

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

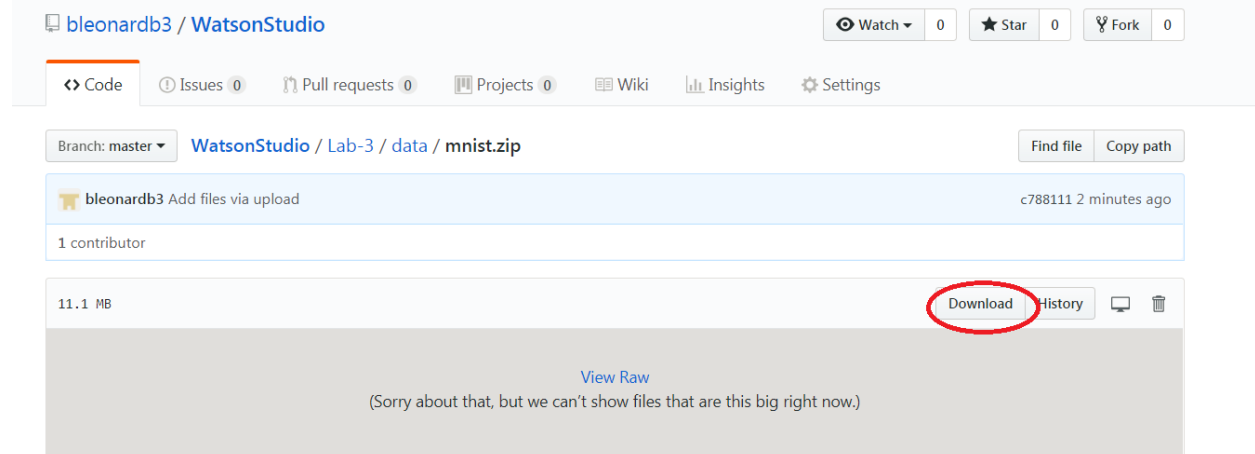
This lab will use the [MNIST](#) computer vision data set to train a deep learning model to recognize handwritten digits. A single layer convolutional neural network will be built in the Watson Studio neural network designer, and then trained using the Watson Studio Experiment Builder. The trained model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

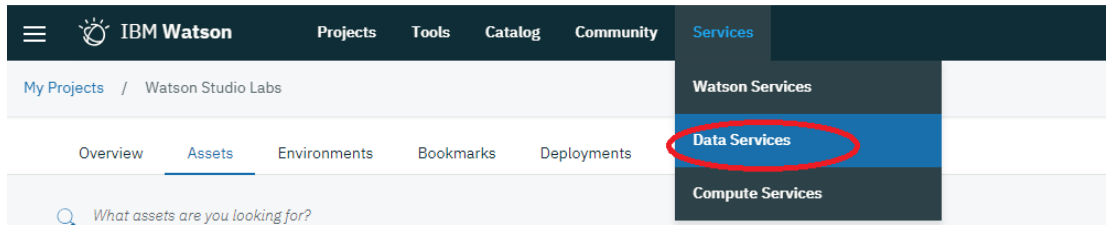
Step 1: Set up the Data Files in IBM Cloud Storage

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.)

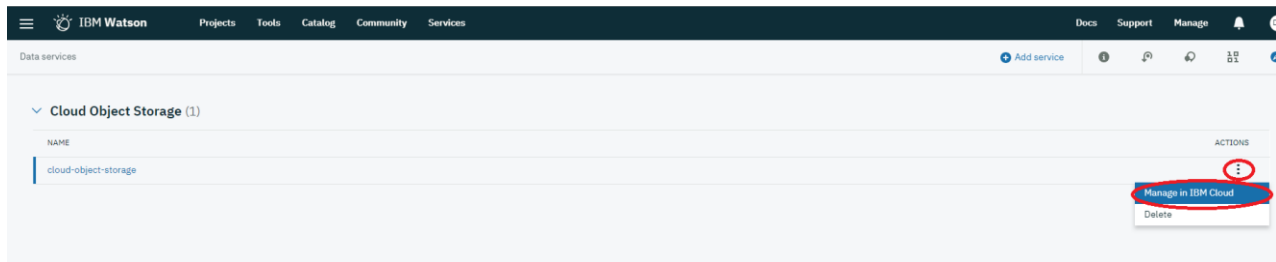
1. Download the [mnist.zip](#) file. Extract the 3 files - a training file (mnist-tf-train.pkl), test file (mnist-tf-test.pkl), and a validation file (mnist-tf-valid.pkl) in pickle format.



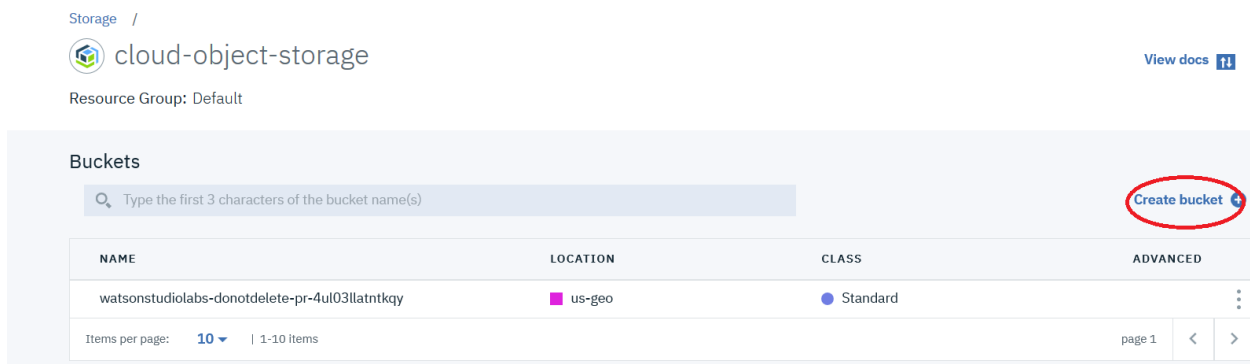
2. Select the **Services** tab and the click on **Data Services**



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



4. Click on **Create Bucket**



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

Create a bucket ×

Name: ⓘ

mnist-lab-5-train-blb

Resiliency: ⓘ

Cross Region

Location: ⓘ

us-geo

Storage class: ⓘ

Standard

ADVANCED CONFIGURATION

☐ Add Key Protect Keys ⓘ
Key Protect is not available in the selected location. To enable, choose another location.

Cancel

Create

6. Click on **Upload** and then **Files** to add files to the bucket.

Storage / cloud-object-storage / mnist-lab-5-train-blb

mnist-lab-5-train-blb

Objects

Type the first 3 characters of the object name(s)

Upload +

Files

<input type="checkbox"/>	Object Name	Size	Last Modified
<div><div>Drag and drop files to upload them.</div></div>			

7. Click on **Select Files**

Upload files ×

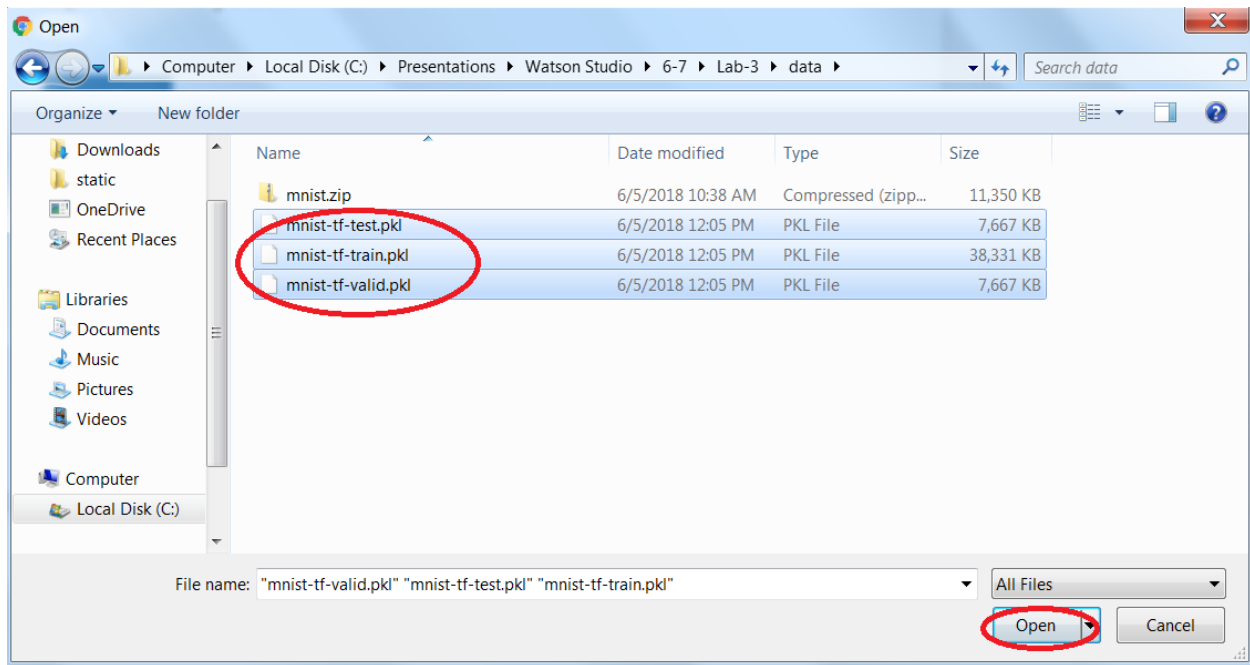
Verify object and folder names to prevent upload failures:

- Existing items with identical names will be replaced.
- Don't include any personal information such as name or address.
- Avoid these characters: / \ " : ' ? < > & * |

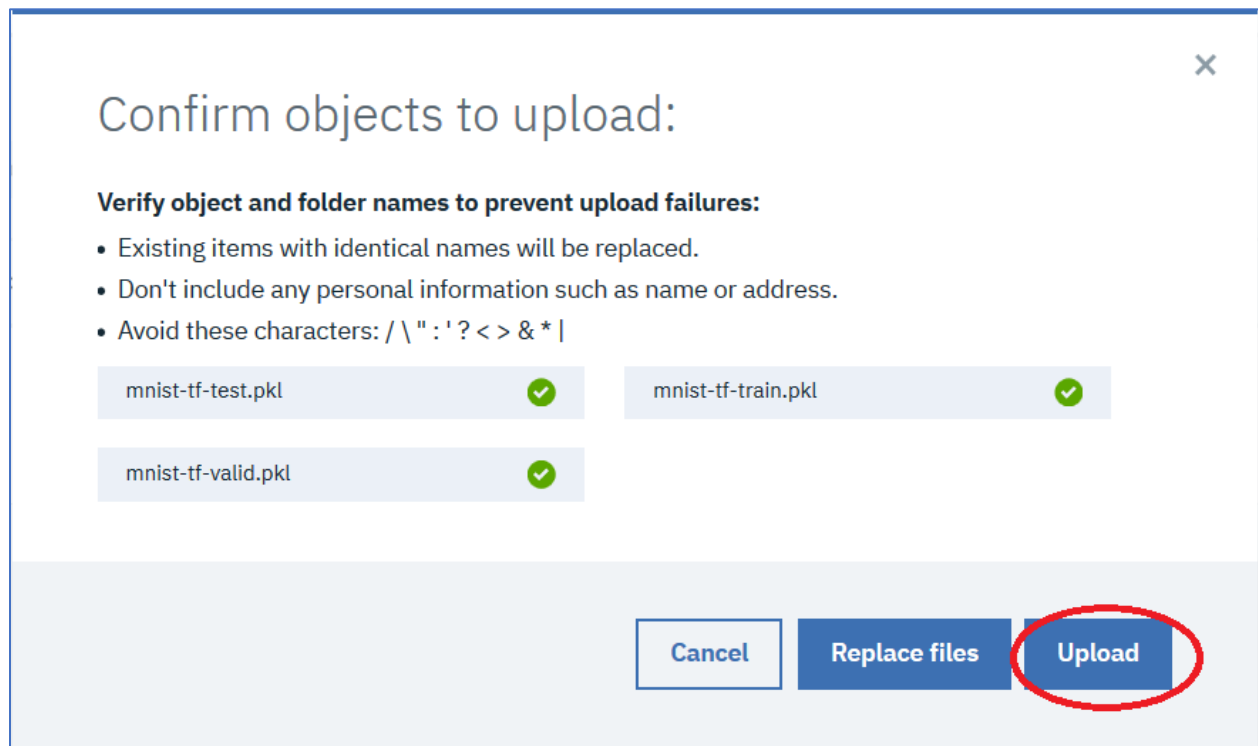
Cancel

Select files

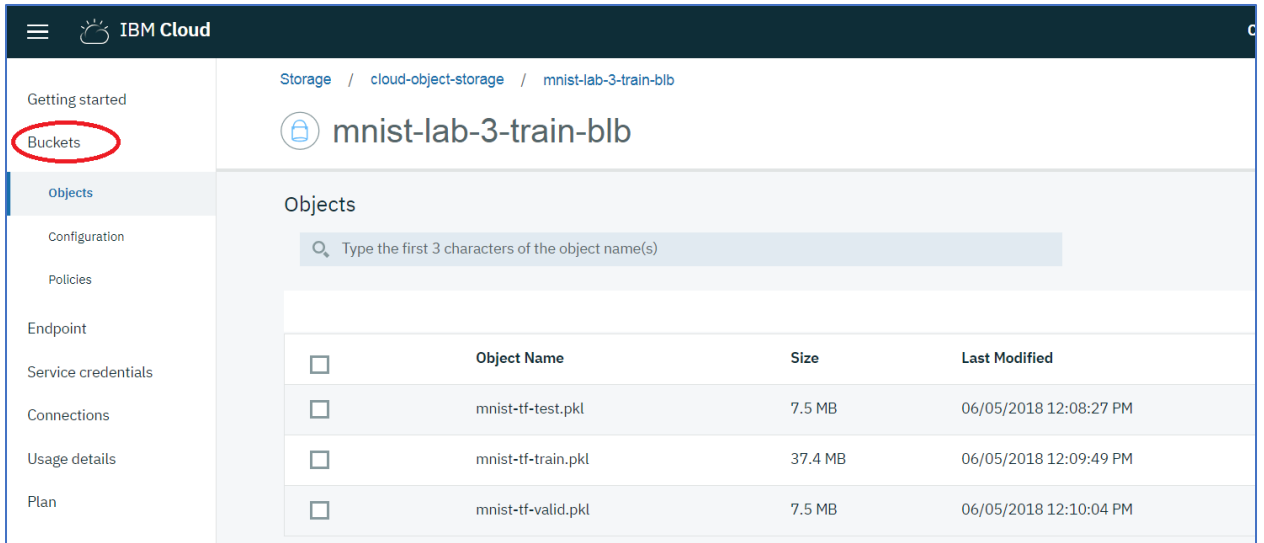
8. Navigate to where you stored the 3 pickle files. Select the files and click **Open**



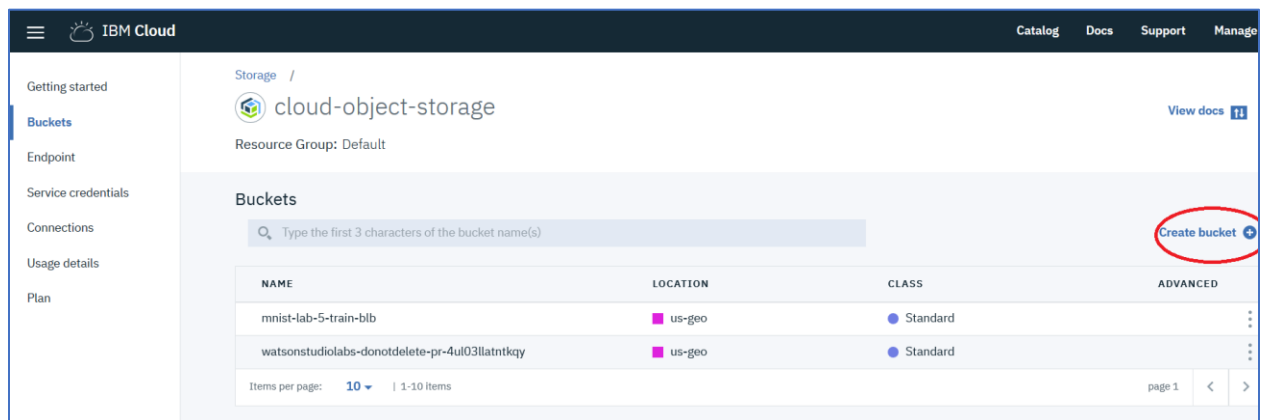
9. Click on **Upload**.



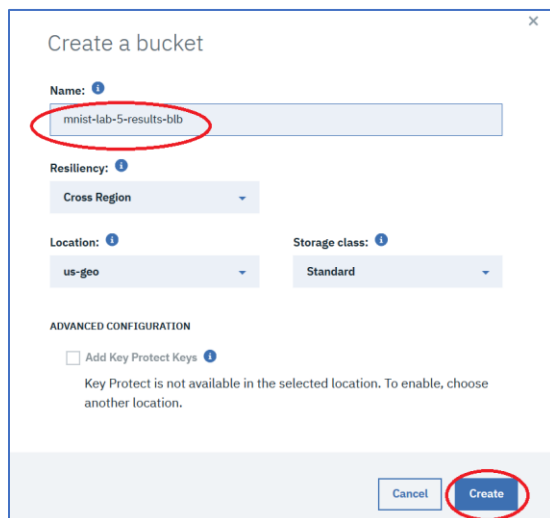
10. Click on Buckets to add a second bucket.



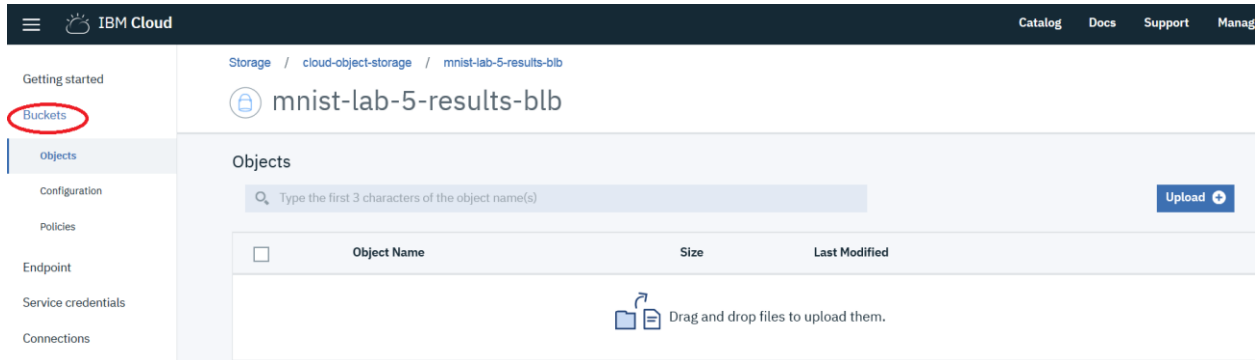
11. Click on **Create Bucket**.



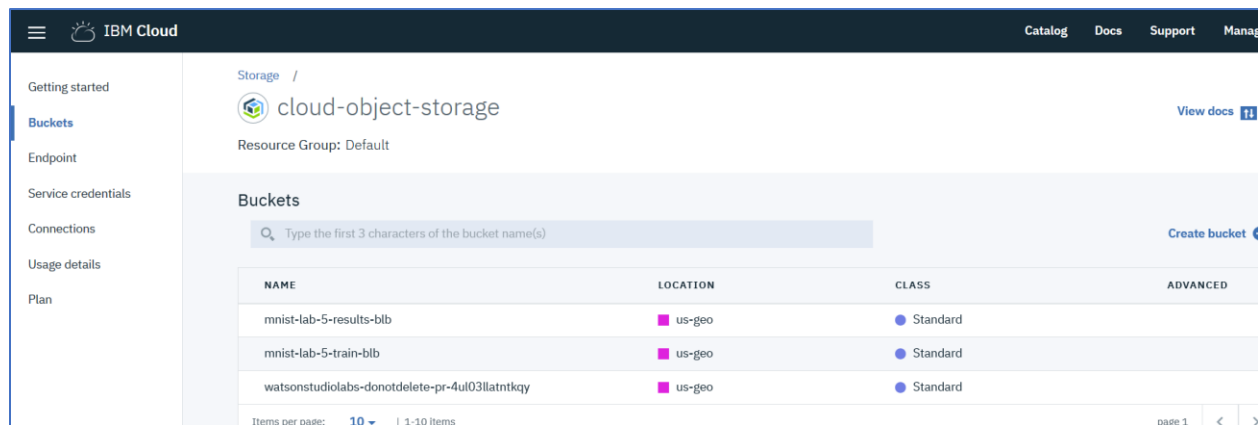
12. Name it mnist-lab-5-results-xxx, where xxx are your initials, and click **Create**.



13. No files need to be added to this bucket, so click on **Buckets**.

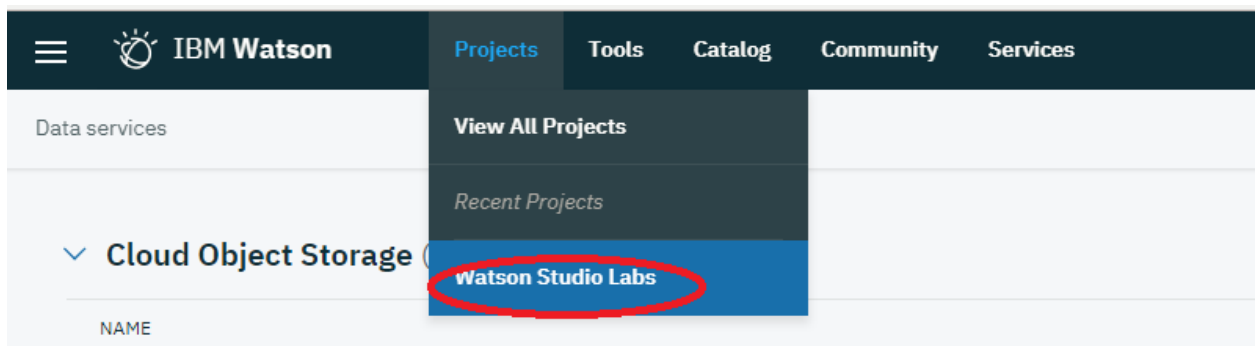


14. The Cloud Object Storage panel should appear as below.



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**.



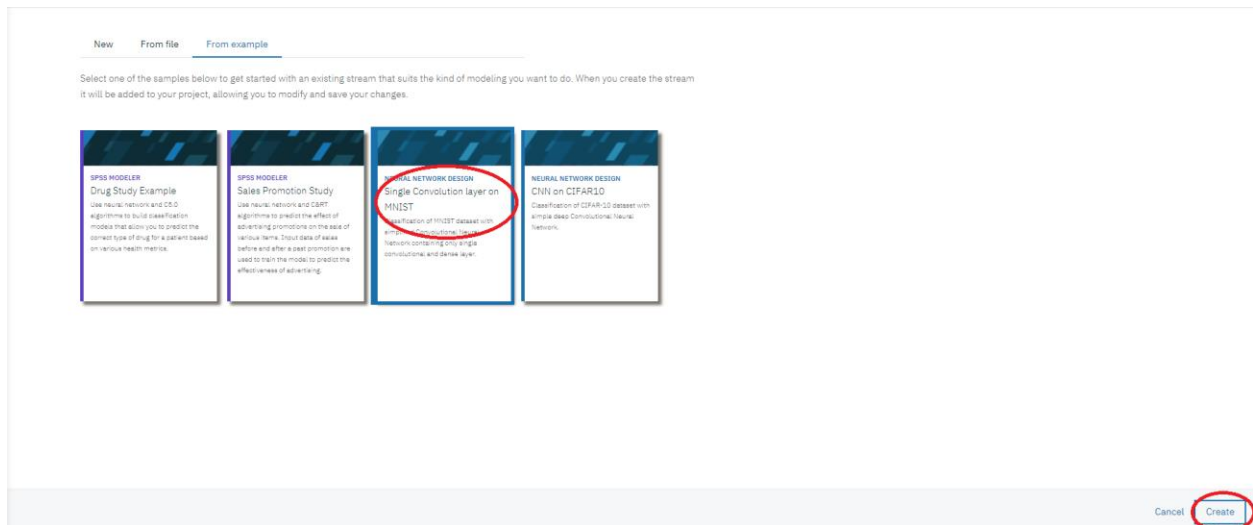
2. Click on the **Add to project** and then click on **Modeler Flow**.

The screenshot shows the IBM Watson Studio interface. At the top, there is a navigation bar with 'My Projects' and 'Watson Studio Labs'. Below this, there are tabs for 'Overview', 'Assets', 'Environments', 'Bookmarks', 'Deployments', 'Collaborators', and 'Settings'. The 'Assets' tab is selected. A search bar is present with the placeholder text 'What assets are you looking for?'. Below the search bar, there are two sections: 'Data assets' and 'Visual recognition models'. The 'Data assets' section shows '0 assets selected.' and a table with columns: NAME, TYPE, SERVICE, CREATED BY, and LAST MODIFIED. The table contains one row with the asset 'female_human_trafficking.csv'. The 'Visual recognition models' section shows a table with columns: NAME, MODEL ID, SERVICE INSTANCE, and LAST MODIFIED. On the right side, there is a dropdown menu titled 'Add to project'. The menu items are: Connected data, Notebook, Connection, Data asset, Model, Experiment ^{BETA}, Modeler flow (highlighted with a red circle), Dashboard, Data flow, Environment, and Visual recognition model.

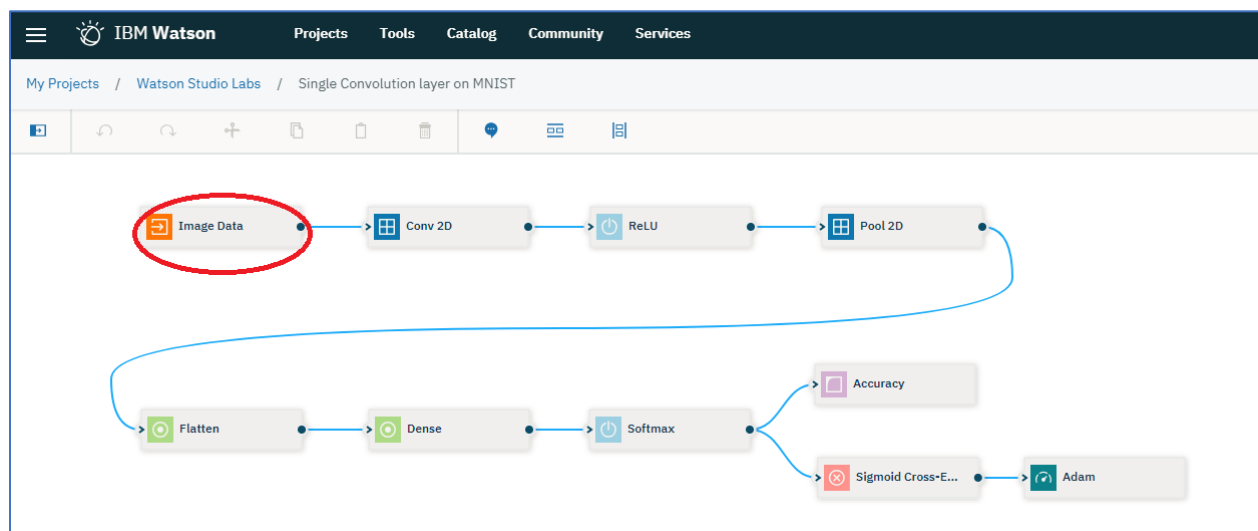
3. Click on **From example**.

The screenshot shows the IBM Watson Studio 'Modeler' page. At the top, there is a navigation bar with 'Projects', 'Tools', 'Catalog', 'Community', and 'Services'. Below this, there is a header 'Modeler'. Under the header, there are three tabs: 'New', 'From file', and 'From example' (highlighted with a red circle). Below the tabs, there is a form with the following fields: 'Name*' (with a placeholder 'Type name here.' and a character count of 50), 'Description' (with a placeholder 'Type description here.' and a character count of 500), 'Select flow type' (with two radio buttons: 'Modeler Flow' (selected) and 'Neural Network Modeler ^{BETA}'), and 'Runtime' (with two radio buttons: 'IBM SPSS Modeler' (selected) and 'Scala Spark 2.1 ^{BETA}').

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



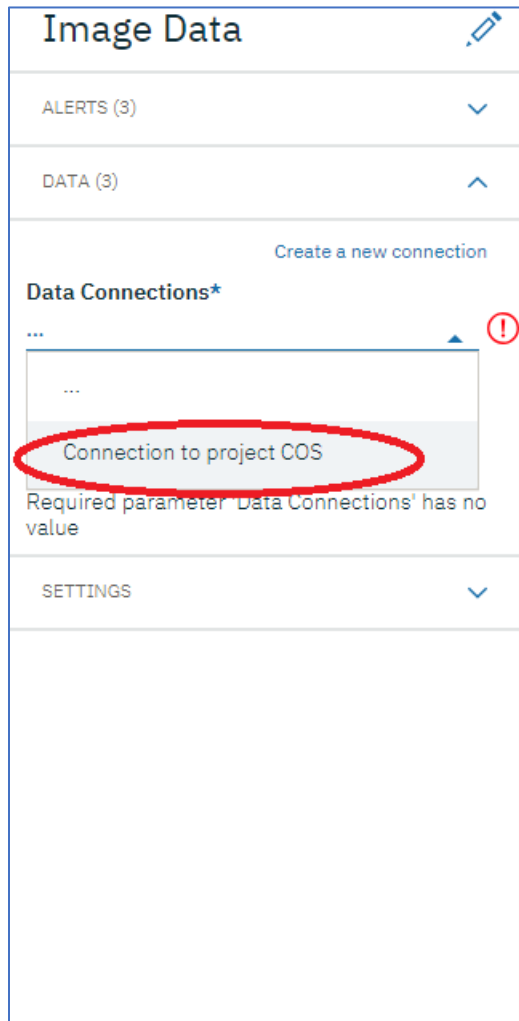
5. We need to configure the Image Data node. Double-click on Image Data.



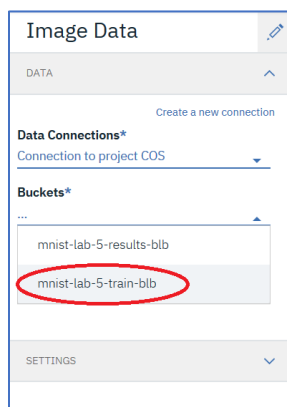
6. Optionally change the default name, and click on **Create a connection**.

The screenshot shows the 'Image Data' configuration panel. It includes sections for 'ALERTS (2)', 'DATA (2)', and 'SETTINGS'. The 'DATA (2)' section contains a message: 'There are no data connections in this project. Would you like to connect your default COS instance to your project?'. Below this, there is a 'Connection Name' field with the text 'Connection to project COS' and a 'Create a connection' button, which is highlighted with a red circle.

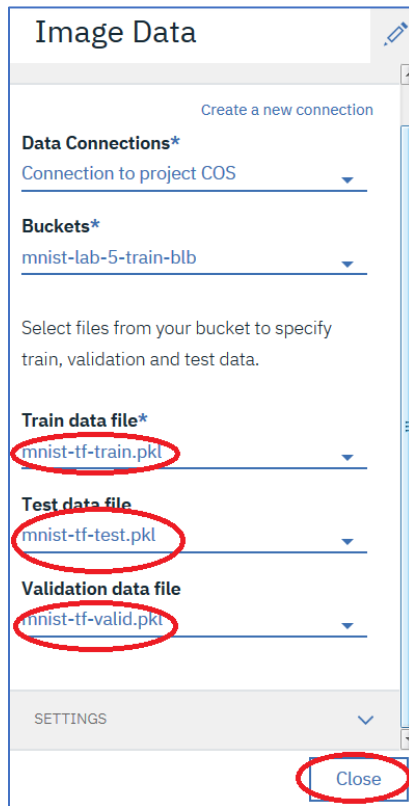
- Click on the downward triangle icon ▼ underneath **Data Connections***. Click on the connection that was just created.





- Click on the downward triangle icon ▼ underneath **Buckets***, and then click on the **mnist-lab-5-train-xxx** where “xxx” are your initials.



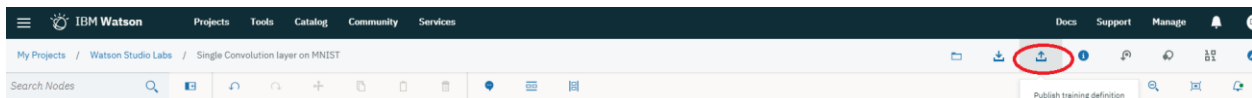
- Click on the ▼ icon underneath **Train data file***, and select the **mnist-tf-train-pkl**. Assign the Test data file(mnist-tf-test-pkl), and Validation data files(mnist-tf-valid-pkl) in the same way and then click on **Save**.



- Explore the neural network flow modeler options

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

- 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.

Publish Training Definition
Fill name and description then select WML instance to publish.

Name*
Single Convolution layer on MNIST

Description
Optional Description

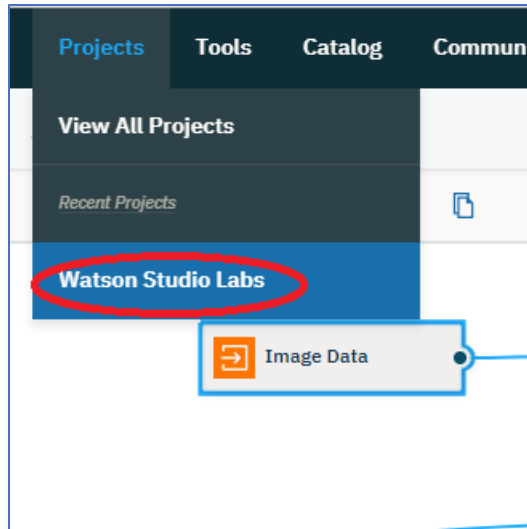
Select WML Instance*
Machine Learning

Cancel Publish

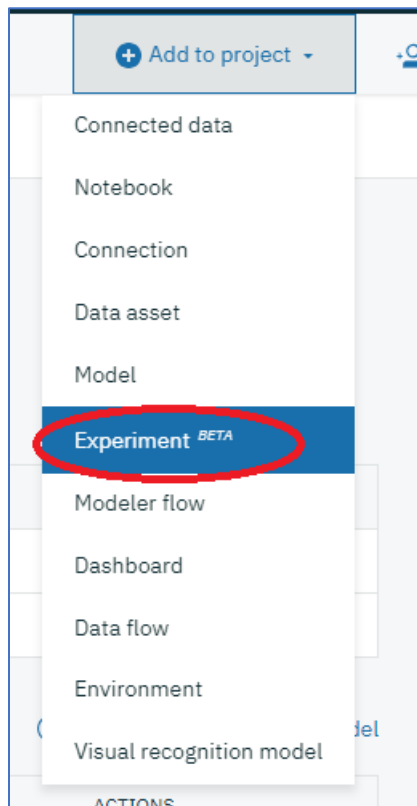
Step 3: Train the Model using Experiment Builder

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.

New experiment BETA

Define experiment details

Name

Single Convolution Layer on MNIST

67

Description

Experiment description

300

Machine Learning Service

Machine Learning

Cloud Object Storage bucket for storing training source and results files

Select

Associate training definitions

[+ Add training definition](#)

NAME	COMPUTE PLAN
No training definitions associated.	

☐ Use global execution command (override training definition values)

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

IBM Watson Projects Tools Catalog Community Services

Cloud Object Storage bucket selection

Existing connections New connection

Cloud Object Storage connection

Select Cloud Object Storage connection

Connection to project COS

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

Cloud Object Storage bucket selection

Existing connections New connection

Cloud Object Storage connection
Connection to project COS

Bucket containing training data
☒ Existing ☐ New
mnist-lab-3-train-bib

Bucket for storing training results
☒ Existing ☐ New
mnist-lab-3-results-bib

Cancel Select

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

IBM Watson Projects Tools Catalog Community Services Docs Support Manage

New experiment ^{BETA}

Define experiment details

Name
Single Convolution Layer on MNIST

Description
Experiment description

Machine Learning Service
Machine Learning

Cloud Object Storage buckets for storing training source and results files
Source: Connection to project COS / mnist-lab-3-train...
Results: Connection to project COS / mnist-lab-3-resul...
Update

Associate training definitions

Add training definition

NAME	COMPUTE PLAN
No training definitions associated.	

☐ Use global execution command (override training definition values)

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**.

Add training definition

Existing training definition

Select training definition
Single Convolution Layer on MNIST

Training definition attributes

Compute plan
1/2 x NVIDIA Tesla K80 (1 GPU)

Hyperparameter optimization method
None

Cancel Select

8. Click **Create and run**.

New experiment BETA

Define experiment details

Name

Single Convolution Layer on MNIST

67

Description

Experiment description

300

Machine Learning Service

Machine Learning

▼

Cloud Object Storage buckets for storing training source and results files

Source: Connection to project COS / mnist-lab-3-train...

Update

Results: Connection to project COS / mnist-lab-3-resul...

Update

If your connection is authorized for dashboard access, click the bucket name above to launch the dashboard. It may take a few seconds for the dashboard link to work for newly created buckets. Alternatively, reference the Cloud Object Storage API.

Associate training definitions

+ Add training definition

NAME	COMPUTE PLAN
Single Convolution layer on MNIST	1/2 x NVIDIA® Tesla® K80 (1 GPU)

☐ Use global execution command (override training definition values)

Cancel

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

My Projects / Watson Studio Labs / Single Convolution Layer on MNIST

Single Convolution Layer on MNIST

Cancel runs in progress Add training runs

Training Runs Compare Runs Overview

1

Runs in total

0 hr, 0 min, 0 sec

Total running time

Queued

NAME	SUBMITTED
Single Convolution layer on MNIST	0 hr, 0 min, 6 sec ago

In progress

NAME	DURATION
No training runs found.	

Completed

NAME	STATUS	DURATION	ACTIONS
No training runs found.			

In-Process Status

My Projects / Watson Studio Labs / Single Convolution Layer on MNIST

Single Convolution Layer on MNIST

Cancel runs in progress Add training runs

Training Runs Compare Runs Overview

1 Runs in total 0 hr, 0 min, 43 sec Total running time

Queued

NAME	SUBMITTED
No training runs found.	

In progress

NAME	DURATION
Single Convolution layer on MNIST	0 hr, 0 min, 43 sec

Completed

NAME	STATUS	DURATION	ACTIONS
No training runs found.			

Completed Status

Single Convolution Layer on MNIST

Cancel runs in progress Add training runs

Training Runs Compare Runs Overview

1 Runs in total 0 hr, 2 min, 45 sec Total running time

Queued

NAME	SUBMITTED
No training runs found.	

In progress

NAME	DURATION
No training runs found.	

Completed

NAME	STATUS	DURATION	ACTIONS
Single Convolution layer on MNIST	completed	0 hr, 2 min, 45 sec	

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

My Projects / Watson Studio Labs / Single Convolution Layer on MNIST

Single Convolution Layer on MNIST

Add training runs

Training Runs Compare Runs Overview

1 Runs in total 0 hr, 2 min, 45 sec Total running time

Queued

NAME	SUBMITTED
No training runs found.	

In progress

NAME	DURATION
No training runs found.	

Completed

NAME	STATUS	DURATION	ACTIONS
Single Convolution layer on MNIST	completed	0 hr, 2 min, 45 sec	

2. Click on **Logs**

Single Convolution layer on MNIST

Monitor Overview **Logs**

✓ **completed**
Training status

✓ **0 hr, 2 min, 45 sec**
Total running time

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

Single Convolution layer on MNIST

Monitor Overview **Logs**

```
> i 8352/10000 [=====>.....] - ETA: 0s
> i 9120/10000 [=====>...] - ETA: 0s
> i 9856/10000 [=====>.] - ETA: 0s
> i 10000/10000 [=====] - 1s 68us/step
> i /usr/local/lib/python3.5/dist-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.float64` is deprecated. In future versions, calling `issubdtype(float, float)` will raise an exception. To avoid this issue, import `np` and use `np.float64` directly.
> i from ._conv import register_converters as _register_converters
> i Using TensorFlow backend.
> i [0.0704007319975819, 0.9811]
> i Saving the model...
> i Model saved in file: /mnt/results/mnist-lab-3-results-blb/training-yGqyOKImg/model/Single-Convolution-layer-on-MNIST.h5
> i Done!
> i -----
```

Most recent 500 logs are displayed.

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**.

Completed			
NAME	STATUS	DURATION	ACTIONS
Single Convolution layer on MNIST	completed	0 hr, 2 min, 45 sec	⋮ Save model

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

Save Model

Name

Single Convolutional layer on MNIST

65

Description

Model description

300

Cancel

Save

- 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.

IBM Watson

Projects Tools Catalog Community Services

My Projects / Watson Studio Labs

Single Convolution Layer on MNIST

✓

Model successfully saved. View model details [here](#).

- Click on the newly saved model

Models

+

New model

NAME	STATUS	TYPE	RUNTIME	LAST MODIFIED	ACTIONS
Single Convolutional layer on MNIST	trained	tensorflow-1.5	python-3.5	5 Jun 2018	

- Click on **Deployments**.

IBM Watson

Projects Tools Catalog Community Services

My Projects / Watson Studio Labs / Single Convolutional layer on MNI...

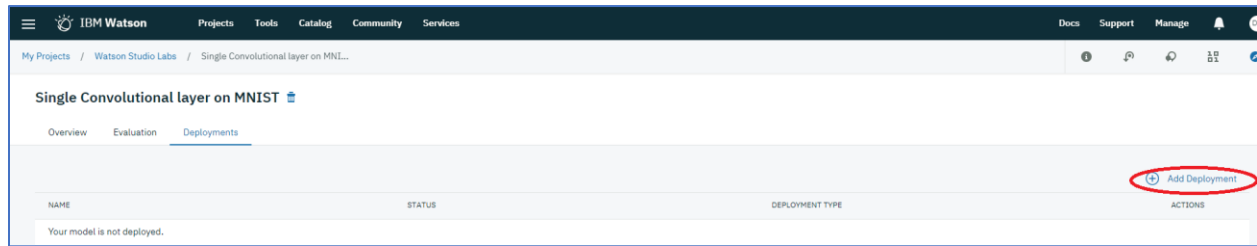
Single Convolutional layer on MNIST

Overview Evaluation Deployments

Summary

Machine learning service	Machine Learning
Model Type	tensorflow-1.5
Runtime environment	python-3.5
Training date	5 Jun 2018, 3:48 PM
Latest version	1c472928-e0ac-4146-9985-5b1c02bb8881

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**.

The screenshot shows the 'Create Deployment' form. The 'Name' field is filled with 'Single Convolutional layer on MNIST Deployed'. The 'Description' field is empty. The 'Deployment type' is set to 'Web service'. The 'Save' button is circled in red.

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

The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with 'My Projects / Watson Studio Labs / Single Convolutional layer on MNIST...'. Below this, the project name 'Single Convolutional layer on MNIST' is displayed. The 'Deployments' tab is selected. A table with columns 'NAME', 'STATUS', 'DEPLOYMENT TYPE', and 'ACTIONS' is shown. The table contains one row with the following data:

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Single Convolution layer on MNIST Deployed	DEPLOY_SUCCESS	Web Service	