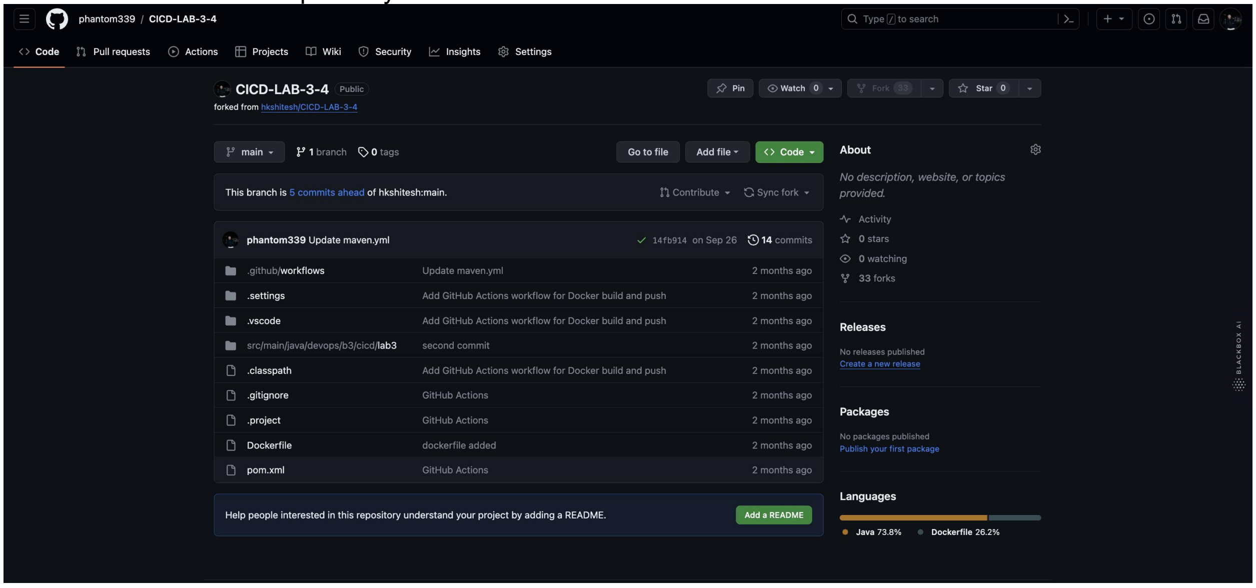
Experiment 3
Maven Build using GitHub Actions

Aim

Set up a GitHub Actions workflow to automatically build a Maven project whenever changes are
pushed to a GitHub repository.

Steps

1. Create a maven project
2. Or fork the Git Hub Repository



forked from hkshitesh/CICD-LAB-3-4

This branch is 5 commits ahead of hkshitesh:main.

phantom339 Update maven.yml ✓ 14fb914 on Sep 26 14 commits

.github/workflows	Update maven.yml	2 months ago
.settings	Add GitHub Actions workflow for Docker build and push	2 months ago
.vscode	Add GitHub Actions workflow for Docker build and push	2 months ago
src/main/java/devops/b3/cicd/lab3	second commit	2 months ago
.classpath	Add GitHub Actions workflow for Docker build and push	2 months ago
.gitignore	GitHub Actions	2 months ago
.project	GitHub Actions	2 months ago
Dockerfile	dockerfile added	2 months ago
pom.xml	GitHub Actions	2 months ago

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

Activity
0 stars
0 watching
33 forks

Releases

No releases published
Create a new release

Packages

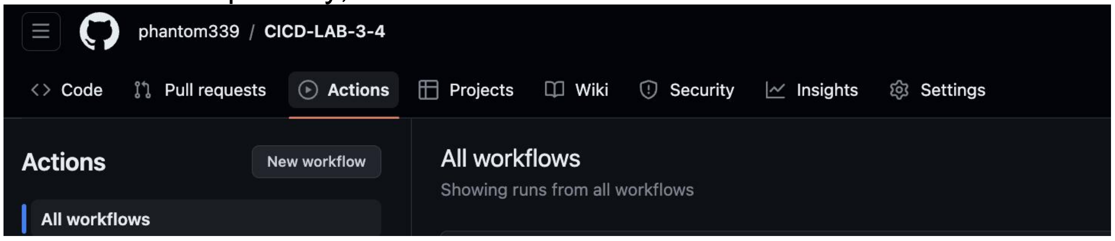
No packages published
Publish your first package

Languages

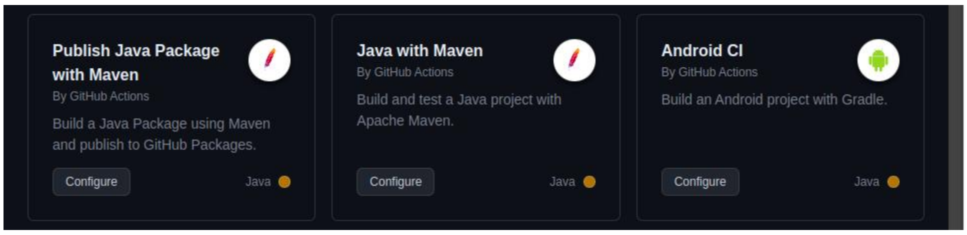
● Java 73.8% ● Dockerfile 26.2%

3.Setup GitHub workflow for the project

a. On GitHub repository, Go to Actions tab



b. Select Java with Maven and click on Configure

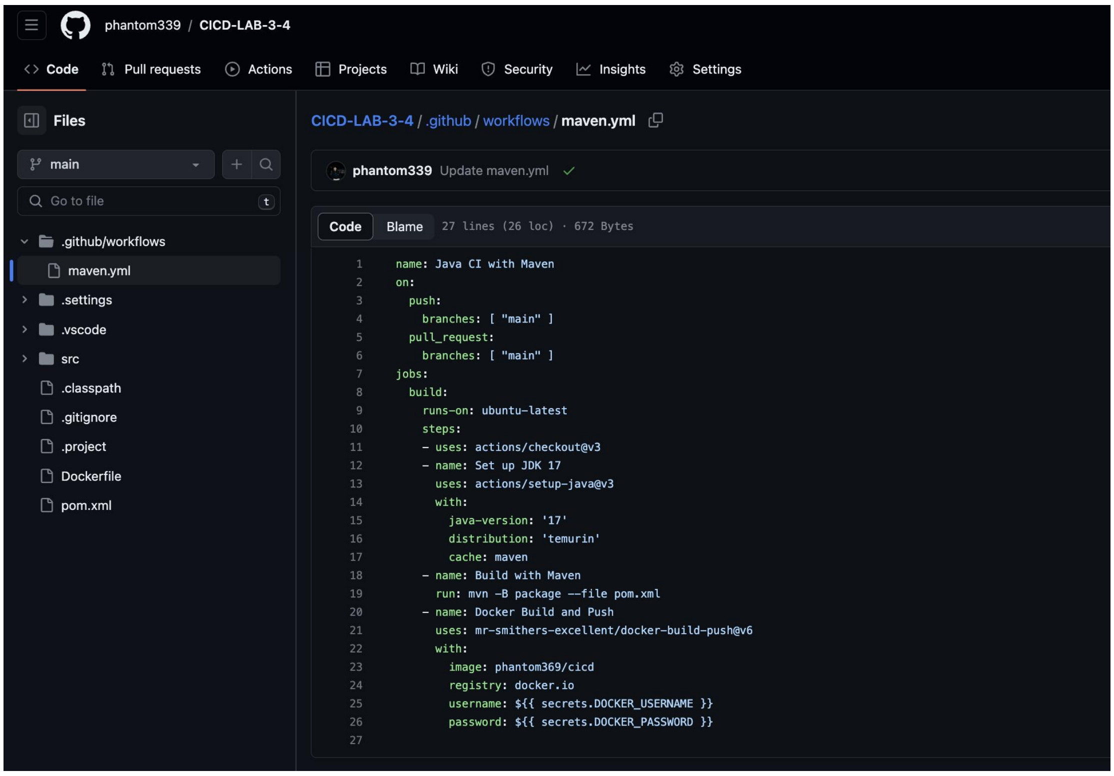


Publish Java Package with Maven
By GitHub Actions
Build a Java Package using Maven and publish to GitHub Packages.

Java with Maven
By GitHub Actions
Build and test a Java project with Apache Maven.

Android CI
By GitHub Actions
Build an Android project with Gradle.

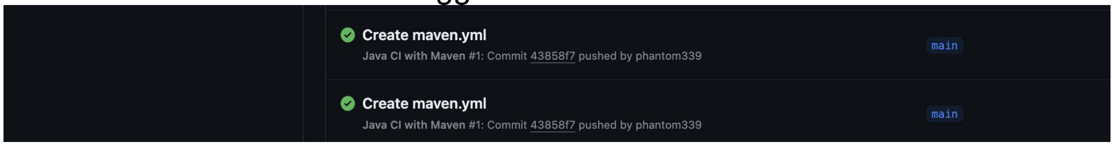
c. Configure the maven.yml file & commit the changes



phantom339 Update maven.yml

```
name: Java CI with Maven
on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 17
      uses: actions/setup-java@v3
      with:
        java-version: '17'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
      run: mvn -B package --file pom.xml
    - name: Docker Build and Push
      uses: mr-smithers-excellent/docker-build-push@v6
      with:
        image: phantom369/cicd
        registry: docker.io
        username: ${{ secrets.DOCKER_USERNAME }}
        password: ${{ secrets.DOCKER_PASSWORD }}

```

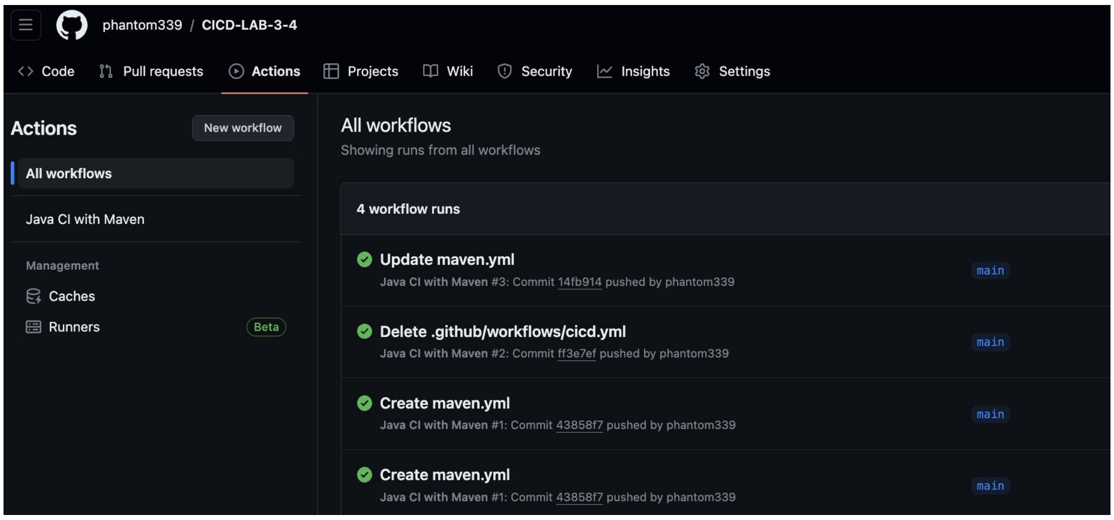
d. An automatic build will be triggered



Create maven.yml
Java CI with Maven #1: Commit 43858f7 pushed by phantom339 main

Create maven.yml
Java CI with Maven #1: Commit 43858f7 pushed by phantom339 main

Create some changes on the Repository now and the changes will be made into builds automatically-



All workflows
Showing runs from all workflows

4 workflow runs

Update maven.yml
Java CI with Maven #3: Commit 14fb914 pushed by phantom339 main

Delete .github/workflows/cicd.yml
Java CI with Maven #2: Commit ff3e7ef pushed by phantom339 main

Create maven.yml
Java CI with Maven #1: Commit 43858f7 pushed by phantom339 main

Create maven.yml
Java CI with Maven #1: Commit 43858f7 pushed by phantom339 main

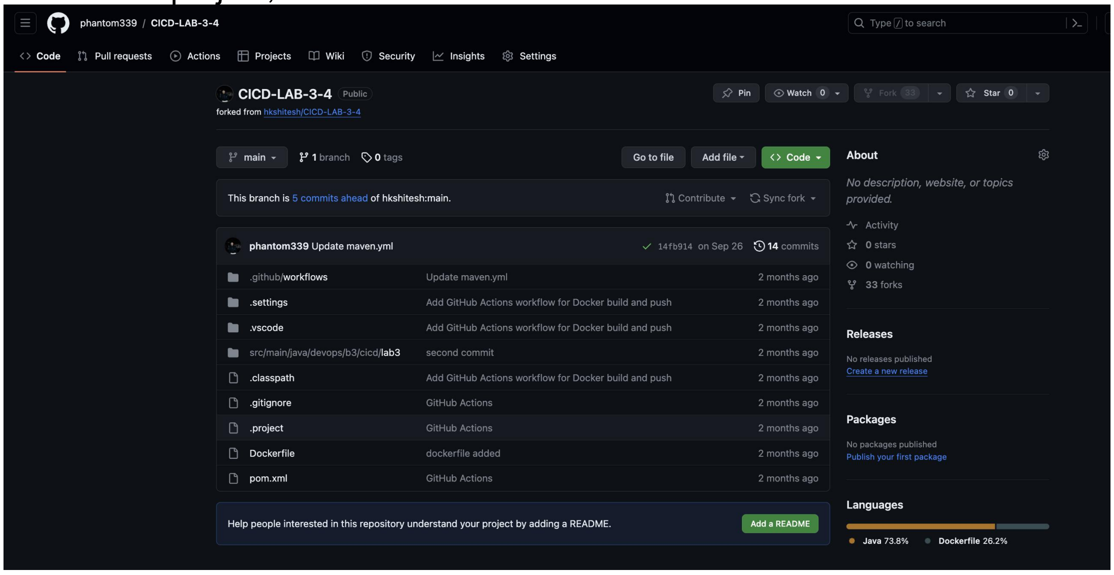
Experiment 4
Docker Build and Push using GitHub Actions

Aim

Set up a GitHub Actions workflow to automatically build a Docker image from a Dockerfile in your
GitHub repository and push it to a container registry (e.g., Docker Hub).

Steps

1. Create a project, Add Dockerfile and Push it on GitHub



phantom339 / CICD-LAB-3-4

CICD-LAB-3-4 Public
forked from hkshitesh/CICD-LAB-3-4

main · 1 branch · 0 tags

This branch is 5 commits ahead of hkshitesh:main.

phantom339 Update maven.yml ✓ 14fb914 on Sep 26 · 14 commits

.github/workflows	Update maven.yml	2 months ago
.settings	Add GitHub Actions workflow for Docker build and push	2 months ago
.vscode	Add GitHub Actions workflow for Docker build and push	2 months ago
src/main/java/devops/b3/cicd/lab3	second commit	2 months ago
.classpath	Add GitHub Actions workflow for Docker build and push	2 months ago
.gitignore	GitHub Actions	2 months ago
.project	GitHub Actions	2 months ago
Dockerfile	dockerfile added	2 months ago
pom.xml	GitHub Actions	2 months ago

Help people interested in this repository understand your project by adding a README.

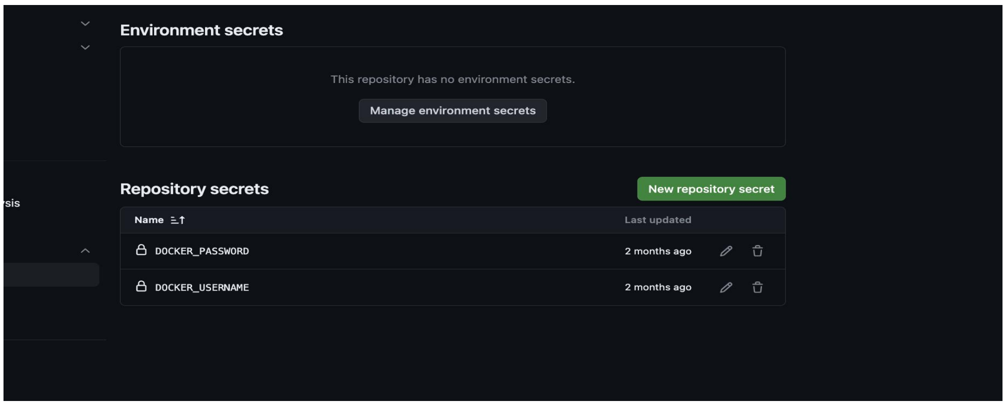
About
No description, website, or topics provided.
Activity
0 stars
0 watching
33 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Languages
● Java 73.8% ● Dockerfile 26.2%

2. Create Docker Hub Access Token
a. Log in to your Docker Hub account.
b. Go to your account settings and click on the "Security" tab.
c. Under "Access Tokens," click "New Access Token." Give it a name, select the required
permissions (e.g., "Write" for pushing Docker images), and click "Create."
d. Copy the generated access token. You will need it to authenticate with Docker Hub in your
GitHub Actions workflow.



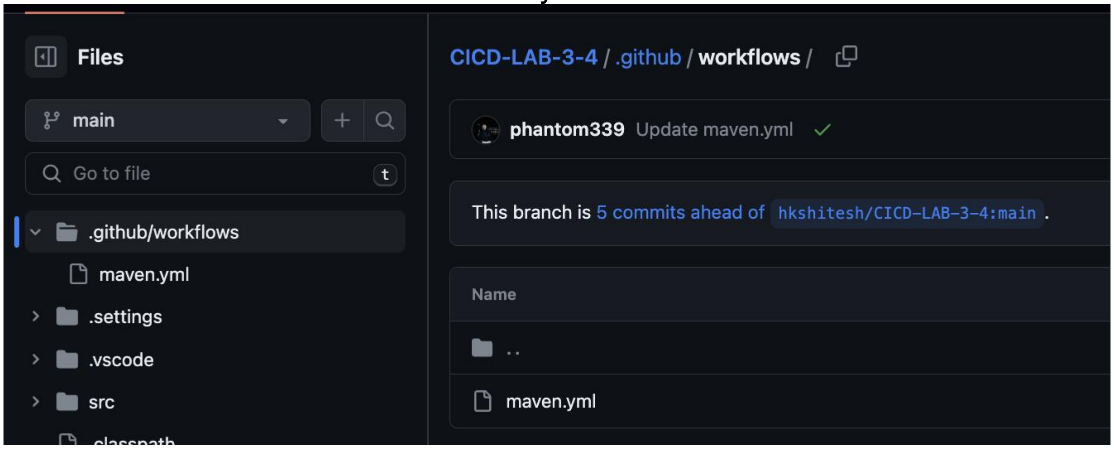
Environment secrets

This repository has no environment secrets.

Repository secrets

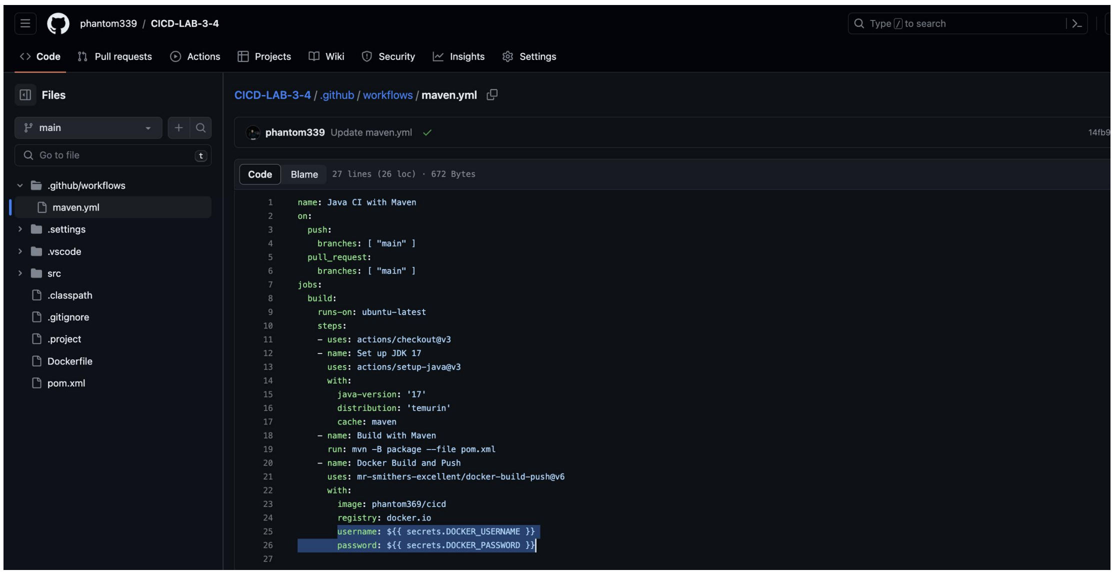
Name	Last updated
DOCKER_PASSWORD	2 months ago
DOCKER_USERNAME	2 months ago

Create a YAML file inside this directory \



This branch is 5 commits ahead of `hkshitesh/CICD-LAB-3-4:main`.

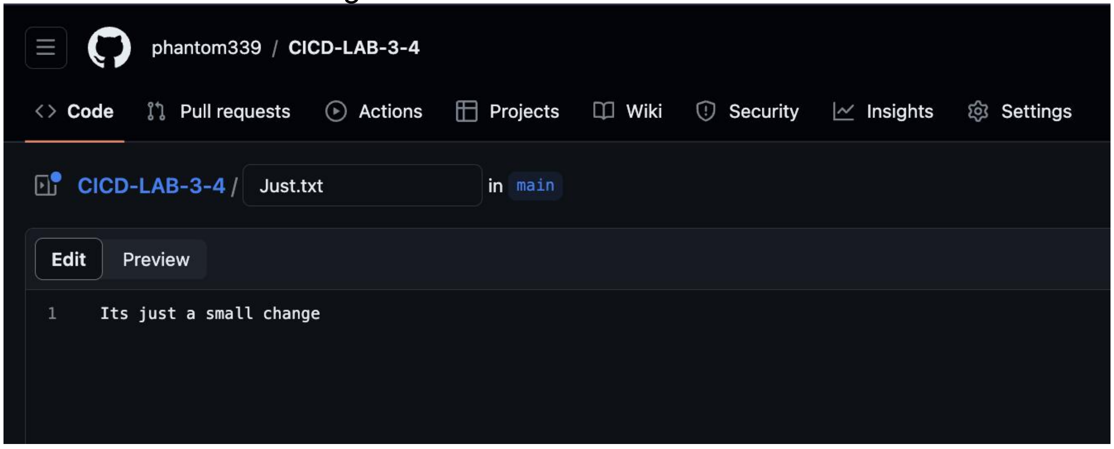
Write the necessary configuration in it



27 lines (26 loc) · 672 Bytes

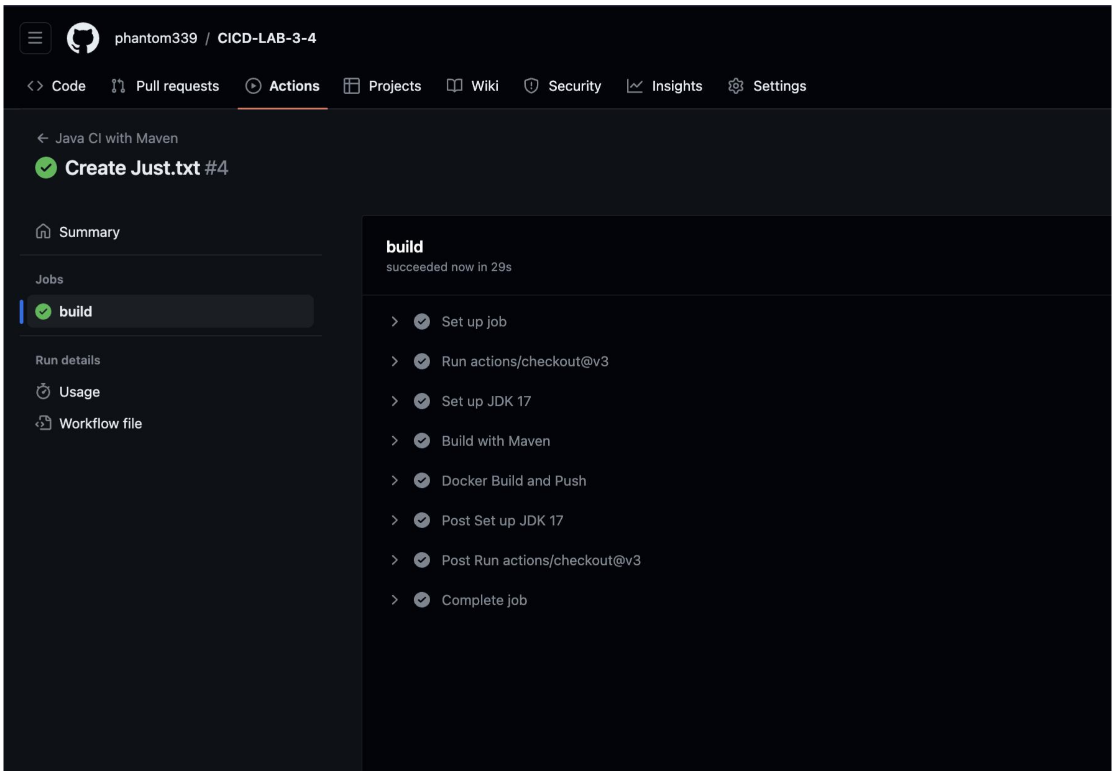
```yaml
name: Java CI with Maven
on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 17
      uses: actions/setup-java@v3
      with:
        java-version: '17'
        distribution: 'temurin'
        cache: maven
    - name: Build with Maven
      run: mvn -B package --file pom.xml
    - name: Docker Build and Push
      uses: mr-smithers-excellent/docker-build-push@v6
      with:
        image: phantom369/cicd
        registry: docker.io
        username: ${{ secrets.DOCKER_USERNAME }}
        password: ${{ secrets.DOCKER_PASSWORD }}

```

Commit and Push Changes



```
Its just a small change
```

. Check the Workflow Status


phantom339 / CICD-LAB-3-4

Code · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

← Java CI with Maven

Create Just.txt #4

Summary

Jobs

build

Run details

Usage

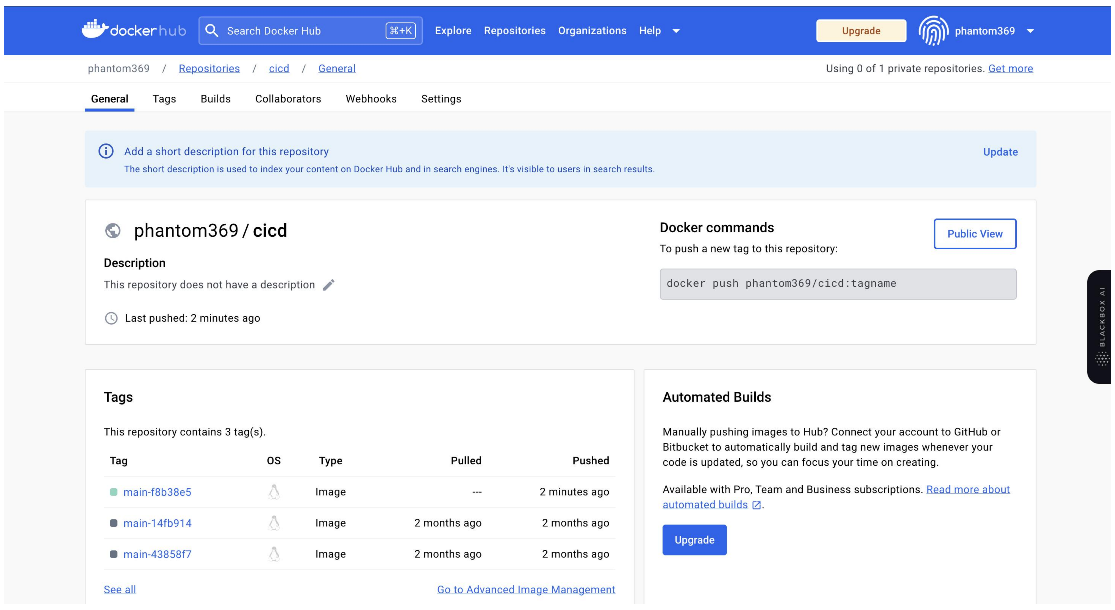
Workflow file

build

succeeded now in 29s

- Set up job
- Run actions/checkout@v3
- Set up JDK 17
- Build with Maven
- Docker Build and Push
- Post Set up JDK 17
- Post Run actions/checkout@v3
- Complete job

. Verify the Docker Image on Docker Hub


phantom369 / Repositories / cicd / General — Using 0 of 1 private repositories. Get more

General · Tags · Builds · Collaborators · Webhooks · Settings

Add a short description for this repository — Update

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

phantom369 / cicd

Description

This repository does not have a description

Last pushed: 2 minutes ago

Docker commands — Public View

To push a new tag to this repository:

```
docker push phantom369/cicd:tagname
```

Tags

This repository contains 3 tag(s).

Tag	OS	Type	Pulled	Pushed
main-f8b38e5		Image	---	2 minutes ago
main-14fb914		Image	2 months ago	2 months ago
main-43858f7		Image	2 months ago	2 months ago

See all — Go to Advanced Image Management

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

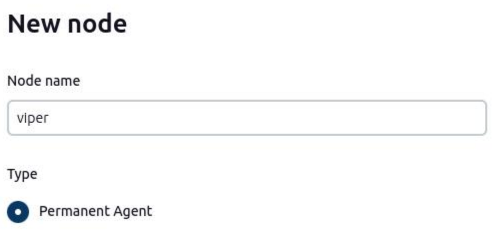
Available with Pro, Team and Business subscriptions. Read more about automated builds.

Upgrade

Experiment 5
Jenkins Master-Slave

Aim
To understand the Master-Slave methodology of Jenkins

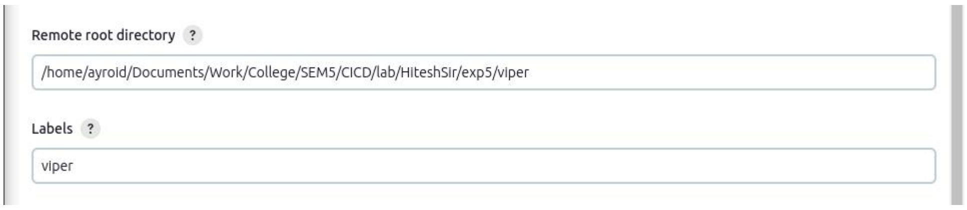
Steps
1. Create a Slave Node on Jenkins
a. Go to Dashboard > Manage Jenkins > Nodes > New Node
b. Give a name to the node, select Permanent Agent & click Create


New node

Node name: viper

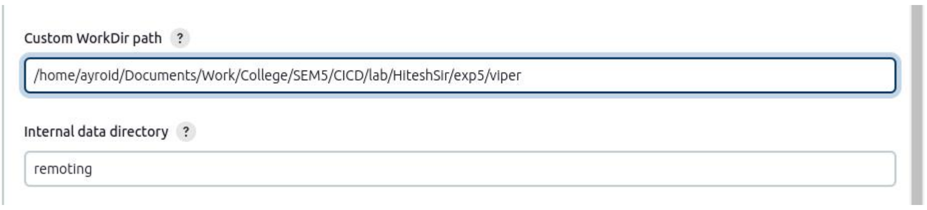
Type: ● Permanent Agent

c. Provide the root directory path & label


Remote root directory: /home/ayroid/Documents/Work/College/SEM5/CICD/lab/HiteshSir/exp5/viper

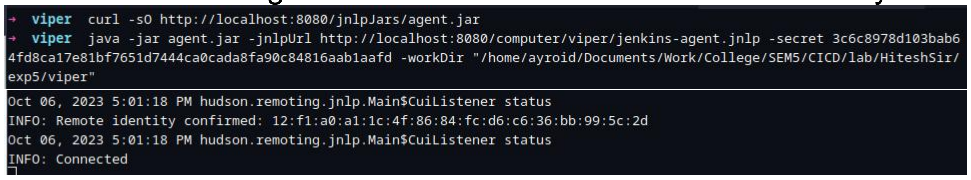
Labels: viper

d. Provide Custom WorkDir path

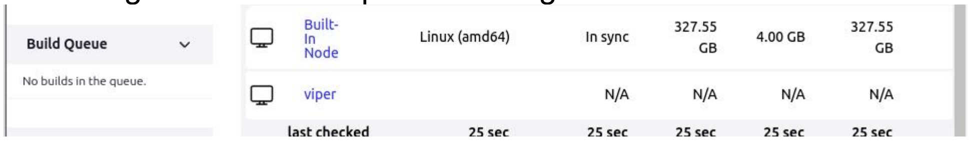

Custom WorkDir path: /home/ayroid/Documents/Work/College/SEM5/CICD/lab/HiteshSir/exp5/viper

Internal data directory: remoting

e. Run the following commands at the location of root directory


```
→ viper  curl -sO http://localhost:8080/jnlpJars/agent.jar
→ viper  java -jar agent.jar -jnlpUrl http://localhost:8080/computer/viper/jenkins-agent.jnlp -secret 3c6c8978d103bab6
4fd8ca17e81bf7651d7444ca0cada8fa90c84816aab1aafd -workDir "/home/ayroid/Documents/Work/College/SEM5/CICD/lab/HiteshSir/
exp5/viper"
Oct 06, 2023 5:01:18 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 12:f1:a0:a1:1c:4f:86:84:fc:d6:c6:36:bb:99:5c:2d
Oct 06, 2023 5:01:18 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

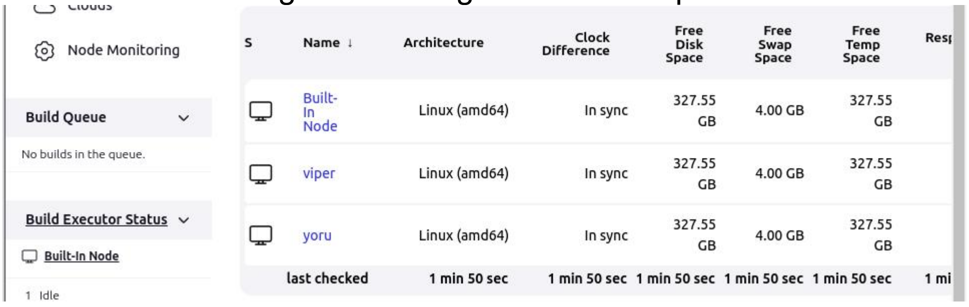
f. Your agent should be up and running


Build Queue

No builds in the queue.

🖥	Built-In Node	Linux (amd64)	In sync	327.55 GB	4.00 GB	327.55 GB
🖥	viper		N/A	N/A	N/A	N/A
	last checked	25 sec	25 sec	25 sec	25 sec	25 sec

2. Create another agent following the above steps



Node Monitoring

Build Queue

No builds in the queue.

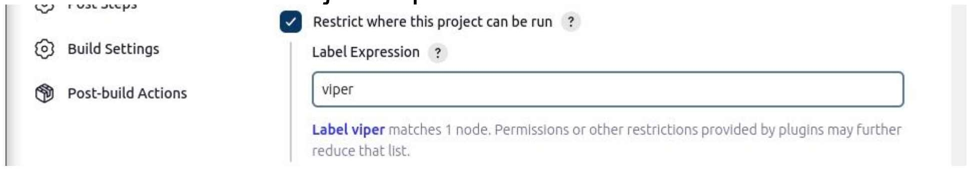
Build Executor Status

Built-In Node

1 Idle

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Resp
	Built-In Node	Linux (amd64)	In sync	327.55 GB	4.00 GB	327.55 GB	
	viper	Linux (amd64)	In sync	327.55 GB	4.00 GB	327.55 GB	
	yoru	Linux (amd64)	In sync	327.55 GB	4.00 GB	327.55 GB	
	last checked	1 min 50 sec	1 min 50 sec	1 min 50 sec	1 min 50 sec	1 min 50 sec	1 mi

3. Create a Maven Project Pipeline and restrict it to run on one of the agents



Build Settings

Post-build Actions

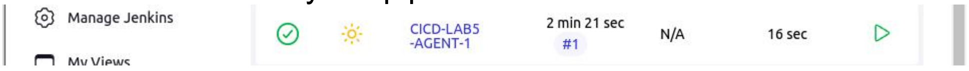
☑ Restrict where this project can be run ?

Label Expression ?

viper

Label viper matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

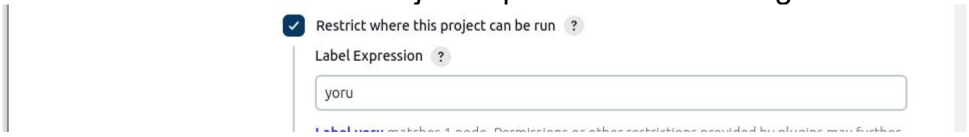
4. Run a build to verify the pipeline



Manage Jenkins

✓	☀	CICD-LAB5 -AGENT-1	2 min 21 sec #1	N/A	16 sec	▷

5. Create another Maven Project Pipeline on the 2nd agent

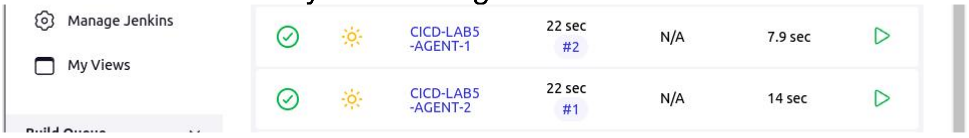


☑ Restrict where this project can be run ?

Label Expression ?

yoru

6. Run a build to verify both the agents



Manage Jenkins

My Views

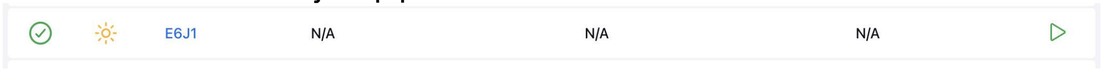
✓	☀	CICD-LAB5 -AGENT-1	22 sec #2	N/A	7.9 sec	▷
✓	☀	CICD-LAB5 -AGENT-2	22 sec #1	N/A	14 sec	▷

Experiment 6
Job Chaining in Jenkins

Aim

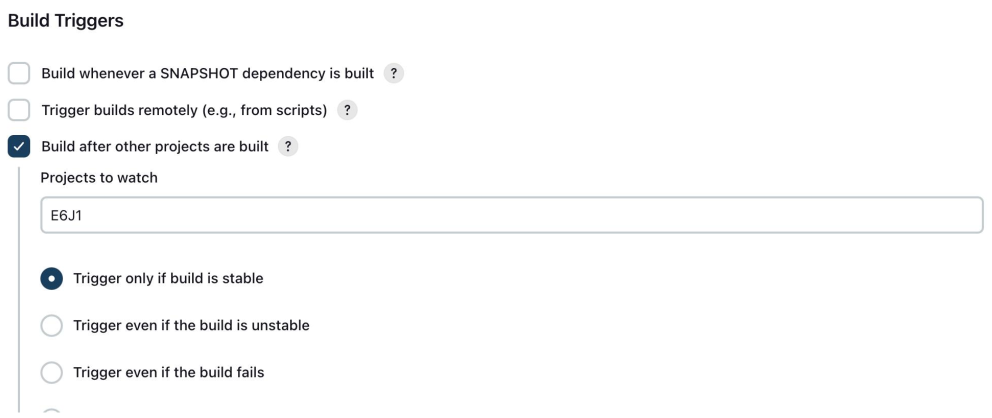
To understand and practice job chaining in Jenkins.

1. Create a Maven Project pipeline E6J1



✓	☀	E6J1	N/A	N/A	N/A	▷

2. Create another pipeline E6J2
a. Choose Build Triggers > Build after other projects are built



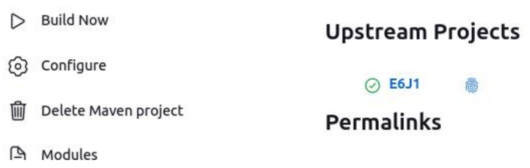
Build Triggers

- [ ] Build whenever a SNAPSHOT dependency is built ?
- [ ] Trigger builds remotely (e.g., from scripts) ?
- [x] Build after other projects are built ?

Projects to watch: E6J1

- [✓ Trigger only if build is stable]
- Trigger even if the build is unstable
- Trigger even if the build fails

b. Previous job is now showing as Upstream Project in the new job

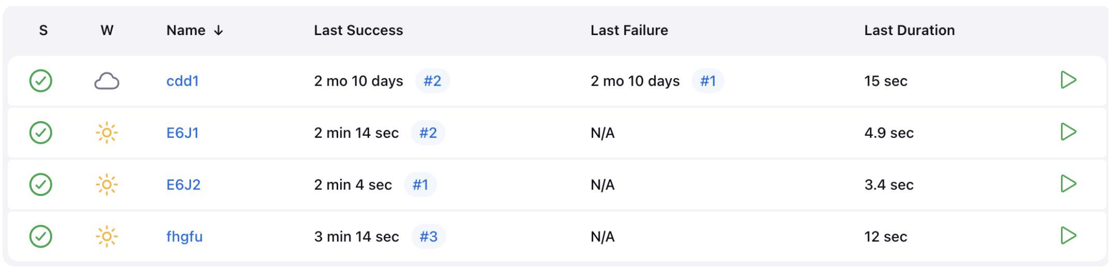


Upstream Projects

✓ E6J1

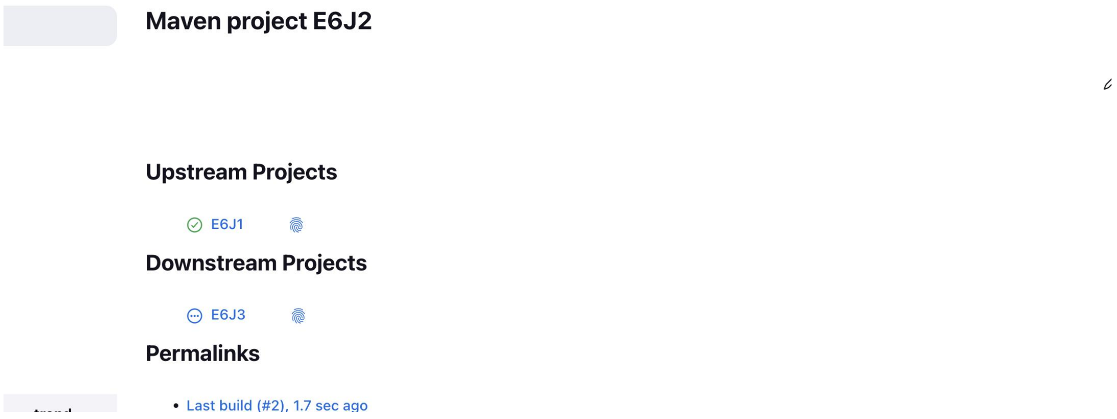
Permalinks

2. Test the chain by building the first pipeline



S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☁	cdd1	2 mo 10 days #2	2 mo 10 days #1	15 sec	▷
✓	☀	E6J1	2 min 14 sec #2	N/A	4.9 sec	▷
✓	☀	E6J2	2 min 4 sec #1	N/A	3.4 sec	▷
✓	☀	fhgfu	3 min 14 sec #3	N/A	12 sec	▷

3. Create 1 more pipelines E6J3, creating a longer and branched chain



Maven project E6J2

Upstream Projects

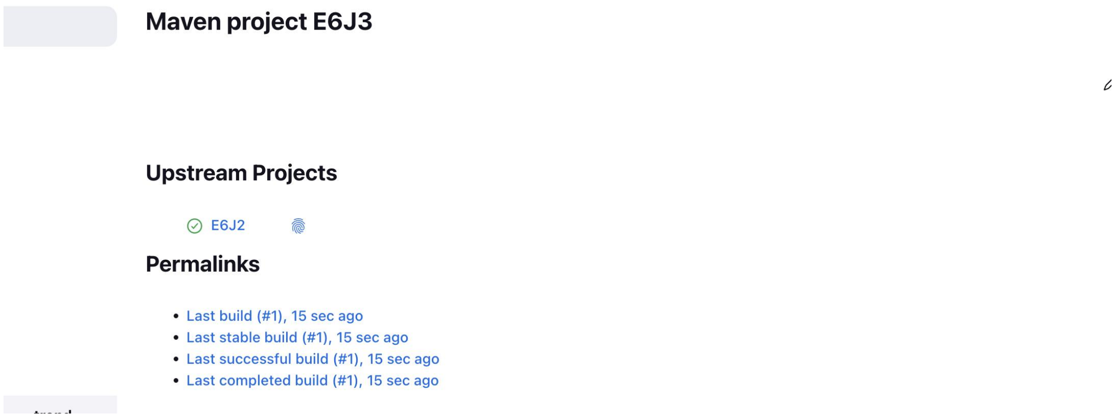
- E6J1

Downstream Projects

- E6J3

Permalinks

- Last build (#2), 1.7 sec ago



Maven project E6J3

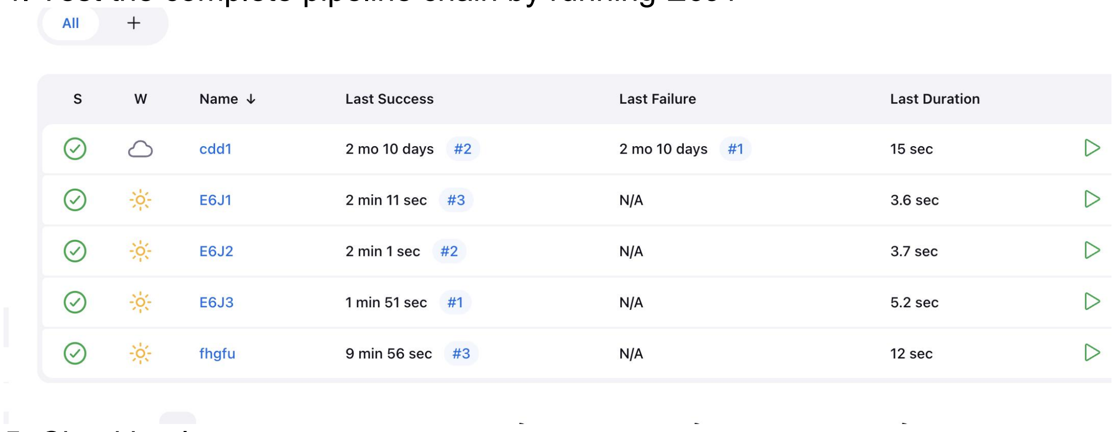
Upstream Projects

- E6J2

Permalinks

- Last build (#1), 15 sec ago
- Last stable build (#1), 15 sec ago
- Last successful build (#1), 15 sec ago
- Last completed build (#1), 15 sec ago

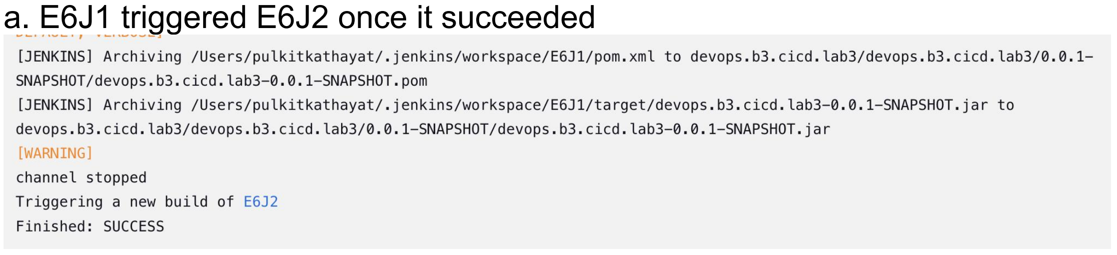
4. Test the complete pipeline chain by running E6J1



S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☁	cdd1	2 mo 10 days #2	2 mo 10 days #1	15 sec	▷
✓	☀	E6J1	2 min 11 sec #3	N/A	3.6 sec	▷
✓	☀	E6J2	2 min 1 sec #2	N/A	3.7 sec	▷
✓	☀	E6J3	1 min 51 sec #1	N/A	5.2 sec	▷
✓	☀	fhgfu	9 min 56 sec #3	N/A	12 sec	▷

5. Checking logs

a. E6J1 triggered E6J2 once it succeeded

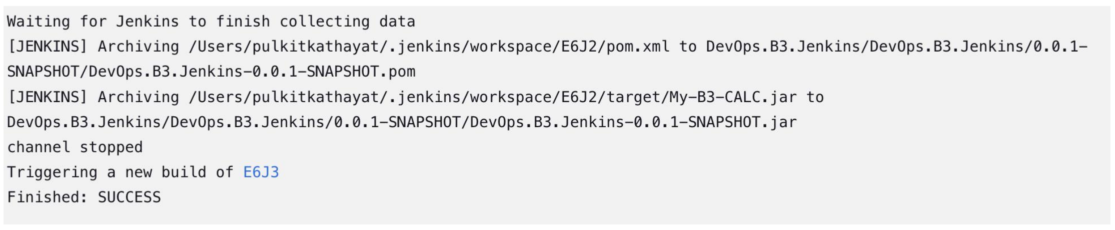


```
[JENKINS] Archiving /Users/pulkitkathayat/.jenkins/workspace/E6J1/pom.xml to devops.b3.cicd.lab3/devops.b3.cicd.lab3/0.0.1-
SNAPSHOT/devops.b3.cicd.lab3-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving /Users/pulkitkathayat/.jenkins/workspace/E6J1/target/devops.b3.cicd.lab3-0.0.1-SNAPSHOT.jar to
devops.b3.cicd.lab3/devops.b3.cicd.lab3/0.0.1-SNAPSHOT/devops.b3.cicd.lab3-0.0.1-SNAPSHOT.jar
[WARNING]
channel stopped
Triggering a new build of E6J2
Finished: SUCCESS
```

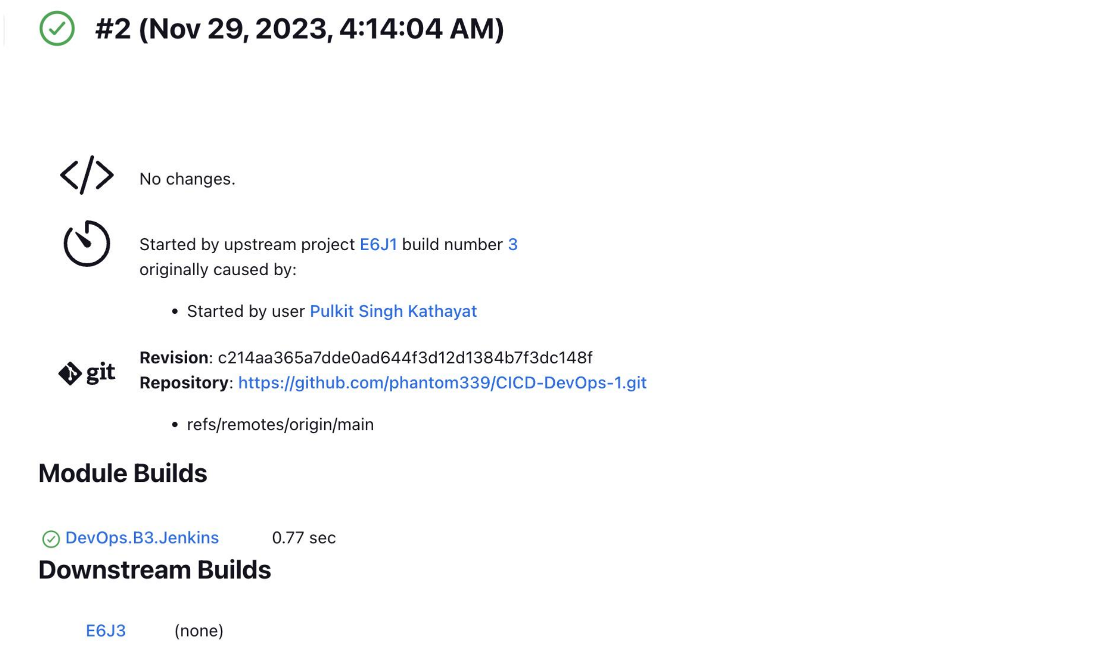
b. Similarly others were also triggered

i.

E6J2 logs



```
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /Users/pulkitkathayat/.jenkins/workspace/E6J2/pom.xml to DevOps.B3.Jenkins/DevOps.B3.Jenkins/0.0.1-
SNAPSHOT/DevOps.B3.Jenkins-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving /Users/pulkitkathayat/.jenkins/workspace/E6J2/target/My-B3-CALC.jar to
DevOps.B3.Jenkins/DevOps.B3.Jenkins/0.0.1-SNAPSHOT/DevOps.B3.Jenkins-0.0.1-SNAPSHOT.jar
channel stopped
Triggering a new build of E6J3
Finished: SUCCESS
```



#2 (Nov 29, 2023, 4:14:04 AM)

No changes.

Started by upstream project E6J1 build number 3
originally caused by:

- Started by user Pulkit Singh Kathayat

Revision: c214aa365a7dde0ad644f3d12d1384b7f3dc148f
Repository: https://github.com/phantom339/CICD-DevOps-1.git

- refs/remotes/origin/main

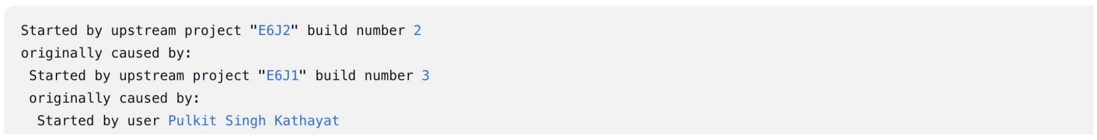
Module Builds

DevOps.B3.Jenkins 0.77 sec

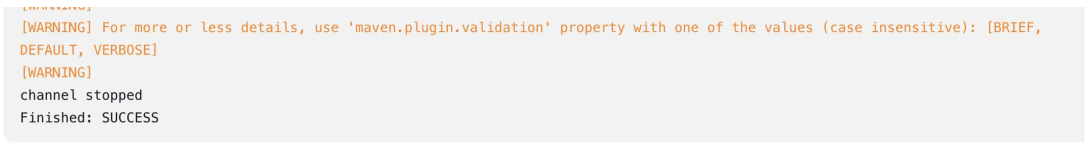
Downstream Builds

E6J3 (none)

ii. E6J3 logs



```
Started by upstream project "E6J2" build number 2
originally caused by:
 Started by upstream project "E6J1" build number 3
 originally caused by:
  Started by user Pulkit Singh Kathayat
```

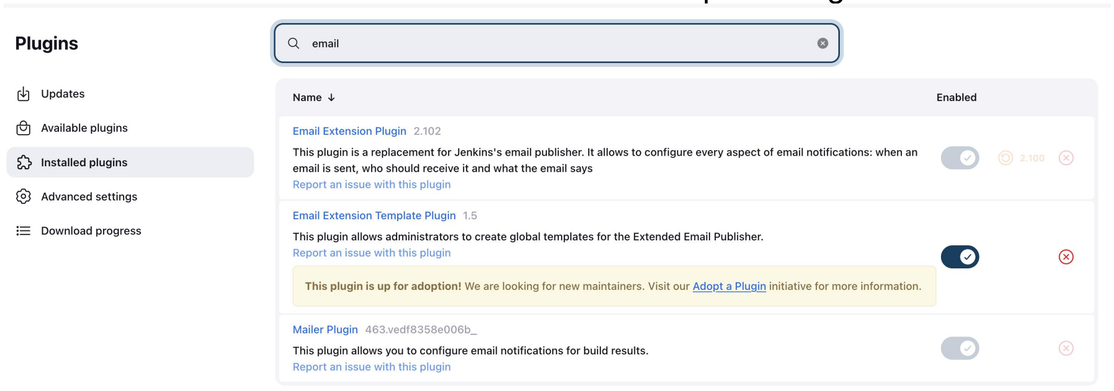


```
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the values (case insensitive): [BRIEF,
DEFAULT, VERBOSE]
[WARNING]
channel stopped
Finished: SUCCESS
```

Experiment 7
Email Notifications in Jenkins

Aim
To understand and practice setting up email.

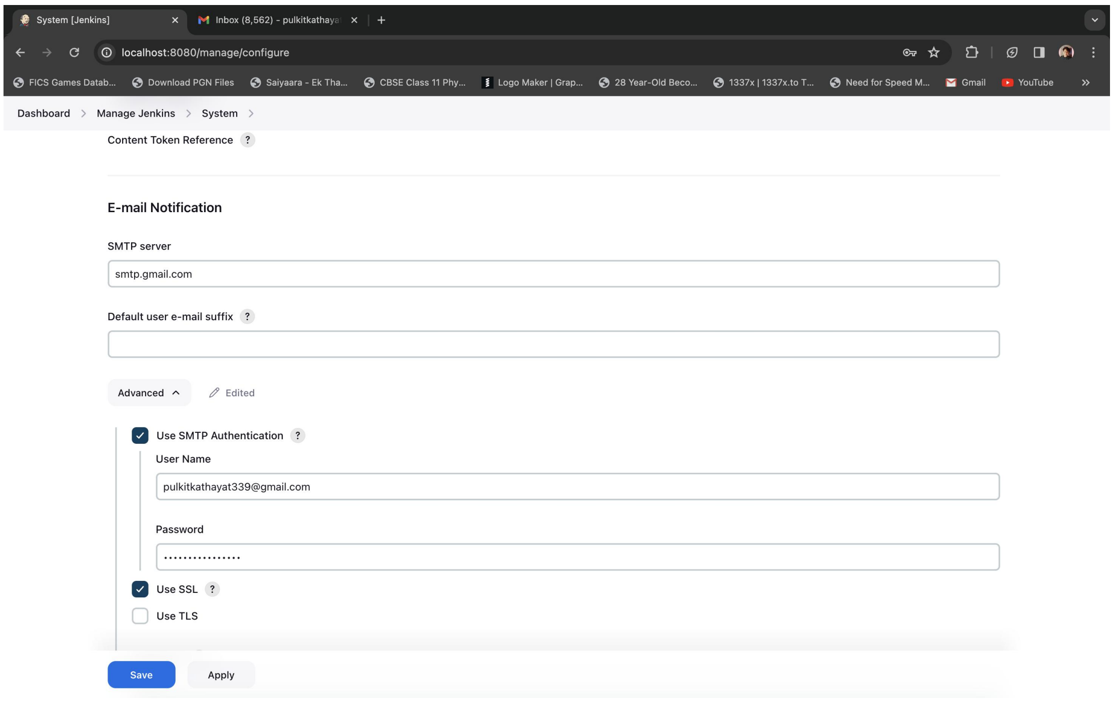
Steps
1. Install Email Extension & Email Extension Template Plugin



Plugins

Name	Enabled
Email Extension Plugin 2.102 This plugin is a replacement for Jenkins's email publisher. It allows to configure every aspect of email notifications: when an email is sent, who should receive it and what the email says Report an issue with this plugin	
Email Extension Template Plugin 1.5 This plugin allows administrators to create global templates for the Extended Email Publisher. Report an issue with this plugin **This plugin is up for adoption!** We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	
Mailer Plugin 463.vedf8358e006b_ This plugin allows you to configure email notifications for build results. Report an issue with this plugin	

2. Configure the SMTP Server by going to Dashboard > Manage Jenkins > System > Email
Notification
If using Gmail, type smtp.gmail.com for the SMTP server.



localhost:8080/manage/configure

Dashboard > Manage Jenkins > System

Content Token Reference

E-mail Notification

SMTP server: smtp.gmail.com

Default user e-mail suffix:

Advanced — Edited

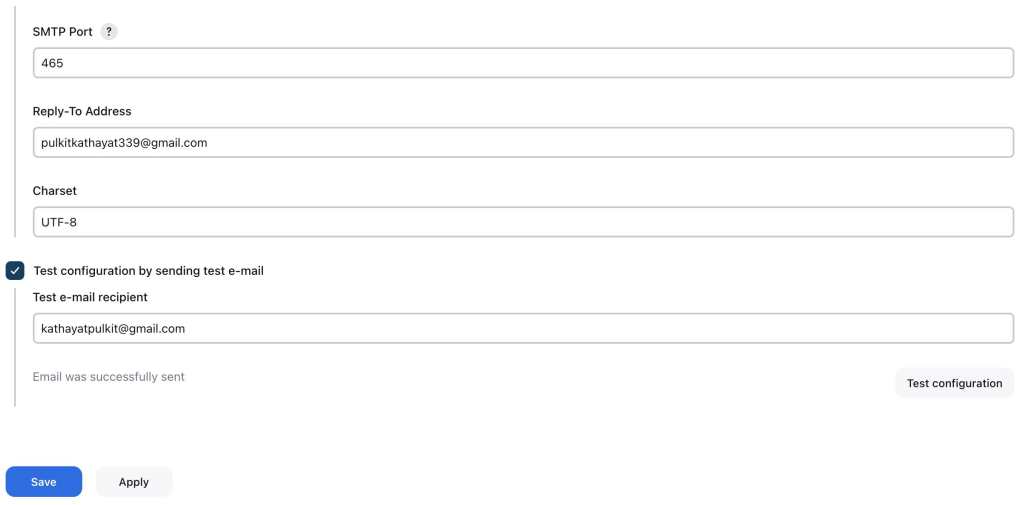
[✓] Use SMTP Authentication

User Name: pulkitkathayat339@gmail.com

Password: ················

[✓] Use SSL

[ ] Use TLS



SMTP Port ?

465

Reply-To Address

pulkitkathayat339@gmail.com

Charset

UTF-8

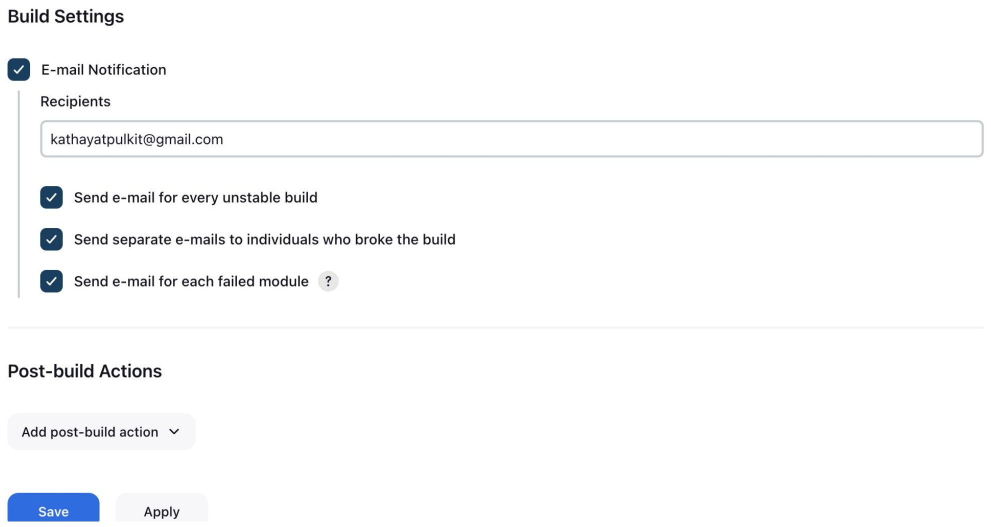
✓ Test configuration by sending test e-mail

Test e-mail recipient

kathayatpulkit@gmail.com

Email was successfully sent

3. Create a job and add email in Build Settings



Build Settings

✓ E-mail Notification

Recipients

kathayatpulkit@gmail.com

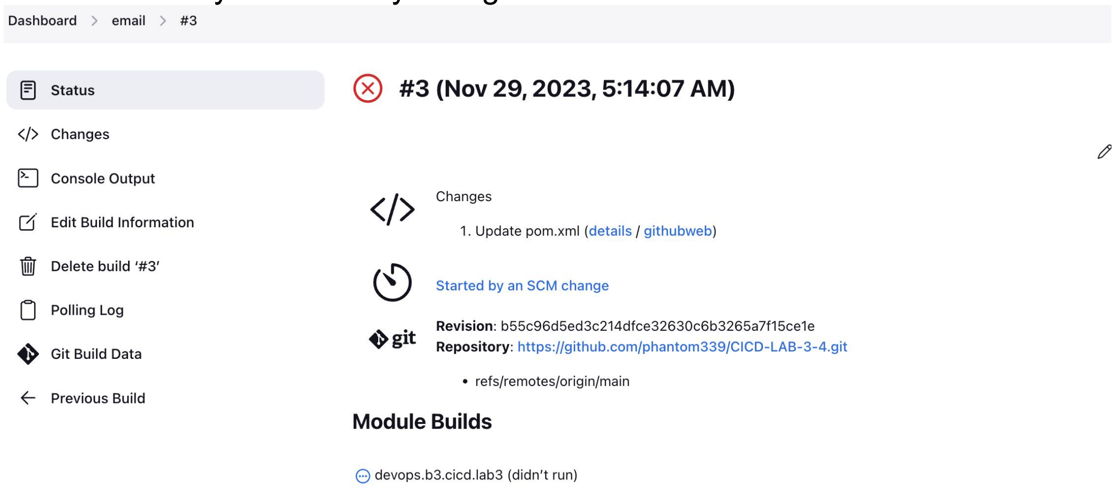
✓ Send e-mail for every unstable build

✓ Send separate e-mails to individuals who broke the build

✓ Send e-mail for each failed module ?

Post-build Actions

Test the mail by intentionally failing a build



#3 (Nov 29, 2023, 5:14:07 AM)

Changes

1. Update pom.xml (details / githubweb)

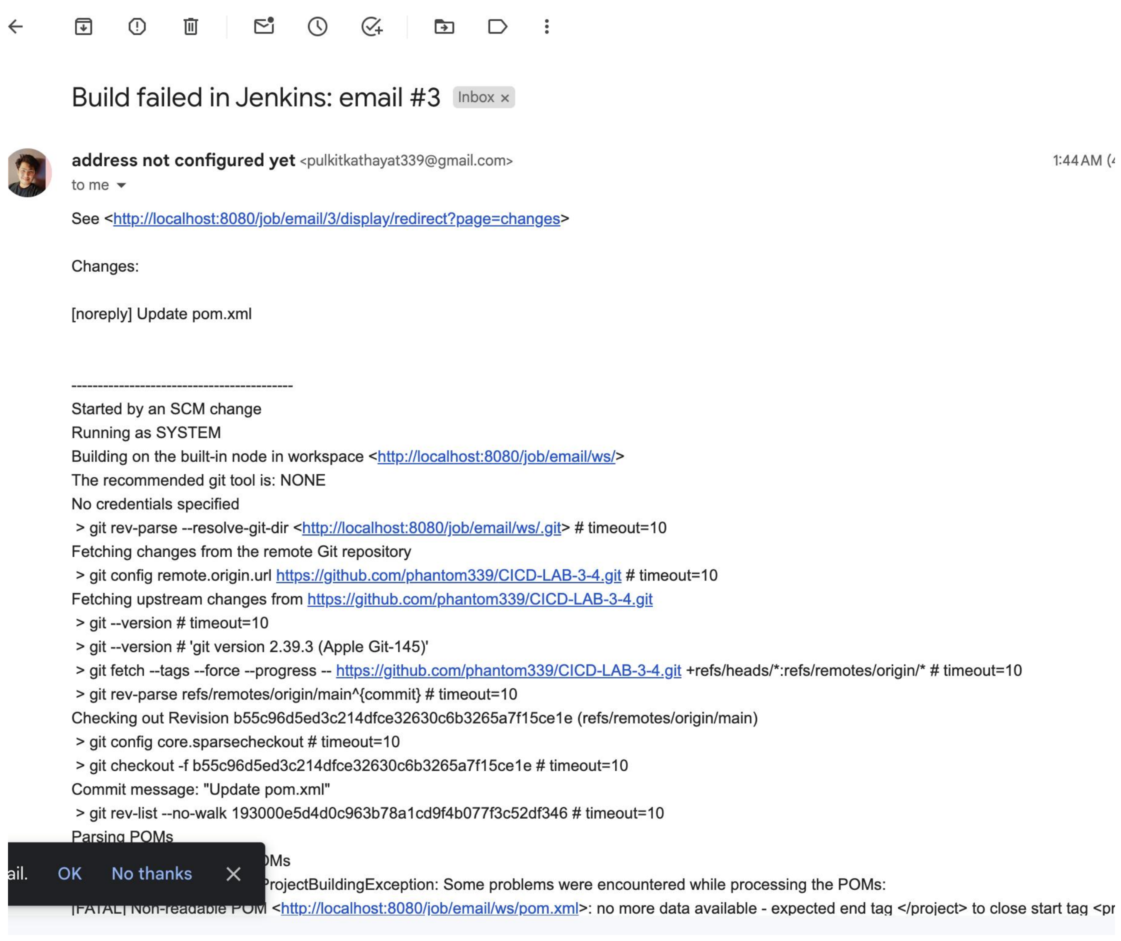
Started by an SCM change

Revision: b55c96d5ed3c214dfce32630c6b3265a7f15ce1e
Repository: https://github.com/phantom339/CICD-LAB-3-4.git

- refs/remotes/origin/main

Module Builds

devops.b3.cicd.lab3 (didn't run)



Build failed in Jenkins: email #3

Inbox ×

address not configured yet <pulkitkathayat339@gmail.com>
to me

1:44 AM (

See <http://localhost:8080/job/email/3/display/redirect?page=changes>

Changes:

[noreply] Update pom.xml

```
------------------------------------------
Started by an SCM change
Running as SYSTEM
Building on the built-in node in workspace <http://localhost:8080/job/email/ws/>
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir <http://localhost:8080/job/email/ws/.git> # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/phantom339/CICD-LAB-3-4.git # timeout=10
Fetching upstream changes from https://github.com/phantom339/CICD-LAB-3-4.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.3 (Apple Git-145)'
 > git fetch --tags --force --progress -- https://github.com/phantom339/CICD-LAB-3-4.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision b55c96d5ed3c214dfce32630c6b3265a7f15ce1e (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f b55c96d5ed3c214dfce32630c6b3265a7f15ce1e # timeout=10
Commit message: "Update pom.xml"
 > git rev-list --no-walk 193000e5d4d0c963b78a1cd9f4b077f3c52df346 # timeout=10
Parsing POMs
OMs
ProjectBuildingException: Some problems were encountered while processing the POMs:
[FATAL] Non-readable POM <http://localhost:8080/job/email/ws/pom.xml>: no more data available - expected end tag </project> to close start tag <pr
```