Jenkins

Day-10

1. Installing jenkins and jvm on the machine.
Refering the documentations and Jenkins website online.

The command for the installation of jvm:

```
sudo apt install default-jdk
```

The command for the installation of jenkins:

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```

In order to start the jenkins after it has been installed:

```
sudo systemctl start jenkins.service
```

Since `systemctl` doesn't display status output, we'll use the `status` command to verify that Jenkins started successfully:

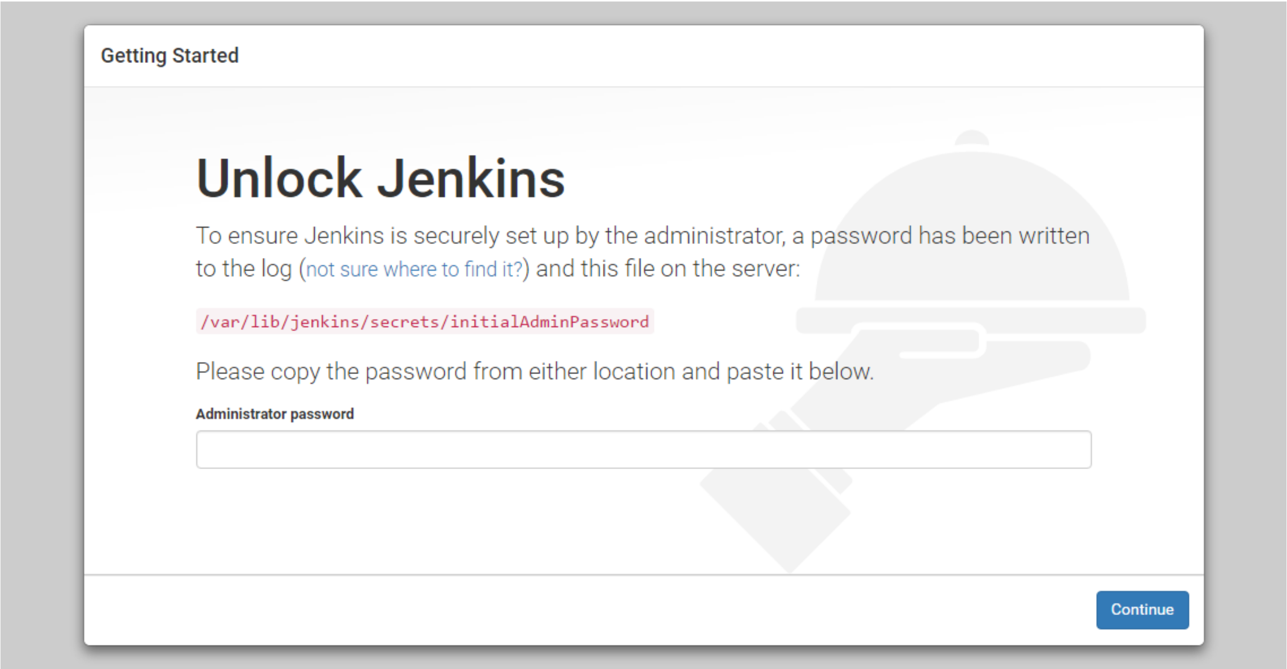
```
sudo systemctl status jenkins
```

In order to access the jenkins we need to have port 8080 open.

2.Setting up Jenkins

To set up your installation, visit Jenkins on its default port, `8080`, using your server domain name or IP address: `http:/<your_Server_ip>/:8080`

You should receive the **Unlock Jenkins** screen, which displays the location of the initial password:



Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

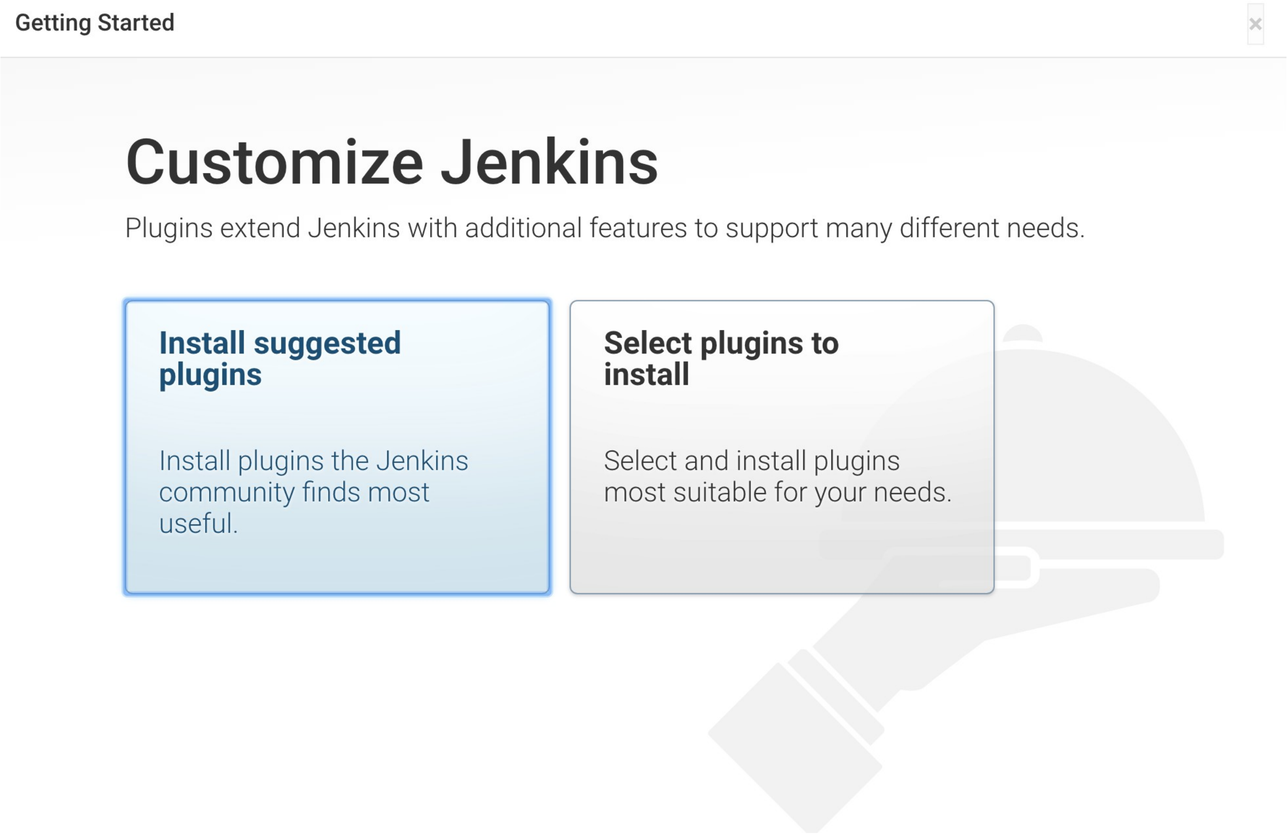
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

In the terminal window, use the `cat` command to display the password:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Copy the 32-character alphanumeric password from the terminal and paste it into the **Administrator password** field, then click **Continue**.



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

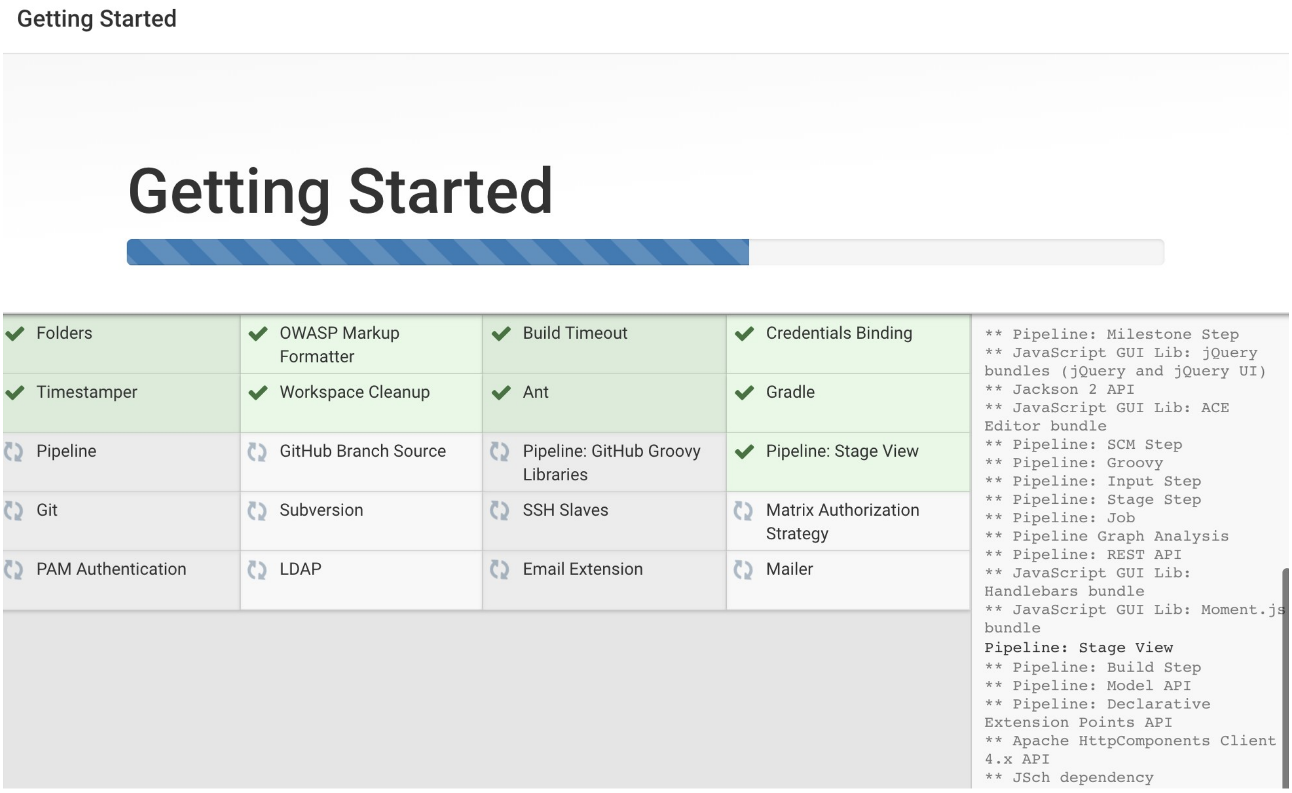
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

We'll click the **Install suggested plugins** option, which will immediately begin the installation process.



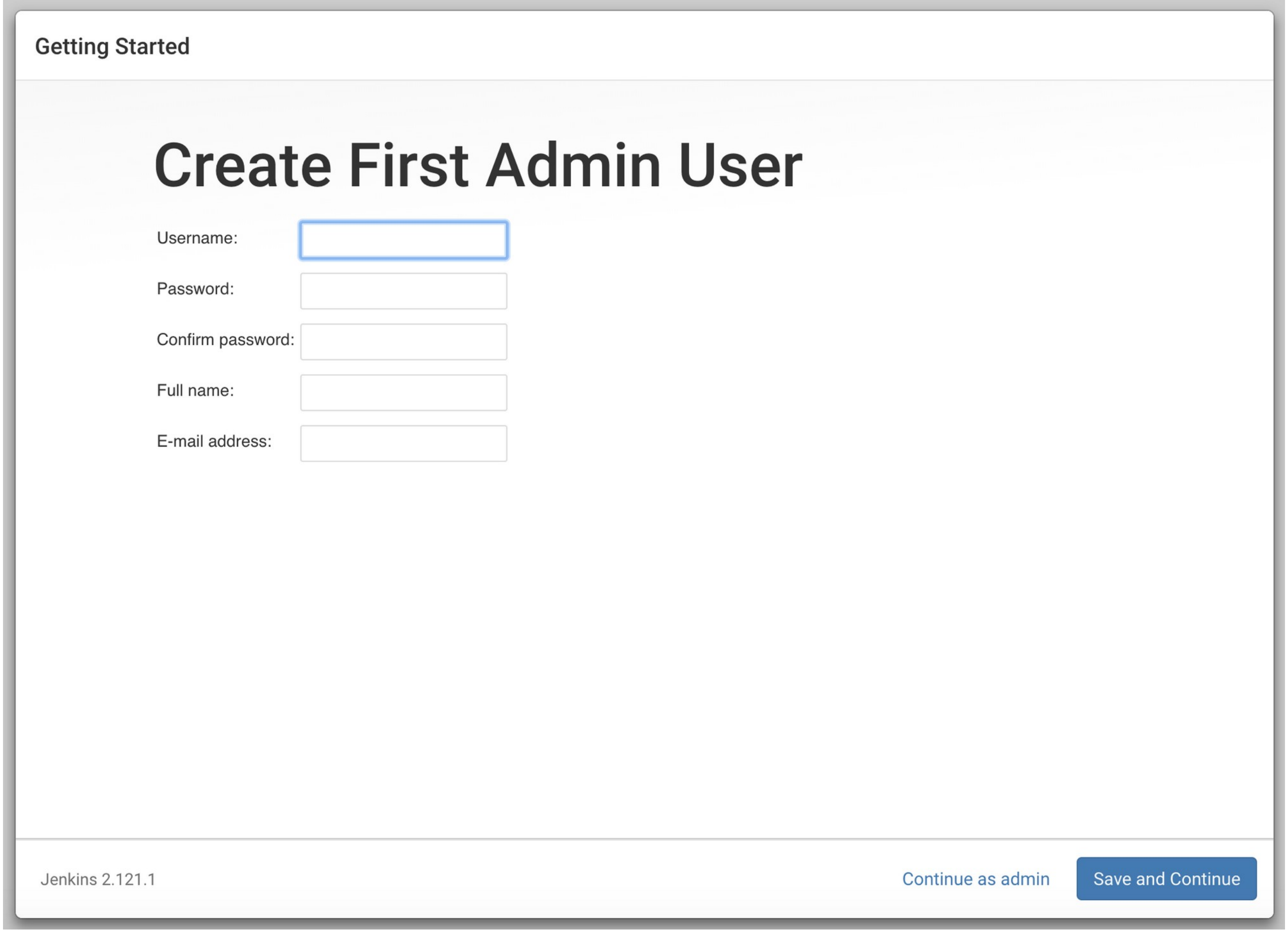
Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
⟳ Pipeline	⟳ GitHub Branch Source	⟳ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
⟳ Git	⟳ Subversion	⟳ SSH Slaves	⟳ Matrix Authorization Strategy
⟳ PAM Authentication	⟳ LDAP	⟳ Email Extension	⟳ Mailer

```
** Pipeline: Milestone Step
** JavaScript GUI Lib: jQuery
bundles (jQuery and jQuery UI)
** Jackson 2 API
** JavaScript GUI Lib: ACE
Editor bundle
** Pipeline: SCM Step
** Pipeline: Groovy
** Pipeline: Input Step
** Pipeline: Stage Step
** Pipeline: Job
** Pipeline Graph Analysis
** Pipeline: REST API
** JavaScript GUI Lib:
Handlebars bundle
** JavaScript GUI Lib: Moment.js
bundle
Pipeline: Stage View
** Pipeline: Build Step
** Pipeline: Model API
** Pipeline: Declarative
Extension Points API
** Apache HttpComponents Client
4.x API
** JSch dependency
```

When the installation is complete, you’ll be prompted to set up the first administrative user. It’s possible to skip this step and continue as `admin` using the initial password from above, but it is always preffered to create a username and password and mentioning all the other details required.



Getting Started

Create First Admin User

Username:

Password:

Confirm password:

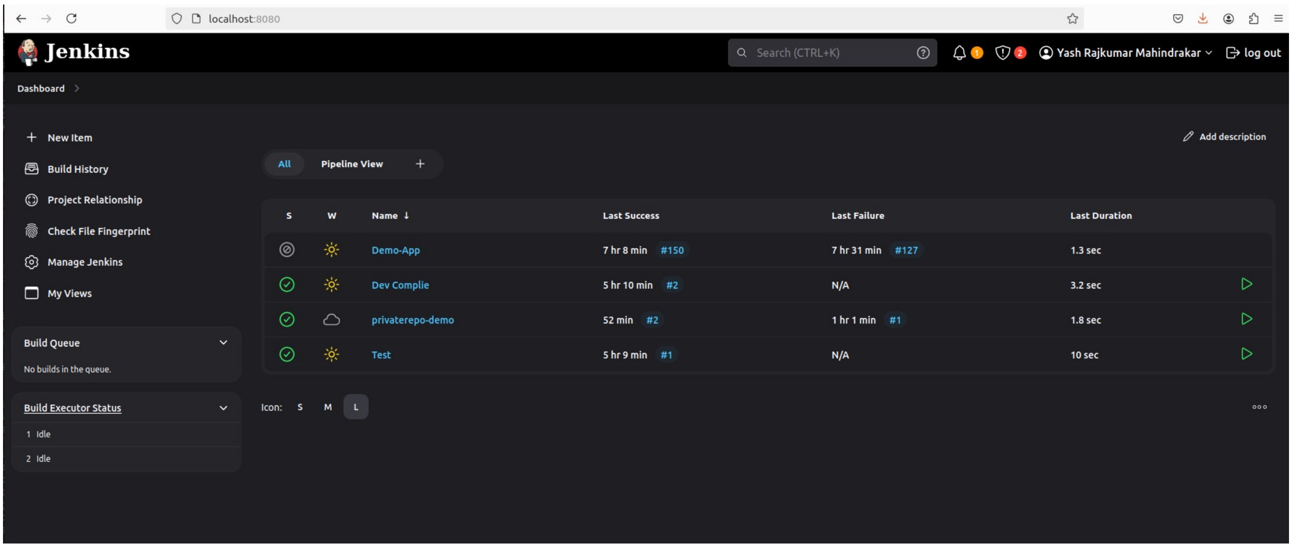
Full name:

E-mail address:

Jenkins 2.121.1

You’ll receive an **Instance Configuration** page that will ask you to confirm the preferred URL for your Jenkins instance. Confirm either the domain name for your server or your server’s IP address. After confirming the appropriate information, click **Save and Finish**. You’ll receive a confirmation page confirming that **“Jenkins is Ready!”.**

The dashboard for the jenkins will be ready at the `http:/<your_Server_ip>/:8080`



S	W	Name ↓	Last Success	Last Failure	Last Duration
⊘	☀	Demo-App	7 hr 8 min #150	7 hr 31 min #127	1.3 sec
✓	☀	Dev Complie	5 hr 10 min #2	N/A	3.2 sec
✓	☁	privaterepo-demo	52 min #2	1 hr 1 min #1	1.8 sec
✓	☀	Test	5 hr 9 min #1	N/A	10 sec

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

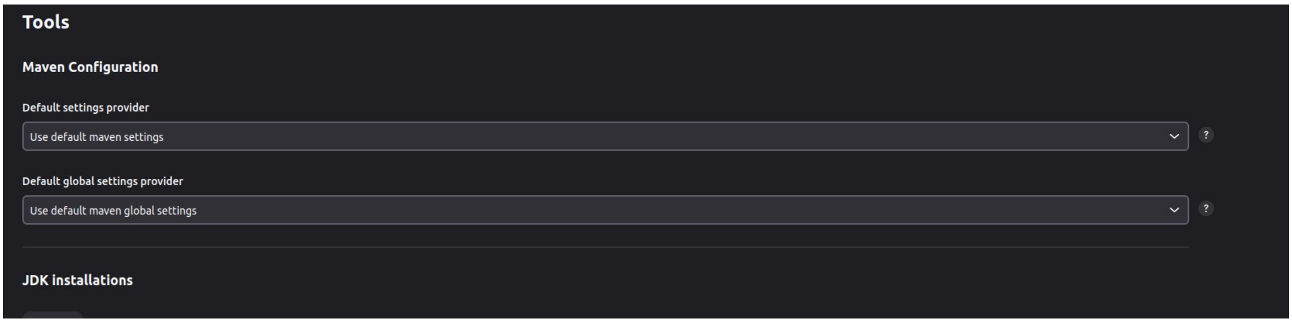
2 Idle

3. Setting up maven

Maven is used as a build tool.

IN order to set that up we need to do the followings steps:

Manage Jenkins ----> Tools



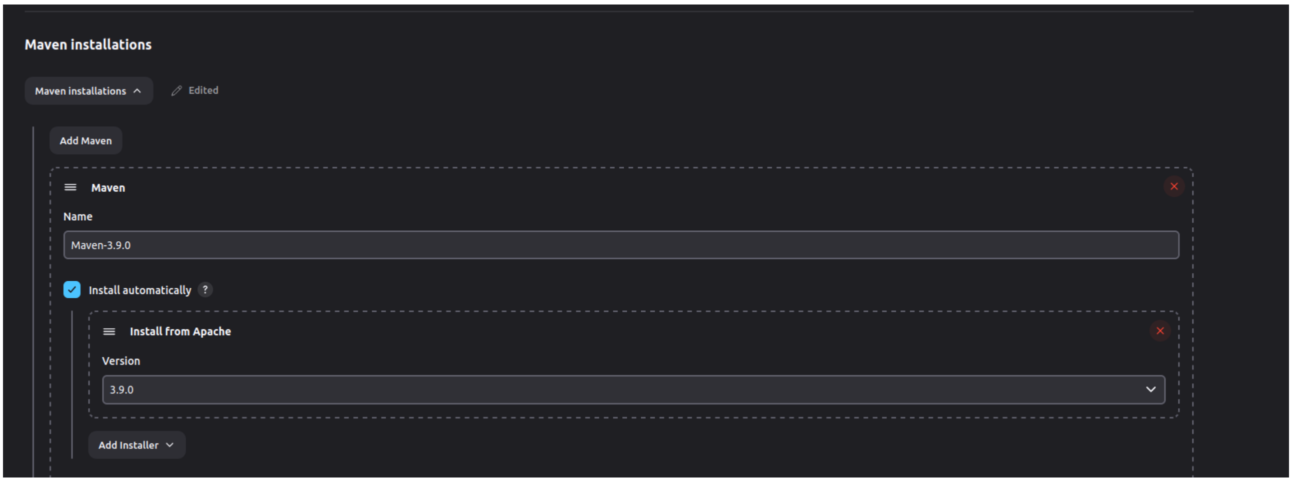
Tools

Maven Configuration

Default settings provider: Use default maven settings

Default global settings provider: Use default maven global settings

JDK installations



Maven installations

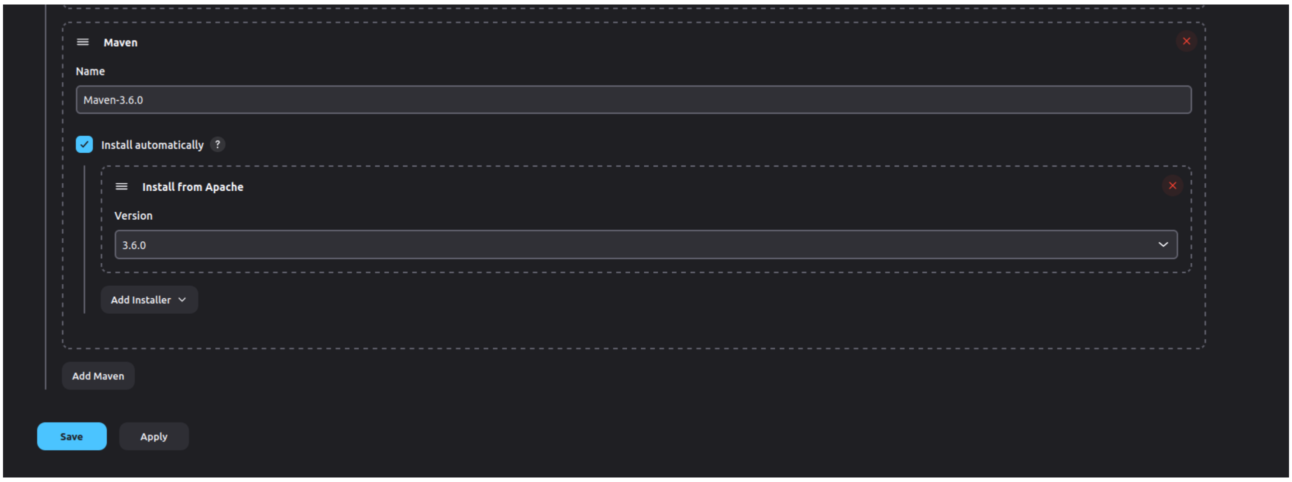
Maven

Name: Maven-3.9.0

☑ Install automatically

Install from Apache

Version: 3.9.0



Maven

Name: Maven-3.6.0

☑ Install automatically

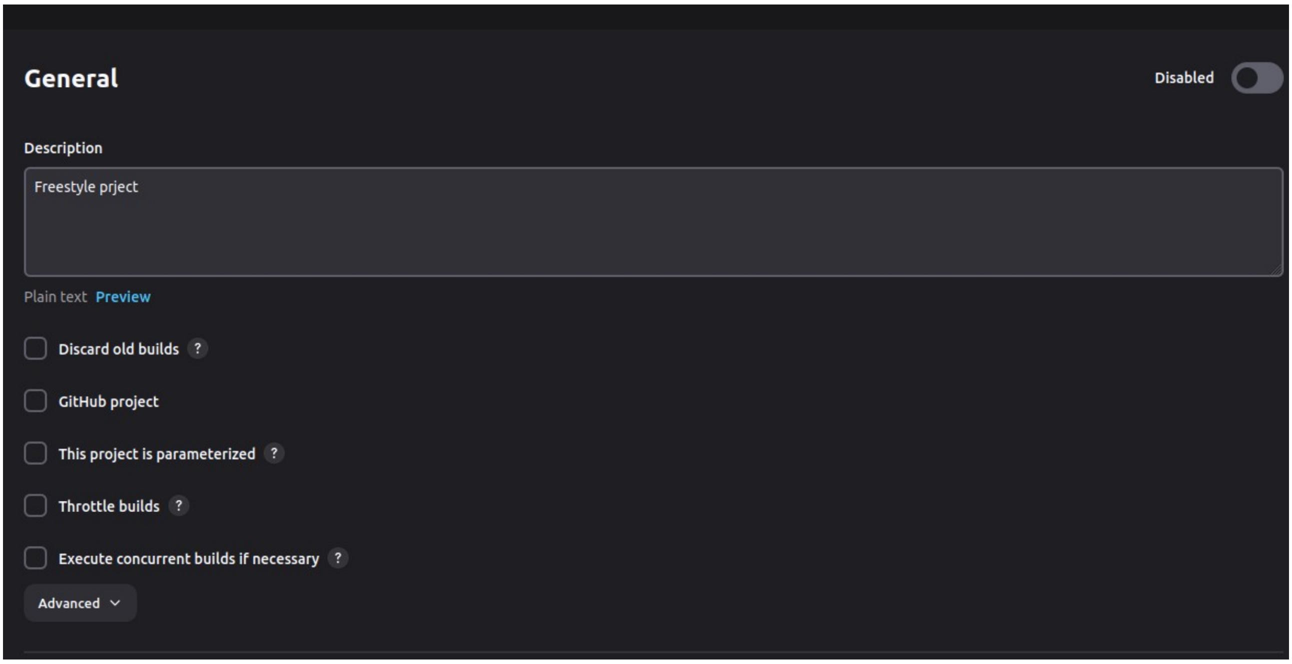
Install from Apache

Version: 3.6.0

Click save and the maven will be created.

4. Creating a JOB named demo-app:

Give the desired name to the job for e.g demo-app.
Use Freestyle Proect.



General

Disabled

Description

Freestyle prject

Plain text Preview

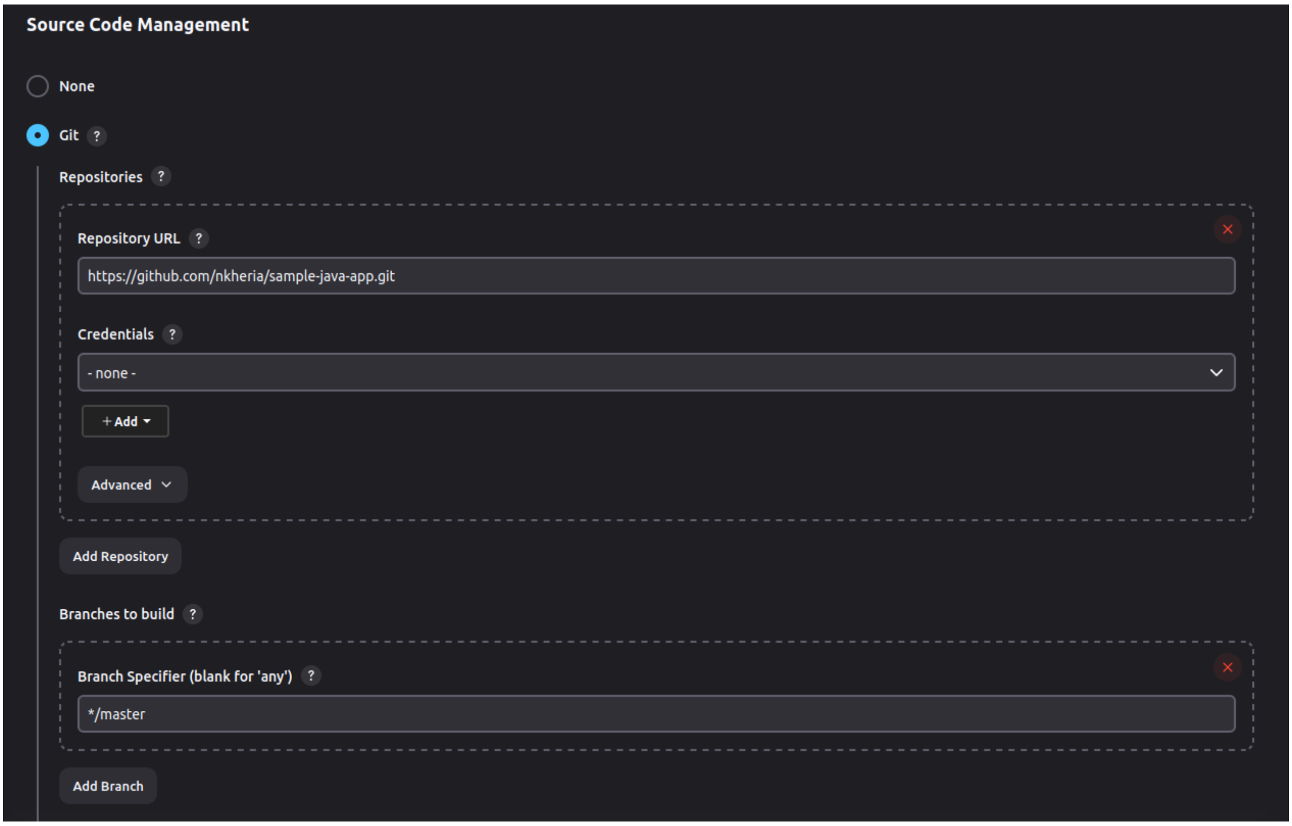
□ Discard old builds ?

□ GitHub project

□ This project is parameterized ?

□ Throttle builds ?

□ Execute concurrent builds if necessary ?



Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

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

Credentials ?

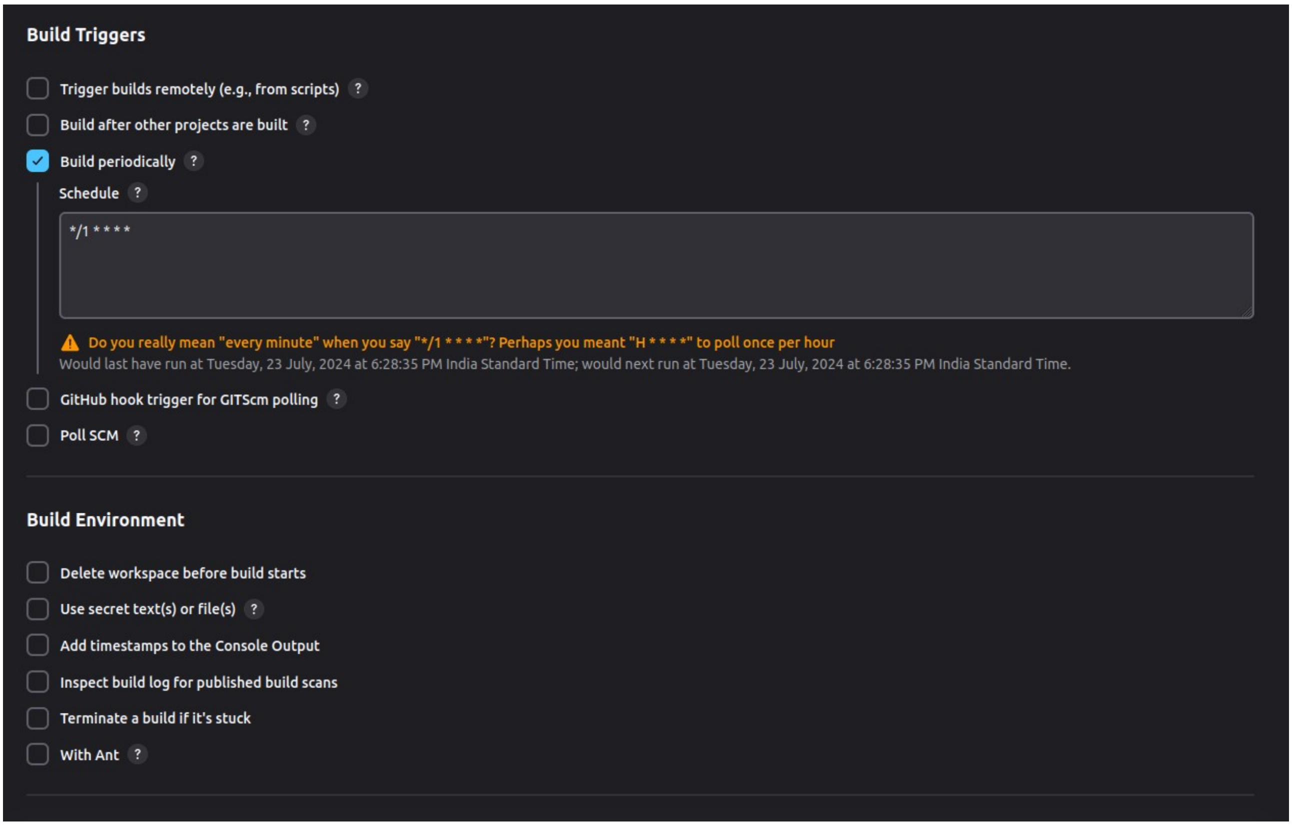
- none -

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Here we have to schedule a time when the job is supposed to be build in the form of minute, hour, day(month), month, day(week).



Build Triggers

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

□ Build after other projects are built ?

☑ Build periodically ?

Schedule ?

```
*/1 * * * *
```

⚠ Do you really mean "every minute" when you say "*/1 * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Tuesday, 23 July, 2024 at 6:28:35 PM India Standard Time; would next run at Tuesday, 23 July, 2024 at 6:28:35 PM India Standard Time.

□ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?

Build Environment

□ 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

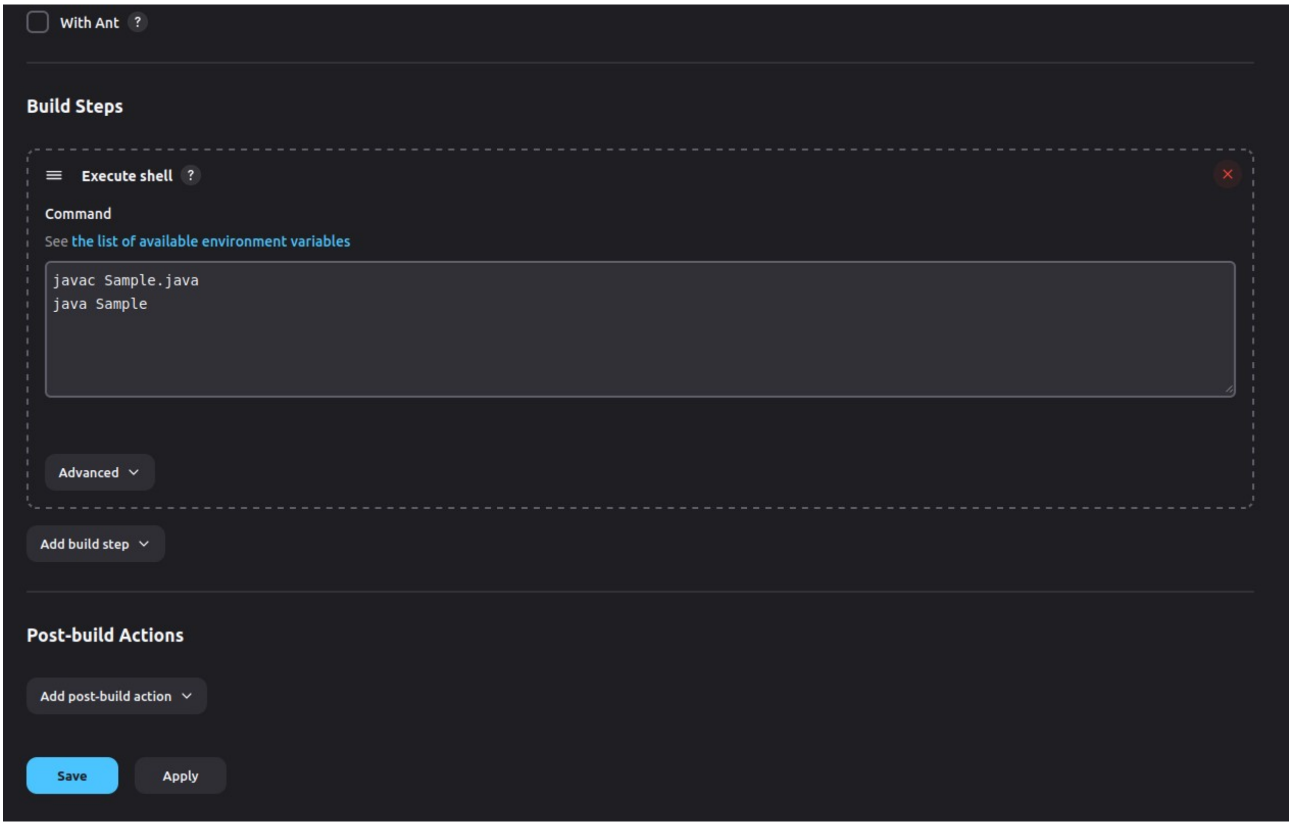
□ Terminate a build if it's stuck

□ With Ant ?

Under the build step we have to choose Execute shell and under command

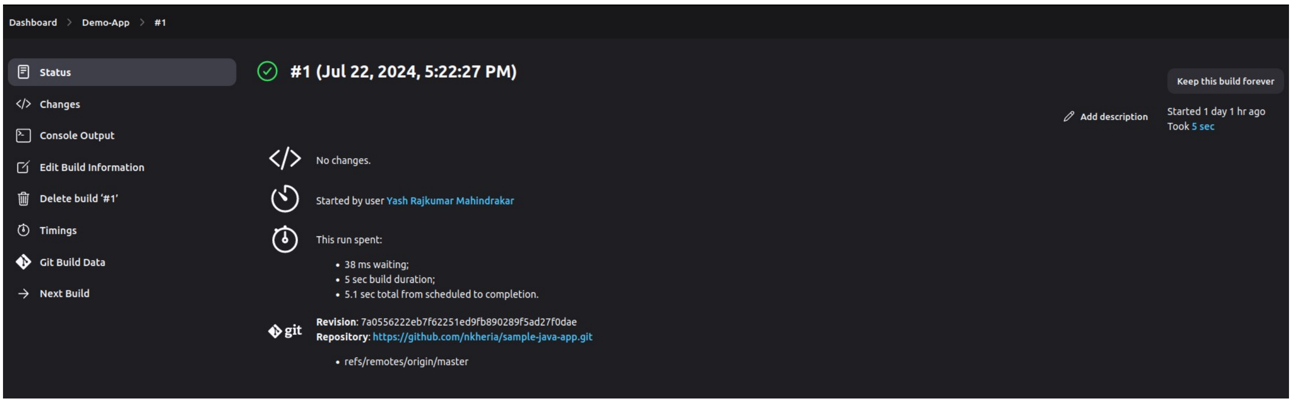
javac <filename.ext> here it would be javac Sample.java

java <function_name> here it would be java Sample



```
javac Sample.java
java Sample
```

Under the build id (#1, #2, #3, etc.) we can check the status and the output from the code.



#1 (Jul 22, 2024, 5:22:27 PM)

Started 1 day 1 hr ago
Took 5 sec

No changes.

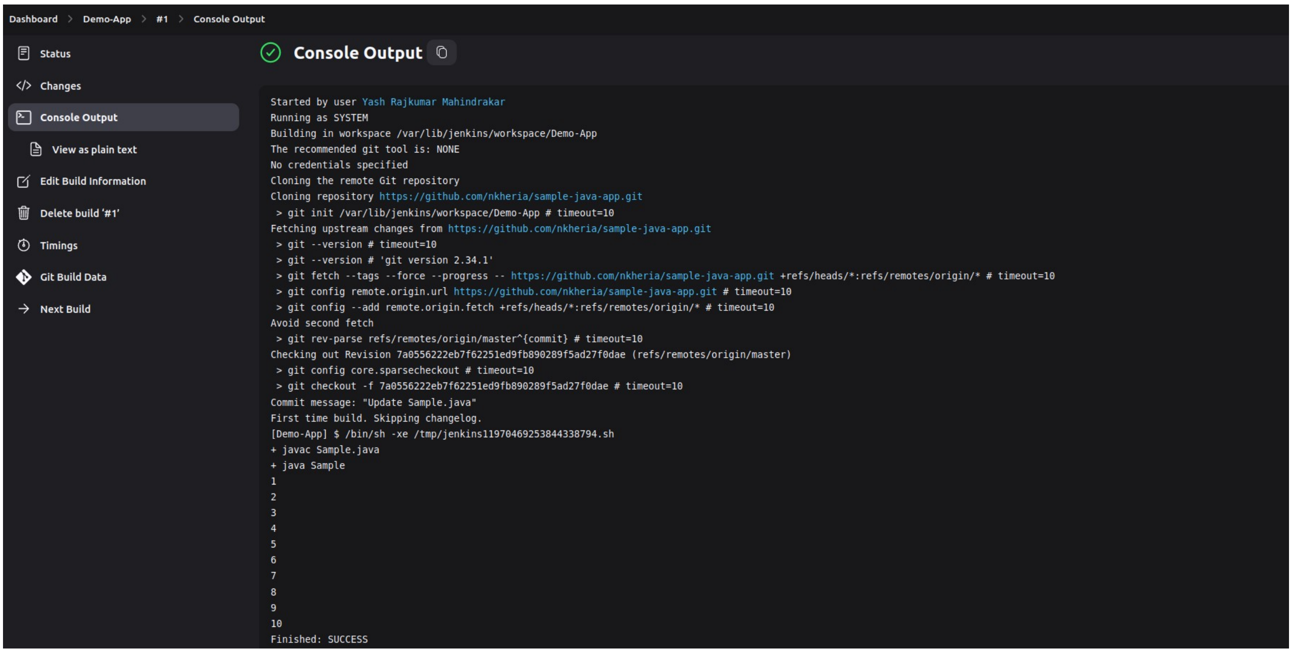
Started by user Yash Rajkumar Mahindrakar

This run spent:

- 38 ms waiting;
- 5 sec build duration;
- 5.1 sec total from scheduled to completion.

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

- refs/remotes/origin/master

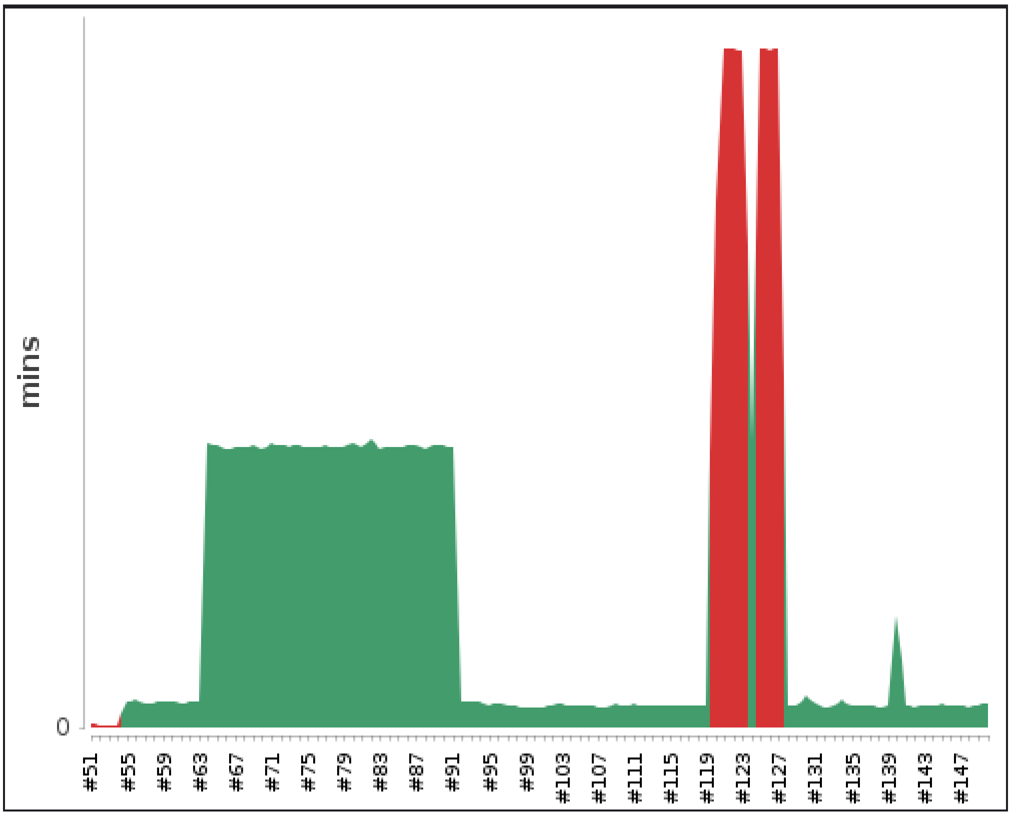


```
Started by user Yash Rajkumar Mahindrakar
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Demo-App
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/nkheria/sample-java-app.git
 > git init /var/lib/jenkins/workspace/Demo-App # 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
Commit message: "Update Sample.java"
First time build. Skipping changelog.
[Demo-App] $ /bin/sh -xe /tmp/jenkins11970469253844338794.sh
+ javac Sample.java
+ java Sample
1
2
3
4
5
6
7
8
9
10
Finished: SUCCESS
```

By doing so we have successfully build and created a job named demo-app which will give us number from 1 to 10 as an output which can be seen under console output.

The build time trend for the created job naed demo-app can be seen in the graphical format by:

tapping on the trend available besides the Build History.



The red part in the graph represnts which build failed and the graph area in green represnts build ids that were successfull.