Deploy a Python Dashboard on AWS

Published on June 18, 2020

Sreejith Munthikodu
Data Scientist at Checkfront

Recently, I created an interactive covid-19 dashboard in Python using plotly dash. I would like to share the steps I followed to get the app running on an AWS EC2 instance. I also scheduled the EC2 instance to fetch up to date data from the data source.

Deployed app

Project Github Repo

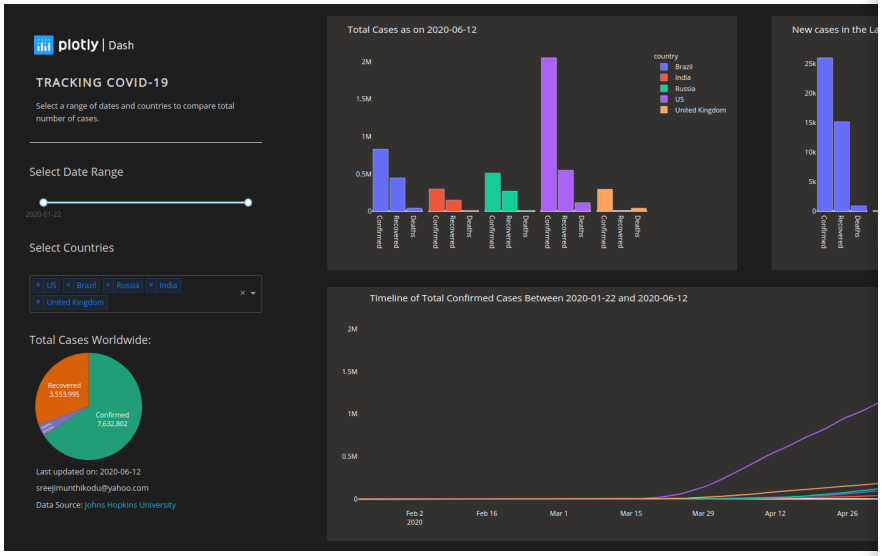
Data Source: CSSE @ Johns Hopkins University

The app

Plotly dash is an opensource framework to build enterprise-ready analytic web apps without having to write javascript code. It empowers data analysts and data scientists to publish their dashboards and data analytics products without having to worry about the complex tasks

quickly learn how to use dash, please refer to this tutorial on the offi

I used dash to build the covid-19 dashboard that I used in this article obtained from the popular COVID-19 Data Repository by the Center and Engineering (CSSE) at Johns Hopkins University. The data was Pandas and numpy. Plots were created using **Plotly Express** in Pyth dashboard looks like.



Deployment

Create an EC2 Instance

The app was created for learning purposes. So I wanted to use a free service to deploy it. Since I am on AWS free tier period, I decided to go with AWS. I used only resources that are eligible for the free tier in this project. I assume you have an AWS account set up already.

1. Create an EC2 t2.micro instance as the server for this web app. From the AWS management console, under services, click on EC2.

2. Click Instances and then Launch Instance

3. Select Ubuntu Server 18.04 LTS as the Amazon Machine Image (AMI)

5. For connecting the instance securely, create a new key pair and d
key file. Keep this file safe as this will enable anyone to connect

6. Launch Instance

7. If you go back to the EC2 service, under Instances, you will find
running. This will be the server for our app.

Configure Inbound Rules

1. It is safer to restrict the access to the EC2 instance only to our IP
instance that is running. Under description -> Security Groups, c
1.

2. Click on Inbound Rules -> Edit Inbound Rules

3. For Port 22, select My IP as the source. This ensures that only yo
connect to the EC2 instance.

4. We need to open port 80 to enable users to access the app via we
add 80 under Port Range. Select Anywhere under Source. Do the
for whichever port you are planning to run the app on.

Copy the Project to AWS S3 Bucket

1. Install AWS CLI. This is used to interact with the AWS console from the command line. Follow the instructions **here**.

2. Follow instructions **here** to get an AWS access key.

3. Configure AWS CLI by typing `aws configure` from your command line

4. Provide the Access Key Id, Access Key, Default Region Name you obtained in step 2. You may leave the default output format.

5. Now create an AWS S3 Bucket to store the project by typing `aws s3 mb s3://bucket-name`.

```
aws configure
aws s3 mb s3://bucket-name
aws s3 cp <your directory path> s3://<your bucket name> --recursiv
```

Connect to the EC2 Instance

1. Right-click on the running EC2 instance on the AWS manageme connect.
2. Follow the instructions to connect to the EC2 instance remotely line. For Ubuntu, using the example command would connect to

Install Dependencies

1. Once in the EC2 instance, you need to install the dependencies t The project root directory has a requirements.txt file.
2. Install pip3 and required dependencies using the below comman

```
sudo apt-get update
sudo apt-get -y install python3-pip
pip3 install -r requirements.txt
```

Run the app on EC2

1. Follow the instructions **here** to enable S3 access from EC2 instance
2. Copy the project directory from AWS S3 to the EC2 instance
3. CD to the project directory and run the app. You should use `screen` to start a detached terminal to run the app so that you can close the connection to the EC2 instance without killing the app. The app should be running on localhost now.

```
cd <Project Directory>

python3 app.py
```

If you are not planning the attach the web app to a domain name, you
web app server to run on 0.0.0.0 instead of localhost. This is to ensu
accessed from anywhere by http://EC2 IP:PORT. This can be done b
script.

```
app.run(host='0.0.0.0', port=8050)
```

You will now be able to access the interactive web app with http:EC

Enable Automatic Data Update

The data source for this web app is the well known **COVID-19 Data**
Center for Systems Science and Engineering (CSSE) at Johns Hopki
source is updated once every day with summary data from around th
configure our EC2 instance to download the data every day to the pr
restart the web app. Firstly, I cloned the source repo to the EC2 insta
script to automate pulling the data from the source repository, movin
directory, and restarting the web app. Then crontab is used to schedu
script every day at around 00:00 hours UTC. The bash script I used i

```
#!/bin/bash

# Change directory to JohnHopkins github repo
cd <Path to source repo in your EC2 instance>

# Update repo
sudo git pull

# Remove old data from current project
cd <Project root directory>
sudo rm -rf data/csse_covid_19_time_series*


# Move updated data to project directory
sudo cp -a <Path to csse_covid_19_time_series on source repository> <Path to data directory

sleep 3s
# Kill the dash app
sudo killall screen
# Restart the dash app
```

You may schedule to run this script multiple times a day around UTC
source is usually updated around this time.

```
crontab -e


15 00 * * * sudo bash <Path to the bash script>
15 01 * * * sudo bash <Path to the bash script>
15 02 * * * sudo bash <Path to the bash script>
15 03 * * * sudo bash <Path to the bash script>
```

I registered a domain name using AWS ROUTE 53. The app is route
name using nginx. I used the answer **here** to configure nginx.

Disclaimer

This is a project I did to learn how to build a dash app and deploy it
be the best approach for this application. I take no responsibility if th
above lead to compromising your AWS account security. Also, I assu
free tier period. Keeping the EC2 instance running beyond the free ti
charges.

Published by

Sreejith Munthikodu
Data Scientist at Checkfront
Published • 8mo

I created a covid-19 dashboard using Plotly dash. The app is deployed on an EC2 instance and it is programmed to update every day. The data source is the Johns Hopkins University covid-19 data repository. So the app will give up to date, reliable information about the covid-19 numbers. Here is a post that summarizes the steps I followed to get the app deployed on AWS.

Deployed app: **https://lnkd.in/gVxAArn**

Git Repository: **https://lnkd.in/gzHQYiw**

Data Source: **https://lnkd.in/emuX9ny**

#dataanalytics #visualizations #Plotly #dashboard #aws #covid19 #coronaravirus

Search

Re

Reactions

3 Comments

Most relevant

Add a comment…

Ricardo Sueiras • 2nd
Helping customers gain strategic advantage through open source, cloud and innovation

Nice write up!

Like · 1 | Reply · 1 Reply

Sreejith Munthikodu • 2nd
Data Scientist at Checkfront

Thank you ☺

Like | Reply

Ryan VanDorp • 3rd+
Benefit Plan Admin & Technical Analyst

Thank you, Sreejith. I've deployed R/Shiny apps to AWS in the past, and I'm swi
Python/Plotly/Dash for another project. I wanted to get a general sense of the
were any big, unexpected challenges awaiting me. It doesn't look too bad! 👍

Like | Reply

Sreejith Munthikodu
Data Scientist at Checkfront

+ Follow

n Sreejith Munthikodu

riptive Statistics

nthikodu on LinkedIn

Messaging