

Week 5 – Deployment on Cloud

Name: Archana Devi Ramesh

Batch code: LISUM16

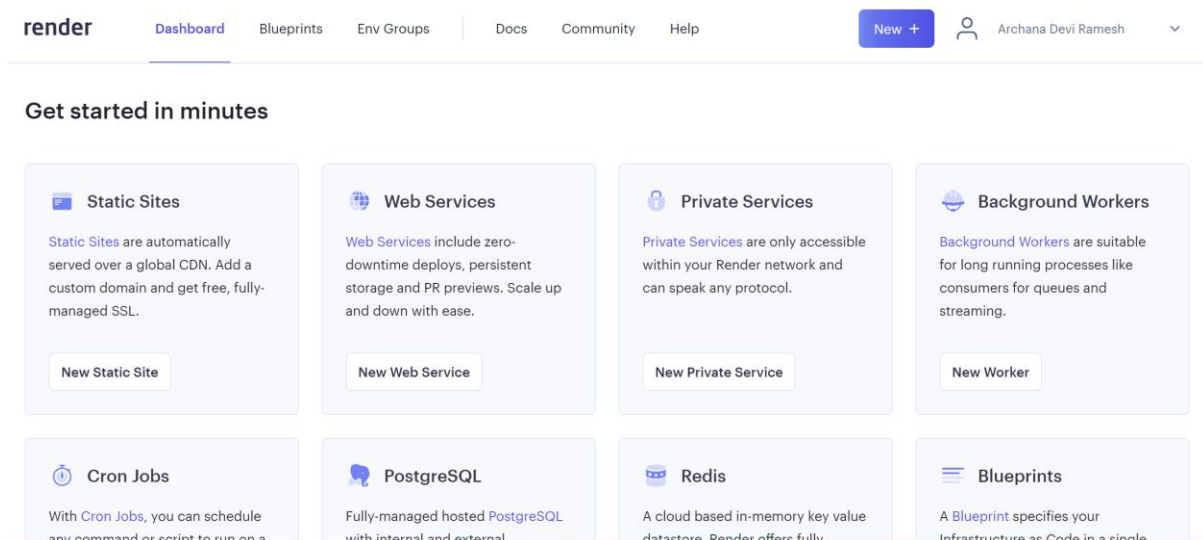
Submission date: 5th January 2023

Submitted to: Data Glacier

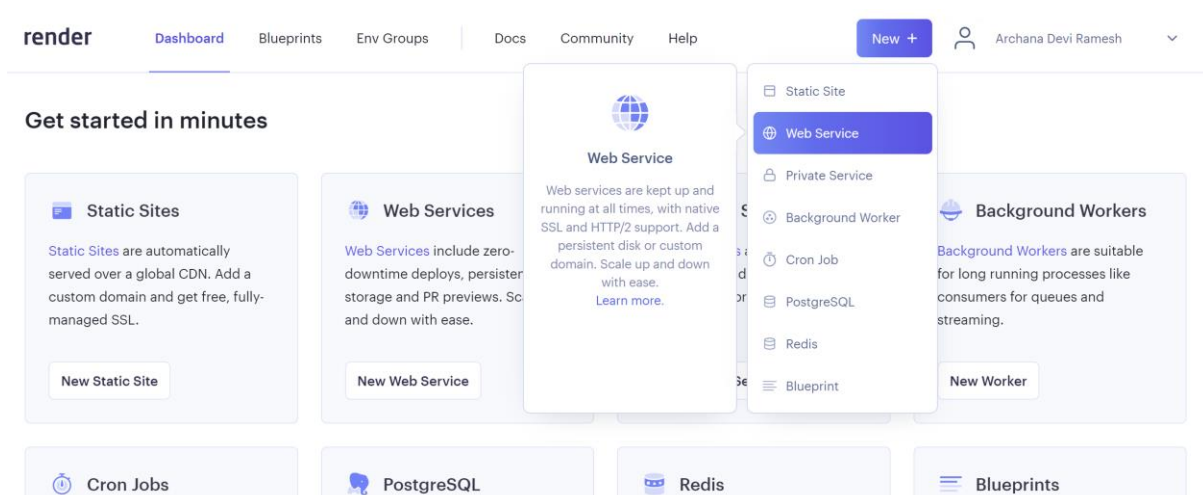
Submission Link: <https://github.com/ArchanaDeviRamesh/Data-Glacier-Week5>

Cloud Deployment steps

1. Since Heroku is not used, the open source cloud **Render** <https://render.com/> is used for this assignment.
2. The steps from [1] are followed in order to deploy the ML application on cloud.
3. Create an account in render and click on **New**



4. Click on **Web Service**



5. Connect to your Github account and select the repository to be deployed (<https://github.com/ArchanaDeviRamesh/Data-Glacier-Week4> already contains the ML application deployed using Flask submitted for Week 4)

The screenshot shows the Render dashboard with the 'Create a new Web Service' section. Under 'Connect a repository', there is a search bar and a list of repositories. The repository 'ArchanaDeviRamesh / Data-Glacier-Week4' is selected, and the 'Connect' button is highlighted. To the right, the GitHub account '@ArchanaDeviRamesh' is shown with 14 repositories, and the 'Configure account' button is visible.

6. Provide a name to the service (**iris-flower-detection-cloud-deployment**)

The screenshot shows the Render dashboard with the 'You are deploying a web service for ArchanaDeviRamesh/Data-Glacier-Week4.' section. The deployment configuration form is filled out with the following values:

- Name:** iris-flower-detection-cloud-deployment
- Region:** Oregon (US West)
- Branch:** main
- Root Directory:** Optional (e.g., src)

7. Enter **gunicorn app:flask_app** as the start command and click **Create Web Service** (**flask_app** is the name of my flask app in the file **app.py**)

render
Dashboard
Blueprints
Env Groups
Docs
Community
Help
New +
Archana Devi Ramesh

Environment
The runtime environment for your web service.

Python 3

Build Command
This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

\$ pip install -r requirements.txt

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:flask_app

<input type="radio"/> Pro	4 GB	2 CPU	\$85 / month
<input type="radio"/> Pro Plus	8 GB	4 CPU	\$175 / month
<input type="radio"/> Pro Max	16 GB	4 CPU	\$225 / month
<input type="radio"/> Pro Ultra	32 GB	8 CPU	\$450 / month

Need a [custom plan](#)? We support up to 512 GB RAM and 64 CPUs.

i Unlike paid services, free services scale down when inactive. They also have slower build times. Learn more about [free instance type limits](#).

Advanced

Create Web Service

8. It will start running and you can see the progress in the terminal. While it's running it will show the status as **“in progress”**. It will first install all the packages given in **requirements.txt**

iris-flower-detection-cloud-deployment
Python 3
Free Plan
ArchanaDeviRamesh/Data-Glacier-Week4
main
Connect
Manual Deploy
https://iris-flower-detection-cloud-deployment.onrender.com

Events
Logs
Disks
Environment
Shell
PRs
Jobs

i Builds too slow? Upgrade to a paid plan to go faster. Learn more about [free instance type limits](#).

January 4, 2023 at 11:21 PM
*** In progress
66c4da0 Update requirements.txt

Search logs
Search
Maximize
Scroll to top

Jan 4 11:21:03 PM ==> Cloning from https://github.com/ArchanaDeviRamesh/Data-Glacier-Week4...
Jan 4 11:21:04 PM ==> Checking out commit 66c4da01bb2162ce0cde053563a5970348d696ff in branch main

render Dashboard Blueprints Env Groups Docs Community Help New + Archana Devi Ramesh

Events

Logs Search logs Search Maximize Scroll to top

Disks

Environment

Shell

PRs

Jobs

Metrics

Scaling

```

Jan 4 11:21:21 PM Collecting MarkupSafe>=2.1.1
Jan 4 11:21:21 PM Downloading MarkupSafe-2.1.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Jan 4 11:21:21 PM Installing collected packages: numpy, scipy, joblib, threadpoolctl, scikit-learn, six, python-dateutil, pytz, p
andas, typing-extensions, zipp, importlib-metadata, itsdangerous, MarkupSafe, Werkzeug, Jinja2, click, flask, gunicorn
Jan 4 11:21:37 PM Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 flask-2.2.2 gunicorn-20.1.0 imp
ortlib-metadata-6.0.0 itsdangerous-2.1.2 joblib-1.2.0 numpy-1.21.6 pandas-1.3.5 python-dateutil-2.8.2 pytz-2022.7 scikit-learn-1.0
.2 scipy-1.7.3 six-1.16.0 threadpoolctl-3.1.0 typing-extensions-4.4.0 zipp-3.11.0
Jan 4 11:21:37 PM WARNING: You are using pip version 20.1.1; however, version 22.3.1 is available.
Jan 4 11:21:37 PM You should consider upgrading via the '/opt/render/project/src/.venv/bin/python -m pip install --upgrade pip' c
ommand.
Jan 4 11:21:38 PM ==> Generating container image from build. This may take a few minutes...

```

render Dashboard Blueprints Env Groups Docs Community Help New + Archana Devi Ramesh

Events 66c4da0 Update requirements.txt

Logs Search logs Search Maximize Scroll to top

Disks

Environment

Shell

PRs

Jobs

Metrics

Scaling

Settings

```

Jan 4 11:21:37 PM Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 flask-2.2.2 gunicorn-20.1.0 imp
ortlib-metadata-6.0.0 itsdangerous-2.1.2 joblib-1.2.0 numpy-1.21.6 pandas-1.3.5 python-dateutil-2.8.2 pytz-2022.7 scikit-learn-1.0
.2 scipy-1.7.3 six-1.16.0 threadpoolctl-3.1.0 typing-extensions-4.4.0 zipp-3.11.0
Jan 4 11:21:37 PM WARNING: You are using pip version 20.1.1; however, version 22.3.1 is available.
Jan 4 11:21:37 PM You should consider upgrading via the '/opt/render/project/src/.venv/bin/python -m pip install --upgrade pip' c
ommand.
Jan 4 11:21:38 PM ==> Generating container image from build. This may take a few minutes...
Jan 4 11:23:09 PM ==> Uploading build...
Jan 4 11:23:43 PM ==> Build uploaded in 26s
Jan 4 11:23:43 PM ==> Build successful 🎉
Jan 4 11:23:43 PM ==> Deploying...
Jan 4 11:23:59 PM ==> Starting service with 'gunicorn app:flask_app'

```

9. After a successful build, the service starts with the **gunicorn app:flask_app** command and the status changes from “in progress” to “**live**”. To run the application in the browser, click on the url provided below the web service name (<https://iris-flower-detection-cloud-deployment.onrender.com>)

render Dashboard Blueprints Env Groups Docs Community Help New + Archana Devi Ramesh

WEB SERVICE

🌐 iris-flower-detection-cloud-deployment

Python 3 Free Plan ArchanaDeviRamesh/Data-Glacier-Week4 main Connect Manual Deploy

<https://iris-flower-detection-cloud-deployment.onrender.com>

Events

Logs

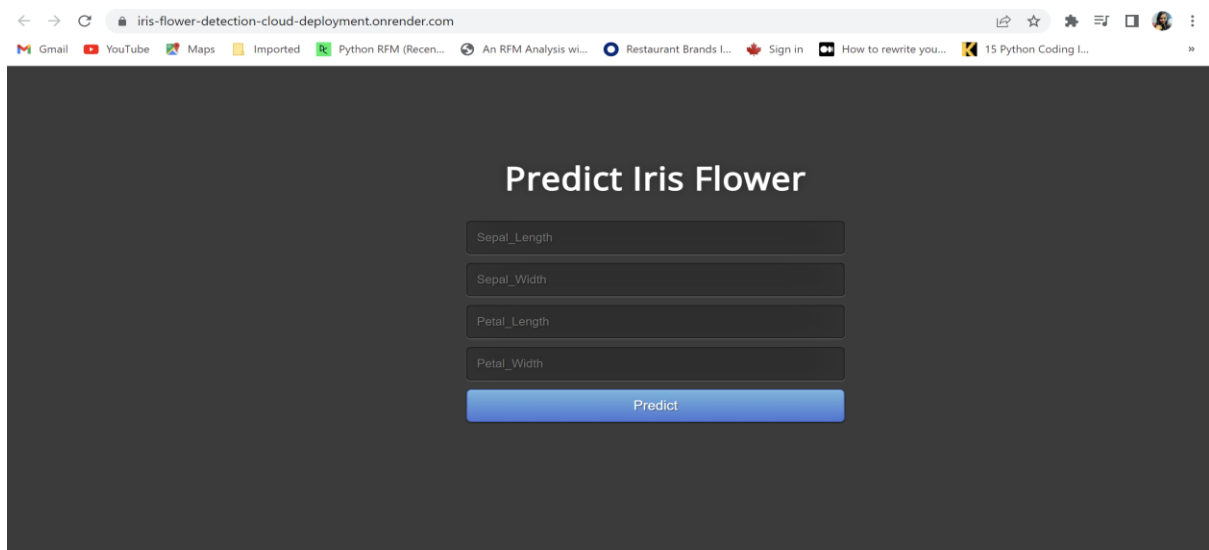
Disks

Environment

January 4, 2023 at 11:21 PM Live

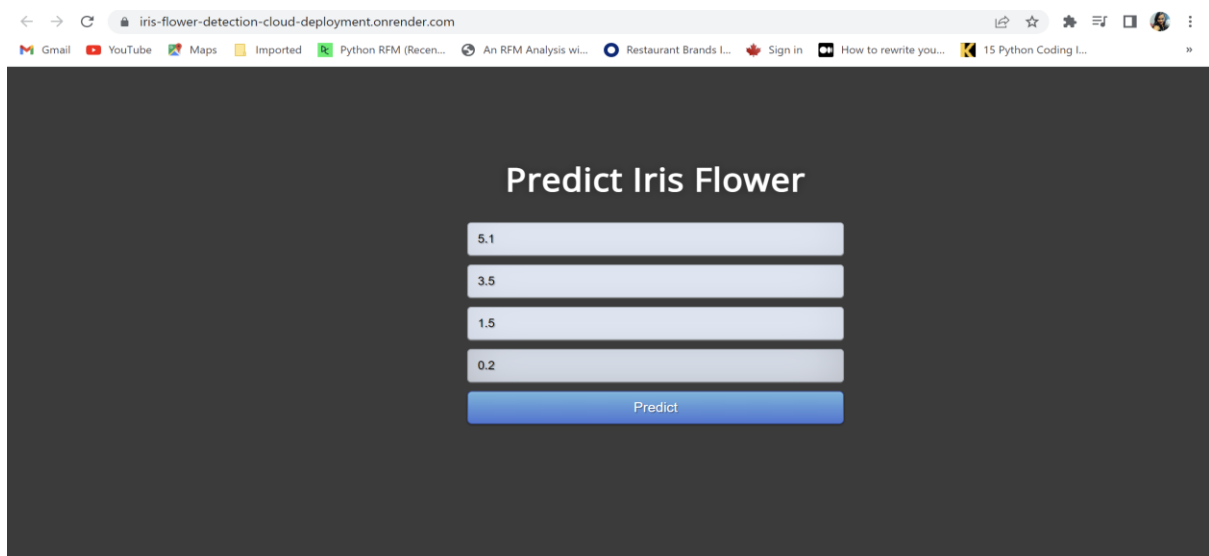
66c4da0 Update requirements.txt

10. It will open the ML application on the web browser



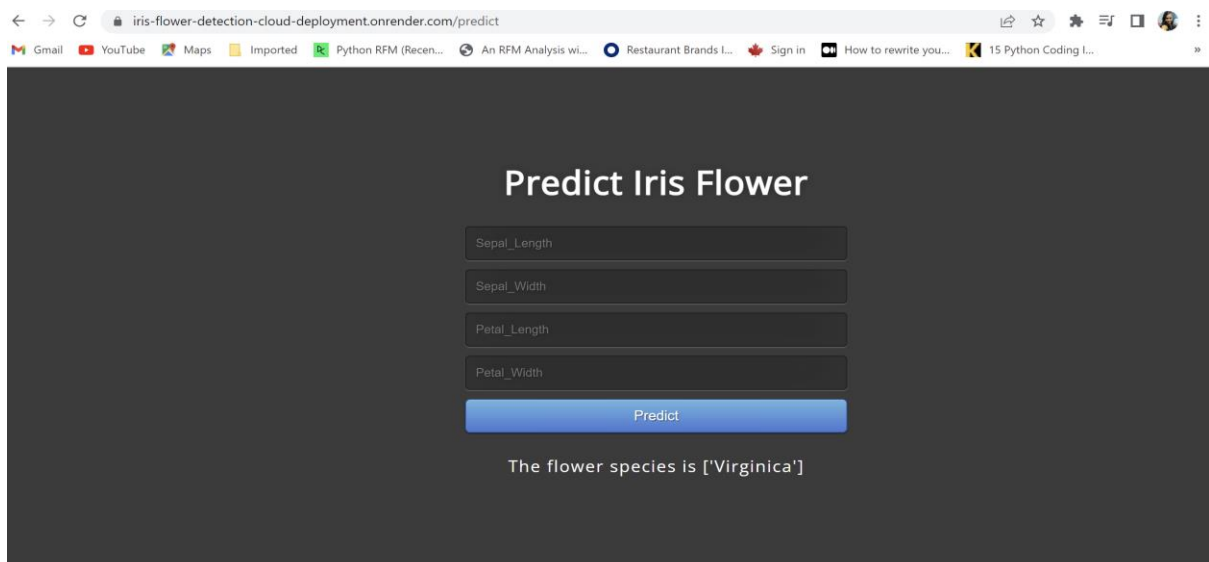
The screenshot shows a web browser window with the URL `iris-flower-detection-cloud-deployment.onrender.com`. The page has a dark background and is titled "Predict Iris Flower". It features four input fields for "Sepal_Length", "Sepal_Width", "Petal_Length", and "Petal_Width", each with a light gray border. Below these fields is a blue "Predict" button. The browser's address bar and tabs are visible at the top.

11. Testing the ML application



This screenshot shows the same "Predict Iris Flower" application, but now the input fields contain numerical values: "5.1" for Sepal_Length, "3.5" for Sepal_Width, "1.5" for Petal_Length, and "0.2" for Petal_Width. The blue "Predict" button remains at the bottom. The browser window and tabs are consistent with the previous screenshot.

12. The ML application successfully predicts the flower species



The screenshot shows the "Predict Iris Flower" application after a successful prediction. The input fields are now empty, and a blue "Predict" button is still present. Below the button, the text "The flower species is ['Virginica']" is displayed in a light gray font. The browser window and tabs are consistent with the previous screenshots.

References:

[1] Heroku Alternative | Learn to deploy Python application on Render | Step by step deployment guide, Raj Kapadia,
<https://www.youtube.com/watch?v=OBGaCULCZzg>