

RÉFÉRENCE RAPIDE GITHUB ACTIONS

Workflows, déclencheurs, jobs, secrets, cache, artefacts

Bases des workflows

Workflow minimal

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

Concepts clés

Workflow Fichier YAML dans `.github/workflows/` qui définit l'automatisation

Event Déclencheur qui démarre un workflow (push, PR, schedule, etc.)

Job Ensemble d'étapes s'exécutant sur le même runner

Step Tâche individuelle — exécute une commande ou utilise une action

Runner VM qui exécute les jobs (`'ubuntu-latest'`, `'macos-latest'`, `'windows-latest'`)

Action Unité de code réutilisable référencée avec ``uses:``

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

branches: Déclencher uniquement pour des branches spécifiques

paths: Déclencher uniquement quand les fichiers correspondants changent

tags: Déclencher sur les push de tags (`"*"`)

types: [opened, synchronize] Filtrer les types d'activité PR

branches-ignore: Exclure des branches spécifiques

paths-ignore: 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

run: Exécuter une commande shell

uses: Utiliser une action publiée

with: Passer des entrées à une action

name: Nom d'affichage dans l'interface

id: Référencer les sorties via ``steps.<id>.outputs``

if: Exécution conditionnelle

continue-on-error: true 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

actions/checkout@v4 Récupérer le code du dépôt

actions/setup-node@v4 Installer Node.js

actions/setup-python@v5 Téléverser Python

actions/upload-artifact@v4 Téléverser les artefacts de build

actions/download-artifact@v4 Télécharger les artefacts d'un autre job

actions/cache@v4 Mettre en cache les dépendances entre les runs

actions/github-script@v7 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

github.sha SHA du commit qui a déclenché le workflow

github.ref Référence de branche ou de tag (`'refs/heads/main'`)

github.repository Nom propriétaire/dépôt

github.actor Utilisateur qui a déclenché le workflow

github.event_name Événement qui a déclenché le workflow

runner.os OS du runner (`'Linux'`, `'macOS'`, `'Windows'`)

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

secrets.GITHUB_TOKEN Token généré automatiquement, limité au dépôt

Settings → Secrets Ajouter des secrets dans les paramètres du dépôt ou de l'organisation

Masquage Les valeurs des secrets sont masquées automatiquement dans les logs

Environment secrets Limités à un environnement de déploiement

Org secrets Partagés entre les dépôts d'une organisation

Stratégie de matrice

Buils 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

matrix: Définir les combinaisons de variables à développer

include: Ajouter des combinaisons supplémentaires à la matrice

exclude: Supprimer des combinaisons spécifiques

fail-fast: false Continuer les autres jobs si l'un échoue

max-parallel: 2 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

retention-days Suppression automatique après N jours (défaut: 90)

path Fichier ou répertoire à téléverser (supporte les globs)

Cross-job Téléverser dans un job, télécharger dans un autre avec ``needs:``

compression-level 0 (aucune) à 9 (max), défaut 6

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

success() Vrai si toutes les étapes précédentes ont réussi

failure() Vrai si une étape précédente a échoué

always() S'exécute quel que soit le statut (nettoyage)

cancelled() Vrai si le workflow a été annulé

contains(github.ref, 'release')

startsWith(github.ref, 'refs/tags')

hashFiles('/lock*')**

Vérification de contenu de chaîne

Vérification de préfixe de chaîne

SHA-256 des fichiers (pour les clés de cache)