|  |  |
| --- | --- |
|  |  |
| Technical Execution Document |  |
|  |  |
|  | Project TitleDeploy React App to Elastic Beanstalk using AWS Code Pipeline |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | INSTRUCTIONSThe Technical Steps mentioned in the execution documents need to execute as it is by students who all is involved in this projects . The Technical steps involved some components which require instructor involvement as well.Some features and UI interface may change as shown this execution steps . The cost of using the AWS resources must be taken care off well before setting up the services | |  |
|  | Node.js is an open source server environment. Node.js allows you to run JavaScript on the server. Node.js is free and runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)  In this lab based project, We'll deploy a NodeJS application from GitHub out to a target environment in AWS Elastic Beanstalk to provide easy integration and deployment from dev to production.  Use a personal account or [create a new AWS account](https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/) for this lab rather than using an organization’s account to ensure that you have full access to the necessary services and do not leave behind any resources from the lab. | Image |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Table of Contents

[1. Create an AWS and GitHub account 5](#_Toc57038482)

[2. Step 2: Create Security Group 5](#_Toc57038483)

[3. Setting up Development Environment 5](#_Toc57038484)

[**1.1** **Download and Install Git for Windows** 6](#_Toc57038485)

[**1.2** **Download and Install Visual Studio Code for Windows** 11](#_Toc57038486)

[4. Prepare the Code available in Gut Hub 14](#_Toc57038487)

[5. Configure Elastic Beanstalk Application 17](#_Toc57038488)

[6. Configure Code Pipeline 19](#_Toc57038489)

[11. Verify the Updated Elastic Beanstalk Site 24](#_Toc57038490)

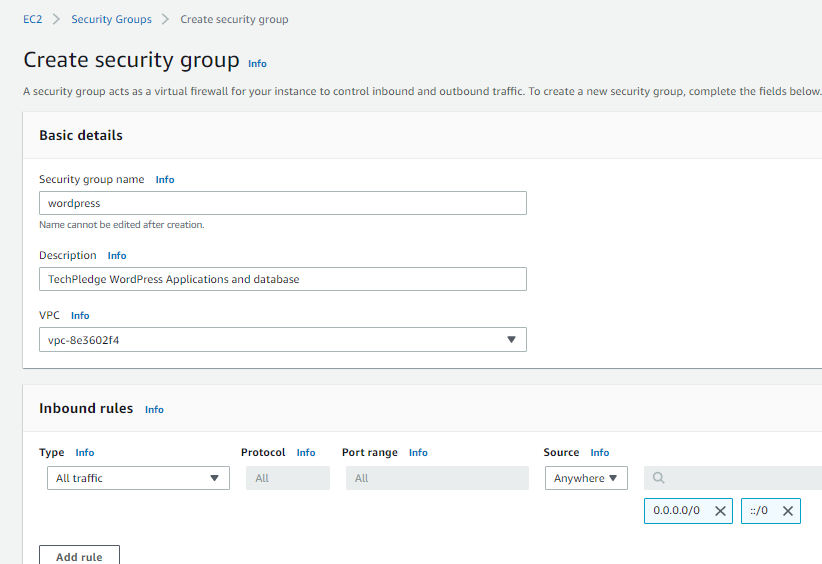
[12. Perform CI/CD 25](#_Toc57038491)

# Create an AWS and GitHub account

1. If you already have the accounts use it .

# Step 2: Create Security Group

1. Create a Security Group and Open all Traffic

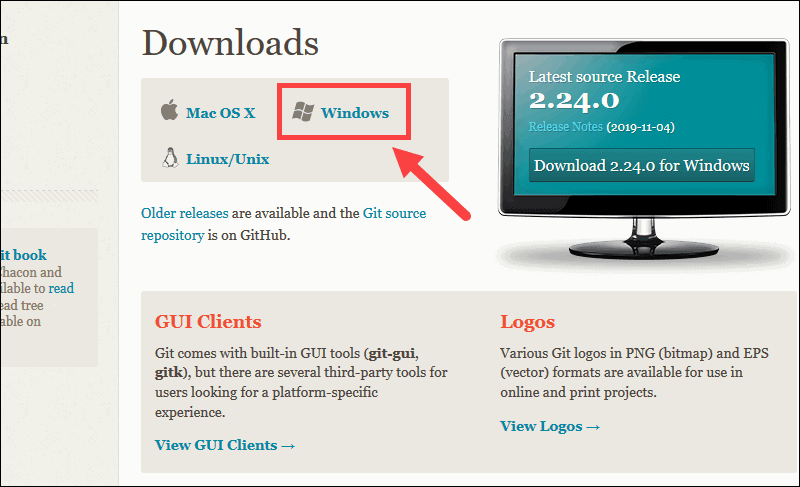


# Setting up Development Environment

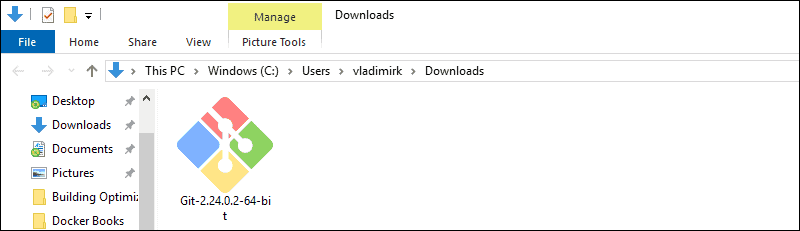
To complete this project we need to setup the local development environment where developer will create and update the code for application.

## **Download and Install Git for Windows**

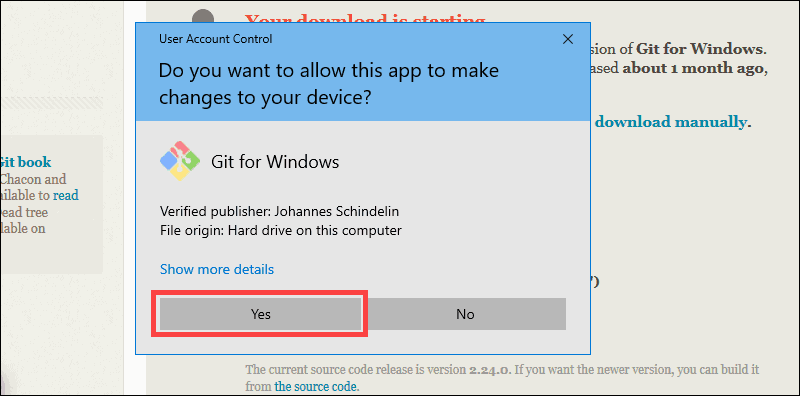
1. Browse to the official Git website: <https://git-scm.com/downloads>
2. Click the download link for Windows and allow the download to complete.



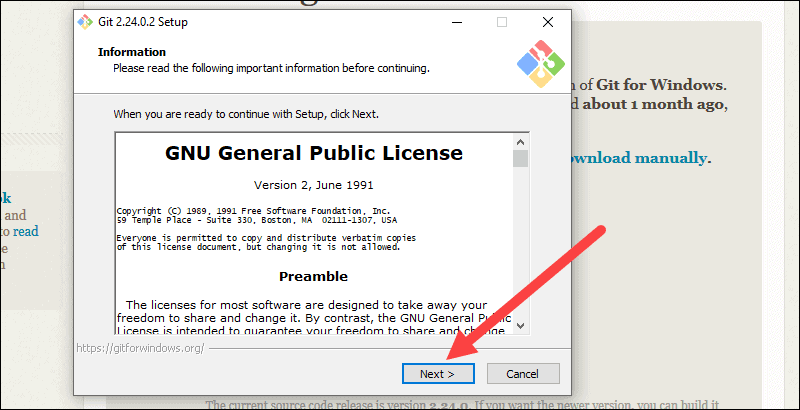
1. Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.



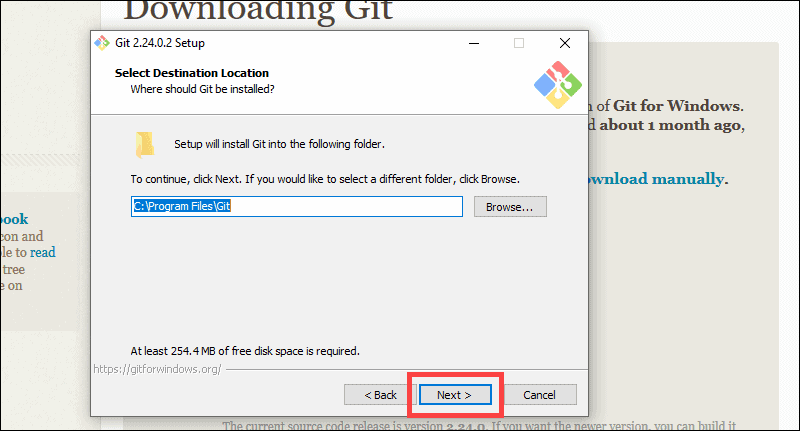
1. Allow the app to make changes to your device by clicking **Yes** on the User Account Control dialog that opens



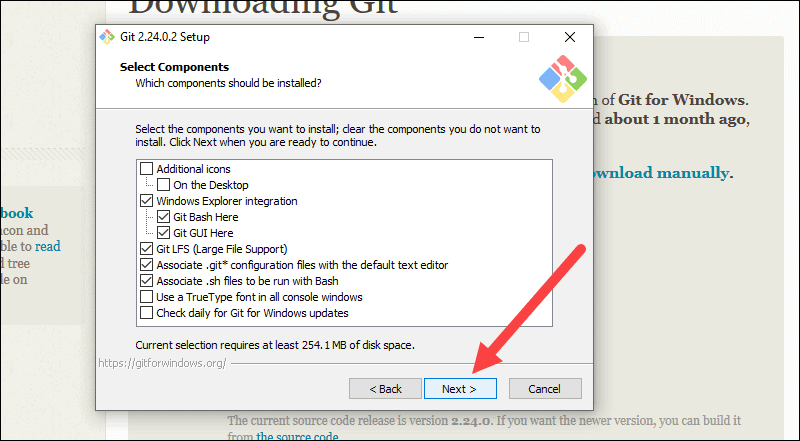
1. Review the GNU General Public License, and when you’re ready to install, click **Next**.



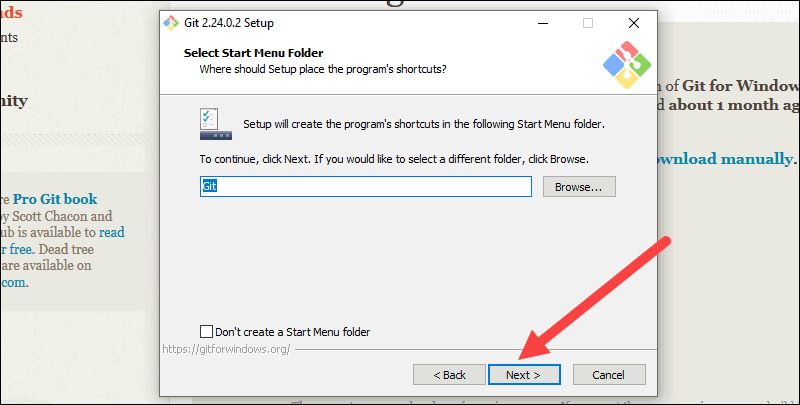
1. The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click **Next**.



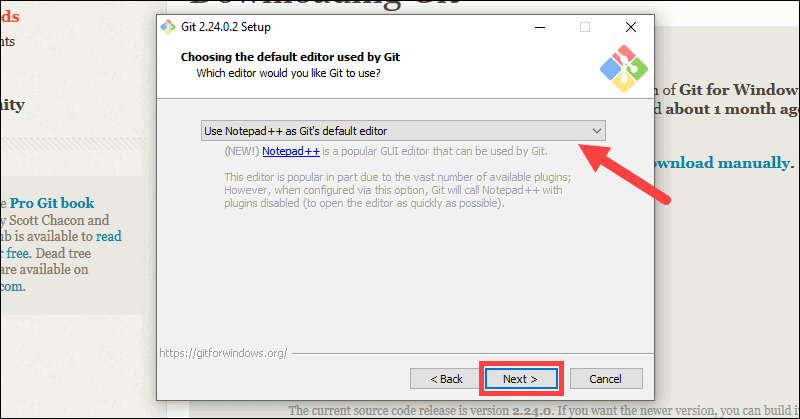
1. A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click **Next**.



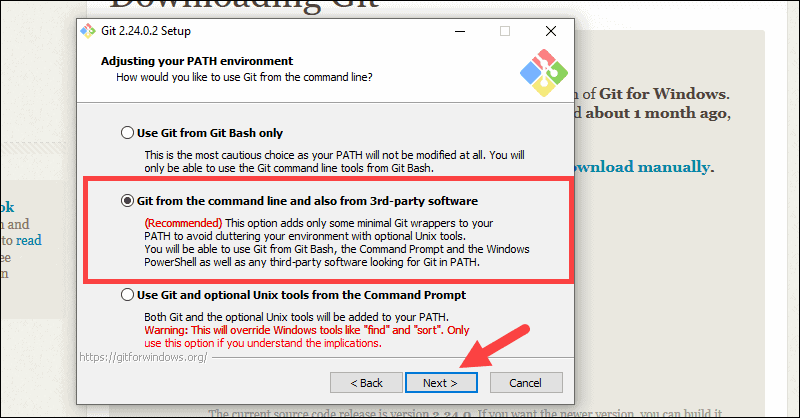
1. The installer will offer to create a start menu folder. Simply click **Next**.



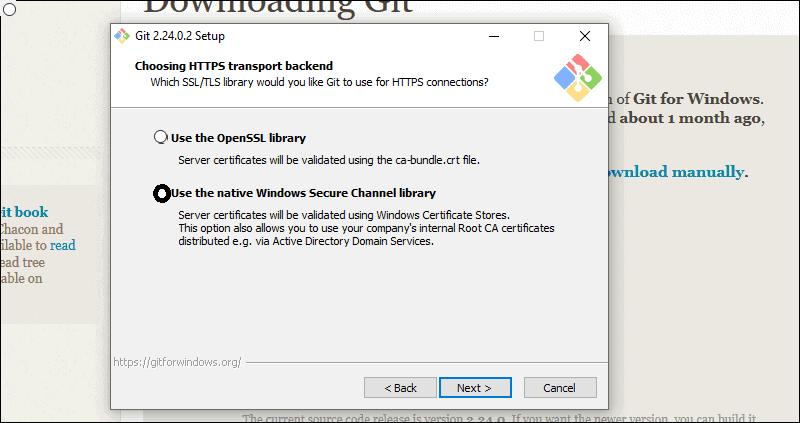
1. Select a text editor you’d like to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.



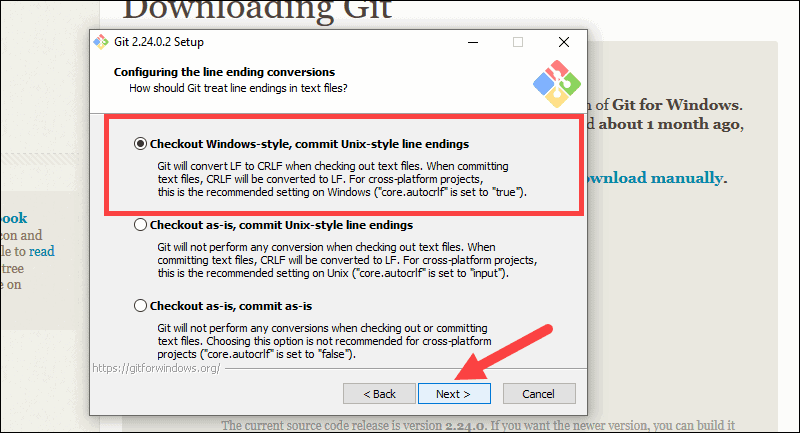
1. This installation step allows you to change the **PATH environment**. The PATH is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click **Next**.



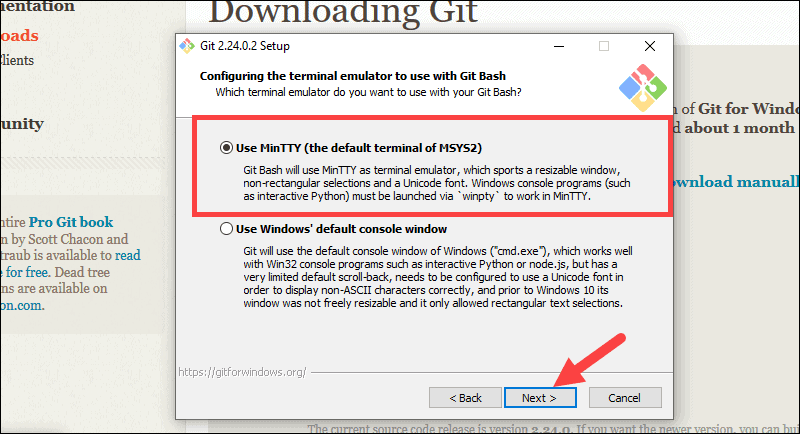
1. The next option relates to server certificates. Select Windows Store certificates. Click **Next**.



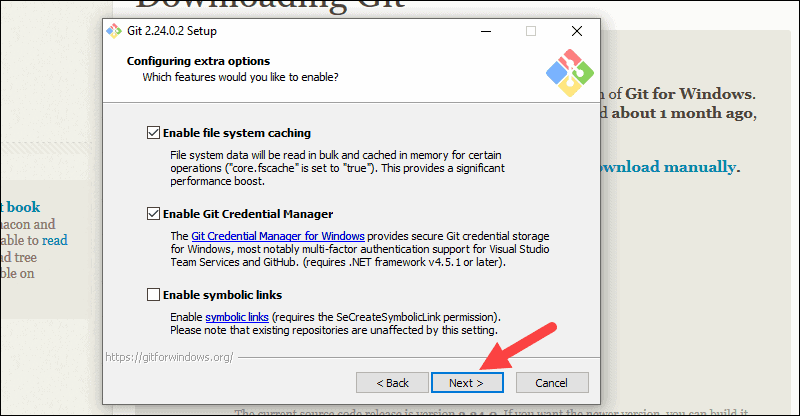
1. Leave the default selection. Click **Next**.



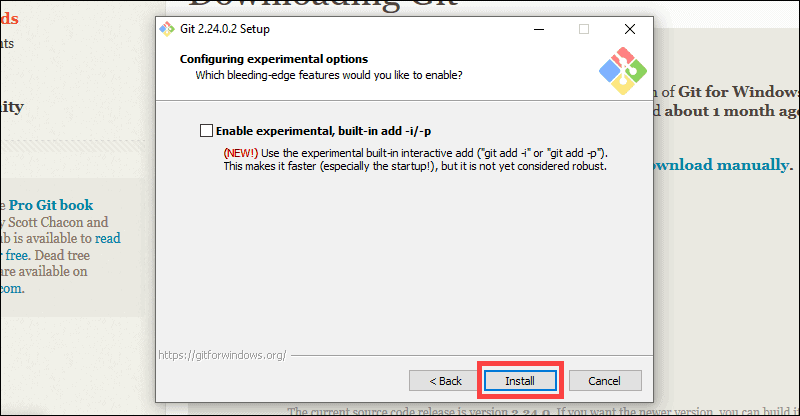
1. Choose the Windows Emulator . Click **Next**.



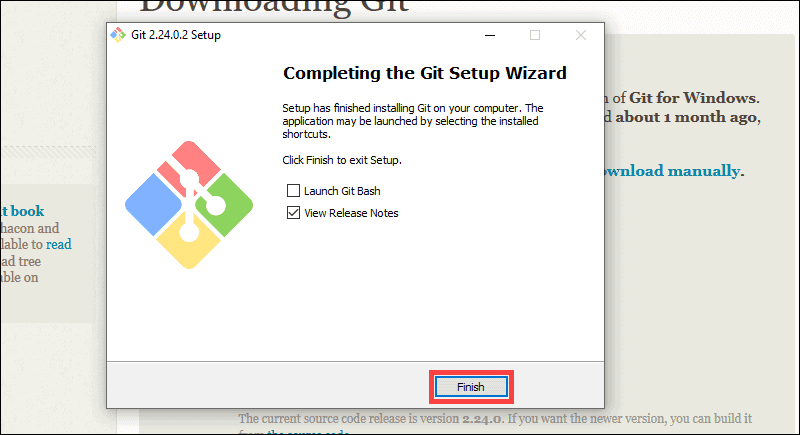
1. Click **Next**



1. Leave them experimental features unchecked and click **Install**.



1. Once the installation is complete, tick the boxes to view the Release Notes or Launch Git Bash, then click **Finish**.



Now Open the command prompt and Configure your local Git installation to use your GitHub credentials by entering the following:

git config ––global user.name “github\_username”

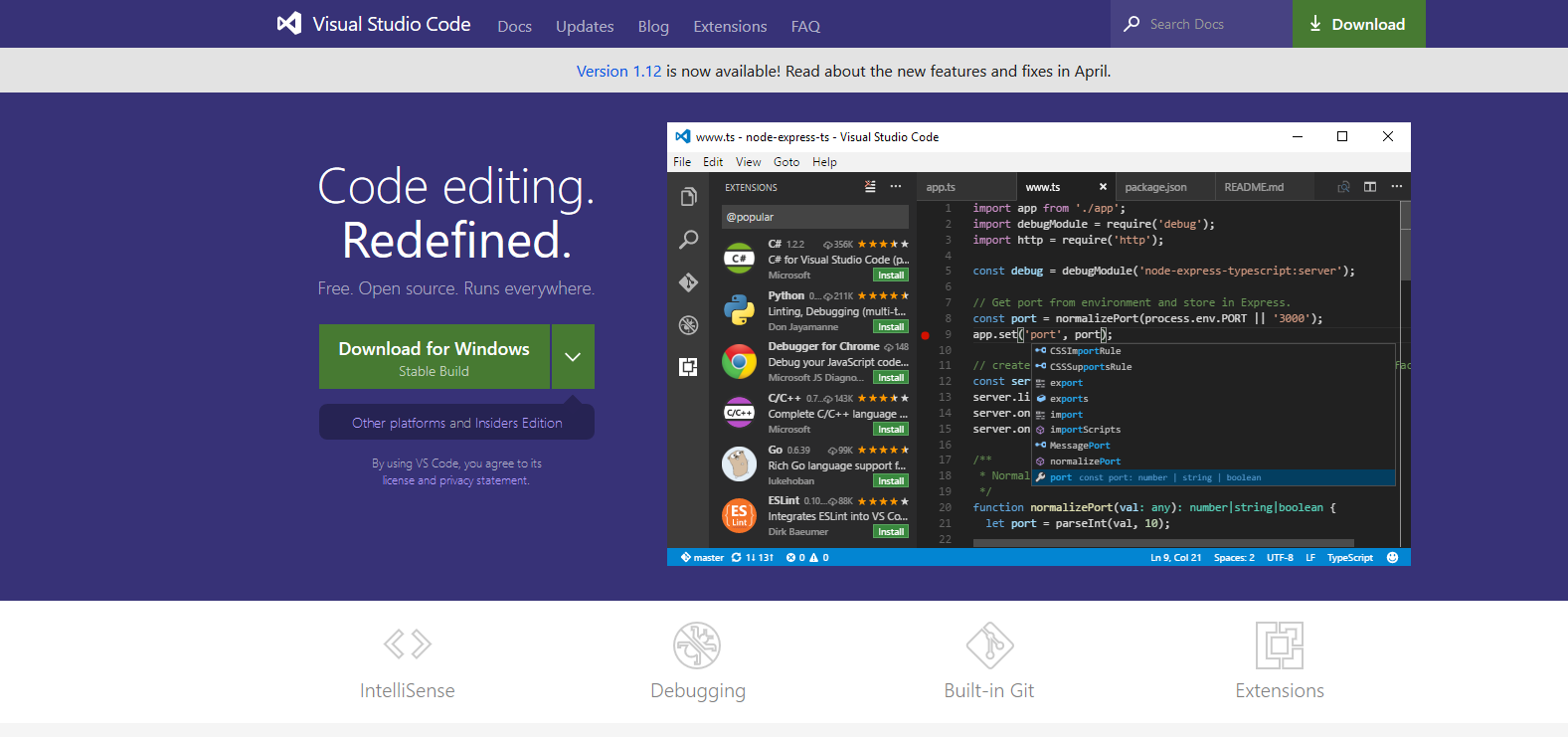
git config ––global user.email “email\_address”

## **Download and Install Visual Studio Code for Windows**

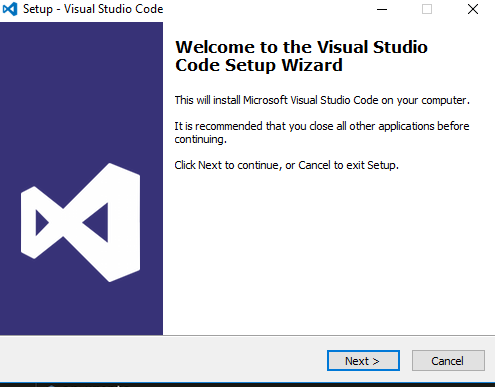
Visual Studio Code editor is an **open source** editor. So you can use it as **free** for your Ionic project development.

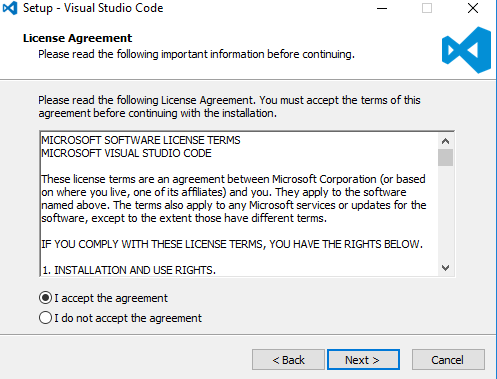
1. To download the Visual Studio Code editor, please go to the below link.

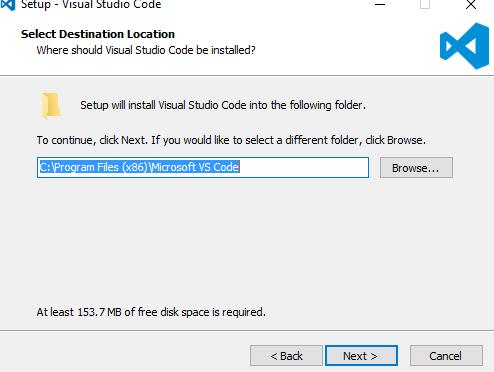
<https://code.visualstudio.com/>

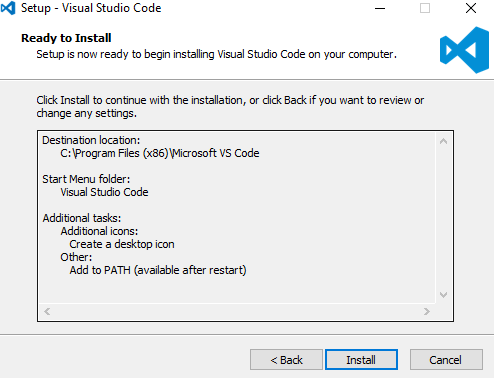


1. Click the download button to download the Visual Studio Code Editor.
2. It is the .exe file. So The installation is as usual windows standard installation. Click next-next.

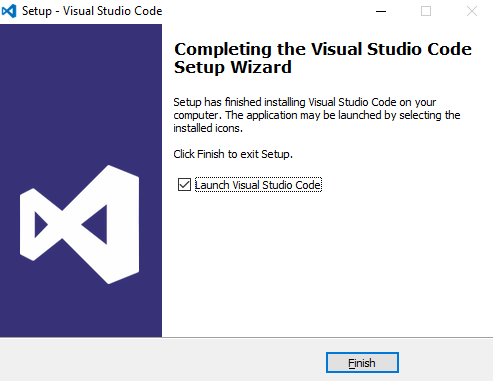






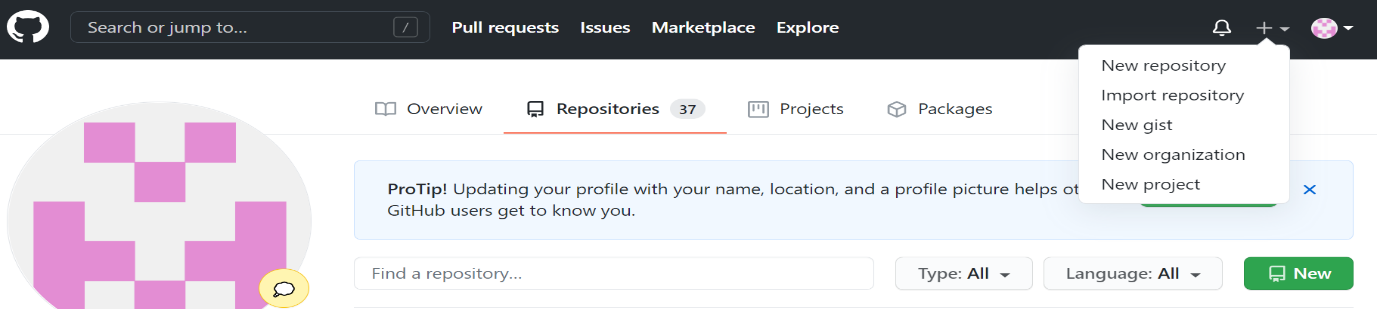


1. Once the installation is done, click the **Launch** option to open the **Visual Studio Code editor**

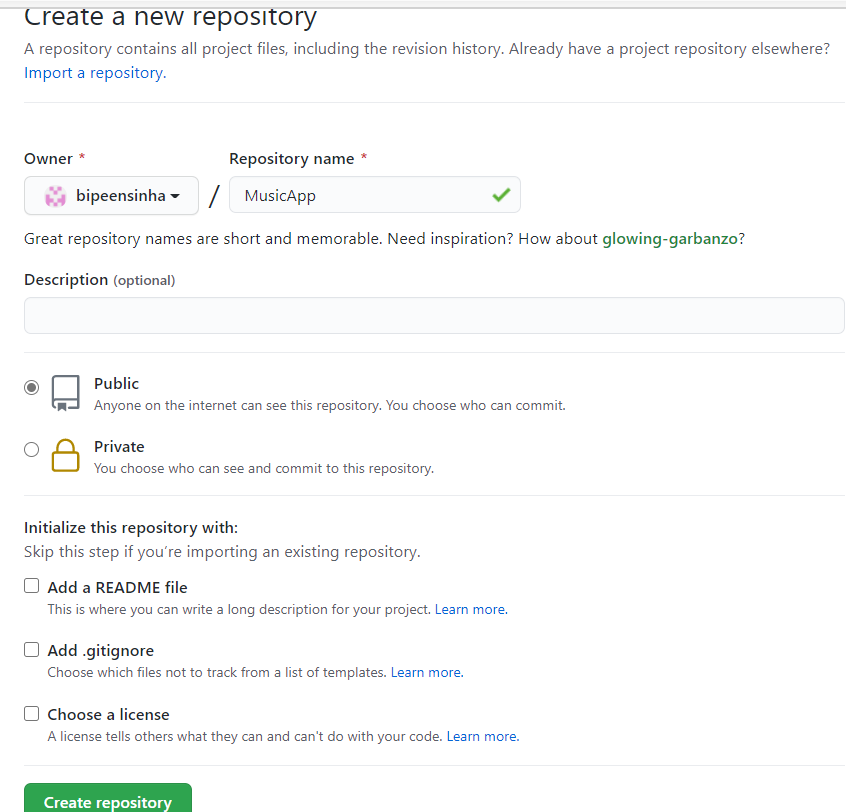


# Prepare the Code available in Gut Hub

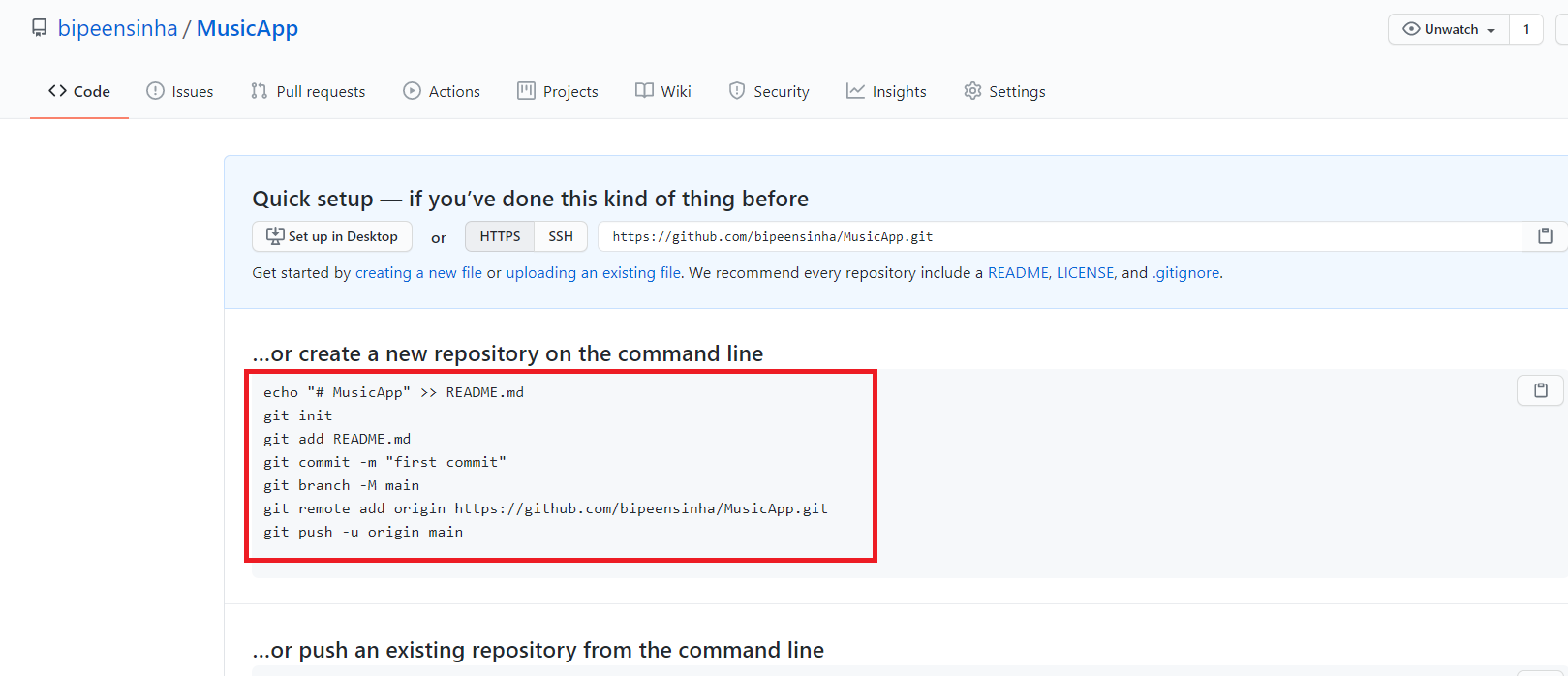
1. Click Create New Repository in git hub



1. Type name **MusicApp** , set the scope **public** but do not initialize the repo and click Create



1. You get the screen similar to this , keep note of the command

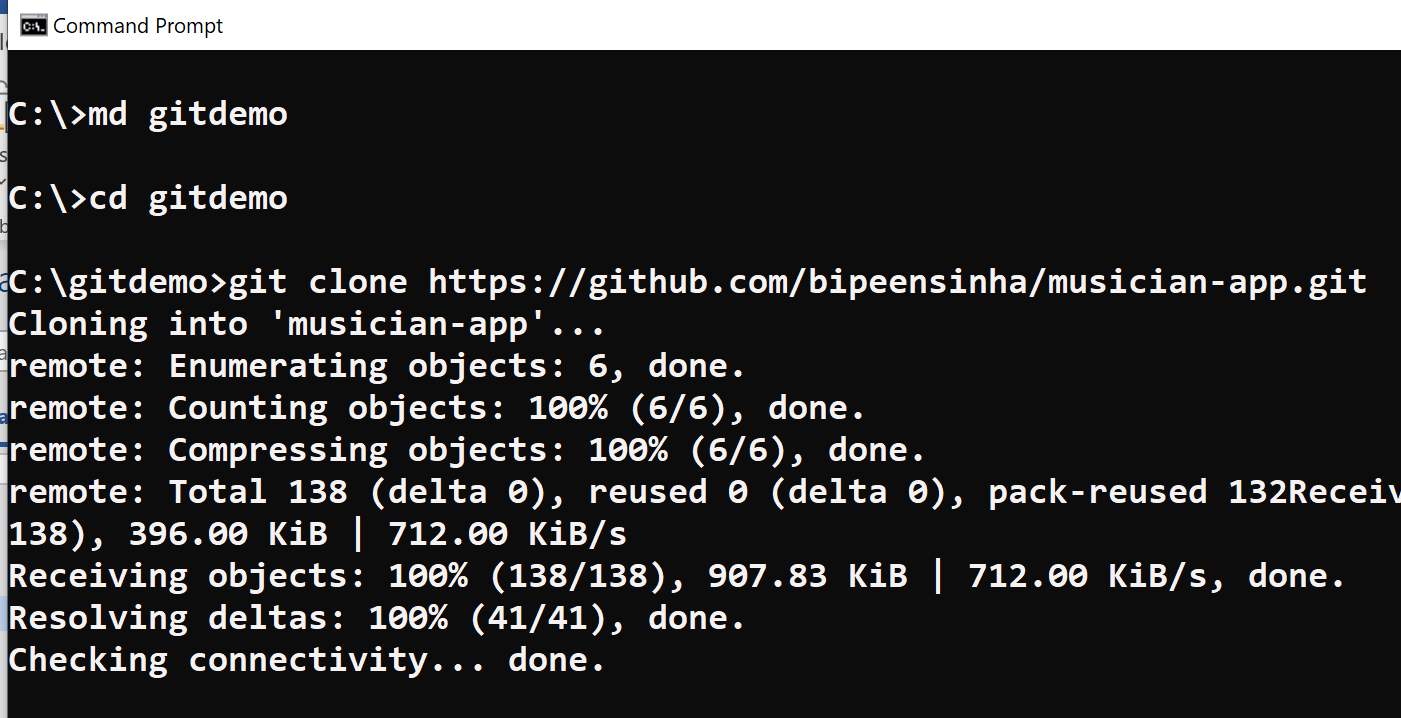


1. Now open the command prompt , create new directory **Gitdemo** and clone the Code from Repo

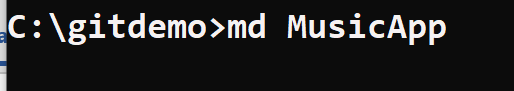
**md gitdemo**

**Cd gitdemo**

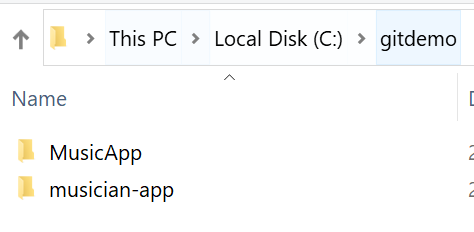
**Git clone https://github.com/MajeedAhZ/musician-app.git**



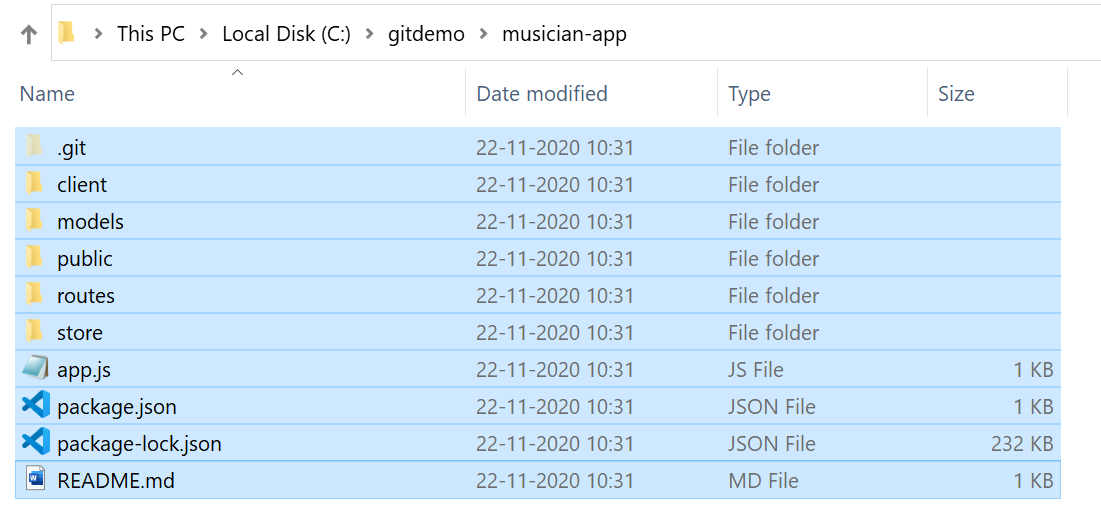
1. Create Another Directory **Music App**



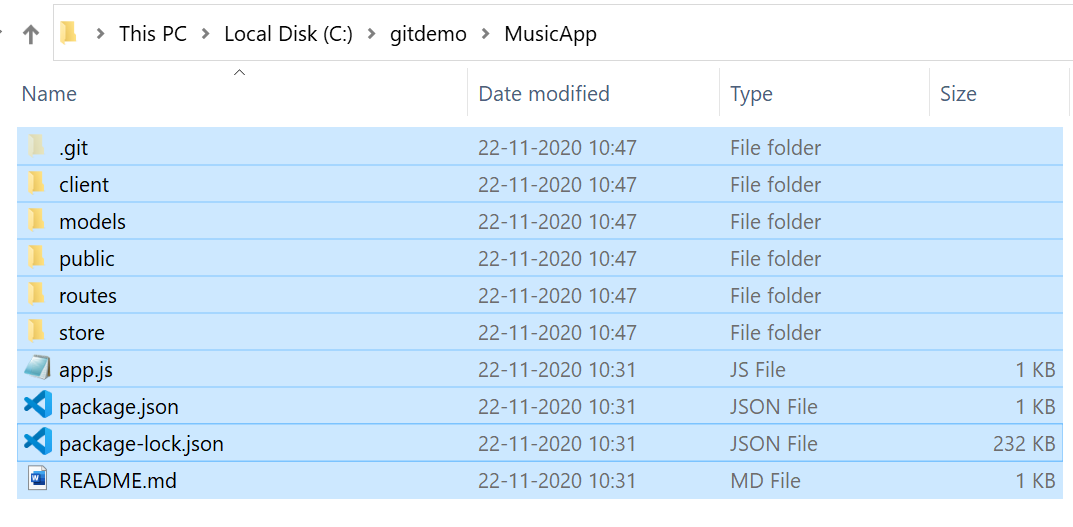
1. Now switch to desktop go to **C:\gitdemo**



1. **Copy** everything from **Musician-app**

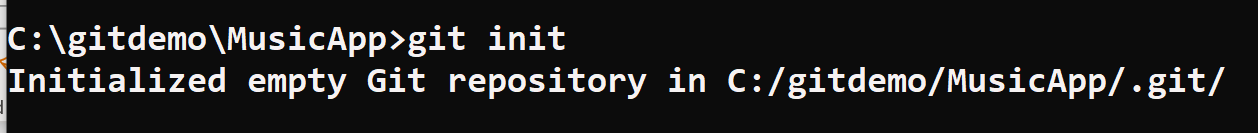


1. **Paste** it to **MusicApp** and make sure **to delete** **.git** folder



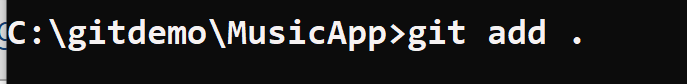
1. Now **initialize** the **git** Repository

**git init**



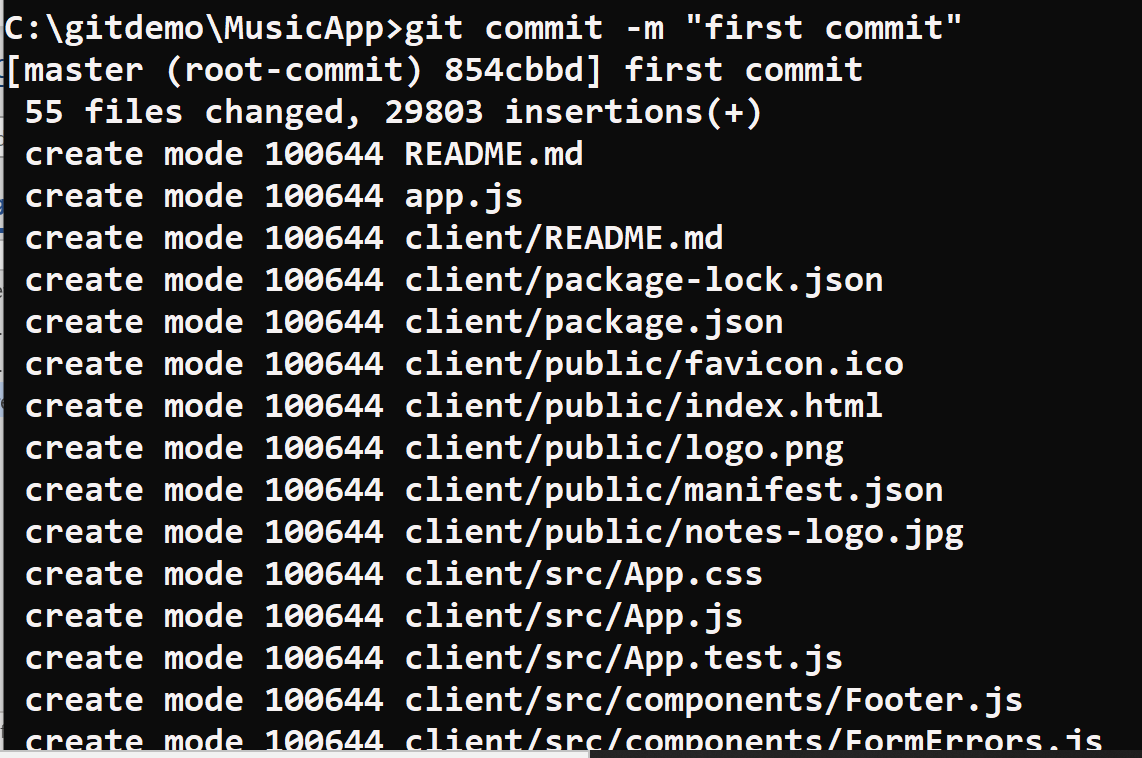
1. Type **git** add to add all the existing files and folder

**git add**



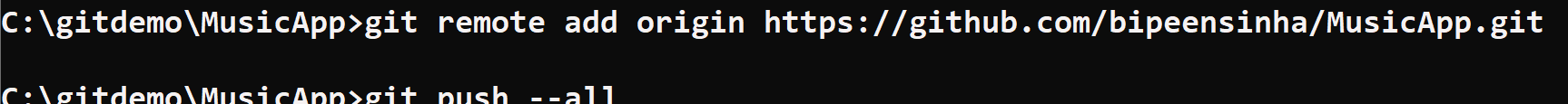
1. Type **git commit** to commit the existing files and folder to git hub

**git commit -m "first commit"**

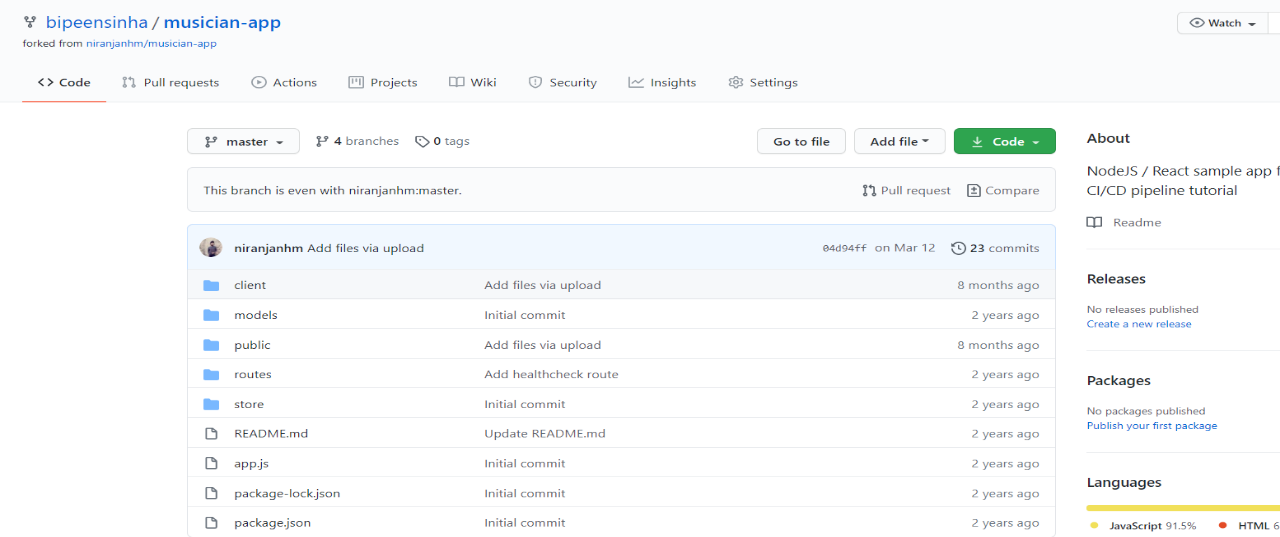


1. Add the **Github Repo**

**git remote add origin https://github.com/bipeensinha/MusicApp.git**

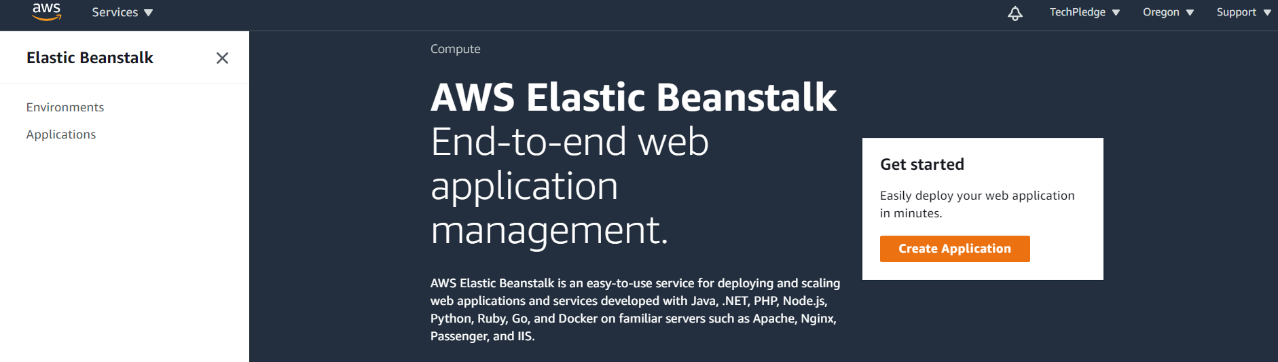


1. You Should Now see all the Code in git Hub

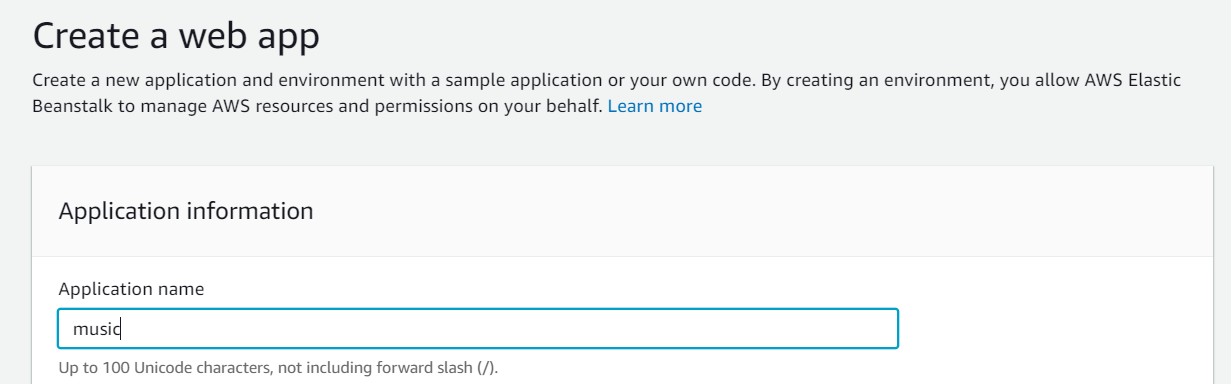


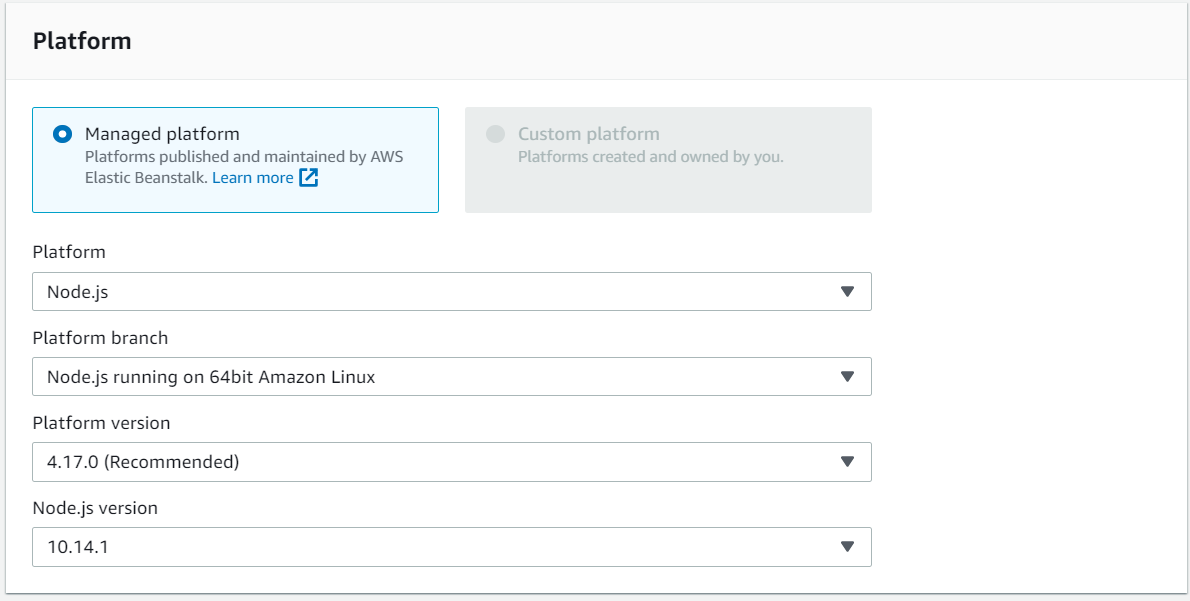
# Configure Elastic Beanstalk Application

1. Go to Elastic Beanstalk in AWS and click Create Application

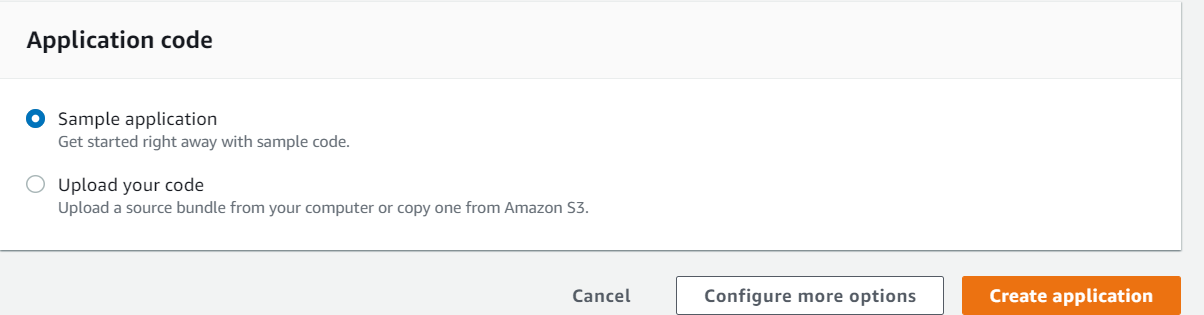


1. Type the app **name** as **Music** and in **Platform** select **Node.js** and version **10.14.1**

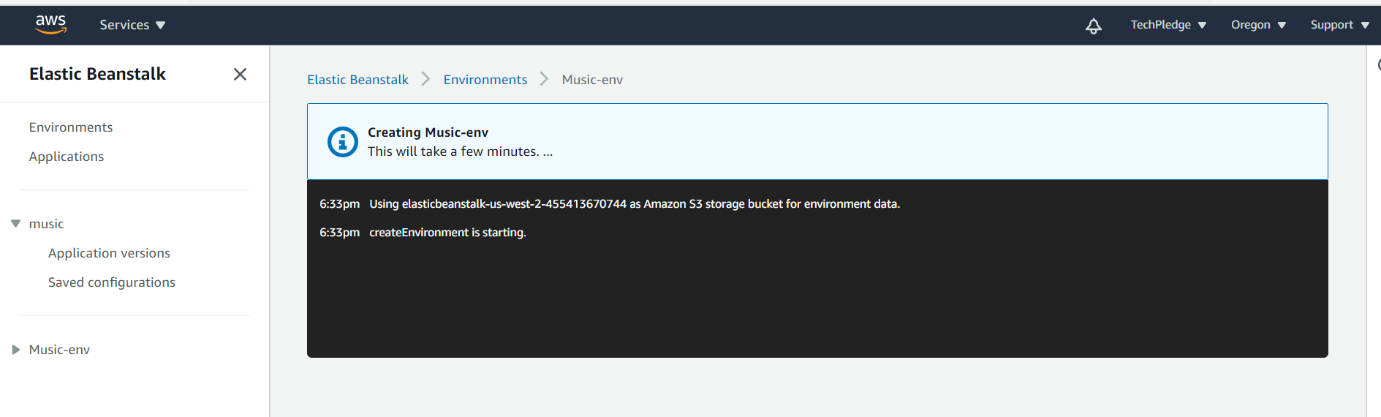




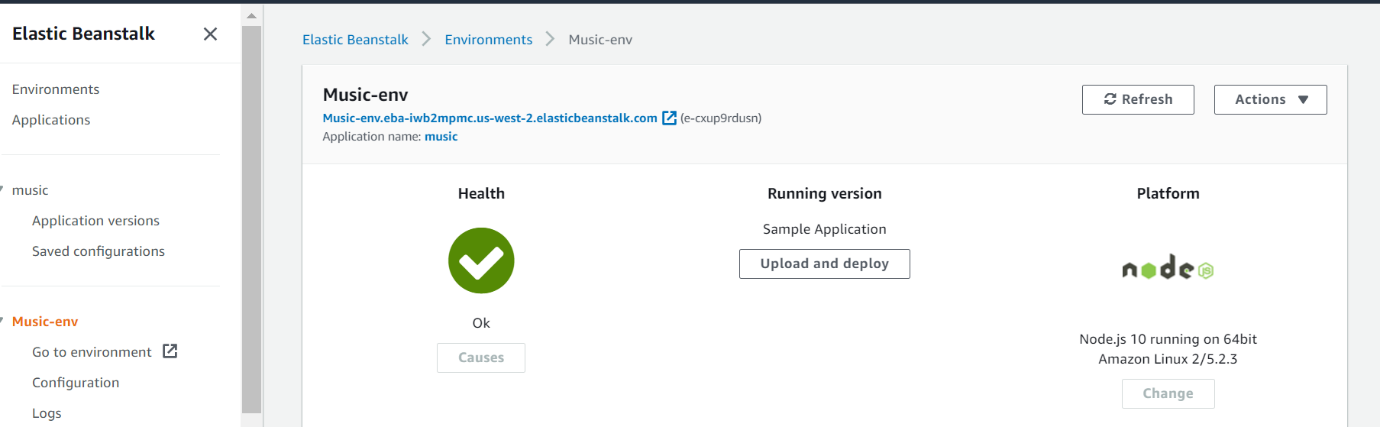
1. Click **Create Application** button



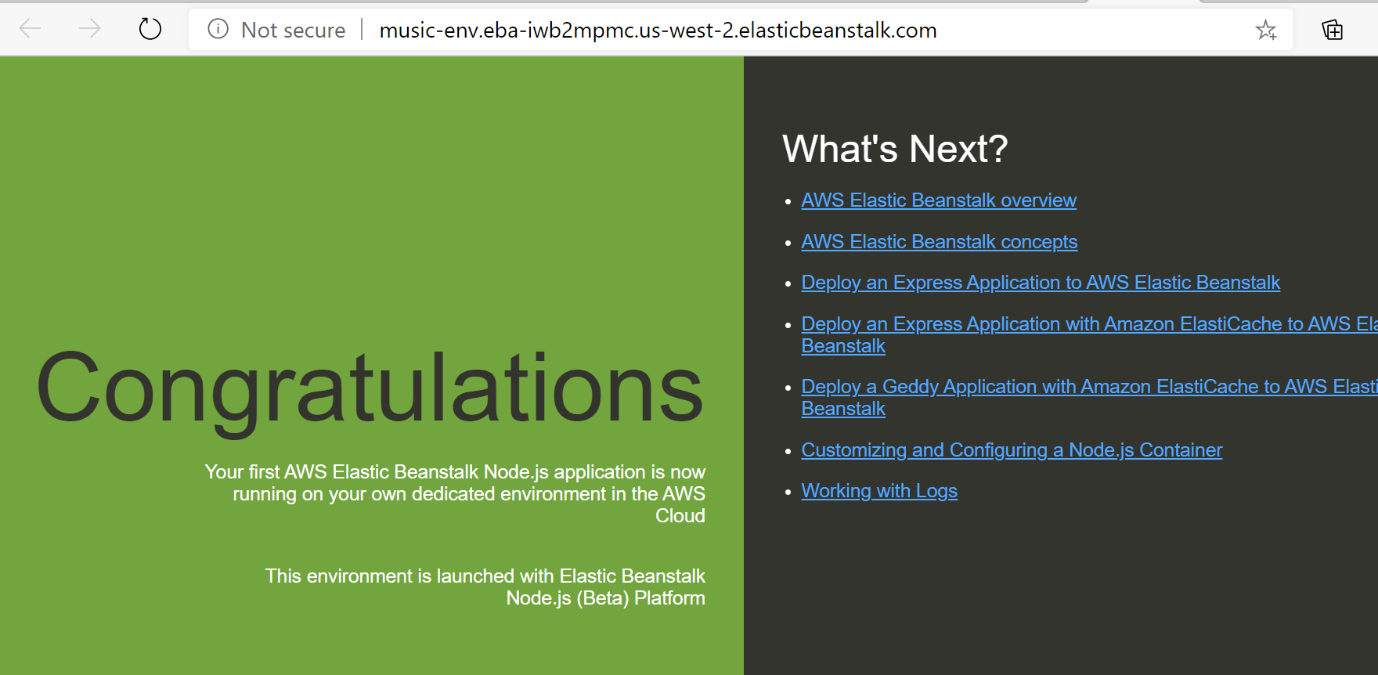
1. You can now see the Application get initialized



1. After 15 Min you’re application get ready

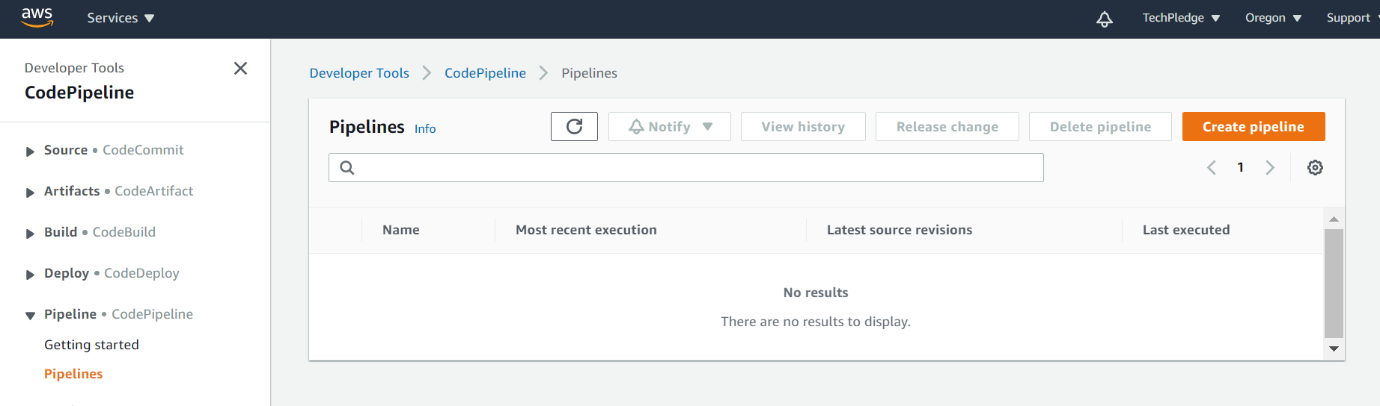


1. Browse the elastic Beanstalk application URL , you should see the application similar to below:

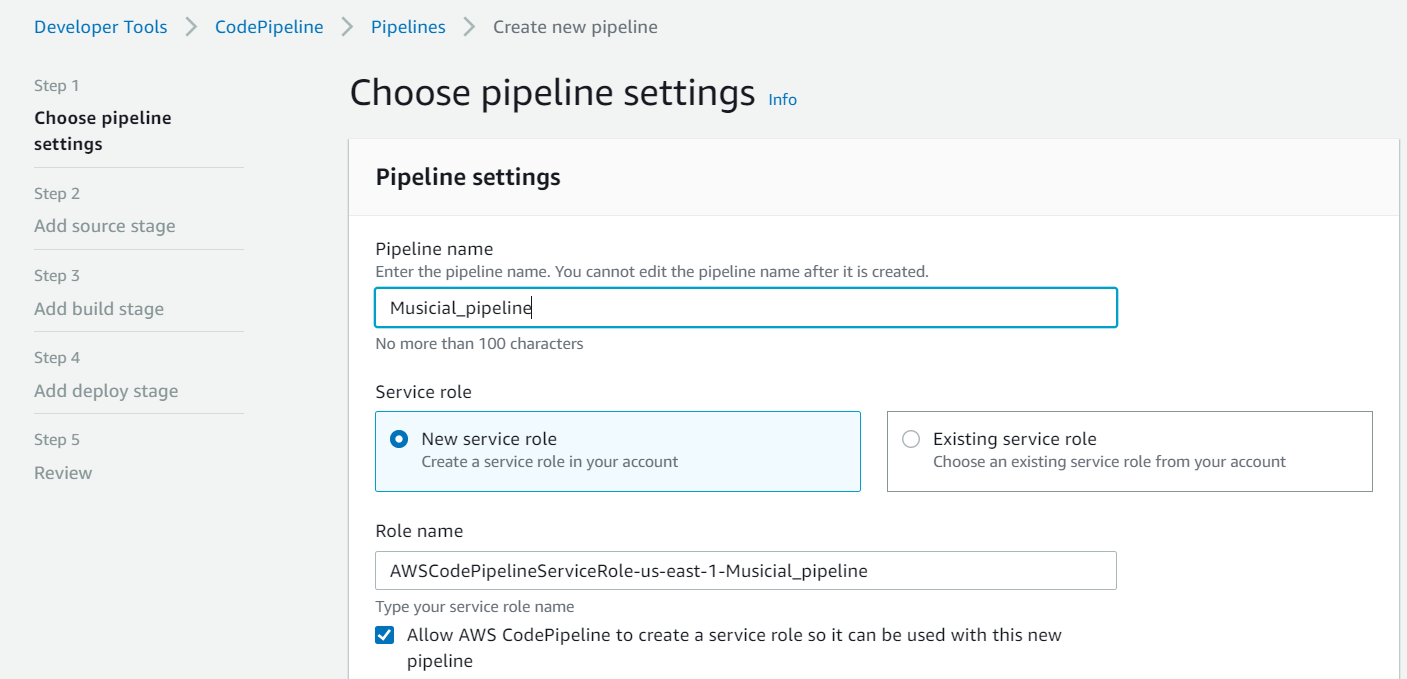


# Configure Code Pipeline

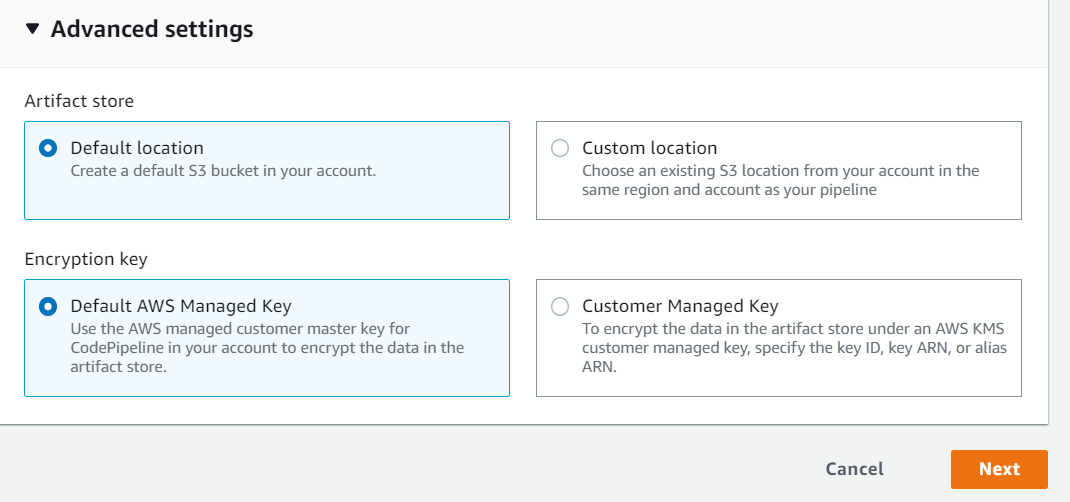
1. Now Open **Code Pipeline** console in AWS and Click **Create Pipeline**



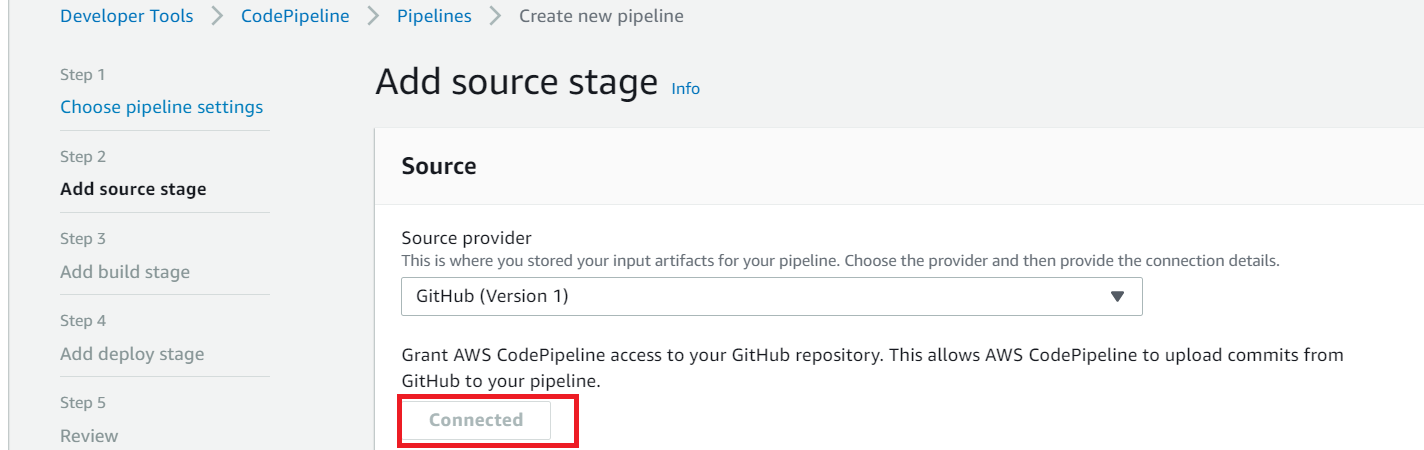
1. Type the Pipeline name “**Musical\_pipeline”** and Select **New Service Role.**



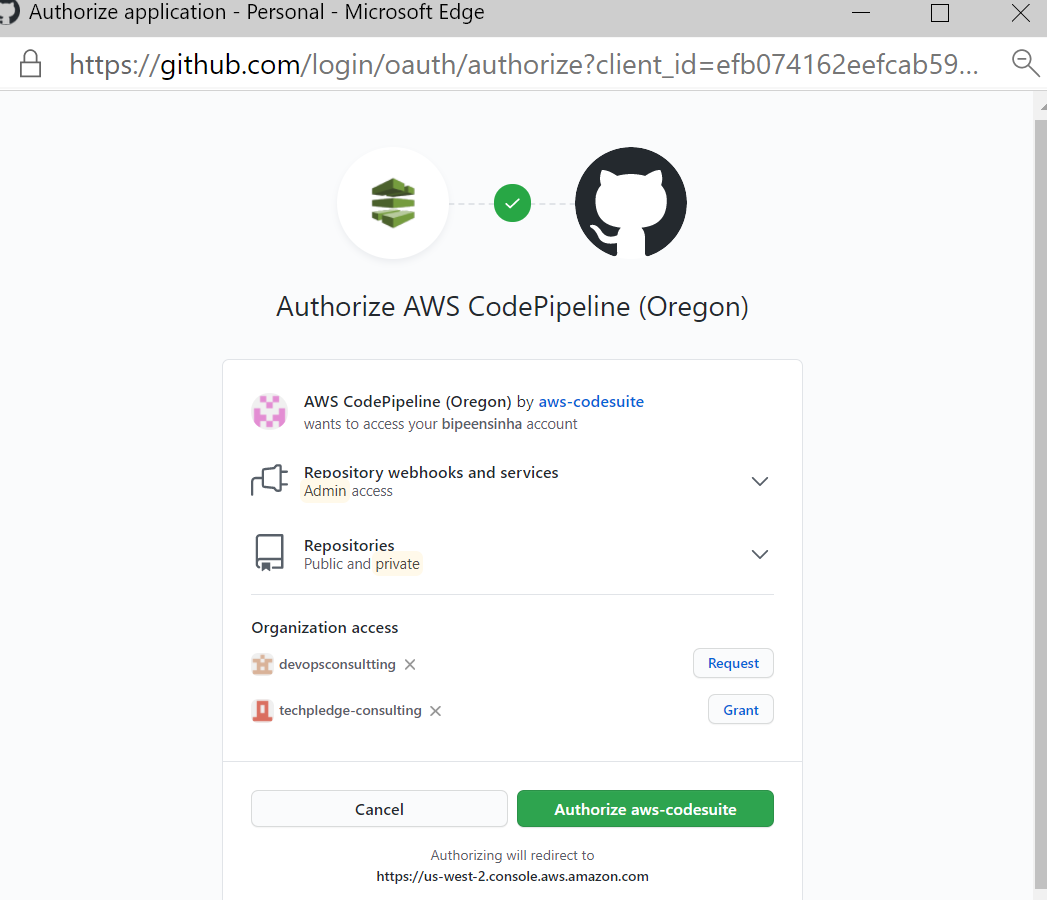
1. Leave all setting default and select **Next**



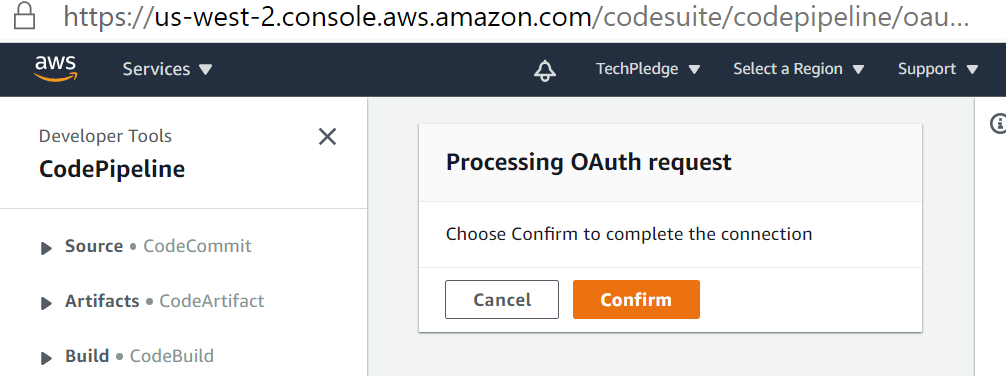
1. Select **GitHub** and select **connect** button



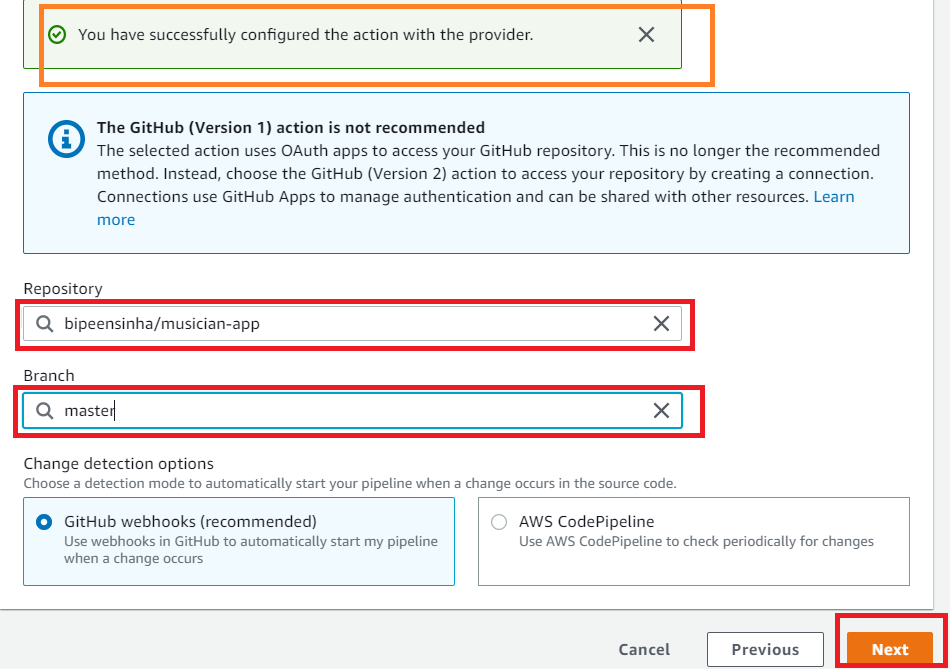
1. Click **Authorized**



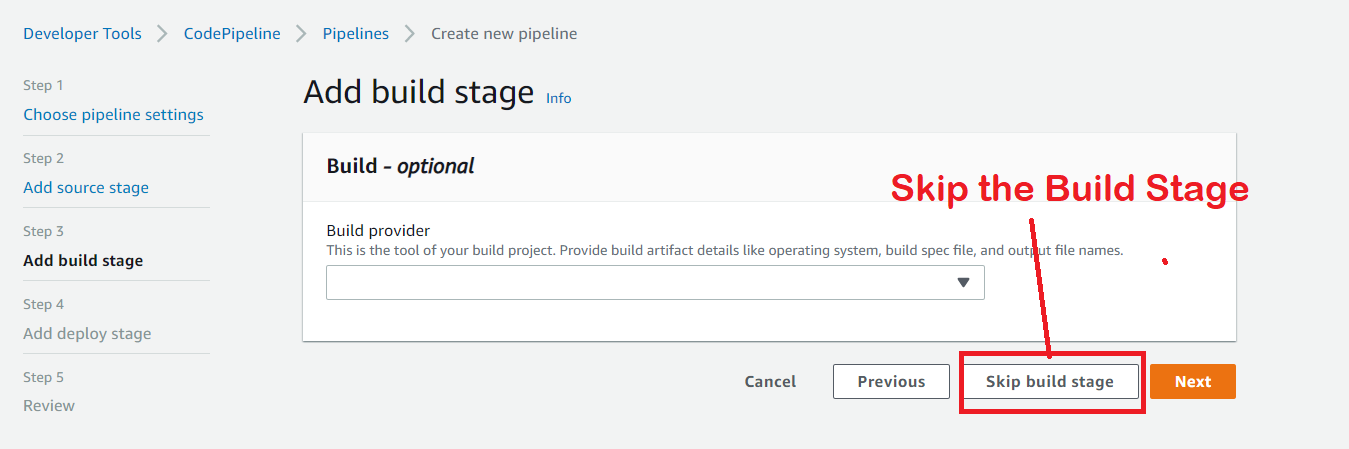
1. Click **Confirm** in **Oath Request**



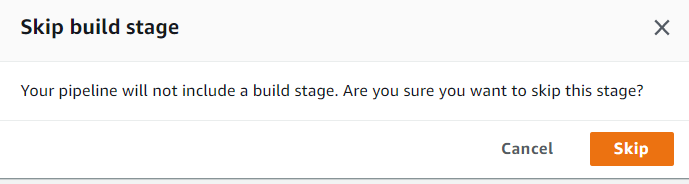
1. Click your repository **MusicApp** and Branch as **Master** and Click **next**



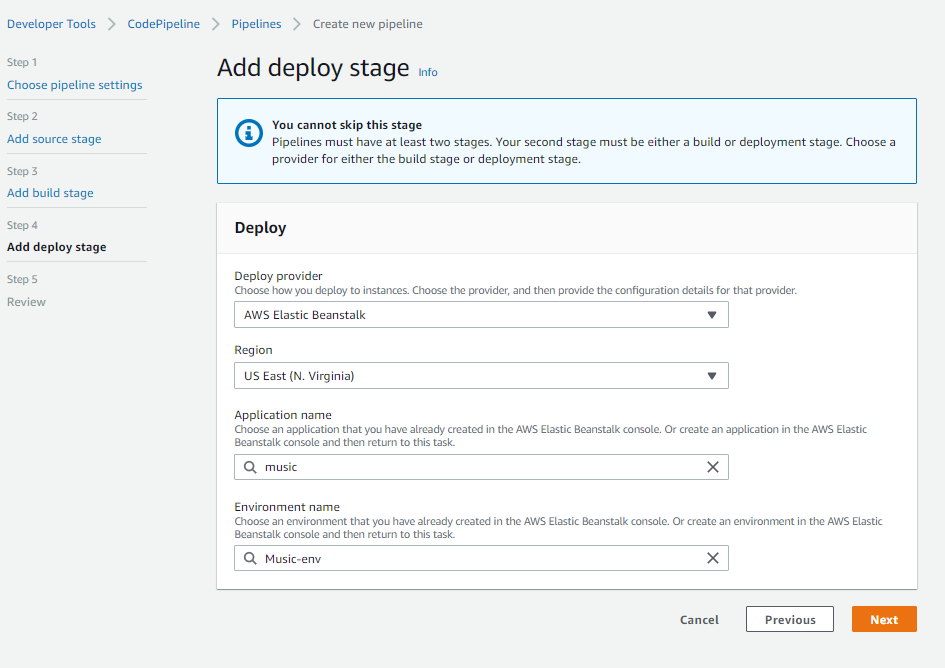
1. Click **Skip Build Stage**



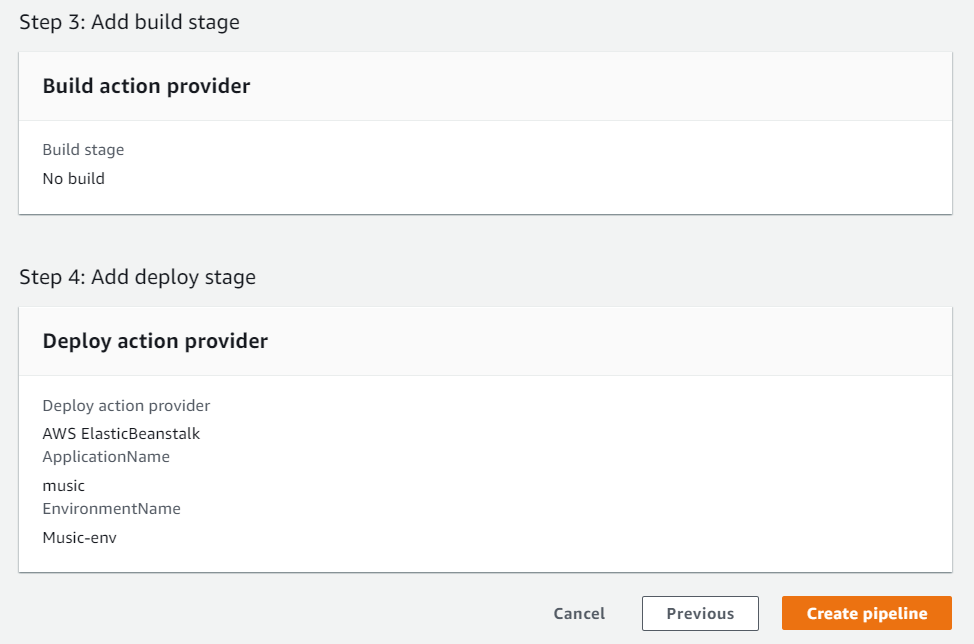
1. Click **Skip**



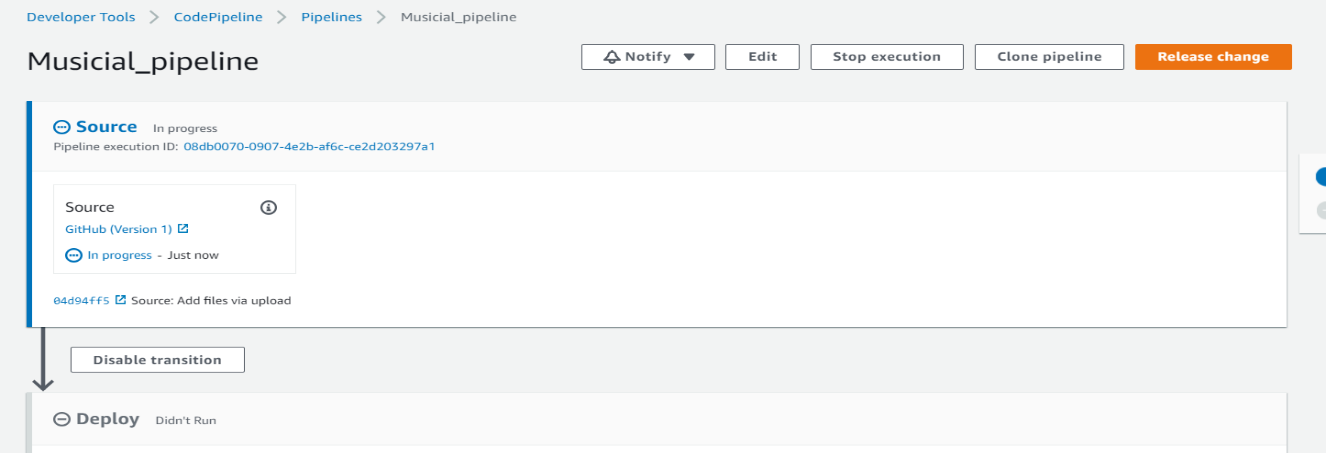
1. Click **Next** in **Add Deploy Stage**

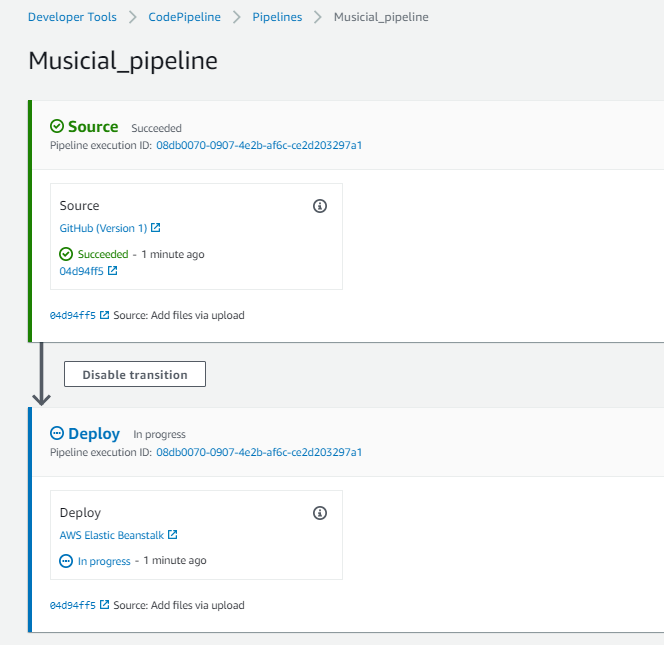


1. Click **Create Pipeline**

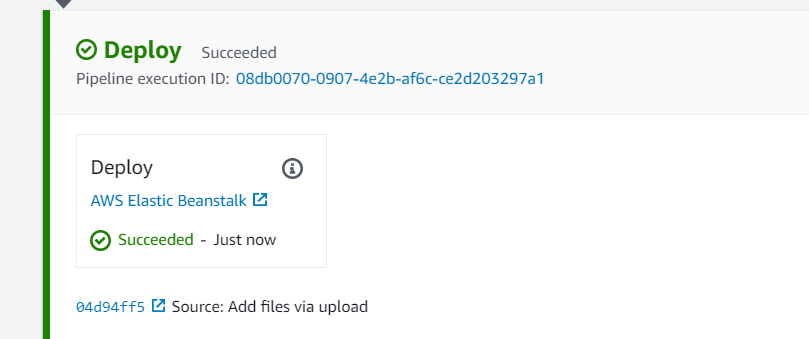


1. Now You see the Pipeline started



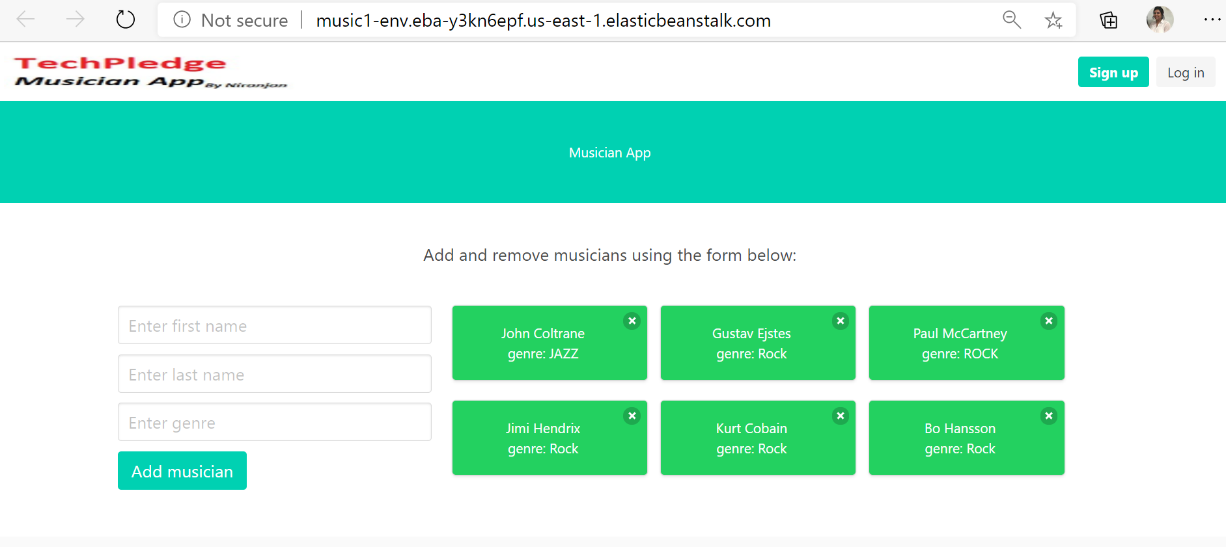


1. Now You see the Pipeline Successful after 5-7 Minute



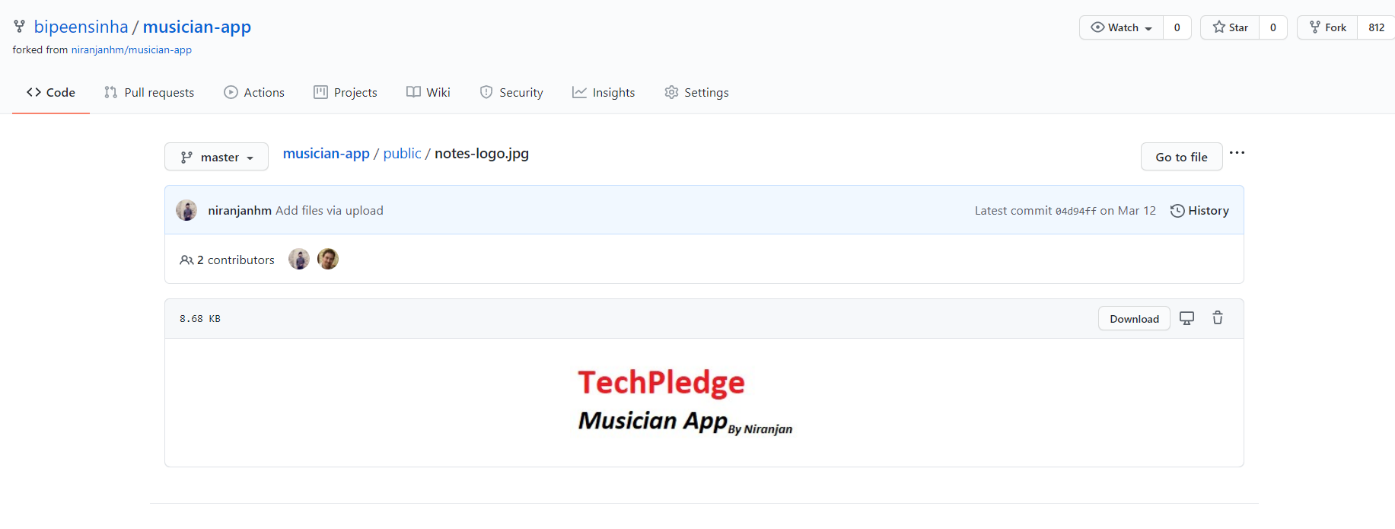
# Verify the Updated Elastic Beanstalk Site

1. Now Refresh the Elastic Beanstalk URL and you will be able to see the updated code

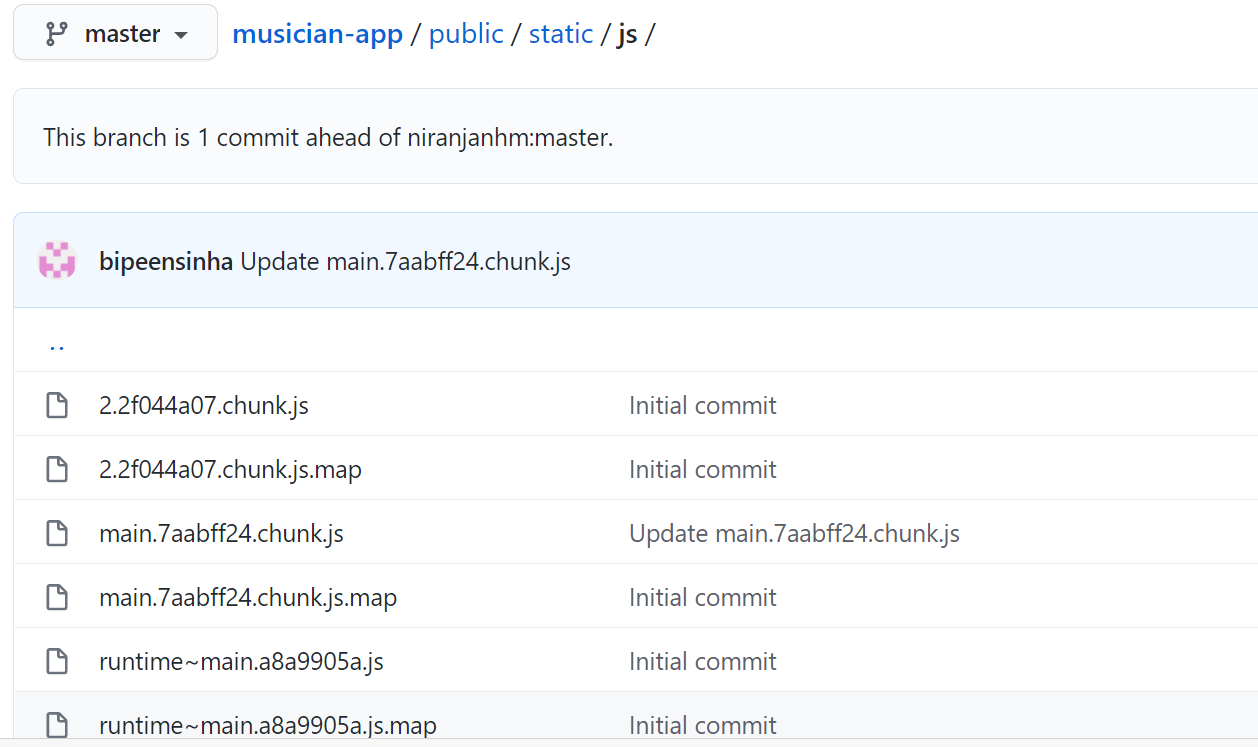


# Perform CI/CD

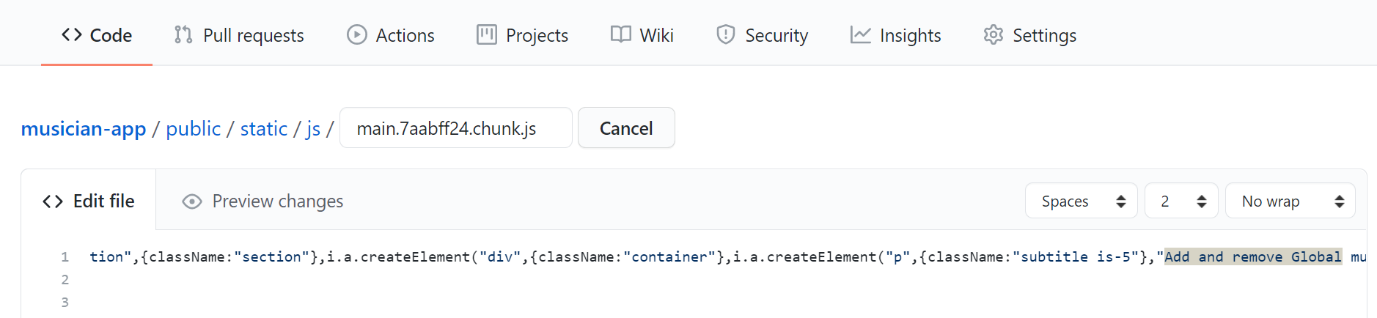
Update the New Logo in **Musician-app/public**/**notes-logo.jpg**

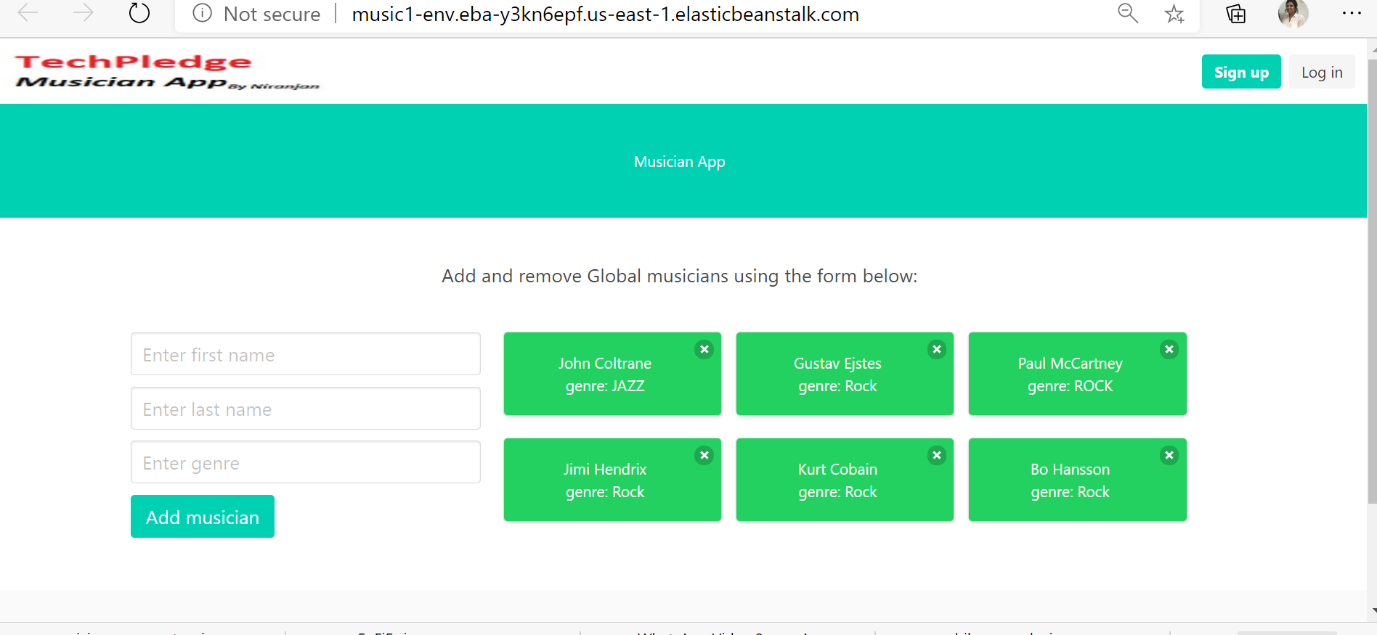


Update the content in **Musician-app/public**/**static/js/**



Add Global word





|  |  |  |
| --- | --- | --- |
|  |  |  |
| **That’s it. You have a live, publicly-accessible NodeJS Web App using full CI/CD Capabilities of AWS Code Pipeline and GitHub Web hook.** |