# Workflows in GitHub Actions

## Hands-on Cheat Sheet

Practical notes distilled into runnable examples

November 3, 2025

## How to use this sheet

This document focuses on the essentials for writing, understanding, and running GitHub Actions *workflows*: YAML structure, triggers, jobs/steps, and runners. Examples are drop-in and minimal. Compile this LaTeX with `pdflatex -shell-escape` (or `xelatex -shell-escape`) to enable `minted`.

## 1 YAML in 90 seconds

YAML is indentation-sensitive and built from **scalars** (numbers, strings, booleans, null) and **collections** (lists with `-` and maps/dictionaries as key: value).

```yaml
number: 42
pi: 3.14159
title: "Hello, YAML"
flag: true
nothing: null   # or ~
list:
  - build
  - test
  - deploy
map:
  name: "Mirana"
  traits:
    color: black
    friendly: true
  skills: [jump, play, nap]
```

# 2 Workflow anatomy

Workflow files live under `.github/workflows/*.yml` and are made of:

- `name`: a human-friendly title.
- `on`: *triggers* that launch runs.
- `jobs`: one or more jobs (parallel by default).
- Within each job: `runs-on` (runner label) and `steps` (commands or actions).

```
name: "Workflow Overview"
on: push
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - name: Build
        run: echo "Building..."

  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Deploy
        run: echo "Deploying..."
```

# 3 Triggers (`on`)

You can launch workflows manually, on a schedule, or when repository events occur.

## Manual

```
on:
  workflow_dispatch: {}
```

## Schedule (CRON)

```
on:
  schedule:
    - cron: "0 9 * * 1-5"  # Weekdays at 09:00 UTC
```

## Repository activity

```
on:
  push:
    branches: [main]
  issues:
    types: [opened, labeled]
```

## 4 Jobs and steps

Jobs define the runner and contain *steps*. Steps either `run` shell commands or `uses` a prebuilt action.

```
jobs:
  my_job:
    name: "Build and Test"
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Use Node.js
        uses: actions/setup-node@v4
        with:
          node-version: "20"

      - name: Unit tests
        run: npm test

      - name: Python step (explicit shell)
        shell: python
        run: |
          print("Hello from Python shell")
```

## 5 A complete minimal workflow

```yaml
name: "Workflow Overview"
on:
  workflow_dispatch:
  push: { branches: [main] }

jobs:
  a_simple_job:
    name: "My First Job"
    runs-on: ubuntu-latest
    steps:
      - name: Echo from bash
        id: stepId
        run: echo "This is the first step from the first job"

      - name: Print from Python shell
        shell: python
        run: |
          print("Hello from Python")

  # Example of a second job that could run in parallel
  another_job:
    runs-on: ubuntu-latest
    steps:
      - run: echo "Another job running in parallel"
```

# 6 Runners: GitHub-hosted vs Self-hosted

**GitHub-hosted** runners are ephemeral VMs (Ubuntu, Windows, macOS) provided by GitHub and preloaded with common tooling. **Self-hosted** runners are machines you manage (on-prem, VM, or personal) where you install and register the runner. Select a runner with `runs-on`:

```
runs-on: ubuntu-latest     # GitHub-hosted
# runs-on: self-hosted     # Your machine/VM
```

## Setup a self-hosted runner (Windows example)

Run these in an elevated PowerShell:

```
mkdir C:\actions-runner; cd C:\actions-runner
# Download the runner package (URL provided by GitHub UI)
Invoke-WebRequest -Uri "<runner-zip-url>" -OutFile actions-runner.zip

# (Optional) Validate integrity (command shown in UI)
# Expand the archive
Add-Type -AssemblyName System.IO.Compression.FileSystem
[IO.Compression.ZipFile]::ExtractToDirectory("actions-runner.zip", ".")

# Configure the runner (token/URL provided by GitHub UI)
.\config.cmd --url https://github.com/<owner>/<repo> --token <TOKEN>

# Install as a service (optional)
.\svc install
.\svc start
```

If you prefer a plain CMD step in a job on Windows, set `shell: cmd` to avoid PowerShell execution policy issues:

```
shell: cmd
run: |
  echo This is a multiline command
  echo Executed on a self-hosted runner
```

# 7 Helpful tips

- Keep each workflow focused (build, test, release) for clarity.
- Use explicit `name` for workflows, jobs, and steps for nicer logs.
- Pin actions to major versions (e.g., `actions/checkout@v4`).
- When using Windows self-hosted, `shell: cmd` is a simple default.
- Combine multiple triggers when useful (e.g., `push` and `workflow_dispatch`).

*These notes summarize core ideas from lesson material on workflows, YAML, triggers, jobs/steps, and runners, plus pragmatic tips and examples.*