IBM Cloud

A Hands-on Introduction to Deep Learning and Neural Network Modeling with IBM Watson Studio

Training and Deploying a deep learning model for digit handwriting recognition

Lab Guide

Notices and Disclaimers

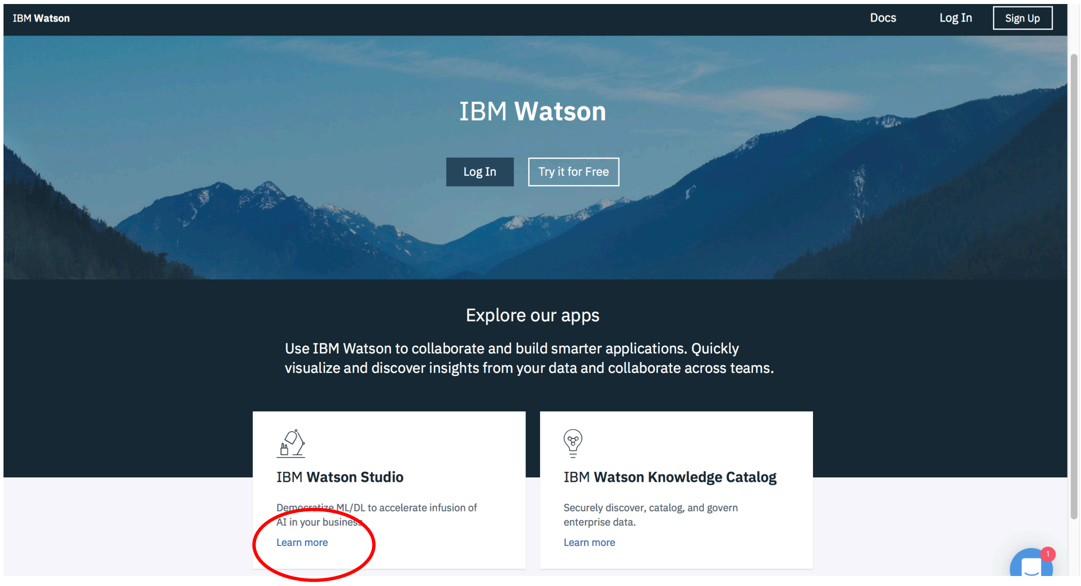
Table of Contents

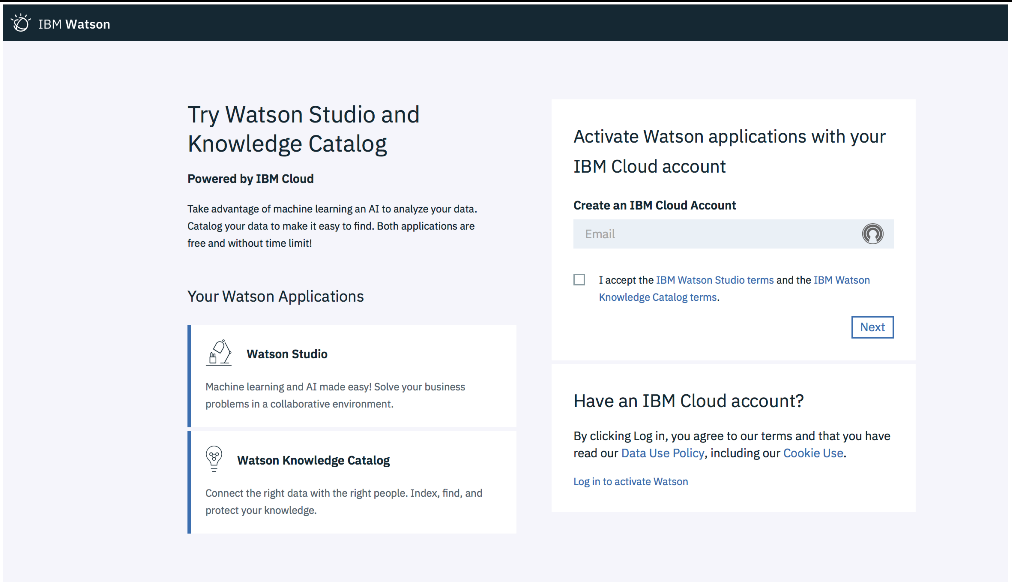
Lab Environment Overview

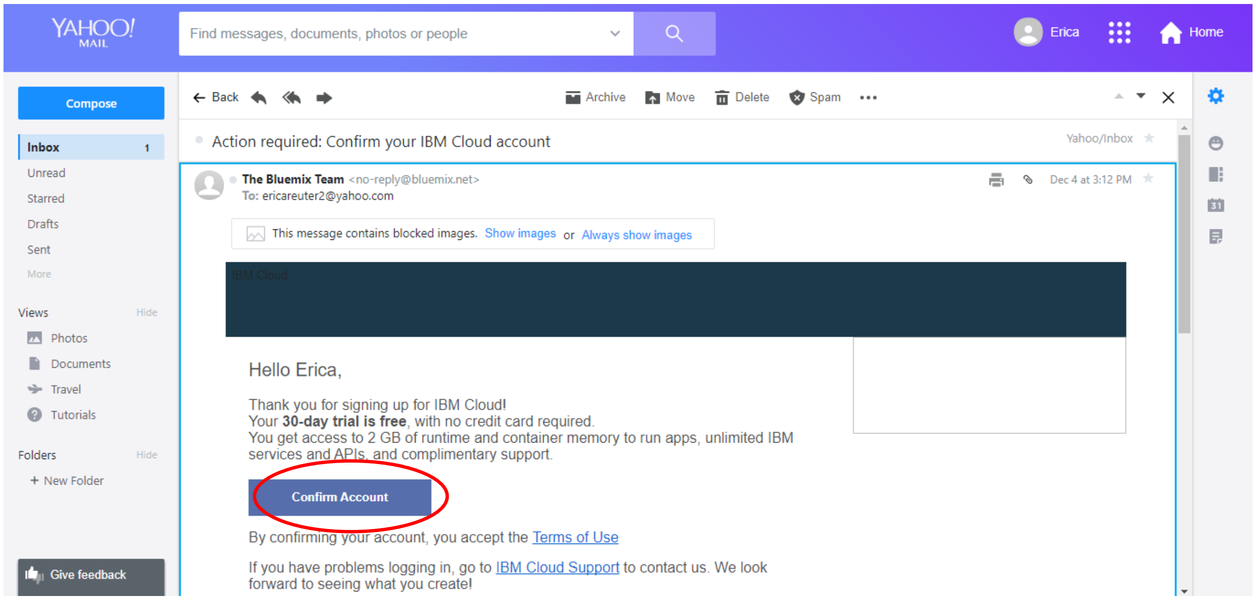
Purpose:	The purpose of this lab is to allow you to explore how you can develop and deploy deep learning models in Watson Studio. Specifically, this includes, • Working with a visual representation of a Neural Network example in the Model Flow editor • Training and optimizing your model using the Experiment Builder • Deploying the model to IBM Watson Machine Learning where it can be utilized by developers for use in applications For this lab, we will employ a standard MNIST dataset which is a sample of handwritten digits that will be used for the training and analysis or our model. By the end of this lab, you will be familiar with the following: • Navigating IBM Watson Studio • Creating and deploying Deep Learning Neural Network models in Watson Studio • Familiarity with IBM Cloud Services

Tasks:	Tasks you will complete in this lab exercise include: • Set up your IBM Account with Watson Studio • Creating a model for the MNIST data and running an experiment • Refinement and deployment of an existing TensorFlow model, using hyperparameter optimizations

Module 1: Create an IBM Cloud Account and Watson Studio Project- Lab Instructions

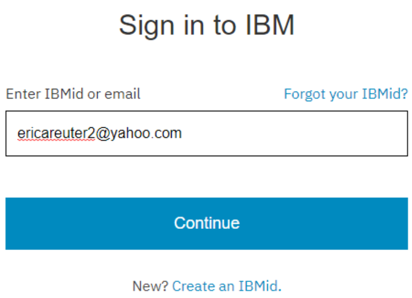
Step	Action
1	**<u>Set up your IBM Cloud Account with Watson Studio</u>** a. Log into IBM Data Science Experience at http://dataplatform.ibm.com/ b. Under IBM Watson Studio, click "Learn More". When the Watson Studio page appears, click "Start your Free Trial". c. Follow the instructions to create an account or use an existing IBM Cloud account. 

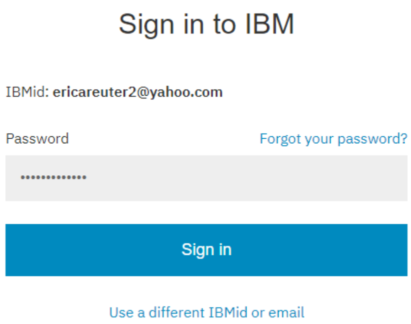
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td></td><td>


d. You will need to log into your email account to confirm and complete account registration.
e. Select the “Confirm Account” link in the email to be redirected to IBM Cloud.


</td></tr>
</table>

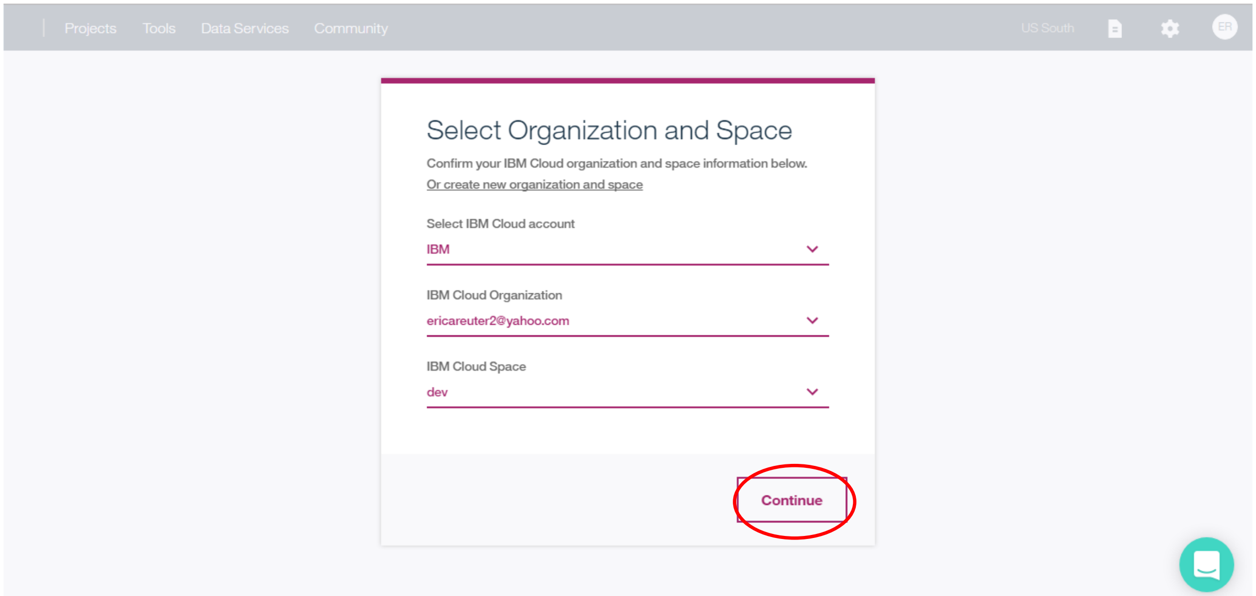
Step	Action



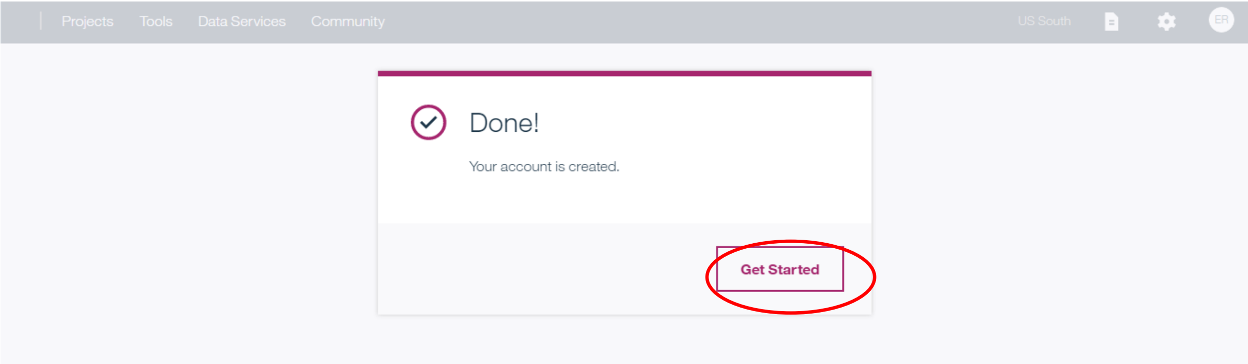


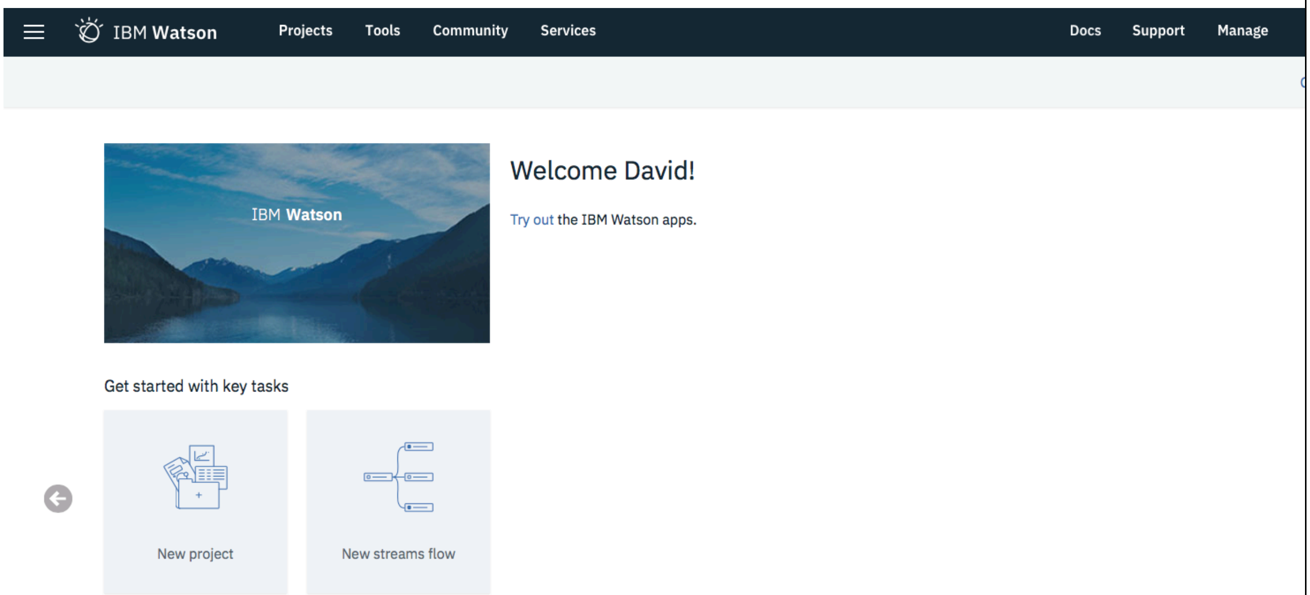
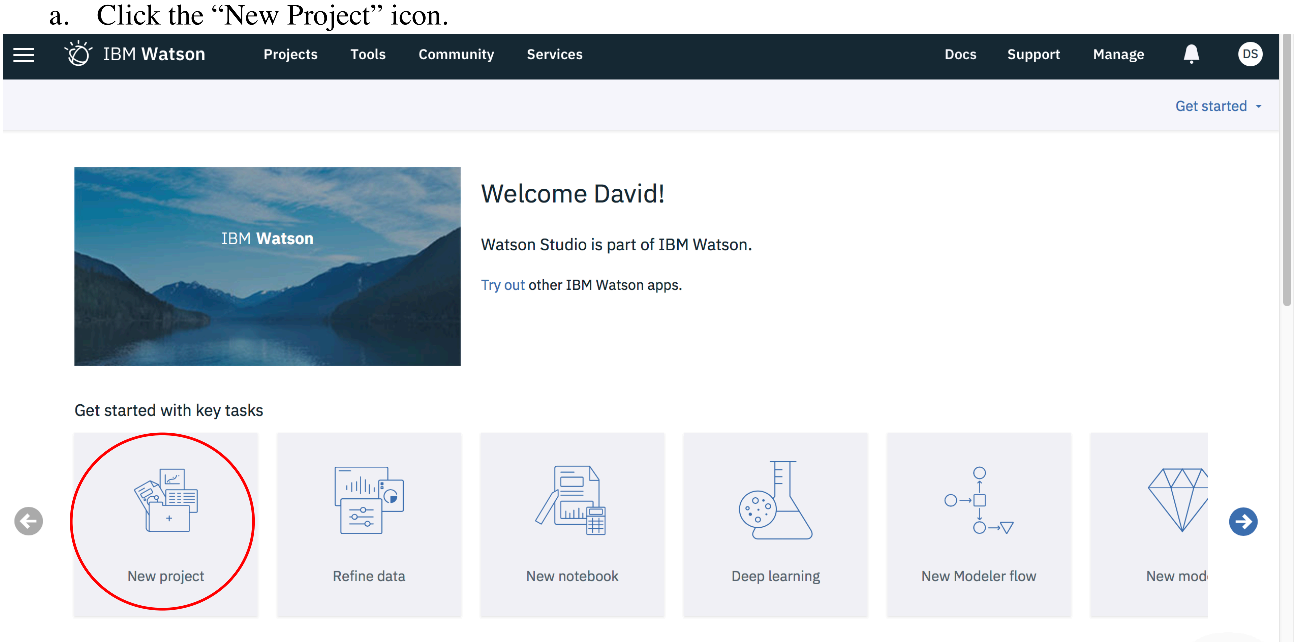
f. When you first sign into your new account, you will need to select and organization and space.

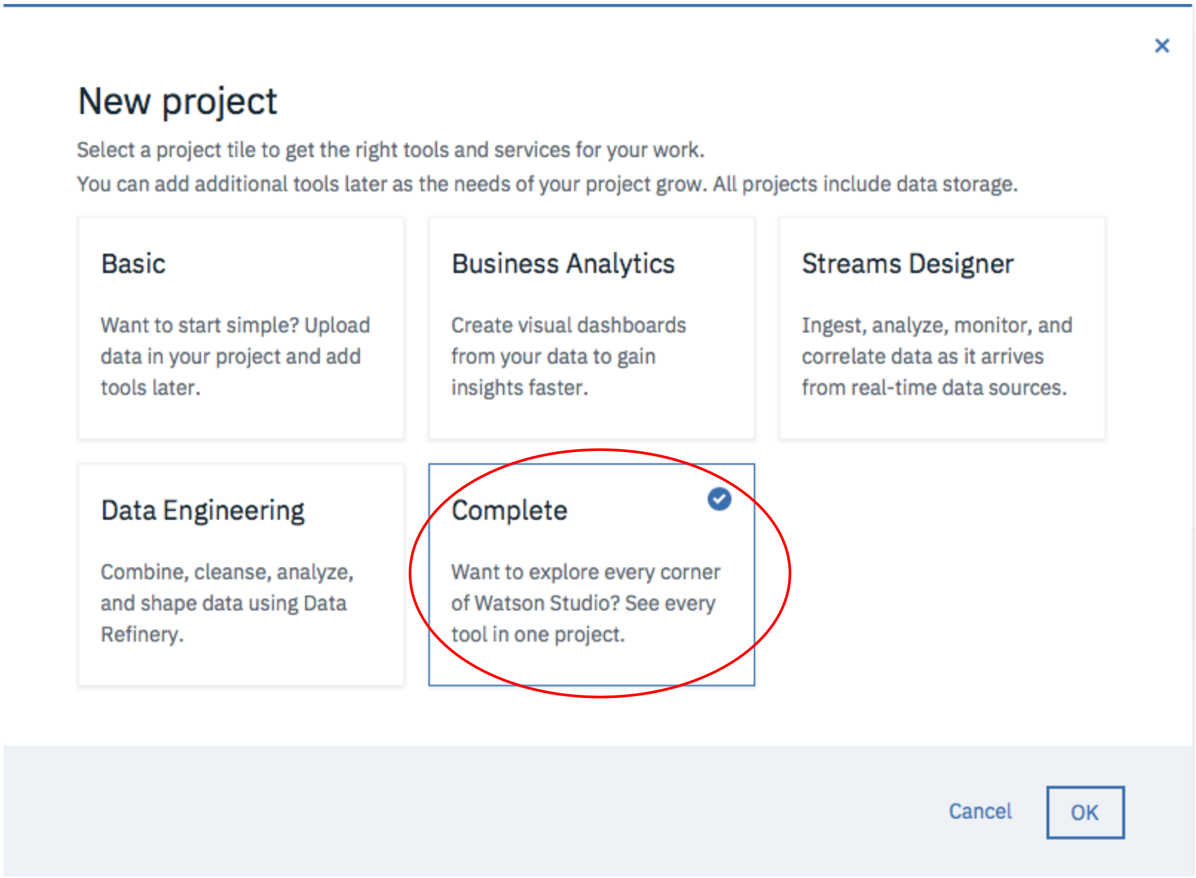
g. We will be using the default options for this lab. Click “Continue”.



h. Once you see that the Object-Storage is done provisioning, click “Get Started”.

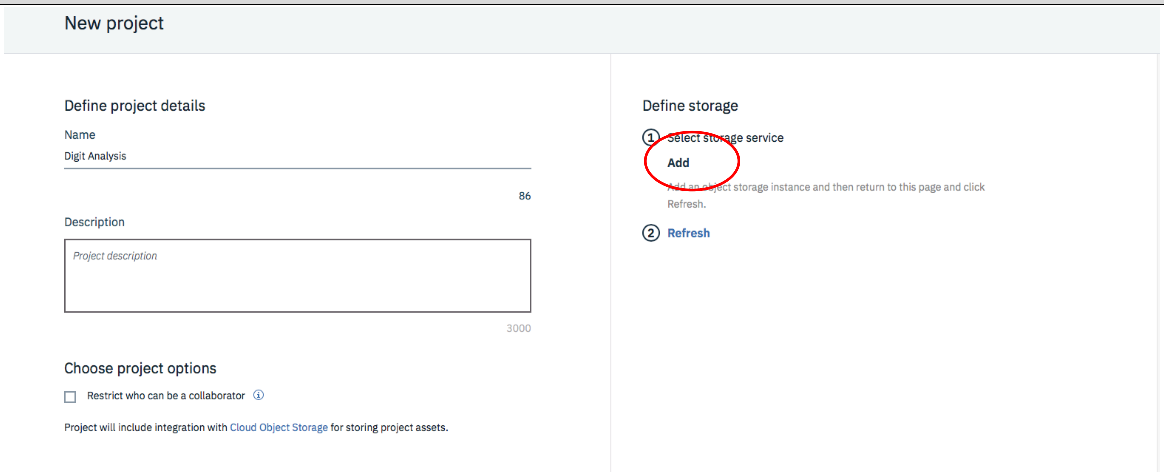


Step	Action
	
2	<u>**Create your Watson Studio Project**</u> a. Click the "New Project" icon.  b. For project type, select "Complete"

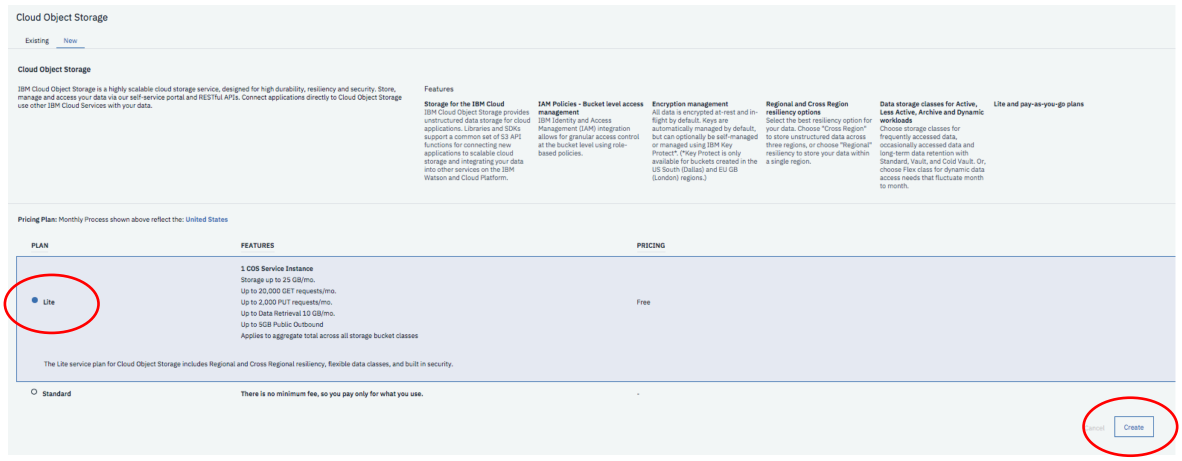
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td></td><td>


c. In order to create a project, you will need to have a place to store your project’s assets. Watson Studio projects utilize an associated IBM Cloud Object Storage (COS) services for this purpose. There is a free tier of COS that can be easily created. To do this, click “Add” under “Select Storage Service” on the new project page.
</td></tr>
</table>

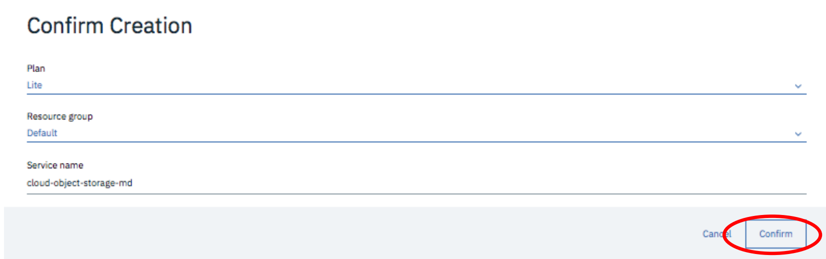
Step	Action



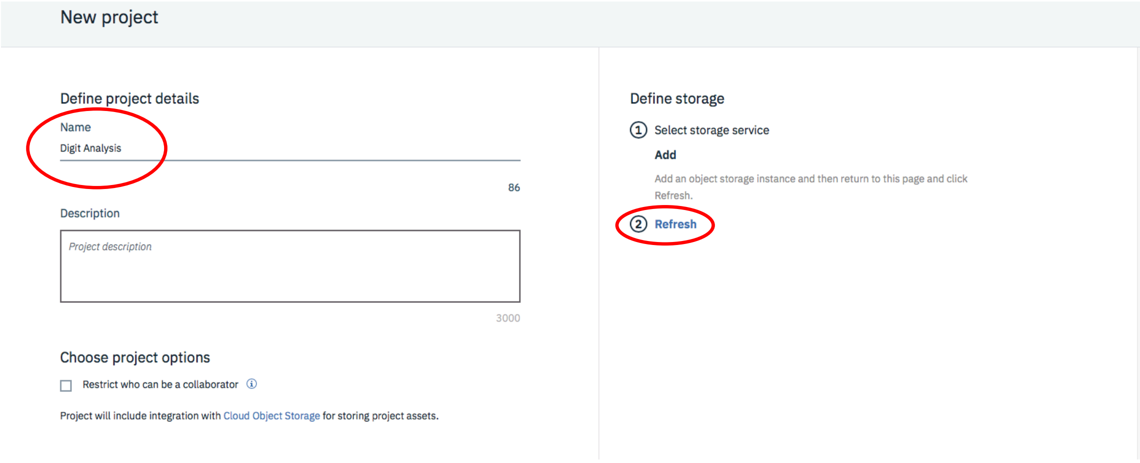
d. Select "Lite" for the service type and Click "Create"



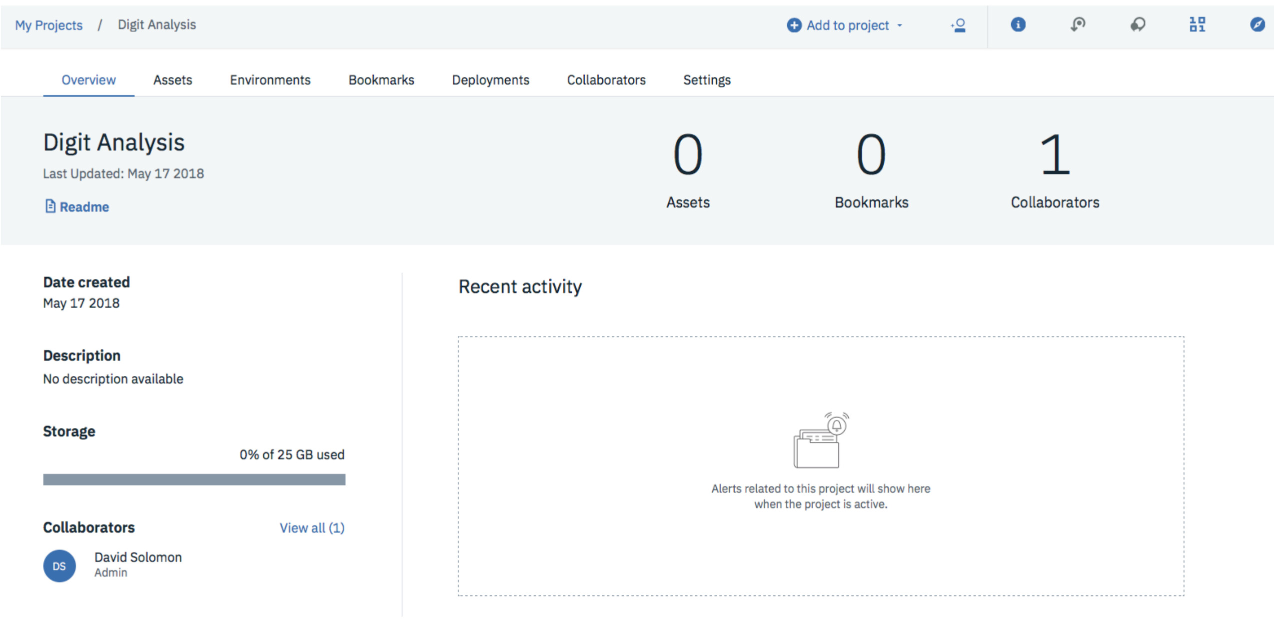
e. Click "Confirm".



f. Once back in the new project page, click "Refresh", enter "Digit Analysis" for the project name, and click "Create".

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td></td><td>


g. You have now successfully created your Watson Studio project. You should see your project's main page.



</td></tr>
</table>

Module 2: Create a Model Flow to Design a Neural Network for MNIST Data

In this module, you will use the Watson Studio Model Flow editor and Experiment Builder to design and build a single layer Neural Network model for the MNIST dataset.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>

Upload the Training and Test Data to the IBM Cloud

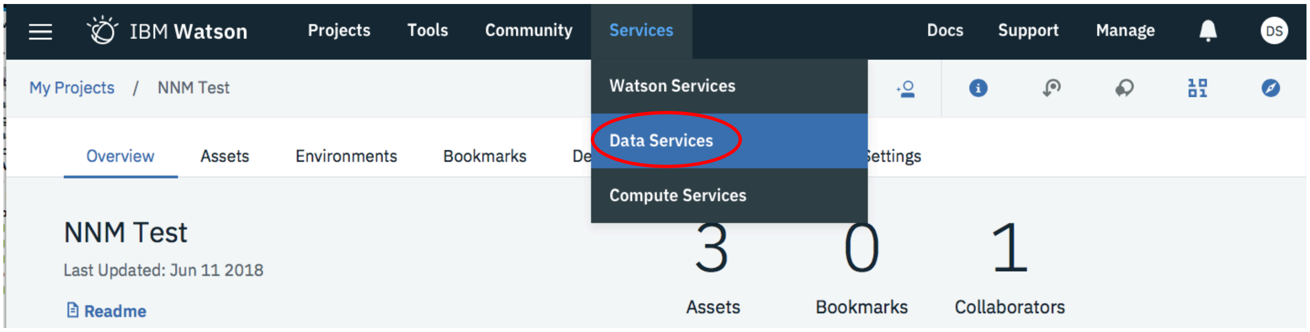
Before we can start working with our Neural Network model, we will need to obtain our training and testing data and make it visible to your project. This data consists of sample handwritten digits. Our Neural Network modeler requires that this data be located in Cloud Object Storage

a. Download the following files from this location:

http://ibm.biz/BdZNbX

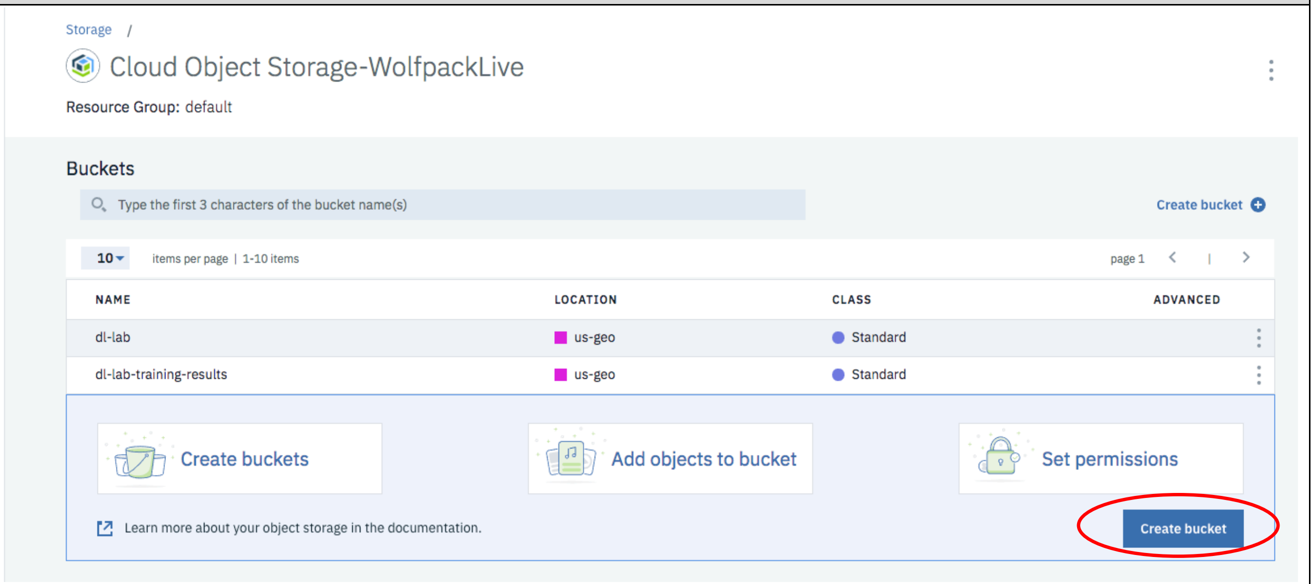
- mnist-tf-train.pkl
- mnist-tf-validate.pkl
- mnist-tf-test.pkl
- tf-mnist-test-payload-pkl.json

b. In Watson Studio, go to the “Services” menu at the top and select “Data Services”

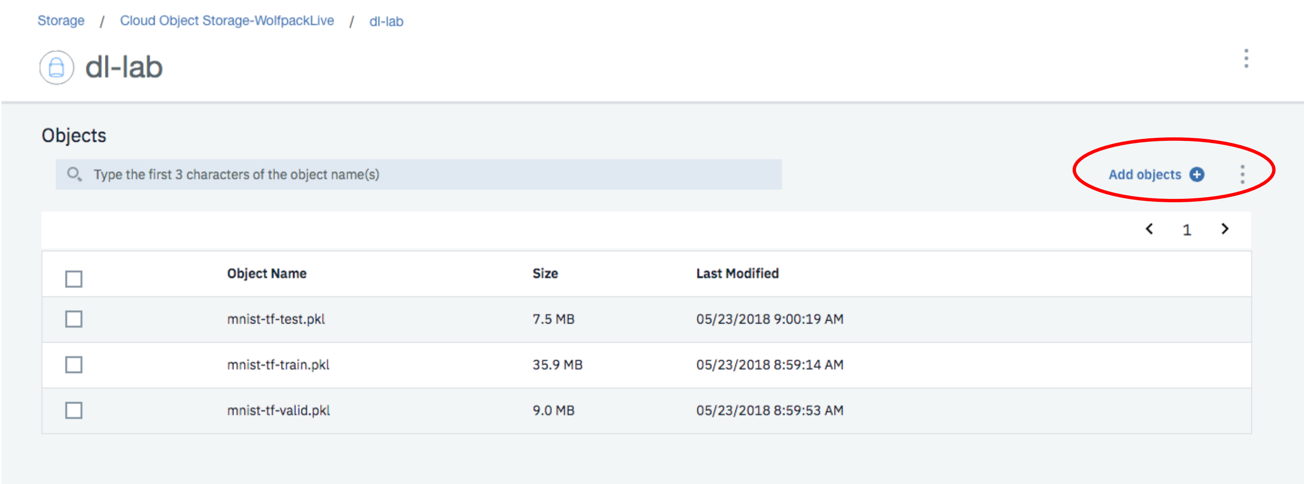


c. You will see your project’s Cloud Object Storage instance listed. Select this instance to go to the details page.

</td></tr>
</table>

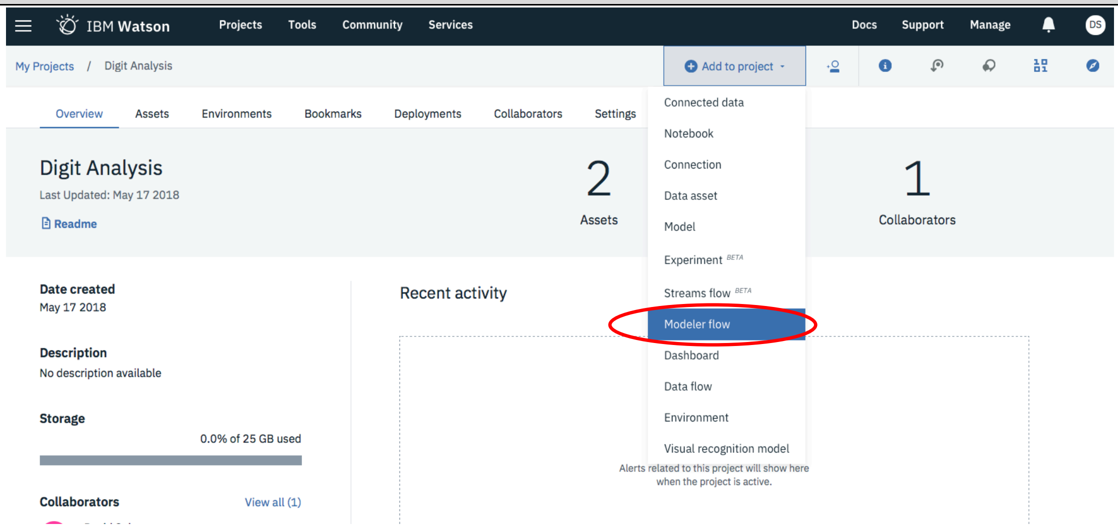
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td></td><td>


d. Using the "Create Bucket" button, highlighted above, create two buckets titled "mnist-training" and "mnist-training-results".
e. Select the "mnist-training" bucket, click on the "Add objects" button, and upload the three pkl files you previously downloaded to that bucket.

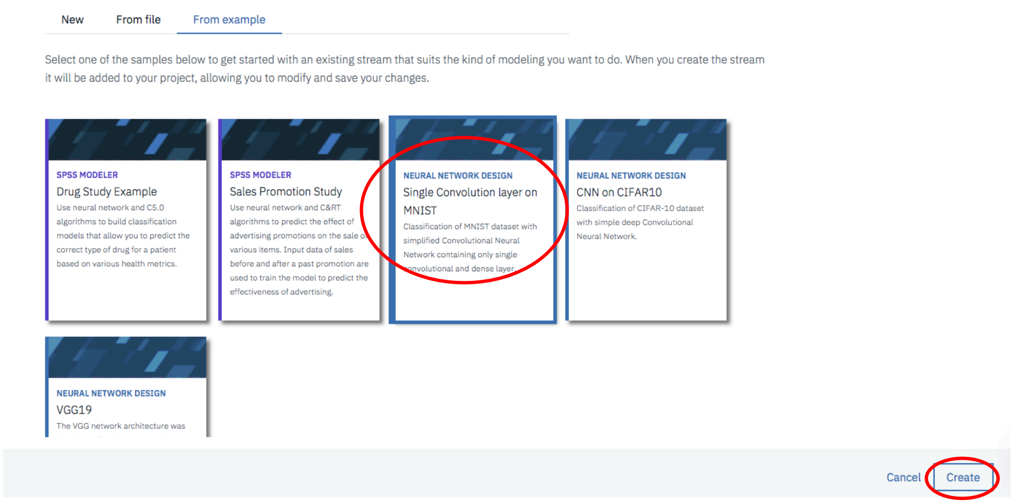


</td></tr>
<tr><td>2</td><td>
<u>Create a Model Flow</u>
a. From your project's overview page, click the "Add to Project" menu and select "Modeler Flow"
</td></tr>
</table>

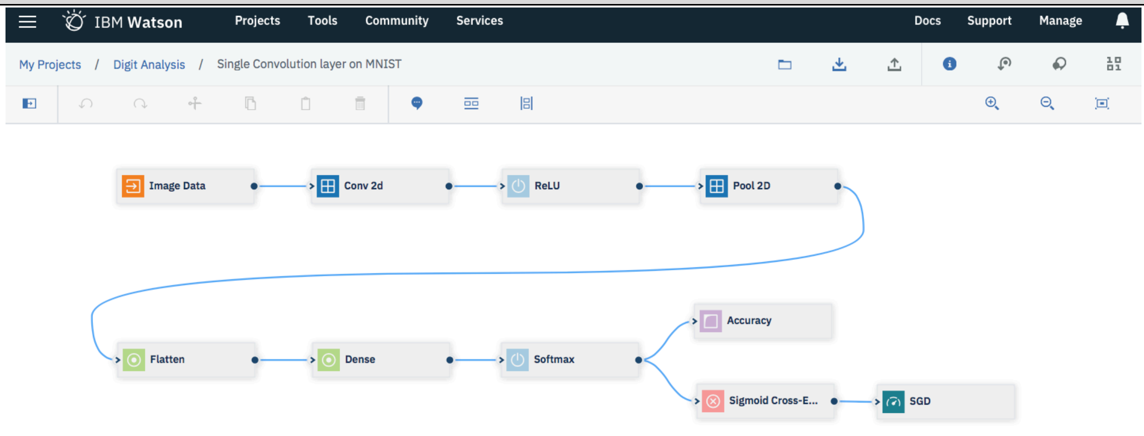
Step	Action



In Modeler Flow creation page, select "From Example". Select the "Single Convolution layer on MNIST" and click "Create".



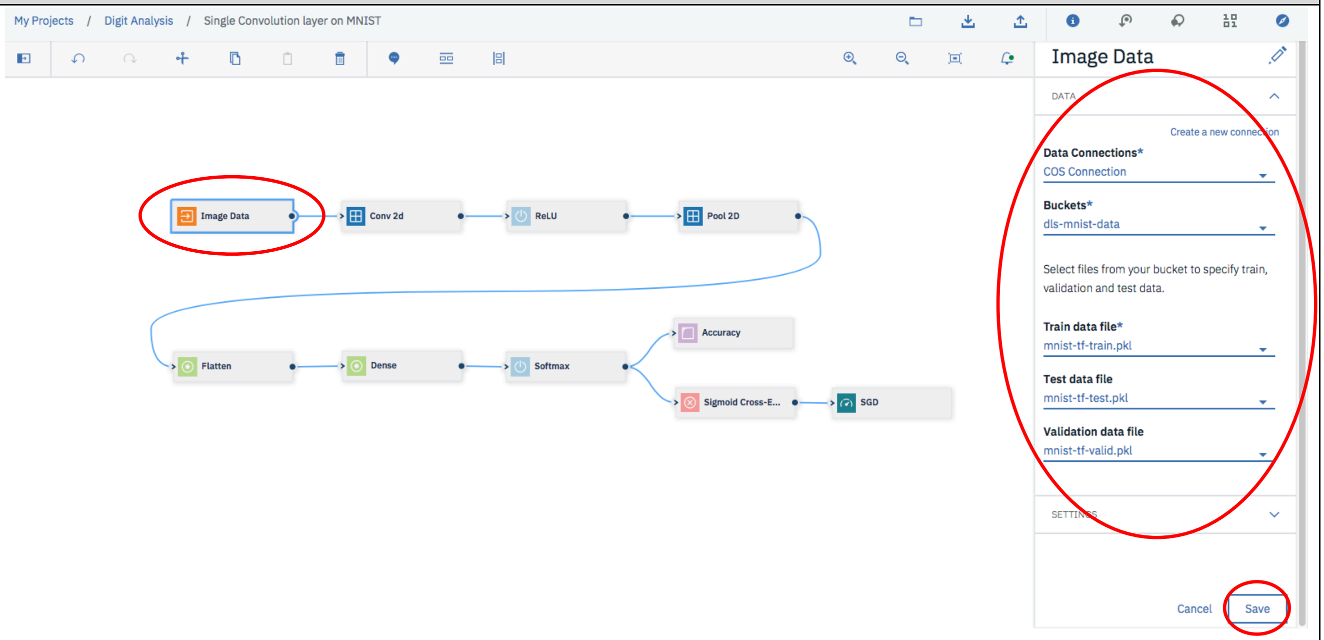
b. You will be brought to the model canvas that has been populated with our example model.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td></td><td>


c. In order to finalize this, we first need to point it to the data files the model will use for training and validation. Perform the following:

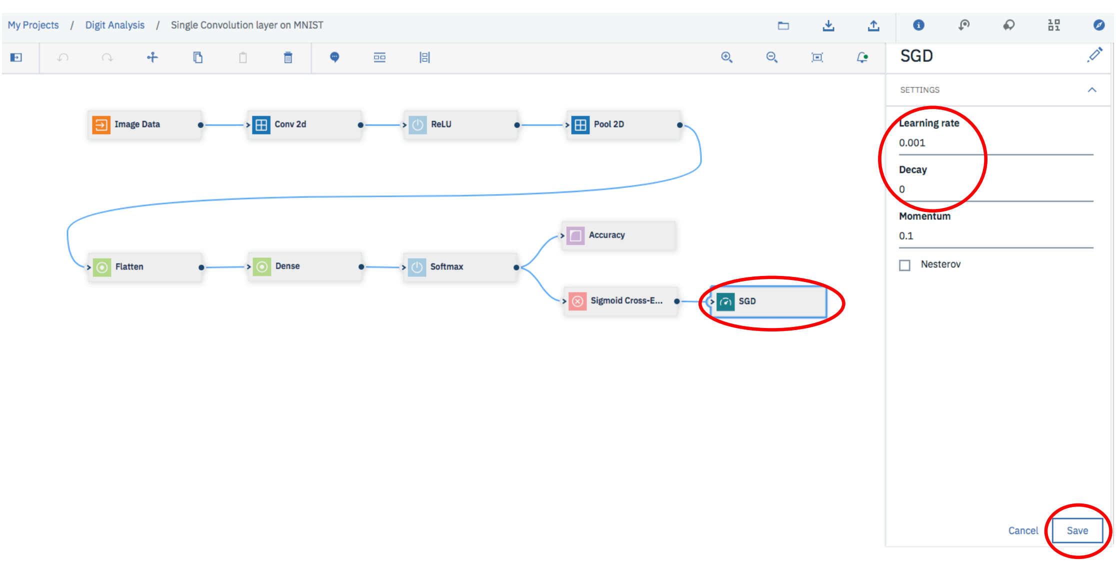
1. Double-click the first node, labeled “Image Data” (this opens the node details side panel.)
2. In the details side panel, in the “DATA” section, in the “Data Connections” drop-down menu, select the Cloud Object Storage connection that you created in the previous step.
3. In the “Buckets” drop-down menu, select the your training data bucket (“mnist-training”).
4. In the “Train data file” drop-down menu, select "mnist-train.pkl".
5. In the “Test data file” drop-down menu, select "mnist-test.pkl".
6. In the “Validation data file” drop-down menu, select "mnist-validate.pkl".
7. Click “Save”.
</td></tr>
</table>

Step	Action



d. Next, you will need to make some changes to the final node (SGD).

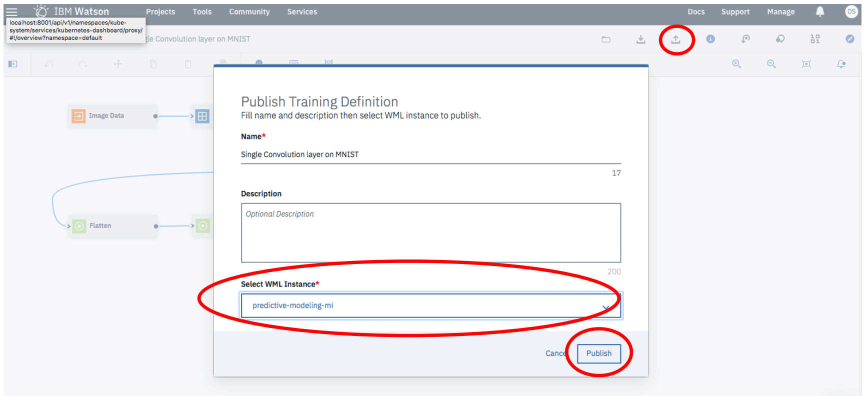
1. Double-click the final node, labeled “SGD” (this opens the node details side panel.)
2. Set “Learning rate” to 0.001
3. Set “Decay” to 0
4. Click “Save”.



e. You can now save our neural network design as a training definition.

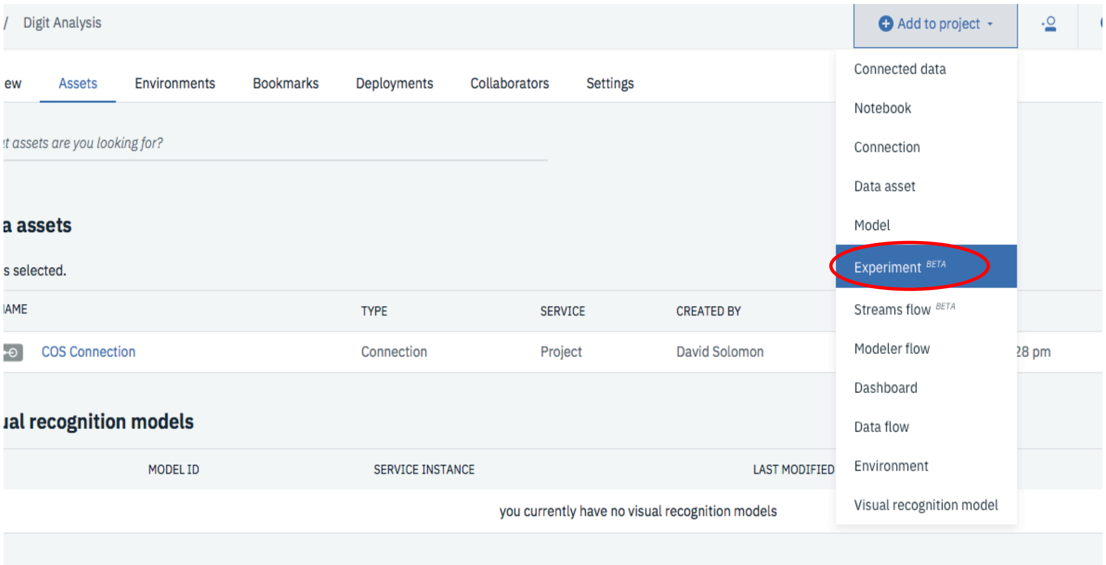
Step	Action

1. Click the publish icon ( ).
2. Specify a name for the training definition.
3. In the “Specify WML Instance” drop-down menu, select the Watson Machine Learning service instance that is associated with the project.
4. Click the “Publish” button.



3 **<u>Train your Model using the Experiment Builder</u>**

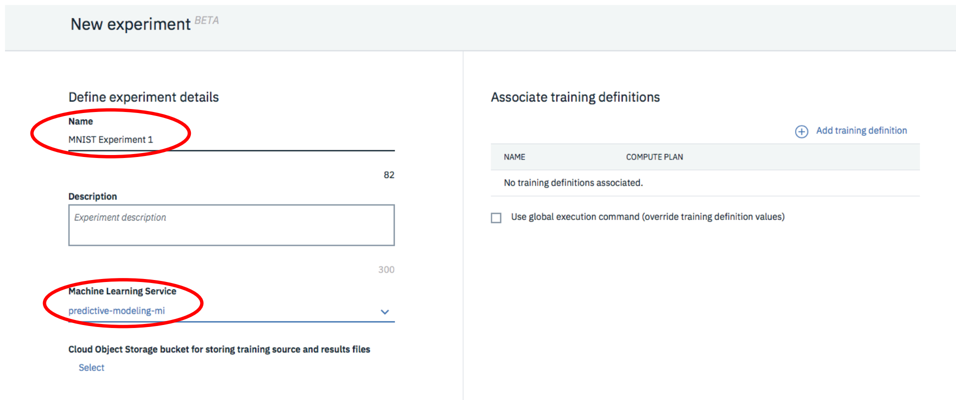
a. From your project’s overview page, click the “Add to Project” menu and select “Experiment”

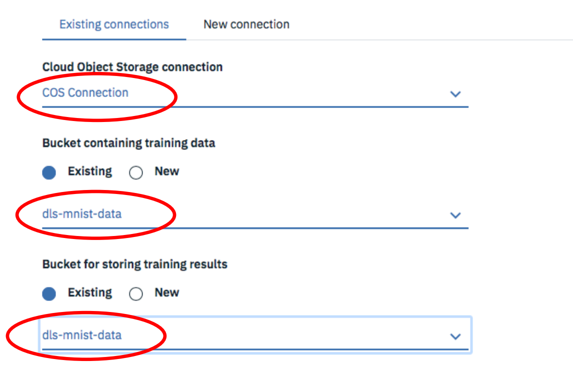


b. Complete the following in the new Experiment page:
   1. Name your experiment “MNIST Experiment 1”

Step	Action

2. In the “Machine Learning Service” drop-down, select the Watson Machine Learning instance associated with your project
3. In the area for “Cloud Object Storage” click “Select”.
4. In the “Cloud Object Storage connection” drop-down list, select the Cloud Object Storage connection you created before.
5. For the “Training Data Bucket”, select the training data bucket you created before.
6. For the “Training Results Bucket”, select the results bucket you created before.
7. Click “Create”

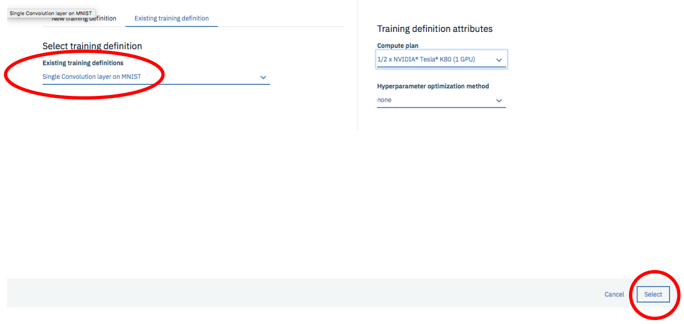




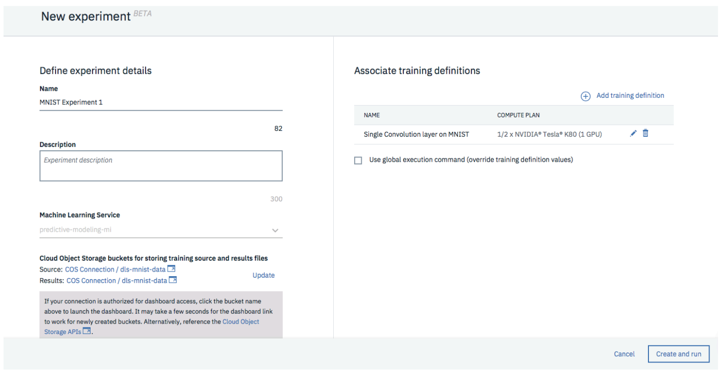
c. Add the training definition you created in the previous step.

1. Click **Add training definition**.
2. Click the **Existing training definitions** tab.
3. From the drop-down menu, select the training definition you published from the flow editor.
4. Select "1/2 x NVIDIA Tesla K80 (1 GPU)" for the compute plan.
5. Select "none" for the hyperparameter optimization method.
6. Click “Select”

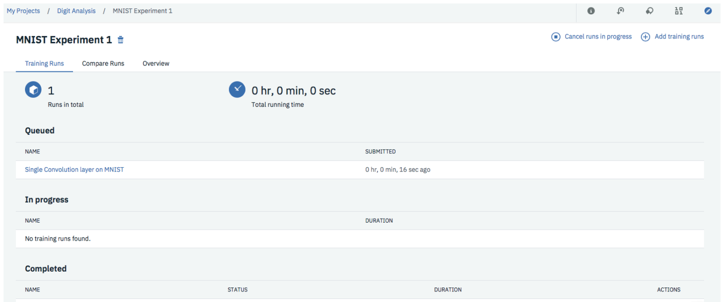
Step	Action



d. Click **Create and run**

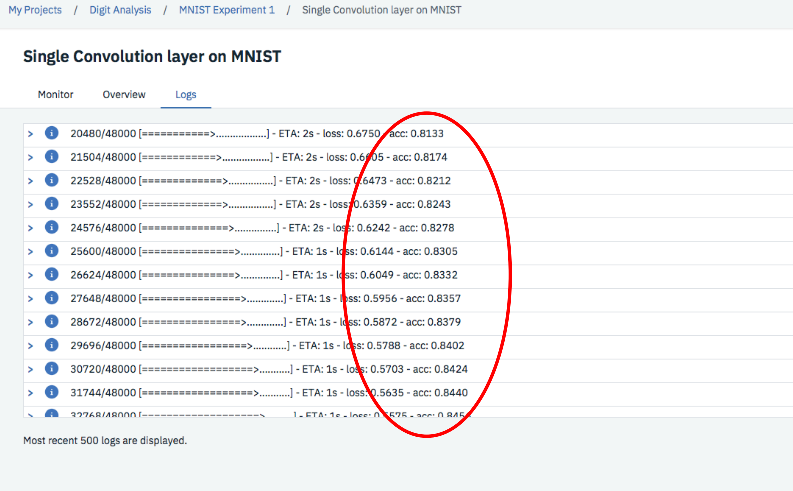


e. You can monitor the status of your training run.

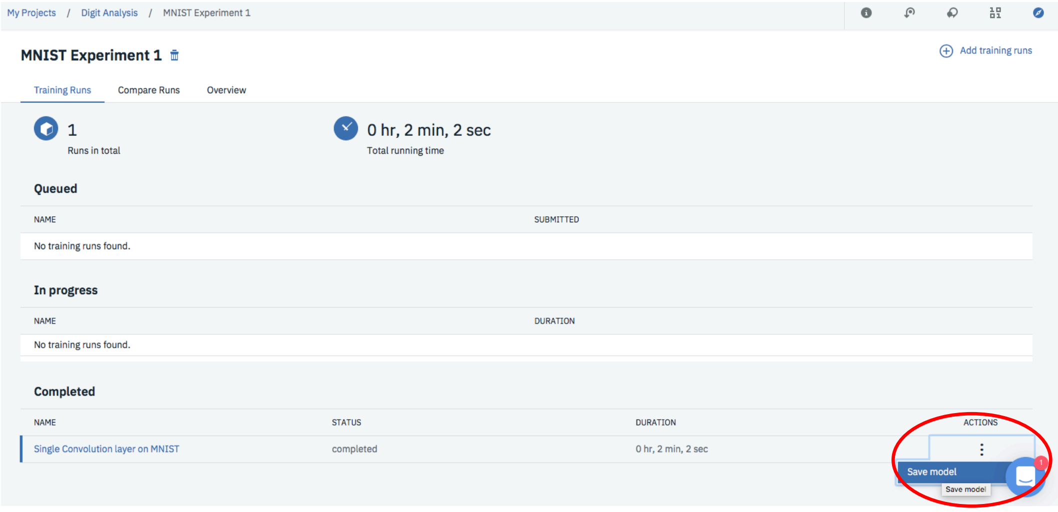


f. Once the training run is completed, you can review the details of the training run results and other output by clicking on the training definition.

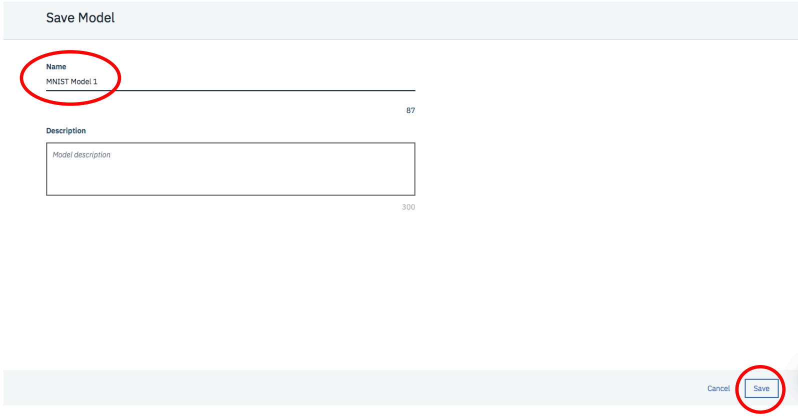
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td></td><td>



g. We will save the model, so it can be deployed into a Watson Machine Learning for testing and use by applications. To do, return to the Experiment page. Select the "Actions" menu for the training definition located in the "Completed" section of the page. From that menu, select "Save Model". Save the model with the name "MNIST Model 1" and click "Save".



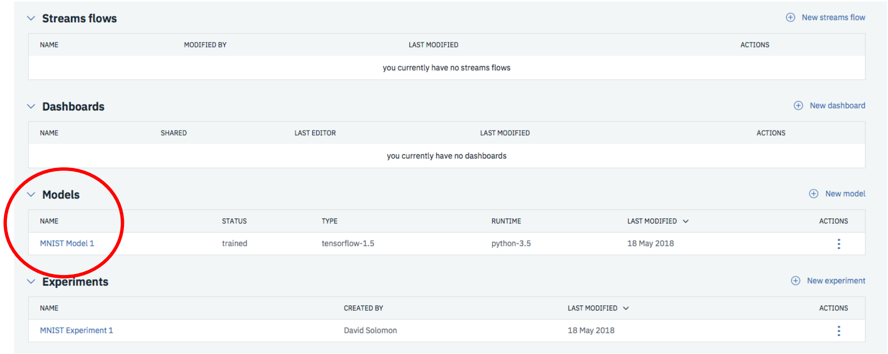
</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td></td><td>


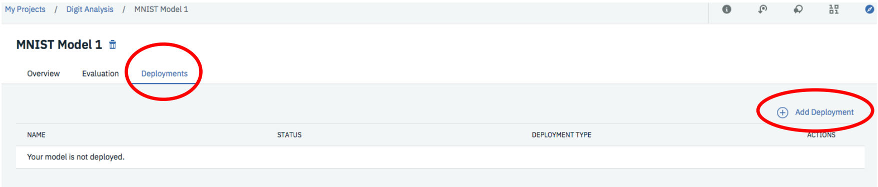
</td></tr>
<tr><td>4</td><td>

<u>**Deploy and Test your Model**</u>

a. We will now deploy the model into your Watson Machine Learning Service. Once the model is deployed, you can test it or have developers access it from their applications in production using a simple Rest URL. To deploy the model, return to the “Assets” page of your project. In the “Models” section, you should see your saved model listed. Click on the model.

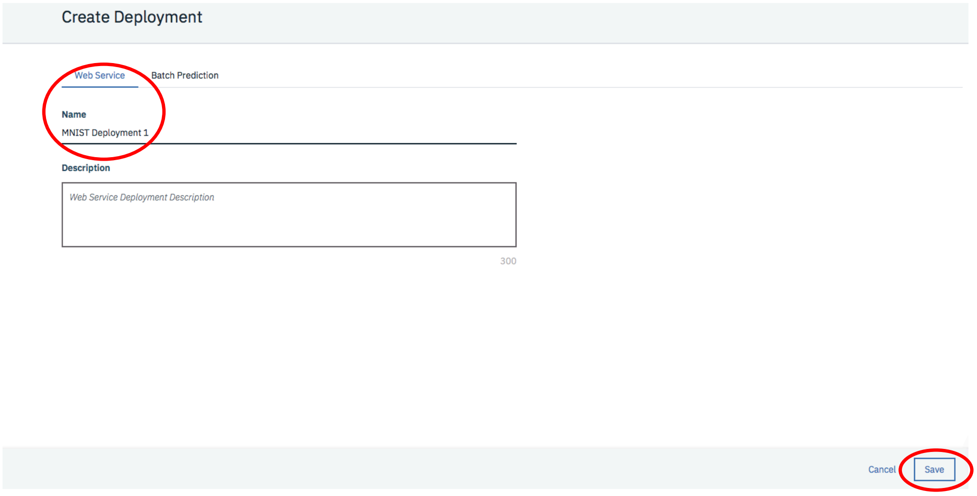


b. When the model opens, go to the “Deployments” tab and click “Add Deployment”

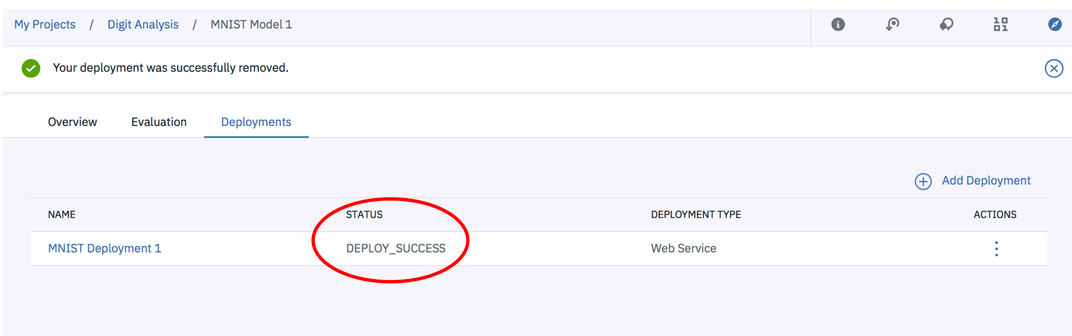


c. On the create deployment page, select the “Web Service” tab, name your deployment “MNIST Deployment 1” and click “Save”.
</td></tr>
</table>

Step	Action

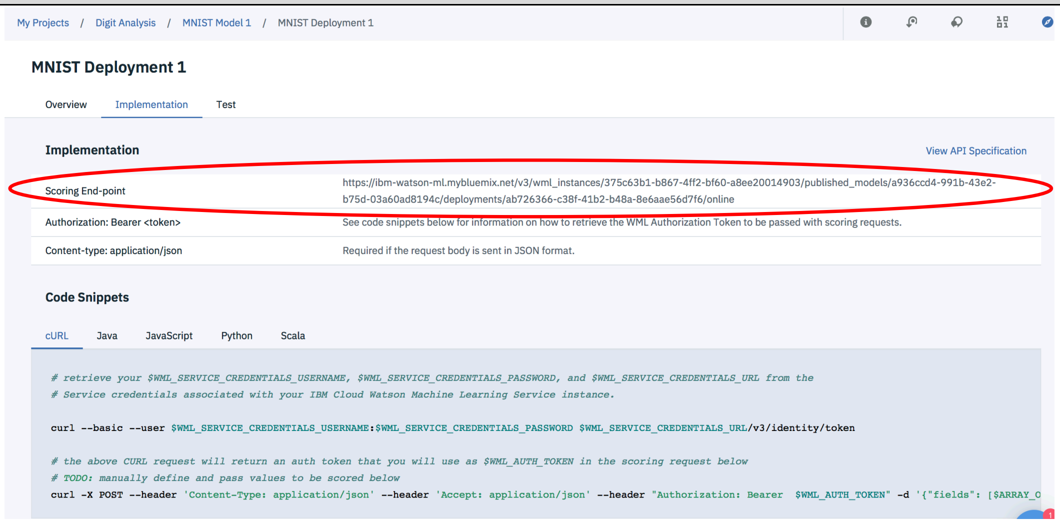


d. Wait until you receive an indication that the deployment was successful.



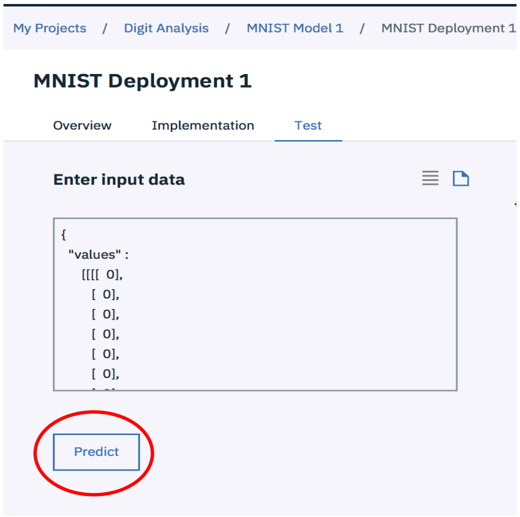
e. Once deployed, we can either test the deployment or provide a URL to developers for use in applications. Click on the deployment name to view the details. Next, click on the “Implementation” tab. Notice the scoring URL, as well as the sample code for accessing it.

Step	Action



f. Click on the "Test" tab. Copy and paste the contents of the "tf-mnist-test-payload-pkl.json" file you previously downloaded into the "Enter Input Data" field.

g. Click "Predict"



h. The JSON file contains an encoded image representing the number "7". Based on this, we would expect our model to correctly identify the image as a "7". The output of our team will be as shown below. In the output, each field represents a digit (0-9). The "0" or "1" is a true or false indication as to whether or not the image is a specific digit. Notice that the 8th entry

Step	Action
	is labeled as “1” meaning that the prediction is that our image is a “7”.


```
"fields": [
  "prediction"
],
"values": [
  [
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    1,
    0,
    0
  ]
]
```


Module 3: Building and Deploying a TensorFlow Model

In this module, you will use the Experiment Builder to train and deploy an existing TensorFlow model. In addition, you will use hyperparameter optimizations as part of the training process. These optimizations can be used to refine the performance of your model.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>

<u>**Download Sample Data and Model**</u>

a. Download the 4 MNIST sample data files from the following URL:

http://yann.lecun.com/exdb/mnist/

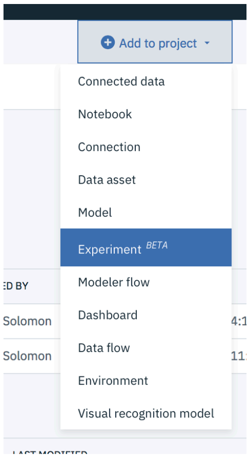
b. Download the example TensorFlow sample code from the following URL

http://ibm.biz/BdZNVi

</td></tr>
<tr><td>2</td><td>

<u>**Create and Run a New Experiment**</u>

a. Go to the overview page of your project. Click on the “Add to Project” menu and select “Experiment”.

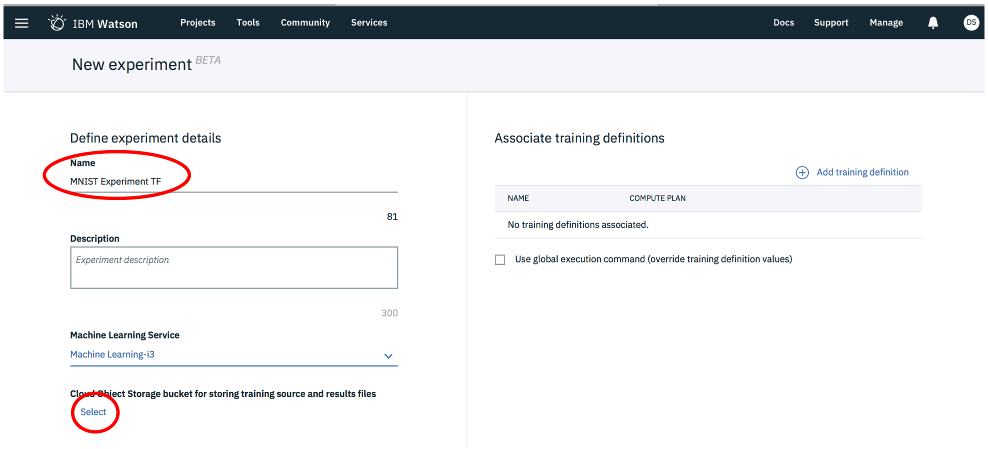


b. In the new experiment page, name your experiment “MNIST Experiment TF”.

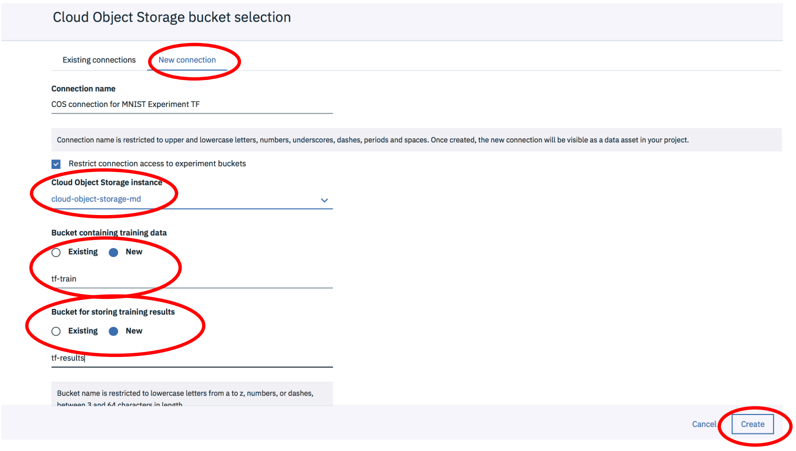
c. We will need to provide a location for the experiment to access training data and to store results. We use Cloud Object Storage for this purpose. Click on “Select” under the Cloud Object Storage Bucket area.

</td></tr>
</table>

Step	Action

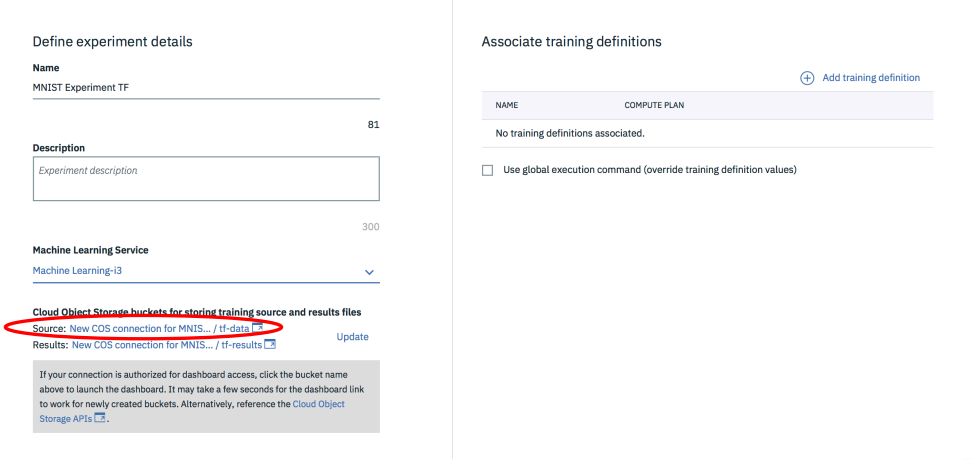


d. In the Cloud Object Storage page, select "New Connection" to create a connection to your project's default COS instance.
e. From the list, select your existing COS instance.
f. Click on "New" for both your training and results bucket selection.
g. For you training data bucket name, enter "tf-data" and for training results enter "tf-results"
h. Click "Create"

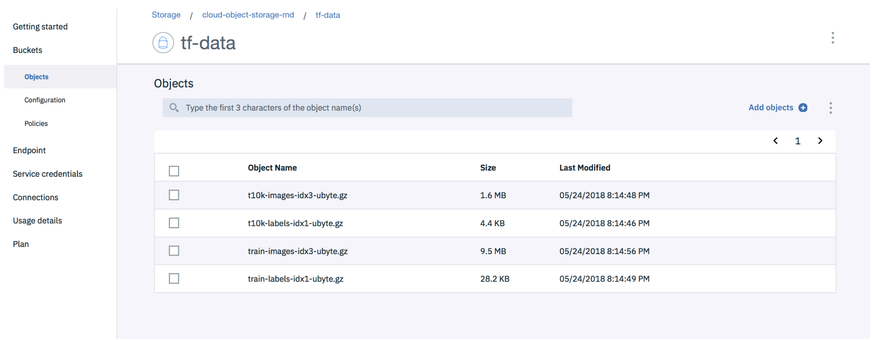


i. Once the experiment is created, click on the link to the "source" bucket, as shown below. This will open the COS bucket so you can add the training data.

Step	Action



j. Once the bucket is open, click on the "Add objects" link and select the MNIST files you downloaded earlier. The files will then be uploaded to the bucket.

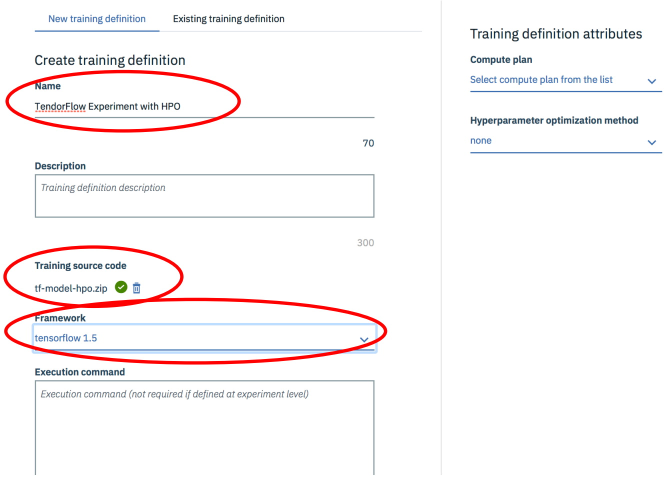


k. We can now add our TensorFlow training definition code to our experiment. Return to the experiment page and click "Add Training Definition". Enter "TensorFlow Experiment with HPO" as the name.

l. Click "browse" and select the training definition file you previously downloaded.

m. Select "TensorFlow 1.5" as the framework

Step	Action

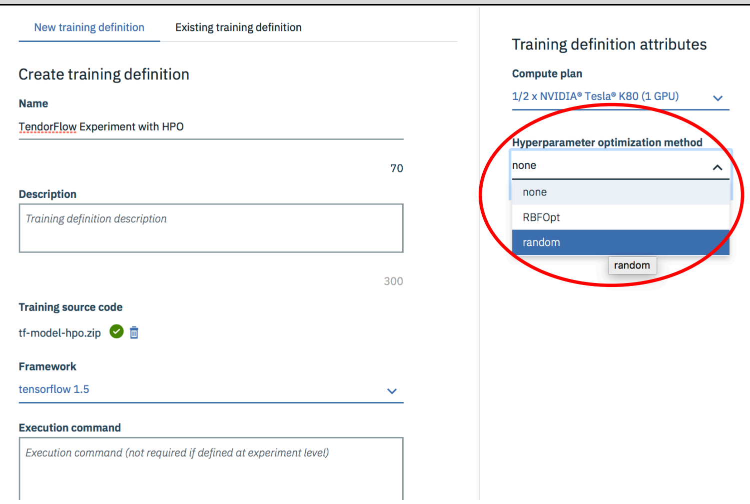


n. For “Compute Plan”, select “1/2 x NVIDIA Tesla K80 (1 GPU)”.
o. In the “Execution command” field, enter the following,

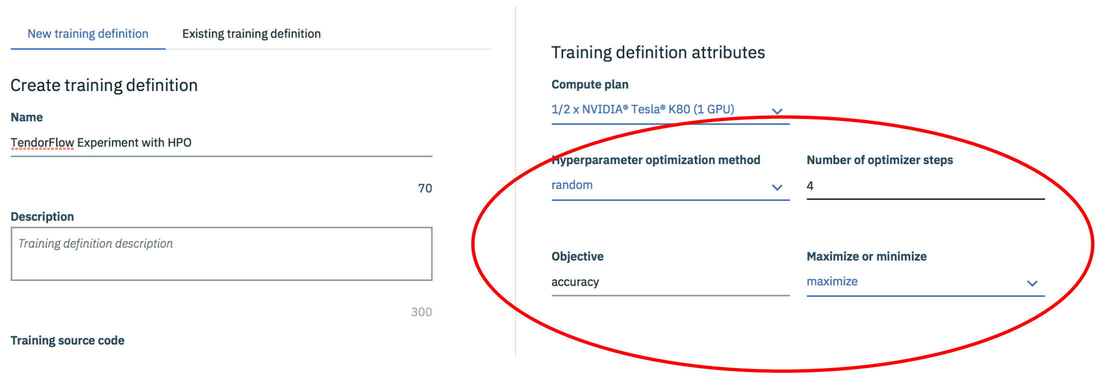
```
python3 convolutional_network.py --trainImagesFile
${DATA_DIR}/train-images-idx3-ubyte.gz --trainLabelsFile
${DATA_DIR}/train-labels-idx1-ubyte.gz --testImagesFile
${DATA_DIR}/t10k-images-idx3-ubyte.gz --testLabelsFile
${DATA_DIR}/t10k-labels-idx1-ubyte.gz --trainingIters 200000
```

p. This training definition allows for the use of specific hyperparameter optimization (HPO) to enable us to assess the model, based on different conditions and assumptions. When the experiment is run, it can run multiple instances, each with a different HPO values to determine which combination of values provide the best performance and accuracy. For this lab, we will specify value ranges for 4 HPOs. To start, click on the “Hyperparameter optimization method” menu and select “Random”.

Step	Action

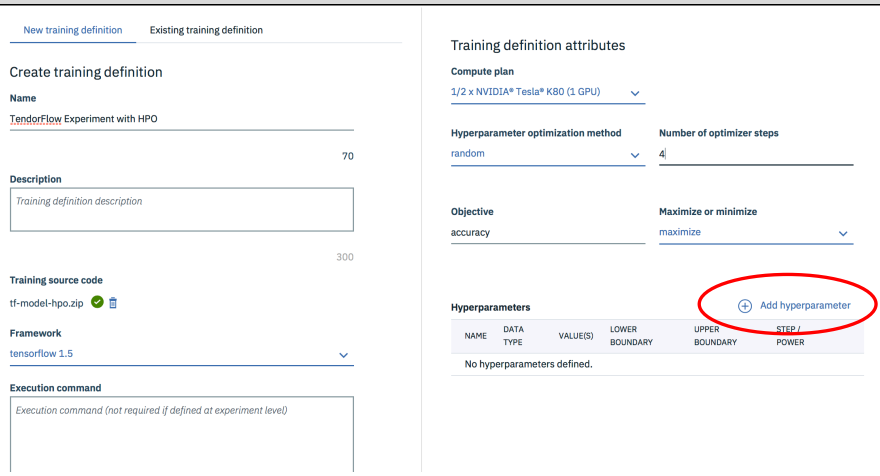


q. Additional options will now appear. Set the "number of optimizer steps" field to "4", "Objective" to "Accuracy" and "Maximize or Minimize" to "maximize".



r. We will now specify the HPO values that will be used for this experiment. To do this, click "add hyperparameter".

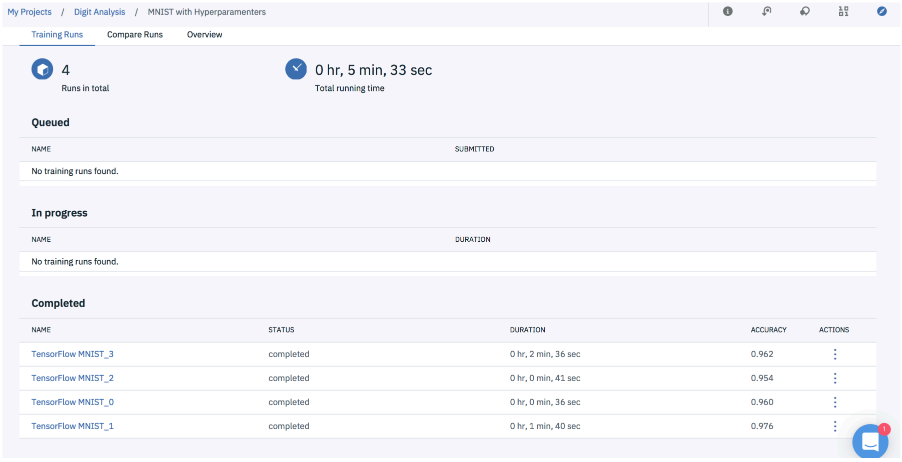
Step	Action



s. Add the following hyperparameters, as specified in the following table.

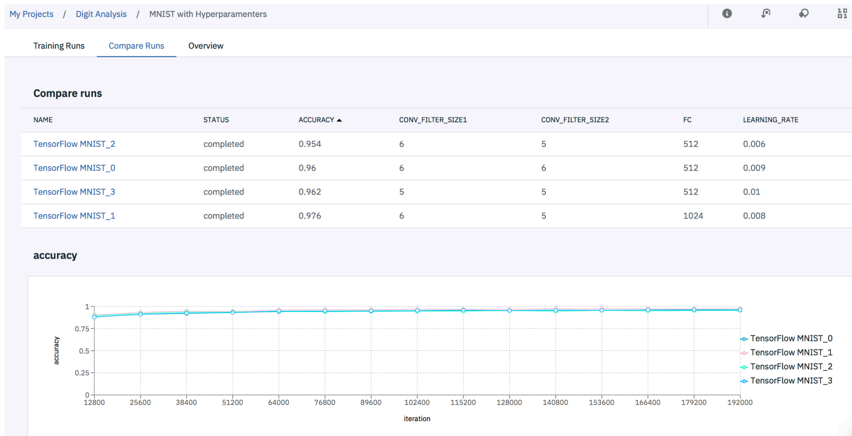
Name	Lower bound	Upper bound	Step/Power	Data type
learning_rate	0.005	0.01	Step: 0.001	Double
conv_filter_size1	5	6	Step: 1	Integer
conv_filter_size2	5	6	Step: 1	Integer
fc	9	10	Power: 2	Integer

t. Click "Create and Run" to run the experiment. The experiment control page will appear. Please note that is may take 5-10 minutes for your experiment to run.



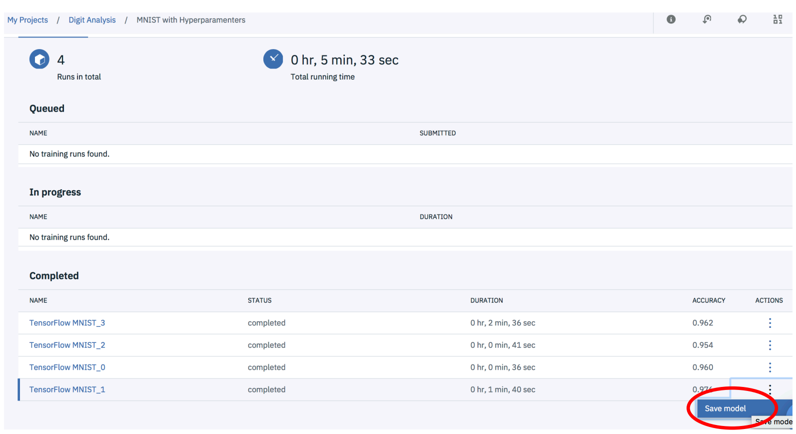
Step	Action
3	**Review Experiment Results**

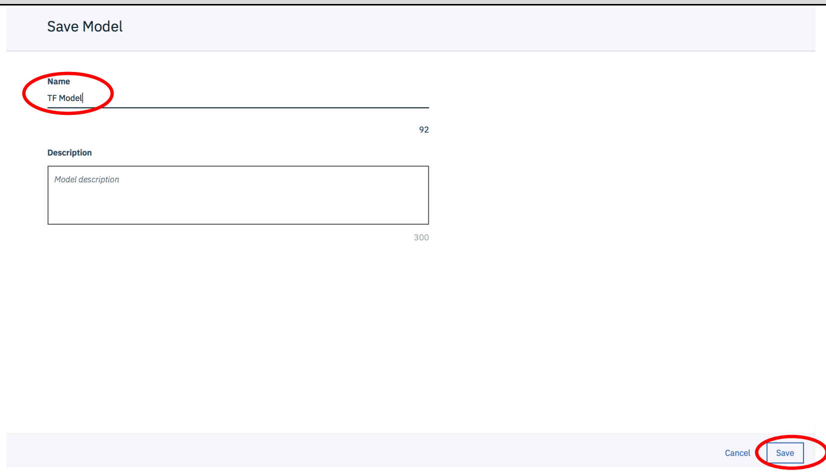
a. Once the experiment is complete, select the “Compare Runs” tab.



b. Note here that all four runs had very similar results in this case. Also note that each run used a different set of hyperparameter values, since we asked for a “random” allocation. We can also view the details of a specific run by clicking on its link.

c. Click on the “Training Runs” tab. From here, we can create a model from any of these runs based on the results. Click on the “Actions” menu for the run with the highest accuracy and select “Save Model”. Name the model, “TF Model” and click “Save”.



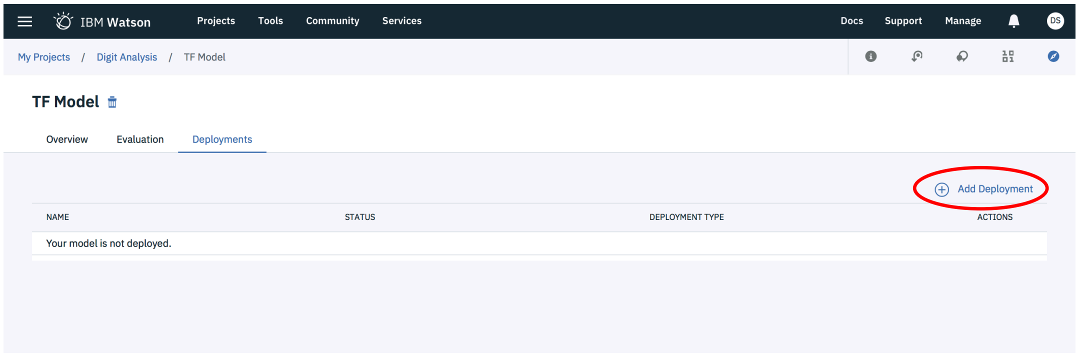
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td></td><td>


</td></tr>
<tr><td>4</td><td>

<u>Deploy and Test the TensorFlow Model</u>

a. We can now deploy the model using the same process as in Module 2. To do this, return to the “Assets” page of your project. Under the “Models” section, select the model you just saved.

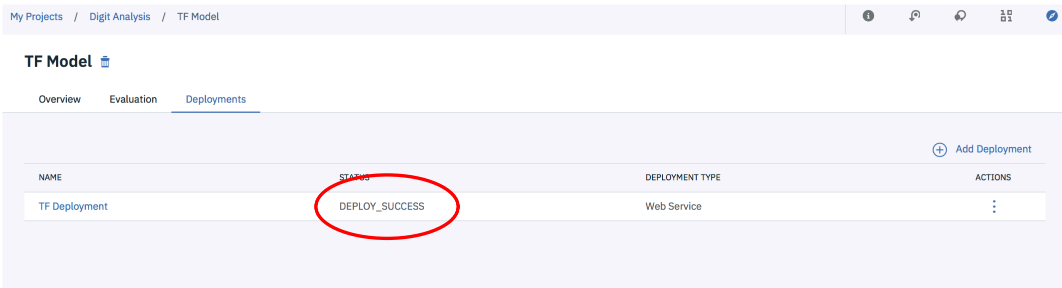
b. On the model overview page, select the “Deployments” tab. Click on “Add Deployment”.



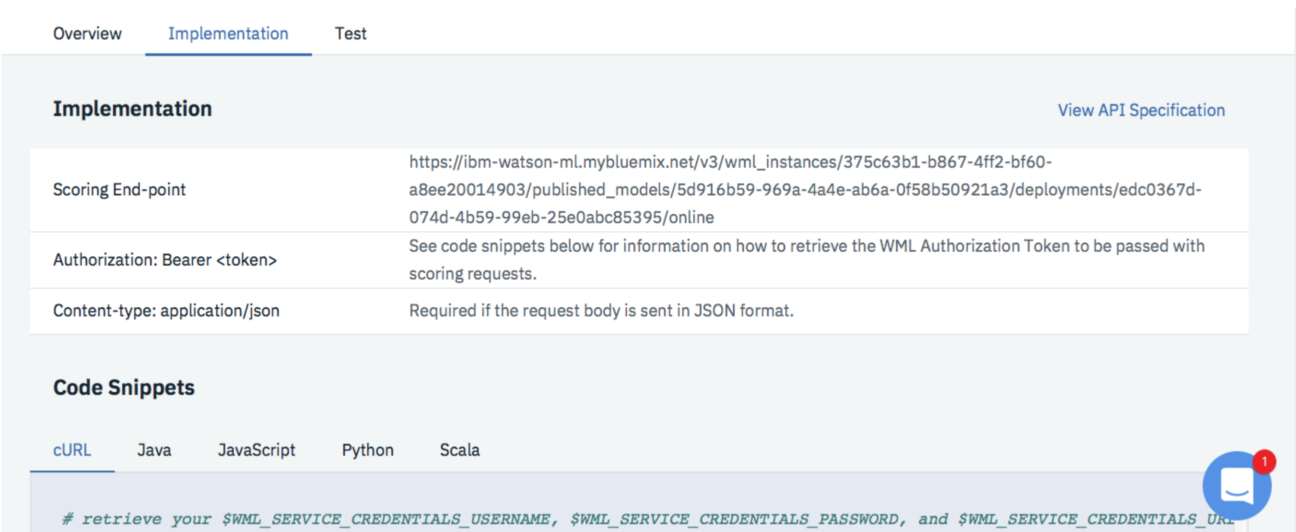
c. On the deployment page, enter “TF Deployment” as the deployment name and click “Save”.
</td></tr>
</table>

Step	Action

d. Your model will be deployed when the status changes to “DEPLOY_SUCCESS”



e. Click on the deployment name to view the details. Next, click on the “Implementation” tab. As with the previous lab exercise, we have generated a URL that developers can use in their applications to access the model, as well as example code.



f. Finally, we can test the model. Click on the “Test” tab. Paste the contents of the following file to the “Enter Input Data” field and click “Predict”.

http://ibm.biz/BdZNWF

g. In this case, the input data is the binary code for images of the numbers 4 and 5. Therefore, the output from this test should be as shown below.

Step	Action
	

Conclusion

In this lab, we learned how Watson Studio can facilitate the development and deployment of Deep Learning models and applications in a highly collaborative and open environment. We welcome you continue to explore this other capabilities of Watson Studio.

Thank you for your time today!!

The IBM Wolfpack Team