



NextWork.org

Connect a GitHub Repo with AWS



tusharyadav796270@gmail.com

```
(7/8): git-core-doc-2.48.1-1.amzn2023.0.3.noarch          12 MB/s | 2.6 MB  00:00
(8/8): git-core-2.48.1-1.amzn2023.0.3.x86_64.rpm        14 MB/s | 4.3 MB  00:00
Total                                         21 MB/s | 7.1 MB  00:00

Preparing :
Installing : git-core-2.48.1-1.amzn2023.0.3.x86_64          1/1
Installing : git-core-doc-2.48.1-1.amzn2023.0.3.noarch      2/8
Installing : perl-lib-0.65+477.amzn2023.0.6.x86_64         3/8
Installing : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64    4/8
Installing : perl-File-Find-1.37+477.amzn2023.0.6.noarch     5/8
Installing : perl-Error-1.0.17029-5.amzn2023.0.2.noarch       6/8
Installing : perl-Git-2.48.1-1.amzn2023.0.3.noarch          7/8
Installing : git-2.48.1-1.amzn2023.0.3.x86_64             8/8

Running scriptlet: git-2.48.1-1.amzn2023.0.3.x86_64          8/8
Verifying   : git-2.48.1-1.amzn2023.0.3.x86_64             1/8
Verifying   : git-core-2.48.1-1.amzn2023.0.3.x86_64          2/8
Verifying   : git-core-doc-2.48.1-1.amzn2023.0.3.noarch      3/8
Verifying   : perl-Error-1.0.17029-5.amzn2023.0.2.noarch     4/8
Verifying   : perl-File-Find-1.37+477.amzn2023.0.6.noarch     5/8
Verifying   : perl-Git-2.48.1-1.amzn2023.0.3.noarch          6/8
Verifying   : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64    7/8
Verifying   : perl-lib-0.65+477.amzn2023.0.6.x86_64          8/8

Installed:
git-2.48.1-1.amzn2023.0.3.x86_64          git-core-2.48.1-1.amzn2023.0.3.x86_64
git-core-doc-2.48.1-1.amzn2023.0.3.noarch  perl-Error-1.0.17029-5.amzn2023.0.2.noarch
perl-File-Find-1.37+477.amzn2023.0.6.noarch perl-Git-2.48.1-1.amzn2023.0.3.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64  perl-lib-0.65+477.amzn2023.0.6.x86_64

Complete!
```

Introducing Today's Project!

What is GitHub?

GitHub is a version control platform for managing and sharing code. In today's project, I used it to store my code and track changes, enabling automated deployments to AWS.

One thing I didn't expect...

One thing I didn't expect in this project was the need to use a personal access token for GitHub authentication instead of a password, which added an extra step for setup.

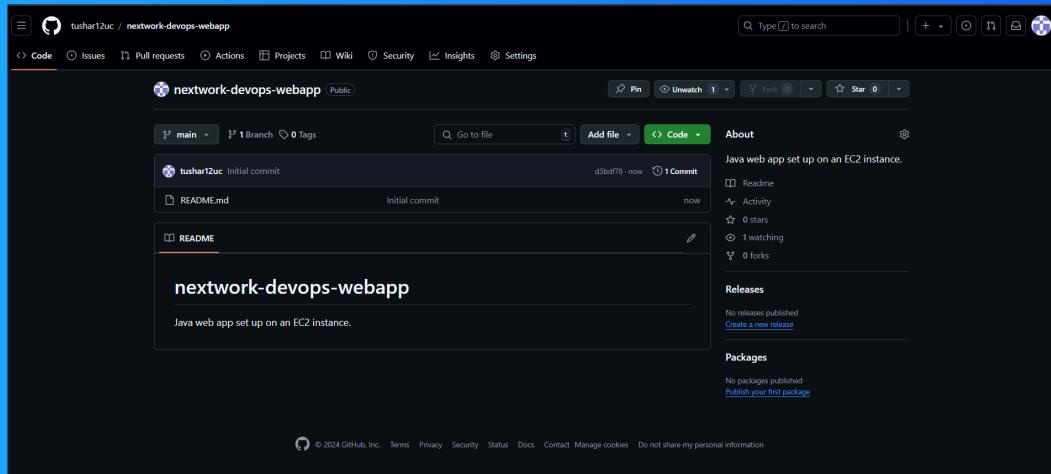
This project took me...

This project took me about 2 hours to complete, including setting up GitHub, making commits, and configuring the AWS deployment.

Git and GitHub

Git is a distributed version control system for tracking code changes. I installed Git using the commands `sudo apt update` and `sudo apt install git`.

GitHub is a web-based platform for version control and collaboration, allowing developers to store, manage, and share their code. I'm using GitHub in this project to enable automatic deployment of my code to AWS whenever I make updates to the repository.



My local repository

A Git repository is a storage space that tracks changes to files, allowing multiple developers to collaborate on a project by managing versions and history of the code.

`git init` is a command that initializes a new Git repository in a specified directory. I ran `git init` in my project folder to start tracking changes and manage my code versioning.

After running `git init`, the response from the terminal was a confirmation of the new repository. A branch in Git is a separate line of development for working on features or changes independently.

```
● [ec2-user@ip-172-31-47-72 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-172-31-47-72 nextwork-web-project]$ █
```

To push local changes to GitHub, I ran three commands

git add

The first command I ran was `git add`. A staging area is a space where I prepare changes to be committed, allowing me to selectively choose which modifications to include in the next commit.

git commit

The second command I ran was `git commit -m "commit message"`. Using `-m` means I'm adding a brief message to describe the changes I've made in this commit.

git push

The third command I ran was `git push -u origin main`. Using `-u` means I'm setting the upstream branch, so future pushes will default to this branch without needing to specify it again.

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 repository.

Local Git identity

Git needs my name and email because they are used to identify the author of each commit, ensuring proper attribution for changes made in the repository.

Running `git log` showed me that it displays the commit history, including each commit's unique ID, author, date, and message, allowing me to track changes over time.

```
● [ec2-user@ip-172-31-47-72 nextwork-web-project]$ git log
commit c47b706766e42f9da2388b52392a9371bb106fc7 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-47-72.ec2.internal>
Date:   Tue Oct 15 17:47:46 2024 +0000

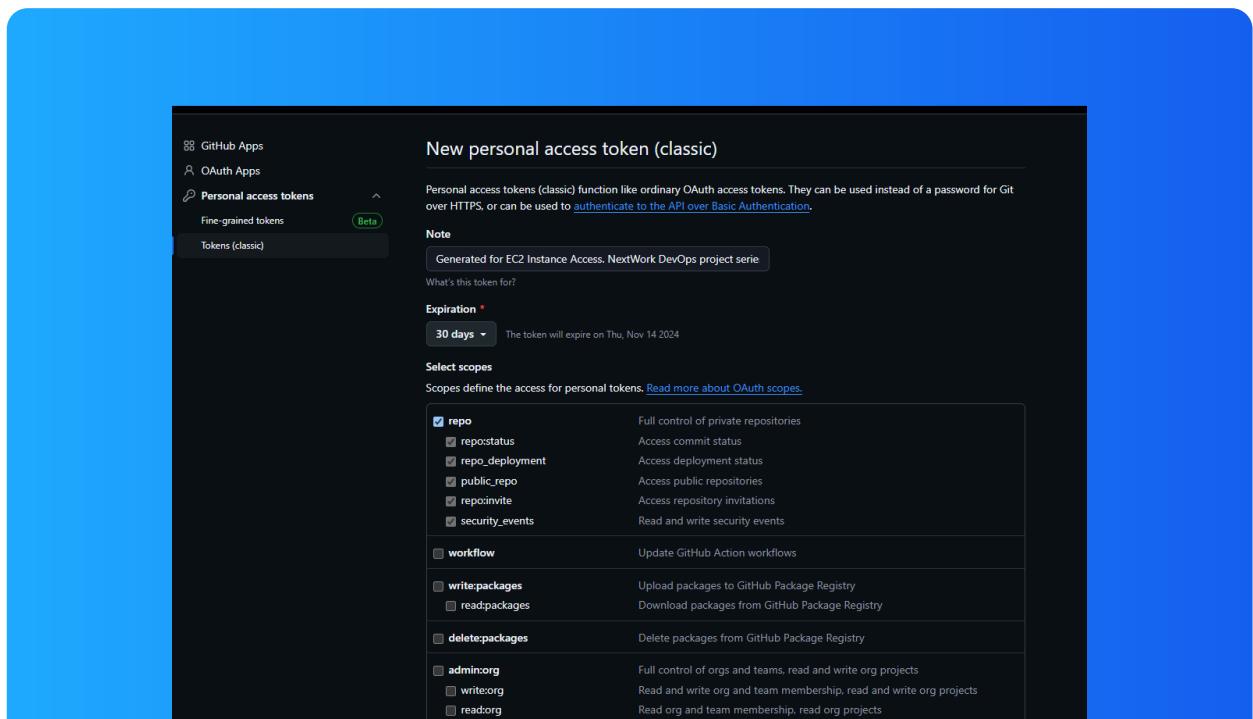
    Updated index.jsp with new content
○ [ec2-user@ip-172-31-47-72 nextwork-web-project]$ █
```

GitHub tokens

GitHub authentication failed when I entered my password because I needed to use a personal access token instead of my password, as GitHub no longer supports password authentication for Git operations.

A GitHub token is a secure string for accessing my GitHub account. I'm using one in this project because GitHub requires tokens for authentication instead of passwords for better security.

I could set up a GitHub token by navigating to my GitHub account settings, selecting "Developer settings," then "Personal access tokens," and generating a new token with the required permissions.



Making changes again

I wanted to see Git in action, so I updated the `index.html` file in the `nextwork-web-project`. I couldn't see the changes in my GitHub repo initially because I hadn't committed and pushed the updates yet.

I finally saw the changes in my GitHub repo after committing the updates and pushing them to the remote repository.

```
      Updated index.jsp with new content
● [ec2-user@ip-172-31-47-72 nextwork-web-project]$ git config --global user.name "Your Name"
git config --global user.email you@nextwork.org
● [ec2-user@ip-172-31-47-72 nextwork-web-project]$ git add .
● [ec2-user@ip-172-31-47-72 nextwork-web-project]$ git diff --staged
○ [ec2-user@ip-172-31-47-72 nextwork-web-project]$ git commit -m "Add new line to index.jsp"
git push
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
Username for 'https://github.com': █
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

