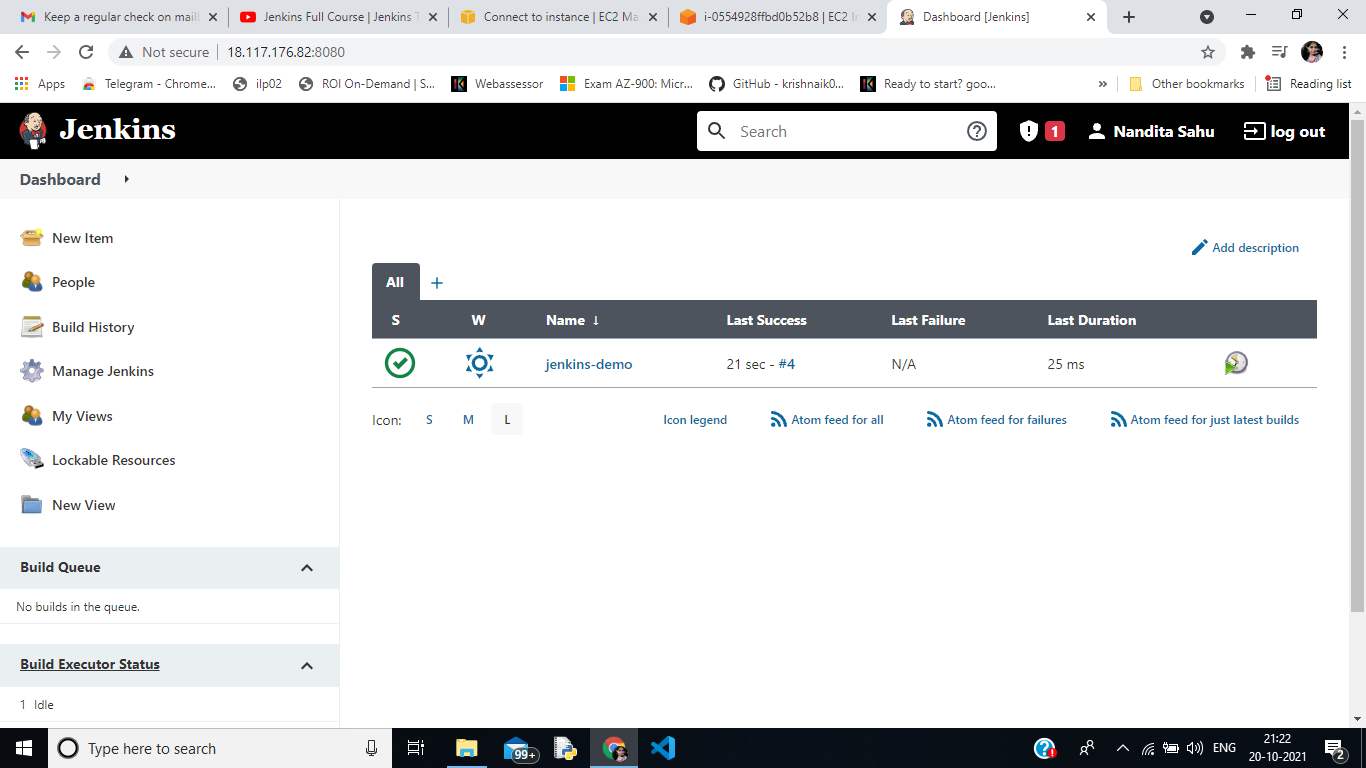
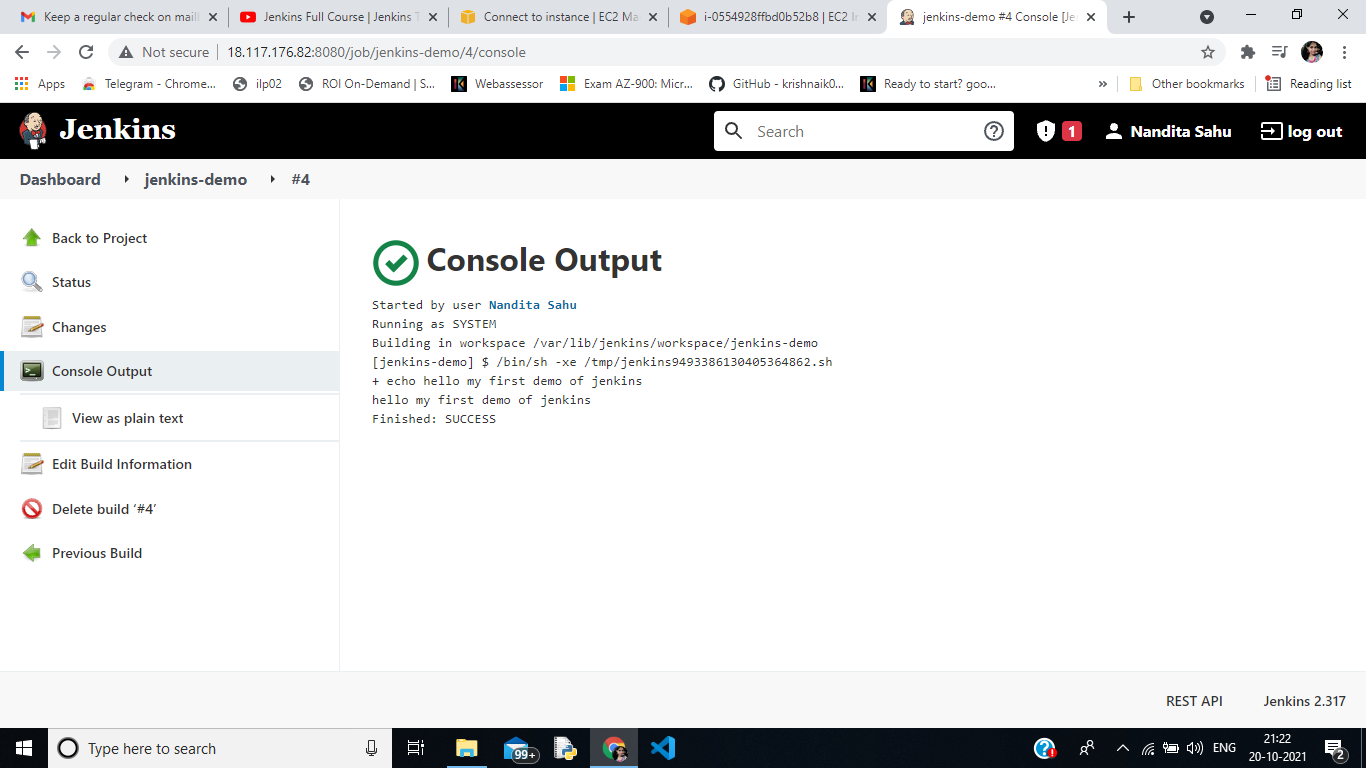
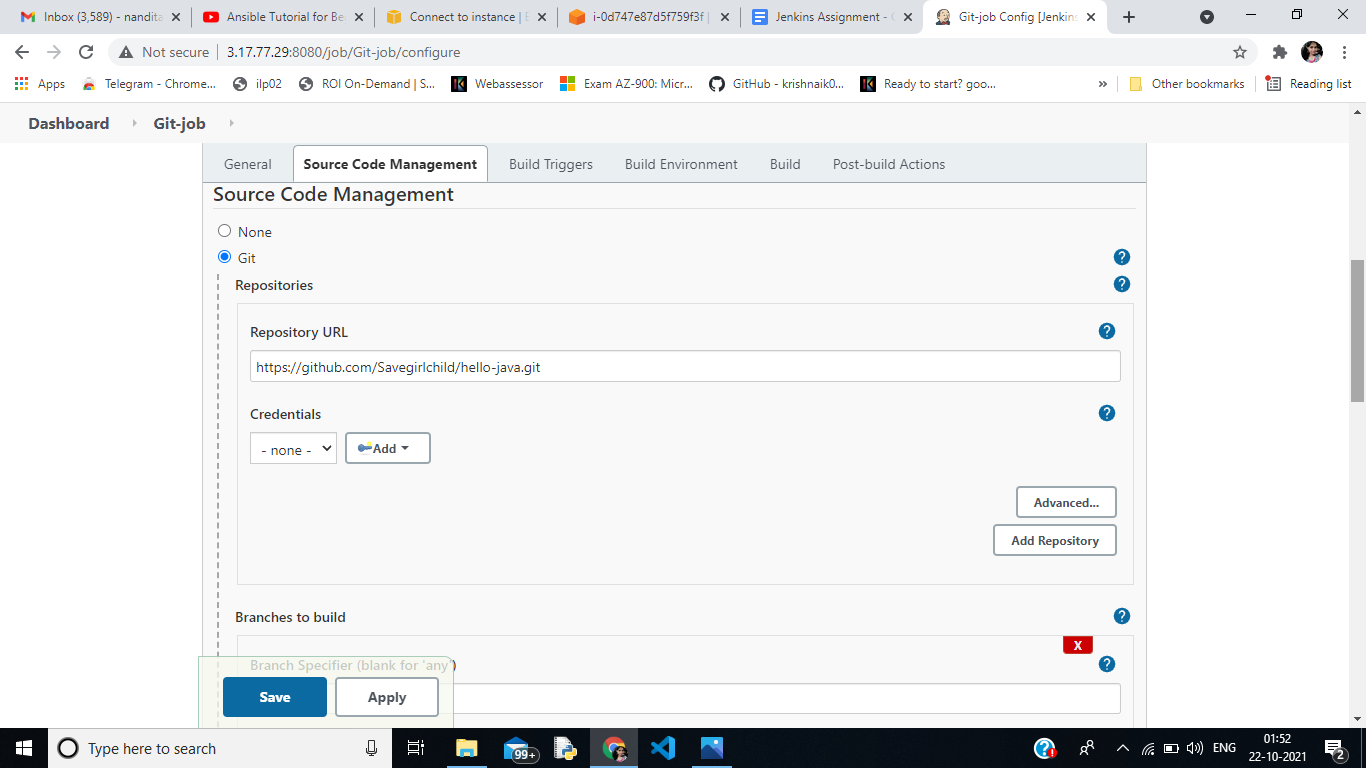
**Jenkins Commands :**

****

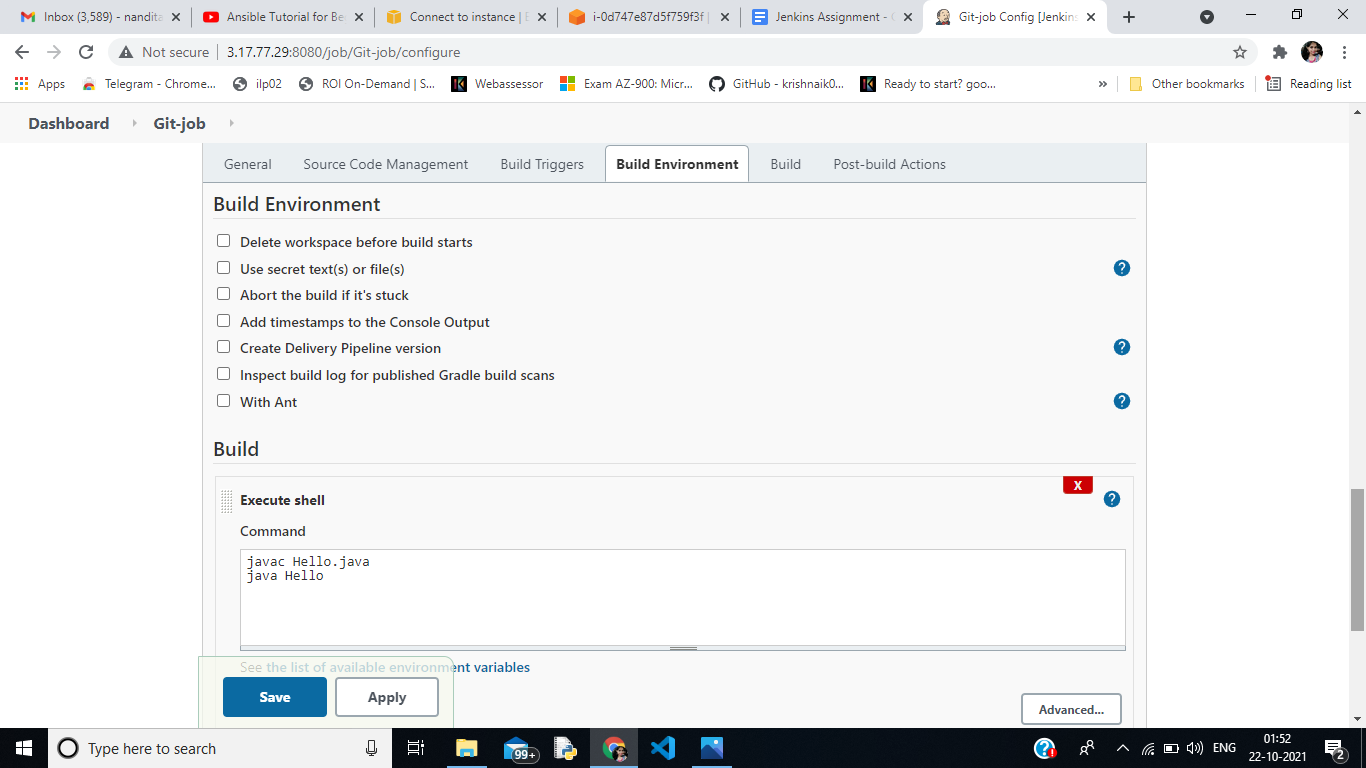
1. **Created first “jenkis-demo” project .**
2. **Go to New Item → Free style Project → Save→ Build→ Add Build steps→ Excute Shell (echo “ “)**

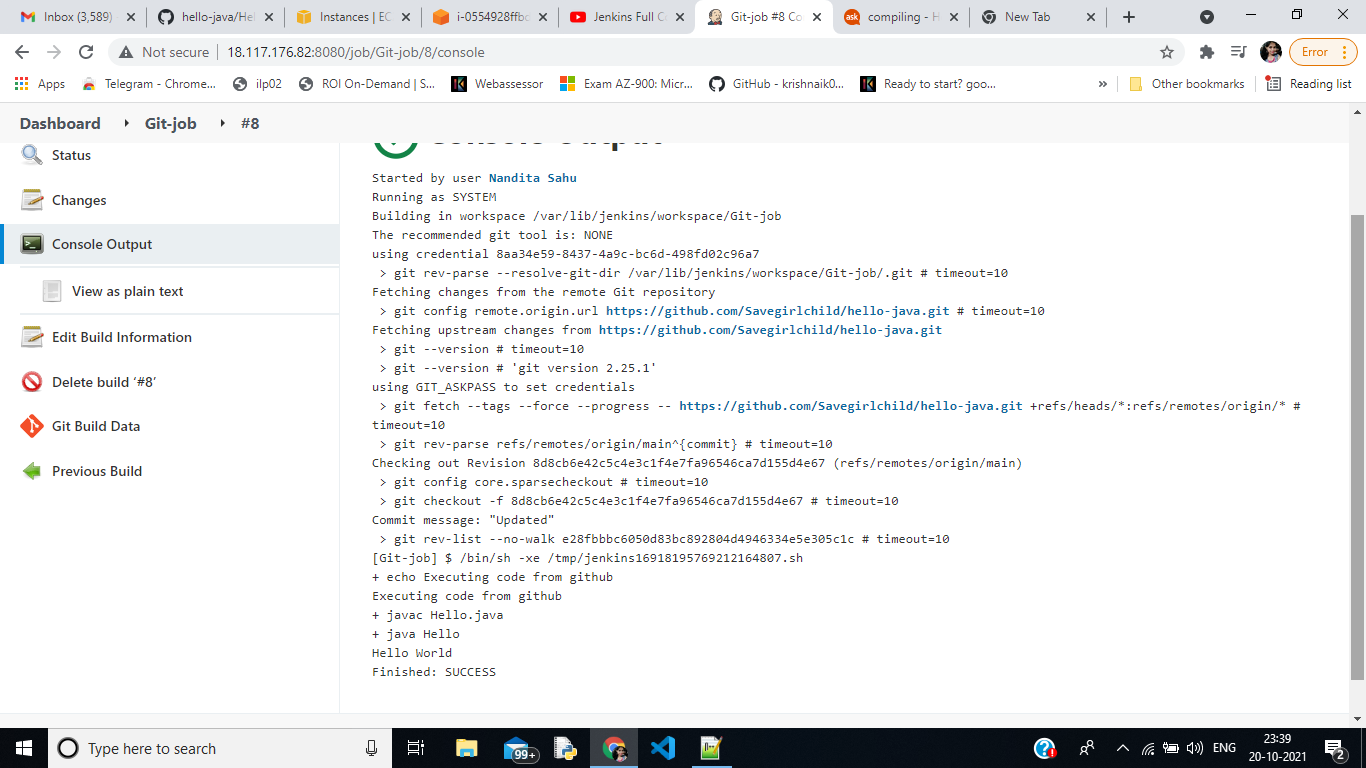
****

1. **Output of the jenkins-demo project**

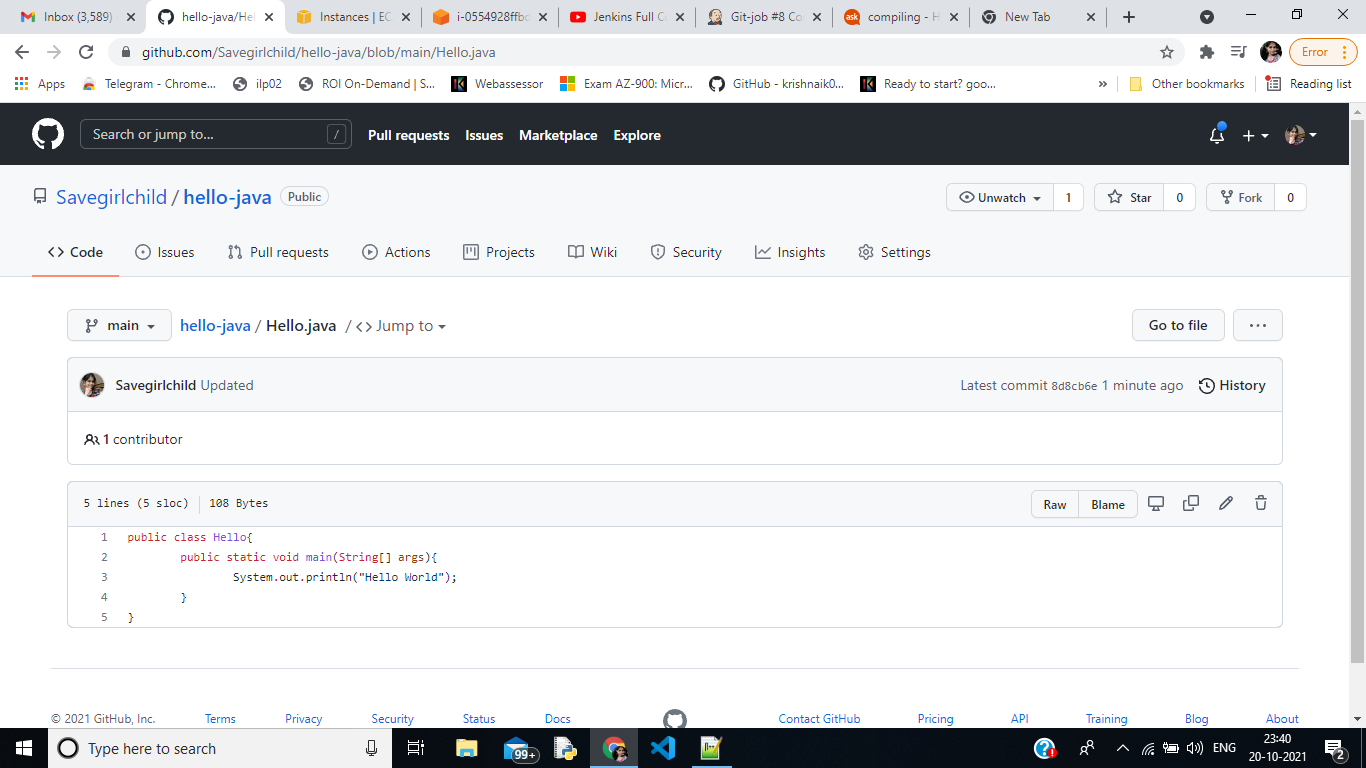
****

1. **Created a git-job project where we clone the project from github by putting prepositoy url , put the branches and then execute command as per the build steps**

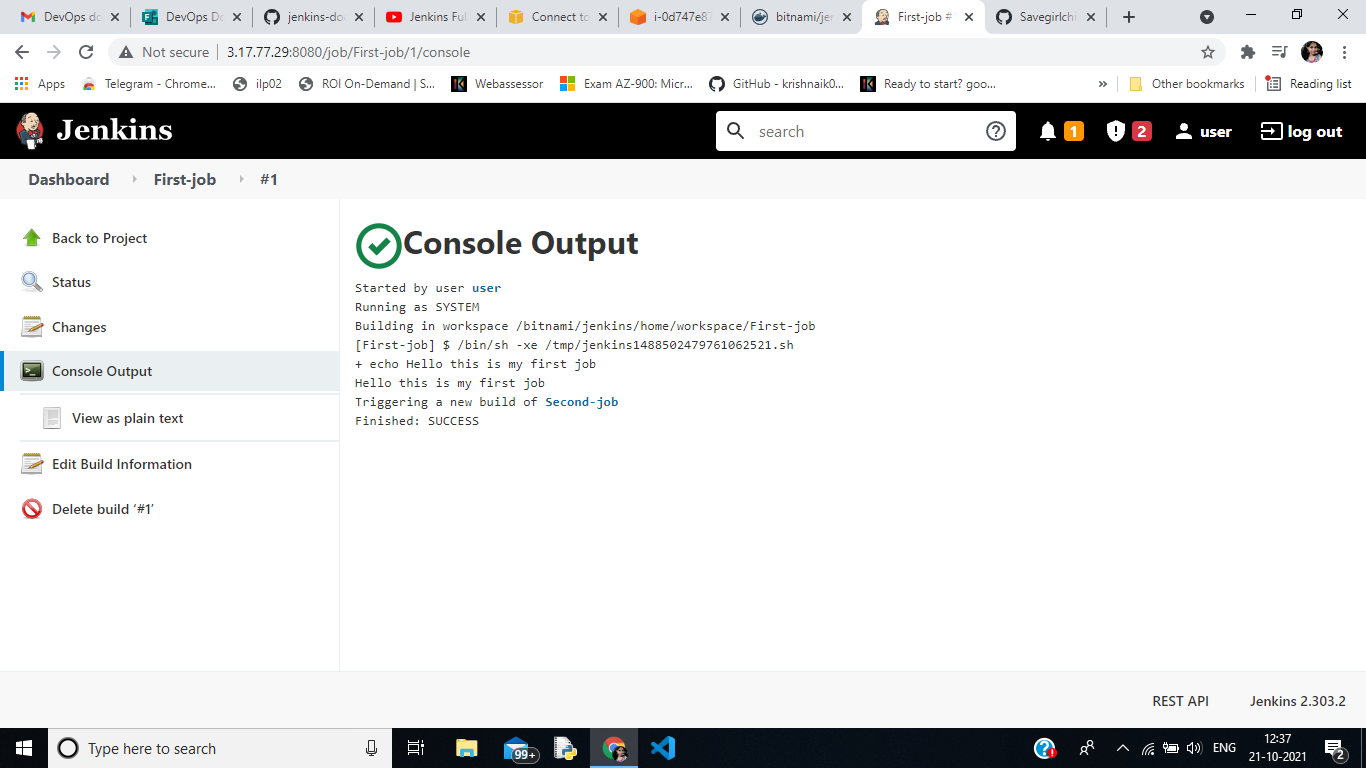
****

****

1. **Output of the git job**

****

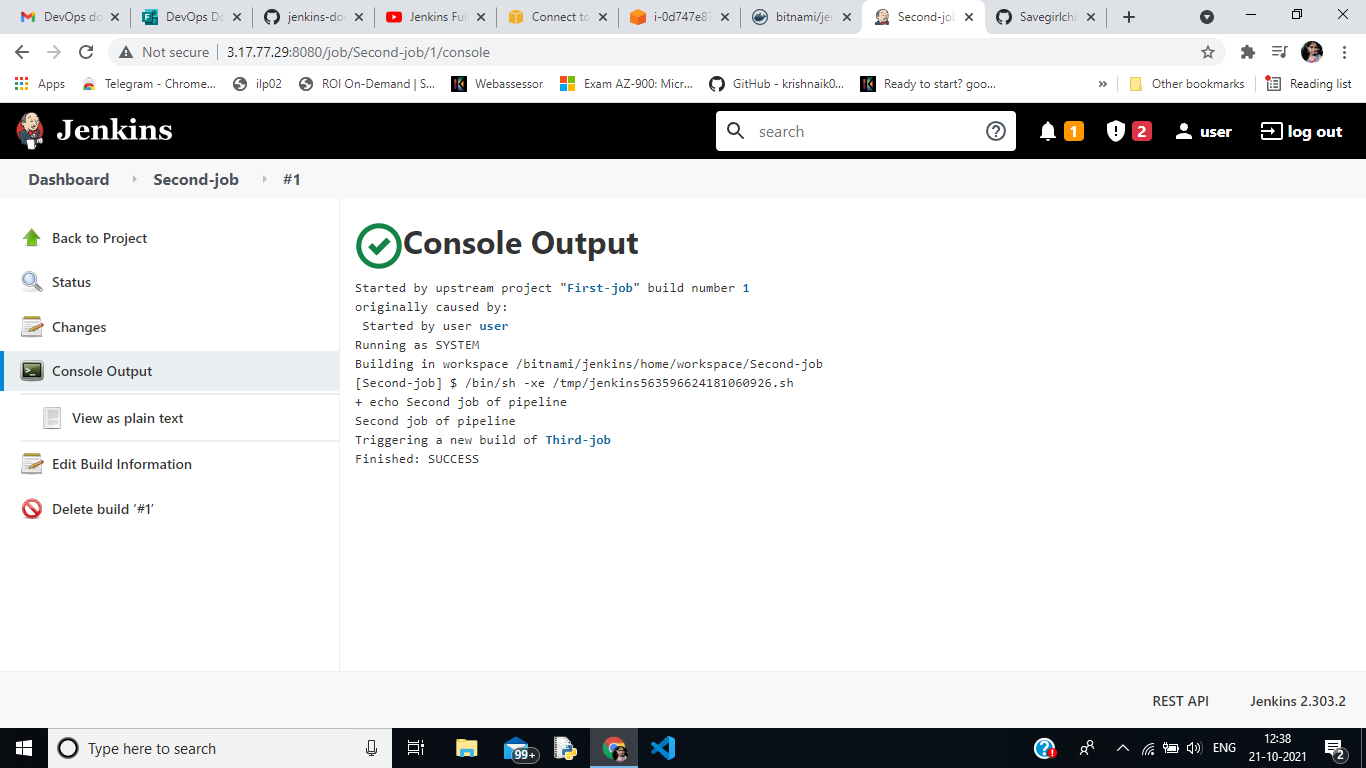
1. **The git hub repository for git-job project**

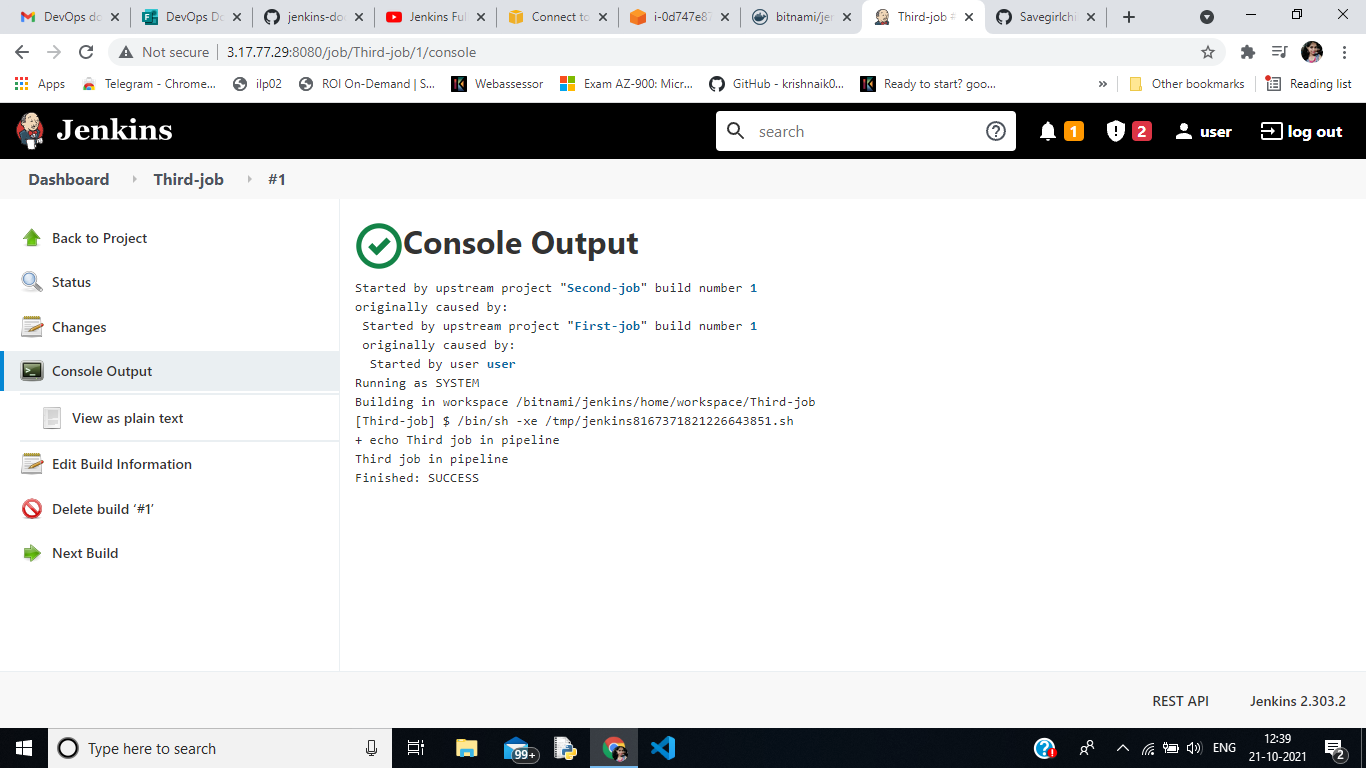
****

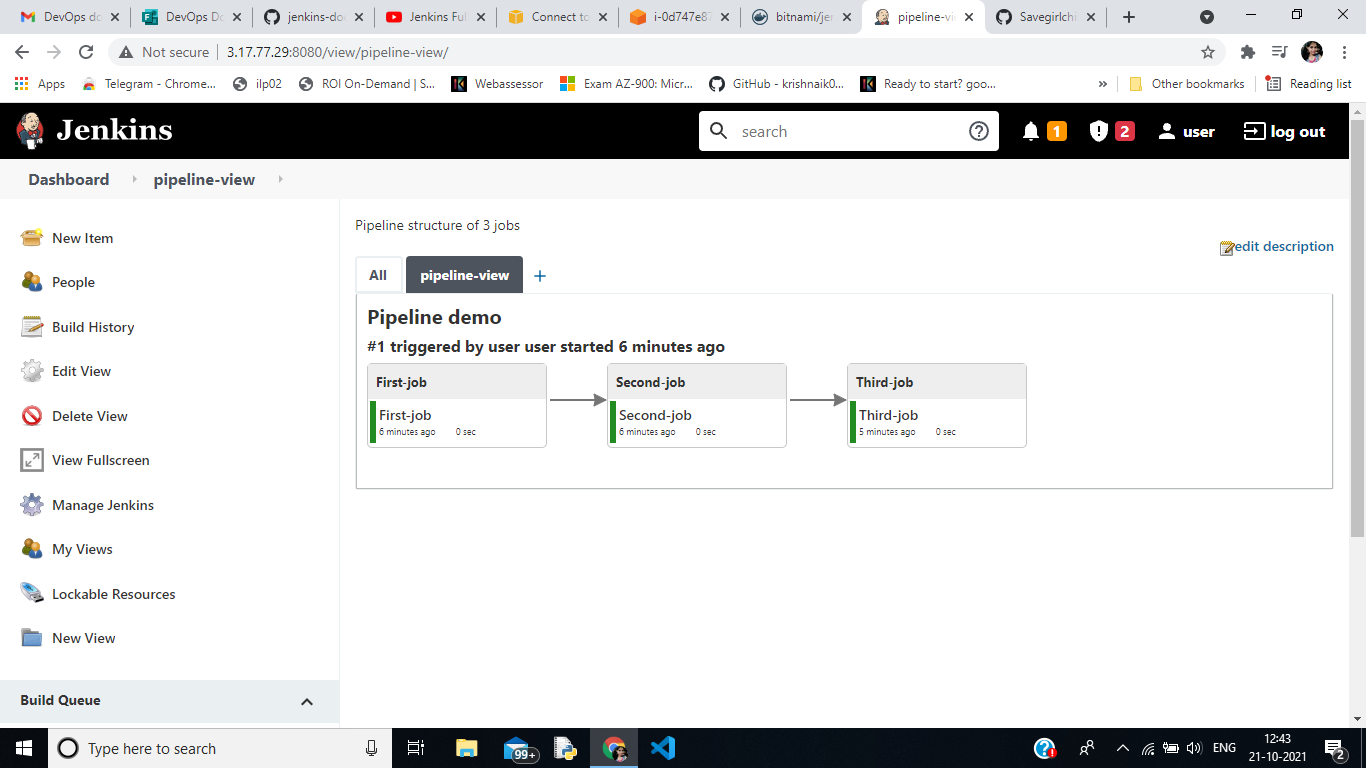
1. **First way to Create Pipelines :**

**Create 3 jobs and in first job post-build actions , choose projects to build :**

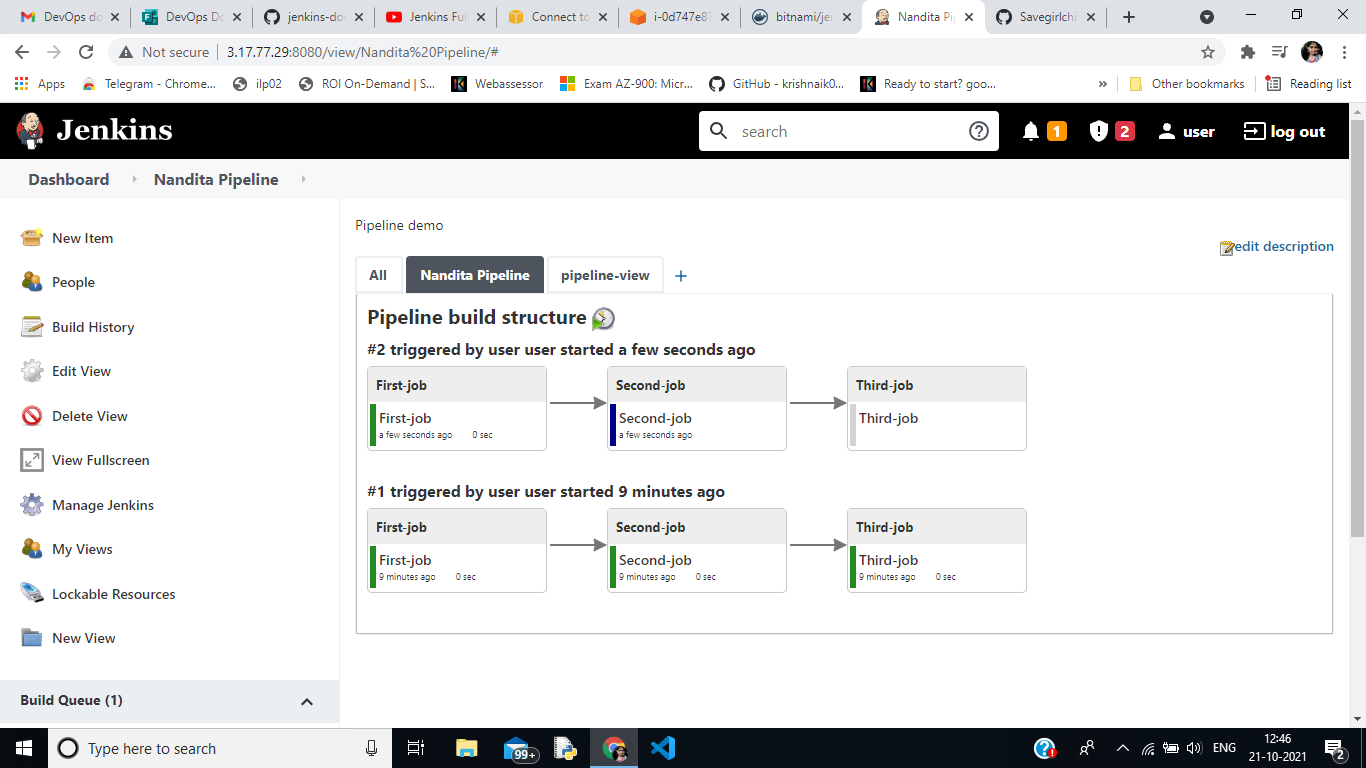
**Choose Second-job , which means that after the first job is completed , it will trigger into second job and viceversa for third job as well**

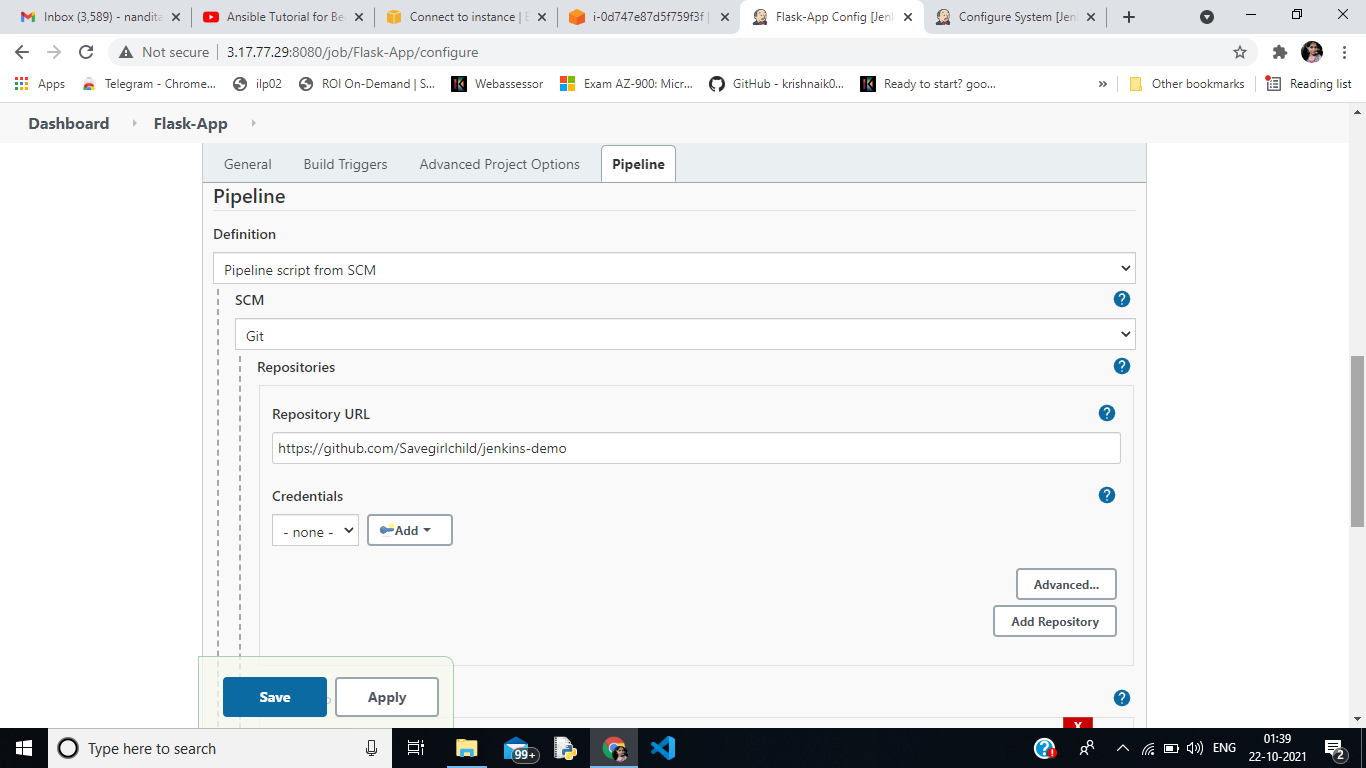
****

****

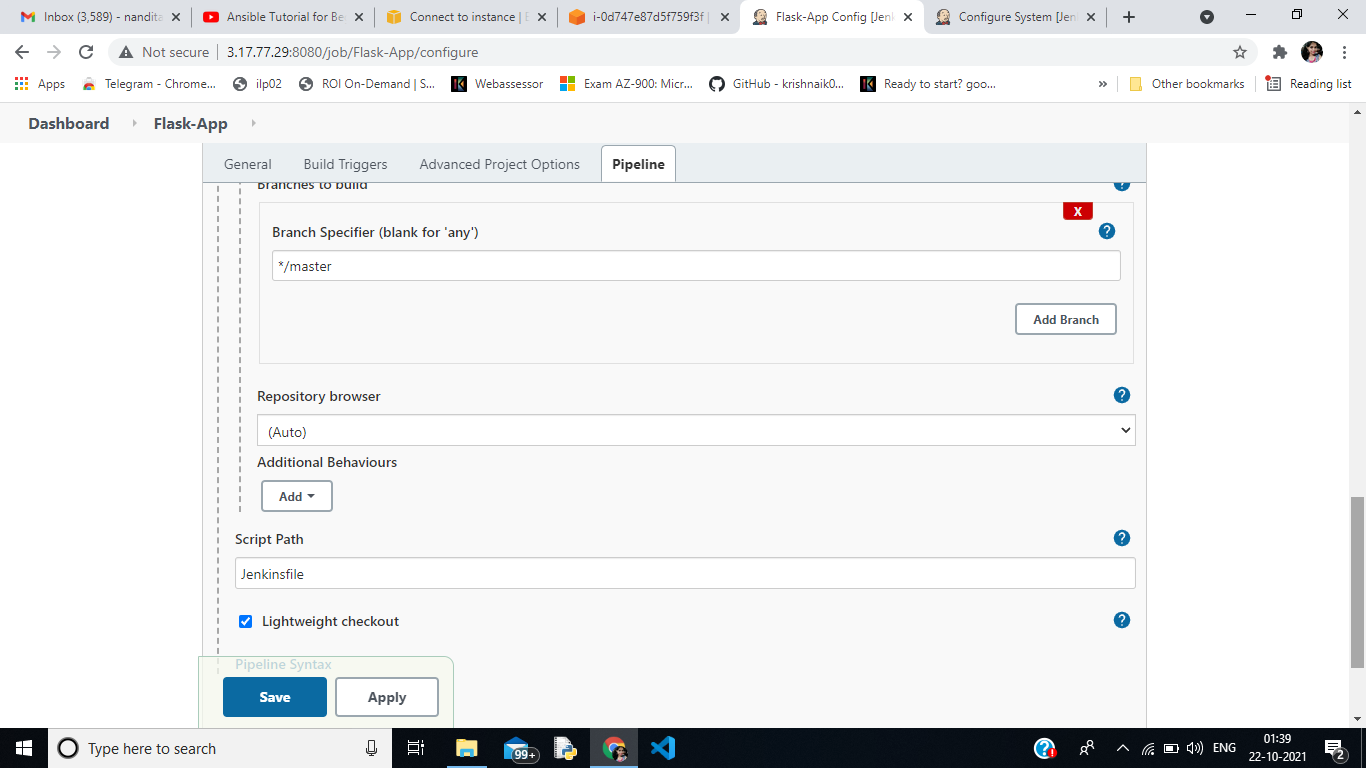
****

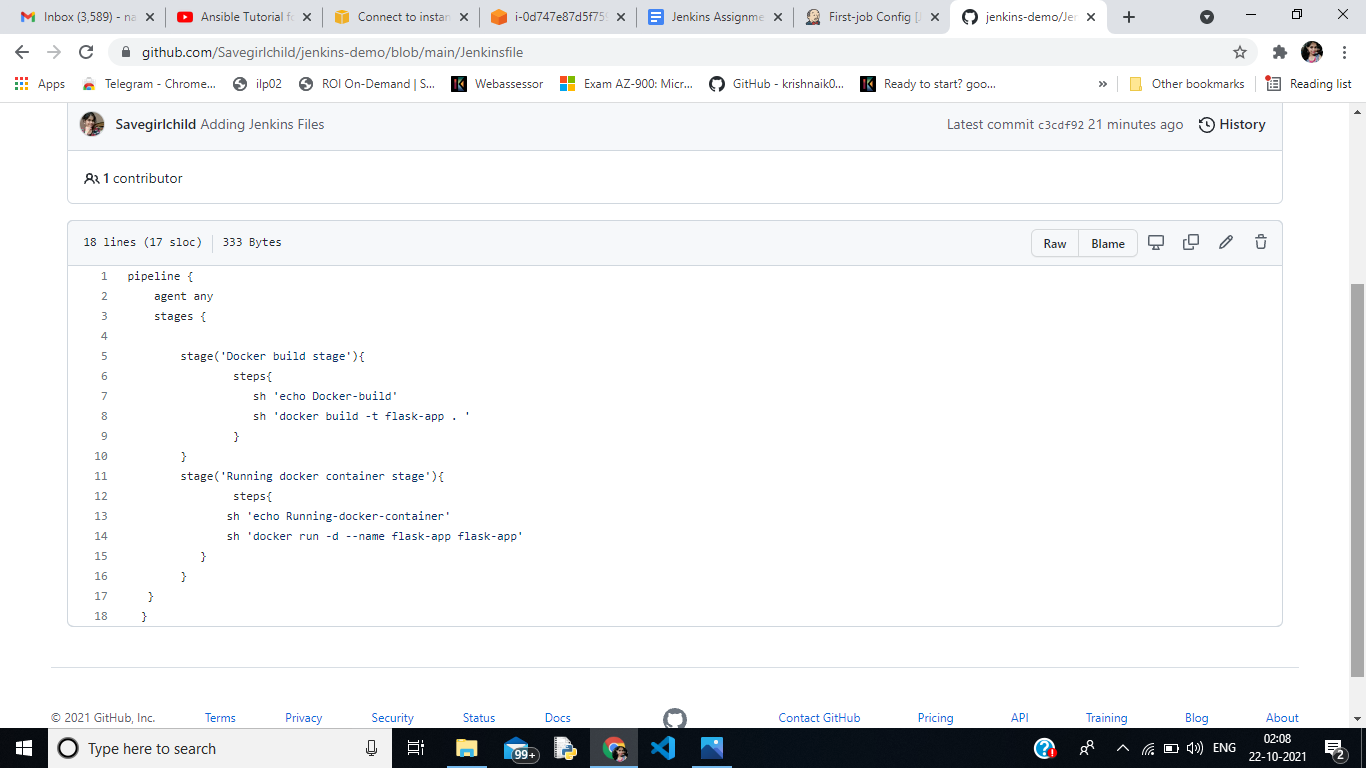
1. **Visualization of the running of jobs in pipeline . Use plugins (Delivery pipeline plugins)**

****

****

1. **Created Flask-App project where in the git we have added all Dockerfile , app.js, Jenkinsfile where the pipeline is structured . In Project → Pipeline , we added the git repository link and apply.**

****

****

1. **In the Jenkins File , pipeline is created by declarative type where agent ‘any’ means any agent can run this file . We have two stages where in 1st stage**

**“ Docker build stage “ is performed and then “Docker container run command”**

**Github Repository :** [**https://github.com/Savegirlchild/jenkins-demo**](https://github.com/Savegirlchild/jenkins-demo)

**Dockerhub Image :** [**https://hub.docker.com/repository/docker/nanditasahu/jenkins-demo**](https://hub.docker.com/repository/docker/nanditasahu/jenkins-demo)