

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

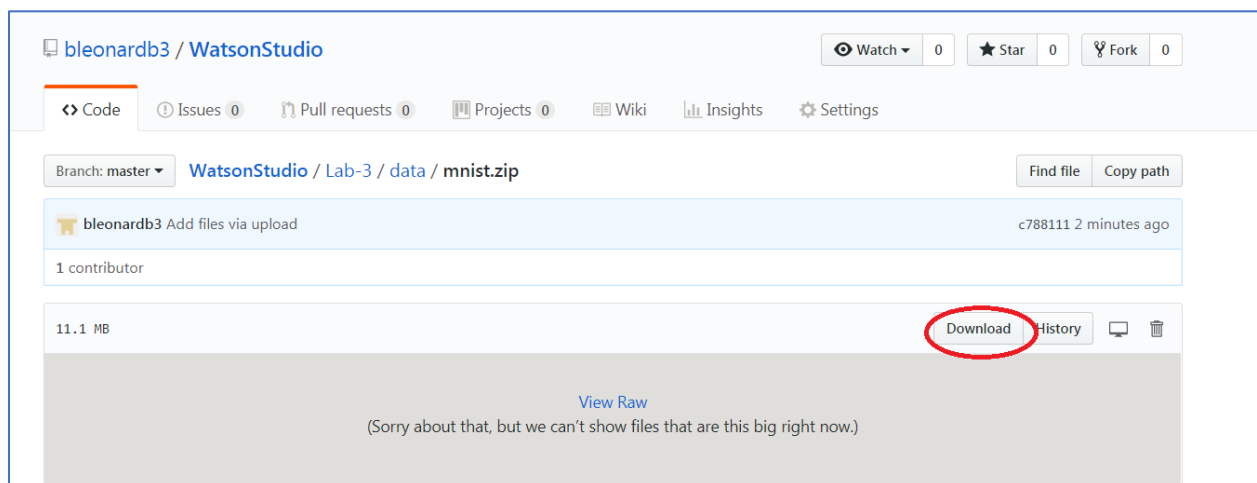
This lab will use the [MNIST](#) computer vision data set to train a deep learning model to recognize handwritten digits. A single layer convolutional neural network will be built in the Watson Studio neural network designer, and then trained using the Watson Studio Experiment Builder. The trained model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

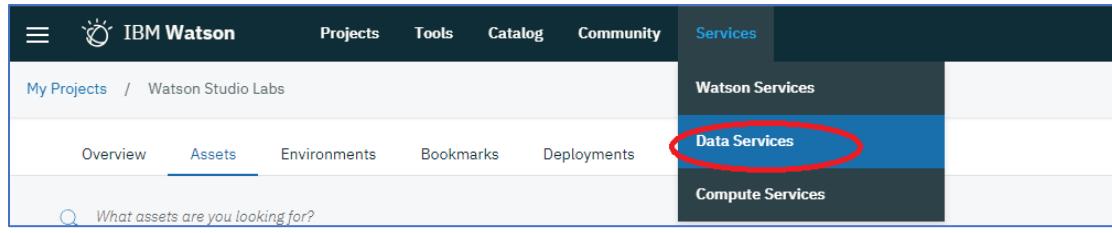
Step 1: Set up the Data Files in IBM Cloud Storage

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.



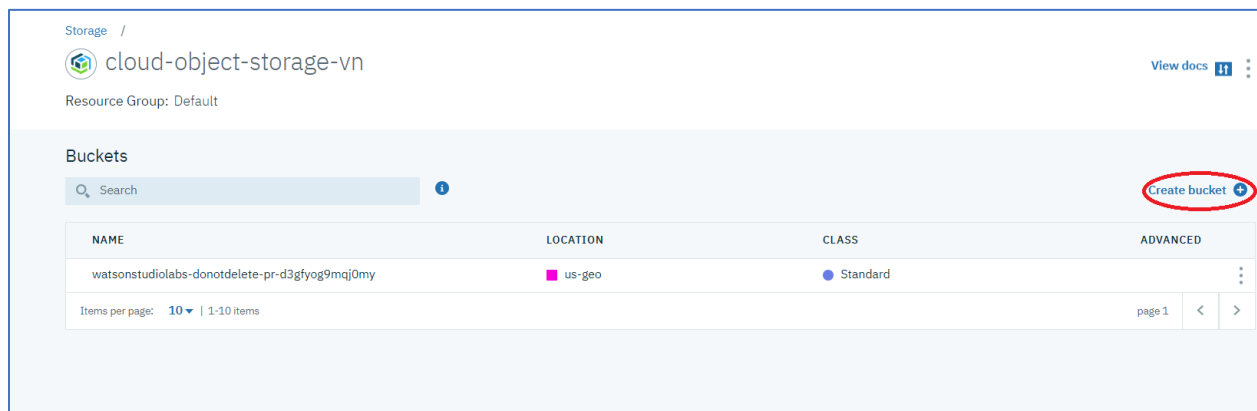
2. Select the **Services** tab and the click on **Data Services**



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), 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

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

Objects

Search ⓘ

☐ Object Name Size Last Modified

Drag and drop files or folders to upload them.

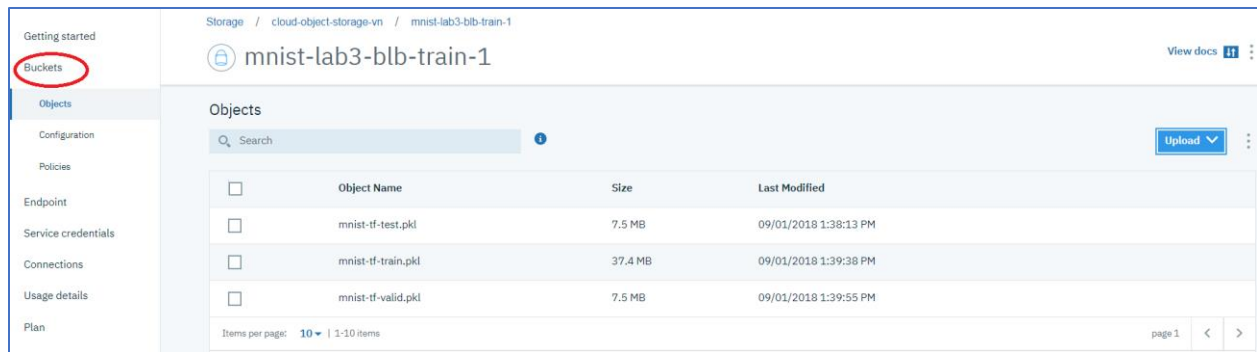
Computer
 Local Disk (C:)
 Presentations
 Machine Learning
 9-6
 Lab-3
 data

Copy to data

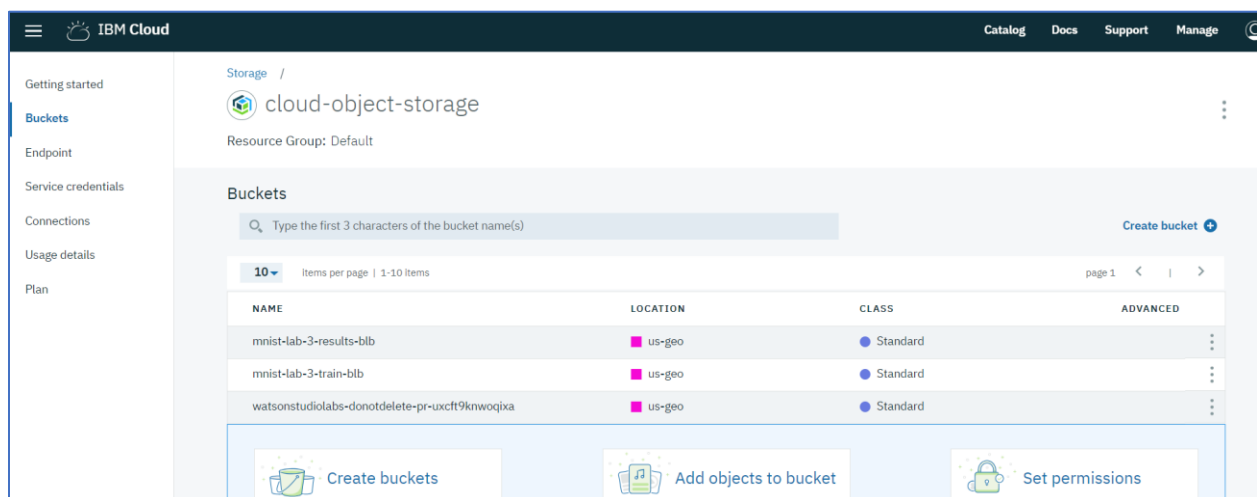
Search data

	Name	Date modified	Type	Size
	mnist.zip	6/5/2018 10:38 AM	Compressed (zipp...	11,350 KB
	mnist-tf-test.pkl	9/1/2018 11:59 AM	PKL File	7,667 KB
	mnist-tf-train.pkl	9/1/2018 11:59 AM	PKL File	38,331 KB
	mnist-tf-valid.pkl	9/1/2018 11:59 AM	PKL File	7,667 KB

- Click on Buckets to add a second bucket.

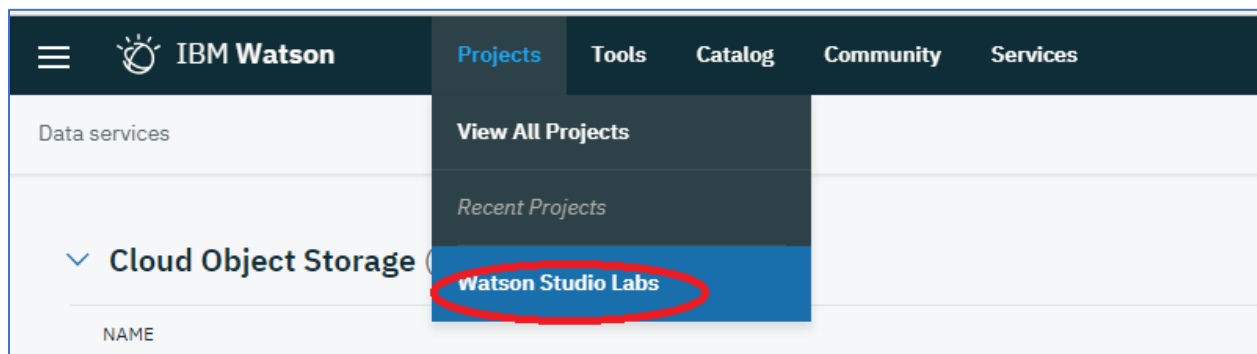


8. Name it mnist-lab-3-results-xxx, where xxx are your initials. Follow the procedure above to create the second bucket. No files need to be added.
9. The Cloud Object Storage panel should appear as below.

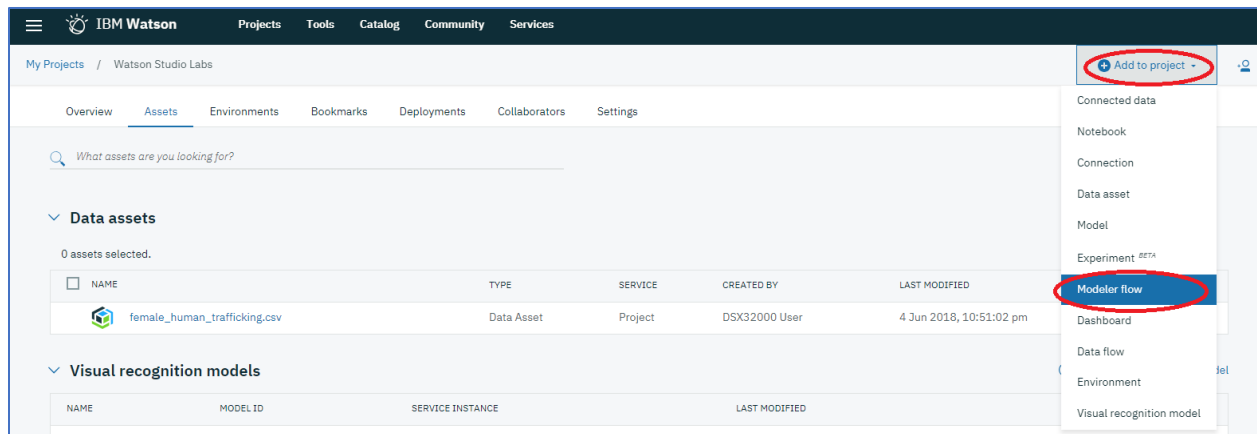


Step 2: Design the Neural Network and Publish Training Definition

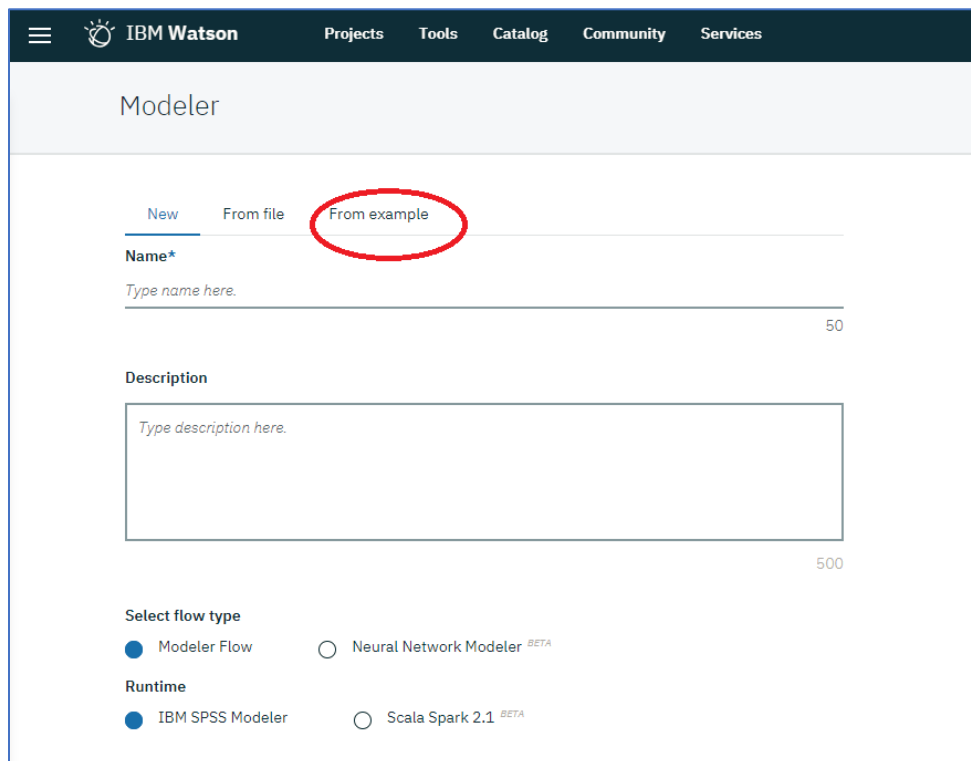
1. Return to Watson Studio, and click on the **Projects** tab, and **Watson Studio Labs**.



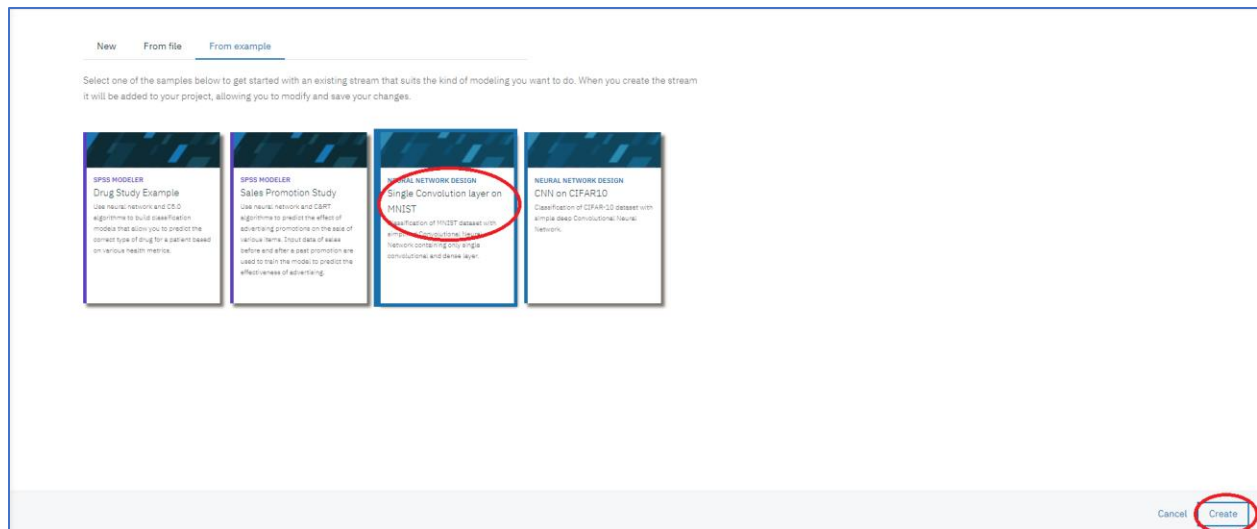
2. Click on the **Add to project** and then click on **Modeler Flow**.



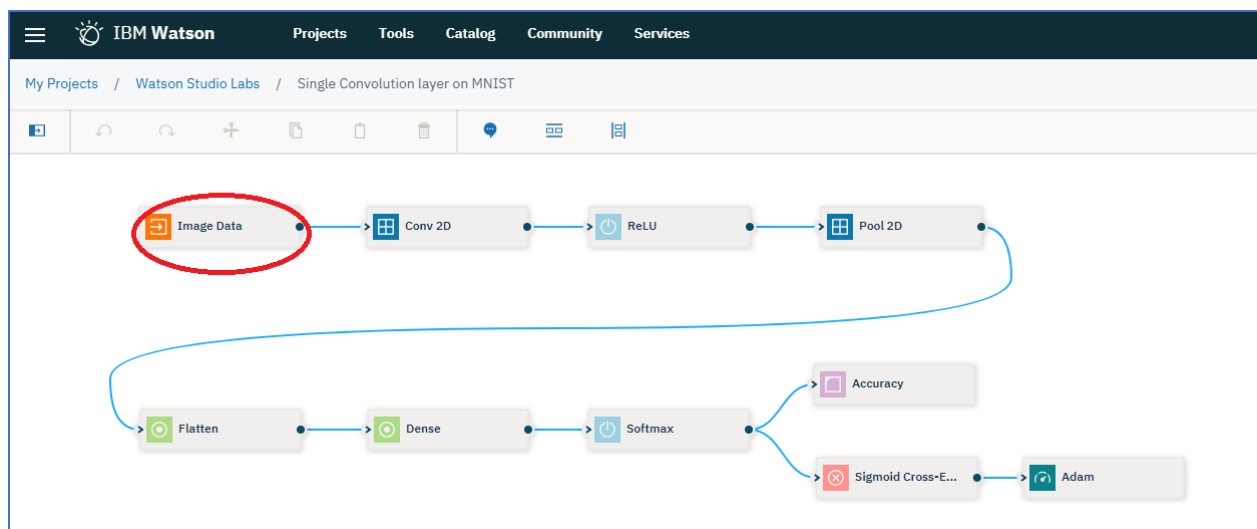
3. Click on **From example**.



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**.

Image Data

ALERTS (2) ▼

DATA (2) ▲

There are no data connections in this project.
Would you like to connect your default COS instance to your project?

Connection Name

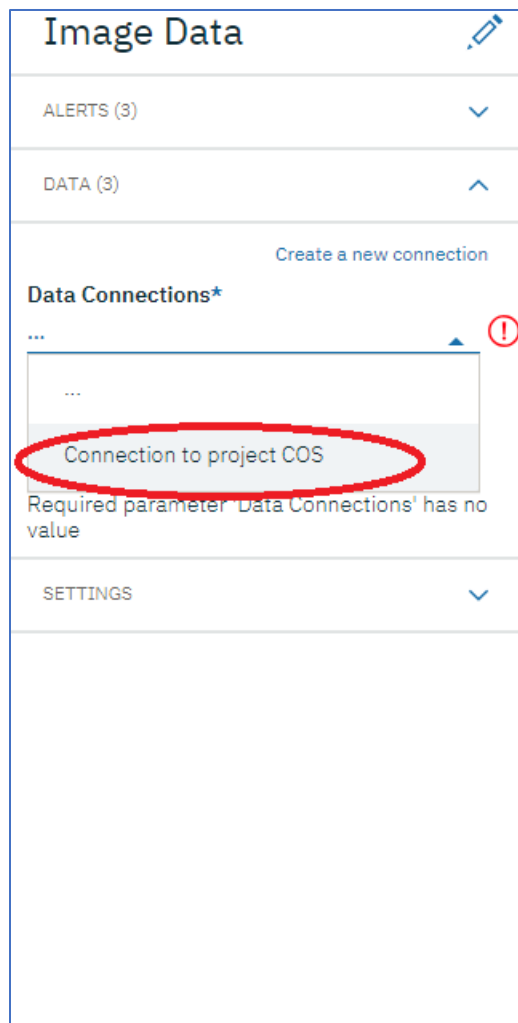
Connection to project COS

25

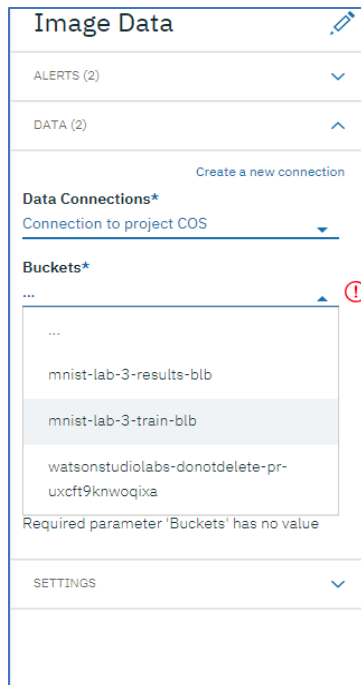
Create a connection

SETTINGS ▼

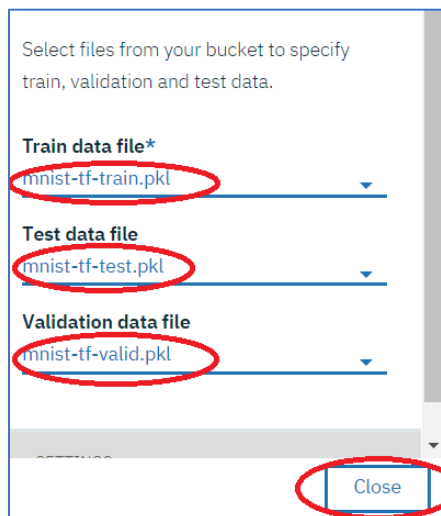
- Click on the downward triangle icon ▼ underneath **Data Connections***. Click on the connection that was just created.





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

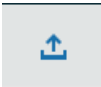


9. Click on the ▼ icon underneath **Train data file***, and select the **mnist-tf-train.pkl**. Assign the Test data file(mnist-tf-test.pkl), and Validation data files(mnist-tf-valid.pkl) in the same way and then click on **Save**.



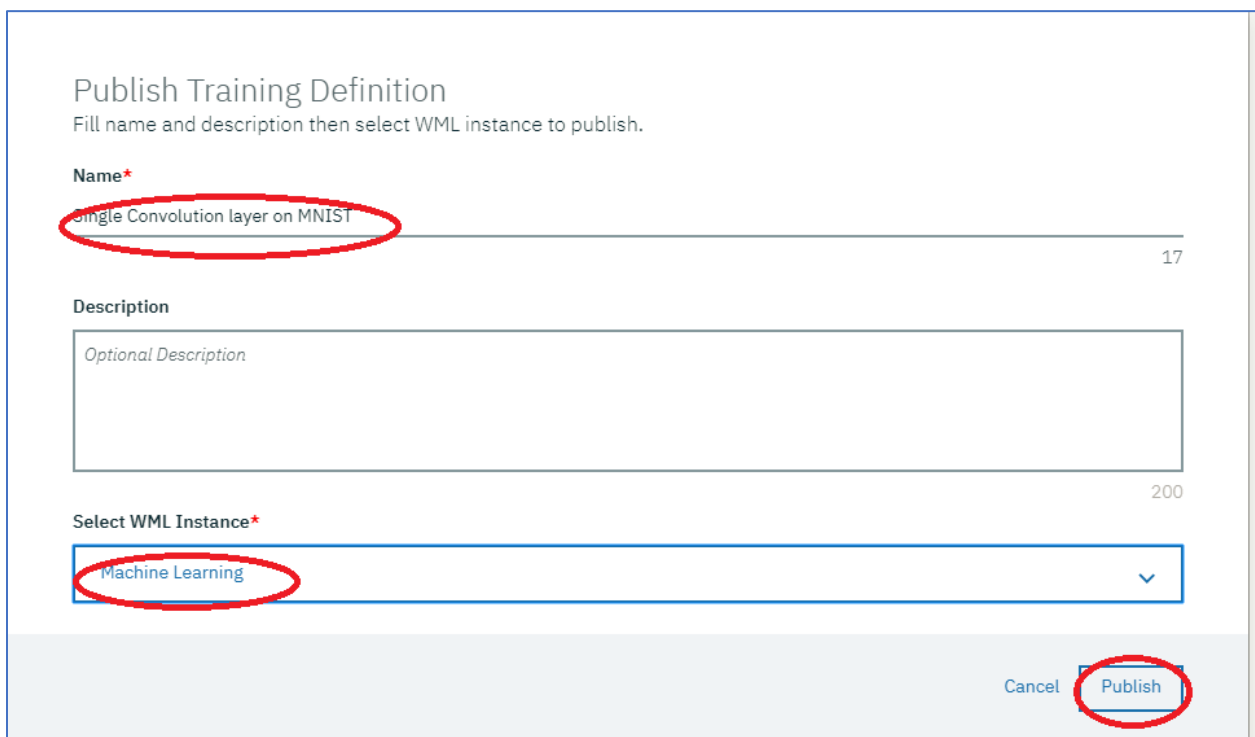
10. Explore the neural network flow modeler options
 1. Click on the  icon to see the list of neural network component categories that are available
 2. Explore the contents in each category. Hover over the components to get a pop-up description.
 3. Drag some nodes on the canvas and double-click to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



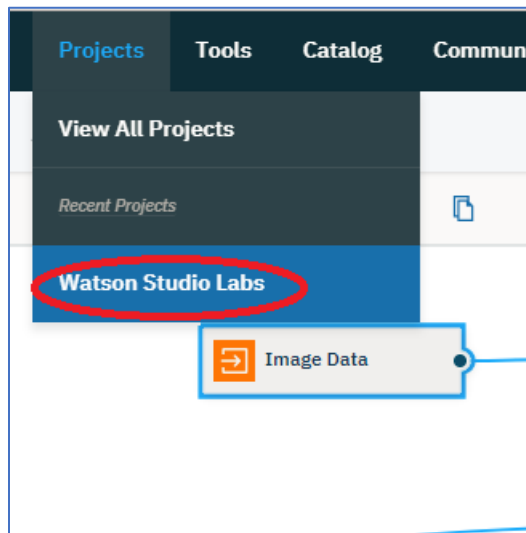
12. Enter a name for the training definition (or leave the default), and select the Machine Learning service that you created. Note, it will not be named Machine Learning unless that is the name that you used. Click on Publish.

A screenshot of the 'Publish Training Definition' form in IBM Watson Studio. The form has three main sections: 'Name*', 'Description', and 'Select WML Instance*'. The 'Name' field contains 'Single Convolution layer on MNIST' and is circled in red. The 'Description' field is empty and has a placeholder 'Optional Description'. The 'Select WML Instance' dropdown menu is open, showing 'Machine Learning' as the selected option, which is also circled in red. At the bottom right, there are 'Cancel' and 'Publish' buttons, with the 'Publish' button being highlighted with a red circle.

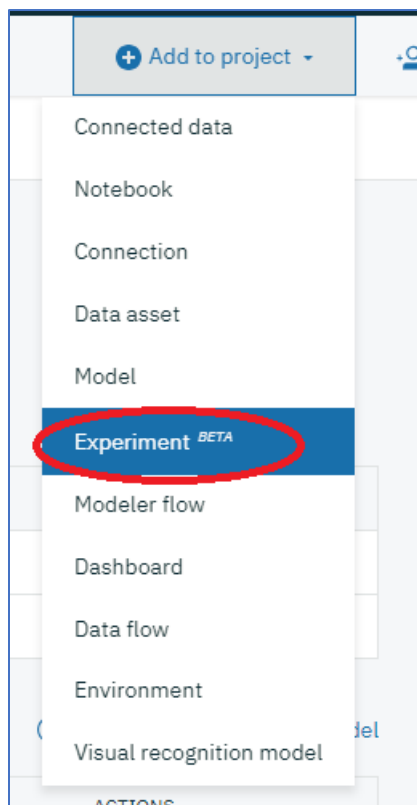
Step 3: Train the Model using Experiment Builder

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.

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...

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)	<div></div>

☐ 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**

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

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.

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

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
> [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	

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
<u>Single Convolutional Layer on MNIST</u>	trained	tensorflow-1.5	python-3.5	5 Jun 2018	

- Click on **Deployments**.

The screenshot shows the IBM Watson Studio interface. The top navigation bar includes 'My Projects', 'Watson Studio Labs', and 'Single Convolutional layer on MNI...'. The main header displays the project name 'Single Convolutional layer on MNIST' with a small icon. Below the header, there are three tabs: 'Overview', 'Evaluation', and 'Deployments', with 'Deployments' being the active tab. A 'Summary' section follows, containing a table with the following details:

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**.

This screenshot shows the 'Deployments' tab of the project. At the top, there are tabs for 'Overview', 'Evaluation', and 'Deployments'. Below these, a table lists deployments with columns for 'NAME', 'STATUS', 'DEPLOYMENT TYPE', and 'ACTIONS'. The table is currently empty, with a message 'Your model is not deployed.' at the bottom. In the top right corner of the table area, there is a button labeled 'Add Deployment' with a plus icon, which is circled in red.

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 displays the 'Create Deployment' form. It is divided into two main sections. The left section, titled 'Define deployment details', contains the following fields:

- Name:** A text input field containing 'Single Convolutional layer on MNIST Deployed', which is circled in red.
- Description:** A text area with a placeholder 'Deployment description' and a character count of 300.
- Deployment type:** A section with three radio button options: 'Web service' (selected and circled in red), 'Batch prediction', and 'Virtual (containers and files)'.

The right section is empty. At the bottom right of the form, there are 'Cancel' and 'Save' buttons, with the 'Save' button circled in red.

8. The model is successfully deployed.

Overview Evaluation Deployments Lineage			
			+ Add Deployment
NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Single Convolution Layer on MNIST Deployed	DEPLOY_SUCCESS	Web Service	⋮

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**.

Single Convolution on MNIST 📄			
Overview Evaluation Deployments Lineage			
			+ Add Deployment
NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Single Convolution Layer on MNIST Deployed	DEPLOY_SUCCESS	Web Service	⋮ View Delete

2. Click on **Test**.

Single Convolution Layer on MNIST Deployed	
Overview Implementation Test	
Deployment	
Name	Single Convolution Layer on MNIST Deployed
Type	Web Service
Deployment ID	89cf10e5-bd95-4b56-a728-e8b65fa98d83

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

Computer > Local Disk (C:) > Presentations > Machine Learning > 9-6 > Lab-3 > data				
Organize	Include in library	Share with	Burn	New folder
Favorites				
Desktop				
Downloads				
static				
OneDrive				
Recent Places				
Libraries				
Documents				
Name	Date modified	Type	Size	
._MACOSX	9/2/2018 10:40 AM	File folder		
mnist.zip	9/2/2018 11:46 AM	Compressed (zipp...	11,437 KB	
mnist-tf-test.pkl	9/2/2018 11:45 AM	PKL File	7,667 KB	
mnist-tf-train.pkl	9/2/2018 11:45 AM	PKL File	38,331 KB	
mnist-tf-valid.pkl	9/2/2018 11:45 AM	PKL File	7,667 KB	
mnistup.py	9/2/2018 10:50 AM	PY File	1 KB	
test.json	9/2/2018 11:36 AM	JSON File	14 KB	



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.

[illegible]

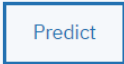
- Copy and paste the content into the **Paste the request payload here** input data section. Make sure you have both the top bracket { at the beginning of the input data section and the bottom bracket } at the end of the input section data section, and then click on **Predict**.

Single Convolution Layer on MNIST Deployed

Overview Implementation **Test**



Enter input data  

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

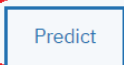


Single Convolution Layer on MNIST Deployed

Overview Implementation **Test**

Enter input data  

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



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],
      [ 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
    ]
  ]
}
```