

Build, Train, Deploy 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 deployed and then tested. 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. Deploy the trained model
6. Test the deployed model

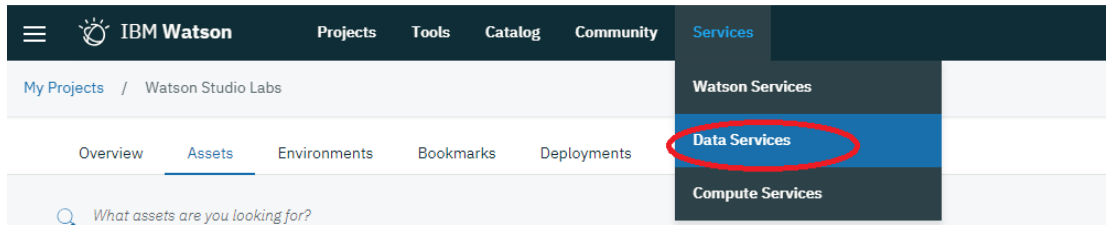
Step 1: Set up the Data Files in IBM Cloud Storage

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.)

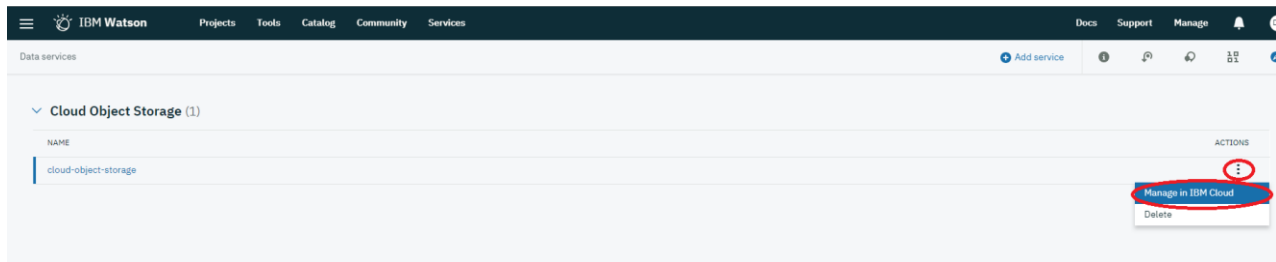
1. Download the [mnist.zip](#) file. Extract the 3 files - a training file (mnist-tf-train.pkl), test file(mnist-tf-test.pkl), and a validation file (mnist-tf-valid.pkl) in pickle format.



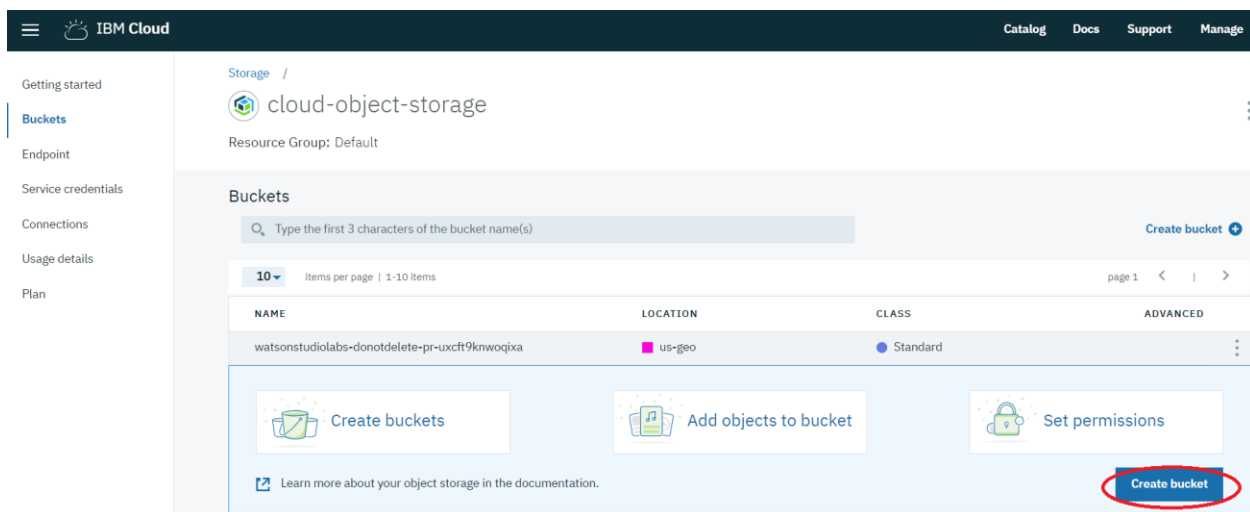
2. Select the **Services** tab and the click on **Data Services**



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



4. Click on **Create Buckets**



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

Create a bucket

Name: ⓘ

mnist-lab-3-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 **click here to add objects** to add files to the bucket.

IBM Cloud

Catalog Docs Support Manage

Storage / cloud-object-storage / mnist-lab-3-train-blb

mnist-lab-3-train-blb

Objects

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

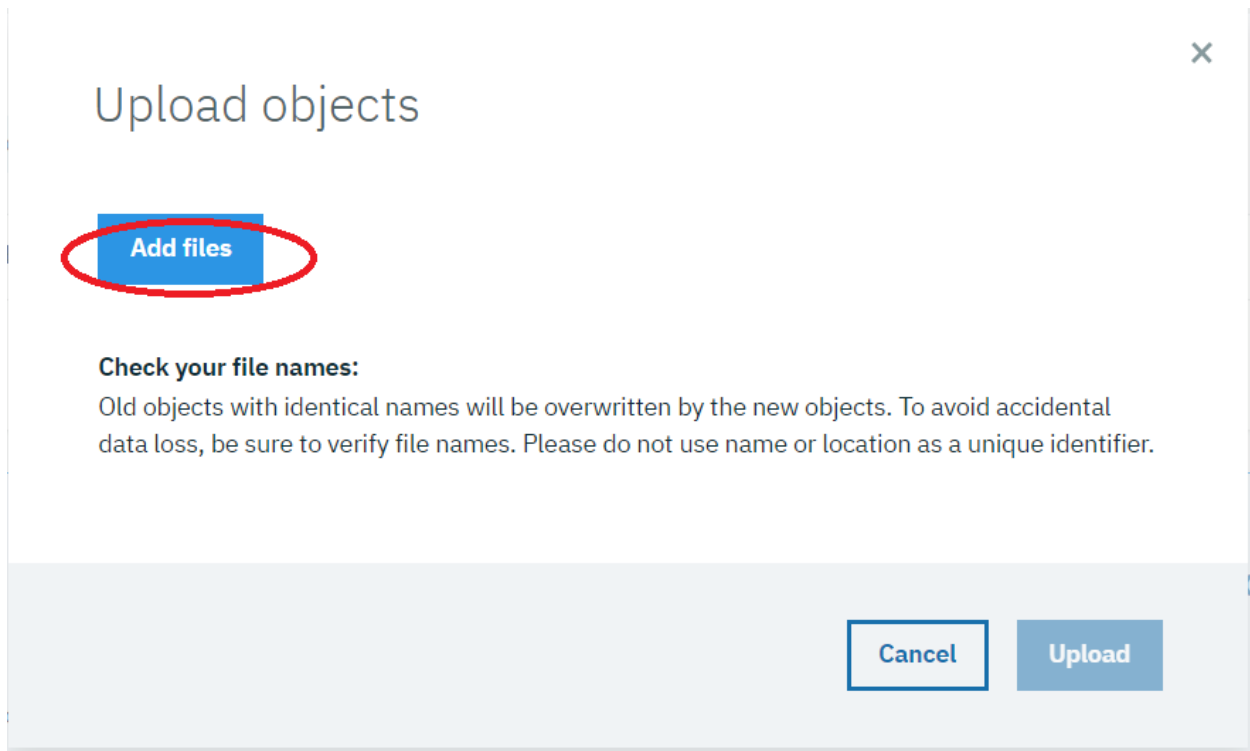
Add objects +

	Object Name	Size	Last Modified
	There are no objects in this bucket. click here to add objects.		

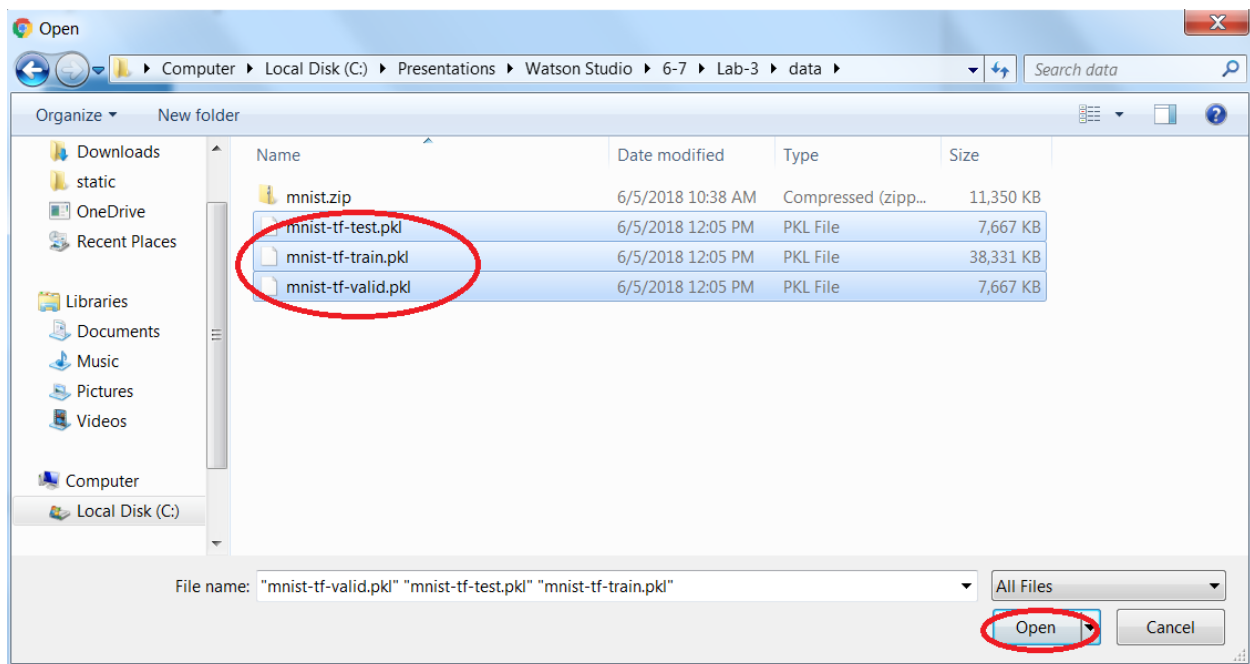
You successfully created a bucket! Now add objects or drag and drop them here.

[Learn more about buckets in Cloud Object Storage](#)

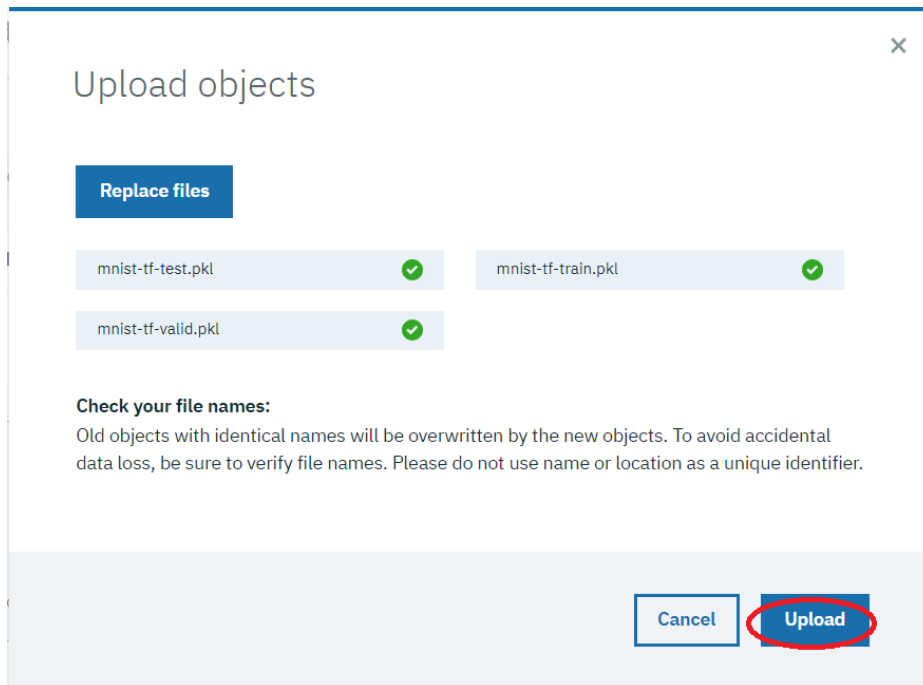
7. Click on **Add Files**



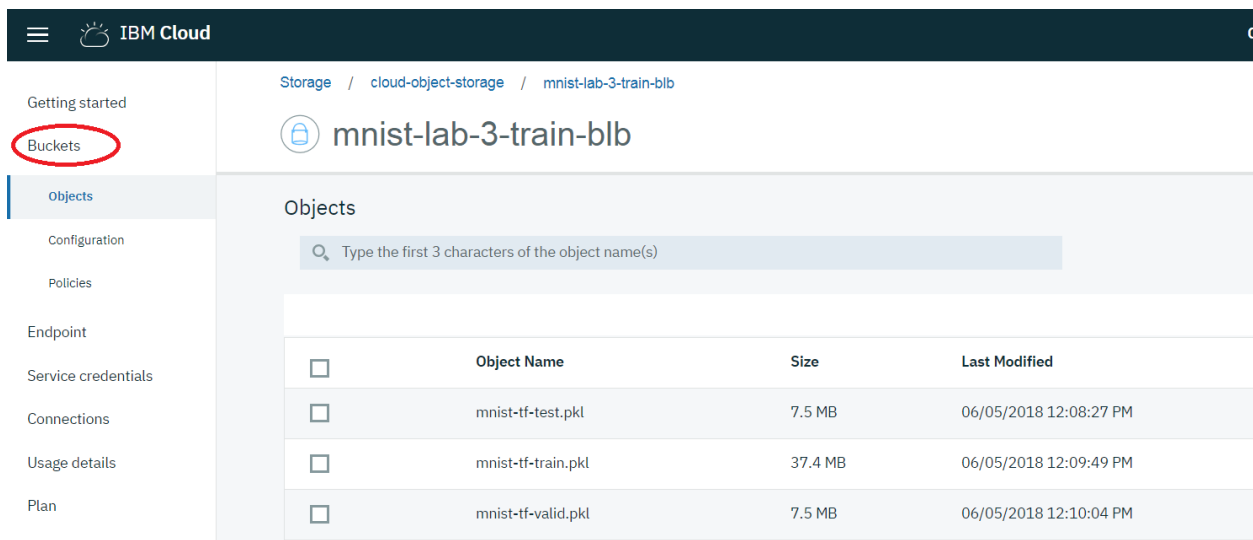
8. Navigate to where you stored the 3 pickle files. Select the files and click **Open**



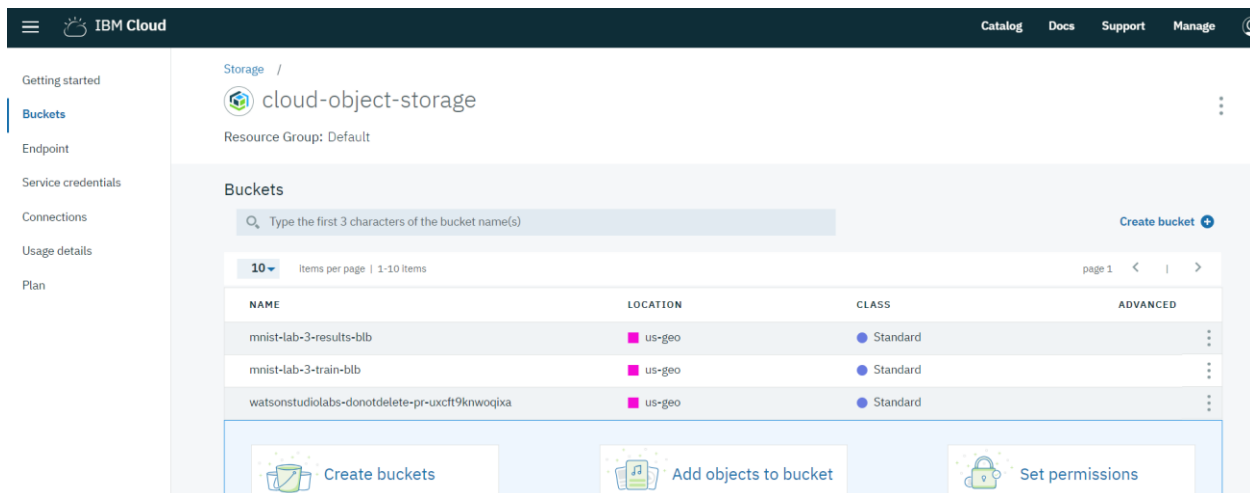
9. Click on **Upload**.



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.

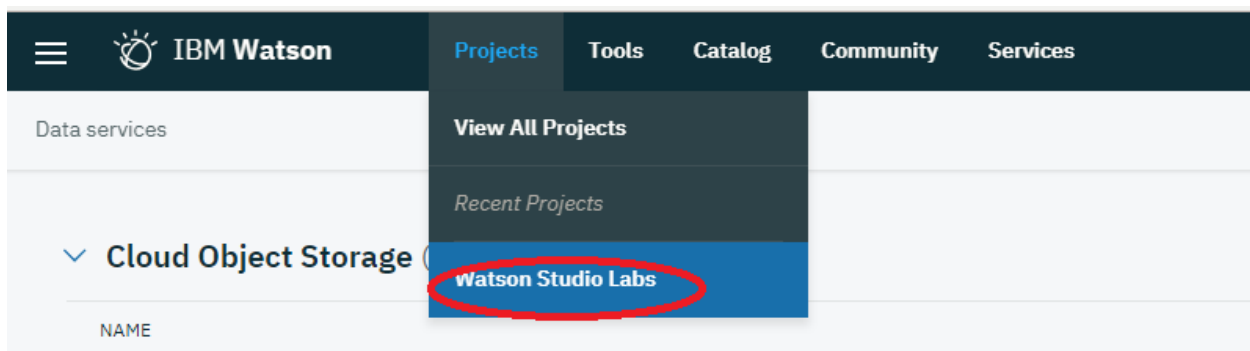


11. The Cloud Object Storage panel should appear as below.

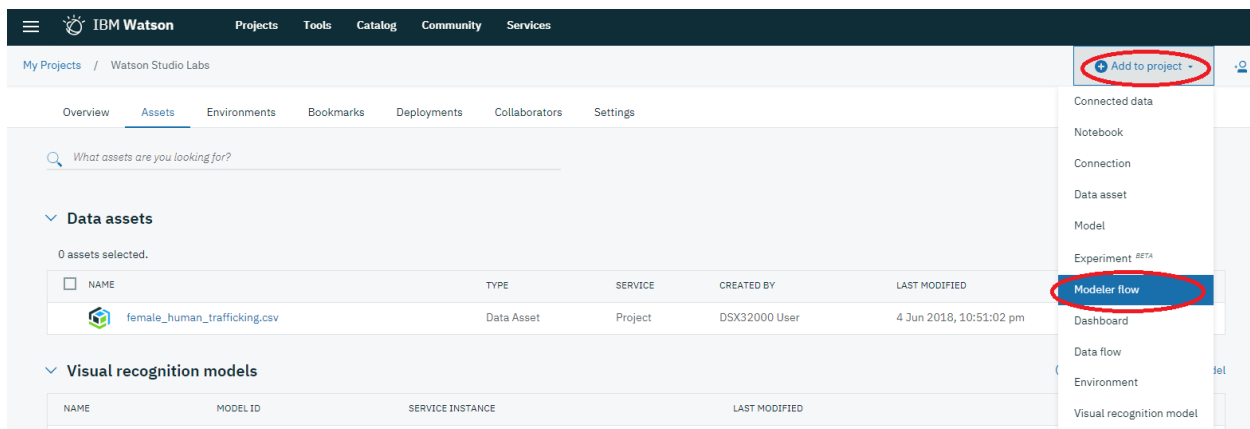


Step 2: Design the Neural Network and Publish Training Definition

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**.

IBM Watson Projects Tools Catalog Community Services

Modeler

New From file **From example**

Name*

Type name here.

50

Description

Type description here.

500

Select flow type

☒ Modeler Flow ☐ Neural Network Modeler BETA

Runtime

☒ IBM SPSS Modeler ☐ Scala Spark 2.1 BETA

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

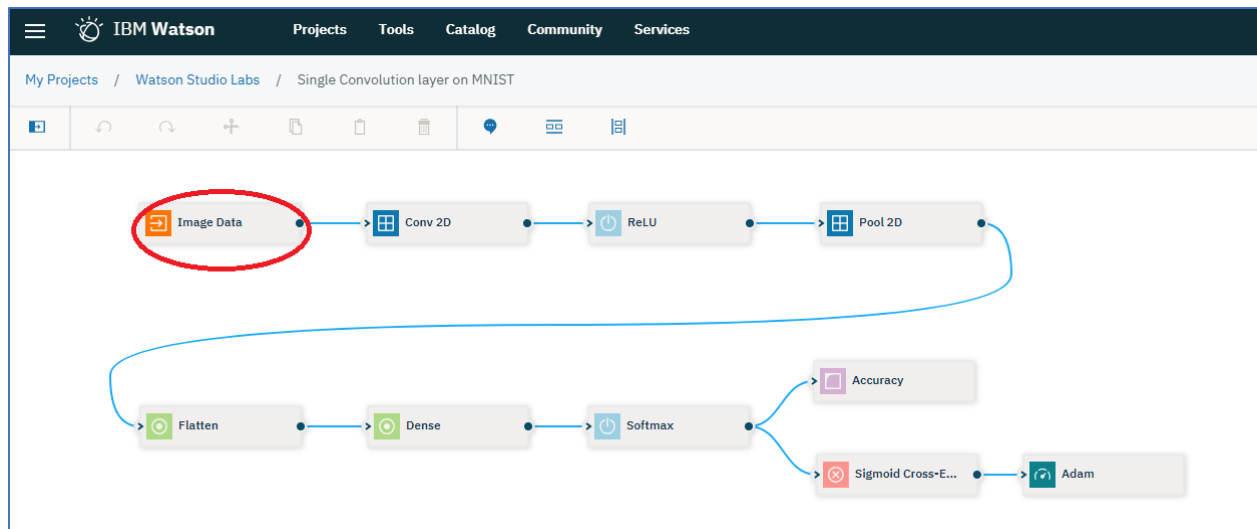
New From file From example

Select one of the samples below to get started with an existing stream that suits the kind of modeling you want to do. When you create the stream it will be added to your project, allowing you to modify and save your changes.

SPSS MODELER Drug Study Example Use neural networks and C2.0 algorithms to build classification models that allow you to predict the correct type of drug for a patient based on various health metrics.	SPSS MODELER Sales Promotion Study Use neural networks and C2.0 algorithms to predict the effect of advertising promotions on the sale of various items. Cross data of sales before and after a past promotion are used to train the model to predict the effect of sales of advertising.	NEURAL NETWORK DESIGN Single Convolution layer on MNIST Classification of MNIST dataset with a neural network containing only single convolutional and dense layer.	NEURAL NETWORK DESIGN CNN on CIFAR10 Classification of CIFAR-10 dataset with simple deep Convolutional Neural Network.
--	---	---	--

Cancel **Create**

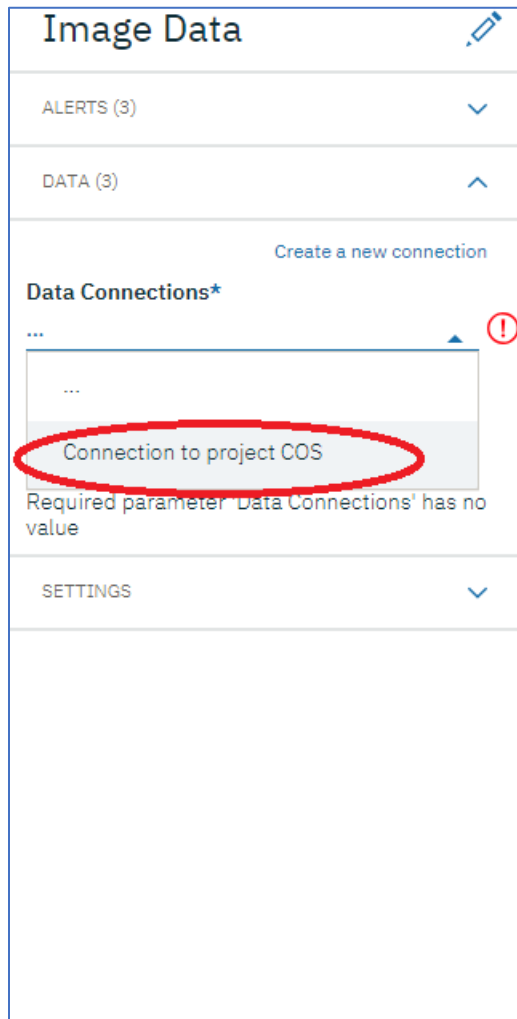
5. We need to configure the Image Data node. Double-click on Image Data.



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

The screenshot shows the configuration panel for the "Image Data" component. 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 message, the "Connection Name" is set to "Connection to project COS". The "Create a connection" button is circled in red.

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.

Image Data

ALERTS (2)

DATA (2)

Create a new connection

Data Connections*

Connection to project COS

Buckets*

...

mnist-lab-3-results-blb

mnist-lab-3-train-blb

watsonstudiolabs-donotdelete-pr-uxcft9knwoqixa

Required parameter 'Buckets' has no value

SETTINGS

- 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**.

Image Data

DATA

Create a new connection

Data Connections*

Connection to project COS

Buckets*

mnist-lab-3-train-blb

Select files from your bucket to specify train, validation and test data.

Train data file*

mnist-tf-train.pkl

Test data file

mnist-tf-test.pkl

Validation data file

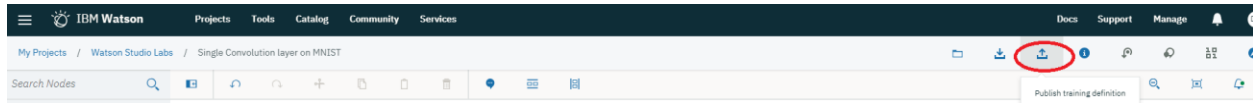
mnist-tf-valid.pkl

SETTINGS

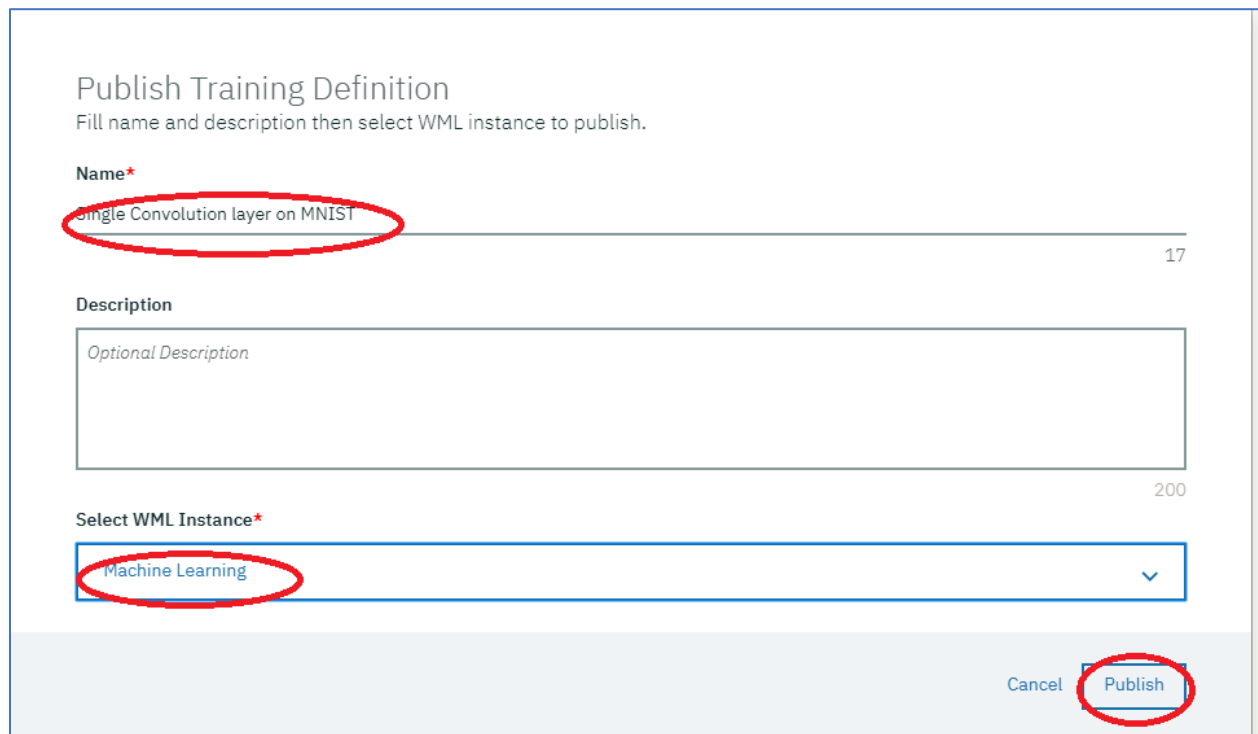
Cancel

Save

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



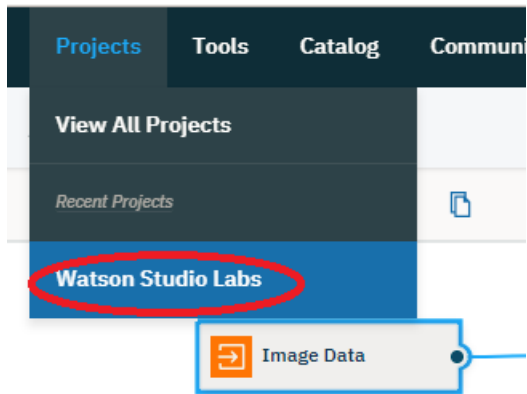
11. Enter a name for the training definition (or leave the default), and select the Machine Learning service that you created. *Note, it will not be named Machine Learning unless that is the name that you used.* Click on **Publish**.

A screenshot of the 'Publish Training Definition' form. The form has a title 'Publish Training Definition' and a subtitle 'Fill name and description then select WML instance to publish.' It contains three main sections: 'Name*' with a text input field containing 'Single Convolution layer on MNIST' (circled in red); 'Description' with a text area containing 'Optional Description'; and 'Select WML Instance*' with a dropdown menu showing 'Machine Learning' (circled in red). At the bottom right, there are 'Cancel' and 'Publish' buttons, with the 'Publish' button circled in red.

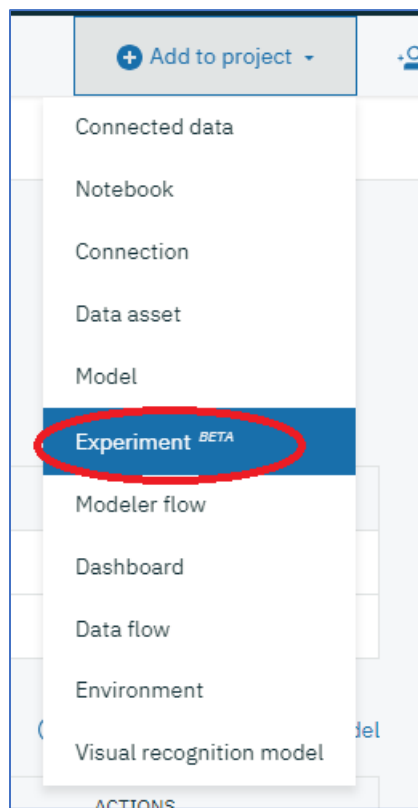
Step 3: Train the Model using Experiment Builder

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-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**.

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

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 APIs

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

New training definition Existing training definition

Select training definition

Existing training definitions
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...
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

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, 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, it will be treated as `np.float64` or `np.float_` depending on the context.
> 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: Deploy the Trained Model

We will now save the trained model to the Watson Machine Learning repository. We will then use the Watson Machine Learning UI to deploy the model. Once deployed, the model can be used to classify new images.

Note failing on the deployment.

