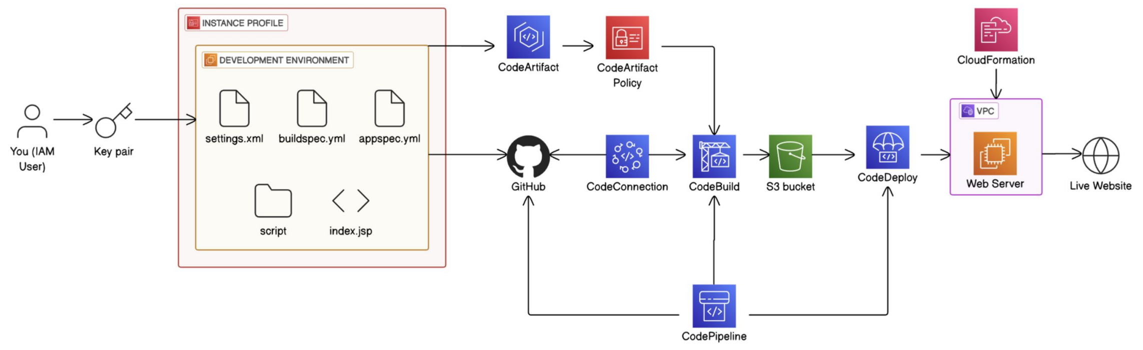
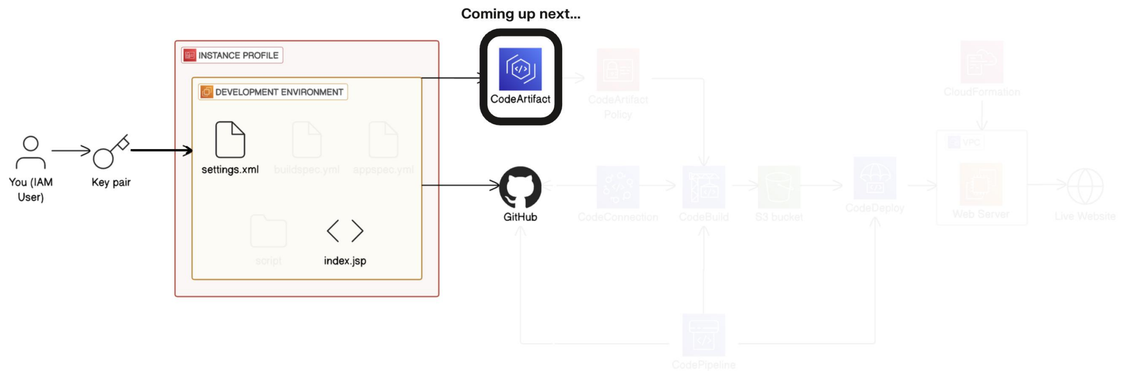
CONNECT A GITHUB REPO WITH AWS



Introducing Today's Project!

Today, I'll set up Git and GitHub for my web app project, initialize a Git repo, connect it to GitHub, and push my files. I'll make changes, track updates, and see them reflect in GitHub, ensuring version control and smooth collaboration!



Key tools and concepts

Services I used were Git, GitHub, and EC2 for saving and sharing code. Key concepts I learned include Git repositories, branches, commits, pushing changes, using tokens or SSH for security, and tracking updates to work easily with others.

Project reflection

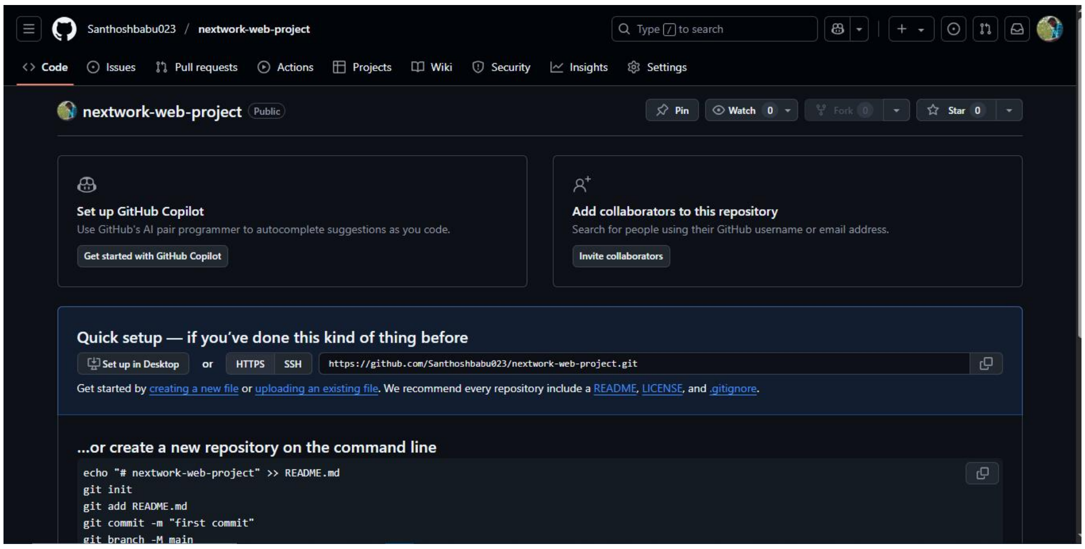
This project took me a few hours to complete. The hardest part was setting up GitHub authentication without using a password. The best part was successfully pushing my code to GitHub and learning how Git helps track and manage changes.
I did this project to learn Git and GitHub for version control and collaboration. Yes, it met my goals! I successfully set up a repository, tracked changes, pushed code to GitHub, and learned authentication, making project management easier.

This project is part two of a DevOps series where I'm building a CI/CD pipeline! I'll be working on the next project soon, focusing on automating deployments, improving workflows, and integrating more DevOps tools for better software delivery.

Git and GitHub

Git is a version control system that tracks code changes, enables collaboration, and manages projects efficiently. I installed Git using the commands:
sudo dnf update -y
sudo dnf install git -y

GitHub is a cloud platform for storing and managing projects tracked by Git. It enables version control, collaboration, and remote access to code. I'm using GitHub in this project to track changes, share updates, and work with others efficiently.



Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository
Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

```
https://github.com/Santhoshbabu023/nextwork-web-project.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

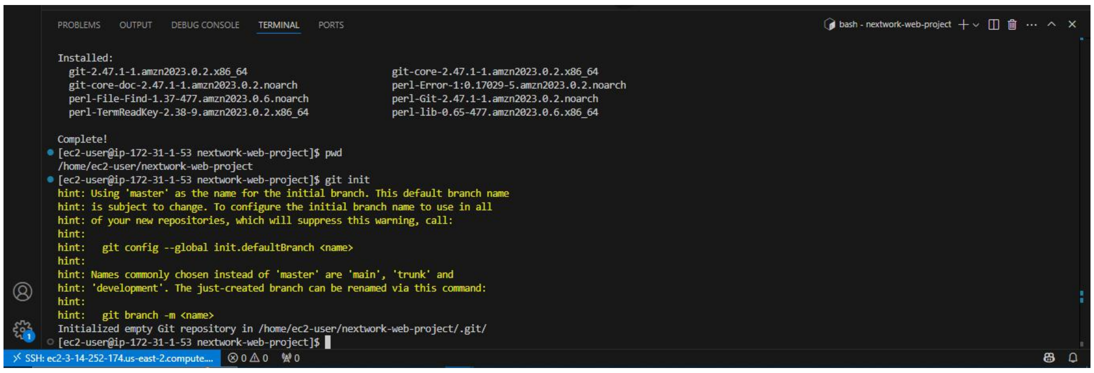
```
echo "# nextwork-web-project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
```

My local repository

A Git repository is a storage space that tracks changes in a project, enabling version control and collaboration. It contains files, commit history, and branches, helping developers manage code efficiently. Repositories can be local or hosted online.

git init is a command that creates a new Git repository in a project folder, enabling version control. It sets up the necessary files for tracking changes. I ran git init in my web app folder to start managing code versions efficiently.

After running git init, the terminal showed "Initialized empty Git repository..", meaning Git is tracking changes. A branch in Git is a separate version of your project where you can work on changes safely without affecting the main code.


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
bash - nextwork-web-project
SSH: ec2-3-14-252-174.us-east-2.compute....
0 0
0

```
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-172-31-1-53 nextwork-web-project]$ pwd
/home/ec2-user/nextwork-web-project
[ec2-user@ip-172-31-1-53 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-172-31-1-53 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 all changes to the staging area before saving them. A staging area is where Git keeps track of changes before they are saved. It helps review and organize updates before committing them to repository.

git commit

The second command I ran was git commit -m "message", which saves the staged changes with a description. Using -m means adding a short message to explain the commit, making it easier to track updates and understand what changes were made in project.

git push

The third command I ran was git push -u origin master, which uploads my committed changes to the GitHub repository.Using -u means setting the upstream branch, so future pushes can be done with just git push, making it easier to update the repository.

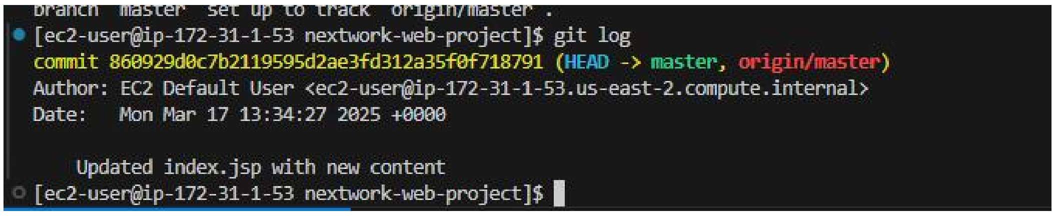
Authentication

When I commit changes to GitHub, Git asks for my credentials because I'm using HTTPS authentication, which requires a username and a personal access token (PAT). To avoid this, I can set up SSH keys for secure and password-free access.

Local Git identity

Git needs my name and email because it identifies me as the author of my commits and records them in the project history.This helps track contributions, maintain accountability, and collaborate efficiently by ensuring changes are properly attributed.

Running git log showed me that Git keeps a history of all commits, including the commit hash, author name, email, date, and message. This helps track changes, see who made updates, and review previous versions of the project for better collaboration.



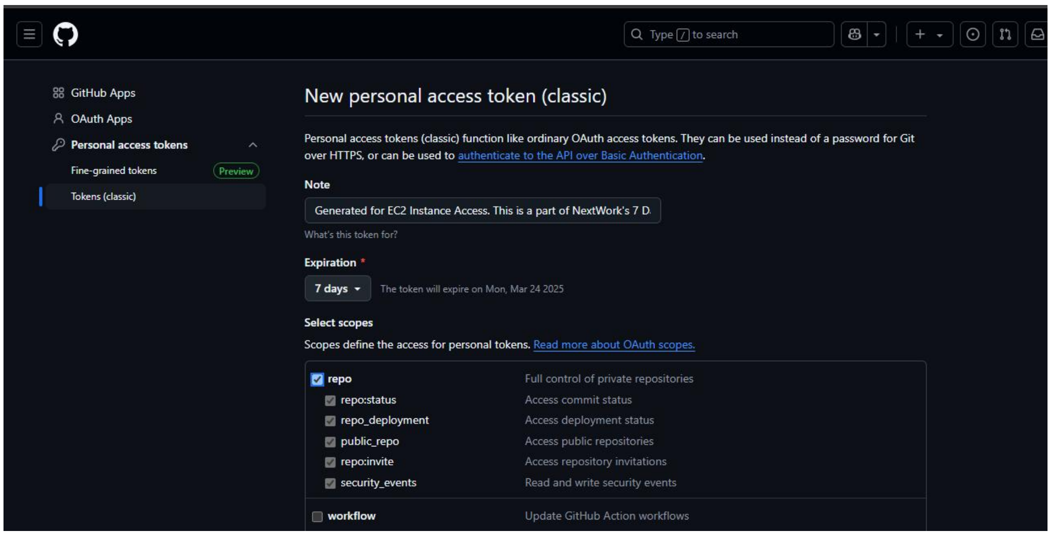
```
branch 'master' set up to track 'origin/master'.
[ec2-user@ip-172-31-1-53 nextwork-web-project]$ git log
commit 860929d0c7b2119595d2ae3fd312a35f0f718791 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-1-53.us-east-2.compute.internal>
Date:   Mon Mar 17 13:34:27 2025 +0000

    Updated index.jsp with new content
[ec2-user@ip-172-31-1-53 nextwork-web-project]$
```

GitHub tokens

GitHub authentication failed when I entered my password because GitHub no longer supports password authentication for Git operations. Instead, I need to use a personal access token (PAT) for HTTPS authentication or set up SSH keys for secure access.

A GitHub token is a secure access key used instead of a password for authentication. I'm using one in this project because GitHub no longer allows password authentication for Git operations, ensuring a safe and secure way to push and manage code.

I could set up a GitHub token by going to GitHub Settings, then Developer settings → Personal access tokens, and clicking Generate new token. After selecting the required repo permissions, I generated the token and used it instead of a password.



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 NextWork's 7 D

What's this token for?

Expiration *

7 days — The token will expire on Mon, Mar 24 2025

Select scopes

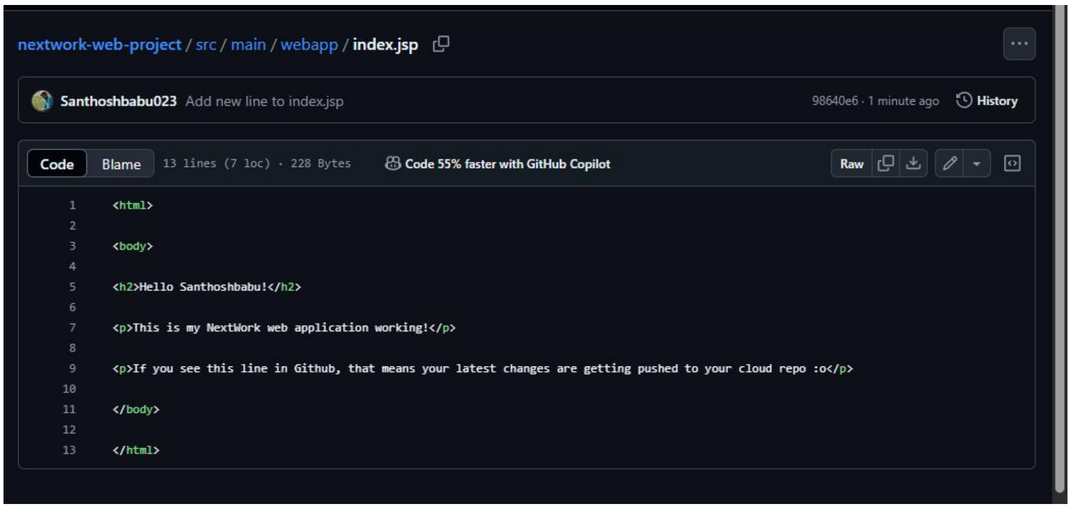
Scopes define the access for personal tokens. Read more about OAuth scopes.

✓ 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
☐ workflow	Update GitHub Action workflows

Making changes again

I wanted to see Git working in action, so I updated index.jsp in the nextwork-web project and committed the changes. I couldn't see the changes in my GitHub repo at first because I hadn't pushed them yet. After running git push, the updates appeared.

I finally saw the changes in my GitHub repo after committing the updates using git commit -m "message" and pushing them with git push. Committing saved the changes locally, and pushing uploaded them to GitHub, making them visible in the repository.



nextwork-web-project / src / main / webapp / index.jsp

Santhoshbabu023 Add new line to index.jsp — 98640e6 · 1 minute ago

13 lines (7 loc) · 228 Bytes

```
<html>

<body>

<h2>Hello Santhoshbabu!</h2>

<p>This is my NextWork web application working!</p>

<p>If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :o</p>

</body>

</html>
```