

# Workflow

## Étapes de génération du portail

### 1. Push vers GitHub

Lorsque vous effectuez un `git push` vers une branche, les fichiers modifiés sont envoyés au dépôt distant. Cela déclenche les workflows configurés dans GitHub Actions si des événements correspondants sont définis (par exemple, sur `push`).

### 2. Déclenchement du Workflow GitHub Actions

GitHub détecte le **push** et exécute le fichier de workflow défini dans `.github/workflows/main.yml`.

### 3. Installation de l'environnement

Dans un projet MkDocs, les étapes incluent :

- Vérifier le code source en le clonant.
- Installer Python et les dépendances requises :
  - MkDocs et ses plugins (`mkdocs-material`, etc.).
  - Toute autre bibliothèque spécifiée dans le workflow.

### 4. Installation de l'environnement

Dans un projet MkDocs, les étapes typiques incluent :

- Vérifier le code source en le clonant.
- Installer Python et les dépendances requises :
  - MkDocs et ses plugins (`mkdocs-material`, etc.).
  - Toute autre bibliothèque spécifiée dans le workflow.

### 5. Génération des fichiers statiques

Une fois l'environnement prêt, la commande `mkdocs build` est exécutée pour convertir les fichiers Markdown en pages HTML statiques, selon la configuration définie dans `mkdocs.yml`.

### 6. Déploiement sur GitHub Pages

Si la branche concernée est `main`, la commande `mkdocs gh-deploy` est exécutée :

- Elle génère et pousse les fichiers HTML vers la branche `gh-pages`.
- GitHub Pages sert automatiquement ces fichiers comme un site web.

### 7. Mise à jour du site

Après le déploiement sur `gh-pages`, le site hébergé sur GitHub Pages (par ex. <https://inria.github.io/datalake/>) est automatiquement mis à jour avec les dernières modifications.

## 8. Notification et suivi

GitHub affiche les logs du workflow dans l'onglet **Actions**, où vous pouvez vérifier le succès ou l'échec des différentes étapes. En cas de problème (ex. : erreur de configuration), vous recevez une notification pour corriger et pousser une mise à jour.

## Debugger le workflow

The screenshot shows the GitHub Actions interface for the 'datalake' repository. The 'Actions' tab is selected, and the 'pages build and deployment' workflow run is highlighted. A red arrow points to the 'Actions' tab in the top navigation bar. Another red arrow points to the 'pages build and deployment' workflow run in the 'All workflows' section. A third red arrow points to the 'build' job in the workflow diagram.

The interface displays the following information:

- Repository:** INRIA / datalake
- Workflow:** pages build and deployment
- Run:** pages-build-deployment #25: by github-pages [bot]
- Status:** Success
- Total duration:** 24s
- Artifacts:** 1

The workflow diagram shows the following jobs:

- build** (5s)
- report-build-status** (3s)
- deploy** (10s) - URL: <https://inria.github.io/datalake/>

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

pages build and deployment #25 Re-run all jobs ⋮

Summary

Jobs

- build
- report-build-status
- deploy

Run details

- Usage

**build**  
succeeded 17 hours ago in 5s

Search logs

- > Set up job 1s
- > Checkout 1s
- > Upload artifact 1s
- > Post Checkout 0s
- > Complete job 0s

**i** Texte générés en partie par ChatGPT !