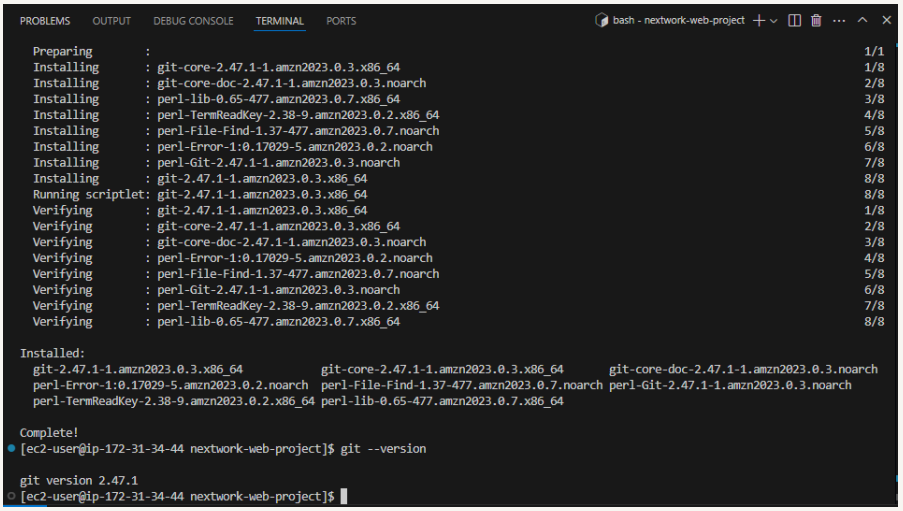
Connect a GitHub Repo with AWS

Thabang Khasebe



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                                   bash - nextwork-web-project

  Preparing        :                                                                                                  1/1
  Installing       : git-core-2.47.1-1.amzn2023.0.3.x86_64                                                            1/8
  Installing       : git-core-doc-2.47.1-1.amzn2023.0.3.noarch                                                        2/8
  Installing       : perl-lib-0.65-477.amzn2023.0.7.x86_64                                                            3/8
  Installing       : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64                                                      4/8
  Installing       : perl-File-Find-1.37-477.amzn2023.0.7.noarch                                                      5/8
  Installing       : perl-Error-1:0.17029-5.amzn2023.0.2.noarch                                                       6/8
  Installing       : perl-Git-2.47.1-1.amzn2023.0.3.noarch                                                            7/8
  Installing       : git-2.47.1-1.amzn2023.0.3.x86_64                                                                 8/8
  Running scriptlet: git-2.47.1-1.amzn2023.0.3.x86_64                                                                 8/8
  Verifying        : git-2.47.1-1.amzn2023.0.3.x86_64                                                                 1/8
  Verifying        : git-core-2.47.1-1.amzn2023.0.3.x86_64                                                            2/8
  Verifying        : git-core-doc-2.47.1-1.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.amzn2023.0.7.noarch                                                      5/8
  Verifying        : perl-Git-2.47.1-1.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.amzn2023.0.7.x86_64                                                            8/8

Installed:
  git-2.47.1-1.amzn2023.0.3.x86_64            git-core-2.47.1-1.amzn2023.0.3.x86_64       git-core-doc-2.47.1-1.amzn2023.0.3.noarch
  perl-Error-1:0.17029-5.amzn2023.0.2.noarch  perl-File-Find-1.37-477.amzn2023.0.7.noarch perl-Git-2.47.1-1.amzn2023.0.3.noarch
  perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.7.x86_64

Complete!
[ec2-user@ip-172-31-34-44 nextwork-web-project]$ git --version

git version 2.47.1
[ec2-user@ip-172-31-34-44 nextwork-web-project]$
```

Thabang Khasebe

Introducing Today's Project!

Today, I will learn how to use Git to store and manage my web app's code. This helps keep track of changes, collaborate with others, and maintain a history of the project's development.

Key tools and concepts

Services I used were AWS EC2 and GitHub. Key concepts I learnt include installing Git, creating and linking GitHub repos, using Git commands to commit and push code, and using personal access tokens for secure authentication.

Project reflection

This project took me approximately 2 hours. The most challenging part was setting up authentication with GitHub tokens. It was most rewarding to successfully push my web app changes to the GitHub repository.

I chose to do this project today to learn how to manage my code with Git and GitHub, essential skills for any developer. The project met my goals by helping me understand version control and remote code management on AWS.

Thabang Khasebe

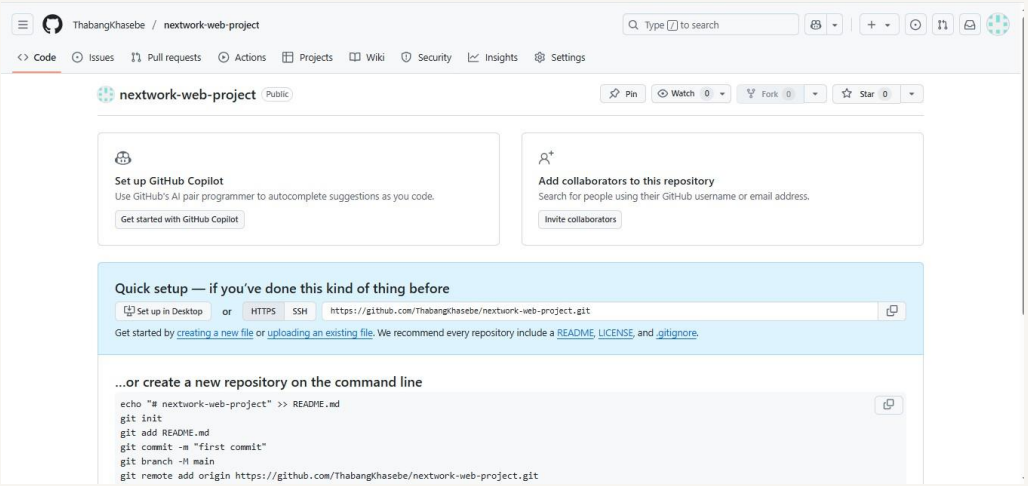
This project is part two of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project tomorrow.

Thabang Khasebe

Git and GitHub

Git is a version control system that helps track changes and manage code history. I installed Git using the commands `sudo dnf update -y` followed by `sudo dnf install git -y` on my EC2 instance.

GitHub is a cloud-based platform for hosting and managing Git repositories. I'm using GitHub in this project to store my web app code, track changes, and make it easier to collaborate, share, and access my project from anywhere.



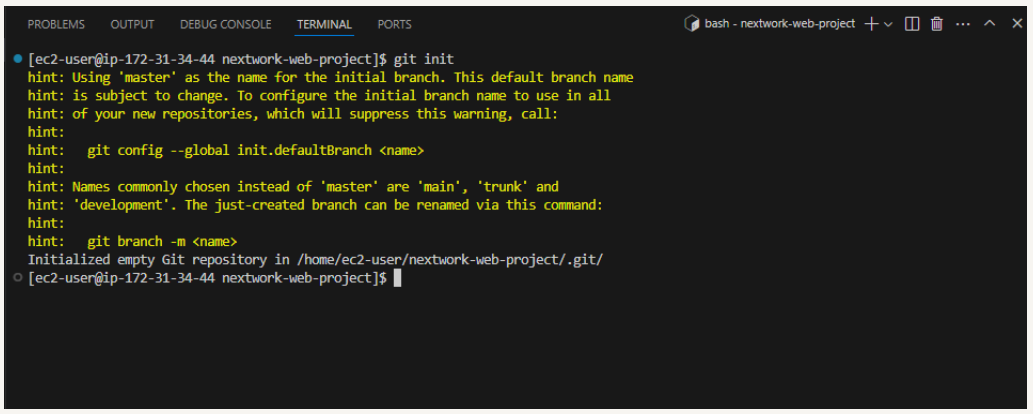
Thabang Khasebe

My local repository

A Git repository is a storage space where your project's files, history, and changes are tracked using Git. It allows you to manage versions, collaborate with others, and easily roll back or review code changes.

git init is a command that initializes a new Git repository in a folder, allowing Git to start tracking changes. I ran git init in my web app project folder on the EC2 instance to begin version control locally.

After running git init, the terminal said "Initialized empty Git repository..." A branch in Git is a pointer to a set of commits, allowing you to work on changes separately without affecting the main project.


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
bash - nextwork-web-project
[ec2-user@ip-172-31-34-44 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-172-31-34-44 nextwork-web-project]$

Thabang Khasebe

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 I want to include in the next commit, allowing me to review and organize updates before saving them permanently.

git commit

The second command I ran was `git commit -m "Updated index.jsp with new content"`. Using '-m' means I'm adding a message to describe the changes in this commit, helping others understand what was updated.

git push

The third command I ran was `git push -u origin master`. Using '-u' means I'm setting the upstream branch, so future pushes can be done simply with `git push` without specifying the remote and branch.

Thabang Khasebe

Authentication

When I commit changes to GitHub, Git asks for my credentials because it needs to verify my identity to ensure I have permission to push changes to the repository. This keeps the code secure and prevents unauthorized access.

Local Git identity

Git needs my name and email because it uses this information to label each commit, helping identify who made changes and when. This is important for tracking contributions and collaborating with others.

Running git log showed me that my commit was successfully recorded, including details like the commit ID, author name and email, date, and the commit message describing the changes I made.

Thabang Khasebe

```
[ec2-user@ip-172-31-34-44 nextwork-web-project]$ git log
commit 122096c288c747c90fd4c4385693cbff979e8021 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-34-44.eu-north-1.compute.internal>
Date:   Wed Jul 2 22:41:52 2025 +0000

    Updated index.jsp with new content
[ec2-user@ip-172-31-34-44 nextwork-web-project]$
```

Thabang Khasebe

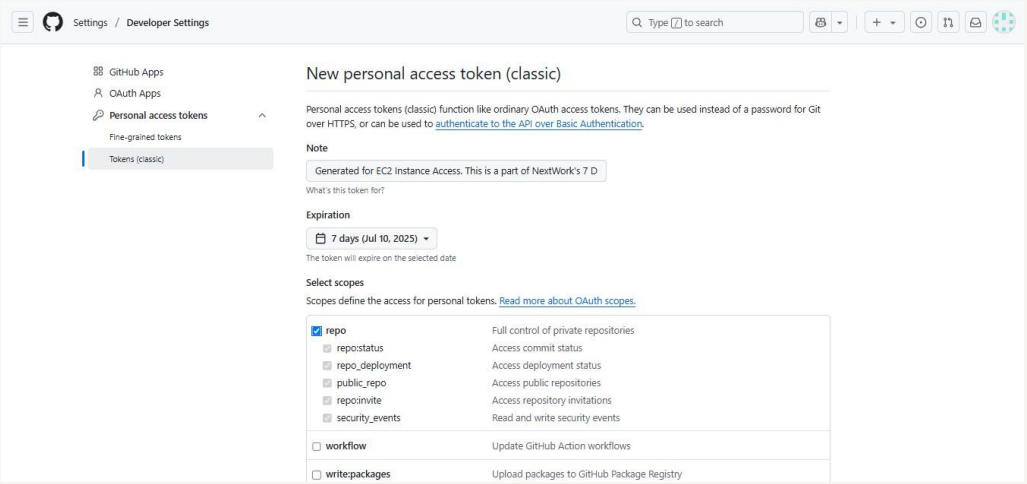
GitHub tokens

GitHub authentication failed when I entered my password because GitHub no longer accepts account passwords for Git operations. Instead, it requires a personal access token (PAT) for secure authentication when pushing code.

A GitHub token is a secure authentication key that replaces your password for Git operations. I'm using one in this project because GitHub no longer accepts passwords when pushing code, and tokens provide a safer way to verify my identity.

I could set up a GitHub token by going to my GitHub account settings, selecting "Developer settings," then "Personal access tokens," and generating a new token with the necessary scopes for accessing and pushing to my repositories.

Thabang Khasebe



Thabang Khasebe

Making changes again

I wanted to see Git working in action, so I updated the `index.jsp` file in my 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 my updates with `git commit` and pushing them using `git push`, which uploaded the latest code from my EC2 instance to the remote repository.