

Jenkins

Jenkins

→ Automation CI/CD tool.

developer

login fetch
Java app

push

github.com

Jenkins

Automation build tool

maven

Maven will compile & package the project.

Jenkins

If error is there
Jenkins will report
the error to the
developer.

Run the test cases in maven

Jenkins

Quality assurance

checkstyle
pmd
findbugs

Deploy

Jenkins

Jenkins → automation tool used for building, testing and deploying app.

→ Open source tool

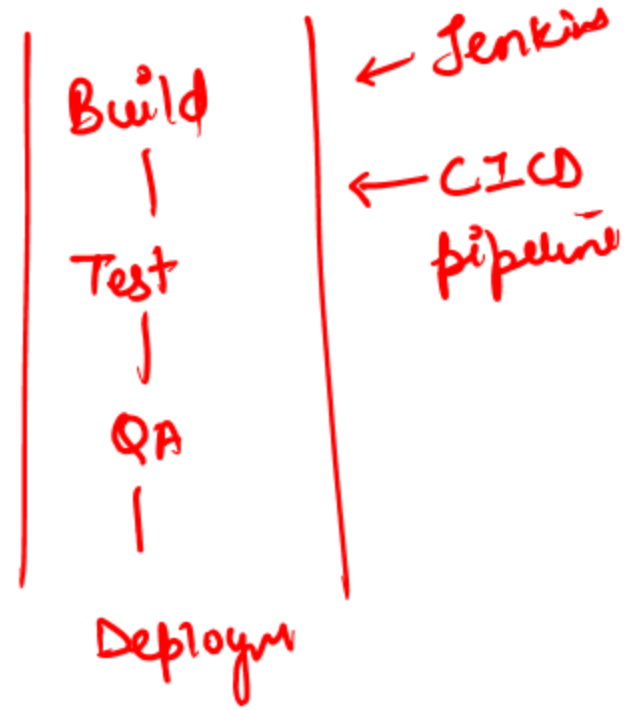
→ Jenkins software developed in Java.

→ Jenkins follow plugin based architecture.

for e.g. → mailer plugin.
maven plugin.

→ To automate the execution of pipeline
we write the script in groovy language.

→ Jenkins can be installed in windows, mac, ubuntu, linux



Maven → Build automation and project management tool

→ It helps building (compiling, packaging), testing Java app automatically
→ (Jar/war)

→ configuration file → pom.xml

- Dependencies
- plugins
- Project Information
- Build Instructions

} Developer

- <https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

lifecycle handles the creation of your project's web site.

A Build Lifecycle is Made Up of Phases

Each of these build lifecycles is defined by a different list of build phases, wherein a build phase represents a stage in the lifecycle.

For example, the default lifecycle comprises of the following phases (for a complete list of the lifecycle phases, refer to the [Lifecycle Reference](#)):

- `validate` - validate the project is correct and all necessary information is available
- `compile` - compile the source code of the project
- `test` - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
- `package` - take the compiled code and package it in its distributable format, such as a JAR.
- `verify` - run any checks on results of integration tests to ensure quality criteria are met
- `install` - install the package into the local repository, for use as a dependency in other projects locally
- `deploy` - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

These lifecycle phases (plus the other lifecycle phases not shown here) are executed sequentially to complete the `default` lifecycle. Given the lifecycle phases above, th

Gradle → competitor of maven

— build automation tool used to build, test and deploy the applications

— you can use gradle for Java, Kotlin, Android apps, Groovy

Gradle uses scripts — build.gradle (Groovy)
— build.gradle.kts (Kotlin)

Lab 1 : Install Jenkins in the ubuntu machine

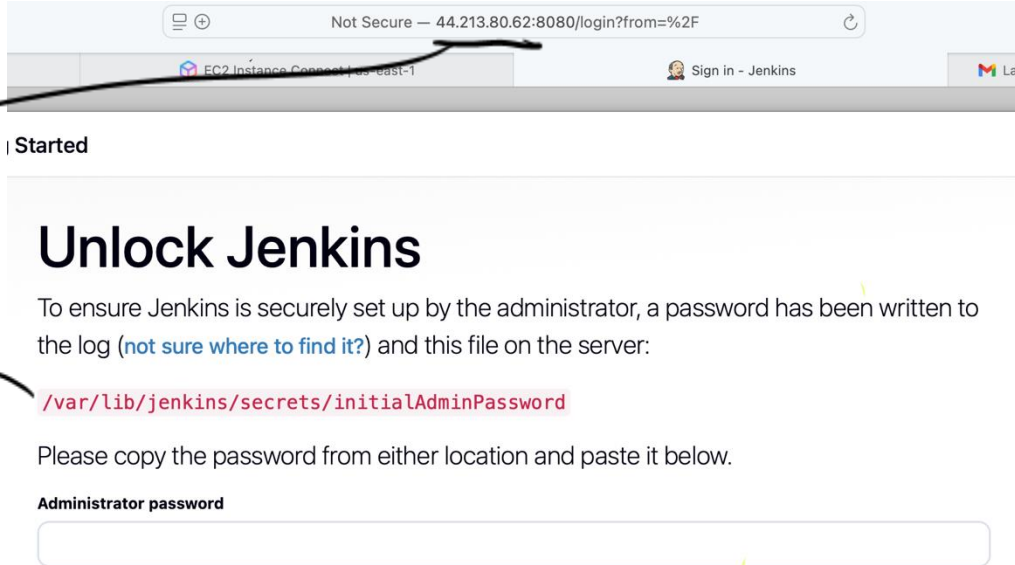
Use the script to download Jenkins

<https://raw.githubusercontent.com/akshu20791/Deployment-script/refs/heads/main/jenkins.sh>

Jenkins works on port 8080 so you need to allow traffic coming from port 8080 on ur machine

http:// public ip : 8080

*cat /var/lib - - - -
give you password*



- Install suggested plugins
- Put user , pass and other details and login to jenkins



Jenkins

+ New Item

Build History

Build Queue

No builds in the queue.

Build Executor Status

0/2

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job



Set up a distributed build

Set up an agent



Configure a cloud



Learn more about distributed builds



manage Jenkins



Add description

you want to create a new Job/Item/Project
e.g. you want to create a pipeline which will
compile - package - Test - QA - deploy

Whenever you
execute the Item
we call it as build.

Lab 2: User management in jenkins

Lets suppose you are working in a team and you have developer , testers ,interns , admin etc in the team to which u don't want to give your main Jenkins account. So we can create the users in Jenkins and give them the restricted permissions

There are multiple ways of doing it ...

1) Role based access control

2) Matrix based method

Doc link:

https://docs.google.com/document/d/1kFlN0dF1zX3MHTTz96JLefwzk0gi3KNPB_NLxs9x1jA/edit?usp=sharing

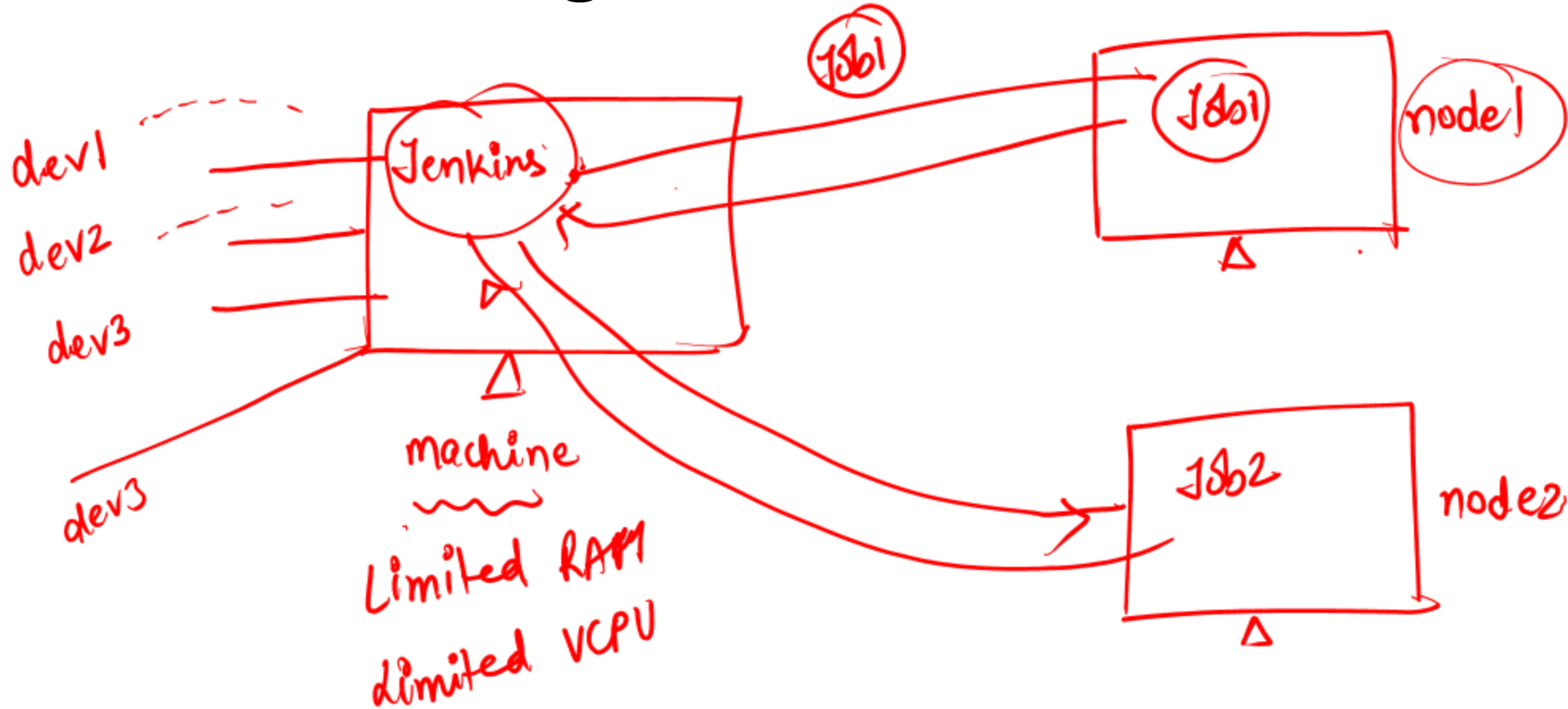
Lab 3: Create a simple hello world job

- We need to create a job in Jenkins which will print hello world message in the machine and we will see the logs from Jenkins

Solution:

<https://drive.google.com/file/d/178Ql7wdMHlcGA3abSHhOwWumFE7HgiUu/view?usp=sharing>

Lab 4: Working in master node architecture



Jenkins delegates the tasks to the nodes (machines) as so as to reduce the load on Jenkins machine (master)

- Go to aws account -> ec2 -> Instances -> Launch a new instance with all traffic enabled and name that machine as node machine
- Connect to the node machine

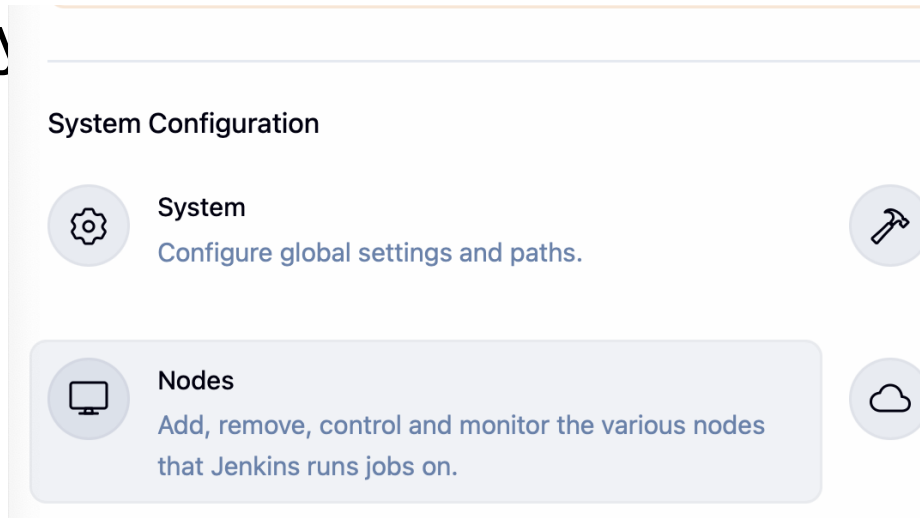
In the terminal write :

```
sudo su
```

```
apt update
```

```
apt install openjdk-17-jdk -y
```

- Go back to your Jenkins master machine -> manage Jenkins -> In sy _____ : nodes -> + NEW NODES



New node

Node name

akshat-node

Type

☒ Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

← any name



Save



Name ?

akshat-node

Description ?

How many jobs this node can execute concurrently

Plain text Preview

Number of executors ?

1

Remote root directory ?

/opt/akshat

any location
(Here your connection files are present)

Labels ?

node1

Identifier

Usage ?








Nodes

+ New Node

Configure Monitors



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	akshat-node		N/A	N/A	N/A	N/A	N/A 
	Built-In Node	Linux (amd64)	In sync	3.44 GiB	 0 B	3.44 GiB	0ms 

This node is not connected

click




 Status

 Delete Agent

 Configure

 Build History

 Load Statistics

 Log

Agent akshat-node

 Add description

Mark this node temporarily offline

Run from agent command line: (Unix)



```
curl -sO http://44.213.80.62:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://44.213.80.62:8080/ -secret
335036eec46ef461336cd8a16a1086959fd53d5a2ffdfc90bc341ea7e9c9a263 -name "akshat-node" -webSocket -workDir
"/opt/akshat"
```

copy this complete code & paste it in node

Build Executor Status

0/1 ▾

```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-66-234:/home/ubuntu# curl -sO http://44.213.80.62:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://44.213.80.62:8080/ -secret 335036eec46ef461336cd8a16a1086959fd53d5a2ffdfc90bc341ea7e9c9a263 -name "akshat-node" -webS
rkDir "/opt/akshat"
Nov 18, 2025 11:12:58 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /opt/akshat/remoting as a remoting work directory
Nov 18, 2025 11:12:58 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /opt/akshat/remoting
Nov 18, 2025 11:12:58 AM hudson.remoting.Launcher createEngine
INFO: Setting up agent: akshat-node
Nov 18, 2025 11:12:58 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3327.v868139a_d00e0
Nov 18, 2025 11:12:58 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /opt/akshat/remoting as a remoting work directory
Nov 18, 2025 11:12:58 AM hudson.remoting.Launcher$CuiListener status
INFO: WebSocket connection open
Nov 18, 2025 11:12:58 AM hudson.remoting.Launcher$CuiListener status
INFO: Connected

```

i-069660b9f5bf64f88 (node-akshat-jenkins)

PublicIPs: 18.206.16.190 PrivateIPs: 172.31.66.234

(node machine)



Jenkins

Manage Jenkins ▾ / Nodes

Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space
	akshat-node	Linux (amd64)	In sync	4.09 GiB	
	Built-In Node	Linux (amd64)	In sync	3.44 GiB	
Data obtained		35 sec	35 sec	35 sec	35 sec

X mark is removed

- Lets run the job in the node
- Go to ur Jenkins machine
- +new item

New Item

Enter an item name

mysecondjob ✓

Select an item type



Freestyle project ✓

Classic, general-purpose job type that checks out
post-build steps like archiving artifacts and sendin



Jenkins

mysecondjob

Configuration

Configure



General



Source Code Management



Triggers



Environment



Build Steps



Post-build Actions

Plain text [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

☒ Restrict where this project can be run ?

Label Expression ?

node1

Label node1 matches 1 node. Permissions or other restrictions provided by

label of the node



Jenkins

/ mysecondjob / Configuration

Configure



General



Source Code Management



Triggers



Environment



Build Steps



Post-build Actions

☐ 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 ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing,

Execute shell ?

Command

See [the list of available environment variables](#)

```
echo hello world
```

Save and Build



Jenkins

/ mysecondjob

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Credentials

mysecondjob

Permalinks

Builds

Today

#1 11:17 am



click (After build)

Jenkins / mysecondjob ▾ / #1

Status

</> Changes

Console Output

✓ Edit Build Information

🗑 Delete build '#1'

🕒 Timings

✅ #1 (18-Nov-2025, 11:17:39 am)

🕒 Started by user [admin](#)

🕒 This run spent:

- 3 ms waiting;
- 0.41 sec build duration;
- 0.42 sec total from scheduled to completion.

Go see the logs

🔍 ⚙️ 👤

✎ Add description Keep this build forever

Started 50 sec ago
Took [0.41 sec](#) on [akshat-node](#)

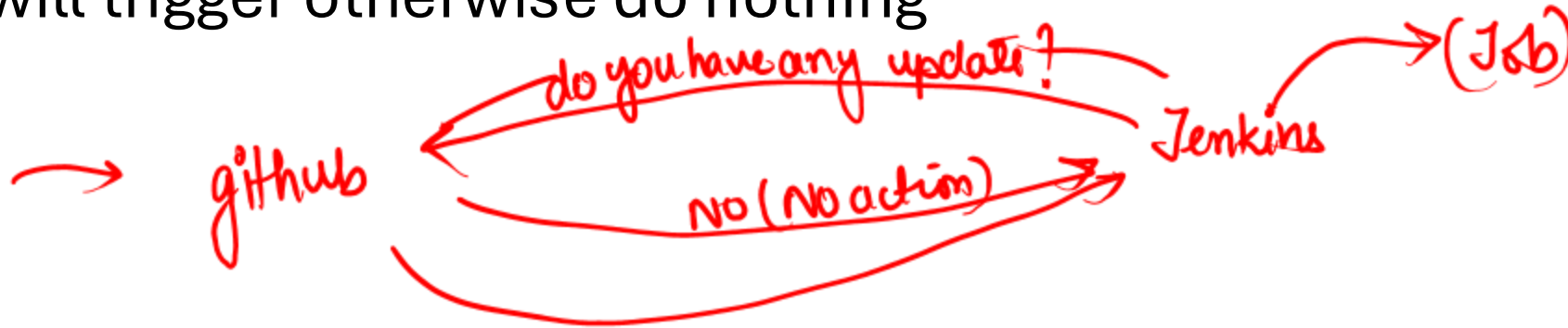
Hence proved that
our Job is executed
in akshatnode.

Lab 5: Automate the Jenkins job whenever the developer push the code to github

- As of now if the developer push the code to github it does not automatically build the project ...we need to manually build the project by clicking on build now button



- This could be done via multiple ways like :
 - > POLL SCM : Jenkins will keep checking the github for updates after a particular duration (CRON JOB) and if there is any update in github it will trigger otherwise do nothing



- > GITHUB WEBHOOK : Whenever there is any update in github ...its github who will go to Jenkins and trigger the job....



LAB 5.1 : POLL SCM LAB

- Go to Jenkins -> +NEW ITEM -> Give any name and freestyle job

Enter an item name

pollscmlab

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes post-build steps like archiving artifacts and sending email notifications.

FORK THE GITHUB REPO :

<https://github.com/akshu20791/apachewebsite>

(Google if you don't know how to fork)

Jenkins / pollscmlab / Configuration

Configure

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

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

☐ None

☒ Git ?

Repositories ?

Repository URL ?

<https://github.com/akshu20791/apachewebsite>

Please enter Git repository.

Credentials ?

- none -

Advanced ▾

+ Add Repository

here put the link
of your forked githubs
repo.

Jenkins / pollscmlab / Configuration

Configure

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

Triggers

Set up automated actions that start your build based on specific e

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

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

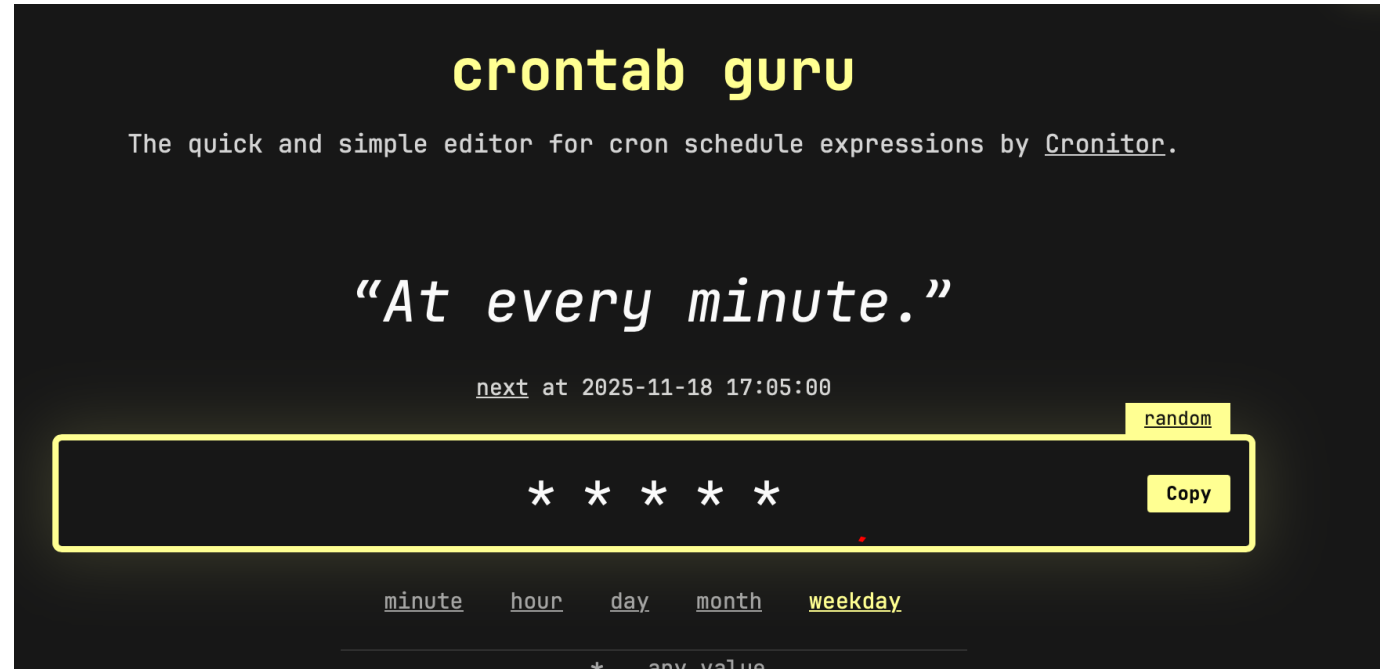
☒ Poll SCM ?

Schedule ?

here cron job will up

- <https://crontab.guru>

• * * * *



☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☒ Poll SCM ?

Schedule ?

*****|

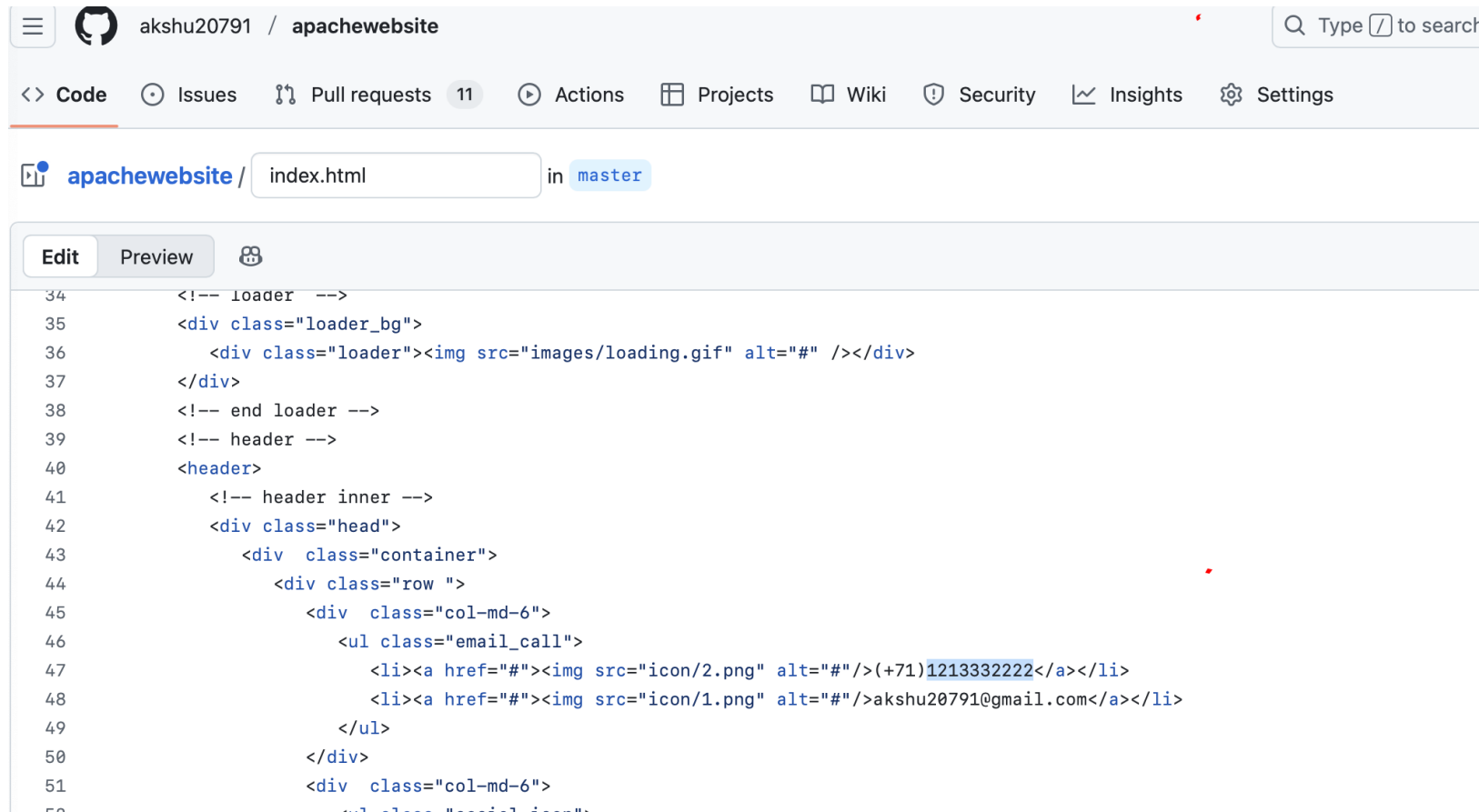
In build steps -> add build steps

execute shell

echo hello world

Save — Build now

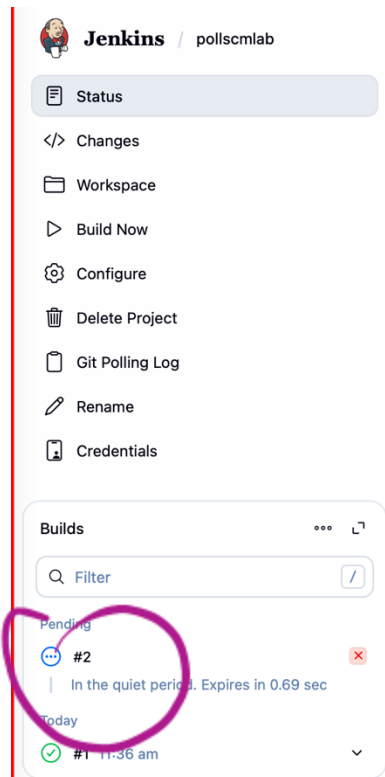
- Go to the github account -> go to the repo which u have forked -> edit the index.html file and add any content



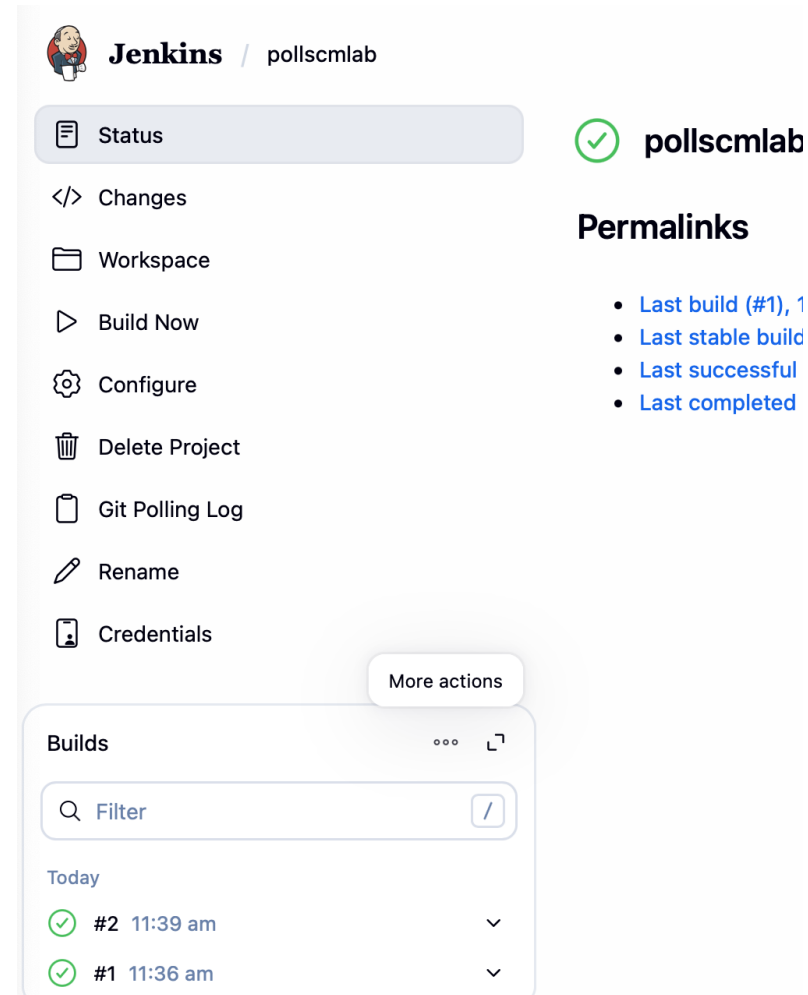
```
34      <!-- loader -->
35      <div class="loader_bg">
36          <div class="loader"></div>
37      </div>
38      <!-- end loader -->
39      <!-- header -->
40      <header>
41          <!-- header inner -->
42          <div class="head">
43              <div class="container">
44                  <div class="row ">
45                      <div class="col-md-6">
46                          <ul class="email_call">
47                              <li><a href="#">(+71)1213332222</a></li>
48                              <li><a href="#">akshu20791@gmail.com</a></li>
49                          </ul>
50                      </div>
51                      <div class="col-md-6">
```

Commit

- Keep checking the Jenkins ...u will see automatically the Jenkins job will be trigger after 1 min because u have made some changes in github which is automatically identified by Jenkins.



The screenshot shows the Jenkins interface for the 'pollscmlab' project. On the left, a sidebar contains links: Status, Changes, Workspace, Build Now, Configure, Delete Project, Git Polling Log, Rename, and Credentials. The main area displays the project status as 'pollscmlab' with a green checkmark. Below this, a 'Permalinks' section lists links for the last build, stable build, successful build, and completed build, all indicating they occurred 1 min 45 sec ago. At the bottom, a 'Builds' section shows a list of builds. Build #2 is in a 'Pending' state, indicated by a blue circle with a clock icon, and is circled in purple. It shows a message 'In the quiet period. Expires in 0.69 sec'. Build #1 is shown below it as a completed build at 11:36 am.

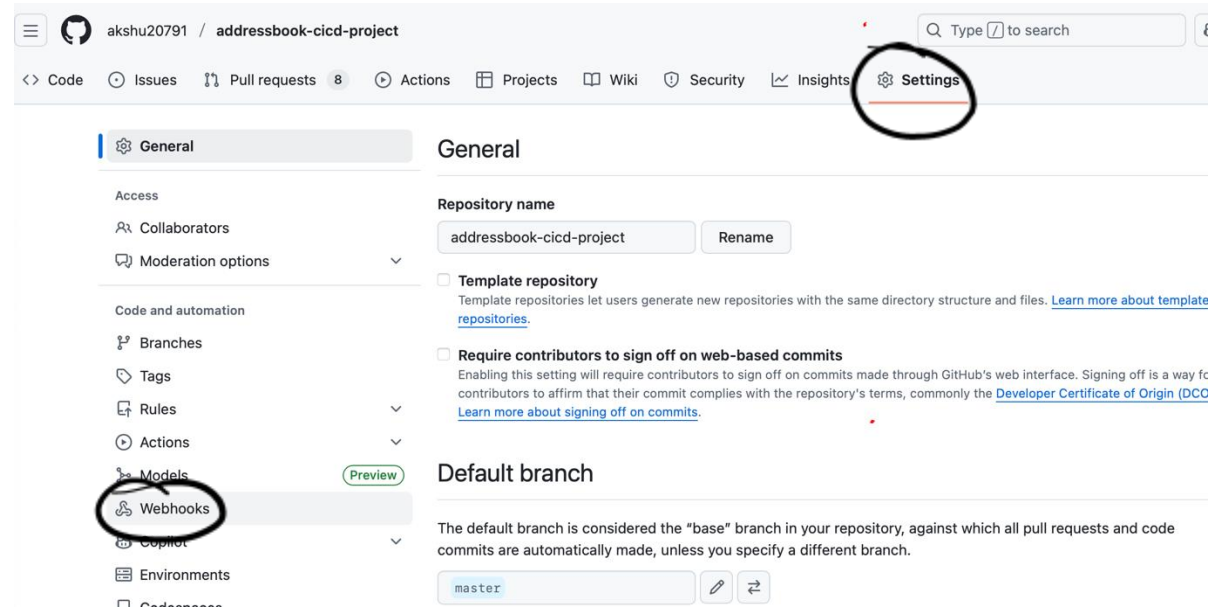


This screenshot shows the same Jenkins project page after a refresh. The 'Builds' section now shows build #2 as completed at 11:39 am, marked with a green checkmark. The 'Permalinks' section remains the same. The sidebar and project status are also visible. A 'More actions' button is located above the builds list.

Lab 5.2: Github webhooks

- Fork <https://github.com/akshu20791/addressbook-cicd-project> (you will be using ur forked github repo everywhere)

Go to github -> go to ur githubrepo -> settings → webhooks



Click on Add webhook -> in payload url : put your Jenkins url
http://publicip:8080/github-webhook/

my Jenkins url
(you have to use yours)

do not forget to put / in the end.

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

http://44.213.80.62:8080/github-webhook/

Content type *

application/json ✓

Secret

SSL verification

☐ By default, we verify SSL certificates when delivering payloads.

☐ Enable SSL verification ☒ Disable (not recommended)

Which events would you like to trigger this webhook?

☐ Just the push event.

☒ Send me everything.

☐ Let me select individual events.

☒ Active

We will deliver event details when this hook is triggered.

Add webhook

- Go to Jenkins -> + new item -> free style item

Jenkins / mynewite / Configuration

Configure

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

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/akshu20791/addressbook-cicd-project

! Please enter Git repository.

Credentials ?

None

Here your forked github repo will come

In trigger -> select GitHub hook trigger for GITScm polling

In build step -> add build step -> Execute shell -> echo hello world

save

Jenkins / mynewite / Configuration

Configure

- General
- Source Code Management
- Triggers**
- Environment
- Build Steps

Triggers

Set up automated actions that start your build based on specific events

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

- Now build the job once
- Now go to github and edit readme file and add any content in it

