



Connect a GitHub Repo with AWS

AM

ampahben3@gmail.com

```
Last metadata expiration check: 22:35:34 ago on Tue May 13 15:25:00 2025.
=====
WARNING:
  A newer release of "Amazon Linux" is available.

Available Versions:

Version 2023.7.20250512:
  Run the following command to upgrade to 2023.7.20250512:

    dnf upgrade --releasever=2023.7.20250512

Release notes:
  https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.7.20250512.html

=====
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 22:35:36 ago on Tue May 13 15:25:00 2025.
Package git-2.47.1-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
> [ec2-user@ip-172-31-25-59 nextwork-web-project]$ git --version
unknown option: --verion
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
           [--no-optimal-locks] [--no-advice] [-bare] [-git-dir=<path>]
           [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
           <command> [<args>]
> [ec2-user@ip-172-31-25-59 nextwork-web-project]$ git --version
git version 2.47.1
> [ec2-user@ip-172-31-25-59 nextwork-web-project]$ 
```

Introducing Today's Project!

Today, I'm exploring connecting a GitHub repo to AWS. A DevOps engineer must master AWS tools like CodePipeline, CodeBuild, and CloudFormation. This project automates builds, tests, and deployments to EC2. A CI/CD Project Ready for something huge!!!!

Key tools and concepts

Services I used were GitHub and AWS EC2. Key concepts I learnt include setting up SSH for secure access, cloning a GitHub repo to a remote server, and deploying a web app by managing files and services directly on an EC2 instance.

Project reflection

This project took me approximately an hour. The most challenging part was dealing with EC2 connectivity issues. It was most rewarding to successfully connect my GitHub repo to the EC2 instance and understand the networking setup better.

I did this project today to improve my DevOps skills with AWS and build a strong portfolio I can confidently share with recruiters. Yes, it met my goals by giving me hands-on experience and reinforcing key cloud deployment concepts.

AM

ampahben3@gmail.com

NextWork Student

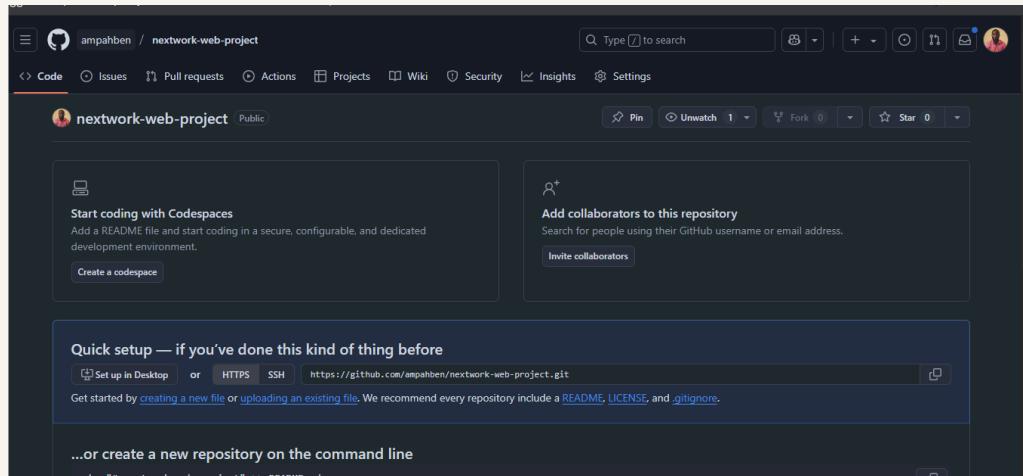
nextwork.org

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 soon as part of my 7-day plan and aim to finish the rest within the coming week.

Git and GitHub

Git is a version control system that helps track code changes and enables collaboration. I installed Git using the commands `sudo dnf update -y` and `sudo dnf install git -y`, then verified the installation with `git --version`.

GitHub is a platform for hosting, managing, and collaborating on code using Git. I'm using GitHub in this project to store my code, track changes, and share my work, making collaboration and version control easier.



My local repository

A Git repository is a storage location for code and project files, managed using Git. It tracks changes, enables collaboration, and allows version control. Repositories can be local or hosted on platforms like GitHub for easy sharing and teamwork.

The "git init" is a command that initializes a new Git repository in a folder, allowing version control and tracking of changes. I ran "git init" in my web app folder to set up a local repository for managing and syncing code with GitHub.

After running git init, the terminal showed a note about the default branch name. A branch in Git is an independent development line that allows multiple code versions to exist separately before merging, enabling better collaboration amongst others.

```
● [ec2-user@ip-172-31-25-59 nextwork-web-project]$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
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>
Initialized empty Git repository in /home/ec2-user/nextwork-web-project/.git/
○ [ec2-user@ip-172-31-25-59 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 where changes are prepared before being committed to the repository. It allows developers to review and organize modifications before finalizing them with a commit.

git commit

The second command I ran was `git commit -m "message"`. Using '-m' means adding a short message describing the changes directly in the command, instead of opening an editor. This helps keep track of modifications in a structured way.

git push

The third command I ran was `git push -u origin main`. Using '-u' means setting the upstream branch, ensuring future pushes and pulls automatically target 'origin main' without needing to specify the branch again. This streamlines workflow efficiency

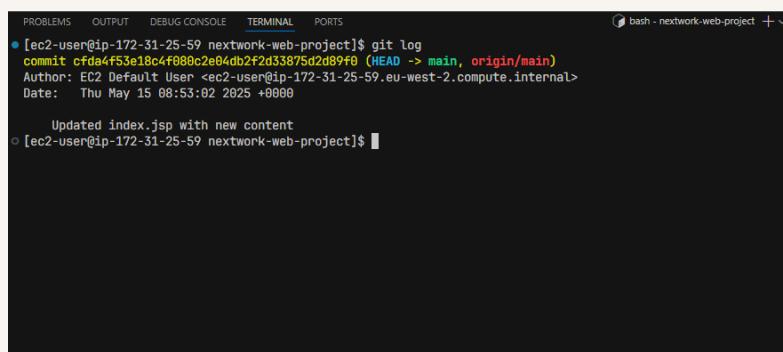
Authentication

When I commit changes to GitHub, Git asks for my credentials because authentication is required to verify the user's identity and grant access to the repository. This ensures that only authorized users can push changes, maintaining security and more.

Local Git identity

Git needs my name and email because they help associate commits with me, ensuring proper attribution and tracking contributions in project history. The email also integrates with GitHub and other remote repositories for collaboration and notification

Running `git log` showed me that the latest commit was made by the EC2 Default User on May 15, 2025, with the message "Updated index.jsp with new content."



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash - nextwork-web-project + 
● [ec2-user@ip-172-31-25-59 nextwork-web-project]$ git log
commit cfd4e53e18c4f080c2e04db2f2d33875d2d89f0 (HEAD -> main, origin/main)
Author: EC2 Default User <ec2-user@ip-172-31-25-59.eu-west-2.compute.internal>
Date:   Thu May 15 08:53:02 2025 +0000

    Updated index.jsp with new content
○ [ec2-user@ip-172-31-25-59 nextwork-web-project]$
```

GitHub tokens

GitHub authentication failed when I entered my password because GitHub no longer accepts password authentication for Git operations. Instead, it requires a personal access token (PAT) or SSH key for security reasons.

A GitHub token is a secure authentication method replacing passwords for Git operations. I'm using one in this project because GitHub no longer supports password authentication, and tokens provide a safer way to access and interact with my repository

I could set up a GitHub token by navigating to GitHub's settings, selecting "Developer settings," then "Personal access tokens." After generating a new token with necessary permissions, I copy it and use it instead of a password for Git authentication

AM

ampahben3@gmail.com

NextWork Student

nextwork.org

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Generated for EC2 Instance Access. This is a part of NextWork's 7 D

What's this token for?

Expiration

7 days (May 22, 2025) ▾

The token will expire on the selected date

Select scopes

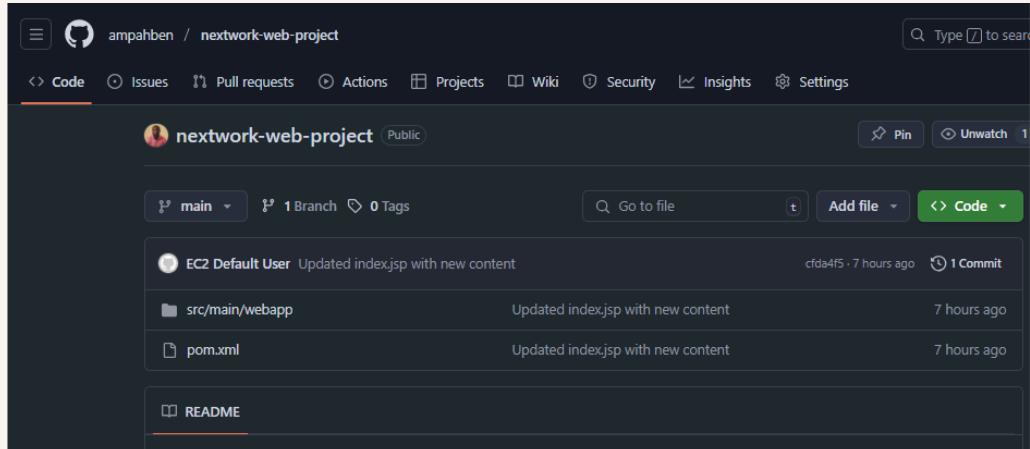
Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> reposstatus	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

Making changes again

I wanted to see Git working in action, so I updated the `index.jsp` file. I couldn't see the changes in my GitHub repo initially because I forgot to push the commit after making local changes.

I finally saw the changes in my GitHub repo after committing the update to `index.jsp` and running `git push` to send the changes from my local machine to the remote repository.





nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

