



Connect a GitHub Repo with AWS

MA

Harish madhavan J U

```
Verifying      : perl-lib-0.65-477.amzn2023.0.7.x86_64
Installed:
git-2.47.1-1.amzn2023.0.3.x86_64          git-core-2.47.1-1.
perl-File-Find-1.37-477.amzn2023.0.7.noarch perl-Git-2.47.1-1.

Complete!
[ec2-user@ip-172-31-12-185 nextwork-web-project]$ git --version
git version 2.47.1
[ec2-user@ip-172-31-12-185 nextwork-web-project]$ █
```

Introducing Today's Project!

Today I will be setting up my git and github and connect my java web project with github repo so that on making changes to code in web app it automatically updates in github.

Key tools and concepts

Services I used where git, github, EC2, IAM. Key concepts I learnt include Personnel access token, local repo, remote repo, branch.

Project reflection

The project took me approximately 3 hrs , the most challenging part was ensuring that VScode stays connected with the instance . It was most rewarding to connect a local repo and remote repo.

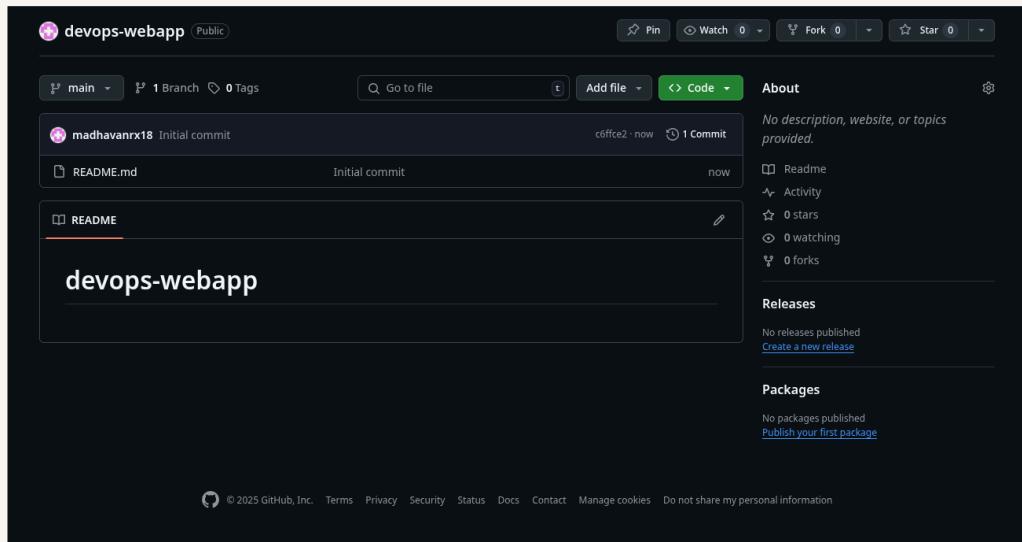
I did this project to set up a CI/CD pipeline on my own.

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 the next day.

Git and GitHub

Git is a version control system. I installed Git using the commands sudo dnf update -y which updates all the dependencies in the instance and sudo dnf install git -y which installs git in our instance and I verified it using git --version.

Github is a cloud based storage service for storing different version of our project and I'm using Github in this project because it's a user friendly way to track changes in the code using Git, Github being a cloud service lets you access your files from anywhere in the world and eases collaboration.



My local repository

A Git repository is a folder containing files and subfolders related to our web application and version history. Since repos are hosted in the cloud it allows you to collaborate with engineers all over the world.

git init is a command that initializes the pwd as a git repository where the changes made, files and subfolders are visible to you alone . I ran git init in my VS code terminal connected with EC2 instance.

After running git init, the response from the terminal was a bunch of yellow text saying that your repo is created with a main branch called master you can rename it as you wish. A branch in Git can be understood as parallel versions of the same project, developers used to create a separate branch for developing and testing new features if no issue rise it is then merged with the master branch.

```
[ec2-user@ip-172-31-12-185 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-172-31-12-185 nextwork-web-project]$ ]
```

To push local changes to GitHub, I ran three commands

git add

The first command I ran was `git add .`, which pushes all the changes in the repo to the staging area. The staging area (aka index) is a place where Git gathers changes before committing them. It allows you to review and be selective with what changes you want to commit to the repository.

git commit

The second command I ran was `git commit -m "first commit"` which saves all the changes listed in the staging area as a new version. Using `-m` means you leave a message of what the commit is about.

git push

The third command I ran was `git push -u origin master`. This command uploads the committed changes from your local repository to the remote repository origin, on the master branch. Using `-u` flag sets the origin/master branch as the default upstream branch for our local master branch. This means that in the future, we can simply use `git push` or `git pull` without specifying the remote and branch every time.

Authentication

When I commit changes to GitHub, Git asks for my credentials because to check whether the user is authorised to perform the commit. If this is not implemented anyone can perform any type of actions.

Local Git identity

Git needs my name and email because it need information about the author who makes changes to the repo if not set manually.

Running git log showed me that github stores the commit made to a username called 'EC2 default user' instead of our own details.

```
[ec2-user@ip-172-31-12-185 nextwork-web-project]$ git log
commit ceb4f2324c6e5fbde97bd7729997aa07048865d4 (HEAD -> master, origin/master)
Author: EC2 Default User <ec2-user@ip-172-31-12-185.ap-south-1.compute.internal>
Date:   Thu Jun 19 10:39:30 2025 +0000

    first commit
[ec2-user@ip-172-31-12-185 nextwork-web-project]$
```

GitHub tokens

Github authentication failed when I entered my password because password based authentication was removed in 2021 due to the method being too risky, more prone to phishing and more secure ways have been established.

A GitHub token is form of authentication based on something you know, its kind of a temporary password used for access. I'm using one in this project because it lets me safely authenticate from the EC2 instance.

I could set up a GitHub token by navigating to Devoleper settings and create a token with necessary permissions with an expiry of 7 days.

Note

`linux-general-purpose`

What's this token for?

Expiration

This token expires *on Sat, Jul 19 2025*. To set a new expiration date, you must [regenerate the token](#).

Select scopes

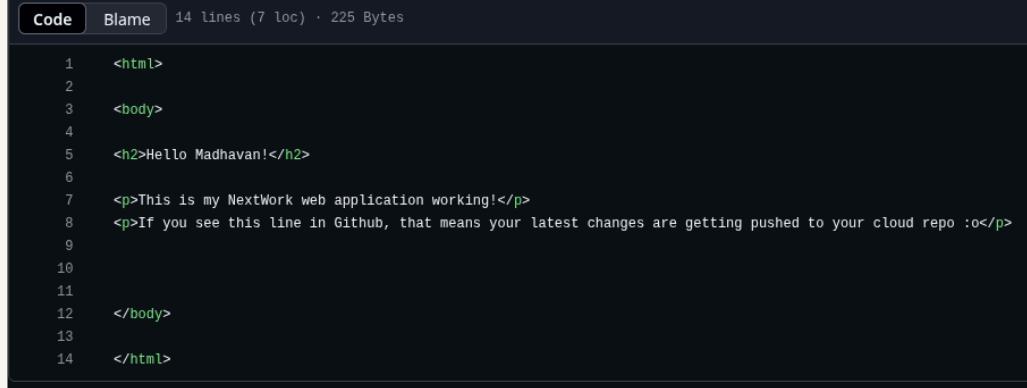
Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> <code>repo:status</code>	Access commit status
<input type="checkbox"/> <code>repo_deployment</code>	Access deployment status
<input type="checkbox"/> <code>public_repo</code>	Access public repositories
<input type="checkbox"/> <code>repo:invite</code>	Access repository invitations
<input type="checkbox"/> <code>security_events</code>	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> <code>read:packages</code>	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> <code>write:org</code>	Read and write org and team membership, read and write org projects
<input type="checkbox"/> <code>read:org</code>	Read org and team membership, read org projects
<input type="checkbox"/> <code>manage_runners:org</code>	Manage org runners and runner groups

Making changes again

I wanted to see Git working in action, so I made change to the index.jsp file ,I couldn't see the changes in my GitHub repo initially because i haven't added, committed and pushed the changes.

I finally saw the changes in my GitHub repo after running the three command to add, commit and push.



The screenshot shows a GitHub code editor interface. At the top, there are two tabs: "Code" (which is selected) and "Blame". Below the tabs, it says "14 lines (7 loc) · 225 Bytes". The code itself is a simple HTML file:

```
1 <html>
2
3 <body>
4
5 <h2>Hello Madhavan!</h2>
6
7 <p>This is my NextWork web application working!</p>
8 <p>If you see this line in Github, that means your latest changes are getting pushed to your cloud repo :o</p>
9
10
11
12 </body>
13
14 </html>
```

Setting up a README file

As a finishing touch to my Github repository, I added a README file, which is a document which contains information about what the project is, how to set up it is a valuable resource. I added a README file by executing the command touch README.md which creates a empty file called README.

My README is written in Markdown because its a language that formats your document with headings, bold text etc with just adding few extra symbols. Special characters can help you format text in markdown such as # for heading ,... etc

My README file has 6 sections that outline the technologies used , steps to set up and a conclusion on what is the outcome of the project.

The screenshot shows a GitHub repository page for 'devops-webapp'. The title bar indicates it's on the 'master' branch. The main content is a README file titled 'Java Web App Deployment with AWS CI/CD'. It starts with a welcome message: 'Welcome to this project combining Java web app development and AWS CI/CD tools!'. Below this is a 'Table of Contents' section with links to 'Introduction', 'Technologies', 'Setup', 'Contact', and 'Conclusion'. The 'Introduction' section contains text about the project's purpose and deployment pipeline. The 'Technologies' section lists 'Amazon EC2', 'VS Code', and 'GitHub' as tools used. The entire screenshot is set against a light beige background.

Java Web App Deployment with AWS CI/CD

Welcome to this project combining Java web app development and AWS CI/CD tools!

Table of Contents

- [Introduction](#)
- [Technologies](#)
- [Setup](#)
- [Contact](#)
- [Conclusion](#)

Introduction

This project is used for an introduction to creating and deploying a Java-based web app using AWS, especially their CI/CD tools.

The deployment pipeline I'm building around the Java web app in this repository is invisible to the end-user, but makes a big impact by automating the software release processes.

Technologies

Here's what I'm using for this project:

- Amazon EC2: I'm developing my web app on Amazon EC2 virtual servers, so that software development and deployment happens entirely on the cloud.
- VS Code: For my IDE, I chose Visual Studio Code. It connects directly to my development EC2 instance, making it easy to edit code and manage files in the cloud.
- Github: All my web app code is stored and versioned in this Github repository.



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

