

---

## Connecting Actions with Workflows

---

### Create a workflow

- >Workflows define the event that triggers actions
- >Workflows define which actions to run
- >Repositories can contain multiple workflows

#### Syntax:

```
name: [workflow name]
on: [triggers]
```

### Add jobs and steps to a workflow

- >Workflows must have at least one job
- >Each job must have a unique identifier
- >Job identifier must start with a letter or underscore

#### Syntax:

jobs:

job1:

name: [name of the job]

runs-on:

#### Some runners:

Operating System	Workflow Keyword
Windows Server 2019 Windows Server 2016	windows-latest or windows-2019 windows-2016
Ubuntu 20.04 Ubuntu 18.04	ubuntu-latest or ubuntu-20.04 ubuntu-18.04
macOS Catalina 10.15 macOS Big Sur 11	macos-latest or macos-10.15 macos-11

1:12 / 2:36 1x

## Steps

->Steps are tasks within a job

->Steps run as processes in the compute resource

->Steps can run commands or actions

## Add actions to a workflow

**uses:** Execute an action in the operating system.

## Where to find the action

Action Location	Syntax
Public repository	<code>uses: {owner}/{repo}@{ref}</code> <code>uses: octocat/super-cool-action@v1</code>
The same repository as the workflow	<code>uses: ./path/to/the/action</code> <code>uses: ../.github/actions/my-local-action</code>
A Docker image registry	<code>uses: docker://{image}:{tag}</code> <code>uses: docker://hello-world:latest</code>

## Adding a command

**run:** Execute commands in the operating system's shell

Bash: Default shell for Ubuntu, macOS

PowerShell: Default shell for Windows

Run	Syntax
Single-line command	<code>run: {command} {parameters} {arguments}</code> <code>run: mv ./output ./archive</code>
Multiline command	<code>run:  </code> Command 1 Command 2  <code>run:  </code> g++ -c -Wall -g Main.cpp g++ -g -o Main.exe Main.o

e.g.

```
name: first
on: push
```

```
jobs:
  job1:
    name: First job
    runs-on: ubuntu-latest
    steps:
      - name: step one
        uses: actions/checkout@v2
      - name: step two
        run: env | sort
  job2:
    name: Second job
    runs-on: windows-latest
    steps:
      - name: Step one
        uses: actions/checkout@v2
      - name: Step two
        run: "Get-ChildItem Env: | Sort-Object Name"
```

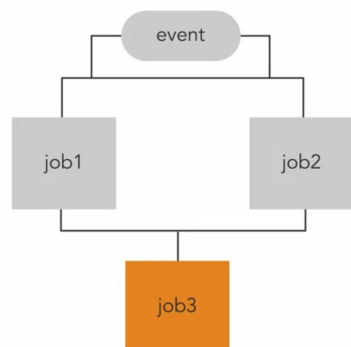
### Adding dependencies.

->Normally, jobs run in parallel.

**needs:** identifies one or more jobs that must complete successfully before a job will run.

e.g.

```
jobs:
  job1:
  job2:
  job3:
    needs: [job1, job2]
```



## Conditionals

Ignore branches	branches, branches-ignore
use tags	tags, tags-ignore
Special characters	'release/*'

## Syntax:

on:

[trigger]:

branches:

- develop
- master

[trigger]:

branches-ignore:

- develop

## Workflows and Action Limitations

->Workflow concurrent is limited 20. In a single repo, you can run 20 workflows at the same time.

Job concurrency is limited based on your plan.

GitHub Plan	Concurrent Jobs per Repository
Free	20
Pro	40
Team	60
Enterprise	180

->1000 API request per hour

->Actions cannot trigger other workflows

->Logs are limited to 64 KB.

->If you exceed limits, job queueing or failures.