



nextwork.org

Connect a GitHub Repo with AWS



Sanjana Tripathy



Introducing Today's Project!

"In this project, I'll demonstrate how to set up Git and GitHub for a web application. My goal is to learn version control basics, connect a project to a remote repo, and track code changes effectively for collaboration."

Key tools and concepts

The services I used were EC2. Key concepts I learnt include understanding Git and GitHub, setting up and using a personal access token, connecting VSCode with Git, and managing version control through staging, committing, and pushing changes.

Project reflection

This project took me approximately 2 hours. The most challenging part was minimal, and it was most rewarding to see Git and GitHub in action, successfully tracking changes, committing updates, and pushing them from my EC2 instance to the cloud repo.

I did this project because I wanted to gain hands-on experience with Git and GitHub, understand version control workflows, practice connecting my local EC2 environment with a cloud repository, and learn how to manage and track code changes effectively.



Sanjana Tripathy
NextWork Student

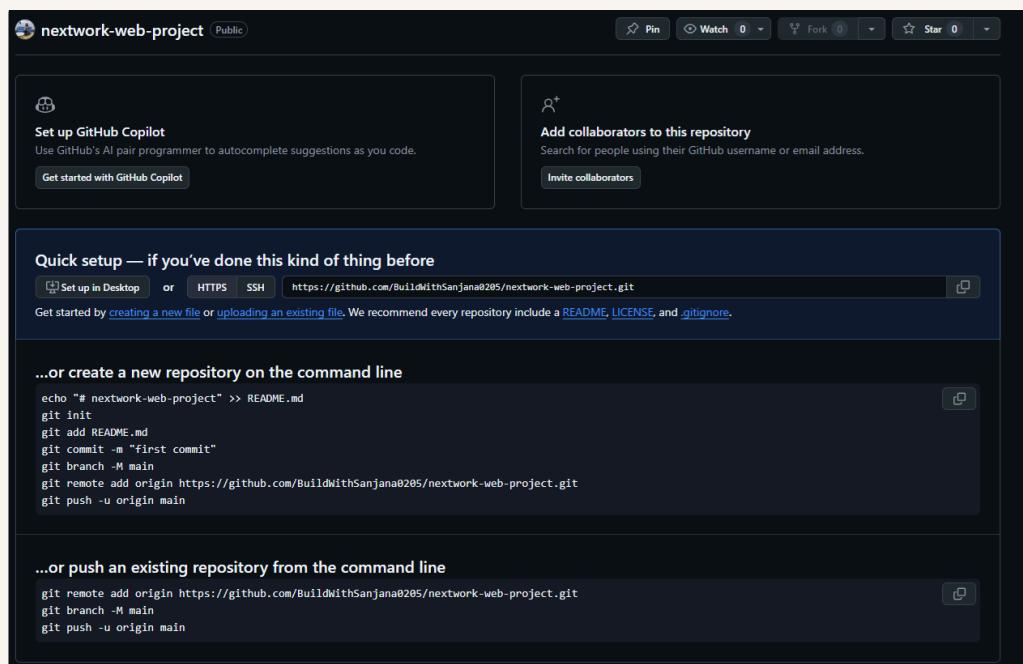
nextwork.org

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 today to continue enhancing my skills in automating deployments, managing code changes, and integrating development workflow.

Git and GitHub

Git is a version control system that tracks changes and supports collaboration. I installed it on my EC2 instance by opening the terminal and running sudo dnf update -y and sudo dnf install git -y.

GitHub is a cloud platform that stores Git repositories, making version tracking and collaboration easy. I'm using GitHub in this project to sync and share my code beyond the EC2 instance, since EC2 alone doesn't provide version control.



A circular profile picture of a young woman with dark hair, wearing a pink patterned top, sitting on a blue couch in an office setting.

Sanjana Tripathy
NextWork Student

nextwork.org

My local repository

A Git repository is a storage space that holds your project's files along with their entire change history. I'm using a Git repository in this project to track modifications, manage versions, and collaborate efficiently across different environments.

Git init is a command that creates a new Git repository in the current folder, enabling version control for its files. I ran git init in my web app project directory because I needed to start tracking changes and connect it with my GitHub repo.

A branch in Git is a parallel version of a project that lets you test changes without affecting the main code. After running git init, the terminal showed a note about the default branch being named master and suggested using names lik



Sanjana Tripathy
NextWork Student

nextwork.org

```
● [ec2-user@ip-172-31-37-217 ~]$ ls
  nextwork-web-project
● [ec2-user@ip-172-31-37-217 ~]$ cd nextwork-web-project
● [ec2-user@ip-172-31-37-217 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>
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-172-31-37-217 nextwork-web-project]$ █
```

A circular profile picture of a young woman with dark hair, wearing a pink top, sitting on a blue couch in an office setting.

Sanjana Tripathy
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 modified files in my project. A staging area is a place where Git collects these changes for review because it allows me to verify updates before committing them permanently.

git commit

The second command I ran was `git commit -m "Updated index.jsp with new content"`. Using `-m` means I can add a descriptive message for the commit, and committing saves all staged changes as a new snapshot in the project's version history for easy tracking

git push

The third command I ran was `git push -u origin master`. Using `-u` means I set an upstream for my local branch, so Git remembers to push to the master branch of my GitHub repo by default, uploading all committed changes to the remote repository.



Authentication

When I commit changes to GitHub, Git asks for my credentials because it needs to verify my identity, ensuring that only authorized users can push updates and maintain the security and integrity of the remote repository.

Local Git identity

Git needs my name and email because it uses this information to identify the author of each commit, ensuring that project history accurately records who made changes for accountability and collaboration.

Running `git log` showed me a detailed history of all commits in my repository, including commit messages, authors, and timestamps, because Git tracks every change to provide a complete version history and make it easy to review project evolution.



Sanjana Tripathy
NextWork Student

nextwork.org

```
[master] (master) - set up to track origin/master
● [ec2-user@ip-172-31-37-217 nextwork-web-project]$ git log
commit 6e0a030277f0e0a09e7930836670bd3 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-37-217.ap-south-1.compute.internal>
Date: Thu Sep 4 12:40:21 2025 +0000

    Updated index.jsp with new content

● [ec2-user@ip-172-31-37-217 nextwork-web-project]$ git config --global user.name "Sanjana"
git config --global user.email tripathysanjana0@gmail.com
● [ec2-user@ip-172-31-37-217 nextwork-web-project]$ git log
commit 6e0a030277f0e0a09e7930836670bd3 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-37-217.ap-south-1.compute.internal>
Date: Thu Sep 4 12:40:21 2025 +0000

    Updated index.jsp with new content
```



GitHub tokens

GitHub authentication failed when I entered my password because GitHub no longer accepts account passwords for Git operations and requires a personal access token to securely verify and authorize remote repository access.

A GitHub token is a secure, unique code that grants access to your repositories without using a password. I'm using one in this project because it enables authenticated Git operations from my local repo while keeping my account secure.

I could set up a GitHub token by going to my GitHub Settings, navigating to Developer settings → Personal access tokens → Tokens (classic), generating a new token with a descriptive note, selecting the repo scope, and copying the generated token.



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 (Sep 11, 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 checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

Making changes again

I wanted to see Git working in action, so I edited the index.jsp file through VSCode. I couldn't see the changes in my GitHub repo initially because edits in my local environment only affect the local repository until I commit and push them to the repo.

I finally saw the changes in my GitHub repo after staging the edits with git add ., committing them with git commit -m "Add new line to index.jsp", and pushing them using git push because only committed and pushed changes update the remote repository.

The screenshot shows a GitHub code editor interface for the file `index.jsp`. The title bar indicates the path: `nextwork-web-project / src / main / webapp / index.jsp`. The commit history shows a single commit from `BuildWithSanjana0205` with the message `Add new line to index.jsp`. The code editor displays the following content:

```
1 <html>
2
3 <body>
4
5 <h2>Hello sanjanaa!</h2>
6
7 <p>This is my NextWork web application working!</p>
8 <p>If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :o</p>
9
10
11
12 </body>
13
14 </html>
```



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

