

AZURE DEPLOYMENT OF DEEP LEARNING MODEL

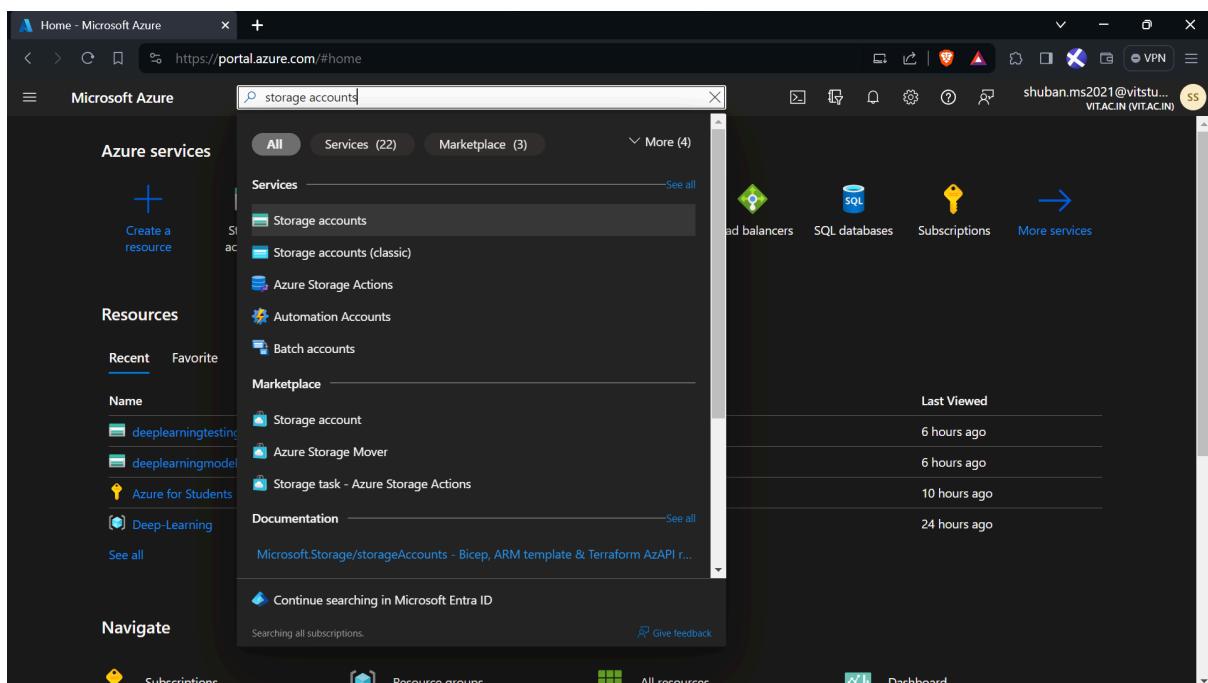
The following steps are after you have successfully signed into your azure account using your VIT student email id.

Creating a STORAGE ACCOUNT

Once you have signed in navigate to the main azure portal page using this link:

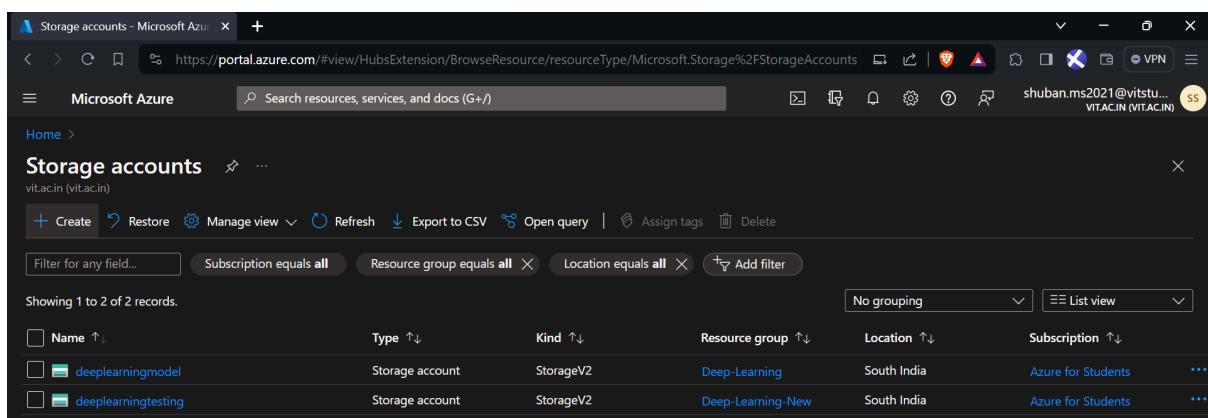
<https://portal.azure.com/#home>

In the search bar type in '**Storage accounts**' and click on the option with the green icon not the classic one.



The screenshot shows the Microsoft Azure portal's home page. In the top search bar, 'storage accounts' has been typed. Below the search bar, there are three tabs: 'All', 'Services (22)', and 'Marketplace (3)'. The 'All' tab is selected. A list of services is displayed, with 'Storage accounts' being the first item and highlighted with a green icon. Other items in the list include 'Storage accounts (classic)', 'Azure Storage Actions', 'Automation Accounts', 'Batch accounts', 'Storage account', 'Azure Storage Mover', and 'Storage task - Azure Storage Actions'. To the right of the search results, there are icons for 'Load balancers', 'SQL databases', 'Subscriptions', and 'More services'. Below the search results, there is a section titled 'Last Viewed' with items like 'Storage account' (6 hours ago), 'Azure Storage Mover' (6 hours ago), 'Storage task - Azure Storage Actions' (10 hours ago), and 'Microsoft.Storage/storageAccounts - Bicep, ARM template & Terraform AzAPI r...' (24 hours ago). At the bottom of the page, there are links for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'.

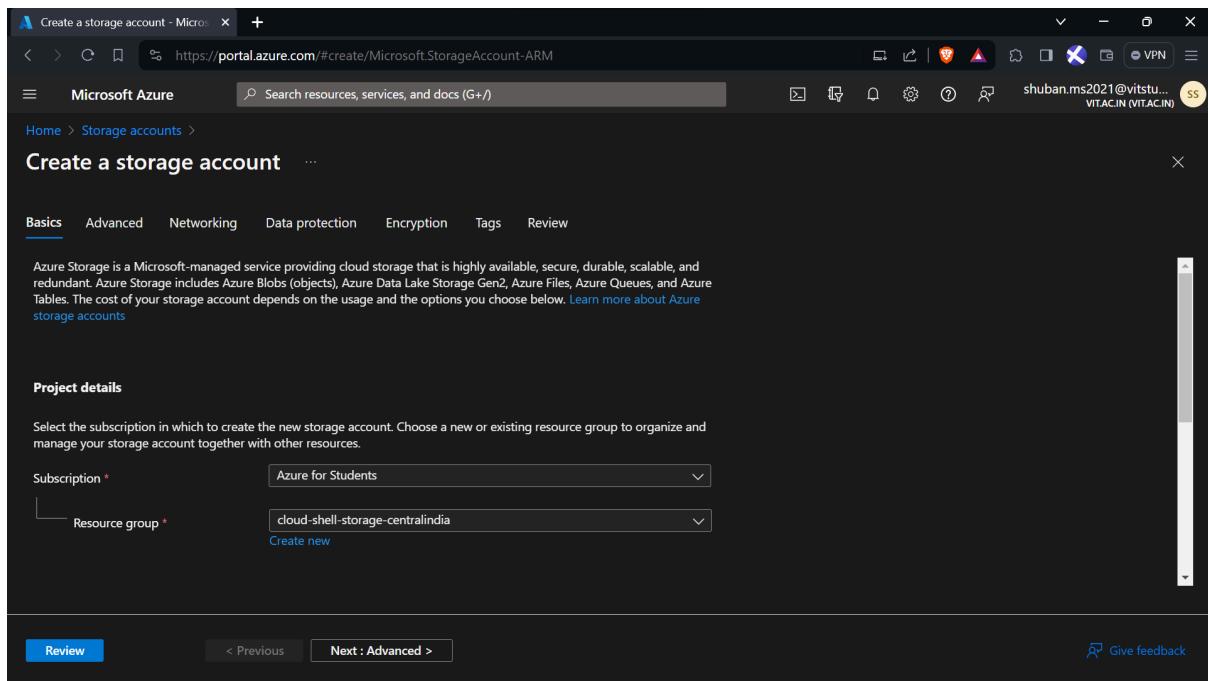
After selecting the storage account option then you should have a screen like the below where you see the option to create a new storage account



The screenshot shows the 'Storage accounts' blade in the Microsoft Azure portal. At the top, there is a search bar and a 'Create' button. Below the search bar, there are filters for 'Subscription equals all', 'Resource group equals all', and 'Location equals all'. The main area displays a table with the following data:

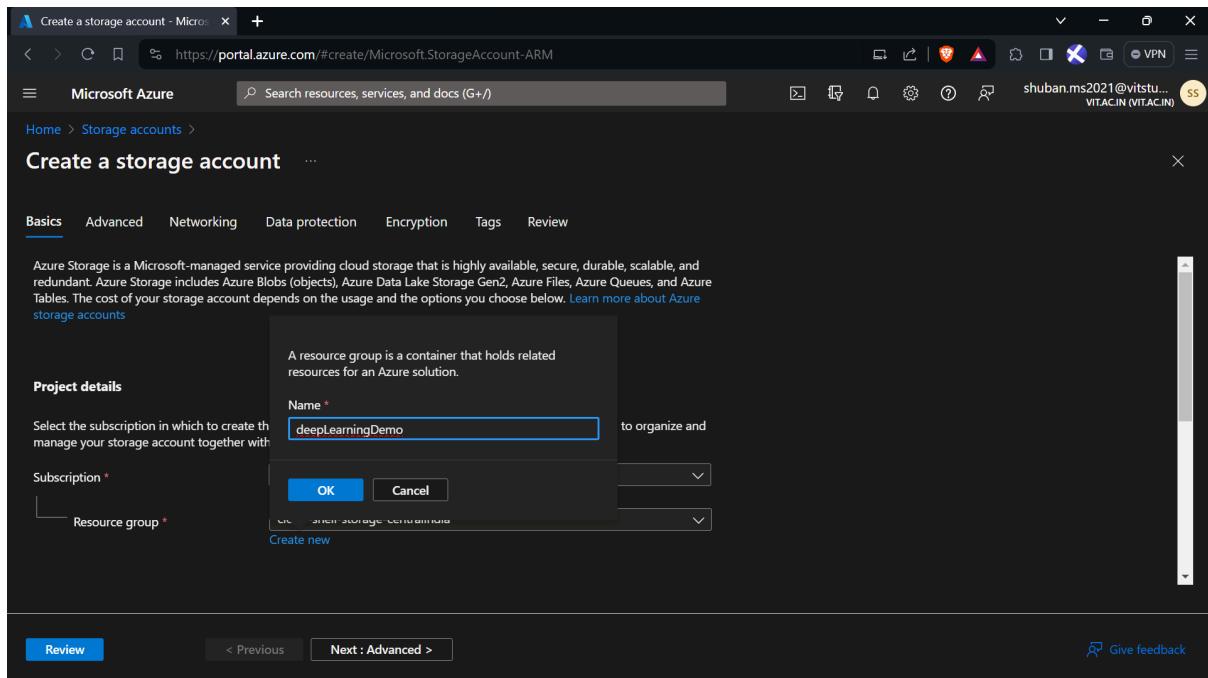
Name	Type	Kind	Resource group	Location	Subscription
deeplearningmodel	Storage account	StorageV2	Deep-Learning	South India	Azure for Students
deeplearningtesting	Storage account	StorageV2	Deep-Learning-New	South India	Azure for Students

Click on create with the plus icon on the left hand side, then you need to fill out some basic details about the storage account

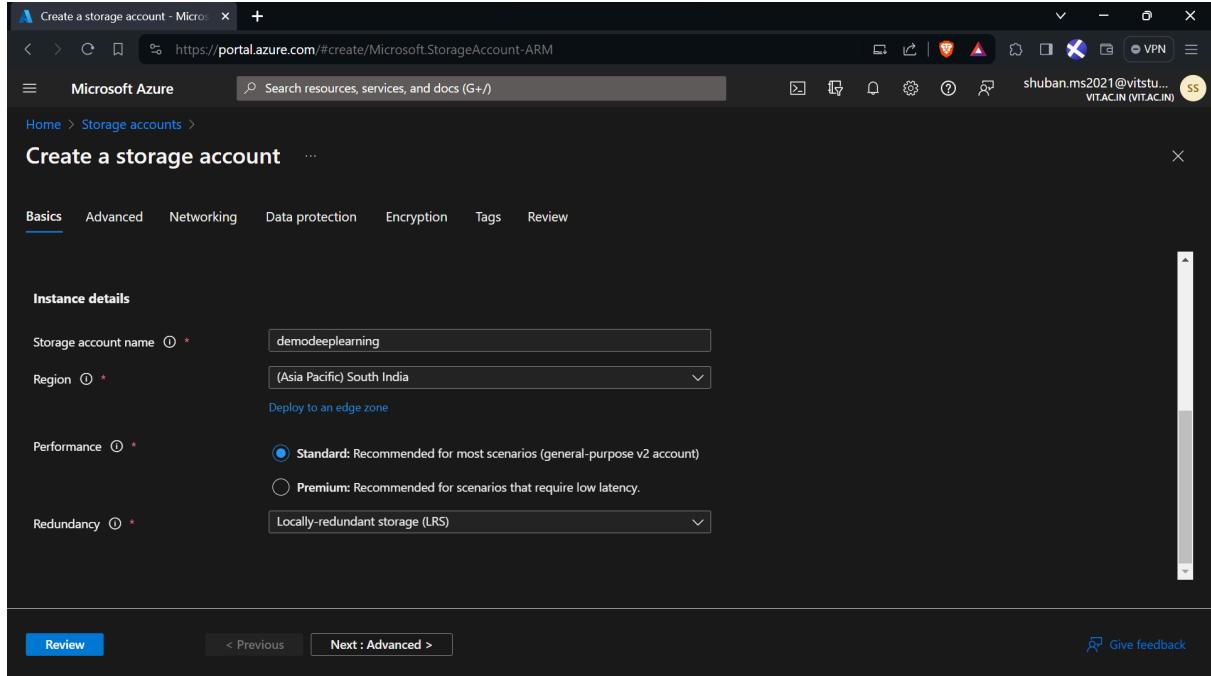


For the subscription '**Azure for students**' should be the default if not select it from the drop down

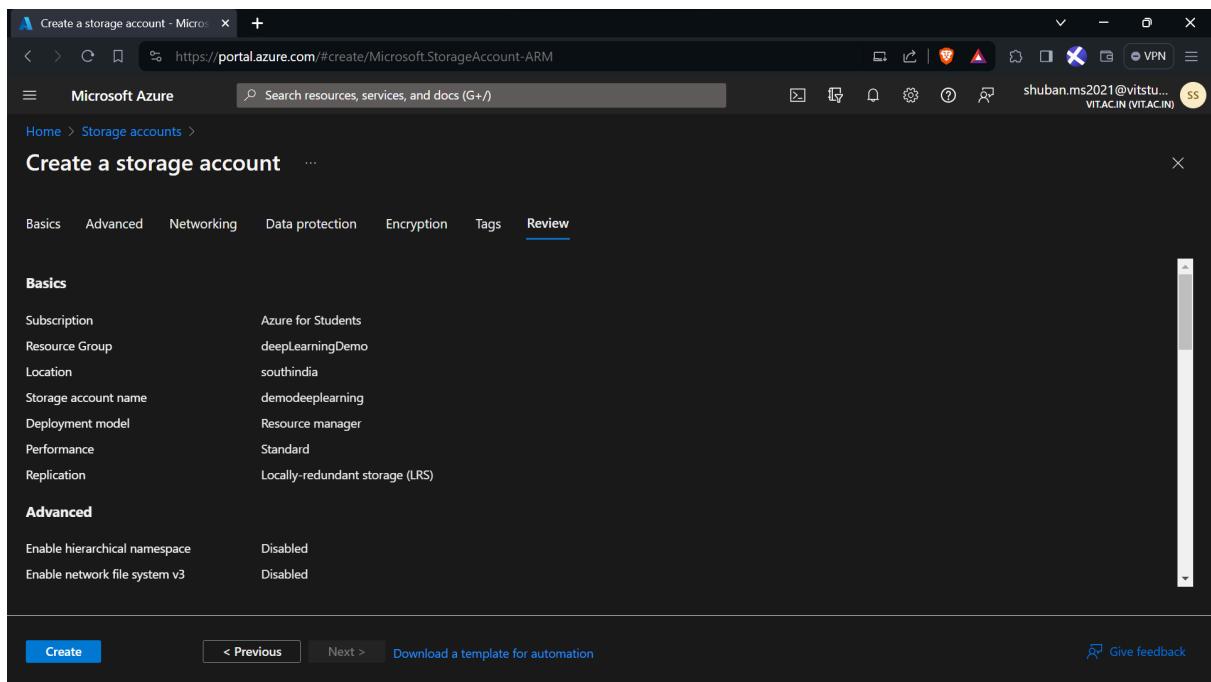
Click on the '**Create new**' for the resource group option and give in a unique name following the rules of name creation and then click ok



Scroll down and give in a unique storage account name along with other options as shown in the screenshot like region, performance and redundancy



After filling all these details click on 'Review'



Make sure your details are correct and click on ‘Create’, you should be taken to a new page where it shows ‘Deployment in progress’

The screenshot shows the Microsoft Azure portal with a deployment named "demodeeplearning_1706809284650". The status is "Deployment is in progress". Deployment details include:

- Deployment name: demodeeplearning_1706809284650
- Subscription: Azure for Students
- Resource group: deepLearningDemo
- Start time: 2/1/2024, 11:11:30 PM
- Correlation ID: 4a62e0dd-4bf7-40d4-8637-95ae8b3640

A sidebar on the right contains links to Microsoft Defender for Cloud, Microsoft tutorials, and Work with an expert.

Wait for some time and you should see a success message for deployment. This means your resource group and your containers are successfully deployed to the azure server and are ready to be used

The screenshot shows the Microsoft Azure portal with the same deployment. The status message now says "Your deployment is complete". Deployment details are identical to the previous screenshot. A sidebar on the right contains links to Cost Management, Microsoft Defender for Cloud, Microsoft tutorials, and Work with an expert.

Then from this page shown above you can click on ‘Home’ and go back to the main portal page of Azure

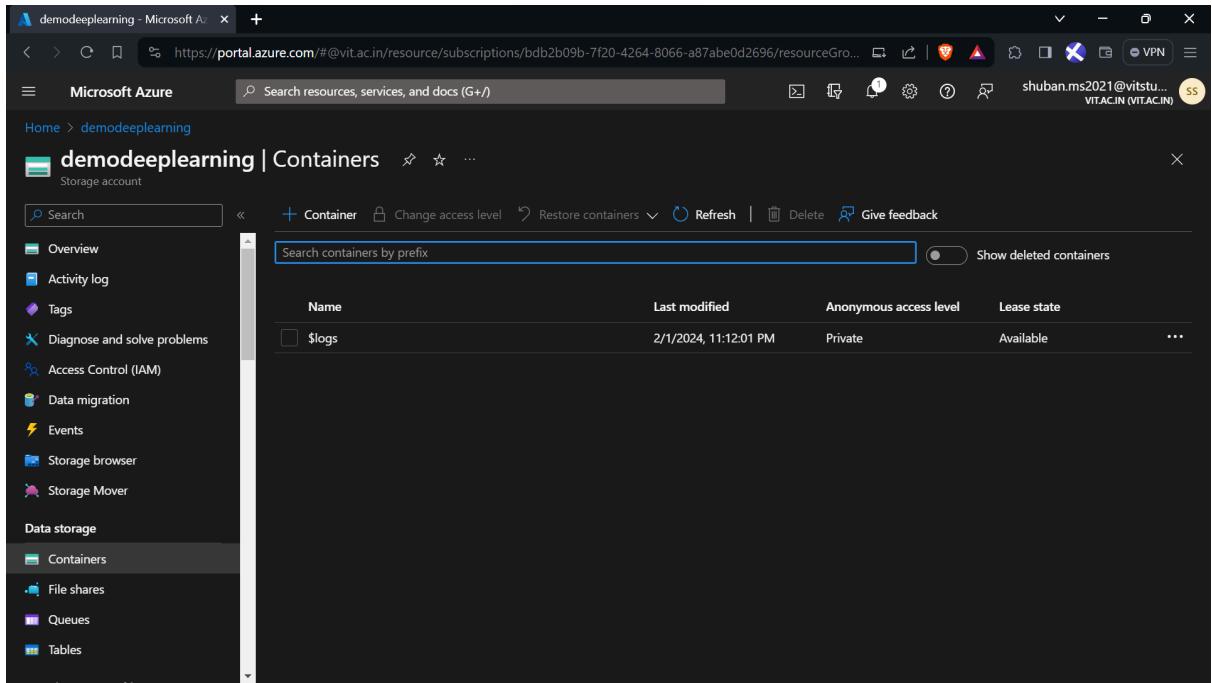
The screenshot shows the Microsoft Azure Home page. At the top, there's a search bar and a user profile icon. Below the header, there's a section titled "Azure services" with various icons for "Create a resource", "Storage accounts", "Azure Machine Learning", "App Services", "Bot Services", "All resources", "Load balancers", "SQL databases", "Subscriptions", and "More services". Under "Resources", there are tabs for "Recent" and "Favorite". The "Recent" tab is selected, displaying a list of resources with columns for "Name", "Type", and "Last Viewed". The resources listed are: "demodeeplearning" (Storage account, 3 minutes ago), "deepLearningDemo" (Resource group, 3 minutes ago), "deeplearningtesting" (Storage account, 6 hours ago), "deeplearningmodel" (Storage account, 7 hours ago), "Azure for Students" (Subscription, 11 hours ago), and "Deep-Learning" (Resource group, 24 hours ago). There's also a "See all" link at the bottom of the list.

Once on this page you should be able to see the storage account name pop up under the resources tab, if not then navigate to the ‘Storage account’ page as discussed in the beginning

The screenshot shows the Microsoft Azure Storage account details page for "demodeeplearning". The left sidebar has sections like Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, Storage Mover, Data storage (Containers, File shares, Queues, Tables), and Blob service. The main content area shows the "Essentials" section with details such as Resource group (deepLearningDemo), Location (southindia), Subscription (Azure for Students), Subscription ID (bdb2b09b-7f20-4264-8066-a87abe0d2696), Disk state (Available), and Tags (edit, Add tags). The "Properties" tab is selected, showing Blob service (Hierarchical namespace: Disabled) and Security (Require secure transfer for REST API). There are also links for Monitoring, Capabilities (0), Recommendations (0), Tutorials, and Tools + SDKs.

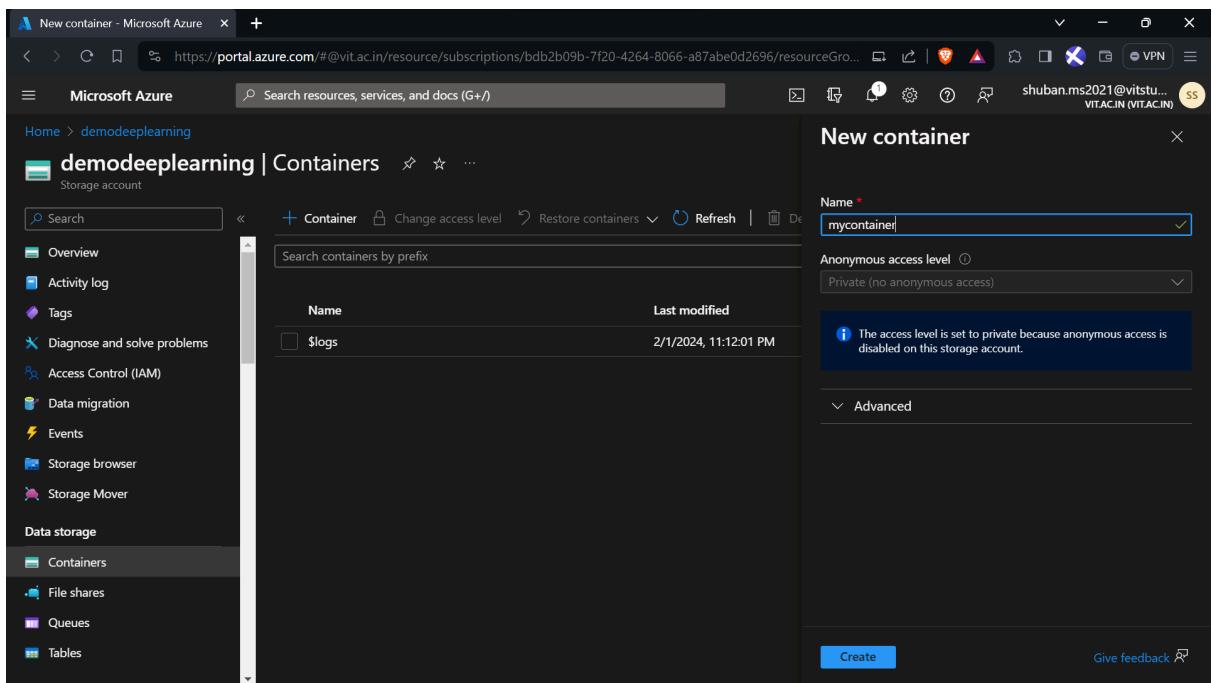
The above image shows the options and methods available within the storage account that you just created

On the left hand options menu click on the '**Containers**' option under the '**Data storage**' option



This screenshot shows the 'Containers' page within the Microsoft Azure portal for a storage account named 'demodeeplearning'. The left sidebar is collapsed, and the main area displays a table of containers. A single row is present in the table, representing the '\$logs' container. The table columns include Name, Last modified, Anonymous access level, and Lease state. The '\$logs' container is listed with a last modification date of 2/1/2024, 11:12:01 PM, Private anonymous access, and Available lease state. Above the table, there is a search bar labeled 'Search containers by prefix' and a button to 'Show deleted containers'. At the top of the page, there are buttons for '+ Container' (with a plus sign), 'Change access level', 'Restore containers', 'Refresh', 'Delete', and 'Give feedback'.

By default there should be just a single logs file. Click on the container button with a + symbol on top to create a new container



This screenshot shows the 'Containers' page within the Microsoft Azure portal for the same storage account ('demodeeplearning'). A new dialog box titled 'New container' is open on the right side of the screen. In this dialog, the 'Name' field is populated with 'mycontainer'. Below it, the 'Anonymous access level' dropdown is set to 'Private (no anonymous access)'. A note in the dialog states: 'The access level is set to private because anonymous access is disabled on this storage account.' At the bottom of the dialog, there is a 'Create' button. The rest of the page shows the existing '\$logs' container and the standard navigation and search controls.

Give it a new name and click on 'Create'

Name	Last modified	Anonymous access level	Lease state
\$logs	2/1/2024, 11:12:01 PM	Private	Available
mycontainer	2/1/2024, 11:18:51 PM	Private	Available

After creation it should show up below the logs file

Click on the container that you just created

Name	Modified	Access tier	Archive status	Blob type	Size
No results					

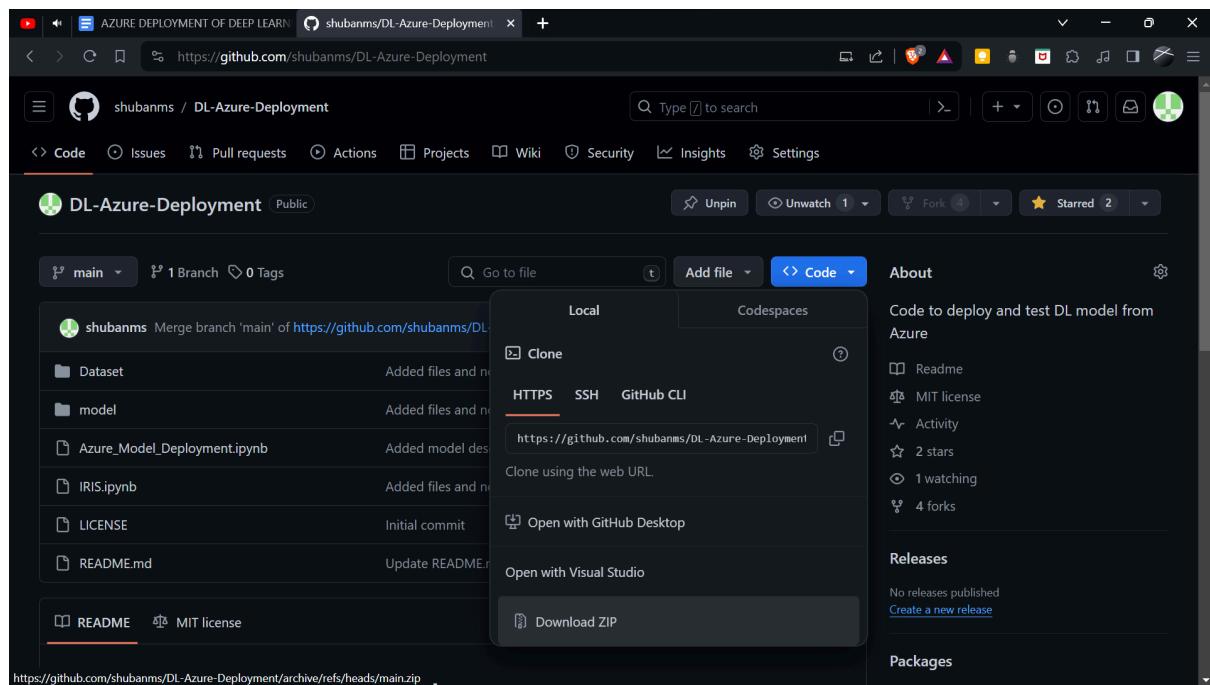
So as you can see the container name shows up on the top left corner and an option to '**Upload**' a file to this container

Now that we have reached till this step, we can shift focus to creating our model and saving it into a '**model_name.h5**' file

Navigate to this github link to access the codes and the dataset along with the final model as well:

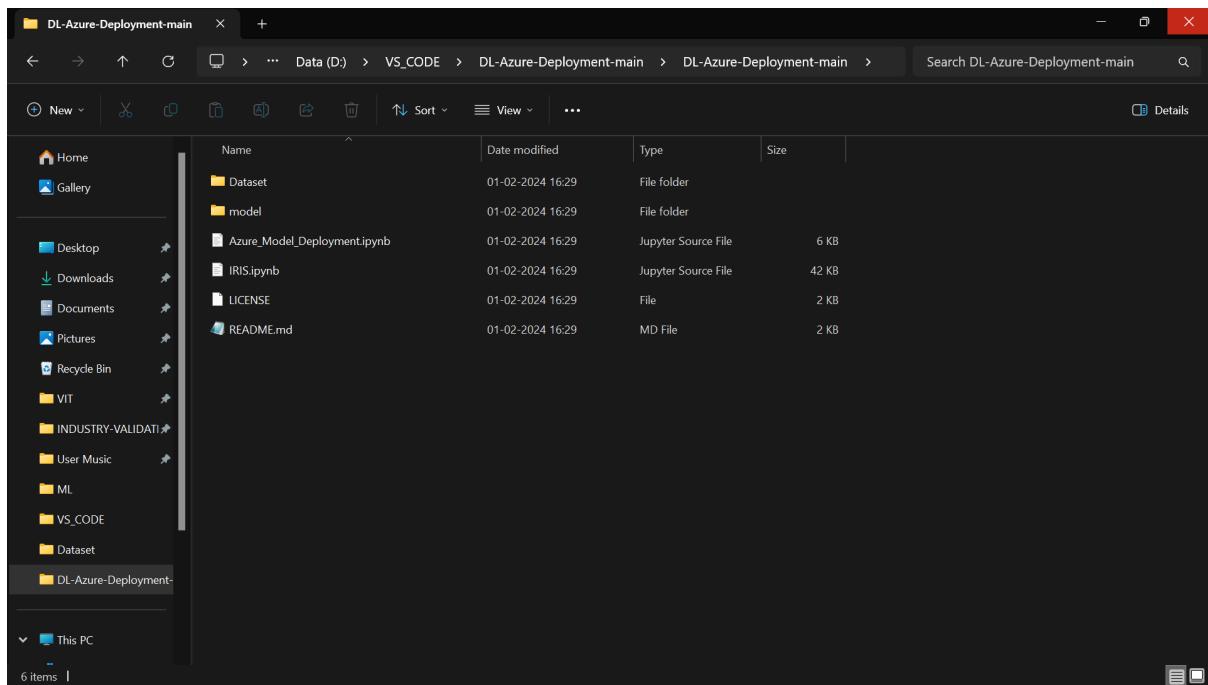
<https://github.com/shubanms/DL-Azure-Deployment>

If you have git set-up locally follow the steps to fork and then clone or clone directly to your local system. If you don't have git set up then you can download the repository locally to your machine



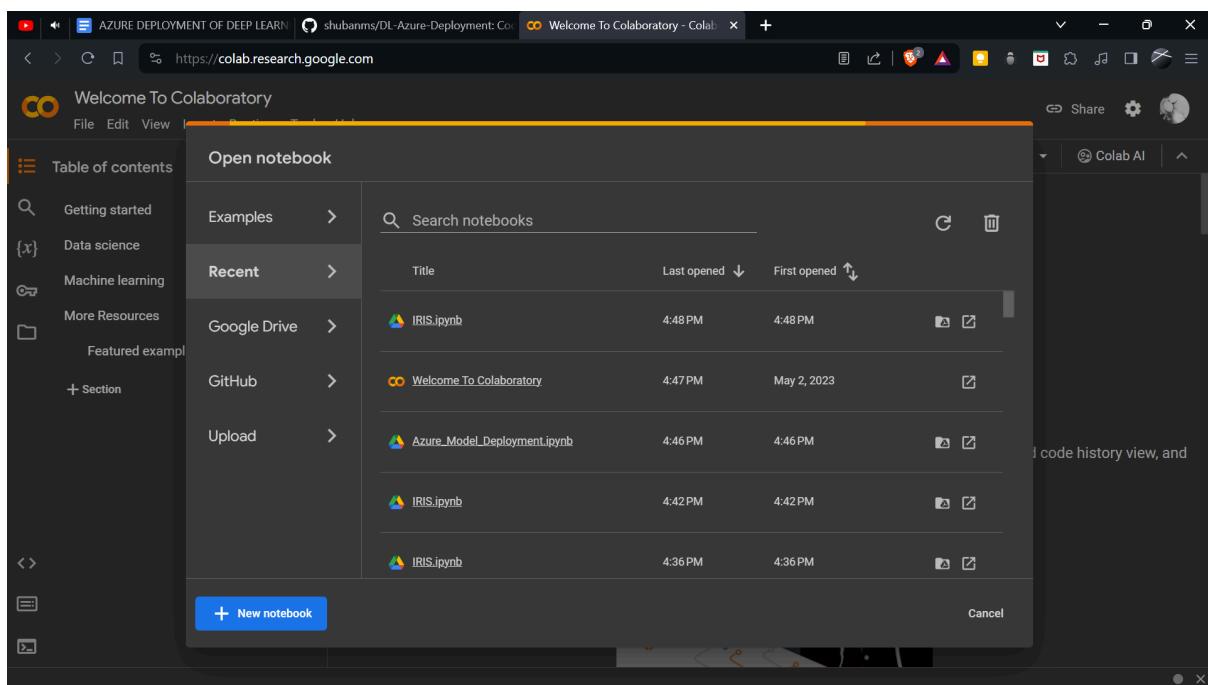
Click on the blue icon '**Code**' and the the option of '**Download ZIP**'

Extract the ZIP file to the preferred location and you should have the contents of the folder like this



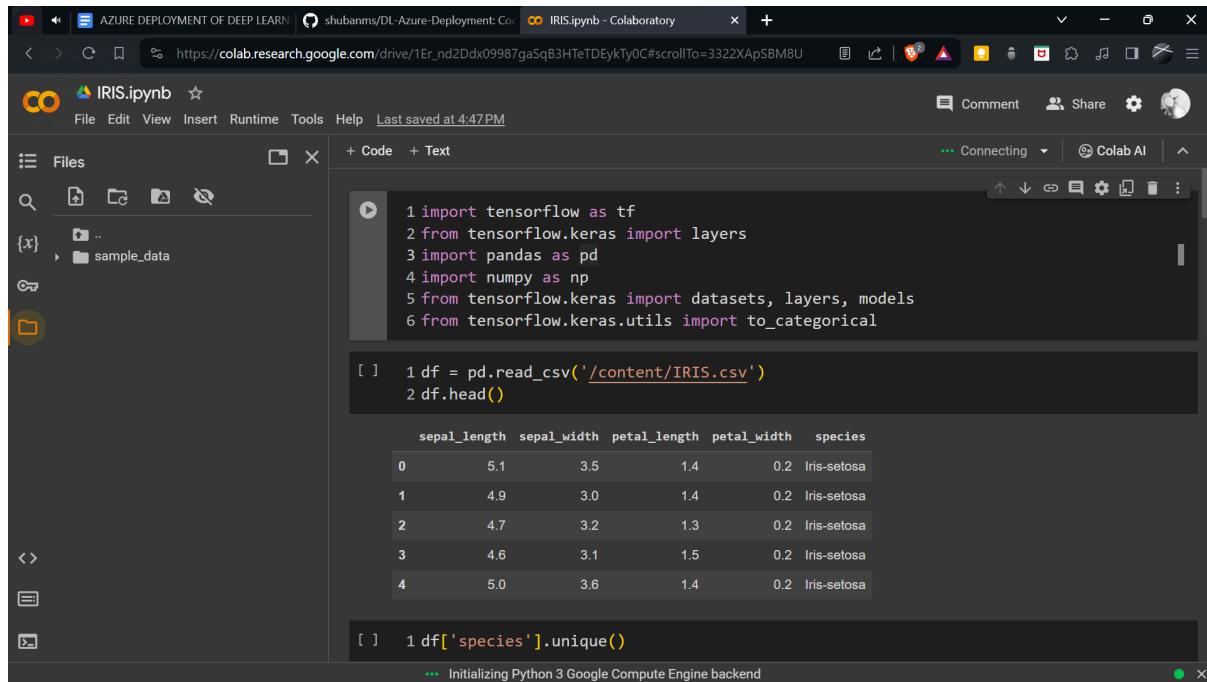
On a new browser tab open google colab:

<https://colab.research.google.com/>



Click on '**Upload**' option on the left menu and then browse and find the IRIS.ipynb file from the extracted folder and upload it to colab

After the jupyter notebook is uploaded to colab then select the folder like option on the left hand menu



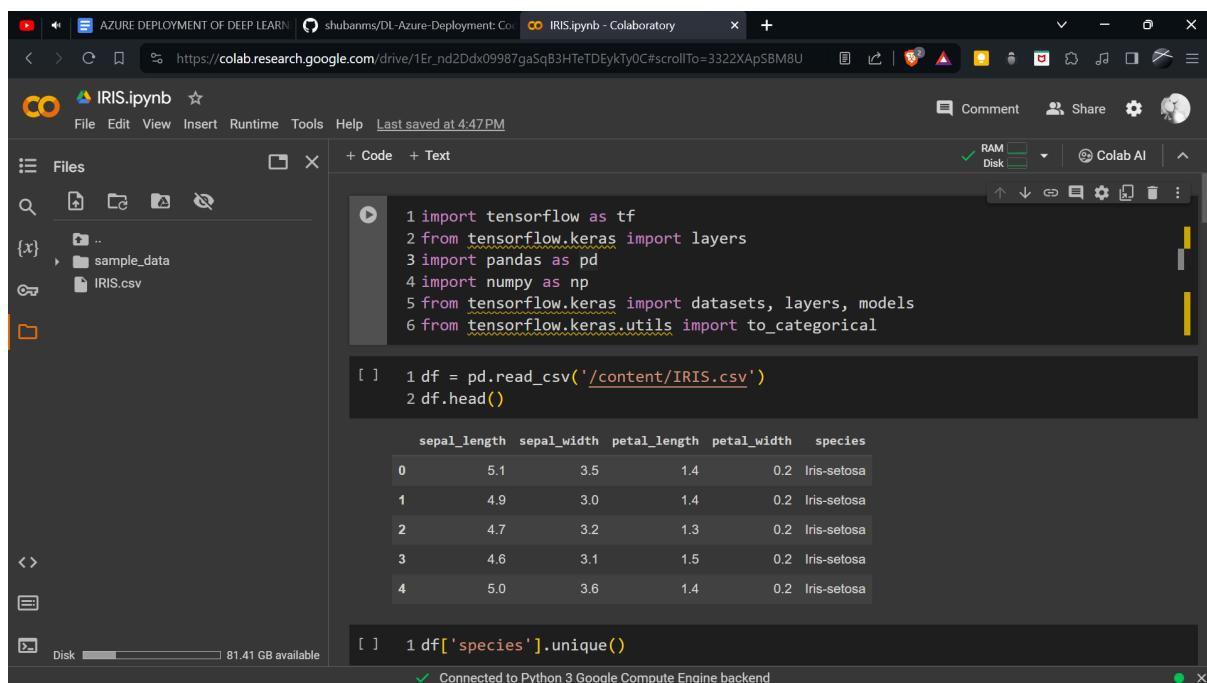
```
1 import tensorflow as tf
2 from tensorflow.keras import layers
3 import pandas as pd
4 import numpy as np
5 from tensorflow.keras import datasets, layers, models
6 from tensorflow.keras.utils import to_categorical

[ ] 1 df = pd.read_csv('/content/IRIS.csv')
2 df.head()

      sepal_length  sepal_width  petal_length  petal_width    species
0            5.1         3.5          1.4         0.2  Iris-setosa
1            4.9         3.0          1.4         0.2  Iris-setosa
2            4.7         3.2          1.3         0.2  Iris-setosa
3            4.6         3.1          1.5         0.2  Iris-setosa
4            5.0         3.6          1.4         0.2  Iris-setosa

[ ] 1 df['species'].unique()
... Initializing Python 3 Google Compute Engine backend
```

Then the first of four smaller icons to upload a file to the session, navigate to the same extracted folder and upload the IRIS.csv file under the dataset directory



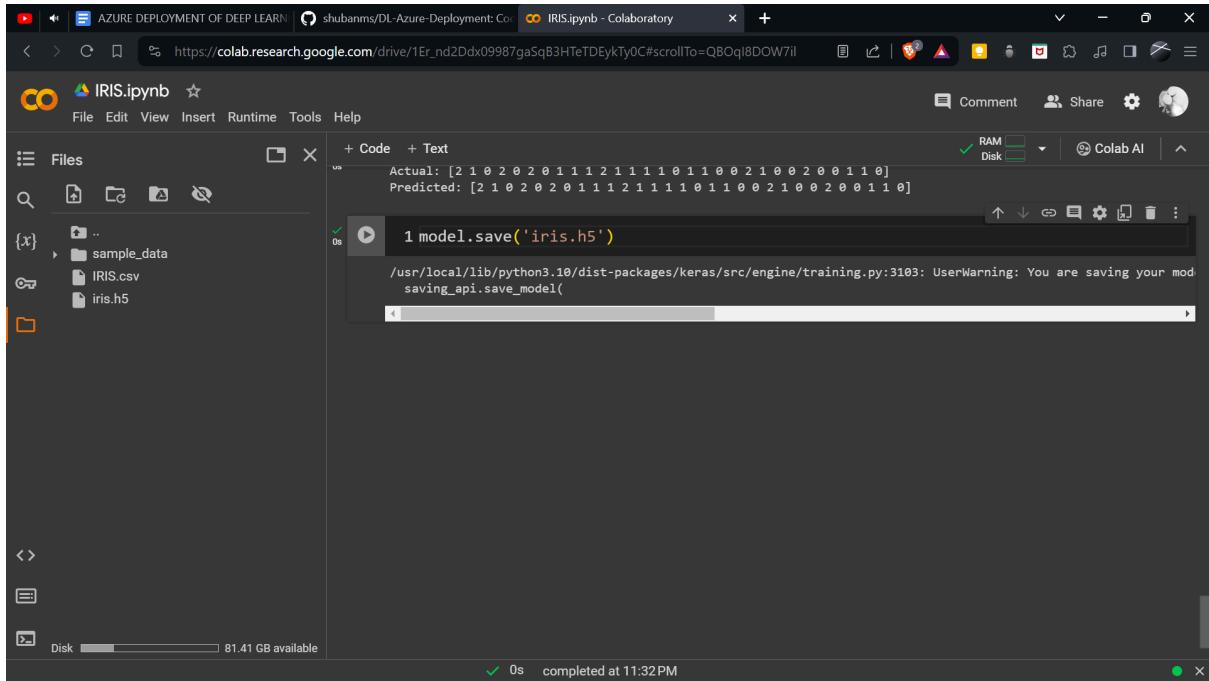
```
1 import tensorflow as tf
2 from tensorflow.keras import layers
3 import pandas as pd
4 import numpy as np
5 from tensorflow.keras import datasets, layers, models
6 from tensorflow.keras.utils import to_categorical

[ ] 1 df = pd.read_csv('/content/IRIS.csv')
2 df.head()

      sepal_length  sepal_width  petal_length  petal_width    species
0            5.1         3.5          1.4         0.2  Iris-setosa
1            4.9         3.0          1.4         0.2  Iris-setosa
2            4.7         3.2          1.3         0.2  Iris-setosa
3            4.6         3.1          1.5         0.2  Iris-setosa
4            5.0         3.6          1.4         0.2  Iris-setosa

[ ] 1 df['species'].unique()
... Connected to Python 3 Google Compute Engine backend
```

After uploading the dataset or the csv file then run all the code cells, you can however go through the code and play around with it making it better



```
Actual: [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0]
Predicted: [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0]

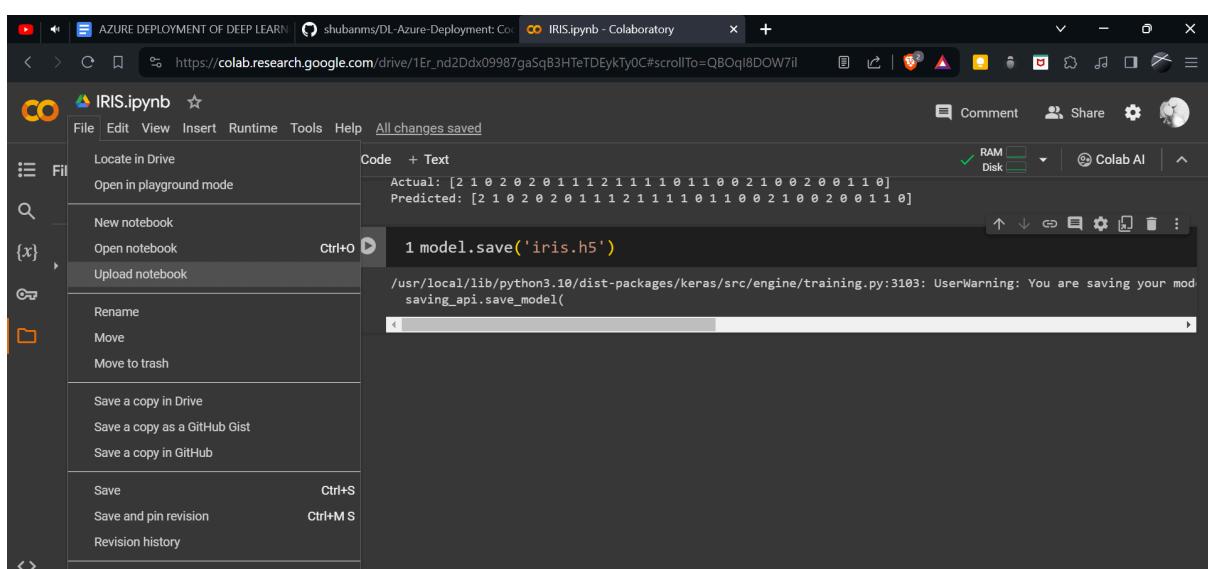
1 model.save('iris.h5')

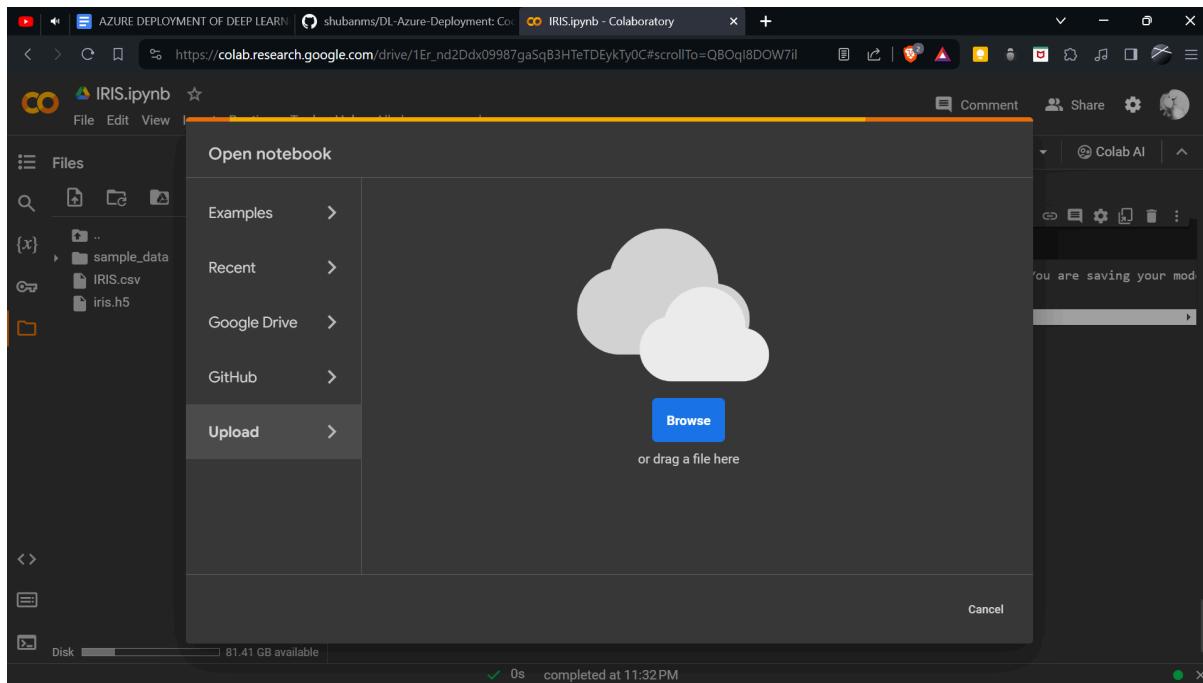
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your mod
saving_api.save_model(
```

After you run the cell in which it saves the model to a '**iris.h5**' file, you should be able to access the file from the left side. Click on the three dots next to the h5 file and download it.

If you face any problem in this whole process there is already a h5 file ready in the extracted folder under the model directory you can use that as well

After this once you have the h5 file with you again click on '**File**' in the colab window and click on '**Upload notebook**'



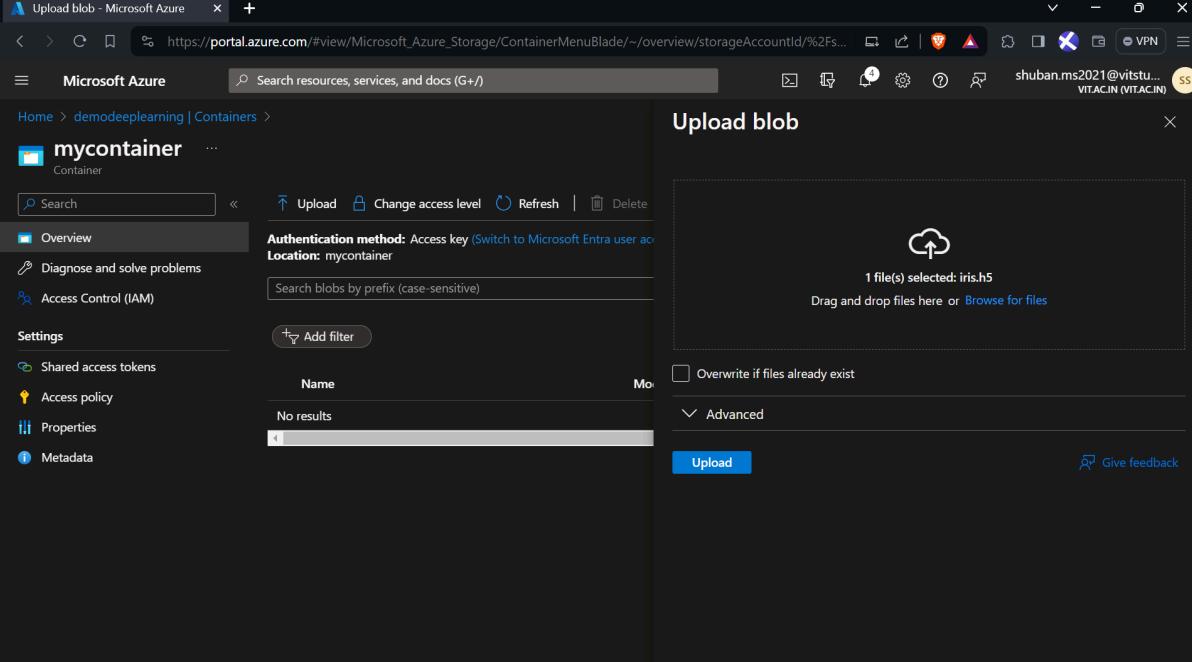


Click on '**Browse**' and then select the other file from the folder called '**Azure_Model_Deployment.ipynb**' and upload it

Now we jump back to azure to get our container name and our connection string to the storage account

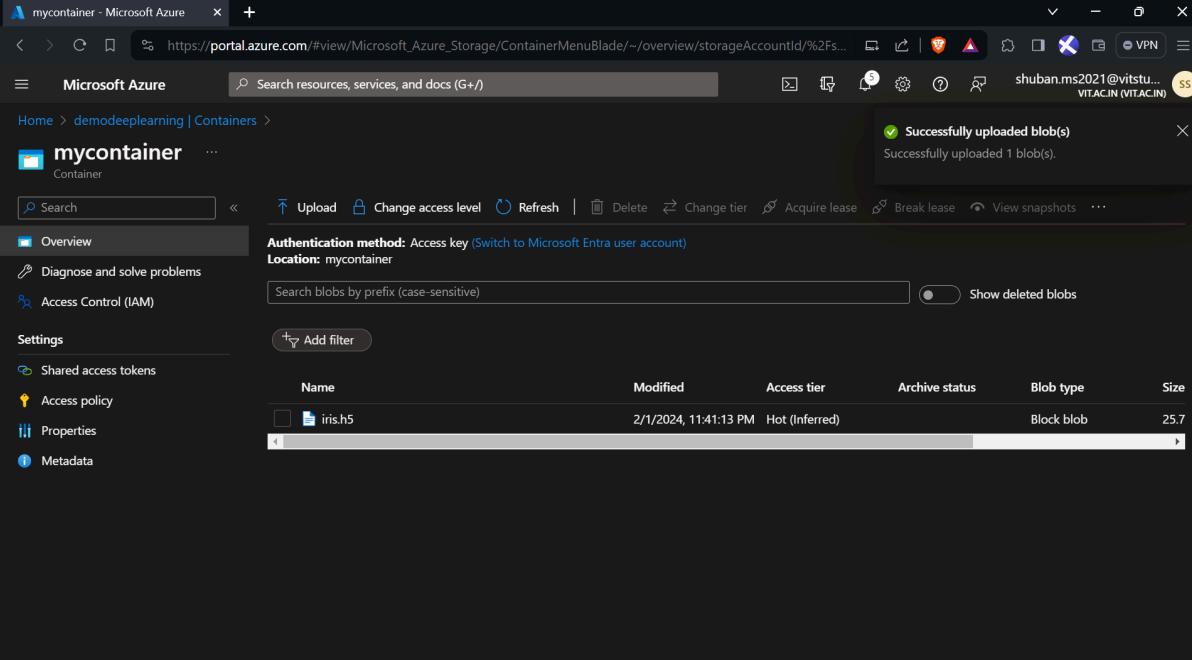
This was the last step on azure

From this page click on ‘Upload’ and then ‘Browse for files’ and upload the h5 file you just created from your model and then click on ‘Upload’



The screenshot shows the 'Upload blob' interface in the Microsoft Azure Storage portal. On the left, there's a sidebar with options like Overview, Diagnose and solve problems, Access Control (IAM), Settings, Shared access tokens, Access policy, Properties, and Metadata. The main area has tabs for Upload, Change access level, Refresh, Delete, and a search bar. It displays the authentication method as 'Access key (Switch to Microsoft Entra user account)' and the location as 'mycontainer'. A central panel shows a cloud icon and a message saying '1 file(s) selected: iris.h5'. Below it is a placeholder text 'Drag and drop files here or Browse for files'. There's also a checkbox for 'Overwrite if files already exist' and a 'Upload' button. A 'Give feedback' link is at the bottom right.

You should then be able to see the file as a blob (Files uploaded to the container are called blobs)



The screenshot shows the 'mycontainer' storage container page in the Microsoft Azure Storage portal. The sidebar is identical to the previous screenshot. The main area shows a table of blobs. The table has columns: Name, Modified, Access tier, Archive status, Blob type, and Size. One row is visible, showing 'iris.h5' as the name, '2/1/2024, 11:41:13 PM' as the modified date, 'Hot (Inferred)' as the access tier, 'Archive status' as 'Not yet archived', 'Blob type' as 'Block blob', and '25.7' as the size. A success message 'Successfully uploaded blob(s)' is displayed at the top right, indicating the file was uploaded successfully.

After uploading the model to the container we can go and get the connection string first

Go back to the portal home:

<https://portal.azure.com/#home>

Then navigate to your container, the steps remain the same as above

The screenshot shows the 'Properties' tab of the Azure Storage account 'demodeeplearning'. The left sidebar lists 'Activity log', 'Tags', 'Diagnose and solve problems', 'Access Control (IAM)', 'Data migration', 'Events', 'Storage browser', and 'Storage Mover'. Under 'Data storage', there are 'Containers', 'File shares', 'Queues', and 'Tables'. The main pane displays the following details:

Essentials	Properties
Resource group (move) deepLearningDemo	Performance Standard
Location southindia	Replication Locally-redundant storage (LRS)
Subscription (move) Azure for Students	Account kind StorageV2 (general purpose v2)
Subscription ID bdb2b09b-7f20-4264-8066-a87abe0d2696	Provisioning state Succeeded
Disk state Available	Created 2/1/2024, 11:11:31 PM
Tags (edit) Add tags	

Below the properties, there are sections for 'Blob service' (Hierarchical namespace: Disabled) and 'Security' (Require secure transfer for REST API).

On the left hand under the '**Security + networking**' tab you should see an option of '**Access keys**', click on that

The screenshot shows the 'Properties' tab of the Azure Storage account 'demodeelearning'. The left sidebar now includes 'Networking', 'Front Door and CDN', 'Access keys', 'Shared access signature', 'Encryption', and 'Microsoft Defender for Cloud'. The main pane displays the same essential information as the previous screenshot.

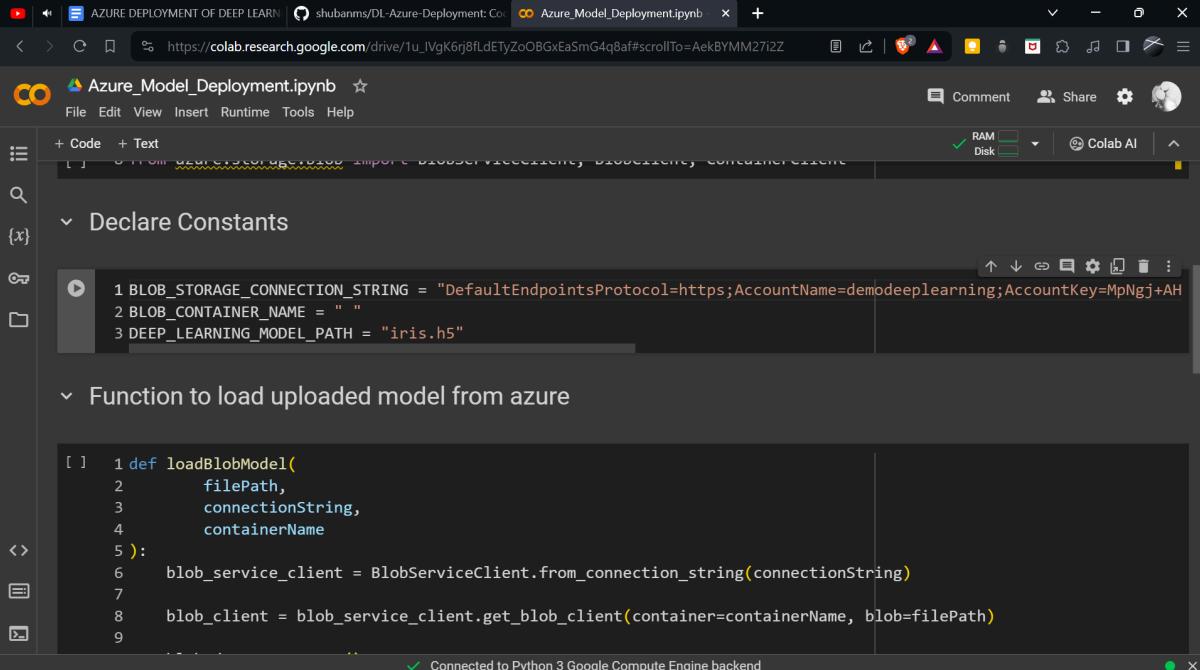
Over here on this page you have two options of key1 and key2 you can use any one of them

The screenshot shows the 'Access keys' page for a storage account named 'demodeelearning'. The left sidebar shows various Azure services like Data storage, Security + networking, and Data management. The main area displays two sets of access keys: 'key1' and 'key2'. Each key includes a 'Rotate key' button, a timestamp indicating it was last rotated on 2/1/2024 (0 days ago), and a 'Show' button next to a redacted key value. Below the keys is a 'Connection string' section with a 'Show' button next to another redacted value.

Under the option of ‘**Connection string**’ click ‘**Show**’ and then copy the string to your clipboard

The screenshot shows the same 'Access keys' page for the 'demodeelearning' storage account. The 'Connection string' section now includes a 'Copy to clipboard' button next to the 'Show' button, which is highlighted in blue, indicating it has been clicked. The rest of the page remains the same, showing the two sets of access keys and their details.

Now after you have copied it jump back to colab where the deployment notebook is open and paste it in the constant cell block with the variable name of '**BLOB_STORAGE_CONNECTION_STRING**'



The screenshot shows a Google Colab notebook titled 'Azure_Model_Deployment.ipynb'. The code cell contains the following Python code:

```
1 BLOB_STORAGE_CONNECTION_STRING = "DefaultEndpointsProtocol=https;AccountName=demodeeplearning;AccountKey=MpNgj+AH...  
2 BLOB_CONTAINER_NAME = ""  
3 DEEP_LEARNING_MODEL_PATH = "iris.h5"
```

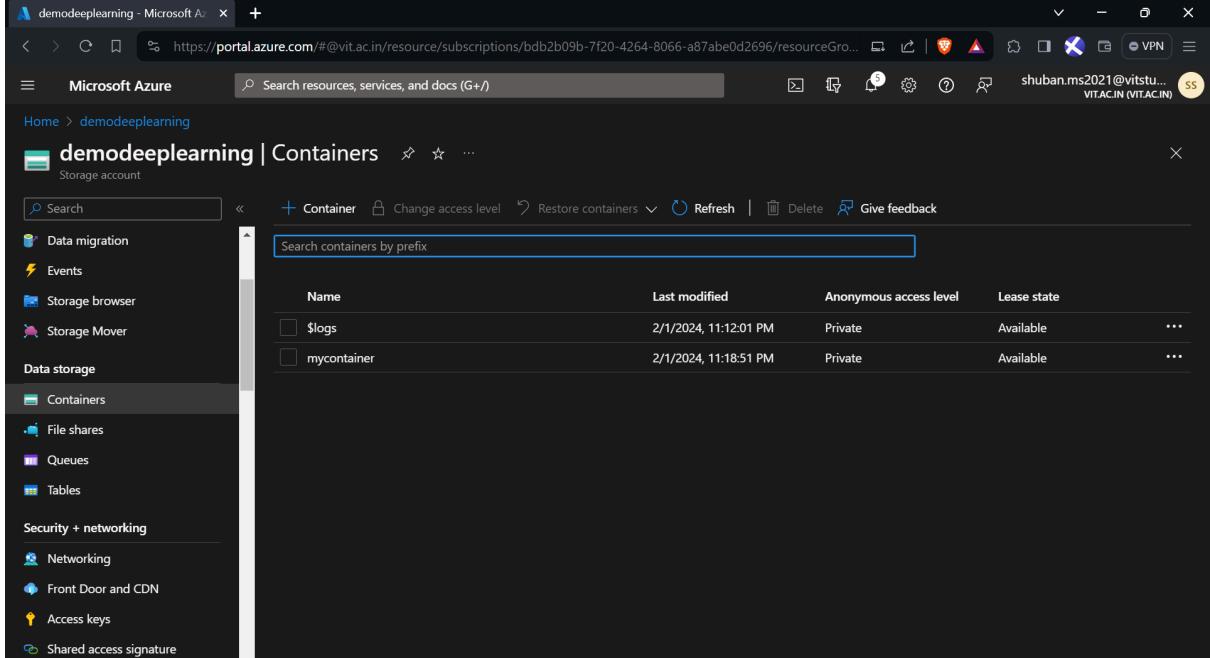
Below the code cell, there is a section titled 'Function to load uploaded model from azure' containing the following Python code:

```
[ ] 1 def loadBlobModel(  
2     filePath,  
3     connectionString,  
4     containerName  
5 ):  
6     blob_service_client = BlobServiceClient.from_connection_string(connectionString)  
7  
8     blob_client = blob_service_client.get_blob_client(container=containerName, blob=filePath)
```

The status bar at the bottom indicates 'Connected to Python 3 Google Compute Engine backend'.

Now go back to azure and then navigate to your container name where you had initially uploaded the h5 file

Which comes under the '**Data storage**' tab and '**Containers**' option on the left hand menu



The screenshot shows the Microsoft Azure portal with the URL 'https://portal.azure.com/#/vit.ac.in/resource/subscriptions/bdb2b09b-7f20-4264-8066-a87abe0d2696/resourceGro...' in the address bar. The left sidebar is expanded, showing 'Storage account' under 'Data storage' and 'Containers' under 'Containers'. The main content area displays a list of containers:

Name	Last modified	Anonymous access level	Lease state
\$logs	2/1/2024, 11:12:01 PM	Private	Available
mycontainer	2/1/2024, 11:18:51 PM	Private	Available

Whatever is the name given for this folder or container type the same in the colab notebook for the variable name '**BLOB_CONTAINER_NAME**'

The screenshot shows a Google Colab notebook interface. The title bar says 'AZURE DEPLOYMENT OF DEEP LEARN' and 'shubanms/DL-Azure-Deployment: Co'. The notebook file is 'Azure_Model_Deployment.ipynb'. The code cell contains:

```
1 BLOB_STORAGE_CONNECTION_STRING = "DefaultEndpointsProtocol=https;AccountName=demodeeplearning;AccountKey=MpNgj+AH
2 BLOB_CONTAINER_NAME = "mycontainer"
3 DEEP_LEARNING_MODEL_PATH = "iris.h5"
```

Below the code, there is a section titled 'Function to load uploaded model from azure' with the following code:

```
1 def loadBlobModel(
2     filePath,
3     connectionString,
4     containerName
5 ):
6     blob_service_client = BlobServiceClient.from_connection_string(connectionString)
7     blob_client = blob_service_client.get_blob_client(containerName, blob=filePath)
```

The status bar at the bottom indicates 'Connected to Python 3 Google Compute Engine backend'.

Now we are so close to finishing up :)

In this colab notebook run all the cell blocks and retrieve the model from azure through connection with the storage account

The screenshot shows a Google Colab notebook interface. The title bar says 'AZURE DEPLOYMENT OF DEEP LEARN' and 'shubanms/DL-Azure-Deployment: Co'. The notebook file is 'Azure_Model_Deployment.ipynb'. The code cell contains:

```
1 model = loadBlobModel(
2     DEEP_LEARNING_MODEL_PATH,
3     BLOB_STORAGE_CONNECTION_STRING,
4     BLOB_CONTAINER_NAME
5 )
```

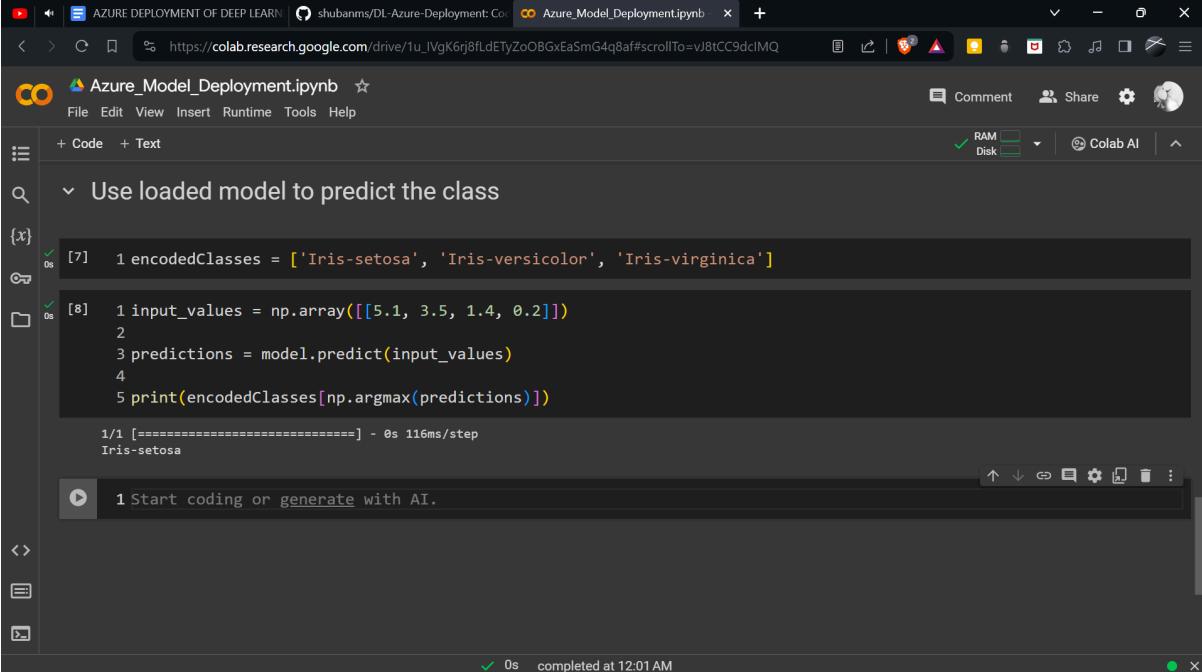
Below the code, the output shows the model summary:

```
Model: "sequential"
=====
Layer (type)      Output Shape       Param #
=====
dense (Dense)    (None, 10)          50
dense_1 (Dense)  (None, 10)          110
dense_2 (Dense)  (None, 3)           33
=====
Total params: 193 (772.00 Byte)
Trainable params: 193 (772.00 Byte)
Non-trainable params: 0 (0.00 Byte)
```

The status bar at the bottom indicates '1s completed at 12:00 AM'.

If all goes well and have followed each step then this code block should print out the summary of the model shown above

The final part is the application of the model in real time cases in this we try to predict the class of the flower from our input data and use the loaded model from azure



A screenshot of a Google Colab notebook titled "Azure_Model_Deployment.ipynb". The code cell contains the following Python code:

```
[7] 1 encodedClasses = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
[8] 1 input_values = np.array([[5.1, 3.5, 1.4, 0.2]])
2
3 predictions = model.predict(input_values)
4
5 print(encodedClasses[np.argmax(predictions)])
1/1 [=====] - 0s 116ms/step
Iris-setosa
```

The output of the code shows the prediction "Iris-setosa". The Colab interface includes a sidebar with file navigation, a search bar, and a code editor tab.

This brings us to the end of deployment and testing our own built model onto the cloud using Azure

Thank you all for following each step by step process, if there are any queries at any step or any difficulties please post your screenshot with a short description of what error you are facing in the group and we can help you out.