NHH

BAN438

Workshop 1 – deployment

Introduction

- **We have used Dash to build applications in pure Python.**

- **After we have built the app, we deploy it to make it accesible to users.**
  - Deployment – connecting an application to a specific URL on a server

- **So far, we have deployed our apps *locally* on our computers.**

- **To share the app with others, we must instead deploy it to an *online* server.**

- **We will be using a combination of GitHub and render to deploy apps online.**

➢ **Let us deploy the spot price app (version 2) from the mandatory assignment. Check out the app here.**

Introduction

There are 5 steps to deploy your app online:

1. **Set up a virtual environment for your app (to debug the app)**
2. **Convert your app from a Jupyter notebook (.ipynb) to a Python script (.py).**
3. **Create the requirements.txt file for your virtual environment.**
4. **Upload your app to GitHub.**
5. **Deploy your app from GitHub to render.**

Step 1: Set up virtual environment

1. **Open Anaconda Prompt (Terminal on Mac)**

2. **Create an environment called «spot-price-app» with Python (version 3.11.8) installed:**
   - `conda create --name spot-price-app python=3.11.8`

3. **Activate the environment:**
   - `conda activate spot-price-app`

4. **Install necessary packages:**
   - `conda install pandas=2.0.3 openpyxl plotly=5.18.0 dash=2.14.2 dash-bootstrap-components dash-bootstrap-templates`

Step 2: Convert app to .py file

- **Convert «spot-price-app.ipynb» to a Python file:**
  a) In Jupyter Notebook, open the notebook and download it as a .py file.
  b) In Anaconda prompt (in base environment), navigate to the folder with the notebook and execute the following command:
     - `jupyter nbconvert --to python spot_price_app.ipynb`

- **Check that the conversion was successful by launching the .py file from Anaconda Prompt:**
  - Activate the “spot-price-app” environment
  - Use the “cd” command to navigate to the folder with “spot_price_app.py”
  - Execute the following command:
    - `python spot_price_app.py`

- **Open «spot_price_app.py» in a text editor (e.g. Notepad) or Spyder. There are three things that we must change in the code to prepare it for online deployment:**

Step 2: Convert app to .py file

1. Add __name__ inside the function call to Dash.

2. Add the line of code “server = app.server” after creating the Dash object.

```
# In[ ]:


app = Dash(__name__, external_stylesheets = [dbc.themes.FLATLY, dbc_css])
server = app.server

text = """
This dashboard shows the hourly spot electricity price in Norway in January, 2023.

Data is extracted from the [ENTSO-E Transparency Platform](https://transparency.entsoe.eu/).

app.layout = dbc.Container(
    children = [
```

Step 2: Convert app to .py file

3. **Place the code for launching the app inside the following if-statement:**

```
    subset = subset[['MTU', 'Day-ahead price']].copy()

    return dbc.Table.from_dataframe(subset, striped = True, bordered = True, hover = True)


if __name__ == '__main__':
    app.run(debug = True)


# In[ ]:
```

Step 3: Create requirements.txt

- **We must now create the requirements.txt file:**
  - A text file that lists all of the necessary packages to run application.
  - This will allow Render to re-create the virtual environment for the app.

- **Open Notepad (or any other text editor):**
  1. List all of the packages necessary to run your application (use double equality sign to specify package version)
  2. Add the package “gunicorn” (deployment will fail without it!)
  3. Store the file as «requirements.txt»

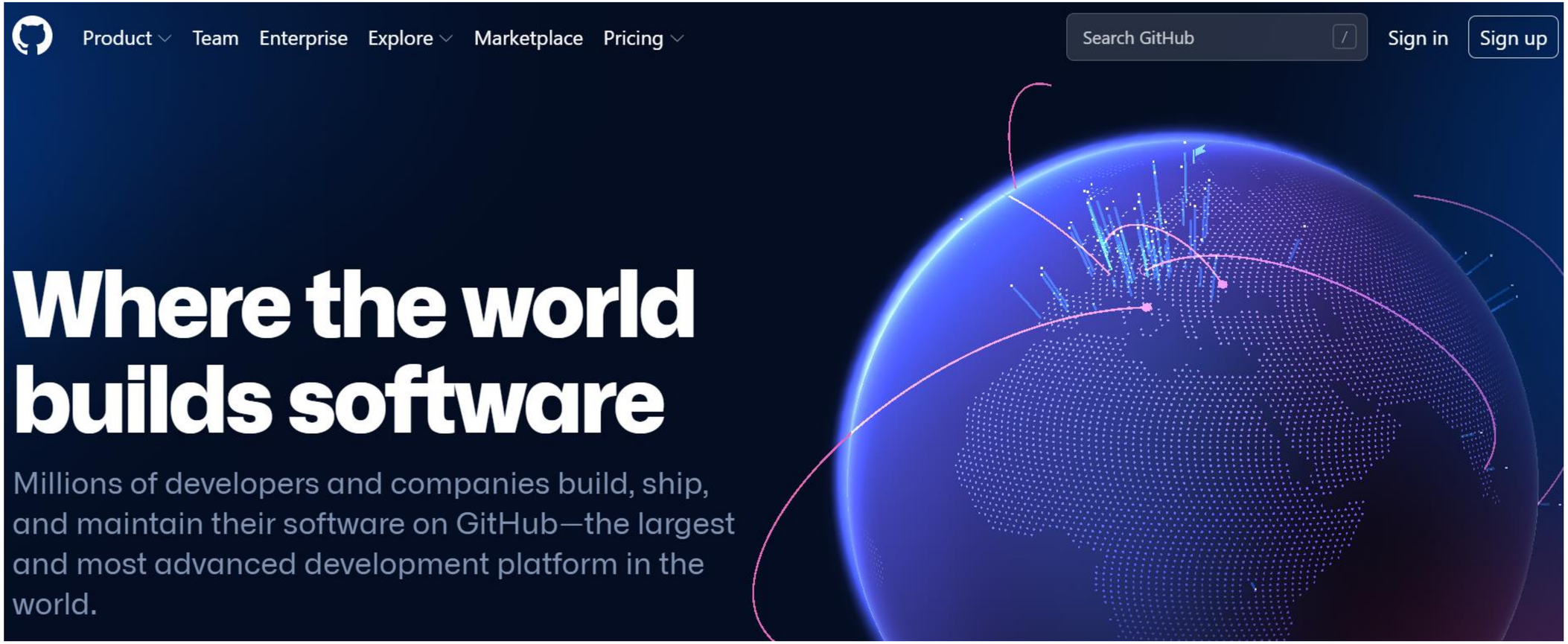
requirements - Notepad

File Edit Format View Help

```
pandas==2.0.3
openpyxl
plotly==5.18.0
dash==2.14.2
dash-bootstrap-components
dash-bootstrap-templates
gunicorn
```

Step 4: Upload your app to GitHub

- **A code hosting platform for version control and collaboration.**
- **Everyone are on GitHub!**



Step 4: Upload your app to GitHub

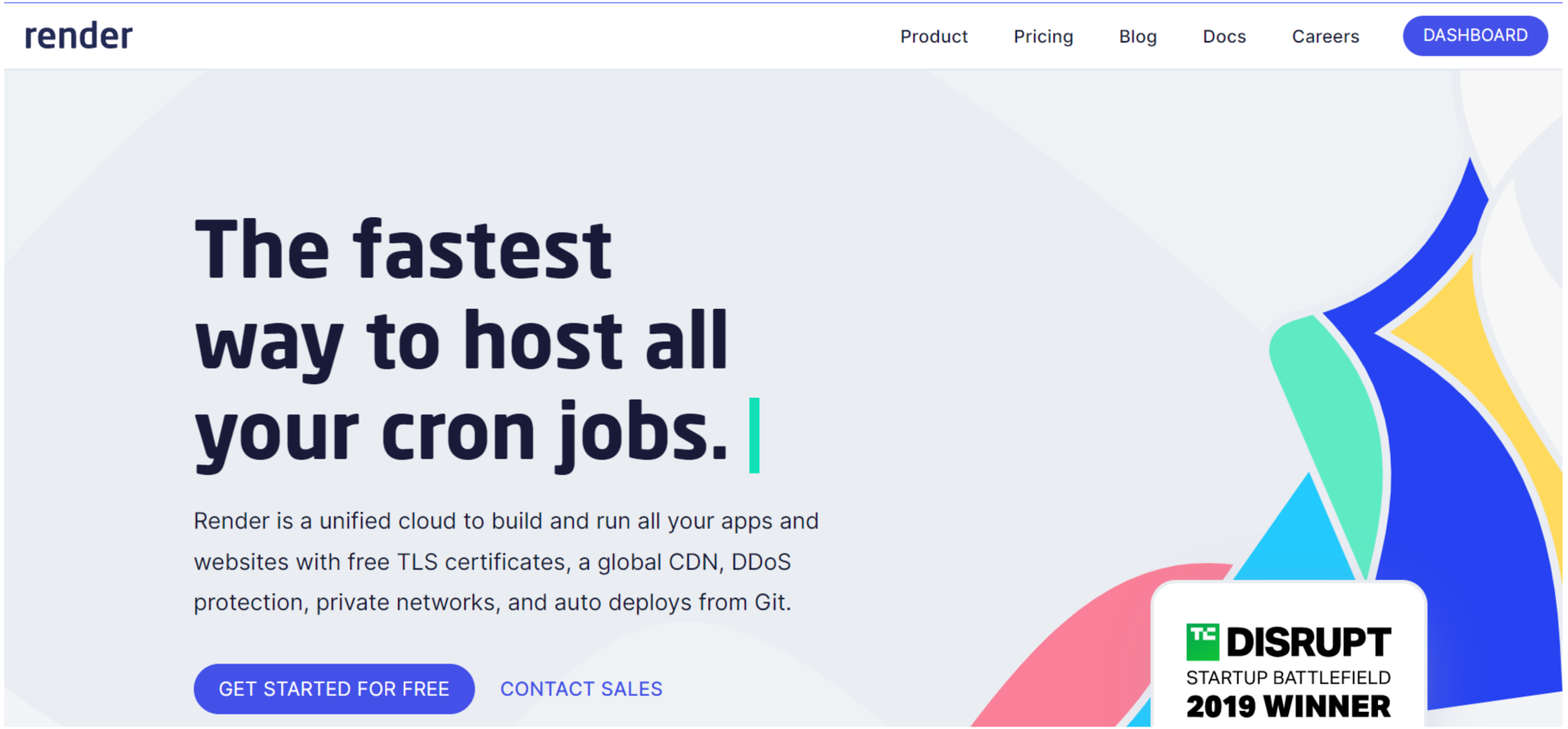
- **Create an account on GitHub (or log into your account if you already have one).**

- **Once logged into your account, create a new (public) repository and give it a unique name, e.g. spot-price-app-<your name>.**

- **Upload the app file, environment file and the data file to the GitHub repo that you created:**
  - spot_price_app.py
  - requirements.txt
  - 2023_01_DayAheadPrices_12.1.D.csv

Step 5: Deploy your app on render

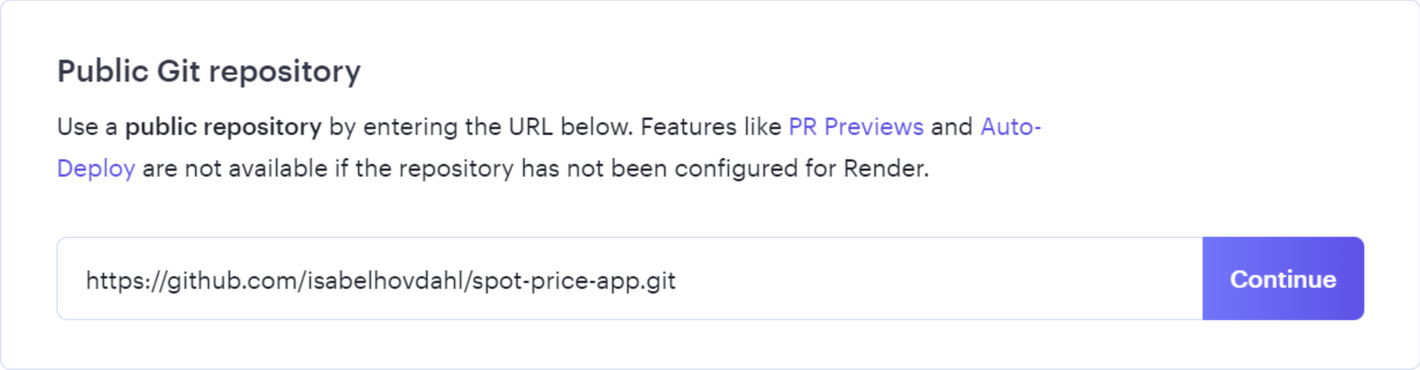
- A cloud platform that offers services to deploy, manage and scale apps.
- One of the easiest (and free) ways to deploy dash apps online.



Step 5: Deploy your app on render

- **Ceate an account on render (or log into your account if you already have one).**

- **Create a new *web service* and connect it to your repo on GitHub:**



Step 5: Deploy your app on render

- **Give your web service a unique name, e.g. «spot-price-app-<your name>» and change the start command from this:**

Start Command

This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.

```
$ gunicorn app:app
```

to this:

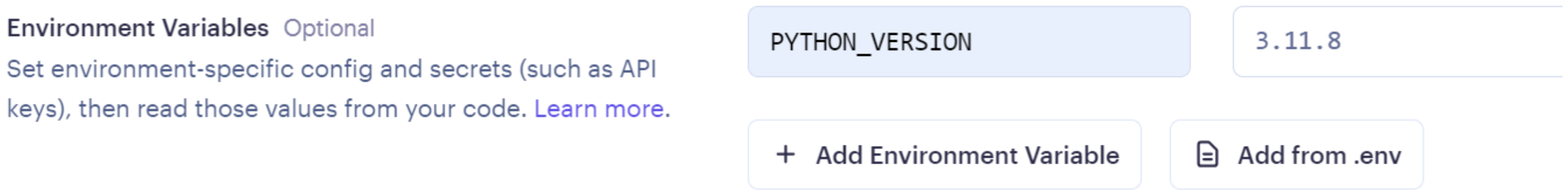
Start Command

This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.

```
$ gunicorn spot_price_app:server
```

Step 5: Deploy your app on render

- **Choose the free plan and add the following environment variable to set the correct Python version:**

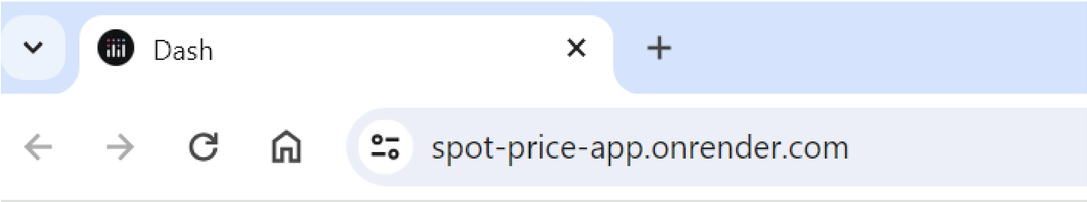


- **Create the web service. The app is now live! (unless an error occurred during the build)**

- **If you want to modify your app:**
  1. Make the changes to the files in your GitHub repo (or use git to connect your local folder to the online repo)
  2. Go to the web service in Render and press: «Manual deploy» --> «Latest commit»

Extra: add favicon and title

- **Notice that as a default, all Dash apps are given the Dash logo and title in the browser…**



- **Let us instead customize the title and logo.**

Extra: add favicon and title

- **There are two modifications that we must do in the GitHub repo:**

  1. Upload the assets folder that contains a favicon.
     - Notice that the favicon must be inside a folder called assets…
     - …and it must be stored as favicon.iso

  2. Inside the app file “spot_price_app.py” add the following line:
     - app.title = ‘Spot price dashboard’

- **Re-deploy your app to render to see the updated title and logo**