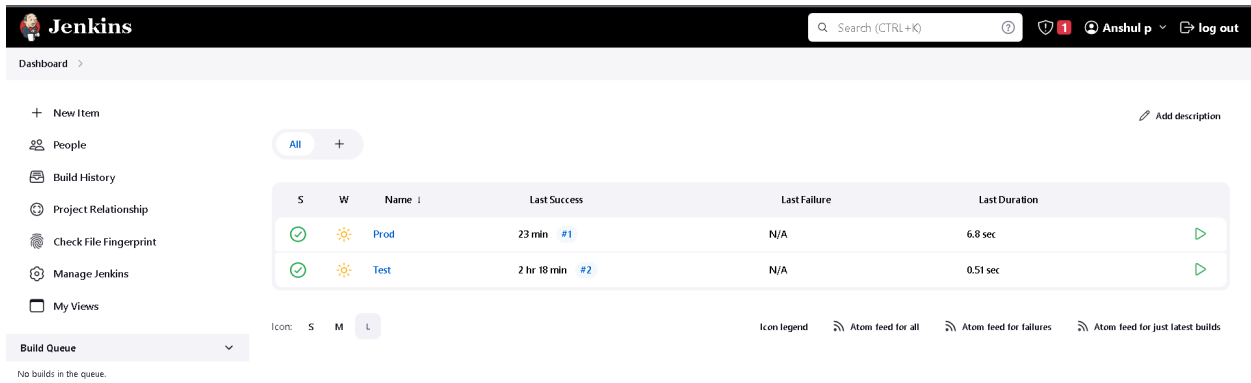


## Jenkins - 3

You have been asked to:

- Create a pipeline in jenkins
- Once push is made to “develop” a branch in git, trigger job “test”. This will copy git files to test node
- If test job is successful, then prod job should be triggered
- Prod jobs should copy files to prod node

We have just renamed job1>>test and job2>>prod to avoid confusion.

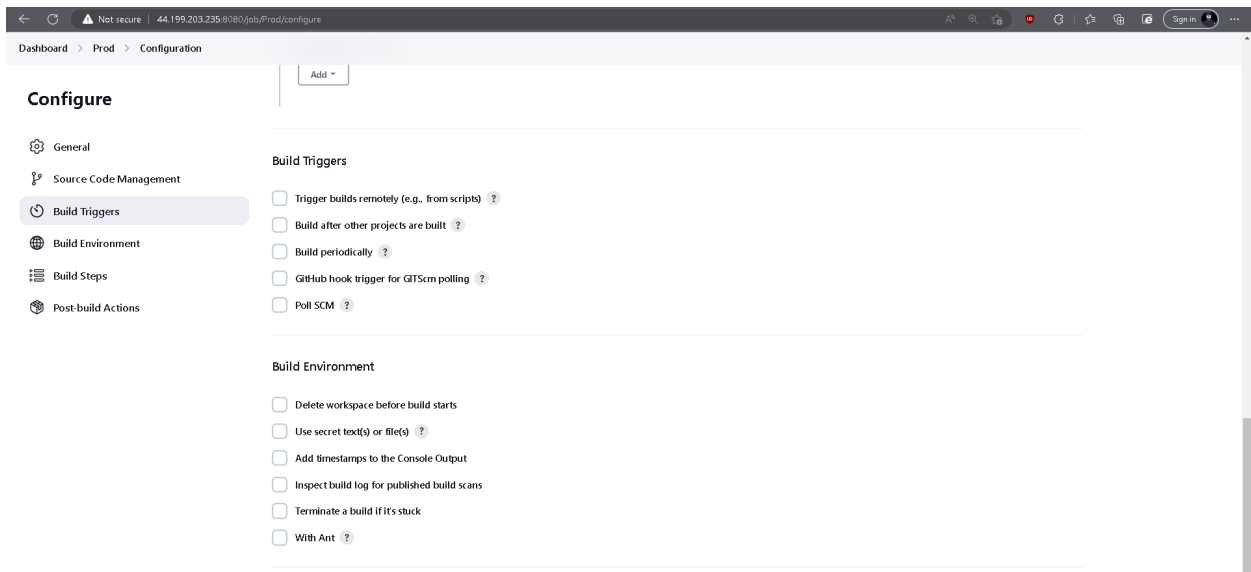


The screenshot shows the Jenkins Dashboard. The top navigation bar includes the Jenkins logo, a search bar, and user information (Anshul p). The left sidebar contains links for New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The main content area displays a table of jobs. The table has columns for S (Status), W (Webhook), Name, Last Success, Last Failure, and Last Duration. Two jobs are listed: 'Prod' and 'Test'. The 'Prod' job has a status of 'Success' (green checkmark) and a duration of 6.8 sec. The 'Test' job has a status of 'Success' (green checkmark) and a duration of 0.51 sec. Below the table, there are icons for S, M, and L, and a section for Atom feeds.

| S | W | Name | Last Success   | Last Failure | Last Duration |
|---|---|------|----------------|--------------|---------------|
| ✓ | ☀ | Prod | 23 min #1      | N/A          | 6.8 sec       |
| ✓ | ☀ | Test | 2 hr 18 min #2 | N/A          | 0.51 sec      |

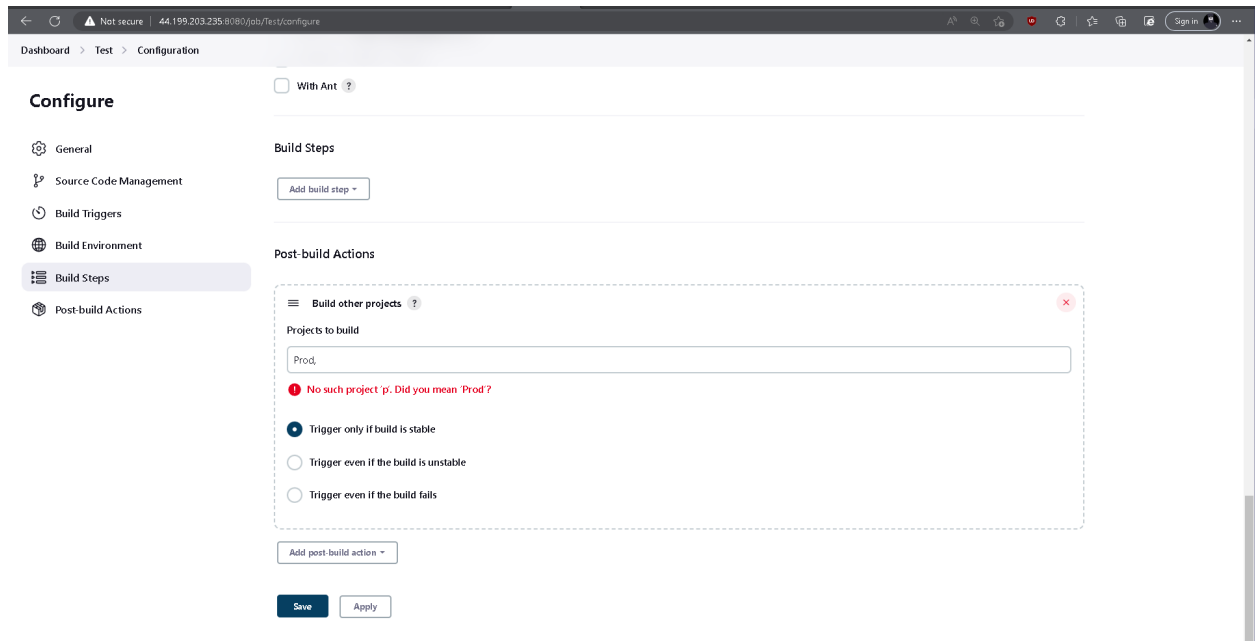
Now according to assignment, we need to create dependency such that until and unless ‘test’ is successfully completed ‘prod’ can’t run.

Now goto: prod job>>configure>>build trigger and uncheck ‘Github hook...’ to not let the job run automatically after changes are made in git.

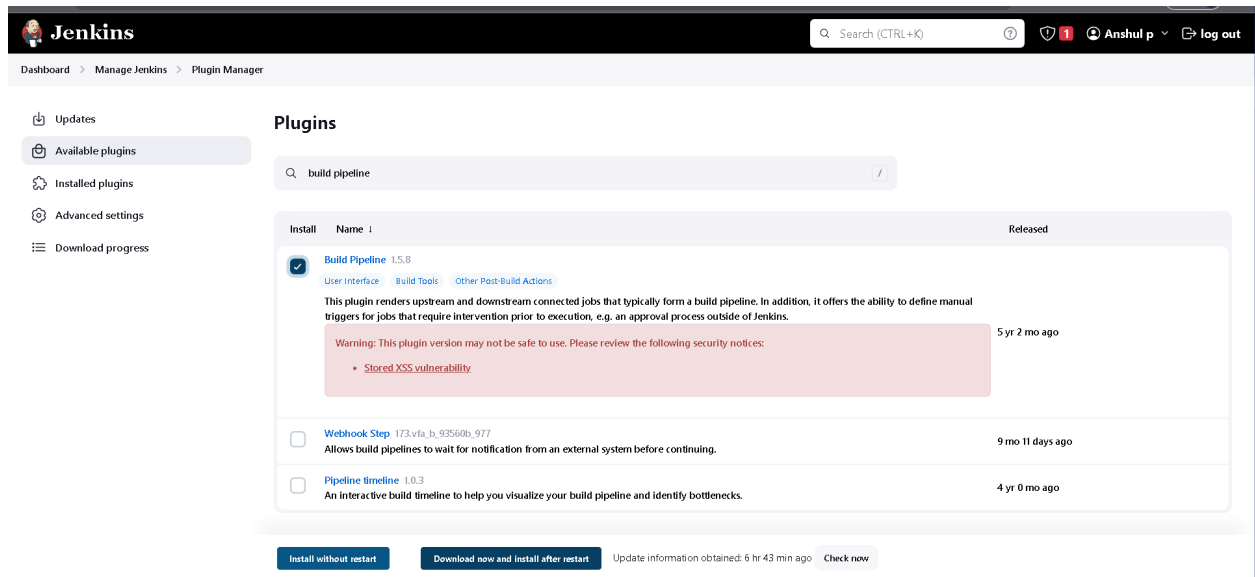


The screenshot shows the Jenkins Configuration page for the 'Prod' job. The left sidebar contains links for General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main content area is titled 'Configure' and has an 'Add +' button. Under the 'Build Triggers' section, there are several checkboxes: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling', and 'Poll SCM'. The 'GitHub hook trigger for GITScm polling' checkbox is unchecked. Under the 'Build Environment' section, there are several checkboxes: 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'. The 'With Ant' checkbox is unchecked.

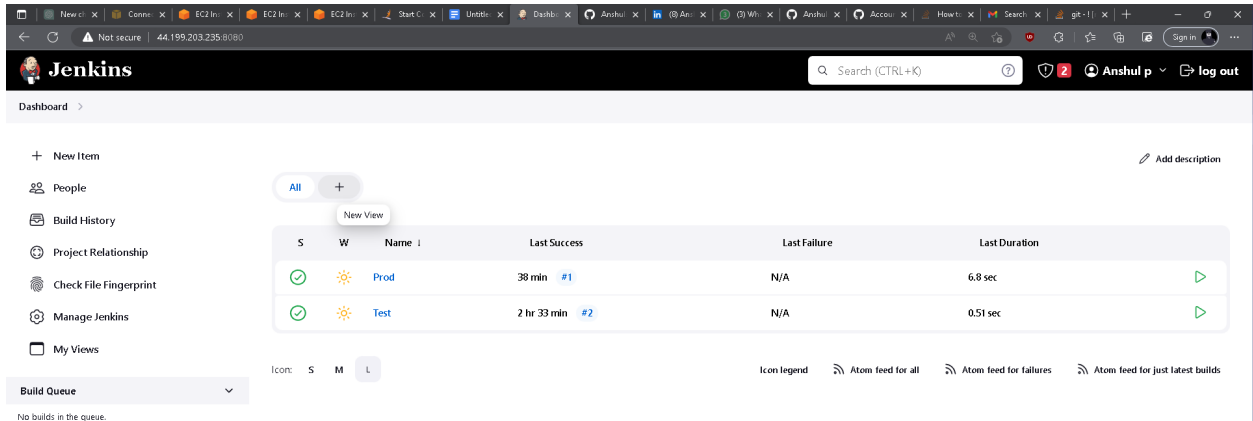
Now goto: test job>>configure>>post-build>>build-steps  
Add 'Prod' there.  
This means Prod is set as downstream to test.



Let us create a pipeline now.  
For that we need to download a plugin first.  
Goto: dashboard>>manage jenkins>>manage plugins>>available plugins  
Then search for “Build Pipeline” and install it.



## Now to build a new pipeline, choose New View.

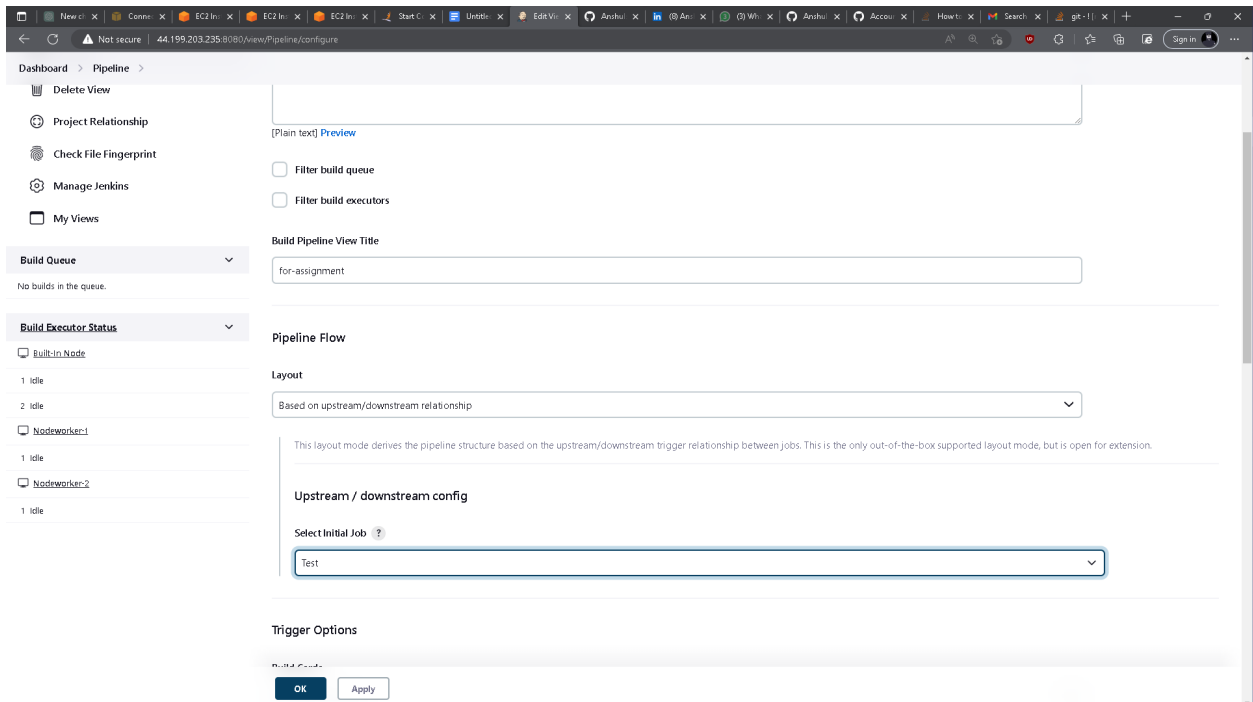


The screenshot shows the Jenkins Dashboard with a sidebar on the left containing links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area displays a table of pipeline views. A 'New View' button is visible above the table. The table has columns for 'S', 'W', 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. Two views are listed: 'Prod' and 'Test'.

| S | W | Name | Last Success   | Last Failure | Last Duration |
|---|---|------|----------------|--------------|---------------|
| ✓ | ☀ | Prod | 38 min #1      | N/A          | 6.8 sec       |
| ✓ | ☀ | Test | 2 hr 33 min #2 | N/A          | 0.51 sec      |

## Name it according to the pipeline view and click next.

## Then just change pipeline flow and select the initial job to Test.



The screenshot shows the Jenkins Pipeline configuration page. The left sidebar has 'Delete View', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area is titled 'Pipeline' and contains several sections: 'Build Pipeline View Title' with a text input field containing 'for-assignment', 'Pipeline Flow' with a 'Layout' dropdown set to 'Based on upstream/downstream relationship', and 'Upstream / downstream config' with a 'Select Initial Job' dropdown set to 'Test'. At the bottom, there are 'OK' and 'Apply' buttons.

## Pipeline is created.

The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is 44.199.203.235:8080/view/Pipeline/. The Jenkins logo and a search bar are at the top. Below the navigation bar, the page title is "Build Pipeline: for-assignment". A row of icons allows for actions: Trigger a Pipeline (Run), Pipeline History, Configure, Add Step, Delete, and Manage. The main content area displays two pipeline runs, both in a green "Success" state. The first run, labeled "#3 Test", occurred on Feb 23, 2023 at 5:39:18 PM and took 0:36 seconds. The second run, labeled "#3 Prod", occurred on Feb 23, 2023 at 5:39:03 PM and took 0:36 seconds. Both runs show a "console" icon and a "re-run" icon. The footer of the page includes the text "REST API" and "Jenkins 2.375.3".

Dashboard > Pipeline >

### Build Pipeline: for-assignment

Trigger a Pipeline Run Pipeline History Configure Add Step Delete Manage

**Pipeline**  
#2

**#3 Test**  
Feb 23, 2023 5:39:18 PM  
0:36 sec  
console re-run

**#3 Prod**  
Feb 23, 2023 5:39:03 PM  
0:36 sec  
console re-run

REST API Jenkins 2.375.3