

Connect a GitHub Repo with AWS



Seth Sekyere

The screenshot shows a terminal window and a code editor side-by-side.

Terminal:

```
HostName: ec2-3-21-27-68.us-east-2.compute.amazonaws.com
IdentityFile: /Users/sekyer/OneDrive/Desktop/DevOps/Devopskey.pem
User: ec2-user

2
3
4
5
6
```

Code Editor:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Verifying : git-2.50.1-1.amzn2023.0.1.x86_64 1/8
Verifying : git-core-2.50.1-1.amzn2023.0.1.x86_64 2/8
Verifying : perl-Git-2.50.1-1.amzn2023.0.1.noarch 3/8
Verifying : perl-Error-1.0.17029-5.amzn2023.0.2.noarch 4/8
Verifying : perl-File-Find-1.37-477.amzn2023.0.7.noarch 5/8
Verifying : perl-Git-2.50.1-1.amzn2023.0.1.noarch 6/8
Verifying : perl-JSON-PP-2.34-1.amzn2023.0.1.x86_64 7/8
Verifying : perl-11b-0.65-477.amzn2023.0.7.x86_64 8/8

Installed:
git-2.50.1-1.amzn2023.0.1.x86_64          git-core-2.50.1-1.amzn2023.0.1.x86_64          git.core-doc-2.50.1-1.amzn2023.0.1.noarch          perl.core-doc-2.50.1-1.amzn2023.0.1.noarch
perl-File-Find-1.37-477.amzn2023.0.7.noarch      perl-Git-2.50.1-1.amzn2023.0.1.noarch      perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64      perl-Error-1.0.17029-5.amzn2023.0.2.noarch
perl-11b-0.65-477.amzn2023.0.7.x86_64          perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64          perl-File-Find-1.37-477.amzn2023.0.7.noarch      perl-11b-0.65-477.amzn2023.0.7.x86_64

Completed:
[ec2-user@ip-172-31-20-0 ~]$ git --version
git version 2.50.1
[ec2-user@ip-172-31-20-0 ~]$
```



Seth Sekyere

Introducing Today's Project!

In this project, I will demonstrate how to use Git and GitHub to manage my web application code. I'm doing this project to learn version control, collaboration, and improve my DevOps skills.

Key tools and concepts

Services I used were Git, GitHub, EC2, and VSCode. Key concepts I learnt include version control, staging and committing changes, pushing to remote repos, and the difference between local and cloud repositories.

Project reflection

This project took me approximately 20 minutes. The most challenging part was understanding how local and remote repositories interact. It was most rewarding to see my changes reflected on GitHub and know my version control is working!

I did this project because I wanted hands-on experience with Git and GitHub, especially understanding how version control works in real-world development environments. It's an essential skill for collaboration and deploying code confidently.



Seth Sekyere

This project is part two of a DevOps series building a CI/CD pipeline! I'll be working on the next project soon to continue improving my skills and move closer to automating deployments end-to-end.

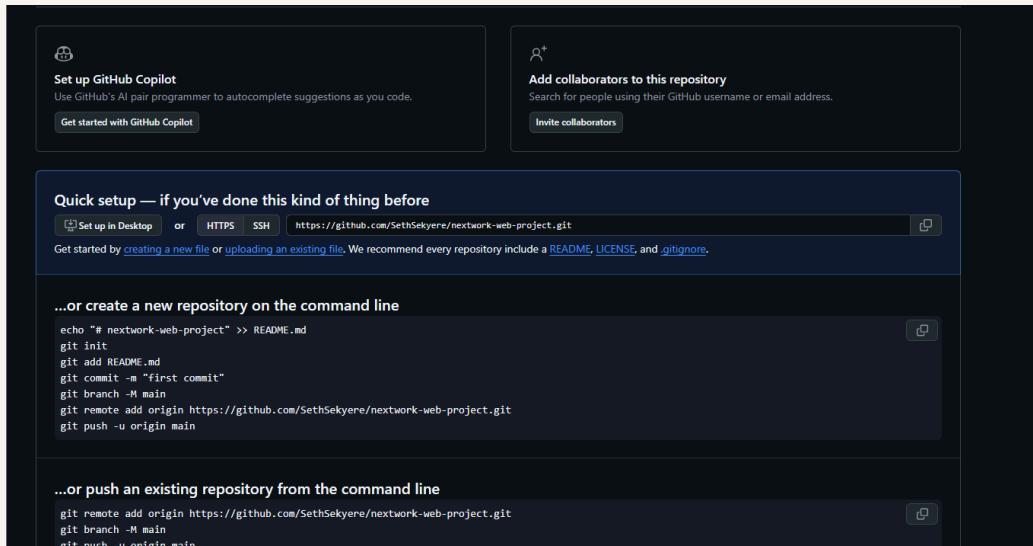


Seth Sekyere

Git and GitHub

Git is a version control system that tracks changes in code and helps with collaboration. I installed Git using the commands `sudo dnf update -y` and `sudo dnf install git -y` on my EC2 instance.

GitHub is a cloud platform that stores code and tracks changes using Git. I'm using GitHub in this project to manage version control, back up my code, and collaborate more easily.





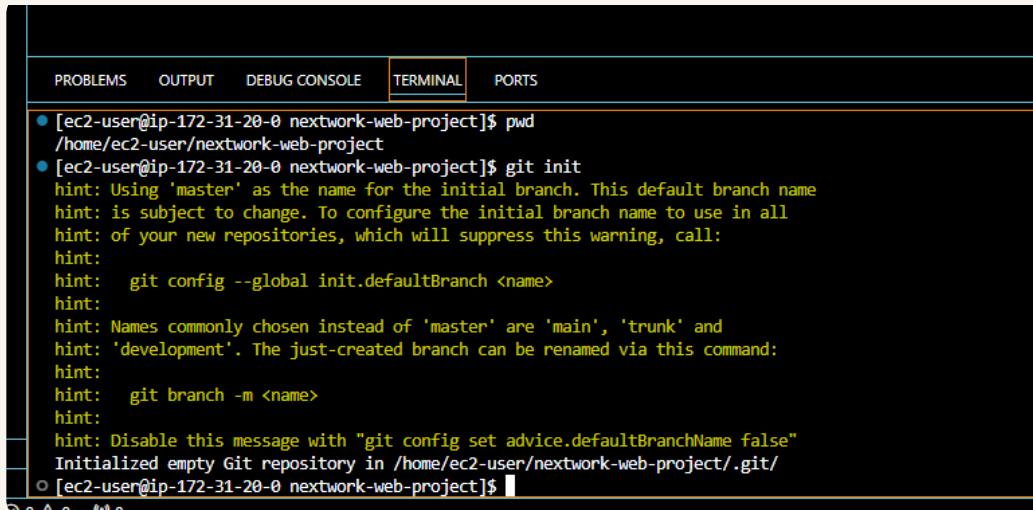
Seth Sekyere

My local repository

A Git repository is a folder that tracks and stores all versions of your project files using Git, enabling you to manage changes, collaborate, and revert to previous versions when needed.

Git init is a command that creates a local Git repository to start tracking changes. I ran git init in my project folder to enable version control for my web app code.

A branch in Git is a parallel version of your project to develop features safely. After running git init, the terminal showed a message about the default branch name (like master or main) you can use to organize your work.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● [ec2-user@ip-172-31-20-0 nextwork-web-project]$ pwd
/home/ec2-user/nextwork-web-project
● [ec2-user@ip-172-31-20-0 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-172-31-20-0 nextwork-web-project]$ █
```



Seth Sekyere

To push local changes to GitHub, I ran three commands

git add

The first command I ran was `git init`, which initializes a new Git repository in the current folder. A staging area is where Git stores changes before committing them, letting you review edits before saving them.

git commit

The second command I ran was `git commit -m "Updated index.jsp with new content"`. Using `-m` means you're adding a message to describe what changes were made in this commit, helping track your project's history clearly.

git push

The third command I ran was `git push -u origin master`. Using `-u` means you're setting the upstream branch so that future pushes can be done with just `git push`, linking your local branch to the remote one on GitHub.



Seth Sekyere

Authentication

When I commit changes to GitHub, Git asks for my credentials because it needs to verify my identity and ensure I have permission to push changes to the remote repository.

Local Git identity

Git needs my name and email to record who made each change in the project. This helps track authorship, makes collaboration clear, and shows accurate commit history for better project management.

Running `git log` showed me a list of all past commits, including each commit's ID, author name, email, date, and the commit message describing what was changed.

```
Username for 'https://github.com': SethSekyere
Password for 'https://SethSekyere@github.com':
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 1.07 KiB | 549.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SethSekyere/nextwork-web-project.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
[ec2-user@ip-172-31-20-0 nextwork-web-project]$ git log
commit f895d459150d475a04b86548e7241ed7bfcc8e668 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-20-0.us-east-2.compute.internal>
Date:  Mon Aug 4 21:07:41 2025 +0000

    Updated index.jsp with new content
[ec2-user@ip-172-31-20-0 nextwork-web-project]$
```



Seth Sekyere

GitHub tokens

GitHub authentication failed when I entered my password because GitHub no longer supports password authentication over HTTPS and now requires a personal access token instead.

A GitHub token is a unique, secure string used instead of a password for authentication. I'm using one in this project because GitHub no longer accepts passwords for security over HTTPS, so the token lets me safely push code to my repository.

I could set up a GitHub token by going to GitHub Settings > Developer settings > Personal access tokens, then generating a new token with the repo scope, setting an expiration, and copying the token to use for authentication.



Seth Sekyere

What's this token for?

Expiration

30 days (Sep 03, 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
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects

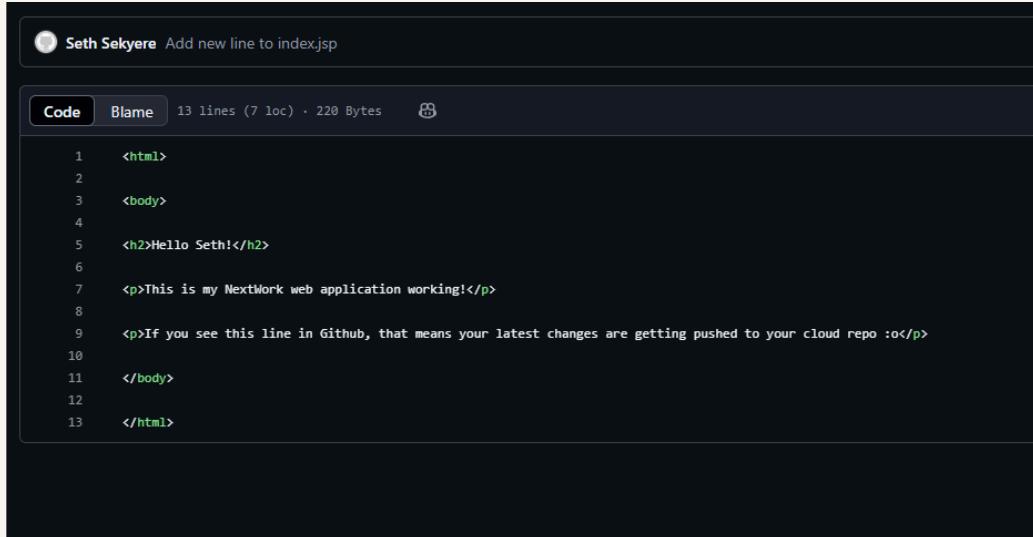


Seth Sekyere

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 saving in VSCode only updates the local repo, not the cloud repo.

I finally saw the changes in my GitHub repo after staging with `git add .`, committing with `git commit -m`, and pushing with `git push`. That's when the updated `index.jsp` appeared in the cloud repo.



The screenshot shows a GitHub code editor interface. At the top, it says "Seth Sekyere Add new line to index.jsp". Below that, there are tabs for "Code" (which is selected) and "Blame", followed by statistics: "13 lines (7 loc) · 220 Bytes". A copy icon is also present. The code itself is a simple HTML file:

```
1 <html>
2
3 <body>
4
5 <h2>Hello Seth!</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>
```