Jenkins

1.To create a Jenkins container using Docker:

* Pull the required image from docker hub

docker pull jenkins (version no)

* Creating a volume for data persistency.

docker create volume sample

2. Creating the container

docker run -d --name=jenkins -p 8080:8080 -p 50000:50000 -v sample:/var/jenkins\_home jenkins/jenkins:lts

* here, -d = it is a detach mode flag. It basically runs the container in the background.
* -p = attach the port to the running container.
* -v = to attach the volumes to the container.

3.To check whether the container is running.

<http://localhost:8080/>

4. To enter into bash or root user.

docker exec -it -u 0 <container name> bash

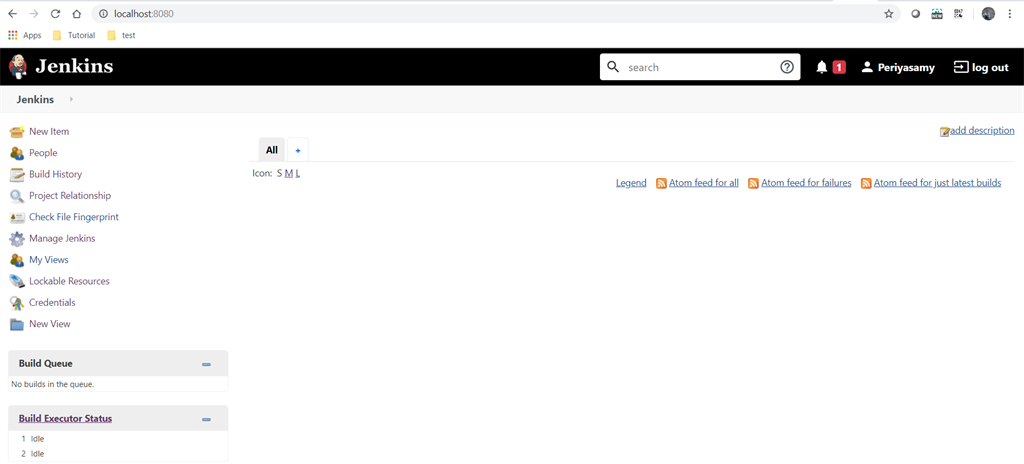


5. To install node.js

* apt update (it updates)
* cat /etc/issue (it checks the OS we are working)
* apt install nodejs npm -y

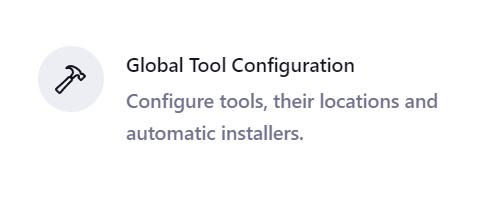
6.To enter the Jenkins UI:

* Give the administrative password.
* Install all the plugins required.
* Set the LOGIN ID

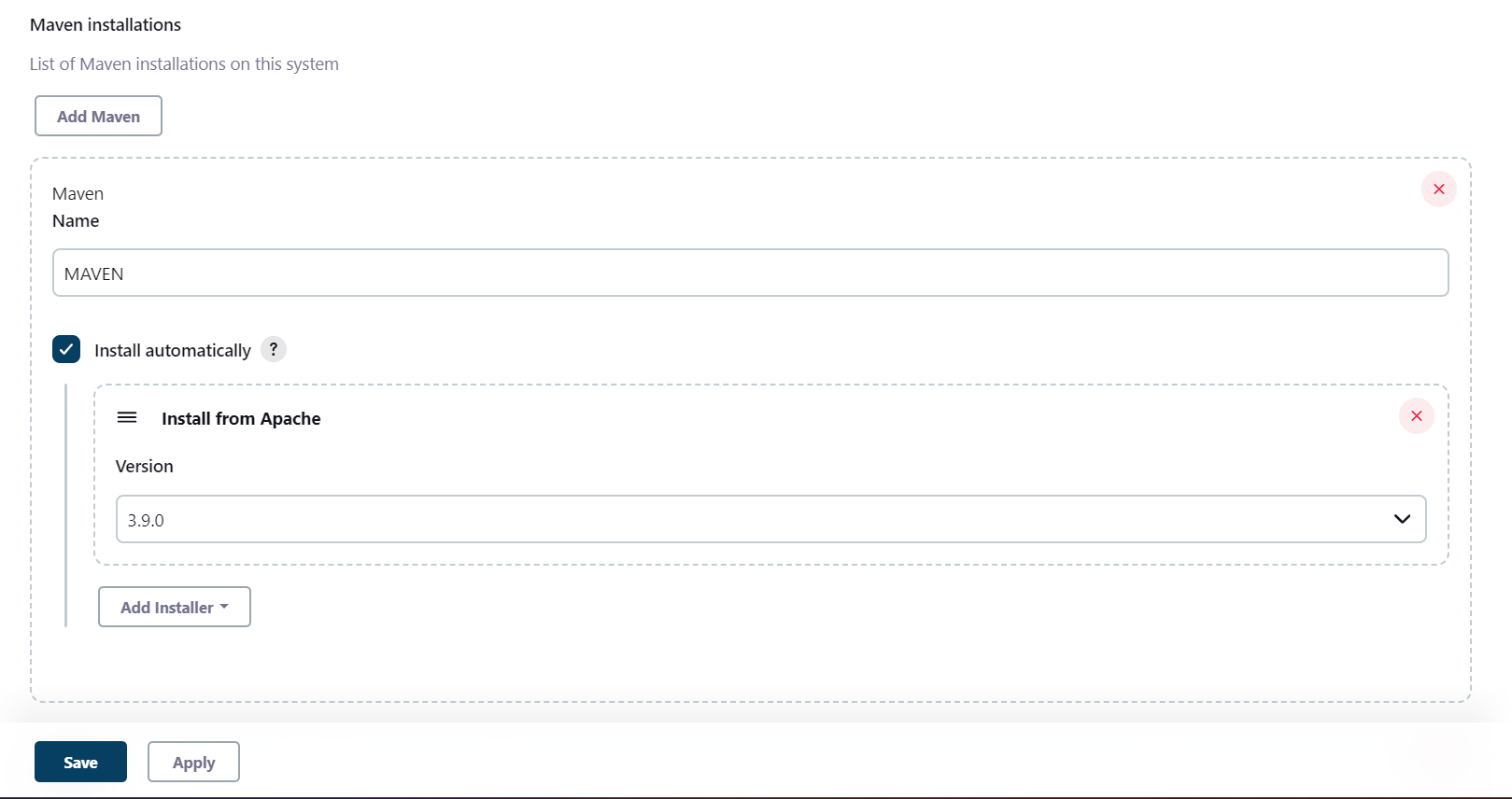


7.To configure various tools that we need to utilize at the time of creating a build job.

* Go to Global configuration



* Give the required for creating and building a job.

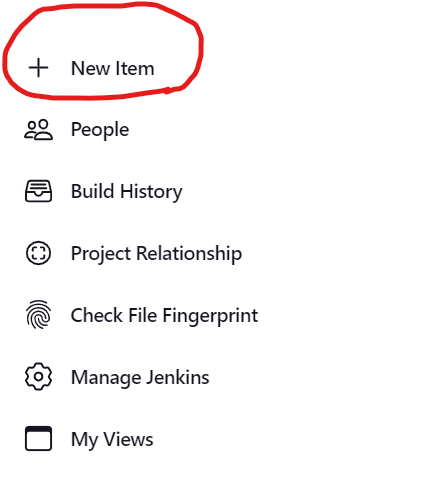


8. To create a new job in the jenkins.

* There are different ways to create a job.

1. Free style project.
2. Pipeline project.
3. Multi-branch pipeline project.

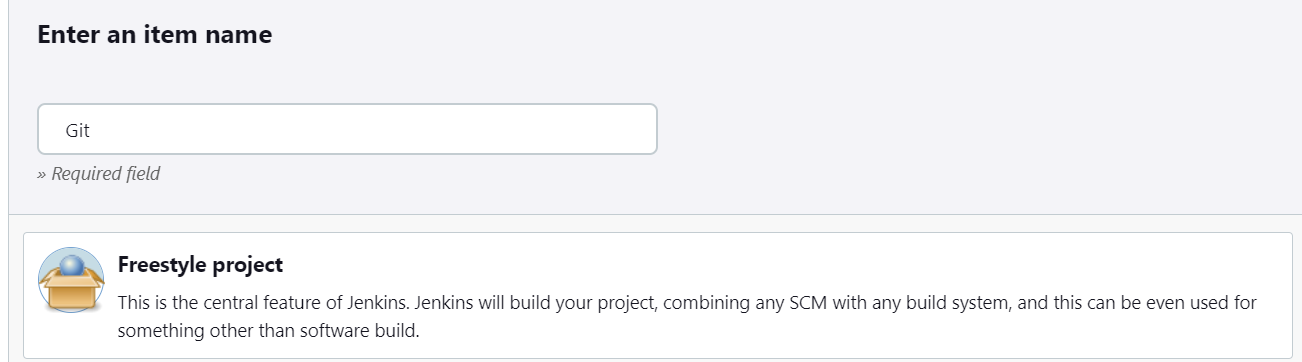
We can create a job here.



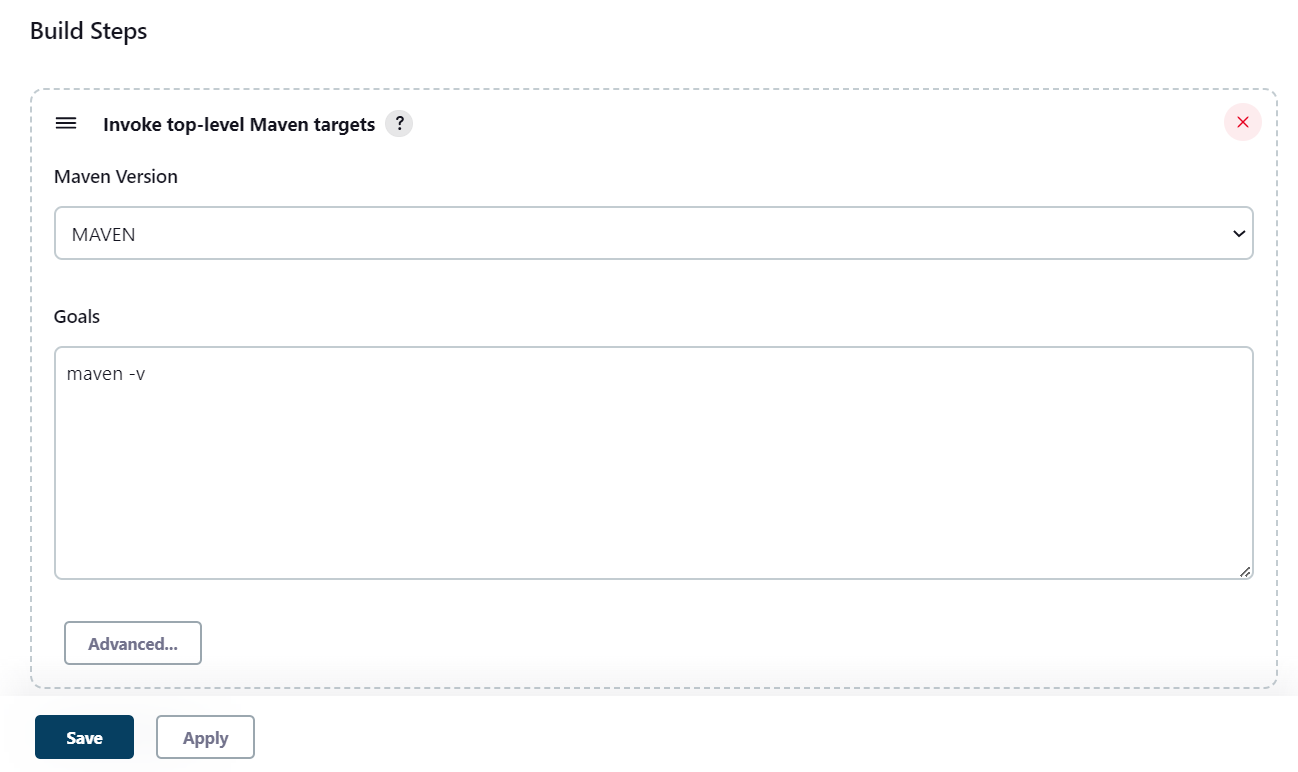
* Here we the job name and select the project type



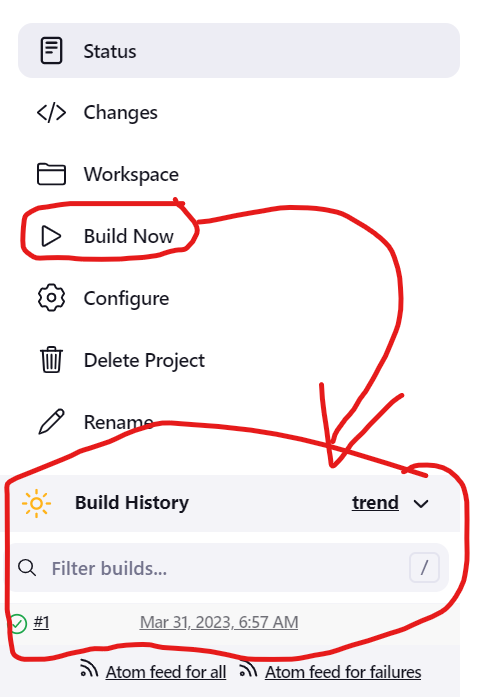
* Free style project:
* Here we give the required project name and select the freestyle project



* Give the requirements in the general page



* After giving requirements, apply the changes and build.



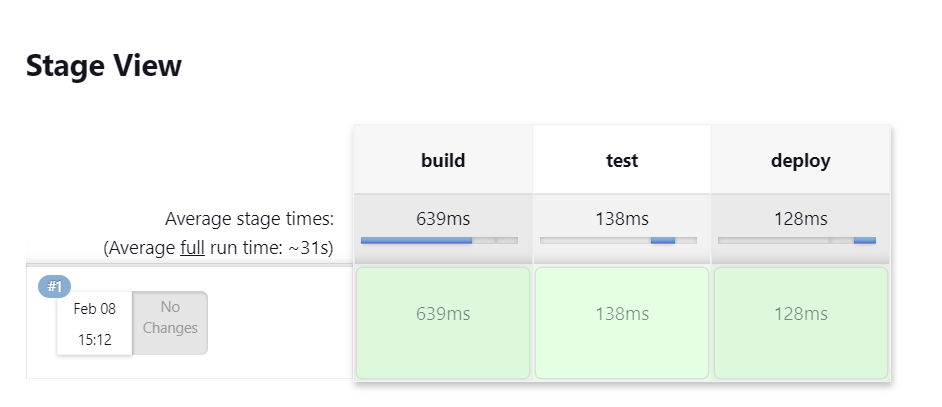
* Whatever the build is, it is shown in the build history.
* Pipeline Project:
  + It creates multiple jobs at a time.
  + It can create in two different ways :

1. Pipeline with SCM
2. Pipeline script

* Pipeline with SCM:
* We need to define the type of pipeline.
* The SCM where the code is present.
* Need to give the URL and give the credentials of the repository.



* After giving the requirements we need to save and apply in order to build the project.



* Using Pipeline Script:
* We define the type of script.
* Here we write script in groovy script.



Example script for Pipeline script

pipeline {

agent none

stages {

stage('build') {

steps {

script {

echo "Building the application..."

}

}

}

stage('test') {

steps {

script {

echo "Testing the application..."

}

}

}

stage('pull') {

steps {

script {

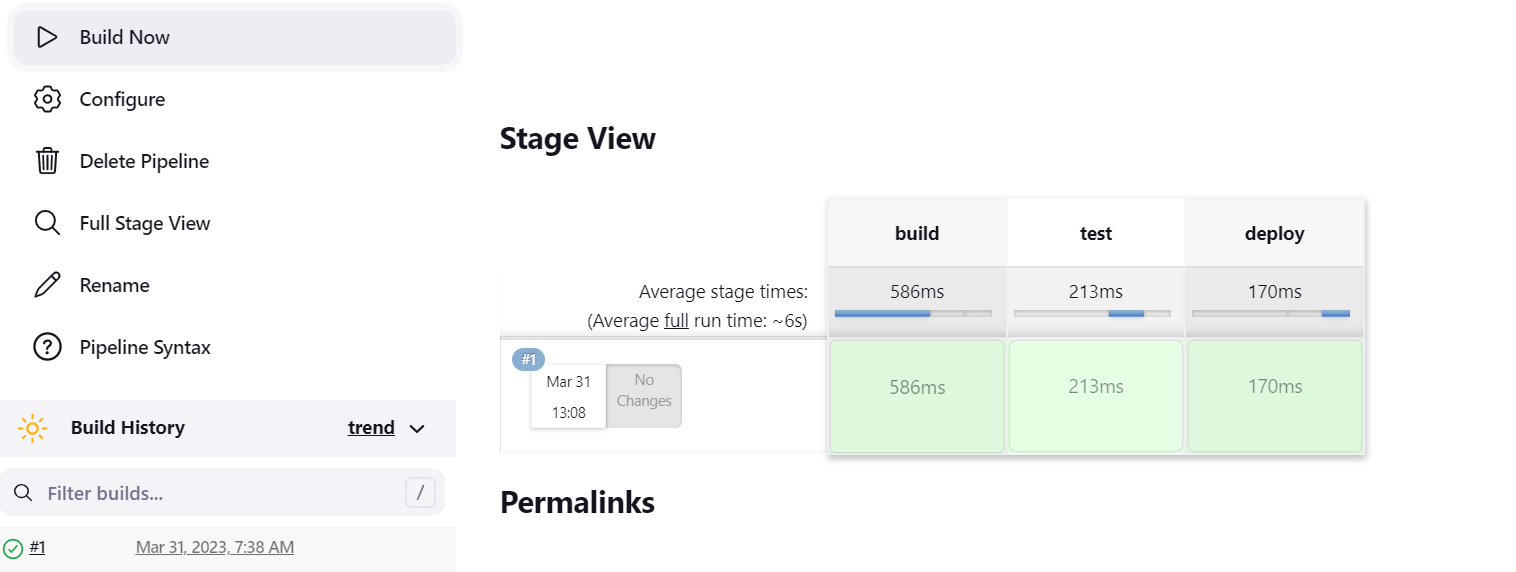
echo "Deploying the application..."

}

}

}

* After writing the script save and apply.
* Build the project.



* WEBHOOKS:

It is a mechanism to automatically trigger the build of jenkins

project in response to a commit pushed to a git repository.

To create a Webhook:

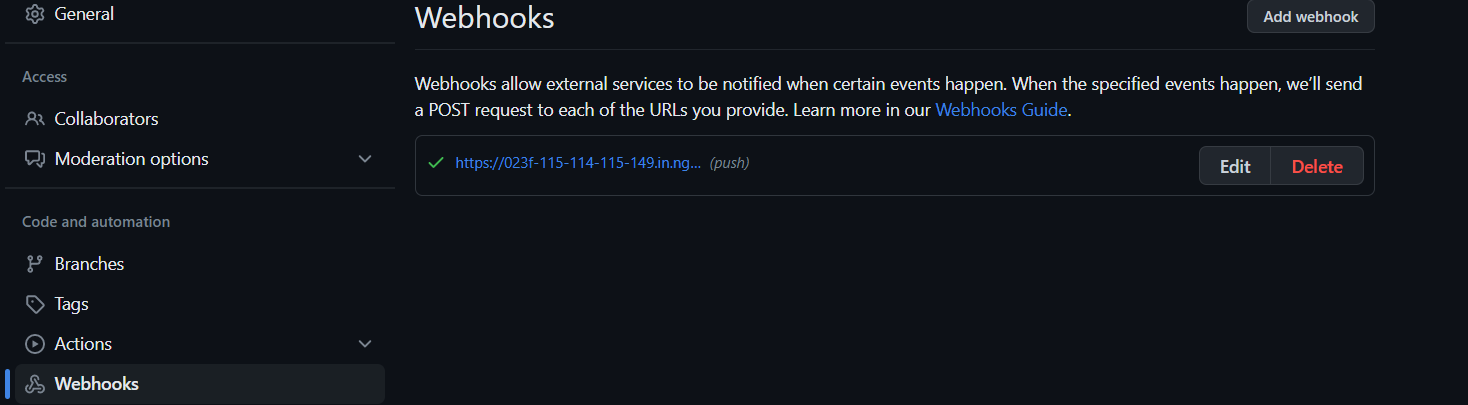
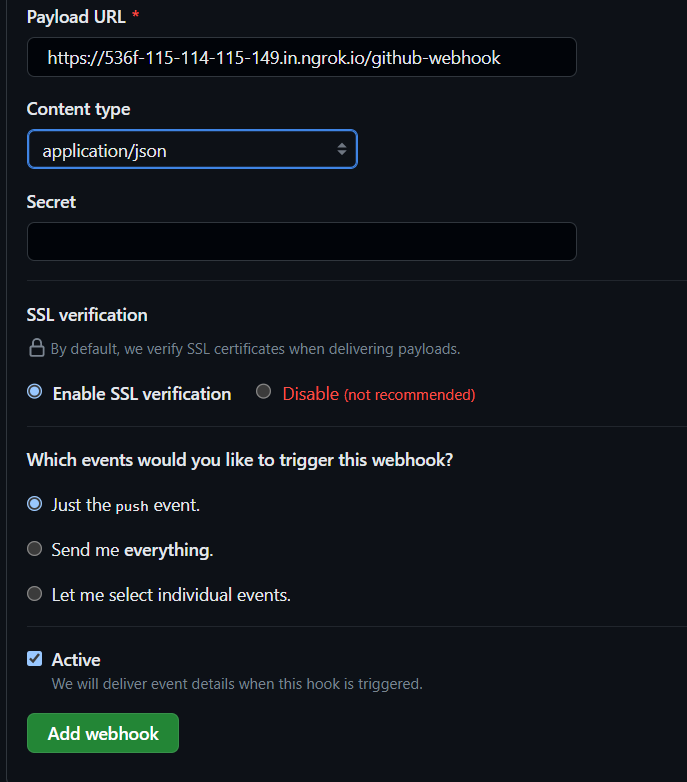
1. Create a Jenkins webserver and Open Git Hub.
2. Download NGROK.exe

* ngrok -version
* ngrok -help
* ngrok http 8080
* copy the forwarding URL

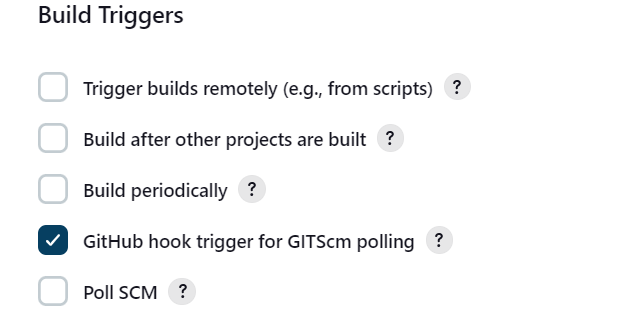
1. Go to a Git repository.
2. Go to settings and open webhook.
3. Add the URL /GitHub-webhook/
4. Open jenkins webserver, add a new project.
5. Add the git repository URL.
6. Add the Git Credentials.
7. Save and apply the project.
8. Jenkins fetches the code from the repository and builds.

Ex:

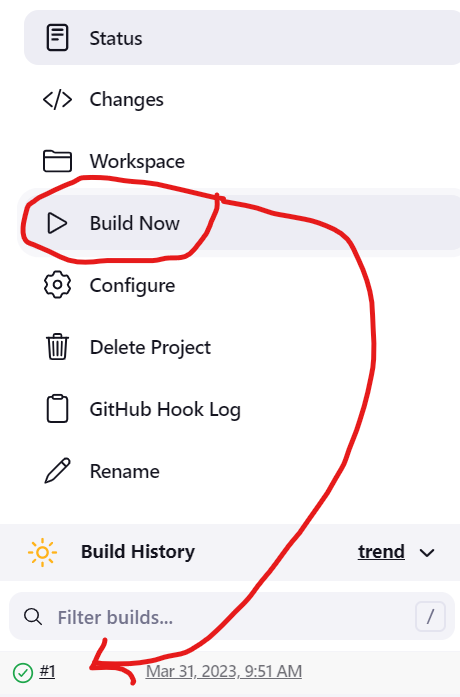
* Adding a webhook:

* 
* 
* Open Jenkins server, create a new job





* After defining the requirements, save and apply. The build happens automatically.



* Console Output:



To Manage Credentials:

