MIRACLE
SOFTWARE SYSTEMS

Build a Classification Model with Azure Custom Vision

Open Lab | Digital Summit 2019

Build a Classification Model with Azure Custom Vision

Introduction

This document contains a step-by-step process of creating a model with Microsoft Custom Vision which classifies the species of the animals. This model is integrated with a web application using Python Flask.

This guide was prepared by **Miracle's Innovation Labs**

Pre-Requisites

All attendees must have their workstation (with Internet) to participate in the lab (Both PC and MAC are compatible). The following prerequisites will help you to make the Hands-on Lab experience easier.

- Active email ID for registering with the Microsoft Custom Vision
- Download Python 3.6
- Text Editor such as Visual Studio (or) Notepad++

Technology Involved

- Microsoft Custom Vision
- Python 3.6

Labs Steps

So, let us get started!

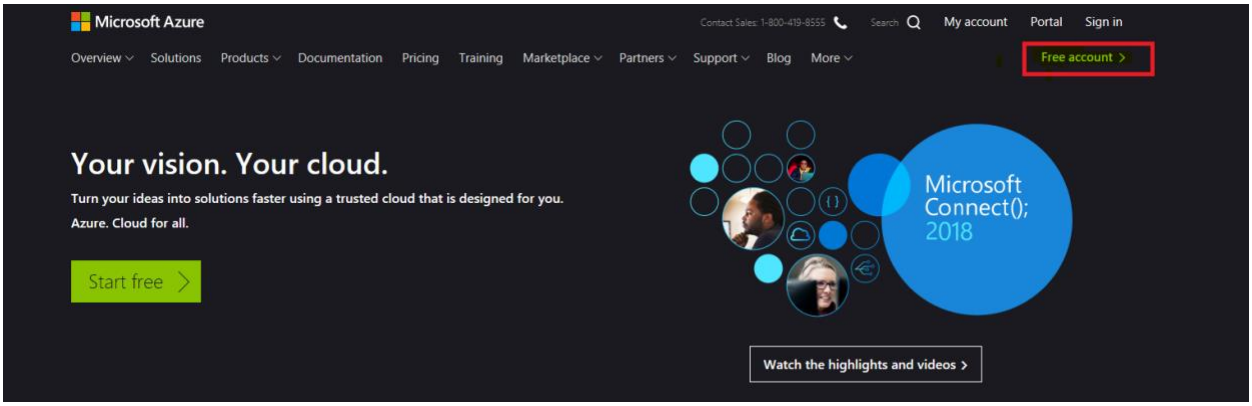
The following steps will describe how to build a model with the Microsoft Custom Vision that performs animal classification and integrate it with Web Application. The user interacts with the web application and can directly give the input image of an animal and the resultant species name will be displayed on the output page.

Step #1 | Create a Microsoft Account

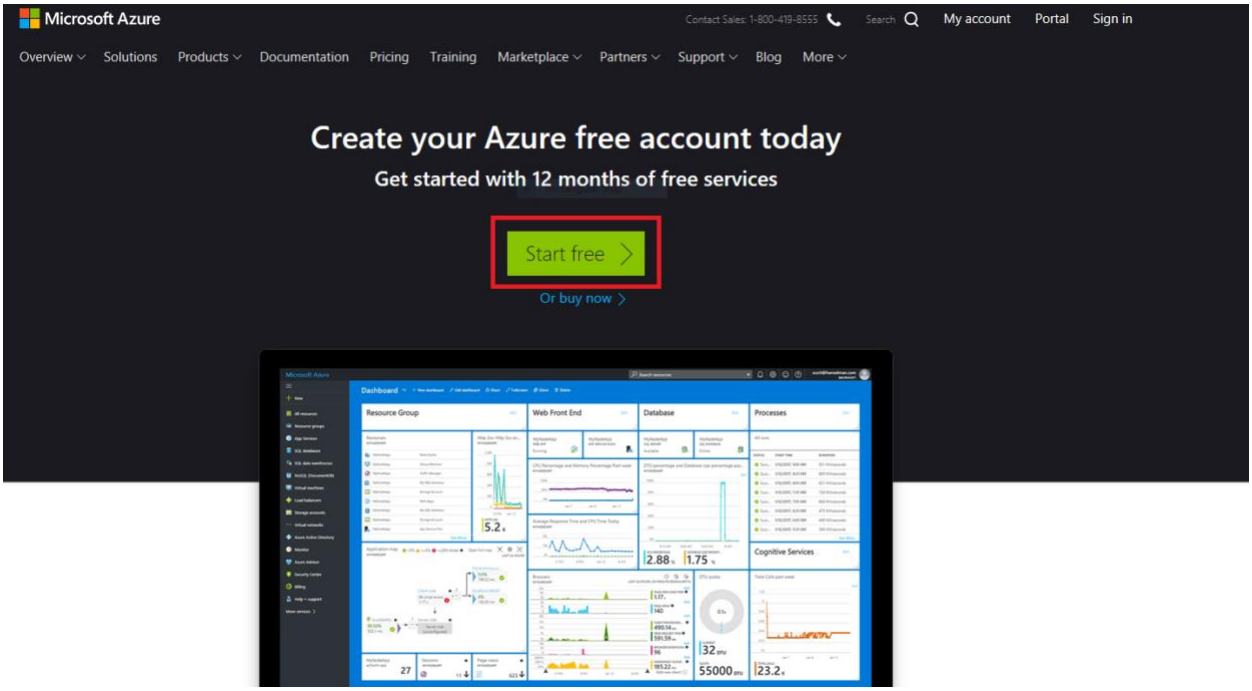
A valid Microsoft account or an Azure Active Directory OrgID ("work or school account") should exist so that you can sign into https://www.customvision.ai/ (customvision.ai) and get started.

If you don't have a Microsoft Account, you can sign-up using the following link **https://signup.live.com/**

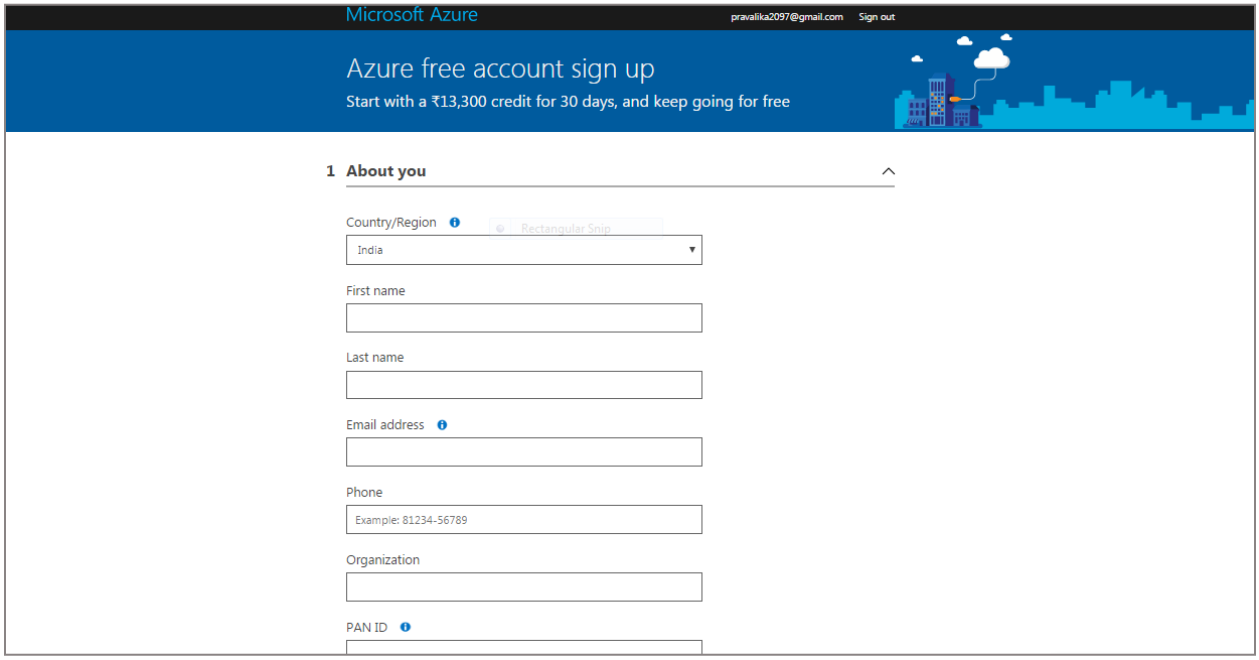
To create an account you need to have a valid phone number or an email id. Please enter this link **https://azure.microsoft.com/** and click on **free account**.



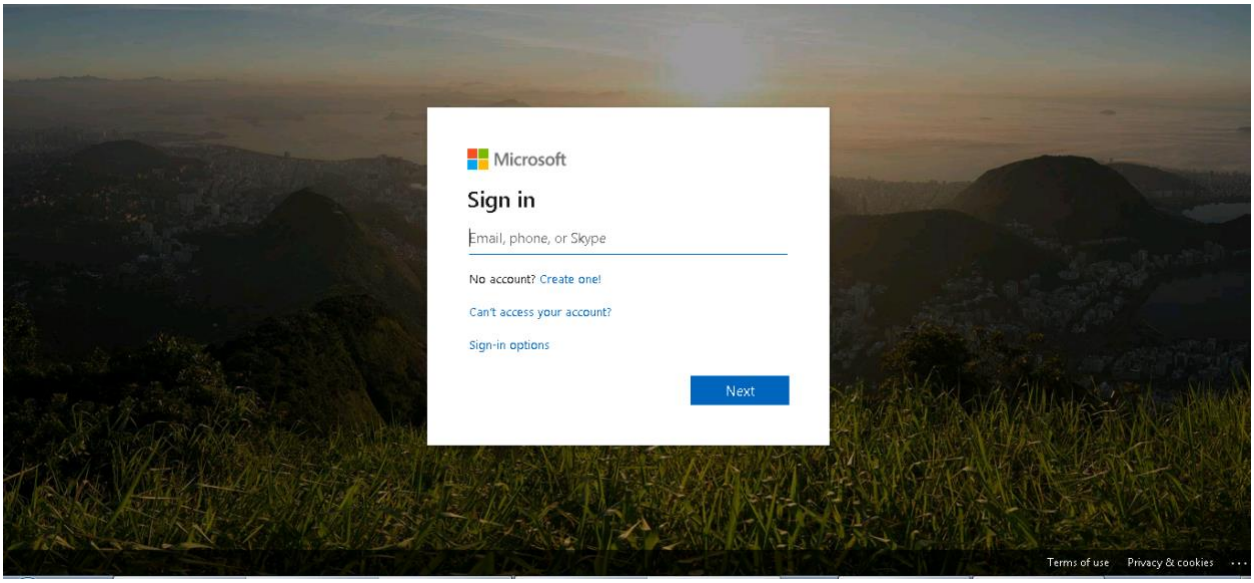
You will be redirected to the next page as shown below. Click on **Start free** option.



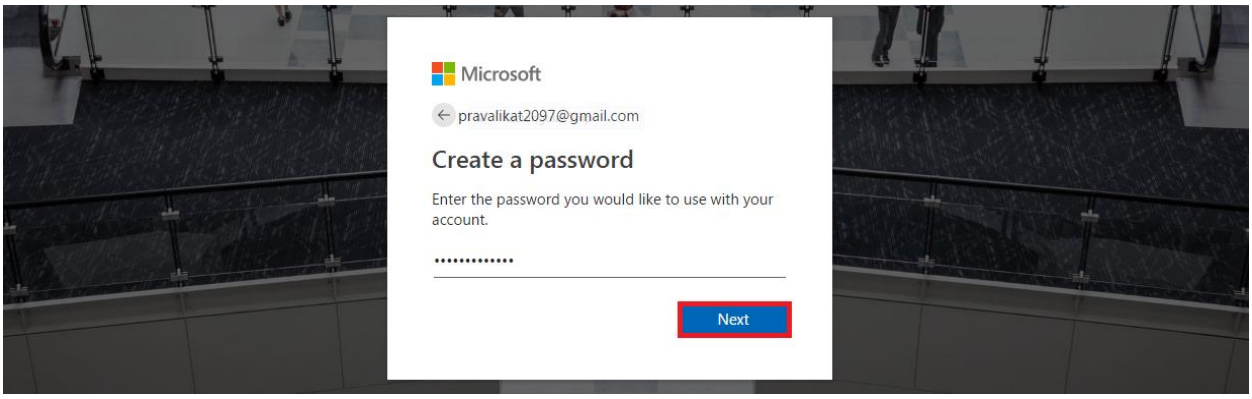
Now, you are redirected to the Azure free account sign up page as shown below. Fill in all the required details.



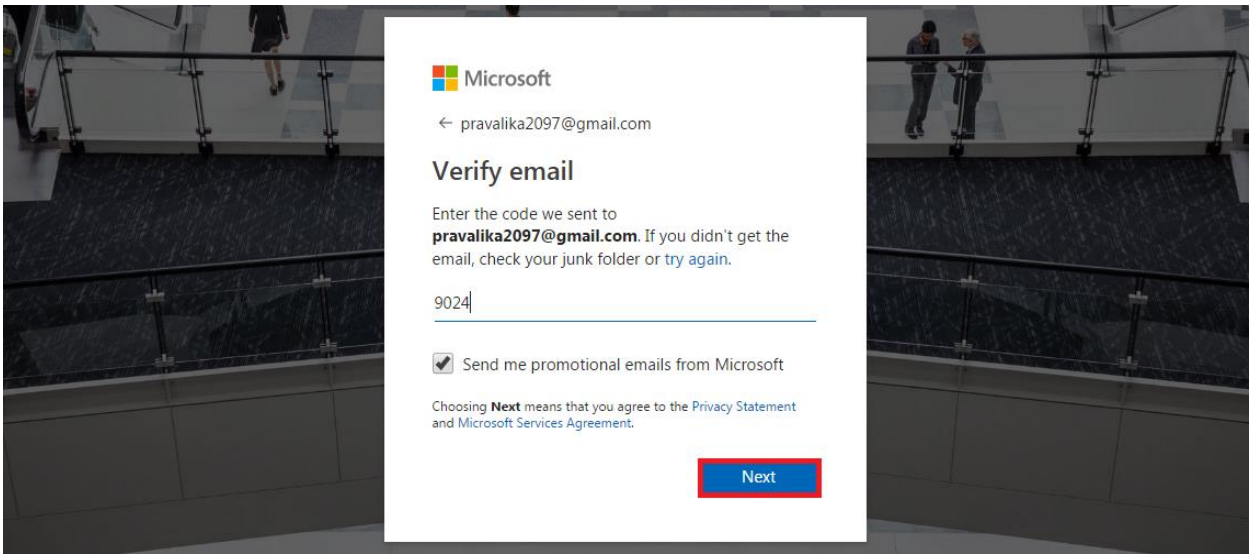
Once you have filled your details, you will be redirected to the sign in page. Enter a valid mail id, phone number or Skype ID and click on **Next** button.



Once you sign in, you will be asked to setup new password as shown in the below image.



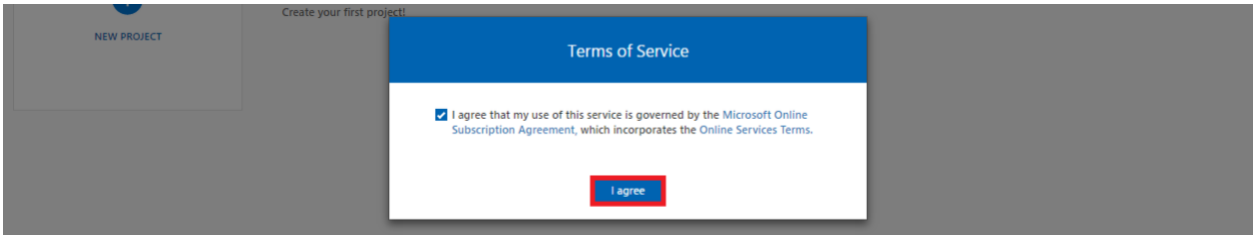
After entering the password, click on **Next** button for the verification. Now, you will receive a verification code to your registered email id. Enter the code and then click on **Next** button.



Now, you will receive a verification code to your registered email id. Enter the code and then click on **Next** button. You will be redirected to the Azure portal.

Open a new window and go to custom vision sign in page, https://www.customvision.ai/. Provide all the credentials that you have created previously and click on **Sign in.**

You will get a popup regarding **Terms of Service**, click on the check box and then click on **I agree** button.



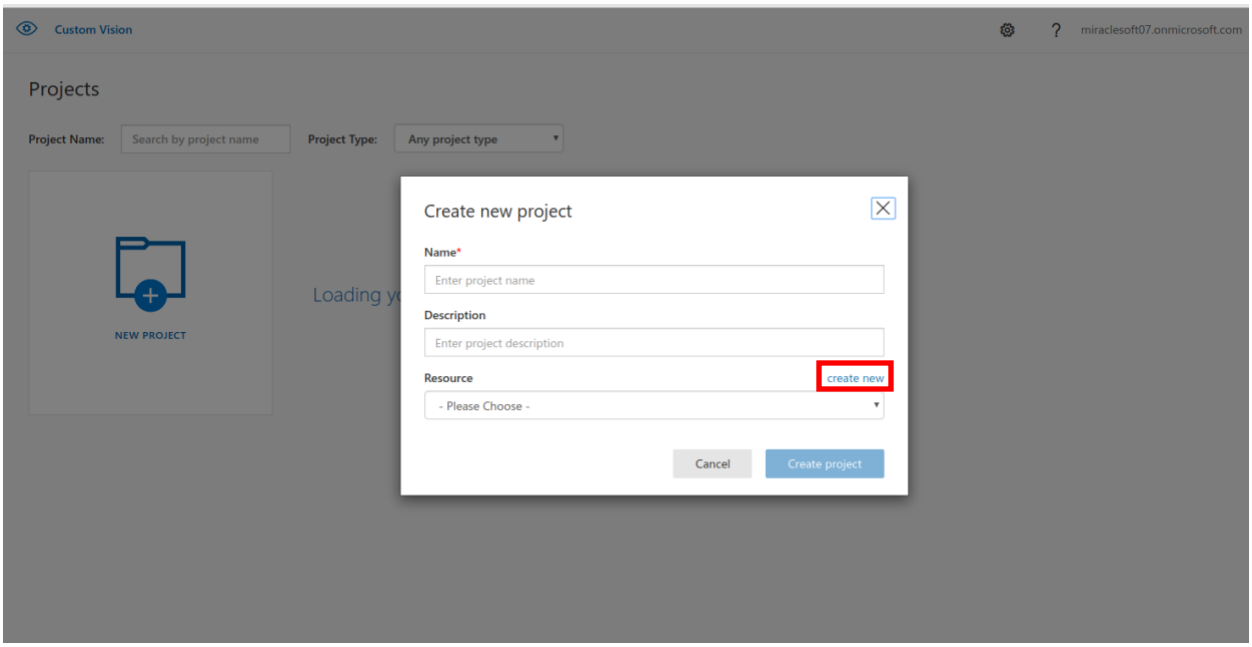
Your custom vision page will now look like this.

This is the process to sign up for **Custom vision service**.

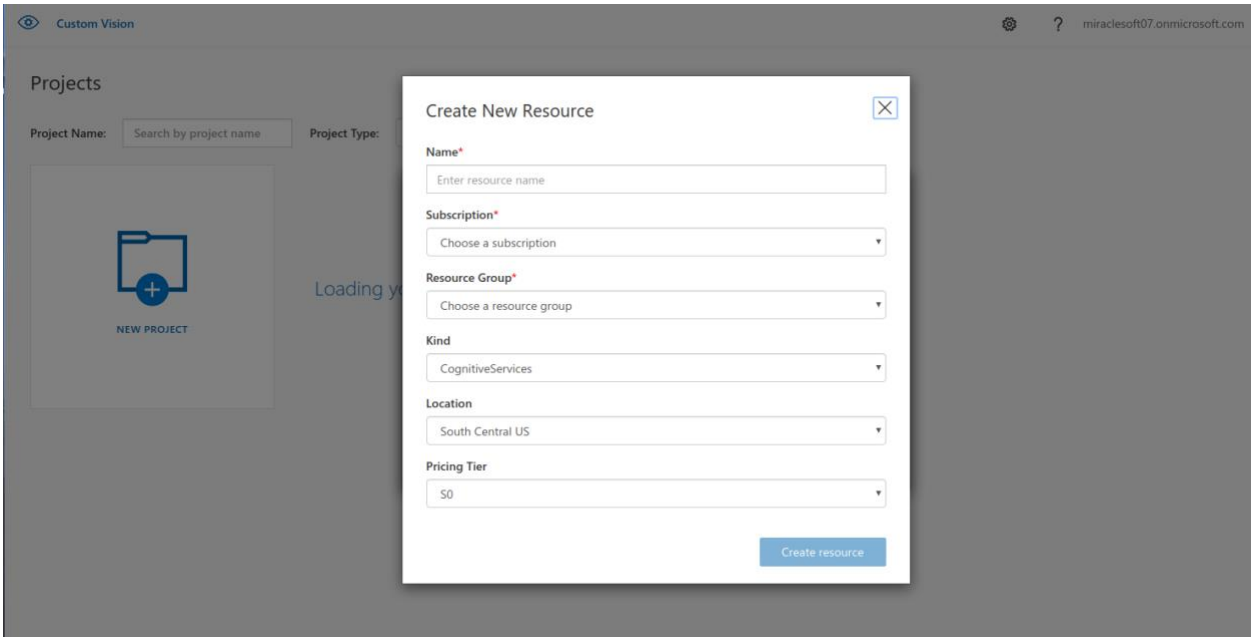
Step #2 | Build a Model in the Microsoft Custom Vision

In the custom vision page https://www.customvision.ai/projects click on **NEW PROJECT.**

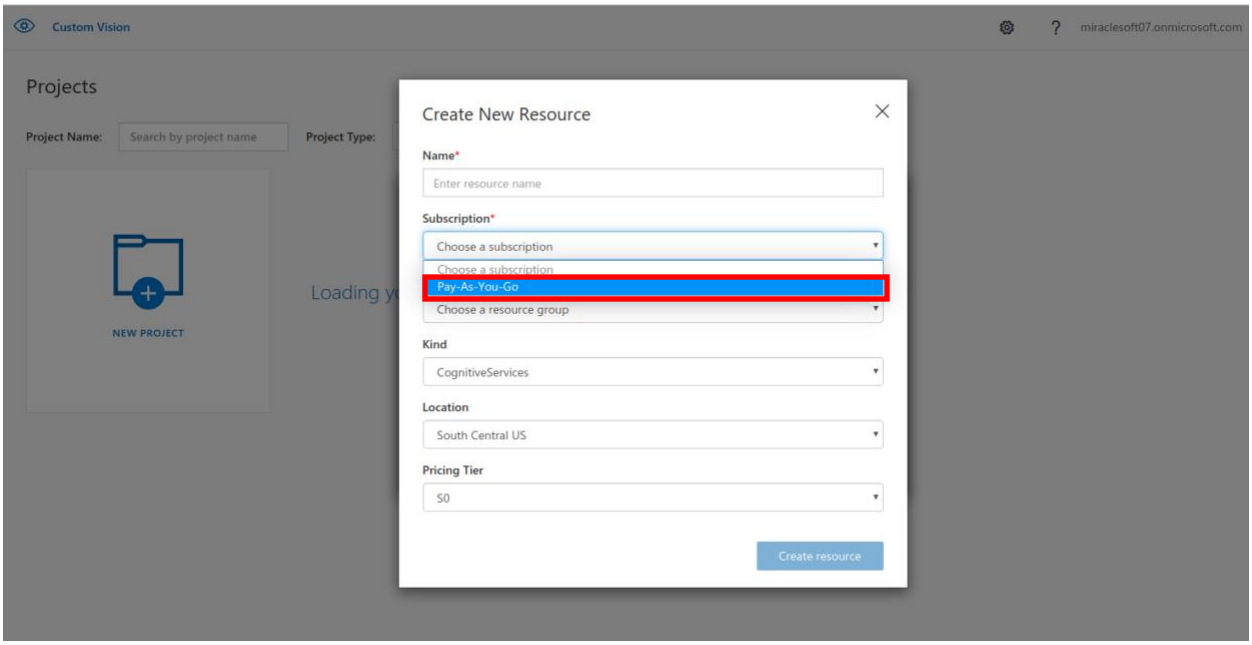
You will see a pop up for creating a new project as shown below. Fill in all the necessary details and for Resource click on **Create New.**



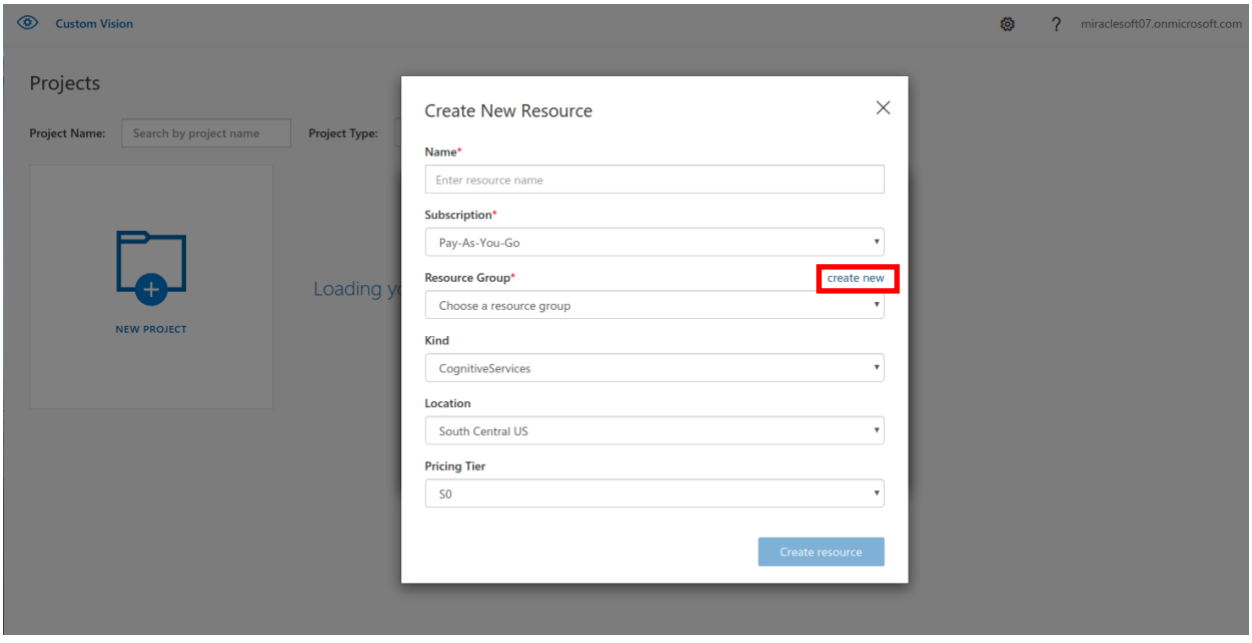
After clicking Create new you will be redirected to the below page. Fill in all the details.



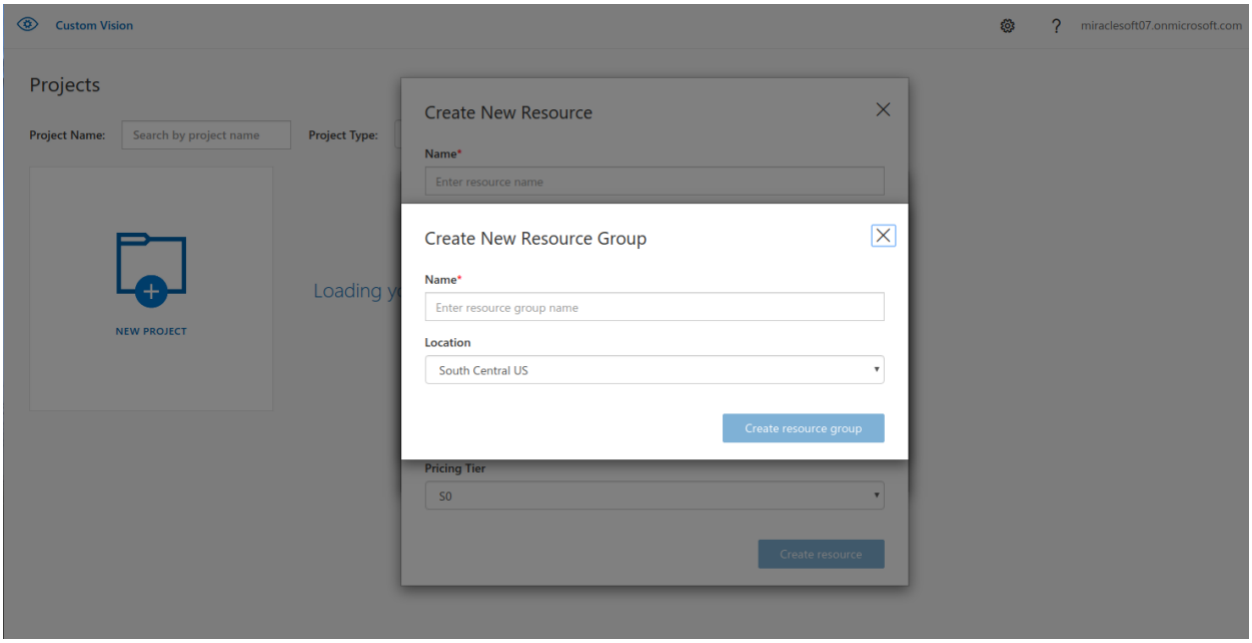
For the Subscription, choose Pay-As-You-Go subscription.



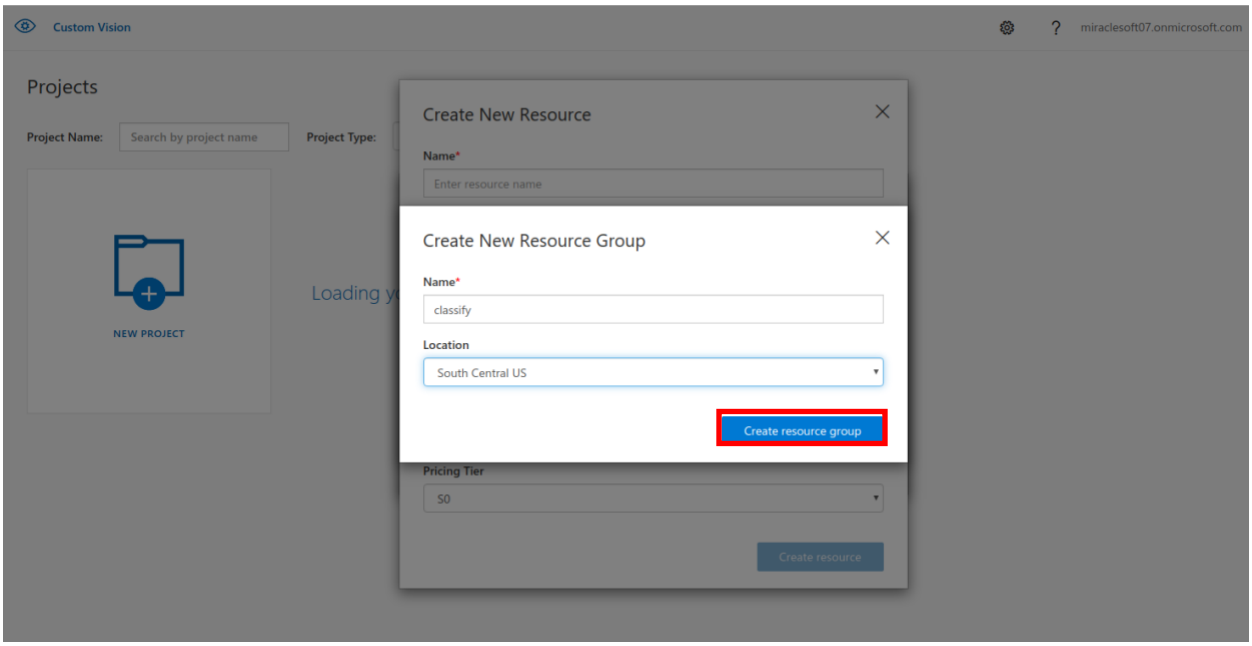
For the Resource Group, Click **Create New.**



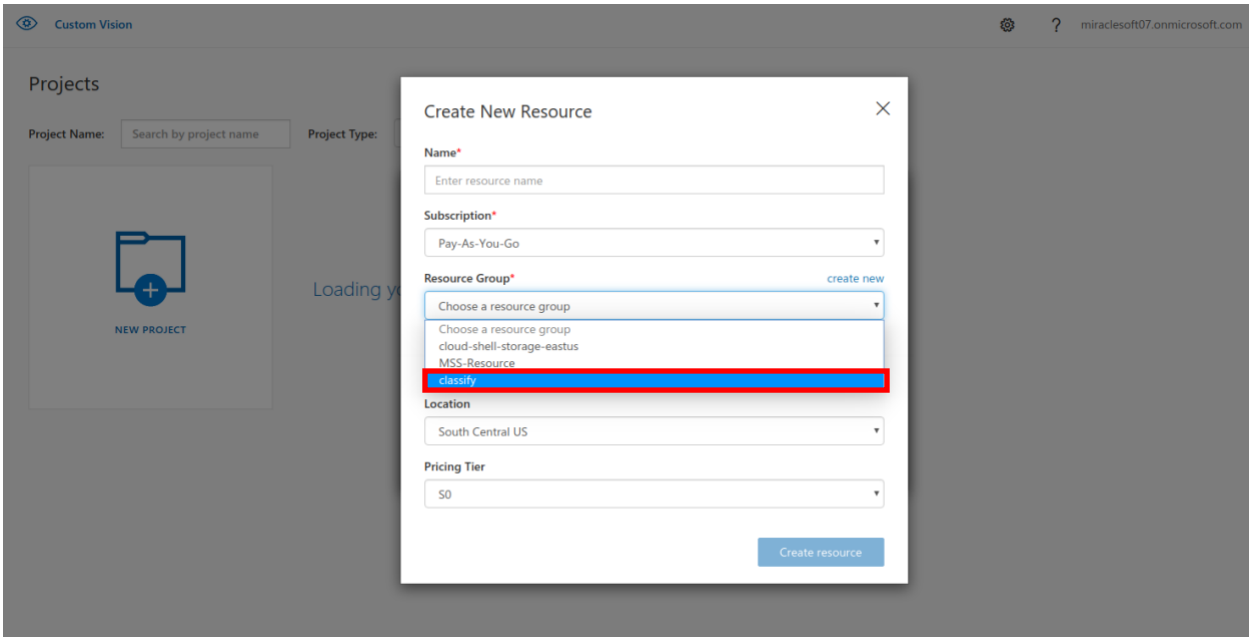
Then a new popup is displayed as shown below. Fill in all the details.



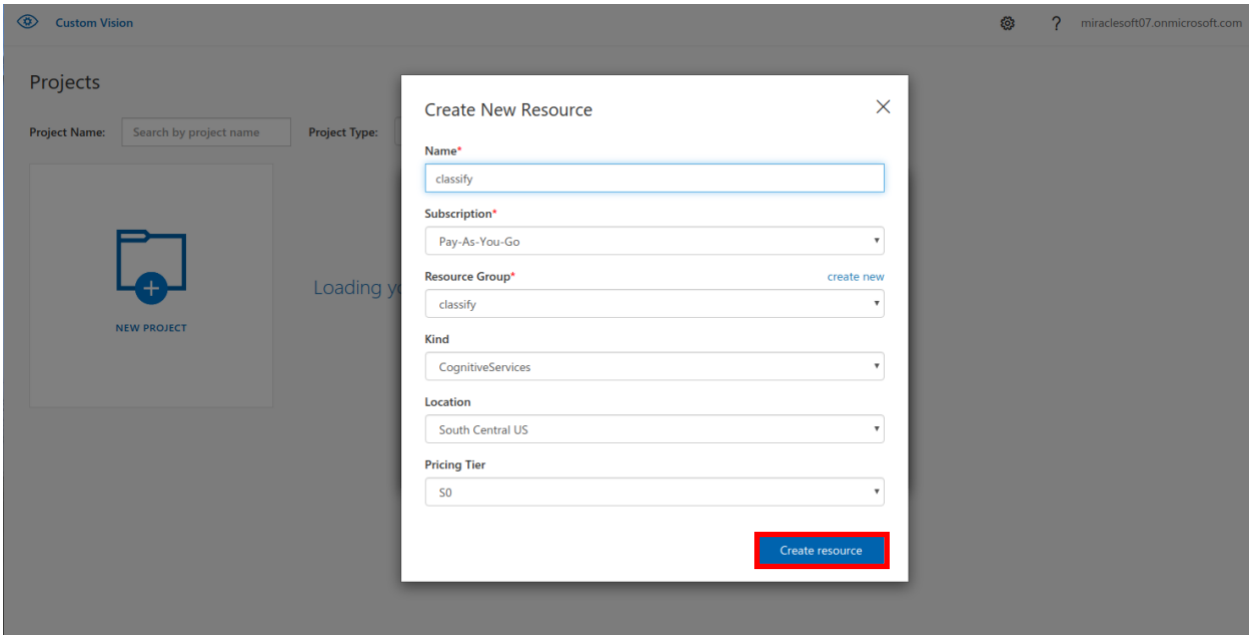
Once the details are filled, the Create resource group button will be highlighted. Click on it.



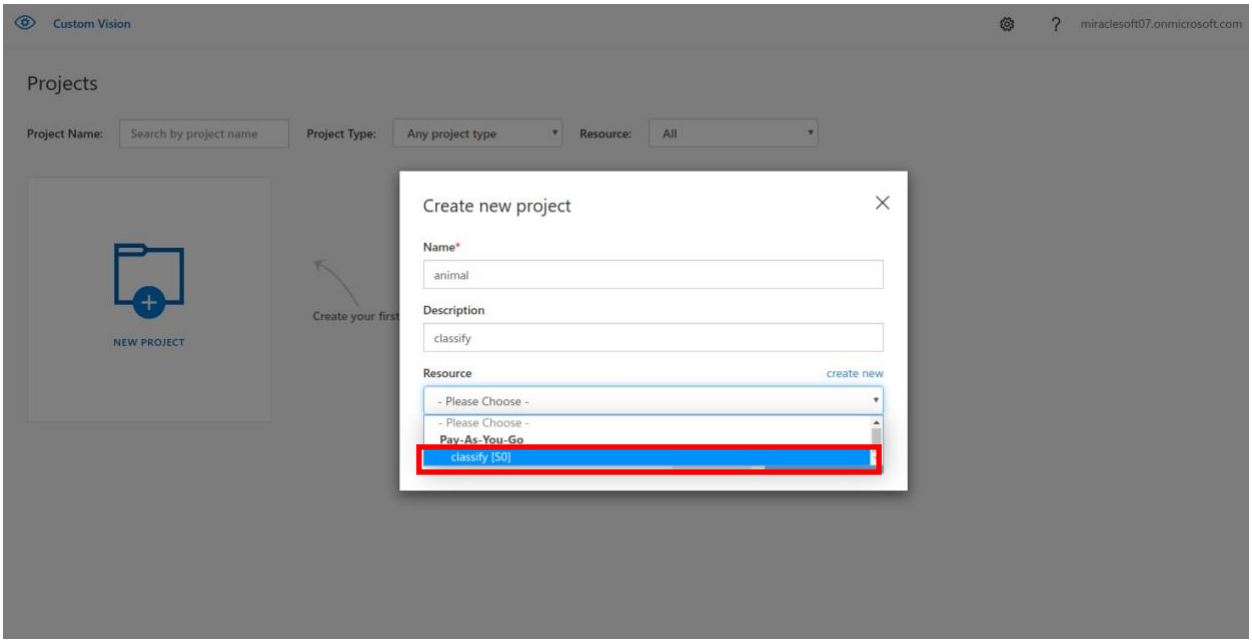
Once done, you will be redirected to the Create New Resource popup. Fill in all the details. For Resource group select the resource group that you have created earlier.



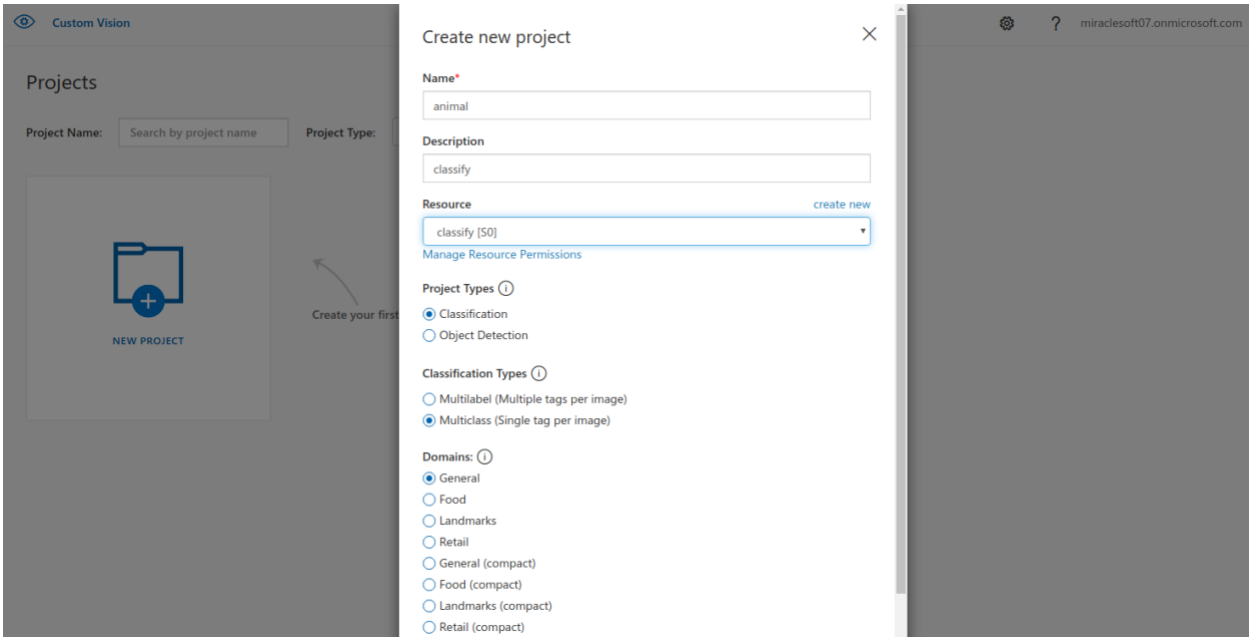
Once done, the create resource button will be highlighted. Click on it.



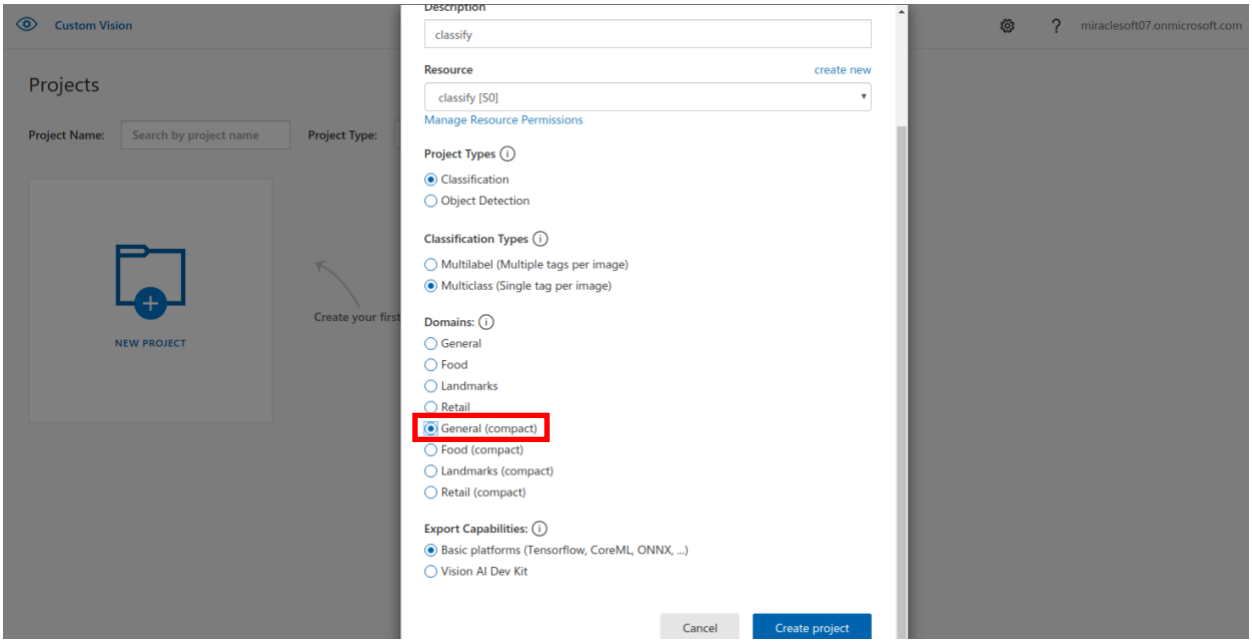
Then you will be redirected to the Create New Project popup. In the Resource, select the Resource that you have created earlier.



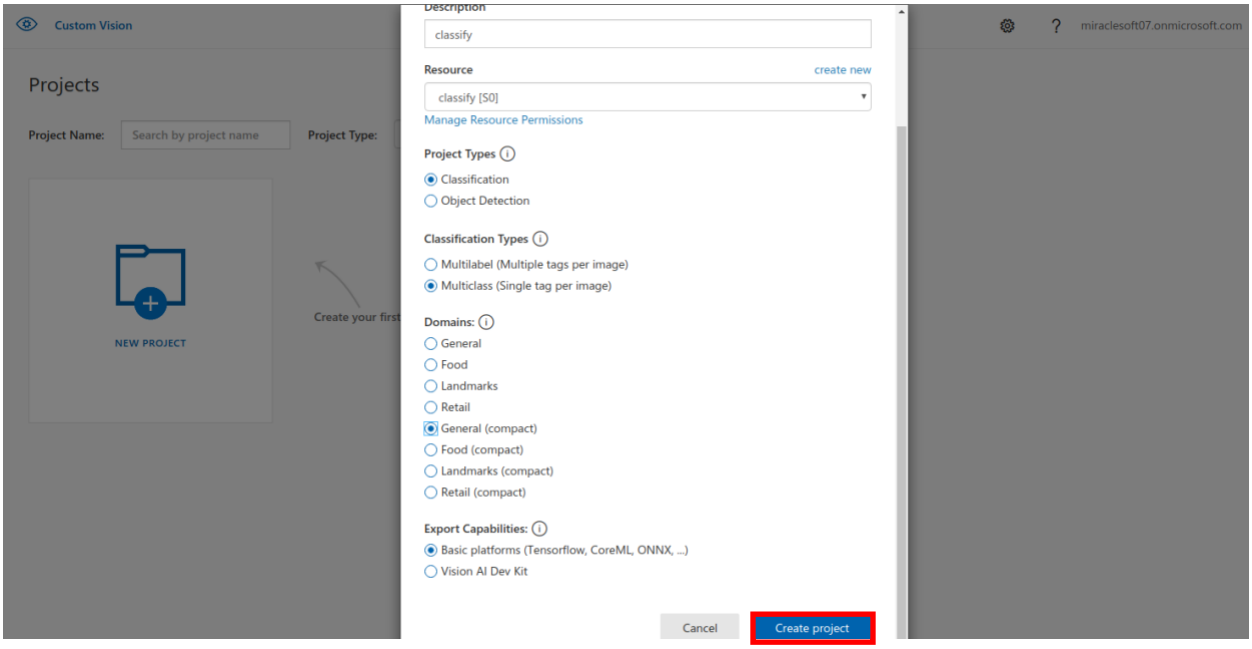
Then, you will be redirected to the next popup as shown below.



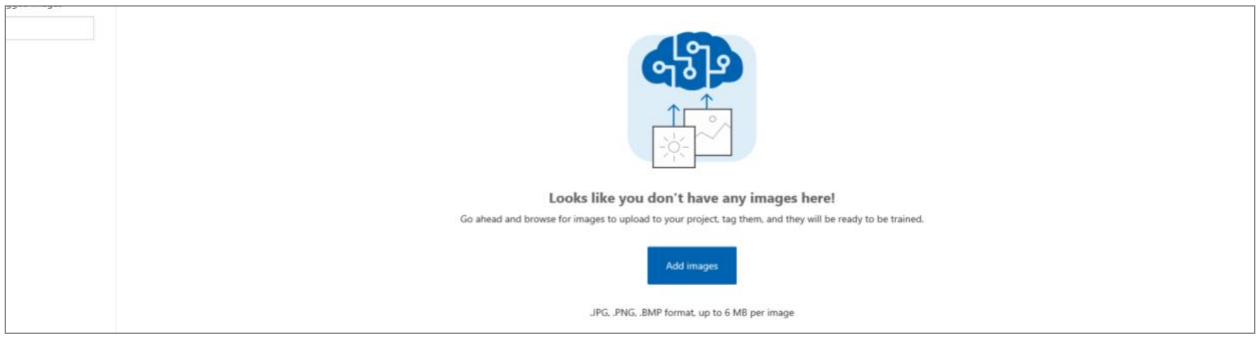
For the Domain, select General (Compact).



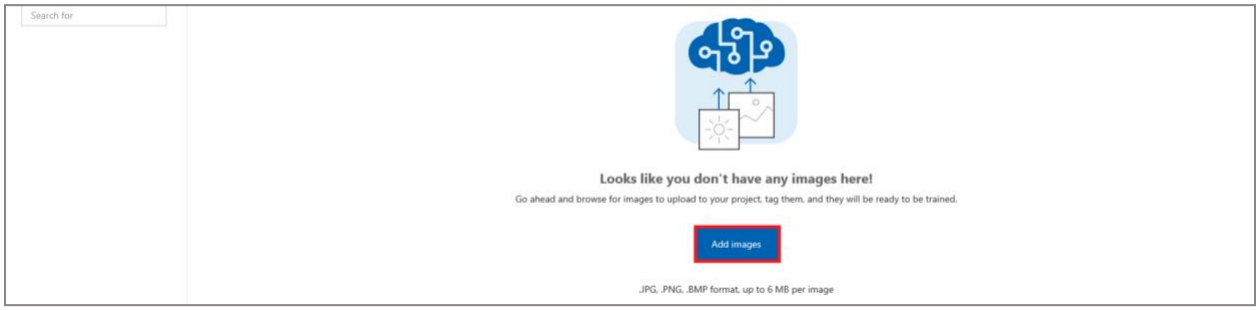
Click on **Create project** button.



Your project is created and will redirect to your project page as shown below.

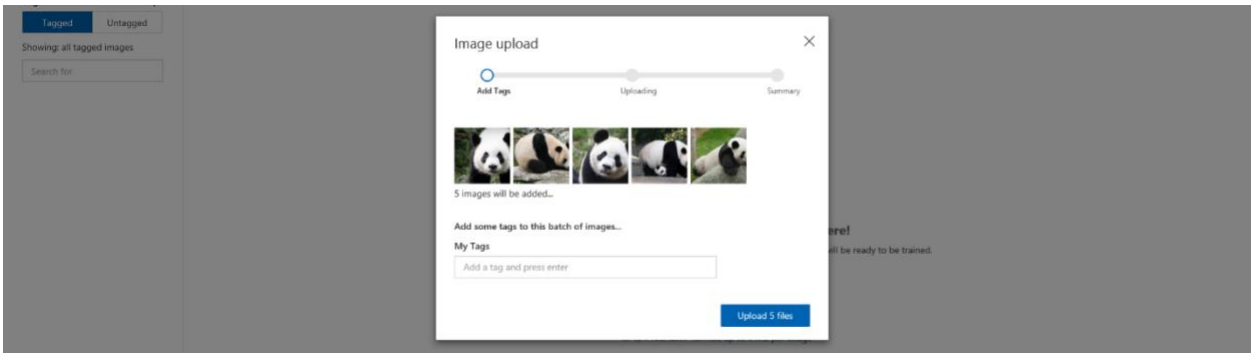


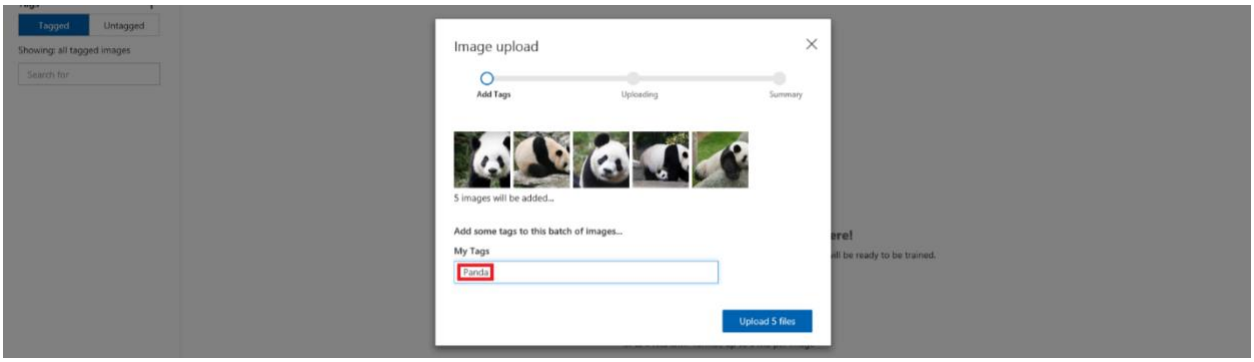
Click on **Add images** to add the dataset that is present in the folder.



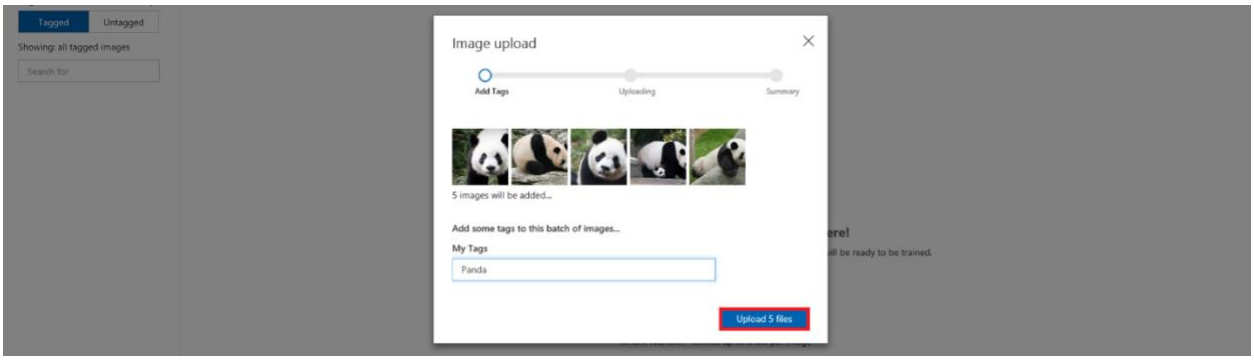
The added dataset contains 15 species and each species has 80-140 images.

Now select the images of a particular species and provide a tag name (name of the animal such as, Panda) in the text box.

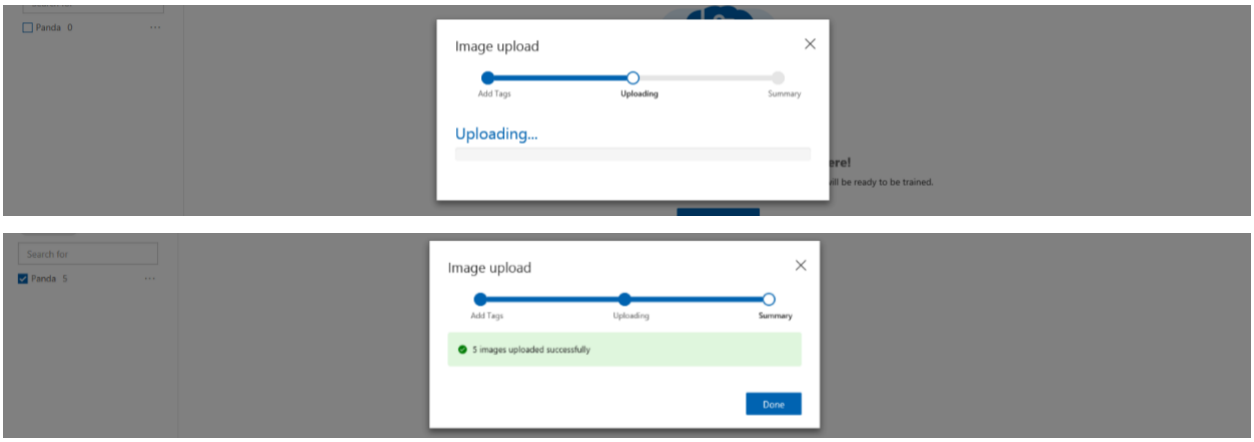




Click on **Upload files** button.

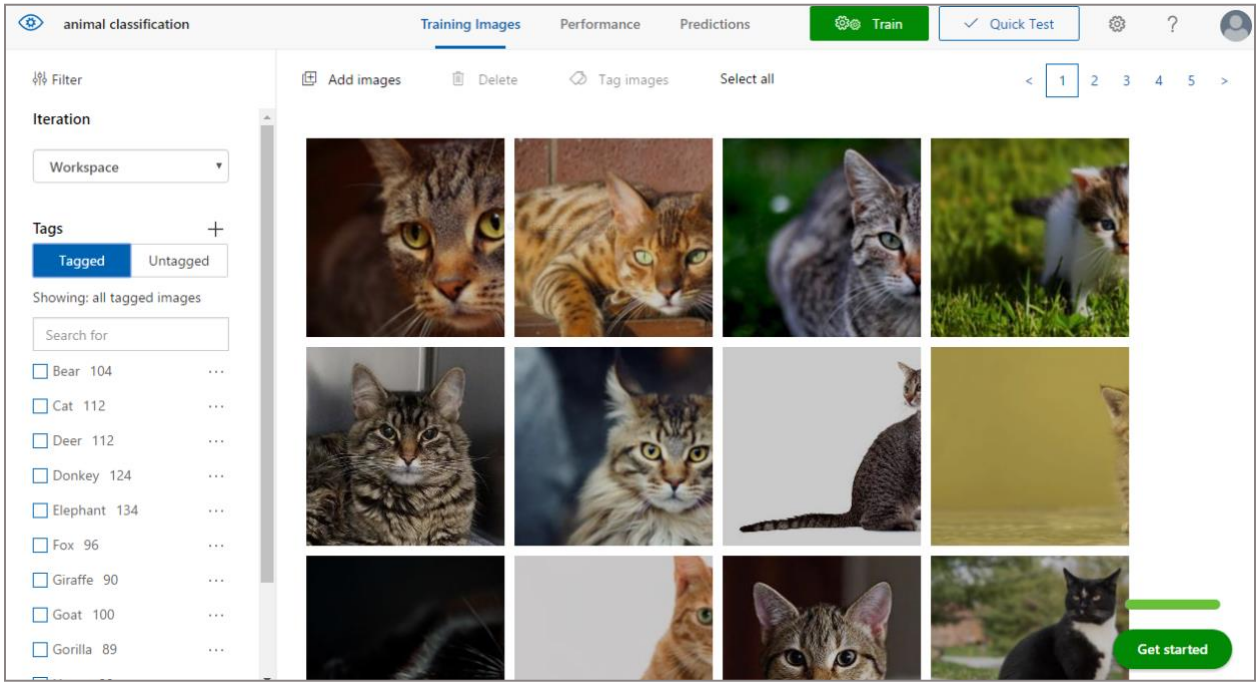


The images will be uploading now.

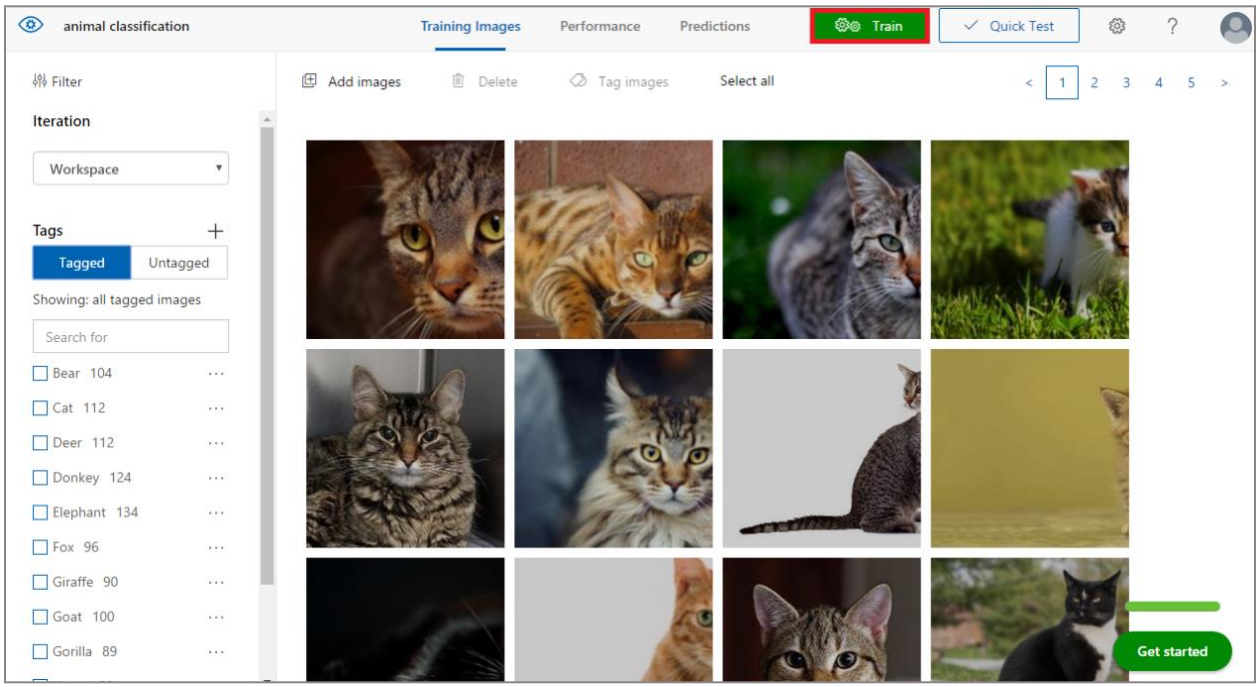


Once the images are successfully uploaded, click on **Done** button.

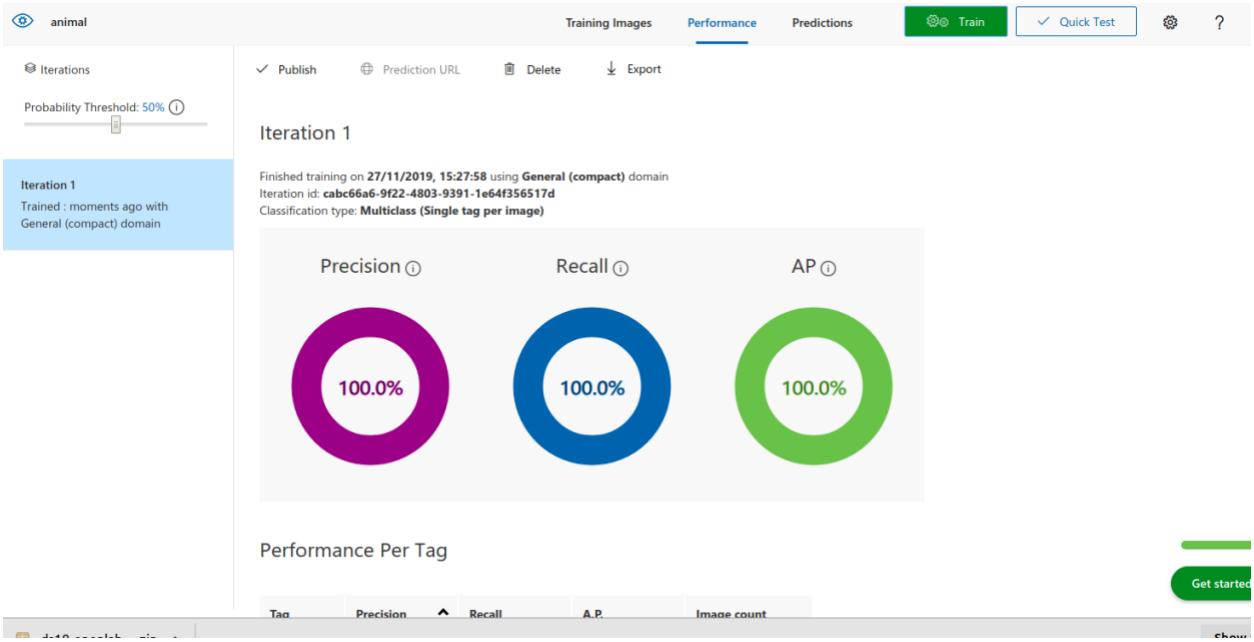
Continue this process to add all the images of the dataset. Once all the images are uploaded your project page will look as shown in the below image.



Click on **Train** button.

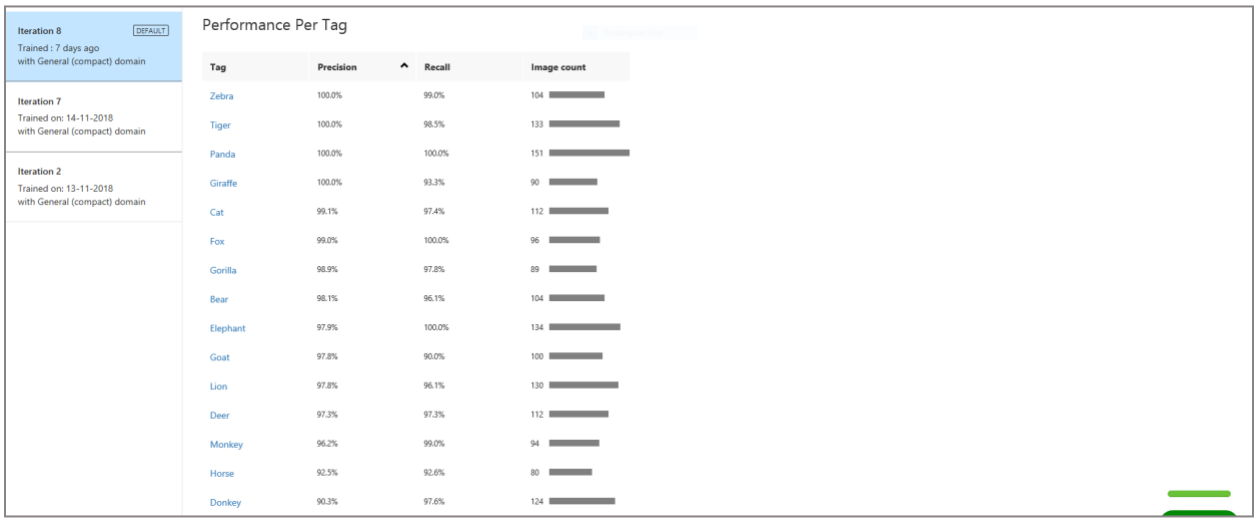


The model will now be trained and can be viewed in **Performance** page.



There will be 10 iterations available for one free account. Your iterations will be trained in the **General Compact Domain.** To change the domain, click on the settings icon.

You can even view the performance for each tag.



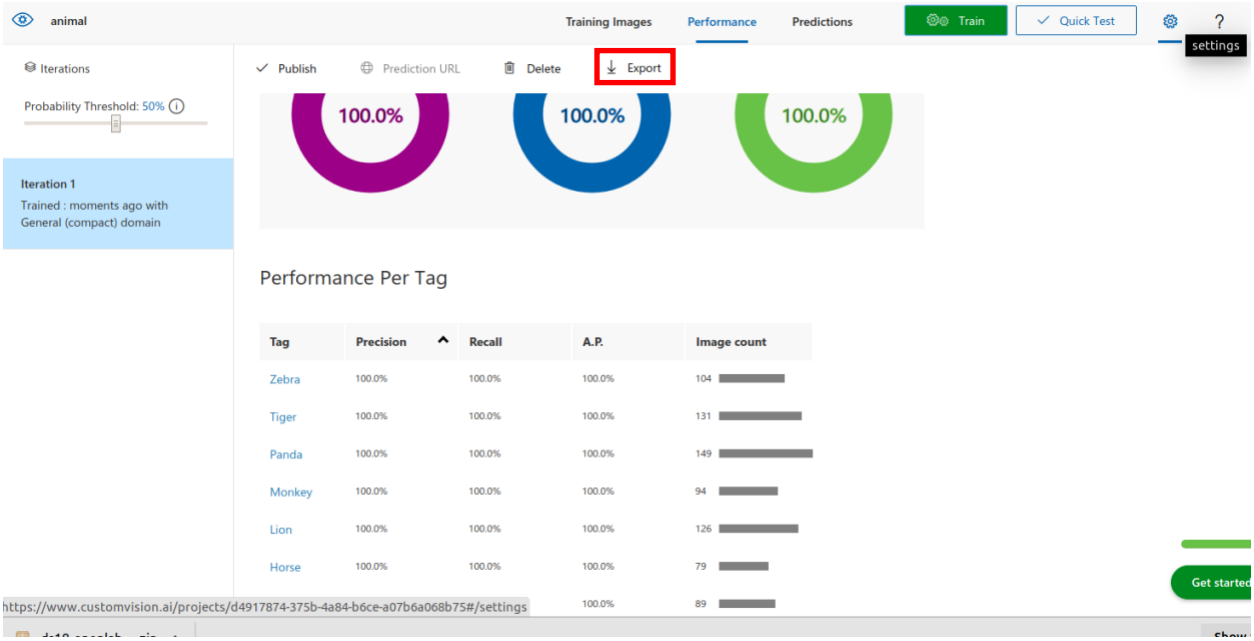
If the Precision and Recall values are above **95%**, you need not train the model again. If it is less than **95%,** then you have to retrain the model.

You can also view the performance for each tag, where you can view **Precision, Recall, A.P.** and **Image count** for each tag. If the values Precision and Recall are above **95%** you need not change otherwise you have to improve your dataset. The Image count for every tag must be uniform.

If all the values are satisfied, the Model is now ready!!

Step #3 | Export the Model

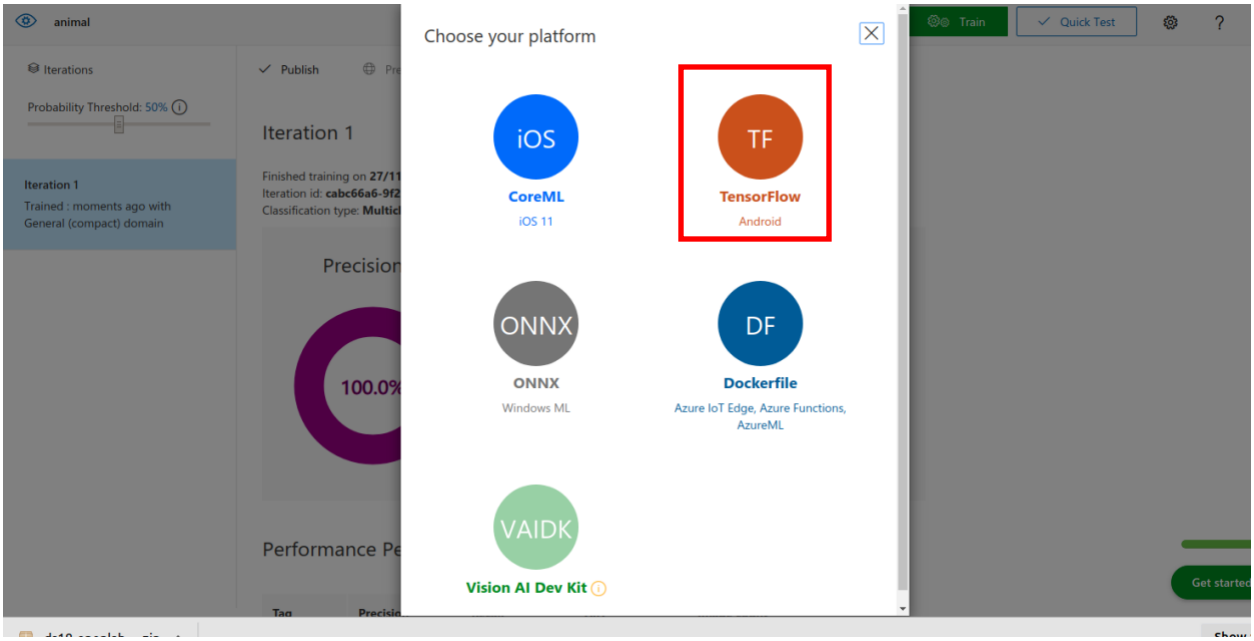
The model is now ready to export. To export the model, click on **Export** button on the top.



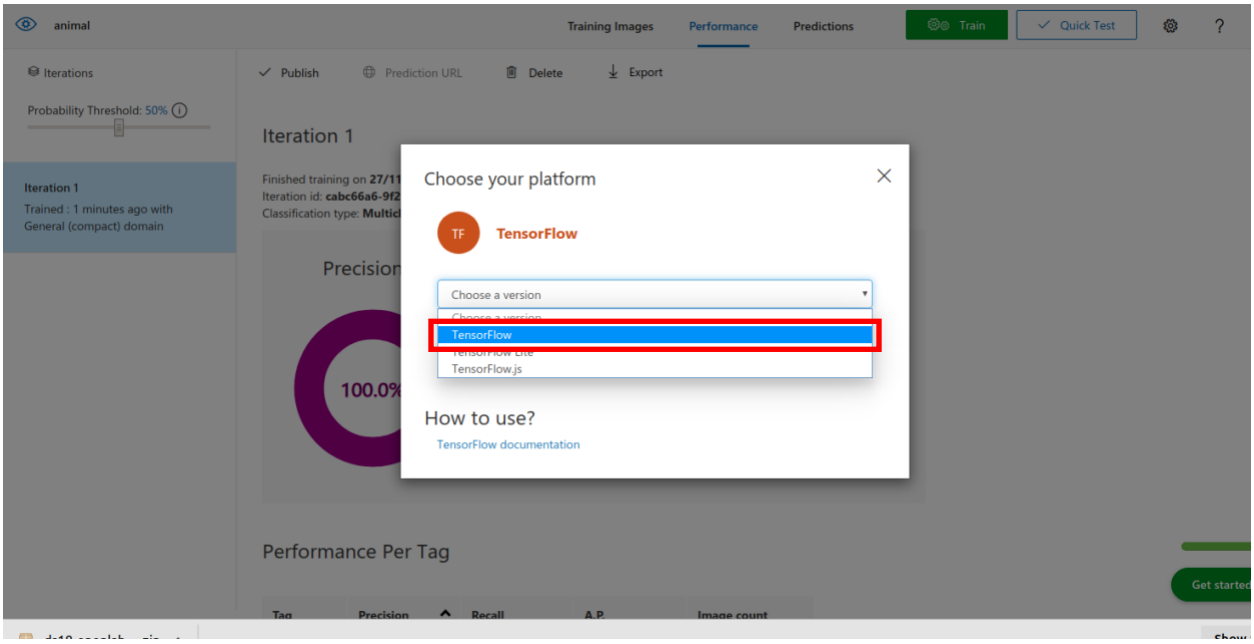
You can now view various platforms to export like,

- Android (TensorFlow)
- iOS (CoreML)
- ONMX (Windows)
- Docker file for Azure to Edge, Azure functions, AzureML

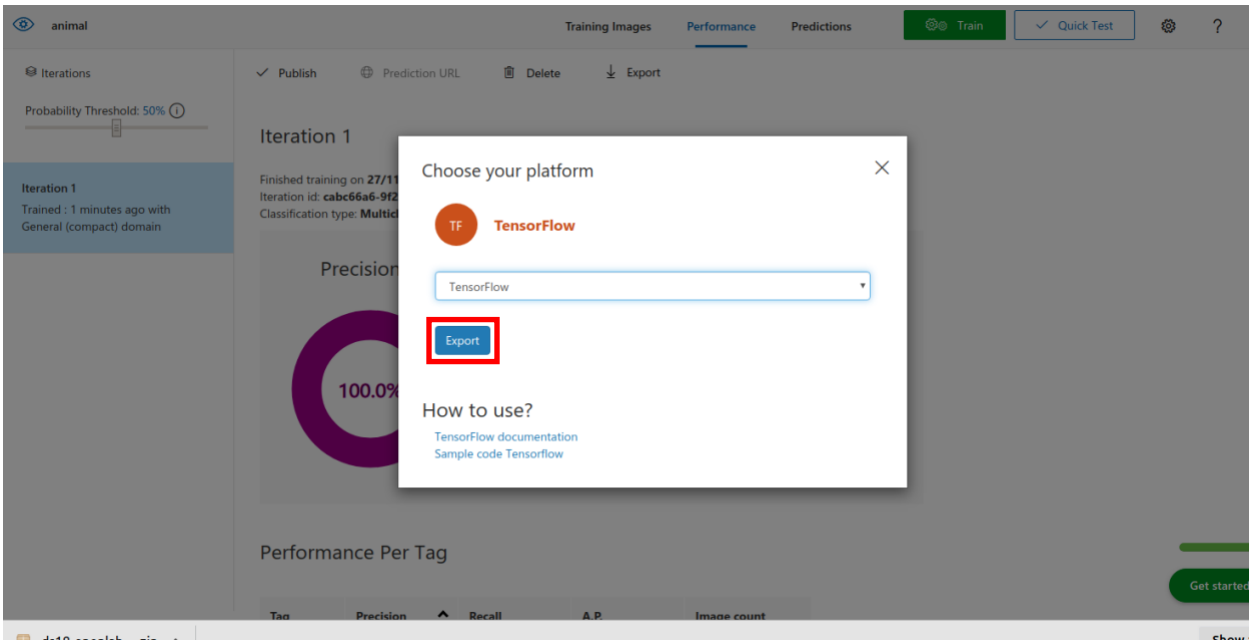
We need TensorFlow model so, click on **TensorFlow.**



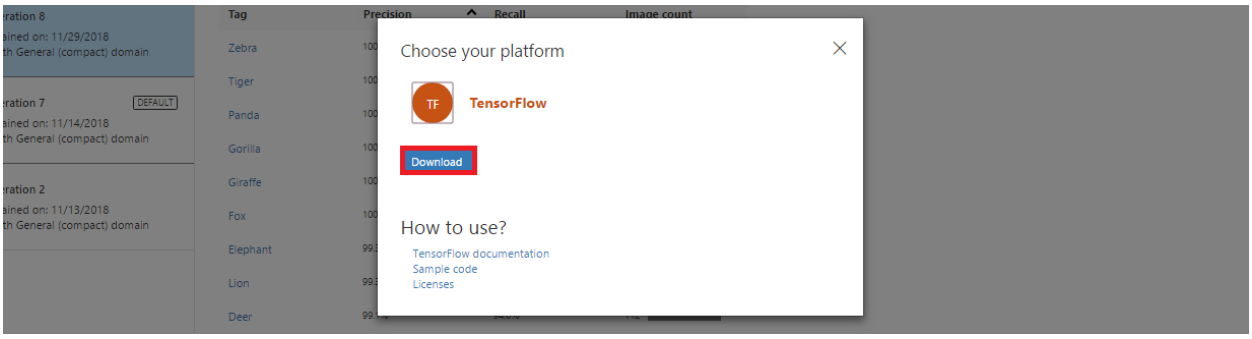
In the Tensorflow model, click on TensorFlow.



Then, click on **Export**.



The following pop up will be displayed. Click on **Download** button as shown below.



The exported model is now downloaded. You can view the model in your local file system in downloads folder. Unzip to use it.

Step #4 | Integrating the Model with Web Application

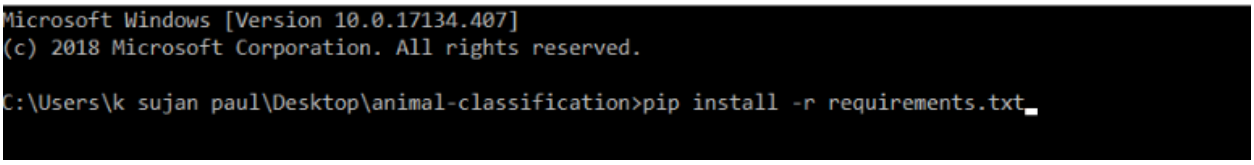
Now we have the model ready with us. We have to integrate the model with the UI for a proper web application to work. For integration we have to follow few steps.

In the main folder, we will have folders such as, templates and static: Files like model.pb, labels.txt, requirements.txt and app.py.

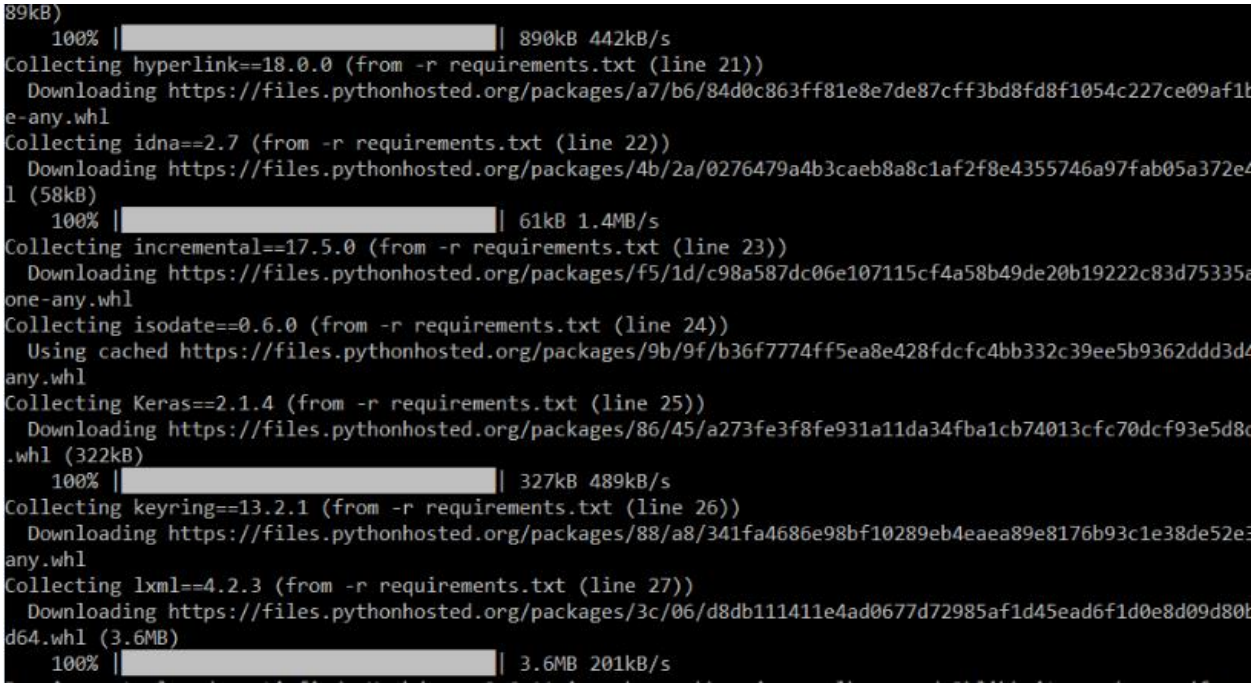
- Model.pb and labels.txt is the model file which we have exported

- Templates folder consists of all the UI code and static folder consists of style sheets
- **Requirements.txt** is the file which has all the dependencies that we have to install

Go to the project directory and open command prompt. For all the dependencies to install just provide the command as, **pip install -r requirements.txt**



Click on **ENTER**.



All the dependencies are installed successfully.

app.py is the file which contains all the integration code. It is basically a python server which is written using python flask. We have to run this in order to operate the web application. In the app.py, we will initialize the model.pb and labels.txt files as follows,

#initializing the model and labels

filename = "./model.pb"

labels_filename = "./labels.txt"

This is present in line number 22

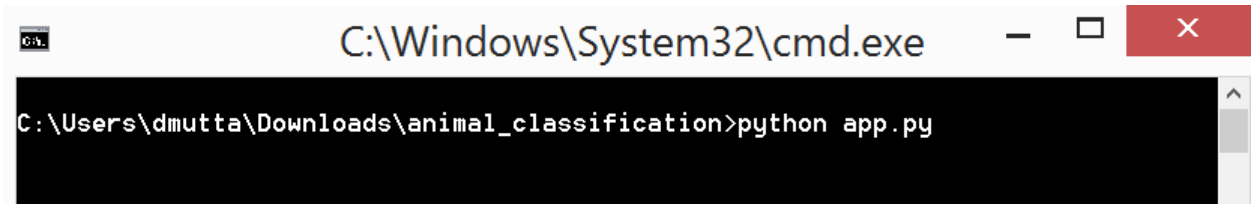
For integrating the only thing we have to do is replace the model.pb and labels.txt present in the folder with the model.pb and labels.txt files that are obtained on exporting your own model.

So, now the integration is done and you can run your application.

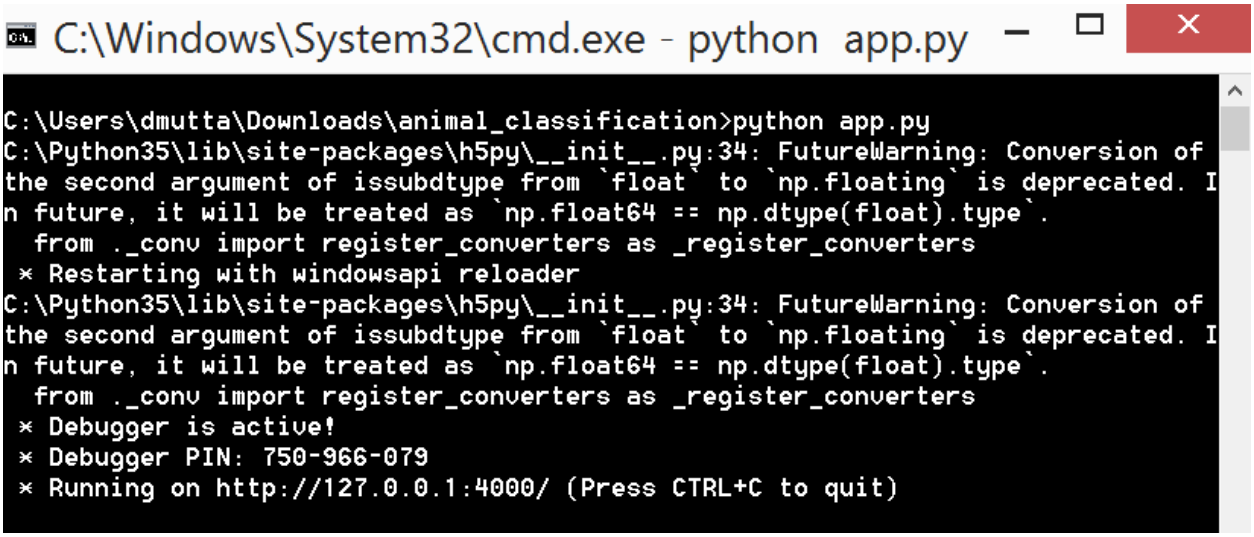
Step #5 | Running the Web Application

Go to **Command Prompt**, provide the command as,

python3 app.py



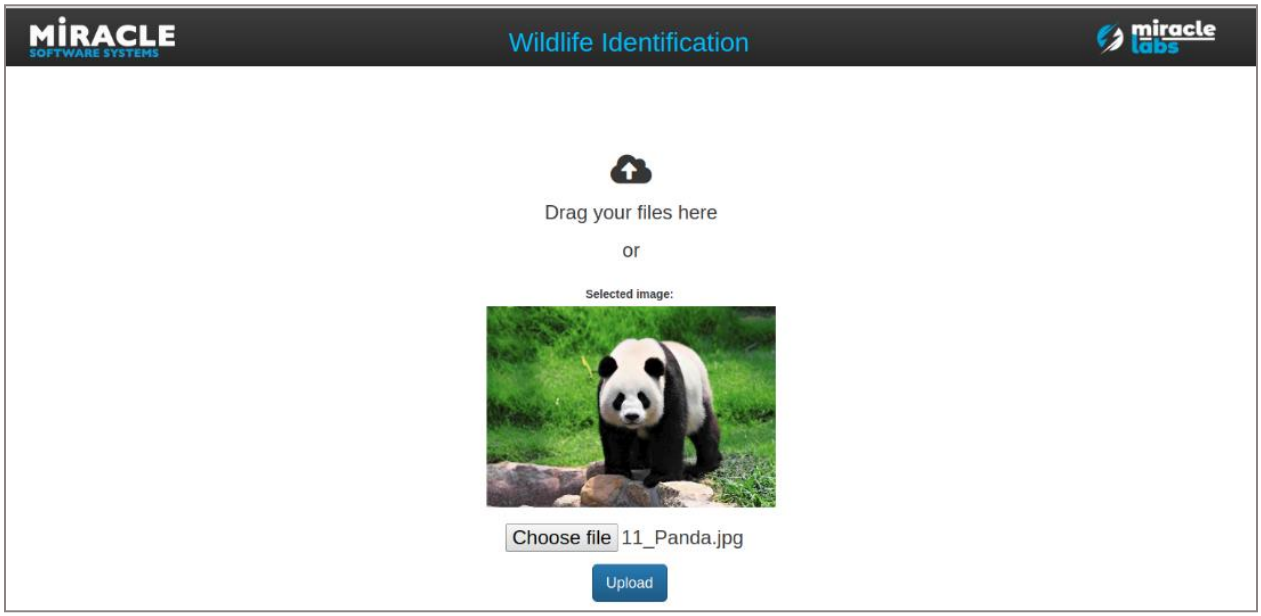
After giving the command click on **ENTER**.



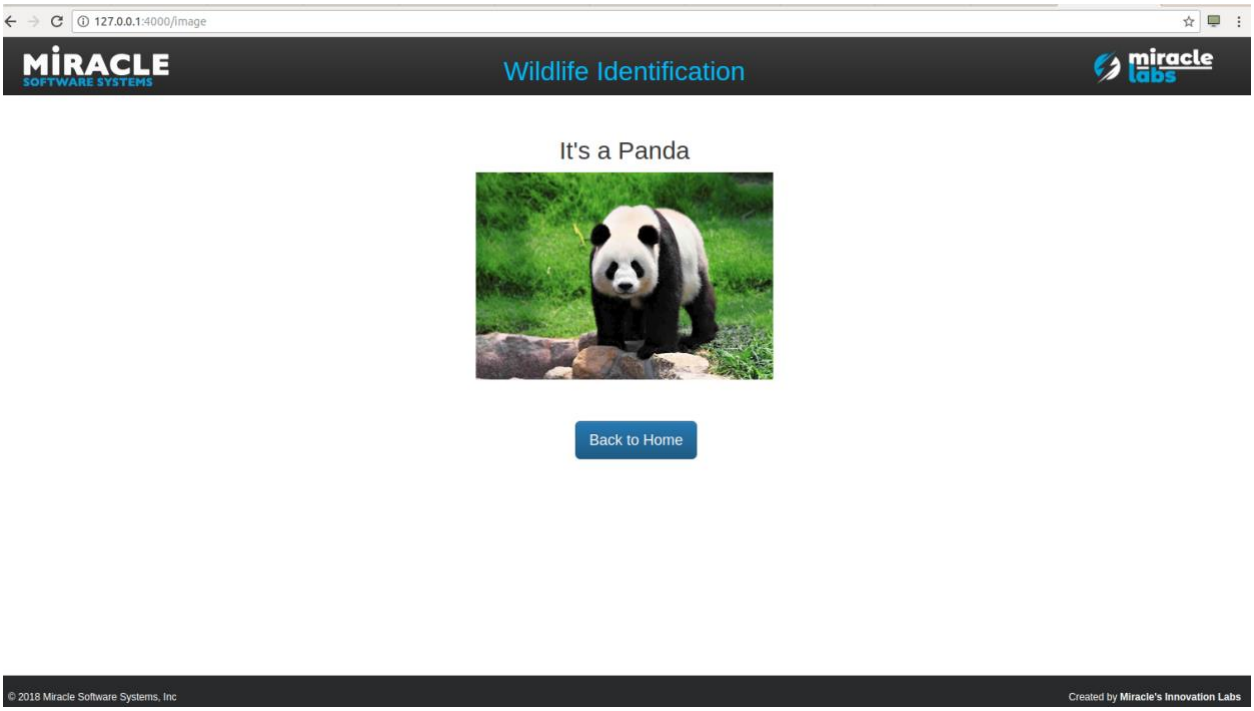
The server is now running on **https://127.0.0.1:4000/**

Now open the link in any browser and you can see your web application. The basic web application will look as shown below.

Click on the **Choose file** button to select any image of an animal.



After giving the image of any animal, click on **Upload** button and you can now view the result in the **Result Page** as Panda.



If you want to try with another image, click on **Back to Home** and repeat the same process.

Hurrah!! With this lab you were able to build a model with **Azure Custom Vision** and integrate with Web Application.

For any questions regarding the lab please feel free to reach out to innovation@miraclesoft.com. **We hope you enjoyed creating Machine Learning models with us.**