



NextWork.org

Connect a GitHub Repo with AWS



Emmanuel Enalpe III

```
Complete!
• [ec2-user@ip-172-31-23-119 nextwork-web-project]$ git --version
git version 2.40.1
○ [ec2-user@ip-172-31-23-119 nextwork-web-project]$ █
```



Introducing Today's Project!

What is GitHub?

GitHub is a platform for version control and collaboration, allowing developers to manage and share code. In today's project, I used GitHub to track changes, collaborate with team members, and manage project documentation efficiently.

One thing I didn't expect...

I didn't expect how useful this project would be for collaborating with other developers and engineers. It also tracks code changes easily through logs, making teamwork more efficient.

This project took me...

It took me over an hour to complete the updated project in the DevOps series.



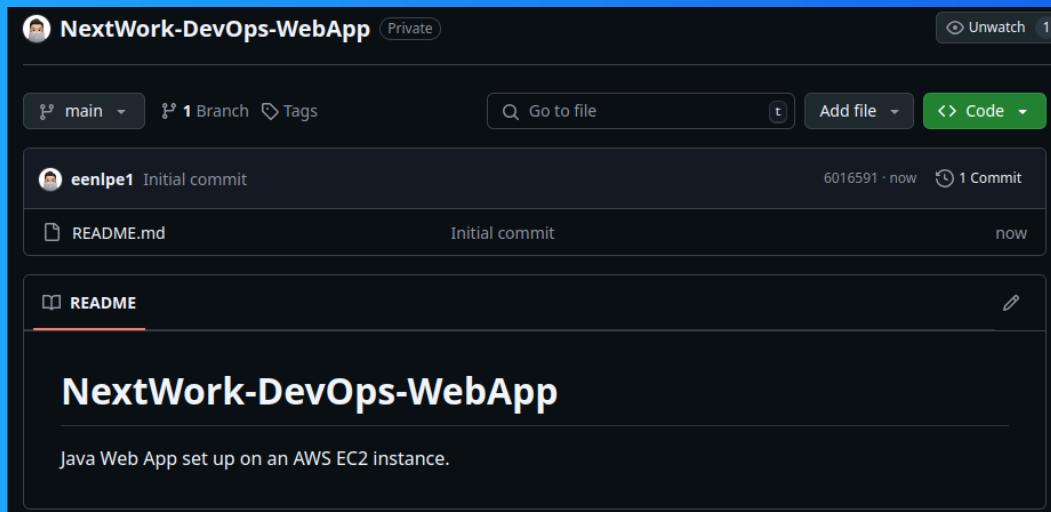
Emmanuel Enalpe III
NextWork Student

NextWork.org

Git and GitHub

Git is a version control system that tracks changes in your local or remote repository. I installed Git using the following commands: sudo dnf update -y and sudo dnf install git -y.

GitHub is a platform for hosting and collaborating on code. I'm using GitHub in this project to manage version control, track changes, and collaborate with others efficiently.





Emmanuel Enalpe III

NextWork Student

NextWork.org

My local repository

A Git repository is a storage space where project files and their version histories are tracked. It allows collaboration by managing changes and ensuring version control, making it easier to revert, compare, and merge code.

`git init` is a command that initializes a new Git repository. I ran `git init` in the project directory to create a repository, allowing me to start tracking changes in the project files.

After running `git init`, the terminal responded with... A branch in Git is a lightweight, movable pointer to a commit. Branches allow you to work on different features or bug fixes independently from the main project.

```
● [ec2-user@ip-172-31-23-119 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-172-31-23-119 nextwork-web-project]$ █
```



Emmanuel Enalpe III

NextWork Student

NextWork.org

To push local changes to GitHub, I ran three commands

git add

The first command I ran was `git add .`, which stages all the files for the remote repository. A staging area is a space where files are prepared before being committed to the repository.

git commit

The second command I ran was `git commit -m "Message of my commit."` Using `-m` means that I can provide a commit message directly in the command, describing the changes made, without opening the default text editor.

git push

The third command I ran was `git push -u origin master`. Using `-u` means setting the upstream tracking reference, allowing future `git push` commands to default to this branch without needing to specify the remote and branch name each time.



Authentication

When I commit changes to GitHub, Git asks for my credentials because I haven't set up credential caching or SSH keys, so it requires authentication for each interaction with the remote repository.

Local Git identity

Git needs my name and email because it associates these details with each commit I make, helping to identify the author and maintain a clear history of contributions in a project. This information is essential for collaboration and accountability.

Running `git log` showed me that there were several recent commits, including bug fixes and feature updates. This helped me understand the project's current status and plan my next steps accordingly.

```
● [ec2-user@ip-172-31-23-119 nextwork-web-project]$ git log
commit 840badf025ff5be13f07395ff1a61536dfeba59d (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-23-119.ap-southeast-1.compute.internal>
Date:  Sun Oct 20 04:55:55 2024 +0000

    Updated index.jsp with new content
○ [ec2-user@ip-172-31-23-119 nextwork-web-project]$ 
```



GitHub tokens

GitHub authentication failed when I entered my password because the password was incorrect or there may be issues with my account settings, such as two-factor authentication being enabled or a problem with the connection.

A GitHub token is a personal access token used to authenticate and authorize access to GitHub's API. I'm using one in this project to securely manage dependencies and automate workflows without exposing my credentials.

I can set up a GitHub token by creating a personal access token in my GitHub account settings. This token will allow me to authenticate and interact with GitHub's API securely.

The screenshot shows the GitHub 'Personal access tokens' section. A new token is being created for 'EC2 Instance Access. NextWork DevOps project'. The token has an expiration of '30 days' and is set to expire on 'Tue, Nov 19 2024'. Under 'Select scopes', the 'repo' scope is selected, granting full control of private repositories, access to commit status, deployment status, public repositories, repository invitations, and security events. Other scopes like 'workflow', 'write:packages', and 'read:packages' are also listed but not selected.

Scope	Description
<input checked="" type="checkbox"/> repo	Full control of private repositories Access commit status Access deployment status Access public repositories Access repository invitations 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 in action, but I couldn't see the changes in my GitHub repo initially because I forgot to commit my changes locally before pushing them to the remote repository. Once I did that, everything updated as expected.

I finally saw the changes in my GitHub repo after pushing the latest updates. It's great to see my progress reflected online! Now I can continue working on the project with confidence and share my work with others.

```
v 5 src/main/webapp/index.jsp
 0 -2,10 +2,13 00
 2 2
 3 3 <body>
 4 4
 5 - <h2>Hello <YOUR Emmanuel>!</h2>
 5 + <h2>Hello World!<YOUR Emmanuel>!</h2>
 6 6
 7 7 <p>This is my NextWork web application working!</p>
 8 8
 9 + <p>If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :o</p>
10 +
11 +
12 </body>
13
14 </html>
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

