Application Deployment

Relevel
by Unacademy

Educator Introduction

In the last session, we learned about unit testing for our application using Jest. In this session, we will learn about deploying the app.

List of Concepts Involved

- Intro of GitHub.
- Repository Creation.
- Get Database for production.
- Intro to Heroku.
- Understand the deployment process of our Ecommerce App.

What is Hosting?

- Now that our project is working fine in the local system. It would be useful only when used by others, i.e., it is available to the world. Our purpose can be achieved when we make our application available to the users via the internet and that is what hosting means.
- There are infrastructure, platforms, or software that are hosted by third-party providers and made available to users for various needs through the internet which is called cloud services.
- Many cloud service providers help us rent their server where our application code runs 24/7. These services are provided based on a free trial and price-based model for individuals and organizations—for example, Amazon AWS, Heroku, Microsoft Azure, Google Cloud, etc.
- We will deploy our application using the services of Heroku.

Run a project locally

- Before deploying the project, ensure it’s running as expected by running the code and accessing the app locally.

- Execute below command in cmd terminals of root directory:

- **npm start**
- `"start": "node server",`
- When we run npm start from terminal it will run above script which we need to add in package.json file which will be responsible to start up the application, moreover this is the standard script which many applications hosting service understand for a node application as a run command.

Intro to GitHub

- Now, you can serve your local setup online and make it live via the internet, but it's not easy to maintain and make your server available 24/7. That's why we need some remote server to deploy it and forget, so the cloud/server provider will maintain our application for us.
- We need to take care of one thing before deploying the app, i.e. maintenance and continuous changes to the application.
- We will likely add more features or make some critical changes to our code in the future or maintain different versions of our code.
- Also, things work differently in the industry. There will be many contributors for a single project; frontend developers, UI designers, backend developers, and software testers all working on the same project but locally on their respective machines.
- Finally, all the code from these machines will be merged as a whole, maintaining the flow and integrity of the code and won't break the code's functionality.
- So, here GitHub comes into the picture. As we can integrate Github with the remote server and the local setup to reflect the changes, we can easily change it without going to the server.

Here we will see how Github works and all the basic functionality/commands of GitHub:

1. **Creating Repositories:**
   a. **git init:** to initialise the repo.
2. **Creating branches**
   a. **git checkout -b <branch_name> :** to create new branch
   b. **git checkout <existing_branch_name> :** to get back to the original branch.
3. **Adding files:**
   a. **git add . :** to add all items
   b. **git add <file_name> :** to add specific elements
4. **Commits**
   a. **git commit -m <commit message> :** to commit the changes locally

5. **Pull**

   a. **git pull origin <branch_name> :** to get the latest code from the repo

6. **Push**

   b. **git push origin <branch_name>:** to push local code to the remote repo.

We have 3 ways to do all these steps:

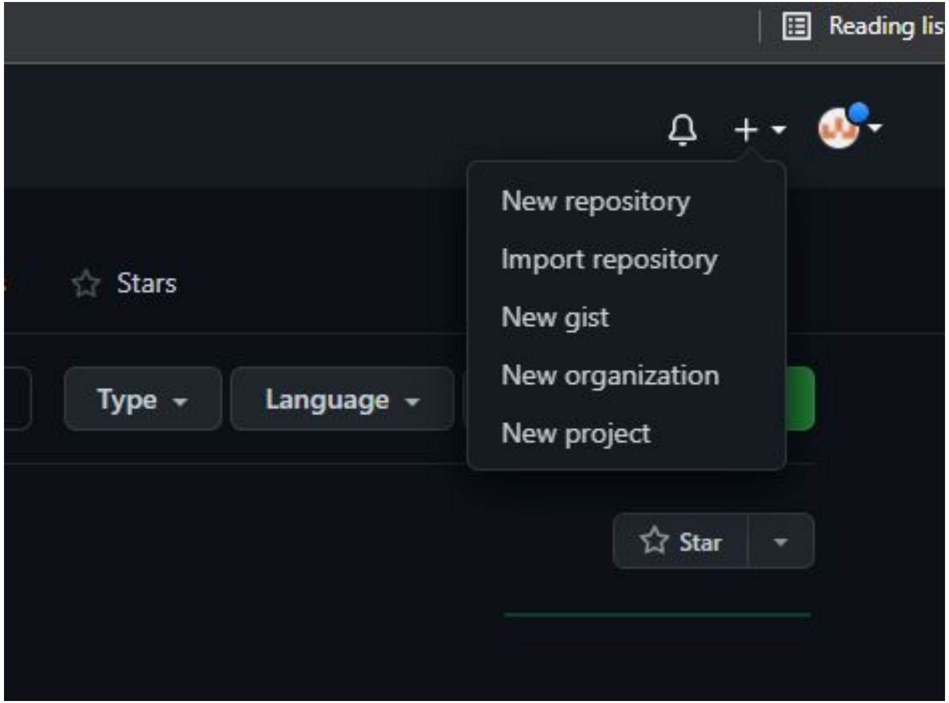
1. Web Interface: Very easy to use but not for complicated steps like resets, rebase, etc.
2. Command Line: Not much user-friendly, but we can perform every operation using this one.
3. Github Desktop: Very much similar to web interface in terms of UI,

Creating Repositories

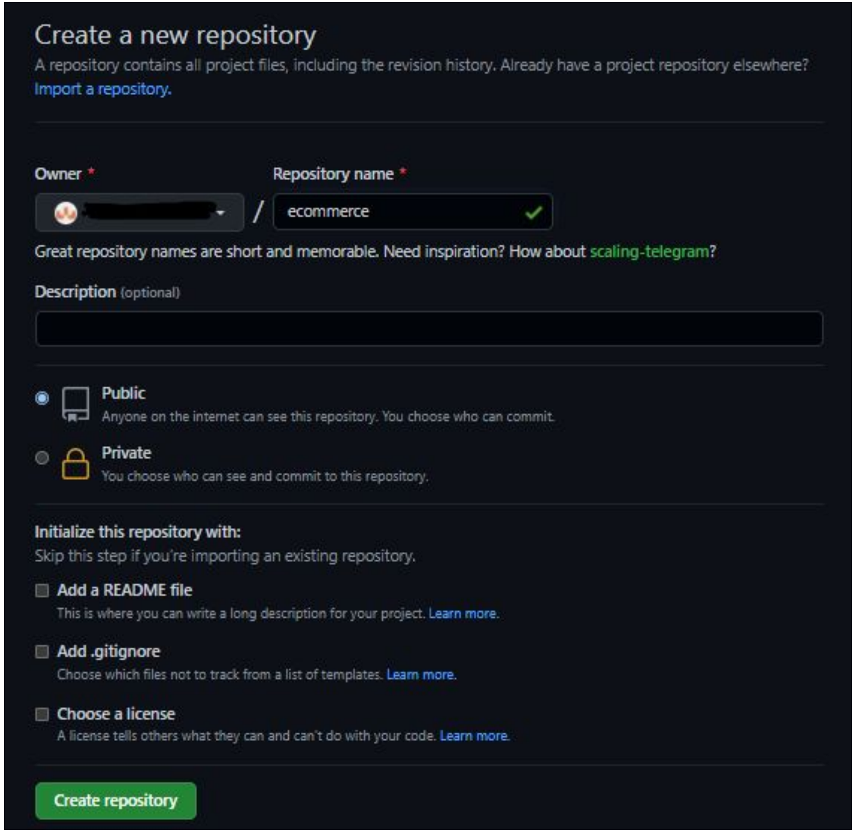
- **Using Web Interface:**

a. Go to **Repositories** click on **“New”** or on the **“+”** icon in the top-right corner.

b. Click on **New repository**



c. Name the repo, create it.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * [redacted] / Repository name * ecommerce ✓

Great repository names are short and memorable. Need inspiration? How about scaling-telegram?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

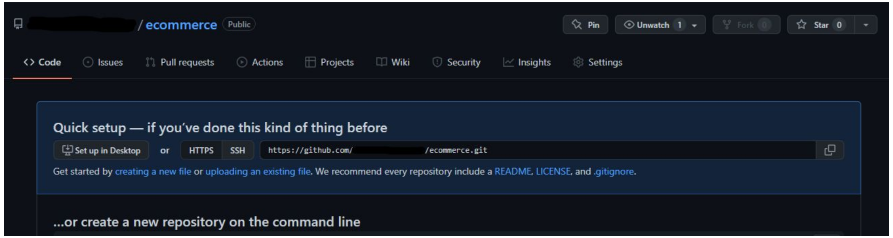
□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

d. Clone it to local, open cmd in the desired directory and write command

git clone <*urlname*>



[redacted] / ecommerce Public

Quick setup — if you've done this kind of thing before

```
https://github.com/[redacted]/ecommerce.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

e. If you have existing code in the local machine that you want to push to Github it can be done by pasting the folders/files in this cloned directory and we can commit and push the files to the remote repo, using command line.

2. Using the Command line:

a. Create a directory in local or open cmd for the root directory of an existing project.

b. Type **git init** will make our current directory a local git repository.

  - Note: Explain to students the difference between a local repo and remote repo,
  - Local repo is the app code present in git repo in local machine
  - Remote repo is where collaborators push/merge their code

a. We can add and commit the existing project files to the remote repository.

b. Type **git add.** (to add all the files) or just **git add <file_name>** you want to push.

c. Type **git commit -m <commit message>** where commit message is a string that informs why this commit happens and its purpose.

d. Now we have to connect it to GitHub:

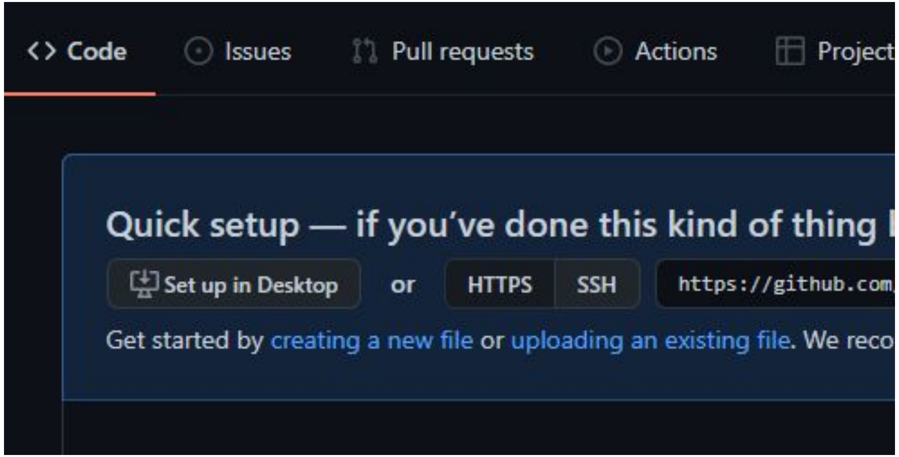
- **git remote add origin git@github.com:username/<repo_name>**

a. Finally, push using: **git push origin master.**

3. Using the Github desktop:

a. We need to install the Github desktop application for this.

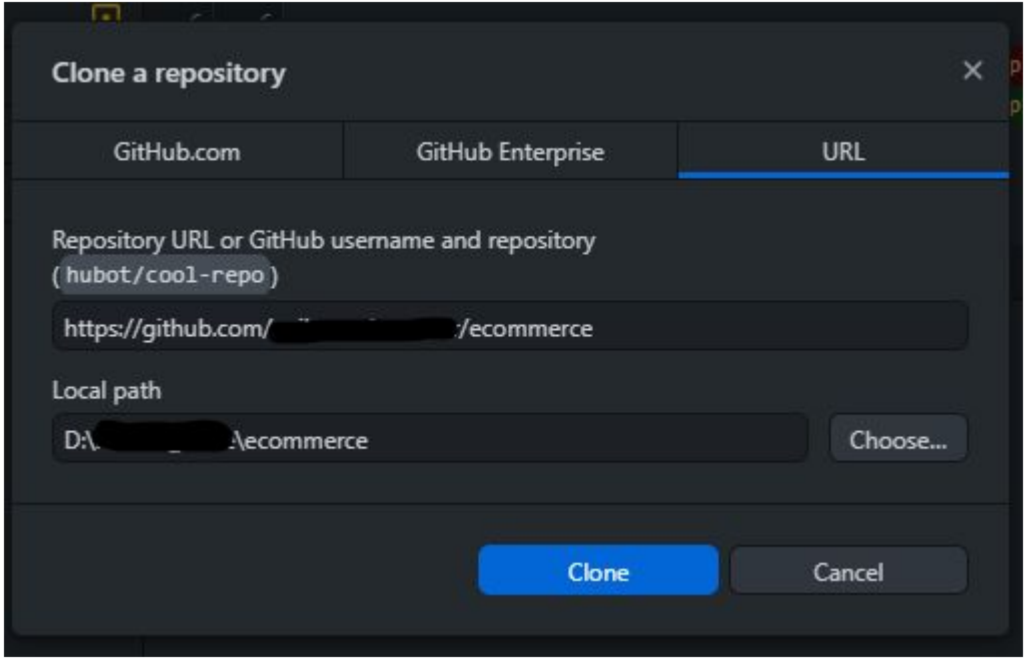
b. To create a new repo Click on **Set up in Desktop.**



Quick setup — if you've done this kind of thing

Get started by creating a new file or uploading an existing file. We reco

c. Choose the path and click on clone.



Clone a repository

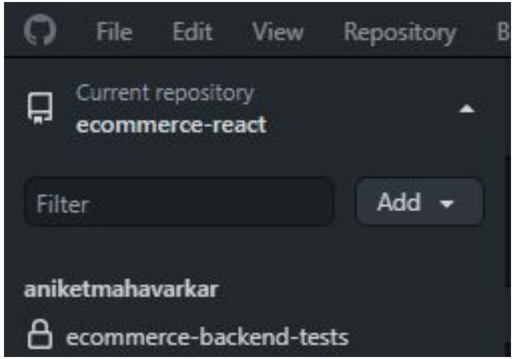
Repository URL or GitHub username and repository
(`hubot/cool-repo`)

https://github.com/[redacted]/ecommerce

Local path

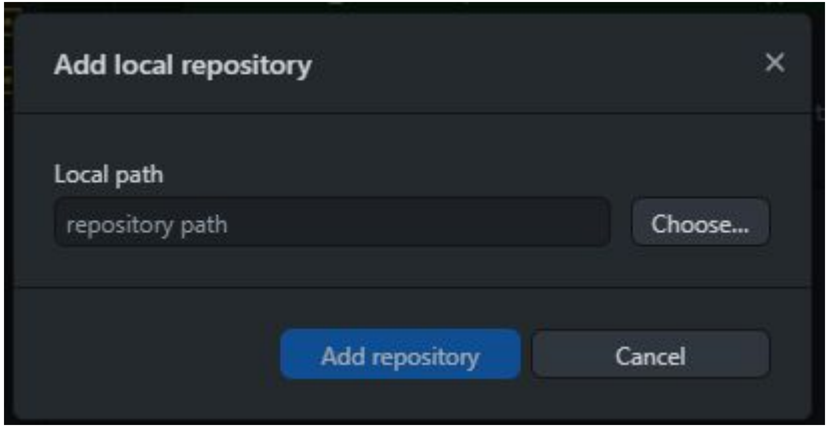
D:\[redacted]\ecommerce

d. To add an existing project or make existing project as local repo click on **Add** in the top left corner of Github desktop application.



e. Click on **Add existing repository**

f. Choose the project folder in the path



Add local repository

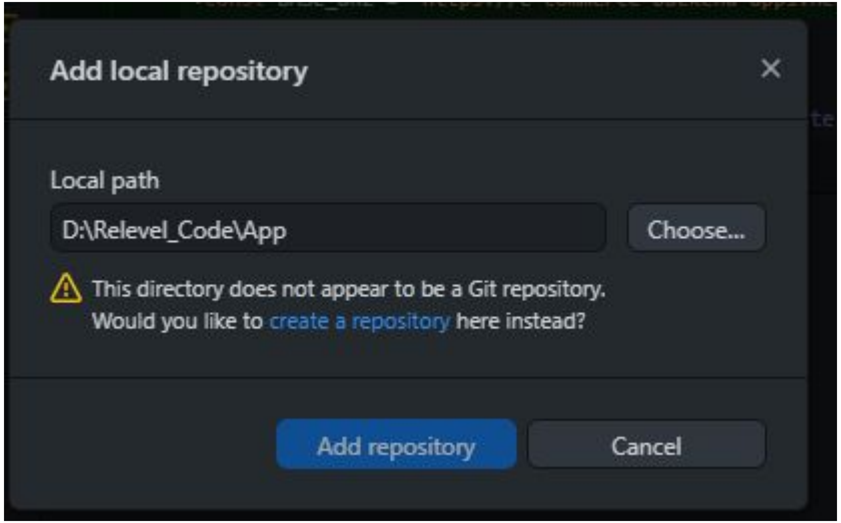
Local path

repository path

g. When your directory is already not a repository it will prompt to make it a repo by asking create a repository here instead click on this and your repository will be created.

code:

https://github.com/Vishwa07dev/eCommerce



Add local repository

Local path

D:\Relevel_Code\App

This directory does not appear to be a Git repository.
Would you like to create a repository here instead?

Get Hosted MySQL Database:

Why hosted DB

When we will be going to host our application on Heroku or any other platform it will be needing a database which will be live 24x7. Now while developing the application, we used MySQL in our local machine which was running in background and serving as hosted DB, which will not be working for application hosted on cloud service app, so we will be opting for a MySQL database hosting service here.

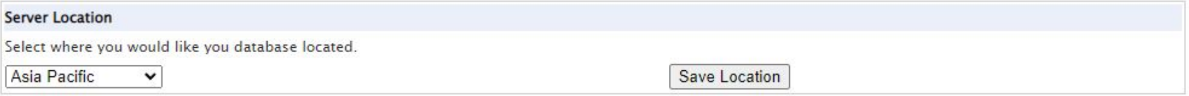
Get Hosted MySQL Database:

Step 1: Go to this website https://www.freemysqlhosting.net/ and register using email, it will send a link to authenticate.

Step 2: Check your email for the mail from this website and click the link on the email

Step 3: This link will redirect you to change password page, update the password and login to your account now.

Step 4: once logged in you will be in https://www.freemysqlhosting.net/account/ where first setup your server location and click on save location

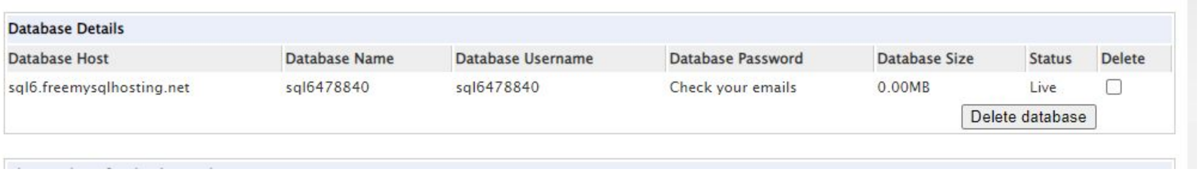


Server Location

Select where you would like you database located.

Asia Pacific

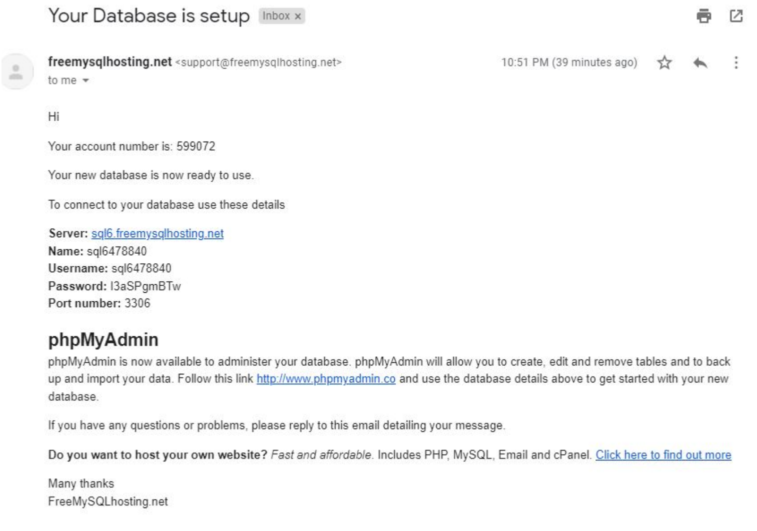
Step 5: After few second you will ser this block below:



Database Details

Database Host	Database Name	Database Username	Database Password	Database Size	Status	Delete
sql6.freemysqlhosting.net	sql6478840	sql6478840	Check your emails	0.00MB	Live	□

Step 6: Check your email and you will get your database connection credentials:



Your Database is setup

freemysqlhosting.net <support@freemysqlhosting.net>
to me

10:51 PM (39 minutes ago)

Hi

Your account number is: 599072

Your new database is now ready to use.

To connect to your database use these details

Server: sql6.freemysqlhosting.net
Name: sql6478840
Username: sql6478840
Password: l3aSPgmBTw
Port number: 3306

phpMyAdmin

phpMyAdmin is now available to administer your database. phpMyAdmin will allow you to create, edit and remove tables and to back up and import your data. Follow this link http://www.phpmyadmin.co and use the database details above to get started with your new database.

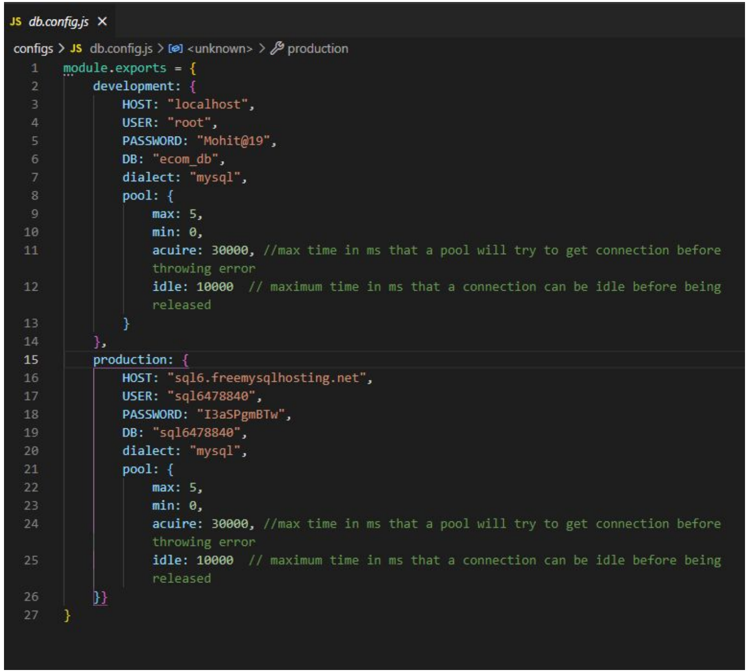
If you have any questions or problems, please reply to this email detailing your message.

Do you want to host your own website? *Fast and affordable.* Includes PHP, MySQL, Email and cPanel. Click here to find out more

Many thanks
FreeMySQLhosting.net

Step 7: Add those creds in your eCommerce application as shown below:

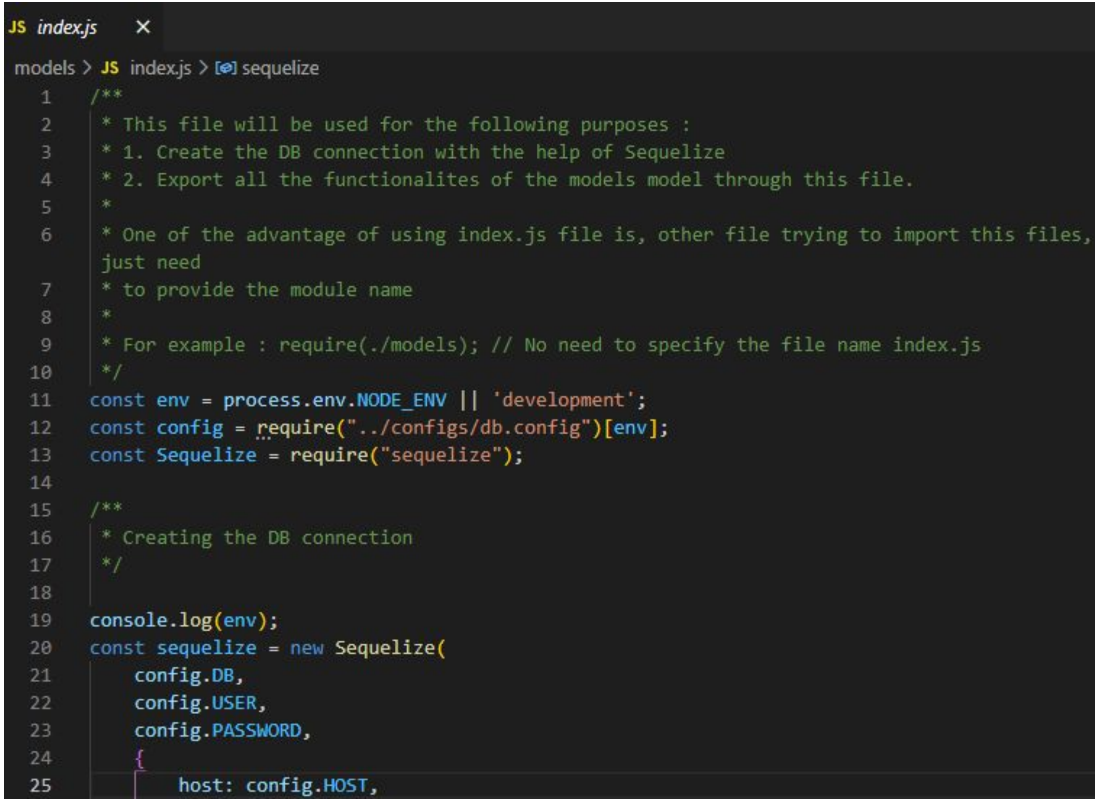
db.config.js



```
module.exports = {
    development: {
        HOST: "localhost",
        USER: "root",
        PASSWORD: "Mohit@19",
        DB: "ecom_db",
        dialect: "mysql",
        pool: {
            max: 5,
            min: 0,
            acuire: 30000, //max time in ms that a pool will try to get connection before throwing error
            idle: 10000  // maximum time in ms that a connection can be idle before being released
        }
    },
    production: {
        HOST: "sql6.freemysqlhosting.net",
        USER: "sql6478840",
        PASSWORD: "I3aSPgmBTw",
        DB: "sql6478840",
        dialect: "mysql",
        pool: {
            max: 5,
            min: 0,
            acuire: 30000, //max time in ms that a pool will try to get connection before throwing error
            idle: 10000  // maximum time in ms that a connection can be idle before being released
        }}
}
```

And,

Index.js in models folder

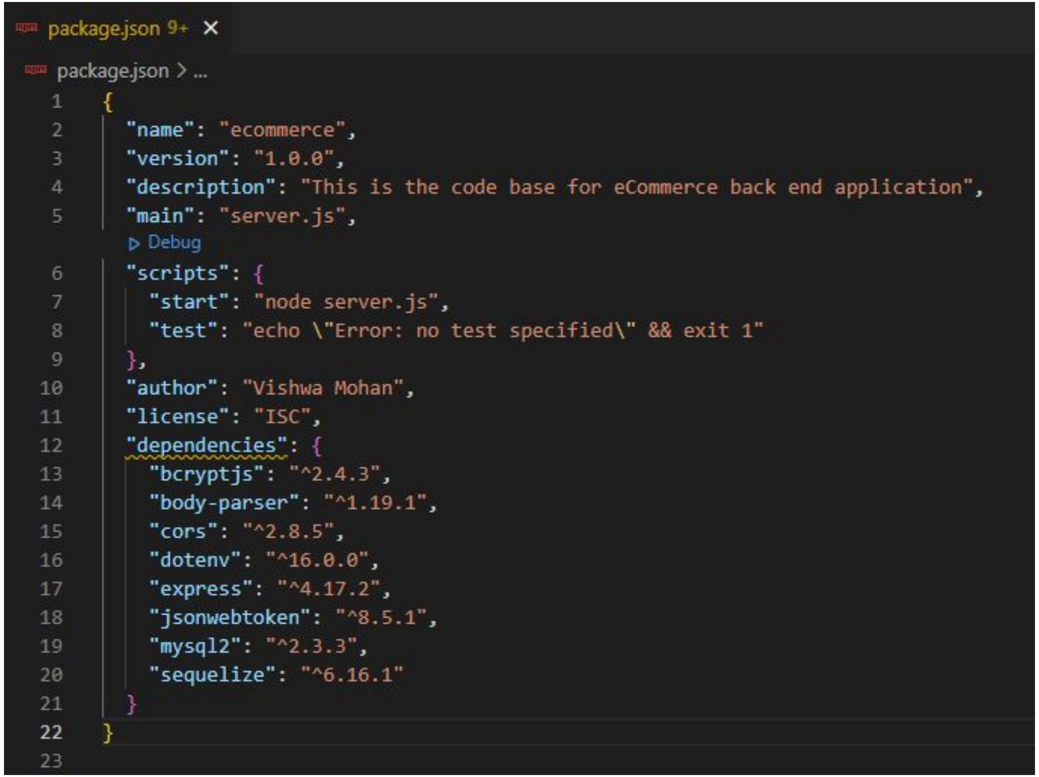


```
/**
 * This file will be used for the following purposes :
 * 1. Create the DB connection with the help of Sequelize
 * 2. Export all the functionalites of the models model through this file.
 *
 * One of the advantage of using index.js file is, other file trying to import this files, just need
 * to provide the module name
 *
 * For example : require(./models); // No need to specify the file name index.js
 */
const env = process.env.NODE_ENV || 'development';
const config = require("../configs/db.config")[env];
const Sequelize = require("sequelize");

/**
 * Creating the DB connection
 */

console.log(env);
const sequelize = new Sequelize(
    config.DB,
    config.USER,
    config.PASSWORD,
    {
        host: config.HOST,
```

And at last start script in package.json.



```json
{
  "name": "ecommerce",
  "version": "1.0.0",
  "description": "This is the code base for eCommerce back end application",
  "main": "server.js",
  "scripts": {
    "start": "node server.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Vishwa Mohan",
  "license": "ISC",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "body-parser": "^1.19.1",
    "cors": "^2.8.5",
    "dotenv": "^16.0.0",
    "express": "^4.17.2",
    "jsonwebtoken": "^8.5.1",
    "mysql2": "^2.3.3",
    "sequelize": "^6.16.1"
  }
}
```

Intro to Heroku

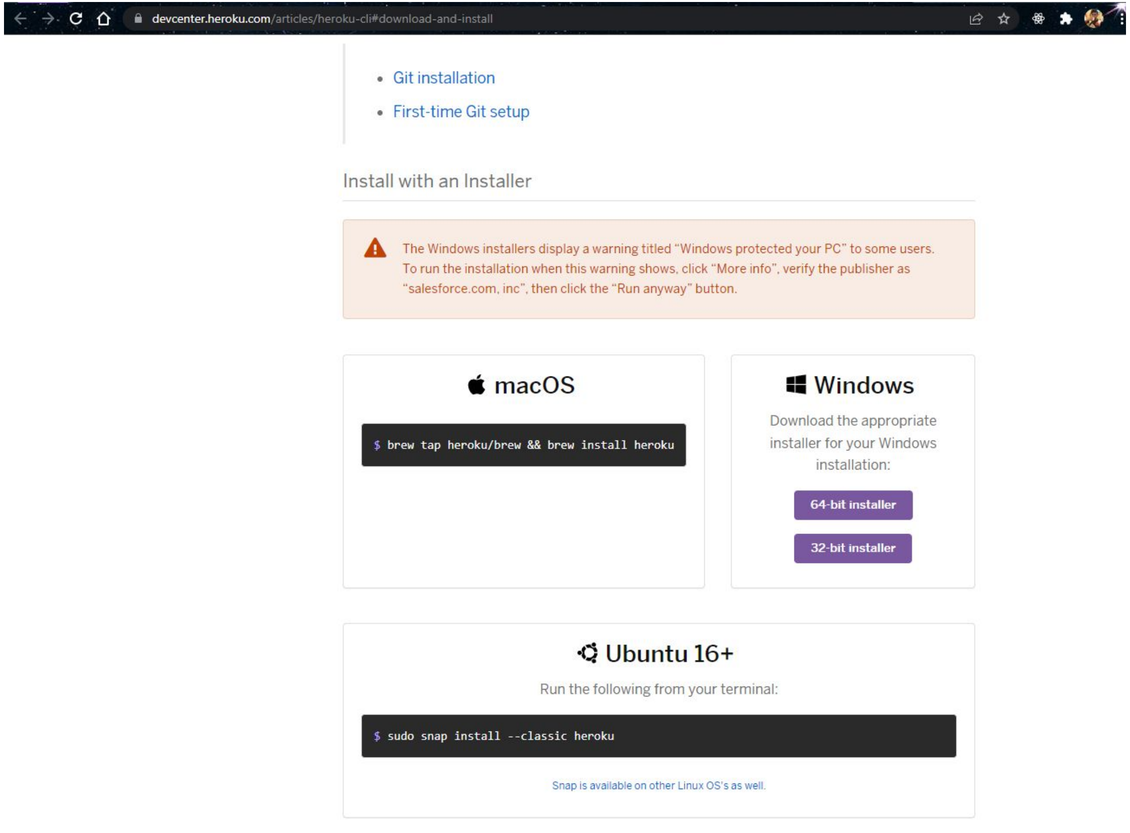
Heroku is a cloud platform that helps developers maintain, build, deliver, and scale apps.

Deploy App:

Let us use our application on Heroku, but before that, we have to do some steps:

1. Create an account on Heroku, (nothing much here just fill the prompted details and verify account for specified email id).

2. Download Heroku CLI: https://devcenter.heroku.com/articles/heroku-cli#download-and-install


devcenter.heroku.com/articles/heroku-cli#download-and-install

- Git installation
- First-time Git setup

Install with an Installer

The Windows installers display a warning titled “Windows protected your PC” to some users. To run the installation when this warning shows, click “More info”, verify the publisher as “salesforce.com, inc”, then click the “Run anyway” button.

macOS

```
$ brew tap heroku/brew && brew install heroku
```

Windows

Download the appropriate installer for your Windows installation:

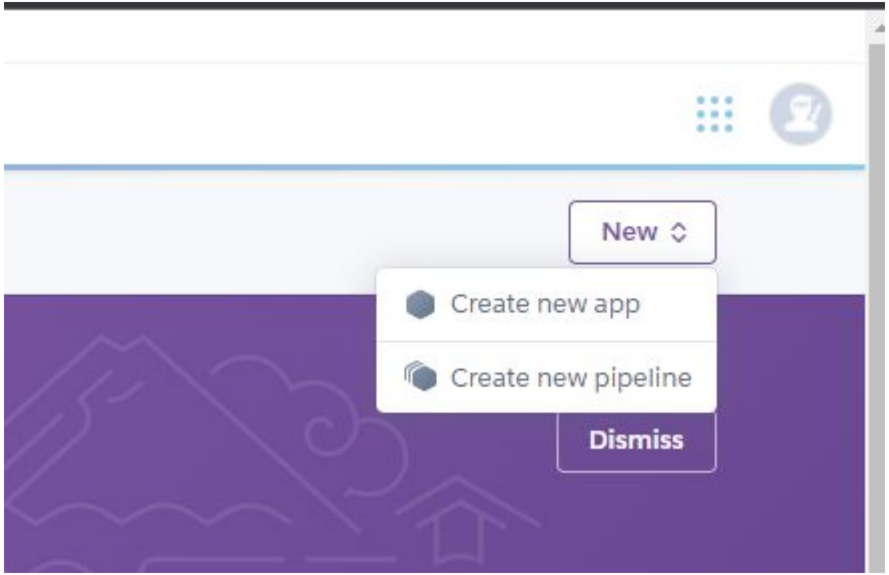
Ubuntu 16+

Run the following from your terminal:

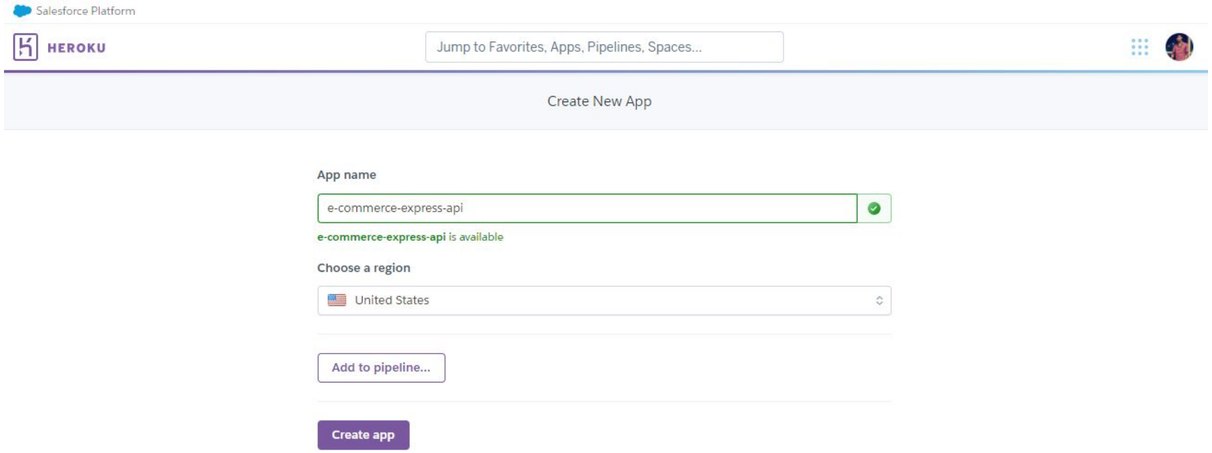
```
$ sudo snap install --classic heroku
```

Snap is available on other Linux OS's as well.

3. Go to the Heroku website https://dashboard.heroku.com/apps and log in. After login, click on “New.” and click “Create new app”.



4. The name must be unique, and click on Create app.



Create New App

App name

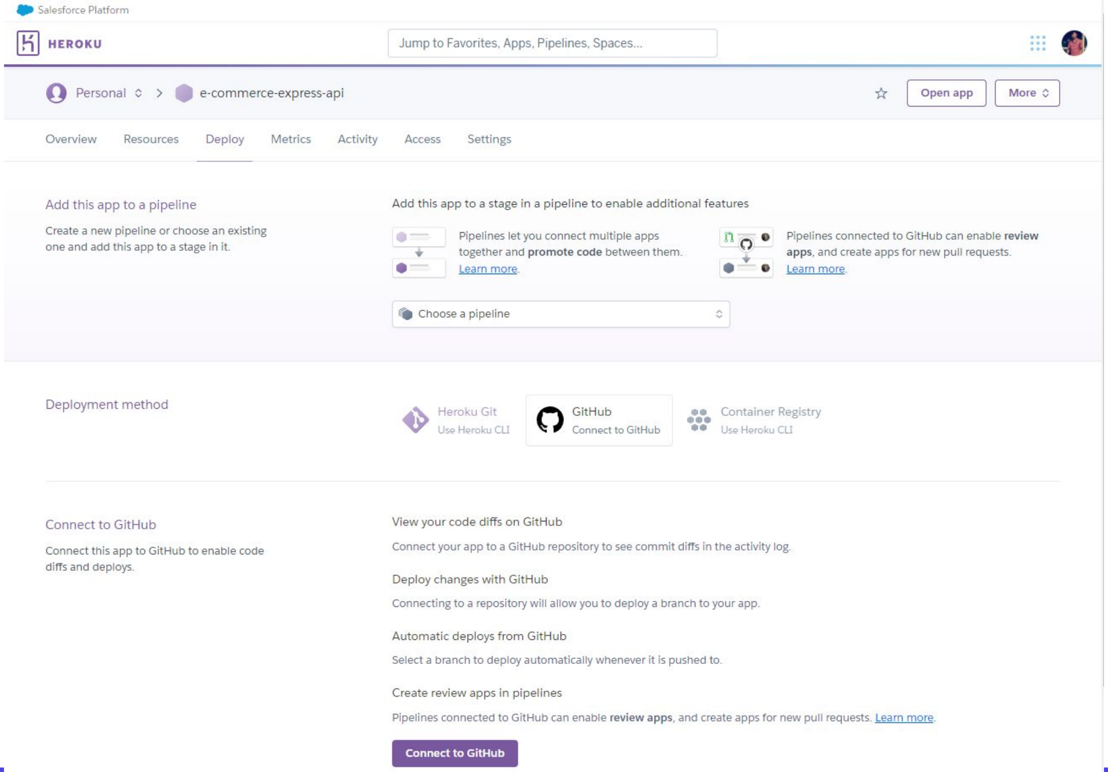
e-commerce-express-api

e-commerce-express-api is available

Choose a region

United States

5. Connect to GitHub now, it will prompt for authorization if Github is not already connected to Heroku



Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

Choose a pipeline

Deployment method

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

View your code diffs on GitHub

Connect your app to a GitHub repository to see commit diffs in the activity log.

Deploy changes with GitHub

Connecting to a repository will allow you to deploy a branch to your app.

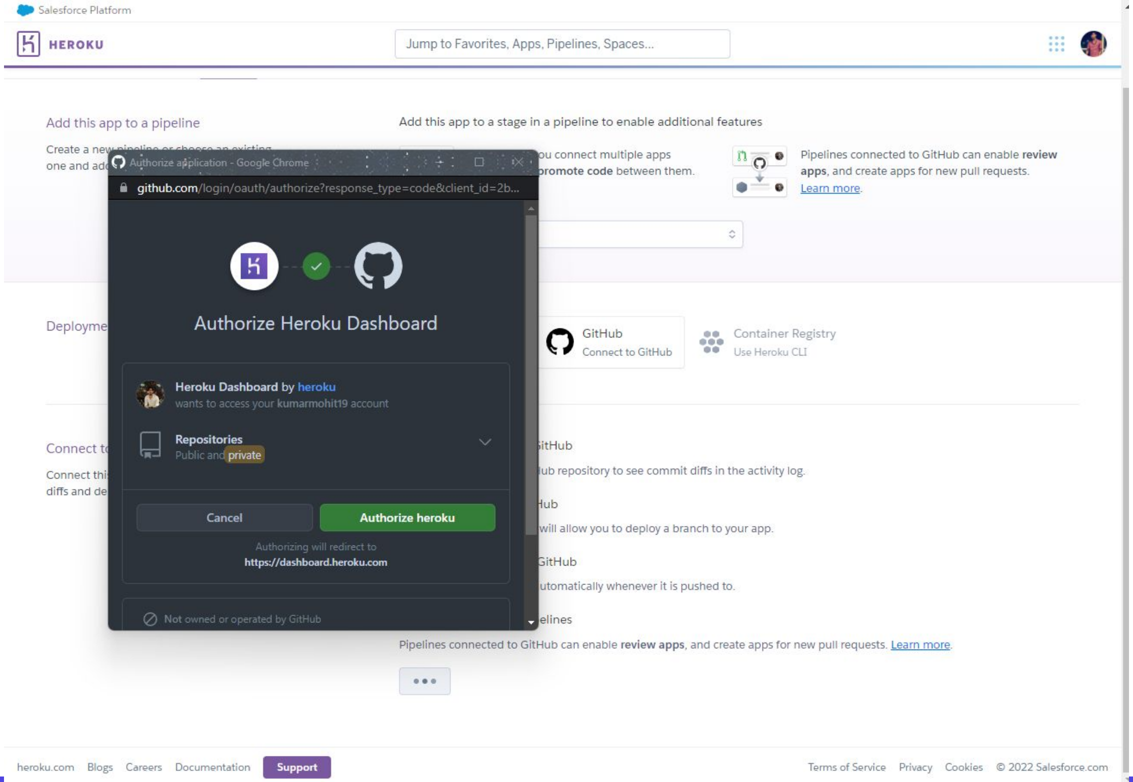
Automatic deploys from GitHub

Select a branch to deploy automatically whenever it is pushed to.

Create review apps in pipelines

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

6. Authorize Github



Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Add this app to a pipeline

Create a new pipeline or choose an existing one and add

Add this app to a stage in a pipeline to enable additional features

ou connect multiple apps promote code between them.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

Authorize application - Google Chrome

github.com/login/oauth/authorize?response_type=code&client_id=2b...

Authorize Heroku Dashboard

Heroku Dashboard by heroku
wants to access your **kumarmohit19** account

Repositories
Public and private

Cancel | Authorize heroku

Authorizing will redirect to
https://dashboard.heroku.com

Not owned or operated by GitHub

Deployme

GitHub
Connect to GitHub

Container Registry
Use Heroku CLI

Connect t

Connect thi diffs and de

itHub

ub repository to see commit diffs in the activity log.

Hub

will allow you to deploy a branch to your app.

GitHub

utomatically whenever it is pushed to.

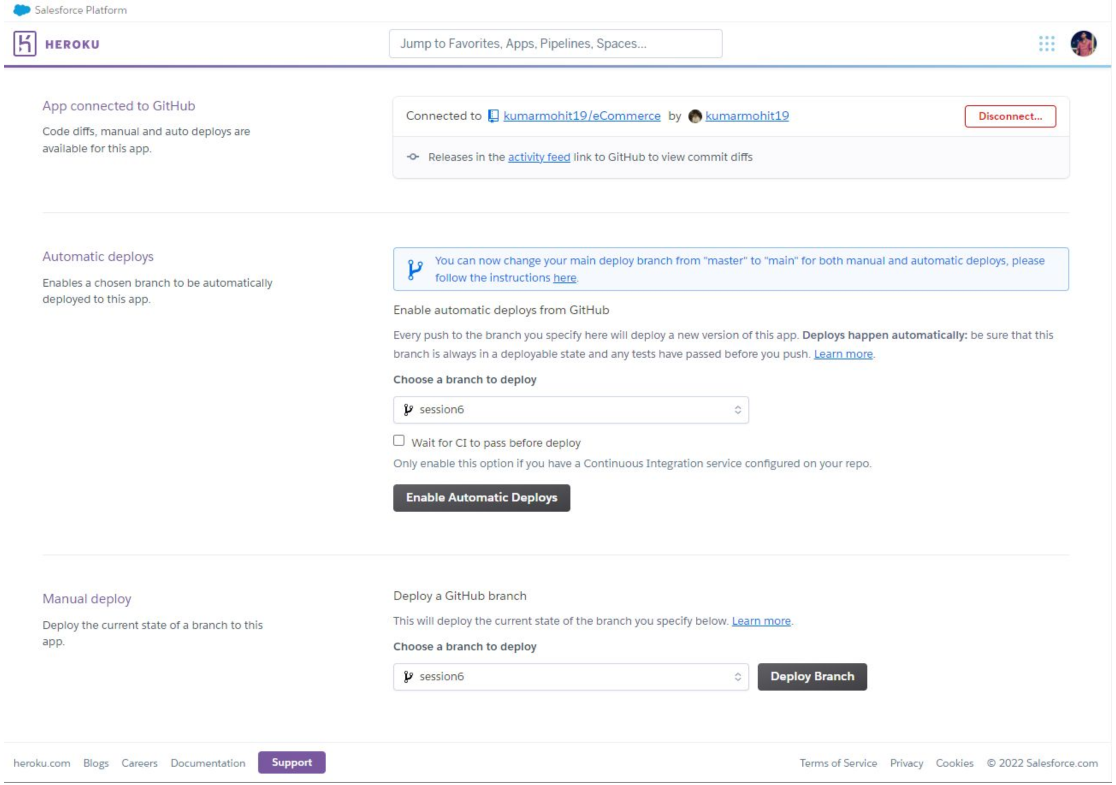
elines

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

heroku.com Blogs Careers Documentation Support

Terms of Service Privacy Cookies © 2022 Salesforce.com

7. Select the branch and deployment setting



Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to kumarmohit19/eCommerce by kumarmohit19

Disconnect...

Releases in the activity feed link to GitHub to view commit diffs

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. Learn more.

Choose a branch to deploy

session6

□ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

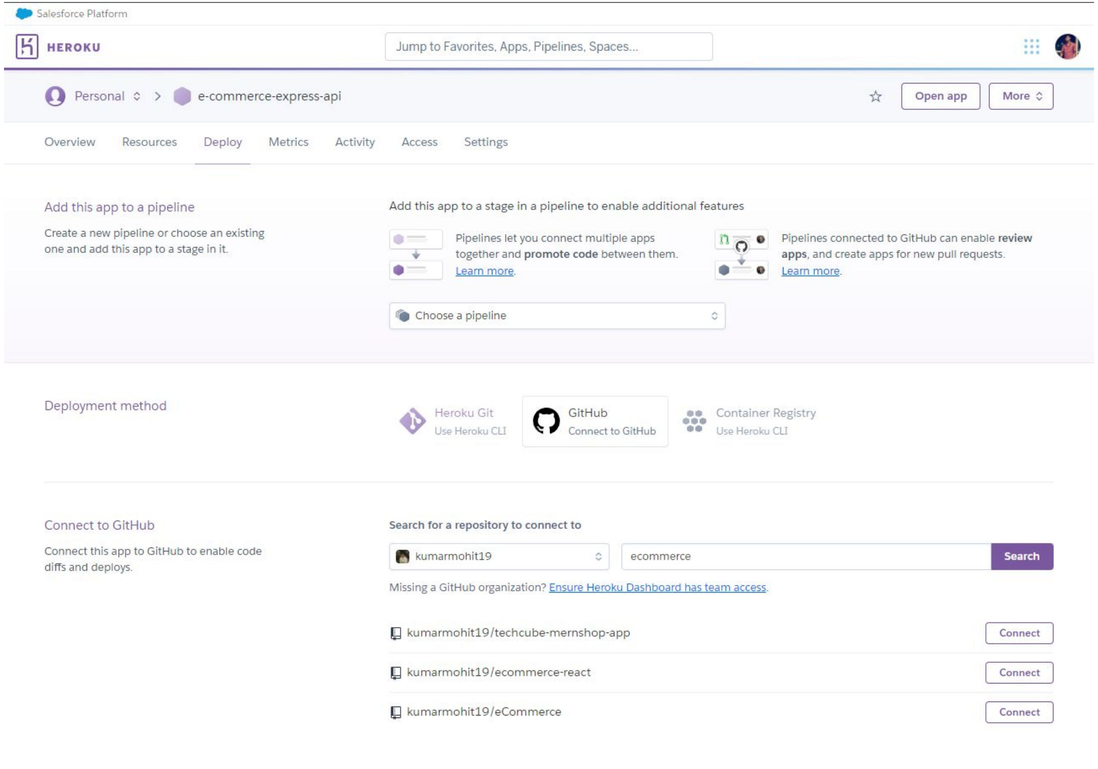
This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

session6

Deploy Branch

8. Search your repo



Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

Choose a pipeline

Deployment method

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

kumarmohit19 | ecommerce | Search

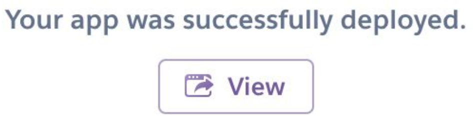
Missing a GitHub organization? Ensure Heroku Dashboard has team access.

kumarmohit19/techcube-mernshop-app Connect

kumarmohit19/ecommerce-react Connect

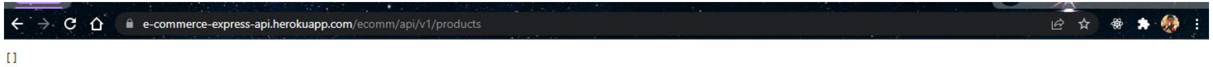
kumarmohit19/eCommerce Connect

9. Do manual deployment, then it will start the deployment process. Once done click on view:



Your app was successfully deployed.

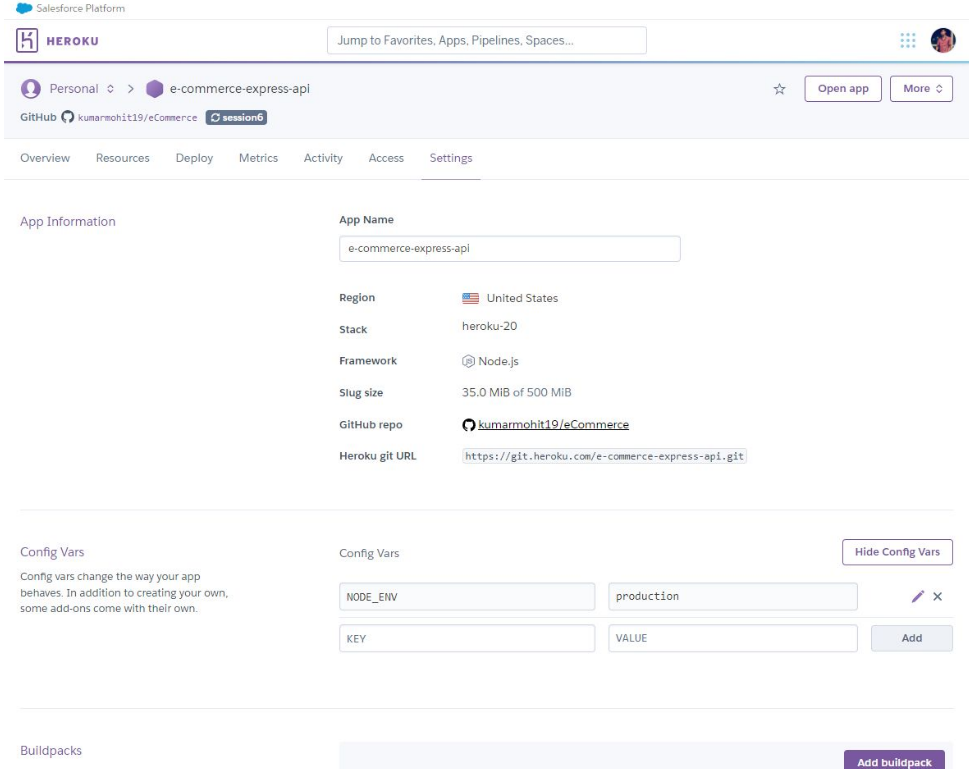
10. This is our backend page, notice that our page has a domain name now.



e-commerce-express-api.herokuapp.com/ecomm/api/v1/products

```
[]
```

11. Add NODE_ENV property in configuration:



App Information

App Name: e-commerce-express-api

Region: United States

Stack: heroku-20

Framework: Node.js

Slug size: 35.0 MiB of 500 MiB

GitHub repo: kumarmohit19/eCommerce

Heroku git URL: `https://git.heroku.com/e-commerce-express-api.git`

Config Vars

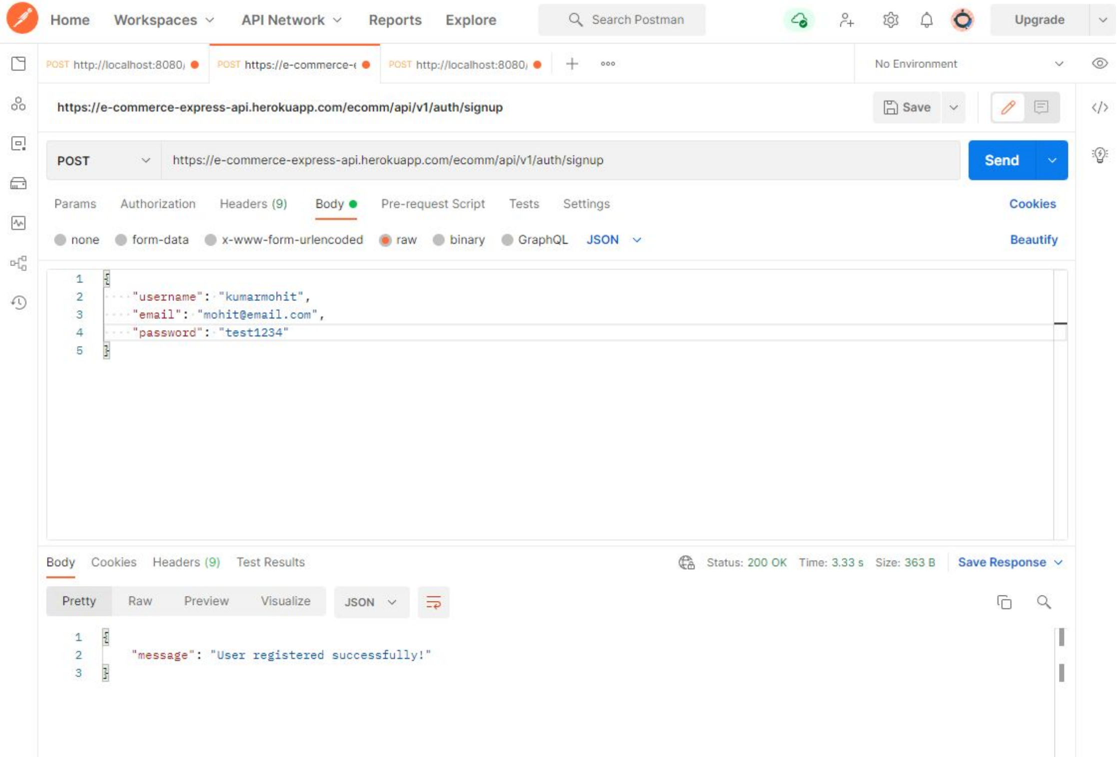
Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

KEY	VALUE
NODE_ENV	production

Testing our hosted App

SignUp



https://e-commerce-express-api.herokuapp.com/ecomm/api/v1/auth/signup

POST https://e-commerce-express-api.herokuapp.com/ecomm/api/v1/auth/signup

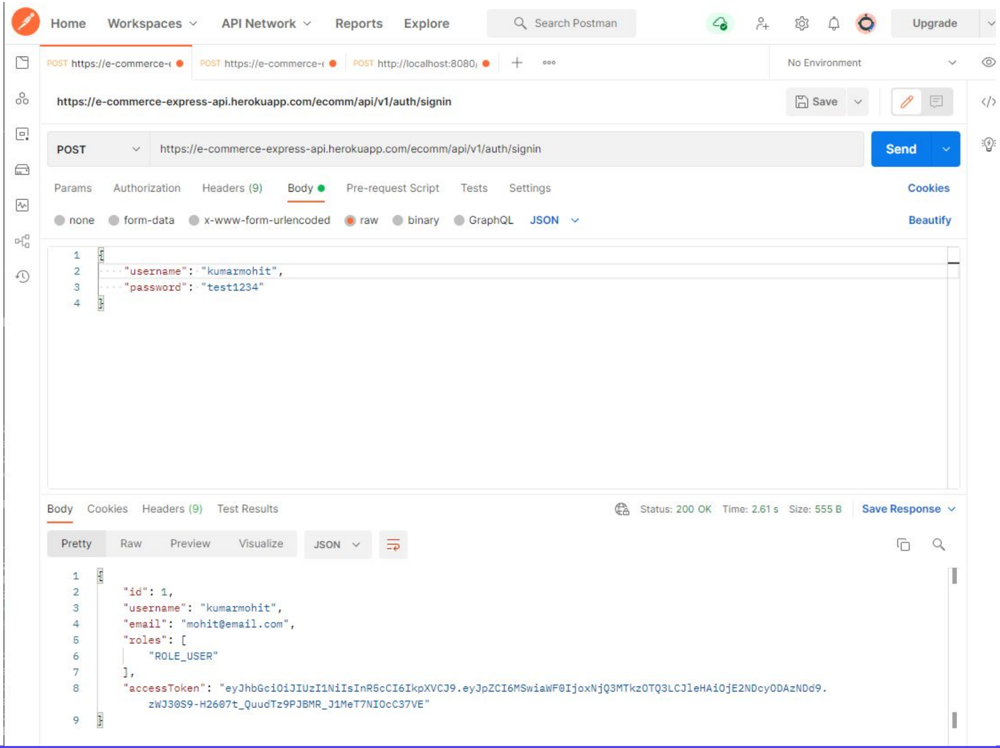
```
{
    "username": "kumarmohit",
    "email": "mohit@email.com",
    "password": "test1234"
}
```

Status: 200 OK Time: 3.33 s Size: 363 B

```
{
    "message": "User registered successfully!"
}
```

Testing our hosted App

SignIn



https://e-commerce-express-api.herokuapp.com/ecomm/api/v1/auth/signin

POST https://e-commerce-express-api.herokuapp.com/ecomm/api/v1/auth/signin

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "username": "kumarmohit",
    "password": "test1234"
}
```

Status: 200 OK Time: 2.61 s Size: 555 B

JSON

```
{
    "id": 1,
    "username": "kumarmohit",
    "email": "mohit@email.com",
    "roles": [
        "ROLE_USER"
    ],
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNjQ3MTkzOTQ3LCJleHAiOjE2NDcyODAzNDd9.
        zWJ30S9-H2607t_QuudTz9PJBMR_J1MeT7NIOcC37VE"
}
```

MCQs

1. What is the meaning of origin in git command?

a. Origin is the name of the branch.

b. It’s the name of the repo.

c. It signifies a remote name where the user wants to push or pull the changes.

d. None of the above.

2. What does a checkout argument do in git command?

a. Checkout means to go out of the repo.

b. It means the act of switching between different versions of a target entity or the branches of the repository.

c. It is used to commit the change and push in one command.

d. None of the above.

3. What Does a Heroku app URL look like with the app name “alpha”?

a. www.herokuapp.com/alpha
b. alpha.heroku.com
c. alpha.herokuapp.com
d. www.alpha.herokuapp.com

4. Is it possible to change the domain of the application deployed on Heroku.

a. No.
b. Yes

5. Which branch is used for deployment from GitHub to Heroku?

a. master

b. We can choose.

c. origin

d. None of the above.

Practice Problems

Deploy all the small or larger applications taught in all the sessions.

THANK YOU