

# GITHUB ACTIONS QUICK REFERENCE

Workflows, triggers, jobs, secrets, caching, artifacts

## Workflow Basics

### Minimal Workflow

```
name: CI
on: push
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - run: echo "Hello from CI"
```

### Key Concepts

**Workflow** YAML file in `.github/workflows/` that defines automation  
**Event** Trigger that starts a workflow (push, PR, schedule, etc.)  
**Job** Set of steps that run on the same runner  
**Step** Individual task — runs a command or uses an action  
**Runner** VM that executes jobs (`ubuntu-latest`, `macos-latest`, `windows-latest`)  
**Action** Reusable unit of code referenced with `uses:`

## Triggers

### Common Events

```
on:
  push:
    branches: [main]
  pull_request:
    branches: [main]
  schedule:
    - cron: "0 6 * * 1" # every Monday 6 AM UTC
  workflow_dispatch: # manual trigger
```

### Event Filters

**branches:** Trigger only for specific branches  
**paths:** Trigger only when matching files change  
**tags:** Trigger on tag pushes (`v*`)  
**types: [opened, synchronize]** Filter PR activity types  
**branches-ignore:** Exclude specific branches  
**paths-ignore:** Exclude specific file paths

## Jobs & Steps

### Job Configuration

```
jobs:
  tests:
    runs-on: ubuntu-latest
    needs: build # depends on build job
    if: github.ref == 'refs/heads/main'
    timeout-minutes: 10
    steps:
      - uses: actions/checkout@v4
      - run: npm test
```

### Step Types

**run:** Execute a shell command  
**uses:** Use a published action  
**with:** Pass inputs to an action  
**name:** Display name in the UI  
**id:** Reference step outputs via `steps.<id>.outputs`  
**if:** Conditional execution  
**continue-on-error: true** Don't fail the job if step fails

## Actions

### Using Actions

```
steps:
  - uses: actions/checkout@v4
  - uses: actions/setup-node@v4
    with:
      node-version: 20
  - uses: ./github/actions/my-action # local action
```

### Popular Actions

**actions/checkout@v4** Check out repository code  
**actions/setup-node@v4** Install Node.js  
**actions/setup-python@v5** Install Python  
**actions/upload-artifact@v4** Upload build artifacts  
**actions/download-artifact@v4** Download artifacts from another job  
**actions/cache@v4** Cache dependencies between runs  
**actions/github-script@v7** Run JS with the GitHub API client

## Environment Variables

### Setting Variables

```
env:
  NODE_ENV: production # workflow-level
jobs:
  build:
    env:
      CI: true # job-level
    steps:
      - run: echo "$MY_VAR"
        env:
          MY_VAR: hello # step-level
```

### Default Variables

**github.sha** Commit SHA that triggered the workflow  
**github.ref** Branch or tag ref (`refs/heads/main`)  
**github.repository** Owner/repo name  
**github.actor** User who triggered the workflow  
**github.event\_name** Event that triggered the workflow  
**runner.os** Runner OS (`Linux`, `macOS`, `Windows`)

## Secrets

### Using Secrets

```
steps:
  - run: deploy --token "${TOKEN}"
    env:
      TOKEN: ${ secrets.DEPLOY_TOKEN }
  - uses: some/action@v1
    with:
      api-key: ${ secrets.API_KEY }
```

### Secret Rules

**secrets.GITHUB\_TOKEN** Auto-generated token scoped to the repo  
**Settings → Secrets (Masking)** Add secrets in repo or org settings  
Secret values are masked in logs automatically  
**Environment secrets (Org secrets)** Scoped to a deployment environment  
Shared across repos in an organization

## Matrix Strategy

### Matrix Builds

```
jobs:
  test:
    strategy:
      matrix:
        os: [ubuntu-latest, macos-latest]
        node: [18, 20]
    runs-on: ${ matrix.os }
    steps:
      - uses: actions/setup-node@v4
        with:
          node-version: ${ matrix.node }
```

### Matrix Options

**matrix:** Define variable combinations to expand  
**include:** Add extra combinations to the matrix  
**exclude:** Remove specific combinations  
**fail-fast: false** Continue other jobs if one fails  
**max-parallel: 2** Limit concurrent matrix jobs

## Caching

### Cache Dependencies

```
- uses: actions/cache@v4
  with:
    path: ~/.npm
    key: npm-${ hashFiles('package-lock.json') }
    restore-keys: npm-
```

### Built-in Caching

```
- uses: actions/setup-node@v4
  with:
    node-version: 20
    cache: npm # auto-cache for npm/yarn/pnpm
- uses: actions/setup-python@v5
  with:
    python-version: "3.12"
    cache: pip # auto-cache for pip
```

## Artifacts

### Upload & Download

```
- uses: actions/upload-artifact@v4
  with:
    name: build-output
    path: dist/
    retention-days: 7
- uses: actions/download-artifact@v4
  with:
    name: build-output
```

### Artifact Notes

**retention-days** Auto-delete after N days (default 90)  
**path** File or directory to upload (supports globs)  
**Cross-job** Upload in one job, download in another with `needs:`

**compression-level** 0 (none) to 9 (max), default 6

## Common Patterns

### Conditional Deployment

```
- name: Deploy to production
  if: github.ref == 'refs/heads/main'
  run: ./deploy.sh
- name: Post PR comment
  if: github.event_name == 'pull_request'
  run: gh pr comment $PR --body "Build passed"
```

### Useful Expressions

**success()** True if all previous steps passed  
**failure()** True if any previous step failed  
**always()** Run regardless of status (cleanup)  
**cancelled()** True if workflow was cancelled  
**contains(github.ref, 'release')** String contains check  
**startsWith(github.ref, 'refs/tags')** String prefix check  
**hashFiles('\*\*/lock\*')** SHA-256 of files (for cache keys)