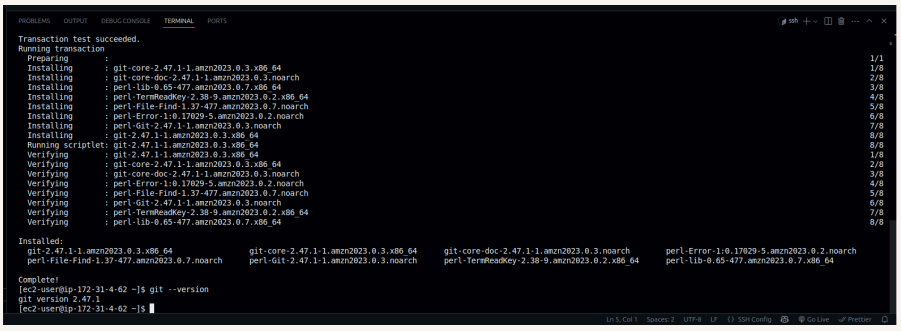
nextwork.org

Connect a GitHub Repo with AWS

Shravan Kumar Satapathy



Shravan Kumar Satapathy
NextWork Student

Introducing Today's Project!

Today, I'll set up Git and GitHub, then connect my web app project to a GitHub repository. I'll practice making code changes and updating the repo, and finally, create a README file for it.

Key tools and concepts

Services I used were GitHub and Git. Key concepts learned include Git repository initialization, remote management, staging/committing changes, pushing updates with branch tracking, and Git identity/secure authentication via Personal Access Tokens.

Project reflection

This project took me approximately a few hours. The most challenging part was troubleshooting GitHub authentication with Personal Access Tokens. It was most rewarding to successfully push changes and see them updated live in the GitHub repository.

I undertook this project today to establish a foundational understanding of Git and GitHub for version control. It successfully met my goal of learning to manage code changes and remote repositories, which is crucial for future DevOps work.

Shravan Kumar Satapathy
NextWork Student
nextwork.org

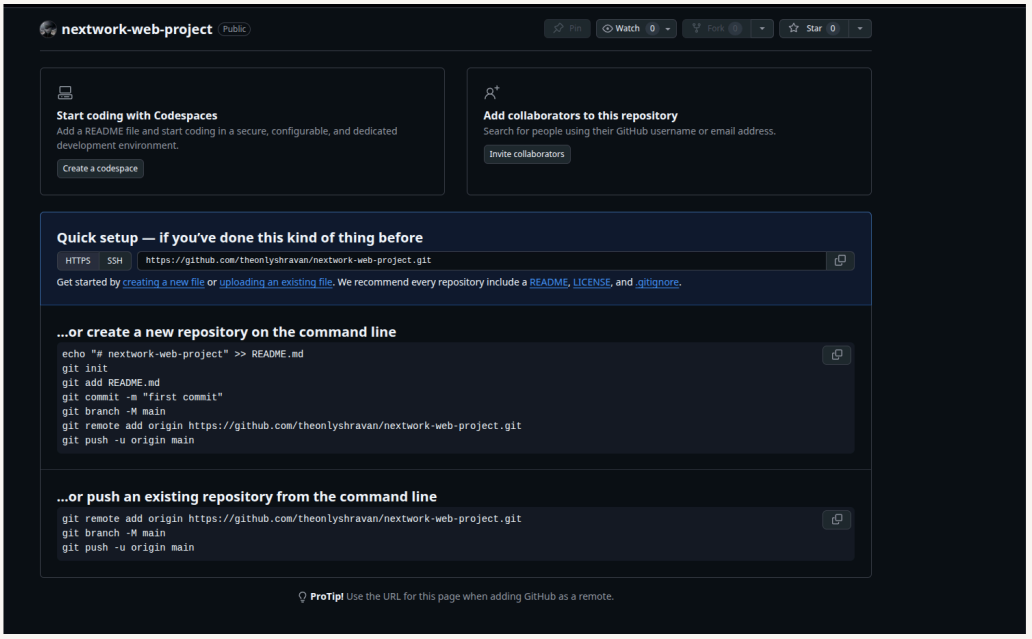
This project is part two of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project next week.

Shravan Kumar Satapathy
NextWork Student

Git and GitHub

Git is a version control system that tracks code changes, enabling easy collaboration and version history. I installed it on my EC2 instance using sudo dnf update -y and sudo dnf install git -y.

GitHub is an online platform for code storage, sharing, and version control using Git. I'm GitHub in this project to secure my web app's code in the cloud, visually track changes, and facilitate team collaboration.



Shravan Kumar Satapathy
NextWork Student

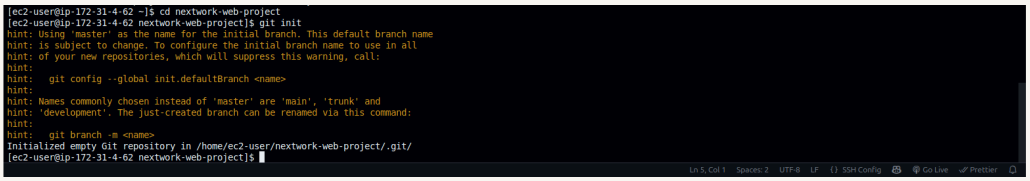
My local repository

A Git repository is a digital folder containing all project files, along with their complete version history. When hosted remotely, it facilitates collaborative development and enables access to the project's codebase from any location.

'git init' is a command that initializes a new local Git repository in a directory, enabling version control. It prepares the directory to track changes. I ran git init in the nextwork-web-project directory.

After running git init, he response from the terminal was yellow text about the default 'master' branch. A branch in Git is a parallel project version, enabling isolated development and safe experimentation with features or fixes.

Shravan Kumar Satapathy
NextWork Student



Shravan Kumar Satapathy
NextWork Student

To push local changes to GitHub, I ran three commands

git add

The first command was 'git add .', which stages all modified files. A staging area is where changes are gathered for review before being committed as a new version, ensuring accuracy and control.

git commit

The second command I ran was 'git commit -m "Updated index.jsp with new content"'. This saves staged changes as a project snapshot. Using '-m' adds a concise, descriptive message, vital for documenting changes and reviewing version history.

git push

The third command I ran was 'git push -u origin master'. This uploads my committed changes to the master branch of my GitHub repository (origin). Using -u establishes an upstream tracking reference, simplifying future push operations to this branch.

Shravan Kumar Satapathy
NextWork Student

Authentication

When committing to GitHub, Git requests credentials to authenticate my identity. This ensures I have the necessary permissions to push changes to the remote repository, maintaining authorized access and integrity.

Local Git identity

Git needs my name and email for commit authorship. This ensures accurate tracking of who made each change in the project's version history. Otherwise, Git might use a generic system default, misrepresenting my identity.

Running git log showed me the project's complete commit history. This output includes details like the commit author's name, email, and message, providing a clear audit trail of all changes.

```
[ec2-user@ip-172-31-4-62 nextwork-web-project]$ git log
commit 0b68ff4157958c06b47da9d60eddade8b3cc1c66 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-4-62.ap-south-1.compute.internal>
Date:   Sat Jun 21 18:16:02 2025 +0000

    Updated index.jsp with new content
[ec2-user@ip-172-31-4-62 nextwork-web-project]$
```

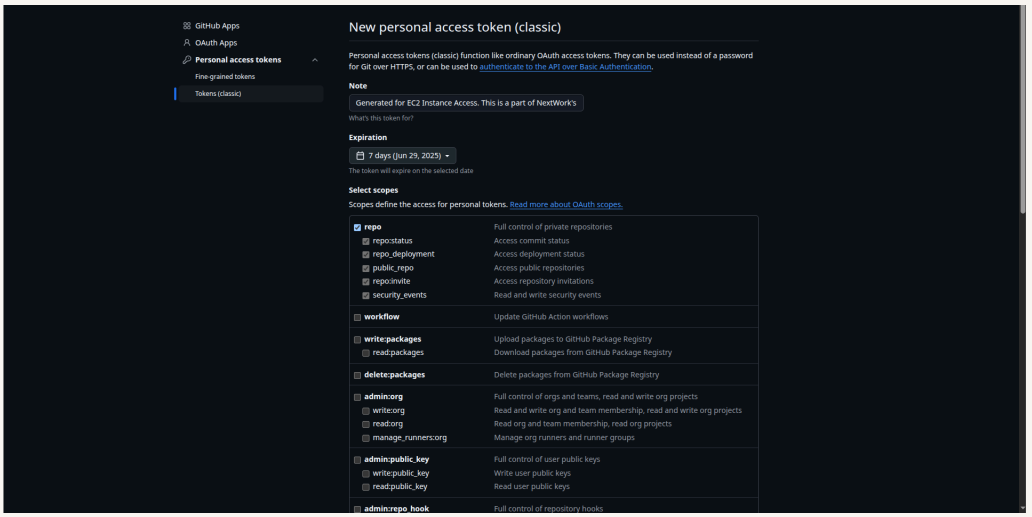
GitHub tokens

GitHub authentication failed when I entered my password because GitHub has deprecated password-based authentication over HTTPS due to security risks. A personal access token, a more secure alternative, is now required for interacting with repo.

A GitHub token is a unique string for secure authentication, replacing passwords. I'm using one because GitHub no longer supports password-based HTTPS authentication due to security risks, making tokens a required, more secure method.

I could set up a GitHub token by navigating to my GitHub account's developer settings then generating a new personal access token with appropriate repository permissions, and then securely copying the generated token for future use.

Shravan Kumar Satapathy
NextWork Student

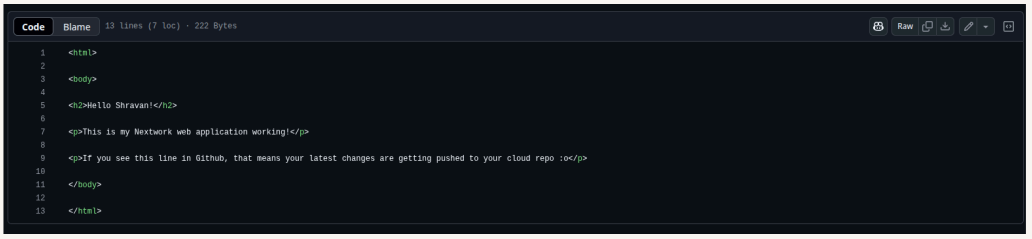


Shravan Kumar Satapathy
NextWork Student

Making changes again

I wanted to see Git working, so I updated index.jsp. I couldn't see changes in GitHub initially because saving only updates the local repo; changes must be explicitly pushed from local to the remote GitHub repo to be visible.

I finally saw the changes in my GitHub repo after staging with git add ., committing with git commit, and pushing with git push. I then authenticated using my username and personal access token when prompted.



The place to learn & showcase your skills

Check out nextwork.org for more projects