**Connect a GitHub Repo with AWS**

**This is Project 2 of 7 Day DevOps challenge!**

In this project, I’ll learn how to store the web application’s code in a Git repository.

**what does a Git repository mean?**

Git repositories are essential in a CI/CD pipeline because they store your code safely in the cloud, and tracks every change your team makes to the code! Using a shared repository makes it much easier for engineering teams to hand off code, collaborate and update each other on changes they've made.

A diagram of a computer

AI-generated content may be incorrect.

In this, I’ll

* Set up Git and GitHub.
* Connect your web app project to a GitHub repo.
* Make changes to your web app code - and watch your GitHub repo update too.
* Set up a README file for your repo.

**Step 1**: Set Up the Web App in the Cloud, which is project 1.

**Step 2**: Install Git

As the development environment is set, the next step is to set up Git on the EC2 instance.

**What is Git?**   
Git is like a time machine and filing system for your code. It tracks every change you make, which lets you go back to an earlier version of your work if something breaks.

You can also see **who** made specific changes and **when** they were made, which makes teamwork/collaboration a lot easier.

**In this step, you're going to:**

1. Install Git on the EC2 instance.
2. Open the EC2 instance's terminal.
3. In the terminal, run these commands to install Git

sudo dnf update -y

sudo dnf install git -y

(or)

sudo yum install git -y

A screen shot of a computer

AI-generated content may be incorrect.

**What do these commands do?**

* sudo dnf update -y tells your EC2 instance to find all the latest updates of software it has (e.g. Java, Maven) and install them straight away.
* sudo dnf install git -y installs Git on your EC2 instance.  
  It's best practice to update your existing software before installing new ones, just in case there are compatibility issues between new and old software.

1. Verify the installation.

git --version

A black background with white text

AI-generated content may be incorrect.

**Step 3:** Set up GitHub

As Git is installed, next will be set up GitHub.

**What is GitHub?**

GitHub is a place for engineers to store and share their code and projects online. It's called **Git**Hub because it uses Git to manage your projects' version history.

In this step, I’ll

1. Set up a GitHub account.
2. Create a GitHub repository.
3. Log in to GitHub

**What is the difference between Git & Github?**  
If Git is the tool for tracking changes, think of GitHub as a storage space for different version of your project that Git tracks. Since GitHub is a cloud service, it also lets you access your work from anywhere and collaborate with other developers over the internet.

**Why would I use Github? Isn't the code in my EC2 instance already in the cloud?**   
Even though your code is on a cloud server like EC2, GitHub helps you use Git and see your file changes in a more user-friendly way. It's just like how using an IDE (VSCode) makes editing code easy.

GitHub is also especially useful in situation where you're working in teams and need to share your updates and reviews to a shared code base.

**Set up a new repository**

Nice, you're ready to set up a new repository on GitHub!

**What is a repository?**  
To store your code using Git, you create repositories (aka 'repos'), which are folders that contain all your project files and their entire version history. Hosting a repo in the cloud, like on GitHub, means you can also collaborate with other engineers and access your work from anywhere.

* After signing in to GitHub, click on the **+** icon next to your GitHub profile icon at the top right hand corner.
* Select **New repository.**
* Select **Create repository.**
* This loads up a new page where you can create a repository.
* Under **Owner**, click on the **Choose an owner dropdown** and select your GitHub username.
* Under **Repository name**, enter project name
* For the **Description**, enter Java web app set up on an EC2 instance.
* Choose your visibility settings. We'd recommend selecting **Public** to make your repository available for the world to see.

A screenshot of a computer

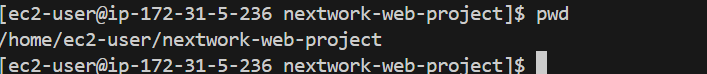
AI-generated content may be incorrect.

**Step 4:** Commit and Push changes to GitHub

In this step, I’ll

1. Set up a local git repo in the web app folder
2. Connect the local repo with the GitHub repo

* Head to VS code and check whether its still connected to the EC2 instance or not.
* I have used pwd command to check if I’m in the right folder or not



What does pwd do?

**pwd** stands for **print working directory**, and this command asks your server "where am I right now?" The terminal will show you the exact location of the directory (folder) you're in.

If its not in the right folder then cd command can be used to navigate to the right folder.

* Now let's tell Git that we'd like to track changes made inside this project folder.

git init

💡 **What does git init do?**  
To start using Git for your project, you need to create a **local repository** on your computer.

When you run git init inside a directory e.g. nextwork-web-project, it sets up the directory as a local Git repository which means changes are now tracked for version control.

💡 **What's a local repository?**  
The local repository is where you use Git **directly on your own EC2 instance.** The edits you make in your local repo is only visible to you and isn't shared with anyone else yet

This is different to the GitHub repository, which is the remote/cloud version of your repo that others can see.

A screenshot of a computer screen

AI-generated content may be incorrect.

* Head back to the GitHub repository's page.
* In the blue section of the page titled **Quick setup — if you’ve done this kind of thing before**, copy the HTTPS URL to your repository page. It will look like https://github.com/username/project.git

A screenshot of a computer

AI-generated content may be incorrect.

Now let's connect the local project folder with the Github repo!

* Head back to the terminal in VSCode.
* Run this command. Don't forget to replace **[YOUR GITHUB REPO LINK]** with the link you've just copied.

git remote add origin [YOUR GITHUB REPO LINK]

A screen shot of a computer program

AI-generated content may be incorrect.

Next, we'll save our changes and push them into GitHub.

* Run this command in your terminal:

git add .

* Run this command next in your terminal:

git commit -m "Updated index.jsp with new content"

git push -u origin master

A screen shot of a computer

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

* Head to your GitHub repository in your web browser.
* Refresh the page, and you’ll see your web app files in the repository, along with the commit message you wrote.

A screenshot of a chat

AI-generated content may be incorrect.