**Setup a sample project with CI/CD using CircleCI**

Setting up a sample project with CI/CD using CircleCI involves several steps. In this guide, I'll walk you through the process of setting up a basic Node.js project with CircleCI. You can adapt these steps to your specific project and technology stack.

**Prerequisites:**

1. A GitHub or Bitbucket account with a repository where your project is hosted.

2. Node.js and npm installed on your local machine.

3. A CircleCI account.

**Step 1: Create a Node.js Project**

First, create a simple Node.js project if you don't have one already. You can use the following commands:

**Bash code**:

mkdir sample-nodejs-project

cd sample-nodejs-project

npm init -y

npm install express # Or any other dependency you want to test

Create an “index.js” file with some code that you want to test using CI/CD.

**Step 2: Version Control**

Make sure your project is hosted on a version control platform like GitHub or Bitbucket. Initialize a Git repository, commit your code, and push it to the remote repository.

**Bash code:**

git init

git add .

git commit -m "Initial commit"

git remote add origin <your-repo-url>

git push -u origin master

**Step 3: Set Up CircleCI**

1. Log in to your CircleCI account.

2. Click on "Add Projects" on the dashboard.

3. Find your repository and click "Set Up Project."

**Step 4: Create a ‘.circleci/config.yml’ file**

This file defines your CI/CD pipeline. Here's a basic example for a Node.js project:

**yaml code:**

version: 2.1

jobs:

build:

docker:

- image: circleci/node:14 # Choose your preferred Node.js version

steps:

- checkout # Check out your code

- run:

name: Install Dependencies

command: npm install

- run:

name: Run Tests

command: npm test

This configuration specifies a single job called "build" that uses a Node.js Docker image, installs dependencies, and runs tests. You can modify this file to fit your project's needs, such as building, testing, and deploying.

**Step 5: Push to Trigger CI/CD**

Once you have your `config.yml` in place, push it to your repository. This will trigger a CI/CD build on CircleCI. You can monitor the build progress on the CircleCI dashboard.

**Step 6: Add Deployment Steps**

If you want to deploy your application as part of the CI/CD pipeline, you can add deployment steps to your `config.yml` file. This might involve deploying to a cloud platform, a server, or a container registry, depending on your project's needs.

**Step 7: Review and Iterate**

Review the CI/CD pipeline's output to ensure everything is working as expected. If there are issues, update your code or configuration as needed and push the changes to your repository to trigger another build.

That's it! You've set up a basic CI/CD pipeline for a Node.js project using CircleCI.