

To bifurcate our jobs, we can create multiple views in Jenkins dashboard:

Dashboard → New View → add Name (Test) → Type > List View → Click on Create → OK

You can create new jobs under these views.

HOW TO RUN PARALLEL & SEQUENTIAL JOBS INSIDE JENKINS?

Scenario: Suppose we have 3 jobs- Job1, Job2 & Job3, when Job1 gets finished Job2 should get triggered and when Job2 finishes Job3 should get triggered -- [Sequential Job]

Can be done with the help of **"Post Build Action"** in Jenkins

SEQUENTIAL JOB:

Create Job1:

Dashboard → New Item → Add Name (Job1) → Freestyle Project → OK → Add build steps → Execute Shell

```
sleep 10
```

→ Apply + Save

Create Job2:

Dashboard → New Item → Add Name (Job2) → Freestyle Project → OK → Add build steps → Execute Shell

```
sleep 10
```

→ Apply + Save

Create Job3:

Dashboard → New Item → Add Name (Job3) → Freestyle Project → OK → Add build steps → Execute Shell

```
sleep 10
```

→ Apply + Save

Now,

Open Job1 → Configure → Post-Build Actions → Add post build action → Build other projects → under, Projects to build add Job2 → select Trigger only if build is stable → Apply + Save

You can see under Status,

Inside Job1's Downstream Project is – Job2

Again, Open Job2 → Configure → Post-Build Actions → Add post build action → Build other projects → under, Projects to build add Job3 → select Trigger only if build is stable → Apply + Save

For Job2 under Status,

Upstream projects – Job1

Downstream Project is – Job3

Go to Job1 and click **Build now**

PARALLEL JOB:

Create a new job named as- trigger

Dashboard → New Item → Add Name (trigger) → Freestyle Project → OK → Post-Build Actions → Add post build action → Build other projects → under, Projects to build add Job1, Job2, Job3 → select Trigger only if build is stable → Apply + Save (***note:** remove post builds created under Job1, Job2 & Job3 in last practical)

Under status for job trigger:

Downstream projects:

Job1 Job2 Job3

Open job trigger and click on Build now

Jenkins will now execute three jobs parallelly.

TO AUTO TRIGGER JOBS:

Build Periodically, Using CRON JOB or CRON TAB EXPRESSION

*	*	*	*	*
MINUTE	HOUR	DAY OF MONTH	MONTH	DAY OF WEEK
(0-59)	(0-23)	(1-31)	(1-12)	(0-7)

We can write time zone as well: TZ

@daily - will run jobs daily

@hourly - Will run the jobs hourly

@weekly - will run jobs weekly

To run job in every 2 minutes:

H/2 * * * *

To run a job at 8:30 am in the morning:

30 08 * * *

To run job Monday to Friday at 9:00 am:

00 09 * * (1-5)

To run a job at 9:00 pm in the evening:

00 21 * * *

To create a job which will execute after 2 minutes automatically:

New Item → Add name (build-pre) → Freestyle Project → OK → Build Triggers → Build Periodically → Inside schedule add our CRONE pattern as

H/2 * * * *

→ Add build step → Execute shell

echo "Hello"

→ Apply + Save

Click on **Build now** next time it will automatically build after every 2 minutes.

Integrating with GitHub:

Dashboard → new item → add name (repo-build-per) → Freestyle Project → OK → click on Git → Add Repository URL (add credentials if repository is private) → Build Triggers → select Build periodically → Schedule

H/2 * * * *

→ Apply + Save

Click on **Build Now**

Our git repository will be cloned after every two minutes, because we have set build periodically after every 2 minutes.

POLL SCM:

It will search our SCM for changes and only if changes found Jenkins will run the build. Suppose we have added CRON as - H/2 * * * * Poll SCM will check GitHub repository after every two minutes for the changes and if changes are found only then it will start the build.

Let's create new job:

New item → Add name (poll-scm) → Freestyle Project → OK → select Git → add Repository URL → Build Triggers → Poll SCM → schedule

H/2 * * * *

→ Build Environment → Delete workspace before build start → Apply + Save → Click on **Build now** once.

Until and unless any changes are made on the GitHub repository, Jenkins will not run a new build.

Make a commit in GitHub repo and check if Jenkins runs the new build.

GITHUB WEBHOOK:

With the help of GitHub as soon as any change made at GitHub repo immediately executed by Jenkins, no Poll SCM or Periodically Build.

First of all let's Create a credentials for Webhook on Jenkins:

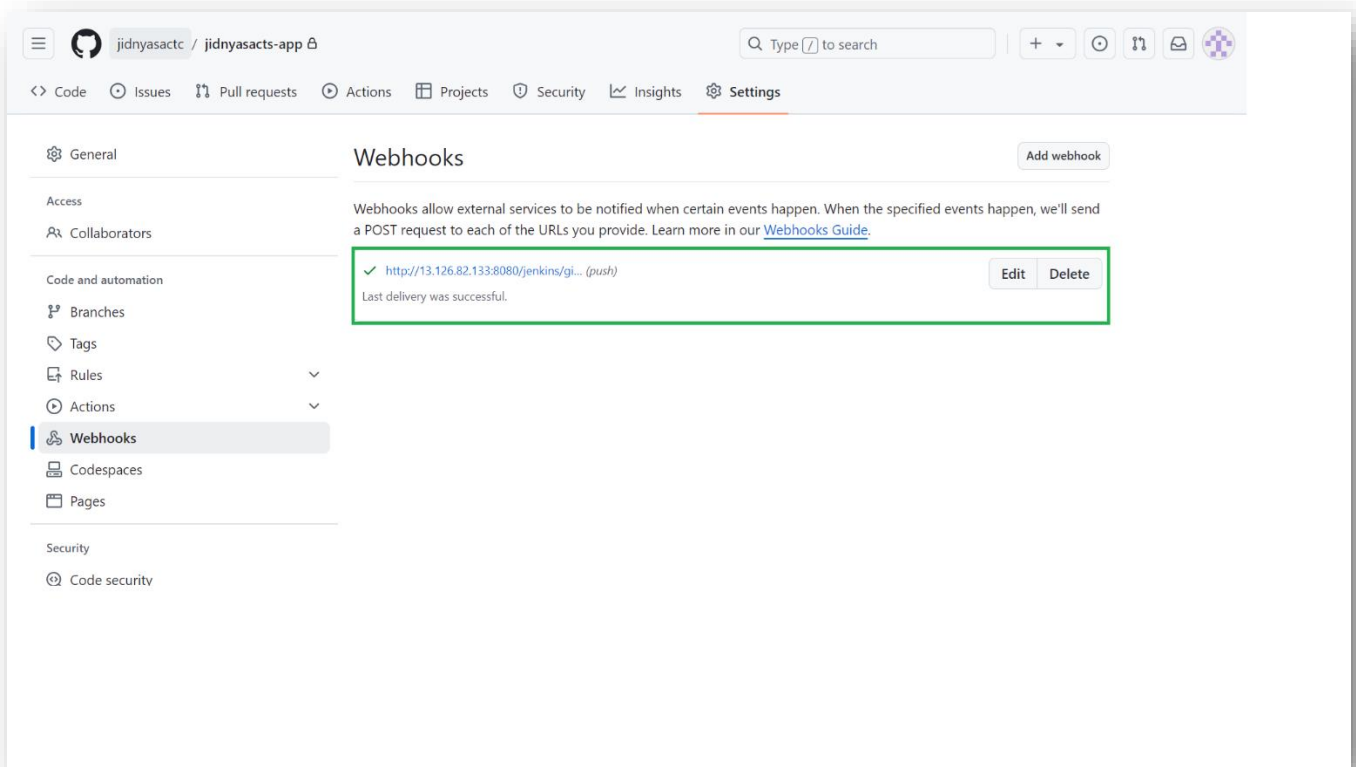
Dashboard → Manage Jenkins → System → Select GitHub → Add GitHub Server → Add name (test) → Credentials > Add → Jenkins → Kind > Secret text → add personal access token in Secret → ID (custom ID) → Description (custom description) → Add → Click on Test connection (should get msg like- credentials verified for user After successful connection) → Apply + Save

Create a new job:

Dashboard → New item → name (webhook) → Freestyle Project → OK → Git → add Repository URL → Build Triggers → Select GitHub hook trigger for GITScm polling → Delete workspace before build start → Apply + Save

Now go on GitHub and open GitHub Repository

settings → Webhooks → Add webhook → Add Payload URL (Add "jenkins URL + github-webhook" eg: <http://65.02.31.65:8080/jenkins/github-webhook/>) → Content type > application/json → select > just the push event → Add webhook → refresh the page you should get a tick as below screenshot



As soon as you commit any changes to the GitHub repository you can see that your job has been triggered to build in Jenkins automatically.