

# Référence rapide GitHub Actions

Workflows, déclencheurs, jobs, secrets, cache, artefacts

## Bases des workflows

### Workflow minimal

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

### Concepts clés

<b>Workflow</b>	Fichier YAML dans <code>.github/workflows/</code> qui définit l'automatisation
<b>Event</b>	Déclencheur qui démarre un workflow (push, PR, schedule, etc.)
<b>Job</b>	Ensemble d'étapes s'exécutant sur le même runner
<b>Step</b>	Tâche individuelle — exécute une commande ou utilise une action
<b>Runner</b>	VM qui exécute les jobs ( <b>ubuntu-latest</b> , <b>macos-latest</b> , <b>windows-latest</b> )
<b>Action</b>	Unité de code réutilisable référencée avec <b>uses:</b>

## Déclencheurs

### Événements courants

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

### Filtres d'événements

<b>branches:</b>	Déclencher uniquement pour des branches spécifiques
<b>paths:</b>	Déclencher uniquement quand les fichiers correspondants changent
<b>tags:</b>	Déclencher sur les push de tags ( <b>v*</b> )
<b>types: [opened, synchronize]</b>	Filtrer les types d'activité PR
<b>branches-ignore:</b>	Exclure des branches spécifiques
<b>paths-ignore:</b>	Exclure des chemins de fichiers spécifiques

## Jobs et étapes

### Configuration d'un job

```
jobs:
  test:
    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
```

### Types d'étapes

<b>run:</b>	Exécuter une commande shell
<b>uses:</b>	Utiliser une action publiée
<b>with:</b>	Passer des entrées à une action
<b>name:</b>	Nom d'affichage dans l'interface
<b>id:</b>	Référencer les sorties via <b>steps.&lt;id&gt;.outputs</b>
<b>if:</b>	Exécution conditionnelle
<b>continue-on-error: true</b>	Ne pas faire échouer le job si l'étape échoue

## Actions

### Utiliser des actions

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

### Actions populaires

<b>actions/checkout@v4</b>	Récupérer le code du dépôt
<b>actions/setup-node@v4</b>	Installer Node.js
<b>actions/setup-python@v5</b>	Installer Python
<b>actions/upload-artifact@v4</b>	Téléverser les artefacts de build
<b>actions/download-artifact@v4</b>	Télécharger les artefacts d'un autre job
<b>actions/cache@v4</b>	Mettre en cache les dépendances entre les runs
<b>actions/github-script@v7</b>	Exécuter du JS avec le client API GitHub

## Variables d'environnement

### Définir des 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
```

### Variables par défaut

<b>github.sha</b>	SHA du commit qui a déclenché le workflow
<b>github.ref</b>	Référence de branche ou de tag ( <b>refs/heads/main</b> )
<b>github.repository</b>	Nom propriétaire/dépôt
<b>github.actor</b>	Utilisateur qui a déclenché le workflow
<b>github.event_name</b>	Événement qui a déclenché le workflow
<b>runner.os</b>	OS du runner ( <b>Linux</b> , <b>macOS</b> , <b>Windows</b> )

## Secrets

### Utiliser des secrets

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

### Règles pour les secrets

<b>secrets.GITHUB_TOKEN</b>	Token généré automatiquement, limité au dépôt
<b>Settings → Secrets</b>	Ajouter des secrets dans les paramètres du dépôt ou de l'organisation
<b>Masquage</b>	Les valeurs des secrets sont masquées automatiquement dans les logs
<b>Environment secrets</b>	Limités à un environnement de déploiement
<b>Org secrets</b>	Partagés entre les dépôts d'une organisation

## Stratégie de matrice

### Builds matriciels

```
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 }
```

### Options de matrice

<b>matrix:</b>	Définir les combinaisons de variables à développer
<b>include:</b>	Ajouter des combinaisons supplémentaires à la matrice
<b>exclude:</b>	Supprimer des combinaisons spécifiques
<b>fail-fast: false</b>	Continuer les autres jobs si l'un échoue
<b>max-parallel: 2</b>	Limiter les jobs matriciels simultanés

## Cache

### Mettre en cache les dépendances

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

### Cache intégré

```
- 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
```

## Artefacts

### Téléverser et télécharger

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

### Notes sur les artefacts

<b>retention-days</b>	Suppression automatique après N jours (défaut : 90)
<b>path</b>	Fichier ou répertoire à téléverser (supporte les globs)
<b>Cross-job</b>	Téléverser dans un job, télécharger dans un autre avec <b>needs:</b>
<b>compression-level</b>	0 (aucune) à 9 (max), défaut 6

# Référence rapide GitHub Actions

---

## Patterns courants

### Déploiement conditionnel

```
- 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"
```

### Expressions utiles

<b>success()</b>	Vrai si toutes les étapes précédentes ont réussi
<b>failure()</b>	Vrai si une étape précédente a échoué
<b>always()</b>	S'exécute quel que soit le statut (nettoyage)
<b>cancelled()</b>	Vrai si le workflow a été annulé
<b>contains(github.ref, 'release')</b>	Vérification de contenu de chaîne
<b>startsWith(github.ref, 'refs/tags')</b>	Vérification de préfixe de chaîne
<b>hashFiles('**/lock*')</b>	SHA-256 des fichiers (pour les clés de cache)