

nextwork.org

Connect a GitHub Repo with AWS



Saish Nar

```
Complete!
• [ec2-user@ip-172-31-15-53 nextwork-web-project]$ git --version
git version 2.47.1
○ [ec2-user@ip-172-31-15-53 nextwork-web-project]$ 
```



Introducing Today's Project!

Today I am here to set up a Git repository for my web app's code. This is project TWO in my 7 day devops challenge. By the end of this project, the code that I write for my Java web app will be stored securely in GitHub.

Key tools and concepts

Services I used were GitHub, Amazon EC2 (development instance), key pairs and VS Code. Key concepts I learnt include setting up a Git repository, the difference between Git and GitHub, and the commands for staging, saving and pushing changes to my code

Project reflection

This project took me approximately 2 hours. The most challenging part was to learn different Git commands. It was most rewarding to set up a README that ties the entire repository together, and seeing my code pop up in my GitHub repository.

I did this project today to learn more about Git and GitHub - AND because it's DAY TWO of my 7 Day DevOps challenge! Excited to now have a repository that stores my web app's source code



Saish Nar
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 in next 24 hours.



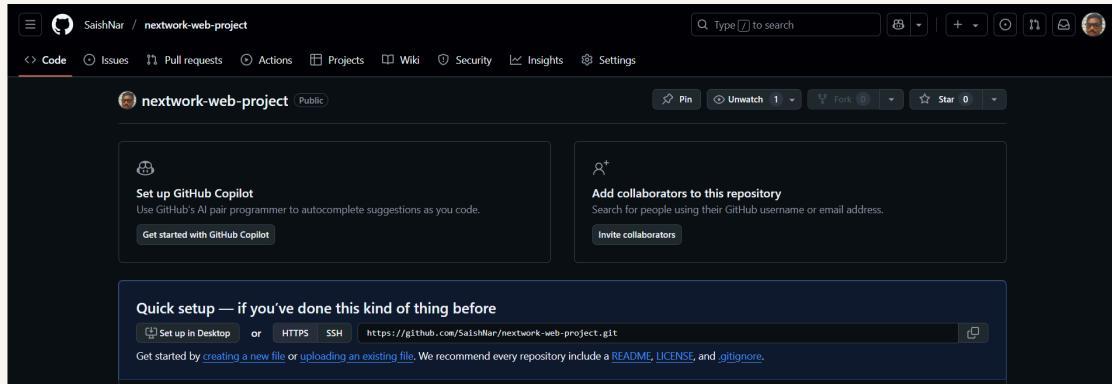
Saish Nar
NextWork Student

NextWork.org

Git and GitHub

Git is often called a version control system since it tracks your changes by taking snapshots of what your files look like at specific moments. I installed Git using the commands: sudo dnf update -y and sudo dnf install git -y.

GitHub is a platform that lets me store and share and collaborate on my code! It's called GitHub because it uses Git as the tool for version control (one of its main features). I'm using GitHub in this project to store my web app's code.





My local repository

A Git repository is like a online folder that you can use to store your web app's code and all the versions of that code. Think of it as the go-to place for all the code and updates and changes related to a specific project.

git init is a command that initializes git in my local repository. I ran git init in my web app project folder, which tells my terminal that I want to start tracking changes locally.

After running git init, the response from the terminal was that I initialized Git, and by default I am using the 'main' branch. A branch in Git is like version of my code, I make changes to my code over a branch and merge those changes to main branch

```
[ec2-user@ip-172-31-15-53 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-172-31-15-53 nextwork-web-project]$ ]
```



To push local changes to GitHub, I ran three commands

git add

The first command I ran was `git add .`, which adds my changes to a staging area. A staging area in Git is like the place to review all changes made to the code, so I can decide what changes I'd like to save in a commit.

git commit

The second command I ran was `git commit -m "Updated index.jsp with new content"`, which is the command for saving the changes in my staging area. Using '`-m`' means I am also leaving a message for that commit e.g. "made changes to `index.jsp`".

git push

The third command I ran was '`git push -u origin master`' This command pushes the code changes I saved to my remote origin. Using '`-u`' means is a special flag that makes my remote origin the default UPSTREAM repository.



Authentication

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

Local Git identity

Git needs my name and email because it's a version control system, which means it's used for tracking WHO made WHAT change to a piece of code. To really identify the people that made x changes, Git will need to know your name and 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.

```
● [ec2-user@ip-172-31-15-53 nextwork-web-project]$ git log
commit 58ac2ccb02435d2fff6201d25dea0efb58b644aa (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-15-53.ap-south-1.compute.internal>
Date:   Fri Apr 4 07:19:27 2025 +0000

    Updated index.jsp with new content
○ [ec2-user@ip-172-31-15-53 nextwork-web-project]$ []
```



GitHub tokens

GitHub authentication failed when I entered my password because password authentication support was already removed in 2021. There are too many security risks associated with entering a password over the terminal to GitHub so more secure ways are now there.

A GitHub token is like a temporary password that grants access to my GitHub account. I am using one in this project because it lets me safely authenticate to my GitHub repository while in my EC2 instance.

I could set up a GitHub token by visiting the Developer Settings in GitHub, and I set one up that expires in 7 days and only allows permissions to repositories.



Saish Nar
NextWork Student

NextWork.org

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. We created this token on Day T.

What's this token for?

Expiration

7 days (Apr 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 made another change to my index.jsp file. I couldn't see the changes in my GitHub repo initially because I hadn't added, committed or pushed those changes.

I finally saw the changes in my GitHub repo after running the same three commands, and then refreshing index.jsp in my GitHub repository.

The screenshot shows a GitHub repository page for 'SaishNar / nextwork-web-project'. The 'Code' tab is selected, showing the file 'index.jsp' under the path 'nextwork-web-project / src / main / webapp'. The code editor displays the following JavaServer Pages (JSP) code:

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



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

