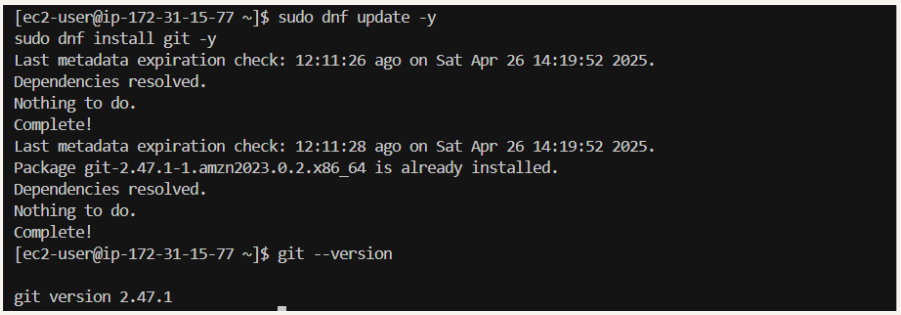
Connect a GitHub Repo with AWS

lue



Introducing Today's Project!

We're here to set up Git and GitHub, connect our web app project to a GitHub repository, and learn how to track and manage code changes. We'll also make updates to our code and see them reflected in GitHub, plus add a README file to our repo.

Key tools and concepts

Services I used were GitHub for code hosting and Git for version control. Key concepts include creating personal access tokens, committing and pushing changes, setting author info, and using git log to track and review project history effecttively.

Project reflection

I completed this project in about 90 minutes, which included setting up GitHub, creating a personal access token, making changes to my web app, committing and pushing those changes, and verifying everything worked correctly.

I did this project today to improve my Git and GitHub skills, essential for a cloud career. It met my goals by giving me hands-on experience with version control, committing, pushing changes, and managing repositories, all key for working in cloud.

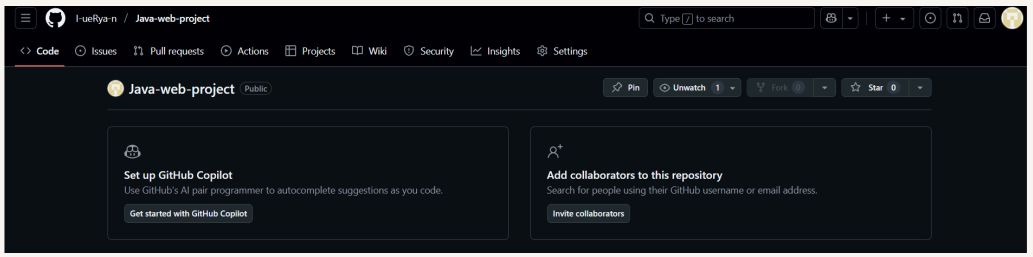
lue
NextWork Student
nextwork.org

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 continue developing my skills and move closer to mastering cloud and DevOps workflows.

Git and GitHub

Git is a version control system that tracks code changes and helps collaboration. I installed Git on my EC2 instance using the command sudo yum install git -y, enabling me to manage and share my project code efficiently

GitHub is a web-based platform that uses Git to help developers create, store, manage, and share code with version control and collaboration tools. I'm using GitHub in this project to track changes, collaborate efficiently, and keep a history.

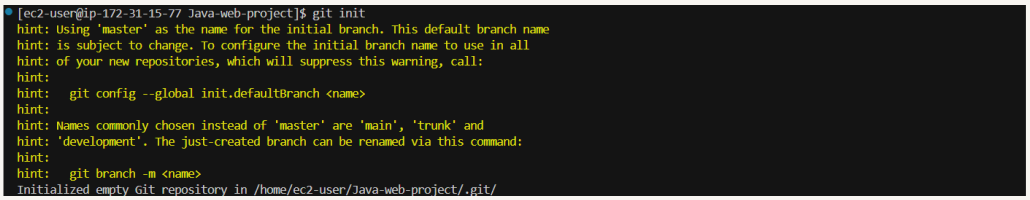


My local repository

A Git repository is a storage for my project's files and their history. It tracks every change, allowing you to manage versions, collaborate with others, and revert to earlier code states easily, making teamwork and code management efficient.

git init creates a new Git repository in my project folder by setting up a hidden .git directory. This enables Git to track changes, manage versions, and start version control for my project locally.

A branch in Git is a separate line of development that lets you work on new features or fixes without affecting the main code. This helps manage changes and collaborate with others safely before merging updates.


[ec2-user@ip-172-31-15-77 Java-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/Java-web-project/.git/

To push local changes to GitHub, I ran three commands

git add

The first command I ran was git add . A staging area is where Git collects changes you want to include in your next commit. This command stages all modified and new files, preparing them to be saved to your project's history with your next commit.

git commit

The second command I ran was `git commit -m "Updated index.jsp with new content"`. Using '-m' means I added a commit message directly, describing my changes. This command saves the staged changes as a new version in my local Git history.

git push

The third command I ran was `git push -u origin master`. Using '-u' means I set the upstream branch, linking my local master to the remote. This uploads my commits to GitHub and lets future pushes use just `git push` without extra details.

Authentication

When I commit changes to GitHub, Git asks for my credentials because authentication is needed to verify my identity and protect the repository. Using HTTPS, Git prompts for my GitHub username and token to ensure only authorized users can push changes

Local Git identity

Git needs my name and email because each commit must show who made the change. This info helps track contributions, keeps the project history clear, and ensures everyone gets proper credit for their work in the repository's commit log.

Running git log showed me that my latest commit was made by "EC2 Default User" with the email "ec2-user@ip-172-31-15-77.ap-southeast-2.compute.internal." It also displayed the commit hash, date, and message: "Updated index.jsp with new content."

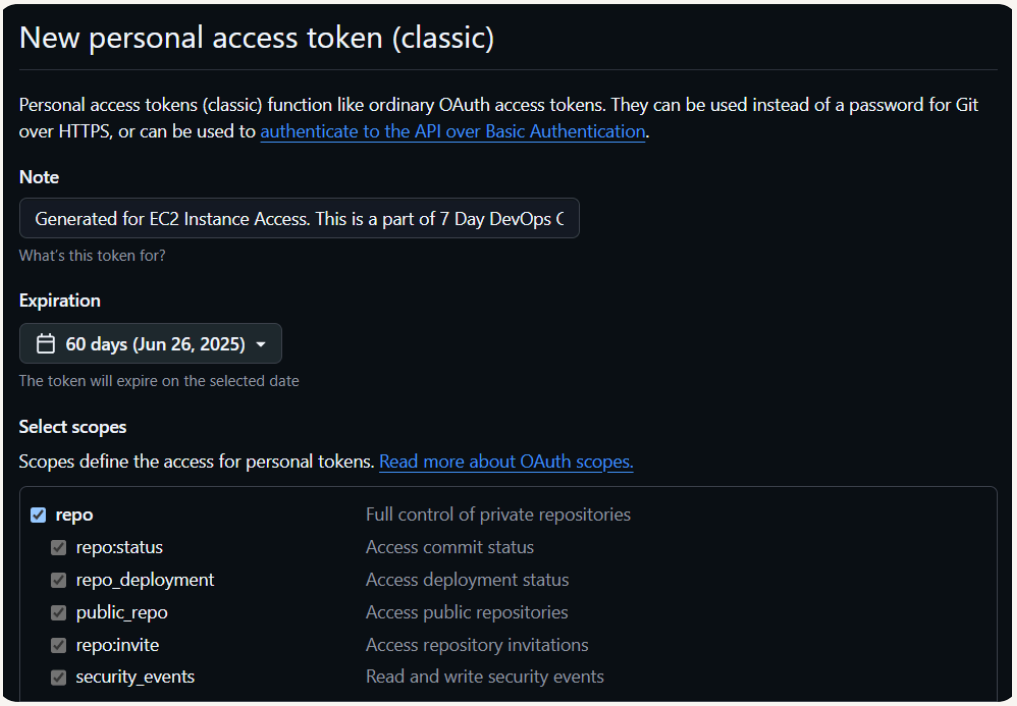
```
[ec2-user@ip-172-31-15-77 Java-web-project]$ git log
commit 5ba97988af60a500cda6f5e69a203de01a4cf230 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-15-77.ap-southeast-2.compute.internal>
Date:   Sun Apr 27 03:49:01 2025 +0000

    Updated index.jsp with new content
```

GitHub tokens

GitHub no longer lets you use your password to push code over HTTPS because it's less secure. Instead, you must use a personal access token-a unique, random string that acts like a safer, one-time password for your GitHub actions.

A GitHub token is a unique, secure code that replaces your password for authenticating with GitHub. I'm using one in this project because it's safer than a password, can be limited in scope, and is required for HTTPS Git operations.

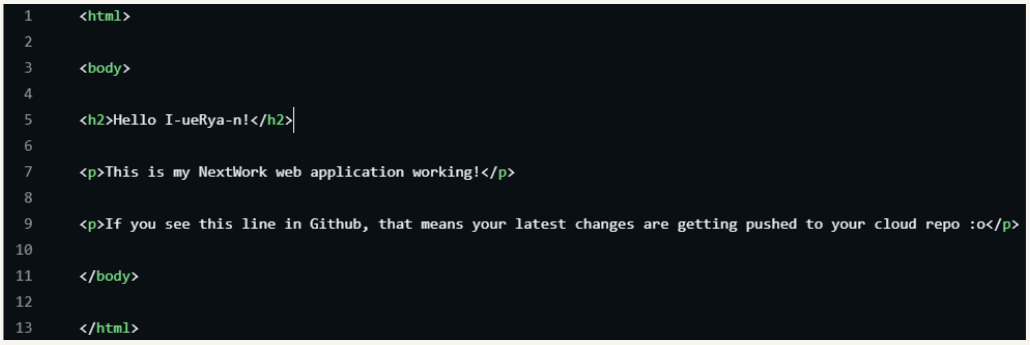
I could set up a GitHub token by going to Settings > Developer settings > Personal access tokens, clicking "Generate new token," choosing the required scopes, and copying the generated token to use as my password for Git operations.


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 7 Day DevOps C
What's this token for?
Expiration
60 days (Jun 26, 2025)
The token will expire on the selected date
Select scopes
Scopes define the access for personal tokens. Read more about OAuth scopes.
repo
Full control of private repositories
repo:status
Access commit status
repo_deployment
Access deployment status
public_repo
Access public repositories
repo:invite
Access repository invitations
security_events
Read and write security events

Making changes again

I wanted to see Git working in action, so I updated the index.jsp file in Java-web-project. I couldn't see the changes in my GitHub repo initially because I hadn't committed and pushed the changes from my local machine to the remote repository yet.

I finally saw the changes in my GitHub repo after committing my updates locally with a message and then pushing them to the remote repo using git push. This synced my local changes with GitHub, making them visible in the repository online.



The place to learn & showcase your skills

Check out nextwork.org for more projects