Github Actions

GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.

https://help.github.com/en/actions/automating-your-workflow-with-github-actions/about-github-actions

About GitHub Actions

GitHub Actions help you automate your software development workflows in the same place you store code and collaborate on pull requests and issues. You can write individual tasks, called actions, and combine them to create a custom workflow. Workflows are custom automated processes that you can set up in your repository to build, test, package, release, or deploy any code project on GitHub.

With GitHub Actions you can build end-to-end continuous integration (CI) and continuous deployment (CD) capabilities directly in your repository. GitHub Actions powers GitHub's built-in continuous integration service

Workflows run in Linux, macOS, Windows, and containers on GitHub-hosted servers. You can create workflows using actions defined in your repository, open source actions in a public repository on GitHub, or a published Docker container image. Workflows in forked repositories don't run by default.

Usage limits

Exceeding usage limits may result in jobs queueing, failing to run, or failing to complete. Limits are subject to change.

You can execute up to 20 workflows concurrently per repository.

You can execute up to 1000 API requests in an hour across all actions within a repository.

Each job in a workflow can run for up to 6 hours of execution time.

The number of jobs you can run concurrently across all repositories in your account depends on your GitHub plan.

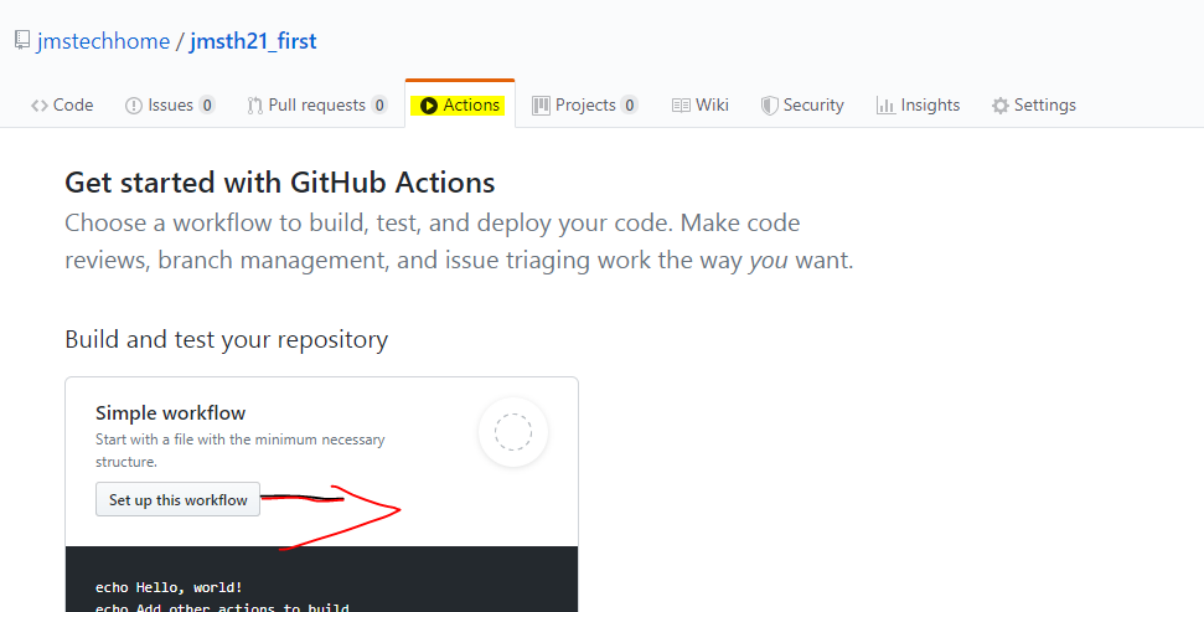
GitHub plan	Total concurrent jobs	Maximum concurrent macOS jobs
Free	20	5
Pro	40	5
Team	60	5
Enterprise	180	15

About billing for GitHub Actions

GitHub Actions usage is free for public repositories. For private repositories, each GitHub account receives a certain amount of free minutes and storage, depending on the product used with the account. For more information, see "About billing for GitHub Actions."

https://help.github.com/en/actions/automating-your-workflow-with-github-actions/workflow-syntax-for-github-actions

create first work flow



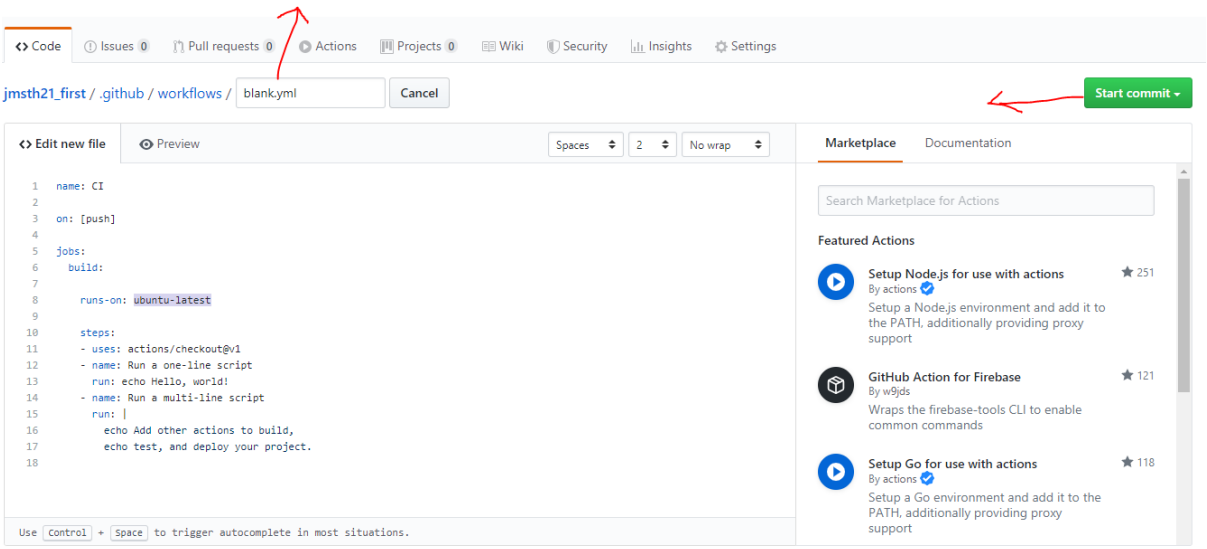
```
name: CI

on: [push]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v1
    - name: Run a one-line script
      run: echo Hello, world!
    - name: Run a multi-line script
      run: |
        echo Add other actions to build,
        echo test, and deploy your project.
```



What are the tools are allowed github actions

```
name:
CI
        on: [push]
        jobs:
          build:
            runs-on: ubuntu-latest
            steps:
            - uses: actions/checkout@v1
            - name: Run a one-line script
              run: echo Hello, world!
            - name: java version
              run: java -version
            - name: java version
              run: mvn --version
            - name: python version
              run: python --version
            - name: docker version
              run: docker -v
```

End to End deployment using github actions.

```
name: web deployment

on: [push]

jobs:

  build:

    runs-on: ubuntu-latest

    steps:

    - uses: actions/checkout@v1

    - name: Set up JDK 1.8

      uses: actions/setup-java@v1

      with:

        java-version: 1.8

    - name: Build with Maven

      run: mvn -B package --file pom.xml
```

- name: tomcat deploy

run: curl -v -u admin:admin -T /home/runner/work/spring3-mvc-maven-xml-hello-world/spring3-mvc-maven-xml-hello-world/target/spring3-mvc-maven-xml-hello-world-1.0-SNAPSHOT.war 'http://ec2-13-126-114-107.ap-south-1.compute.amazonaws.com:8080/manager/text/deploy?path=/github_action_spring'

Build Docker image and push to ECR repo using github actions

```
name:
Publish
Dockerhub
and ecr
node app
            on: [push]
            jobs:
              push_docker_hub:
                runs-on: ubuntu-latest
                steps:
                  - uses: actions/checkout@master
                  - run: |
                      $(aws ecr get-login --no-include-email --region us-east-1)
                      docker build -t ${{ secrets.ECR_REPO_NAME }}/node_action:v1 .
                      docker push ${{ secrets.ECR_REPO_NAME }}/node_action:v1
                    env:
                      AWS_ACCESS_KEY_ID: ${{ secrets.AWS_ACCESS_KEY_ID }}
                      AWS_SECRET_ACCESS_KEY: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
                      AWS_REGION: ${{ secrets.AWS_REGION }}
```

Push docker images to github docker package registry.

name: Docker Image CI

on:

push:

branches: [master]

pull_request:

branches: [master]

jobs:

```
 push_to_registries:
   name: Push Docker image to multiple registries
   runs-on: ubuntu-latest
   steps:
    - name: Check out the repo
      uses: actions/checkout@v2
    - name: Build and Publish head Docker image
      uses: VaultVulp/gp-docker-action@1.1.7
      with:
        github-token: ${{ secrets.TOKEN }} # Provide GITHUB_TOKEN to login into the GitHub Packages
        image-name: myjavaapp # Provide Docker image name
        image-tag: head # Provide Docker image tag
```

publish maven package into github repo using github actions

added in pom.xml below dependencies

```
<plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-deploy-plugin</artifactId>
        <version>2.8.2</version>
</plugin>
--------
--------
<distributionManagement>
 <repository>
  <id>github</id>
  <name>GitHub OWNER Apache Maven Packages</name>
```

<url>https://maven.pkg.github.com/jmstechhome8/my-maven-repo</url>

</repository>

</distributionManagement>

refer below url

https://github.com/jmstechhome8/spring3-mvc-maven-xml-hello-world/blob/master/pom.xml

https://github.com/jmstechhome8/spring3-mvc-maven-xml-hello-world/

github actions workflow code

```
name: Java CI with Maven

on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]
  release:
    types: [created]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - name: Set up JDK 1.8
```

```
    uses: actions/setup-java@v1
    with:
      java-version: 1.8
  - name: Build with Maven
    run: mvn -B package --file pom.xml
  - name: Deploy to Github Package Registry
    env:
      TOKEN: ${{ secrets.TOKEN }}
    run: |
      mkdir -p ~/.m2
      echo "<settings><servers><server><id>gh</id><username>$(echo "$GITHUB_REPOSITORY" | awk -F / '{print $1}')</username><password>\${env.TOKEN}</password></server></servers></settings>" > ~/.m2/settings.xml
      REPO="gh::default::https://maven.pkg.github.com/${GITHUB_REPOSITORY}"
      mvn deploy -DaltReleaseDeploymentRepository="${REPO}" -DaltSnapshotDeploymentRepository="${REPO}"
```

https://stackoverflow.com/questions/57711558/deploy-to-github-package-registry-from-github-action