Build, Train, Save, Deploy and Test a Convolutional Neural Network Model using MNIST

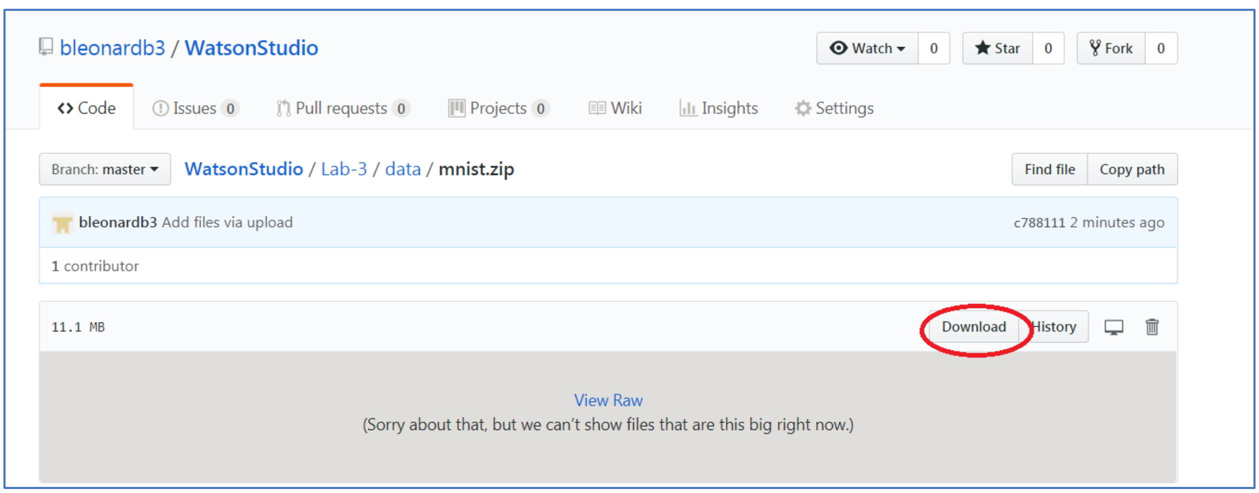
This lab will use the [MNIST]() computer vision data set to train a deep learning model to recognize handwritten digits. A single layer convolutional neural network will be built in the Watson Studio neural network designer, and then trained using the Watson Studio Experiment Builder. The trained model will be saved in the model repository, deployed, and then tested with sample image data. The lab consists of the following steps:

1. Set up the data files in IBM Cloud Storage.
2. Design the neural network
3. Train the model
4. Monitor the training progress and results
5. Save and Deploy the Trained Model
6. Test the Deployment

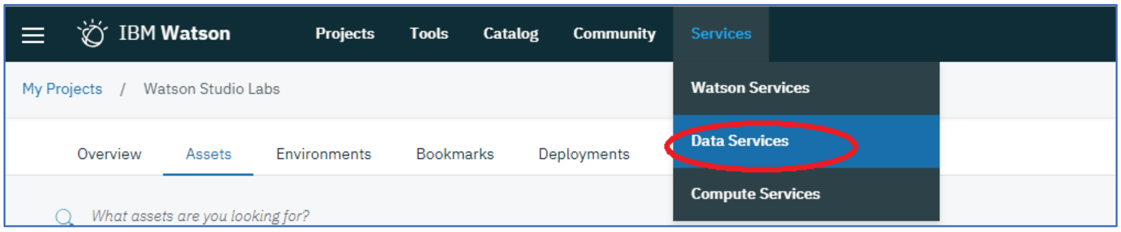
Step 1: Set up the Data Files in IBM Cloud Storage

Training a deep learning model using Watson Machine Learning relies on using Cloud Object Storage for reading input (such as training data) as well as for storing results (such as log files.)

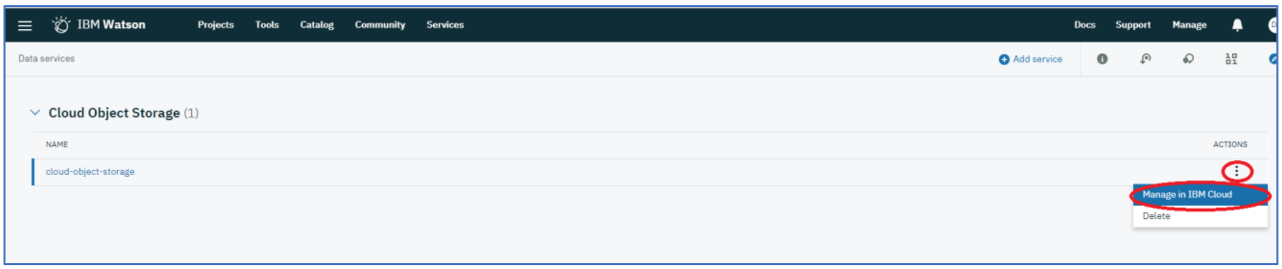
1. Download the mnist.zip file. Extract the 3 files - a training file (mnist-tf-train.pkl), test file(mnist-tf-test.pkl), and a validation file (mnist-tf-valid.pkl) in pickle format.



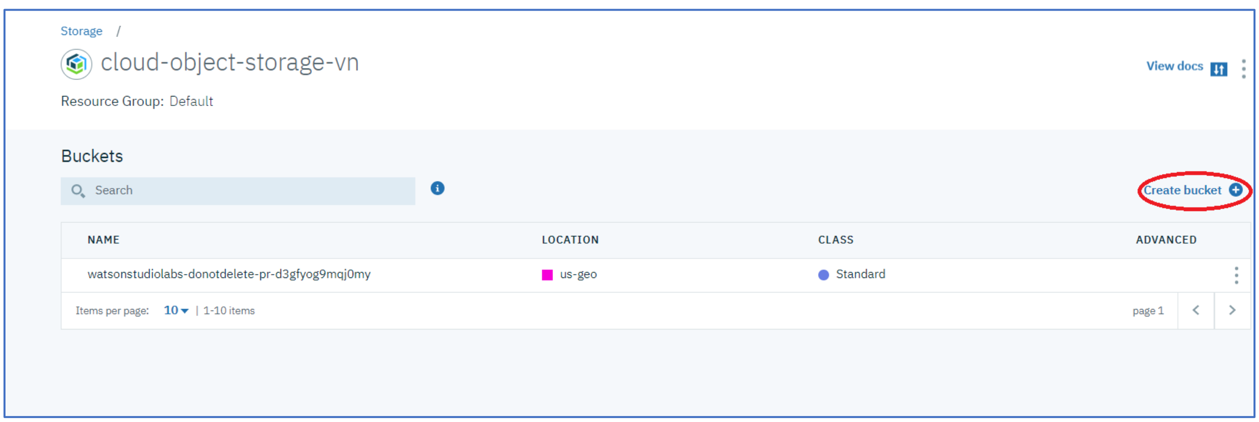
2. Select the **Services** tab and the click on **Data Services**



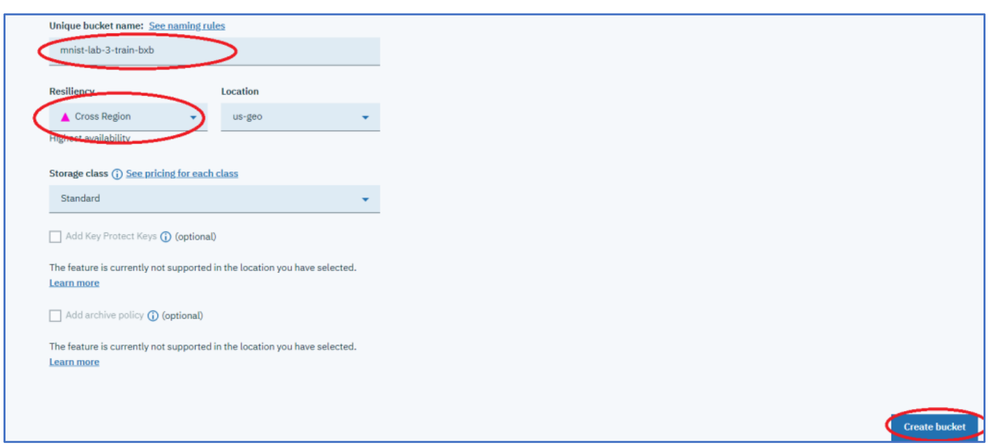
3. Select the vertical **ellipse** on the right hand side of the cloud object storage entry, and then click on **Manage in IBM Cloud**



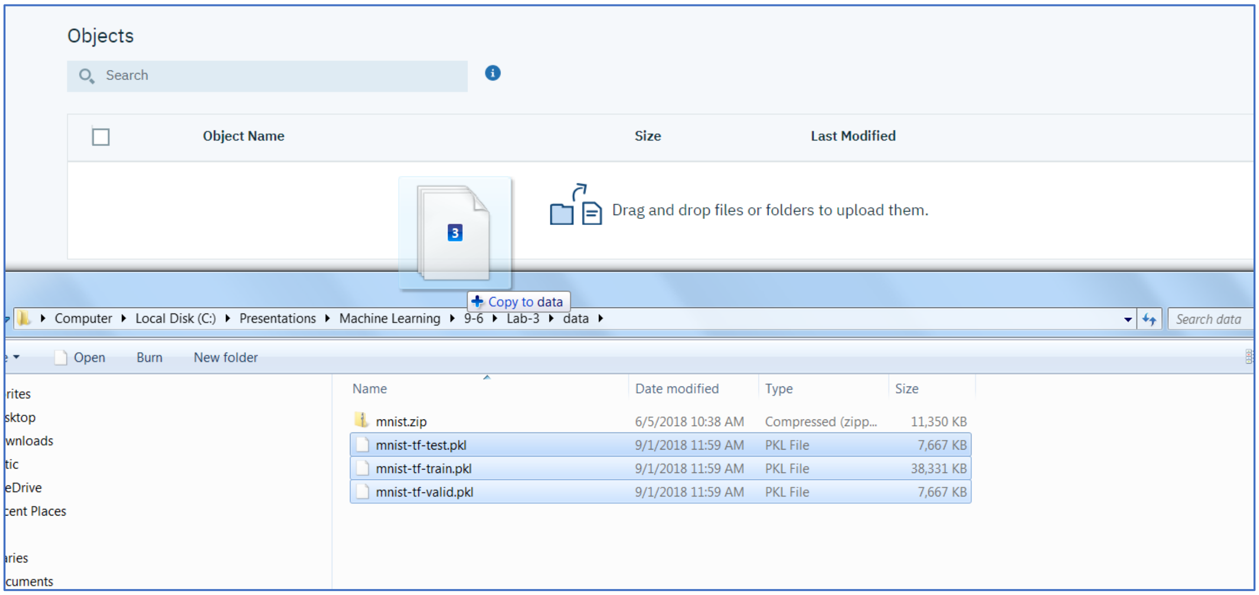
4. Click on **Create Buckets**



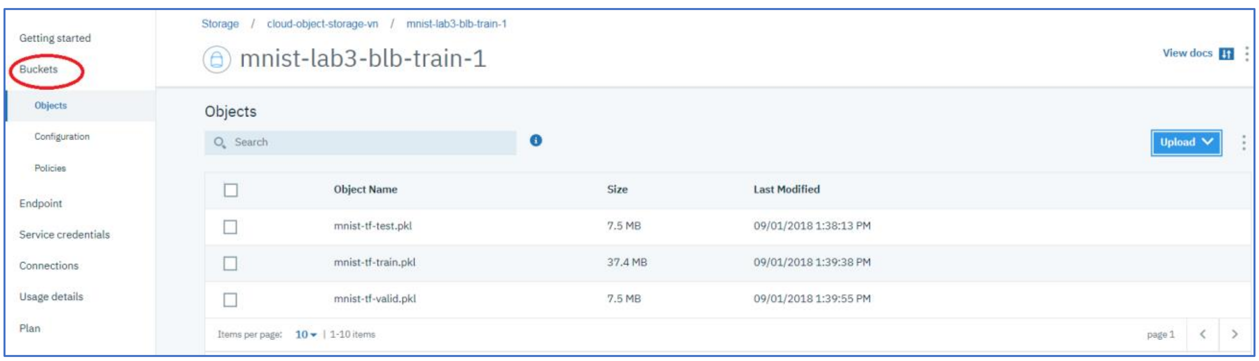
5. Enter a unique name for the bucket - mnist-lab-3-train-xxx (replace xxx with your initials), click on **Cross-Region** for the Resiliency and click **Create**.



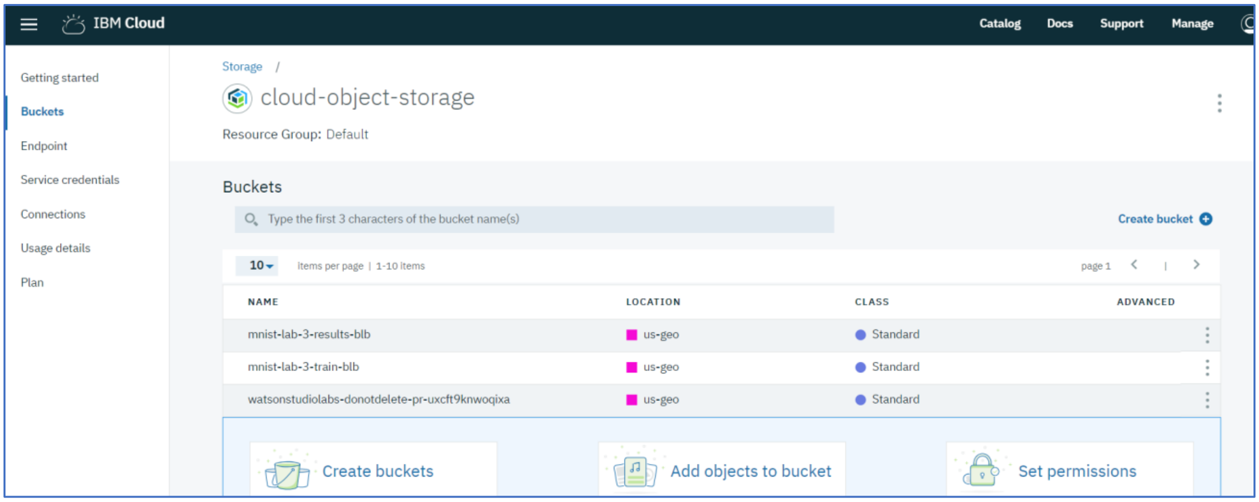
6. Navigate to the directory where the 3 mnist files are stored. Select these 3 files and drag and drop where indicated.



7. Click on Buckets to add a second bucket.

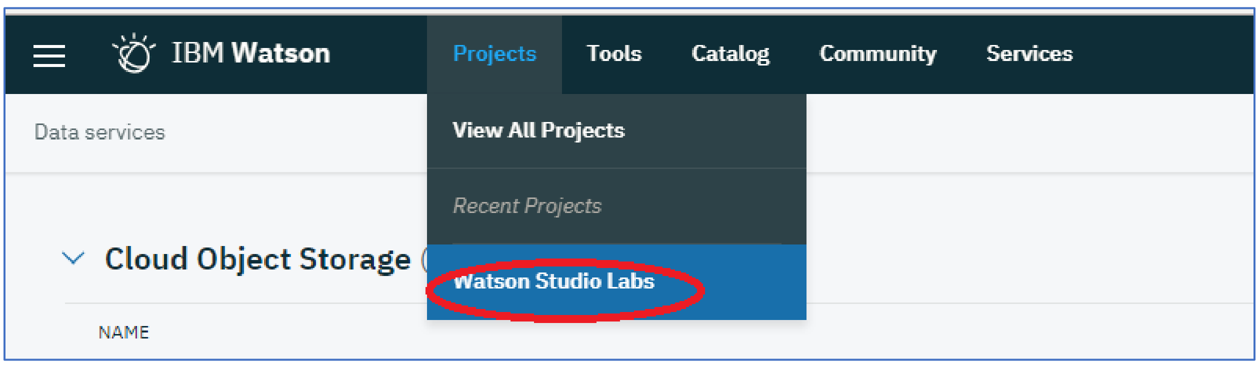


8. Name it mnist-lab-3-results-xxx, where xxx are your initials. Follow the procedure above to create the second bucket. No files need to be added.
9. The Cloud Object Storage panel should appear as below.

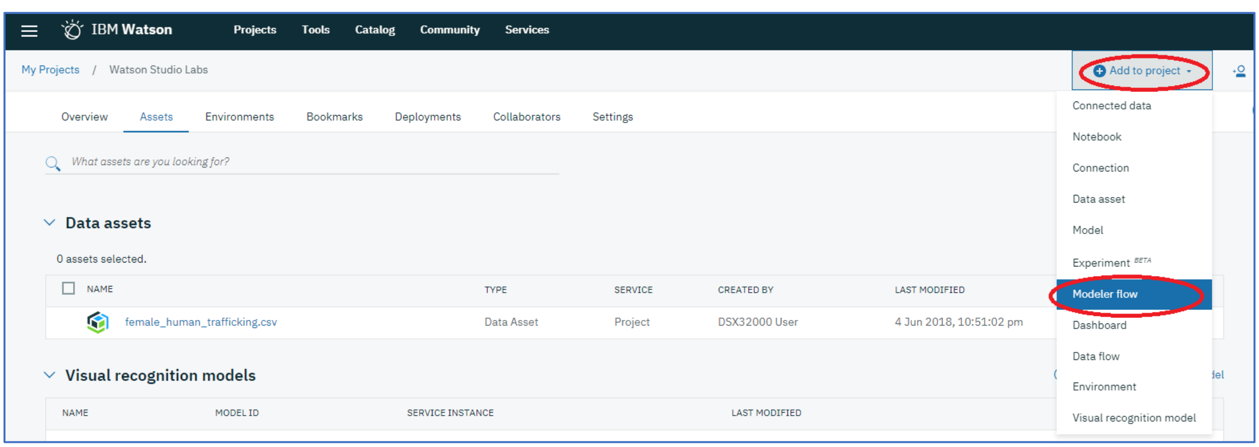


Step 2: Design the Neural Network and Publish Training Definition

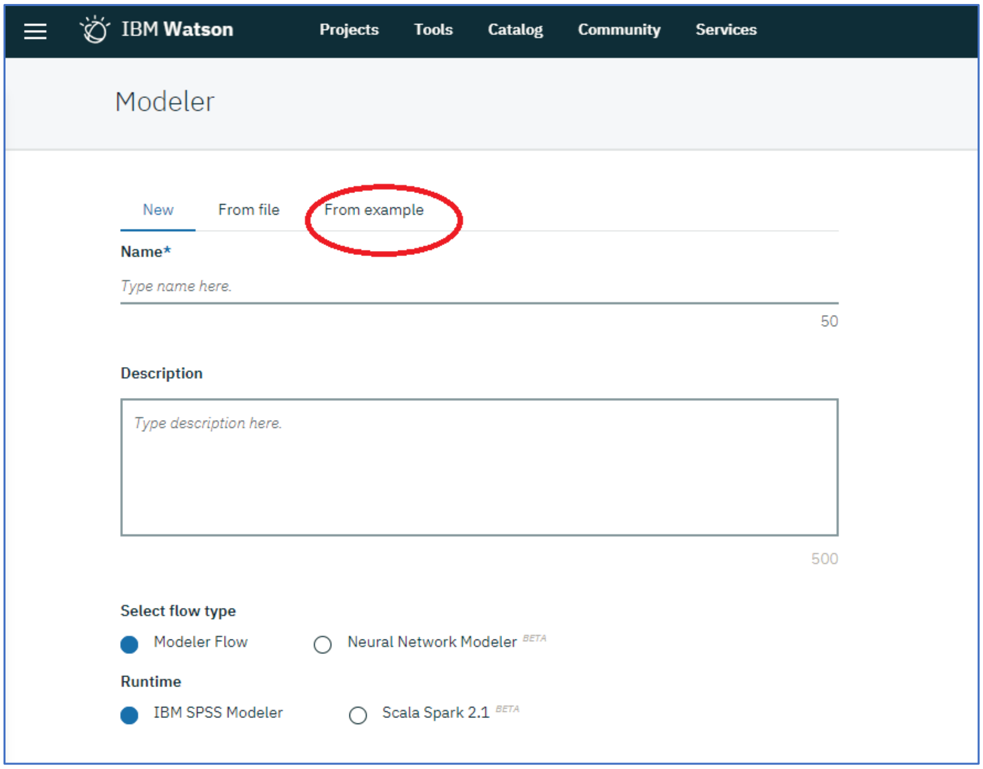
1. Return to Watson Studio, and click on the **Projects** tab, and **Watson Studio Labs**.



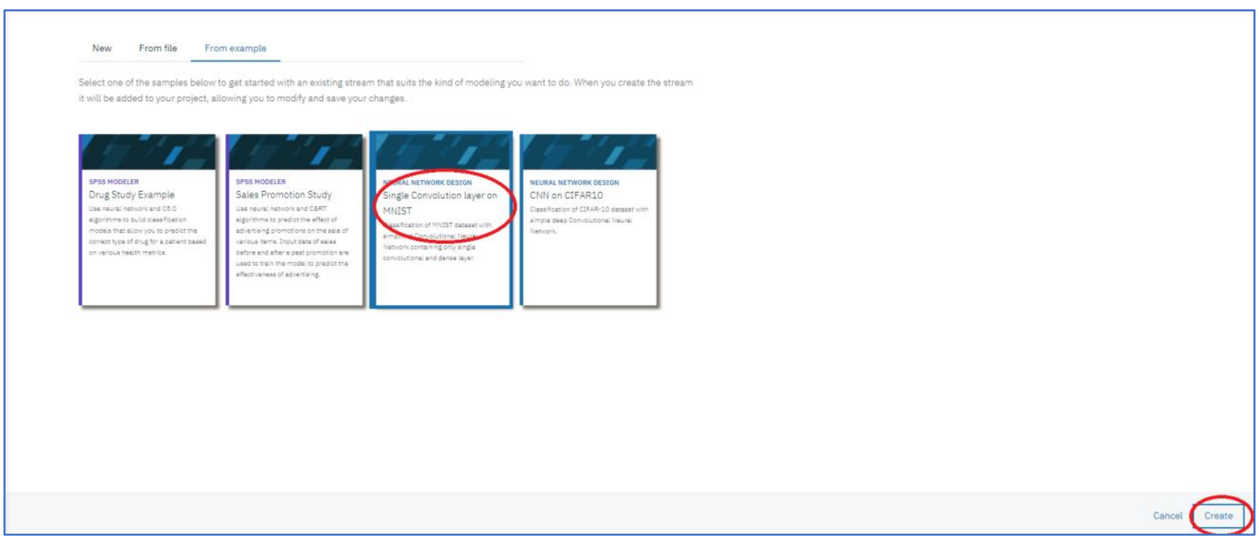
2. Click on the **Add to project** and then click on **Modeler Flow**.



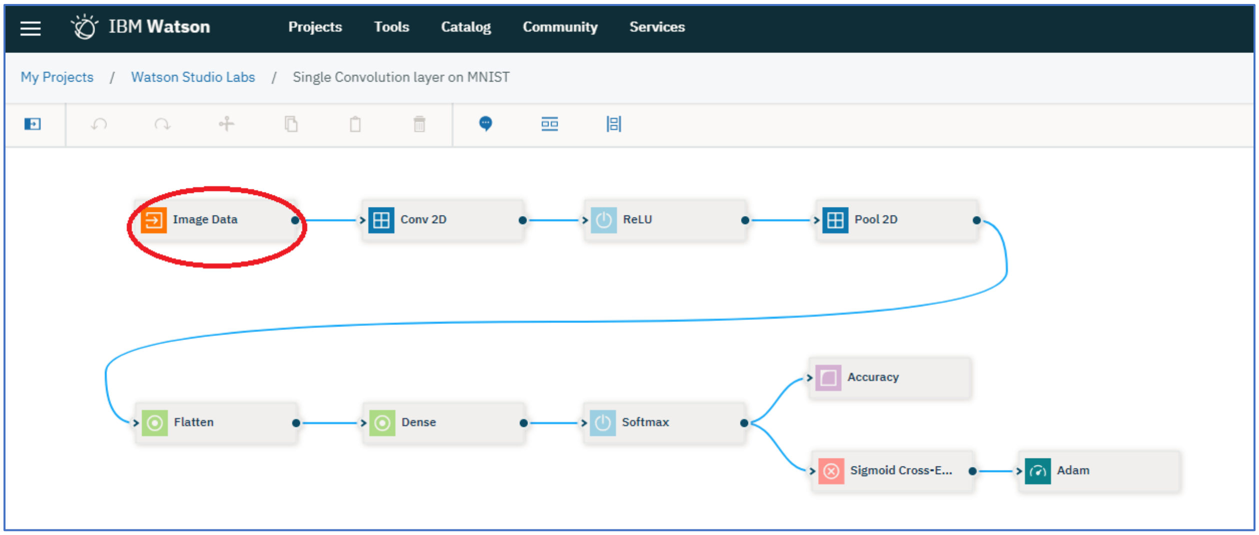
3. Click on **From example**.



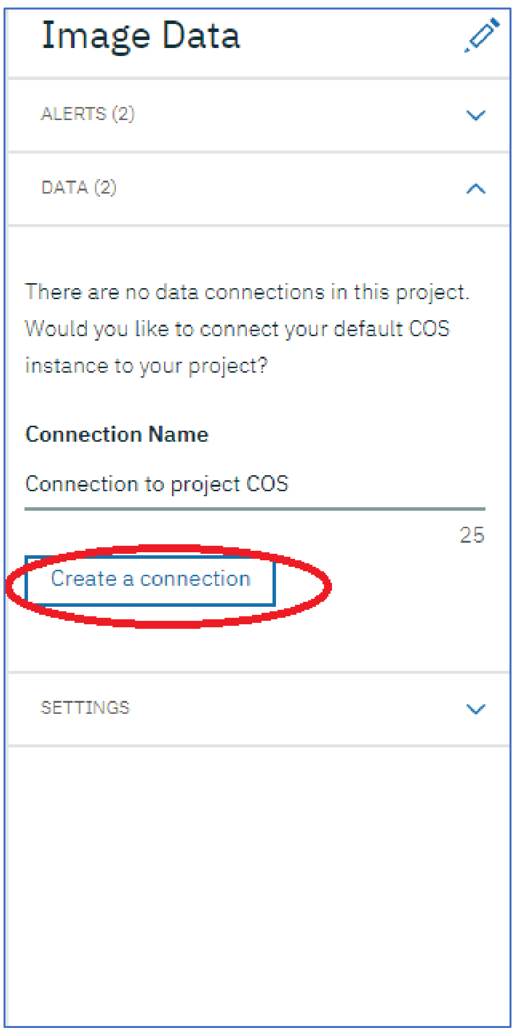
4. Click on the **Single Convolution Layer on MNIST** and then click on **Create**



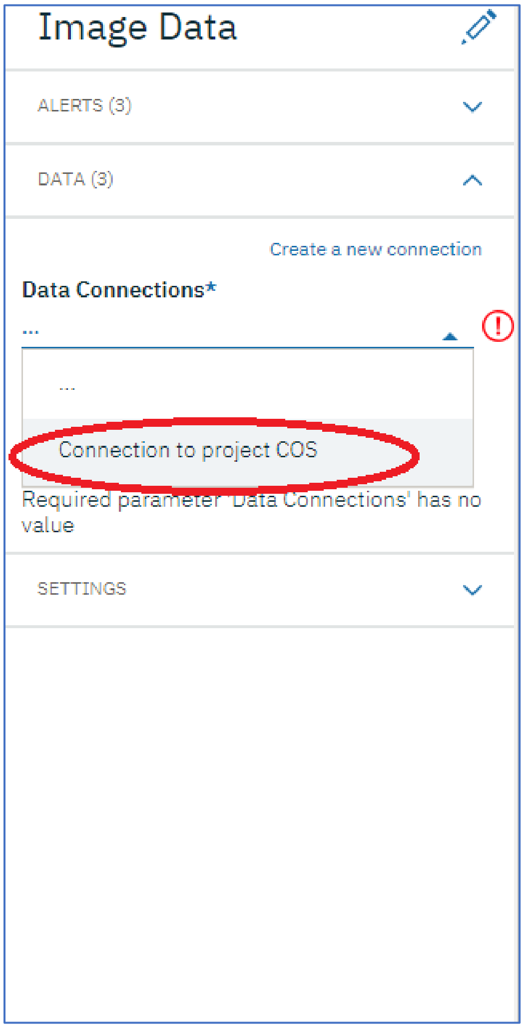
5. A standard convolution neural network (CNN) architecture is displayed. We need to configure the Image Data node. Double-click on Image Data.



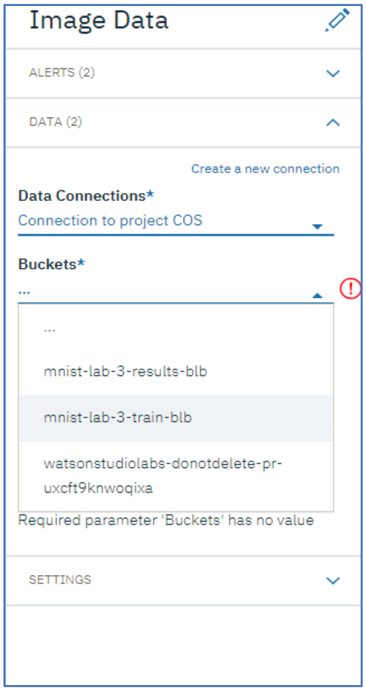
6. Optionally change the default **Connection Name**, and then click on **Create a connection**.



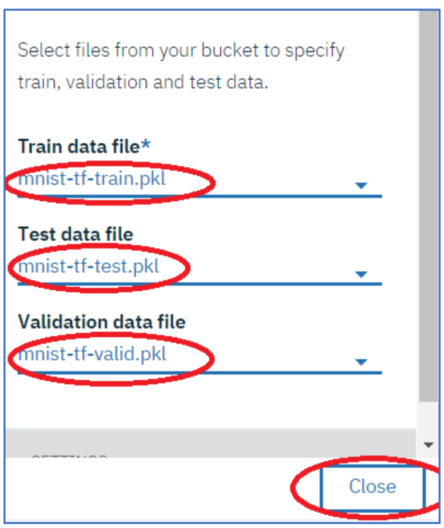
7. Click on the downward triangle icon ▾ underneath **Data Connections*.** Click on the connection that was just created.



8. Click on the downward triangle icon ▾ underneath **Buckets***, and then click on the **mnist-lab-3-train-xxx** where "xxx" are your initials.

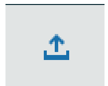


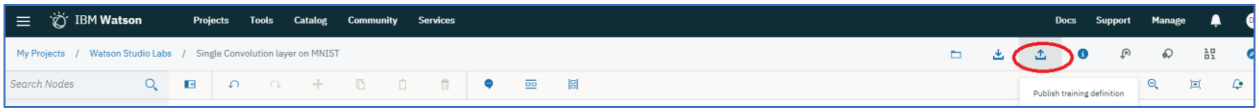
9. Click on the icon underneath **Train data file*** and select the **mnist-tf-train-pkl.** Assign the Test data file(mnist-tf-test-pkl), and Validation data files(mnist-tf-valid-pkl) in the same way and then click on **Close**.



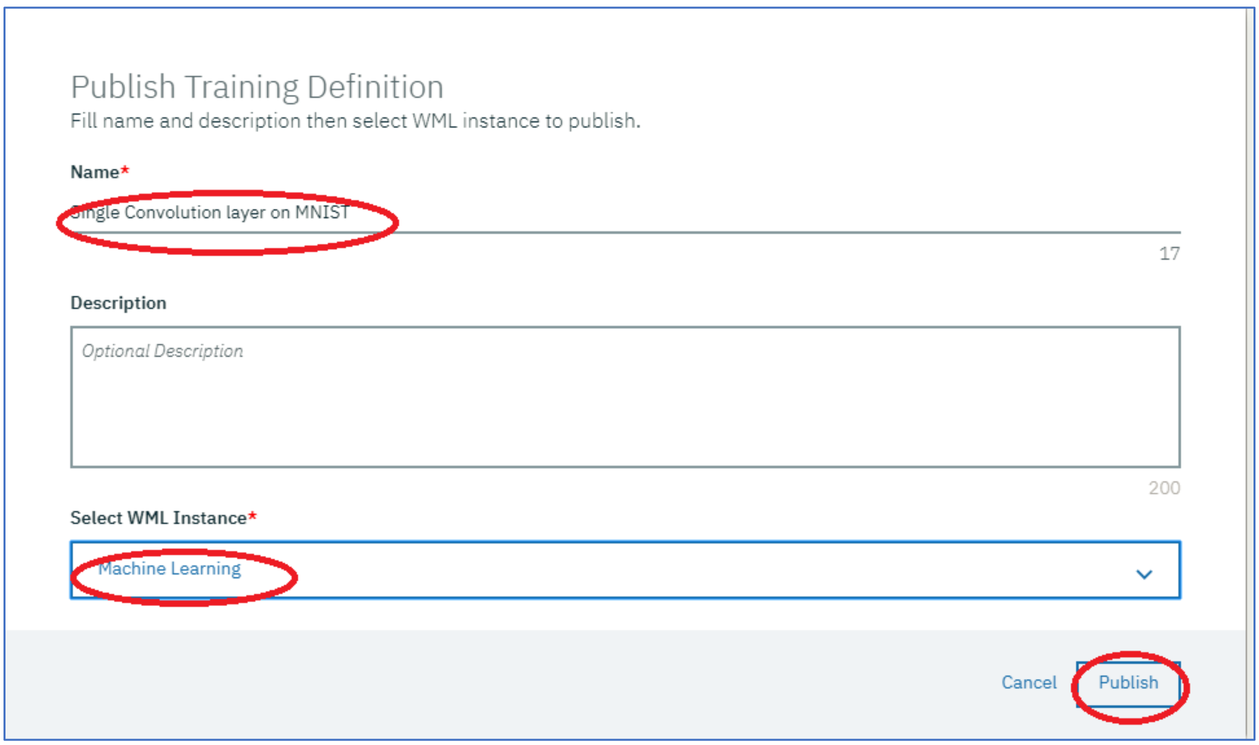
10. Explore the neural network flow modeler options
    1. Click on the icon to see the list of neural network component categories that are available
    2. Explore the contents in each category. Hover over the components to get a pop-up description.
    3. Drag some nodes on the canvas and double-click to see the parameters. **Note remove these nodes before doing step 11.**

4. Click on the download icon  to see the multiple options for code generation.

11. Click on the **Publish** icon  to create a training definition.



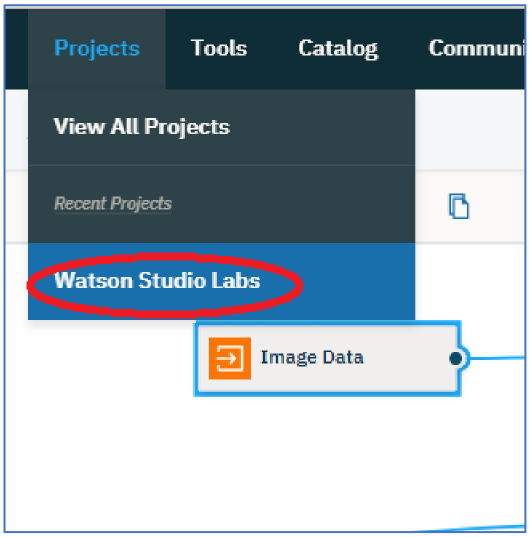
12. Enter a name for the training definition (or leave the default), and select the Machine Learning service that you created. Note, it will not be named Machine Learning unless that is the name that you used. Click on Publish.



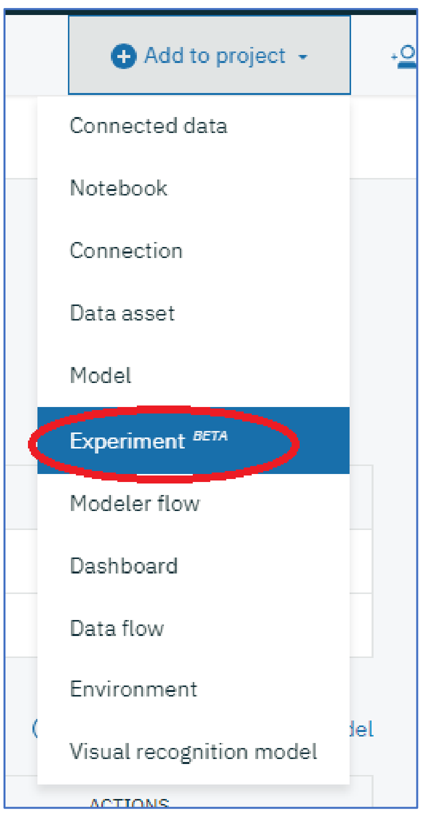
Step 3: Train the Model using Experiment Builder

As part of the model building process, we want to be able to compare different algorithms, and/or different algorithmic parameters to determine the best model. Experiment Builder is a facility in Watson Studio that supports this effort. Different training runs can be defined and run in parallel and their results can then be compared. In this lab, we will define only one training run to minimize the training time.

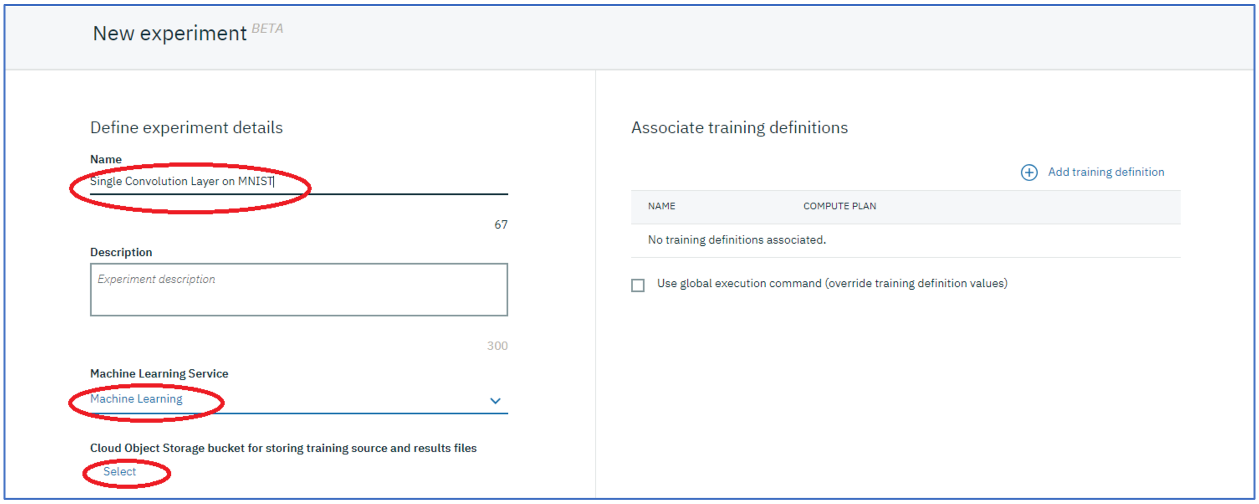
1. Return to the Watson Studio Labs Assets panel by clicking on the **Projects** tab and then **Watson Studio Labs**. Click on the **Assets** tab if the Assets panel is not displayed.



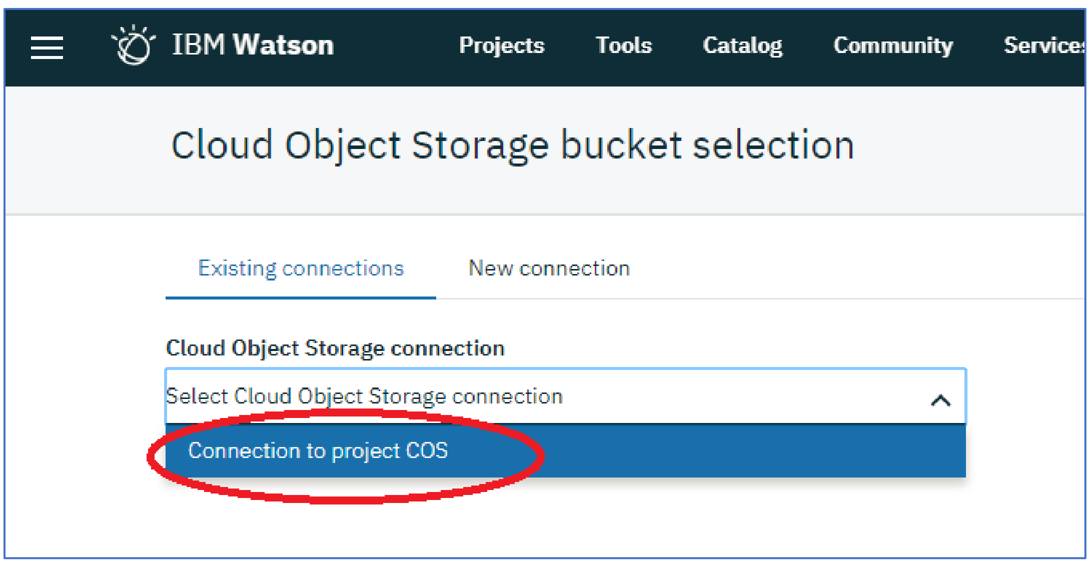
2. Click on **Add to project**, and then click **Experiment** to create a new Experiment.



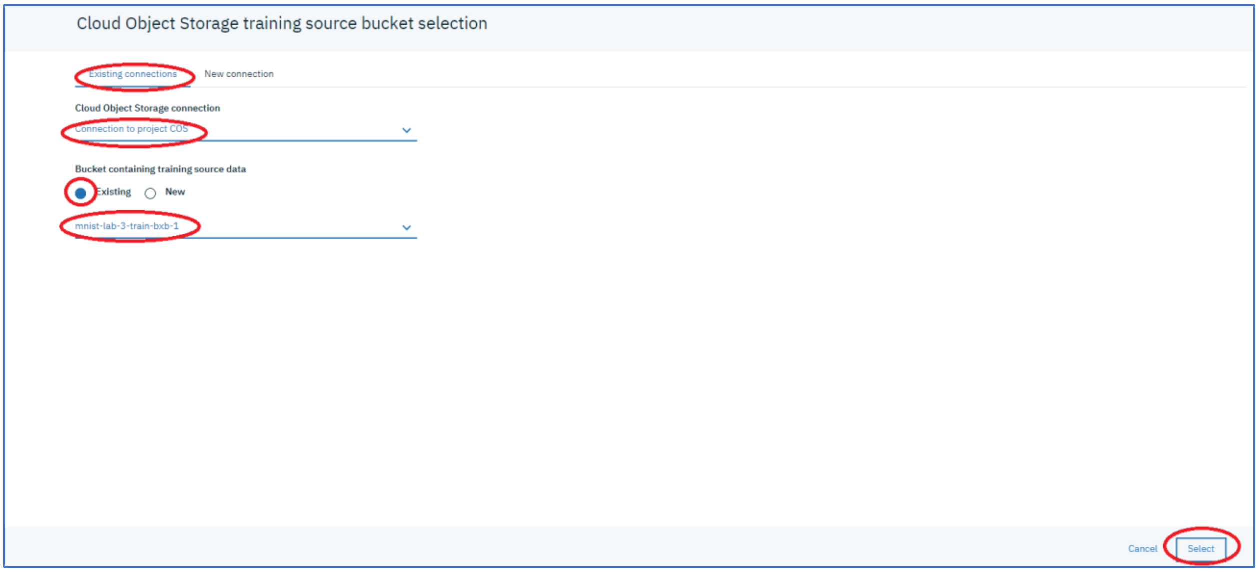
3. Enter an Experiment **Name**, select the **Machine Learning** service, and then click on **Select** to assign a Cloud Storage bucket.



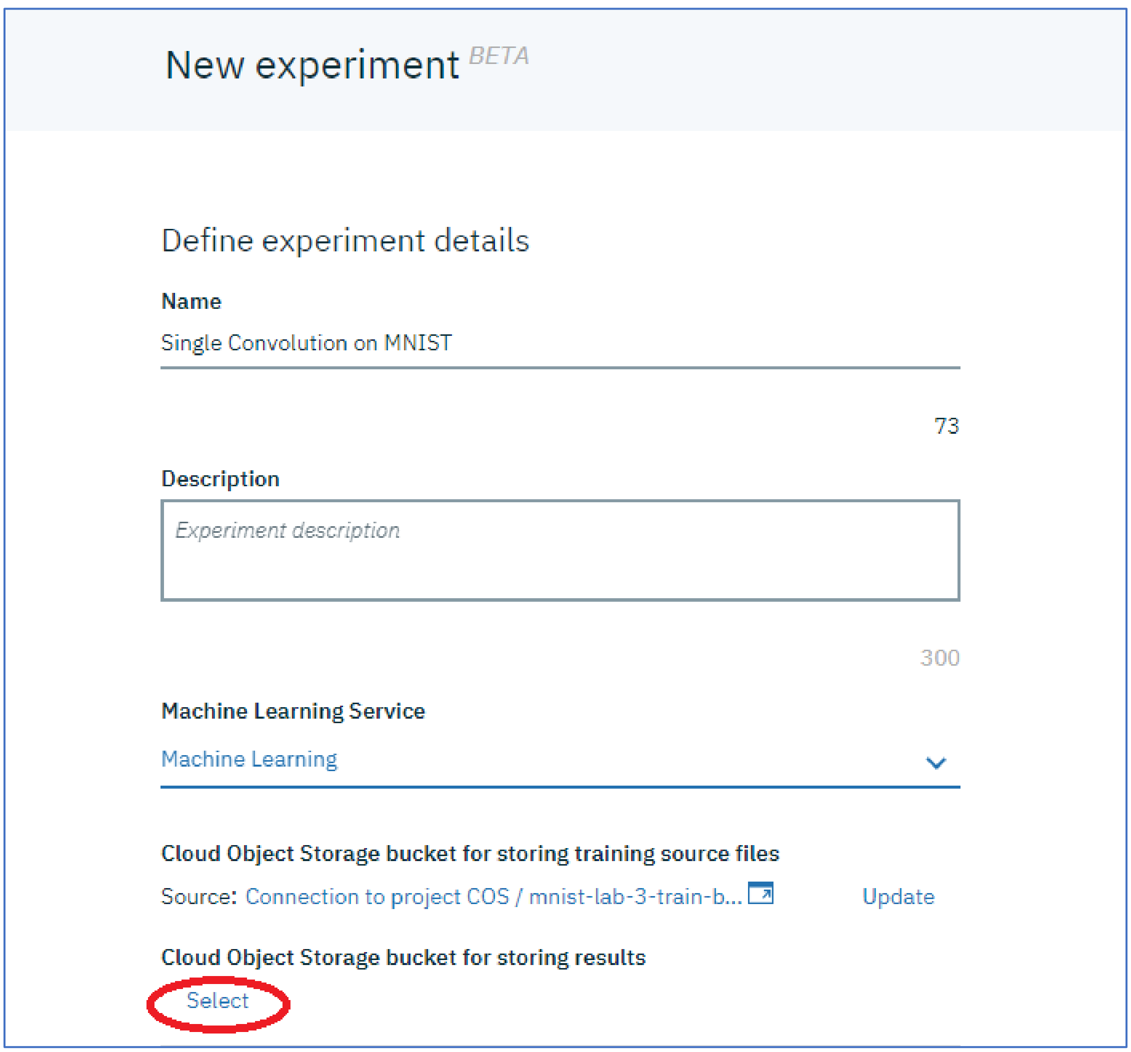
4. Select **Existing connections**, and then select the **Connection to project COS** connection.



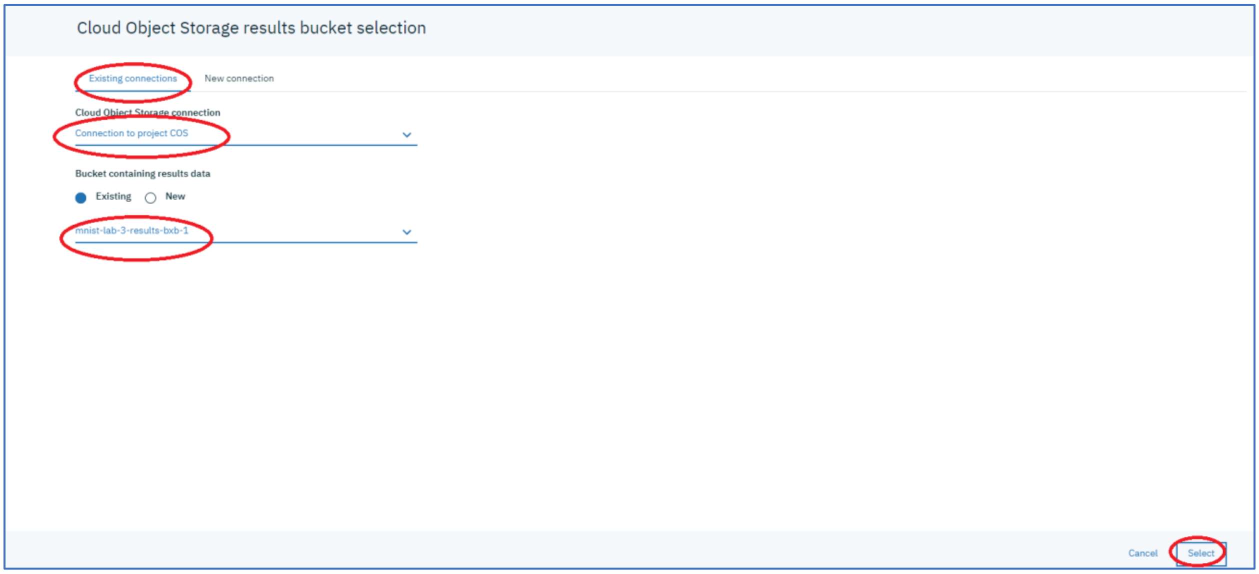
5. We now need to assign the Training and Results buckets. Select **Existing** underneath **Bucket containing training data**, and click on mnist-lab-3-train-xxx, where “xxx” are your initials. Click on **Select**.



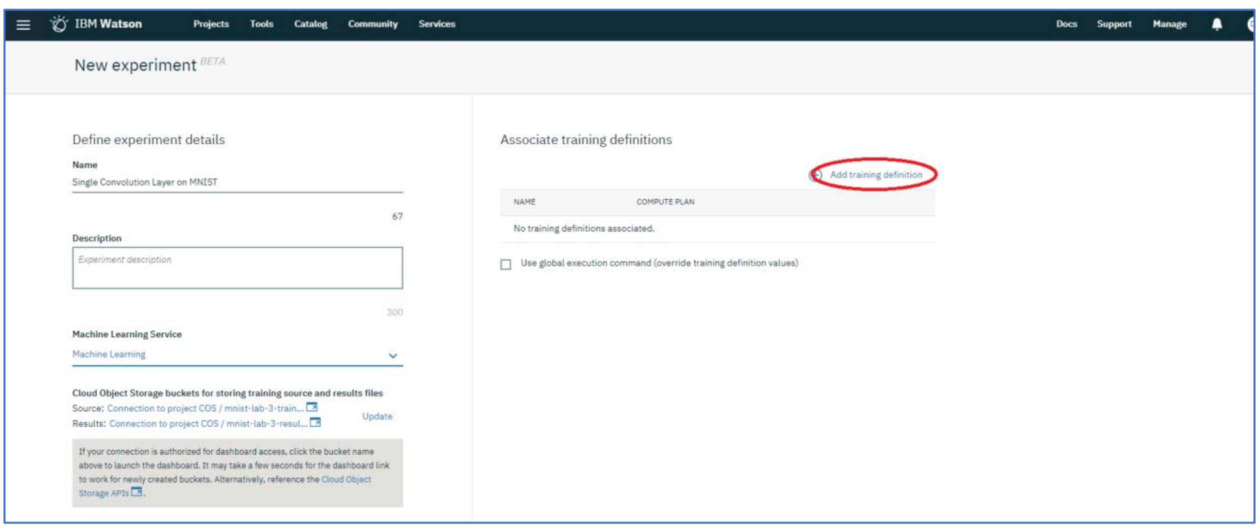
6. Click on **Select** underneath **Cloud Object Storage for storing results.**



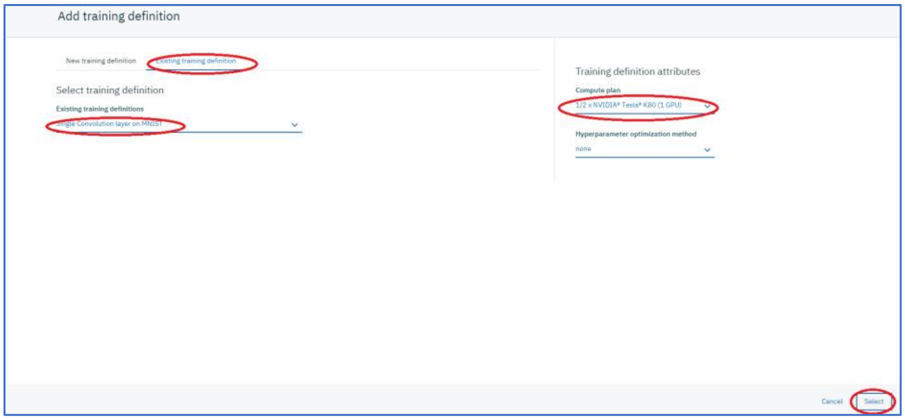
7. Follow the same procedure used to assign the training bucket to assign the results bucket. Assign bucket mnist-lab-3-results-xxx, where “xxx” are your initials, and then click on **Select.**



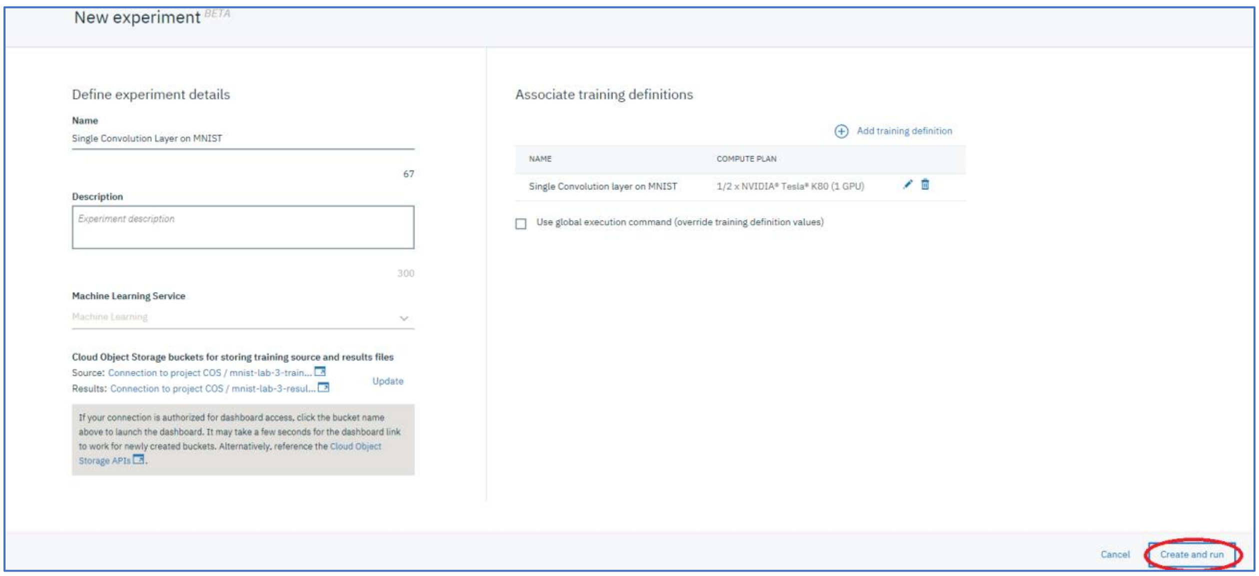
8. We now need to associate a Training Definition. Click on **Add Training Definition**.



9. Click on **Existing training definition,** and select **Single Convolution Layer on MNIST**, select 1/**2 x NVIDIA Tesla K80 (1 GPU)** for the compute plan, and then click **Select**.



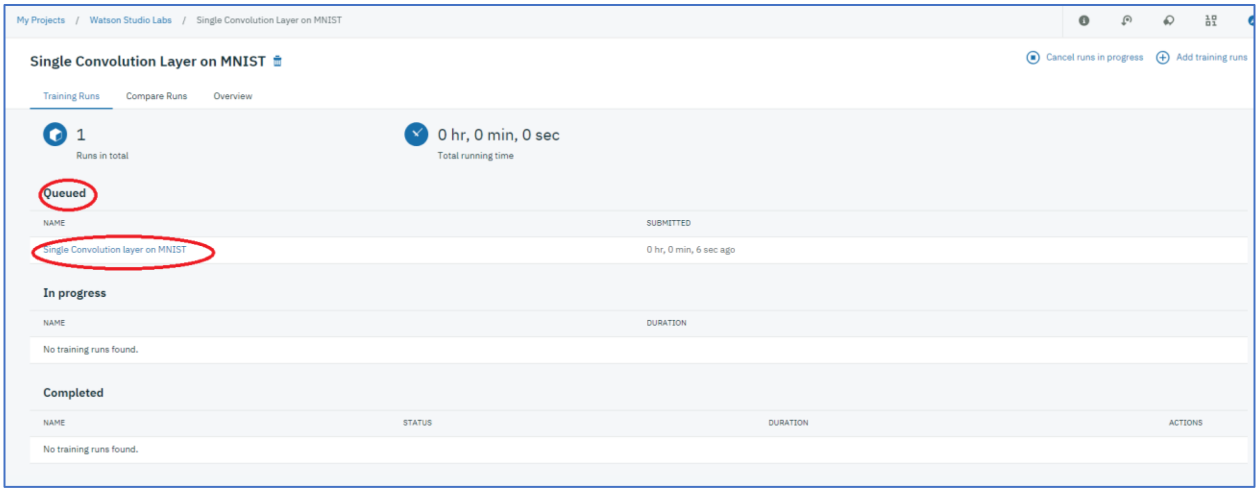
10. Click **Create and run**.



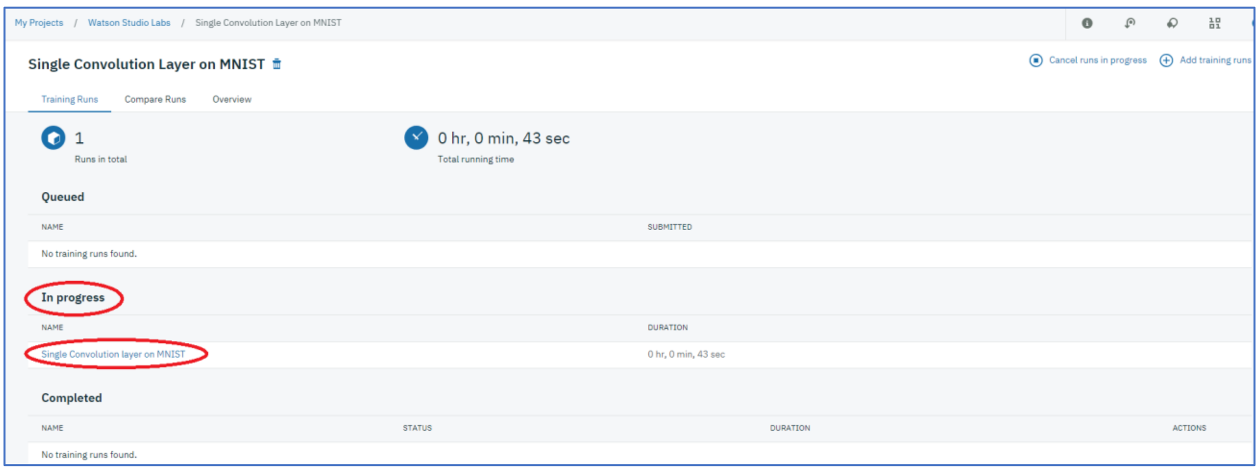
Step 4: Monitor the Training Progress and Results

Training runs will be first queued, then in-process, and then completed. Use the **Training Runs** tab to keep track of progress.

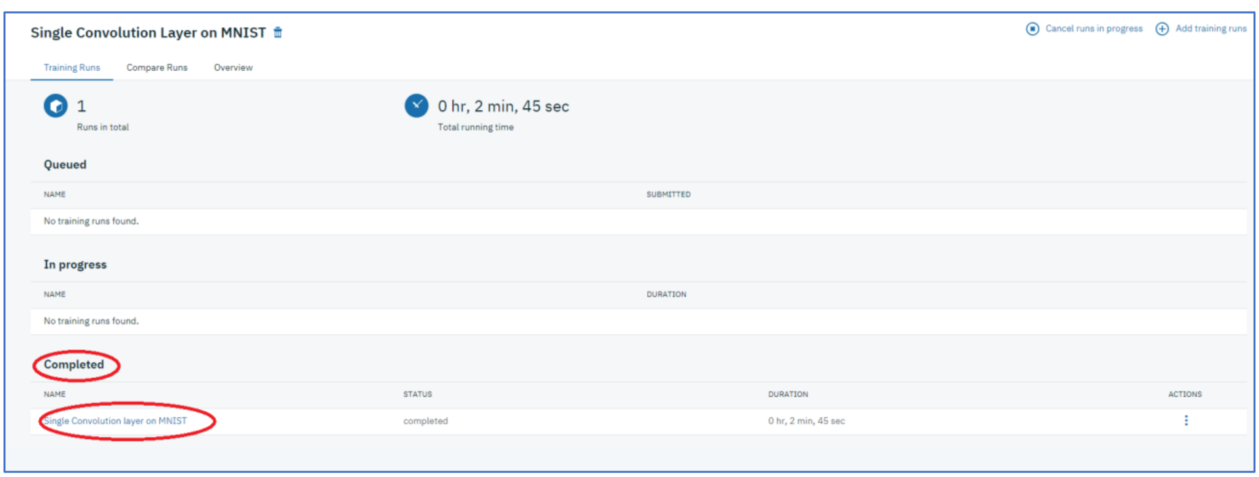
Queued Status



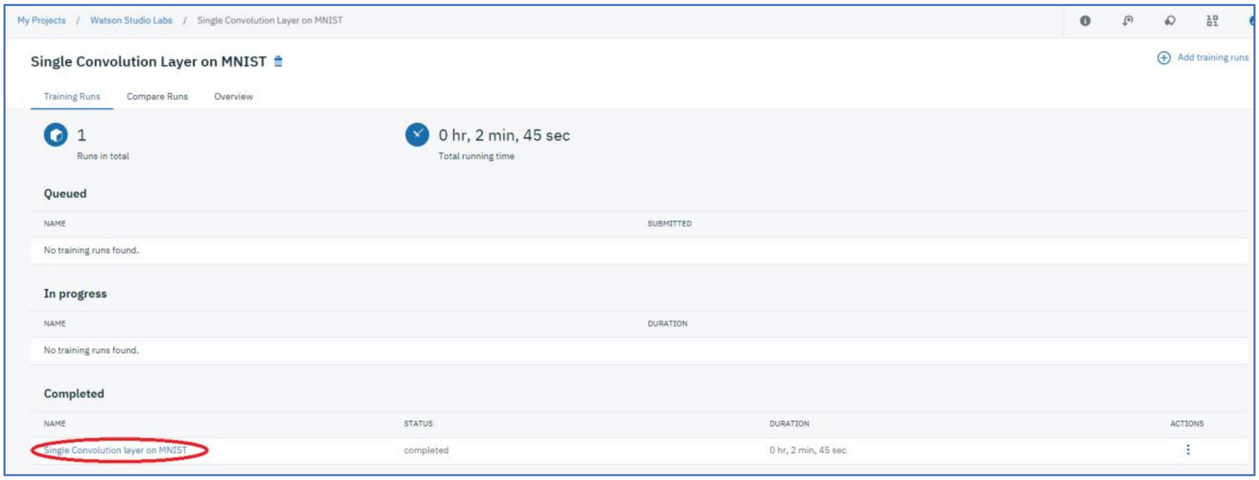
In-Process Status



Completed Status

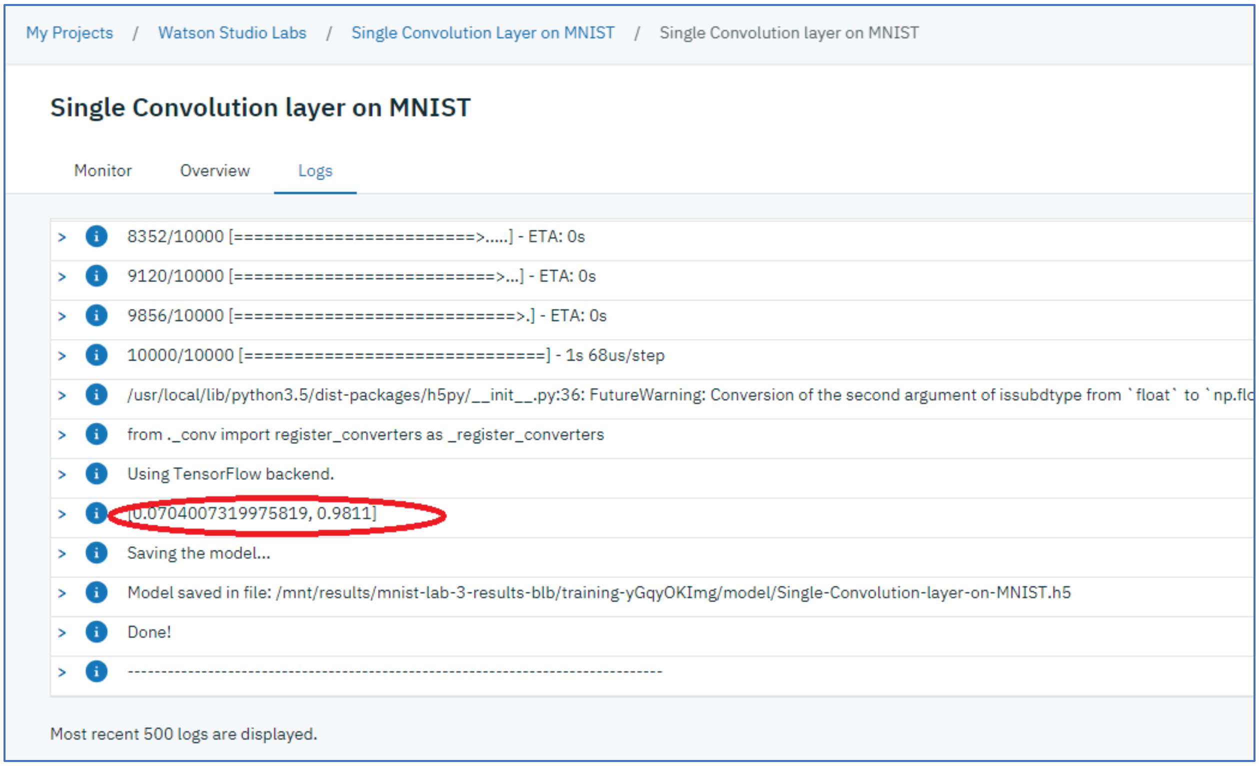


1. Click on the Single Convolution layer on MNIST link to check the results.



2. Click on **Logs**

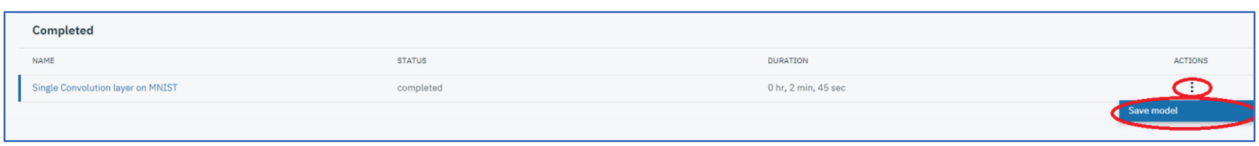
3. Scoll down to the bottom to check accuracy measure.



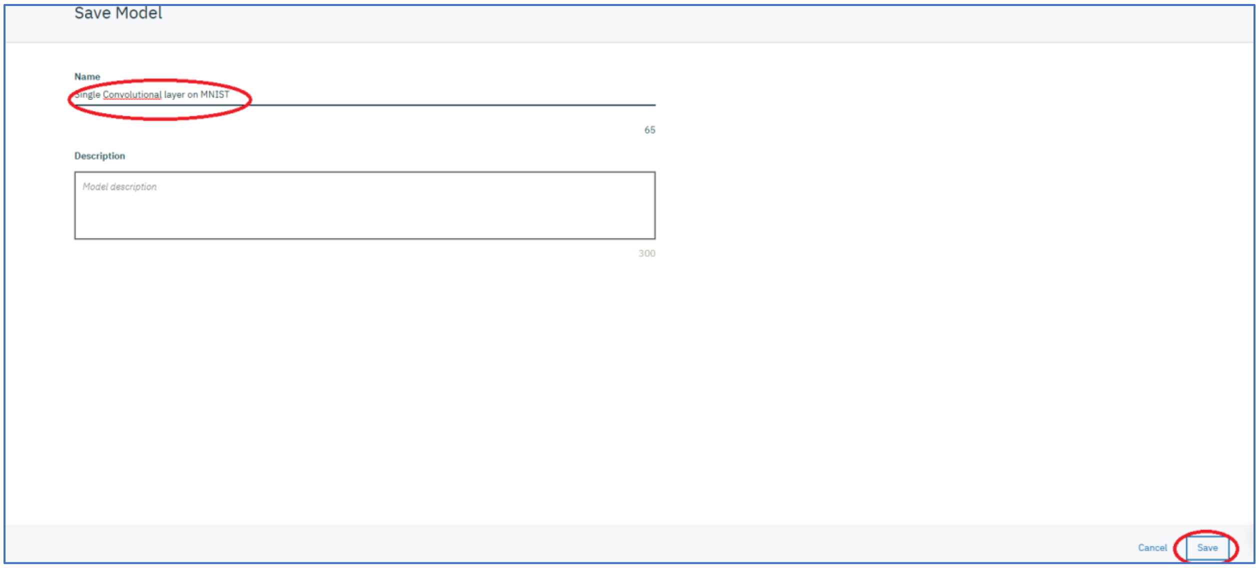
Step 5: Save and Deploy the Trained Model

We will now save the trained model to the Watson Machine Learning repository.

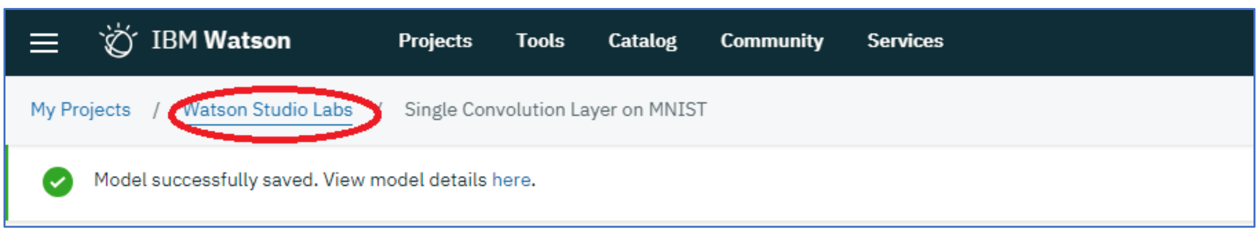
1. Click on the vertical ellipse under ACTIONS, and click **Save model**.



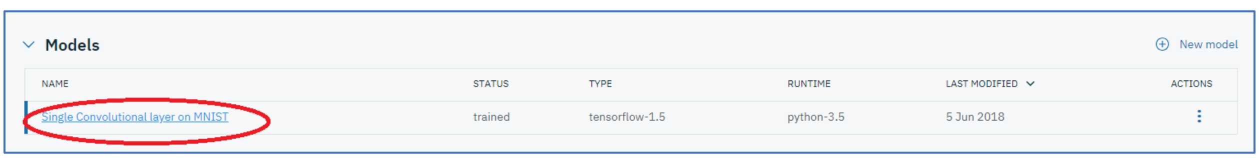
2. Enter a **Name** for the model (Single Convolution layer on MNIST) and click **Save.**



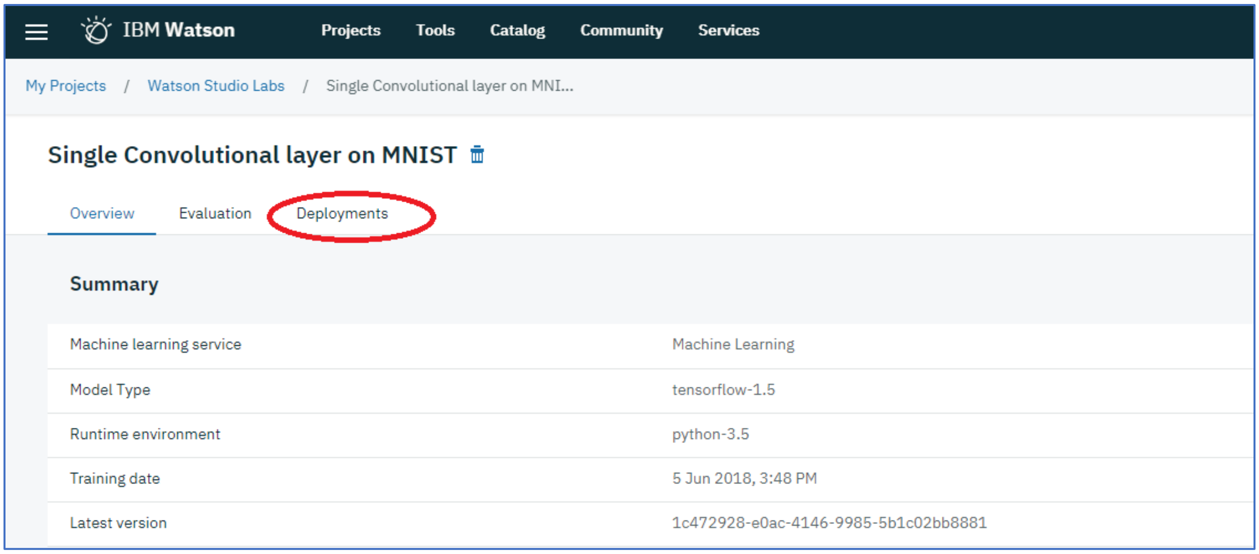
3. Return to the Watson Studio **Assets** panel, by clicking on **Watson Studio Labs** in the breadcrumb path. Click on the **Assets** tab if the Assets panel is not showing.



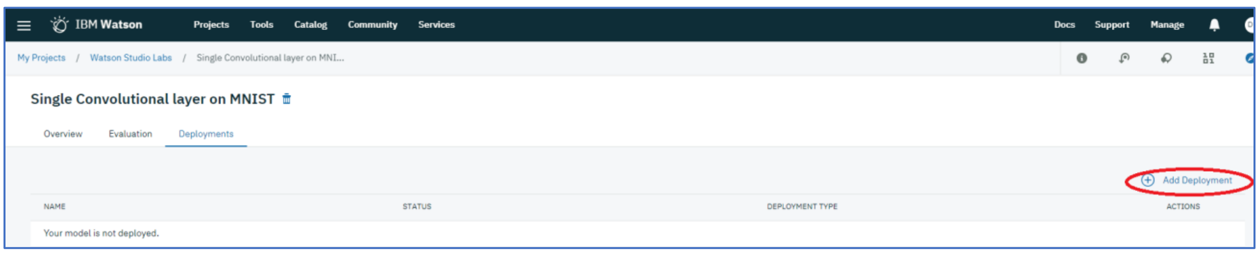
4. Click on the newly saved model



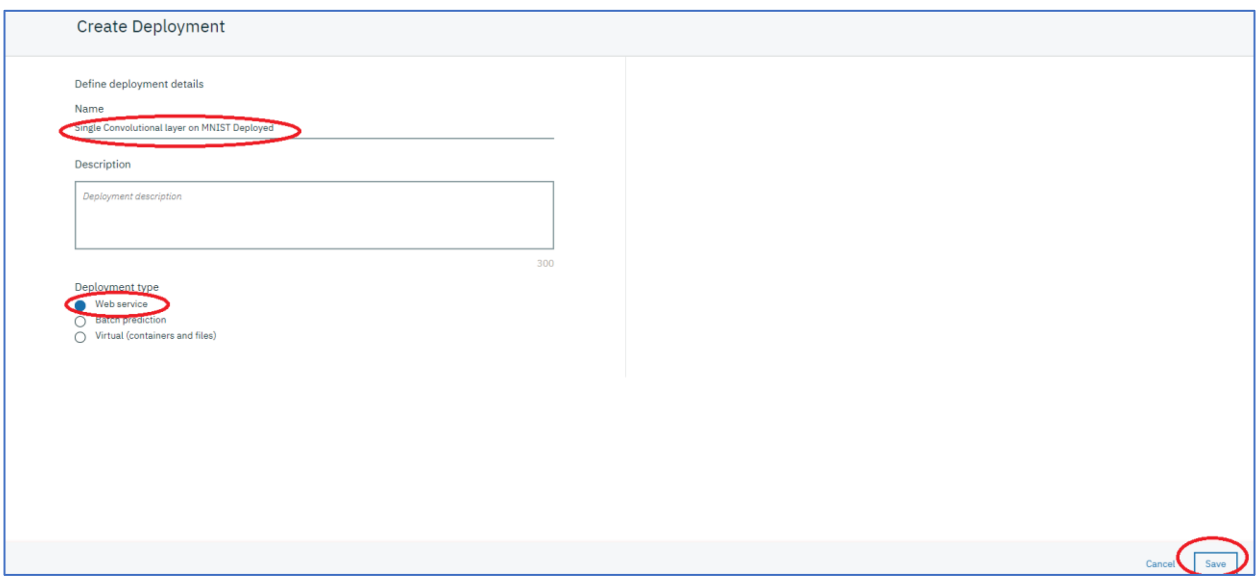
5. Click on **Deployments.**



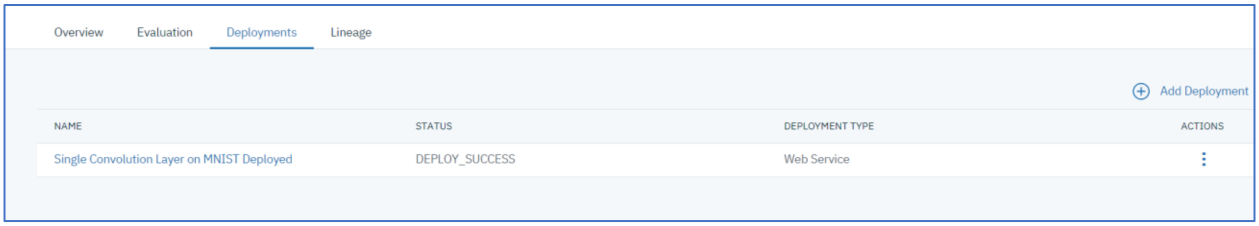
6. Click on **Add Deployment**.



7. Enter a **Name** (e.g. Single Convolution layer on MNIST Deployed), select **Web Service** (should be the default), and click on **Save**.



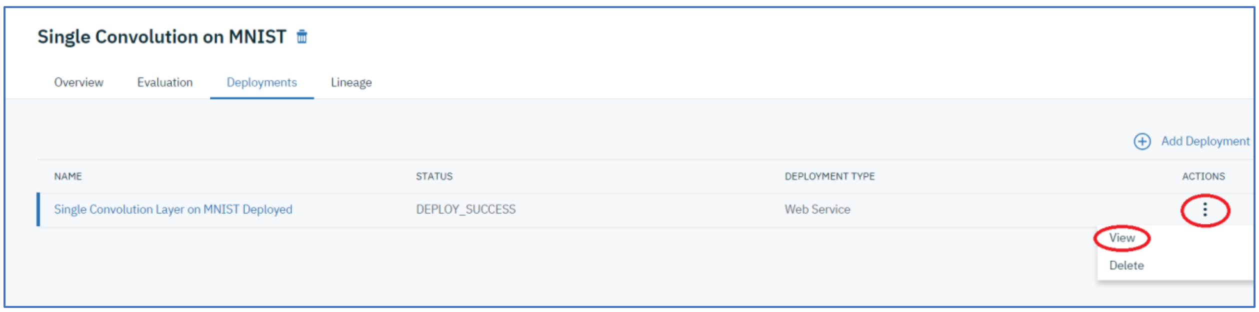
8. The model is successfully deployed.



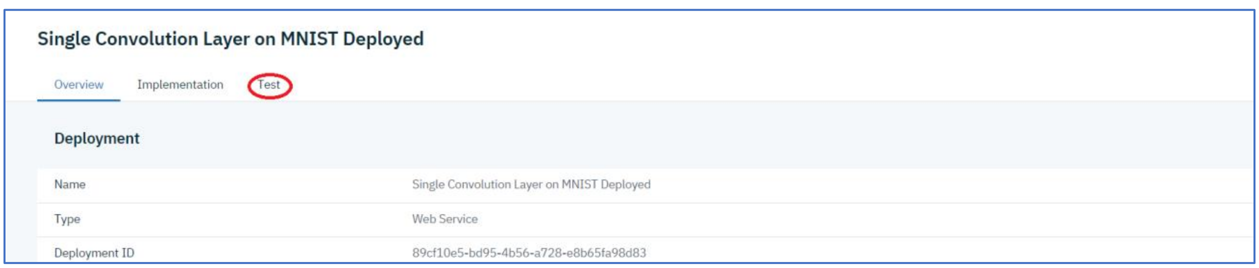
Step 6: Test the Deployed Model

We will now test the deployed model using the sample image data contained in the file test.json that was extracted from the mnist.zip file previously.

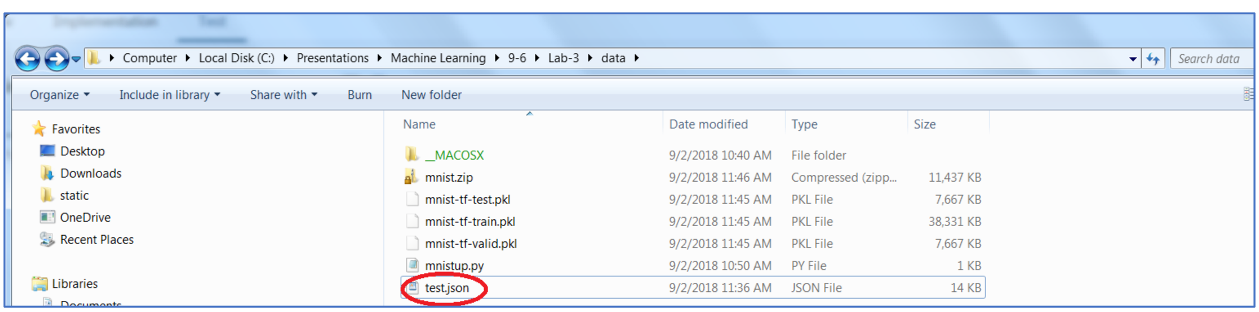
1. Click on the vertical ellipse, and then click on **View**.



2. Click on **Test**.



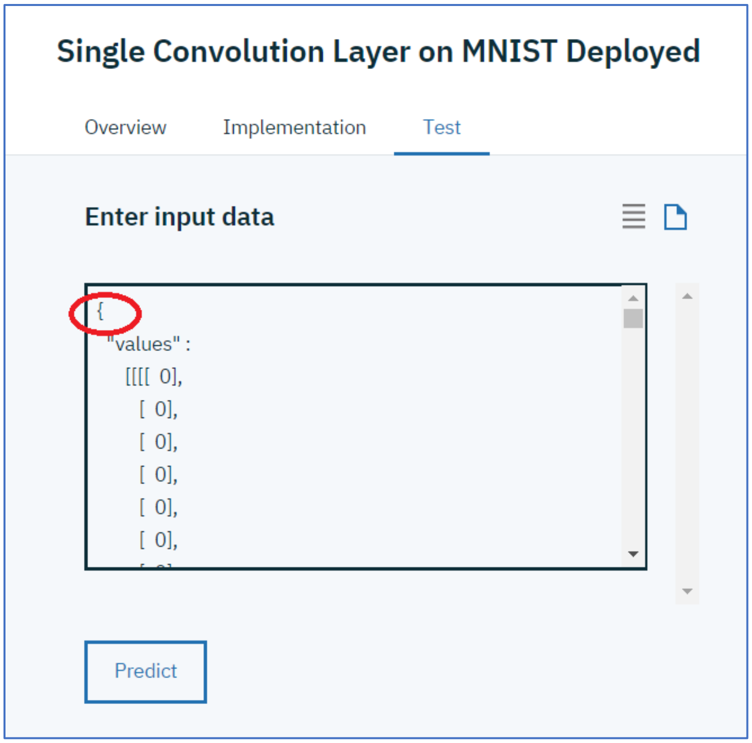
3. Go to the file directory where you have the “test.json” file stored, and double-click on the file.

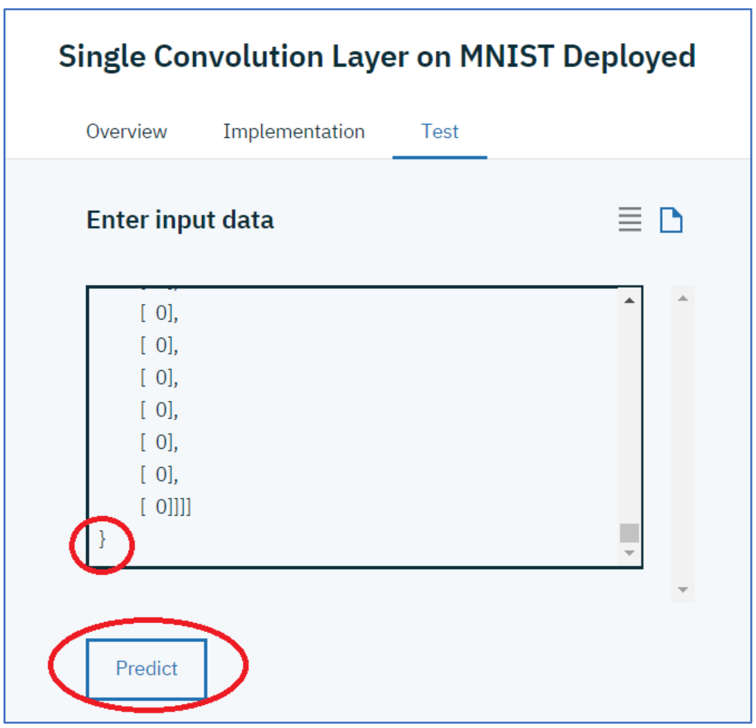


4. Select the contents of the file by placing the cursor to the left of the { and pressing and holding the <Shift><Ctrl><End> keys.

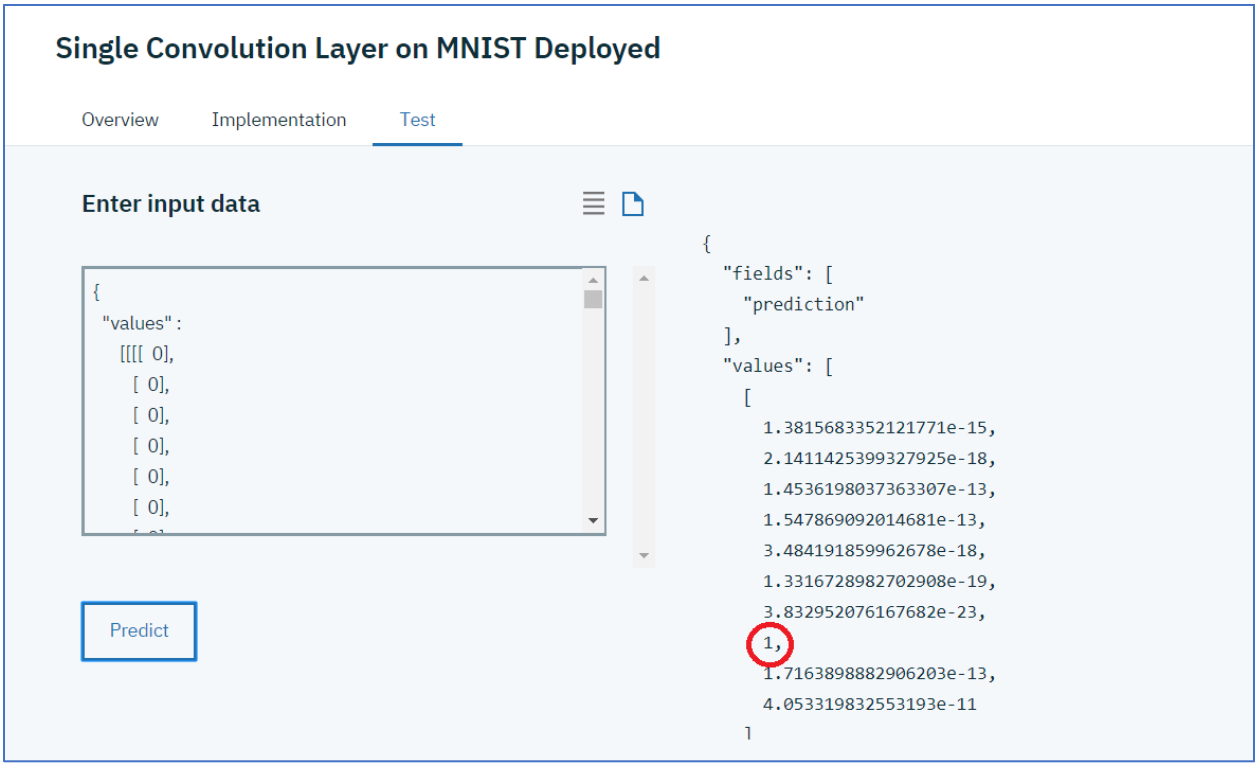
```
{
  "values" :
       [[[[   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
          [   0],
```


5. Copy and paste the content into the **Paste the request payload here** input data section. Make sure you have both the top bracket { at the beginning of the input data section and the bottom bracket } at the end of the input section data section, and then click on **Predict**.





6. Based on the confidence levels returned, we can see that the number 8 would be selected as the best fit for this sample image.


Single Convolution Layer on MNIST Deployed
Overview
Implementation
Test
Enter input data
{
"values" :
[[[[0],
[0],
[0],
[0],
[0],
[0],
Predict
{
"fields": [
"prediction"
],
"values": [
[
1.3815683352121771e-15,
2.1411425399327925e-18,
1.4536198037363307e-13,
1.547869092014681e-13,
3.484191859962678e-18,
1.3316728982702908e-19,
3.832952076167682e-23,
1,
1.7163898882906203e-13,
4.053319832553193e-11
]