

## Purpose

The purpose of this lab is to guide students through the process of deploying a full-stack application to a live environment using cloud hosting services. It covers creating cloud resources such as databases, configuring environment variables, and deploying both backend and frontend parts of the project, ensuring the application is accessible and functional over the internet.

## Learning Outcomes

By completing this lab, students will be able to:

* Apply for and utilize free DigitalOcean credits via the GitHub Student Developer Pack.
* Create and configure cloud databases and manage connection credentials securely.
* Modify project configuration files to support environment variables for flexible deployment.
* Deploy backend and frontend applications using DigitalOcean's App Platform connected to GitHub repositories.
* Manage environment variables and continuous deployment settings on a cloud platform.
* Verify and test the deployed application to ensure correct functionality and responsiveness.

## Introduction

Among the many [offers](https://education.github.com/pack/join) provided by the GitHub Student Developer Pack, free credits are available for your DigitalOcean account. This lab leverages these credits to deploy your project to the cloud, enabling you to create databases and application hosting environments without incurring costs. You will learn how to connect your GitHub repository to DigitalOcean's App Platform for automated deployments, and configure your application to use remote database connections securely through environment variables.

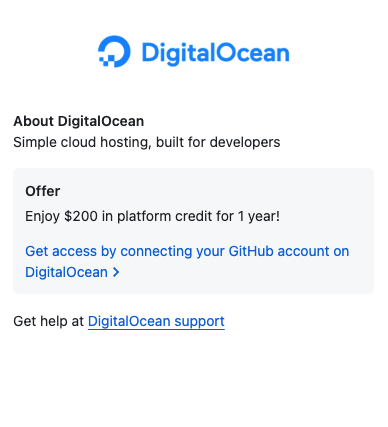
## Pre-Lab Requirements

Before starting this lab, you should have:

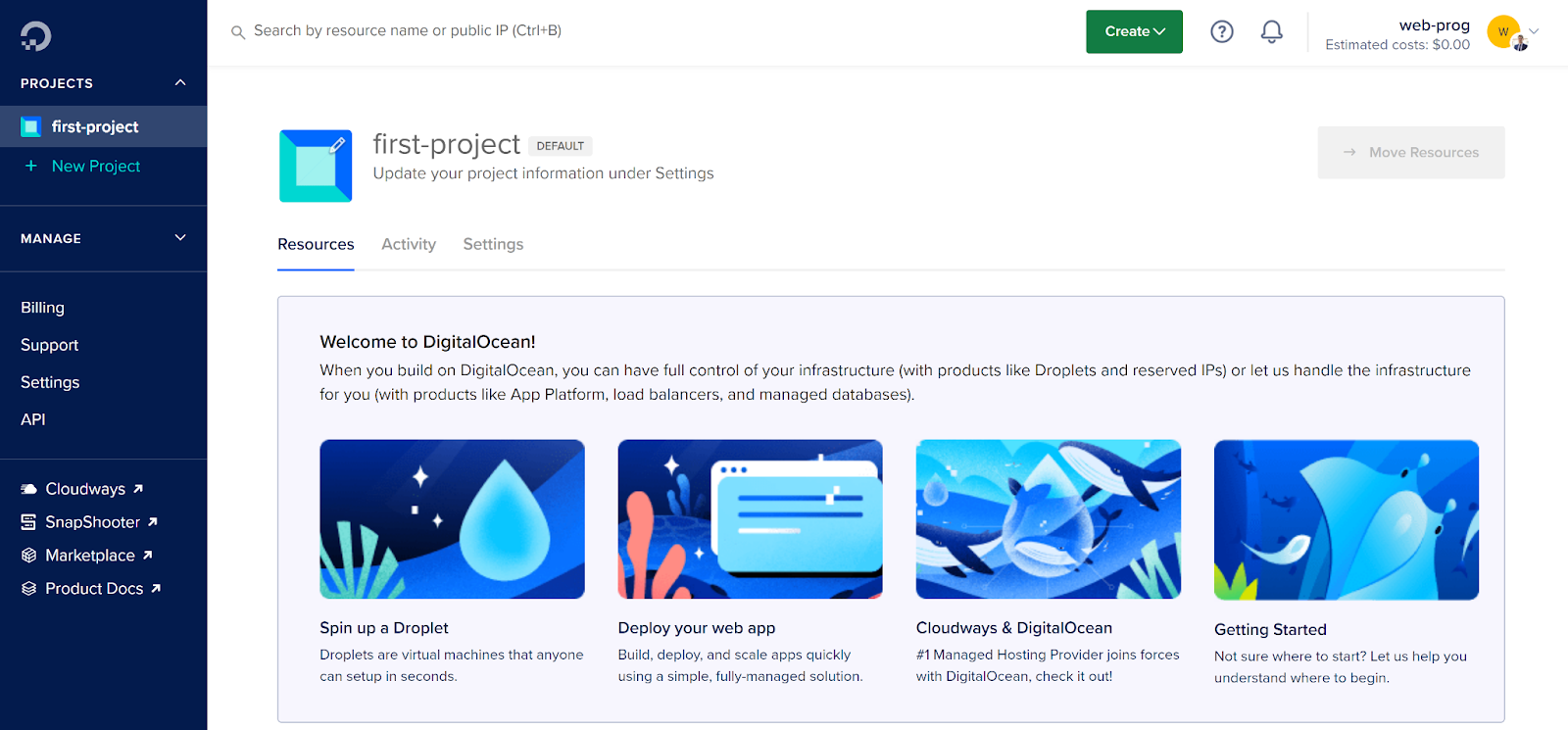
* Registered for the GitHub Student Developer Pack and claimed the DigitalOcean free credits.
* A completed project code pushed to a repository.
* Basic familiarity with MySQL database tools and exporting/importing databases.
* Understanding of environment variables and configuration in your project files.
* Access to the DigitalOcean control panel and basic knowledge of navigating cloud dashboards.

## Step-by-Step Instructions

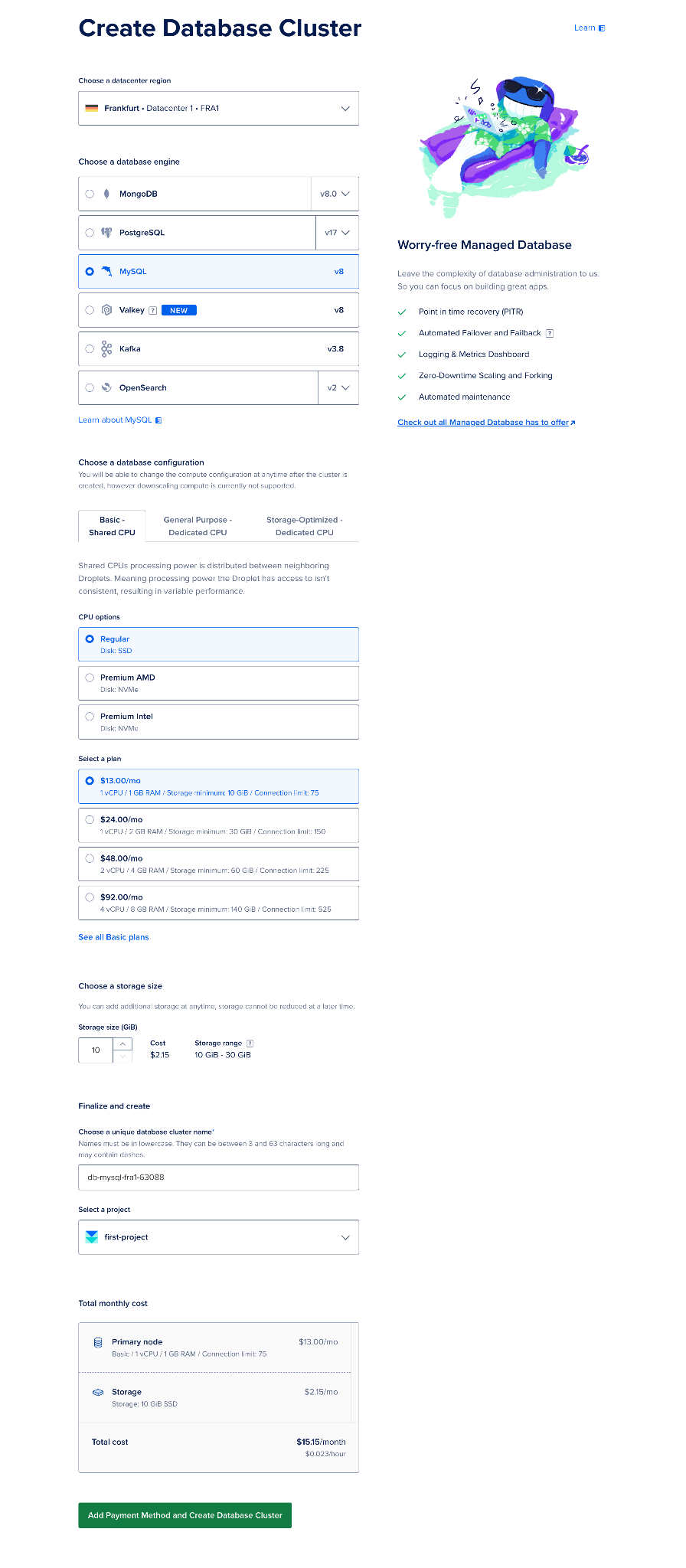
1. Once you applied for the github student pack, under the all offers section you can apply for DigitalOcean free credits. You can follow the link “Get access by connecting your GitHub account on DigitalOcean” to claim this offer and properly register your free credits.



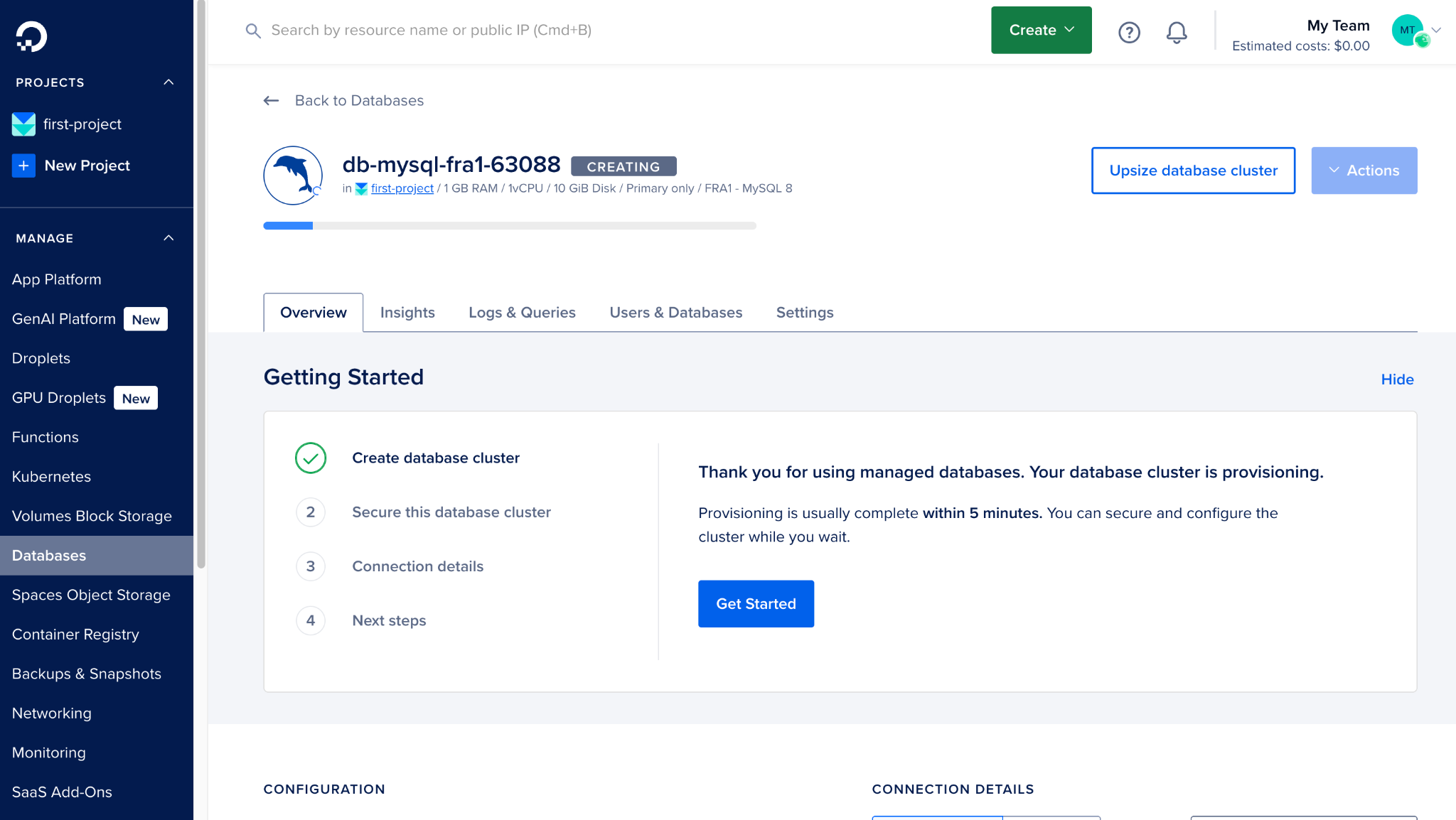
1. Create a profile via github and after you have your DigitalOcean account, and it’s connected to Github, and you have a team created, you can start working on [projects](https://cloud.digitalocean.com/login?redirect_url=https%3A%2F%2Fcloud.digitalocean.com%2Fprojects). By default, a project named “first-project” should be created for you.



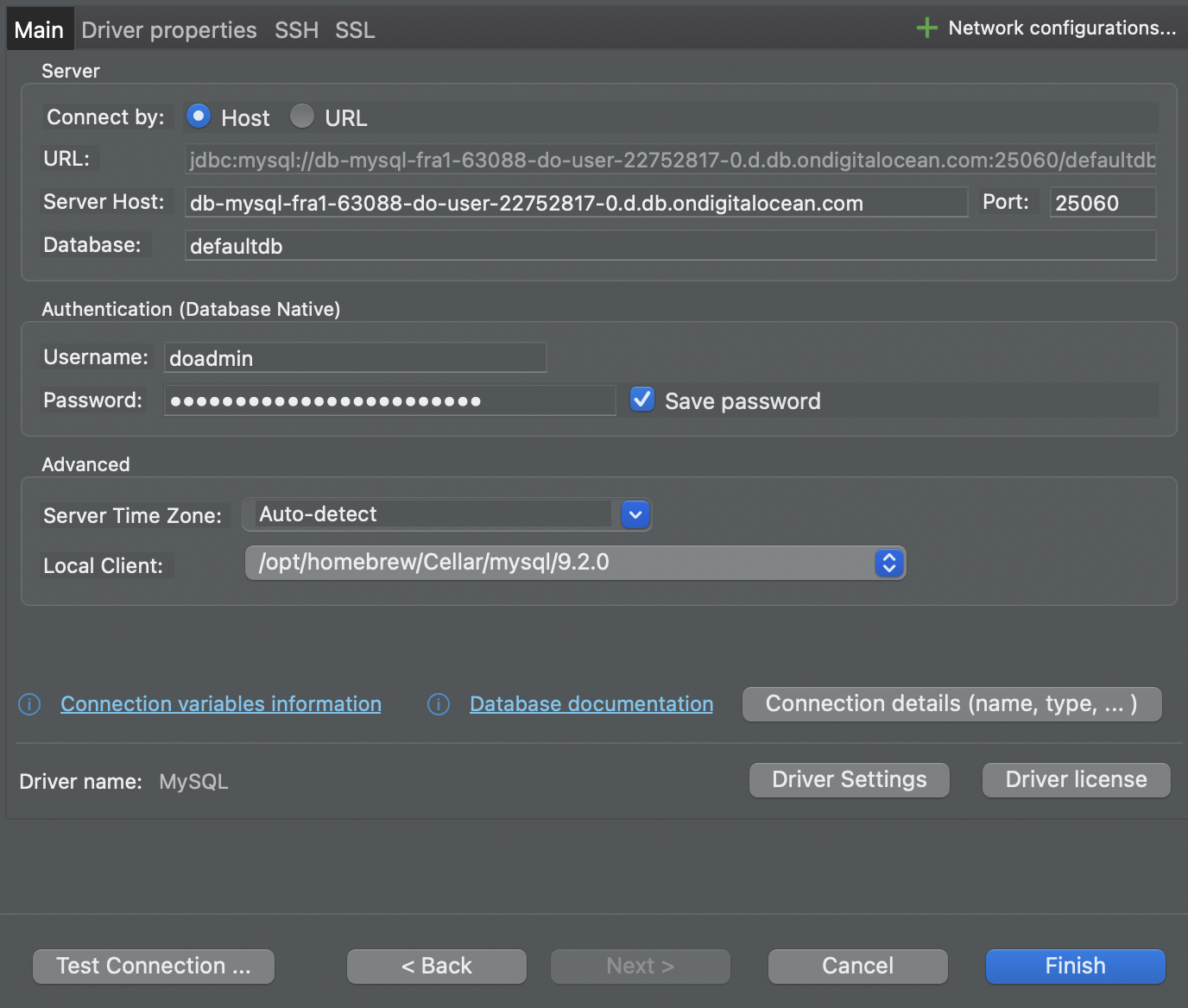
1. Create Database  
   Click on the green “Create” button and choose “Database”, then follow the example below to create your first database cluster on DigitalOcean. The creation process may take a while, and afterwards you should have your database credentials that you can use for connections.



Once the database is created you will see this screen. Under Connection details you will see your credentials for deployed database



1. Using any MySQL tool, you can connect to your database by using the provided credentials.



NOTE: Once you click on SSL and enable SSL don’t forget to upload your CA certificate available her on connection details



1. When you create a connection in your environment you need to add your database to this server. To do this you need to export your database from a localhost connection and import it to a deployed one.
2. Modify your config.php file   
   We want to use environment variables in our projects now, so we will modify the Config.php file and add a function that checks if a variable is available as an environment variable (by using the \_ENV superglobal) and if it’s not then a default value will be used.

<?php

class Config {

public static function DB\_NAME() {

return Config::get\_env("DB\_NAME", "university");

}

public static function DB\_PORT() {

return Config::get\_env("DB\_PORT", 3306);

}

public static function DB\_USER() {

return Config::get\_env("DB\_USER", 'root');

}

public static function DB\_PASSWORD() {

return Config::get\_env("DB\_PASSWORD", '');

}

public static function DB\_HOST() {

return Config::get\_env("DB\_HOST", '127.0.0.1');

}

public static function JWT\_SECRET() {

return Config::get\_env("JWT\_SECRET", ',dpPL,Se%fM-UVQBwf/X0T&B!DF6%}');

}

public static function get\_env($name, $default){

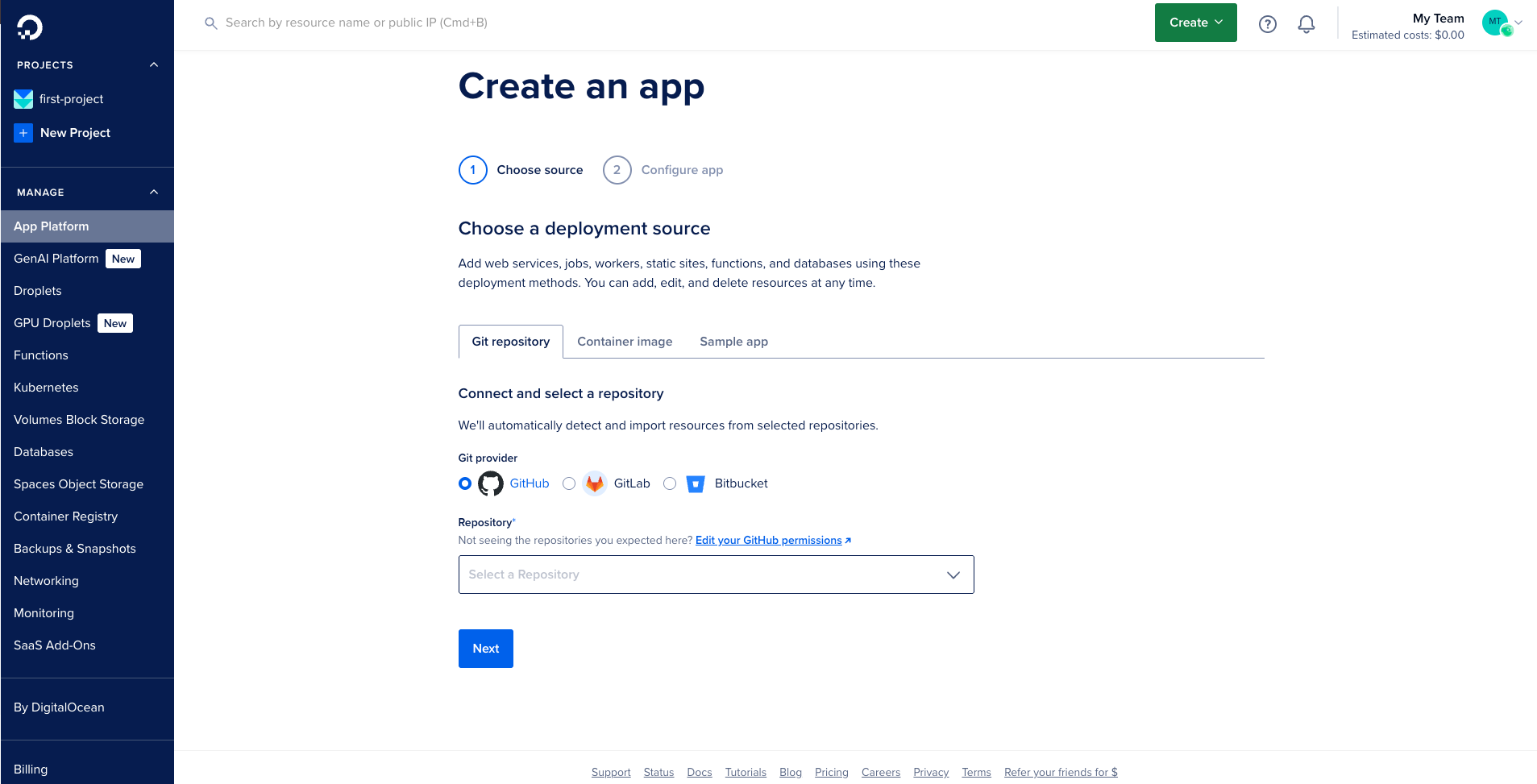
return isset($\_ENV[$name]) && trim($\_ENV[$name]) != "" ? $\_ENV[$name] : $default;

}

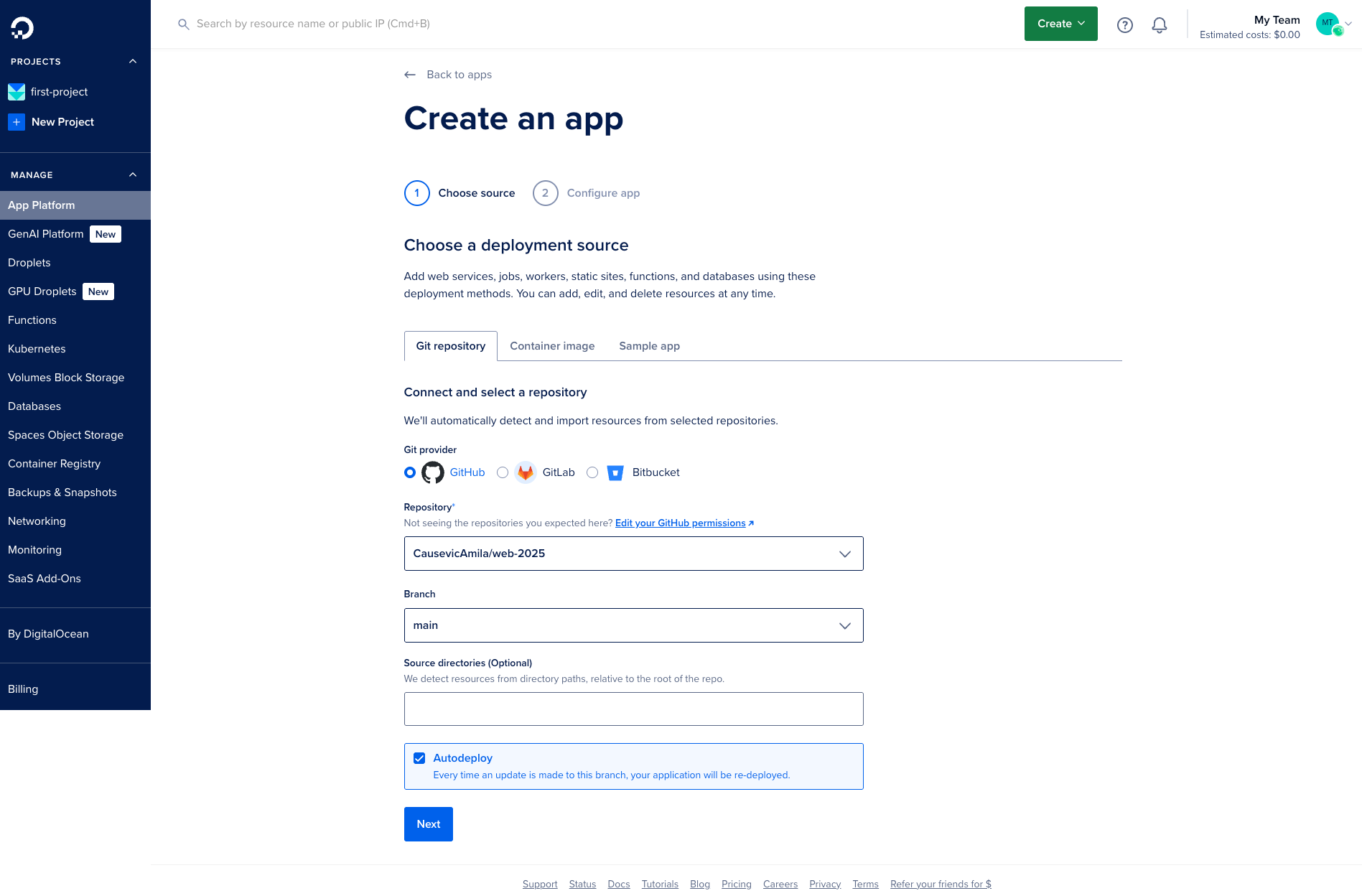
}

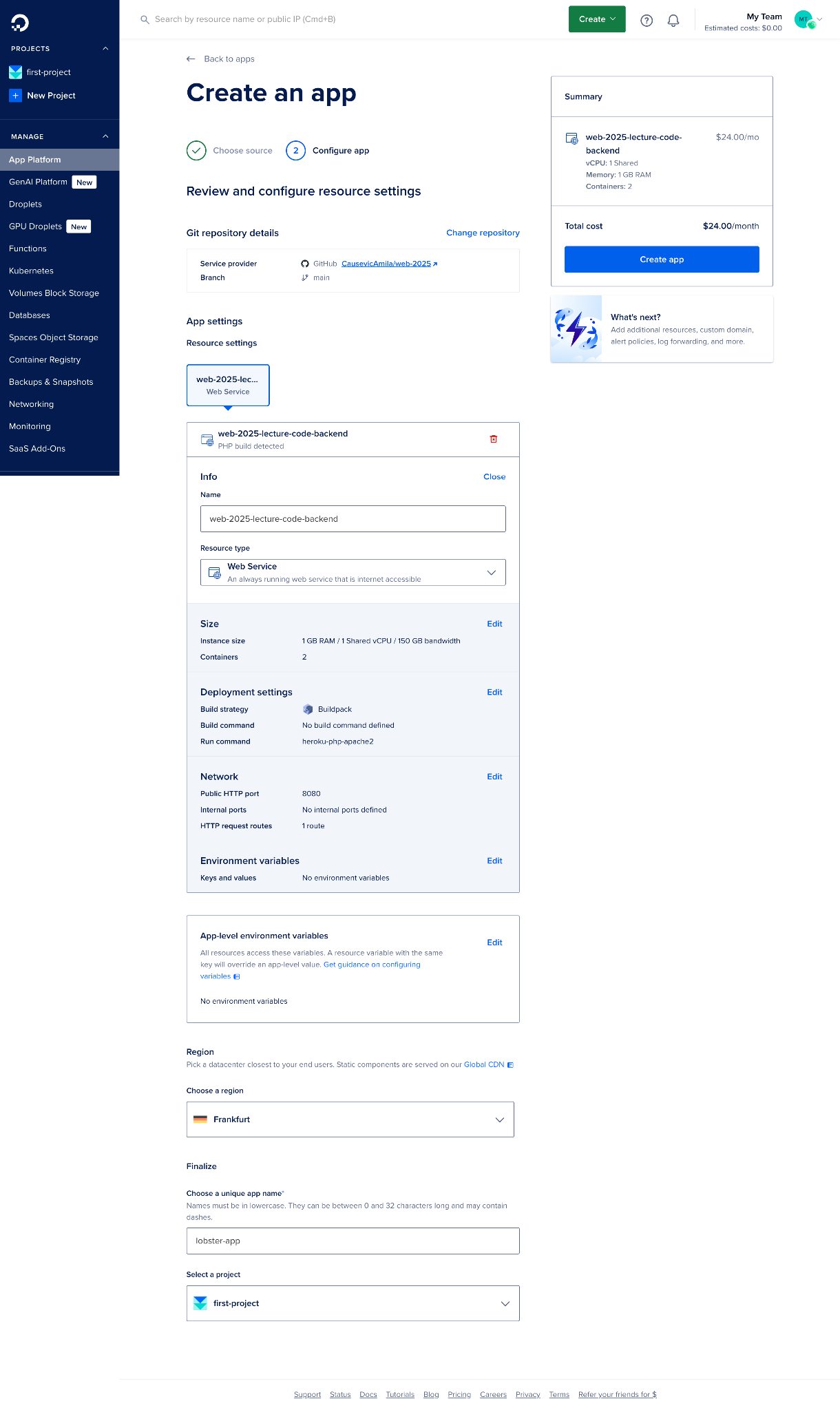
This essentially means that when you deploy your application you can pass remote database connection credentials as environment variables, and they will be used in the production version of the application, while during local development the default values will be used. Make sure that you use the same names for environment variables as the names passed in the functions.

1. Deploy your Application  
   On the DigitalOcean control panel, use the green “Create” button and click “App Platform”. The first step requires choosing where to retrieve the application source code from - we will be using GitHub. You will need to authorize DigitalOcean to access your GitHub repository.

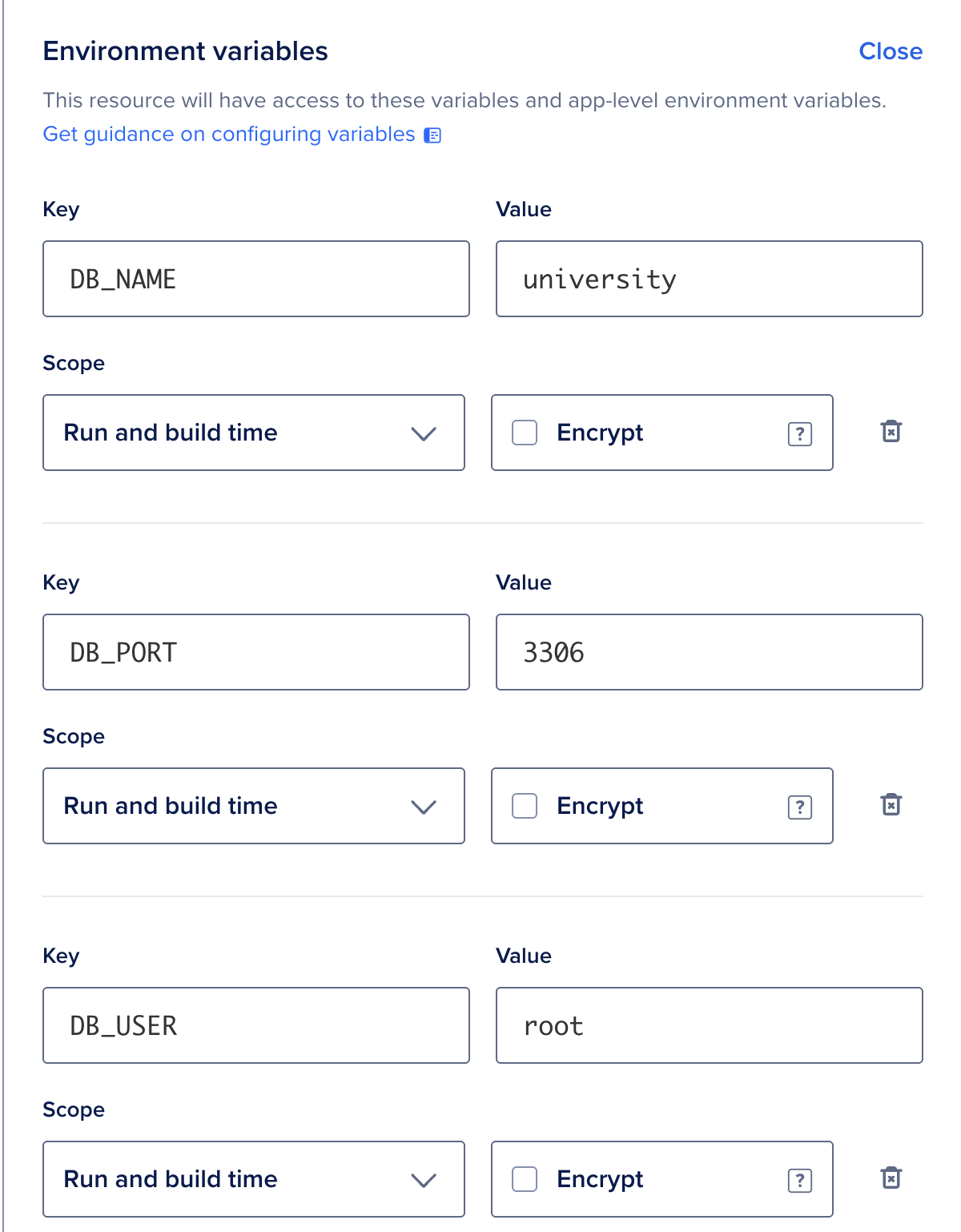
After you have authorized GitHub, you can choose your repository from the list, and configure additional options. In this case, the branch that will be deployed will be the main branch and it will have Autodeploy enabled (continuous delivery).

Note: in the source directions section you will need to provide a path to your backend folder (project/backend) for backend deployment, and for the fronted part you will need to provide a path for the frontend code (project/frontend). After that click on next button.

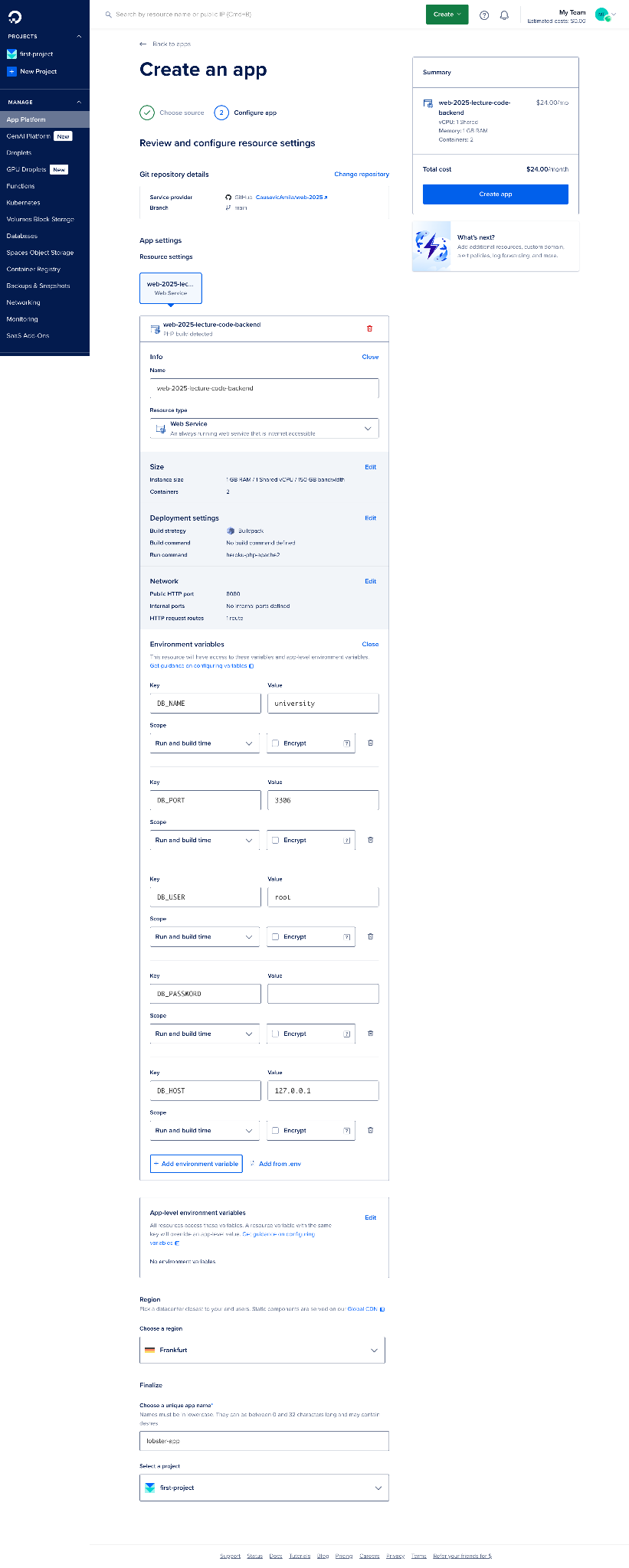


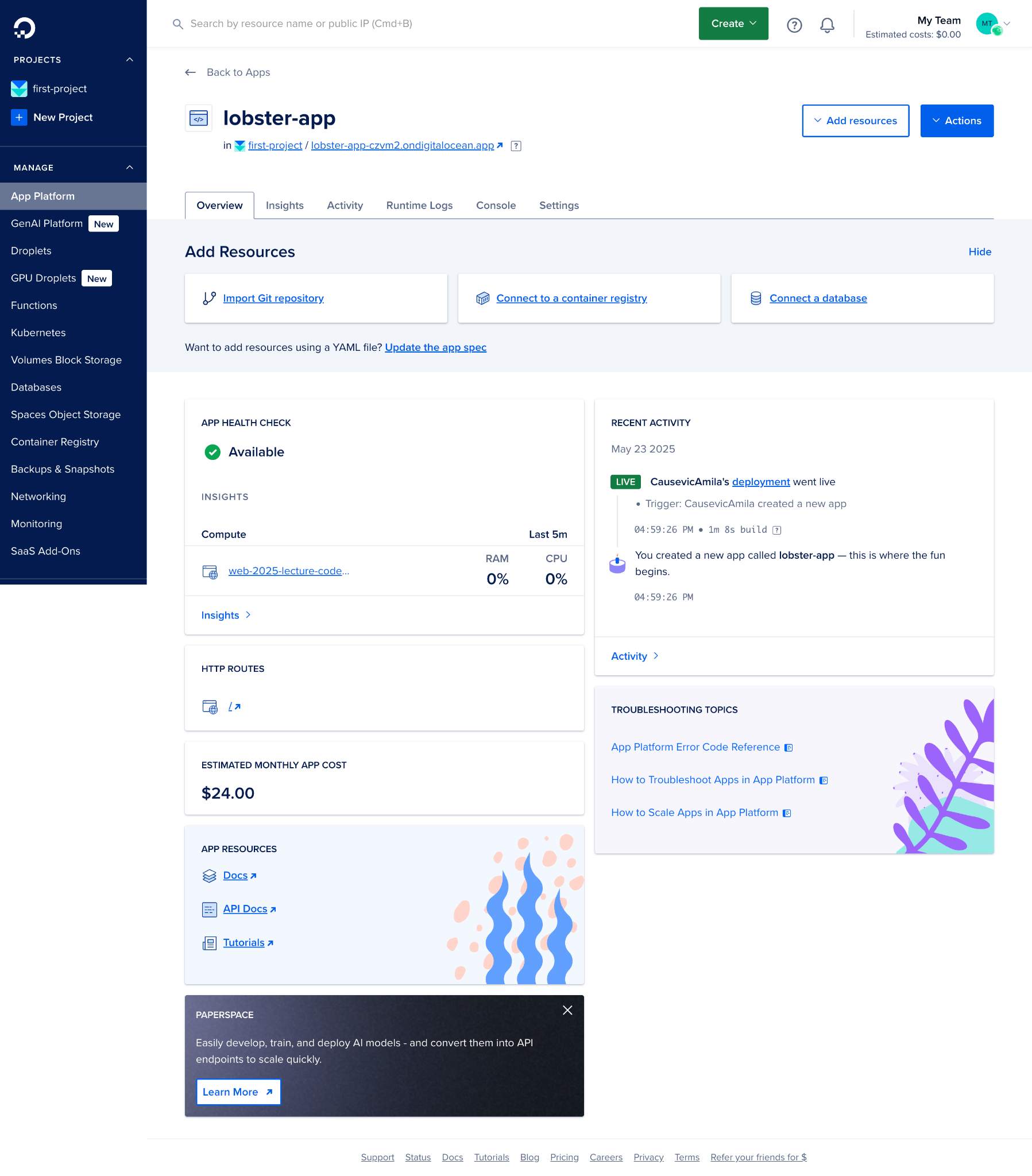


1. Don’t forget to add env variables. You can add here data you got from digital ocean once db is deployed if this is not working!



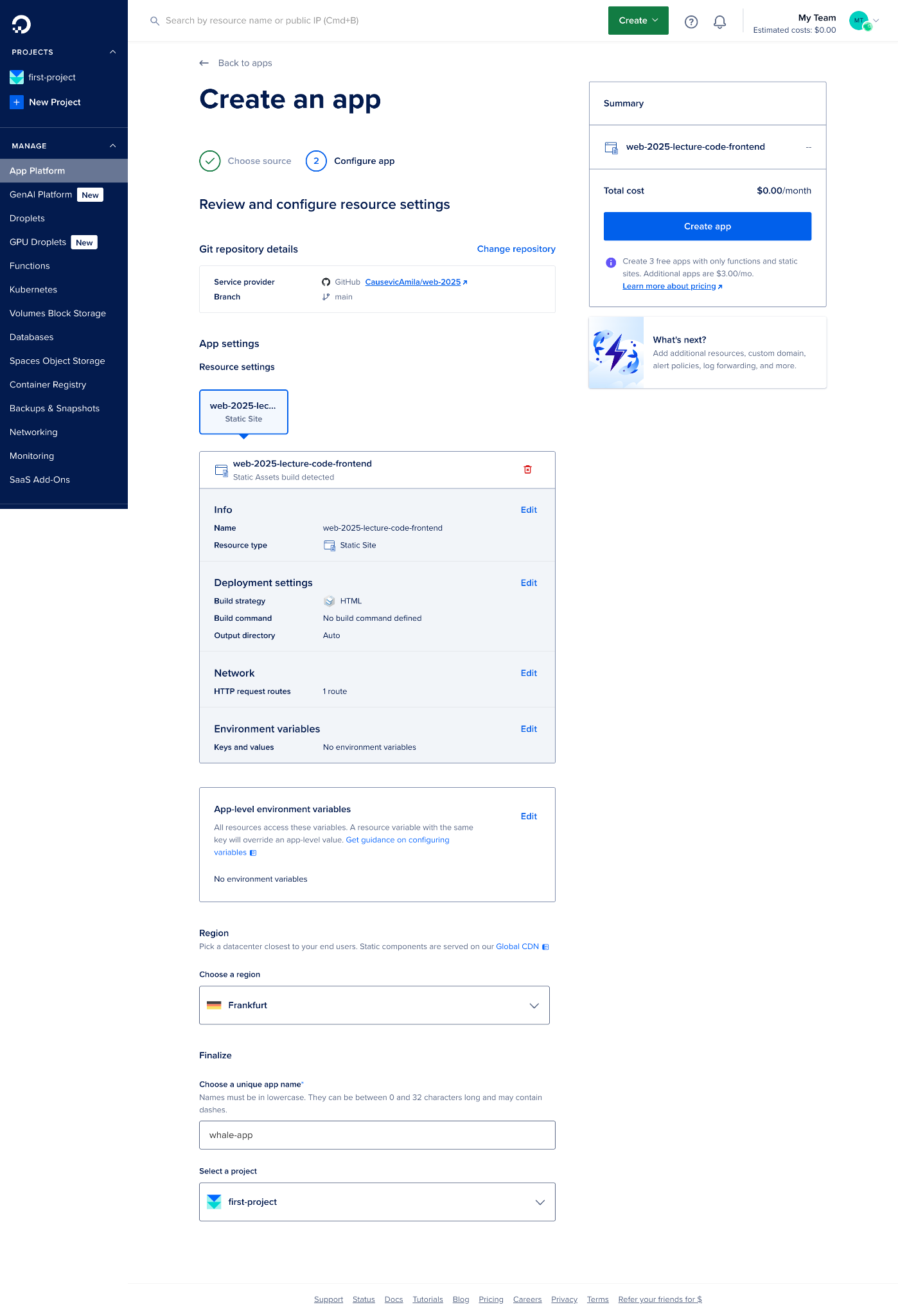
1. Once everything is done review all fields and then click on create app button, and wait few moment until backend is deployed





This is how it should look after everything is done.

1. Repeat the same process for frontend folder



Last year tutorial: <https://drive.google.com/drive/u/0/folders/1NsYZxwywhbrUKjuJtuLAmPowlNsvLBk6>

<https://docs.google.com/document/d/181uzoVUN1cit5iOHb0rGQlTrwl-dtXkABrvNRCqK548/edit?tab=t.0>

## Tasks

1. Your task is to deploy the complete project to a live environment so that it is accessible over the internet. This includes setting up the necessary infrastructure, configuring the environment, and making sure both the backend and frontend are properly connected and running smoothly. You should choose an appropriate hosting platform (such as DigitalOcean, Heroku, AWS, or similar) and configure all environment variables securely. Make sure to push your code to a version control system like GitHub and use it as the source for deployment. After deployment, test your application to verify that all features work correctly and that the application is stable and responsive.