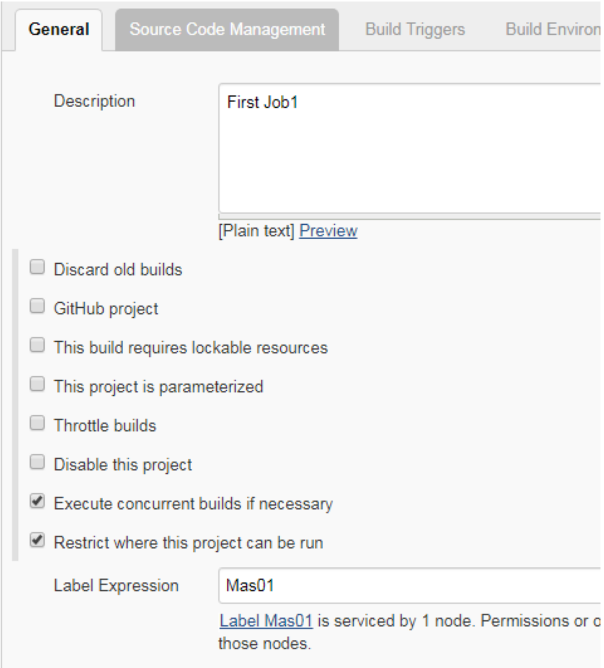
Objective

1. How to **link** the jobs
2. How to run the jobs with **Labels**
3. Execute **Concurrent** builds
4. **Trigger Remotely**

Steps:

A. Create the first Job with



Description: First Job1

[Plain text] Preview

□ Discard old builds

□ GitHub project

□ This build requires lockable resources

□ This project is parameterized

□ Throttle builds

□ Disable this project

☑ Execute concurrent builds if necessary

☑ Restrict where this project can be run

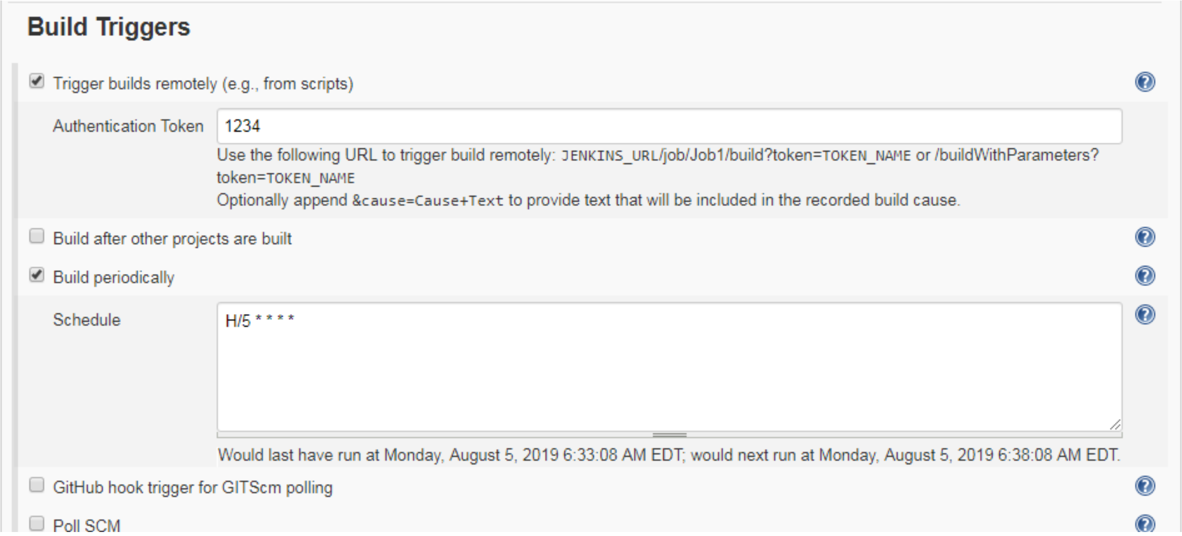
Label Expression: Mas01

Label Mas01 is serviced by 1 node. Permissions or c
those nodes.

Here, we are allowing the job to run “Concurrently” if required.

Also, we are specifying the Job to run only on the “master” node.

Under the trigger part



Build Triggers

☑ Trigger builds remotely (e.g., from scripts)

Authentication Token: 1234

Use the following URL to trigger build remotely: `JENKINS_URL`/job/Job1/build?token=`TOKEN_NAME` or /buildWithParameters?token=`TOKEN_NAME`
Optionally append `&cause=Cause+Text` to provide text that will be included in the recorded build cause.

□ Build after other projects are built

☑ Build periodically

Schedule: H/5 * * * *

Would last have run at Monday, August 5, 2019 6:33:08 AM EDT; would next run at Monday, August 5, 2019 6:38:08 AM EDT.

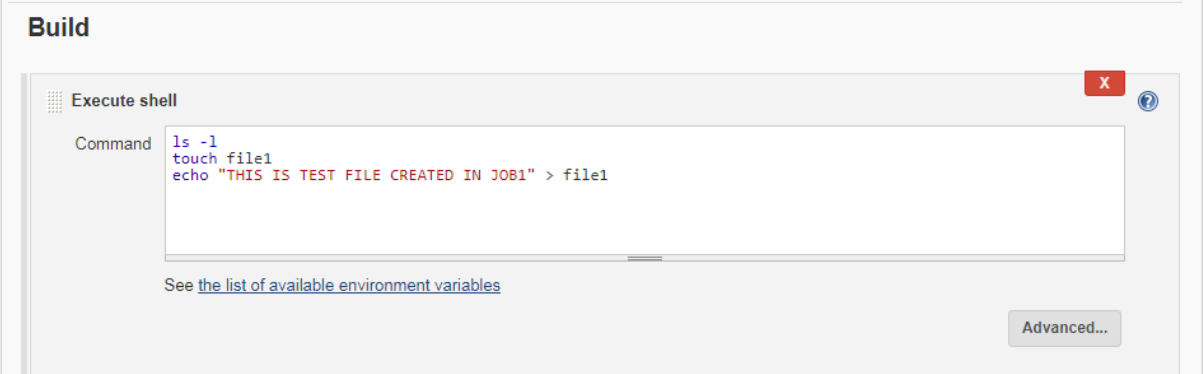
□ GitHub hook trigger for GITScm polling

□ Poll SCM

Here we are creating, 2 kinds of trigger.

1. Trigger remotely -> we can use a link and trigger this job remotely any whr in the network.
   Eg: -- **http://192.168.1.5:8080/job/Job1/build?token=1234**
2. Build Periodically – H/5 * * * * → means automatically trigger every 5 min

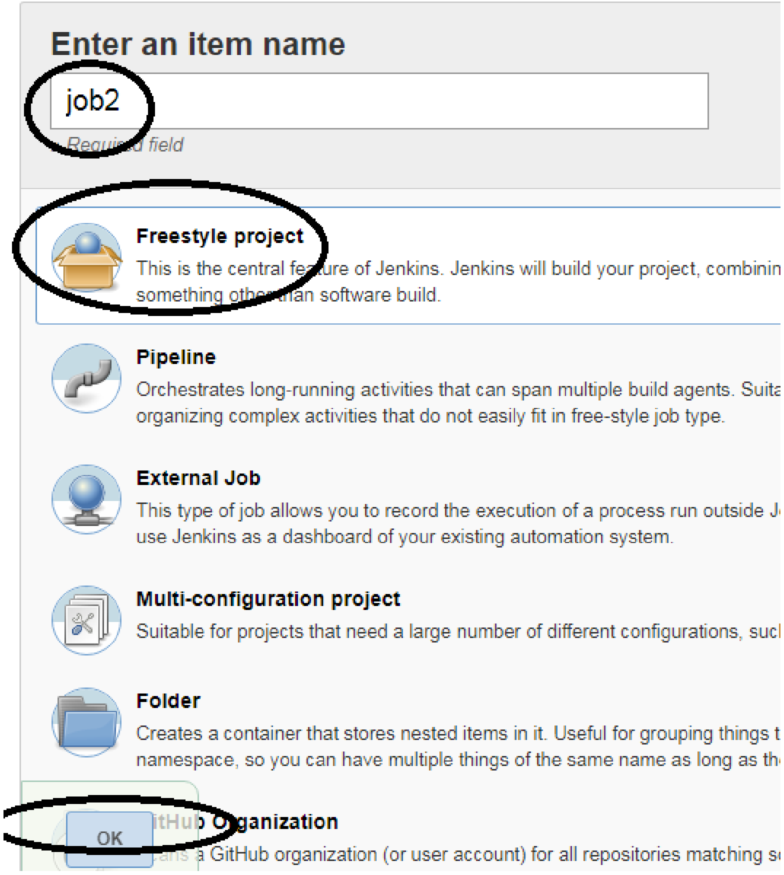
Finally the build part



```
ls -l
touch file1
echo "THIS IS TEST FILE CREATED IN JOB1" > file1
```

Save and apply.

B. Create another Job “Job2” and link this to Job1.



Enter an item name

job2

Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combinin
something other than software build.

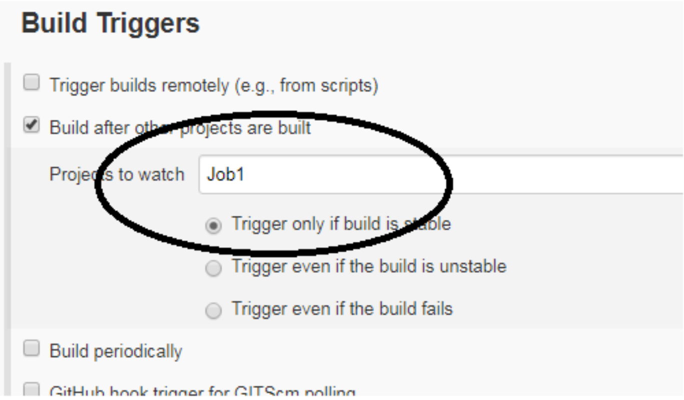
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suita
organizing complex activities that do not easily fit in free-style job type.

External Job
This type of job allows you to record the execution of a process run outside J
use Jenkins as a dashboard of your existing automation system.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such

Folder
Creates a container that stores nested items in it. Useful for grouping things t
namespace, so you can have multiple things of the same name as long as th

GitHub Organization
ans a GitHub organization (or user account) for all repositories matching s



Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☑ Build after other projects are built

Projects to watch: Job1

● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

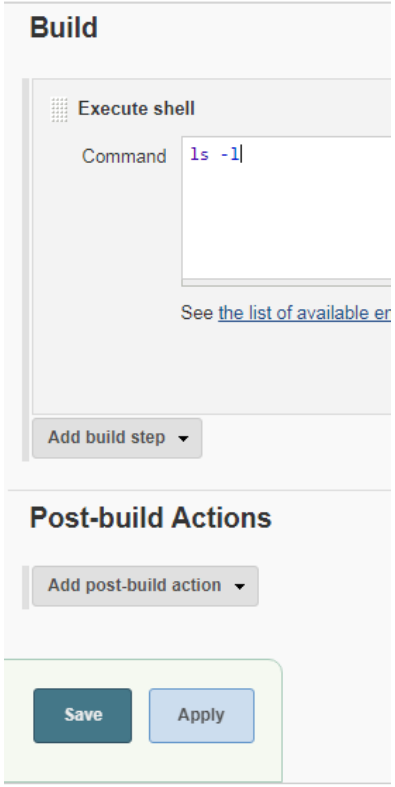
☐ Build periodically

☐ GitHub hook trigger for GITScm polling

on the Job2, under the Build Trigger option, Select the “Job1” to link it.

Which means, the Job2 would get triggered automatically, when ever the Job1 is completed successfully.

Then under the “Build” , give some sample build like below



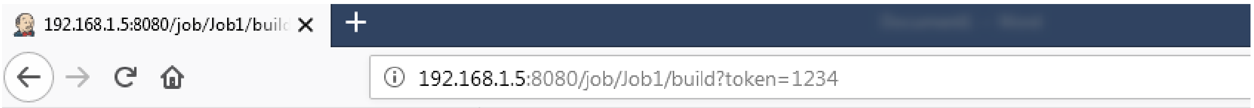
Command

```
ls -l
```

Save and Apply.

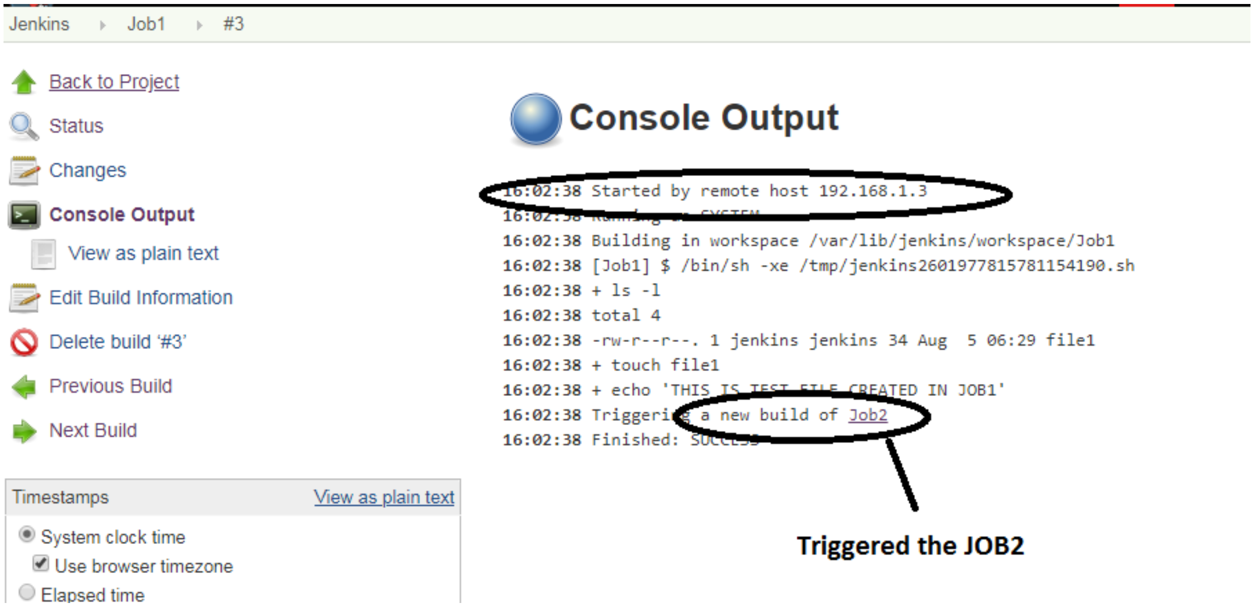
C. Test the above setup.

1. Trigger the build remotely.
   Use the link previously generated and run it on any browser other than the current Jenkins browser.



```
192.168.1.5:8080/job/Job1/build?token=1234
```

Let observe the Build that got triggered.
Under the console, it would say "started by remote" which shows clearly that it would triggered remotely.



● System clock time
☑ Use browser timezone
○ Elapsed time

```
16:02:38 Started by remote host 192.168.1.3
16:02:38 Running as SYSTEM
16:02:38 Building in workspace /var/lib/jenkins/workspace/Job1
16:02:38 [Job1] $ /bin/sh -xe /tmp/jenkins2601977815781154190.sh
16:02:38 + ls -l
16:02:38 total 4
16:02:38 -rw-r--r--. 1 jenkins jenkins 34 Aug  5 06:29 file1
16:02:38 + touch file1
16:02:38 + echo 'THIS IS TEST FILE CREATED IN JOB1'
16:02:38 Triggering a new build of Job2
16:02:38 Finished: SUCCESS
```

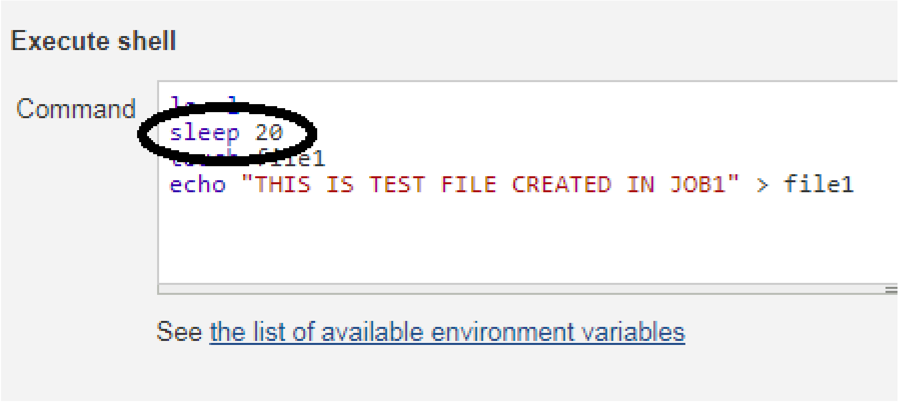
Triggered the JOB2

In turn, this job1 has triggered the "Job2" .

This is how you can link a job and create a chain of reactions of the jobs, which is the precise way that is done in the production.

2. Execute the builds for the same job simultaneously.

For this edit the build and add **“sleep 20”**



Execute shell

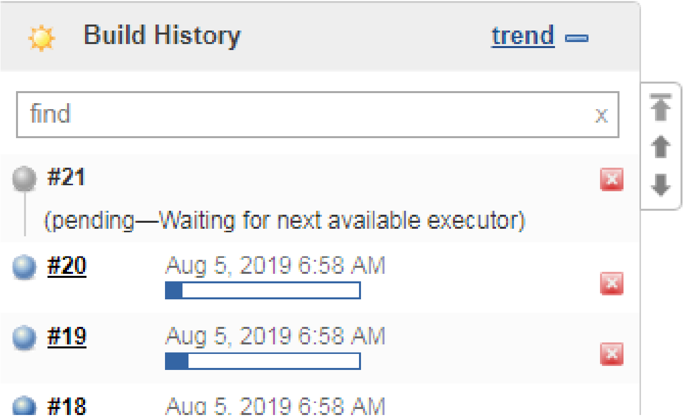
Command

```
ls -l
sleep 20
touch file1
echo "THIS IS TEST FILE CREATED IN JOB1" > file1
```

See the list of available environment variables

So that the build waits for 20 sec before it completes it.

Now, click on the build multiple times.



#21 (pending—Waiting for next available executor)

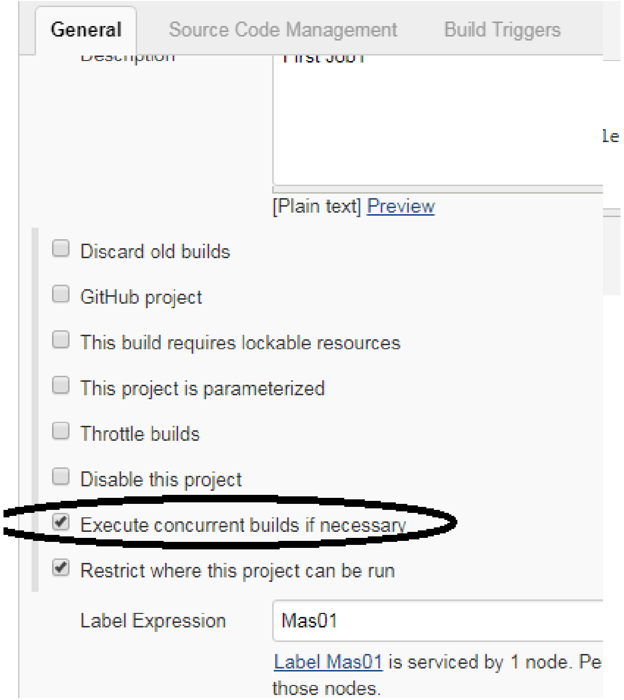
#20 Aug 5, 2019 6:58 AM

#19 Aug 5, 2019 6:58 AM

#18 Aug 5, 2019 6:58 AM

Two build for the same job is executed parallely but the 3rd build is waiting because the Jenkins master has the “executor” limitation of ‘2’.

Now, lets disable the “concurrent build” by unticking the below option.



[Plain text] Preview

☐ Discard old builds

☐ GitHub project

☐ This build requires lockable resources

☐ This project is parameterized

☐ Throttle builds

☐ Disable this project

☑ Execute concurrent builds if necessary

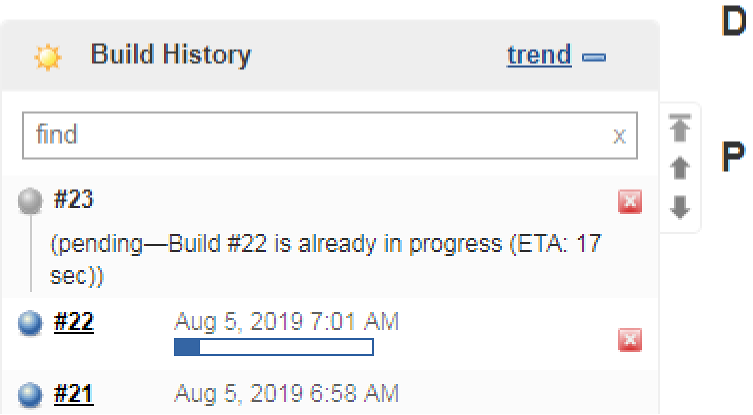
☑ Restrict where this project can be run

Label Expression: Mas01

Label Mas01 is serviced by 1 node. Pe those nodes.

Untick the one in the circle.

And now, if you try to click on the build multiple times.



#23 (pending—Build #22 is already in progress (ETA: 17 sec))

#22 Aug 5, 2019 7:01 AM

#21 Aug 5, 2019 6:58 AM

The wait message is very clear that the other build is in progress before it can start the next build.