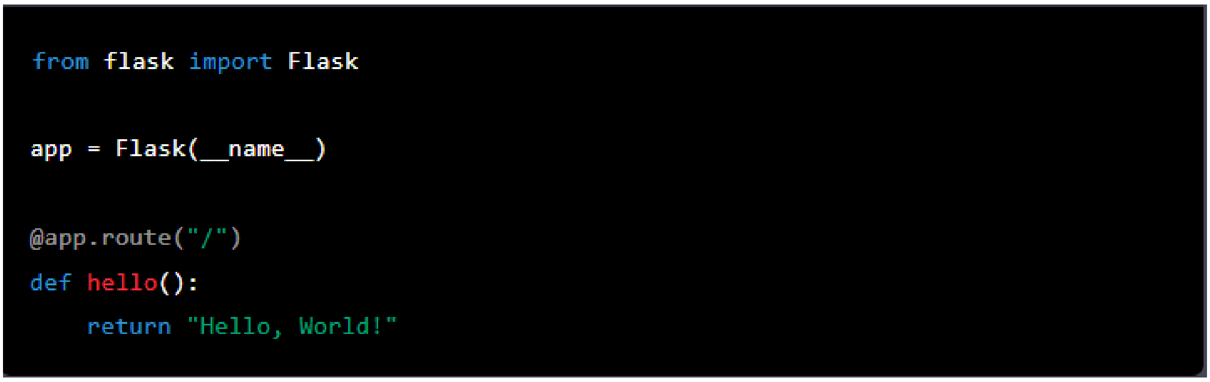
Deploying a Flask app to Netlify

Step 1:

If you don't already have a Flask app, you can create a new one by following the Flask quickstart guide. For example, you can create a file named app.py and add the following code to it:

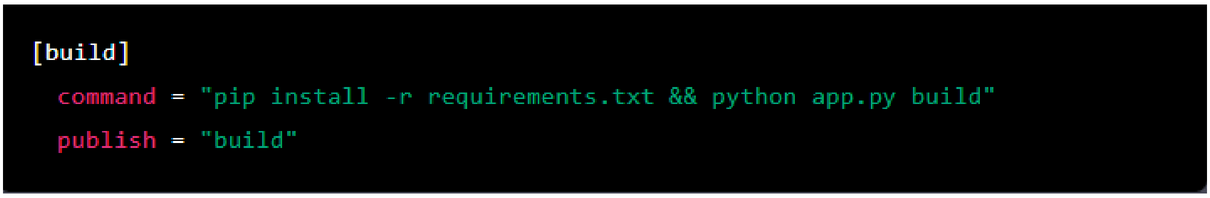


```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello, World!"
```

This app simply returns the string "Hello, World!" when you visit the root URL.

Step 2:

Create a new file named netlify.toml at the root of your project directory. In this file, specify the build command to build the Flask app and the publish directory where the built app will be stored.



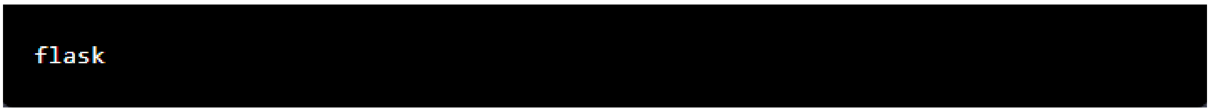
```toml
[build]
  command = "pip install -r requirements.txt && python app.py build"
  publish = "build"
```

Here, we specify the build command to install the dependencies listed in requirements.txt and then run the Flask app using the app.py file. The publish directory specifies where the built app will be stored.

Step 3:

Create a requirements.txt file listing all the Python dependencies required for your Flask app to run.

For example, if your Flask app requires flask, you can add the following line to your requirements.txt file:



```
flask
```

Step 4:

Create a new repository on GitHub and push your Flask app code to it.

You can create a new repository on GitHub by clicking on the "New" button on the main page of your GitHub account. Once you have created the repository, you can push your Flask app code to it using the git push command.

Step 5:

Sign up for a Netlify account, and from the Netlify dashboard, create a new site by linking it to your GitHub repository.

After signing up for a Netlify account, you can create a new site by clicking on the "New site from Git" button on the Netlify dashboard. Follow the prompts to link your GitHub repository to Netlify.

Step 6:

In the Netlify site settings, configure the build command to python app.py build and the publish directory to build.

Once you have linked your GitHub repository to Netlify, you can configure the build settings by going to the "Build & Deploy" section of the Netlify dashboard. Under "Build settings", set the build command to python app.py build and the publish directory to build.

Step 7:

Commit the changes to the netlify.toml file and push them to your GitHub repository.

After making the necessary changes to the netlify.toml file, commit the changes and push them to your GitHub repository using the git push command.

Step 8:

Netlify will automatically build and deploy your Flask app to a URL provided in the Netlify dashboard.

Once the changes have been pushed to GitHub, Netlify will automatically build and deploy your Flask app. You can view the deployment progress and status on the Netlify dashboard. Once the deployment is complete, you can access your Flask app by visiting the URL provided by Netlify.

Step 9:

You can access your Flask app by visiting the URL provided by Netlify.

After the deployment is complete, you can visit the URL provided by Netlify to access your Flask app. For example, if your app returns "Hello, World!" when you visit the root URL, you should see that same message when you visit the URL provided by Netlify.