



# Connect a GitHub Repo with AWS

A

Amirul Zaol-kefli

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
[ec2-user@ip-172-31-46-235 nextwork-web-project]$ sudo dnf update -y
sudo dnf install git -y
Verifying      : perl-Git-2.50.1-1.amzn2023.0.1.noarch
Verifying      : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64
Verifying      : perl-lib-0.65-477.amzn2023.0.7.x86_64

Installed:
  git-2.50.1-1.amzn2023.0.1.x86_64      git-core-2.50.1-1.amzn2023.0.1.x86_64
  perl-Git-2.50.1-1.amzn2023.0.1.noarch  perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64

Complete!
● [ec2-user@ip-172-31-46-235 nextwork-web-project]$ git --version
git version 2.50.1
○ [ec2-user@ip-172-31-46-235 nextwork-web-project]$ 
```

# Introducing Today's Project!

In this project, I will demonstrate connect a GitHub Repo with AWS. I'm doing this project to learn how to setup Git and Github repo and connect my web app to it.

## Key tools and concepts

Services I used were AWS EC2 and Github CLI. Key concepts I learnt include using Github CLI and SSH to upload changes into Github Repo.

## Project reflection

This project took me approximately 3 hours. The most challenging part was dealing with github authentication stuff in CLI. It was most rewarding to be able to see my changes in git as EC2 User.

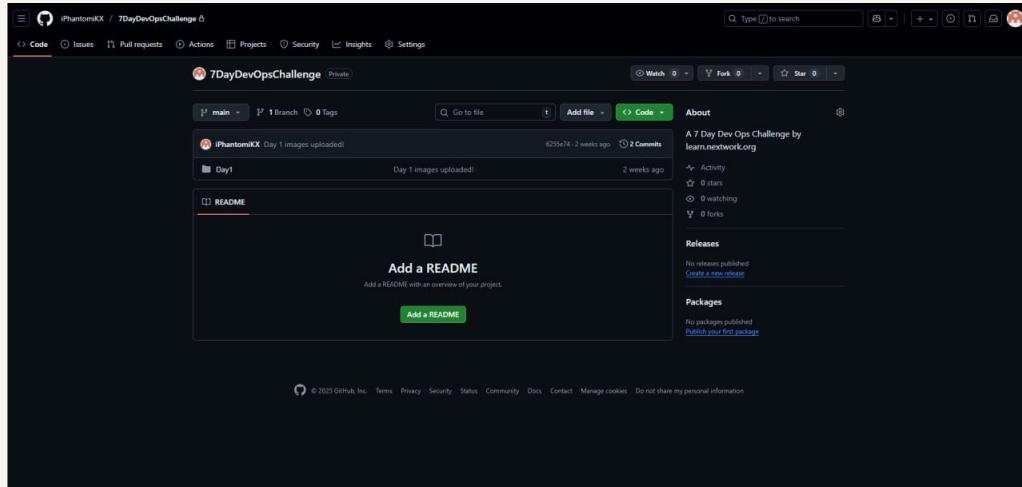
I did this project because I wanted to learn how to use git with AWS EC2.

This project is part two of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project next Thursday, for part 3.

# Git and GitHub

Git is a version control that tracks every changes I make. I installed Git using the commands `sudo dnf update -y` `sudo dnf install git -y` and verified it using `git --version`.

GitHub is a website that stores all my repos. I'm using GitHub in this project to be able to see my changes in all my project repos.

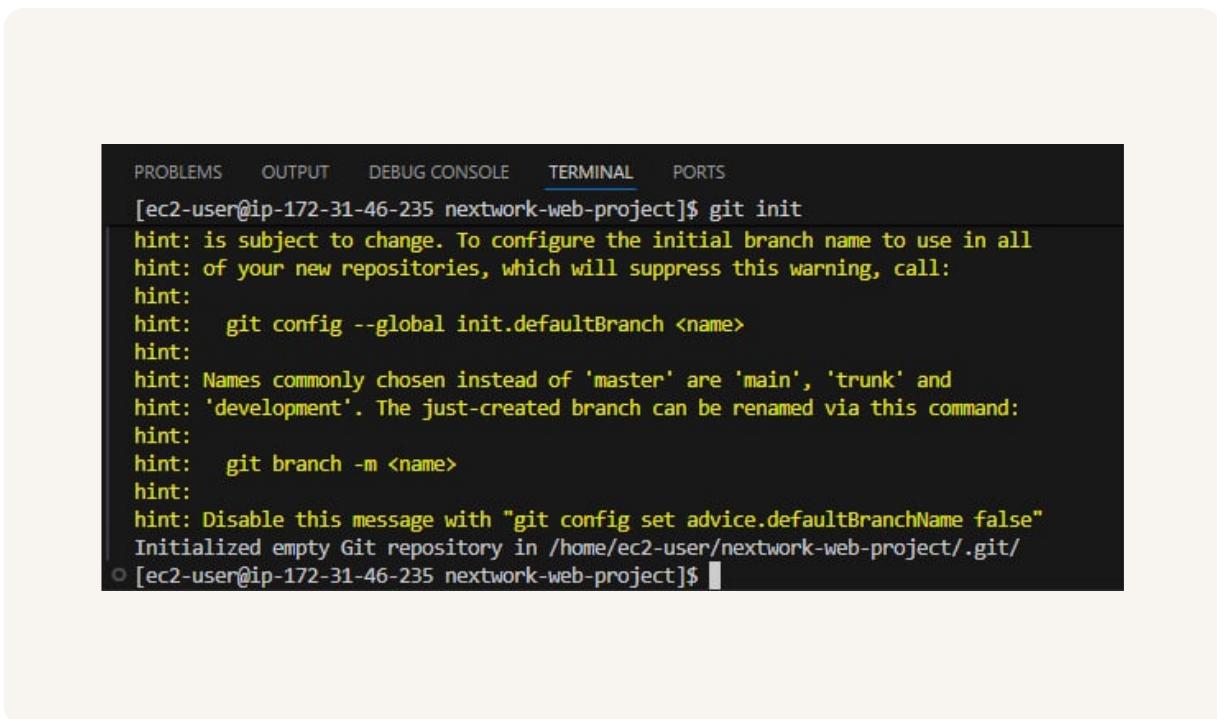


# My local repository

A Git repository are folders that contain all my project files and their entire version history.

Git init is a command that sets up the directory as a local Git repository which means changes are now tracked for version control.

A branch in Git is a parallel version or 'alternate universe' of the same project. After running git init, the response from the terminal was it is now on the main(master) branch.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[ec2-user@ip-172-31-46-235 nextwork-web-project]$ git init
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
hint:
hint: Disable this message with "git config set advice.defaultBranchName false"
Initialized empty Git repository in /home/ec2-user/nextwork-web-project/.git/
[ec2-user@ip-172-31-46-235 nextwork-web-project]$ █
```

## To push local changes to GitHub, I ran three commands

`git add`

The first command I ran was `git add .` A staging area is telling Git to put together all your modified files for a final review before you commit them.

`git commit`

The second command I ran was `git commit -m <message>`. Using '`-m`' means adding message to the commit.

`git push`

The third command I ran was `git push -u origin master`. Using '`-u`' means upstream to my local branch, telling git to push to master branch by default.

# Authentication

When I commit changes to GitHub, Git asks for my credentials because it needs to know who is committing the changes for security purposes.

## Local Git identity

Git needs my name and email because it needs to track who made the changes.

Running git log showed me that who made what change.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
commit 473e707e7aaae4d7e276939645eba21a192605c9 (HEAD -> main, origin/main, origin/HEAD)
Author: EC2 Default User <ec2-user@ip-172-31-46-235.ap-southeast-1.compute.internal>
Date:   Thu Sep 25 05:09:26 2025 +0000

    Changed index.jsp to 3!

commit f37b6178e7fd3868410d2f2a6748ba04e6467a18
Author: iPhantomikX <cmpsun98@gmail.com>
Date:   Thu Sep 25 13:05:51 2025 +0800

    Push Day 2 stuff

commit b1e29857d6090325e99b2a9768a034536a58316d
:|
```

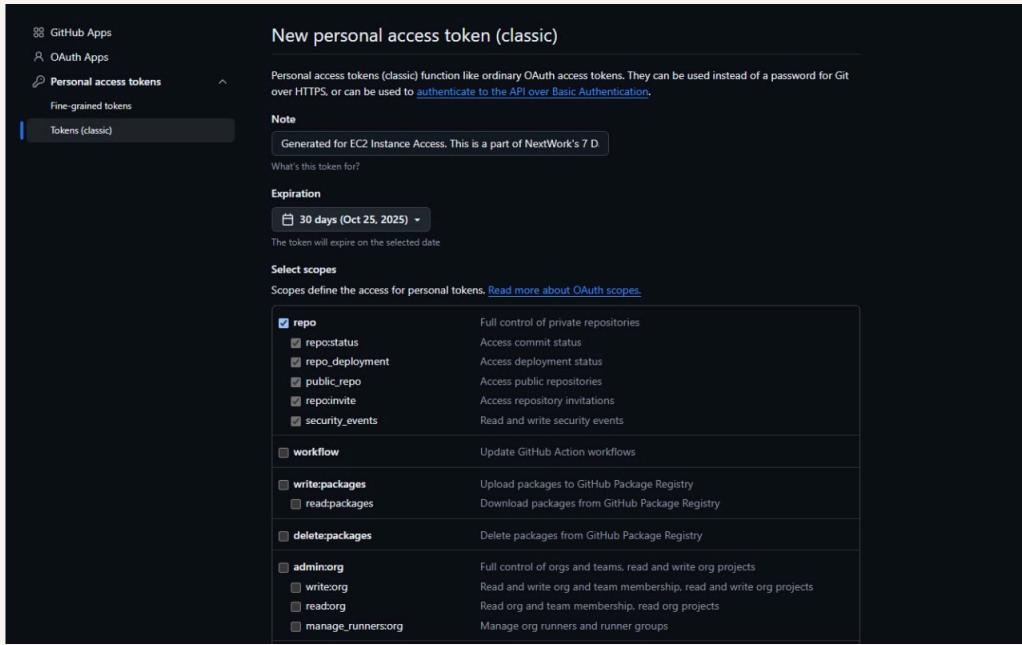


## GitHub tokens

GitHub authentication failed when I entered my password because Github phased out password authentication to connect with repositories over HTTPS - there are too many security risks and passwords can get intercepted over the internet.

A GitHub token is a key to access the repo from CLI. I'm using one in this project because Github Password Authentication is phased out for CLI.

I could set up a GitHub token by going into Developer Settings and generate a key for that gives me write access to all my repos.



The screenshot shows the GitHub 'Personal access tokens' page under the 'Tokens (classic)' tab. It displays a form for generating a new token:

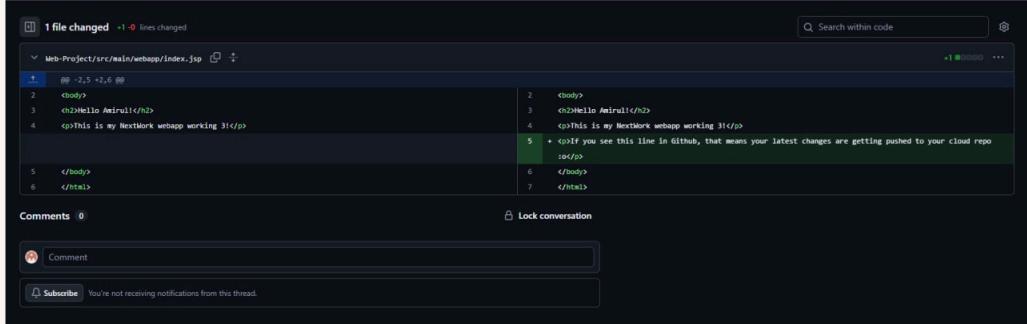
- Note:** Generated for EC2 Instance Access. This is a part of NextWork's 7 D.
- Expiration:** Set to 30 days (Oct 25, 2025).
- Select scopes:** A table showing available OAuth scopes:

Scope	Description
<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> <b>repo:status</b>	Access commit status
<input checked="" type="checkbox"/> <b>repo:deployment</b>	Access deployment status
<input checked="" type="checkbox"/> <b>public_repo</b>	Access public repositories
<input checked="" type="checkbox"/> <b>repo:invite</b>	Access repository invitations
<input checked="" type="checkbox"/> <b>security_events</b>	Read and write security events
<input type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input type="checkbox"/> <b>read:packages</b>	Download packages from GitHub Package Registry
<input type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams. Read and write org projects
<input type="checkbox"/> <b>write:org</b>	Read and write org and team membership, read and write org projects
<input type="checkbox"/> <b>read:org</b>	Read org and team membership, read org projects
<input type="checkbox"/> <b>manage:runner:org</b>	Manage org runners and runner groups

# Making changes again

I wanted to see Git working in action, so I pushed changes into the git. I couldn't see the changes in my GitHub repo initially because I am apparently uploading changes to local repo.

I finally saw the changes in my GitHub repo after writing commands that send (push) them from my local repository into my origin.



A screenshot of a GitHub commit interface. The commit summary at the top says "1 file changed +1-0 lines changed". Below it, a file named "index.jsp" is shown with its code. The code contains several lines of JSP scriptlets and comments. A specific line of code is highlighted with a green background: "+ <p>If you see this line in Github, that means your latest changes are getting pushed to your cloud repo</p>". This indicates that the changes have been successfully pushed to the remote repository.

```
diff --git a/WEB-INF/jsp/index.jsp b/WEB-INF/jsp/index.jsp
--- a/WEB-INF/jsp/index.jsp
+++ b/WEB-INF/jsp/index.jsp
@@ -2,2 +2,6 @@
 2   <body>
 3     <h2>Hello Amirul!</h2>
 4     <p>This is my NextWork webapp working 3!</p>
+
+   <p>If you see this line in Github, that means your latest changes are getting pushed to your cloud repo</p>
 5   </body>
 6 </html>
```



[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

