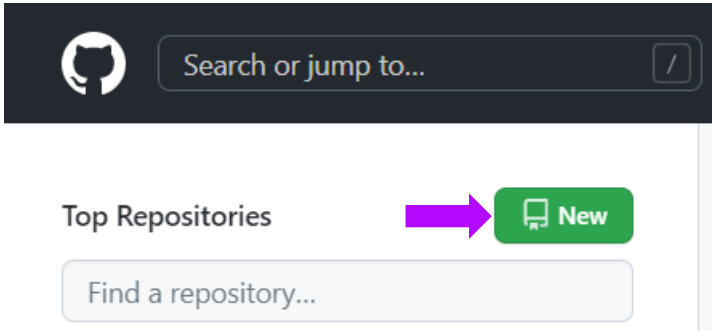


GitOps

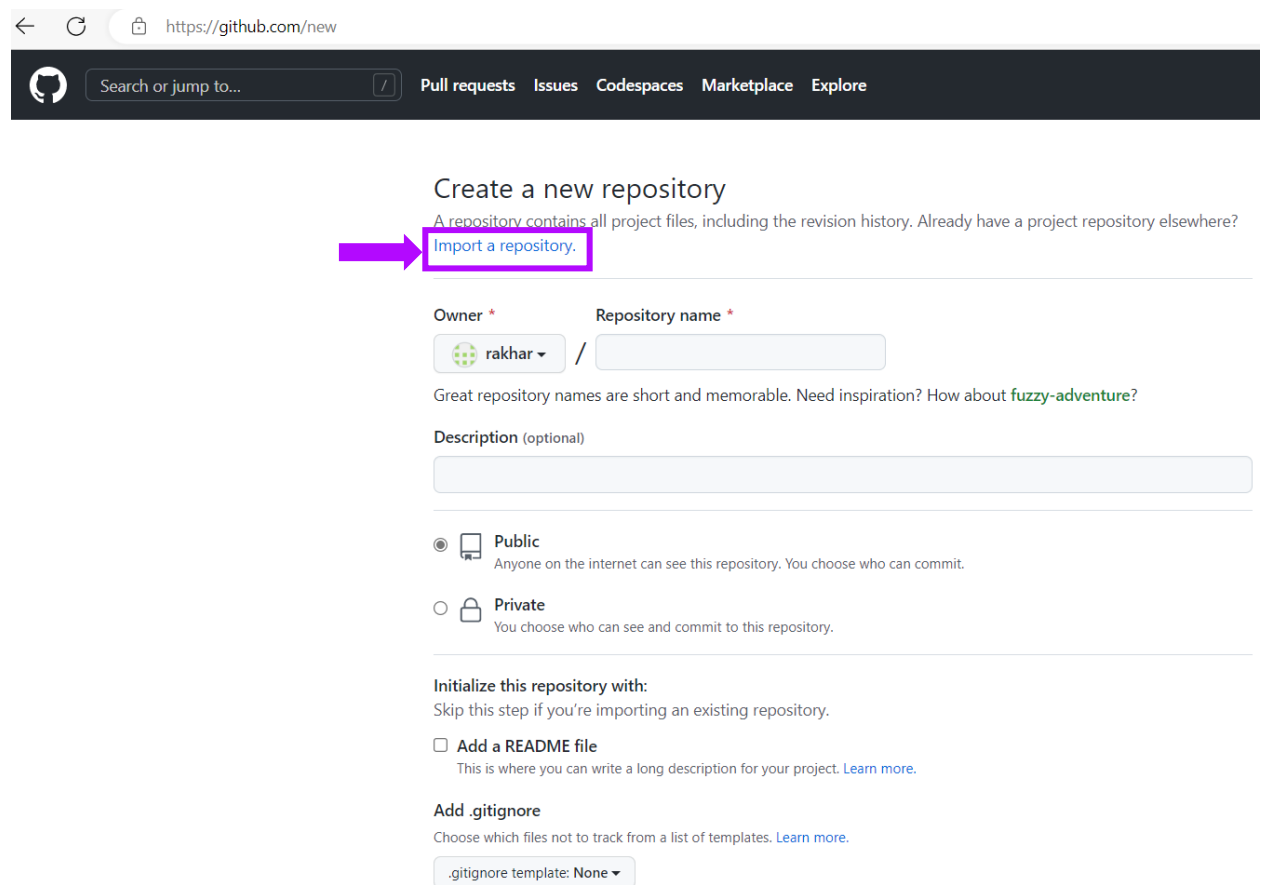
Activity 3: Github Actions workflow CI/CD.

Version	Revision Date	Description	Author	Reviewed by	Approved by
V1.0	21-Apr-23	Initial Version	Rakhi Parashar	Rathna Perumalsamy	CPCL, Sreenivasa Rao

This Activity demonstrates how to make use of Actions and using configuration files creating CICD workflow.

1	<ol style="list-style-type: none"> 1. Create account in github by signing up in github.com with your any valid mail id. 2. Complete the verification by checking your mail box. 3. Login to github.com
2	<ol style="list-style-type: none"> 1. Click on new button to create new project repository as shown below. 

2. Click on Import a repository to get the source code from below repository of the Trainer.



← ↻ 🔒 https://github.com/new


Search or jump to... Pull requests Issues Codespaces Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

Import a repository.

Owner * Repository name *

 rakhar /

Great repository names are short and memorable. Need inspiration? How about **fuzzy-adventure**?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None**

3. Provide below URL to be imported as highlighted in the screenshot.

- <https://github.com/rakhar/spring-hello-rakhi.git>
- Give the Repository name : spring-hello-yourname (e.g. spring-hello-rakhi , you can give yourname)
- Select Public option and click on Begin Import button.

← ↻ <https://github.com/new/import>

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

Import your project to GitHub

Import all the files, including revision history, from another version control system.

Your old repository's clone URL

<https://github.com/rakhar/spring-hello-rakhi.git>

Learn more about the types of [supported VCS](#).

Your new repository details

Owner * [rakhar](#) / Repository name * [spring-hello-yourname](#)

✓ spring-hello-yourname is available.

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

ⓘ You are creating a public repository in your personal account.

[Cancel](#) [Begin import](#)

- Once you see the message project Import is done, go to your project repository and click on Settings to enable Actions as shown below.
- Select Actions -> General -> select Allow all actions and reusable workflows as shown in below screenshot and click on save button.

[rakhar / GoAppsFlyer_Rakhi](#)

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

[rakhar / spring-hello-rakhi](#) [Public](#) [Pin](#) [Unwatch](#) 1 [Fork](#) 0

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

Actions permissions

☒ **Allow all actions and reusable workflows**
Any action or reusable workflow can be used, regardless of who authored it or where it is defined.

☐ **Disable actions**
The Actions tab is hidden and no workflows can run.

☐ **Allow rakhar actions and reusable workflows**
Any action or reusable workflow defined in a repository within rakhar can be used.

☐ **Allow rakhar, and select non-rakhar, actions and reusable workflows**
Any action or reusable workflow that matches the specified criteria, plus those defined in a repository within rakhar, can be used. [Learn more about allowing specific actions and reusable workflows to run.](#)

[Save](#)

Now click on Actions as shown in below screenshot and observe below features.

GitHub repository: **rakhar / spring-hello-rakhi** (Public)

Navigation: <> Code Issues Pull requests **Actions** Projects Wiki Security Insights

Actions

New workflow

All workflows

Java CI with Maven

Management

Caches

All workflows

Showing runs from all workflows

0 workflow runs

Suggested for this repository

- Publish Java Package with Maven** By GitHub Actions
Build a Java Package using Maven and publish to GitHub Packages.
[Configure](#) Java
- Java with Maven** By GitHub Actions
Build and test a Java project with Apache Maven.
[Configure](#) Java
- Publish Java Package with Gradle** By GitHub Actions
Build a Java Package using Gradle and publish to GitHub Packages.
[Configure](#) Java
- Java with Gradle** By GitHub Actions
Build and test a Java project using a Gradle wrapper script.
[Configure](#) Java
- Android CI** By GitHub Actions
Build an Android project with Gradle.
[Configure](#) Java
- Java with Ant** By GitHub Actions
Build and test a Java project with Apache Ant.
[Configure](#) Java

Observe the workflow configurations mentioned in the source code `.github/workflows` YAML file i.e. `maven-publish.yml`.

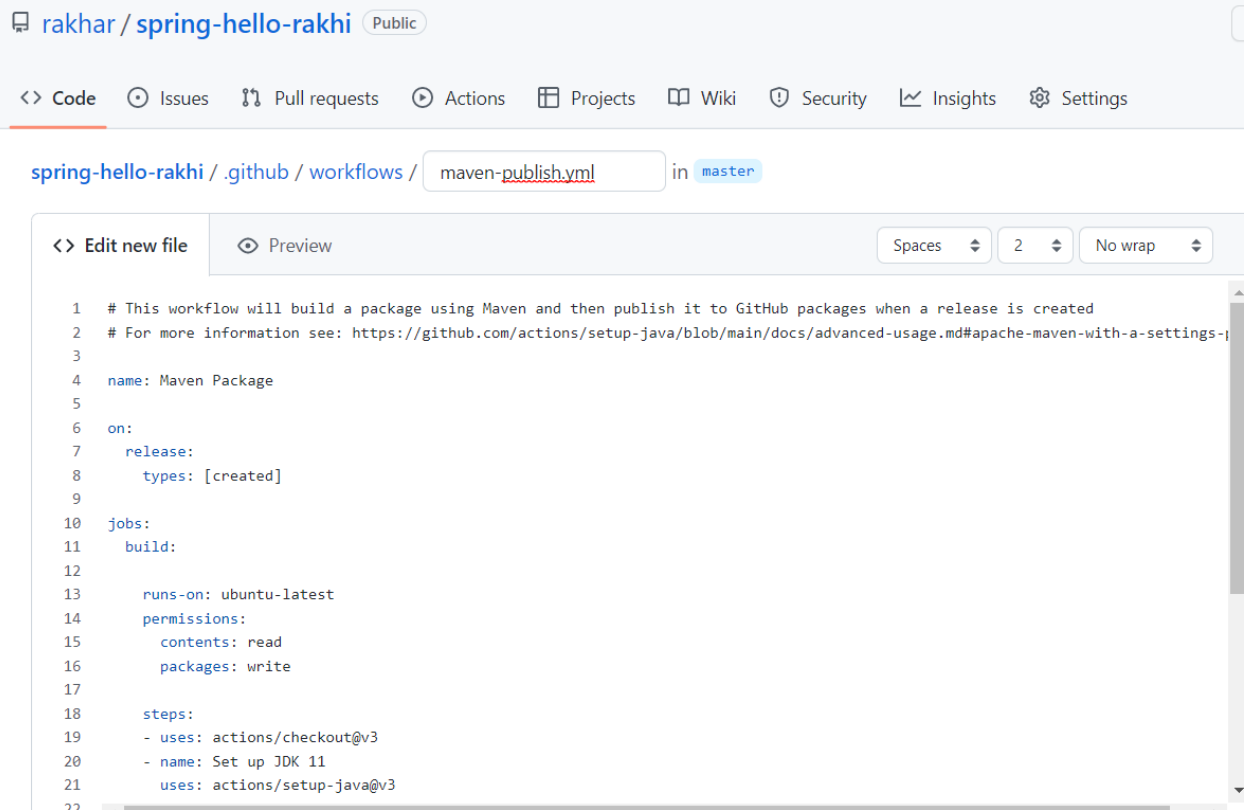
Repository: **rakhar / spring-hello-rakhi** (Public)

Navigation: <> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

master 1 branch 0 tags

[Go to file](#) [Add file](#) [Code](#)

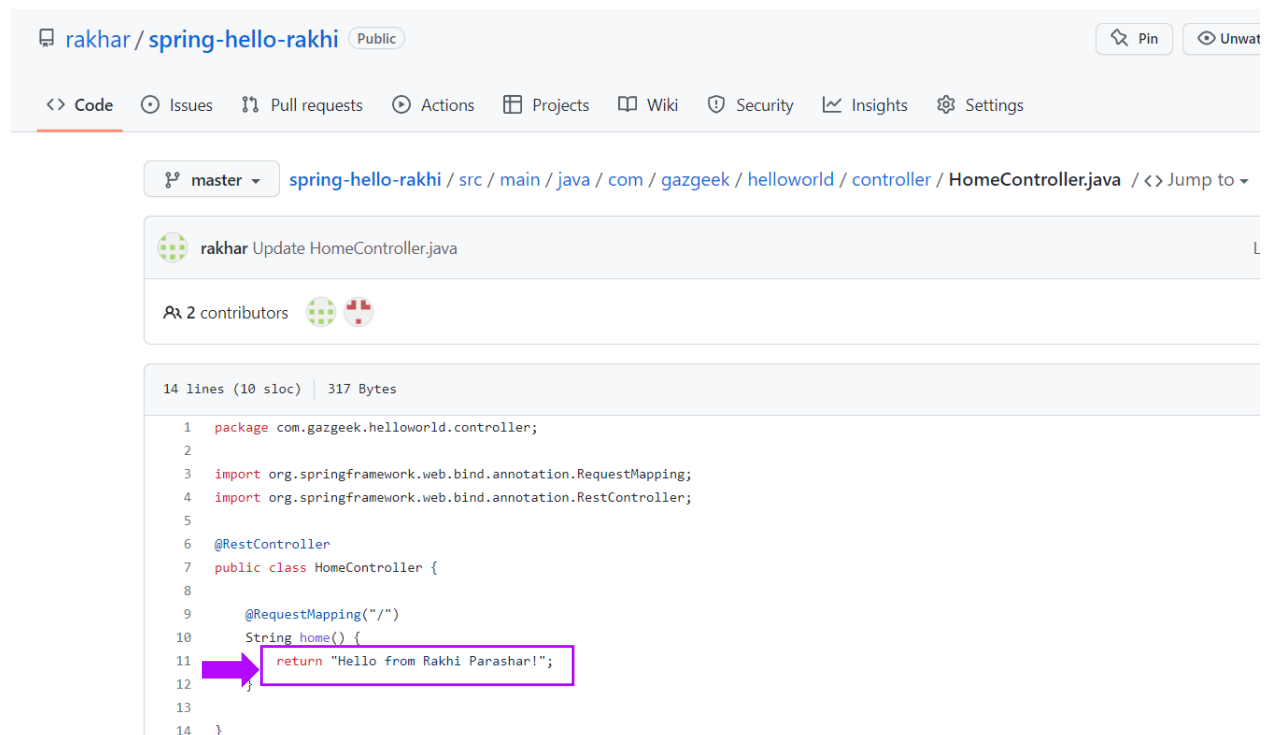
	rakhar Update README.md	✓ 36473a2 3 days ago 4 commits
	.github/workflows	Create maven-publish.yml 2 years ago
	src/main	Update HomeController.java 3 days ago
	README.md	Update README.md 3 days ago
	pom.xml	first commit 2 years ago



```
1 # This workflow will build a package using Maven and then publish it to GitHub packages when a release is created
2 # For more information see: https://github.com/actions/setup-java/blob/main/docs/advanced-usage.md#apache-maven-with-a-settings-f
3
4 name: Maven Package
5
6 on:
7   release:
8     types: [created]
9
10 jobs:
11   build:
12
13     runs-on: ubuntu-latest
14     permissions:
15       contents: read
16       packages: write
17
18     steps:
19       - uses: actions/checkout@v3
20       - name: Set up JDK 11
21         uses: actions/setup-java@v3
```

In the process of continuously integrating the code and continuously build and getting our artifacts ready for deployment, we will make change in the `HelloController.java` file as shown below.

Note: we can make change in `README.md` file also as alternative.



```
1 package com.gazgeek.helloworld.controller;
2
3 import org.springframework.web.bind.annotation.RequestMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HomeController {
8
9     @RequestMapping("/")
10     String home() {
11         return "Hello from Rakhi Parashar!";
12     }
13
14 }
```

Scroll down and click on commit.



Commit changes

Update HomeController.java

Add an optional extended description...

rakhi.parashar@accenture.com

Choose which email address to associate with this commit

- ☒ Commit directly to the `master` branch.
- ☐ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)



Commit changes

Cancel

To show you Zero Touch, as soon as you commit your changes workflow gets triggered for build and binary artifacts gets created which can be deployed anywhere.

To observe the build jobs and see the logs, click on Actions and then click on build as below:

The screenshot shows the GitHub Actions interface for the repository 'rakhar / spring-hello-rakhi'. The 'Actions' tab is selected and highlighted with a purple box. Below it, the workflow 'Update README.md #2' is shown. The 'Summary' tab is active, displaying a table of workflow runs. The first run, triggered by a push to the 'master' branch, is in 'Success' status and took 50s. Below the table, the 'maven-publish.yml' workflow file is shown, with the 'build' job highlighted by a purple box and a purple arrow. The 'build' job is shown as completed with a green checkmark and a duration of 37s.

Triggered via push 1 minute ago	Status	Total duration	Artifacts
rakhar pushed -> 36473a2 master	Success	50s	1

maven-publish.yml
on: push

build 37s

The screenshot shows the GitHub Actions interface for a workflow named 'Java CI with Maven' in the repository 'rakhar / spring-hello-rakhi'. The workflow run is titled 'Update README.md #2' and is in a successful state. The 'build' job is selected, showing a list of steps: 'Set up job', 'Run actions/checkout@v2', 'Set up JDK 11', 'Build with Maven', 'Save artifact', 'Post Set up JDK 11', 'Post Run actions/checkout@v2', and 'Complete job'. A purple bracket highlights the 'Build with Maven' and 'Save artifact' steps. Below the main view, the 'Run details' for the 'Save artifact' step are expanded, showing the logs of the 'Run actions/upload-artifact@v2' action. A purple arrow points to the log entry: 'Uploaded /home/runner/work/spring-hello-rakhi/spring-hello-rakhi/artifacts/helloworld-0.0.1-SNAPSHOT.jar (73.7%) bytes 0:8388607'. The logs also show the total size of the files uploaded and the final status of the artifact upload.

rakhar / spring-hello-rakhi Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

← Java CI with Maven

Update README.md #2

Summary

Jobs

- build

Run details

- Usage
- Workflow file

build

succeeded 4 minutes ago in 37s

- Set up job
- Run actions/checkout@v2
- Set up JDK 11
- Build with Maven
- Save artifact
- Post Set up JDK 11
- Post Run actions/checkout@v2
- Complete job

Expand the each step and observe the stages are successful or not. Like here we showed Build with Maven and artifacts step.

Summary

Jobs

- build

Run details

- Usage
- Workflow file

build

succeeded 5 minutes ago in 37s

- Set up job
- Run actions/checkout@v2
- Set up JDK 11
- Build with Maven
- Save artifact

Run actions/upload-artifact@v2

With the provided path, there will be 1 file uploaded

Starting artifact upload

For more detailed logs during the artifact upload process, enable step-debugging: <https://docs.github.com/actions/monitoring-and-troubleshooting-workflows/enabling-debug-logging#enabling-step-debug-logging>

Artifact name is valid!

Container for artifact "github-actions-artifact" successfully created. Starting upload of file(s)

Uploaded /home/runner/work/spring-hello-rakhi/spring-hello-rakhi/artifacts/helloworld-0.0.1-SNAPSHOT.jar (73.7%) bytes 0:8388607

Uploaded /home/runner/work/spring-hello-rakhi/spring-hello-rakhi/artifacts/helloworld-0.0.1-SNAPSHOT.jar (100.0%) bytes 8388608:11376745

Total size of all the files uploaded is 11376746 bytes

File upload process has finished. Finalizing the artifact upload

Artifact has been finalized. All files have been successfully uploaded!

The raw size of all the files that were specified for upload is 12726735 bytes

The size of all the files that were uploaded is 11376746 bytes. This takes into account any gzip compression used to reduce the upload size, time and storage

Note: The size of downloaded zips can differ significantly from the reported size. For more information see: <https://github.com/actions/upload-artifact#zipped-artifact-downloads>

Artifact github-actions-artifact has been successfully uploaded!

Summary

Jobs

build

Run details

Usage

Workflow file

build

succeeded 3 days ago in 37s

Build with Maven

1292 [INFO] Downloading from central: <https://repo.maven.apache.org/maven2/asm/asm-analysis/3.2/asm-analysis-3.2.jar>

1293 [INFO] Downloaded from central: <https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-dependency-tree/2.1/maven-dependency-tree-2.1.jar> (68 kB at 384 kB/s)

1294 [INFO] Downloading from central: <https://repo.maven.apache.org/maven2/asm/asm-util/3.2/asm-util-3.2.jar>

1295 [INFO] Downloaded from central: <https://repo.maven.apache.org/maven2/org/yafer/jdependency/0.7/jdependency-0.7.jar> (12 kB at 59 kB/s)

1296 [INFO] Downloading from central: <https://repo.maven.apache.org/maven2/com/google/guava/guava/11.0.2/guava-11.0.2.jar>

1297 [INFO] Downloaded from central: <https://repo.maven.apache.org/maven2/org/jdom/jdom/1.1/jdom-1.1.jar> (153 kB at 758 kB/s)

1298 [INFO] Downloading from central: <https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/1.3.9/jsr305-1.3.9.jar>

1299 [INFO] Downloaded from central: <https://repo.maven.apache.org/maven2/asm/asm-analysis/3.2/asm-analysis-3.2.jar> (18 kB at 87 kB/s)

1300 [INFO] Downloading from central: <https://repo.maven.apache.org/maven2/commons-io/commons-io/1.3.2/commons-io-1.3.2.jar> (88 kB at 416 kB/s)

1301 [INFO] Downloaded from central: <https://repo.maven.apache.org/maven2/com/google/code/findbugs/jsr305/1.3.9/jsr305-1.3.9.jar> (33 kB at 149 kB/s)

1302 [INFO] Downloading from central: <https://repo.maven.apache.org/maven2/asm/asm-util/3.2/asm-util-3.2.jar> (37 kB at 165 kB/s)

1303 [INFO] Downloaded from central: <https://repo.maven.apache.org/maven2/com/google/guava/guava/11.0.2/guava-11.0.2.jar> (1.6 MB at 6.7 MB/s)

1304 [INFO] -----

1305 [INFO] BUILD SUCCESS

1306 [INFO] -----

1307 [INFO] Total time: 15.897 s

1308 [INFO] Finished at: 2023-03-20T08:34:31Z

1309 [INFO] -----

1310 helloworld-0.0.1-SNAPSHOT.jar

Save artifact

Post Set up JDK 11


Post Run actions/checkout@v2

Complete job

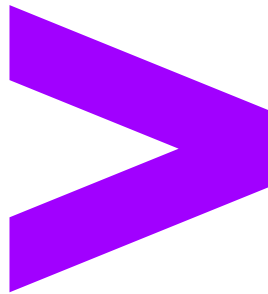
Click on Summary and scroll down to see your binary artifacts which can be deployed anywhere.

Artifacts

Produced during runtime

Name	Size
 github-actions-artifact	12.1 MB

End of the Activity 1.



Copyright © 2025 Accenture
All rights reserved.
Accenture and its logo are trademarks of Accenture.