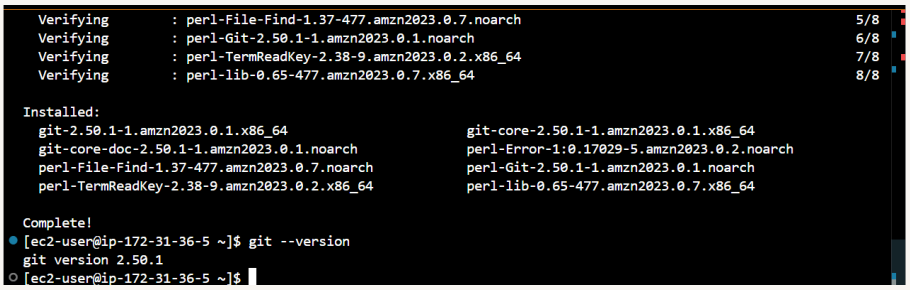
nextwork.org

Connect a GitHub Repo with AWS

naveen msd



Introducing Today's Project!

In this project, I will demonstrate how to create a GitHub repo for a Web app and connect it with AWS, setting up the foundation for automated builds and deployments.

Key tools and concepts

The key services I used for this are GitHub. in this I learnt how to set up a git hup repo, connect git to EC2 Instance, generate tokens and commands used in git and performed commit and push.

Project reflection

This project took me approximately took about an hour. It was most rewarding to see hwo git communicates and all the changes made were pushed, and I could see these reflecting on my GitHub repo

I did this project to understand how the Code is being maintained, stored, and updated automatically.

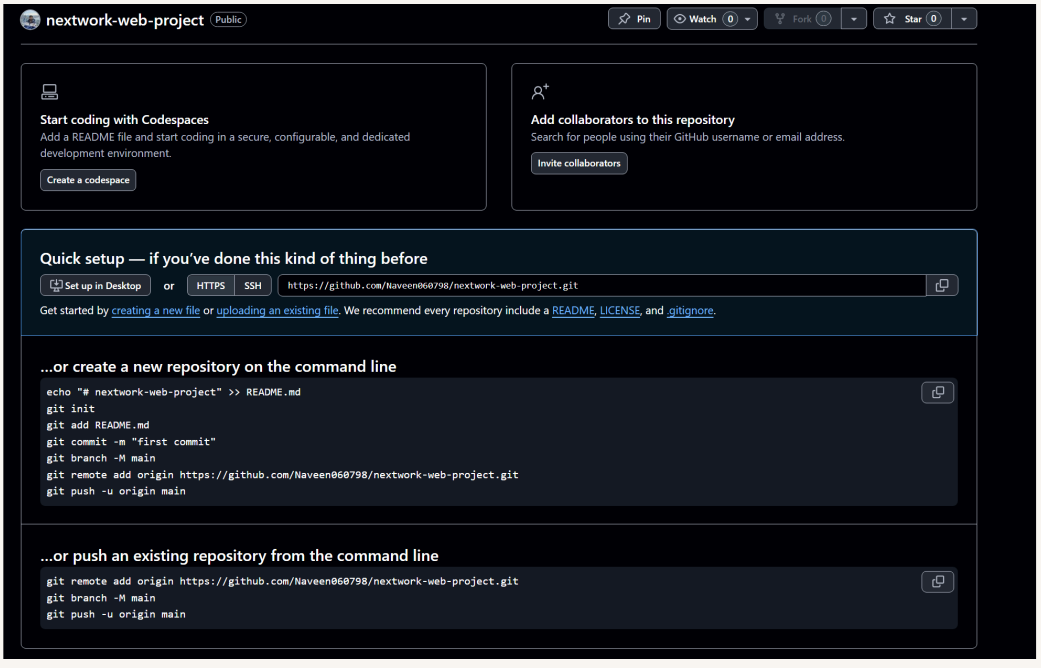
naveen msd
NextWork Student

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.

Git and GitHub

Git is a Version Control it tracks changes by taking snapshots of what files look like at specific moments, and each snapshot is considered a 'version'. I installed Git using the command "sudo dnf install git -y"

GitHub is a storage space for different versions of the project that Git tracks. It lets me access the work from anywhere over the internet. GitHub helps me use Git and see my file changes in a more user-friendly way.

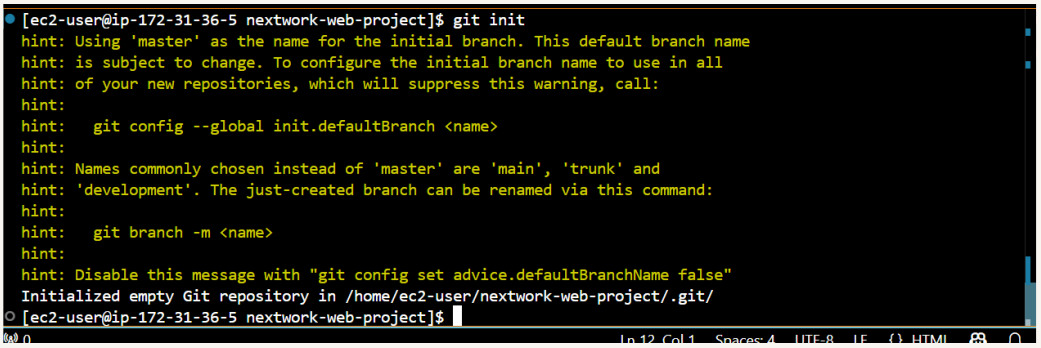


My local repository

A Git repository is like a folder that stores all the project files and their entire version history. Also, lets me collaborate with other developers over the internet.

Git init is a command that initiates to creation of a local git repo on the EC2 instance, where changes are now tracked for version control.

A branch in Git is a parallel version or 'alternate universe' of the same project. After running git init, the response from the terminal was successful in creating an empty repo and suggested changing the Main branch name.



To push local changes to GitHub, I ran three commands

git add

The first command I ran was git diff --staged. A staging area tells Git to put together all your modified files for a final review before you commit them. This is incredibly handy because all the edits are in one spot.

git commit

The second command I ran was git commit -m "message". Using '-m' means leaving a message describing what the commit is about, making it easier to review what changed in this version.

git push

The third command I ran was git push -u origin master. Using '-u' means setting an 'upstream' for the local branch, which means you're telling Git to remember to push to master by default.

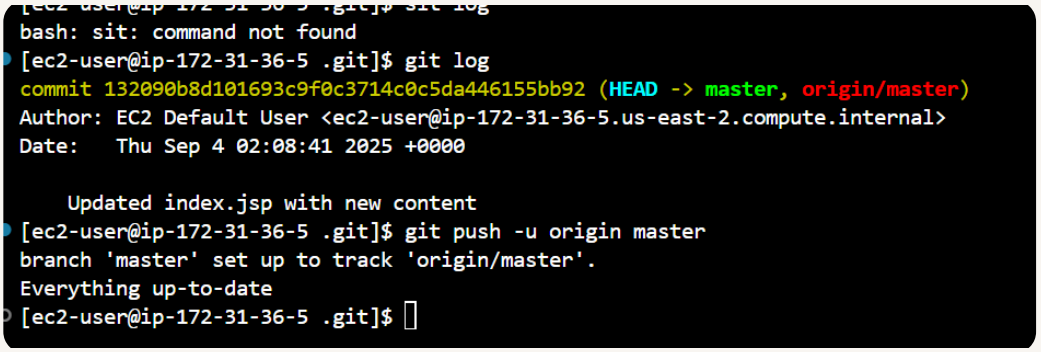
Authentication

When I commit changes to GitHub, Git asks for my credentials because Git needs to double-check that I have the right to push any changes to the remote origin your local repo is connected with.

Local Git identity

Git needs my name and email because Git needs author information for commits to track who made what change. If not set it manually, Git uses the system's default username, which might not accurately represent your identity in the project's.

Running git log showed me the changes that have been committed along with the commit ID, and also it had information like who made the changes, with the date and time.

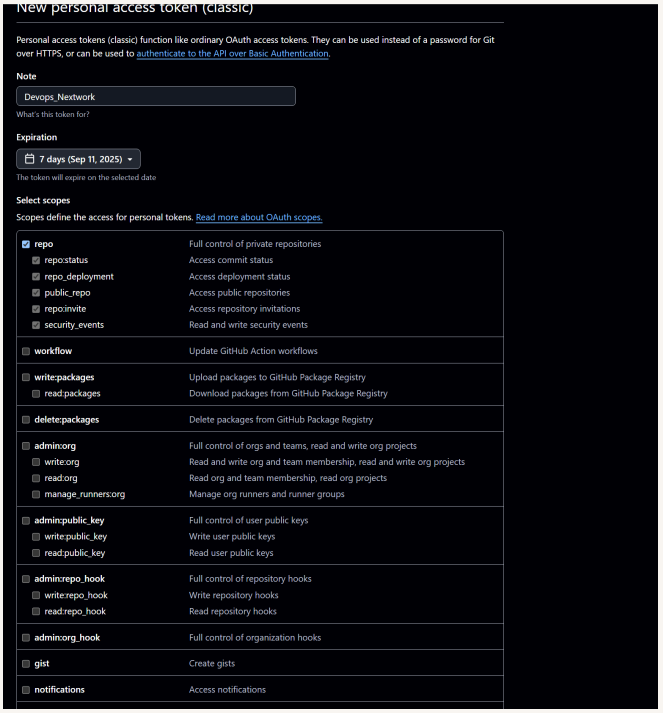


GitHub tokens

GitHub authentication failed when I entered my password because GitHub phased out password authentication to connect with repositories over HTTPS - there are too many security risks and passwords can get intercepted over the internet.

A GitHub token is A GitHub token is a secure string of characters (like a password) that GitHub generates for to authenticate and interact with GitHub services. I'm using one in this project to gain control.

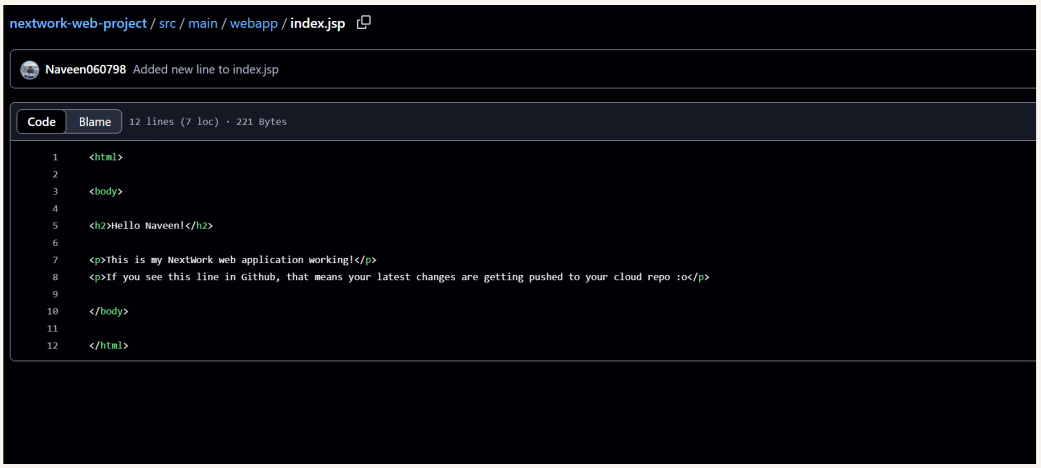
I could set up a GitHub token by going to the settings page in GitHub> developer options> select Tokens> generate Token(classic)>Token expiration in 7days>select repo>and generate token.


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
Devops_Nextwork
What's this token for?
Expiration
7 days (Sep 11, 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
workflow
Update GitHub Action workflows
write:packages
Upload packages to GitHub Package Registry
read:packages
Download packages from GitHub Package Registry
delete:packages
Delete packages from GitHub Package Registry
admin:org
Full control of orgs and teams, read and write org projects
write:org
Read and write org and team membership, read and write org projects
read:org
Read org and team membership, read org projects
manage_runners:org
Manage org runners and runner groups
admin:public_key
Full control of user public keys
write:public_key
Write user public keys
read:public_key
Read user public keys
admin:repo_hook
Full control of repository hooks
write:repo_hook
Write repository hooks
read:repo_hook
Read repository hooks
admin:org_hook
Full control of organization hooks
gist
Create gists
notifications
Access notifications

Making changes again

I wanted to see Git working in action, so I updated the index.jsp where I added a new line.

I finally saw the changes in my GitHub repo after adding the file, committing the changes, and pushing them. I could see the changes reflecting on the repository.



The place to learn & showcase your skills

Check out nextwork.org for more projects