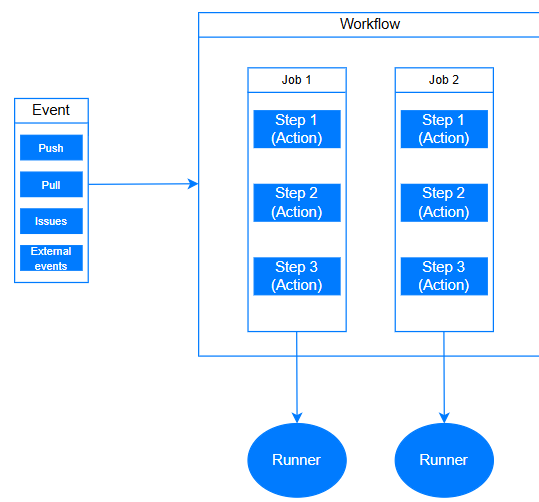
GitHub actions allow us to automate, customize and execute our software development workflows right in our repository. These workflows are composed of different tasks or actions that can be run automatically on certain events. This enables us to include Continuous Integration (CI) and continuous deployment (CD) capabilities and many other features directly in our repository.

**Components of GitHub Actions**



GitHub actions is an event-driven paradigm, meaning we can execute a series of commands after a certain event has occurred. We can create multiple workflows and multiple jobs in GitHub actions. As we see in the figure above, each and every job we create is running on a runner which separates virtual machines inside the GitHub server.

GitHub's action is composed of five main parts:

* **Events**- initiate or trigger an execution of a GitHub action workflow. An event may be an internal GitHub event (such as a push, a release, fork, issues, or a pull request), a scheduled event (triggered at a specific time, like a cron job), or an arbitrary external event (triggered by a webhook call to the GitHub API).
* **Workflows**- are composed of jobs that can be started by an event or by scheduling them to run at a specific time. Workflows assist in building, testing, releasing, or deploying our project to GitHub.
* **Job** - is a set of steps that run inside the same virtual machine runner, or inside a container. By default, they are executed in parallel, in other words, they don't wait for the previous job to be finished. Moreover, we can configure the jobs to run one after another if needed.
* **Actions**- in the workflow, we are going to specify our actions, which are the individual tasks that we use to achieve our goal. We can create our own actions, or use actions created by the GitHub community.
* **Runners** - is a virtual machine that runs a job. The runner is responsible for looking for the available jobs, then executes each job at a time. After the job is finished running, the runner reports the progress and logs of the job to GitHub. The runners hosted by GitHub consist of Ubuntu, Linux, Windows, and macOS.

**Workflow syntax**

To get started using GitHub Actions, we need to add a folder to the root of the GitHub repository.

.github/workflows

The workflow directory will contain the YAML files.

We can also go to the remote repository and go to the actions tab to create the workflow files directly.

YAML uses a fixed indentation scheme to represent relationships between data layers.

# your-repo-name/.github/workflows/first\_workflow.yml

name: First Workflow

on: push

jobs:

first-job:

name: Name of first step

runs-on: ubuntu-latest

steps:

#step 1

- name: Print a greeting

run: echo Hi from our first workflow!

#step 2

- uses: actions/checkout@v2.3.4

second-job:

strategy:

matrix:

runtimes: [10, 12, 14]

os\_version: [ubuntu-latest, windows-latest]

runs-on: ${{ matrix.os\_version}}

steps:

- uses: actions/setup-node@v3

with:

node-version: ${{ matrix.version }}

The name of our workflow is First Workflow and our workflow is triggered by the push event. We have jobs that make the workflow.

In the first job, the following terms are laid out as follows:

* runs-on: the machine each job should run
* steps:the tasks each job should run
* run: the command the step should run
* uses: signature of the action we want to use from the GitHub marketplace

GitHub Actions allows us to automate the tasks to run on multiple versions of OS and runtimes. In the second job, a job will run for each possible combination of the variables. In this example, the workflow will run six jobs, one for each combination of the OS and version variables.

You don't need to understand all the details from the snippet as of now. We will cover them in future topics.

**Conclusion**

GitHub Actions consists of workflows that are stored in a git repository at the location .github/workflows. The workflows are generally triggered by certain events we specified in the workflow. After the workflow is triggered, the runner executes the jobs one by one. Each job consists of steps that are made up of actions that run as a part of the workflow.

## Matrix strategy

name: Matrix

on: push

jobs:

first-job:

strategy:

matrix:

runtimes: [10, 12, 14, 15]

os\_version: [ubuntu-latest, windows-latest, macOs-latest]

How many jobs will the workflow run?

The matrix has:

* 4 runtimes = [10, 12, 14, 15]
* 3 os\_versions = [ubuntu-latest, windows-latest, macOs-latest]

Total combinations = 4 runtimes × 3 OS versions = **12 jobs**

✅ **Correct answer: 12**

## Workflow language

Which syntax is used in a workflow file?

Select one option from the list

The correct answer is **YAML**.

GitHub workflows use YAML because it's simple, easy to read, and great for writing configuration files. It's designed to be human-friendly while still structured enough for machines to parse.

Shell script is used **inside** the workflow—for the commands that run—but the file itself is written in YAML. JSON and XML are too strict and harder to write by hand for this kind of setup.

## Directory

What is the address of the directory where workflow files are stored?

.github/workflows

## Identify from YAML file

name: node

on:

pull\_request:

branches: ["staging"]

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: checkout

uses: actions/checkout@v2

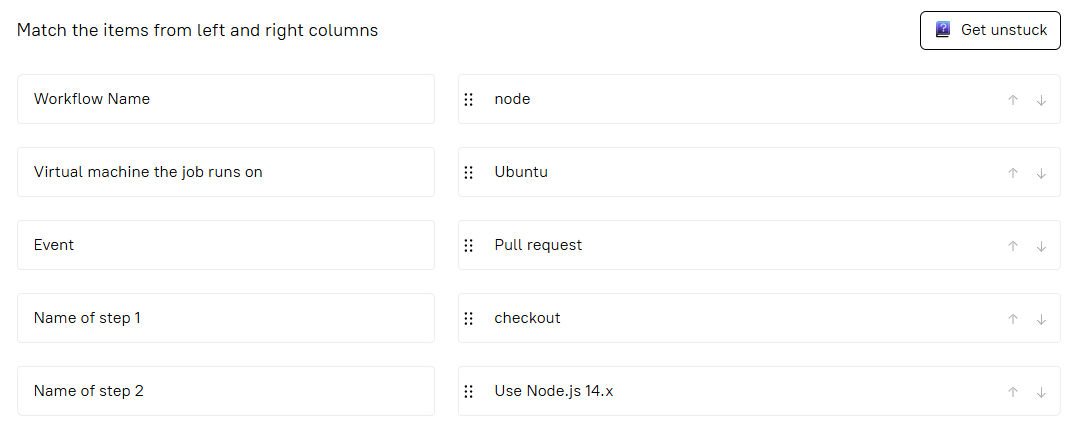
- name: Use Node.js 14.x

uses: actions/setup-node@v1

with:

node-version: 14.x

Match the corresponding key value with the help of the YAML file.

****

**Role of a runner**

Which of the following best describes how a runner works?

**Correct answer: A runner can run a single job at a time**

### Explanation:

A **runner** is a machine (virtual or physical) that GitHub uses to **execute jobs** in a workflow.

* It **runs one job at a time** per runner instance.
* If you want to run multiple jobs in parallel, you need **multiple runners**.
* Each job runs in a clean environment (like a fresh container or VM), depending on the runner type.

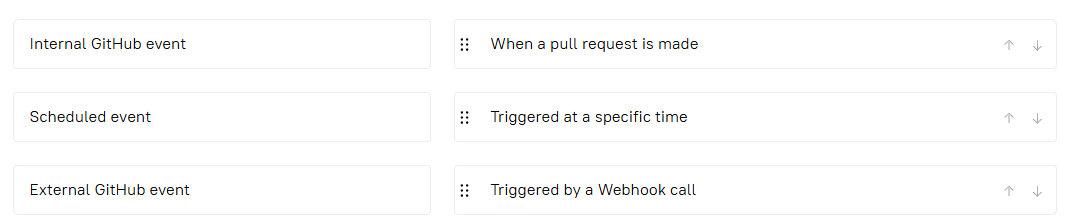
So:

* ❌ It doesn't run multiple jobs at once.
* ❌ It doesn’t manage virtual machines itself.
* ✅ It focuses on executing **one job at a time** per runner.

Types of Events

Match the corresponding events with the correct description

Match the items from left and right columns



Which of the following leads to the trigger of an event?

 Select one or more options from the list

From the text:

"An event may be an internal GitHub event (such as a push, a release, fork, issues, or a pull request)..."

So based on this, the correct options are:

✅ **Issue**  
✅ **Release**  
✅ **Fork**  
✅ **Pull request**

❌ **Branch checkout** – This is not an event. It’s a step that happens **inside** a job, not something that **triggers** a workflow.

### ✅ Final answer:

* ✔️ **Issue**
* ✔️ **Release**
* ❌ **Branch checkout**
* ✔️ **Fork**
* ✔️ **Pull request**