15 July – 2024

Day 4

Integration and deployment

CI and CD tool : Continuous Integration and Continuous deployment or delivery

Dev1 pull login page push

Dev2 pull database Remote Repository

shared organization repository. CI and CD -🡪 testing server

Dev3 pull spring boot code production server or other team

Push the code

Manager

Whenever each developer push the code in shared repository then some person need to merge the code and then re-compile and re – run the application ie continuous integration. After merge it may work or may not.

CI and CD tool we need to configure with Shared repository ie Github whenever any developer push the code in remote repository it will pull it and build the project(compile project, run the project, test the project, creating jar or war or build) if build successfully then send this code to another machine or other team etc. if anything go wrong they can send the notification to respective developer.

CD and CD tools

Jenkin

Gitlab

Circle CI

Travis CI

Github action etc

We are going to learn Jenkin CI and CD tools.

Jenkin : Jenkin is a type of CI and CD tool. It is an open source CI and CD tool base upon java technologies. It is a plugin base CI and CD tool. On demand we can configure different type of plugins. It is GUI base CI and CD tools.

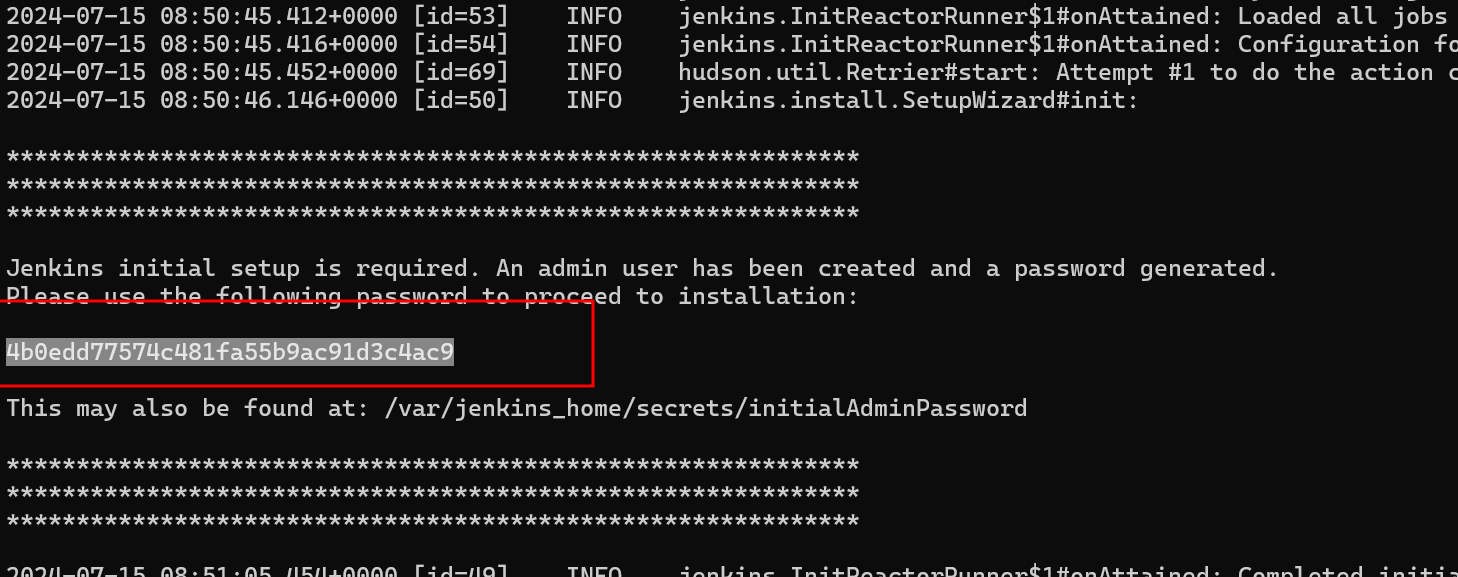


Using Jenkin server using Docker container

docker run -p 8181:8080 -p 50000:50000 --restart=on-failure -v jenkins\_home:/var/jenkins\_home jenkins/jenkins:lts-jdk17

after loaded Jenkin image and run the Jenin on port number 8181

you can verify Jenkin random password



Please open browser and write the URL as

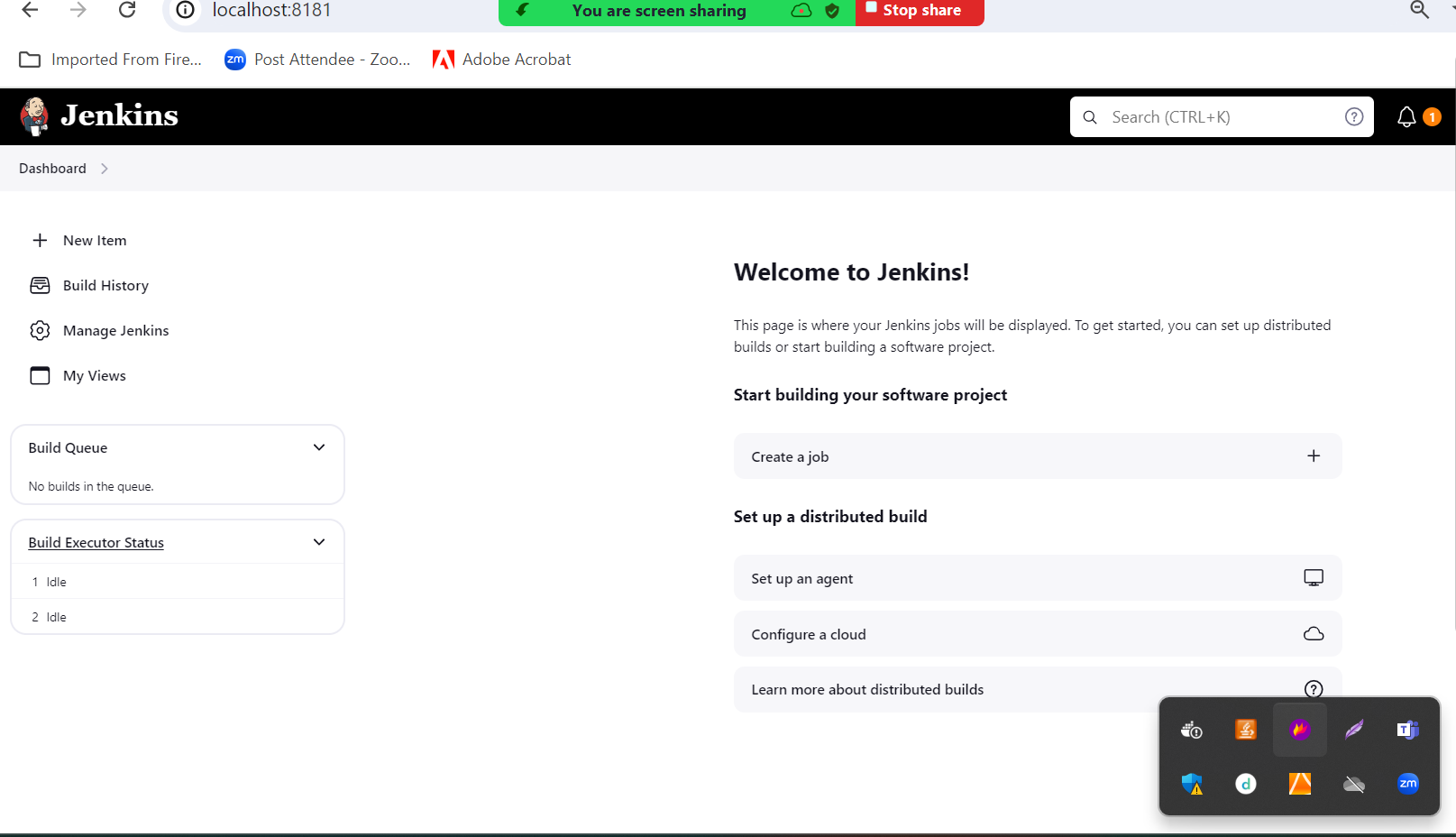
http://localhost:8181

if will ask you password and please verify your password in terminal or console.

Once you click on next button

Please select suggested plugin option to install all required plugin to use Jenkin dashboard.

After installed all plugin we need to create the Account.



Running Jenkin using war

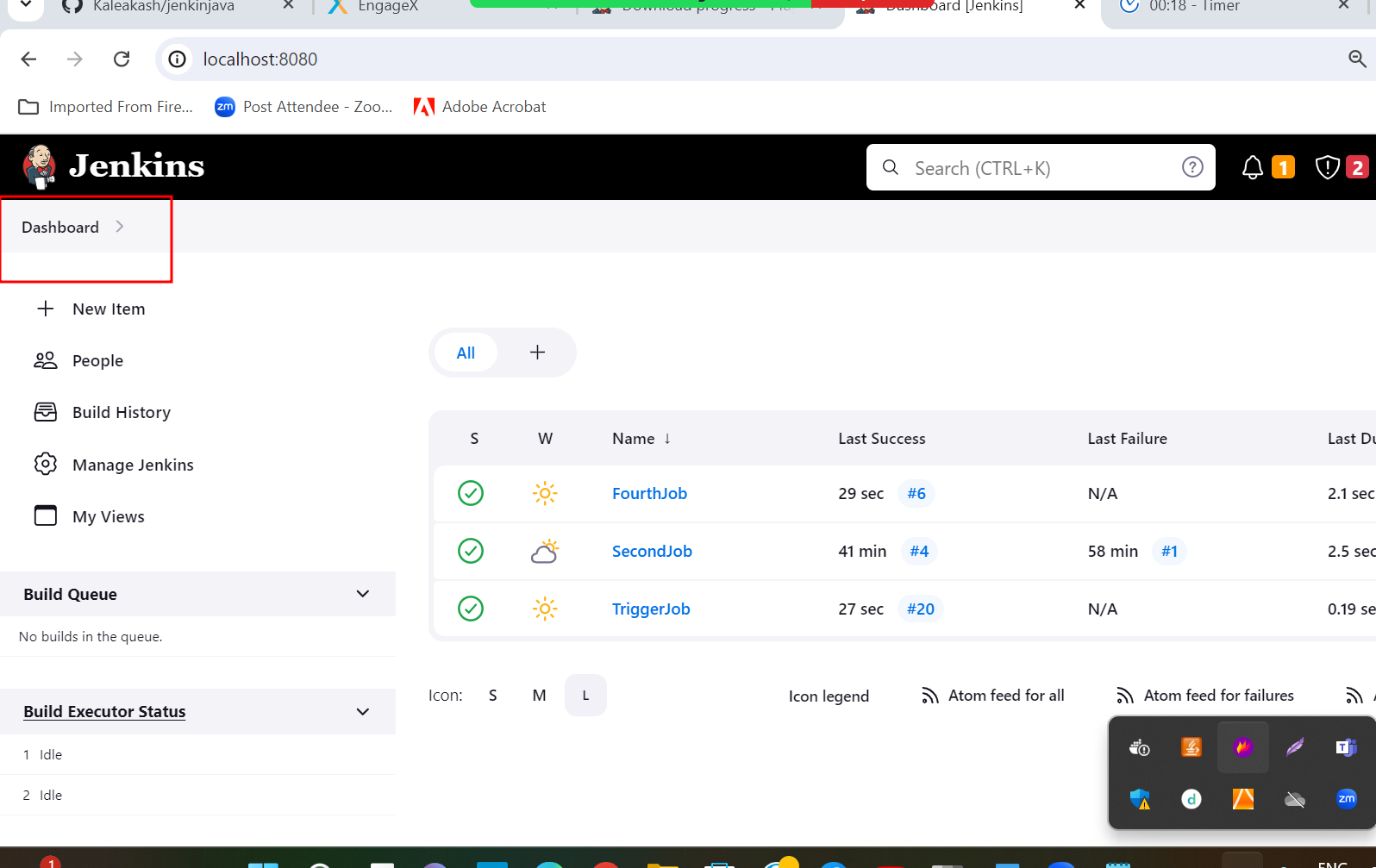
First download war file and open the terminal in that location where war file present and run the below command as

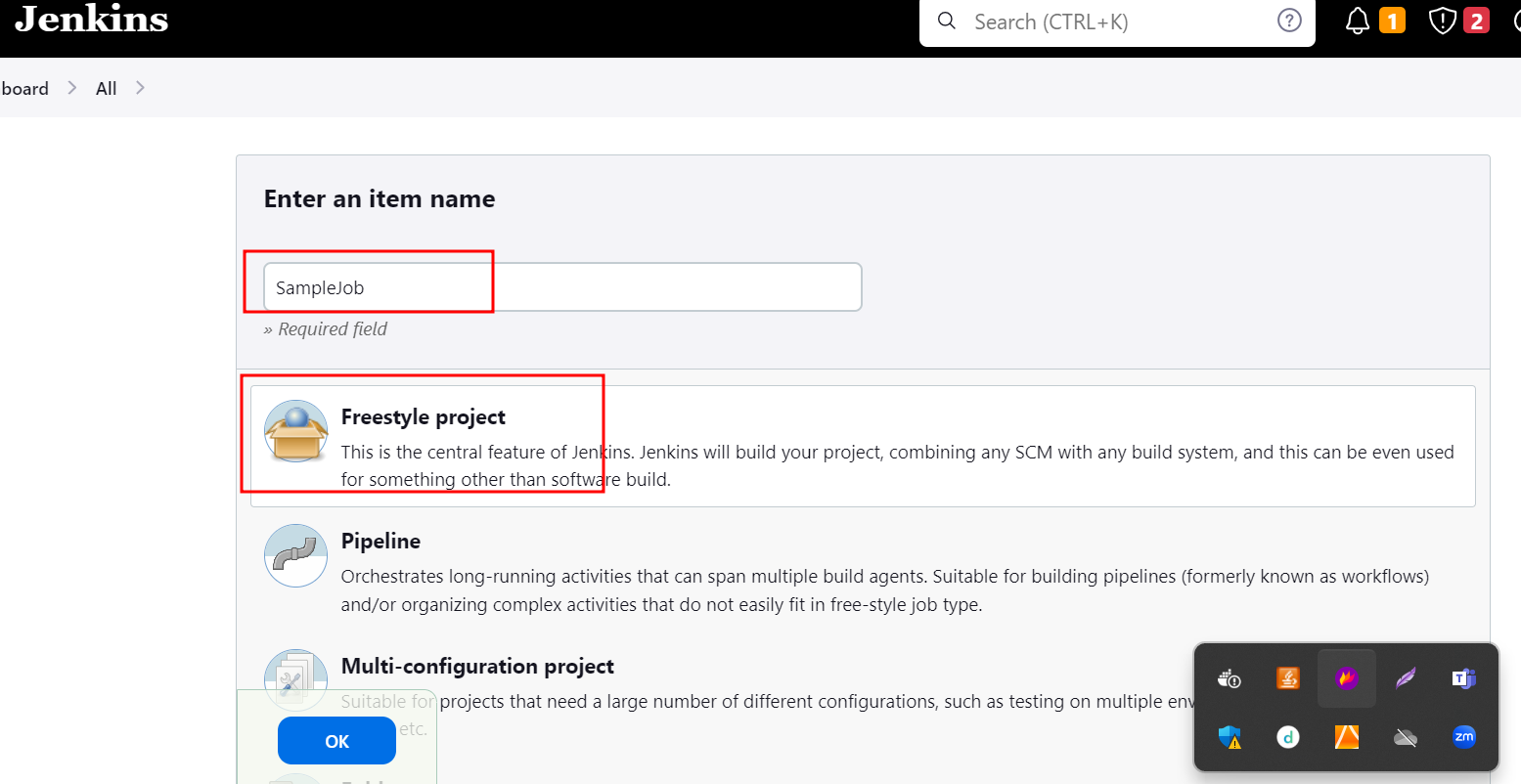
java version check the version of java

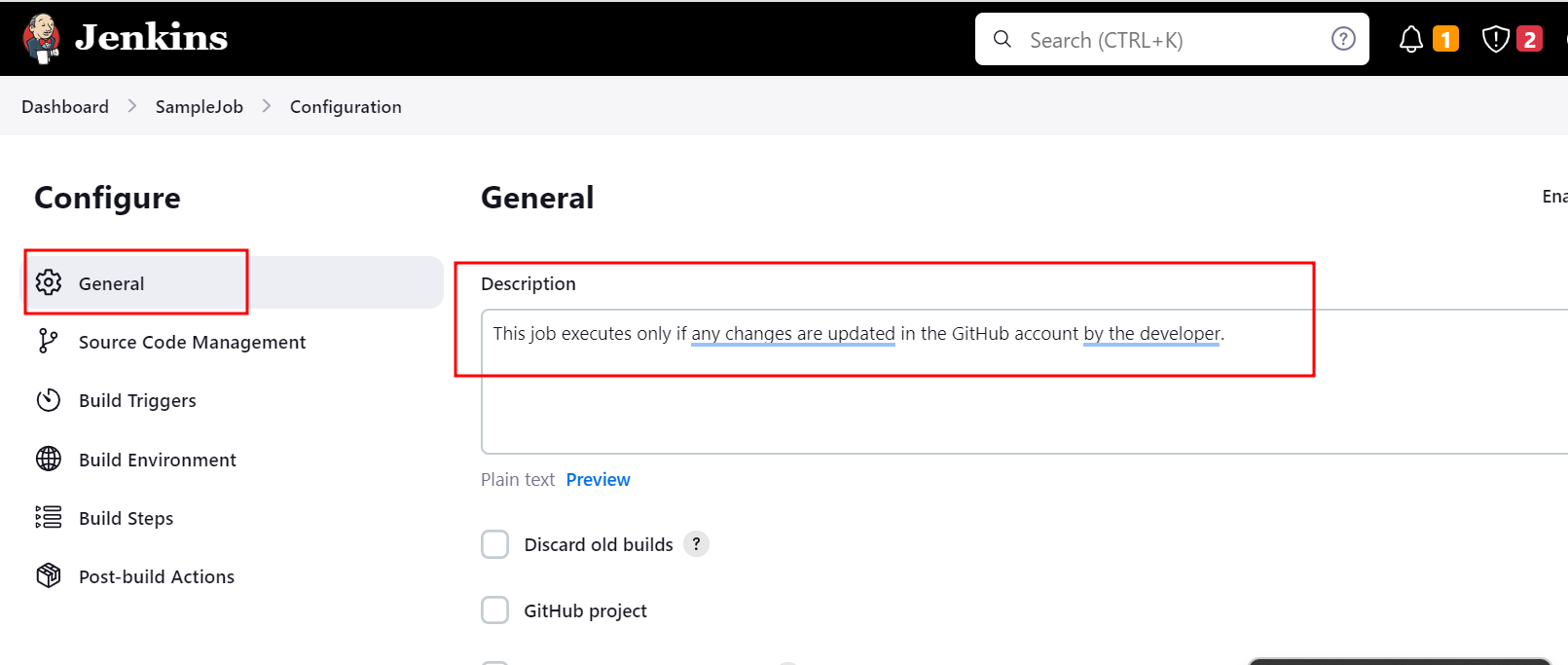
java -jar jenkins.war

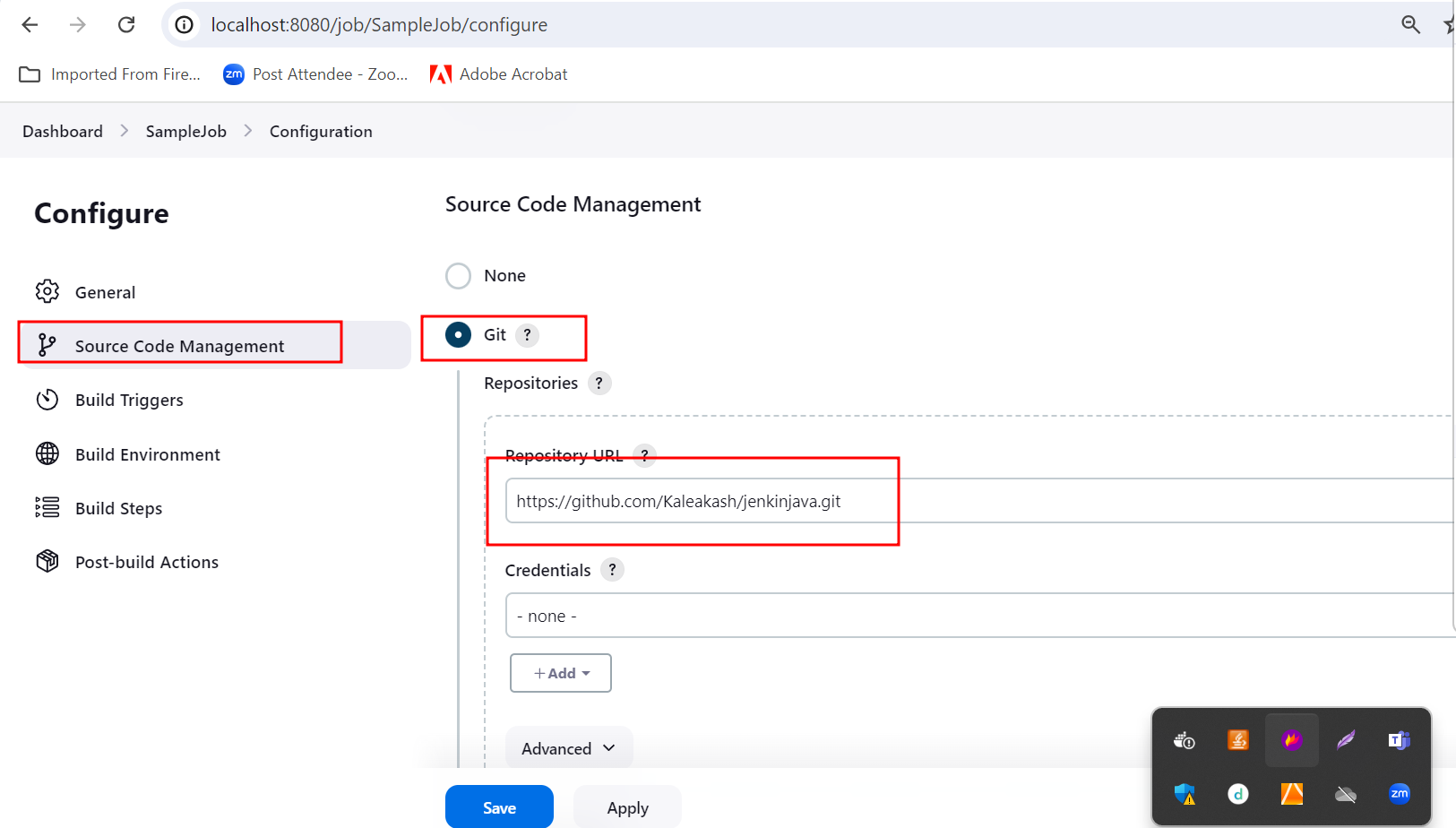
1. Creating simple job to display message.
2. Second job responsible to pull the sample project from Git Repository and build it.
3. Creating Jenkin job responsible to do trigger(this job executed base upon time or conditions).
4. Creating Jenkin job responsible to pull the project from github account and build the job after every 1 min with trigger options.
5. We need to build the project (Compiling and Running Core Java Program ) whenever owner push the new changes in Github account.

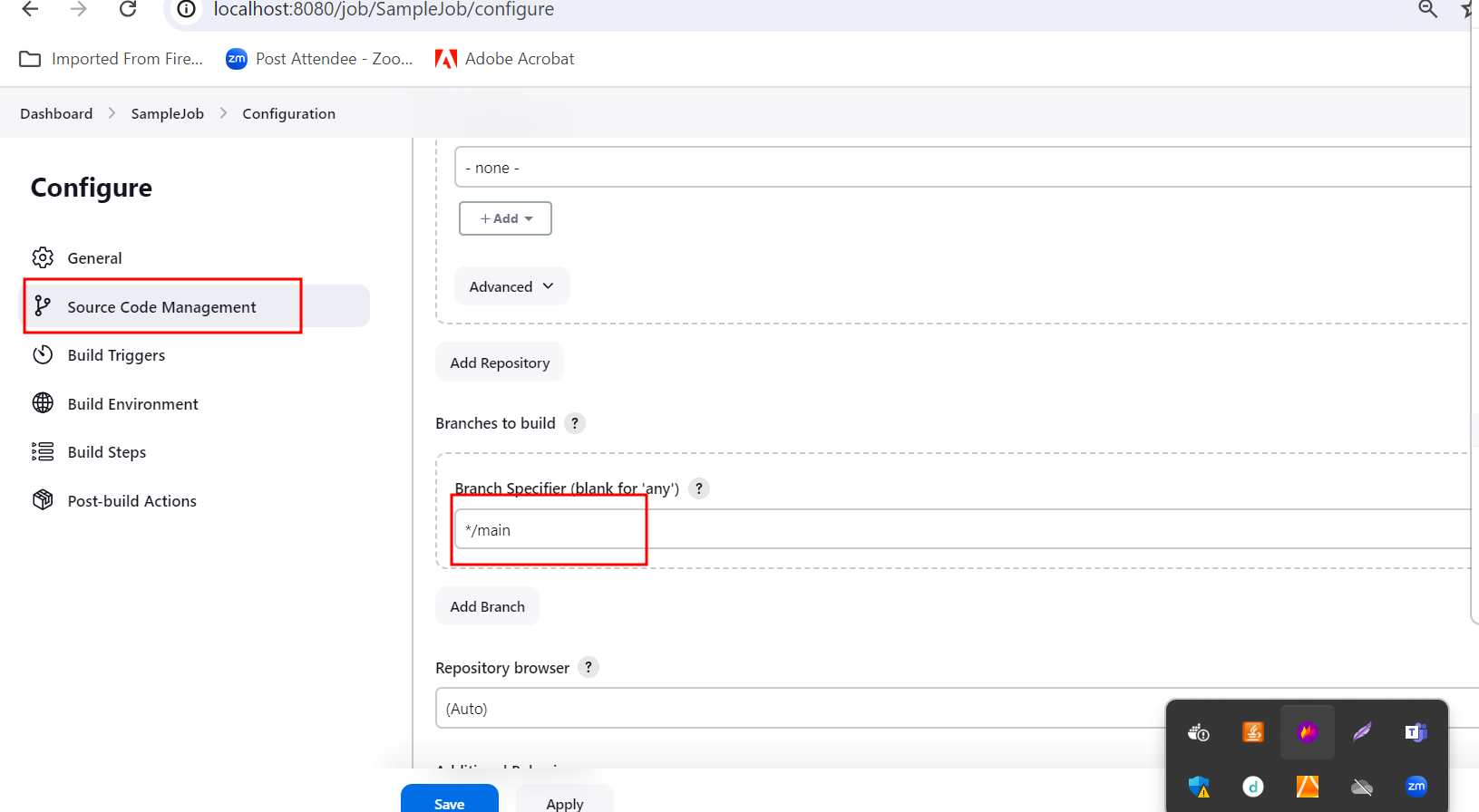
Steps

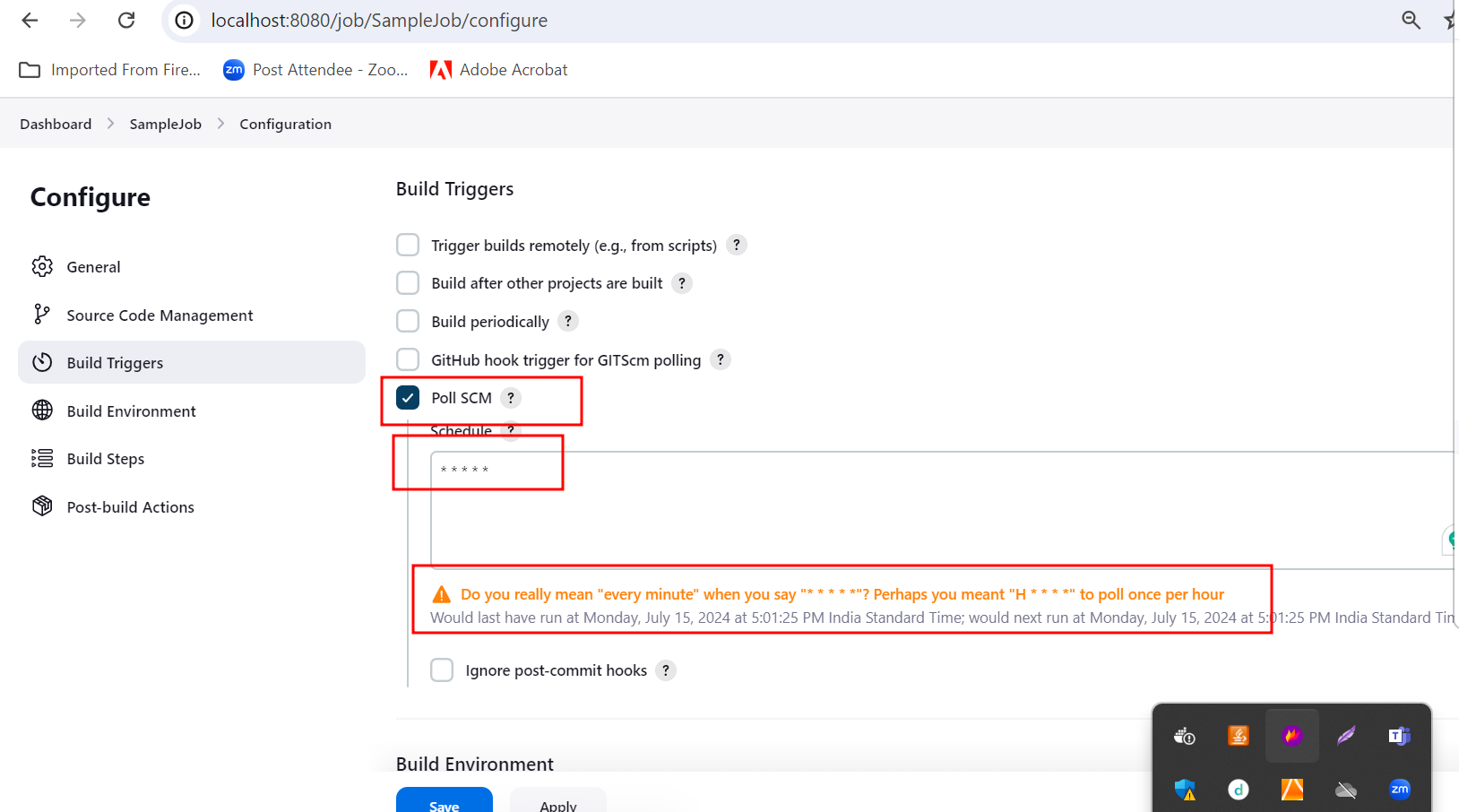




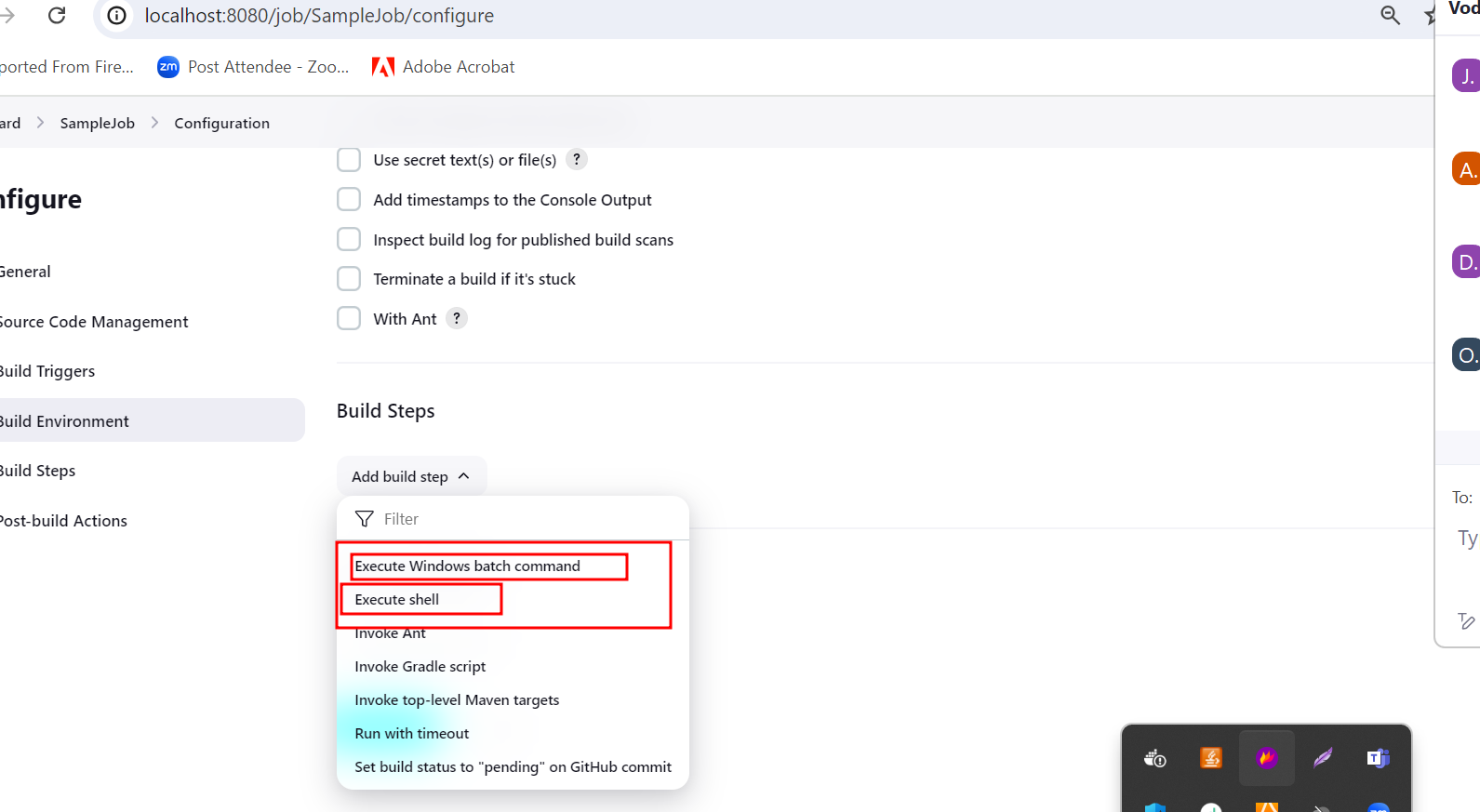






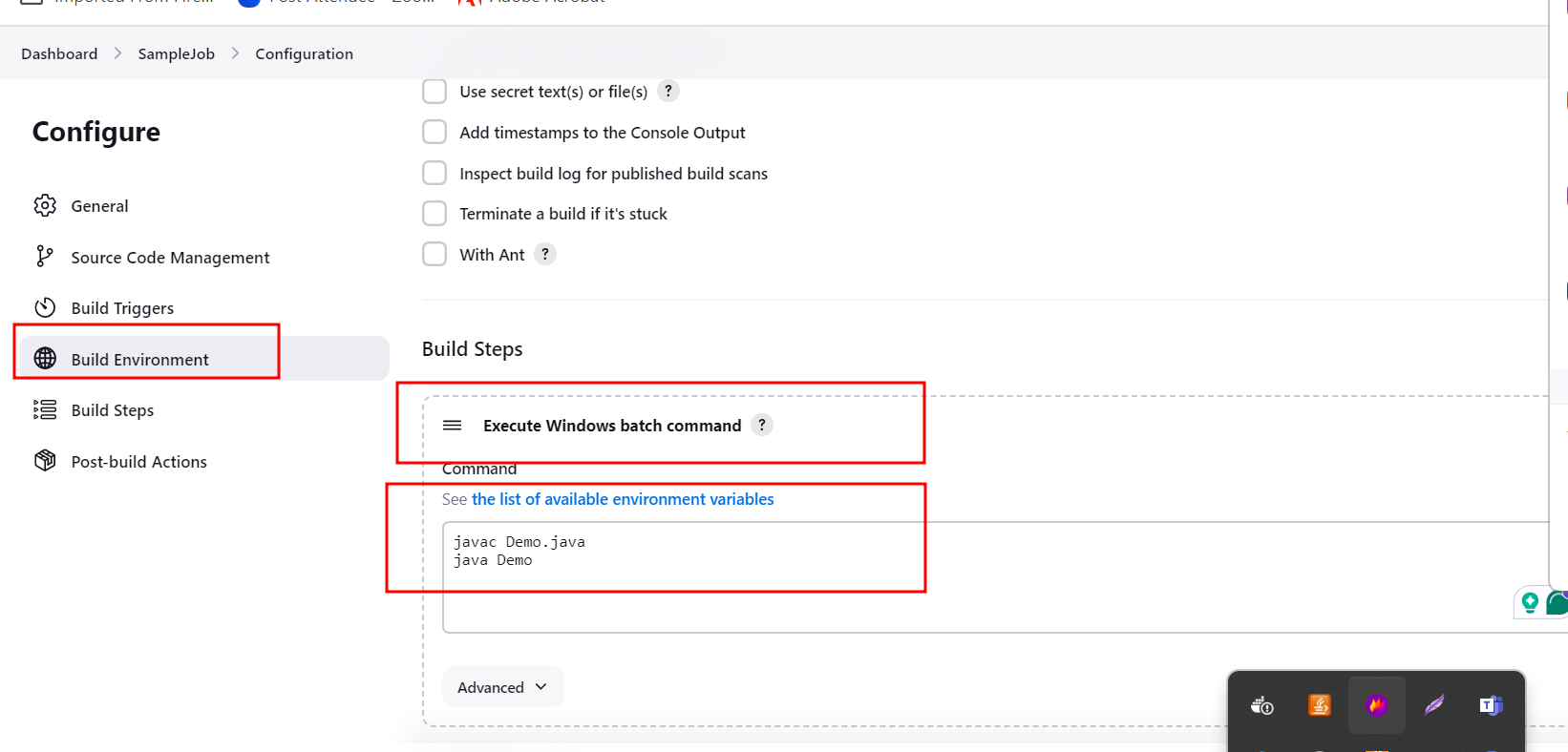


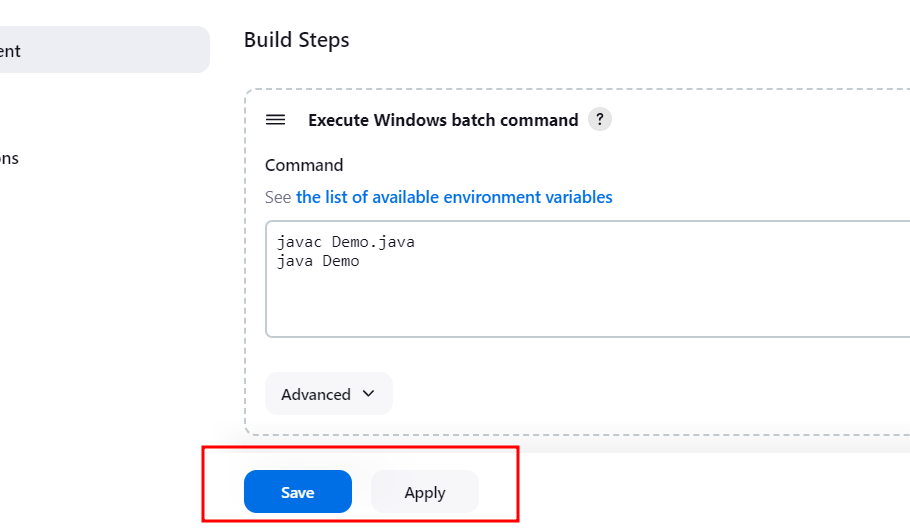
Poll SCM : Trigger option : it check very minute in Github account. If any changes happen then only it build the project.



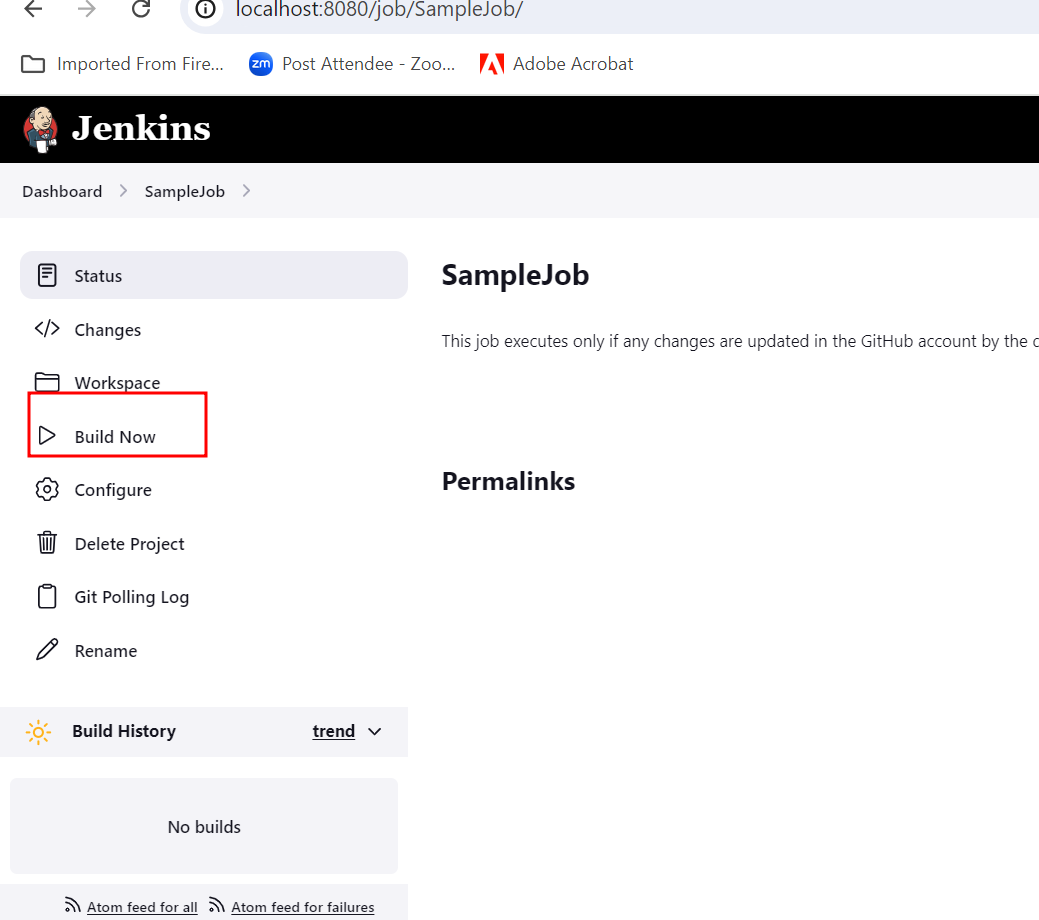
Window user 🡪 Execute window batch command

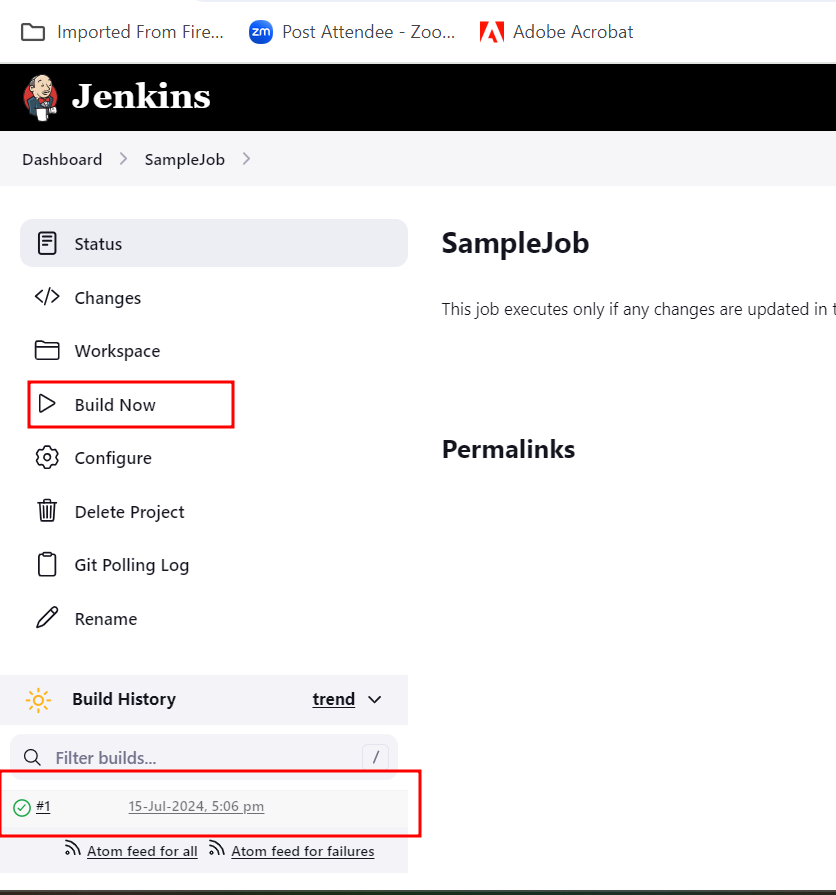
Non window user 🡪 execute shell



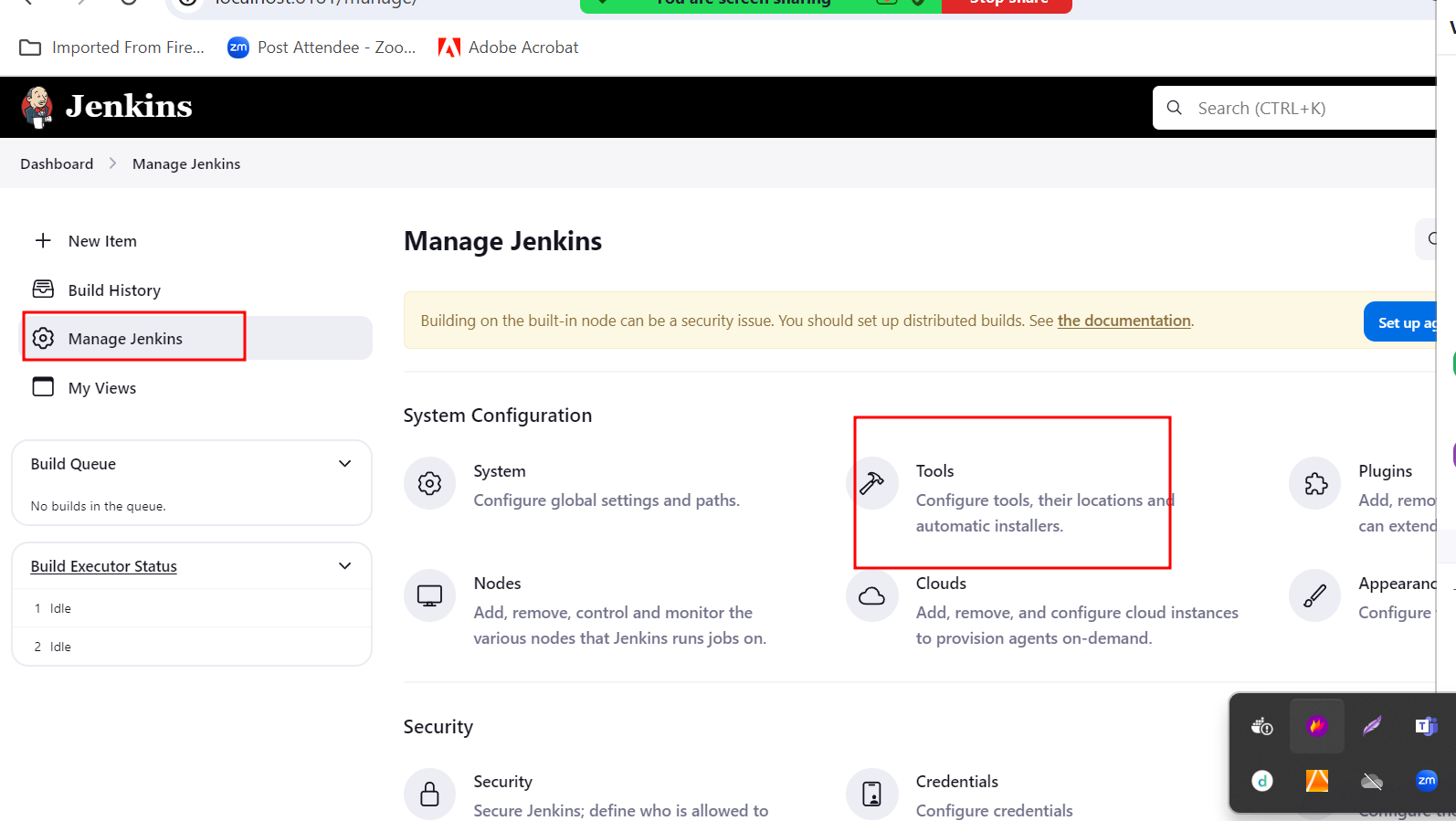


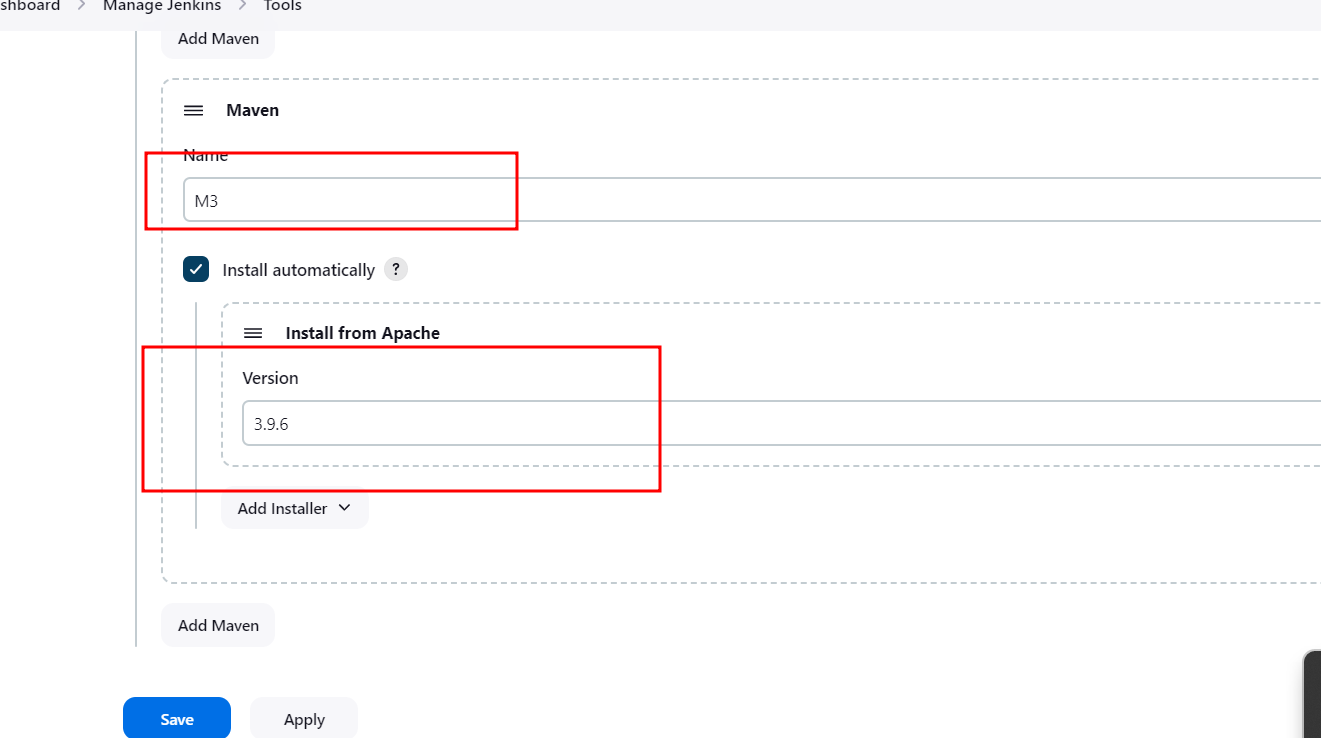
Click apply and save





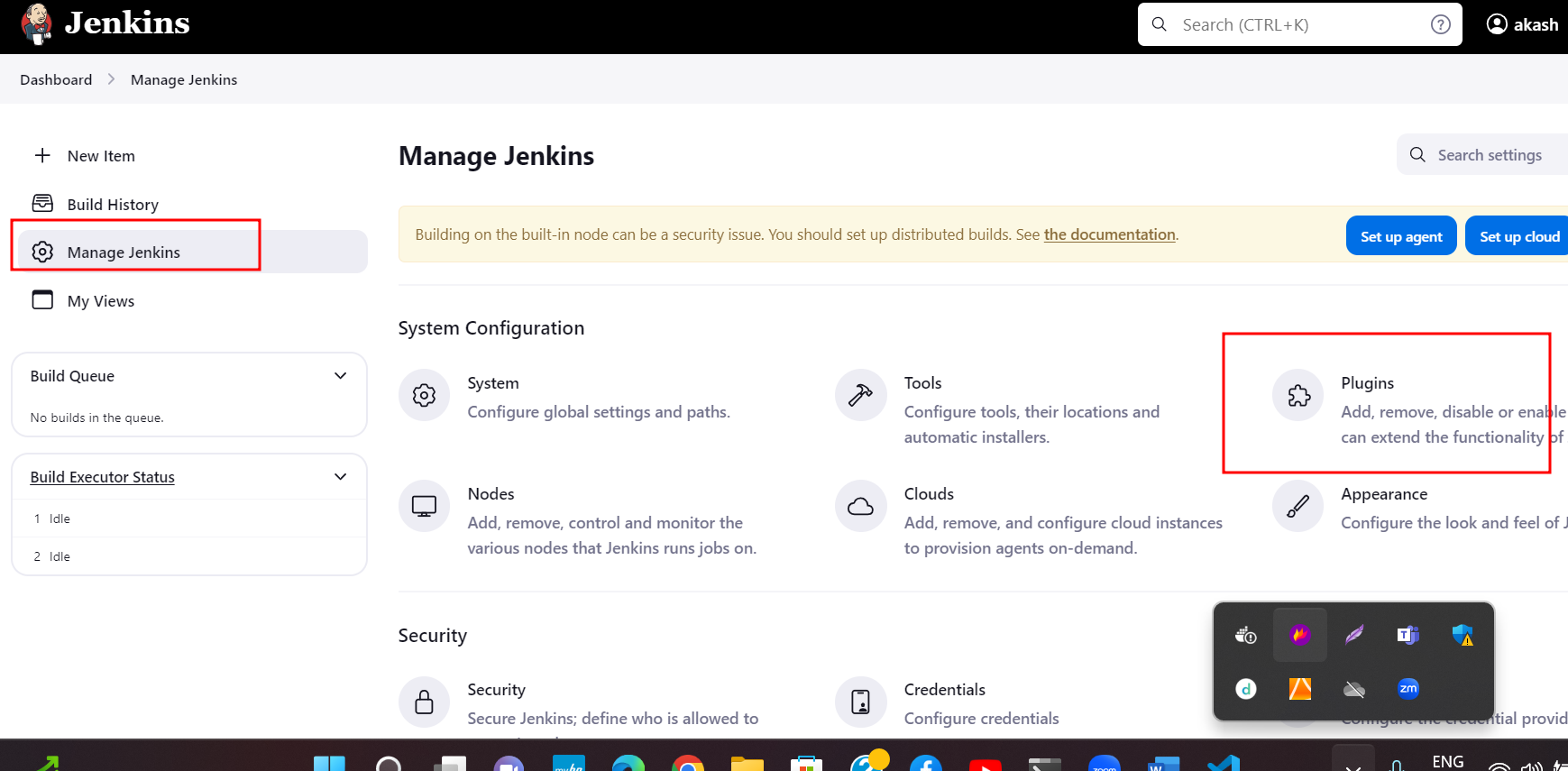
Steps to configure Maven software in Jenkin Dashboard if Jenkin running through docker.

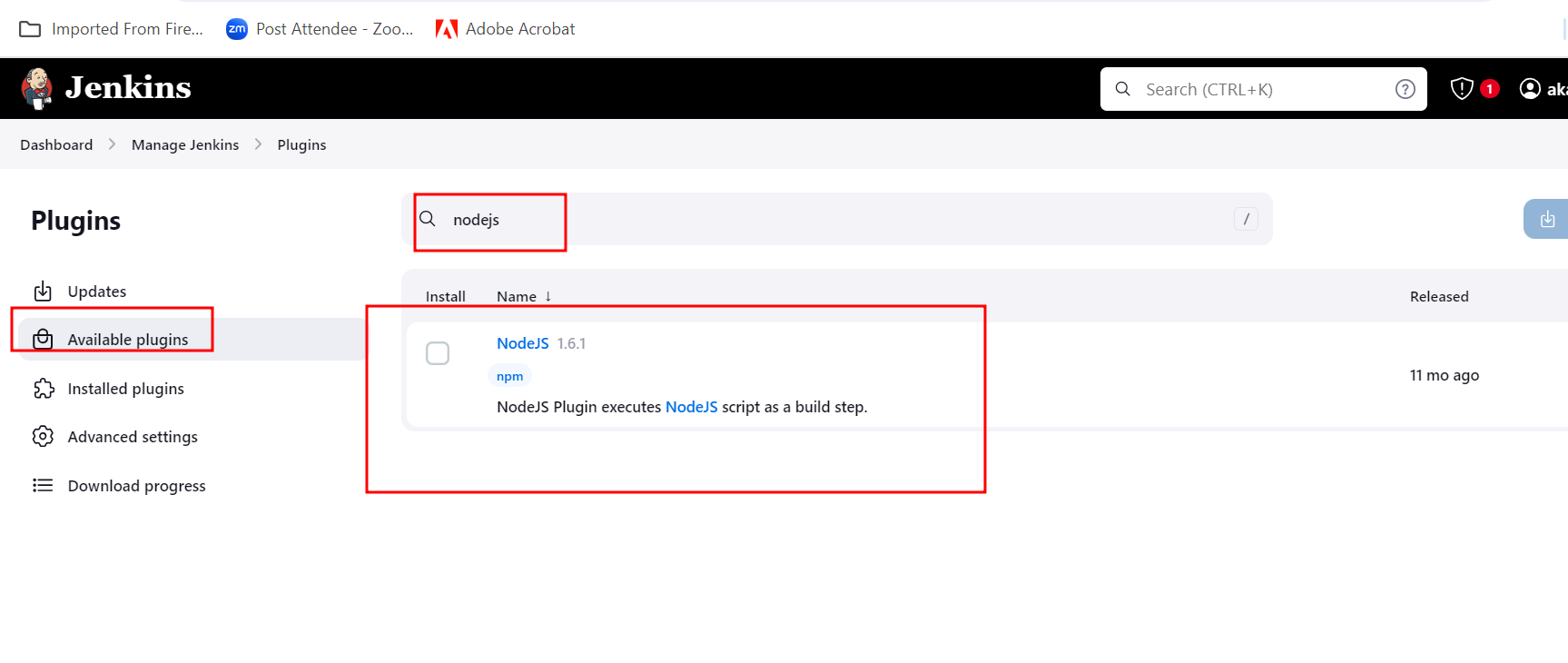


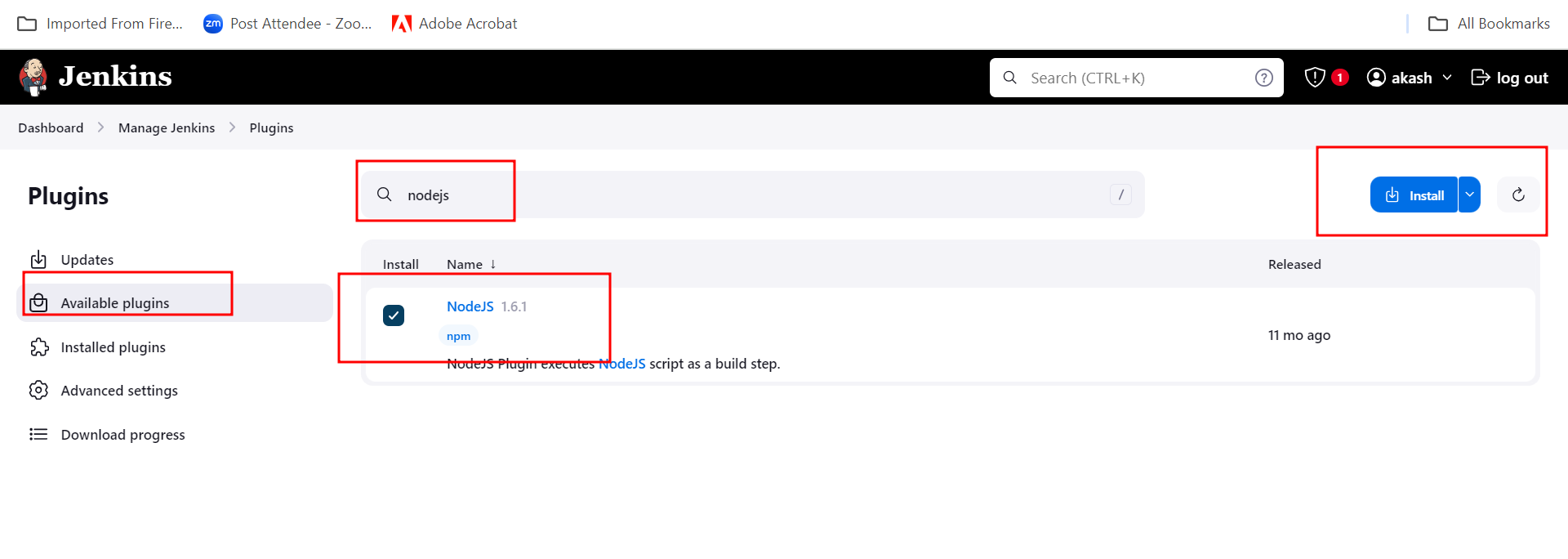


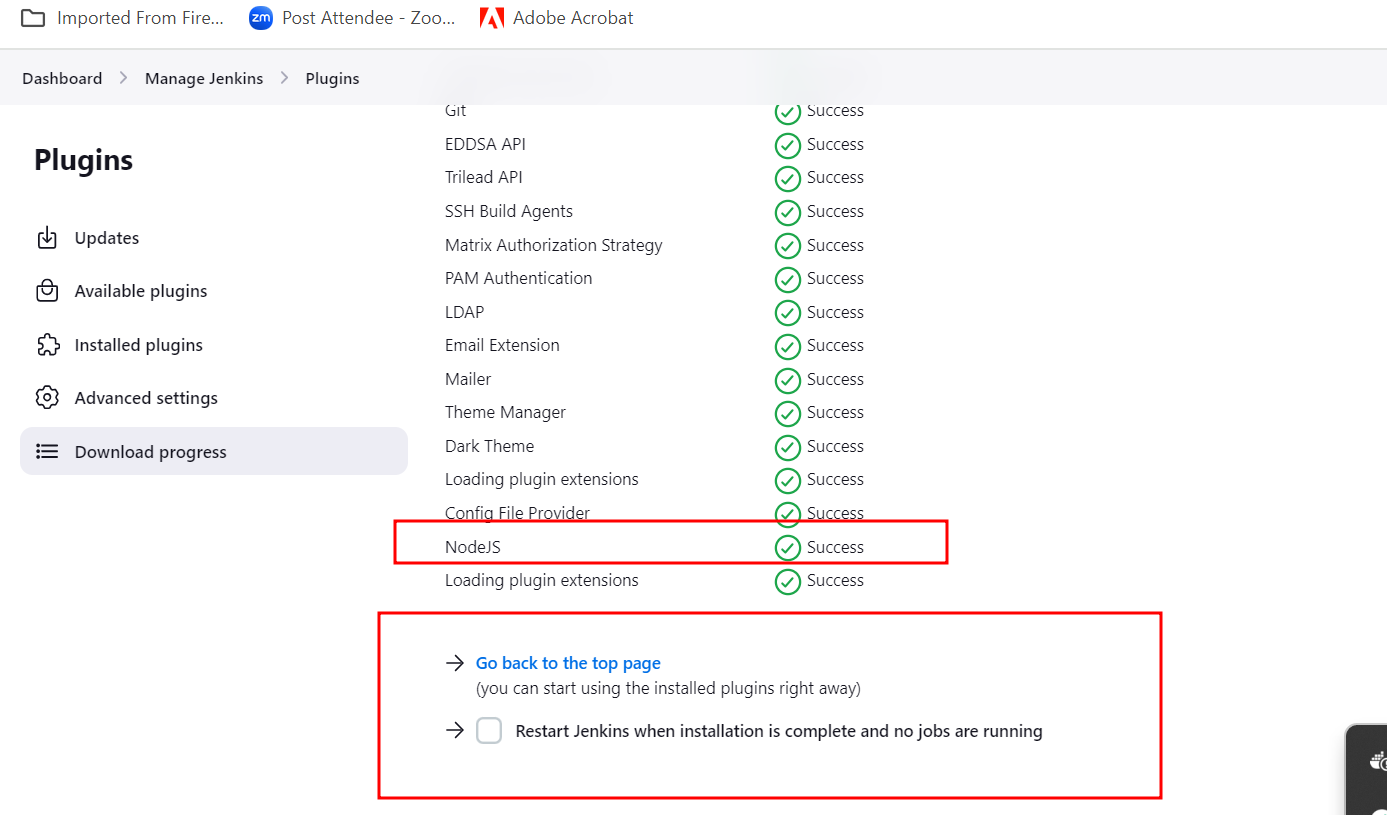
By default Jenkin provide only Maven, Gradle, Ant and Java Plugin

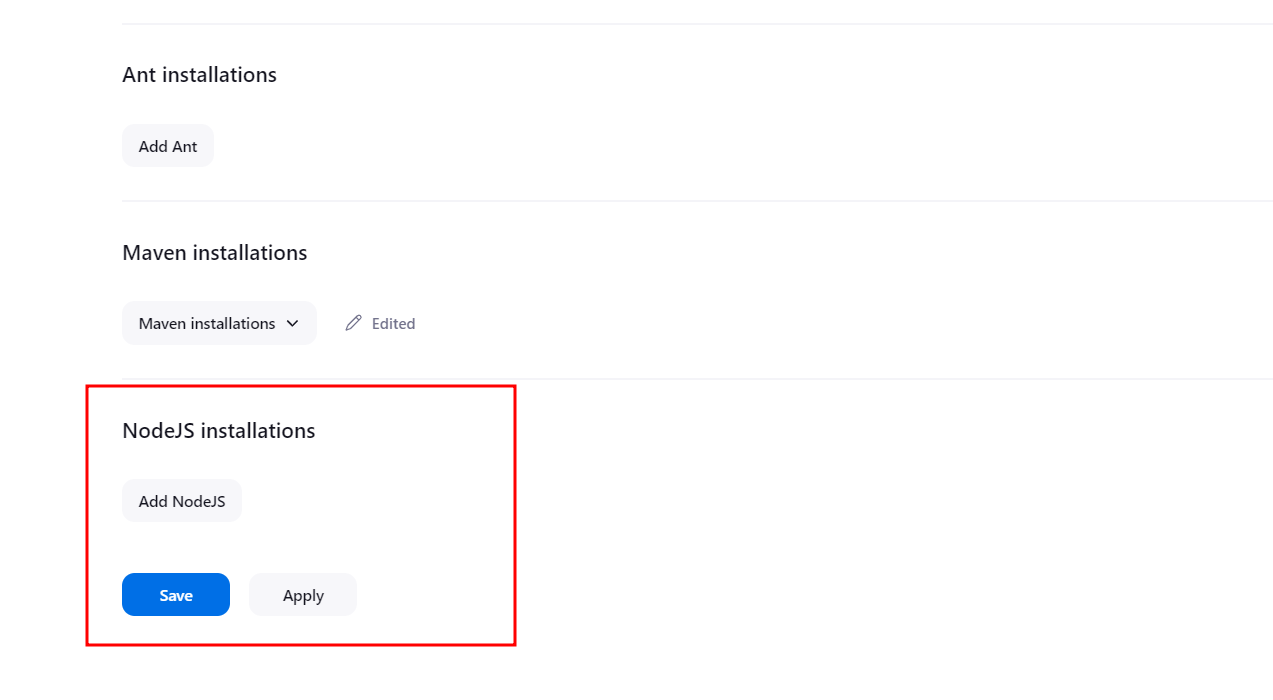
Adding Node JS Plugin

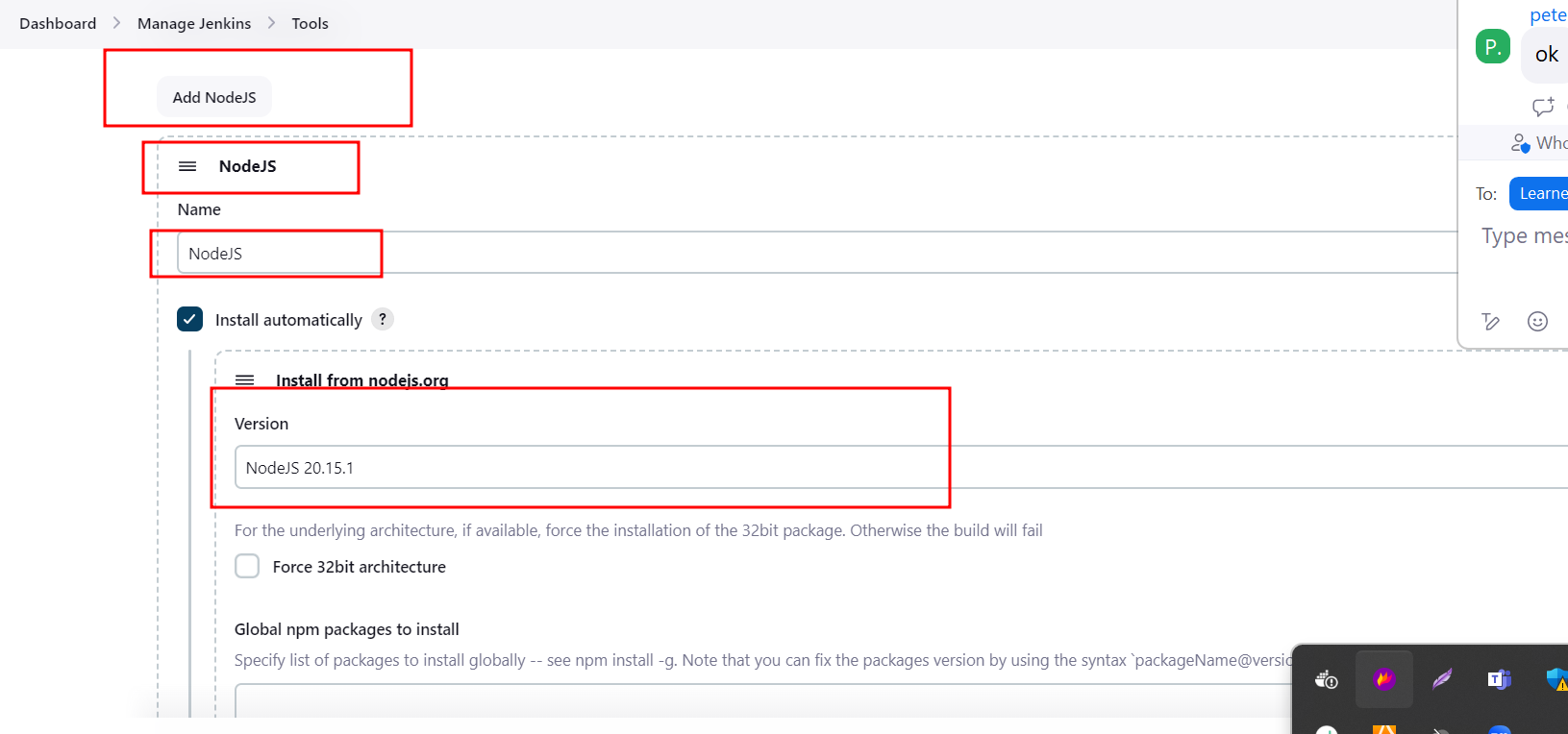








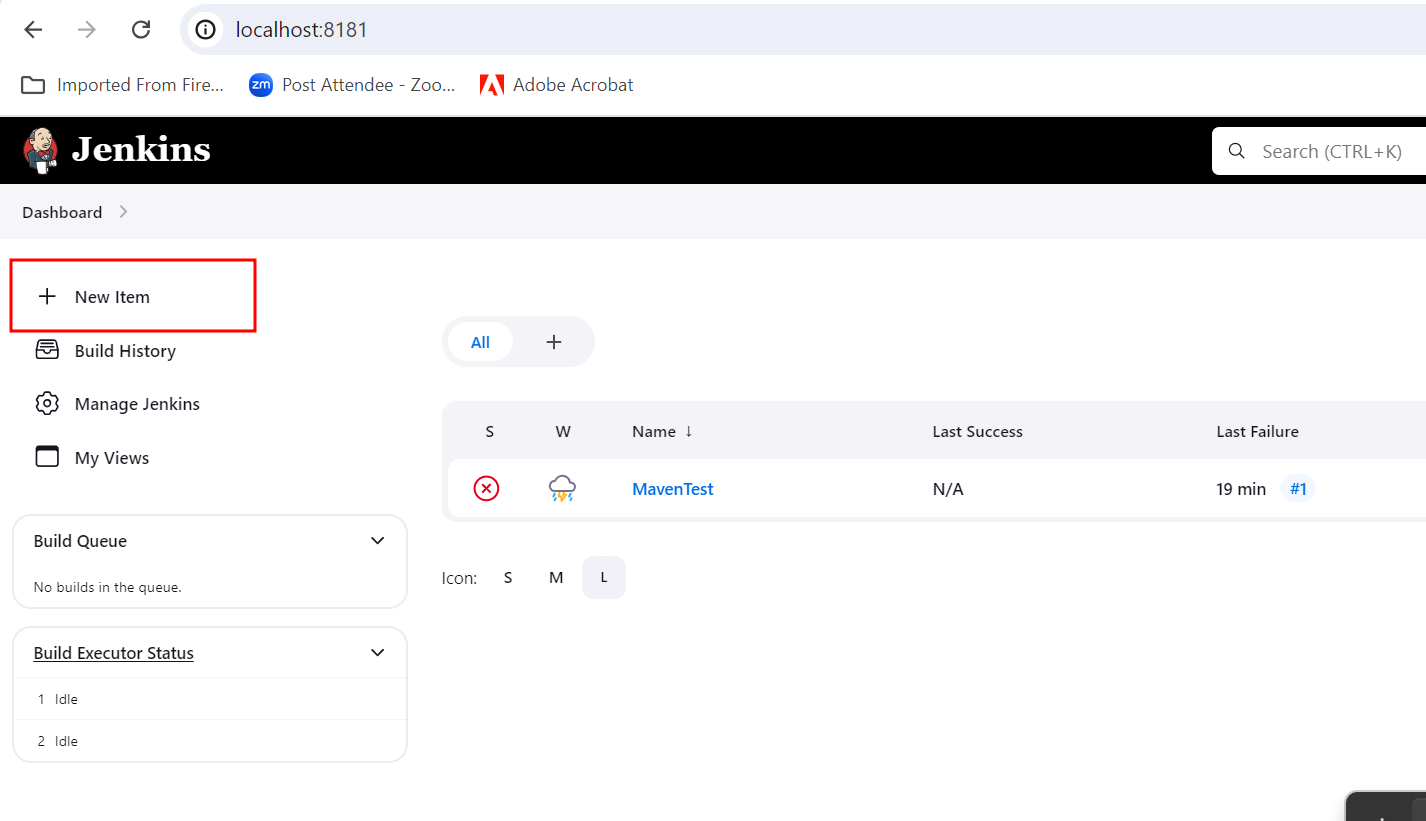


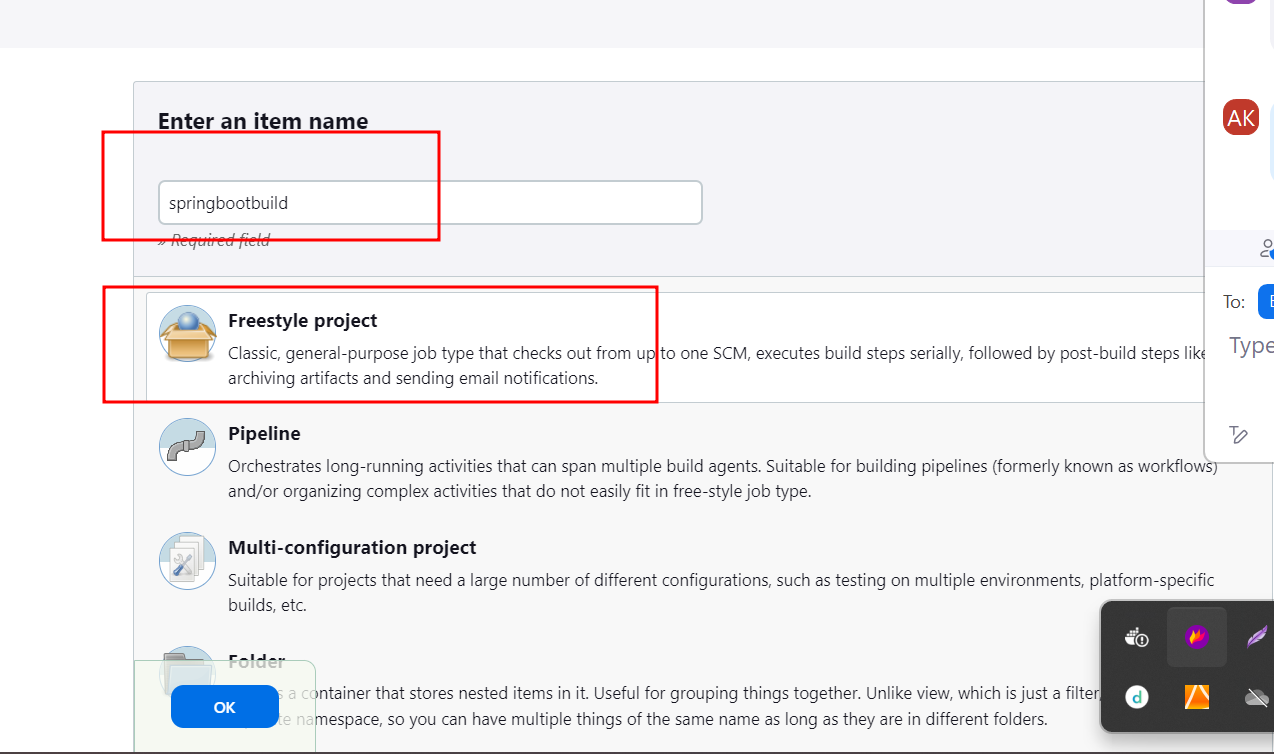


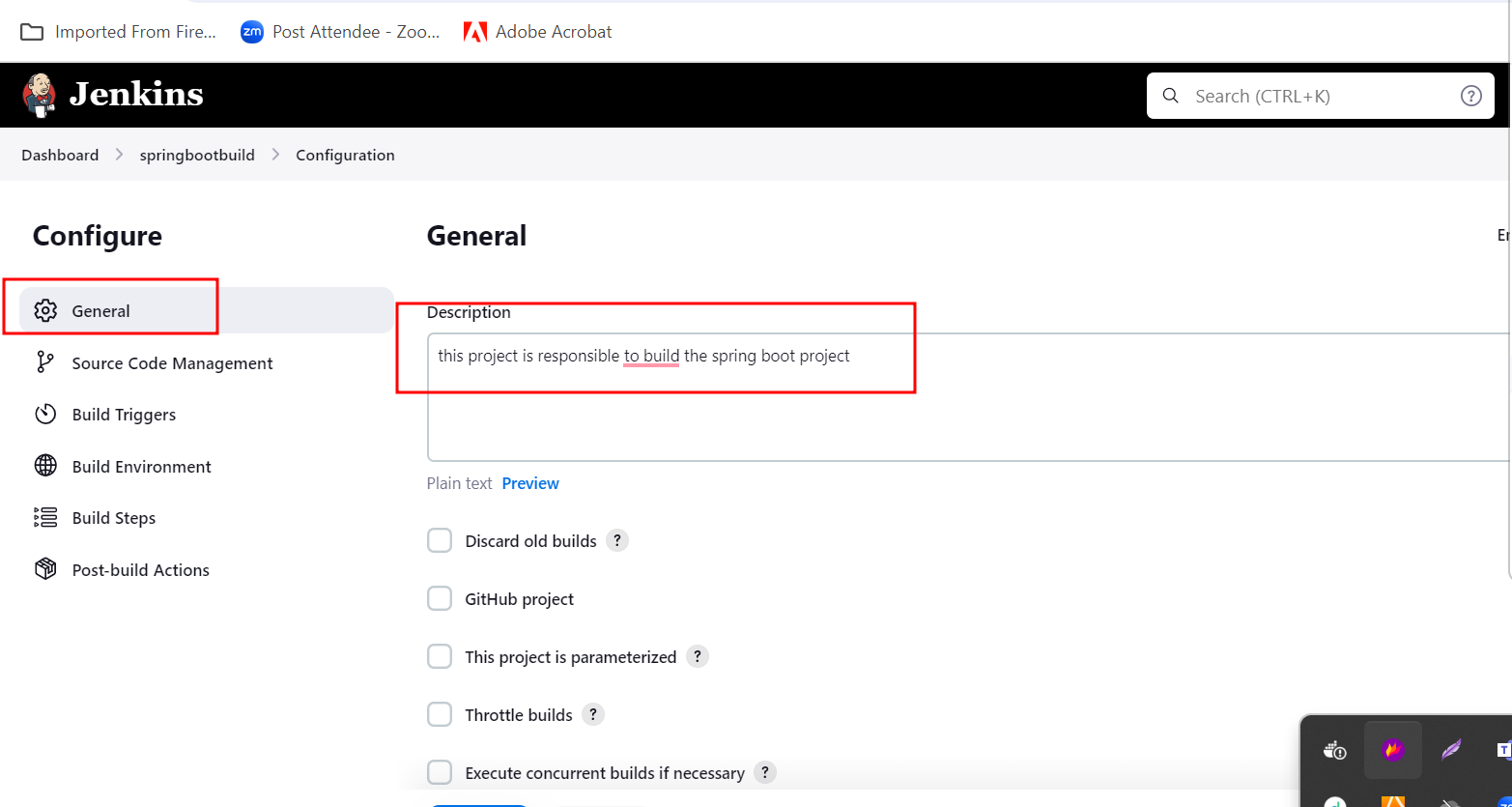
<https://github.com/Kaleakash/springmavenwithjenkin.git>

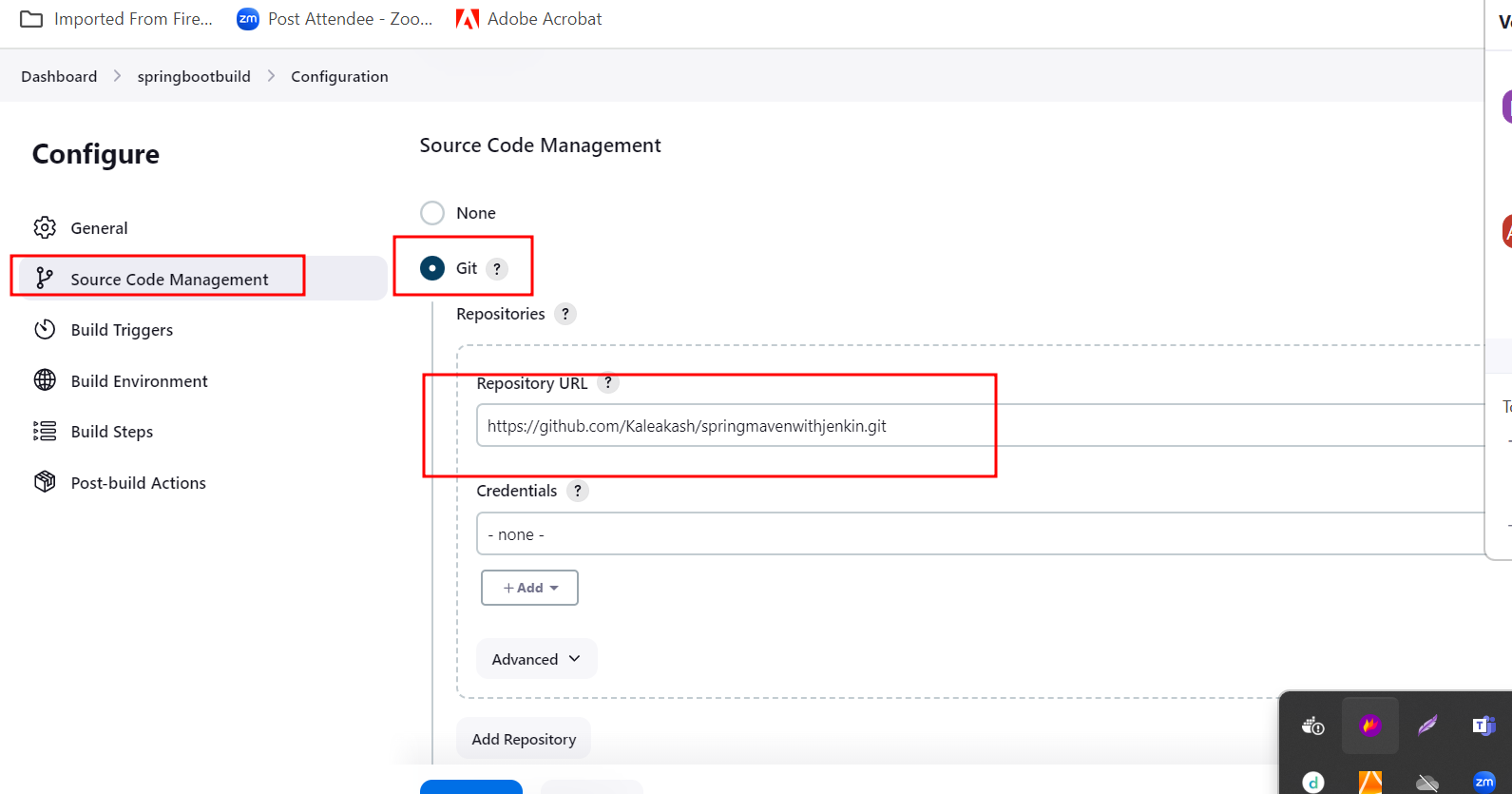
this git hub contains simple spring boot project with simple controller.

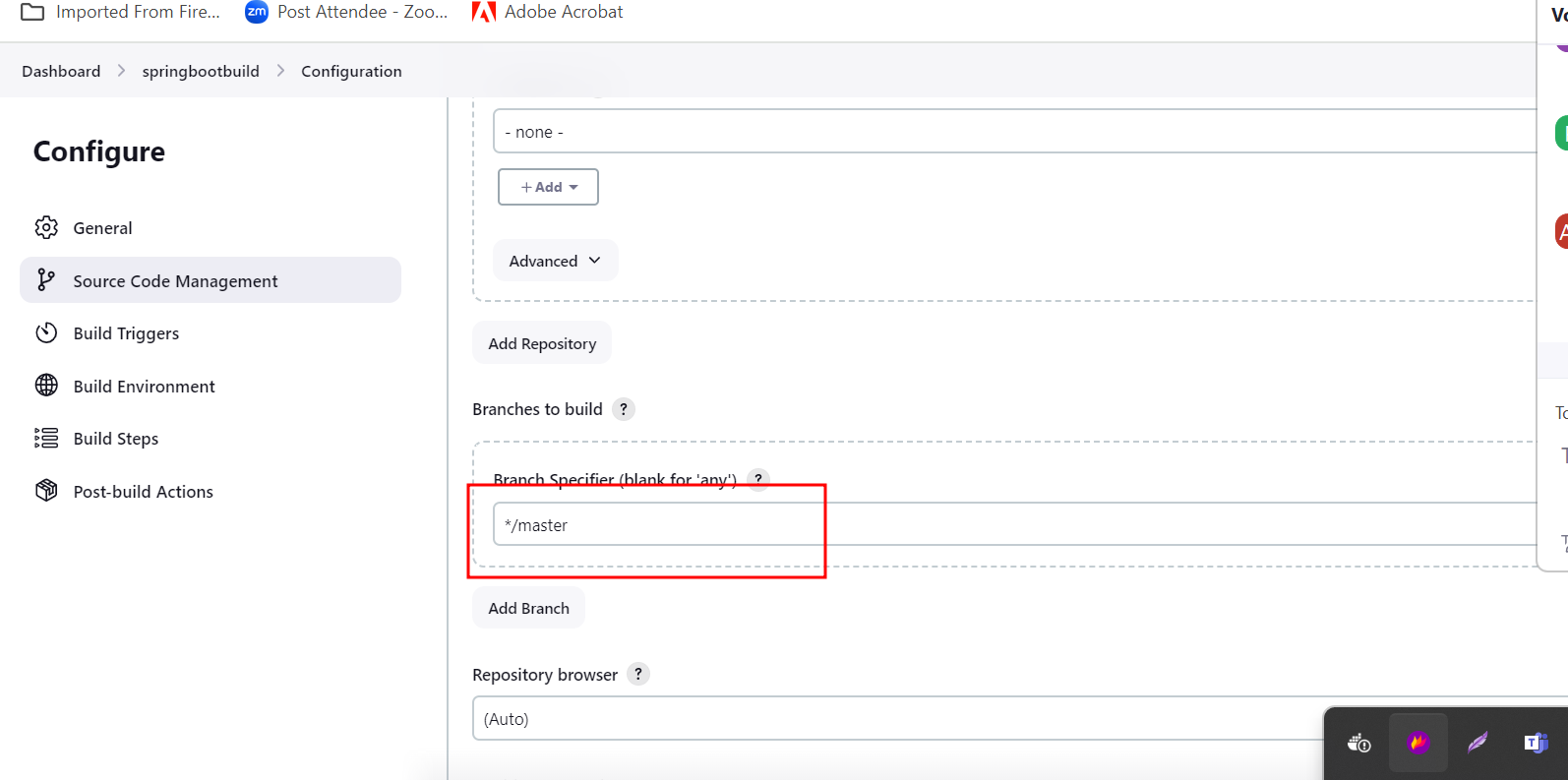
Creating job which is responsible to build spring boot project in Jenkin Environment.

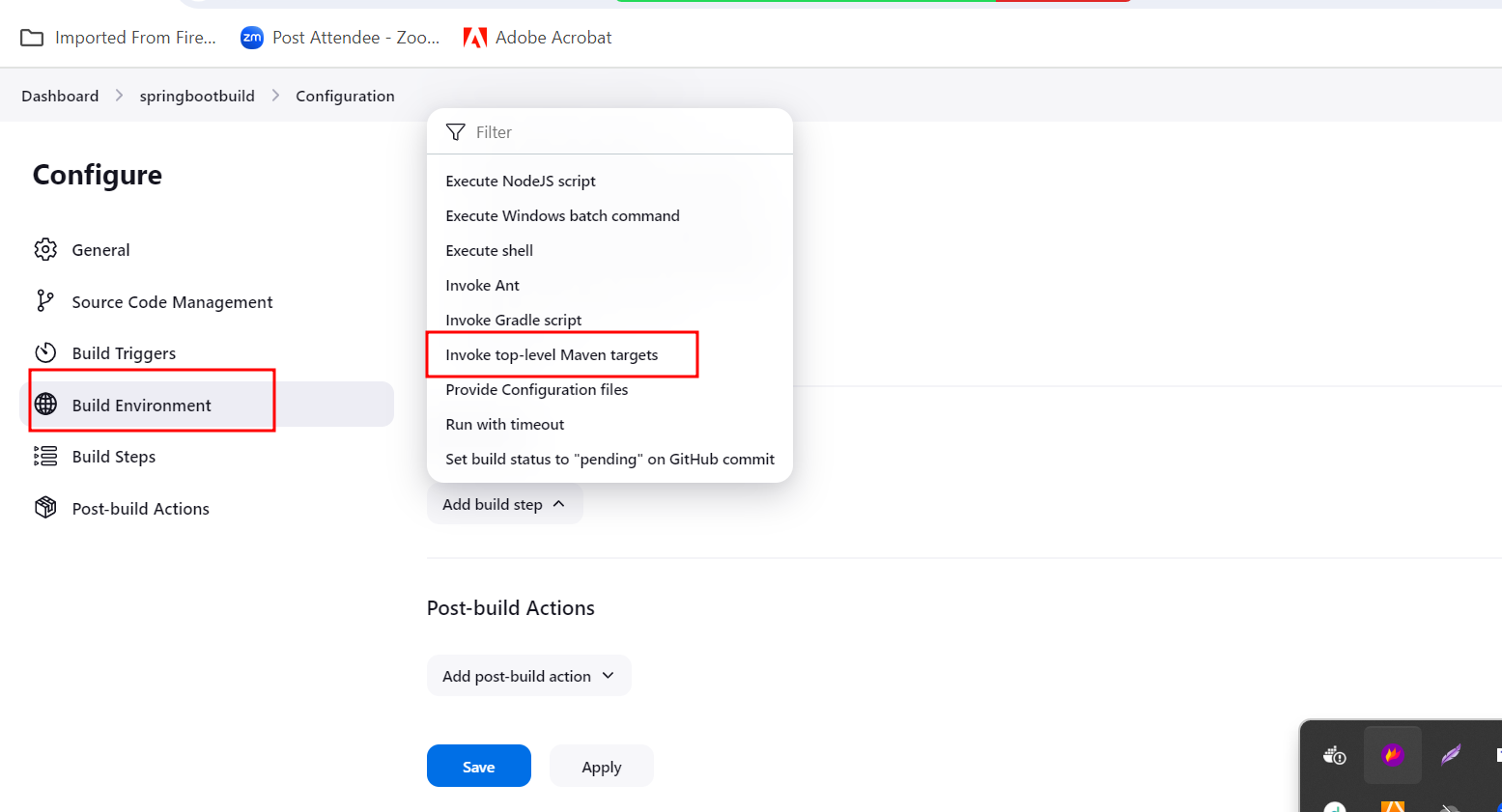


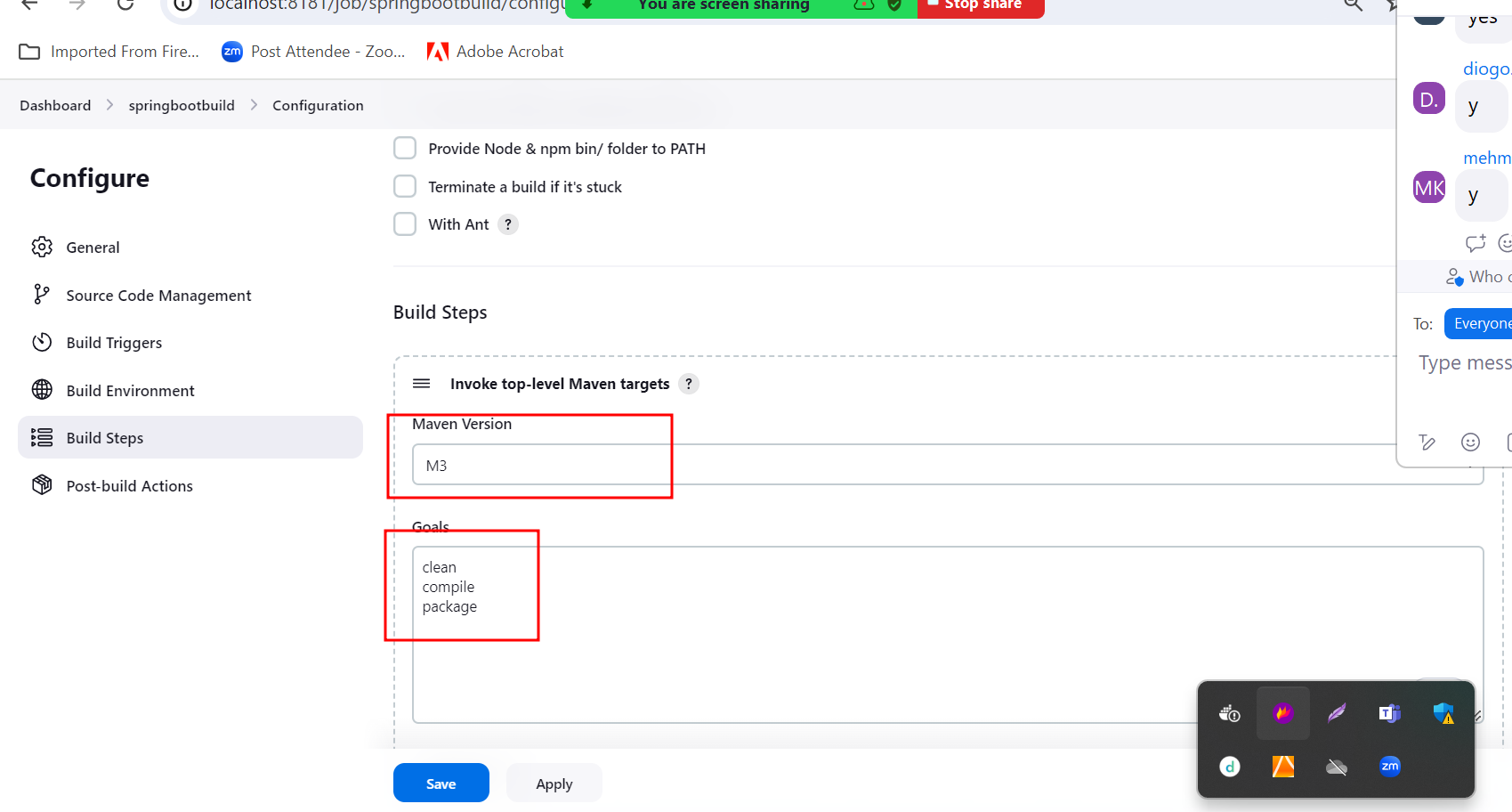


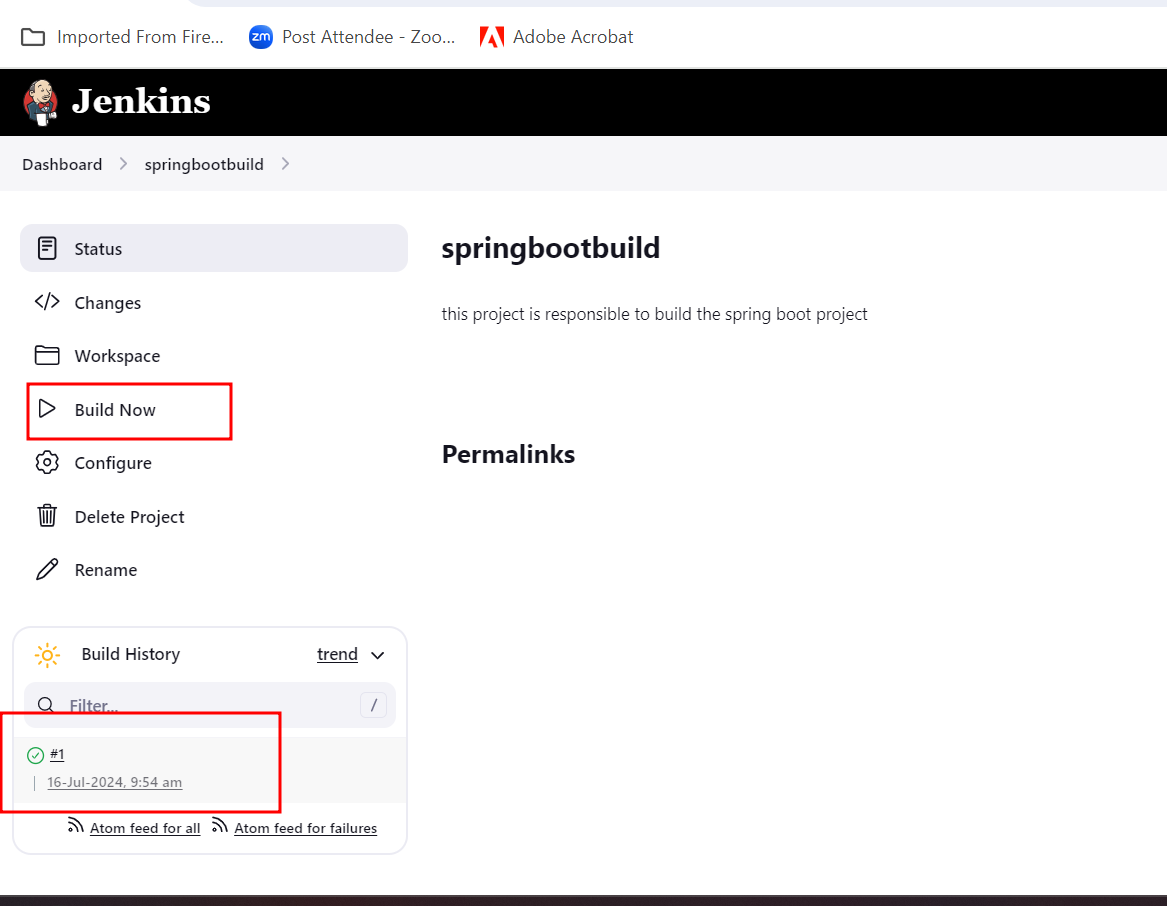


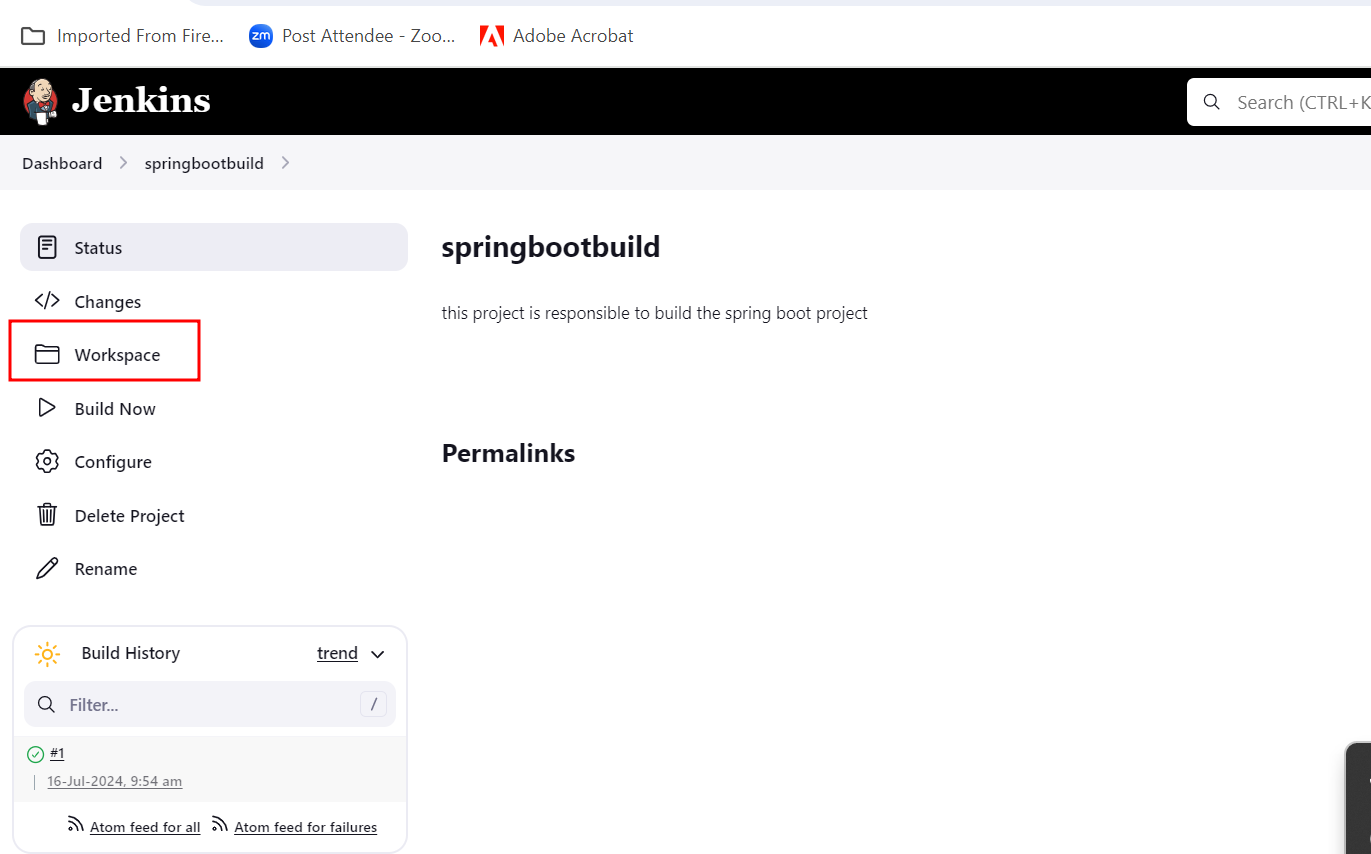


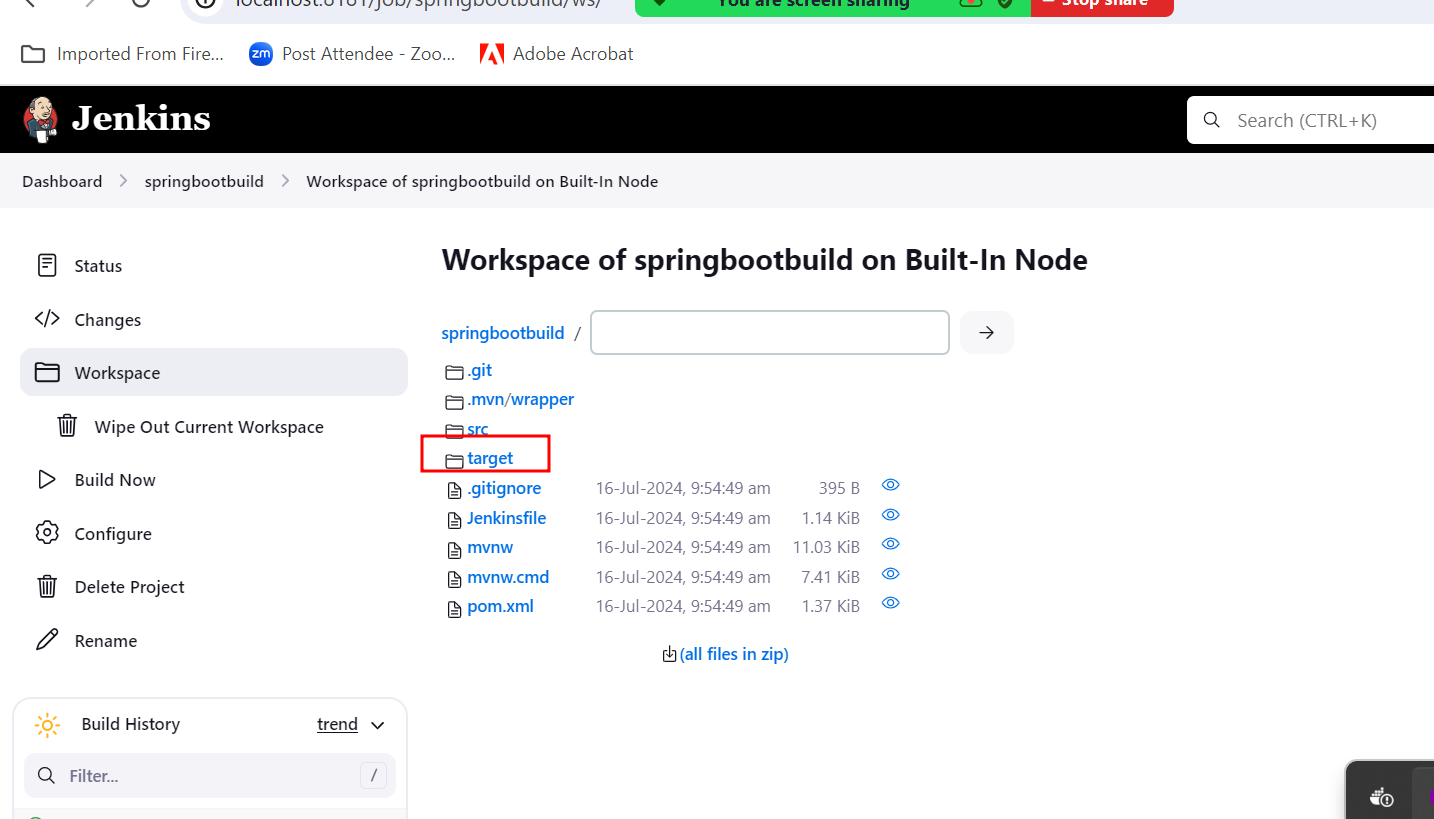


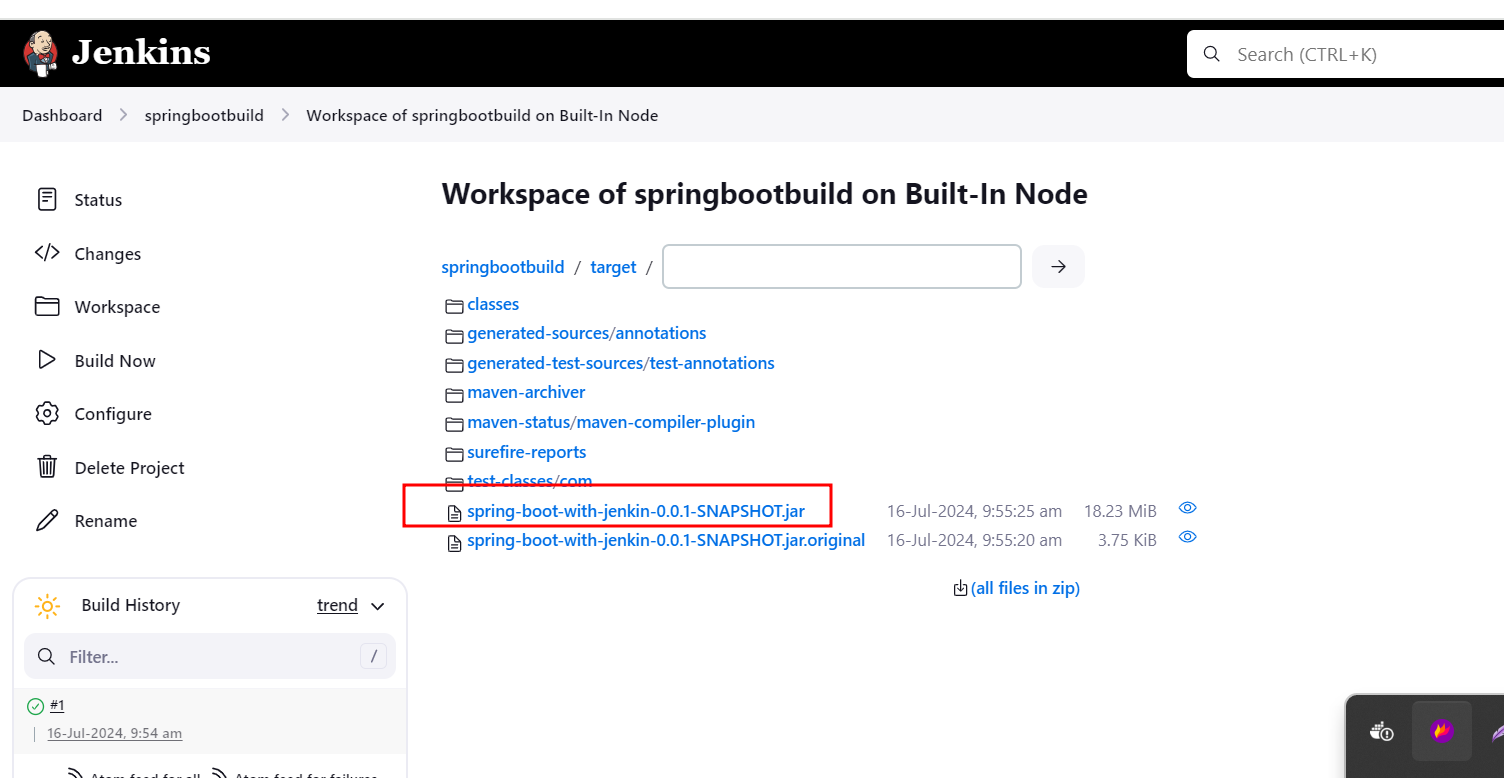












We successfully build spring boot project in Jenkin