

## Deploying Nodejs-based application with CircleCI CI/CD Pipeline

In this project, we are going to build and deploy a Nodejs application using CircleCI.

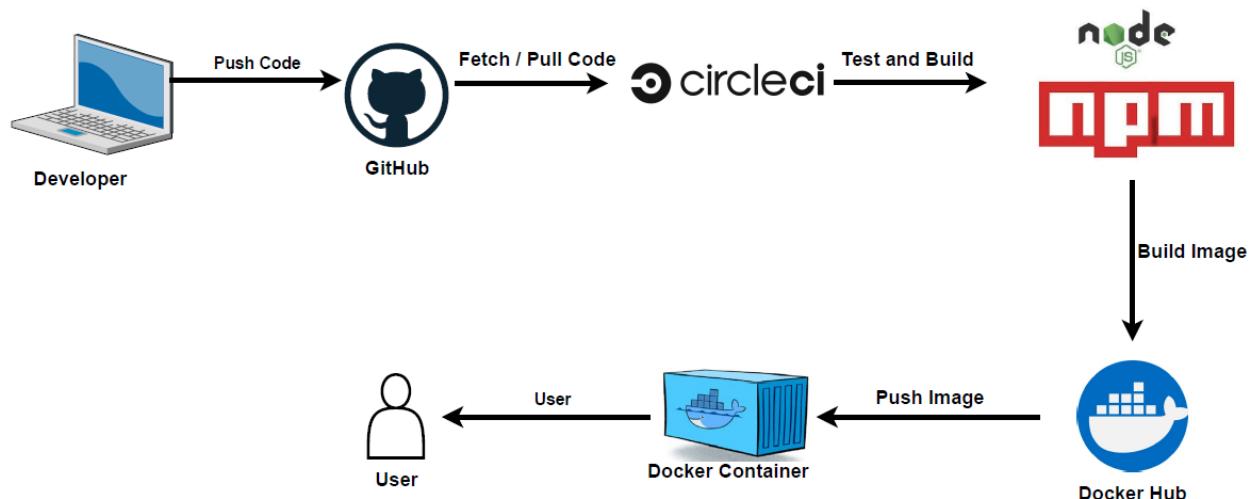
In CircleCI, you do not need to install any application on the system. You just directly login into CircleCI or signup using your Gmail account or any other account and you can start building your application directly from there.

It will directly fetch your source code from GitHub. So, you do not need to specifically store code on your local system or somewhere.

We will use the following resources:

1. Github
2. CircleCI
3. Docker
4. VM for Deployment

### Architecture



We have a simple NodeJS based application pushed to GitHub, CircleCI will pull the application from GitHub. The application is built and packaged using Nodejs and npm. npm is the world's largest software library, a package manager, and installer. It is also known as Node Package Manager, and is free to use.

Docker is used to build a Docker image. Then we will use the built image to deploy the application inside a Docker container.

## Implementation

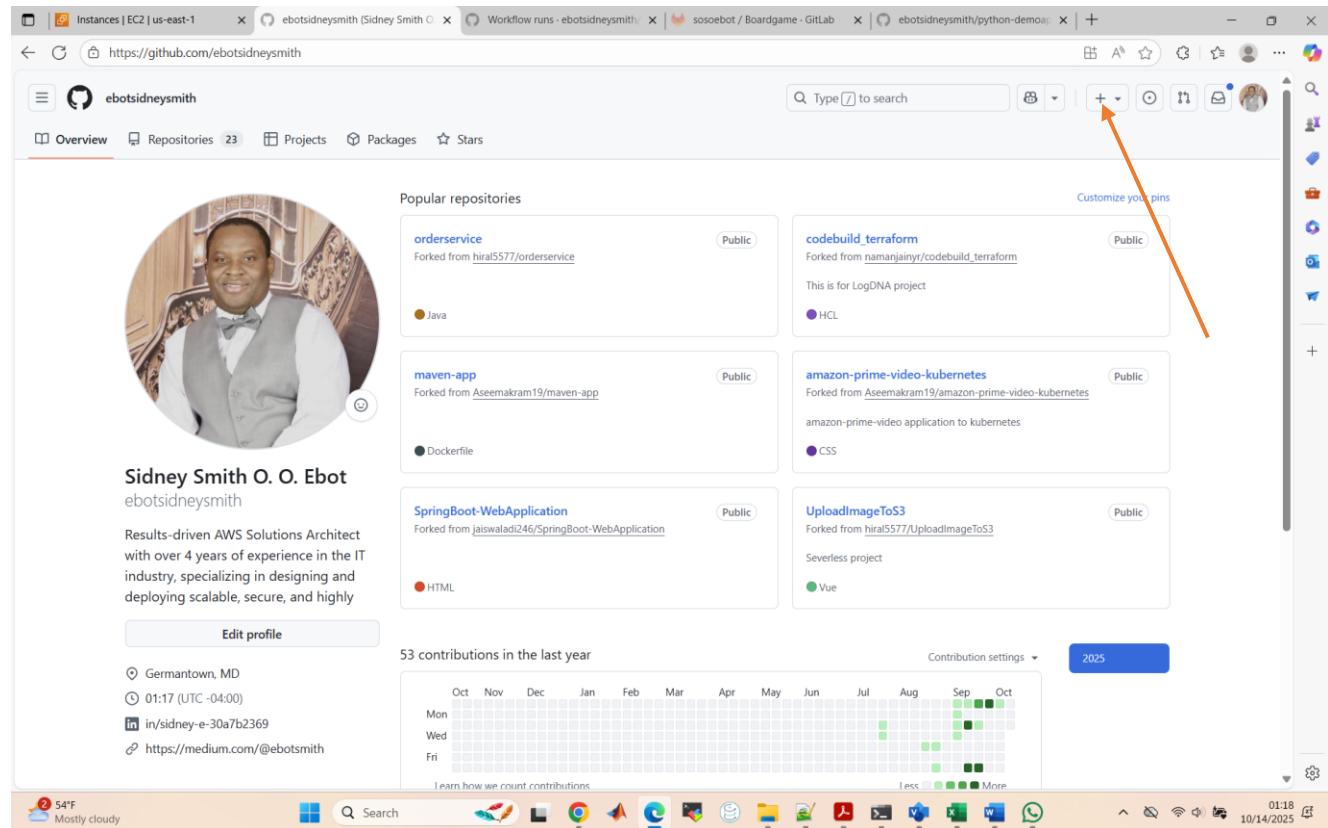
In the implementation, we will do the following:

- Create a GitHub repository and upload the project files to it
- Login to CircleCI
- Connect to GitHub Repo from CircleCI
- Creating Pipeline yaml file
- Push the Pipeline yaml file to the GitHub repository
- Adding Docker Environment Variables
- Verification

### STEP 1: Create a GitHub repository and upload the project files to it

Now, we have to create a repository on GitHub and upload our project files from our laptop to the GitHub repository.

#### Part 1: Create GitHub repository



We will start by creating the repository. Click on “+”

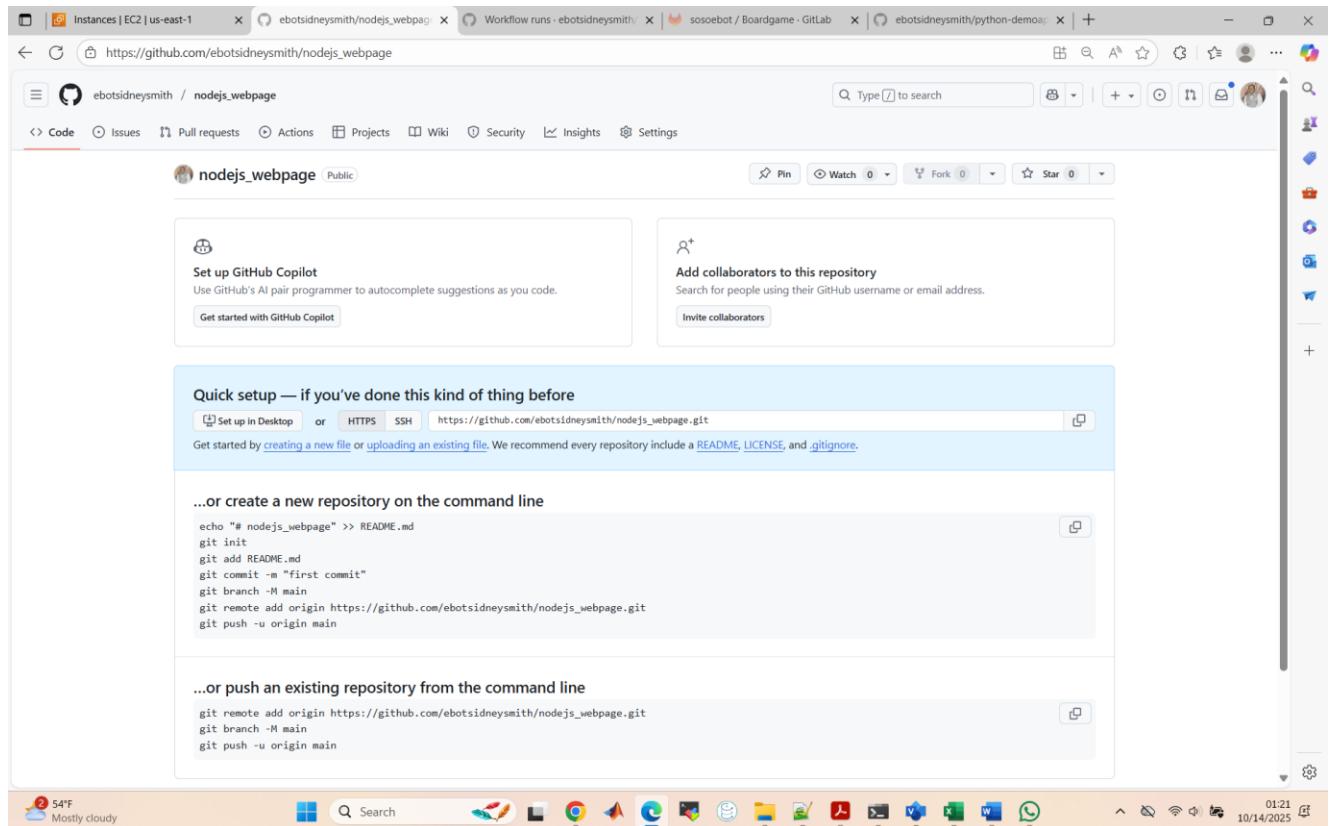
## Prepared by Sidney Smith Ebot

The screenshot shows a GitHub profile for 'ebotsidneysmith'. On the left, there's a large circular profile picture of a man in a tuxedo. Below it, the user's name 'Sidney Smith O. O. Ebot' and GitHub handle 'ebotsidneysmith' are displayed. A bio states: 'Results-driven AWS Solutions Architect with over 4 years of experience in the IT industry, specializing in designing and deploying scalable, secure, and highly available systems.' To the right, a 'Popular repositories' section lists several projects like 'orderservice', 'maven-app', 'amazon-prime-video-kubernetes', and 'UploadImageToS3'. A context menu is open in the top right corner, with the 'New repository' option highlighted by a red arrow. The bottom of the screen shows a Windows taskbar with various icons.

Select “New Repository”. Enter the name of the repository, we will call it “**nodejs\_webpage**”

The screenshot shows the 'Create a new repository' form on GitHub. In the 'General' tab, the 'Repository name' field contains 'nodejs\_webpage'. In the 'Configuration' tab, the 'Choose visibility' dropdown is set to 'Public'. The 'Add README' switch is off. Under 'Add .gitignore', the dropdown says 'No .gitignore'. Under 'Add license', the dropdown says 'No license'. A red arrow points to the 'Create repository' button at the bottom right of the form. The bottom of the screen shows a Windows taskbar with various icons.

Click on “Create Repository”

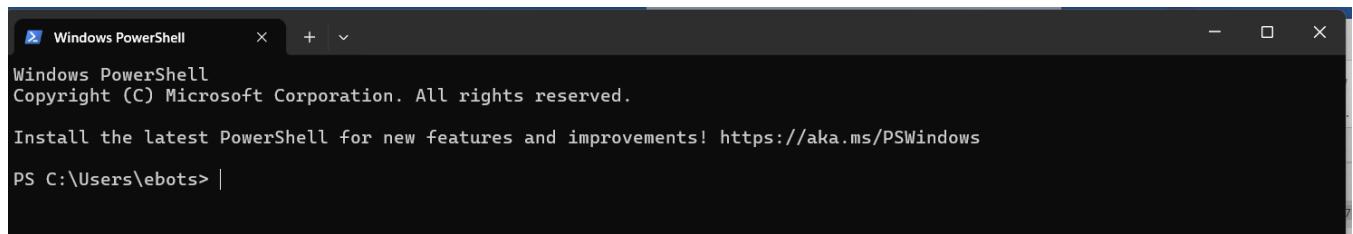


We have created the repository on GitHub

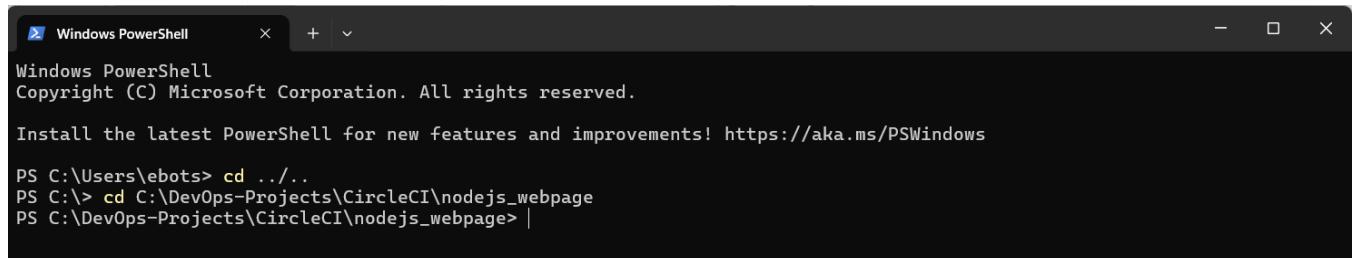
## Part 2: Upload project files to GitHub Repository

We have to navigate to the folder in our Local machine where we have our project files. The files are in this path: C:\DevOps-Projects\CircleCI\nodejs\_webpage

Open PowerShell



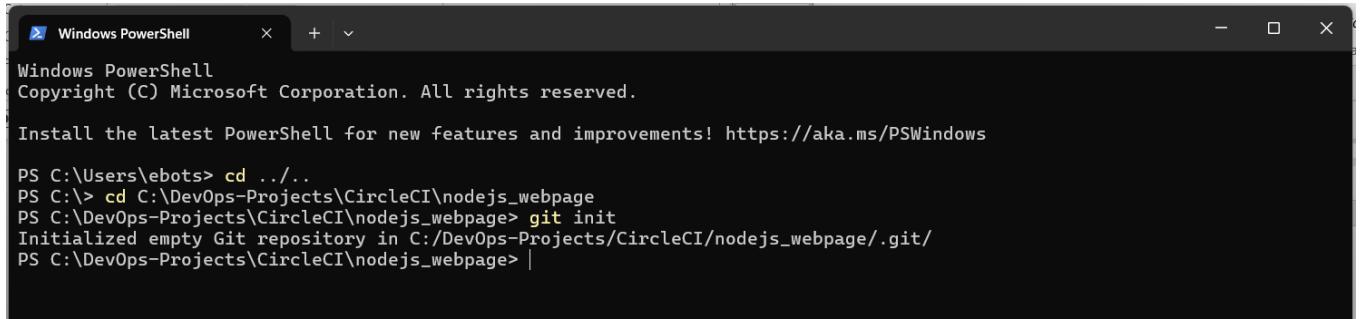
We now navigate to the folder



We are now in the repository in our local machine where the project files are stored

Initialize the repository in the local machine by using the command:

```
git init
```



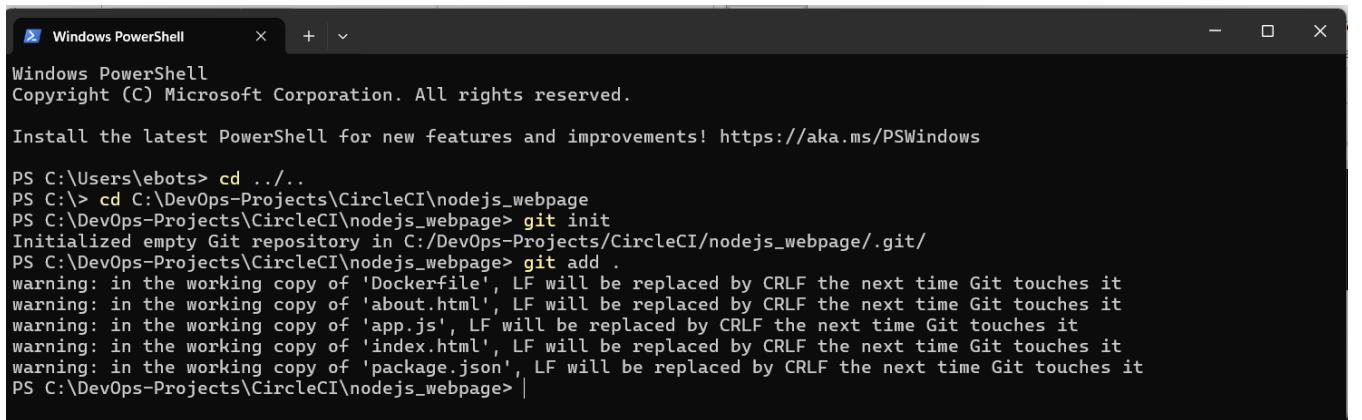
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ebots> cd ../..
PS C:\> cd C:\DevOps-Projects\CircleCI\nodejs_webpage
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git init
Initialized empty Git repository in C:/DevOps-Projects/CircleCI/nodejs_webpage/.git/
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

We stage the files by using the command:

```
git add .
```



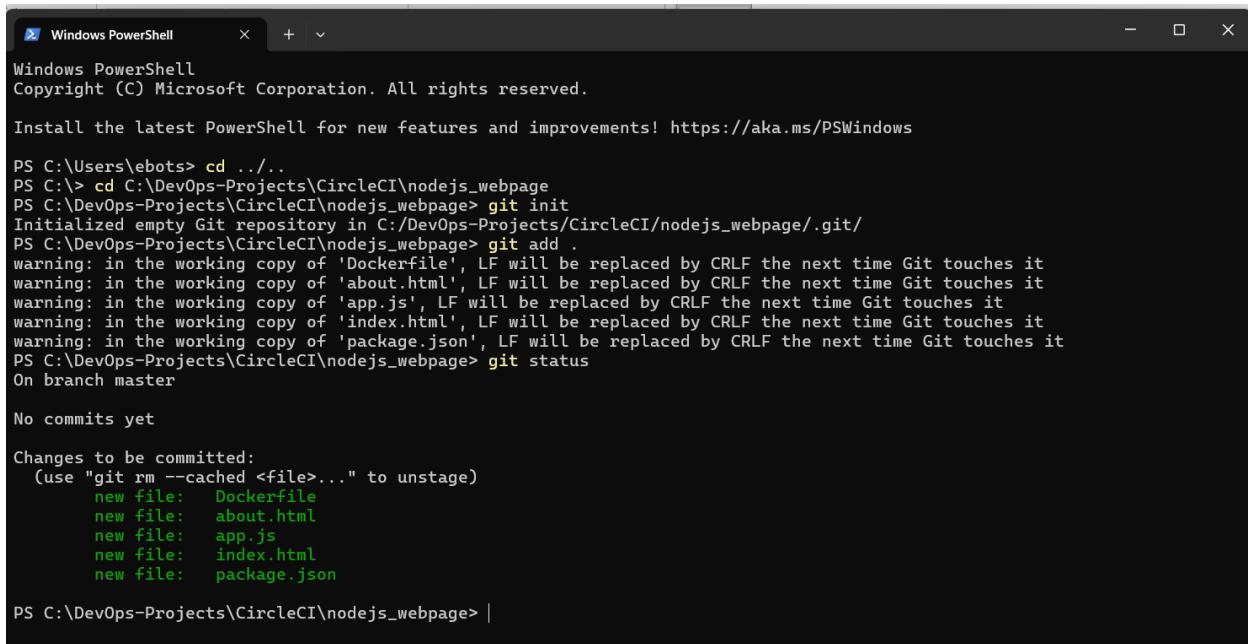
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ebots> cd ../..
PS C:\> cd C:\DevOps-Projects\CircleCI\nodejs_webpage
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git init
Initialized empty Git repository in C:/DevOps-Projects/CircleCI/nodejs_webpage/.git/
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git add .
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'about.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

We check the status of the files by using the command:

```
git status
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ebots> cd ../..
PS C:\> cd C:\DevOps-Projects\CircleCI\nodejs_webpage
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git init
Initialized empty Git repository in C:/DevOps-Projects/CircleCI/nodejs_webpage/.git/
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git add .
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'about.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git status
On branch master

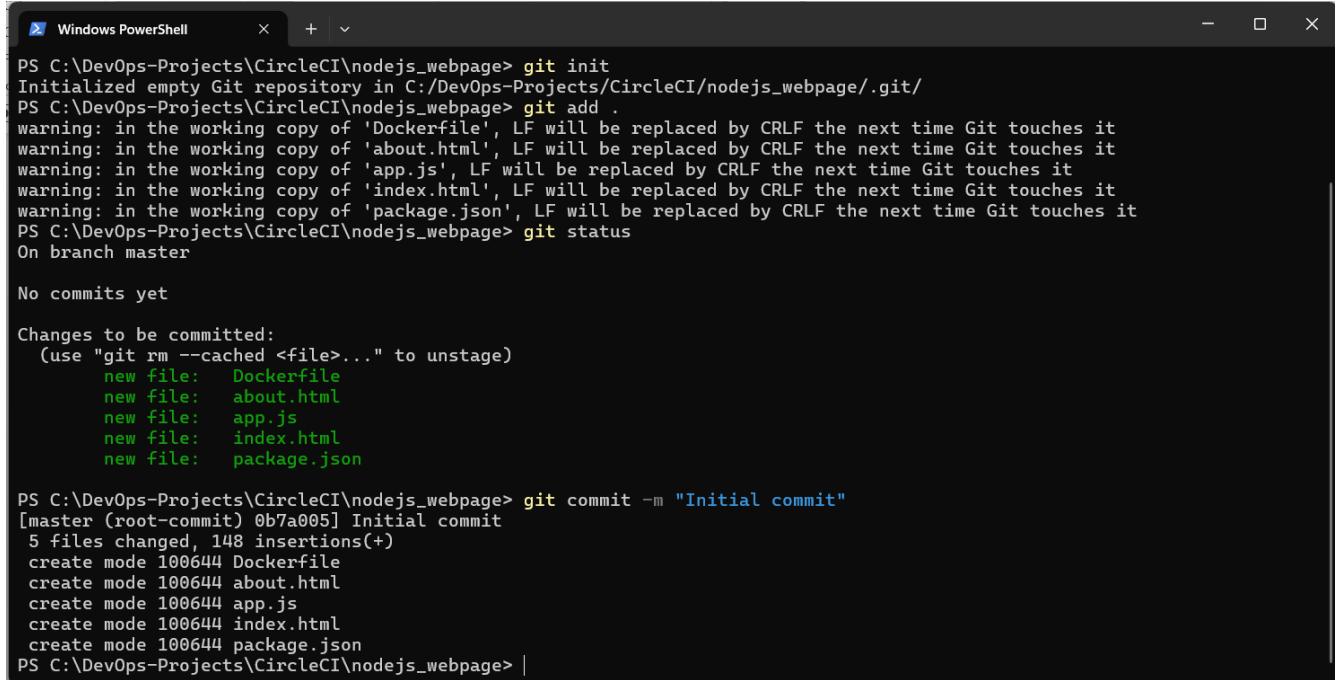
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:  Dockerfile
    new file:  about.html
    new file:  app.js
    new file:  index.html
    new file:  package.json

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

You can see the files that have been staged but not committed. Let us now commit the files by using the command:

```
git commit -m "Initial commit"
```



```
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git init
Initialized empty Git repository in C:/DevOps-Projects/CircleCI/nodejs_webpage/.git/
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git add .
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'about.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git status
On branch master

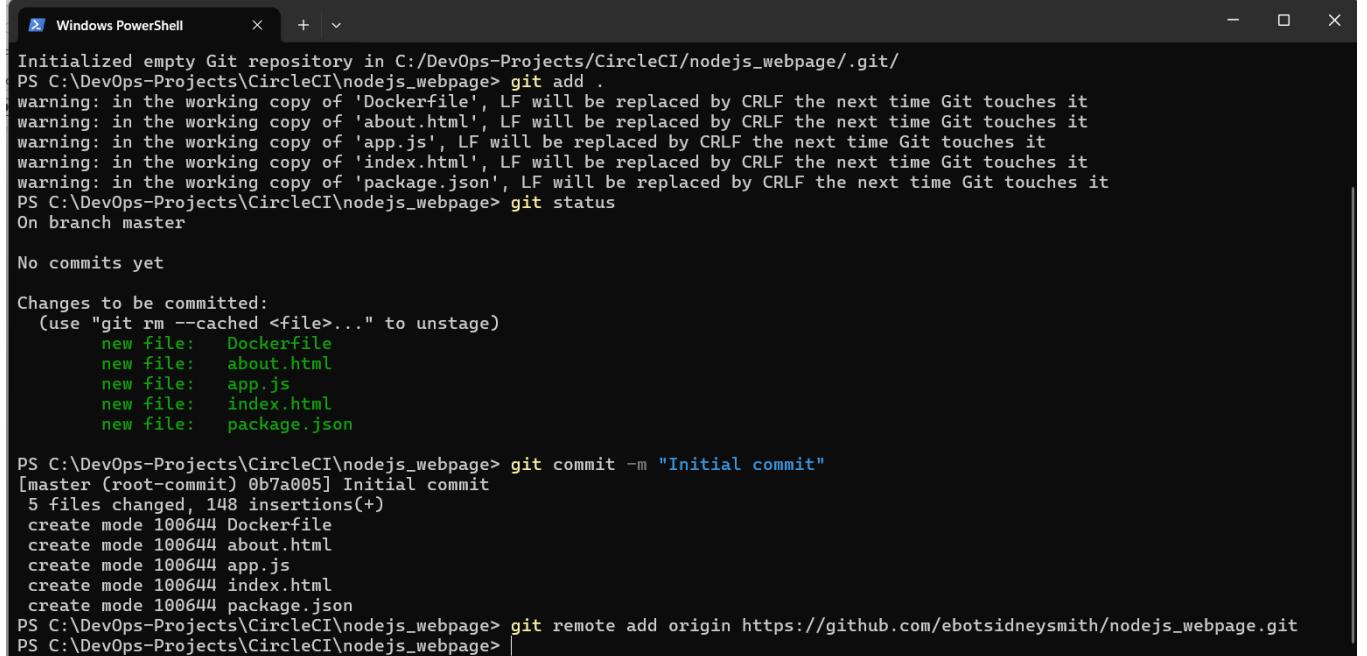
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:  Dockerfile
    new file:  about.html
    new file:  app.js
    new file:  index.html
    new file:  package.json

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git commit -m "Initial commit"
[master (root-commit) 0b7a005] Initial commit
  5 files changed, 148 insertions(+)
  create mode 100644 Dockerfile
  create mode 100644 about.html
  create mode 100644 app.js
  create mode 100644 index.html
  create mode 100644 package.json
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

Let us add the files to our remote repository by using the command:

```
git remote add origin https://github.com/ebotsidneysmith/nodejs_webpage.git
```



```
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git add .
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'about.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git status
On branch master

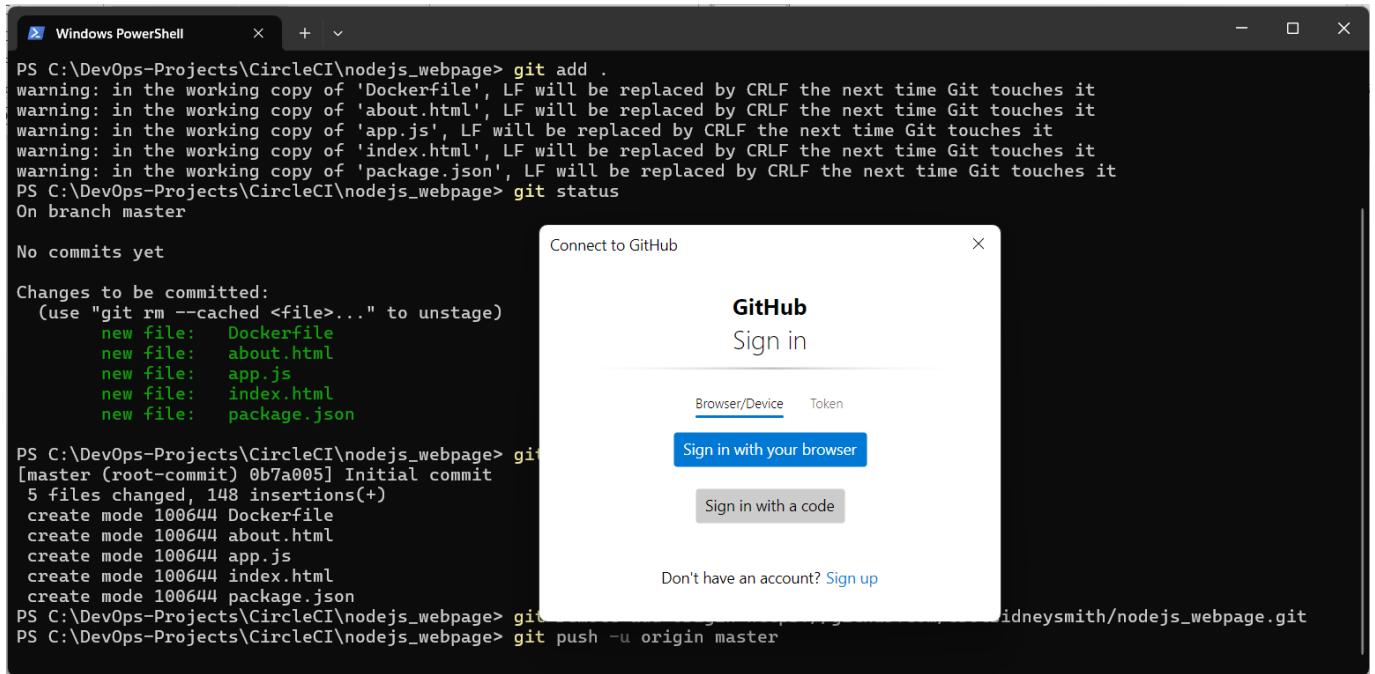
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:  Dockerfile
    new file:  about.html
    new file:  app.js
    new file:  index.html
    new file:  package.json

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git commit -m "Initial commit"
[master (root-commit) 0b7a005] Initial commit
  5 files changed, 148 insertions(+)
  create mode 100644 Dockerfile
  create mode 100644 about.html
  create mode 100644 app.js
  create mode 100644 index.html
  create mode 100644 package.json
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git remote add origin https://github.com/ebotsidneysmith/nodejs_webpage.git
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

Let us push the files to our master branch in the GitHub repository by using the command:

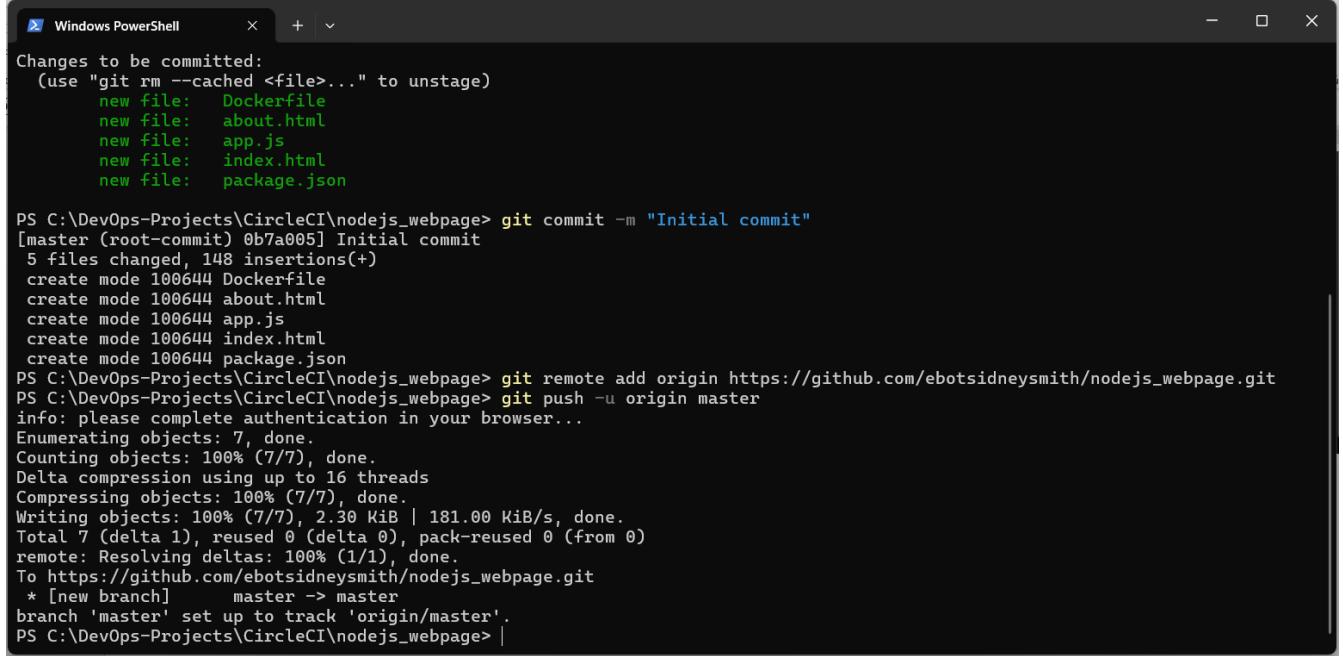
```
git push -u origin master
```



PS C:\DevOps-Projects\CircleCI\nodejs\_webpage> git add .  
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'about.html', LF will be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'app.js', LF will be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it  
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it  
PS C:\DevOps-Projects\CircleCI\nodejs\_webpage> git status  
On branch master  
No commits yet  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
new file: Dockerfile  
new file: about.html  
new file: app.js  
new file: index.html  
new file: package.json  
  
PS C:\DevOps-Projects\CircleCI\nodejs\_webpage> git commit -m "Initial commit"  
[master (root-commit) 0b7a005] Initial commit  
5 files changed, 148 insertions(+)  
create mode 100644 Dockerfile  
create mode 100644 about.html  
create mode 100644 app.js  
create mode 100644 index.html  
create mode 100644 package.json  
PS C:\DevOps-Projects\CircleCI\nodejs\_webpage> git remote add origin https://github.com/ebotsidneysmith/nodejs\_webpage.git  
PS C:\DevOps-Projects\CircleCI\nodejs\_webpage> git push -u origin master  
remote: Please complete authentication in your browser...  
Enumerating objects: 7, done.  
Counting objects: 100% (7/7), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (7/7), done.  
Writing objects: 100% (7/7), 2.30 KiB | 181.00 KiB/s, done.  
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (1/1), done.  
To https://github.com/ebotsidneysmith/nodejs\_webpage.git  
 \* [new branch] master -> master  
branch 'master' set up to track 'origin/master'.  
PS C:\DevOps-Projects\CircleCI\nodejs\_webpage>

The screenshot shows a Windows PowerShell window with a command-line interface for a Git repository. The user has run several commands: `git add .` to stage all new files, `git status` to check the staging area, `git commit -m "Initial commit"` to create the first commit, `git remote add origin https://github.com/ebotsidneysmith/nodejs\_webpage.git` to add the GitHub repository as a remote, and finally `git push -u origin master` to push the commit to the remote. A separate window titled "Connect to GitHub" is overlaid on the PowerShell window, prompting the user to "Sign in" with either a browser or a code token. The "Sign in with your browser" button is highlighted.

Click on “Sign up with browser”



PS C:\DevOps-Projects\CircleCI\nodejs\_webpage> git commit -m "Initial commit"  
[master (root-commit) 0b7a005] Initial commit  
5 files changed, 148 insertions(+)  
create mode 100644 Dockerfile  
create mode 100644 about.html  
create mode 100644 app.js  
create mode 100644 index.html  
create mode 100644 package.json  
PS C:\DevOps-Projects\CircleCI\nodejs\_webpage> git remote add origin https://github.com/ebotsidneysmith/nodejs\_webpage.git  
PS C:\DevOps-Projects\CircleCI\nodejs\_webpage> git push -u origin master  
remote: Please complete authentication in your browser...  
Enumerating objects: 7, done.  
Counting objects: 100% (7/7), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (7/7), done.  
Writing objects: 100% (7/7), 2.30 KiB | 181.00 KiB/s, done.  
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (1/1), done.  
To https://github.com/ebotsidneysmith/nodejs\_webpage.git  
 \* [new branch] master -> master  
branch 'master' set up to track 'origin/master'.  
PS C:\DevOps-Projects\CircleCI\nodejs\_webpage>

Refresh the GitHub page

## Prepared by Sidney Smith Ebot

The screenshot shows a GitHub repository page for 'nodejs\_webpage'. The repository was created by 'ebotsidneysmith' and has one commit from 'ebotsmith2000' titled 'Initial commit'. The files listed are Dockerfile, about.html, app.js, index.html, and package.json, all of which were committed 5 minutes ago. On the right side, there are sections for 'About', 'Activity', 'Releases', 'Packages', 'Languages', and 'Suggested workflows'. The 'Languages' section shows HTML at 91.4%, JavaScript at 7.0%, and Dockerfile at 1.6%. A 'Docker image' workflow is suggested.

You can see that we have uploaded our project file from the local machine to the GitHub repository.

## STEP 2: Login to CircleCI

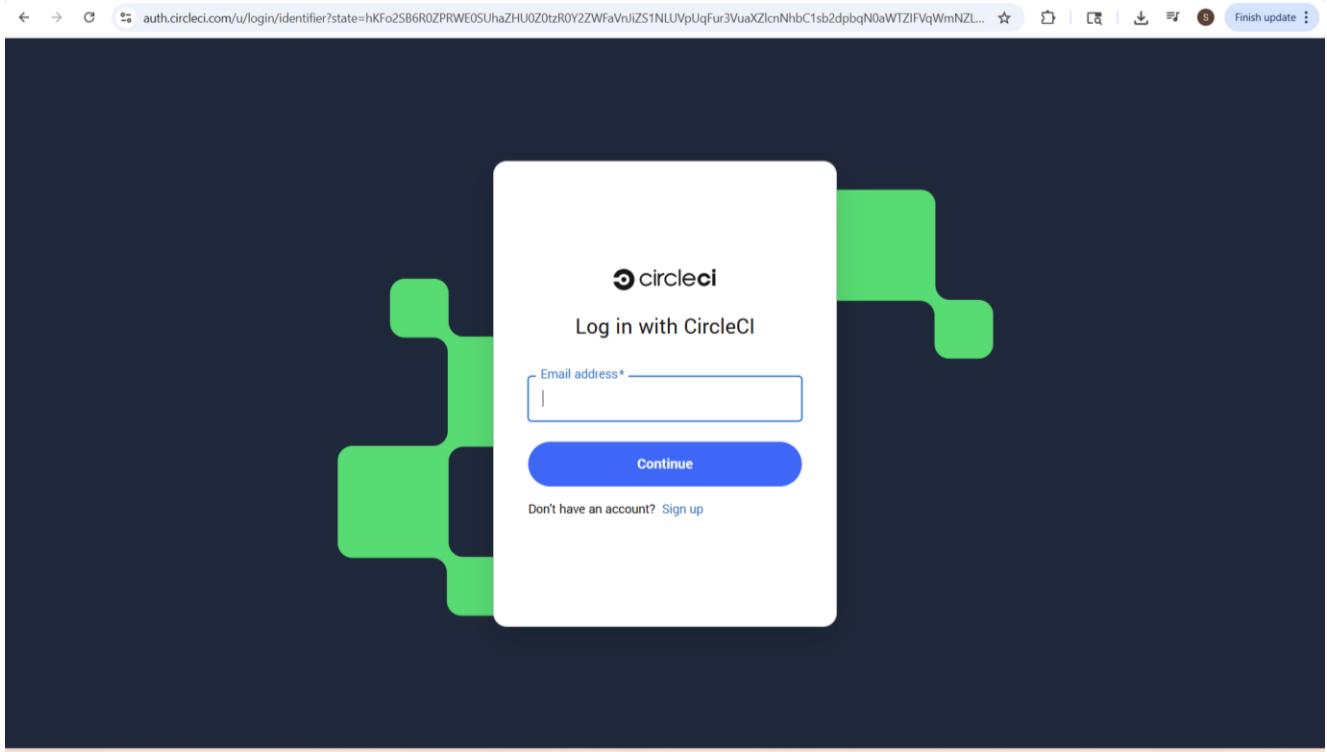
We have to first login to CircleCI. Search for “**CircleCI**” on google

A screenshot of a Google search results page for "circleci". The top result is the official CircleCI website, which features a dark header with the text "CircleCI: Autonomous validation for the AI era" and a "Log in" button. An orange arrow points to the "Log in" button. Below the main result are links for "Careers", "About Us", "Pricing plans", and "Documentation". A "People also ask" section is visible at the bottom.

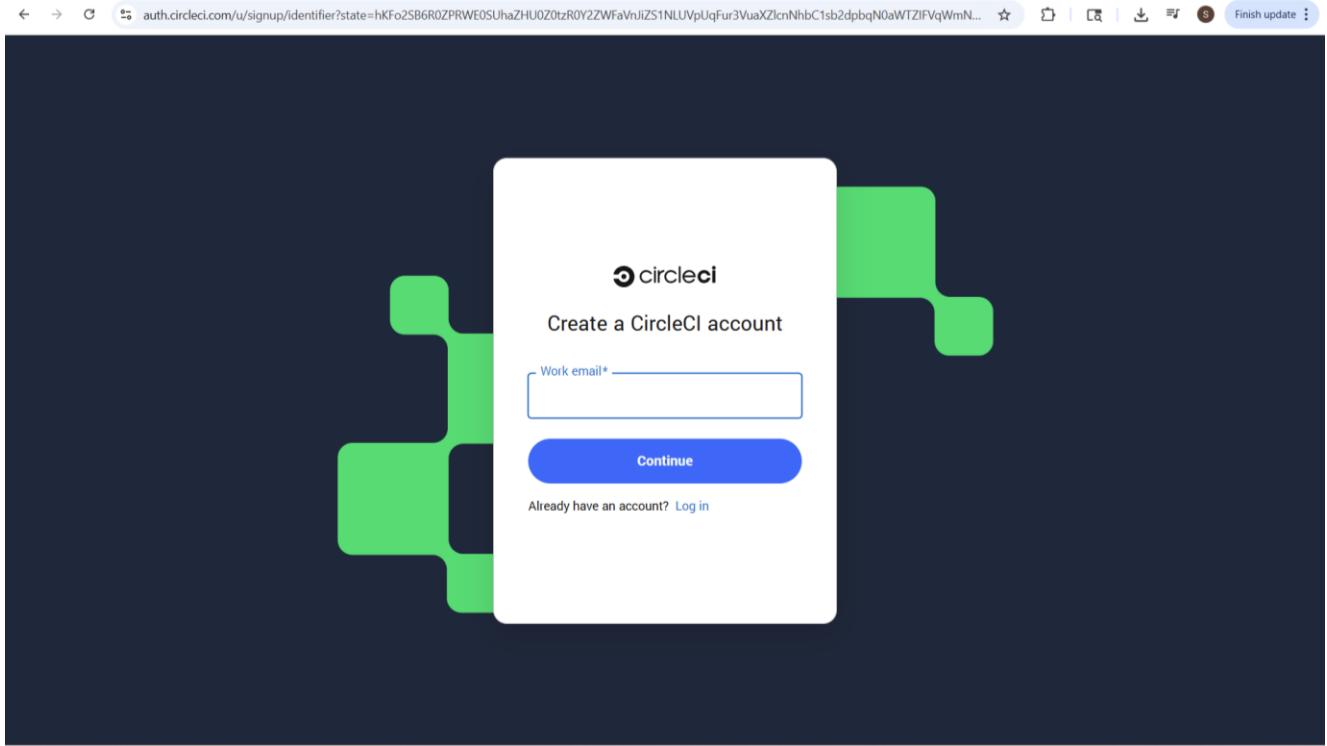
Click on “**Log in**”

A screenshot of the CircleCI login page. The page has a dark background with white text. On the left, there is a large, bold message: ">build something great.". On the right, there is a "Log in to CircleCI" section with three buttons: "Log in with email", "Log in with GitHub", and "Log in with Bitbucket Cloud". Below these buttons, there is small text about agreeing to SaaS Agreement and Privacy Policy, and mentions of reCAPTCHA, Google Privacy Policy, and Terms of Service. At the bottom of the page, there are links for Terms of Use, Privacy Policy, Cookie Policy, Security, and a copyright notice for Circle Internet Services, Inc. There is also an "Ask AI" button in the bottom right corner.

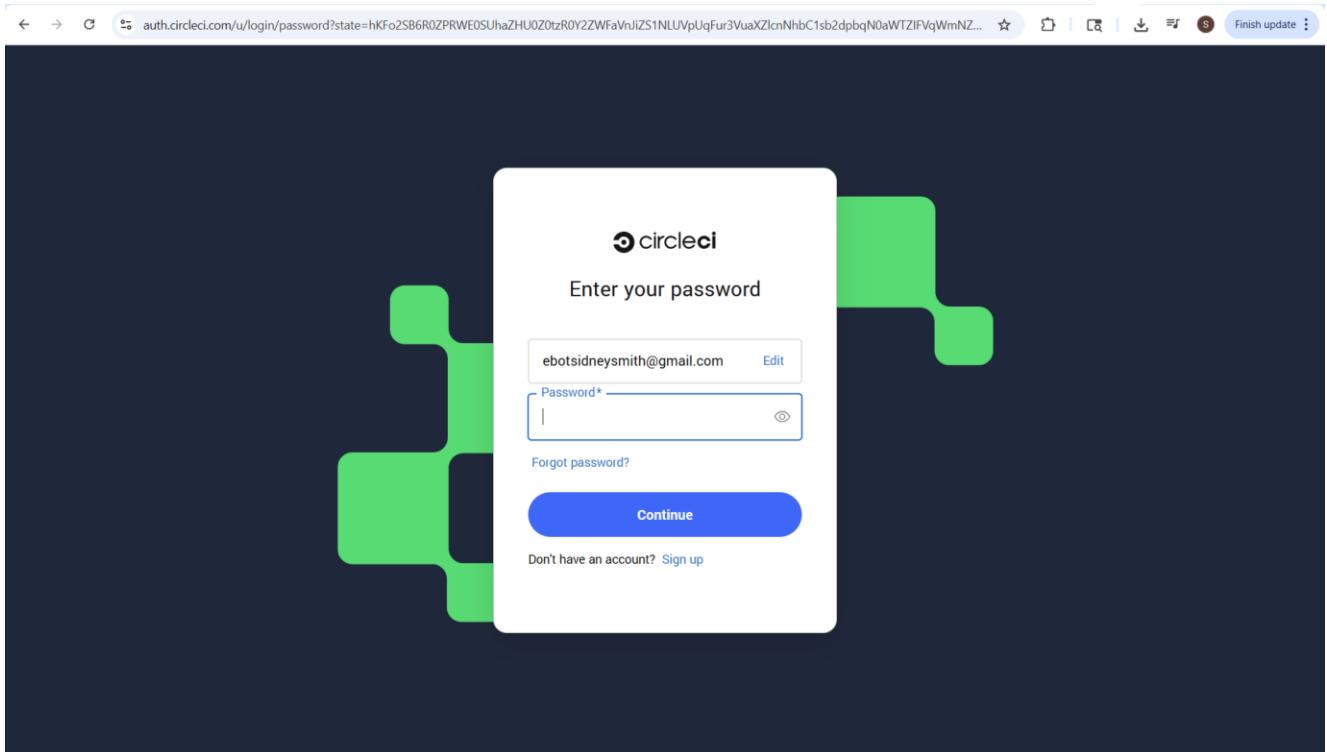
You can choose the method you want to use to log in. I will log in using my email. Click on “**Log in with email**”



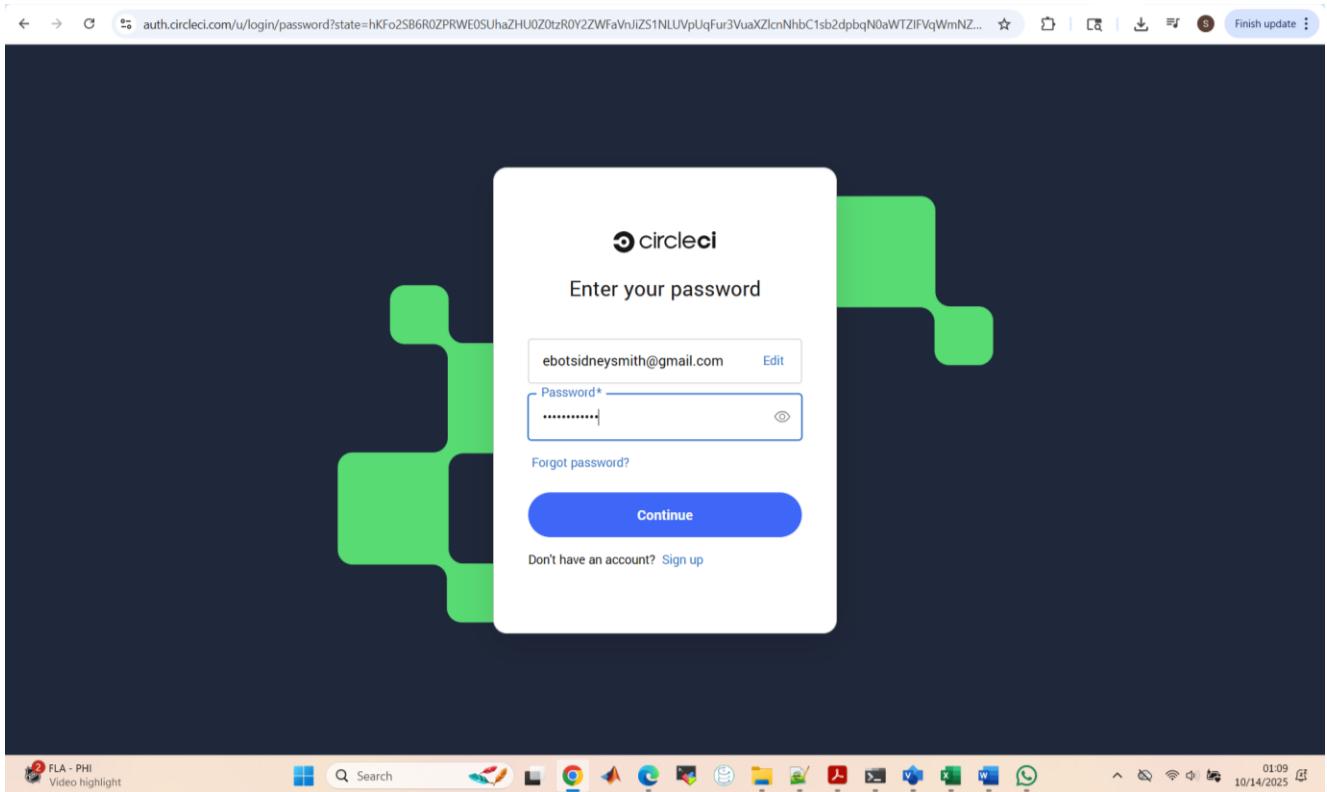
I do not yet have an account. I will click on “**Sign up**”



Then enter my email address



Then, I will enter my password



Click on “Continue”

The screenshot shows the CircleCI web interface. At the top, there's a navigation bar with icons for back, forward, search, and user profile. The URL is app.circleci.com/pipelines/github/ebotsidneysmith. On the right of the header are various status indicators and a 'Finish update' button.

The main area has a dark header 'All Pipelines'. To the left is a sidebar with links: Organization Home, Pipelines (which is selected and highlighted in grey), Projects, Deployments, Insights, Self-Hosted Runners, Organization Settings, and a 'Plan' section with an 'UPGRADE' button. A tooltip in the 'Plan' section says 'Checkout method is changing' and 'Blobless checkout becomes the default - verify your builds still work or switch to full checkout by November 3'.

A prominent message box at the top right says: 'Xcode Image Deprecation - Xcode images 13.4.1, 15.1.0, 15.2.0, 15.3.0, 16.0.0, and 16.1.0 will be removed on November 7, 2025. Brownouts: Oct 22, Oct 29, Nov 5. Update your configs now to avoid job failures. [Learn more.](#)'

The central part of the page features a 'Start running pipelines with your team' section with a 'Add Project' button. Below it is a visual representation of three pipeline stages: a green stage with a checkmark, a purple stage with a pixelated icon, and a red stage with a circular icon.

The bottom of the screen shows a Windows taskbar with various pinned application icons like File Explorer, Edge, Google Chrome, and others. The system tray shows the date as 10/14/2025 and time as 01:09.

I have logged in with my email.

### STEP 3: Connect to GitHub Repo from CircleCI

The screenshot shows the CircleCI All Pipelines dashboard. On the left, there's a sidebar with navigation links: Organization Home, Pipelines (selected), Projects, Deployments, Insights, Self-Hosted Runners, Organization Settings, and Plan (with an Upgrade button). A message in the top right corner informs about Xcode Image Deprecation. In the center, there's a section titled "Start running pipelines with your team" with a "Add Project" button. Below it is a visual representation of pipeline stages with colored bars and circular icons.

Click on “Projects”

The screenshot shows the CircleCI Projects dashboard. The sidebar has 'Projects' selected. A message at the top says "Set up new projects or follow projects already building on CircleCI. Following a project adds it to your dashboard." The main area displays a list of GitHub repositories with their names and "Set up" buttons. An orange arrow points to the "Set up" button for the repository "nodejs\_webpage".

Click on “Set Up” on the GitHub repository of the project

The screenshot shows the CircleCI project dashboard for the organization 'ebotsidneysmith'. A modal window titled 'Select your config.yml file' is open, prompting the user to choose a configuration file from their repository. The options are:

- Fastest:** Use the `.circleci/config.yml` in my repo  
Selected option: `nodejs_webpage`
- Faster:** Commit a starter CI pipeline to a new branch
- Fast:** Take me to a config.yml template that I can edit

A blue 'Set Up Project' button is at the bottom right of the modal.

I won't select any of them. I will go back to my GitHub repository and create a new branch called "developer"

The screenshot shows the GitHub repository page for 'nodejs\_webpage' owned by 'ebotsidneysmith'. The repository has 1 branch and 0 tags. The 'Code' tab is selected. An orange arrow points to the 'master' dropdown menu, indicating where to click to create a new branch.

Click on the drop down on "master"

## Prepared by Sidney Smith Ebot

The screenshot shows a GitHub repository page for 'nodejs\_webpage'. A red arrow points from the text 'Enter the name of the new branch "developer"' to the search input field in the 'Switch branches/tags' dropdown menu. The menu also shows options like 'Create branch developer from master'.

Enter the name of the new branch “developer”

The screenshot shows the same GitHub repository page after entering 'developer' in the search field. A red arrow points from the text 'Click on “create branch developer from master”' to the 'Create branch developer from master' button in the dropdown menu.

Click on “create branch developer from master”

The screenshot shows a GitHub repository page for 'nodejs\_webpage'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, the repository name 'nodejs\_webpage' is displayed, along with a 'Public' badge. A dropdown menu shows the current branch is 'developer'. There are also links for '2 Branches' and '0 Tags'. On the right side of the header, there are buttons for Pin, Watch, Fork, Star, and Settings. A search bar at the top right says 'Type [ ] to search'. The main content area shows a list of files: Dockerfile, about.html, app.js, index.html, and package.json, all committed by 'ebotsmith2000' as 'Initial commit' 35 minutes ago. To the right of the file list, there's an 'About' section with a note that the branch is up-to-date with 'master'. Below the file list, there's a 'README' section with a link to 'Add a README'. On the far right, there are sections for Activity, Releases, Packages, Languages (HTML 91.4%, JavaScript 7.0%, Dockerfile 1.6%), and Suggested workflows (Jekyll using Docker image).

The branch has been created.

#### STEP 4: Creating pipeline yaml file

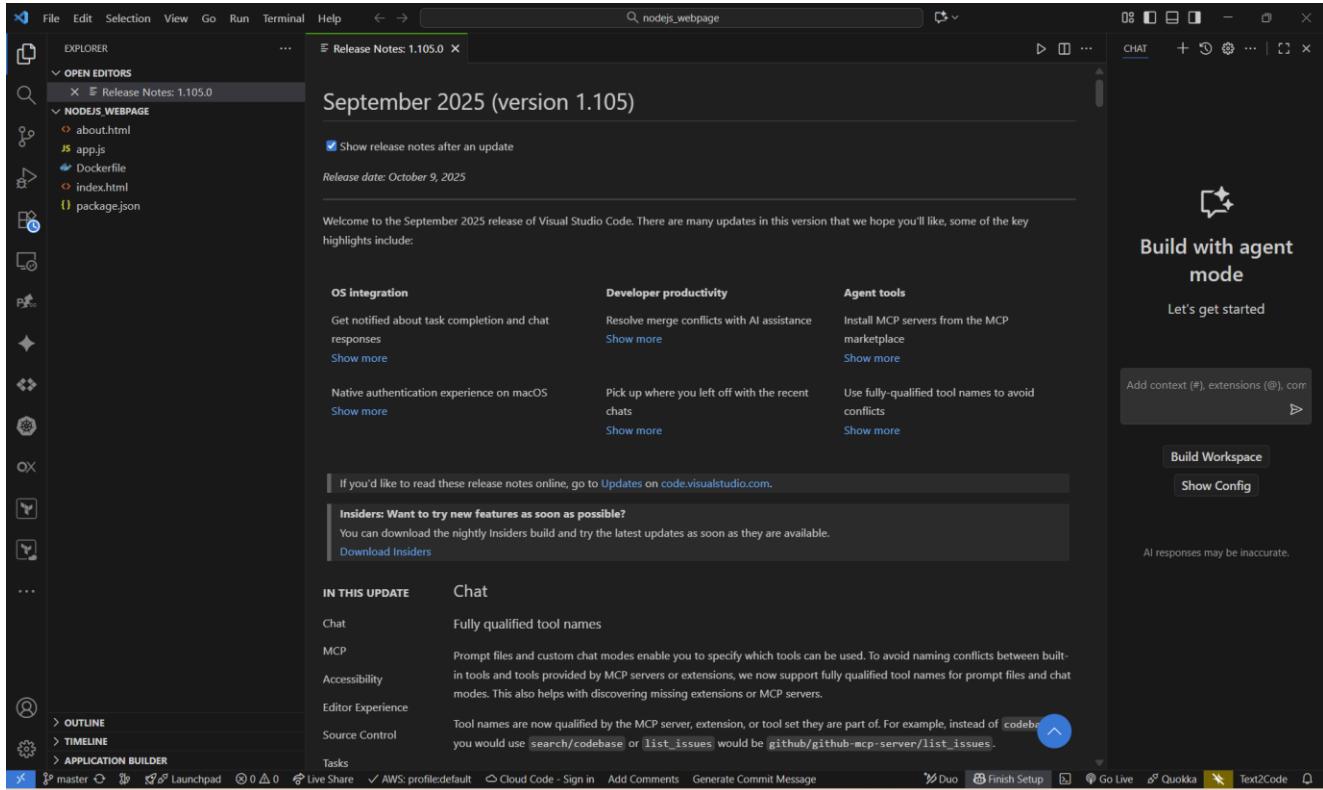
Open the project folder in our local machine on VSCode

```
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file: Dockerfile
  new file: about.html
  new file: app.js
  new file: index.html
  new file: package.json

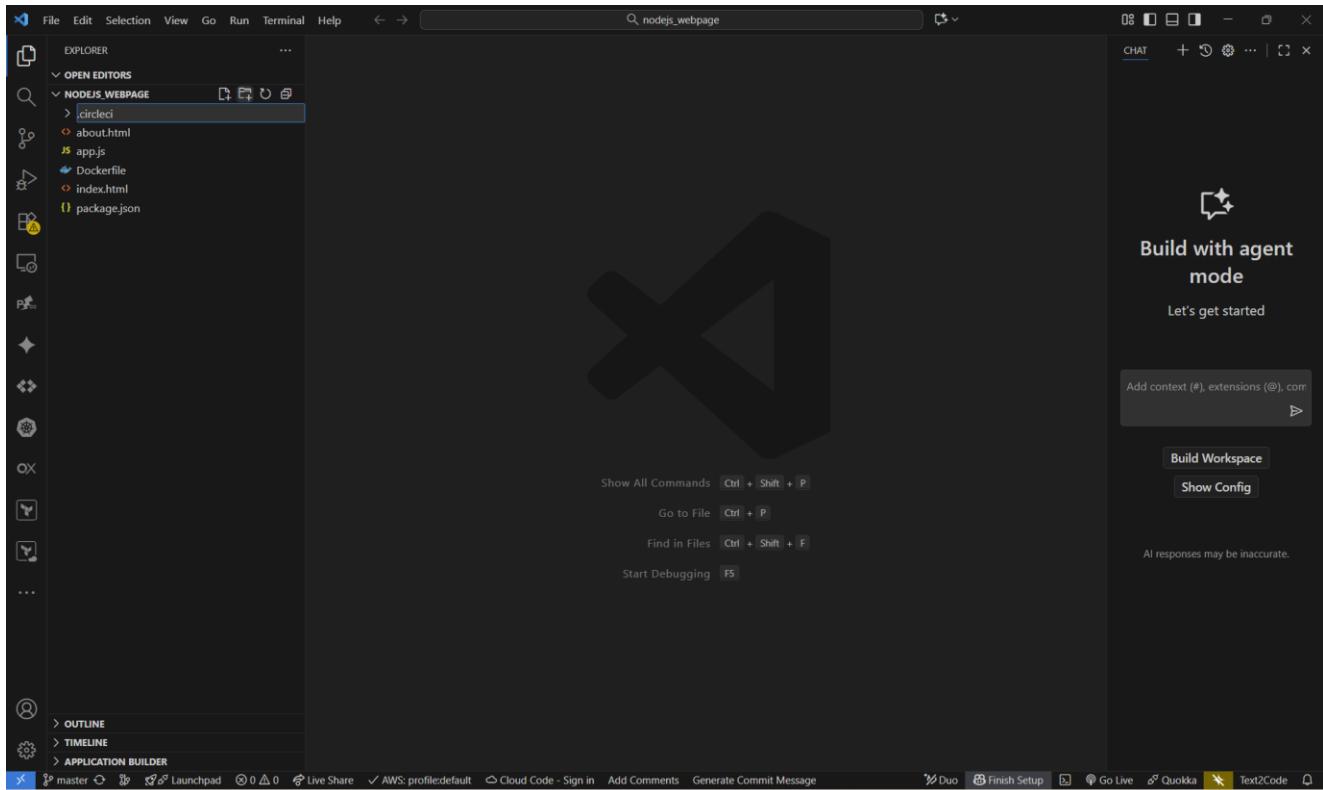
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git commit -m "Initial commit"
[master (root-commit) 0b7a005] Initial commit
 5 files changed, 148 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 about.html
 create mode 100644 app.js
 create mode 100644 index.html
 create mode 100644 package.json
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git remote add origin https://github.com/ebotsidneysmith/nodejs_webpage.git
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.30 KiB | 181.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ebotsidneysmith/nodejs_webpage.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

We can do this by running the command:

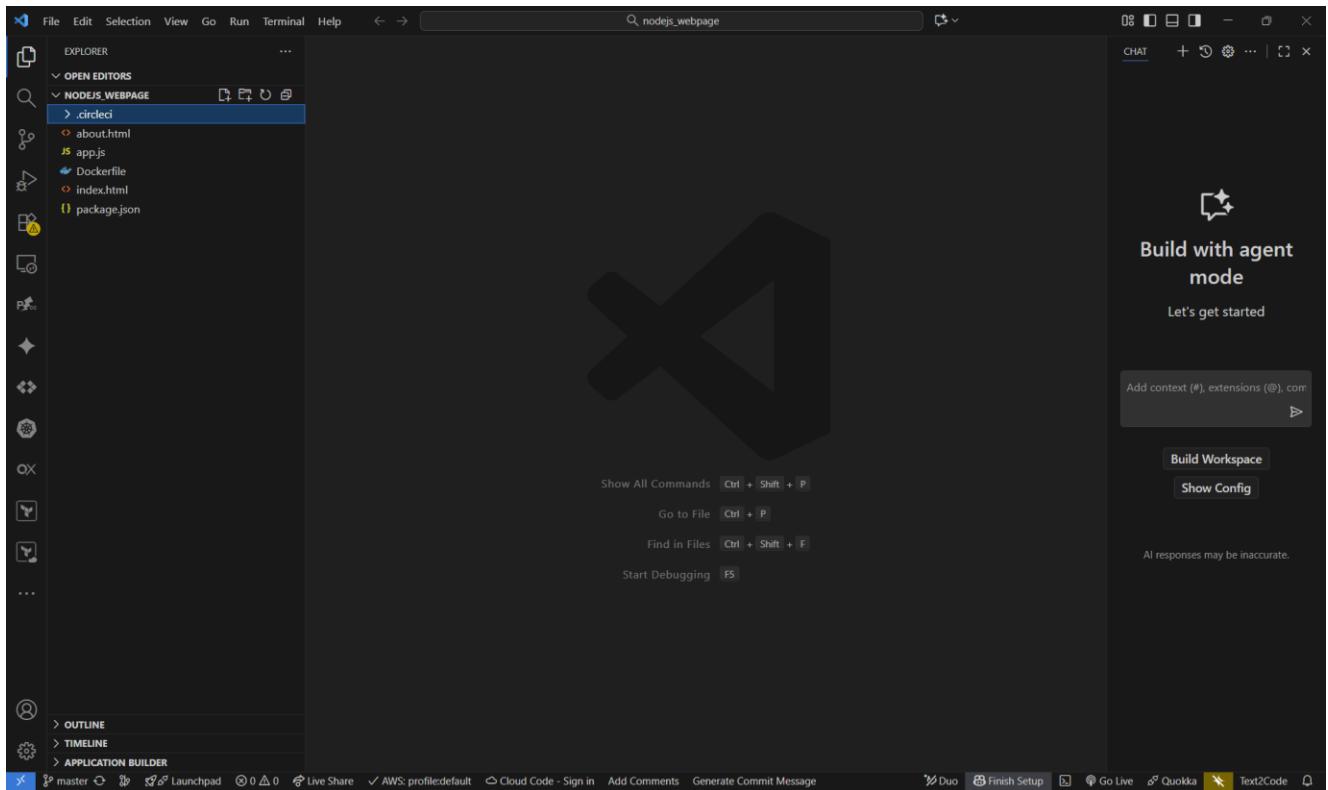
code .



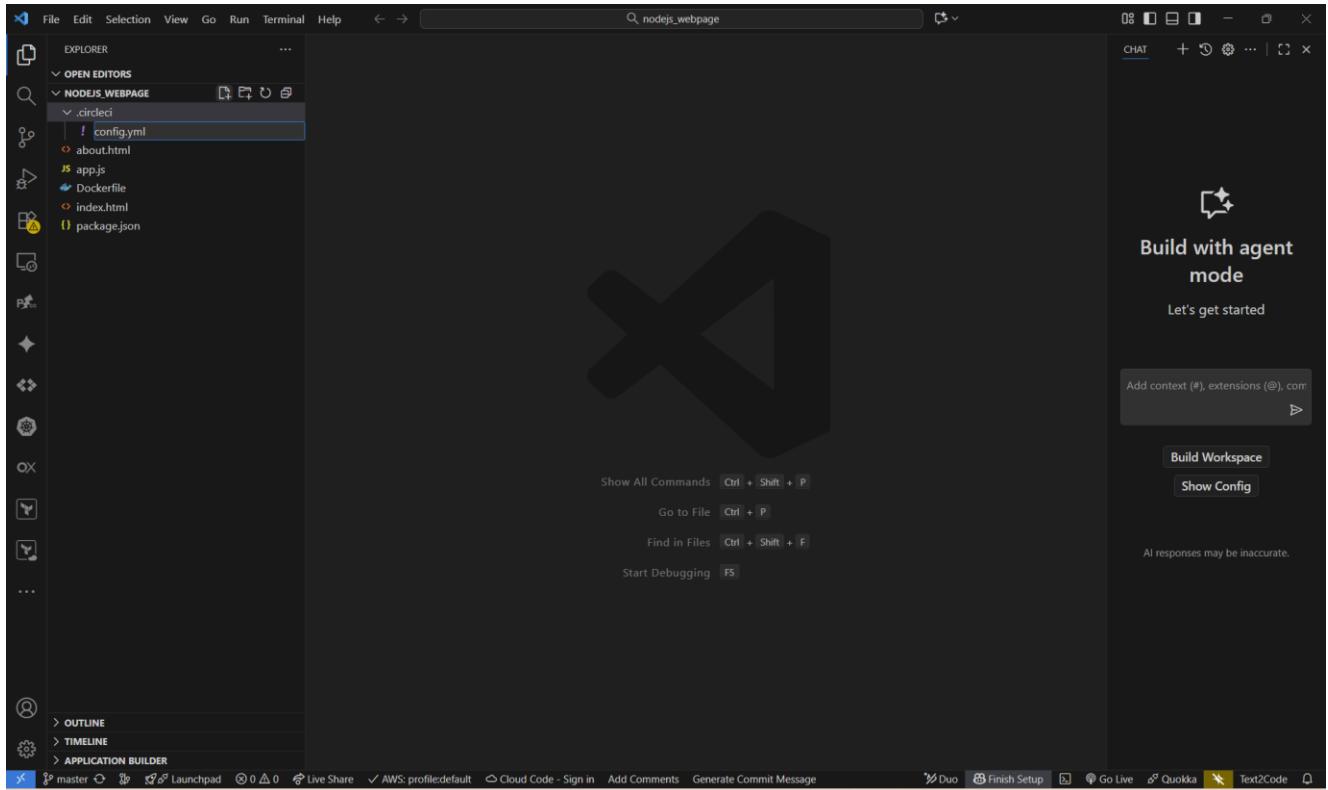
We have to create one folder and one file. The name of the folder will be “.circleci”



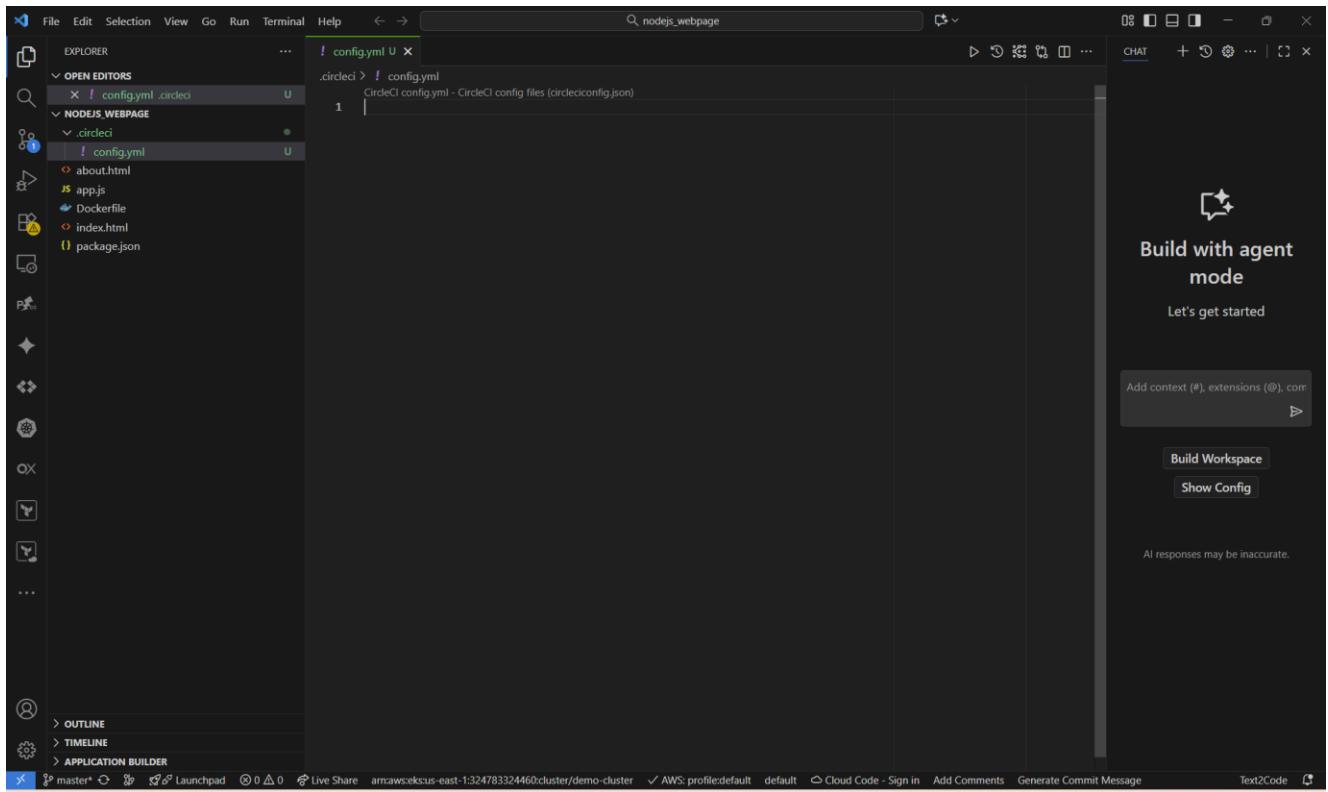
Press Enter



Then we have to create a file in this folder. We will call the file “**config.yml**”

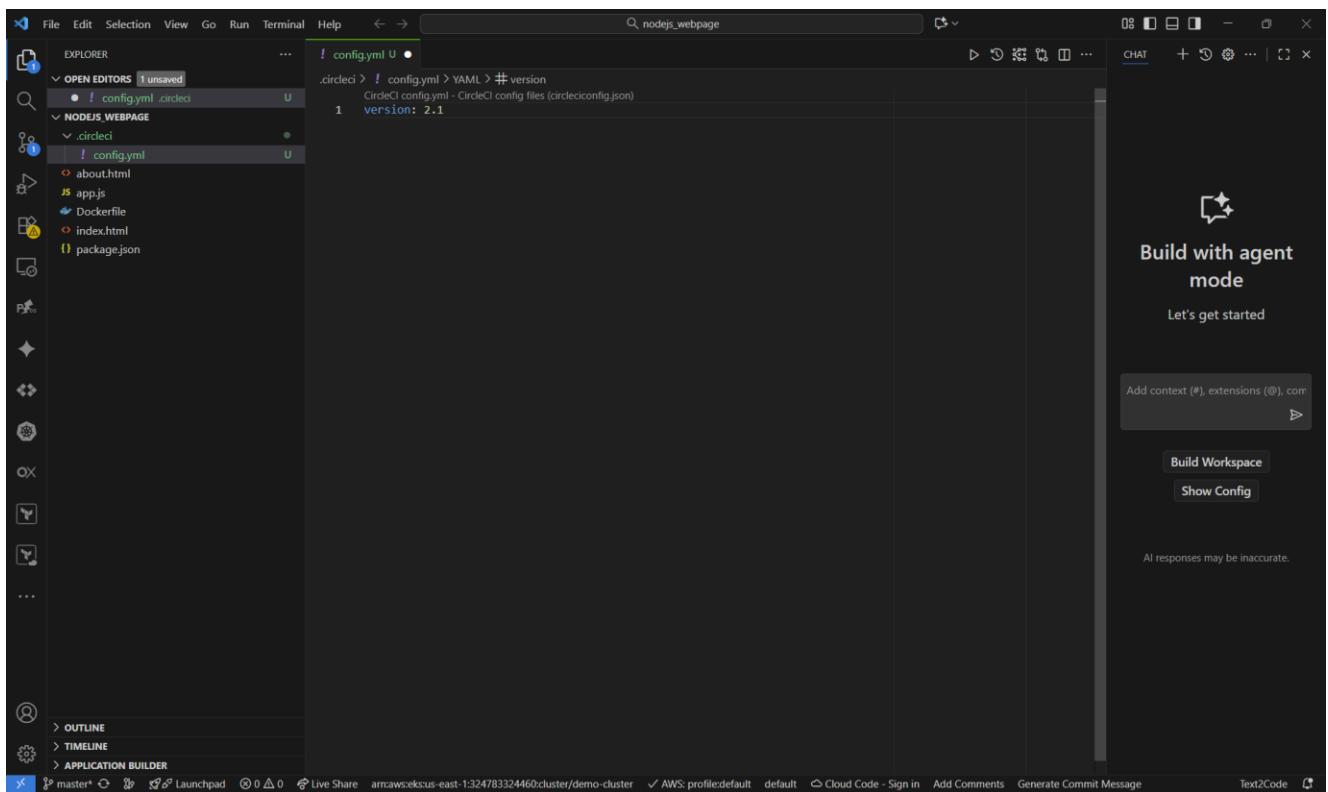


And press Enter



We have created the **config.yml** file. This is the file where we will write our Pipeline commands which needs to be executed.

In the pipeline, the first thing to do is to define the version of Nodejs. We will use **version 2.1**



The next thing is to add jobs. We will use two jobs in the Pipeline. One for the test and package and the other for the Docker image build and push.

### Part 1: Adding the “build\_test” job

For this project I am running the tests with Nodejs and npm. The first job will be called “**build\_test**”

```
build_test:  
  docker:  
    - image: circleci/node:12  
  steps:  
    - checkout  
    - run:  
        name: Install npm dependencies  
        command: |  
          npm install --save
```

### Part 2: Adding the “build\_docker\_image” job

This is where the application’s Docker image is built and pushed to Docker Hub. In a chronological order, it:

- Builds a Docker image from the Dockerfile in the current directory (.)
- Reads a password from the CircleCI environment variable and passes it to the docker login command for authentication purposes.
- Pushes the image to Docker Hub

We will add the second job called “**build\_docker\_image**”

```
build_docker_image:  
  docker:  
    - image: circleci/node:12  
  steps:  
    - checkout  
    - setup_remote_docker:  
        docker_layer_catching: false  
    - run:  
        name: Build Docker image  
        command: |  
          docker build -t ebotsidneysmith/nodejswebapp:latest .  
    - run:  
        name: Push application Docker image  
        command: |  
          echo "$DOCKER_PASSWORD" | docker login -u ebotsidneysmith --password-stdin  
          docker push ebotsidneysmith/nodejswebapp:latest
```

Final Code

```

version: 2.1

jobs:
  build_test:
    docker:
      - image: circleci/node:12
    steps:
      - checkout
      - run:
          name: Install npm dependencies
          command: |
            npm install --save

  build_docker_image:
    docker:
      - image: circleci/node:12
    steps:
      - checkout
      - setup_remote_docker:
          docker_layer_catching: false
      - run:
          name: Build Docker image
          command: |
            docker build -t ebotsidneysmith/nodejswebapp:latest .

    - run:
        name: Push application Docker image
        command: |
          echo "$DOCKER_PASSWORD" | docker login -u ebotsidneysmith --password-stdin
          docker push ebotsidneysmith/nodejswebapp:latest

workflows:
  build_test:
    jobs:
      - build_test
      - build_docker_image

```

```

version: 2.1

jobs:
  build_test:
    docker:
      - image: circleci/node:12
    steps:
      - checkout
      - run:
          name: Install npm dependencies
          command: |
            npm install --save

  build_docker_image:
    docker:
      - image: circleci/node:12
    steps:
      - checkout
      - setup_remote_docker:
          docker_layer_catching: false
      - run:
          name: Build Docker image
          command: |
            docker build -t ebotsidneysmith/nodejswebapp:latest .

    - run:
        name: Push application Docker image
        command: |
          echo "$DOCKER_PASSWORD" | docker login -u ebotsidneysmith --password-stdin
          docker push ebotsidneysmith/nodejswebapp:latest

workflows:
  build_test:
    jobs:
      - build_test
      - build_docker_image

```

## Save the config.yml file

The screenshot shows the CircleCI configuration interface. On the left, the file tree displays a project structure with files like config.yml, .circleci, app.js, Dockerfile, index.html, and package.json. The main pane shows the content of config.yml:

```
version: 2.1
jobs:
  build_test:
    docker:
      - image: circleci/node:12
    steps:
      - checkout
      - run:
          name: Install npm dependencies
          command:
            npm install --save
  build_docker_image:
    docker:
      - image: circleci/node:12
    steps:
      - checkout
      - setup_remote_docker:
          docker_layer_caching: false
      - run:
          name: Build Docker Image
          command:
            docker build -t nodejswebapp -t ebotsidneysmith/nodejswebapp:latest .
            echo $DOCKER_PASSWORD | docker login -u ebotsidneysmith --password-stdin
            docker push ebotsidneysmith/nodejswebapp:latest
workflows:
  build_test:
    jobs:
      - build_docker_image
```

The right side of the interface includes a sidebar with "Build with agent mode" and "Let's get started" buttons, and a bottom bar with various project management and sharing options.

## Part 3: Docker file

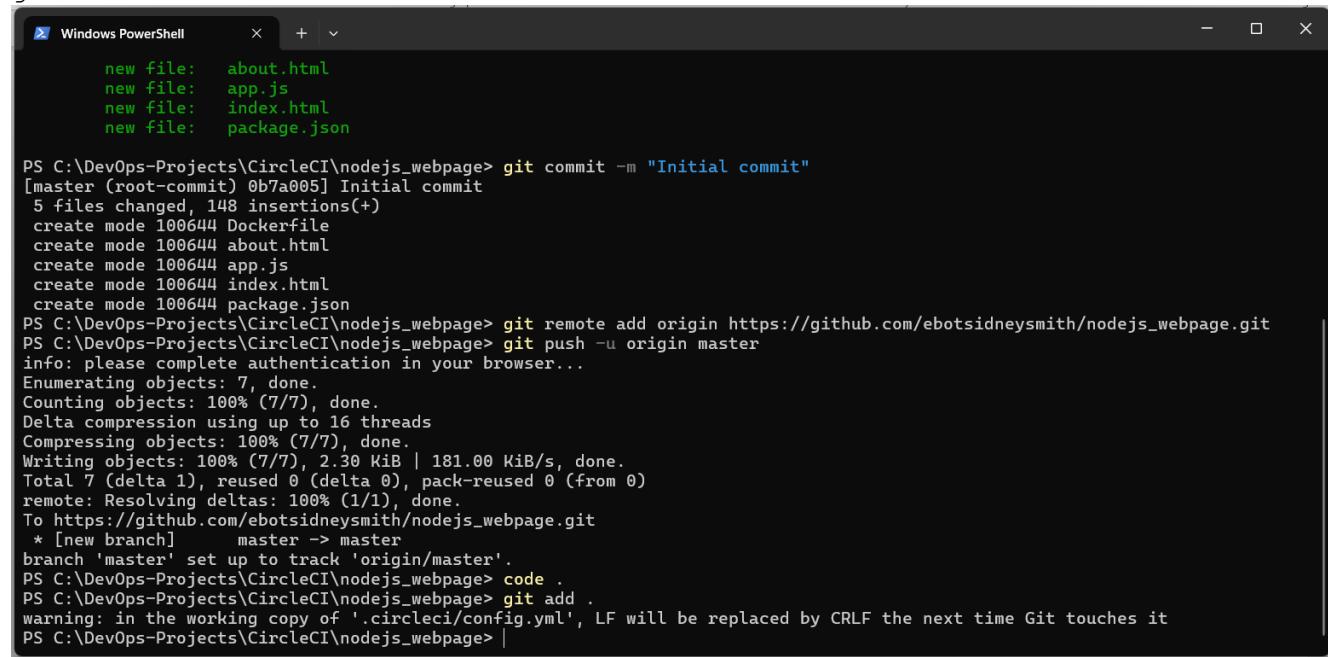
We also have to add a docker file for our project

```
FROM node:alpine
COPY ./ .
RUN npm install
EXPOSE 8081
CMD ["npm", "start"]
```

## STEP 5: Push the Pipeline yaml file to GitHub repository

Let us no head back to our PowerShell terminal and commit the changes. We first start by add the new file using the command:

```
git add .
```

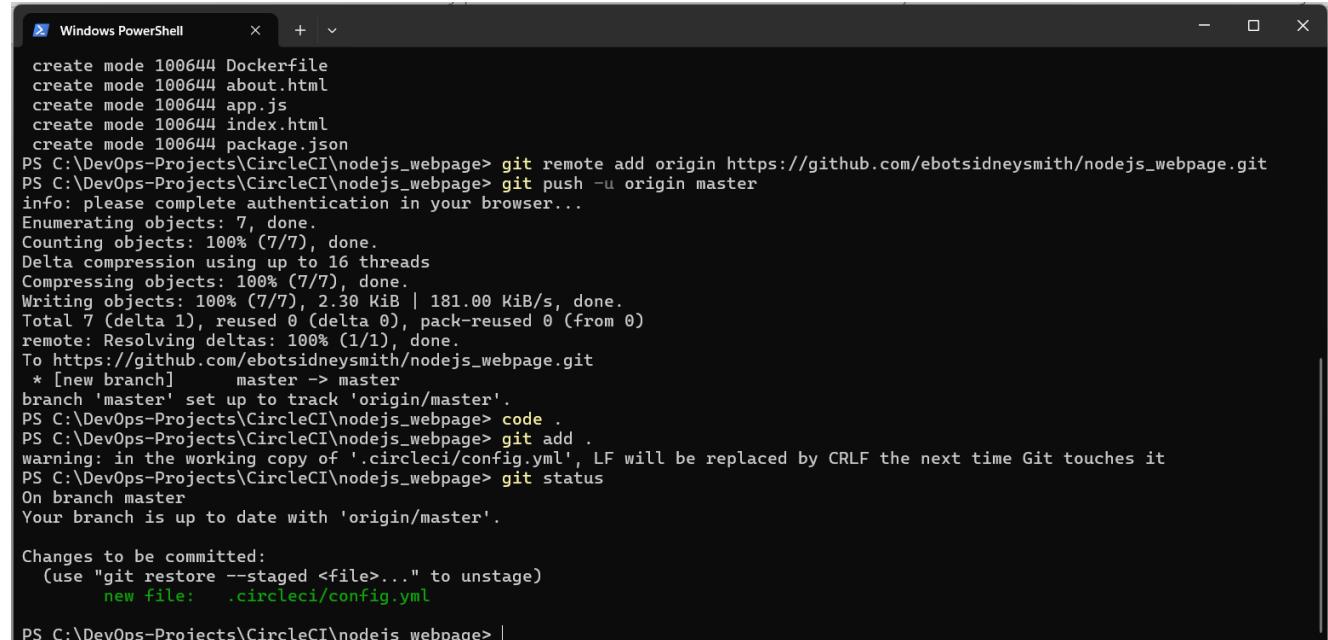


```
Windows PowerShell      +  x
new file:  about.html
new file:  app.js
new file:  index.html
new file:  package.json

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git commit -m "Initial commit"
[master (root-commit) 0b7a005] Initial commit
 5 files changed, 148 insertions(+)
create mode 100644 Dockerfile
create mode 100644 about.html
create mode 100644 app.js
create mode 100644 index.html
create mode 100644 package.json
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git remote add origin https://github.com/ebotsidneysmith/nodejs_webpage.git
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.30 KiB | 181.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ebotsidneysmith/nodejs_webpage.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> code .
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git add .
warning: in the working copy of '.circleci/config.yml', LF will be replaced by CRLF the next time Git touches it
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

Check the status using the command:

```
git status
```



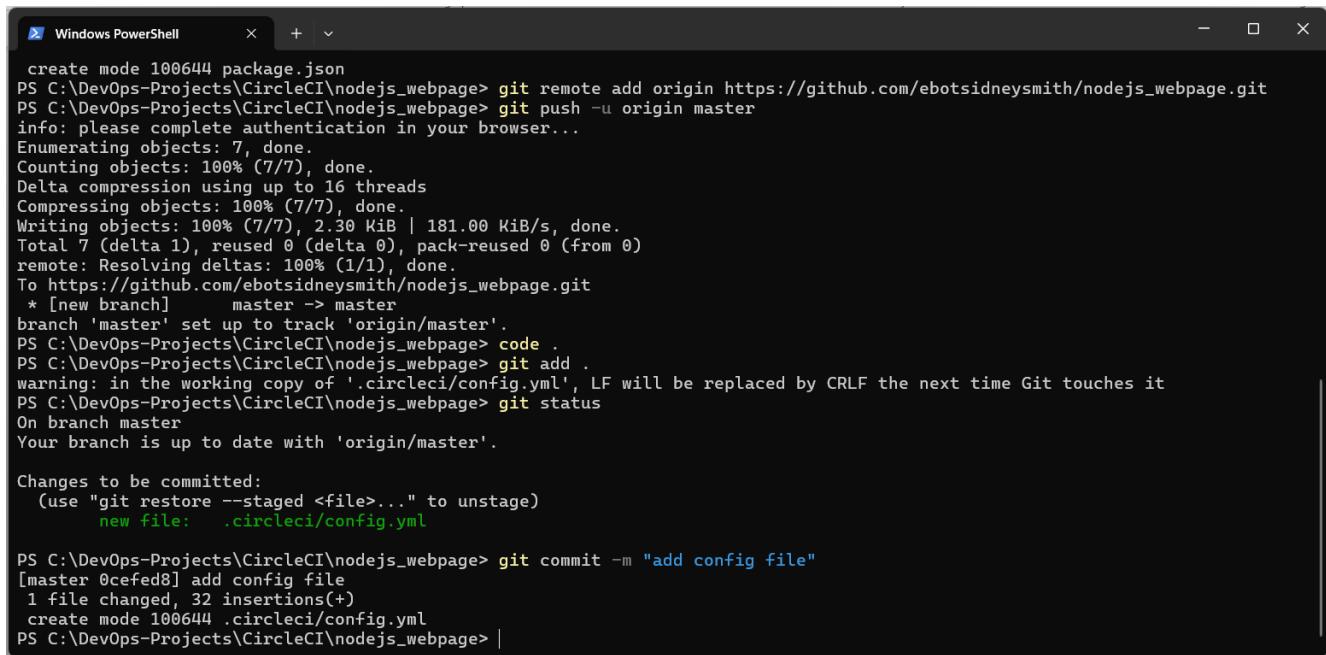
```
Windows PowerShell      +  x
create mode 100644 Dockerfile
create mode 100644 about.html
create mode 100644 app.js
create mode 100644 index.html
create mode 100644 package.json
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git remote add origin https://github.com/ebotsidneysmith/nodejs_webpage.git
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.30 KiB | 181.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ebotsidneysmith/nodejs_webpage.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> code .
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git add .
warning: in the working copy of '.circleci/config.yml', LF will be replaced by CRLF the next time Git touches it
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  .circleci/config.yml

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

Then commit the changes using the command:

```
git commit -m "add config file"
```



```
Windows PowerShell

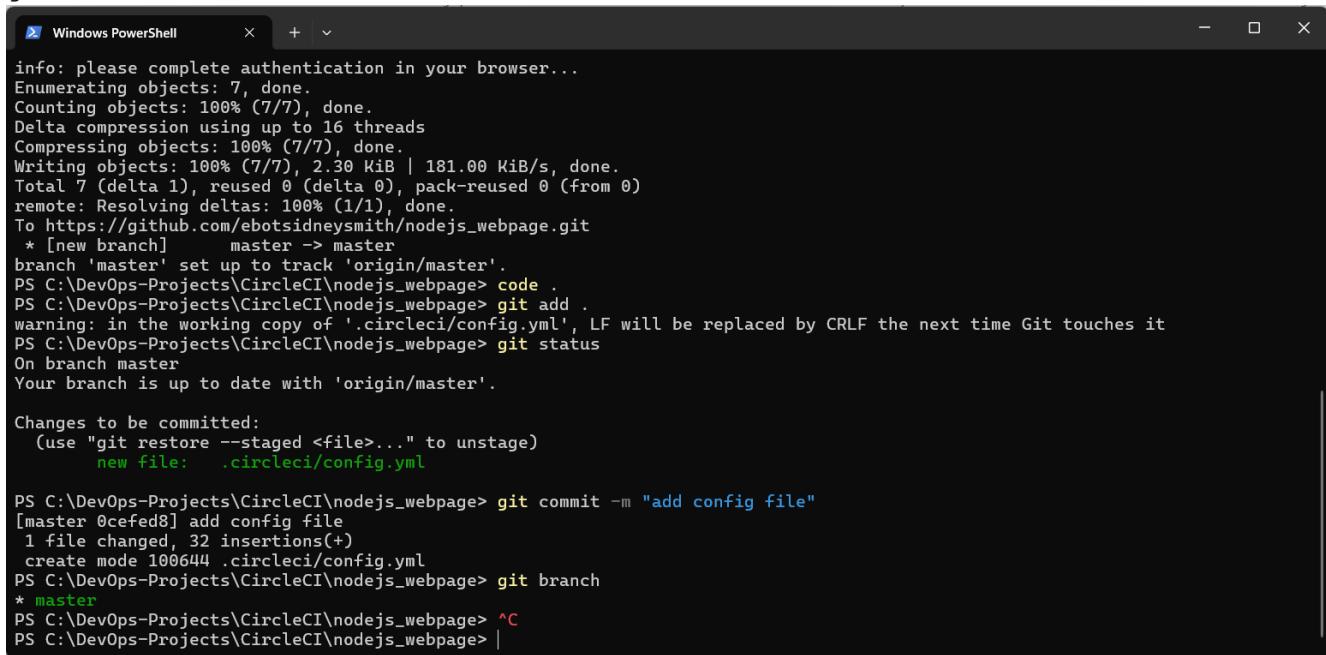
create mode 100644 package.json
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git remote add origin https://github.com/ebotsidneysmith/nodejs_webpage.git
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.30 KiB | 181.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ebotsidneysmith/nodejs_webpage.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> code .
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git add .
warning: in the working copy of '.circleci/config.yml', LF will be replaced by CRLF the next time Git touches it
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  .circleci/config.yml

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git commit -m "add config file"
[master 0cefef8] add config file
 1 file changed, 32 insertions(+)
 create mode 100644 .circleci/config.yml
PS C:\DevOps-Projects\CircleCI\nodejs_webpage>
```

Let us check the branch using the command

```
git branch
```



```
Windows PowerShell

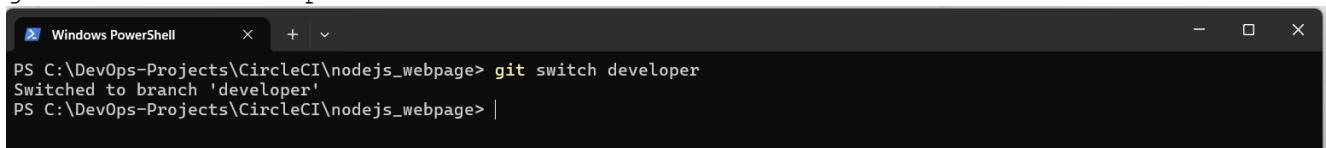
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.30 KiB | 181.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ebotsidneysmith/nodejs_webpage.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> code .
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git add .
warning: in the working copy of '.circleci/config.yml', LF will be replaced by CRLF the next time Git touches it
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  .circleci/config.yml

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git commit -m "add config file"
[master 0cefef8] add config file
 1 file changed, 32 insertions(+)
 create mode 100644 .circleci/config.yml
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git branch
* master
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> ^C
PS C:\DevOps-Projects\CircleCI\nodejs_webpage>
```

Let us switch to the developer branch using the command

```
git switch developer
```

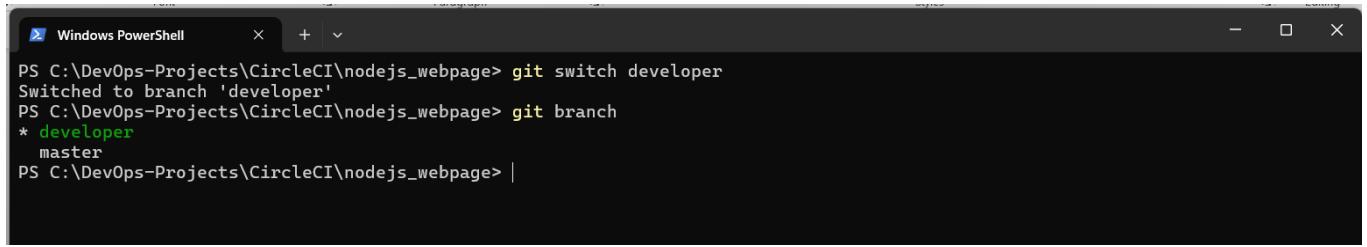


```
Windows PowerShell

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git switch developer
Switched to branch 'developer'
PS C:\DevOps-Projects\CircleCI\nodejs_webpage>
```

Check the branch again using the command:

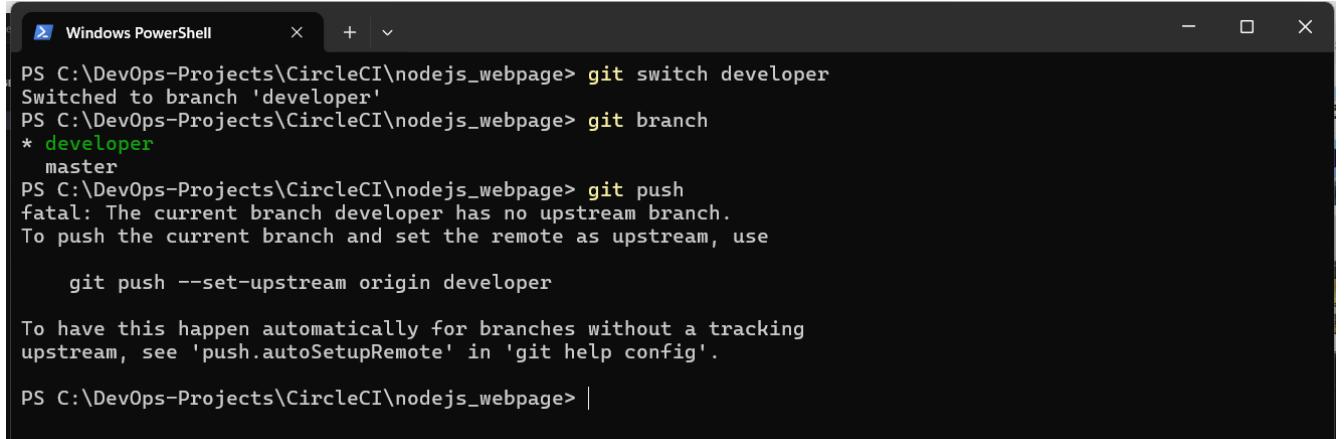
```
git branch
```



```
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git switch developer
Switched to branch 'developer'
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git branch
* developer
  master
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

You can see that we are in the developer branch. Let us push the committed file now using the command:

```
git push
```



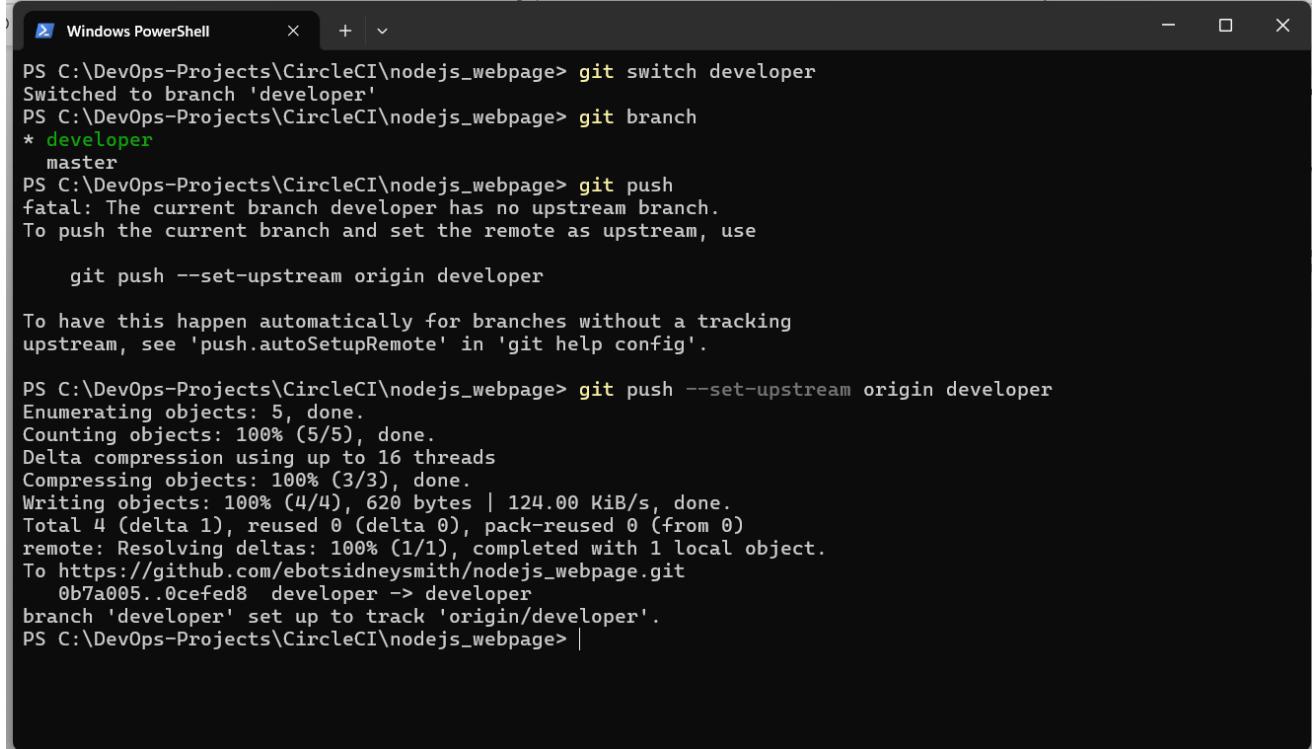
```
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git switch developer
Switched to branch 'developer'
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git branch
* developer
  master
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git push
fatal: The current branch developer has no upstream branch.
To push the current branch and set the remote as upstream, use

  git push --set-upstream origin developer

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

```
git push --set-upstream origin developer
```



```
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git switch developer
Switched to branch 'developer'
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git branch
* developer
  master
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git push
fatal: The current branch developer has no upstream branch.
To push the current branch and set the remote as upstream, use

  git push --set-upstream origin developer

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

PS C:\DevOps-Projects\CircleCI\nodejs_webpage> git push --set-upstream origin developer
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 620 bytes | 124.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ebotsidneysmith/nodejs_webpage.git
  0b7a005..0cefef8  developer -> developer
branch 'developer' set up to track 'origin/developer'.
PS C:\DevOps-Projects\CircleCI\nodejs_webpage> |
```

Go back to the GitHub repository and refresh the page

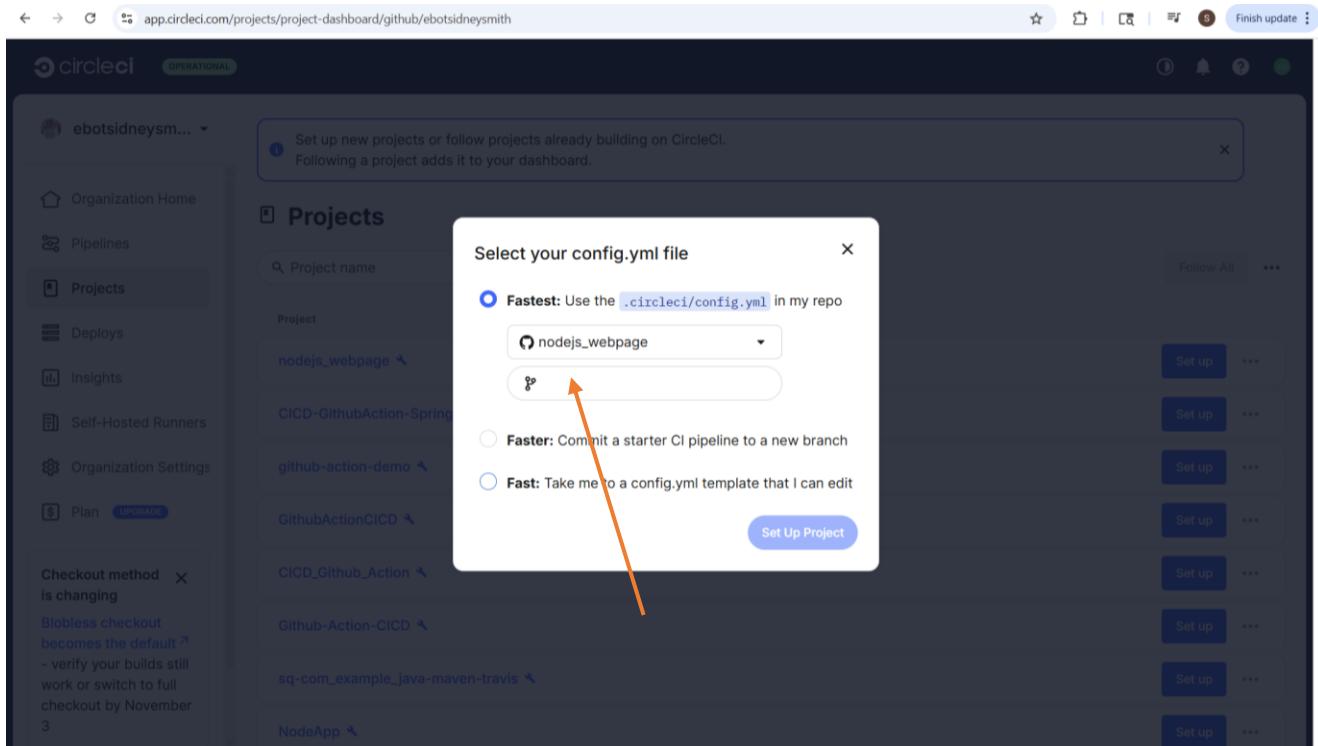
## Prepared by Sidney Smith Ebot

The screenshot shows a GitHub repository page for 'nodejs\_webpage'. At the top, there's a yellow banner indicating recent activity: 'developer had recent pushes 49 seconds ago'. Below this, the repository details show 2 branches and 0 tags. The commit history lists several initial commits from 'ebotsmith2000' for files like .circleci, Dockerfile, about.html, app.js, index.html, and package.json, all made 6 hours ago. A 'README' file is present. On the right side, there are sections for 'About' (no description), 'Releases' (no releases published), 'Packages' (no packages published), 'Languages' (HTML 91.4%, JavaScript 7.0%, Dockerfile 1.6%), and 'Suggested workflows' (Jekyll using Docker image). A red arrow points to the 'Set up' button for the 'nodejs\_webpage' project.

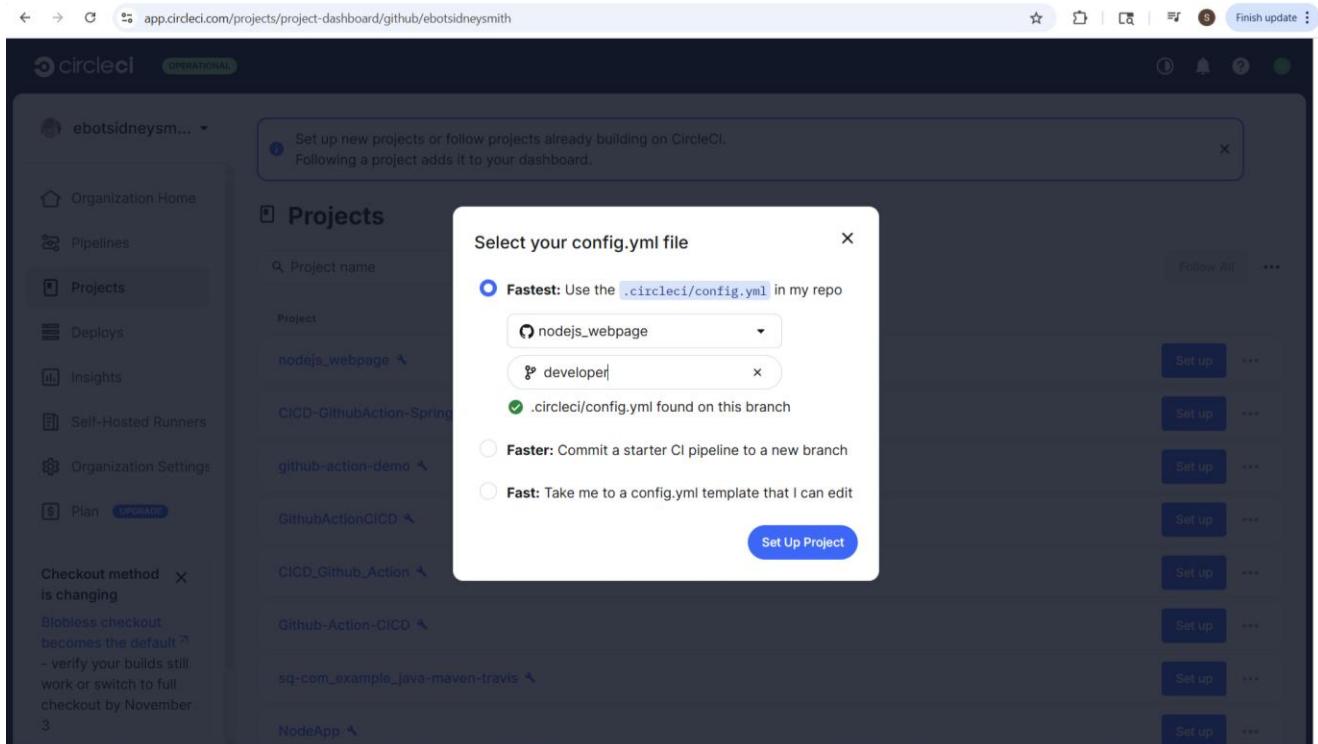
You can now see that the folder “.circleci” has been added. Go back to the CircleCi web browser

The screenshot shows the CircleCI web interface. On the left, a sidebar menu includes 'Organization Home', 'Pipelines', 'Projects' (which is selected and highlighted in grey), 'Deploys', 'Insights', 'Self-Hosted Runners', and 'Organization Settings'. A note at the bottom left says 'Checkout method is changing' and 'Blibless checkout becomes the default'. In the main area, a message box says 'Set up new projects or follow projects already building on CircleCI. Following a project adds it to your dashboard.' Below this, a 'Projects' section lists several projects: nodejs\_webpage, CICD-GithubAction-Springboot, github-action-demo, GithubActionCICD, CICD\_Github\_Action, Github-Action-CICD, sq-com\_example\_java-maven-travis, and NodeApp. Each project entry has a 'Set up' button and a three-dot menu icon. A red arrow points to the 'Set up' button for the 'nodejs\_webpage' project.

Click on “Set up”



Click on the GitHub icon and type “developer”



Then click on “set up Project”

The screenshot shows the CircleCI web interface for a project named "nodejs\_webpage". The pipeline status is "Running". The table below shows the pipeline details:

Pipeline	Status	Workflow	Checkout source	Trigger event	Start	Duration	Actions
nodejs_webpage 1	Running	build_test	developer Ocefed8	API	12s ago	14s	<a href="#">View</a> <a href="#">Logs</a> <a href="#">Cancel</a> <a href="#">...</a>

Under the "Jobs" section, there are two entries:

- build\_test 1: Status is green checkmark, Duration: 5s.
- build\_docker\_image 2: Status is grey question mark, Duration: 0s.

A sidebar message states: "Checkout method is changing. Blobless checkout becomes the default. - verify your builds still work or switch to full checkout by November 3".

You can see that the pipeline has started running

The screenshot shows the CircleCI web interface for the same project "nodejs\_webpage". The pipeline status is now "Failed". The table below shows the pipeline details:

Pipeline	Status	Workflow	Checkout source	Trigger event	Start	Duration	Actions
nodejs_webpage 1	Failed	build_test	developer Ocefed8	API	1m ago	34s	<a href="#">View</a> <a href="#">Logs</a> <a href="#">Cancel</a> <a href="#">...</a>

Under the "Jobs" section, there are two entries:

- build\_test 1: Status is green checkmark, Duration: 5s.
- build\_docker\_image 2: Status is red X, Duration: 1s.

A sidebar message states: "Checkout method is changing. Blobless checkout becomes the default. - verify your builds still work or switch to full checkout by November 3".

The pipeline failed. This is because we have not set up the Environment variables.

## STEP 6: Adding Environment Variables

We have to add environment variable for Docker

The screenshot shows the CircleCI pipeline interface. On the left, a sidebar lists 'Organization Home', 'Pipelines' (selected), 'Projects' (highlighted with an orange arrow), 'Deploys', 'Insights', 'Self-Hosted Runners', and 'Organization Settings'. A message box indicates 'Checkout method is changing' and 'Blobless checkout becomes the default - verify your builds still work or switch to full checkout by November 3'. The main area displays a pipeline named 'build\_docker\_image' which has failed. It shows the following steps: 'Spin up environment' (0s), 'Spin up container environment' (14s), 'Preparing environment variables' (0s), 'Checkout code' (0s), 'Setup a remote Docker engine' (0s), and 'Build Docker Image' (failed, 0s). The failed step's log output is '#!/bin/bash \_eo\_pipefail'. There are tabs for 'Steps', 'Tests', 'Timing', 'Artifacts', and 'Resources'.

Click on “Projects”

The screenshot shows the CircleCI project overview page. The sidebar is identical to the previous one. The main area shows a list of projects under the heading 'Projects'. The first project listed is 'nodejs\_webpage' (selected with an orange arrow), followed by 'CI\_CD-GithubAction-Springboot', 'github-action-demo', 'GithubActionCI\_CD', 'CI\_CD\_Github\_Action', 'Github-Action-CI\_CD', 'sq-com\_example\_java-maven-travis', 'NodeApp', and 'boardgame'. Each project entry includes a 'Set up' button and a '...' menu. A search bar at the top is labeled 'Project name'.

Click on “Overview”

## Prepared by Sidney Smith Ebot

The screenshot shows the CircleCI project dashboard for 'nodejs\_webpage'. On the left sidebar, under 'Projects', there is a message: 'Checkout method is changing. Blobless checkout becomes the default - verify your builds still work or switch to full checkout by November 3'. In the main area, under 'Latest Pipelines', a pipeline named 'nodejs\_webpage' is shown with a status of 'Failed'. A tooltip for the 'Settings' button is visible, pointing to the right.

Click on “Settings”

The screenshot shows the 'Overview' section of the CircleCI project settings for 'nodejs\_webpage'. The left sidebar lists various settings categories: Project Setup (selected), Deploy, Advanced, Environment Variables, SSH Keys, API Permissions, LLM Ops, Slack Integration, Insights Snapshot Badge, Status Badges, Webhooks, and Docker Layer Caching. The main panel displays basic project information: Project ID (524e403e-89bd-469d-a3a7-53f038916146), Project slug (gh/ebotsidneysmith/nodejs\_webpage), and a note about following the project. It also includes a section on how to configure the project, mentioning parallelism.

Click on “Environment Variables”

The screenshot shows the CircleCI project settings interface. On the left, a sidebar lists various project configuration options like Overview, Project Setup (marked as NEW), Deployments, Advanced, Environment Variables (which is selected and highlighted in grey), SSH Keys, API Permissions, LLM Ops, Slack Integration, Insights Snapshot Badge, Status Badges, Webhooks, and Docker Layer Caching. The main content area is titled "Environment Variables". It contains a brief description of environment variables and a note about sharing them across projects via Contexts. Two buttons are present: "Add environment variable" (highlighted with a blue arrow) and "Import variables".

Click on “Add Environment Variable”

The screenshot shows the "Add environment variable" dialog box overlaid on the CircleCI settings page. The dialog has fields for "Name" (containing "MY\_ENV\_VAR") and "Value\*" (containing "my-value-usd\\\$"). At the bottom right of the dialog are two buttons: "Cancel" and "Add environment variable" (highlighted with a blue arrow). The background settings page is visible, showing the "Environment Variables" section and the sidebar.

Here we will add the Docker password

The screenshot shows the CircleCI web interface for managing project settings. The left sidebar lists various configuration options like Overview, Project Setup, Deployments, Advanced, Environment Variables (which is selected and highlighted in grey), SSH Keys, API Permissions, LLM Ops, Slack Integration, Insights Snapshot Badge, Status Badges, Webhooks, and Docker Layer Caching. The main content area is titled "Environment Variables". A modal window titled "Add environment variable" is open, prompting the user to enter a name and value. The "Name" field contains "DOCKER\_PASSWORD" and the "Value\*" field contains "my-value-usd\\\$". At the bottom of the modal are two buttons: "Cancel" and "Add environment variable".

Then enter the password

This screenshot is similar to the previous one, showing the "Environment Variables" section of the CircleCI settings. The "Environment Variables" option is selected in the sidebar. A modal window titled "Add environment variable" is displayed, asking for a name and value. The "Name" field has "DOCKER\_PASSWORD" and the "Value\*" field has "....." (represented by several dots). An orange arrow points from the bottom right towards the "Add environment variable" button at the bottom right of the modal.

Click on “Add environment variable”

The Docker password has been added as a variable. Let us try to run the pipeline again

Click on “Rerun”

## Prepared by Sidney Smith Ebot

The screenshot shows a CircleCI pipeline interface. On the left, there's a sidebar with navigation links like Organization Home, Pipelines (which is selected), Projects, Deployments, Insights, Self-Hosted Runners, and Organization Settings. A prominent message box states: "Checkout method is changing. Blobless checkout becomes the default ↗ - verify your builds still work or switch to full checkout by November 3". The main area displays a pipeline named "build\_docker\_image" which has just completed successfully. It shows the following steps:

Step	Duration	Details
Spin up environment	0s	Success
Spin up container environment	29s	Success
Preparing environment variables	0s	Success
Checkout code	0s	Success
Setup a remote Docker engine	0s	Success
Build Docker Image	10s	Success

You can see that the Pipeline is successful. Let us go to Docker hub and check if the image is there.

The screenshot shows the Docker Hub interface. In the top bar, the URL is https://hub.docker.com/repositories/ebotsidneysmith. The sidebar on the left includes links for Hardened Images, Collaborations, Settings (Default privacy, Notifications), Billing, Usage, Pulls, and Storage. The main content area is titled "Repositories" and shows two entries:

Name	Last Pushed	Contains	Visibility	Scout
ebotsidneysmith/demoapp	4 days ago	IMAGE	Public	inactive
ebotsidneysmith/demorepo	17 days ago	IMAGE	Public	Inactive

At the bottom of the page, there's a cookie consent banner with options: Cookies Settings, Reject All, and Accept All Cookies.

Refresh the page

## Prepared by Sidney Smith Ebots

The screenshot shows the Docker Hub interface for the user ebotsidneysmith. On the left, there's a sidebar with options like Repositories, Hardened Images, Collaborations, Settings, Billing, Usage, Pulls, and Storage. The main area is titled 'Repositories' and shows three entries:

Name	Last Pushed	Contains	Visibility	Scout
ebotsidneysmith/nodejswebapp	8 minutes ago	IMAGE	Public	Inactive
ebotsidneysmith/demoapp	4 days ago	IMAGE	Public	Inactive
ebotsidneysmith/demorepo	17 days ago	IMAGE	Public	Inactive

An orange arrow points to the first repository, 'nodejswebapp'.

You can see that a new repository has been added to the docker hub. Click on it to check if the image is there.

The screenshot shows the details of the 'nodejswebapp' repository under the ebotsidneysmith namespace. The sidebar on the left is identical to the previous screen. The main page includes a general description, Docker commands (with a 'push' command example), and a 'Tags' section.

**General Tab (Visible):**

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	9 minutes

An orange arrow points to the 'Image' entry in the 'Type' column of the tags table.

You can see the tag of the image and type image.

## STEP 7: Verification

We have to verify if the Pipeline is automated. Let us make some changes in our GitHub repository and commit the changes.

The screenshot shows a GitHub repository page for 'nodejs\_webpage'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation is a search bar and a 'Code' dropdown menu. An orange arrow points from the text 'Click on the drop down on "Add file"' to the 'Add file' button in the 'Code' dropdown. The main content area displays a list of commits. One commit by 'ebotsidneysmith' titled 'Update config.yml' is highlighted. Below the commit list is a section titled 'Add a README' with a 'Add a README' button. To the right of the main content, there are sections for About, Releases, Packages, Languages, and Suggested workflows.

Let us create a text file called “**testfile.txt**” in our GitHub repository. Click on the drop down on “Add file”

This screenshot is identical to the one above, showing the same GitHub repository page for 'nodejs\_webpage'. The 'Add file' dropdown menu is open, and an orange arrow points to the '+ Create new file' option. The rest of the interface, including the commit list, README section, and sidebar, remains the same.

Select “Create new file”

## Prepared by Sidney Smith Ebots

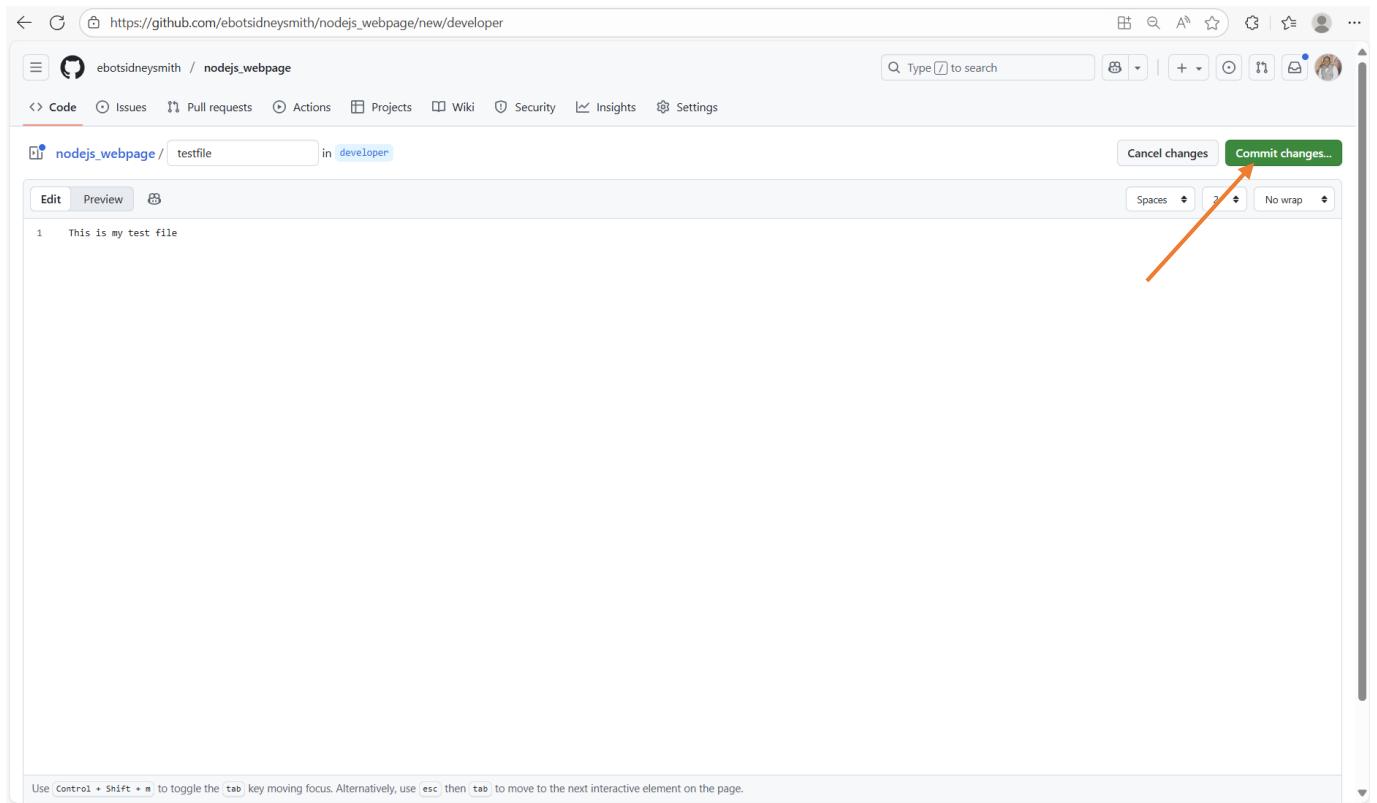
A screenshot of a GitHub code editor interface. At the top, the URL is https://github.com/ebotsidneysmith/nodejs\_webpage/new/developer. The repository name is ebotsidneysmith / nodejs\_webpage. The 'Code' tab is selected. A search bar at the top right contains the placeholder 'Type to search'. Below the search bar is a toolbar with icons for file operations like 'Copy', 'Search', 'Find', 'Star', 'Unstar', 'Watch', 'Unwatch', and 'Close'. A red arrow points from the text 'Enter the file name' to the input field where 'Name your file...' is typed. The input field has a dropdown menu showing 'nodejs\_webpage /' and 'in developer'. To the right of the input field are 'Cancel changes' and 'Commit changes...' buttons. Below the input field is a text area with the placeholder 'Enter file contents here'. At the bottom of the editor, there is a note: 'Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.'

Enter the file name “**testfile**”

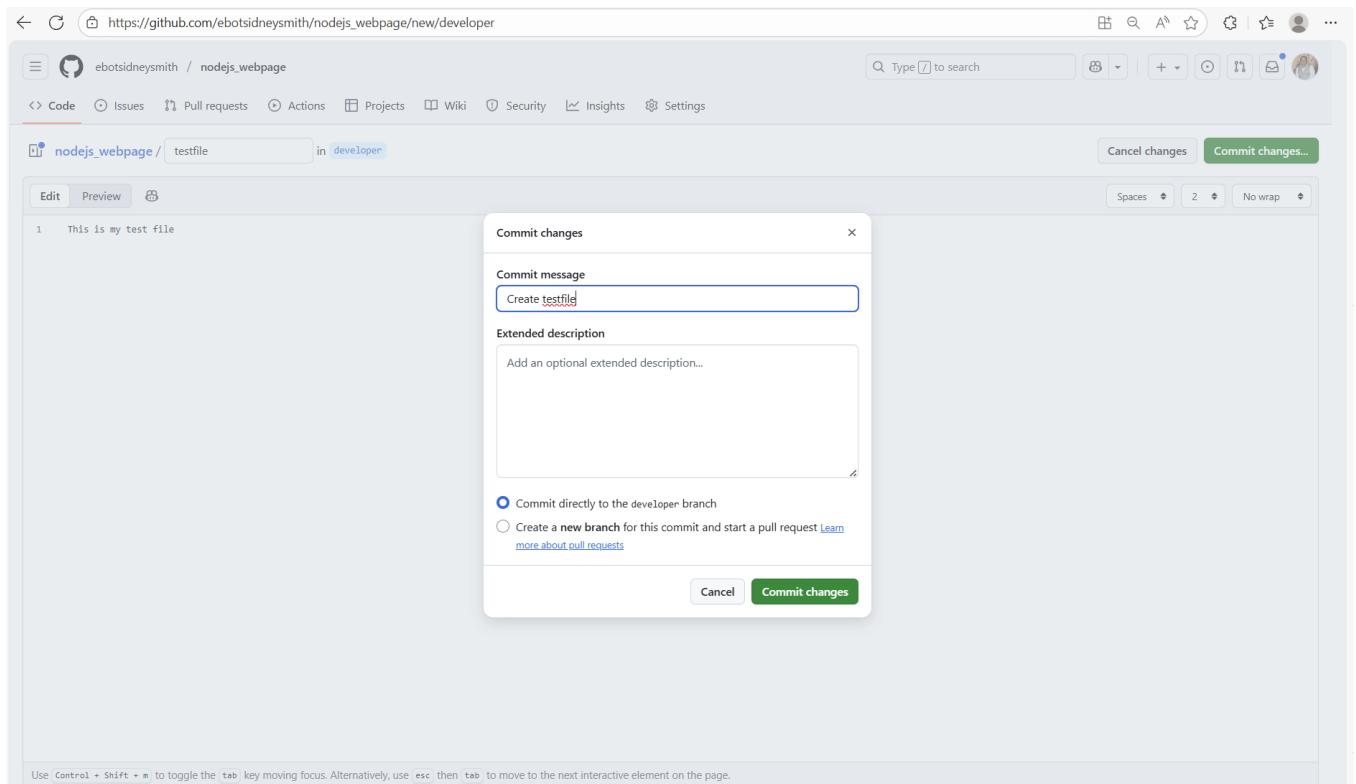
A screenshot of the same GitHub code editor interface after the file has been created. The URL remains https://github.com/ebotsidneysmith/nodejs\_webpage/new/developer. The repository name is ebotsidneysmith / nodejs\_webpage. The 'Code' tab is selected. The input field now shows 'testfile' and 'in developer'. The text area below is empty, with the placeholder 'Enter file contents here'. The bottom note is identical to the previous screenshot.

Add some text on the file, I will add “**This is my test file**”.

## Prepared by Sidney Smith Ebot



Click on “Commit Changes”



Click on “Commit Changes” again

## Prepared by Sidney Smith Ebot

The screenshot shows a GitHub repository page for 'ebotsidneysmith / nodejs\_webpage'. The 'developer' branch is selected. The commit history is as follows:

Name	Last commit message	Last commit date
.circleci	Update config.yml	2 hours ago
Dockerfile	Initial commit	15 hours ago
about.html	Initial commit	15 hours ago
app.js	Initial commit	15 hours ago
index.html	Initial commit	15 hours ago
package.json	Initial commit	15 hours ago
testfile	Create testfile	now

Head back to CircleCI browser

The screenshot shows the CircleCI web interface with the 'All Pipelines' view. There are three pipeline runs listed:

- Pipeline 13:** Status: Running, Triggered by 'Push Commit pushed' (791b2c2 Create testfile). It has two jobs: 'build\_test' (Success) and 'build\_docker\_image' (Running).
- Pipeline 12:** Status: Success, Triggered by 'Push Commit pushed' (dacac2b Update config.yml). It has two jobs: 'build\_test' (Success) and 'build\_docker\_image' (Success).
- Pipeline 11:** Status: Failed, Triggered by 'Push Commit pushed' (f348ebf Update config.yml). It has two jobs: 'build\_test' (Success) and 'ld\_docker\_image' (Running).

A message in the bottom right corner says: "Reconnected to Internet. Please refresh for updates."

You can see that our pipeline has started running. So, our pipeline is automated.

The screenshot shows the CircleCI interface for managing pipelines. On the left, there's a sidebar with links for Organization Home, Pipelines (which is selected), Projects, Deployments, Insights, Self-Hosted Runners, Organization Settings, and a Plan section with an 'UPGRADE' button. A note about 'Checkout method' changing to 'blobless checkout' is displayed. Below the sidebar, the URL is https://app.circleci.com/pipelines/github/ebotsidneysmith/.

The main area is titled 'All Pipelines' and lists three recent runs for the 'nodejs\_webpage' pipeline:

Pipeline	Status	Workflow	Checkout source	Trigger event	Start	Duration	Actions
nodejs_webpage 13	Success	build_test	developer 791b2c2 Create testfile	Push Commit pushed	35m ago	30s	<a href="#">View</a> <a href="#">Logs</a> <a href="#">X</a> ...
nodejs_webpage 12	Success	build_test	developer dacac2b Update config.yml	Push Commit pushed	3h ago	1m 1s	<a href="#">View</a> <a href="#">Logs</a> <a href="#">X</a> ...
nodejs_webpage 11	Failed	build_test	developer f348ebf Update config.yml	Push Commit pushed	3h ago	23s	<a href="#">View</a> <a href="#">Logs</a> <a href="#">X</a> ...

Each pipeline run has a 'Jobs' section below it, showing the status of individual build steps: 'build\_test' and 'build\_docker\_image'. The first two runs show both steps as successful (green checkmarks), while the third run shows the 'build\_docker\_image' step as failed (red X).

You can see that the Pipeline is successful