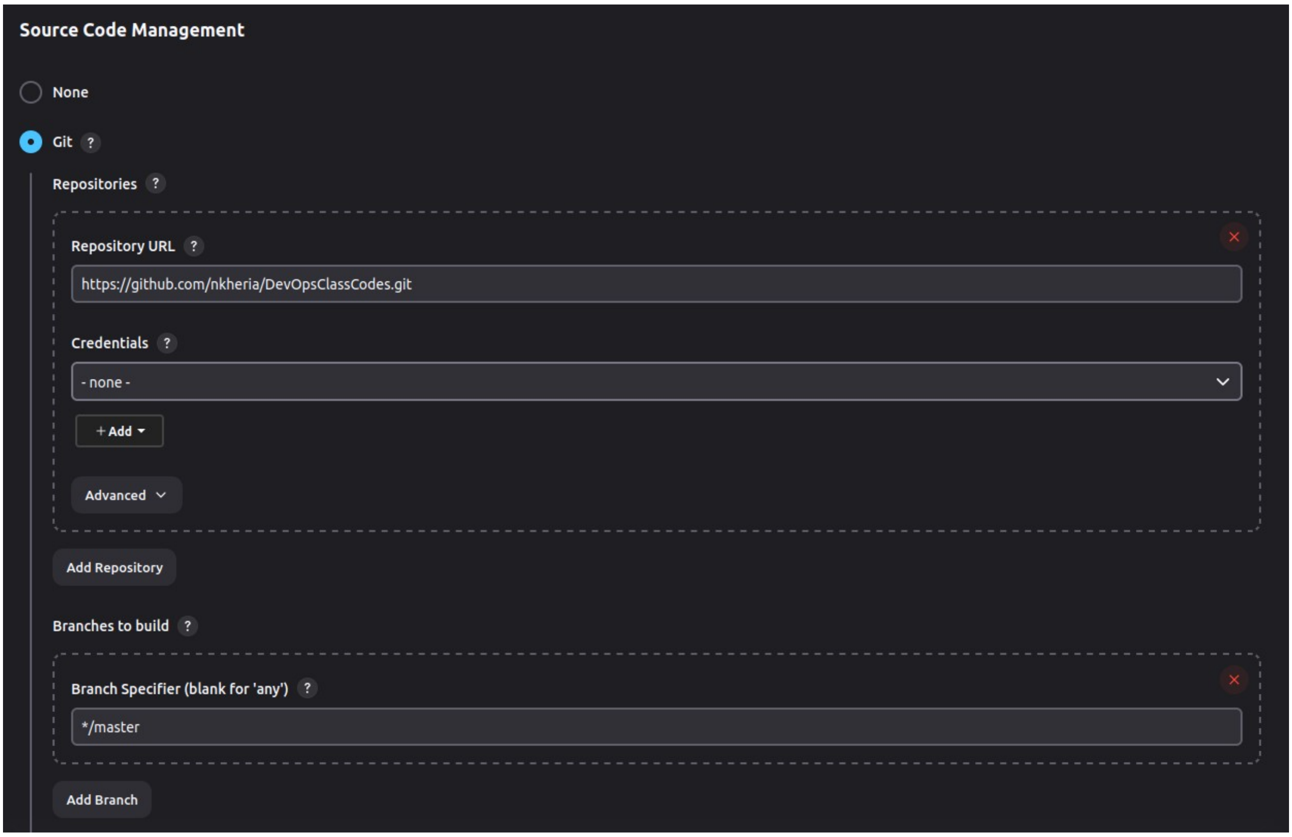
Jenkins

Day 11

1. We created a job named dev compile and test.

The purpose of the creation of the job dev compile was to invoke maven targets as the reffered github repository had a pom.xml file.
Here maven was created beforehand in the previous day (day_10) which was used here.

Fetching the code from github repo can be done with the help of SCM(Source Code Management)



Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

https://github.com/nkheria/DevOpsClassCodes.git

Credentials ?

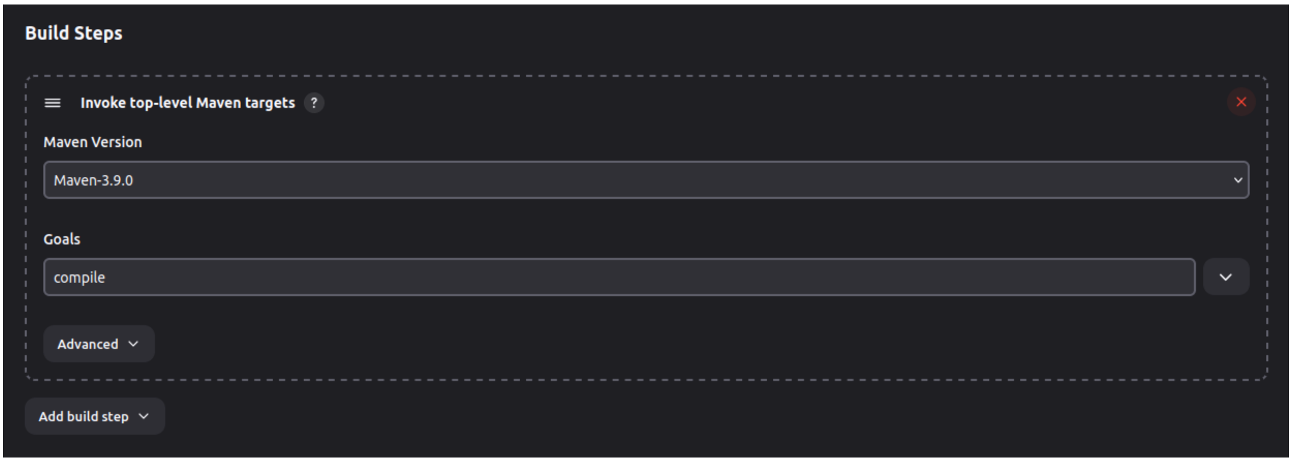
- none -

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

In the build steps we have to choose Invoke top-level maven targets and give the version of the maven.
Under goals we have to write compile which is fetched from the dependencies mentioned under pom.xml file in the github repo mentioned.



Build Steps

Invoke top-level Maven targets ?

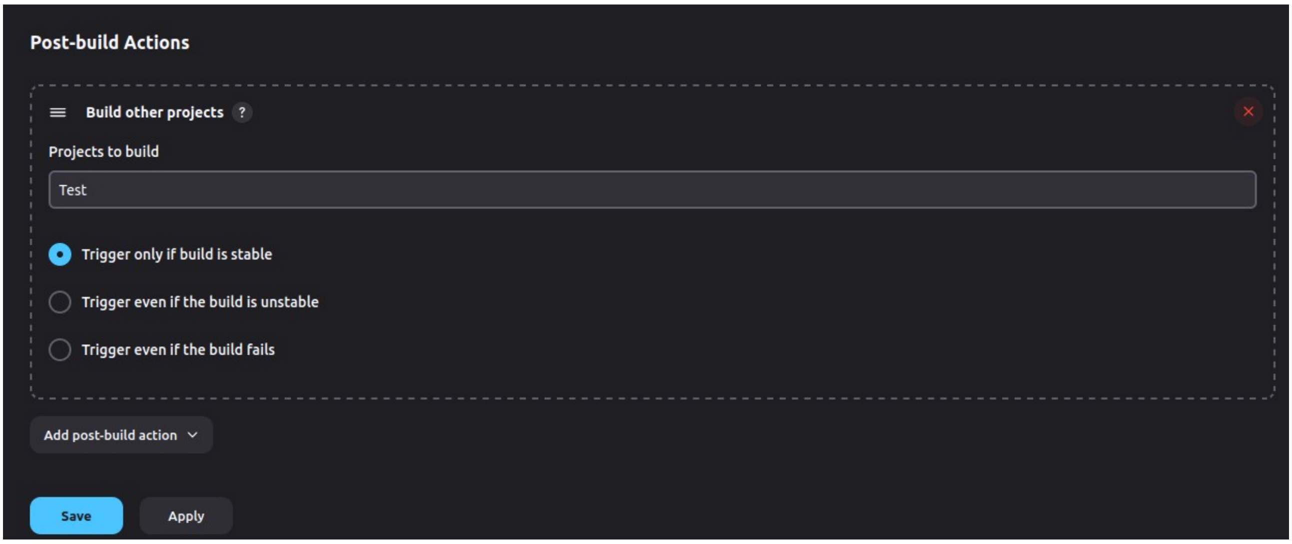
Maven Version

Maven-3.9.0

Goals

compile

Under post-build actions we have to choose build other projects for this we need to have a project named test created which can be done following the same steps and mentioning the same github repo link.
In Test job we won’t mention any Post-build actions.
By clicking the Trigger only if build is stable it will make sure that the Test job is build only when the Dev Compile has been succesfully build and is stable.



Post-build Actions

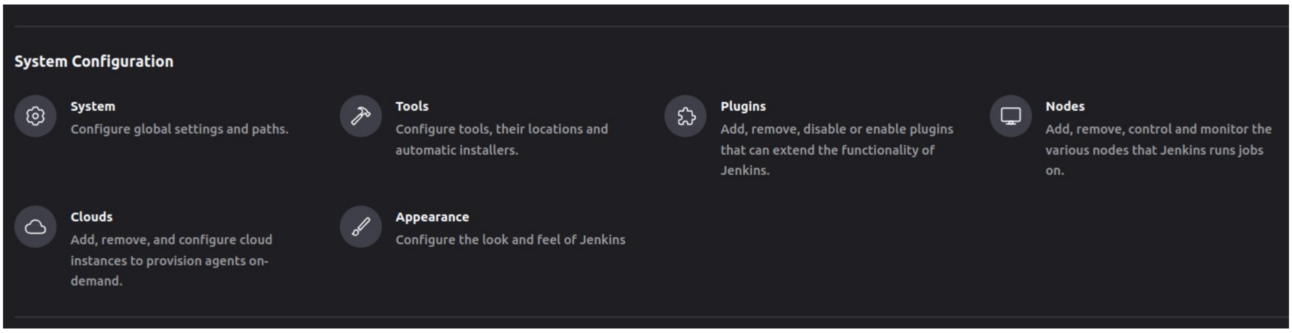
Build other projects ?

Projects to build
Test

● Trigger only if build is stable
○ Trigger even if the build is unstable
○ Trigger even if the build fails

Save the changes.

2. Install a new plugin called Build Pipeline.
Navigate to manage jenkins ---> System Configurations-----> Plugins



System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

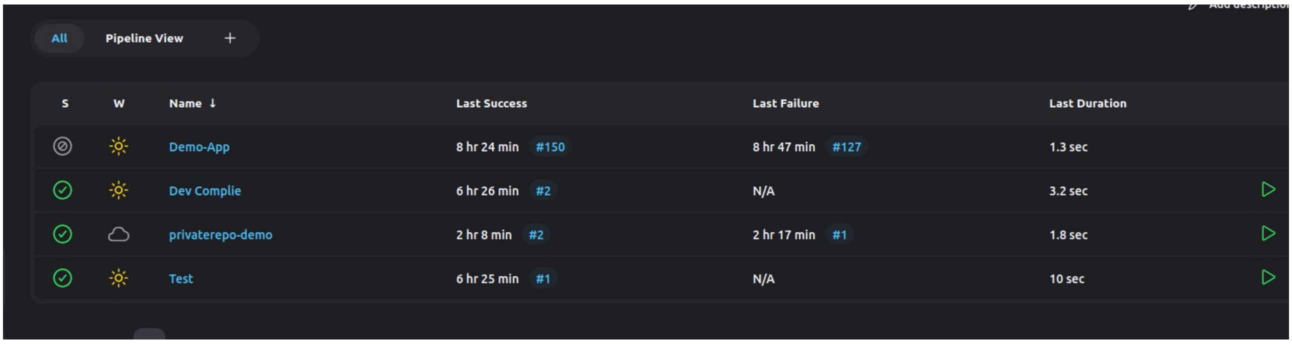
Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Appearance
Configure the look and feel of Jenkins

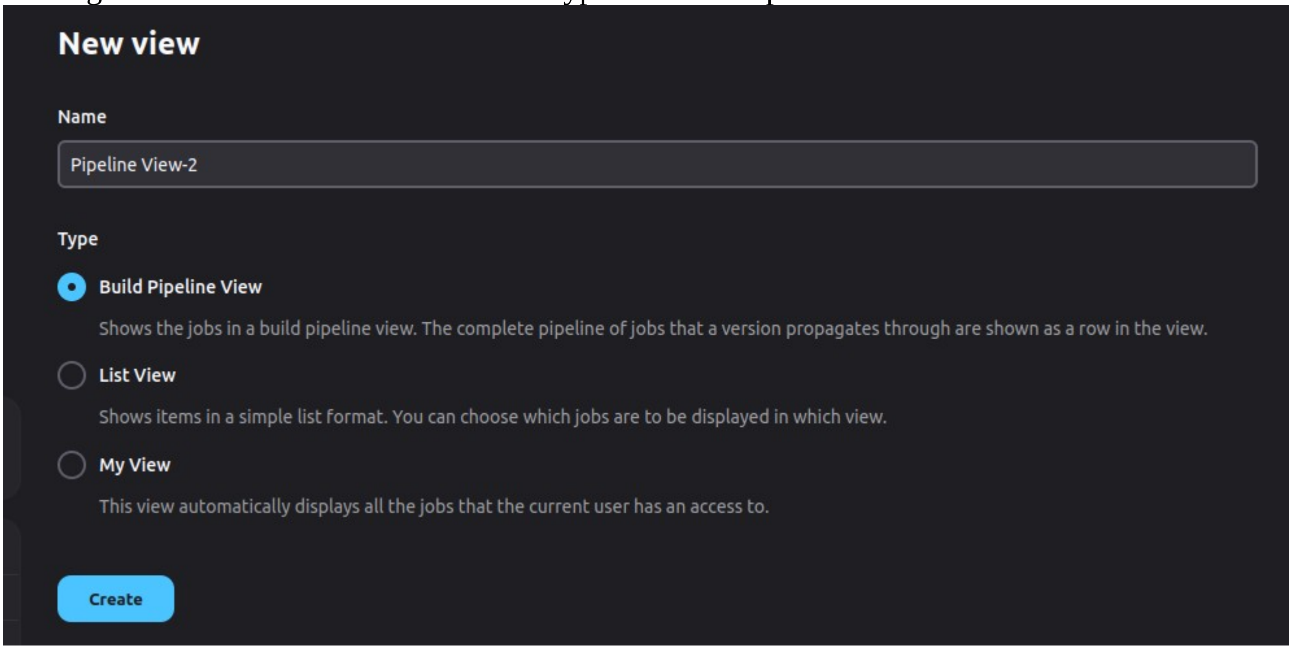
Go to available plugins and search for Build Pipeline.
Select the install checkbox and install the plugin.
Restart Jenkins as directed.

Now under dashboard select the + sign to use the plugin.



S	W	Name ↓	Last Success	Last Failure	Last Duration	
⊘	☀	Demo-App	8 hr 24 min #150	8 hr 47 min #127	1.3 sec	
✓	☀	Dev Complie	6 hr 26 min #2	N/A	3.2 sec	▷
✓	☁	privaterepo-demo	2 hr 8 min #2	2 hr 17 min #1	1.8 sec	▷
✓	☀	Test	6 hr 25 min #1	N/A	10 sec	▷

There a New View tab will open
There give a name for the view and select Type as Build Pipeline View.



New view

Name

Pipeline View-2

Type

● Build Pipeline View

Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

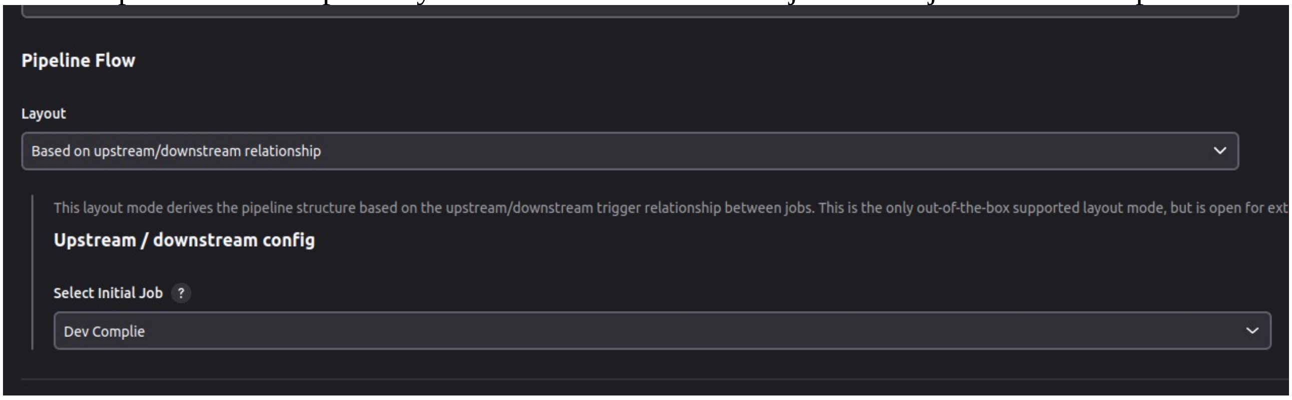
○ List View

Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

○ My View

This view automatically displays all the jobs that the current user has an access to.

Under Pipeline Flow keep the layout as it is and under initial job set the job as Dev Compile



Pipeline Flow

Layout

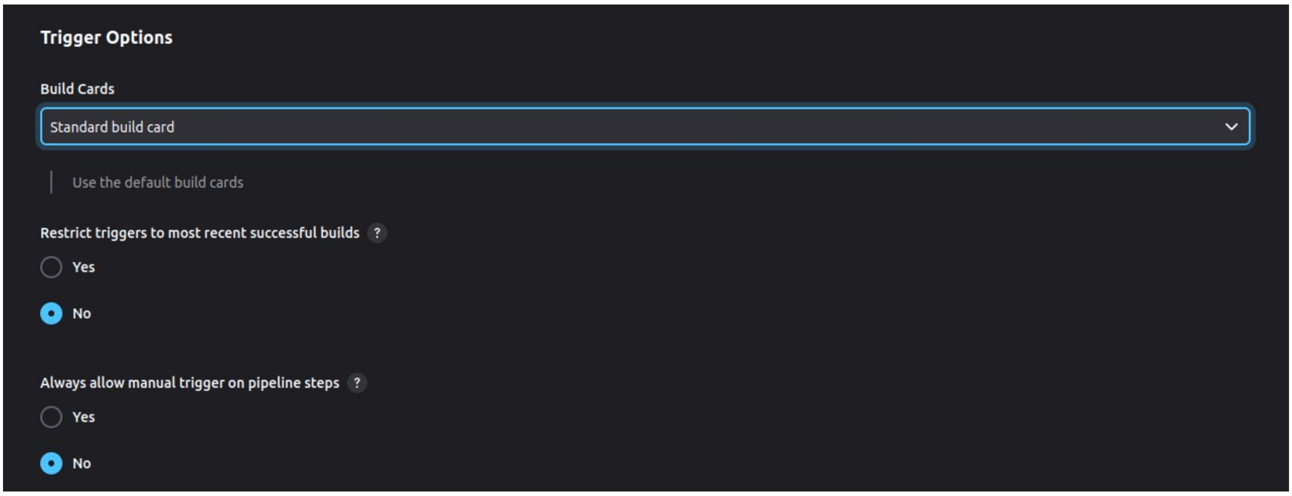
Based on upstream/downstream relationship

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for ext

Upstream / downstream config

Select Initial Job ?

Dev Complie



Trigger Options

Build Cards

Standard build card

Use the default build cards

Restrict triggers to most recent successful builds ?

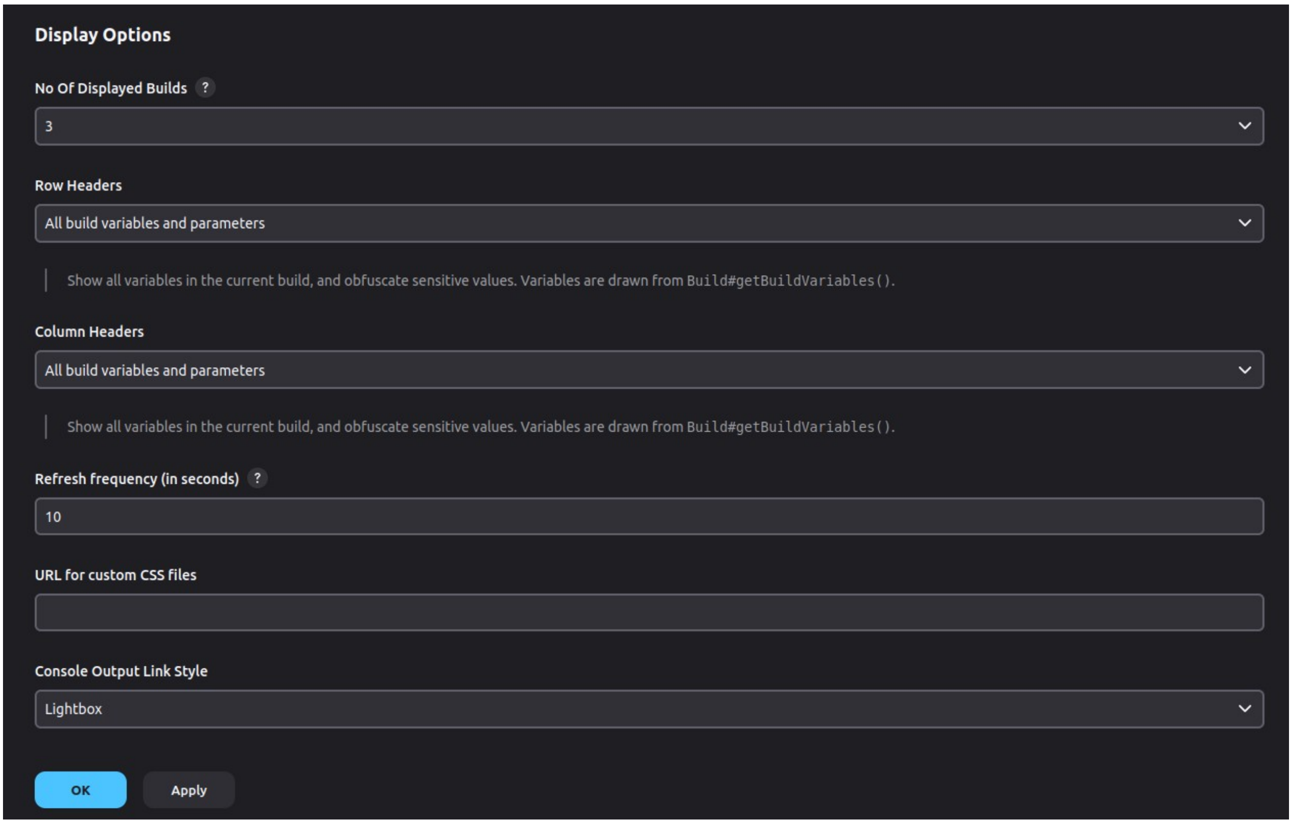
○ Yes

● No

Always allow manual trigger on pipeline steps ?

○ Yes

● No



Display Options

No Of Displayed Builds ?

3

Row Headers

All build variables and parameters

Show all variables in the current build, and obfuscate sensitive values. Variables are drawn from `Build#getBuildVariables()`.

Column Headers

All build variables and parameters

Show all variables in the current build, and obfuscate sensitive values. Variables are drawn from `Build#getBuildVariables()`.

Refresh frequency (in seconds) ?

10

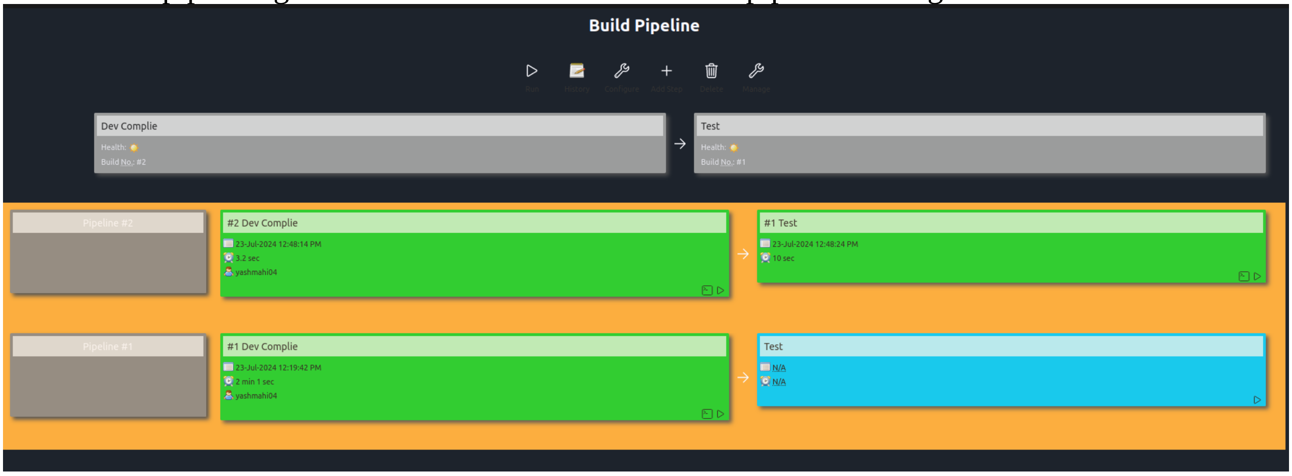
URL for custom CSS files

Console Output Link Style

Lightbox

Click OK to create the view.

To view the pipeline go to the dashboard and name of the pipeline view given will be visible



Build Pipeline

Dev Complie
Health:
Build No.: #2

Test
Health:
Build No.: #1

Pipeline #2

#2 Dev Complie
23-Jul-2024 12:48:14 PM
3.2 sec
yashmahi04

#1 Test
23-Jul-2024 12:48:24 PM
10 sec

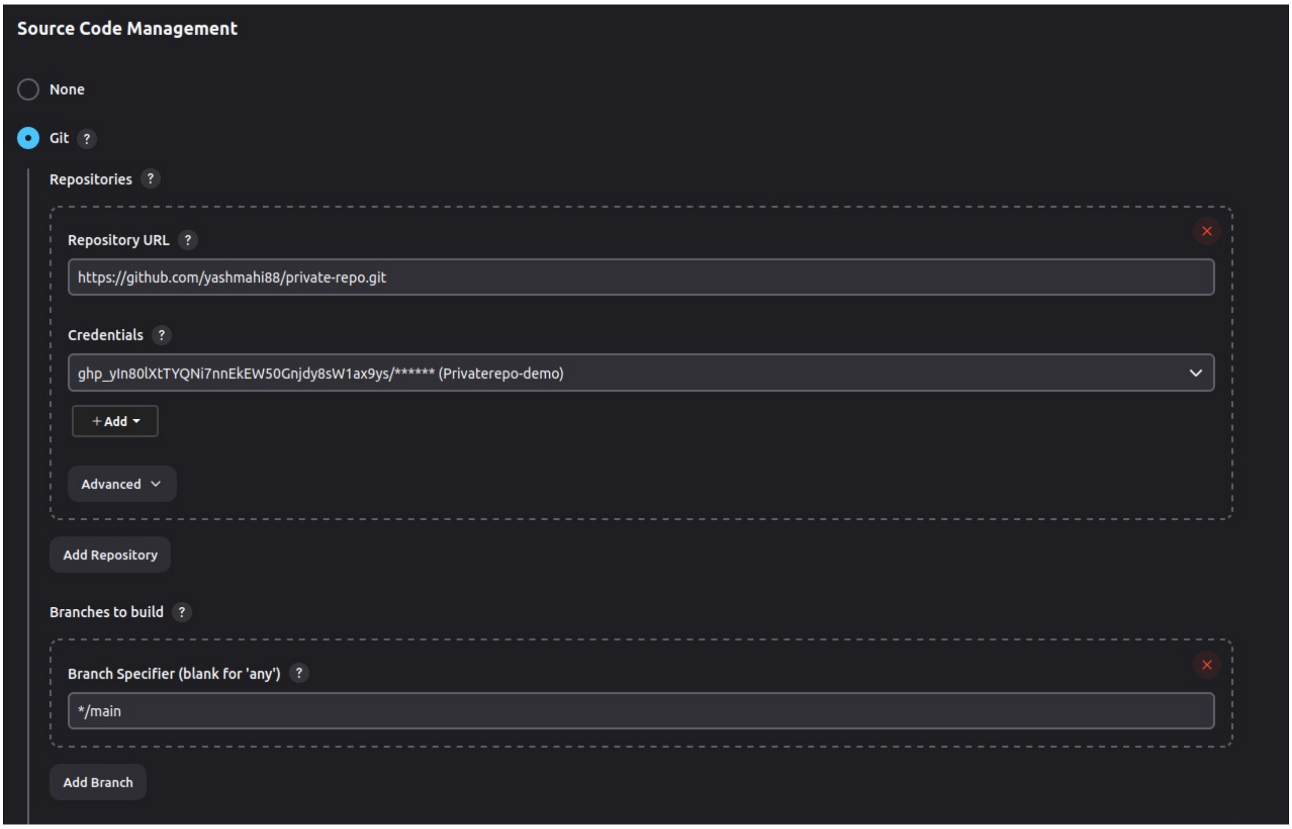
Pipeline #1

#1 Dev Complie
23-Jul-2024 12:19:42 PM
2 min 1 sec
yashmahi04

Test
N/A
N/A

3. Private repo in SCM.

Create a new job named it as privaterepo-demo.
Created a github repo whose visibilty status has been kept to private and that has to be accessed through the credentials.



Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

https://github.com/yashmahi88/private-repo.git

Credentials ?

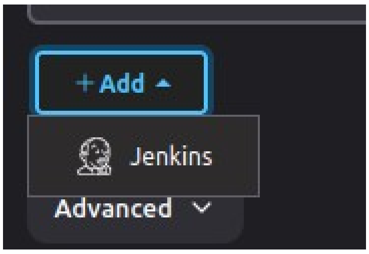
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/****** (Privaterepo-demo)

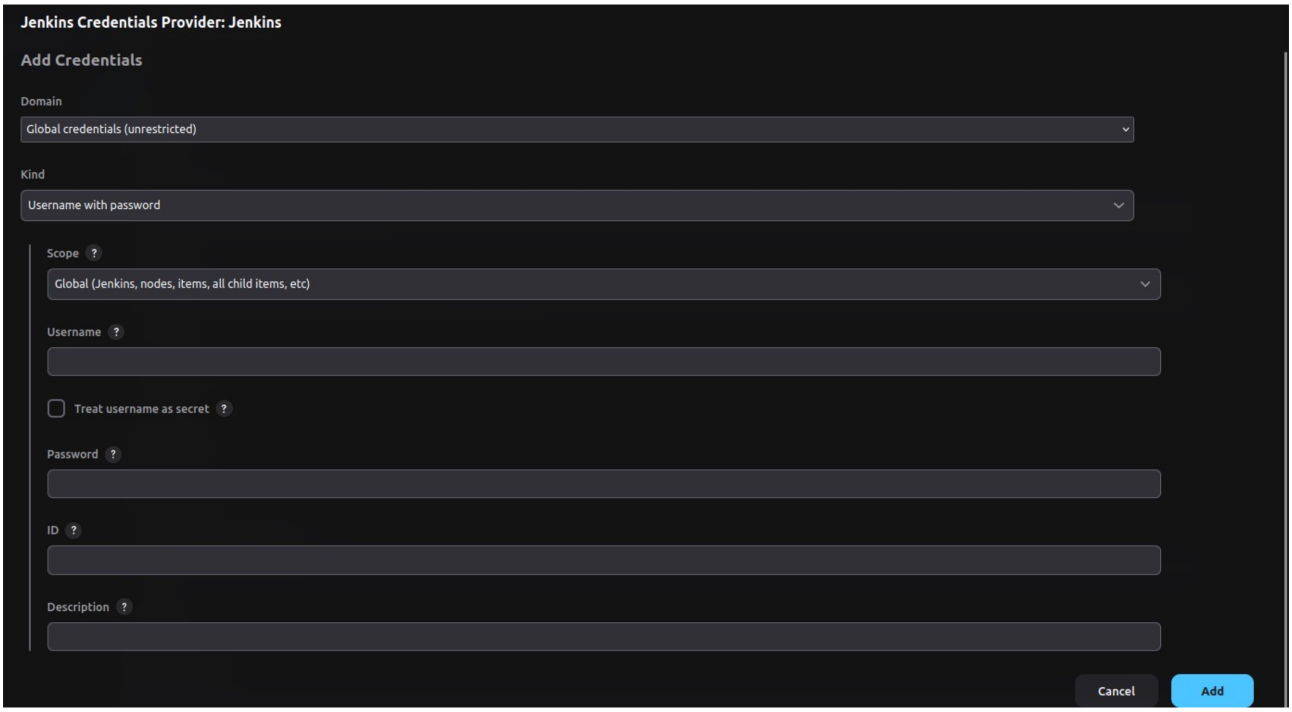
Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

In order to add new credentials, generate a token using github token generation service.
Use the add button.





Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username:

□ Treat username as secret

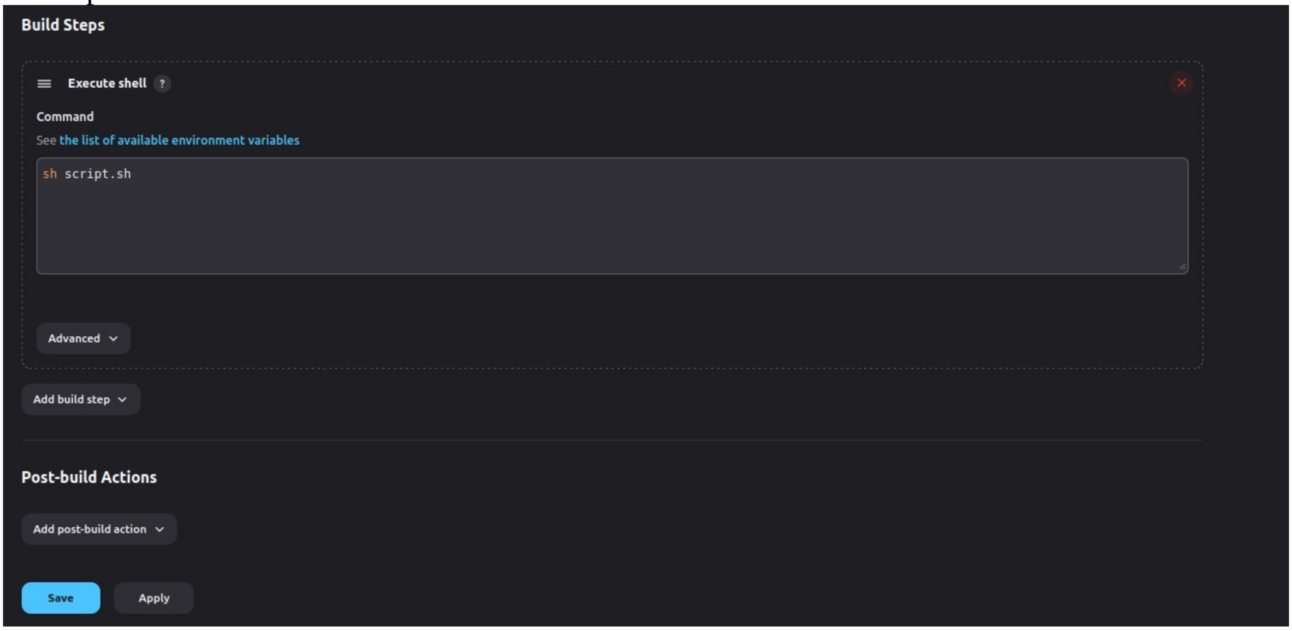
Password:

ID:

Description:

Under username and password copy the token that has been created through github.

As I have used a simple shell script code I selected execute shell and given the command as sh script.sh



Build Steps

Execute shell

Command

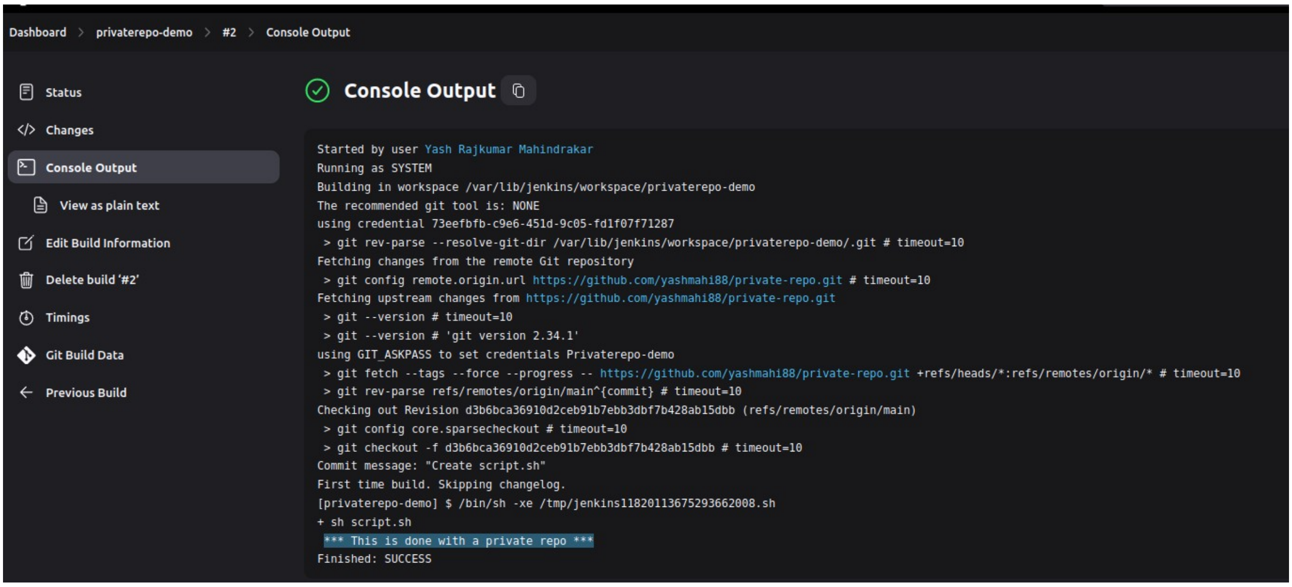
See the list of available environment variables

```
sh script.sh
```

Post-build Actions

After building and tapping on the console output we can see the output as

*** This is done with a private repo ***



Console Output

```
Started by user Yash Rajkumar Mahindrakar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/privaterepo-demo
The recommended git tool is: NONE
using credential 73eefbfb-c9e6-451d-9c05-fd1f07f71287
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/privaterepo-demo/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/yashmahi88/private-repo.git # timeout=10
Fetching upstream changes from https://github.com/yashmahi88/private-repo.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials Privaterepo-demo
 > git fetch --tags --force --progress -- https://github.com/yashmahi88/private-repo.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision d3b6bca36910d2ceb91b7ebb3dbf7b428ab15dbb (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f d3b6bca36910d2ceb91b7ebb3dbf7b428ab15dbb # timeout=10
Commit message: "Create script.sh"
First time build. Skipping changelog.
[privaterepo-demo] $ /bin/sh -xe /tmp/jenkins11820113675293662008.sh
+ sh script.sh
 *** This is done with a private repo ***
Finished: SUCCESS
```