Jenkins

Jenkins is certainly one of the most popular tools in **DevOps**. It can automate building and testing the code at a faster rate and because of this software companies can speed up their development processes. Jenkins provides you with an email notification service through which you can report the build status and testing results to the team.

Use Of Jenkins:

Jenkins, organizations can accelerate the software development process through automation. Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis, and much more.

Jenkins achieves Continuous Integration with the help of plugins. Plugins allow the integration of Various DevOps stages. If you want to integrate a particular tool, you need to install the plugins for that tool. For example, Git, Maven 2 project, Amazon EC2, HTML publisher etc.

Advantages of Jenkins:

- It is an open-source tool with great community support.
- It is easy to install.
- It has 1000+ plugins to ease your work. If a plugin does not exist, you can code it and share it with the community.
- It is free of cost.
- It is built with Java and hence, it is portable to all the major platforms.

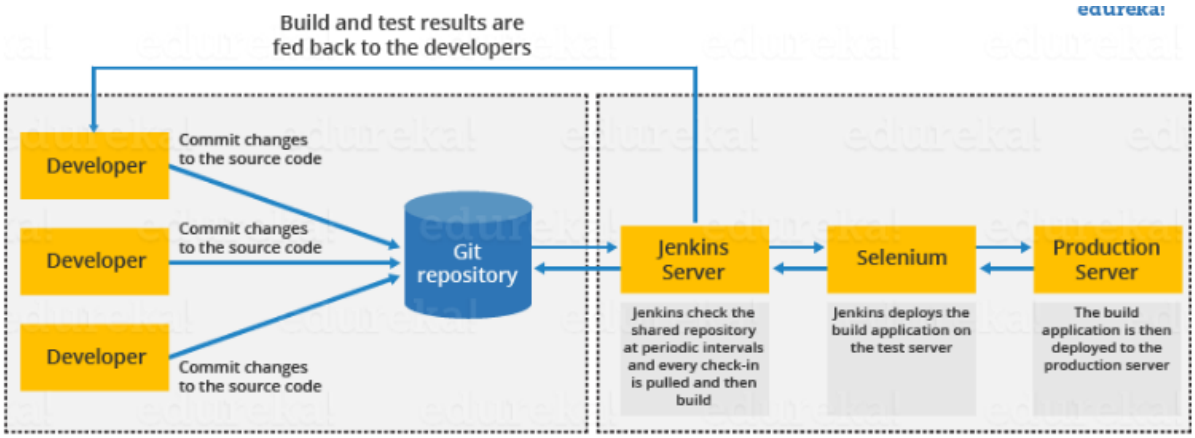
Jenkins Features:

- **Adoption:** Jenkins is widespread, with more than 147,000 active installations and over 1 million users around the world.
- **Plugins:** Jenkins is interconnected with well over 1,000 plugins that allow it to integrate with most of the development, testing and deployment tools.

Continuous Integration:

Continuous Integration is a development practice in which the developers are required to commit changes to the source code in a shared repository several times a day or more frequently. Every commit made in the repository is then built. This allows the teams to detect the problems early. Apart from this, depending on the Continuous Integration tool, there are several other functions like deploying the build application on the test server, providing the concerned teams with the build and test results, etc.

Continuous Integration with Jenkins:



The above diagram is depicting the following functions:

- First, a developer commits the code to the source code repository. Meanwhile, the Jenkins server checks the repository at regular intervals for changes.
- Soon after a commit occurs, the Jenkins server detects the changes that have occurred in the source code repository. Jenkins will pull those changes and will start preparing a new build.
- If the build fails, then the concerned team will be notified.
- If built is successful, then Jenkins deploys the built in the test server.
- After testing, Jenkins generates feedback and then notifies the developers about the build and test results.
- It will continue to check the source code repository for changes made in the source code and the whole process keeps on repeating.

Jenkins Email Notification:

Jenkins email notifications is the way to notify based on event occurred or some action happened. Jenkins email notifications is kind of message that is automatically sent to you and update that, there has been activity on one of your social media accounts like Google, slack. As email is the primary means of notification among other social media for Jenkins email notifications. Jenkins provided a plugin to extend functionality of e-mail notifications. It's basically informing users about some event or status or any information's that needs to be update to their concern users.

Three parts of Jenkins Email Notification

- **Triggers:** This is a defined condition which causes an e-mail notification to be sent.
- **Content:** Basically, defined content for email subject and body.
- **Recipients:** We can mention the concerned user who is supposed to receive an e-mail when any event occurs.

Configure Gmail SMTP Server in Jenkins:

After the successful installation of Jenkins, it comes with several built in functionalities and several plugins like mail sending. Just we need to configure any plugins or email sending plugin with defined necessary values as in case of email what SMTP server expected.

When the Jenkins email notification triggers, the email is sent to the respective recipients. For Automated build job, there are many events such as success, failure, abort and so on After this configuration click on trigger option which is basically a selection of required event for which mail has to be triggered. Once all the changes are made, save the configurations.

Sending Email Notification In Jenkins:

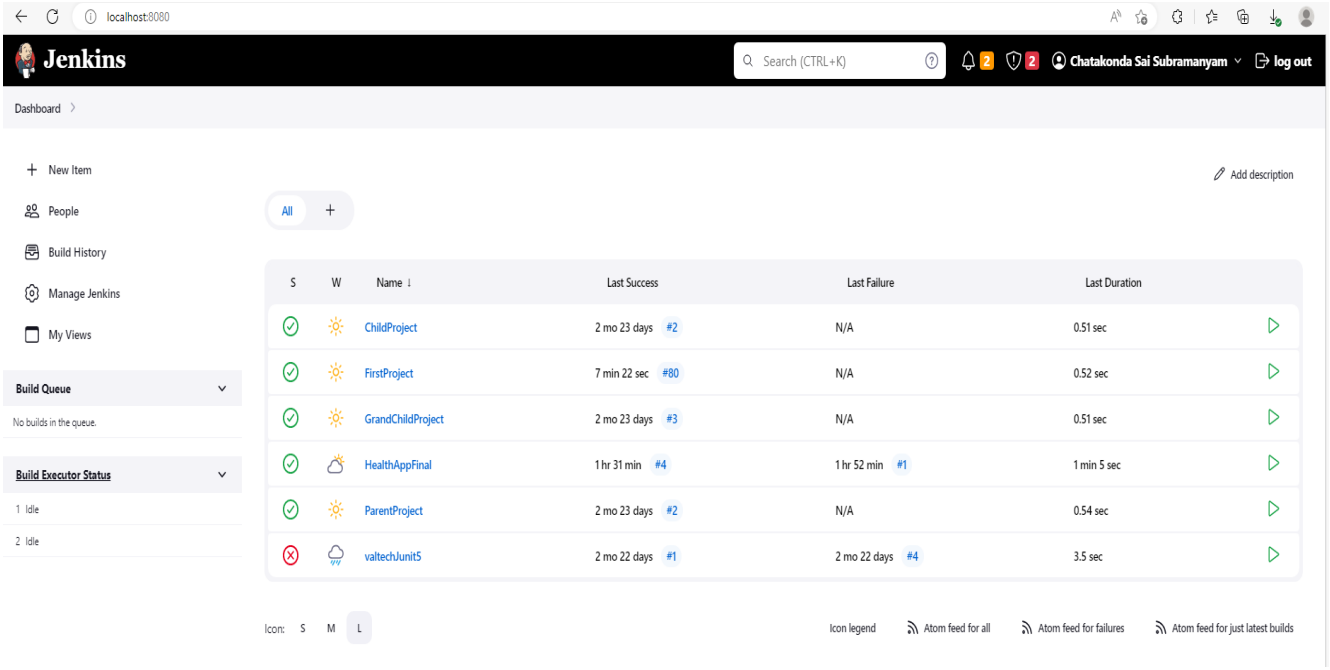
There are basically two ways to configure email notifications in Jenkins.

1. **Using Email Extension Plugin** – This plugin lets you configure every aspect of email notifications. You can customize things such as when to send the email, who receives it, and what the email says.
2. **Using Default Email Notifier** – This comes with Jenkins by default. It has a default message consisting of a build number and status.

Steps:

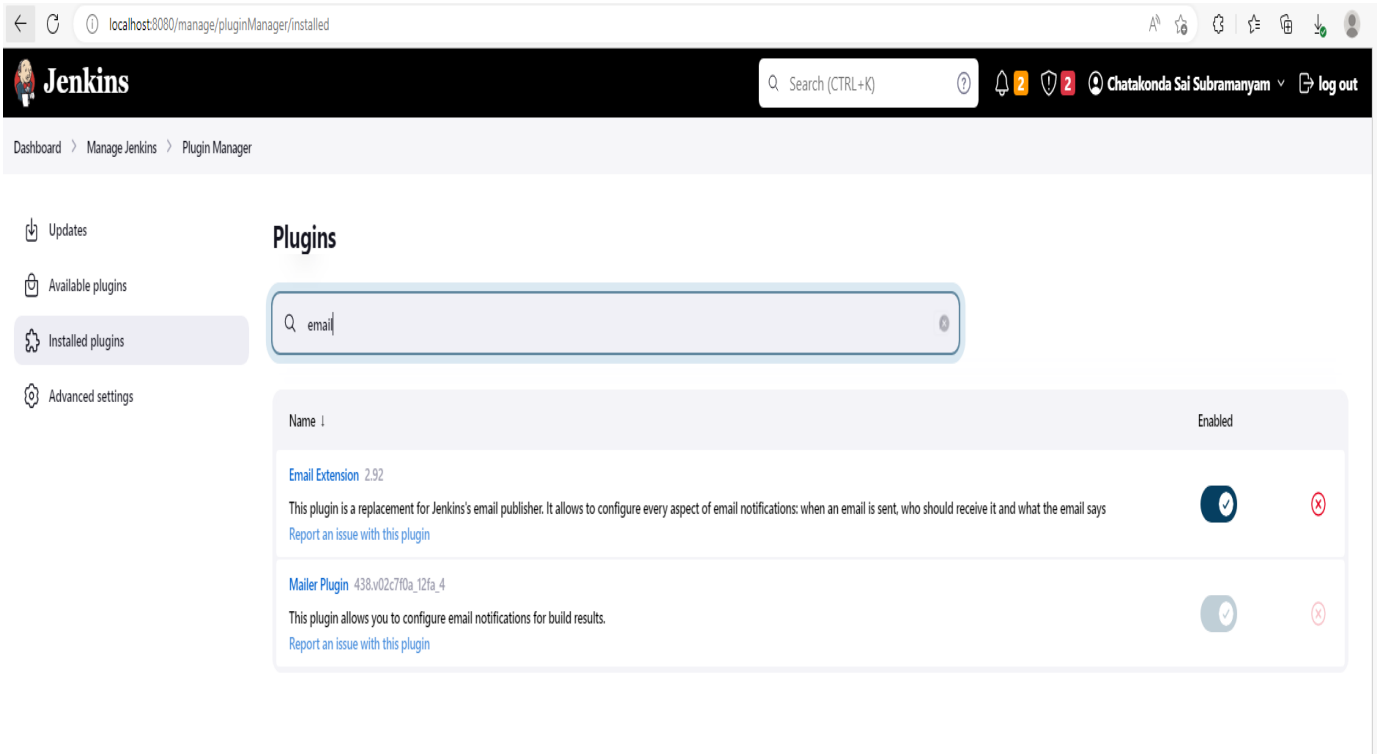
Go to Jenkins home page using the URL localhost:8080. The port number by default is 8080. Sign in using your username and password.

1. Go to Jenkins dashboard.



2. Install Email Extension Plugin:

After that on the Jenkins homepage click on **Manage Jenkins-> Manage Plugins**. In the available tab search for Email Extension Plugin. If it is found there, install it. If it is not found there, check for it in the installed tab.



3. Configure System:

Now go to **Manage Jenkins-> Configure System**. Here scroll down to the email notification section. If you are using Gmail, then type **smtp.gmail.com** for the SMTP server. Click on Advanced and select Use SMTP authentication. Enter your Gmail username and password. Select the Use **SSL** option and enter the port number as **465**. Click on Apply and then Save.

Under E-mail Notification, put the mentioned details below:

- SMTP server → smtp.gmail.com
- Select Use SMTP Authentication
- Put your Gmail id
- Put your Gmail password
- Use SSL select
- SMTP port 465
- Save the configuration

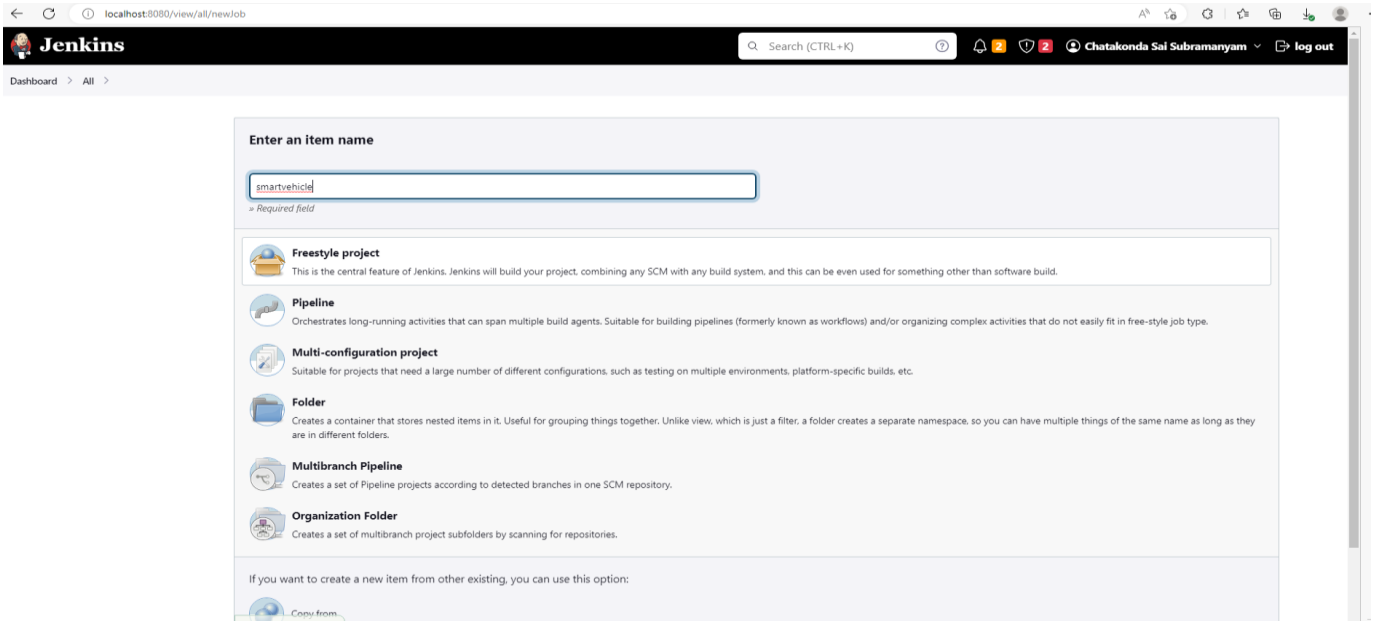
- To verify the email notification functionality by clicking the checkbox below to the ‘Test configuration by sending Test e-mail recipient’ option.

- Enter a valid email id and click the ‘Test configuration’ button to check whether the email id is valid or not.

- Checking on the box that is used for SMTP Authentications, to provide the user credentials for the SMTP account.

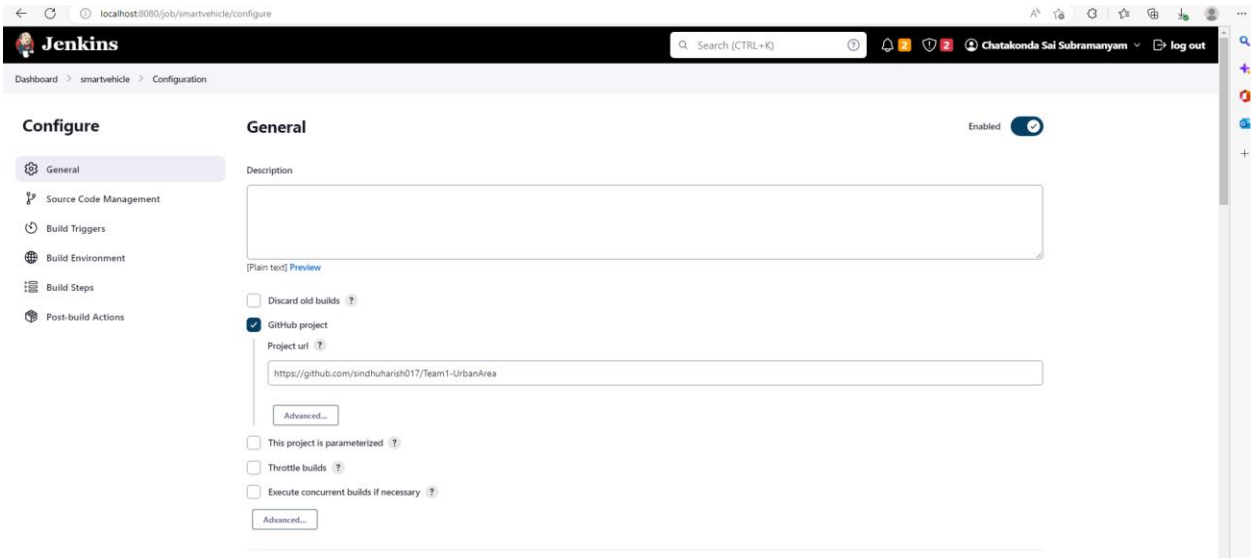
4. Create Jenkins freestyle project:

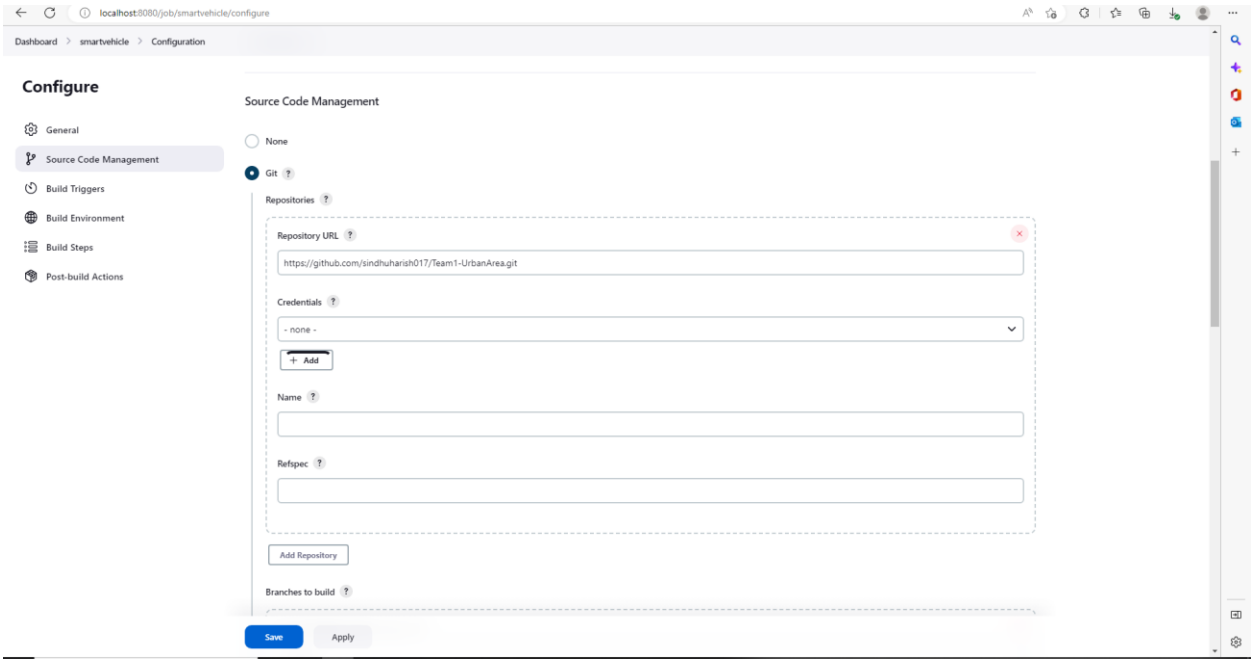
Now go to Jenkins' homepage and create a new job. Name the job with whatever name that you want and select freestyle project. Click on OK.

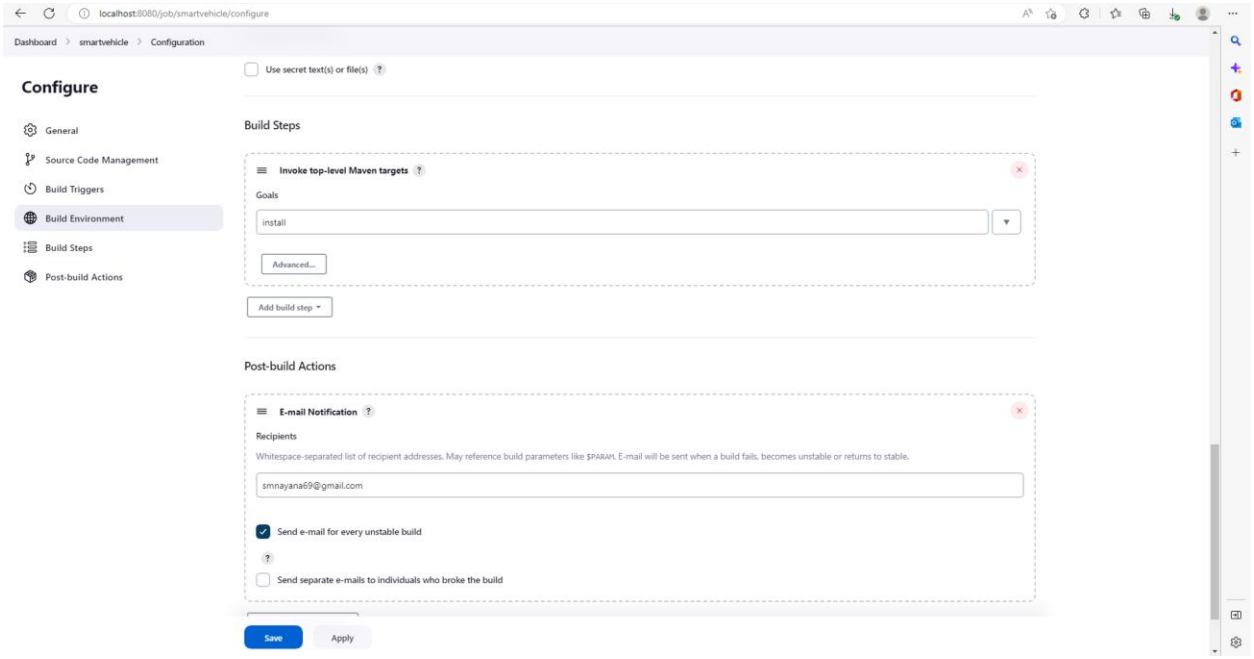


5. Add post-build action:

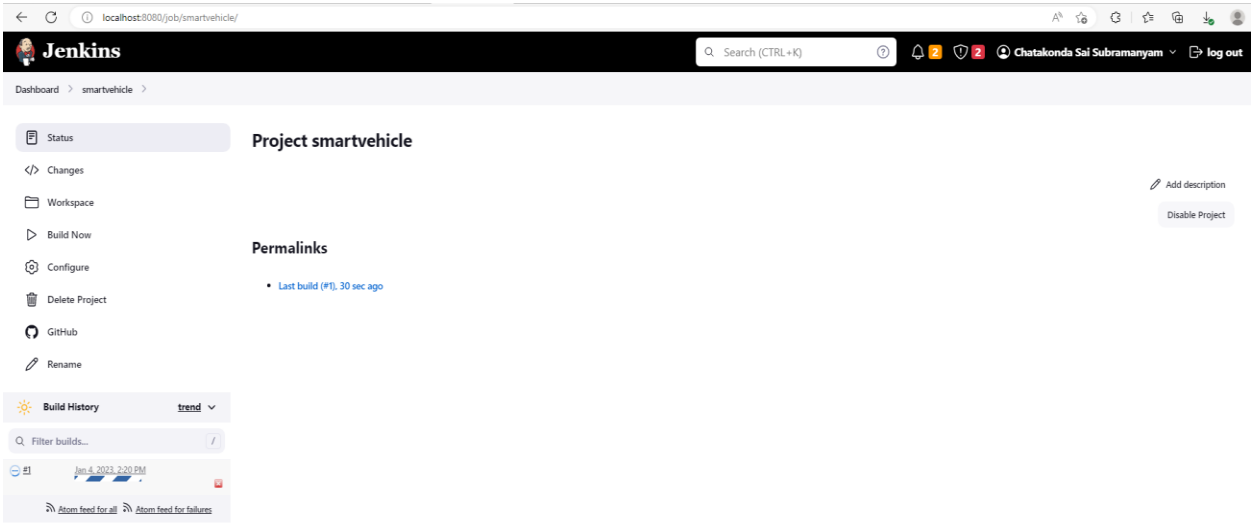
To allow a Jenkins project to send an email on successful build or failed build, we need to add **Post Build Action** and select "**Email Notification** from the drop-down list. This will provide you with the below interface, where you can add a list of email addresses that the email is required to be sent to.




localhost:8080/job/smartvehicle/configure
Dashboard
smartvehicle
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/sindhuharish017/Team1-UrbanArea.git
Credentials
- none -
Add
Name
Refspec
Add Repository
Branches to build
Save
Apply

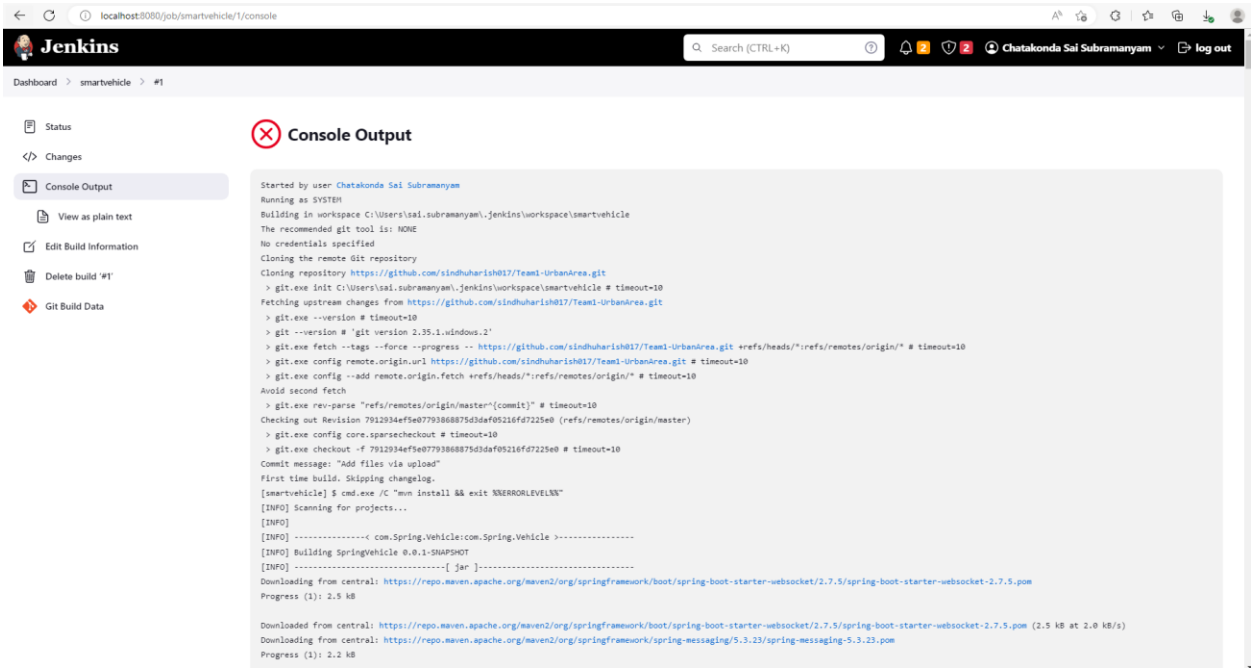

localhost:8080/job/smartvehicle/configure
Dashboard
smartvehicle
Configuration
Use secret text(s) or file(s)
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build Steps
Invoke top-level Maven targets
Goals
install
Advanced...
Add build step
Post-build Actions
E-mail Notification
Recipients
Whitespace-separated list of recipient addresses. May reference build parameters like $PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.
smnayana69@gmail.com
Send e-mail for every unstable build
Send separate e-mails to individuals who broke the build
Save
Apply

6. After the successful configuration click on the **build now** option in order to trigger the build. Based on the build event occurred like success or failed, server send mail.



7. View Console Output

Click on Build Number "#1" and click on "Console Output" on the build menu. The output will be like this.



8. Check The Mail:

We can get the build failure notification as failure event occurred at Jenkins server.