Design of Plant Leaf Recognition Architecture Using Python with Azure Storage and Azure Databricks

1. Problem Statement

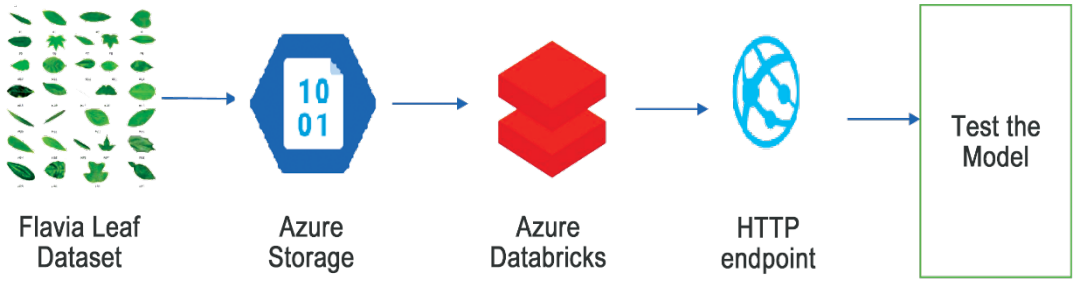
- Plants are very important for the existence of human life. The total number of identified plant species is nearing four hundred thousand as of date The current work aims at designing a **deep learning model** for the **multiclass classification** of plant leaf images using **python** and **Azure services**. The major focus of this work is to automatically classify and predict the plant species of the plant leaf images of **Flavia leaf dataset**. The first step in getting started with building a Plant Leaf Recognition(PLR) model is to have a leaf image dataset. Leaf Image Dataset is required for training and thus creating machine learning models or CNN models for PLR. The Flavia leaf dataset used in this work consists of 1907 leaf images generated from 32 China plant species.

2. Architecture

To get started with the PLR exercise, image acquisition is the first step. A properly labeled set of plant leaf images is needed for the purpose of training and validating the PLR model. For PLR model creation, machine learning techniques or deep learning techniques may be used. The model is trained using the leaf images. On completion of the model training, the model validation is done using the validation data set. The model is used to predict the leaf names of images in the validation data set. The predicted leaf names are compared with the actual leaf names of the validation data set. This information is used to assess the performance of the model. To implement the specified methodology Azure services are used.

Azure Services	Description
Azure Storage	For storage of the Flavia leaf image dataset
Azure Databricks	For training of the classification model implemented with Keras and TensorFlow
Azure ML Service	For versioning and deployment of the classification model as HTTP endpoint

The architecture for the above methodology is shown in the below figure

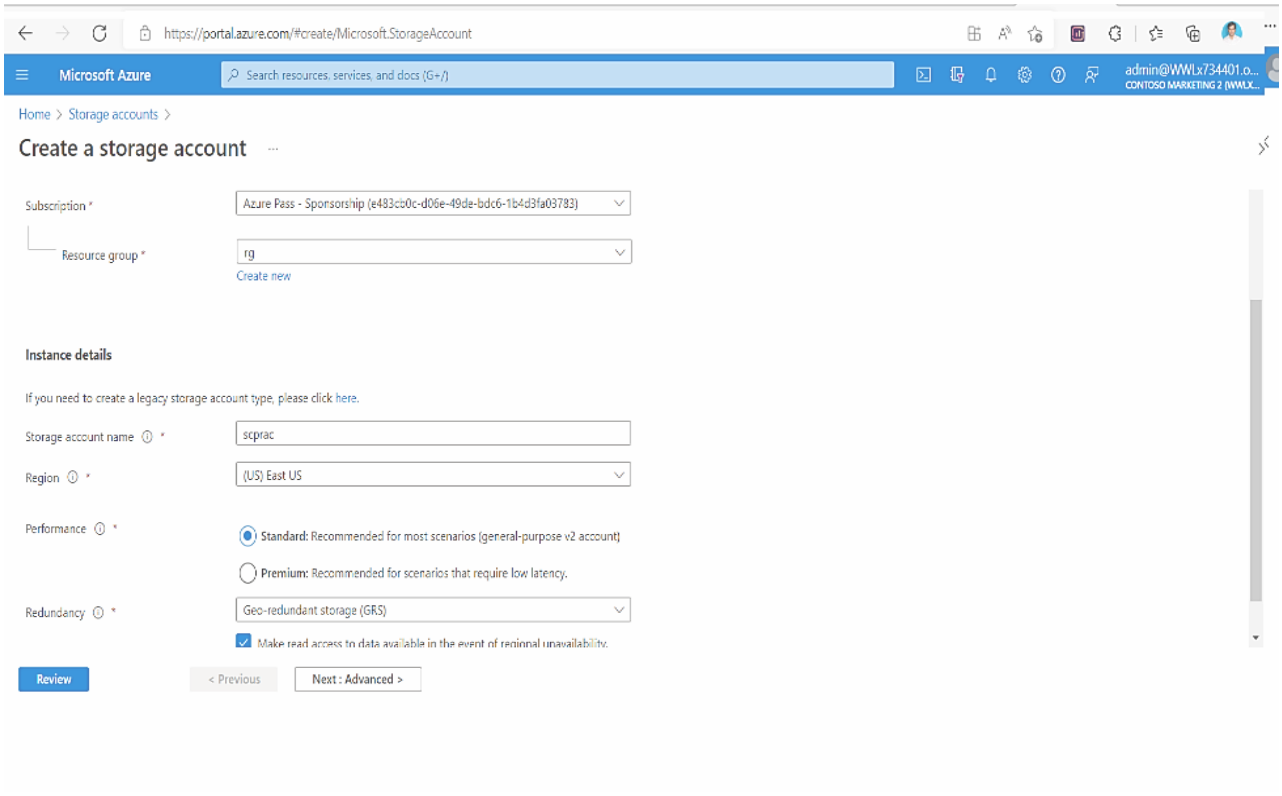


The Azure Databricks which is an Apache Spark-based analytics platform is the key solution for the given architecture. It integrates with Azure services such as Azure storage and provides efficient analytic solutions for end user.

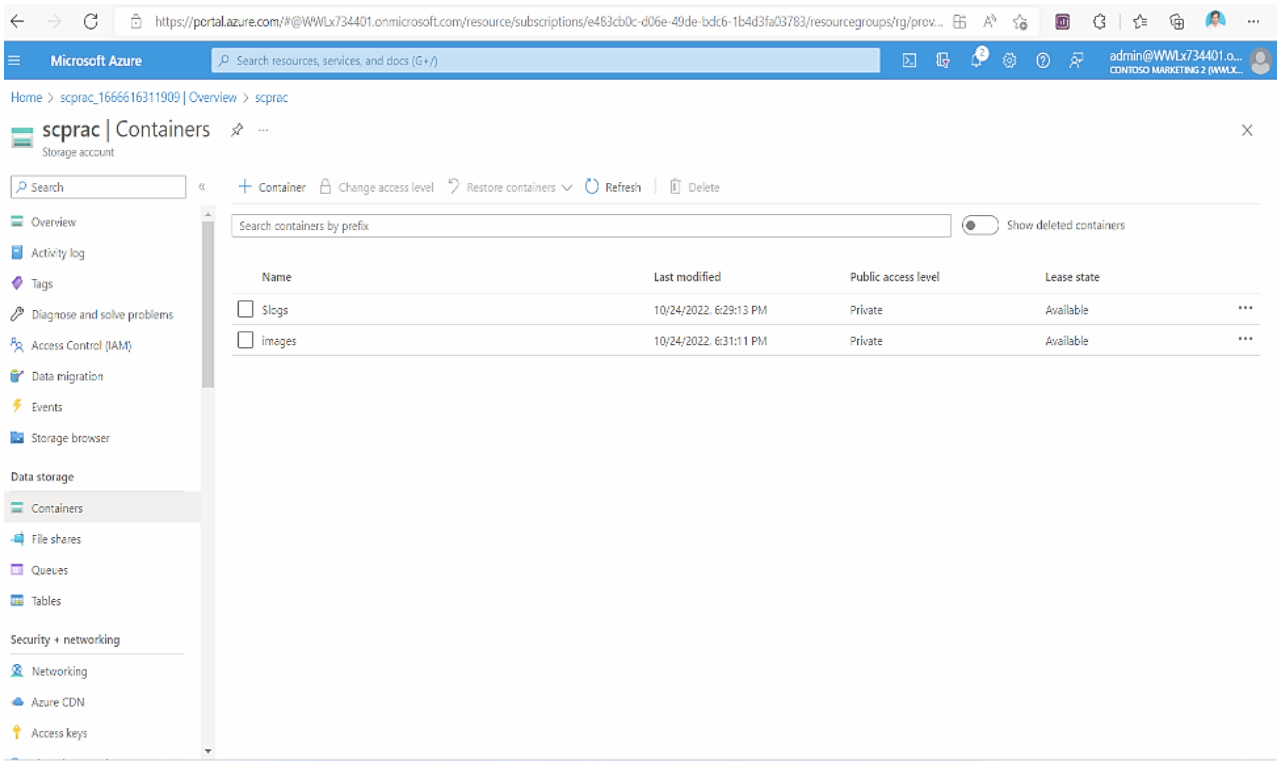
3. Technical Details and Implementation of Solution

The first and foremost requirement to be fulfilled in this work is to add images to a storage account. The images will be required to train the deep learning model. All the images are downloaded from the URL https://flavia.sourceforge.net/. The downloaded images are brought into the Azure environment by storing them in the storage account of Azure portal. To implement this a proper Azure subscription is the prerequisite. The steps involved in this process are

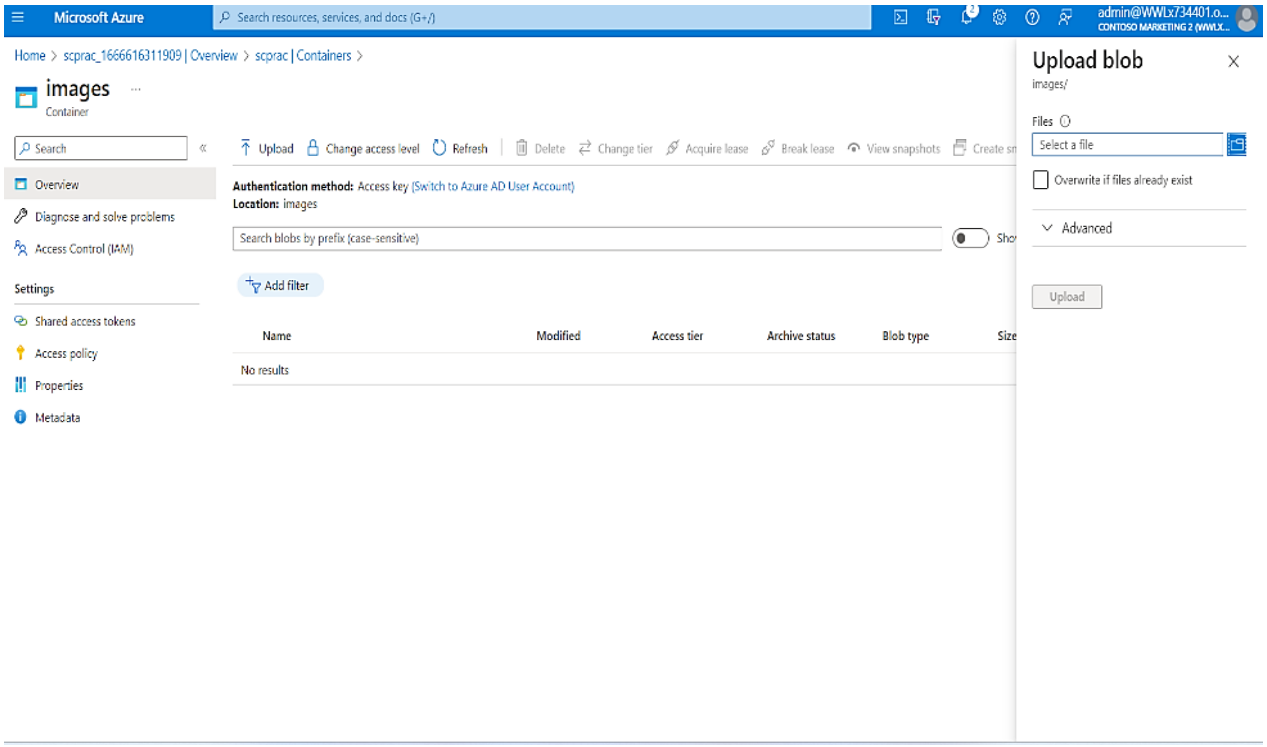
a) Create a Storage account in the Azure portal



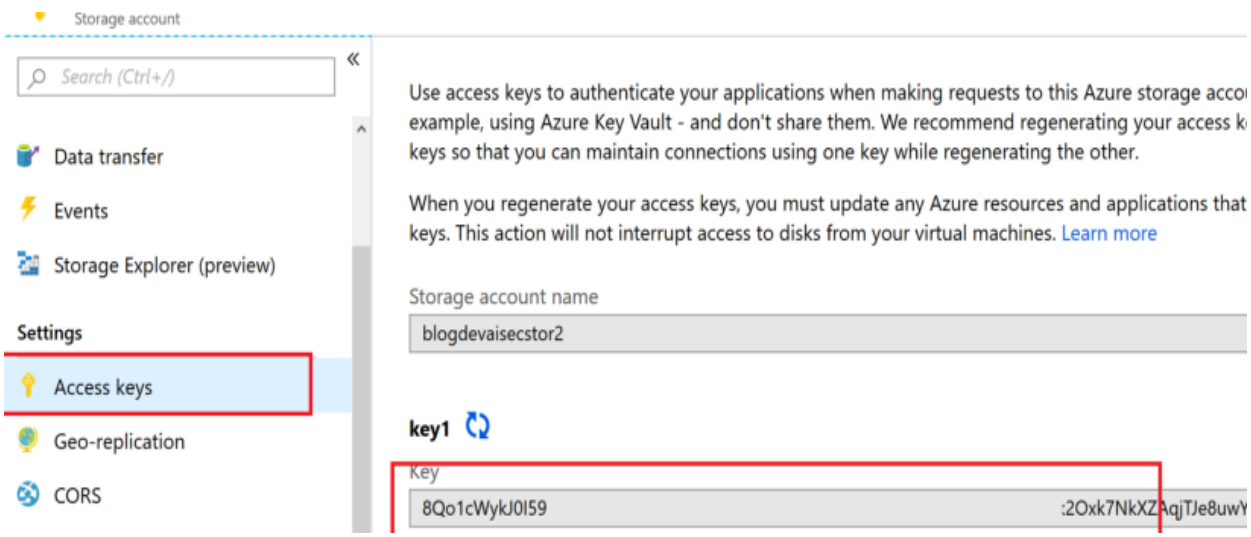
b) Select Containers in the Storage account and create it.



c) Upload blob in the created container

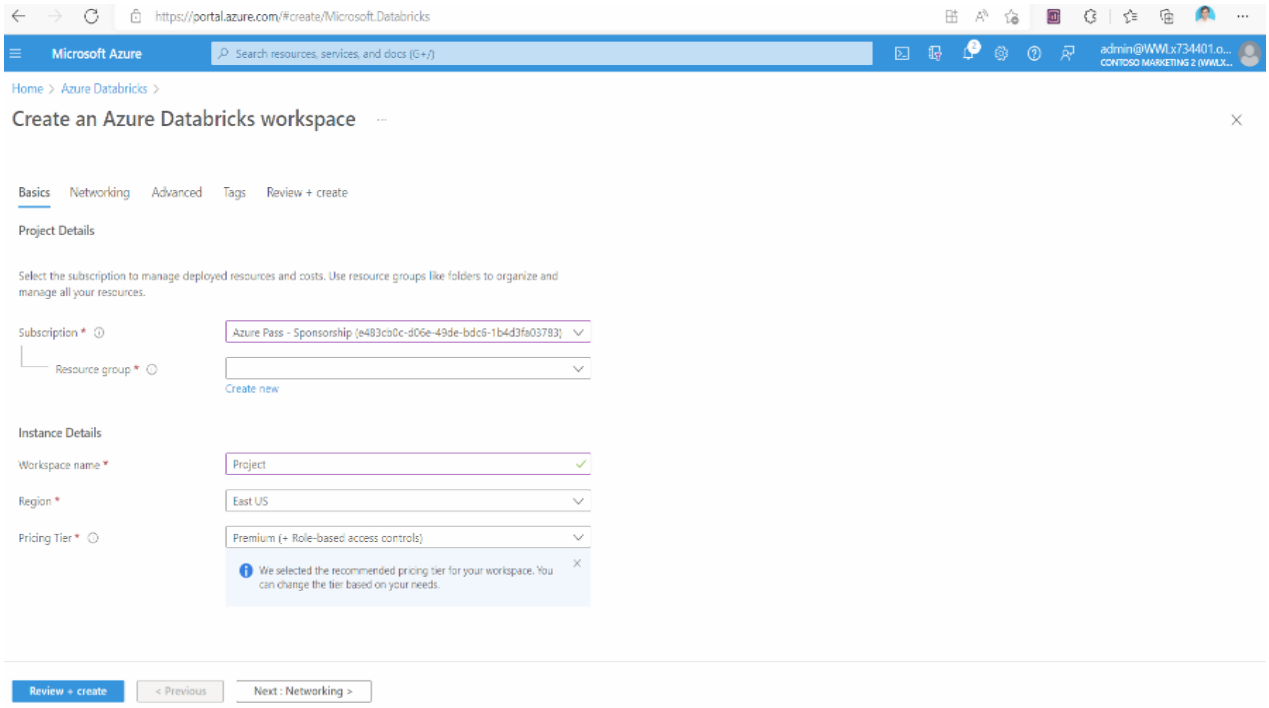


d) After uploading the images to storage account , it is needed to copy the Access Keys. This key will be useful in mounting storage account to Azure Databricks.

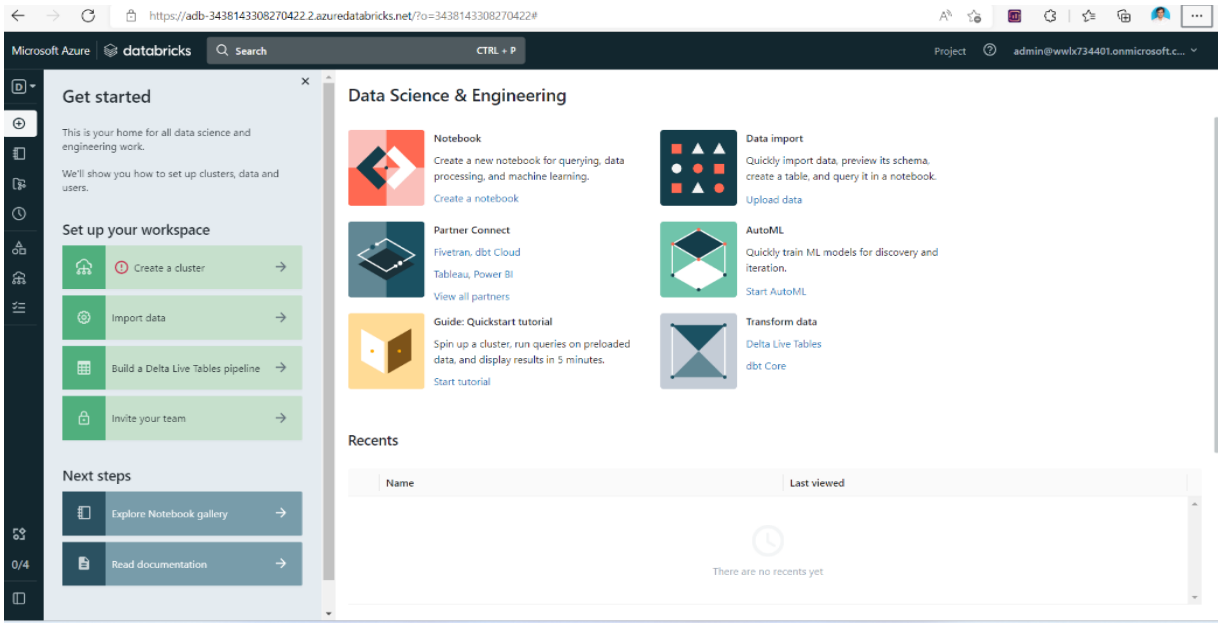


The next important phase of the work is to create deep learning cluster.To implement the mentioned requirement the following steps are to be followed

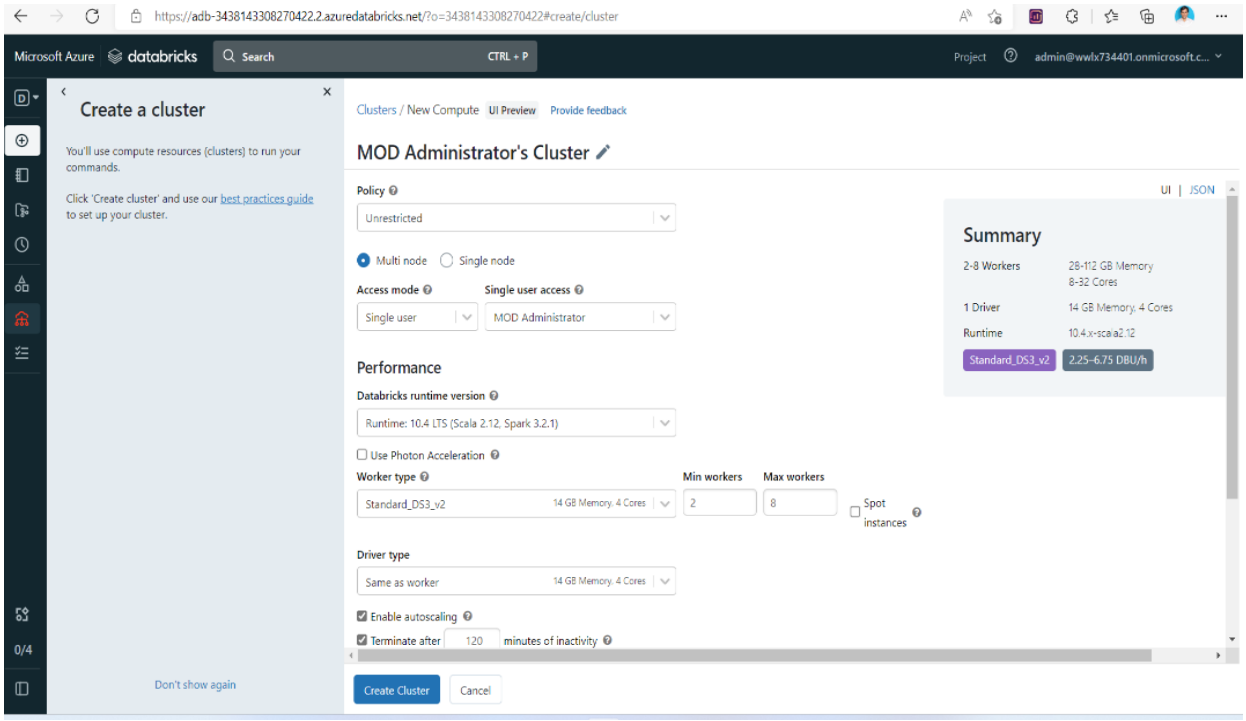
e) Create an Azure Databricks Workspace in the Azure portal



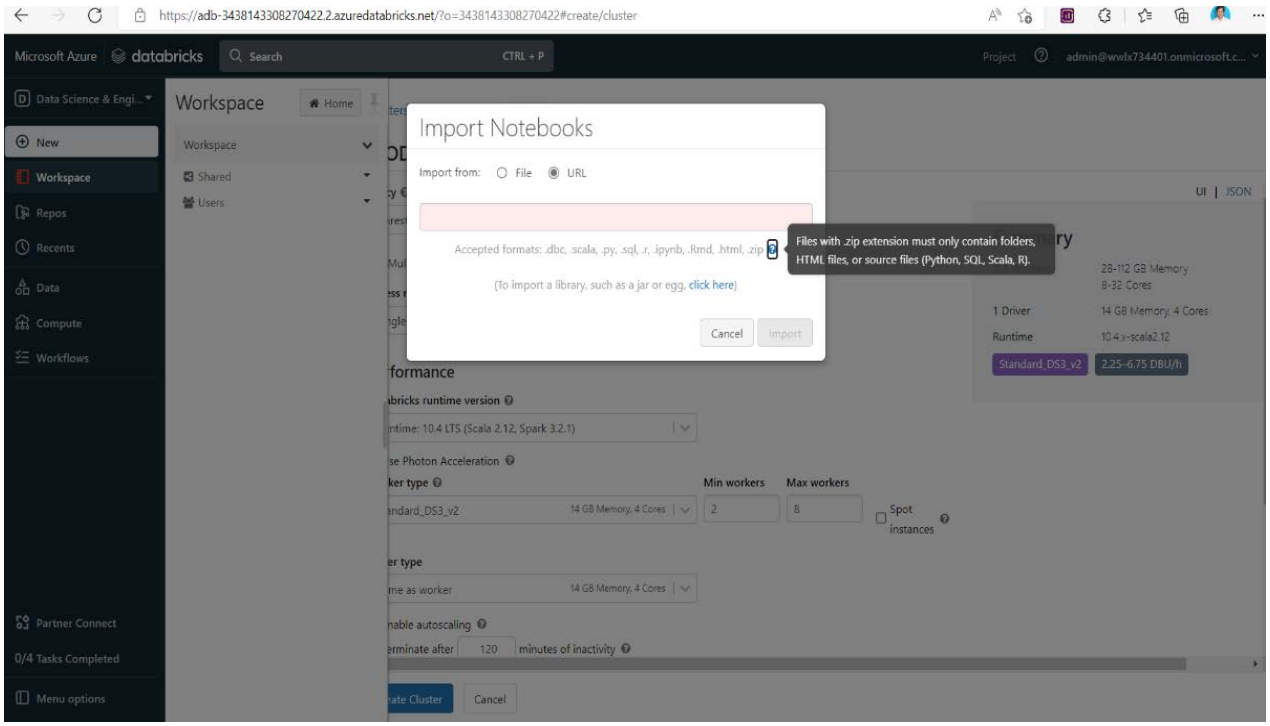
f) Sign in to the Azure Databricks after the creation of Azure Databricks Workspace



g) Create a Cluster with the required settings needed to implement the deep learning model. It is preferable to go through the best practices guide to do this task(https://learn.microsoft.com/en-us/azure/databricks/clusters/cluster-config-best-practices) to set up the node.

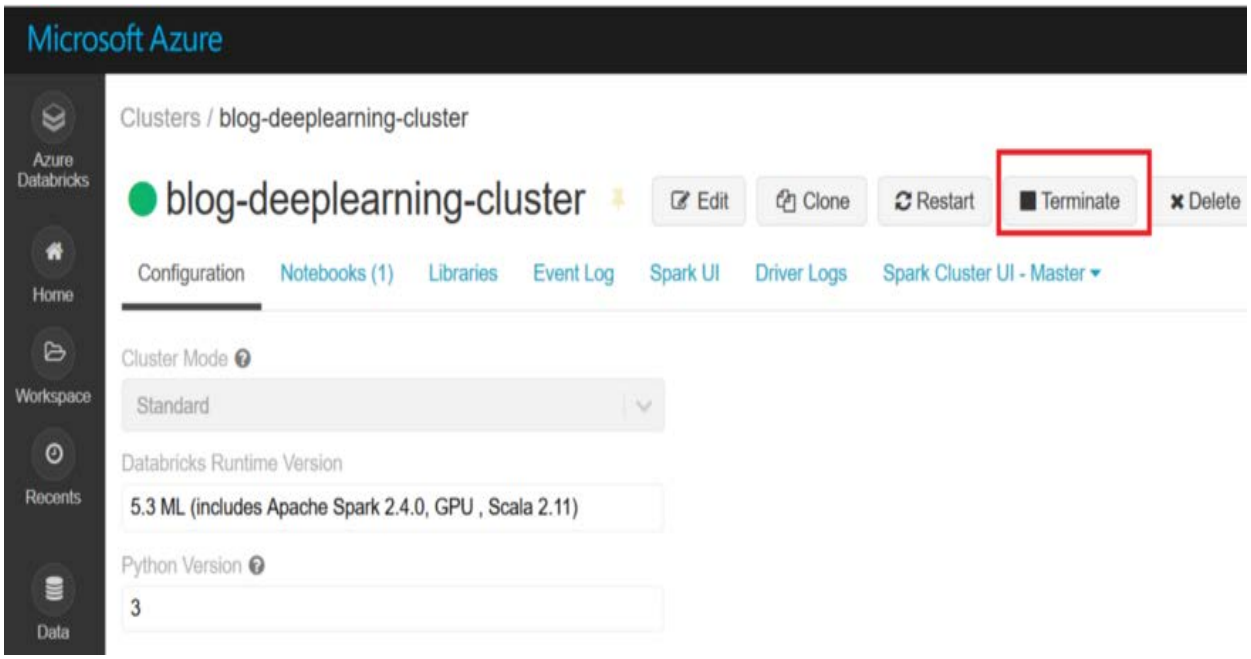


h)The storage account created in the earlier step needs to be mounted to Azure Databricks workspace .This can be done by using the import option available in the workspace. The python codewritten(https://github.com/kvn-rajesh/Azure-Blogathon/blob/main/Mount%20Storage%20Account) to import the storage account to workspace should be selected in the import notebook option as represented in the below screenshot.



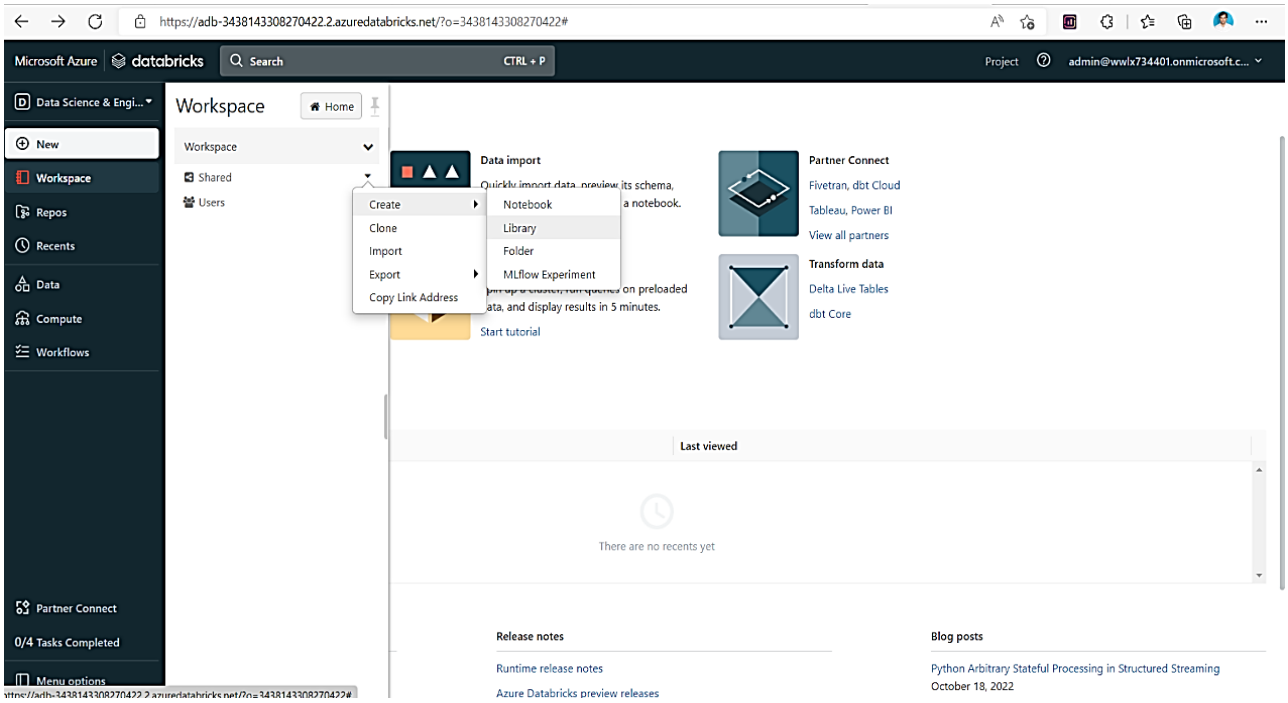
i) The next important phase of the work is to train the deep learning model. This can be done by importing the python based model training code notebook(https://github.com/kvn-rajesh/Azure-Blogathon/blob/main/Leafclassifier) into the Azure Databricks workspace. The import option available in the databricks workspace can be again used for this task. The code should mainly focus on the training of the model based on the input images.

j) The cost of computing resources used in this work is high. It is advisable to terminate the computing clusters when they are not in use.

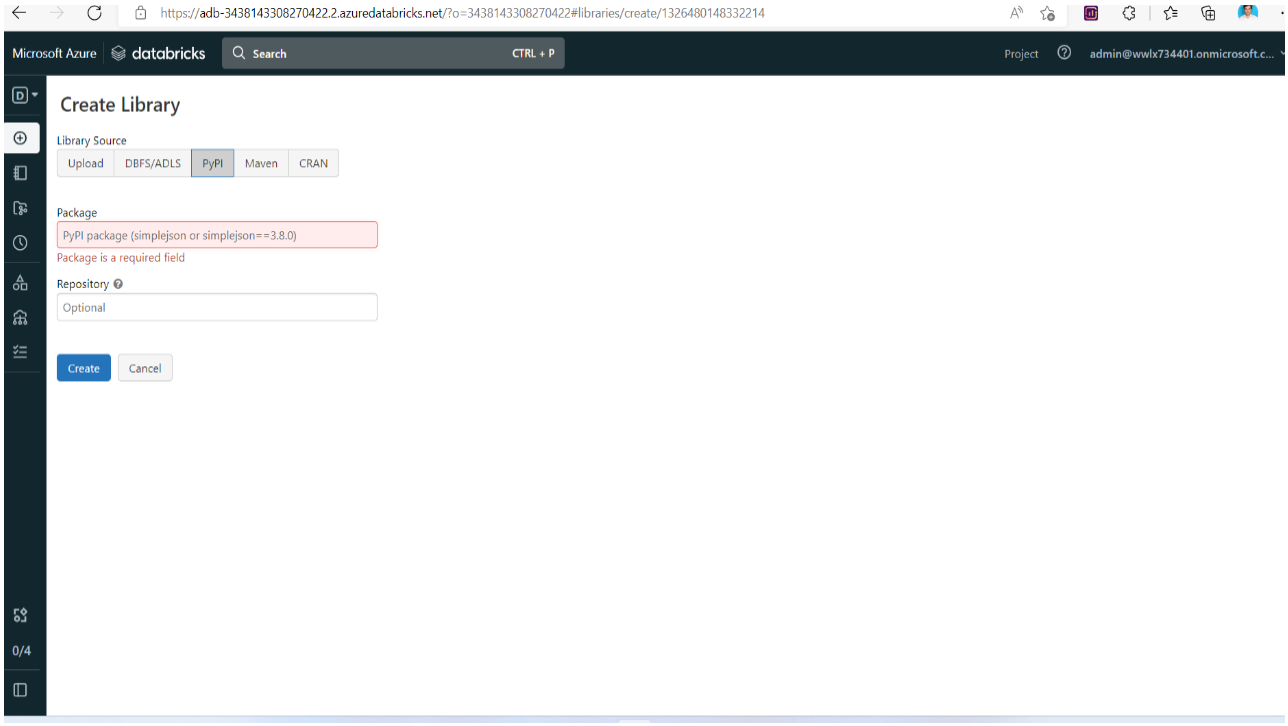


Deep learning project deployment is the next important phase of implementation in the present work.The deployment of Deep Learning project is implemented using the steps mentioned below

k) The deployment of the deep learning model is done by creating a new cluster in Azure Databricks. The process of cluster creation is the same as explained in the previous section of this article. Additionally, a few libraries are needed to deploy the model. To implement this the create library option of shared folder option should be used.



l) After selecting the library option an interface as shown below will appear



m) It is advised to select PyPI option and add the libraries as mentioned below

azureml -sdk(databricks)

keras

tensorflow

After adding the required libraries, the next important step is to register the model and its log metrics into the Azure ML service workspace. After running the log metrics related to python code(https://github.com/kvn-rajesh/Azure-Blogathon/blob/main/Flavia%20deploy%20model) the result of this step can be observed in the experiments section of the Azure ML service workspace. Finally, to create HTTP model there is a need to create an endpoint out of the trained model on the Flavia leaf dataset. The IP address of the endpoint will be found in the Azure Container Instance. There is always a need to test the model. To test the model, we need to provide a test leaf image that matches one of the categories of the Flavia leaf dataset. We need to convert the leaf image in png format to base 64 encoding using an online website such as https://onlinepngtools.com/convert-png-to-base64. The predictions can be done by using a tool like postman (https://www.postman.com/). This tool is a standalone software testing Application Programming interface that can be used to test our deep learning model.

4.Challenges in implementing the solution

The whole process of learning about deep learning models and designing the architecture in the Azure platform itself was a big challenge. The whole process of using Azure Databricks for training the model was time-consuming. My certification in the Microsoft course Designing and Implementing a Microsoft Azure AI Solution(AI-102) has helped in developing solutions for the given problem.

5. Business Benefit

The architecture designed in this work can be further advanced to build a mobile application for Plant leaf classification. Mobile applications are much in demand because of the wide usage of mobile phones in the current generation. Various **ayurvedic pharmaceutical companies** can also use this application as the plant leaves are widely used in ayurvedic medicines. An application of this nature can be very useful in identifying various species of plants across the globe.

6.Conclusion

The motivation behind the work described in this article was to come up with an automatic computer vision-based system to identify the leaves of the plant. A model of such nature was created using Python and Azure services,

Note: I have recorded the entire work and it can be accessed using the link: https://github.com/kvn-rajesh/Azure-Blogathon/blob/main/Blogathon%20Presentation.pptx

7.References

1. https://flavia.sourceforge.net/

2.https://learn.microsoft.com/en-us/azure/storage/common/storage-account-create?tabs=azure-portal

3.https://learn.microsoft.com/en-us/azure/databricks/scenarios/quickstart-create-databricks-workspace-portal?tabs=azure-portal

4.https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/ai/training-deep-learning

-

Dr.K.V.N.Rajesh(Ph.D in Deep Learning),
Subject Matter Expert (CloudThat)