



nextwork.org

Connect a GitHub Repo with AWS

J

Joseph Cole-Showers

```
Installed:
git-2.47.1-1.amzn2023.0.2.x86_64          git-core-2.47.1-1.amzn2023.0.2.x86_64
git-core-doc-2.47.1-1.amzn2023.0.2.noarch   perl-Error-1:0.17029-5.amzn2023.0.2.noarch
perl-File-Find-1.37-477.amzn2023.0.6.noarch perl-Git-2.47.1-1.amzn2023.0.2.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64  perl-lib-0.65-477.amzn2023.0.6.x86_64

Complete!
● [ec2-user@ip-172-31-21-33 nextwork-web-project]$ git --version
git version 2.47.1
```



Introducing Today's Project!

This project is day 2 of a 7 days DevOps Challenge. In this project, I will setup a git repo for the web apps code. By the end of this project, the code for the web app will be stored securely in github.

Key tools and concepts

Services I used were Github, Amazon EC2 (dev instance), key pairs, VS Code. Key concepts I learnt include setting up a git repo, the difference between git and github, and git commands for staging and pushing changes.

Project reflection

This project took me approximately 1 hr. It was most rewarding to connect Amazon EC2 instance with GitHub

I did this project to understand how to connect GitHub to AWS.

This project is part two of a series of DevOps projects where I'm building a CI/CD pipeline.



Git and GitHub

Git is like a time machine and filing system for your code. It is a version control system used to track changes that has been made to a codebase. I installed Git using the commands 'sudo dnf update -y' and 'sudo dnf install git -y'.

GitHub is a place for engineers to store and share their code and projects online. I'm using GitHub in this project to store the web apps code.

The screenshot shows a GitHub repository page for 'nextwork-web-project'. The repository is public and owned by 'uncleejay'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. Below the navigation bar, there are sections for 'Set up GitHub Copilot' and 'Add collaborators to this repository'. A large callout at the bottom provides quick setup instructions, suggesting to 'Set up in Desktop' or use 'HTTPS' or 'SSH' to clone the repository. The URL 'https://github.com/uncleejay/nextwork-web-project.git' is also displayed. A note at the bottom encourages users to create a new file or upload an existing one, and recommends including a README, LICENSE, and .gitignore.

My local repository

A Git repository is like an online folder you can use to store your web apps code or files. Think of it as the go to place for all the code and update including changes related to a specific project.

git init is a command that sets up the directory as a local Git repository which means changes are now tracked for version control.

After running git init, the response from the terminal was that I initialized git, and by default I'm using the main branch. A branch in Git is like a version of your code.

```
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/
```

To push local changes to GitHub, I ran three commands

git add

The first command I ran was 'git add .' which adds the changes to a staging area. A staging area is like a place to review all changes to the code, so you can decided what changes you want to save.

git commit

The second command I ran was 'git commit -m'. Which is the command for saving the changes in the staging area. Using '-m' means I'm also leaving a message for that commit.

git push

The third command I ran was 'git push -u origin master'. This command pushes the code changes I save to the remote origin. Using '-u' makes the remote origin the default.

Authentication

When I commit changes to GitHub, Git asks for my credentials because it needs to authenticate me before letting me make code changes to the gitub repository. It needs to do this to know I have the right to change the code.

Local Git identity

Git needs my name and email because it is a version control system. Which means it is used for tracking changes to code and who made what change to a piece of code. To really identify who made what change, git will need to know the name/email.

Running git log showed me that by default git is saving my code changes to a username called 'EC2 default user' instead of my actual name and details.

```
commit f919348784bcf26ecc0dd472a387ecc0
Author: EC2 Default User <ec2-user@ip-172-31-10-2>
Date:   Sat Apr 19 14:49:15 2025 +0000
```

GitHub tokens

GitHub authentication failed when I entered my password because password authentication was already removed in 2021. There are too many security risks associated with entering a password over the terminal to Github.

A GitHub token is like a temporary password that grants access to your GitHub account. I'm using one in this project because it lets me safely authenticate to the GitHub repo while in the EC2 instance.

I could set up a GitHub token by visiting the developer settings in GitHub and setting up the expiry date.

The screenshot shows the 'Select scopes' section of the GitHub token creation interface. It includes a descriptive text about scopes, a list of available scopes with checkboxes, and a note about reading more about OAuth scopes.

Select scopes

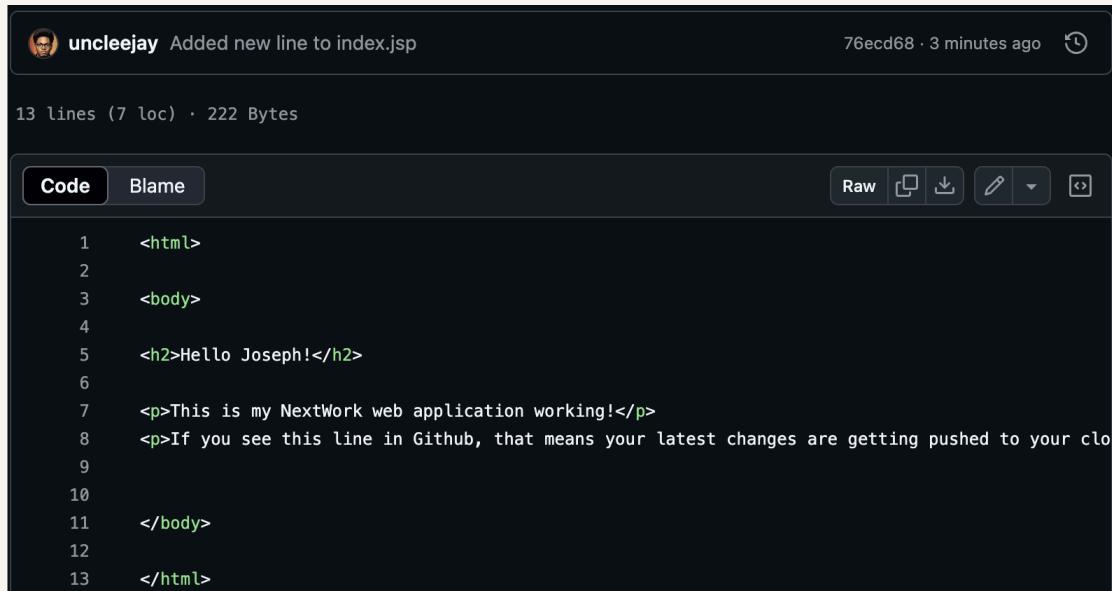
Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<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 made another change to the index.jsp file. I couldn't see the changes in my GitHub repo initially because I had not committed or push the changes.

I finally saw the changes in my GitHub repo after running the commands ' git add ., git commit -m and git push'.



A screenshot of a GitHub commit interface. The commit message is "uncleejay Added new line to index.jsp" with a timestamp of "76ecd68 · 3 minutes ago". It shows 13 lines (7 loc) of code with 222 Bytes. The code editor displays the following HTML:

```
1 <html>
2
3 <body>
4
5 <h2>Hello Joseph!</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 clo
9
10
11 </body>
12
13 </html>
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

