



Connect a GitHub Repo with AWS

NE negbenosepierre@gmail.com

```
File Edit Selection View Go Run ... network-web-project [SSH: ec2-3-8-120-17.eu-west-2.compute.amazonaws.com]
EXPLORER
  NETWORK-WEB-PROJECT [SSH: EC2-3-8-120-17.eu-west-2.compute.amazonaws.com]
    src/main
    resources
    webapp
    WEB-INF
    index.jsp
    pom.xml
index.jsp
  1 <html>
  2 <body>
  3 <h2>Hello World!</h2>
  4
  5 <p>
  6
  7
  8
  9
  10
  11 Hello !
  12
  13 This is my NextWork web application working!
  14
  15
  16
  17
  18 </p>
  19 </body>
  20 </html>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS EXPOSED PORTS
Verifying : perl-lib-0.65-477.amzn2023.0.6.x86_64
Installed:
git-2.40.1-1.amzn2023.0.3.x86_64 git-core-2.40.1-1.amzn2023.0.3.x86_64 git-core-doc-2.40.1-1.amzn2023.0.3.noarch
perl-Error-1.0.1-1029-5.amzn2023.0.2.noarch perl-File-Find-1.37-477.amzn2023.0.6.noarch perl-Git-2.40.1-1.amzn2023.0.3.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-172-31-27-158 network-web-project] git --version
git version 2.40.1
[ec2-user@ip-172-31-27-158 network-web-project]
Type here to search 6°C 11:38 22/11/2024
```



Introducing Today's Project!

What is GitHub?

GitHub is a cloud-based platform for version control and collaboration using Git. In today's project, I used GitHub to store my web app's code, track changes, and sync updates from my EC2 instance to a remote repository for easier access.

One thing I didn't expect...

the level of ease

This project took me...

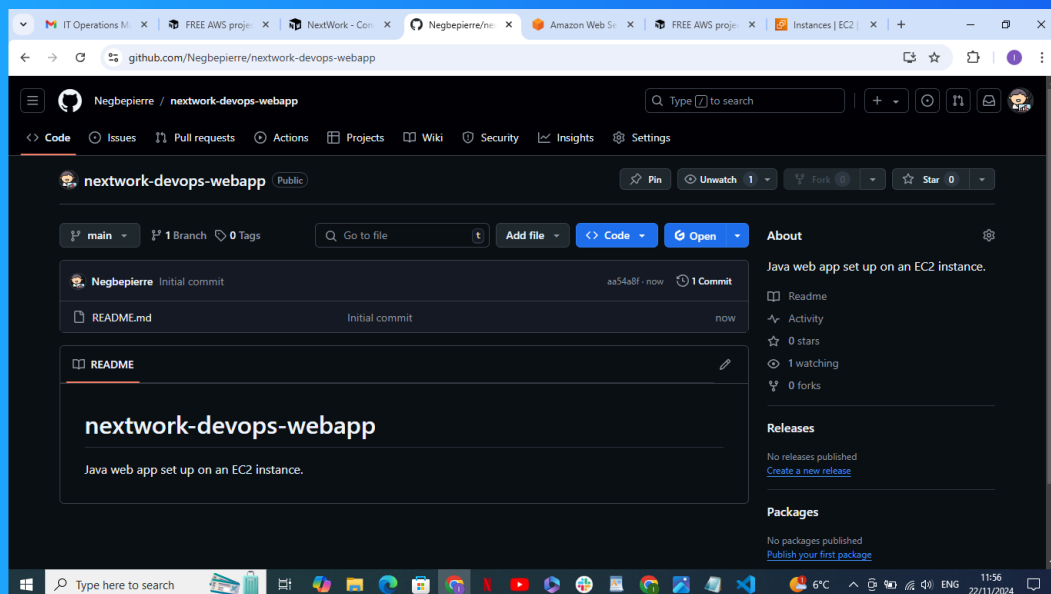
3 hours



Git and GitHub

Git is a version control system that helps developers track code changes, collaborate on projects, and manage code versions. I installed Git using the commands `sudo yum update -y` to update packages and `sudo yum install git -y` to install Git on my EC2

GitHub is a cloud-based platform for version control and collaboration. I'm using GitHub in this project to store and manage my project's code remotely, track changes, and collaborate with others easily.





My local repository

A Git repository is a storage space for your project's code, including its entire history of changes. It tracks updates, additions, and deletions, allowing developers to manage versions, collaborate properly, and revert to previous states if needed

git init is a command that initializes a new Git repository in a project folder, enabling version control for the files. I ran git init in the nextwork-web-project directory to start tracking changes and manage the project's version history.

After running git init, the response from the terminal was about setting the default branch. A branch in Git is a lightweight, movable pointer to a commit, allowing you to work on different versions of a project independently

The screenshot shows a Visual Studio Code editor window with a dark theme. The Explorer panel on the left shows a project structure for 'NEXTWORK-WEB-PROJECT' with folders 'src', 'resources', and 'webapp', and files 'WEB-INF', 'index.jsp', and 'pom.xml'. The main editor area displays the content of 'index.jsp', which is an HTML file with a body containing 'Hello World!'. Below the editor, the TERMINAL panel is open, showing the output of the 'git init' command. The output includes several hints about the default branch name and the command to create a new branch. The terminal text is as follows:

```
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/
```




To push local changes to GitHub, I ran three commands

git add

The first command I ran was `git add ..` This command moves changes from the working directory to the staging area, marking files for the next commit. A staging area is where changes are prepared before being saved in Git's history as a new snapshot.

git commit

The second command I ran was `git commit -m "Updated index.jsp with new content"`. Using `-m` means adding a message to describe the changes, creating a snapshot of the staged files in Git's history. It helps track why and what updates were made.

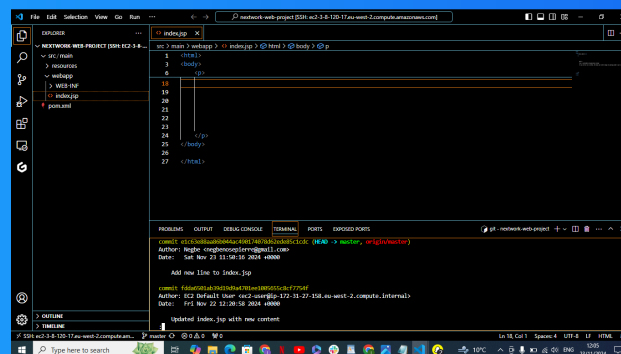
git push

The third command I ran was `git push -u origin master`. Using `-u` means setting an upstream branch, so Git remembers to push future changes to the master branch of the remote repository (origin) by default. This simplifies future push operations.



Local Git identity

Running `git log` showed me the commit history of my project, including details like commit hashes, author names, email addresses, timestamps, and commit messages. It displayed the progression of changes, such as adding new lines or updating the index.



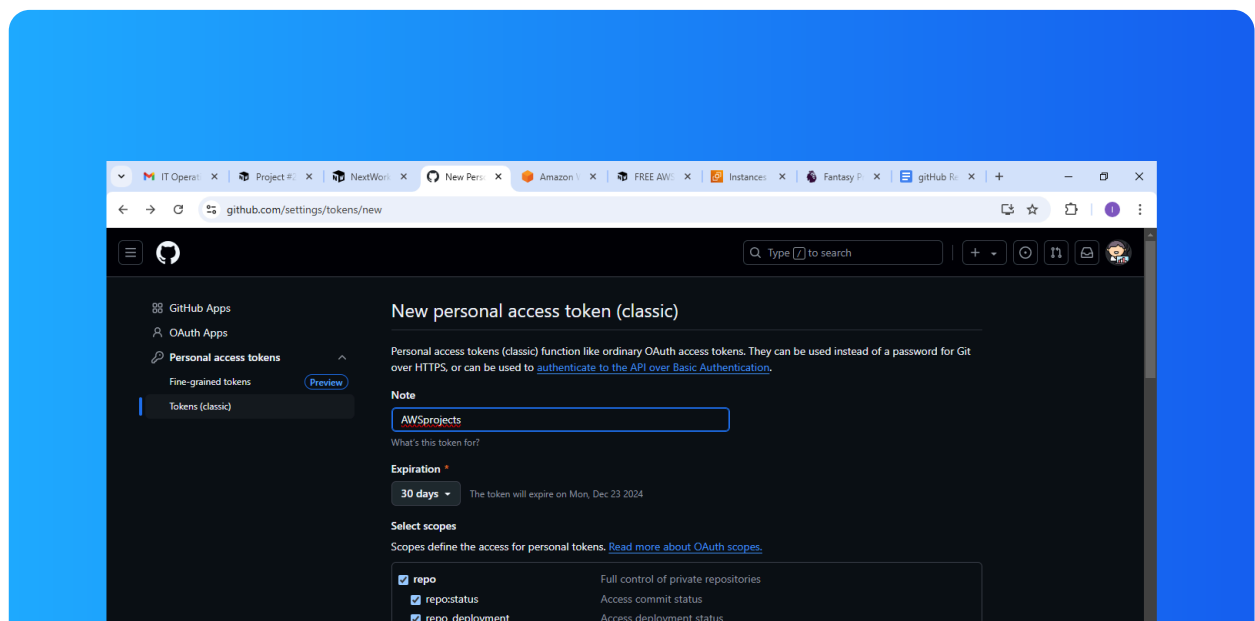


GitHub tokens

GitHub authentication failed when I entered my password because GitHub no longer allows password-based authentication for Git operations. Instead, it requires a personal access token

A GitHub token is a secure, auto-generated key used for authenticating Git operations instead of passwords. I'm using one in this project because GitHub no longer supports password authentication for increased security, ensuring safe repository acces

I could set up a GitHub token by logging into my GitHub account, navigating to Settings > Developer settings > Personal access tokens, and generating a new token with the required scopes like repo access. I used this token for secure authentication





Making changes again

I wanted to see Git working in action, so I updated the index.jsp file in my nextwork-web-project by adding a new line. I couldn't see the changes in my GitHub repo initially because the changes weren't committed and pushed from my local repository

I finally saw the changes in my GitHub repo after committing the updates to the index.jsp file using git commit and pushing the changes with git push. These commands sent the modified files from my local repository to the remote GitHub repository.

The screenshot shows a web browser displaying a GitHub repository page for the file `index.jsp`. The file is located at `src/main/webapp/index.jsp` in the `master` branch. The code content is as follows:

```
1 <html>
2
3 <body>
4   <h2>Hello World!</h2>
5
6   <p>
7
8
9
10
11
12     Hello !
13
14     This is my NextWork web application working!
15
16     If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :o
17
18
19
20
21
22
23   </p>
24 </body>
25
26 </html>
```

The browser's address bar shows the URL: `github.com/Negbepierre/nextwork-devops-webapp/blob/master/src/main/webapp/index.jsp`. The Windows taskbar at the bottom indicates the time is 11:53 on 23/11/2024.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

