**Machine Learning Model Deployment using Flask**

This document discusses the various steps involved in deploying the flask app created to the cloud. This is a continuation of the steps followed in the Machine Learning Model Deployment using Flask file.

**Dataset Used:** Palmer Penguins

**ML Model:** Random Forest Classifier

**Cloud Provider:** AWS

Prerequisite Steps:

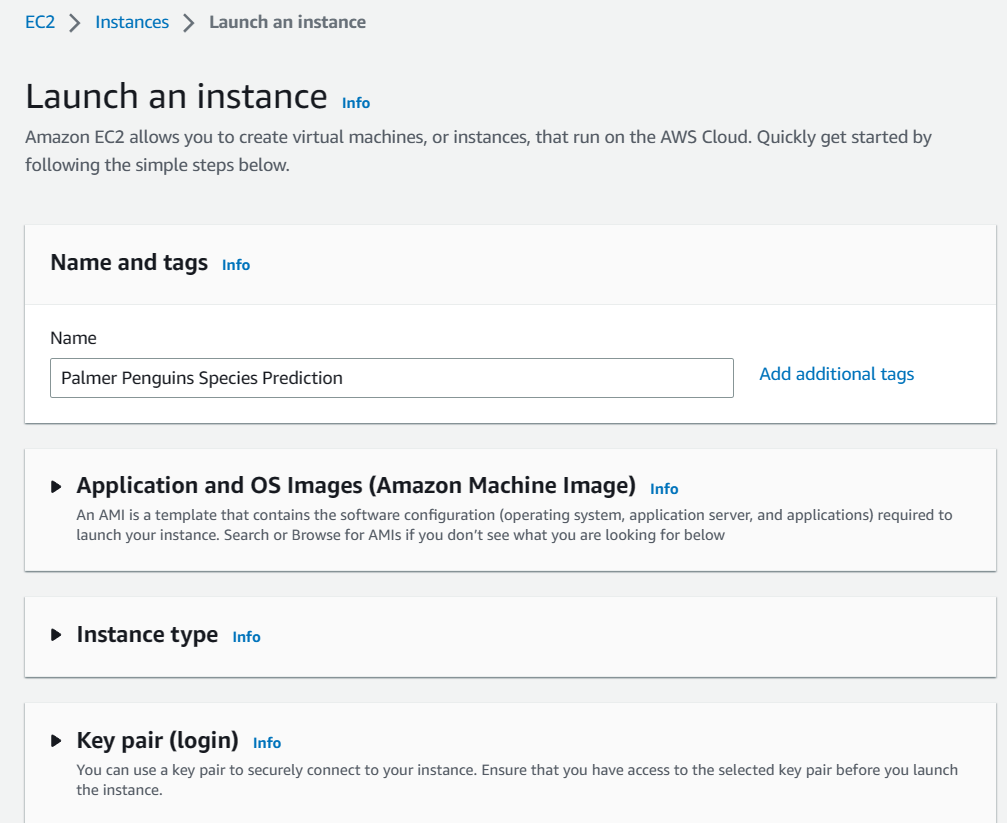
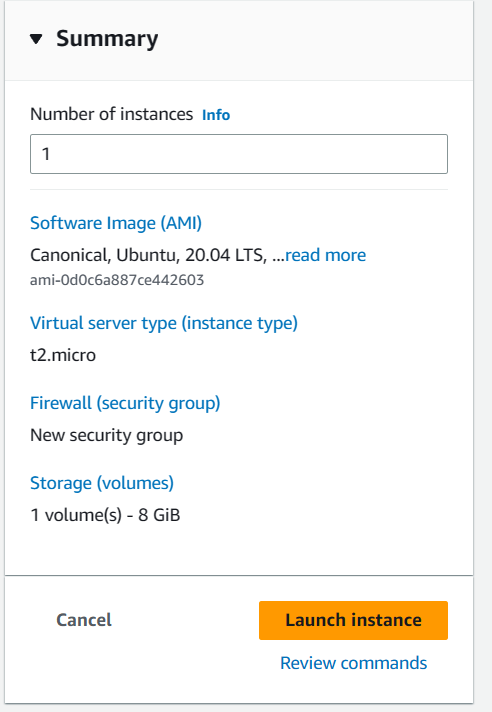
* Build a ML model in python
* Create a html template
* Create a flask app that uses the ML model and the template to create a webpage.

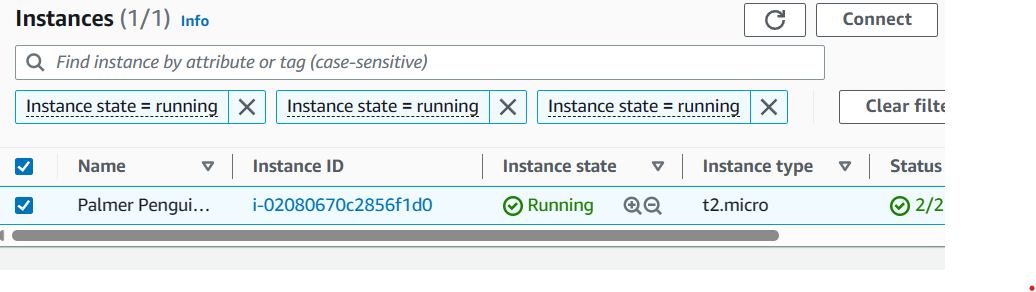
Now let us host this webpage on the AWS cloud.

* I am using the AWS free tier account.

Step 1:

The first step is to create and launch an EC2 instance. I have used a t2 micro instance with an Ubuntu environment.

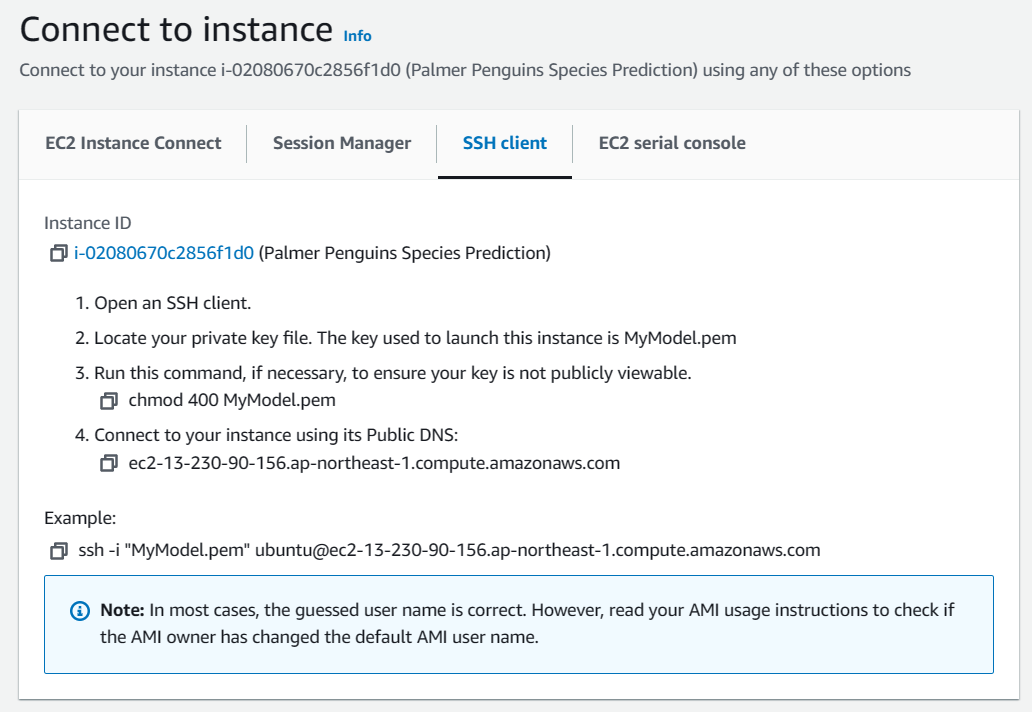
 



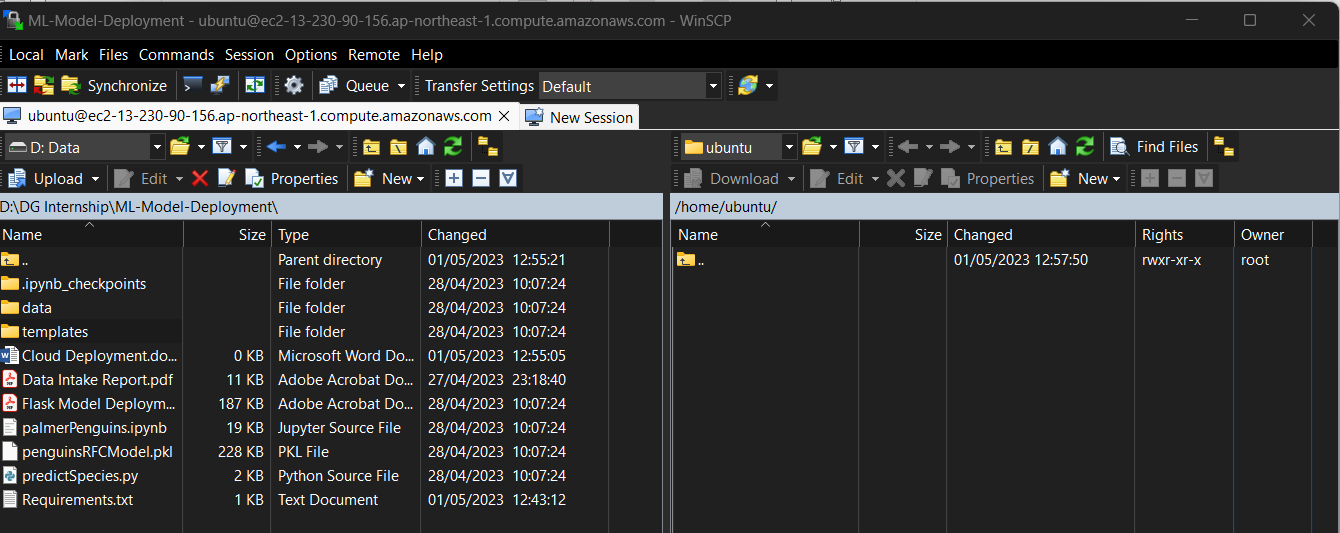
Step 2:

The next step is to connect to the created instance so that the created model can be loaded to it. We do thjs with the help of Putty and WinSCP.

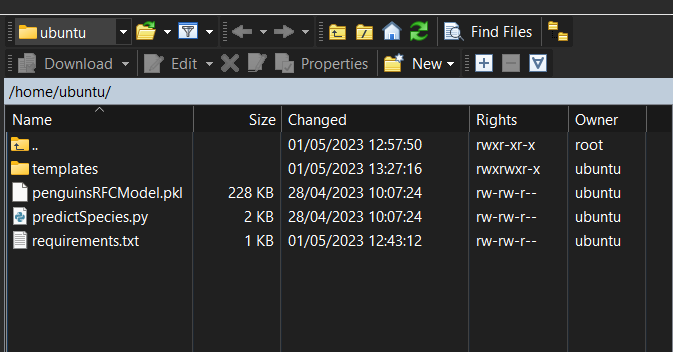
We use an SSH connection, the details for which can be found in the connection tab for the instance.



We login to WinSCP first to transfer the files that we had previously created.



The figure below shows the required files that have been transferred to the ubuntu instance. We add a requirements.txt file that contains all the libraries needed for the execution of our model and flask app..



Step 3:

We then connect to the EC2 instance using Putty. This will allow us to interact with the terminal of the instance. We download python3 and the using pip3 install to install the required packages from the requirements.txt file. The commands required are:

>> sudo apt install python3

>> sudo apt-get update && sudo apt-get install python3-pip

>>pip3 install -r requirements.txt

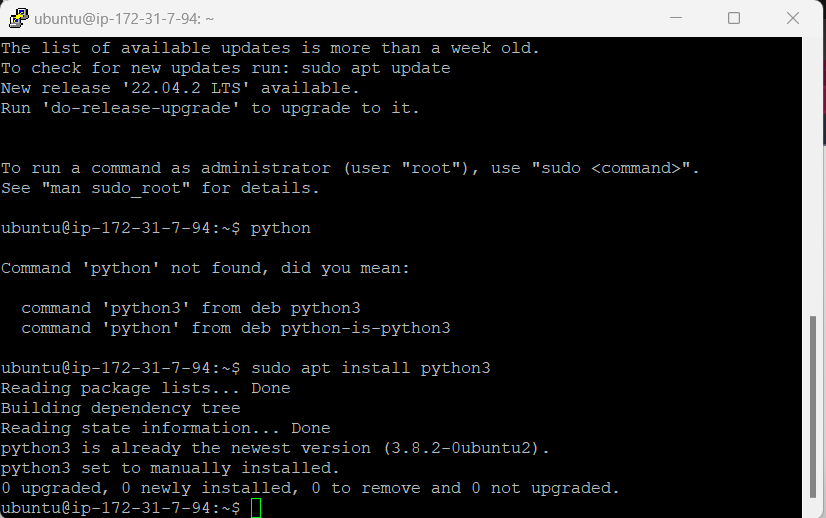
We then open the flask app in the instance and change the host name to 0.0.0.0 and add port as 8080.

Run the app using the command python3 predictSpecies.py

To let the app run continuously even after we close Putty, we can use the following command:

Screen -R deploy python3 predictSpecies.py

Close Putty and WinSCP.



Step 4:

The model is now up and running in the cloud. We can now access it using the public DNS with port 8080.

url for the model:

[Penguin Species Prediction (ec2-35-78-200-13.ap-northeast-1.compute.amazonaws.com)](http://ec2-35-78-200-13.ap-northeast-1.compute.amazonaws.com:8080/)

