# Build, Train, and Save Convolutional Neural Network Model using MNIST

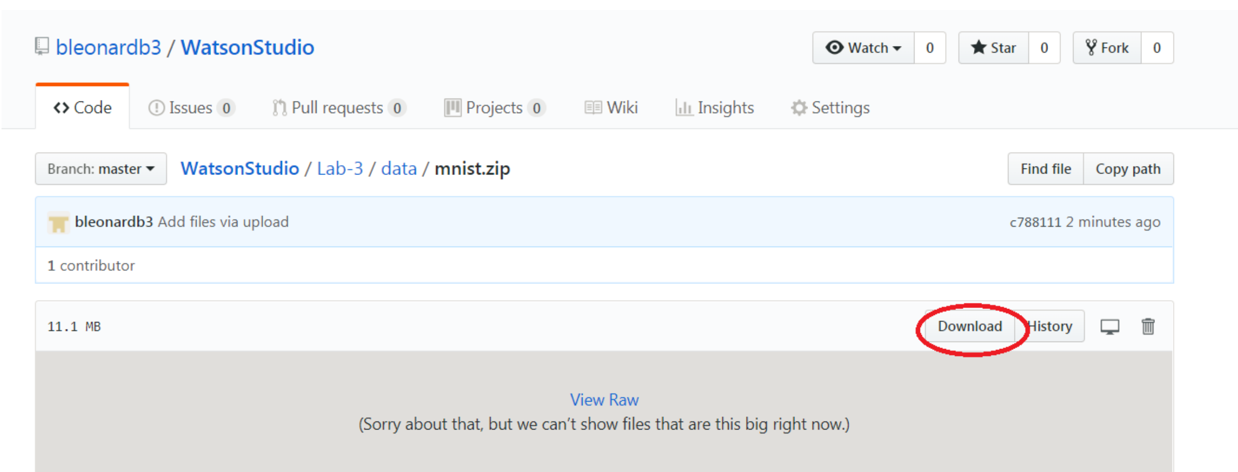
This lab will use the MNIST computer vision data set to train a deep learning model to recognize handwritten digits. A single layer convolutional neural network will be built in the Watson Studio neural network designer, and then trained using the Watson Studio Experiment Builder. The trained model will be saved in the model repository. 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 the Trained Model

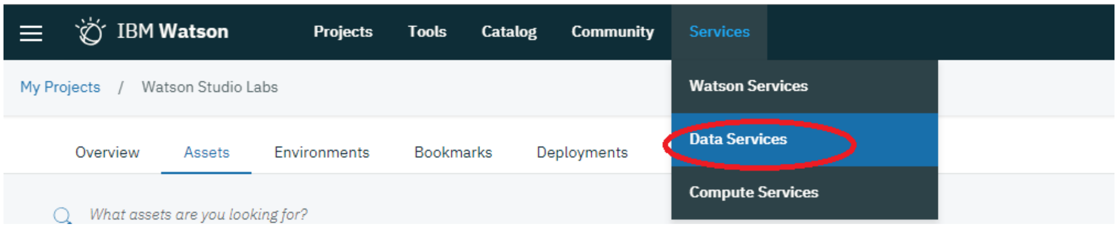
## Step 1: Set up the Data Files in IBM Cloud Storage

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.)

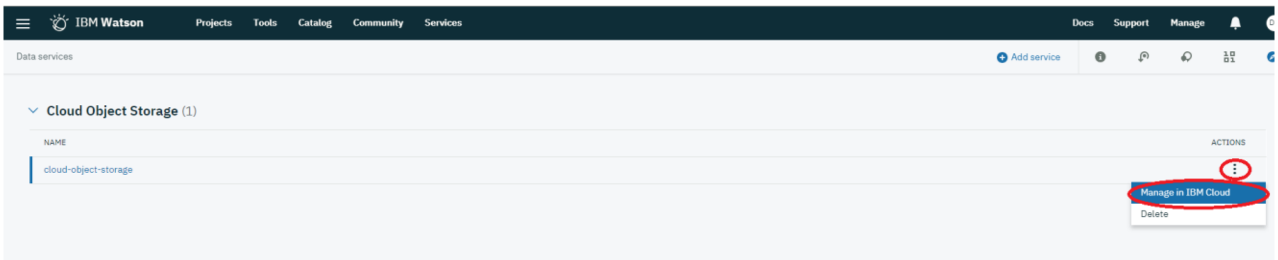
1. Download the mnist.zip file. Extract the 3 files - a training file (mnist-tf-train.pkl), test file(mnist-tf-test.pkl), and a validation file (mnist-tf-valid.pkl) in pickle format.



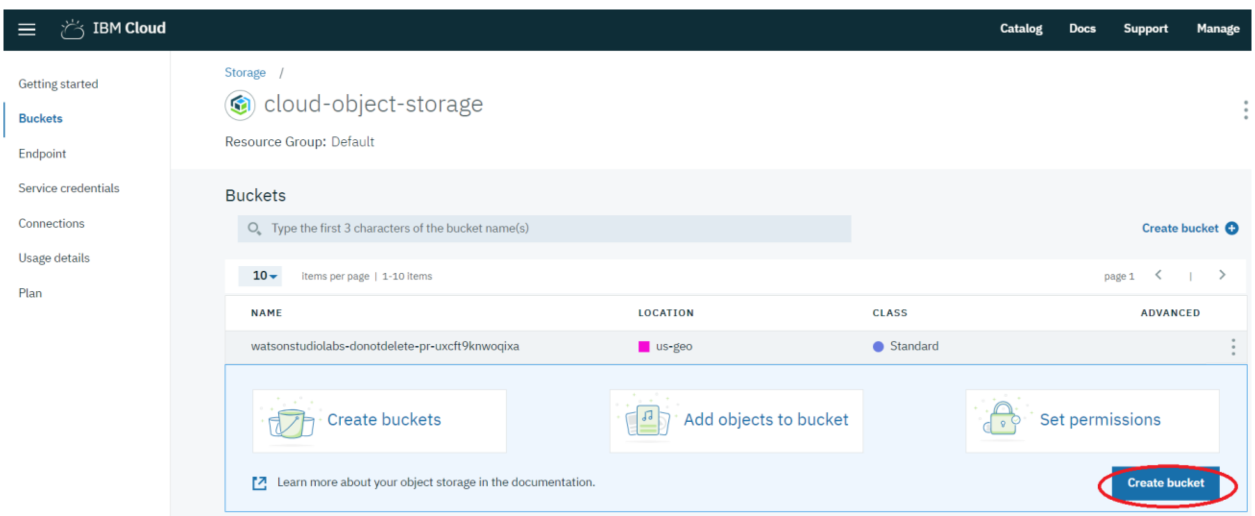

2. Select the **Services** tab and the click on **Data Services**



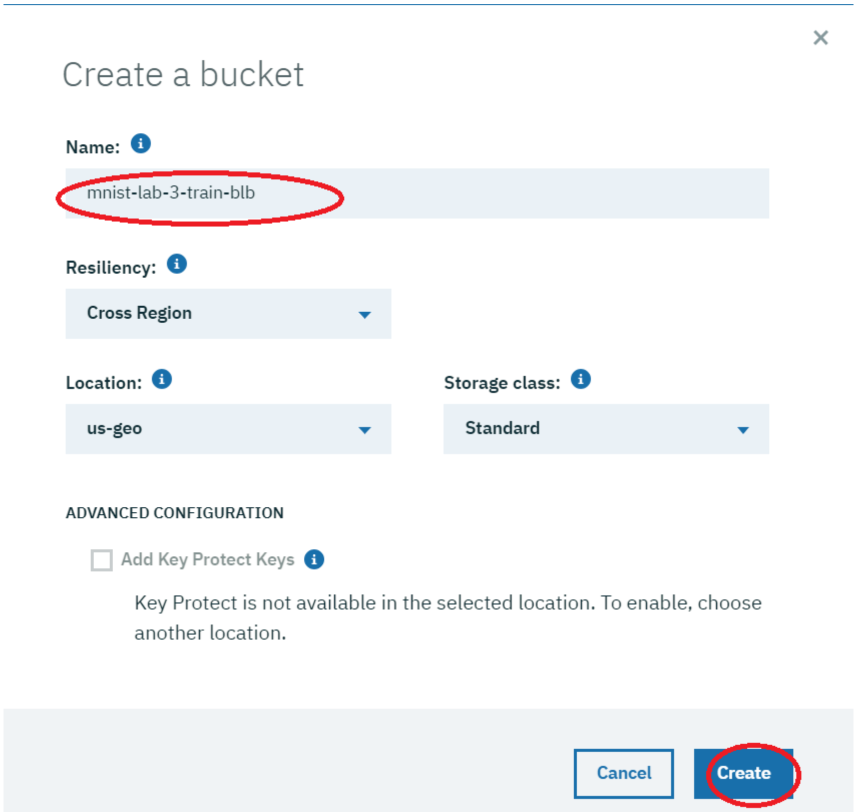

3. Select the vertical **elipse** and then click on **Manage in IBM Cloud**



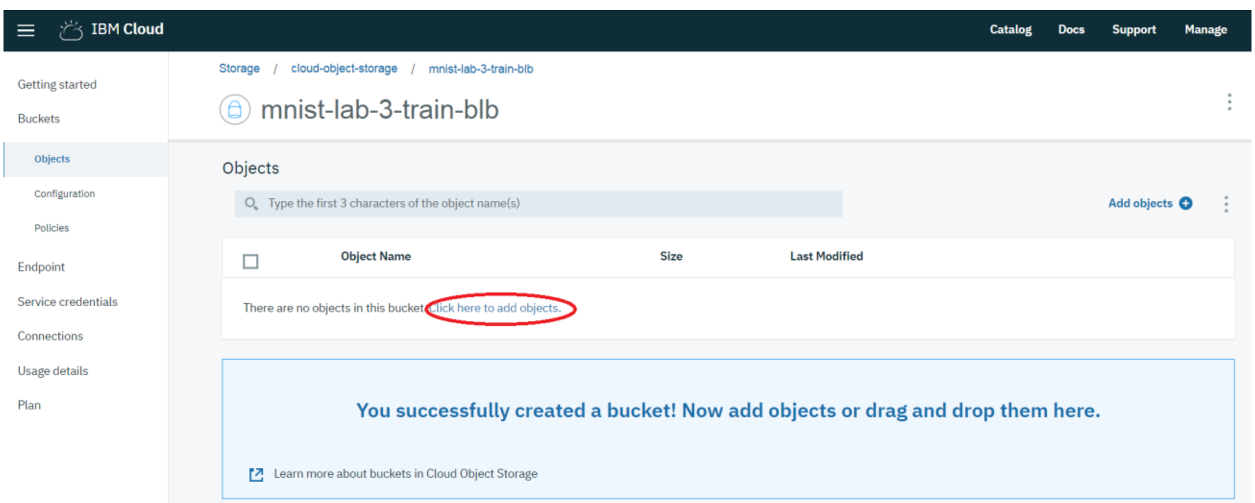

4. Click on **Create Buckets**



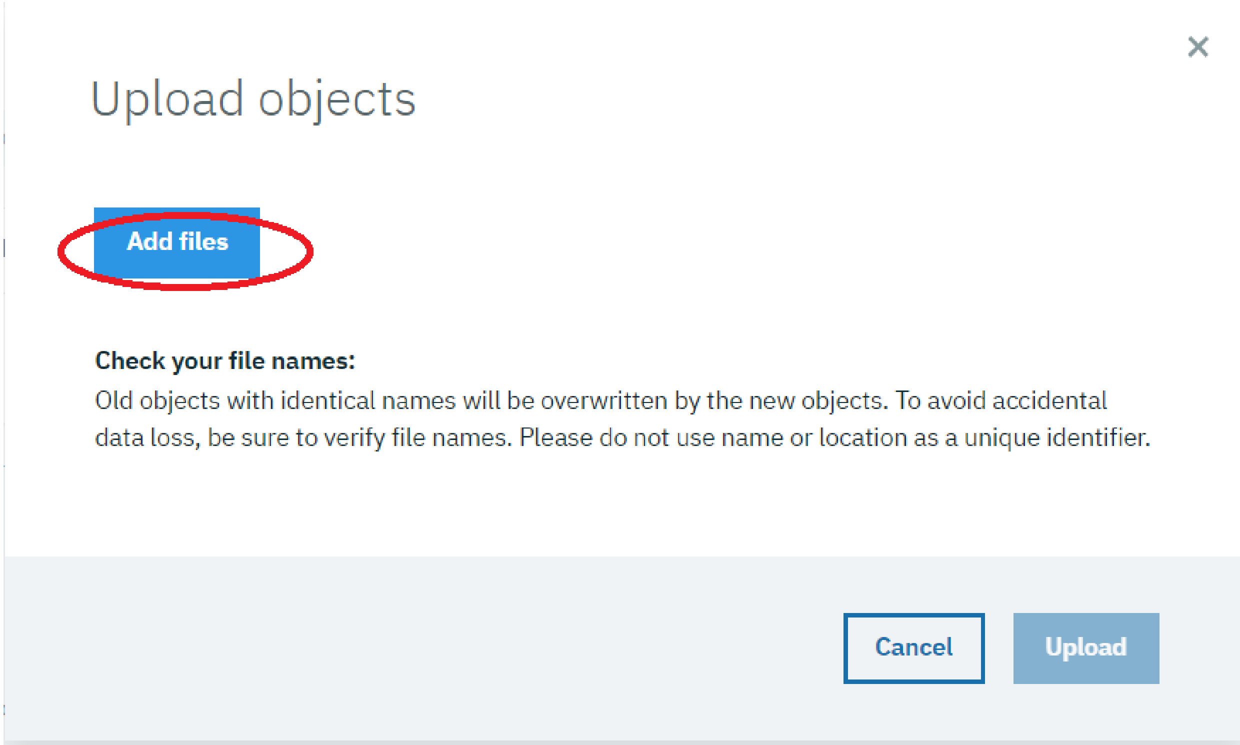

5. Enter a unique name for the bucket - mnist-lab-3-xxx (replace xxx with your initials), and click **Create**.



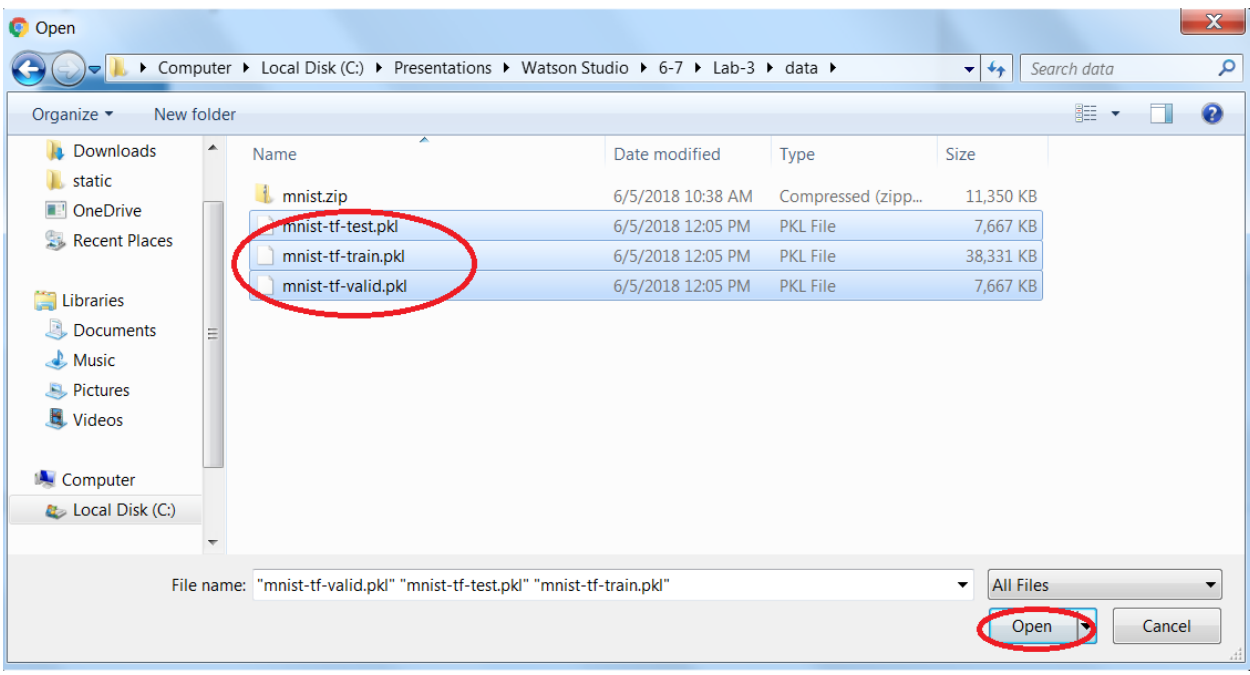

6. Click on **click here to add objects** to add files to the bucket.



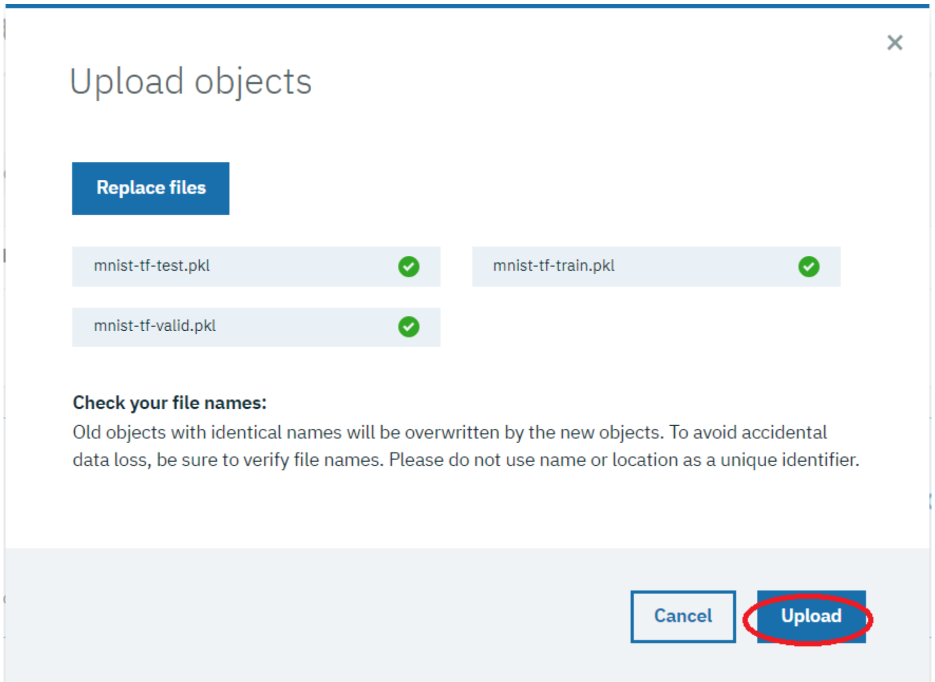

7. Click on **Add Files**



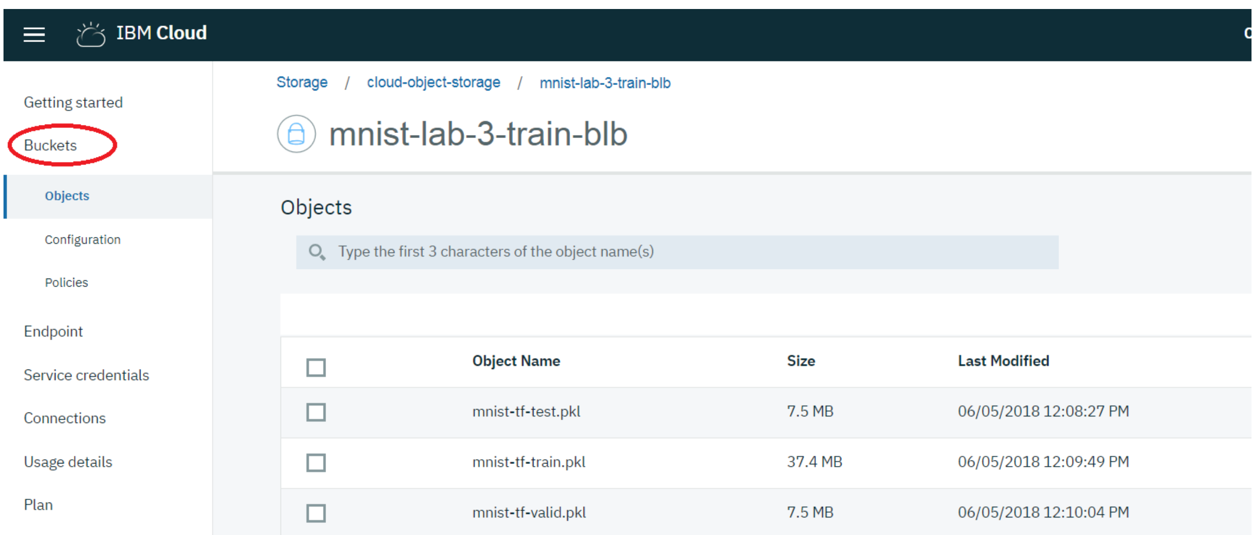

8. Navigate to where you stored the 3 pickle files. Select the files and click **Open**



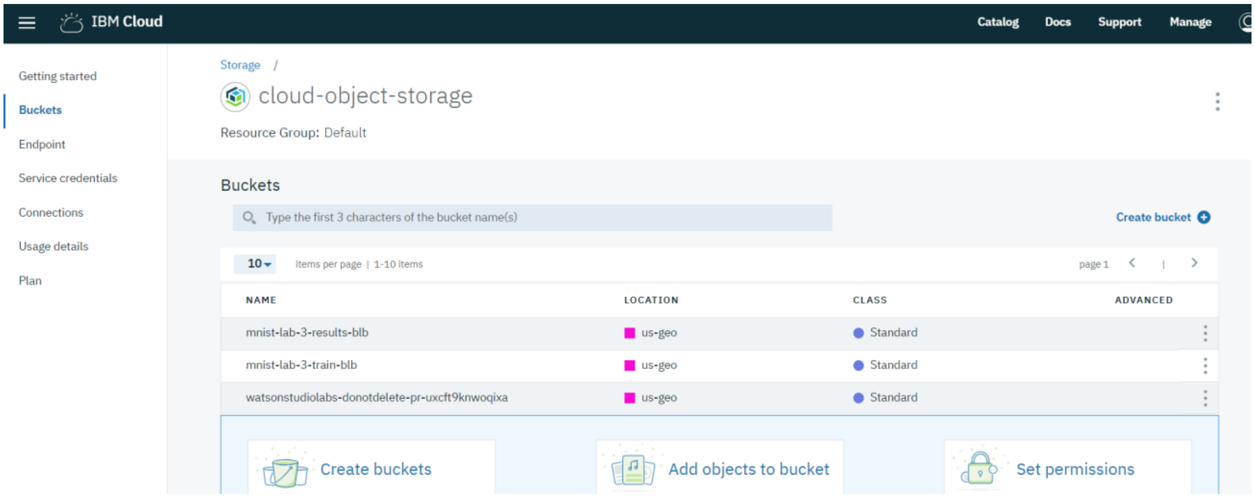

9. Click on **Upload**.




10. Click on Buckets to add a second bucket. Name it mnist-lab-3-xxx, where xxx are your initials. Follow the procedure above to create the second bucket. No files need to be added.

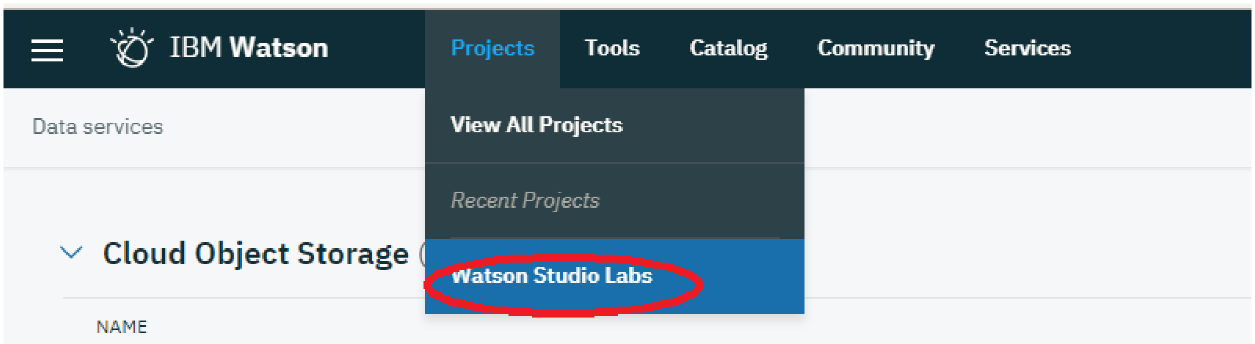



11. The Cloud Object Storage panel should appear as below.

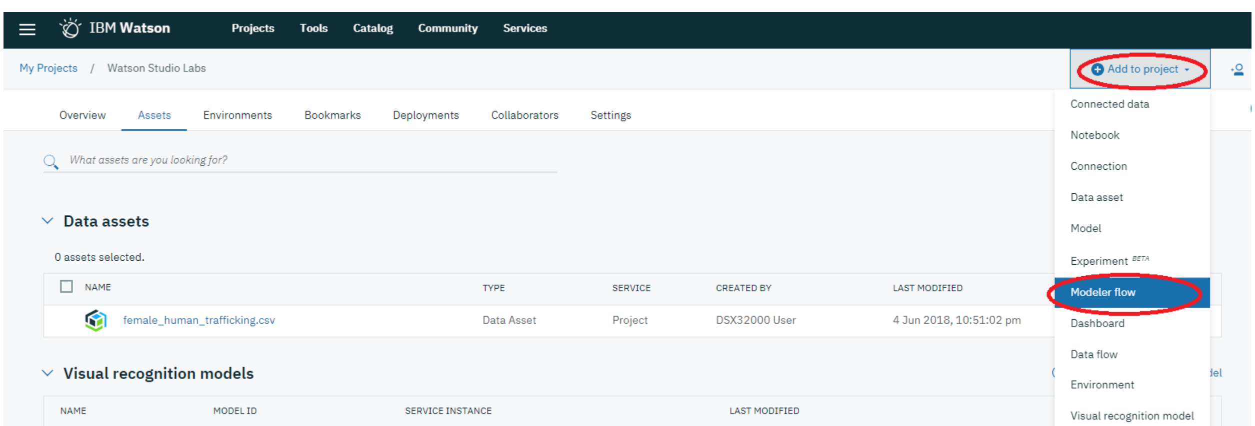



## Step 2: Design the Neural Network and Publish Training Definition

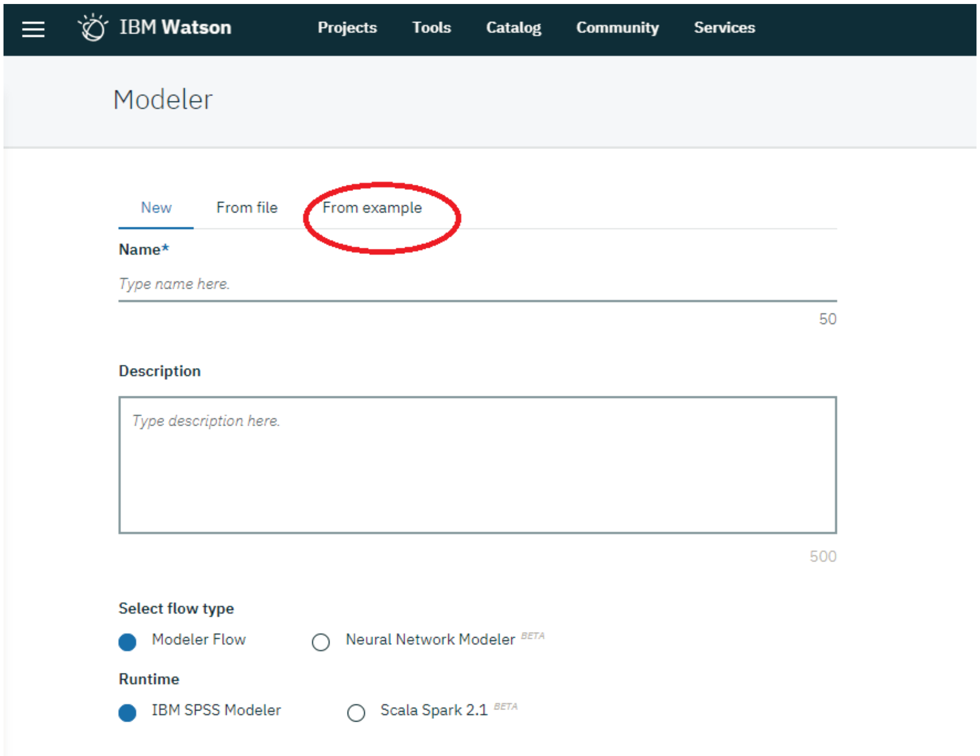
1. Return to Watson Studio, and click on the **Projects** tab, and **Watson Studio Labs**.



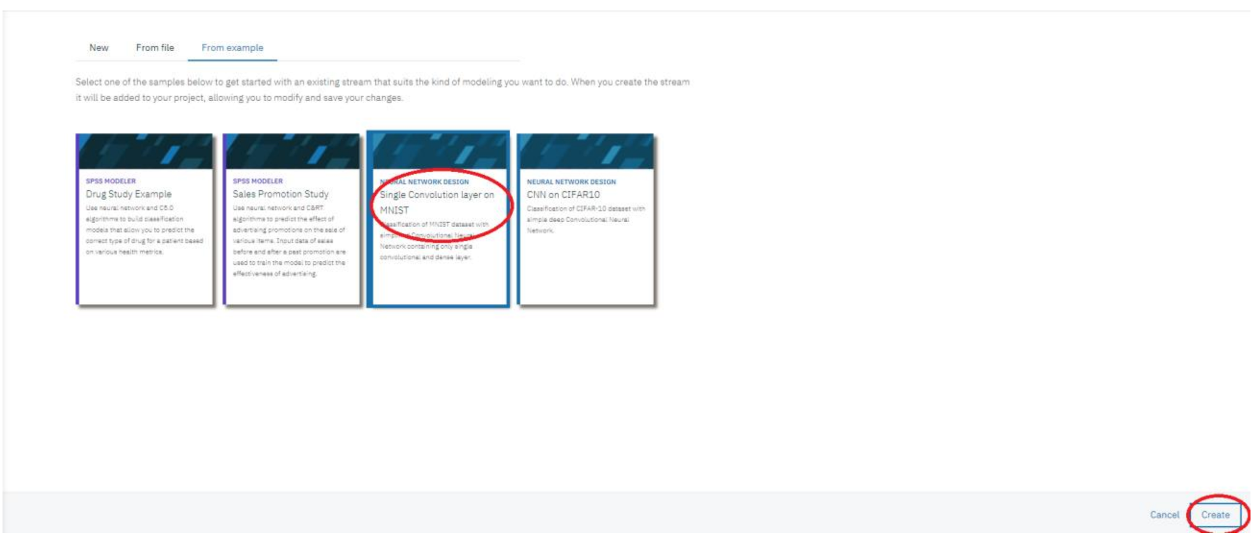

2. Click on the **Add to project** and then click on **Modeler Flow**.



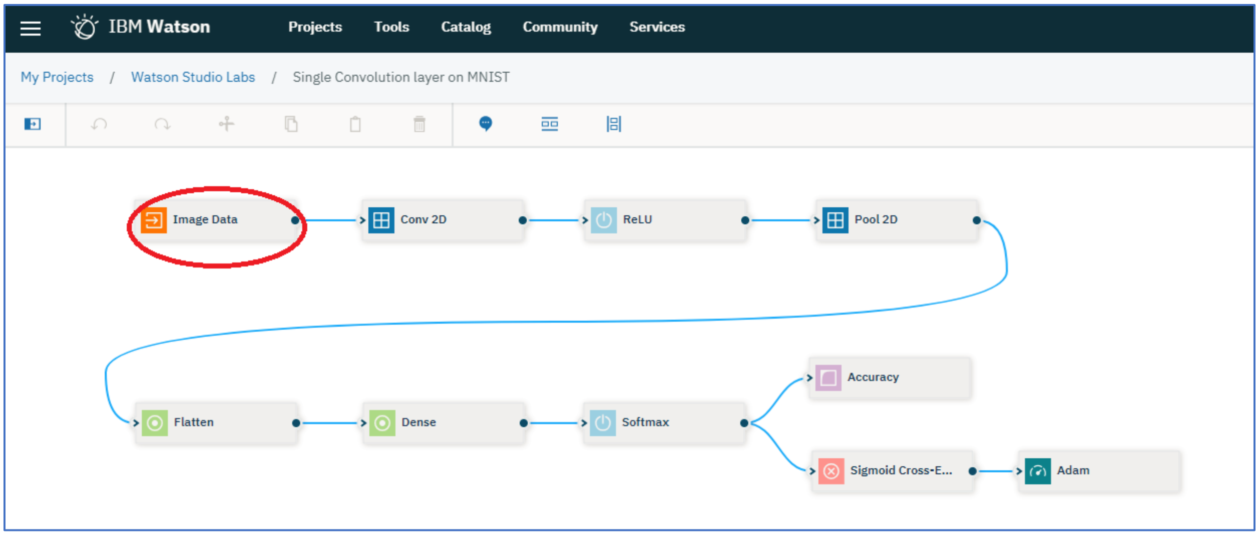

3. Click on **From example**.



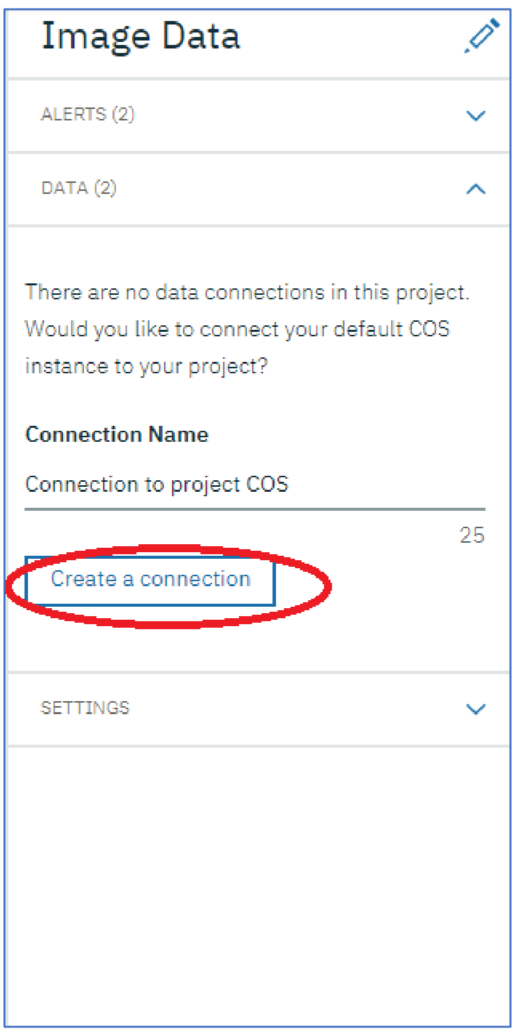

4. Click on the **Single Convolution Layer on MNIST** and then click on **Create**



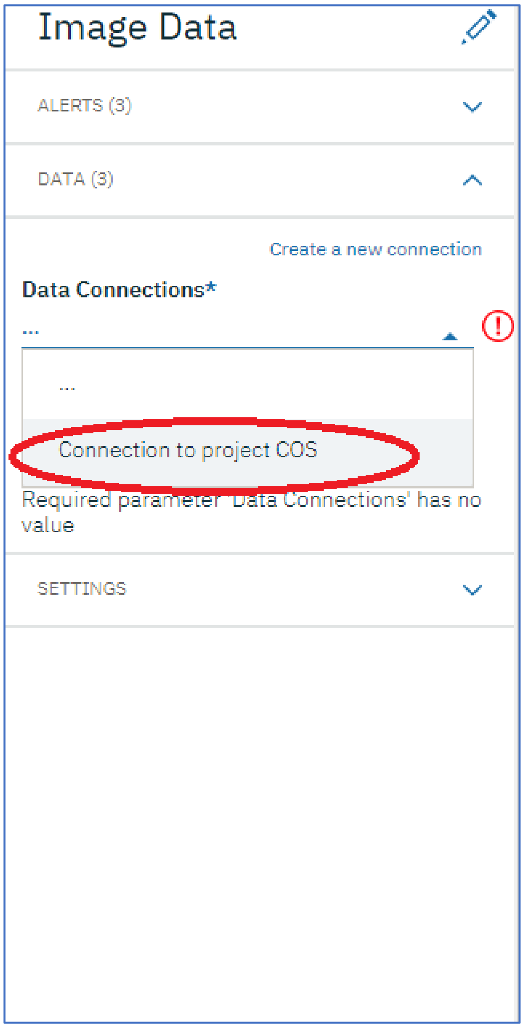

5. We need to configure the Image Data node. Double-click on Image Data.



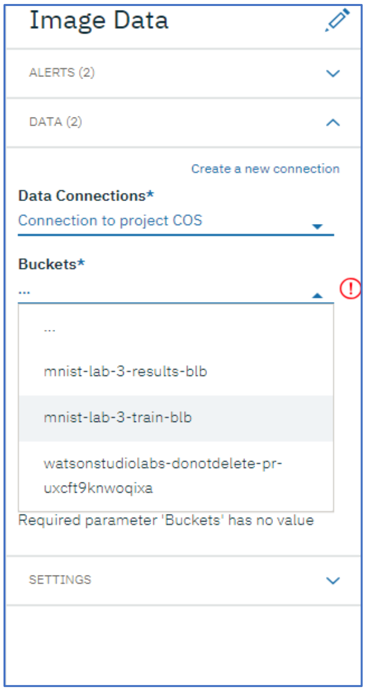

6. Optionally change the default name, and click on **Create a connection**.



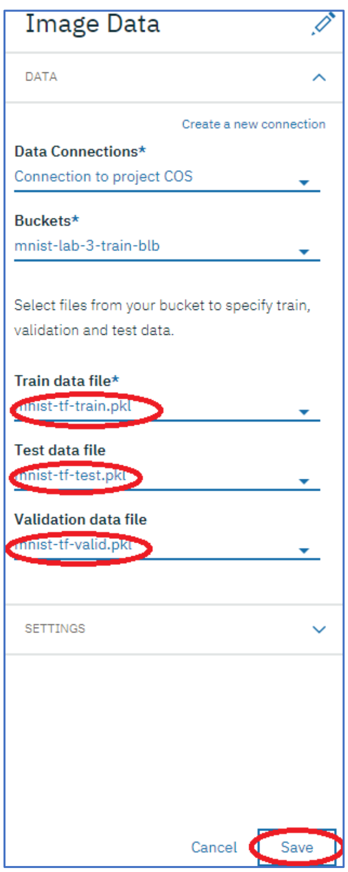
7. Click on the downward triangle icon underneath **Data Connections*.** Click on the connection that was just created.

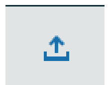


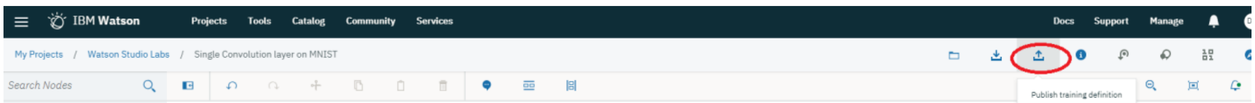

8. Click on the downward triangle icon underneath **Buckets***, and then click on the **mnist-lab-3-train-xxx** where "xxx" are your initials.



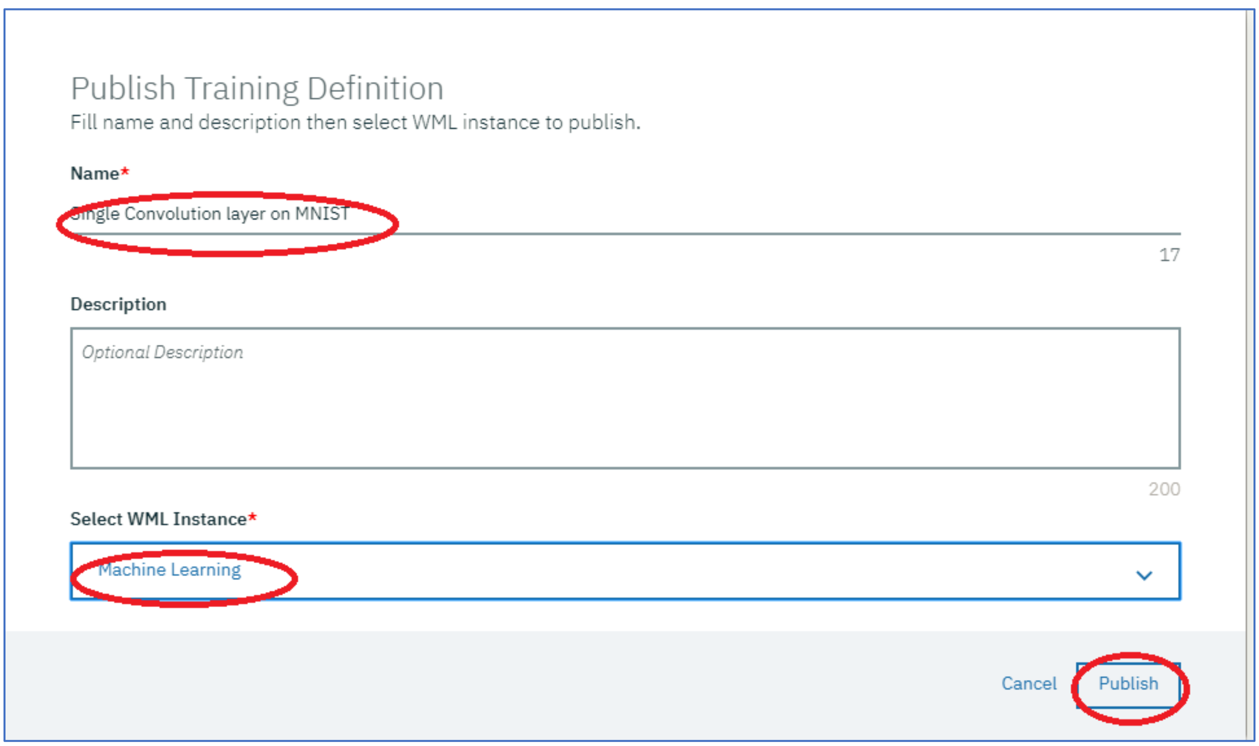

9. Click on the ▾ icon underneath **Train data file***, and select the **mnist-tf-train-pkl.** Assign the Test data file(mnist-tf-test-pkl), and Validation data files(mnist-tf-valid-pkl) in the same way and then click on **Save**.



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.



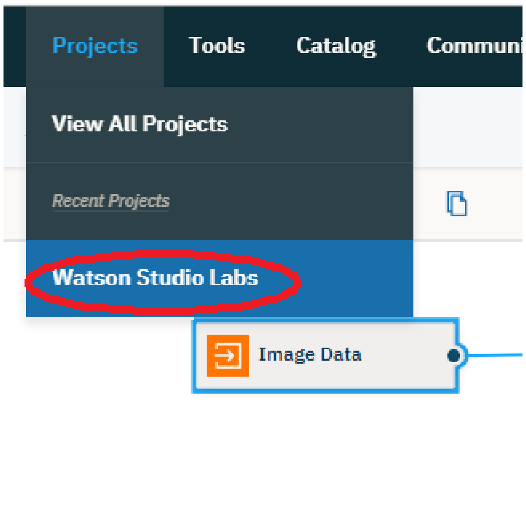

12. Enter a name for the training definition ( or leave the default), and select the Machine Learning service that you created. Note, it will not be named Machine Learning unless that is the name that you used. Click on Publish.



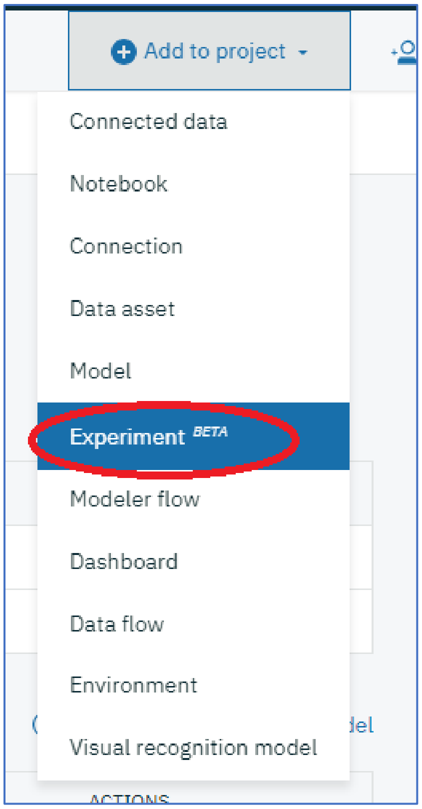
## Step 3: Train the Model using Experiment Builder

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 have defined only one training run to minimize the training time.

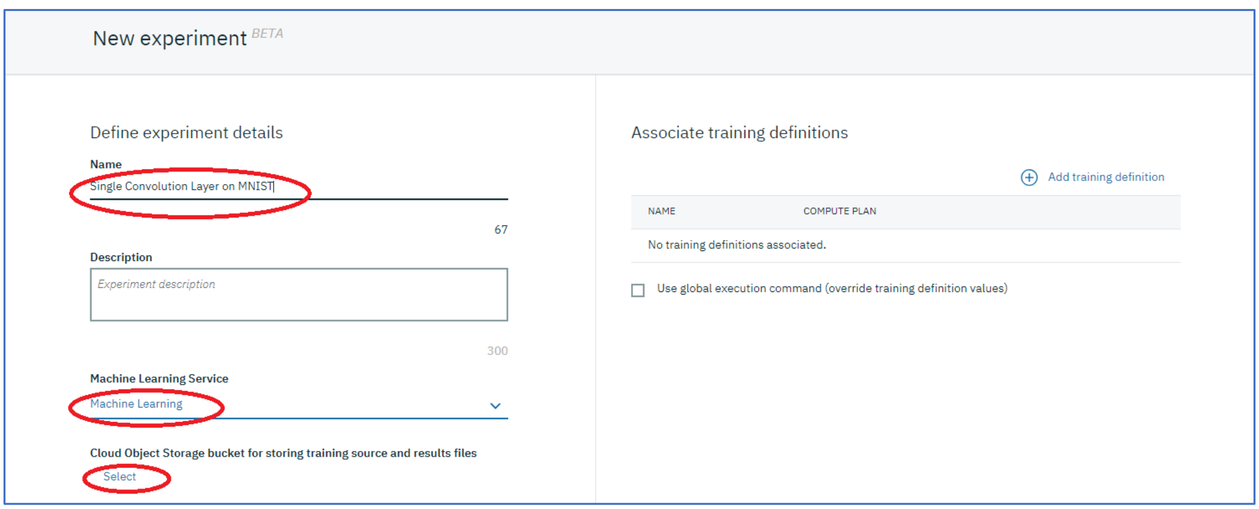
1. Return to the Watson Studio Labs Assets panel by clicking on the **Projects** tab and then **Watson Studio Labs**. Click on the **Assets** tab if the Assets panel is not displayed.



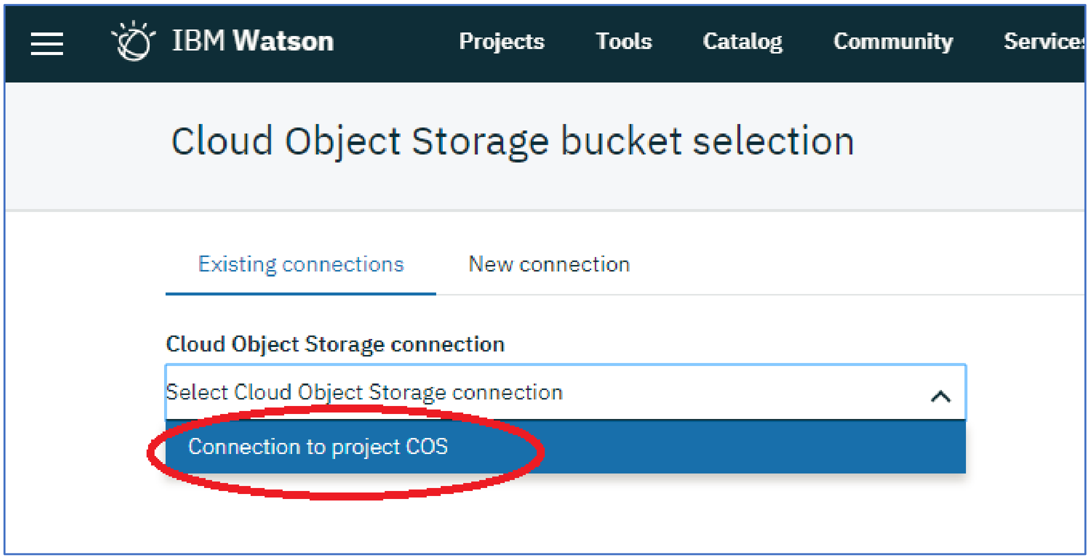

2. Click on **Add to project**, and then click **Experiment** to create a new Experiment.



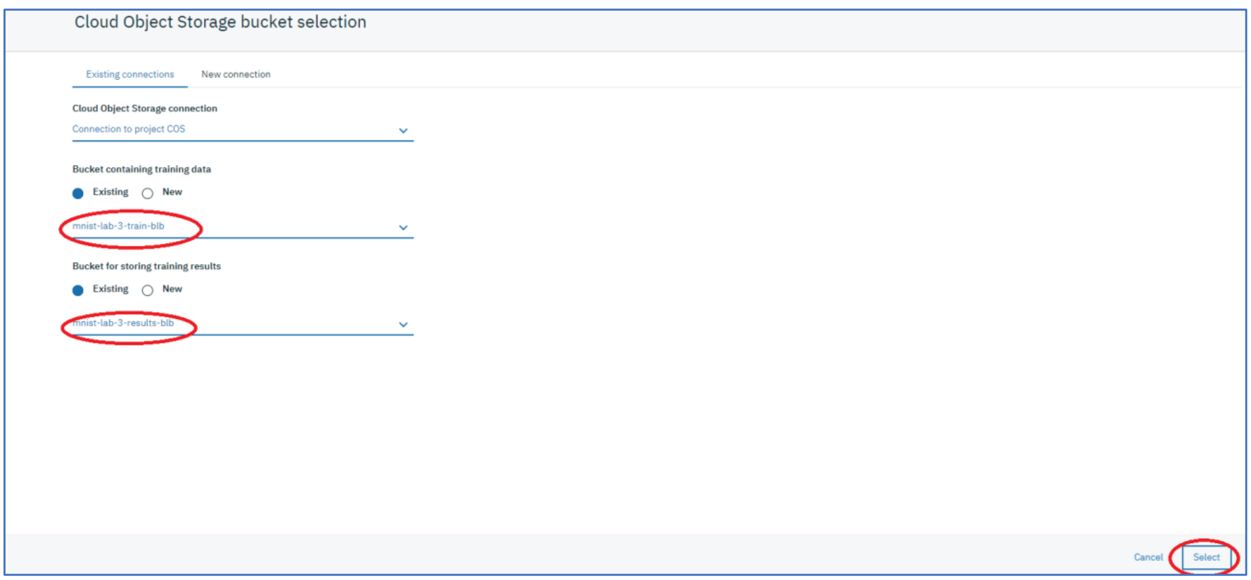

3. Enter an Experiment **Name**, select the **Machine Learning** service, and then click on **Select** to assign a Cloud Storage bucket.



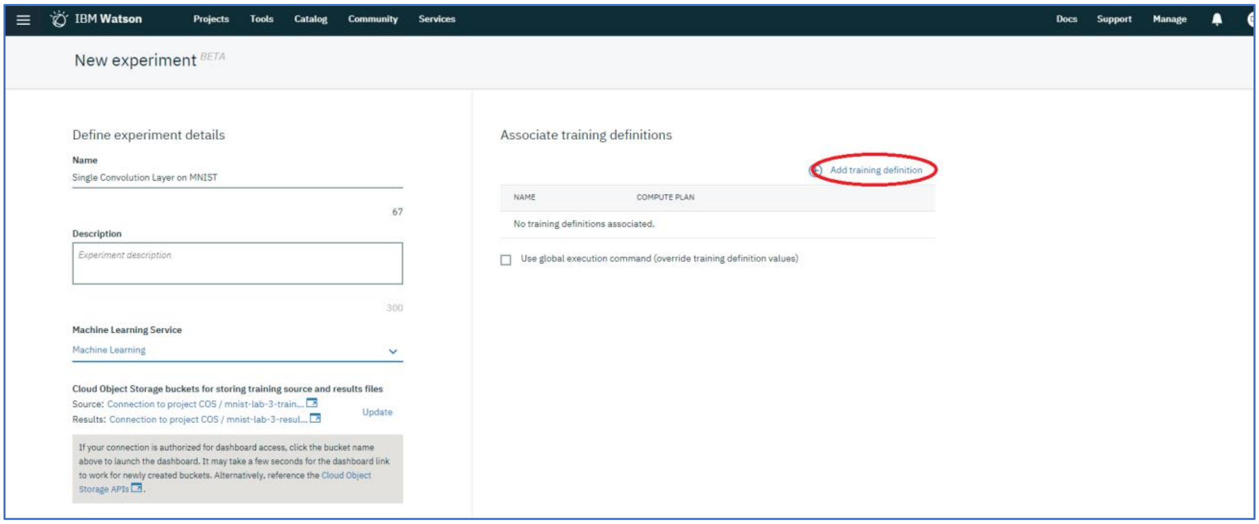

4. Select **Existing connections**, and then select the **Connection to project COS** connection.



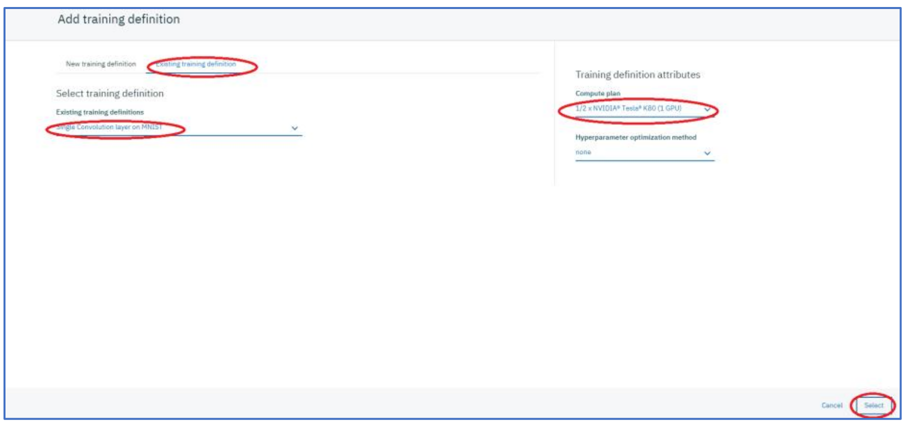

5. We now need to assign the Training and Results buckets. Select **Existing** underneath **Bucket containing training data**, and click on mnist-lab-3-train-xxx, where “xxx” are your initials. Select **Existing** underneath **Bucket for storing training results,** and click on mnist-lab-3-results-xxx,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



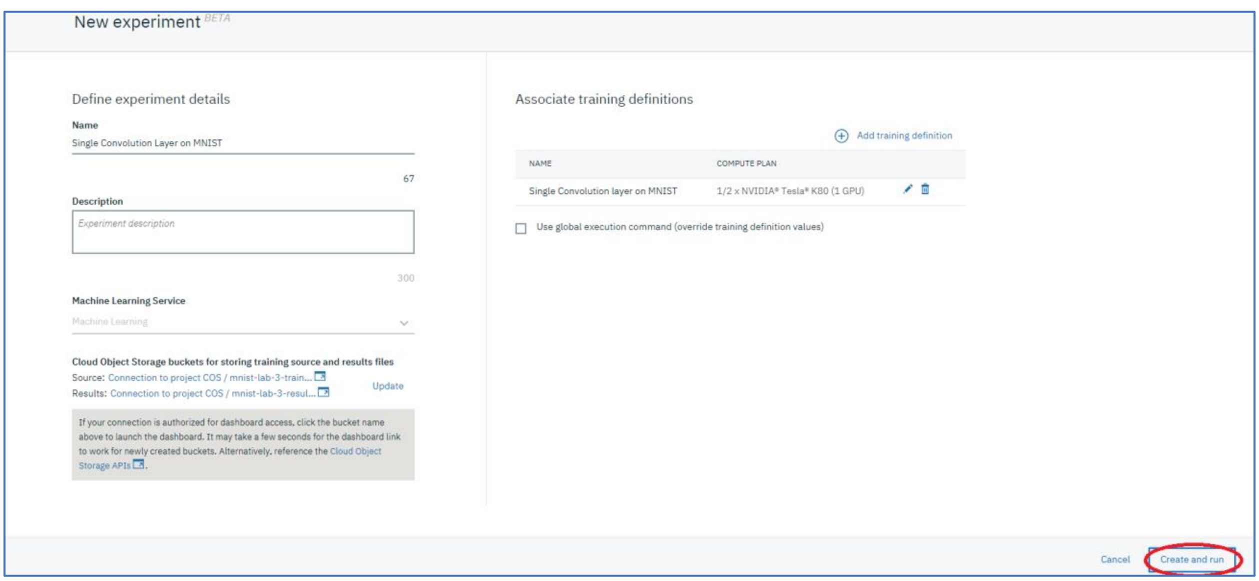

6. We now need to assign a Training Definition. Click on **Associate Definition**.




7. 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**.



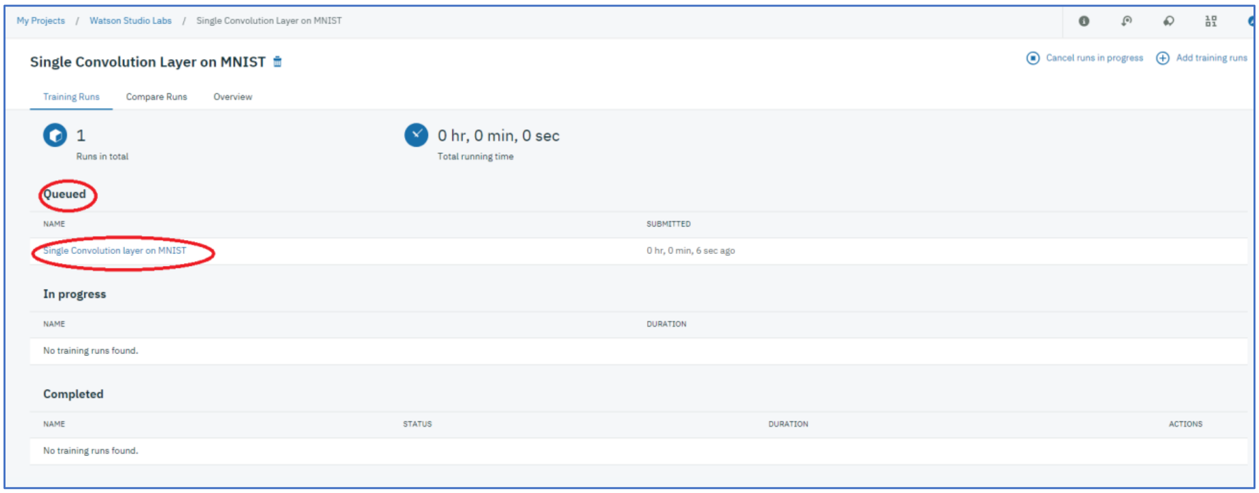

8. Click **Create and run**.



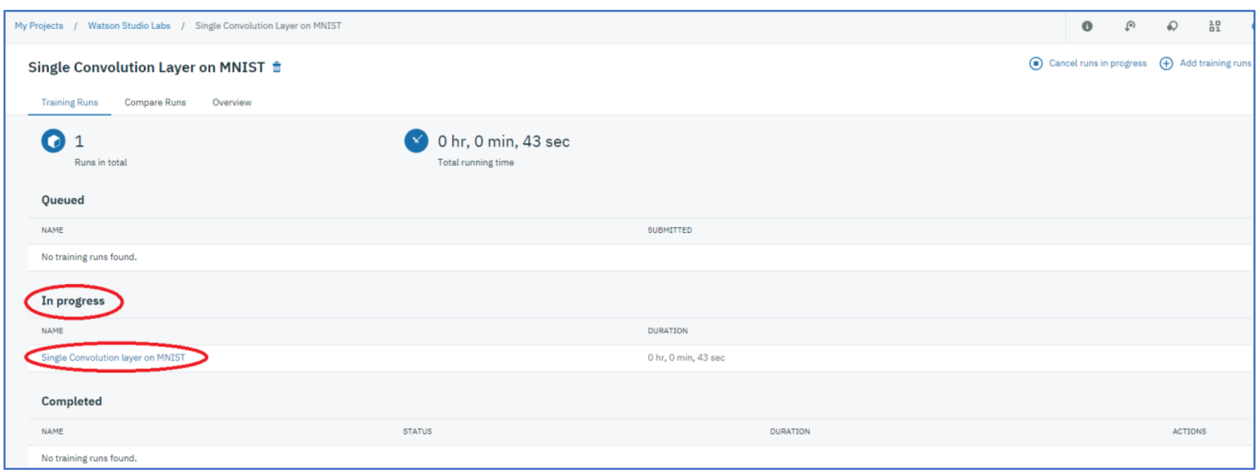
## Step 4: Monitor the Training Progress and Results

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

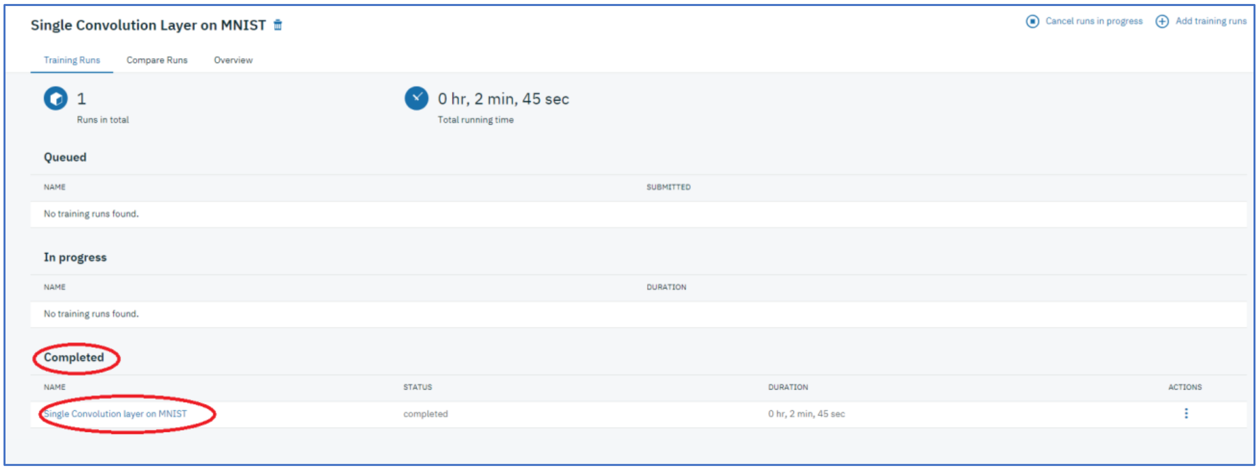
### Queued Status



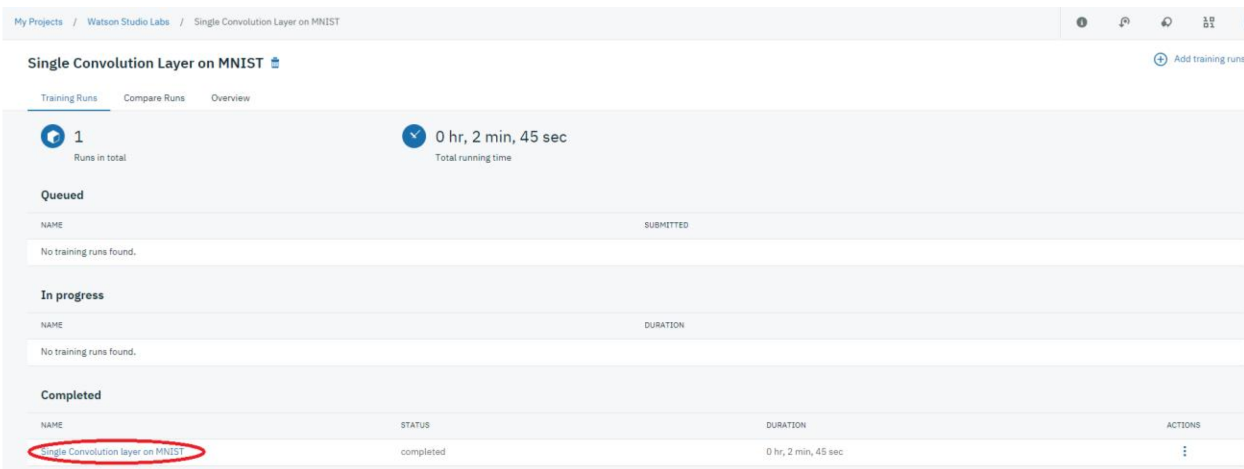

### In-Process Status



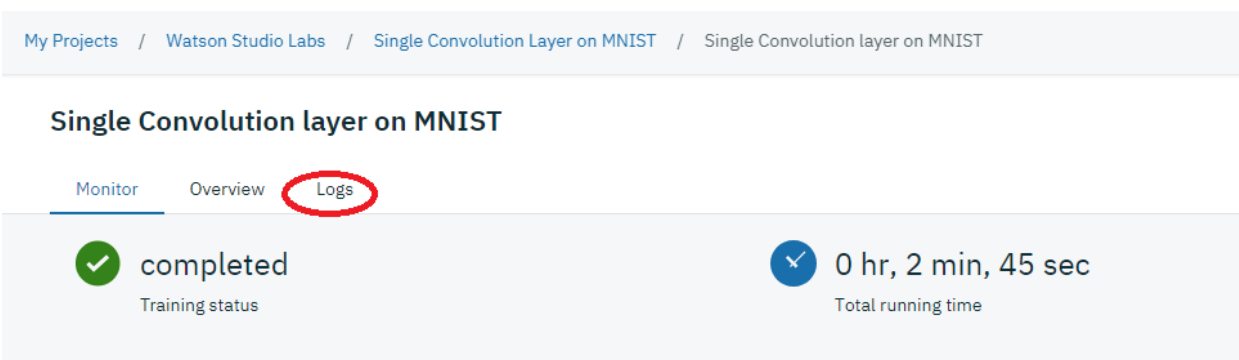

### Completed Status



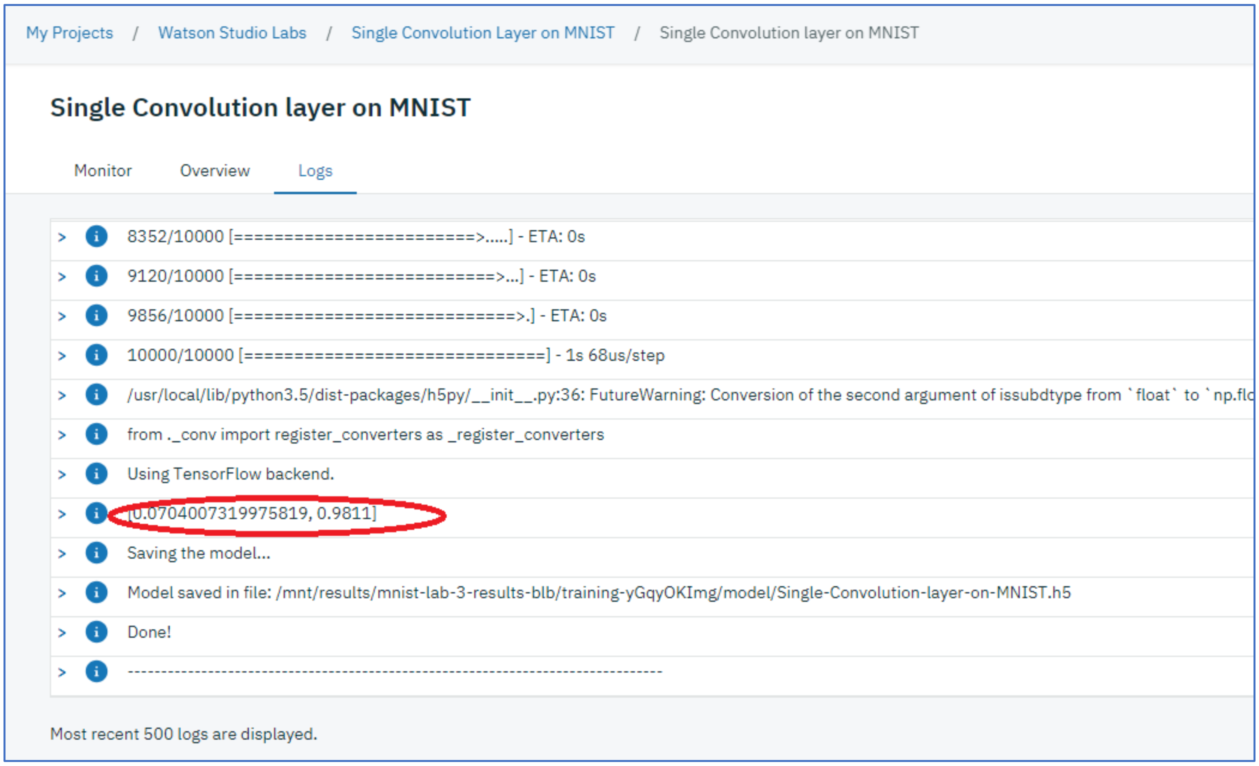

1. Click on the Single Convolution layer on MNIST link to check the results.




2. Click on **Logs**



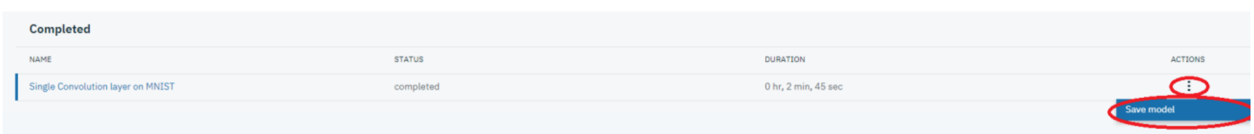

3. Scoll down to the bottom to check accuracy measure.



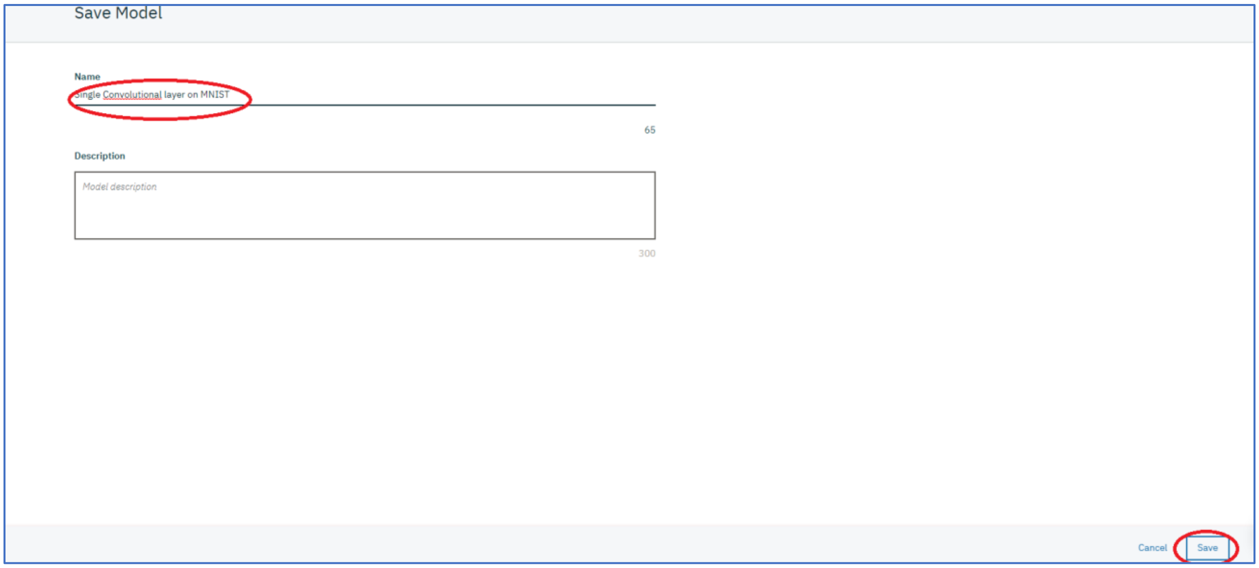

## Step 5: Save and Deploy the Trained Model

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

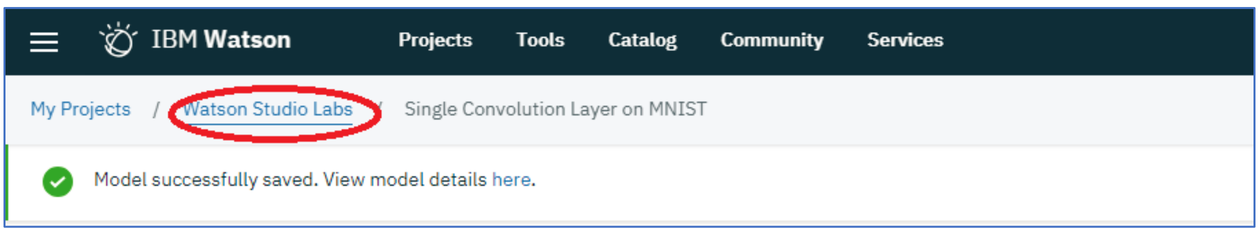
1. Click on the vertical ellipse under ACTIONS, and click **Save model**.



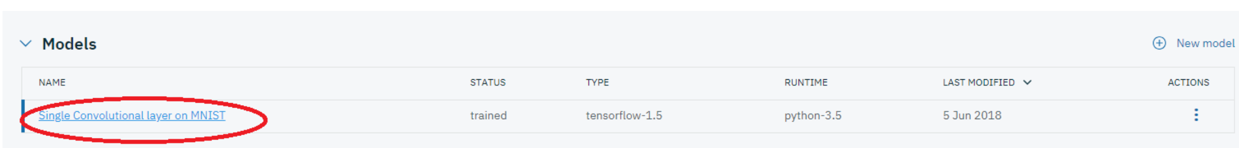

2. Enter a **Name** for the model (Single Convolution layer on MNIST) and click **Save.**



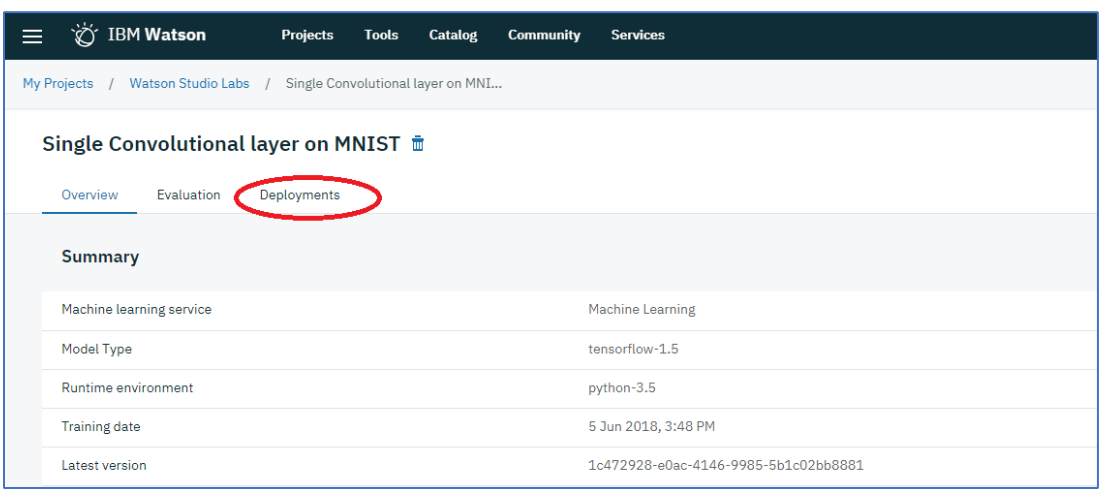

3. Return to the Watson Studio **Assets** panel, by clicking on **Watson Studio Labs** in the breadcrumb path. Click on the **Assets** tab if the Assets panel is not showing.



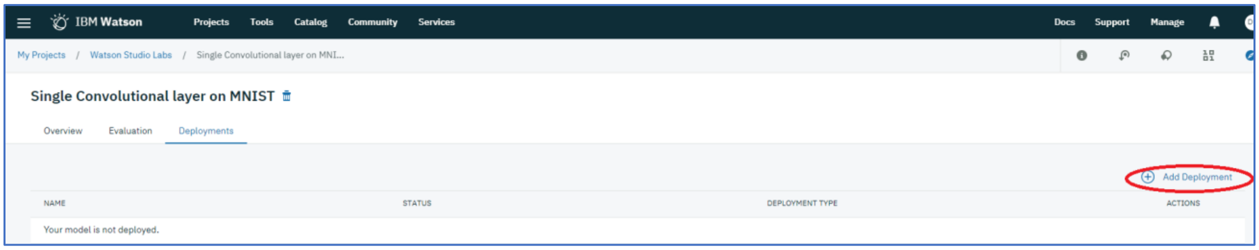

4. Click on the newly saved model



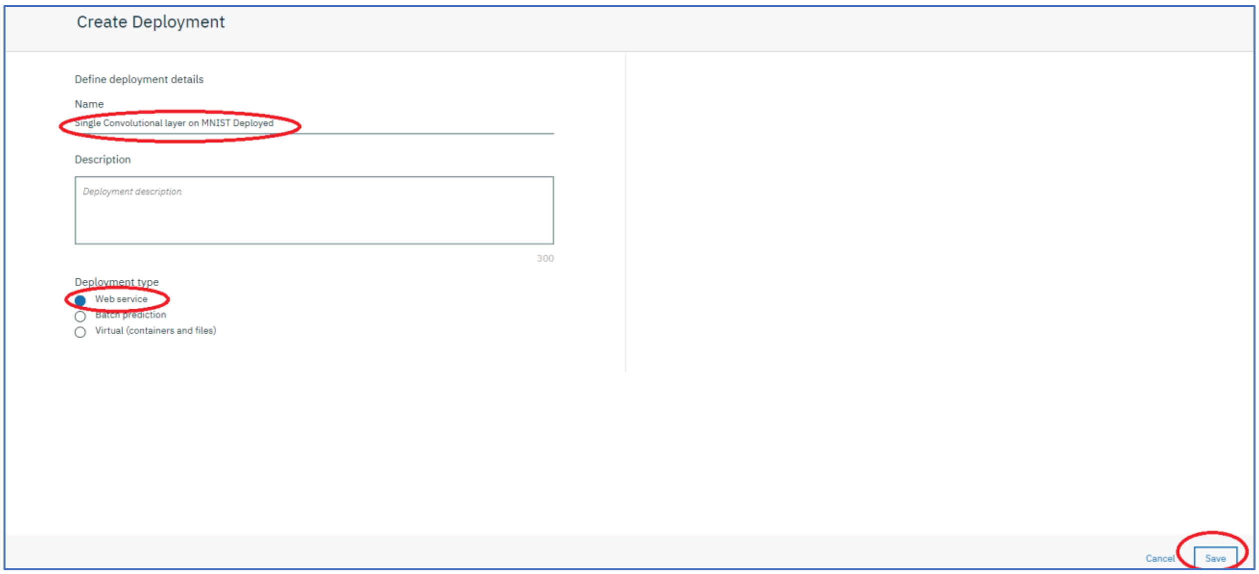

5. Click on **Deployments.**



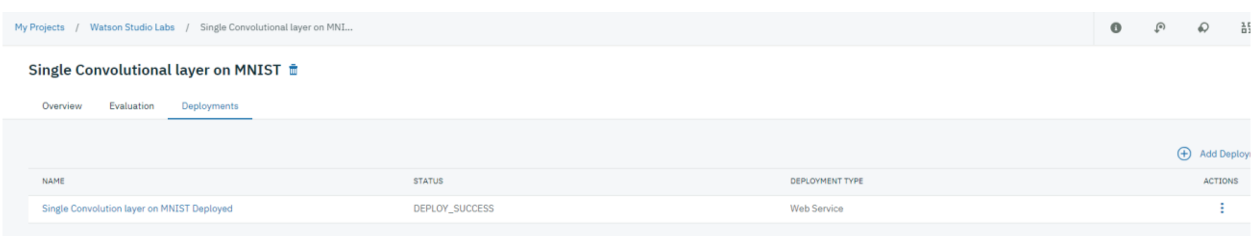
6. Click on **Add Deployment**.




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



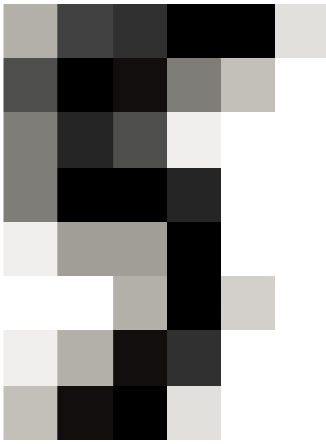

8. The Model **STATUS** should indicate DEPLOY_SUCCESS.




## Step 6: Test the Deployed Model

1. Download this sample test file: mnist-scikit-learn-test-payload.json.

That test image is for the digit "5" and looks like this:



2. In the **Test** area of the deployment details page in Watson Studio, paste the contents of mnist-scikit-learn-test-payload.json:

```
{
"fields" : [ "f0", "f1", "f2", ...
"values" : [ [ 0.0, 5.0, 12.0, ...
}
```

3. Click **Predict**.

   The results will look something like this:

```
{
  "fields": [
    "prediction"
  ],
  "values": [
    [
      5
    ]
  ]
}
```

   The `values` field contains the digit class ("0" through "9") that the model has classified the test data into. The expected digit class is "5".