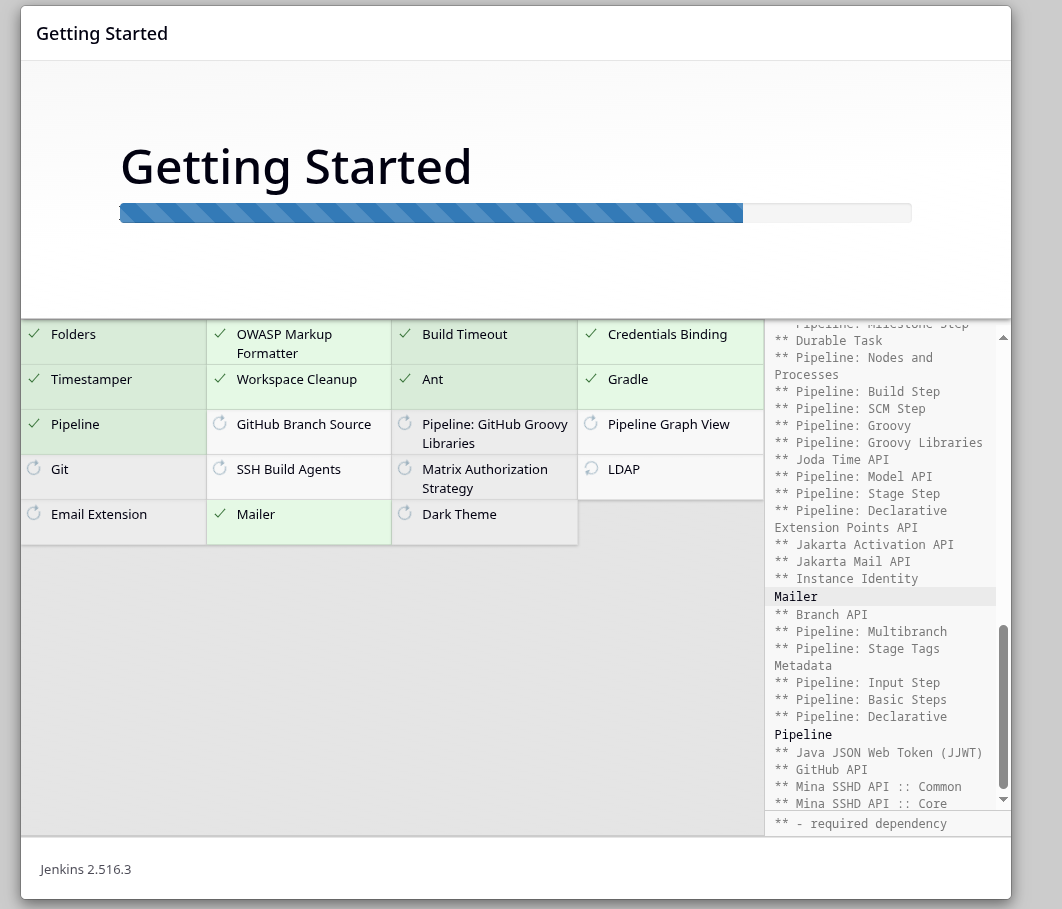
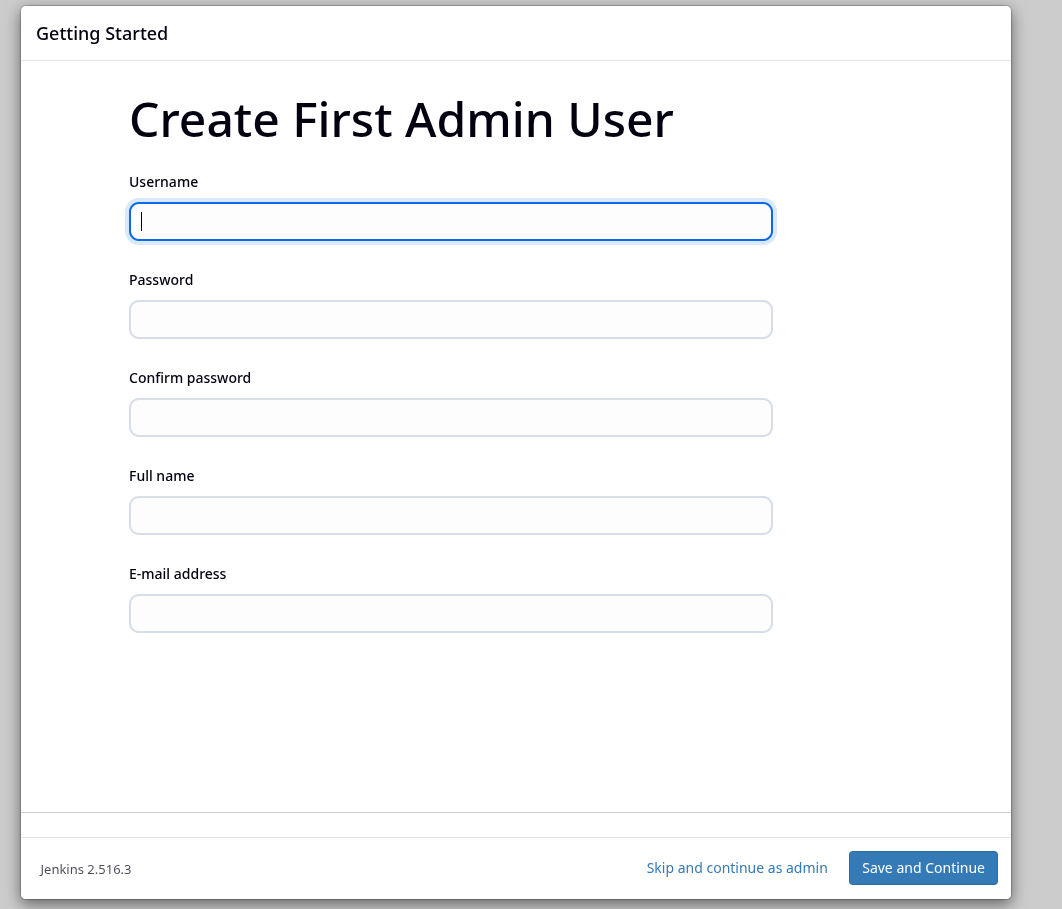
Install Suggested plugins

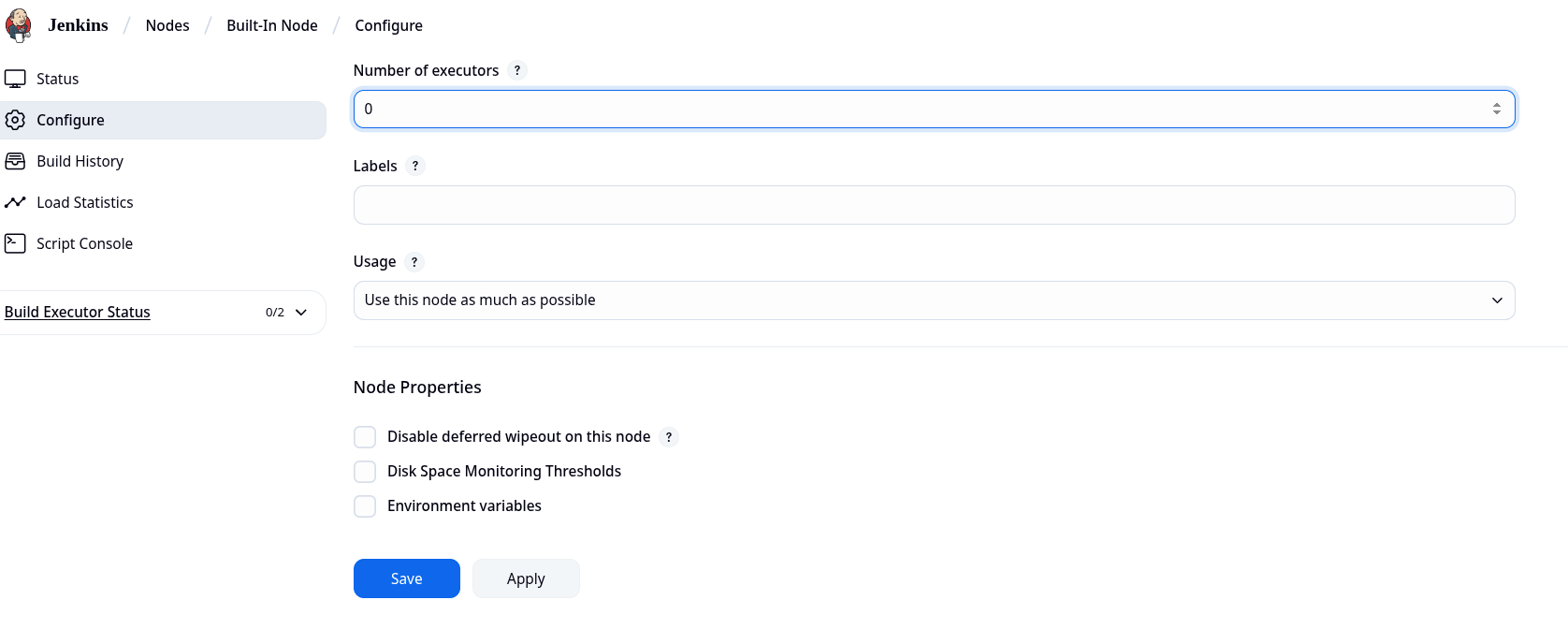


Create user



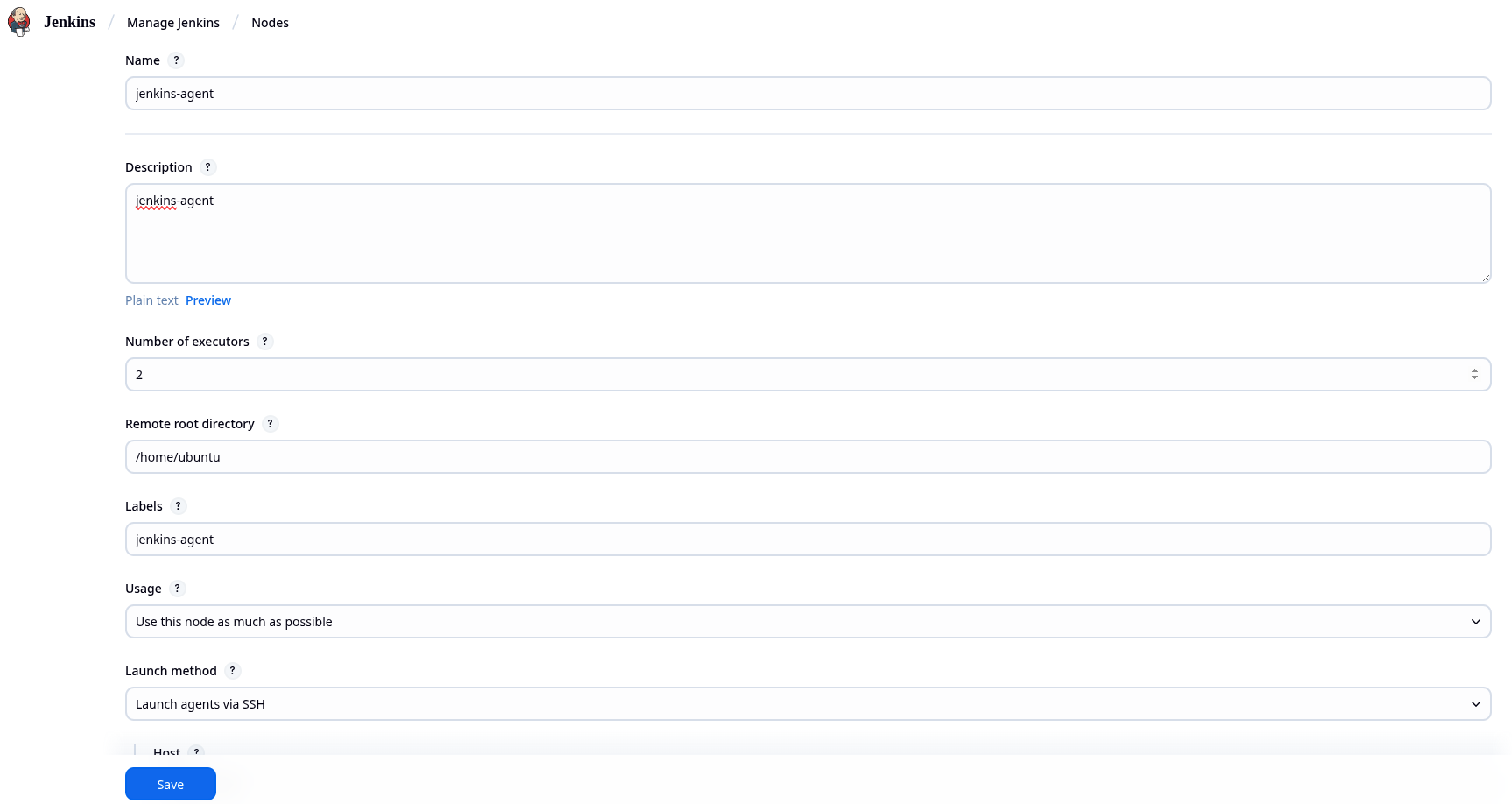
Save and finish after creating user

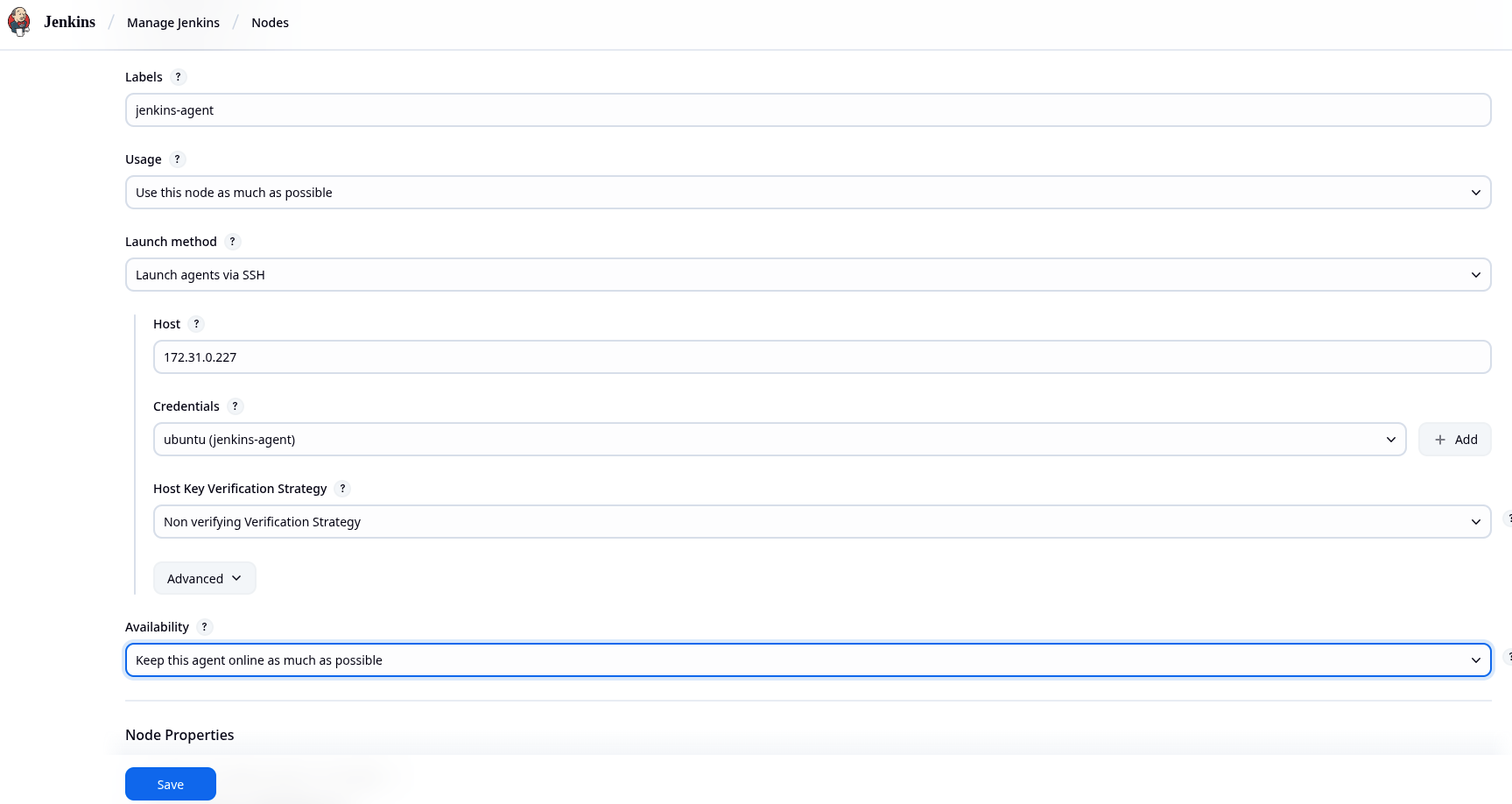
in nodes go to build in nodes and make number of executors 0.



Make a new node called jenkins-agent – should be a permanent node

Configurations of the Node

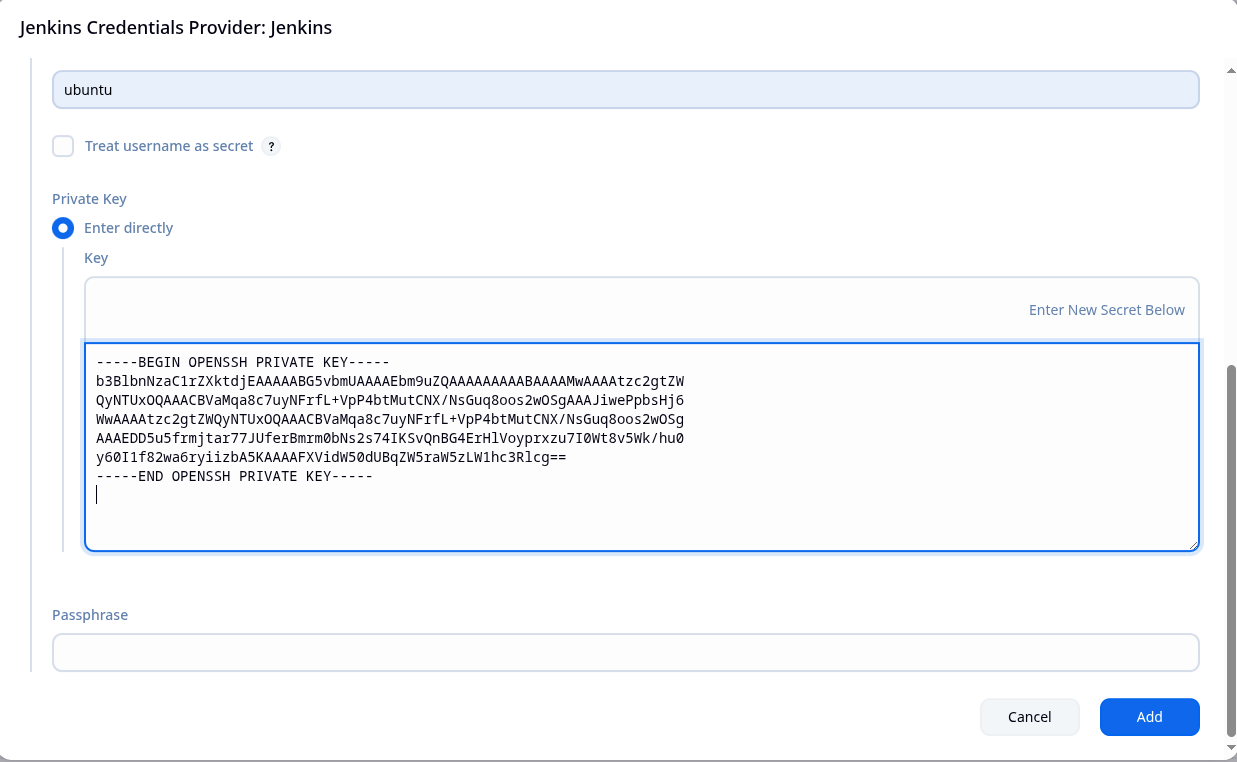




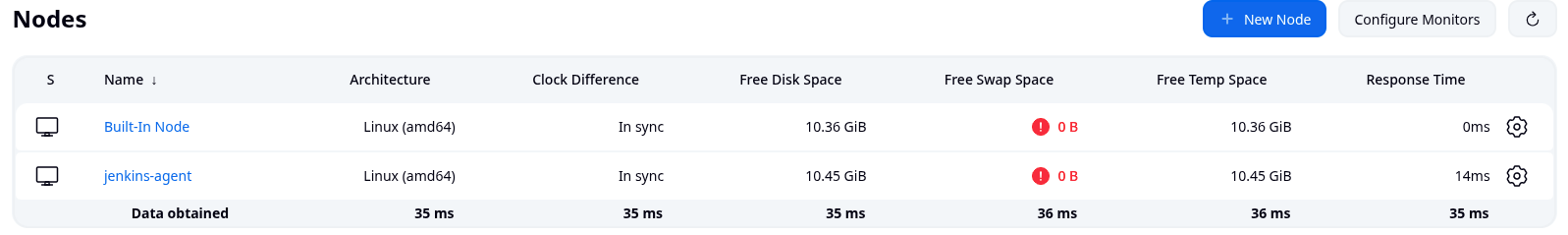
Credentials Configuration , click on add +



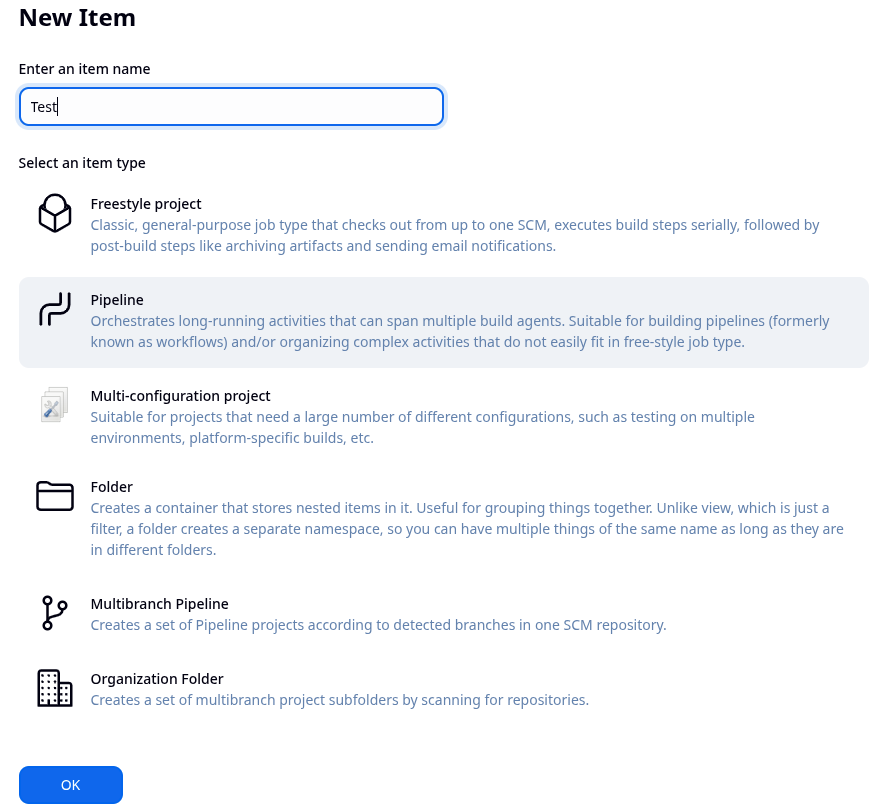
Enter the private ssh key from the jenkins-master instance



Click refresh and make sure agent is added

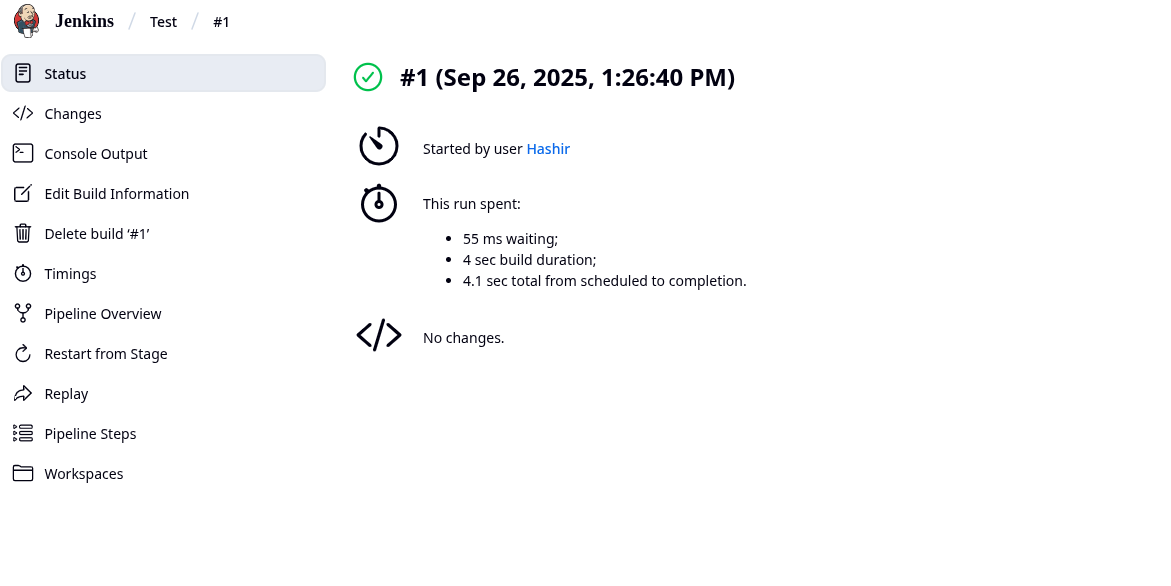


Create job called test , as pipeline then do these configurations



Select pipeline script as hello word and then apply and save

Then run the build and check if it works on the agent ( successful build should look like this )



This confirms us that the connection between the agent and master is successful, you can delete this job now, it was made only for testing connection purpose.

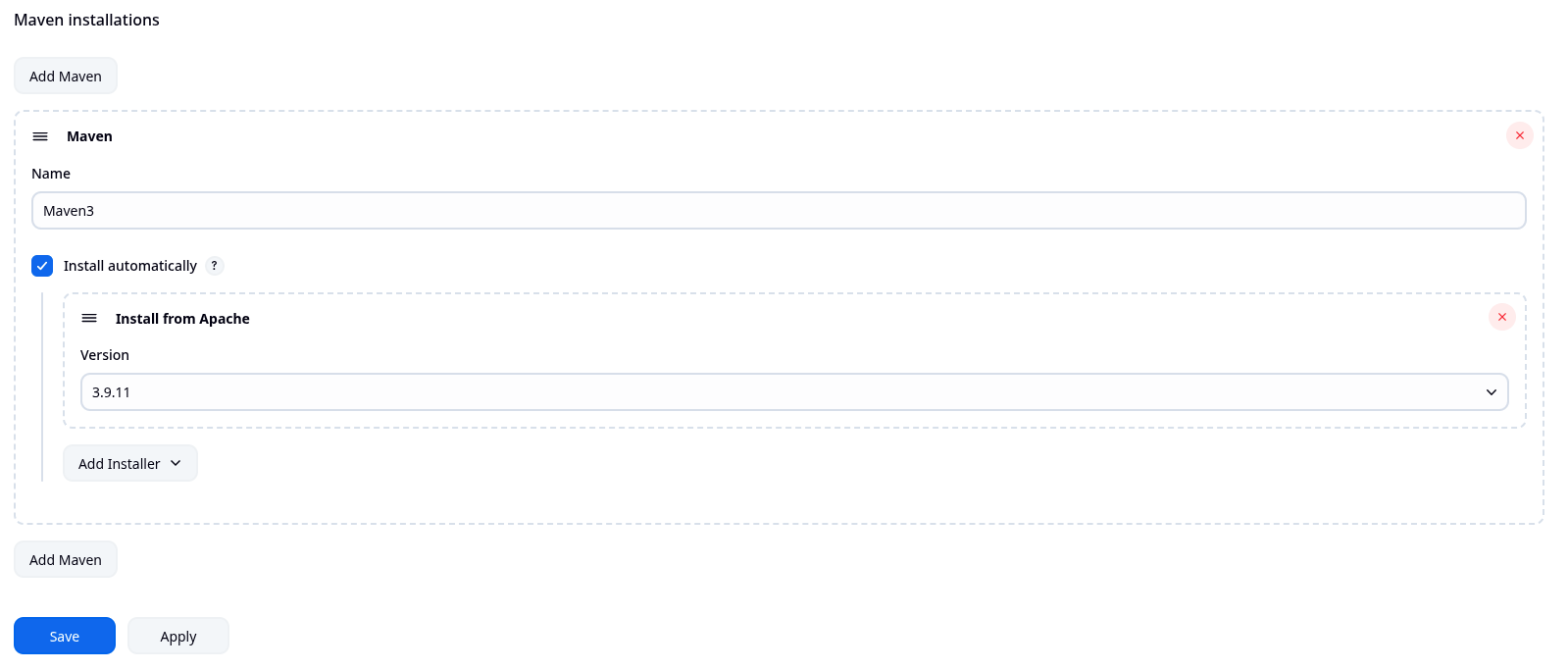
Lets now integrate Maven to jenkins and add the github credentials to jenkins

Install plugins required

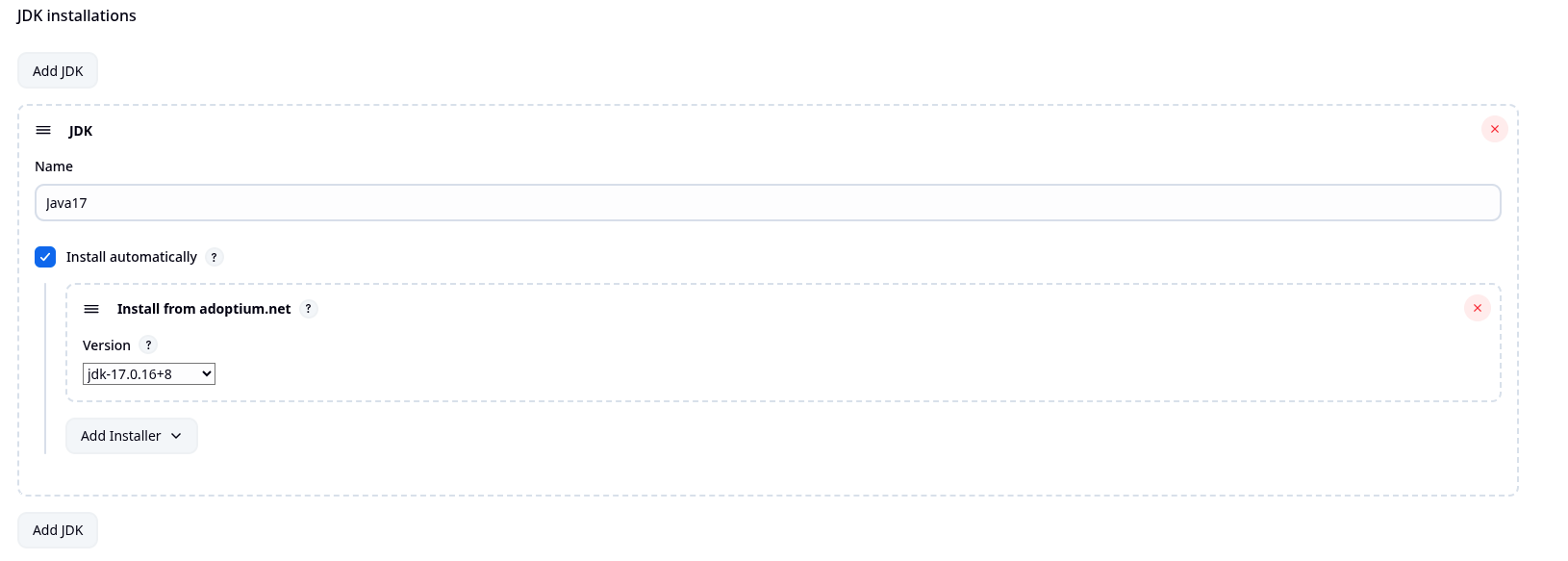
* Maven Integration
* Pipeline Maven Integration
* Eclipse temurin installer

After installing the plugins we need to configure them

Head to tools in manage jenkins and scroll to maven installations

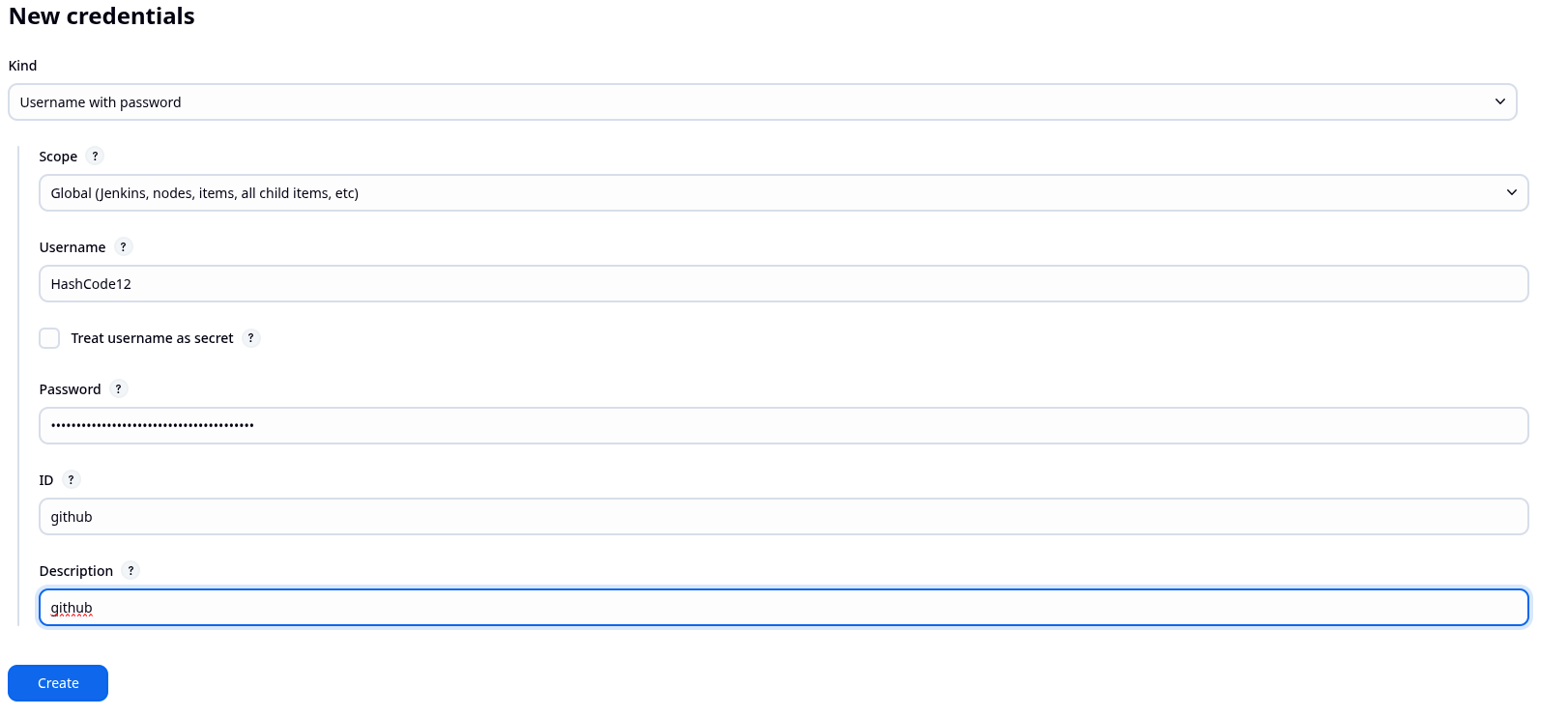


Also do JDK installations



Now lets add github credentials, go to credentials in security section and add credentials in stores scoped to jenkins

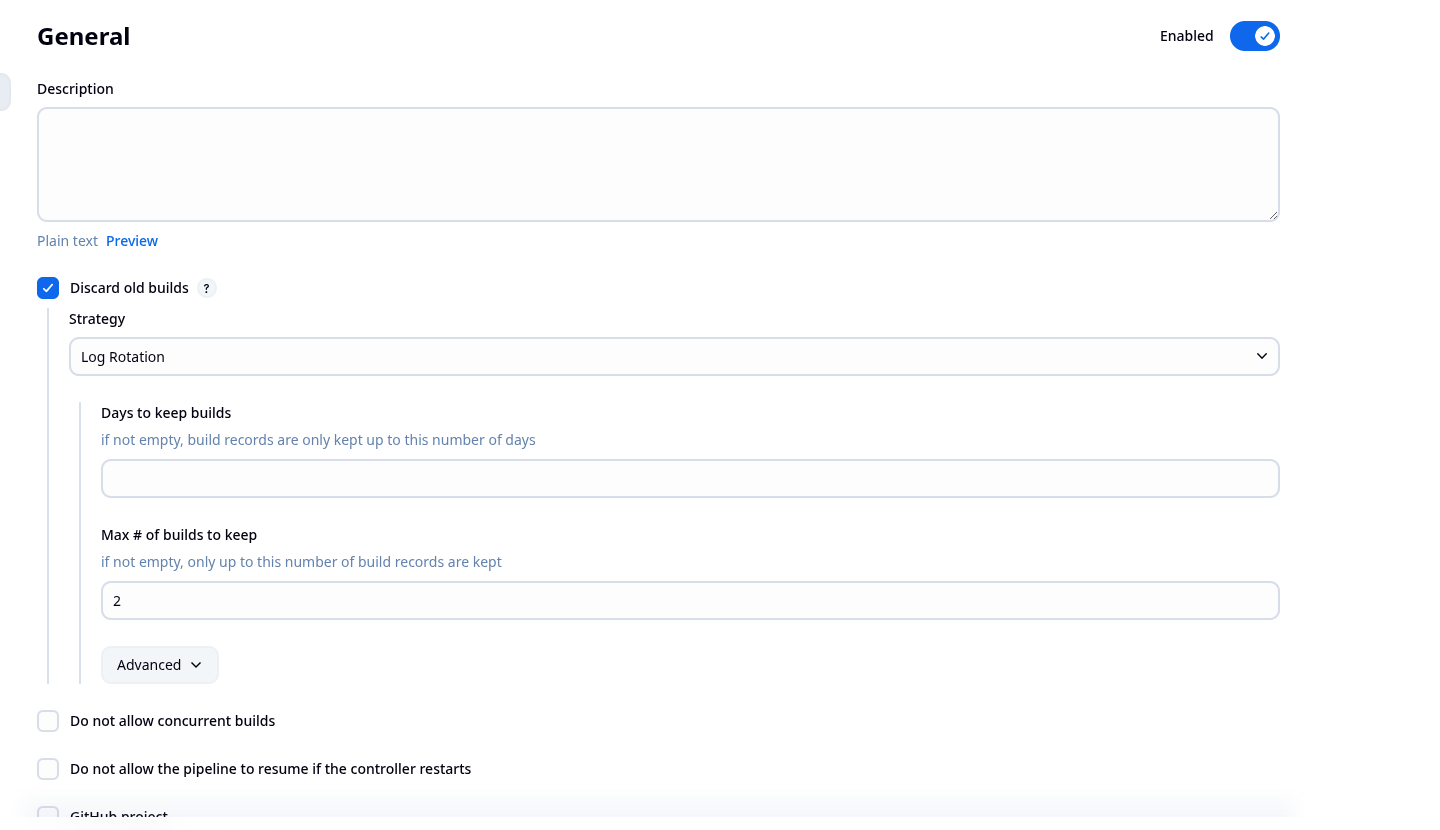
Write your github username in Username section and use a classic token in the password section.

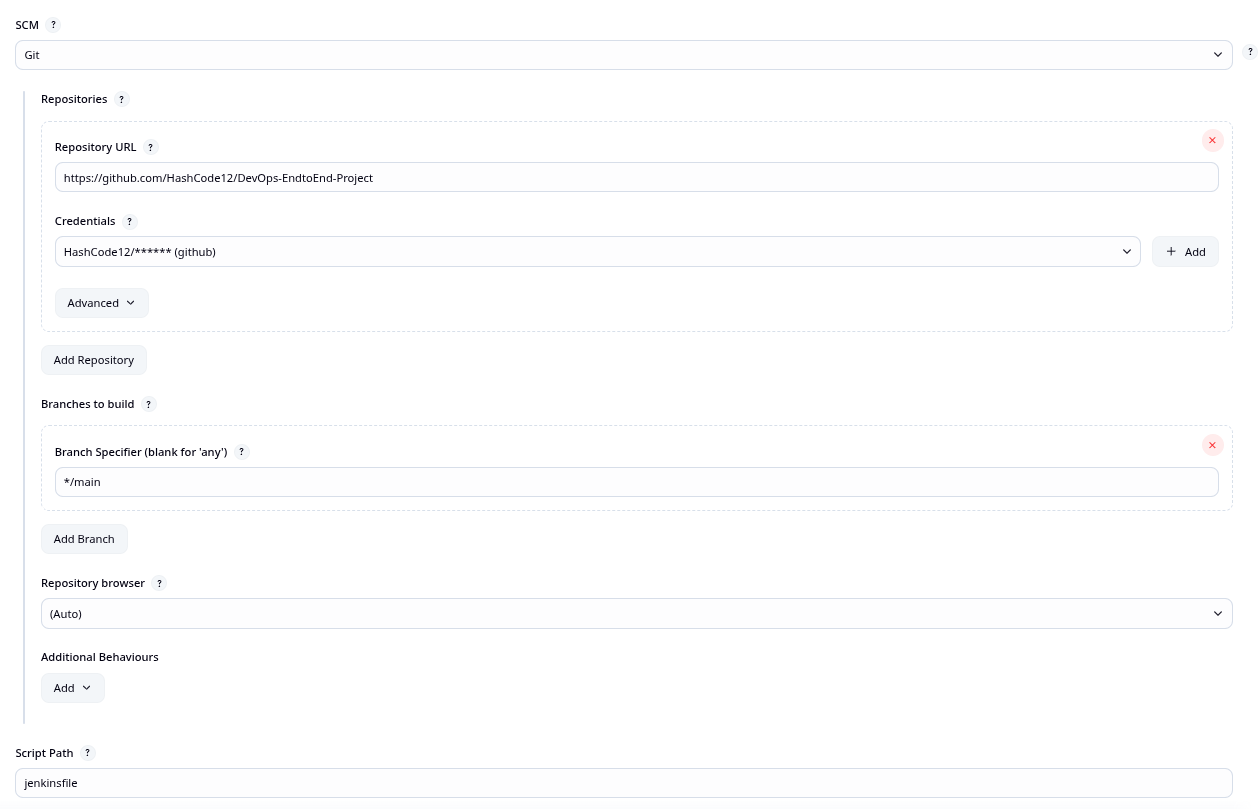


We will create the pipeline now in github

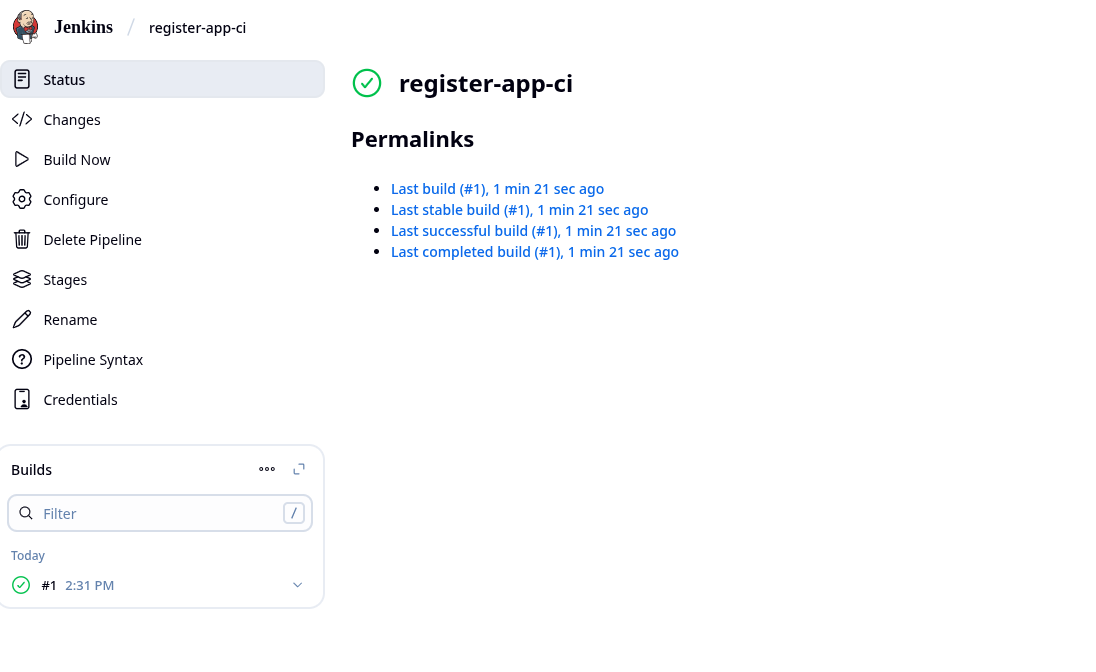
Go into the repo and take the jenkins script file

Now in jenkins create a job register-app-ci as pipeline, follow the configurations done below





After making the pipeline you can build it, note it might take a while.



Should be successfully built, we can move to SonarQube now.