



nextwork.org

Connect a GitHub Repo with AWS



Gursimran SINGH

```
Installed:  
git-2.47.1-1.amzn2023.0.3.x86_64      git-core-2.4  
perl-Git-2.47.1-1.amzn2023.0.3.noarch  perl-TermRea  
  
Complete!  
● [ec2-user@ip-172-31-27-104 ~]$ git --version  
  
git version 2.47.1  
● [ec2-user@ip-172-31-27-104 ~]$
```



Introducing Today's Project!

Today, I'll be setting up Git and GitHub to manage my web app project. I'll connect my local project to a remote repo, track changes using Git, update my code, and see it reflect in GitHub. I'll also add a clean README to explain the project.

Key tools and concepts

Services I used were AWS EC2 and GitHub. Key concepts I learnt include launching and connecting to a server, setting up a basic Java web app, using Git for version control, and pushing code changes to a remote repository.

Project reflection

This project took me approximately 1–2 hours to complete. The most challenging part was setting up the GitHub connection. It was most rewarding to see the code changes reflect in the repo after a successful push.

I did this project today to build hands-on experience with deploying and managing code using Git and AWS. It met my goals by helping me understand real-world DevOps practices and bringing me one step closer to becoming a better cloud engineer.



Gursimran SINGH
NextWork Student

nextwork.org

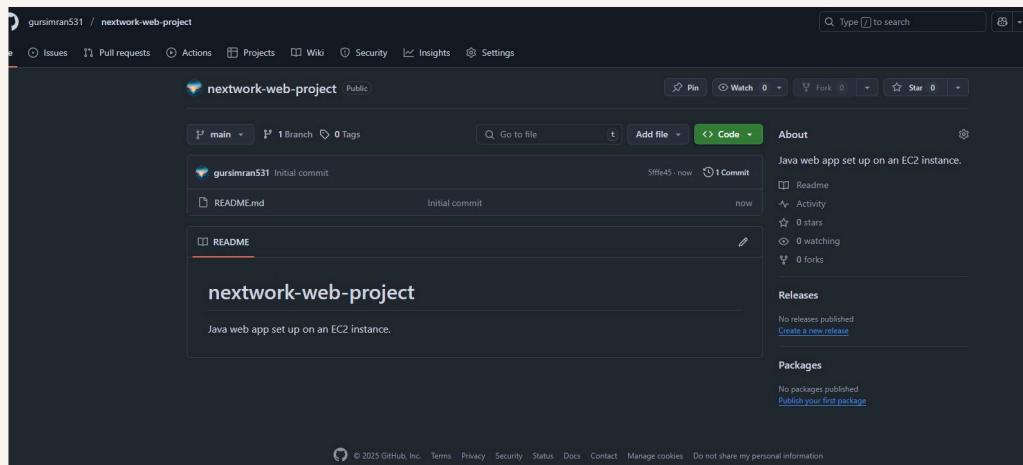
This project is part two of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project tomorrow to keep up the momentum and continue sharpening my AWS and DevOps skills.



Git and GitHub

Git is a version control system that helps track changes in code, collaborate with others, and manage project history. I installed Git using the commands: sudo yum update -y sudo yum install git -y

GitHub is a cloud-based platform for hosting Git repositories. I'm using GitHub in this project to store my code, track changes, and enable version control, so I can collaborate easily and keep my work backed up and organized.





My local repository

A Git repository is a storage space where your project's files and their revision history are tracked. It allows you to manage changes, collaborate with others, and roll back to previous versions if needed.

`git init` is a command that initializes a new Git repository in your current directory. I ran `git init` in my project folder to start tracking changes in my web app locally using Git.

After running `git init`, the response from the terminal was a message saying an empty Git repository was initialized and a default branch called `master` was created. A branch in Git is like a separate workspace where you can make changes without af



Gursimran SINGH

NextWork Student

nextwork.org

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

● [ec2-user@ip-172-31-27-104 ~]$ ls
apache-maven-3.5.2-bin.tar.gz  nextwork-web-project
● [ec2-user@ip-172-31-27-104 ~]$ cd nextwork-web-project/
● [ec2-user@ip-172-31-27-104 nextwork-web-project]$ pwd
/home/ec2-user/nextwork-web-project
● [ec2-user@ip-172-31-27-104 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-172-31-27-104 nextwork-web-project]$ 
```



To push local changes to GitHub, I ran three commands

git add

The first command I ran was `git add .` — it adds all the changed files to the staging area. A staging area is where Git tracks files you want to commit, letting you review and organize changes before saving them permanently in your repository.

git commit

The second command I ran was `git commit -m "Updated index.jsp with new content"`. Using `-m` means I'm adding a message to describe what changes I made. This helps keep track of updates and makes the project history easy to understand.

git push

The third command I ran was `git push -u origin master`. Using `-u` means I'm setting the upstream branch, so future pushes can be done with just `git push`. This sends my committed code to the `master` branch of my GitHub repo named `origin`.



Authentication

When I commit changes to GitHub, Git asks for my credentials because it needs to verify that I have permission to push changes to the repository. This ensures only authorized users can update the code, keeping the project secure.

Local Git identity

Git needs my name and email because it uses them to track who makes each change in a project. This info appears in the commit history and helps collaborators understand who contributed what. It's essential for accountability and collaboration.

Running `git log` showed me that my commits were successfully recorded. It displayed details like the commit ID, author name and email, date, and the commit message — helping me track changes made to the project over time.



Gursimran SINGH

NextWork Student

nextwork.org

```
● [ec2-user@ip-172-31-27-104 nextwork-web-project]$ git log
commit 17f831b19682f0006201cde617fc0c3f3795e9db (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-27-104.ec2.internal>
Date:   Sat Jul 5 13:28:20 2025 +0000

    Updated index.jsp with new content
○ [ec2-user@ip-172-31-27-104 nextwork-web-project]$ ]
```



GitHub tokens

GitHub authentication failed when I entered my password because GitHub no longer accepts account passwords for Git operations. Instead, it requires a **personal access token (PAT)** for secure authentication when pushing code via HTTPS.

A GitHub token is a secure, auto-generated key that replaces your password when performing Git operations. I'm using one in this project because GitHub no longer allows password authentication for pushing code, making tokens a safer alternative.

I could set up a GitHub token by going to my GitHub account settings, selecting "Developer settings" > "Personal access tokens," generating a new token with the required scopes, and using it in place of my password when pushing code to GitHub.



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 (Jul 12, 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
<input type="checkbox"/> workflow	Update GitHub Action workflows

Making changes again

I wanted to see Git working in action, so I updated the `index.jsp` file in the `nextwork-web-project`. I couldn't see the changes in my GitHub repo initially because I forgot to run `git push` after committing.

I finally saw the changes in my GitHub repo after running `git add`, `git commit -m`, and then `git push` to upload the updates from my local project to the remote repository.

A screenshot of a GitHub code editor interface. The path shown is `nextwork-web-project / src / main / webapp / index.jsp`. The code editor shows the following content:

```
1 <html>
2
3 <body>
4
5 <h2>Hello {YOUR NAME}</h2>
6
7 <p>This is my NextWork web application working!</p>
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 </body>
12
13 </html>
```

The code editor has tabs for "Code" and "Blame". It also includes a note: "Code 55% faster with GitHub Copilot". There are buttons for "Raw", "Copy", "Edit", and "Diff". The commit message "SinghOps Add new line to index.jsp" is visible at the top right, along with the timestamp "45885e7 · 2 minutes ago".



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

