

# Atelier N°1 (DevOps)

Mise en place du dépôt Git et introduction au pipeline CI/CD avec gestion des branches

## Partie 2: Création du projet local

- ✓ Créer le dossier de projet → `mkdir projet_devops ; cd projet_devops`
- ✓ Initialiser le dépôt Git → `git init`
- ✓ Créer un premier fichier → `echo "# Atelier1 DevOps – CI/CD" > README.md`
- ✓ Ajouter → `git add .`
- ✓ Valider → `git commit -m "Initialisation du projet"`

```
Documents/DevOps/AtelierN°1
> mkdir projet_devops

Documents/DevOps/AtelierN°1
> git init
Initialised empty Git repository in /home/pc/Documents/DevOps/AtelierN°1/.git/

AtelierN°1 on ↙ main
> echo "# Atelier1 DevOps – CI/CD" > README.md

AtelierN°1 on ↙ main [?]
> ls -a
.  ..  .git  projet_devops  README.md

AtelierN°1 on ↙ main [?]
> git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add" to track)

AtelierN°1 on ↙ main [?]
> git add .
```

```
AtelierN°1 on ◀ main [+]
> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

AtelierN°1 on ◀ main [+]
> git commit -m "Initialisation du projet"
[main (root-commit) a4ddfb5] Initialisation du projet
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

AtelierN°1 on ◀ main
> □
```

### Partie 3: Gestion des branches

- ✓ Créer une nouvelle branche → git branch feature-readme
- ✓ Se déplacer sur la nouvelle branche → git checkout feature-readme
- ✓ Modifier le fichier README.md → echo "Cet atelier illustre le début d'un pipeline DevOps." >> README.md
- ✓ Enregistrer les modifications → git add README.md

git commit -m "Mise à jour du README dans la branche feature-readme"

- ✓ Revenir sur la branche principale git checkout main
- ✓ Fusionner les modifications → git merge feature-readme

```
AtelierN°1 on ⌵ main
> git branch feature-readme

AtelierN°1 on ⌵ main
> git checkout feature-readme
Switched to branch 'feature-readme'

AtelierN°1 on ⌵ feature-readme
> echo "Cet atelier illustre le début d'un pipeline DevOps." >> README.md

AtelierN°1 on ⌵ feature-readme [!]
> git add README.md

AtelierN°1 on ⌵ feature-readme [+]
> git commit -m "Mise à jour du README dans la branche feature-readme"
[feature-readme eaba97f] Mise à jour du README dans la branche feature-readme
1 file changed, 1 insertion(+)
```

```
AtelierN°1 on ⌵ feature-readme
> git checkout main
Switched to branch 'main'

AtelierN°1 on ⌵ main
> git merge feature-readme
Updating a4ddfb5..eaba97f
Fast-forward
 README.md | 1 +
1 file changed, 1 insertion(+)
```

## Partie 4: Dépôt GitHub

✓ Créer le dépôt distant → Sur <https://github.com> → New repository → nommer atelier\_devops.

**Create a new repository**

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

*Required fields are marked with an asterisk (\*).*

**1 General**

**Owner \*** / **Repository name \***

AnasMAC / atelier\_devops

✓ atelier\_devops is available.

Great repository names are short and memorable. How about congenial-system?

**Description**

just atelier for course devops with Prof : M.GHAILANI

54 / 350 characters

**2 Configuration**

**Choose visibility \***  
Choose who can see and commit to this repository

Public

**Add README**  
READMEs can be used as longer descriptions. [About READMEs](#)

Off

✓ Lier le dépôt local à GitHub → `git remote add origin`

`https://github.com/votre-nom-utilisateur/atelier_devops.git`

Explication :

- `git remote` → permet de gérer les dépôts distants (remotes).

- `add` → ajoute un nouveau dépôt distant.

- `origin` → c'est le nom symbolique (alias) donné au dépôt distant principal.

→ Par convention, on l'appelle toujours `origin`.

- `https://github.com/votre-nom-utilisateur/atelier_devops.git` → c'est l'URL du dépôt GitHub.

→ On dit à Git : "Mon dépôt local doit être relié à ce dépôt GitHub que j'appelle `origin`."

✓ Renomme la branche actuelle en `main`, qui devient la branche principale du projet.

`git branch -M main`

→ l'option `-M` force le renommage même si une branche `main` existe déjà.

`git push -u origin main`


→ envoie la branche `main` locale vers le dépôt distant nommé `origin` (par exemple sur GitHub).


L'option `-u` crée un lien de suivi entre la branche locale et la branche distante, ce qui permet ensuite



d'utiliser simplement : `git push` et `git pull`

```
telierN°1 on ↙ main
git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 572 bytes | 572.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AnasMAC/atelier_devops.git
* [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

telierN°1 on ↙ main took 11s
```

 Public repository


 main ↕


 


[Go to file](#)

[+](#)


[Code](#) ↕

 **AnasMAC**

Mise à jour du README dans la branche featu... eaba97f · 48 minutes ago 

 README.md


Mise à jour du README dans la b... 48 minutes ago

 README ✎

# Atelier1 DevOps – CI/CD

---

Cet atelier illustre le début d'un pipeline DevOps.



## Releases