

[Open in app ↗](#)

★ Get unlimited access to all of Medium for less than \$1/week. [Become a member](#) X

A quick tutorial on how to deploy your Streamlit app to Heroku.

The ultimate guide on how to host your Streamlit app on Heroku.



Navid Mashinchi · Follow

Published in Towards Data Science

7 min read · Mar 17, 2021

[Listen](#)[Share](#)[More](#)

Photo by [Kevin Ku](#) on [Unsplash](#)

Welcome to the step-by-step tutorial on how to deploy your Streamlit app to Heroku. For the past two weeks, I have been working with Streamlit, and I was very impressed with its capabilities. It's ideal for sharing your data science project with the world since it's very straightforward. In this tutorial, I assume you have already completed your app and you are at the final stage of deploying your project. Based on my experiences from the past two weeks, you would think you can immediately deploy your project using Streamlit Sharing. However, you need to apply for the functionality, and it could potentially take a couple of days for them to send you the invite. Worst come worse, I even heard of stories where people waited a week to get the invitation to take advantage of Streamlit Sharing. If waiting isn't an option for you, you can always deploy your app to Heroku or some other host. In this guide, I will go only over how to deploy your app to Heroku and note that the steps below won't work if you would like to host it on AWS or somewhere else.

Step 1: Run your Streamlit App Locally

The first step is to make sure you have a python file. If you are working in Jupyter Notebook, make sure to download your file as a .py file. This is necessary if you want to run your code locally with Streamlit and push it later to Heroku. I have created a sample Streamlit app so we can follow along here. The name of my python script is app.py. To run your code locally, go to the directory where your script is saved using the command-line interface (CLI). Once you are in the directory, run the following command.

```
streamlit run app.py
```

A window will open automatically in your browser, as you can see below.



Deploy me to Heroku

Welcome to my app.

Rerun	R
Clear cache	C
Deploy this app	
Record a screencast	
Documentation	
Ask a question	
Report a bug	
Streamlit for Teams	
Settings	
About	

Made with Streamlit

Image by author

Voila, you have your Streamlit app running on your local machine.

Step 2: Create a GitHub repository

The next step is to create a GitHub repo for your app. Log in into your GitHub account and follow the steps below:

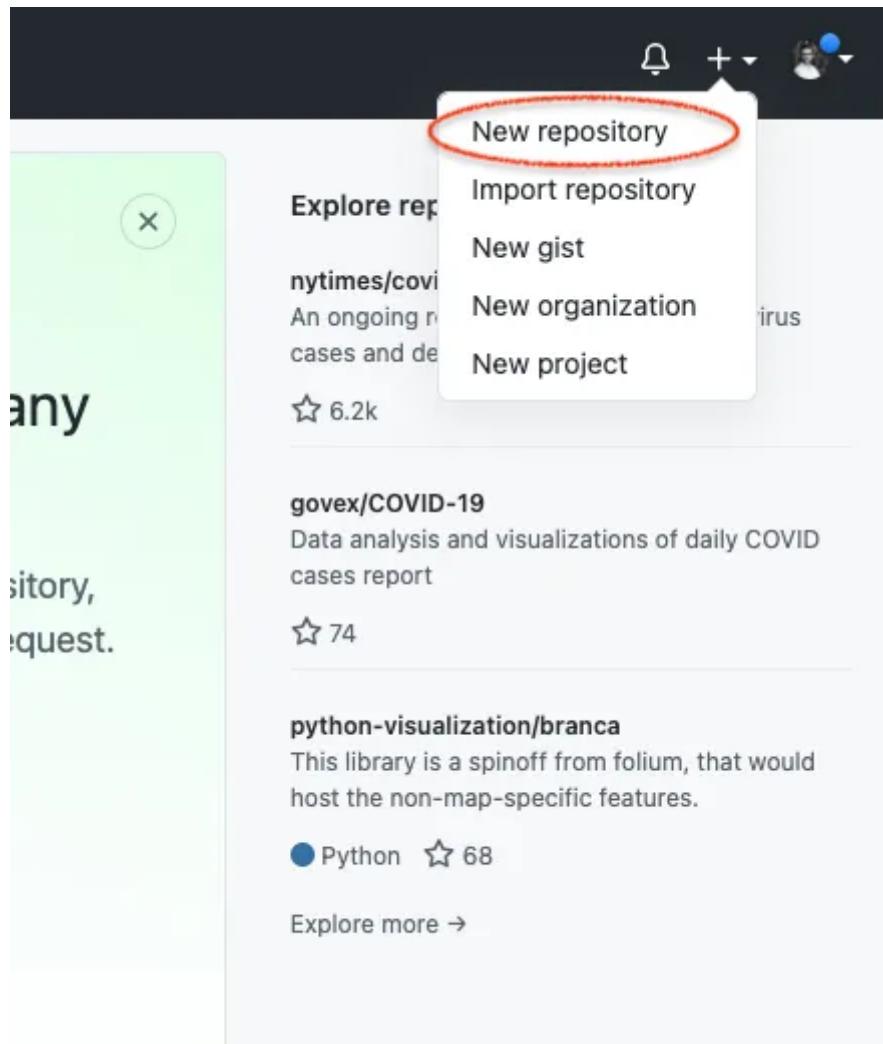


Image by author

Type in your repository name and click “Create repository.”

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Repository template

Start your repository with a template repository's contents.

[No template ▾](#)

Owner *

 navido89 ▾

Repository name *

/ streamlit-to-heroku-tutorial ✓

Great repository names are short and memorable. Need inspiration? How about [furry-tribble](#)?

Description (optional)

A tutorial on how to deploy your Streamlit app to Heroku

 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](#).

[Create repository](#)

Image by author

Now that you have created the repo, it's time to make your first commit and push to the repo. In your terminal, move to the directory where your app.py is in and copy-paste the following code.

```
echo "# streamlit-to-heroku-tutorial" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin
```

```
https://github.com/your_username/your_repo_name.git
git push -u origin master
```

I am initializing my directory in the above code, adding a README.md file, a remote origin with the URL you see above, and conducting my first push.

Step 3: Create a requirements.txt, setup.sh, and Procfile.

The next step is crucial, and you want to make sure you don't have any typos or make any mistakes. Otherwise, Heroku will run an error when you deploy your app. Before you move on, make sure you have the app.py and the README.md file in your directory.

requirements.txt: Create a requirements.txt file (**No capital letters!**) and add all the libraries to that file you are using in your python script. In my app.py example, I just imported the following library:

```
import streamlit as st
```

As a result, I add “*streamlit*” to my requirements.txt file. This file is necessary for Heroku to know which libraries it needs to run your application.

setup.sh: The next file is the setup.sh file. This is a shell file, and you need to add the following shell command inside the file. Copy-paste the code below into the file:

```
mkdir -p ~/.streamlit/
echo "\
[server]\n\
headless = true\n\
port = $PORT\n\
enableCORS = false\n\
\n\
" > ~/.streamlit/config.toml
```

If you don't add this file, you will get an error message when you deploy your app to Heroku. So make sure you create this app and name it as I wrote it above. **No capital letters!**

Procfile: Inside your directory, create a text file and call it “Procfile.” Inside the file, copy-paste the following code:

```
web: sh setup.sh && streamlit run app.py
```

The “web” means that it’s a web app. The Procfile pretty much specifies the commands once you run the app on Heroku. We specify the shell file that we created above and then call Streamlit to run app.py.

Once you have created the files above, review the steps and make sure you named the files as I named them above. It’s without saying that if you have more libraries for your app that you need to list them in your requirements.txt file. Any library that you are importing needs to be in that file.

Step 3: Connect to Heroku

Once you have created all the required files, it’s now time to set up your app so it can interact with Heroku.

The first thing you will do is make sure you have installed Heroku on your machine and created a free account. See the link below to create an account.

Cloud Application Platform | Heroku

Whether you're building a simple prototype or a business-critical product, Heroku's fully-managed platform gives you...

www.heroku.com

According to the documents, Heroku CLI requires Git. Hence, if you don’t have Heroku, click on the link below, and you can install it.

The Heroku CLI

The Heroku Command Line Interface (CLI) makes it easy to create and manage your Heroku apps directly from the terminal...

devcenter.heroku.com

When you installed Heroku on your machine and created an account, it's time to log in to your Heroku account from the terminal. Run the following command in your terminal:

```
heroku login
```

The terminal returns the following line:

"heroku: Press any key to open up the browser to login or q to exit:"

Just click any key, and your browser will automatically open up. You should see the following:

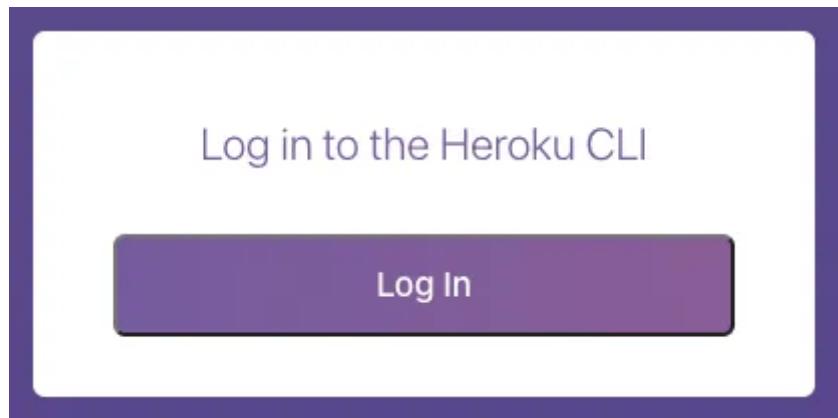


Image by author

Next, you click Log In and enter your credentials. After that, you will see the following:

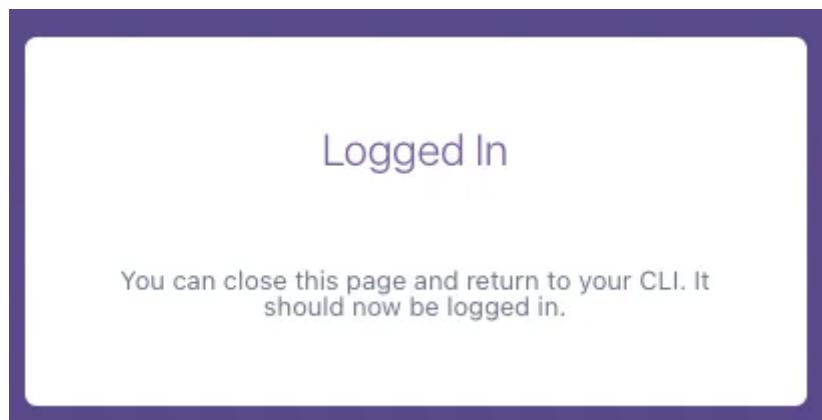


Image by author

As the picture above says, you can close the window and return to the CLI. The next step is to create the Heroku app. Type in the following command:

```
heroku create sample_app22a
```

Here I am creating an app called “sample_app22a”. If you don’t specify a name, it will create an app with random numbers. Hence I suggest picking a name for your app. The last step is to push our files to Heroku. Since Heroku uses git, the command line will be easy to understand. Type in the following:

```
git push heroku master
```

Once you executed the command above, it will take a couple of seconds for Heroku to install all the libraries and set up your app. You will see a URL in the output, and that will be the URL of your app. Please see below for reference.

```
b.1 zipp-3.4.1
remote: ----> Discovering process types
remote:          Procfile declares types -> web
remote:
remote: ----> Compressing...
remote:          Done: 137.2M
remote: ----> Launching...
remote:          Released v3
remote:          https://sample-app22a.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sample-app22a.git
```

Image by author

We didn’t receive an error message, and it’s now time to check out the app. Click on the link below, and you can see our Streamlit app on Heroku.

Streamlit

[Edit description](#)

sample-app22a.herokuapp.com

There you have it, folks. Now that your app is working, you want to push your changes to your repo as well.

```
git add -A  
git commit -m "Type in a message"  
git push
```

Final Comments:

I suggest waiting until you get the Streamlit Sharing invitation, so you can take advantage of that feature instead of deploying it to Heroku. I created an app myself, which I will write about in the coming days. The rendering time took a bit longer for a few plots when using Heroku. However, if you don't have the time and need to deploy as soon as possible, Heroku will get the job done too.

Hopefully, this article gives you a better understanding of how to deploy your Streamlit app to Heroku. Let me know if you have any questions on this topic or have some feedback. If you enjoyed this article, I'd be very grateful if you would share it on any social media platforms. I will introduce you to the Covid19-Dashboard web app that I created with Streamlit and its Sharing feature in my next article. Stay tuned for that! Until next time 🤗

Navid Mashinchi Portfolio

I am Navid Mashinchi a Data Scientist based in Sacramento, California. I have been working in the field of analytics...

www.navidma.com

Data Science

Tech

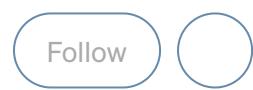
Streamlit

Heroku

Data Science Projects



Follow

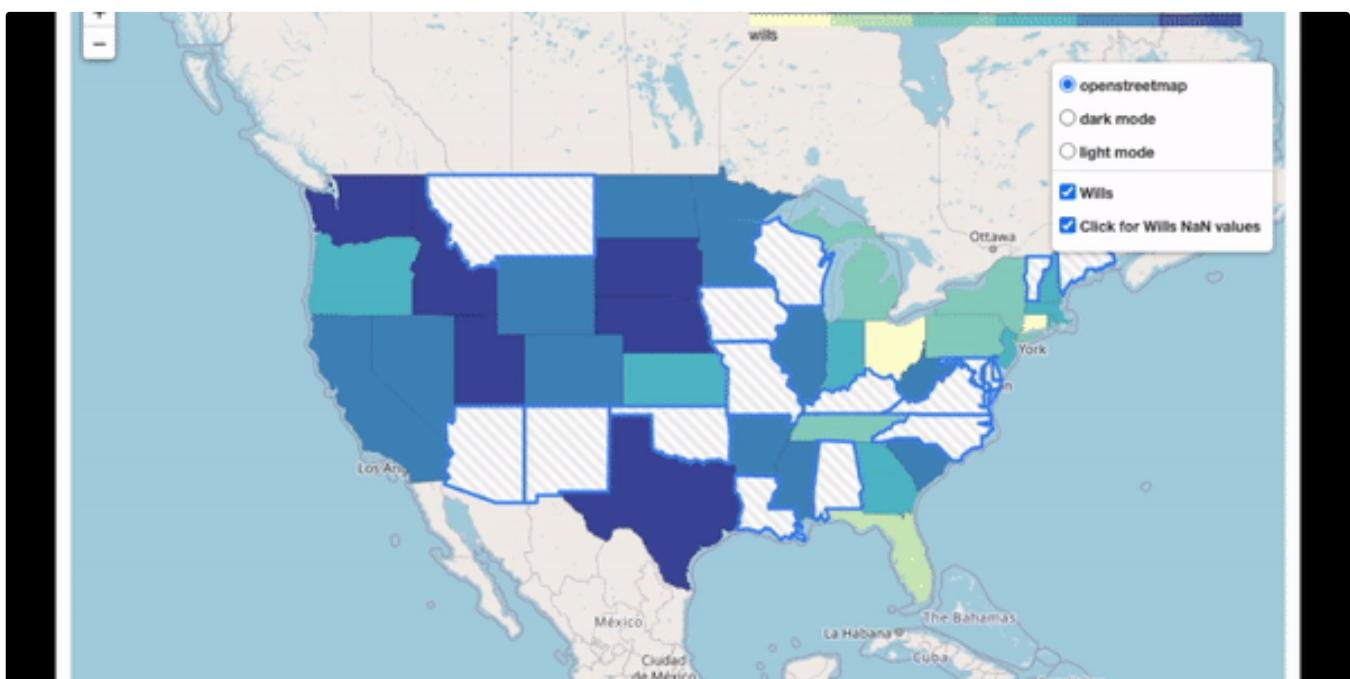


Written by Navid Mashinchi

348 Followers · Writer for Towards Data Science

Data Scientist at Kohl's | Adjunct Professor at University of Denver | Data Science Mentor at SharpestMinds

More from Navid Mashinchi and Towards Data Science



 Navid Mashinchi in Towards Data Science

How to step up Your Folium Choropleth Map skills

Take your Choropleth Map to the next level

10 min read · Feb 24, 2021

 104

 1



...



Bex T. in Towards Data Science

130 ML Tricks And Resources Curated Carefully From 3 Years (Plus Free eBook)

Each one is worth your time

★ · 48 min read · Aug 1

👏 2.4K

💬 9



...



Kenneth Leung in Towards Data Science

Running Llama 2 on CPU Inference Locally for Document Q&A

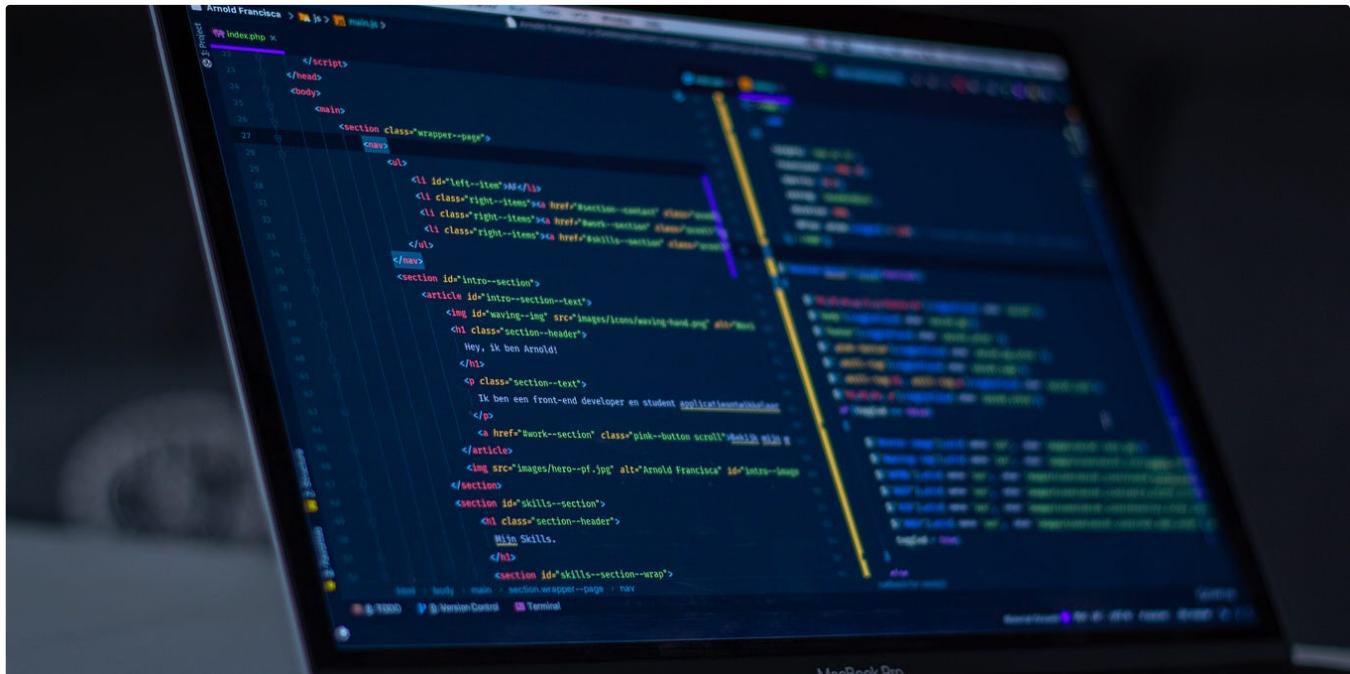
Clearly explained guide for running quantized open-source LLM applications on CPUs using LLama 2, C Transformers, GGML, and LangChain

◆ · 11 min read · Jul 18

2K 27



...



Navid Mashinchi in CodeX

3 Interview Prep Platforms That Prepared Me For Data Science Interviews

Resources That Will Help You Ace Your Technical Assessment

5 min read · Aug 13, 2021

88



...

See all from Navid Mashinchi

See all from Towards Data Science

Recommended from Medium



 Prashun chauhan in Fasal Engineering

Streamlit: Whip up a Web App In Minutes

Streamlit is an open-source Python library that is used to make data apps without any knowledge of frontend service. The best part is if you...

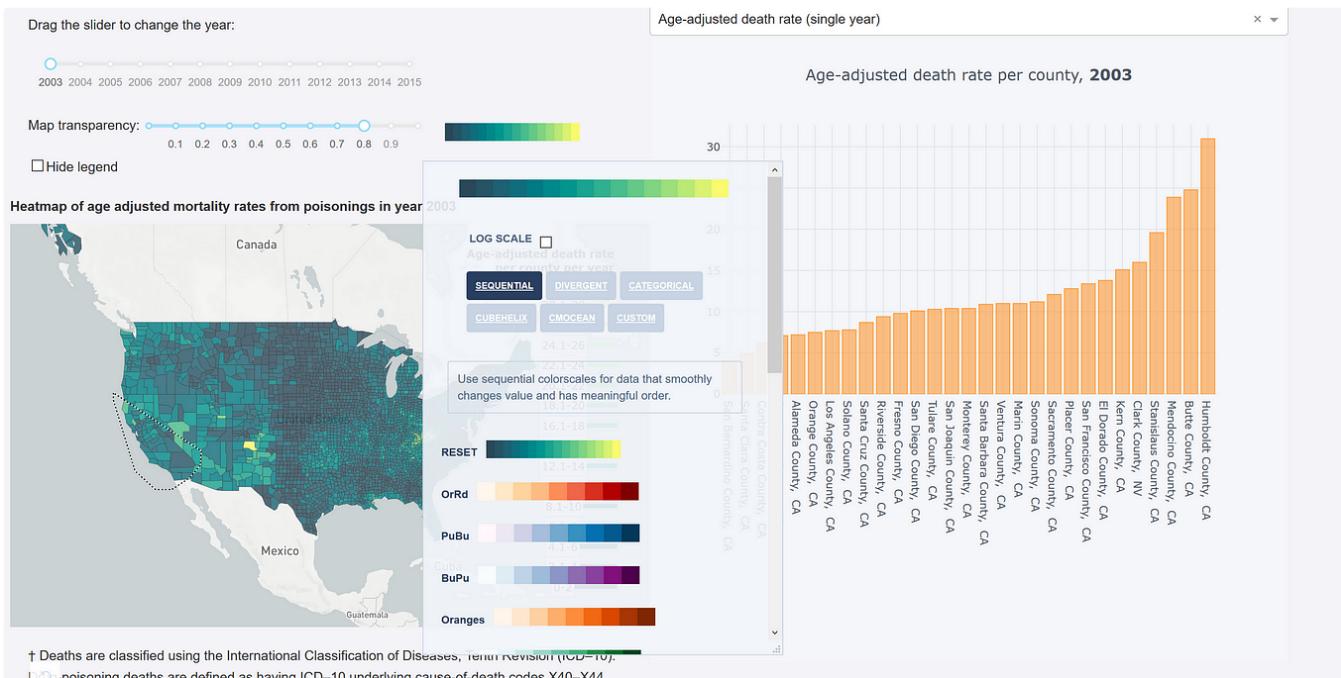
6 min read · Mar 7

 27

 1



...



Yahya sghiouri

Dockerize Your Dash App

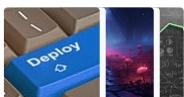
In today's world of software development, containerization has become a game-changer. Docker, a popular containerization platform, has...

10 min read · Jul 12

6

...

Lists



Predictive Modeling w/ Python

20 stories · 265 saves



New_Reading_List

174 stories · 66 saves



Practical Guides to Machine Learning

10 stories · 279 saves



Apple's Vision Pro

7 stories · 15 saves



 Anuj Karn in Towards Data Engineering

Deploy Streamlit Web Apps to the Huggingface and Streamlit Cloud

The growing popularity of web apps as a way to share data science projects is driven by the emergence of user-friendly Python libraries...

6 min read · Mar 31



FREE Web Hosting

Deploy to Firebase in 5 Minutes

 Hayk Simonyan in Level Up Coding

Deploying Your Website to Firebase

Deploy your website frontend to Firebase for FREE in less than 5 minutes with this complete guide!

6 min read · Mar 20



...

RENDER

Product Pricing Blog Docs Careers

GET STARTED

SIGN IN

The fastest way to host all your APIs.

Render is a unified cloud to build and run all your apps and websites with free TLS certificates, a global CDN, DDoS protection, private networks, and auto deploys from Git.

GET STARTED FOR FREE

CONTACT SALES



Olusegun Ajose

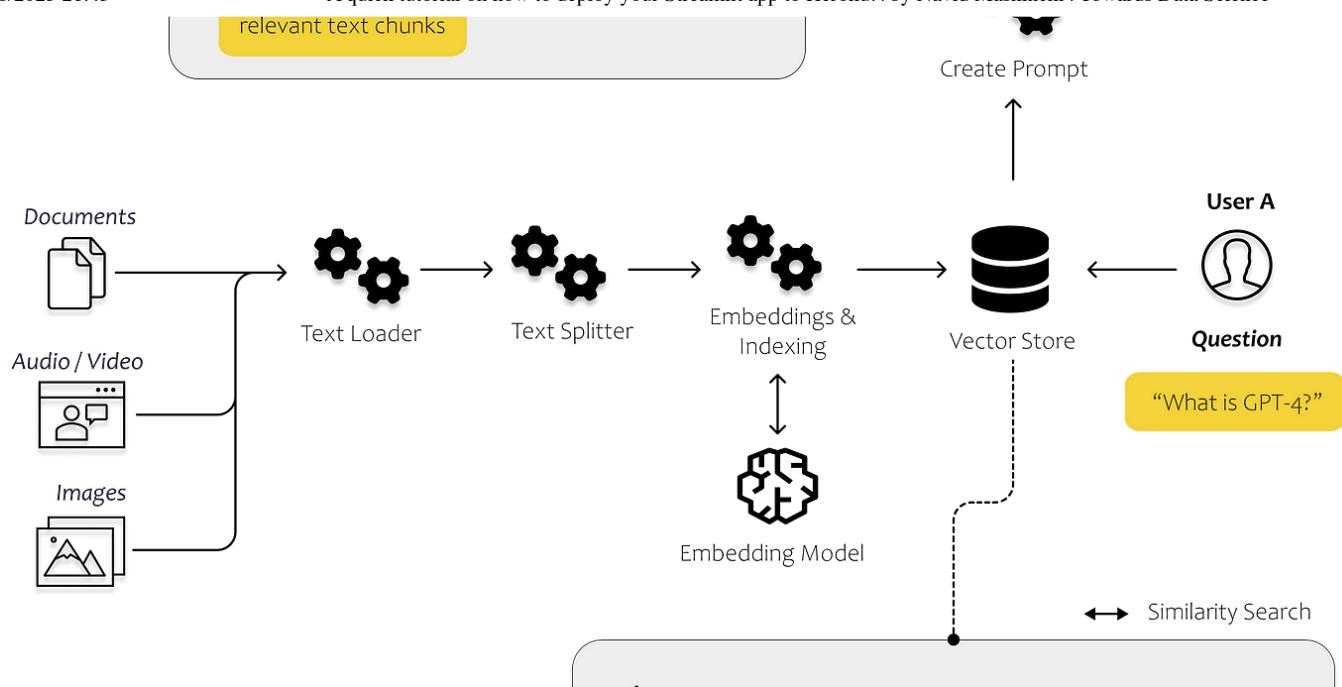
7 Free One-Click Solutions to Host Python Apps

In this article, I will present you with 7 one-click solutions hosting platforms you can use to deploy your Python app or script. Most of...

7 min read · Mar 4



...



 Dominik Polzer in Towards Data Science

All You Need to Know to Build Your First LLM App

A step-by-step tutorial to document loaders, embeddings, vector stores and prompt templates

★ · 26 min read · Jun 22

 4.3K

 40



...

See more recommendations