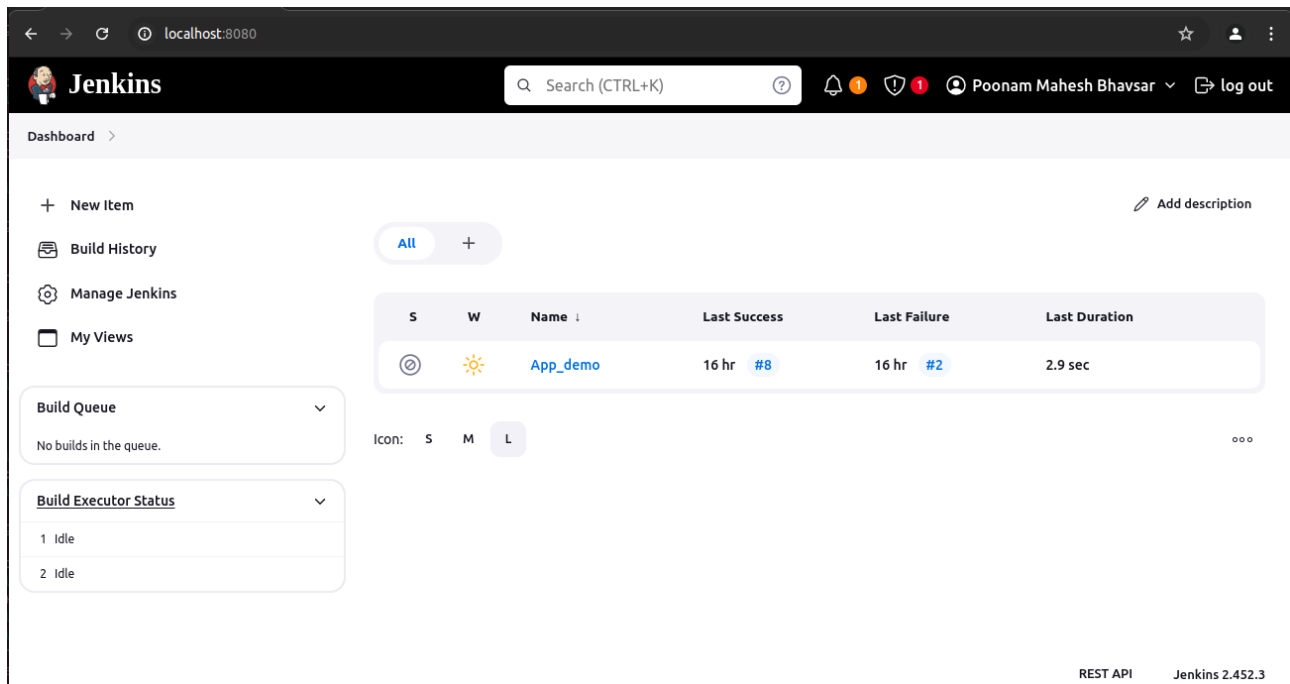
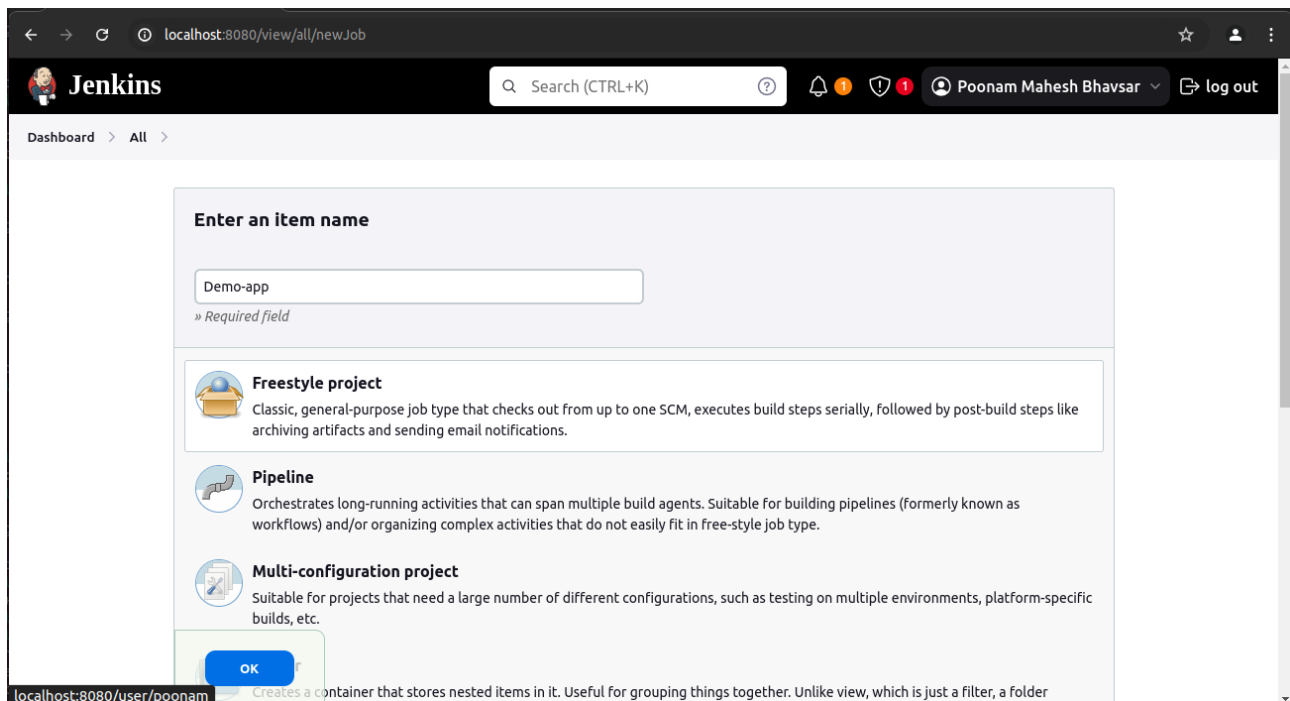


Jenkins Basic Freestyle project

- Jenkins Dashboard



- By clicking on **New Item** we can create a new project and then we can see the following screen

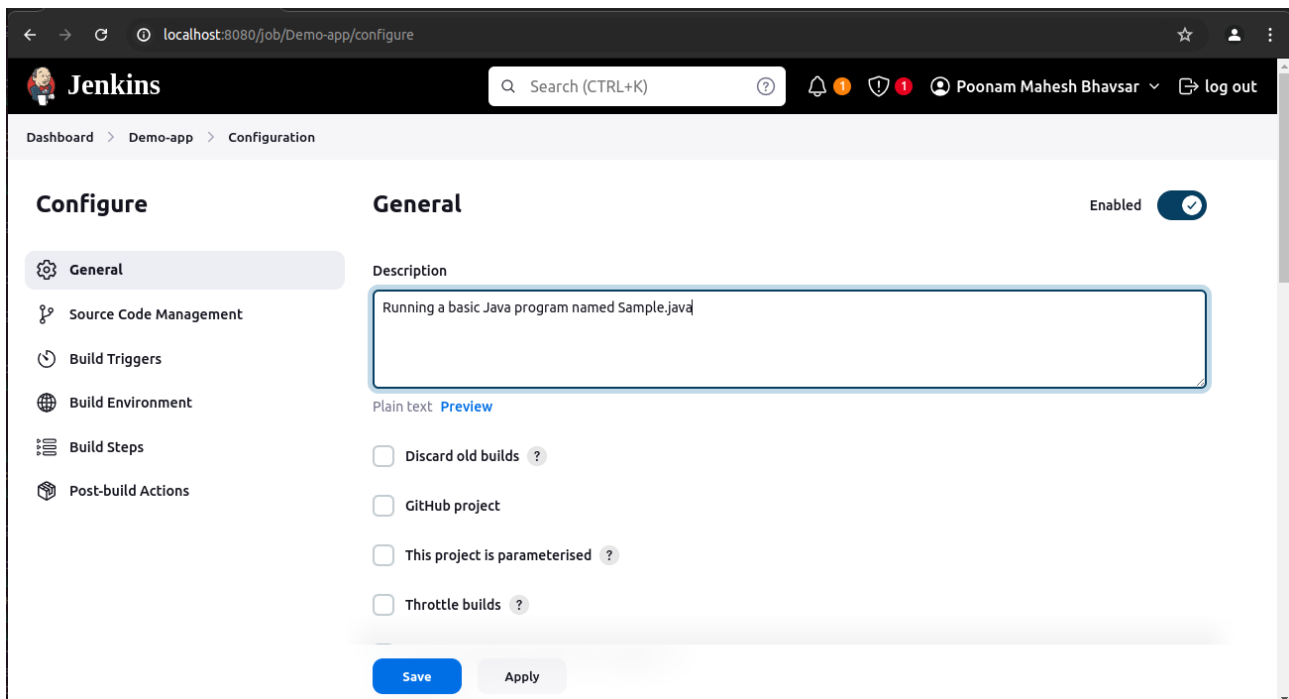


- So here I have created this project named Demo_app.

Then we can select which type of project it is whether it is Freestyle project or a Pipeline or a Multi-configuration project and so on.

Then we can click on **OK**.

- Then we will need to do configuration of our project.



The screenshot shows the Jenkins configuration page for a job named 'Demo-app'. The 'General' tab is selected in the left sidebar. The 'Description' field contains the text 'Running a basic Java program named Sample.java'. Below the description, there are four unchecked checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterised', and 'Throttle builds'. At the bottom, there are 'Save' and 'Apply' buttons. The top navigation bar shows the user 'Poonam Mahesh Bhavsar' and a 'log out' link.

localhost:8080/job/Demo-app/configure

Jenkins

Search (CTRL+K)

Poonam Mahesh Bhavsar log out

Dashboard > Demo-app > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

General

Enabled

Description

Running a basic Java program named Sample.java

Plain text Preview

☐ Discard old builds

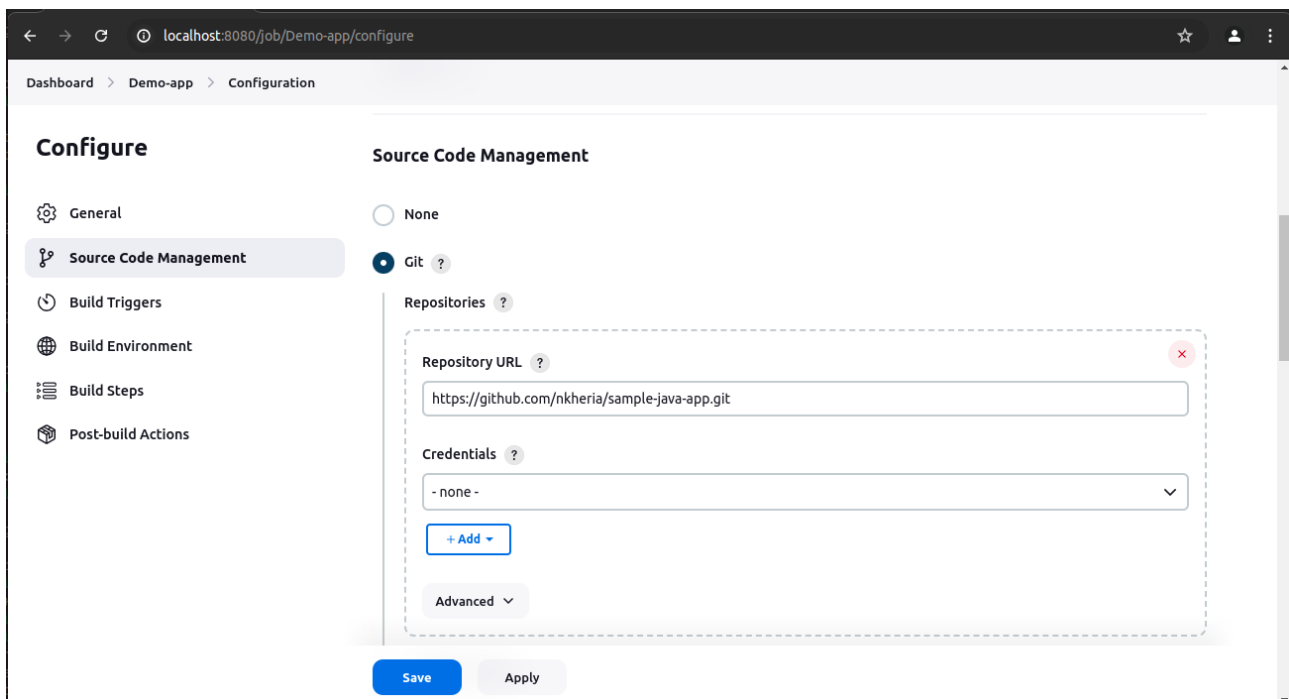
☐ GitHub project

☐ This project is parameterised

☐ Throttle builds

Save Apply

Add a description for your project.



The screenshot shows the Jenkins configuration page for the same job, but with the 'Source Code Management' tab selected. The 'Git' option is selected under 'Source Code Management'. In the 'Repositories' section, the 'Repository URL' field contains 'https://github.com/nkheria/sample-java-app.git' and the 'Credentials' dropdown is set to '- none -'. There is a '+ Add' button and an 'Advanced' dropdown. At the bottom, there are 'Save' and 'Apply' buttons.

localhost:8080/job/Demo-app/configure

Jenkins

Search (CTRL+K)

Poonam Mahesh Bhavsar log out

Dashboard > Demo-app > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Source Code Management

None

Git

Repositories

Repository URL

https://github.com/nkheria/sample-java-app.git

Credentials

- none -

+ Add

Advanced

Save Apply

In source code management, add Git repository path

localhost:8080/job/Demo-app/configure

Dashboard > Demo-app > Configuration

Configure

- General
- Source Code Management
- Build Triggers**
- Build Environment
- Build Steps
- Post-build Actions

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Build Triggers

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

Save Apply

- We can add which branch do build

localhost:8080/job/Demo-app/configure

Dashboard > Demo-app > Configuration

Configure

- General
- Source Code Management
- Build Triggers**
- Build Environment
- Build Steps
- Post-build Actions

Build Triggers

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

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

/ * * * *

⚠ Spread load evenly by using '*/2 * * * *' rather than '*/* * * * *'

Would last have run at Tuesday, 23 July, 2024 at 10:36:15 AM India Standard Time; would next run at Tuesday, 23 July, 2024 at 10:36:15 AM India Standard Time.

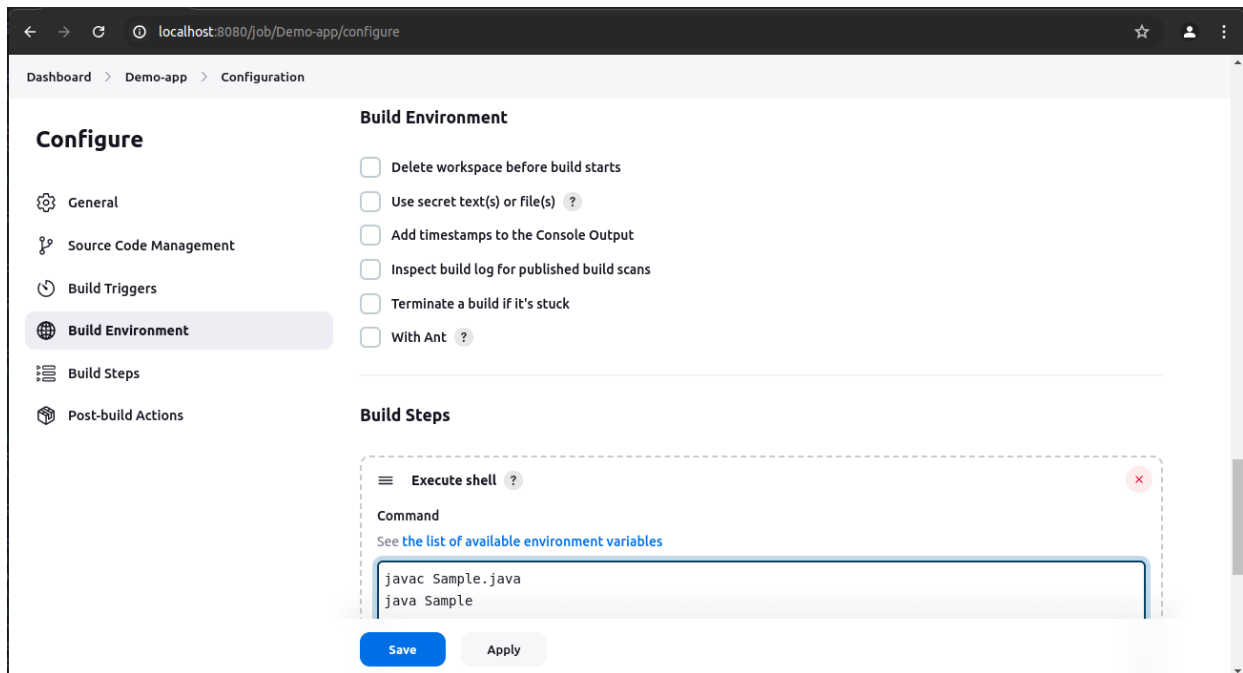
☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

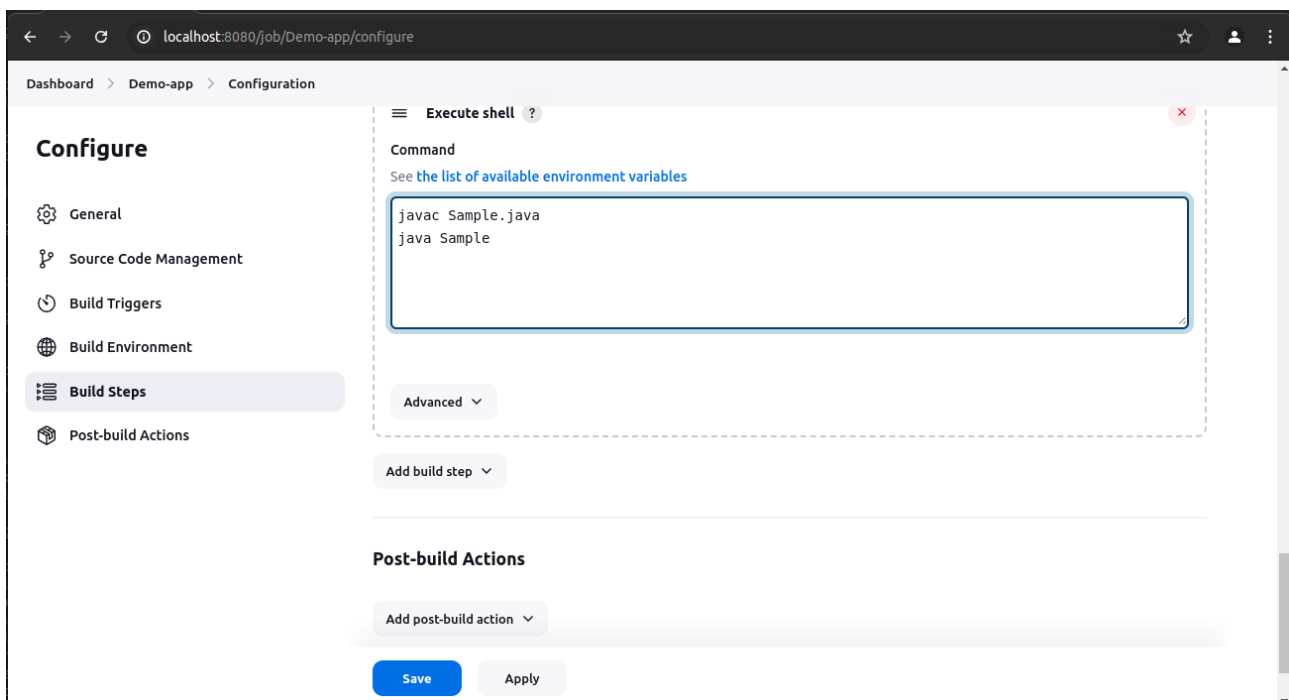
Build Environment

Save Apply

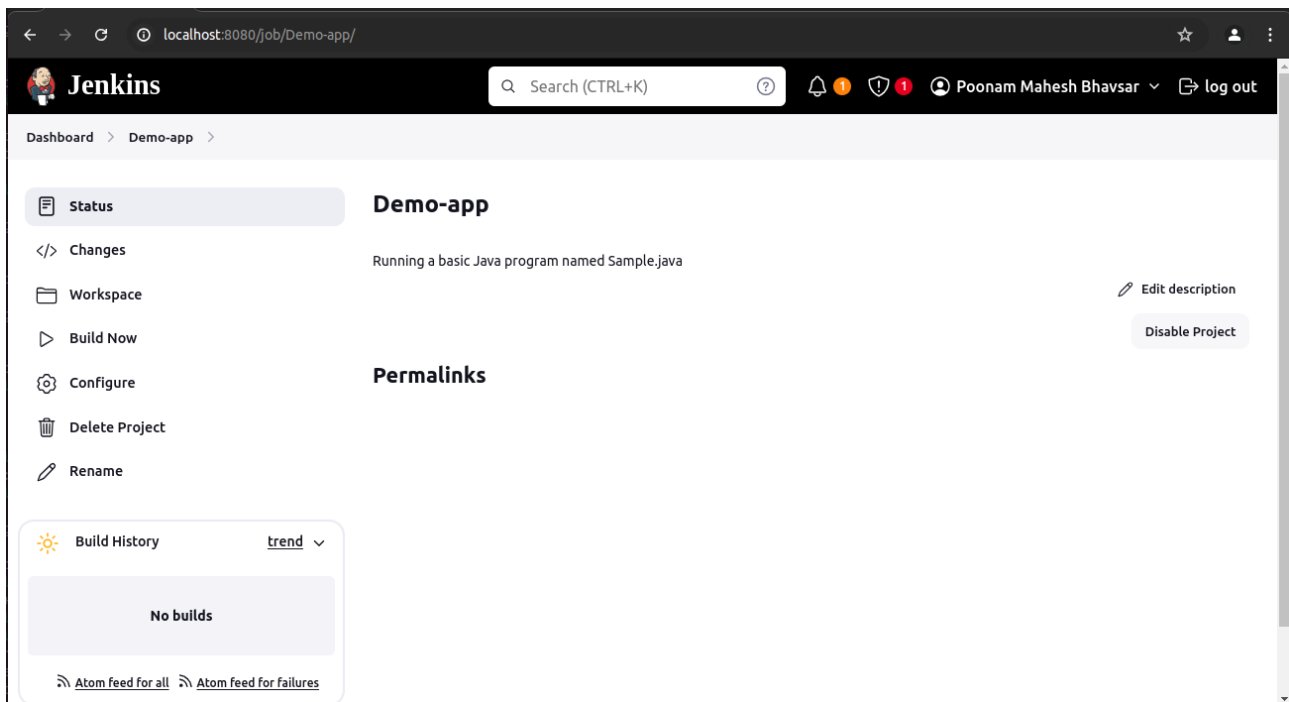
- Build Triggers and schedule the build.



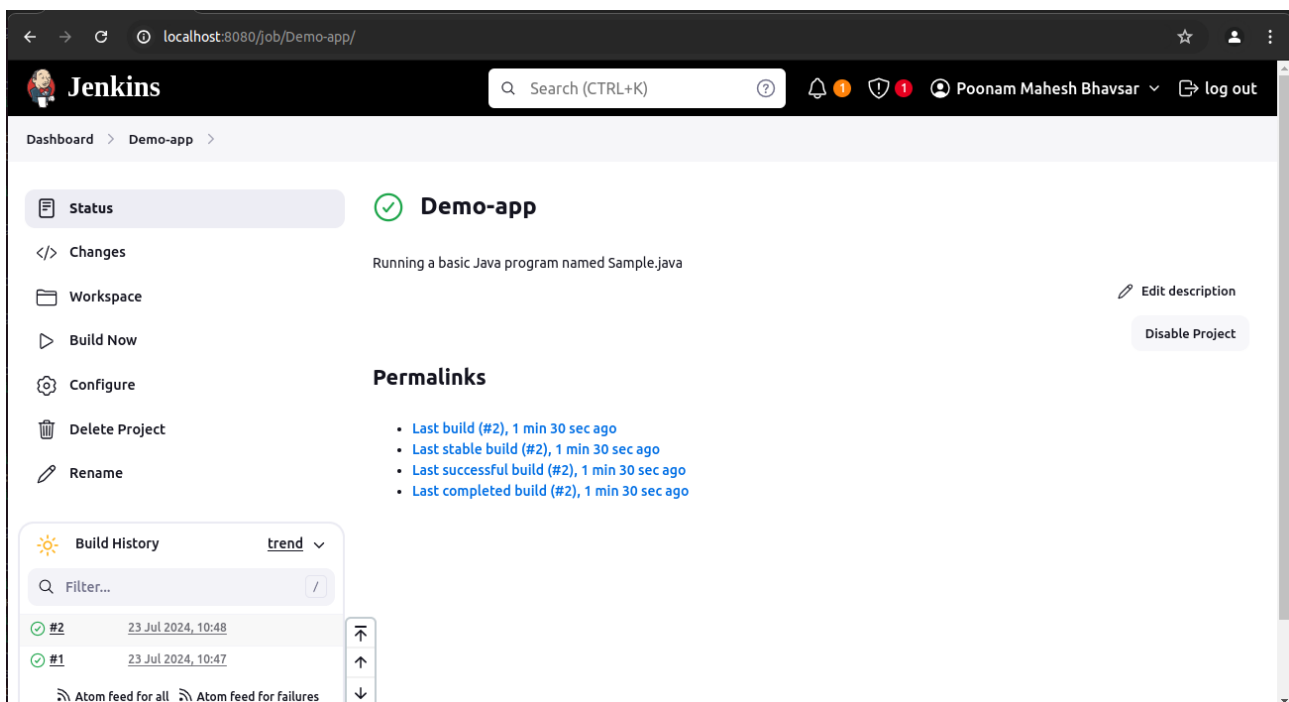
- Add **Build Environment** and **Build Steps**



- We can also add **post build actions** if needed. Then **Save**.



Once created a project you can build the project by clicking on **Build Now**



We can see the **Build History** of the project.

localhost:8080/job/Demo-app/1/

Jenkins

Search (CTRL+K) Poonam Mahesh Bhavsar log out

Dashboard > Demo-app > #1

Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Git Build Data

✓ #1 (23 Jul 2024, 10:47:22)

Keep this build forever

Add description Started 33 sec ago Took 1.6 sec

No changes.

Started by user Poonam Mahesh Bhavsar

This run spent:

- 14 ms waiting;
- 1.6 sec build duration;
- 1.6 sec total from scheduled to completion.

Revision: 7a0556222eb7f62251ed9fb890289f5ad27f0dae
Repository: <https://github.com/nkheria/sample-java-app.git>

- refs/remotes/origin/master

REST API Jenkins 2.452.3

The **Build Status** of the project

Console output

The screenshot shows the Jenkins web interface at localhost:8080/job/Demo-app/1/console. The left sidebar contains navigation links: Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Delete build '#1', Timings, Git Build Data, and Next Build. The main area displays the console output for build #1, which is a successful build. The output text is as follows:

```
Started by user Poonam Mahesh Bhavsar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Demo-app
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/nkheria/sample-java-app.git
> git init /var/lib/jenkins/workspace/Demo-app # timeout=10
Fetching upstream changes from https://github.com/nkheria/sample-java-app.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/nkheria/sample-java-app.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/nkheria/sample-java-app.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 7a0556222eb7f62251ed9fb890289f5ad27f0dae (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 7a0556222eb7f62251ed9fb890289f5ad27f0dae # timeout=10
```

This screenshot shows the continuation of the Jenkins console output. The output text is as follows:

```
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 7a0556222eb7f62251ed9fb890289f5ad27f0dae (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 7a0556222eb7f62251ed9fb890289f5ad27f0dae # timeout=10
Commit message: "Update Sample.java"
First time build. Skipping changelog.
[Demo-app] $ /bin/sh -xe /tmp/jenkins9472666565217540260.sh
+ javac Sample.java
+ java Sample
1
2
3
4
5
6
7
8
9
10
Finished: SUCCESS
```

At the bottom right of the interface, the text "REST API" and "Jenkins 2.452.3" is visible.

- We can also configure the project and build the project again.

The screenshot shows the Jenkins configuration page for a project named 'Demo-app'. The left sidebar contains a 'Configure' section with a list of tabs: General, Source Code Management (selected), Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main content area is titled 'Configure' and shows the 'Source Code Management' configuration. It includes an 'Advanced' dropdown, an 'Add Repository' button, a 'Branches to build' section with a 'Branch Specifier (blank for 'any')' input field containing '*/featured', an 'Add Branch' button, a 'Repository browser' dropdown set to '(Auto)', and an 'Additional Behaviours' section with an 'Add' button. At the bottom are 'Save' and 'Apply' buttons.

The screenshot shows the same Jenkins configuration page for 'Demo-app', but with the 'Build Triggers' tab selected in the sidebar. The main content area is titled 'Build Triggers' and contains several options: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically' (checked), 'Schedule' (input field containing '*/* * * * *'), 'GitHub hook trigger for GITScm polling', and 'Poll SCM'. A warning message is displayed below the schedule field: 'Spread load evenly by using 'H/5 * * * *' rather than '*/* * * * *'. Would last have run at Tuesday, 23 July, 2024 at 11:05:33 AM India Standard Time; would next run at Tuesday, 23 July, 2024 at 11:10:33 AM India Standard Time.' At the bottom are 'Save' and 'Apply' buttons.

Changed the branch to **featured** and updated the **build trigger**.

The screenshot shows a web browser at localhost:8080/job/Demo-app/. The page has a breadcrumb 'Dashboard > Demo-app'. A 'Build History' widget is displayed with a 'trend' dropdown and a 'Filter...' search bar. The table lists 16 builds, all successful, with IDs #3 through #18. The timestamps range from 10:50 to 11:20 on 23 Jul 2024. A vertical scrollbar is on the right.

✓ #18	23 Jul 2024, 11:20	↑
✓ #17	23 Jul 2024, 11:18	↑
✓ #16	23 Jul 2024, 11:16	↓
✓ #15	23 Jul 2024, 11:14	
✓ #14	23 Jul 2024, 11:12	
✓ #13	23 Jul 2024, 11:10	
✓ #12	23 Jul 2024, 11:08	
✓ #11	23 Jul 2024, 11:06	
✓ #10	23 Jul 2024, 11:04	
✓ #9	23 Jul 2024, 11:02	
✓ #8	23 Jul 2024, 11:00	
✓ #7	23 Jul 2024, 10:58	
✓ #6	23 Jul 2024, 10:56	
✓ #5	23 Jul 2024, 10:54	
✓ #4	23 Jul 2024, 10:52	
✓ #3	23 Jul 2024, 10:50	

Build history before changing the build trigger

The screenshot shows the same web application after a trigger change. The 'Build History' widget now displays 17 builds, with a new entry #19 at the top (23 Jul 2024, 11:25). The previous builds #3 through #18 remain in the list, with their timestamps unchanged. The table is scrollable, as indicated by the scrollbar on the right.

✓ #19	23 Jul 2024, 11:25	↑
✓ #18	23 Jul 2024, 11:20	↑
✓ #17	23 Jul 2024, 11:18	↓
✓ #16	23 Jul 2024, 11:16	
✓ #15	23 Jul 2024, 11:14	
✓ #14	23 Jul 2024, 11:12	
✓ #13	23 Jul 2024, 11:10	
✓ #12	23 Jul 2024, 11:08	
✓ #11	23 Jul 2024, 11:06	
✓ #10	23 Jul 2024, 11:04	
✓ #9	23 Jul 2024, 11:02	
✓ #8	23 Jul 2024, 11:00	
✓ #7	23 Jul 2024, 10:58	
✓ #6	23 Jul 2024, 10:56	
✓ #5	23 Jul 2024, 10:54	
✓ #4	23 Jul 2024, 10:52	

Build history after changing build trigger.

Console Output

Started by timer
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Demo-app
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Demo-app/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/nkheria/sample-java-app.git # timeout=10
Fetching upstream changes from https://github.com/nkheria/sample-java-app.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/nkheria/sample-java-app.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/featured^{commit} # timeout=10
Checking out Revision 5fbd29c59993f2ac8aabc3f28ae0a07824966e93 (refs/remotes/origin/featured)
> git config core.sparsecheckout # timeout=10
> git checkout -f 5fbd29c59993f2ac8aabc3f28ae0a07824966e93 # timeout=10
Commit message: "Delete pom.xml"
> git rev-list --no-walk 5fbd29c59993f2ac8aabc3f28ae0a07824966e93 # timeout=10
[Demo-app] \$ /bin/sh -xe /tmp/jenkins3769984472804996983.sh

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Demo-app/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/nkheria/sample-java-app.git # timeout=10
Fetching upstream changes from https://github.com/nkheria/sample-java-app.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/nkheria/sample-java-app.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/featured^{commit} # timeout=10
Checking out Revision 5fbd29c59993f2ac8aabc3f28ae0a07824966e93 (refs/remotes/origin/featured)
> git config core.sparsecheckout # timeout=10
> git checkout -f 5fbd29c59993f2ac8aabc3f28ae0a07824966e93 # timeout=10
Commit message: "Delete pom.xml"
> git rev-list --no-walk 5fbd29c59993f2ac8aabc3f28ae0a07824966e93 # timeout=10
[Demo-app] \$ /bin/sh -xe /tmp/jenkins3769984472804996983.sh
+ javac Sample.java
+ java Sample
2
3
4
3
4
5
4
5
6
Finished: SUCCESS

Building a pipeline between 2 projects: Dev Compile & Dev Test

First created a **Dev Compile** project with a goal to **compile** the project

The screenshot shows the Jenkins configuration page for a project named 'Dev Compile'. The breadcrumb navigation at the top reads 'Dashboard > Dev Compile > Configuration'. On the left, the 'Configure' sidebar has several options: 'General', 'Source Code Management' (which is selected and highlighted), 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The main content area is titled 'Source Code Management' and contains two radio buttons: 'None' and 'Git'. The 'Git' option is selected. Below the radio buttons is a dashed box labeled 'Repositories'. Inside this box, there is a 'Repository URL' field containing 'https://github.com/02Poonam/private-repo.git', a 'Credentials' dropdown menu showing 'poonam/*****', an '+ Add' button, and an 'Advanced' dropdown. At the bottom of the configuration area are 'Save' and 'Apply' buttons.

Added credentials for pulling a private git repo.

This screenshot shows the same Jenkins configuration page, but with the 'Build Steps' tab selected in the left sidebar. The 'With Ant' checkbox is visible at the top of the main area. The 'Build Steps' section contains a dashed box with a single step named 'Invoke top-level Maven targets'. This step has a 'Maven Version' dropdown set to 'Maven-3.9.0' and a 'Goals' dropdown set to 'compile'. There is an 'Advanced' dropdown below the goals. Below the dashed box is an 'Add build step' button. At the bottom of the page are 'Save' and 'Apply' buttons.

Added the Maven version and goal **compile** and to build steps