



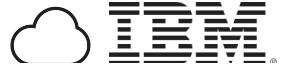
IBM Cloud

A Hands-on Introduction to Deep Learning and Neural Network Modeling with IBM Watson Studio

Training and Deploying a deep learning model for digit handwriting
recognition

Lab Guide





Notices and Disclaimers

© Copyright IBM Corporation 2018.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Other company, product and service names may be trademarks or service marks of others



Table of Contents

Lab Environment Overview.....	4
Module 1: Create an IBM Cloud Account and Watson Studio Project- Lab Instructions	5
Module 2: Create a Model Flow to Design a Neural Network for MNIST Data.....	12
Module 3: Building and Deploying a TensxorFlow Model	25
Conclusion.....	34

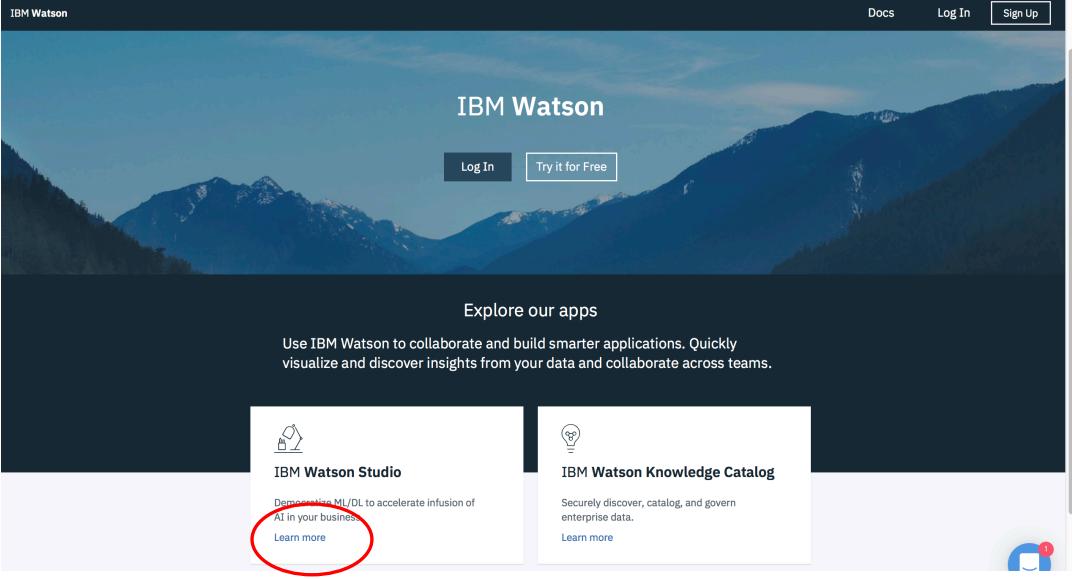


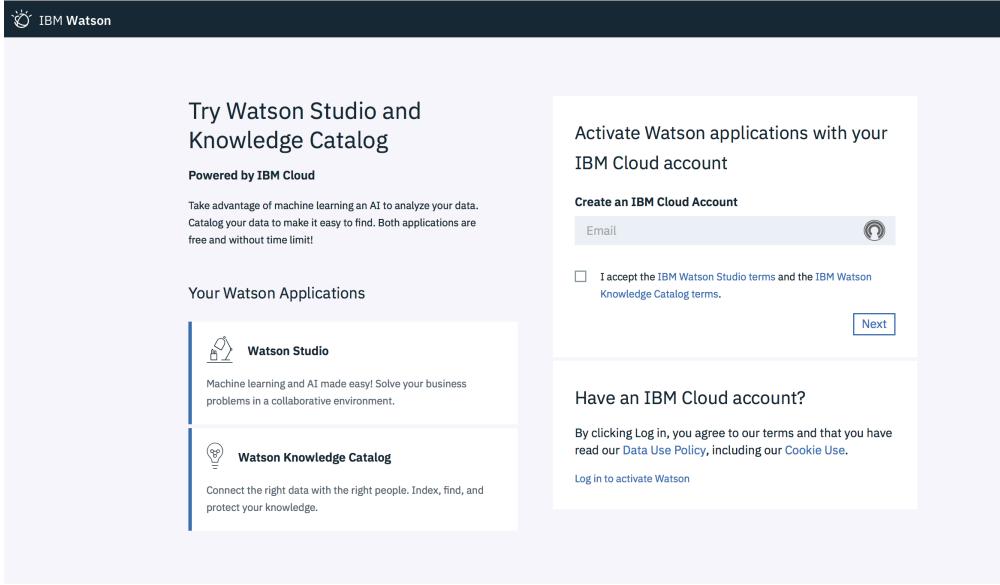
Lab Environment Overview

Purpose:	<p>The purpose of this lab is to allow you to explore how you can develop and deploy deep learning models in Watson Studio. Specifically, this includes,</p> <ul style="list-style-type: none">• Working with a visual representation of a Neural Network example in the Model Flow editor• Training and optimizing your model using the Experiment Builder• Deploying the model to IBM Watson Machine Learning where it can be utilized by developers for use in applications <p>For this lab, we will employ a standard MNIST dataset which is a sample of handwritten digits that will be used for the training and analysis of our model.</p> <p>By the end of this lab, you will be familiar with the following:</p> <ul style="list-style-type: none">• Navigating IBM Watson Studio• Creating and deploying Deep Learning Neural Network models in Watson Studio• Familiarity with IBM Cloud Services
----------	--

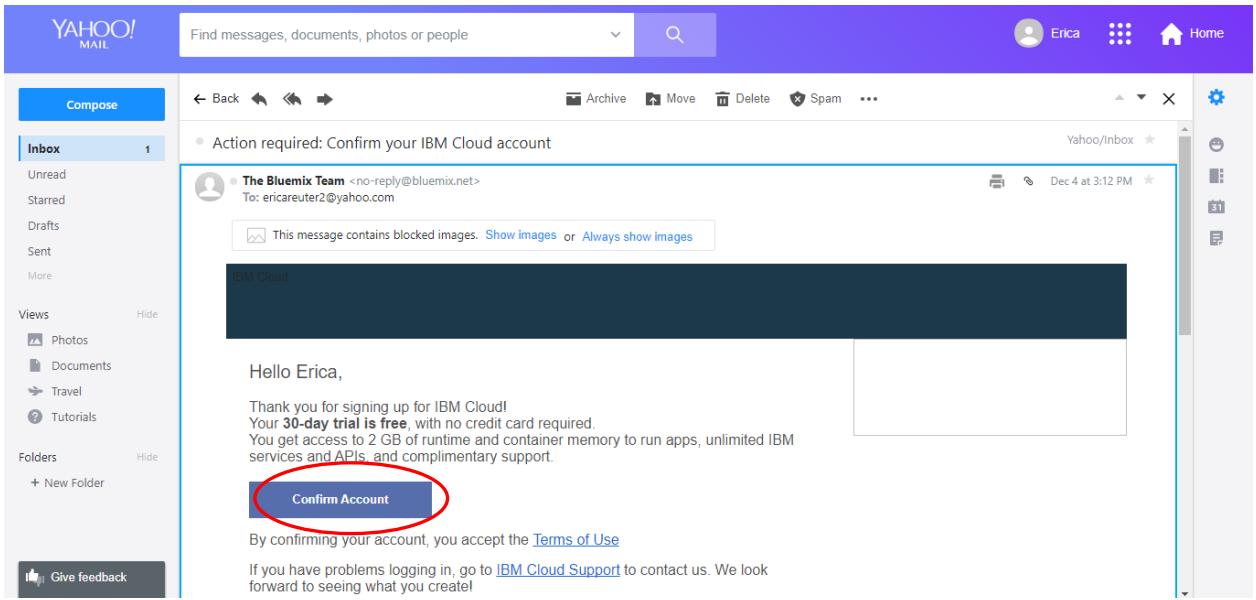
Tasks:	<p>Tasks you will complete in this lab exercise include:</p> <ul style="list-style-type: none">• Set up your IBM Account with Watson Studio• Creating a model for the MNIST data and running an experiment• Refinement and deployment of an existing TensorFlow model, using hyperparameter optimizations
--------	---

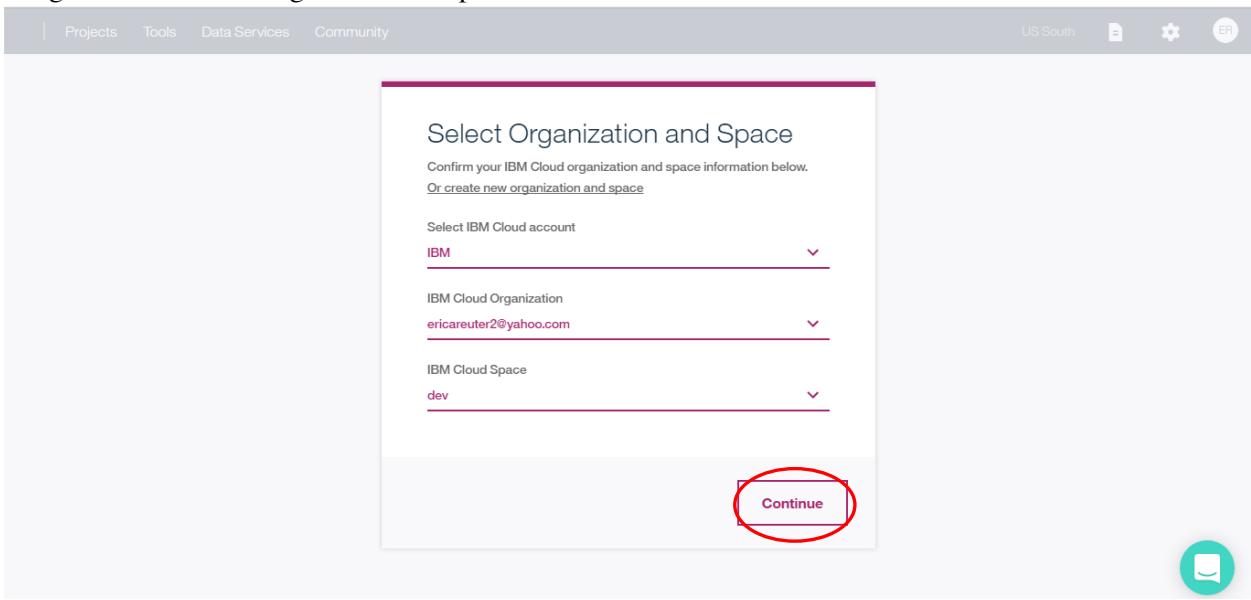
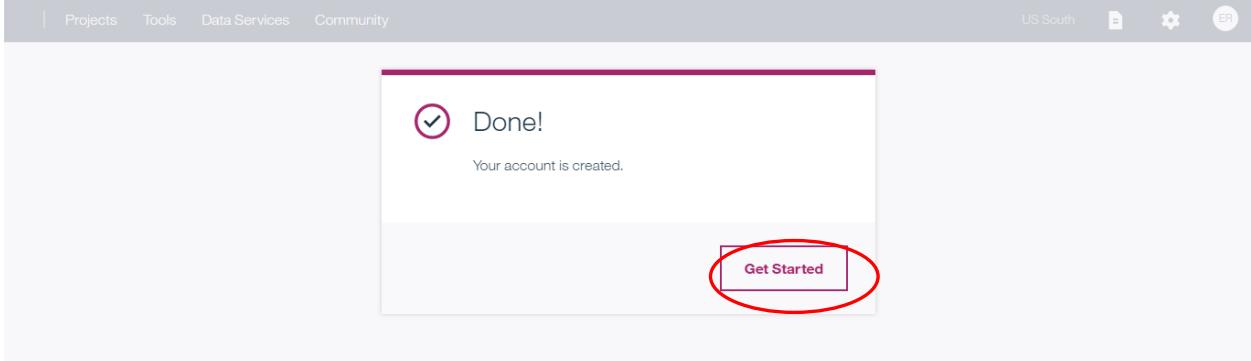
Module 1: Create an IBM Cloud Account and Watson Studio Project-Lab Instructions

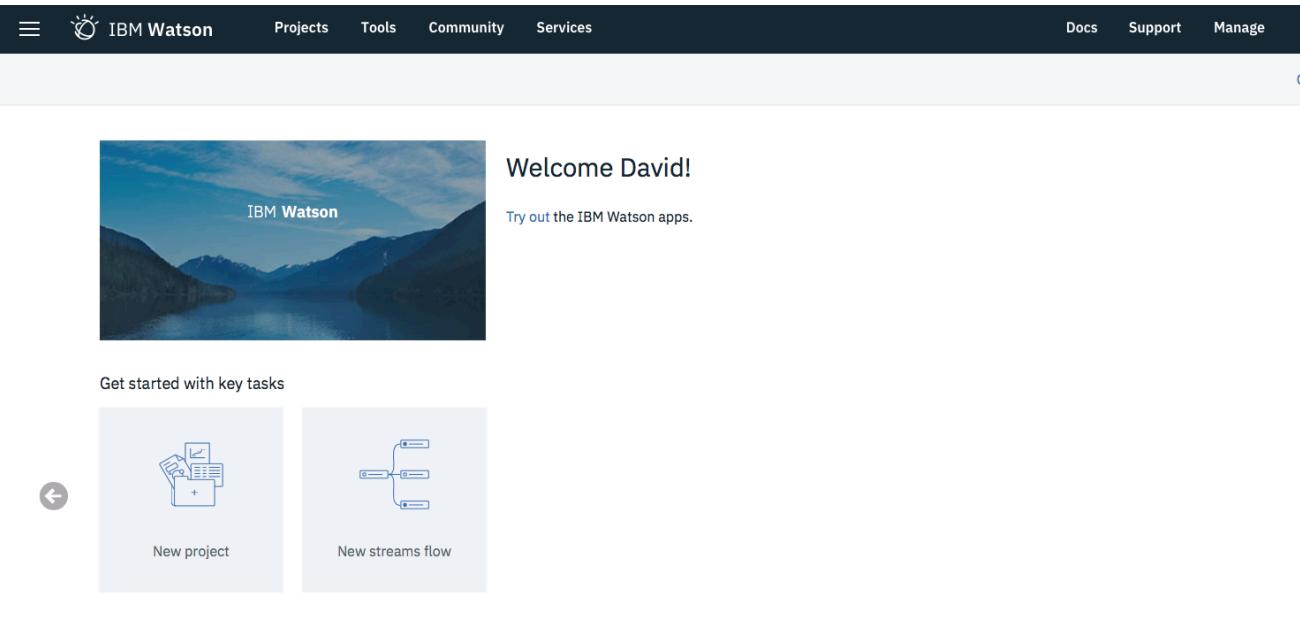
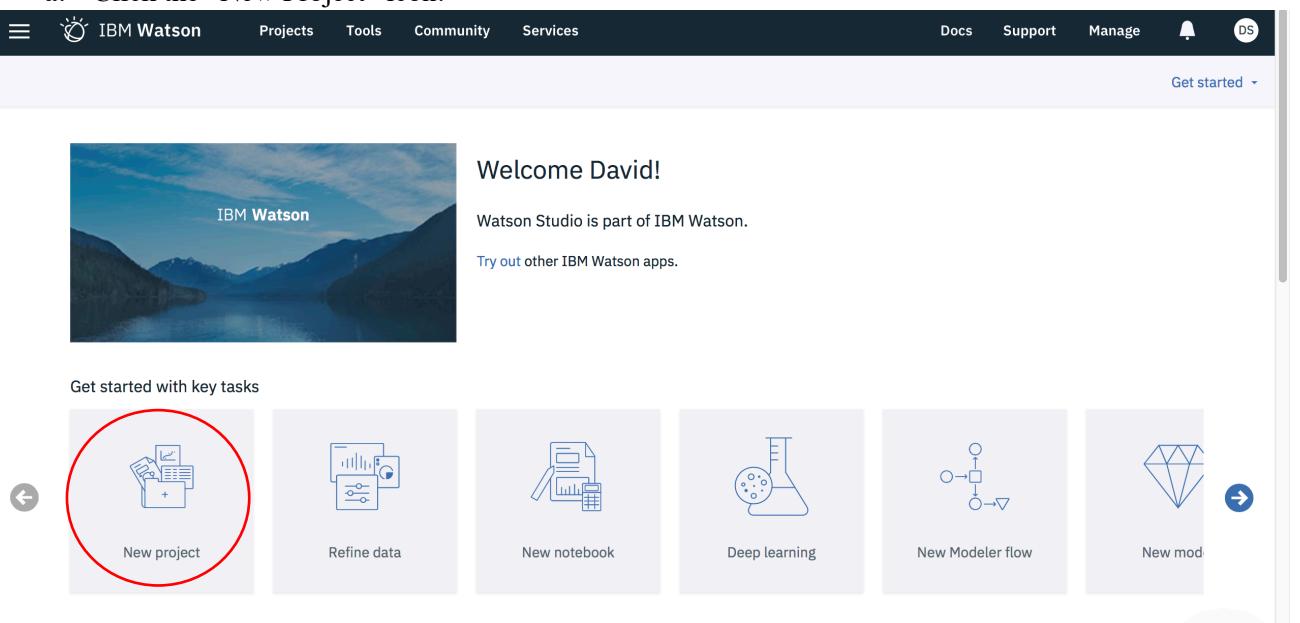
Step	Action
1	<p><u>Set up your IBM Cloud Account with Watson Studio</u></p> <ol style="list-style-type: none"> Log into IBM Data Science Experience at http://dataplatform.ibm.com/ Under IBM Watson Studio, click “Learn More”. When the Watson Studio page appears, click “Start your Free Trial”. Follow the instructions to create an account or use an existing IBM Cloud account. 

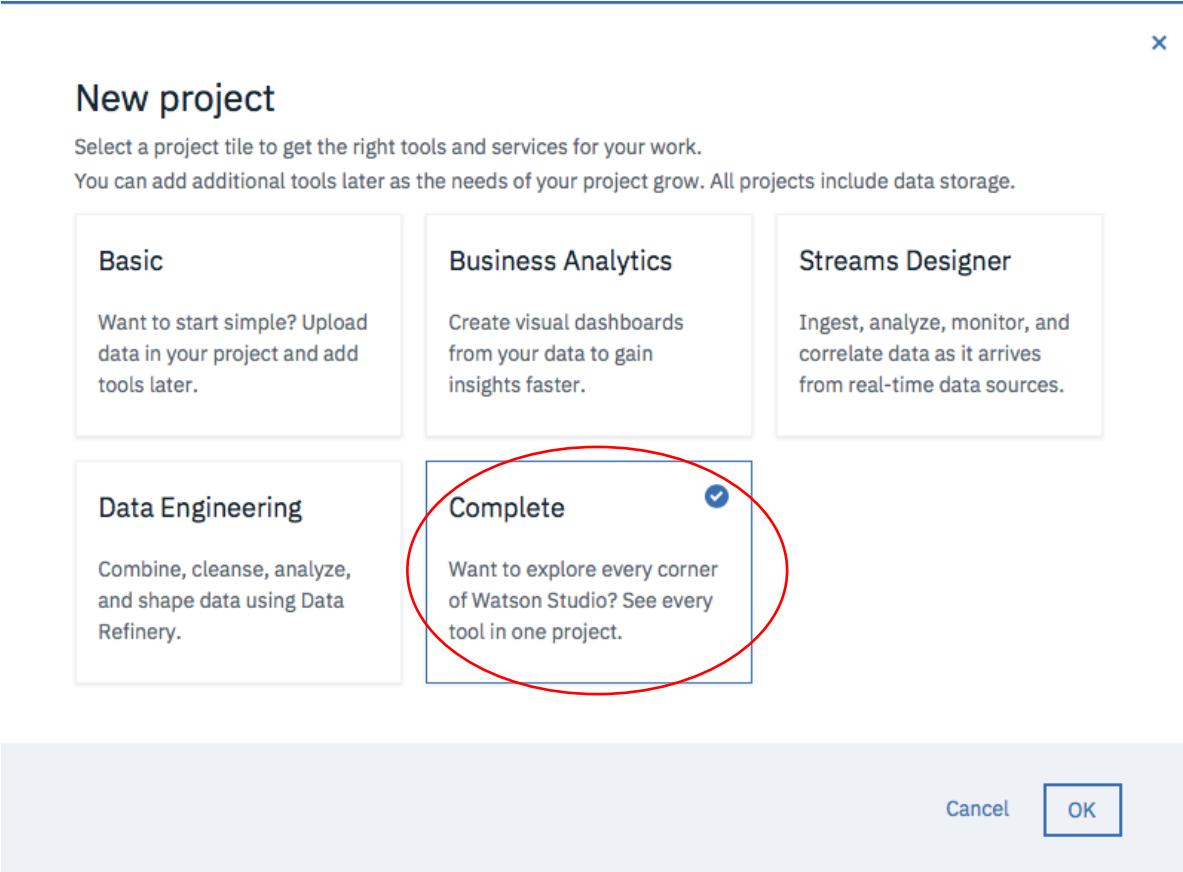
Step	Action
	

d. You will need to log into your email account to confirm and complete account registration.
 e. Select the “Confirm Account” link in the email to be redirected to IBM Cloud.

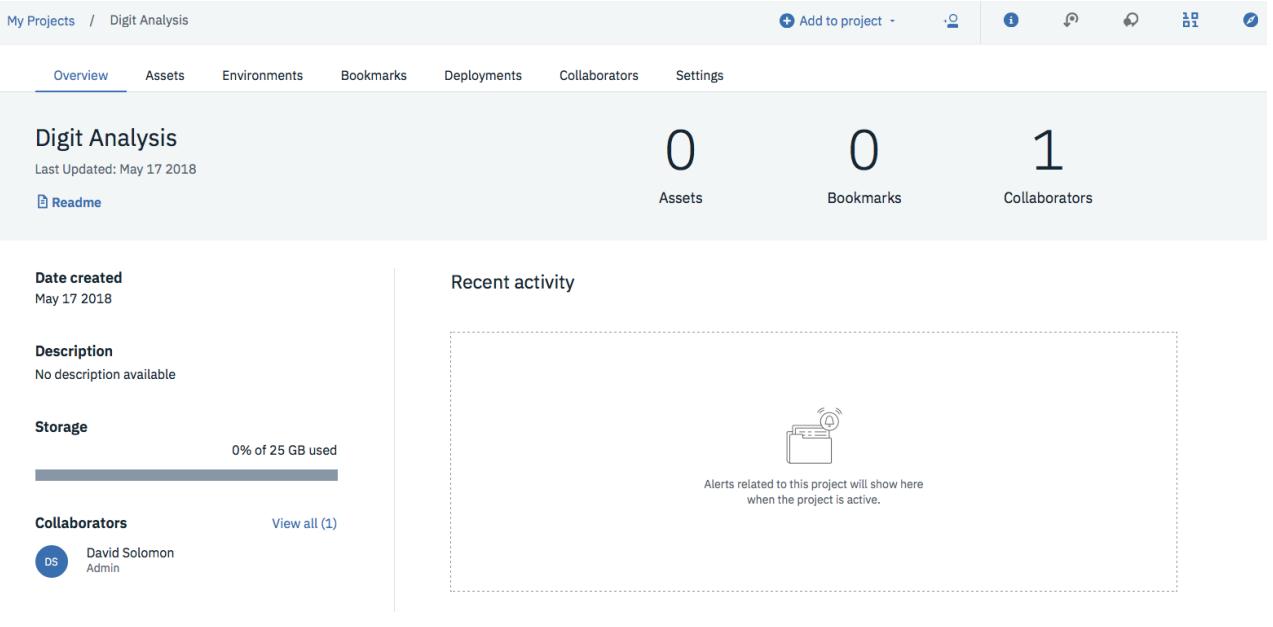


Step	Action
	<p>Sign in to IBM</p> <p>Sign in to IBM</p> <p>IBMid: ericreuter2@yahoo.com</p> <p>Forgot your password?</p> <p>New? Create an IBMid.</p> <p>Use a different IBMid or email</p> <p>Continue</p> <p>Sign in</p>
	<p>f. When you first sign into your new account, you will need to select and organization and space.</p> <p>g. We will be using the default options for this lab. Click “Continue”.</p>  <p>h. Once you see that the Object-Storage is done provisioning, click “Get Started”.</p> 

Step	Action
	
2	<p>Create your Watson Studio Project</p> <p>a. Click the “New Project” icon.</p>  <p>b. For project type, select “Complete”</p>

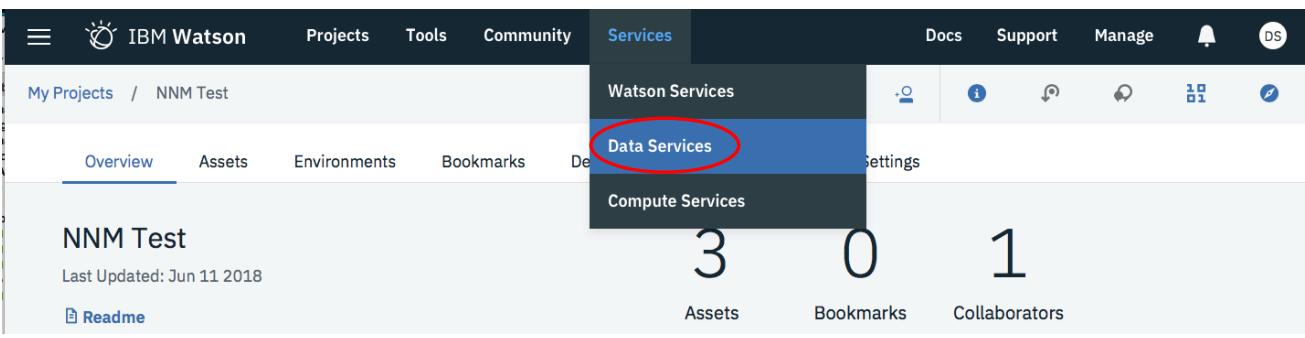
Step	Action
	 <p>New project</p> <p>Select a project tile to get the right tools and services for your work. You can add additional tools later as the needs of your project grow. All projects include data storage.</p> <p>Basic Want to start simple? Upload data in your project and add tools later.</p> <p>Business Analytics Create visual dashboards from your data to gain insights faster.</p> <p>Streams Designer Ingest, analyze, monitor, and correlate data as it arrives from real-time data sources.</p> <p>Complete <input checked="" type="checkbox"/> Want to explore every corner of Watson Studio? See every tool in one project.</p> <p>Cancel OK</p> <p>c. In order to create a project, you will need to have a place to store your project's assets. Watson Studio projects utilize an associated IBM Cloud Object Storage (COS) services for this purpose. There is a free tier of COS that can be easily created. To do this, click "Add" under "Select Storage Service" on the new project page.</p>

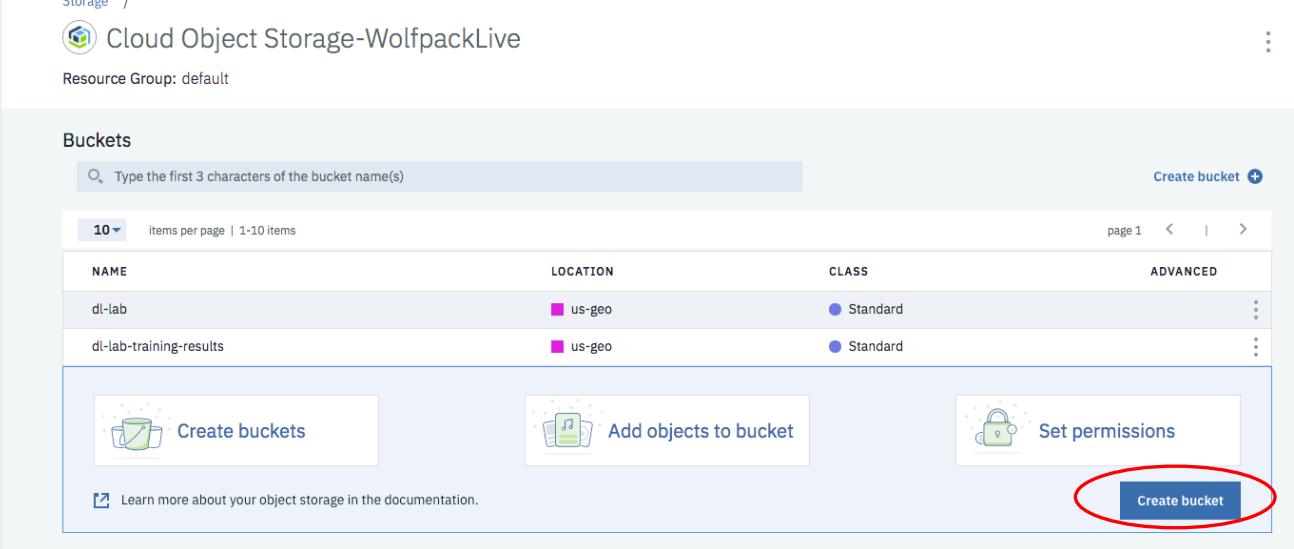
Step	Action									
	<p>New project</p> <p>Define project details</p> <p>Name Digit Analysis</p> <p>Description Project description</p> <p>Choose project options</p> <p><input type="checkbox"/> Restrict who can be a collaborator <small>(i)</small></p> <p>Project will include integration with Cloud Object Storage for storing project assets.</p> <p>Define storage</p> <p>① Select storage service Add Add an existing storage instance and then return to this page and click Refresh.</p> <p>② Refresh</p> <p>d. Select “Lite” for the service type and Click “Create”</p> <p>Cloud Object Storage</p> <p>Pricing Plan: Monthly Process shown above reflect the: United States</p> <table border="1"> <thead> <tr> <th>PLAN</th> <th>FEATURES</th> <th>PRICING</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="radio"/> Lite</td> <td> 1 COS Service Instance Storage up to 25 GiB/mo. Up to 20,000 GET requests/mo. Up to 2,000 PUT/DELETE/mo. Up to 100 Reference ID/mo. Up to 500 Public Outbound <small>Applies to aggregate total across all storage bucket classes</small> </td> <td>Free</td> </tr> <tr> <td><input type="radio"/> Standard</td> <td><small>This is no minimum fee, so you pay only for what you use.</small></td> <td></td> </tr> </tbody> </table> <p>The Lite service plan for Cloud Object Storage includes Regional and Cross Regional resiliency, flexible data classes, and built-in security.</p> <p>e. Click “Confirm”.</p> <p>Confirm Creation</p> <p>Plan Lite</p> <p>Resource group Default</p> <p>Service name cloud-object-storage-md</p> <p>Cancel Confirm</p> <p>f. Once back in the new project page, click “Refresh”, enter “Digit Analysis” for the project name, and click “Create”.</p>	PLAN	FEATURES	PRICING	<input checked="" type="radio"/> Lite	1 COS Service Instance Storage up to 25 GiB/mo. Up to 20,000 GET requests/mo. Up to 2,000 PUT/DELETE/mo. Up to 100 Reference ID/mo. Up to 500 Public Outbound <small>Applies to aggregate total across all storage bucket classes</small>	Free	<input type="radio"/> Standard	<small>This is no minimum fee, so you pay only for what you use.</small>	
PLAN	FEATURES	PRICING								
<input checked="" type="radio"/> Lite	1 COS Service Instance Storage up to 25 GiB/mo. Up to 20,000 GET requests/mo. Up to 2,000 PUT/DELETE/mo. Up to 100 Reference ID/mo. Up to 500 Public Outbound <small>Applies to aggregate total across all storage bucket classes</small>	Free								
<input type="radio"/> Standard	<small>This is no minimum fee, so you pay only for what you use.</small>									

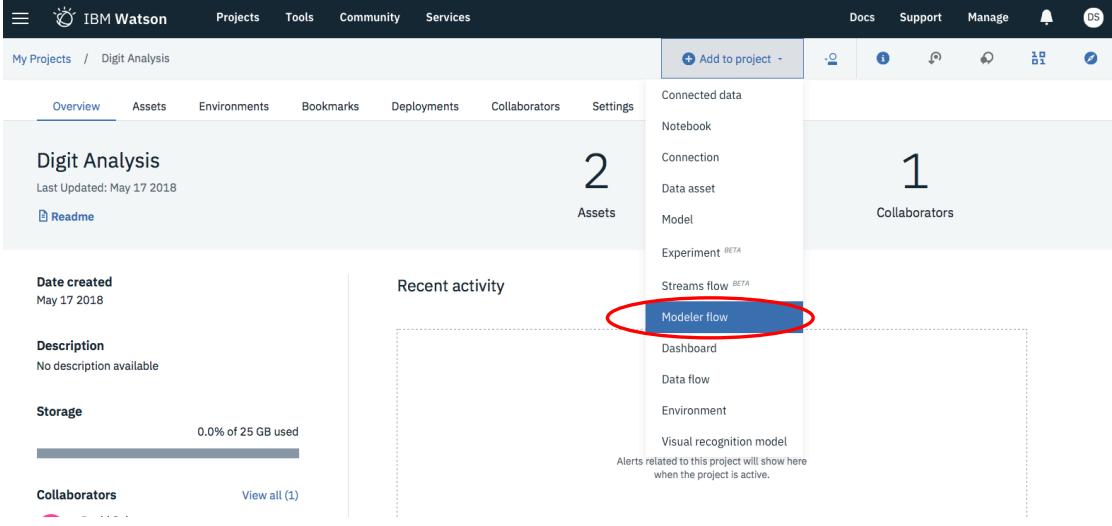
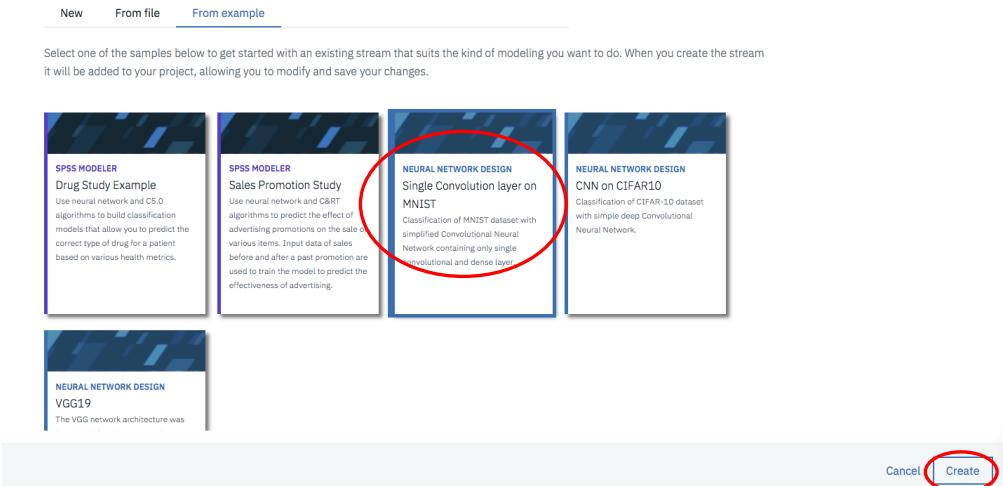
Step	Action
	<p>New project</p> <p>Define project details</p> <p>Name Digit Analysis</p> <p>Description Project description</p> <p>Choose project options</p> <p><input type="checkbox"/> Restrict who can be a collaborator <small>①</small></p> <p>Project will include integration with Cloud Object Storage for storing project assets.</p> <p>Define storage</p> <p>① Select storage service Add Add an object storage instance and then return to this page and click Refresh. ② Refresh</p> <p>g. You have now successfully created your Watson Studio project. You should see your project's main page.</p> 

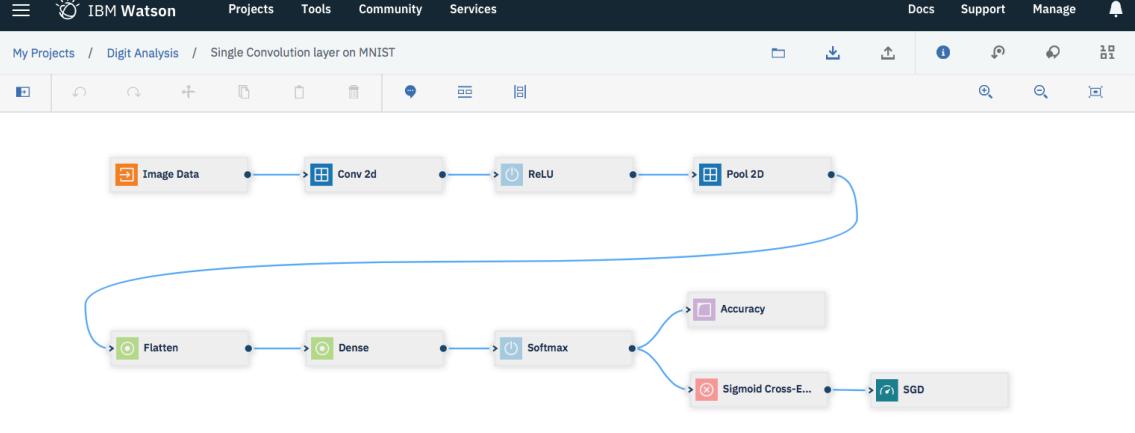
Module 2: Create a Model Flow to Design a Neural Network for MNIST Data

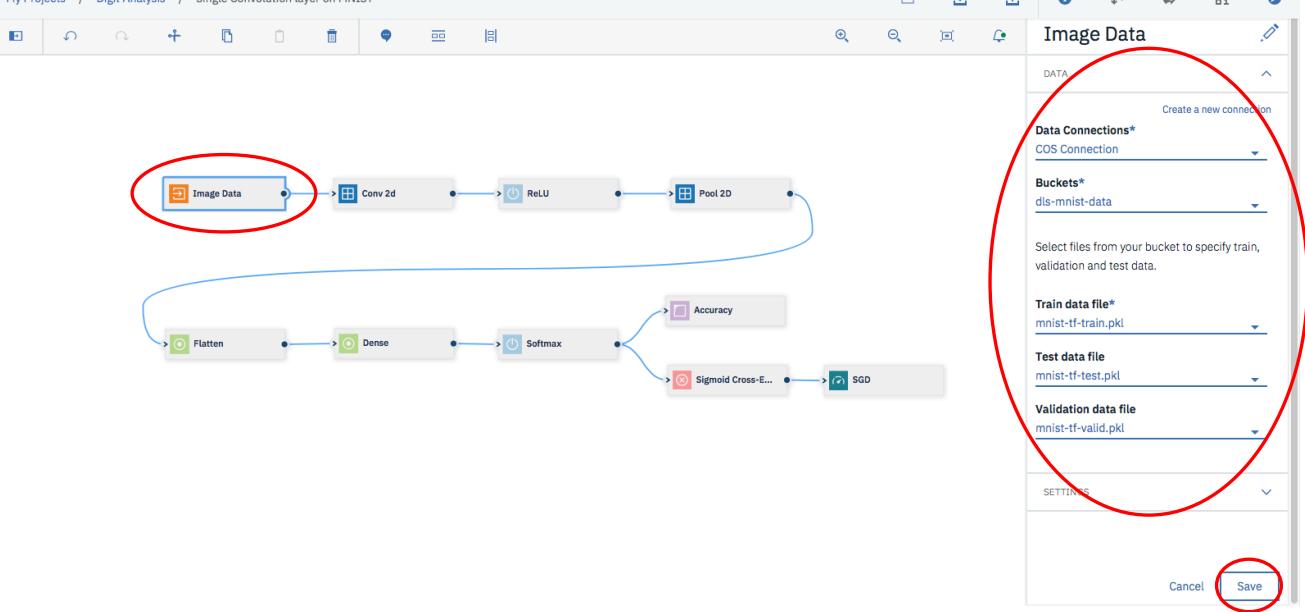
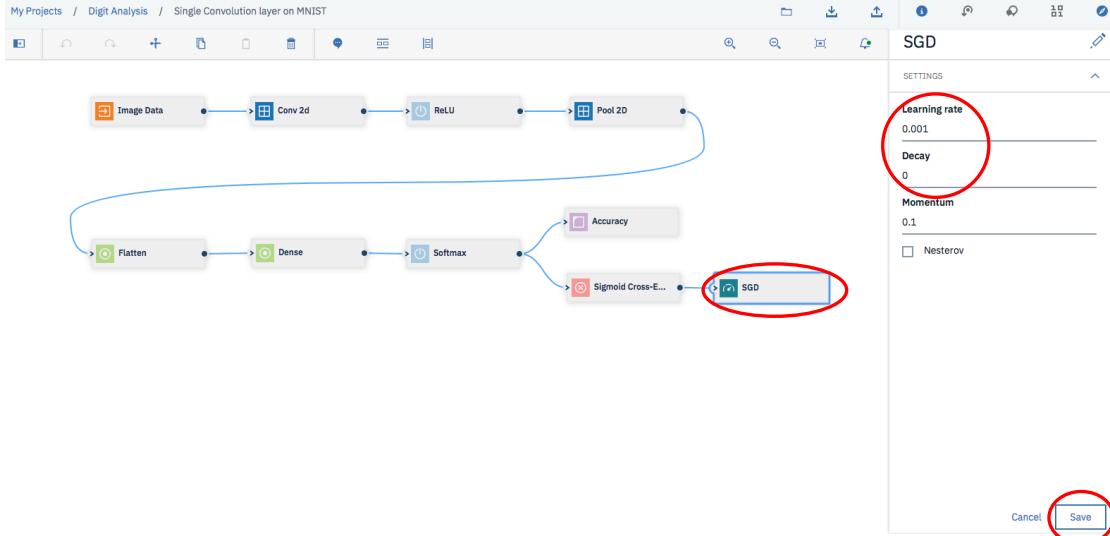
In this module, you will use the Watson Studio Model Flow editor and Experiment Builder to design and build a single layer Neural Network model for the MNIST dataset.

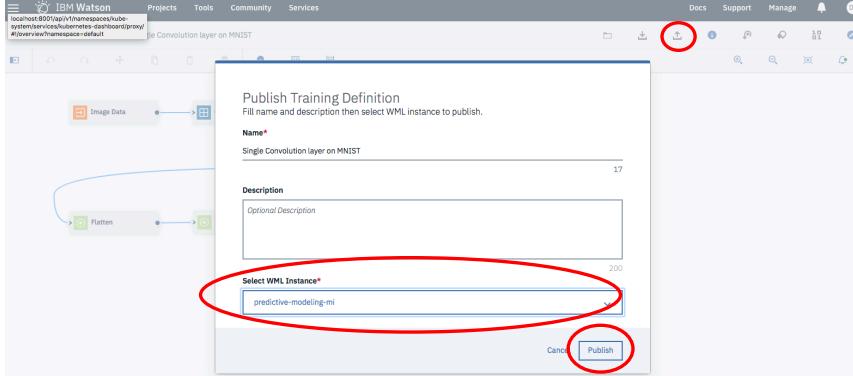
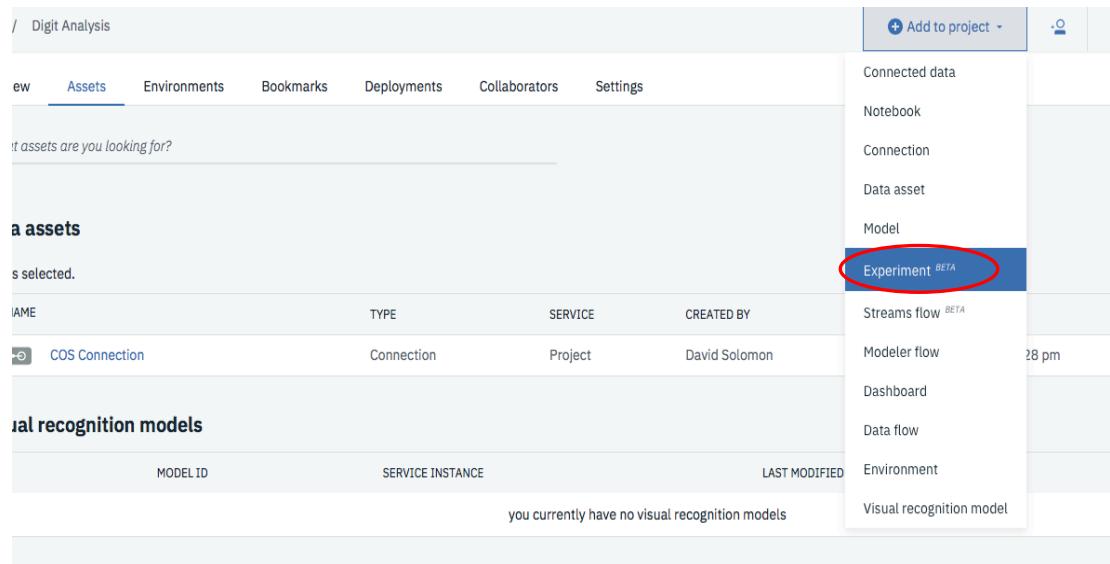
Step	Action
1	<p><u>Upload the Training and Test Data to the IBM Cloud</u></p> <p>Before we can start working with our Neural Network model, we will need to obtain our training and testing data and make it visible to your project. This data consists of sample handwritten digits. Our Neural Network modeler requires that this data be located in Cloud Object Storage</p> <p>a. Download the following files from this location:</p> <p style="text-align: center;">http://ibm.biz/BdZNbX</p> <ul style="list-style-type: none"> ○ mnist-tf-train.pkl ○ mnist-tf-validate.pkl ○ mnist-tf-test.pkl ○ tf-mnist-test-payload-pkl.json <p>b. In Watson Studio, go to the “Services” menu at the top and select “Data Services”</p>  <p>c. You will see your project’s Cloud Object Storage instance listed. Select this instance to go to the details page.</p>

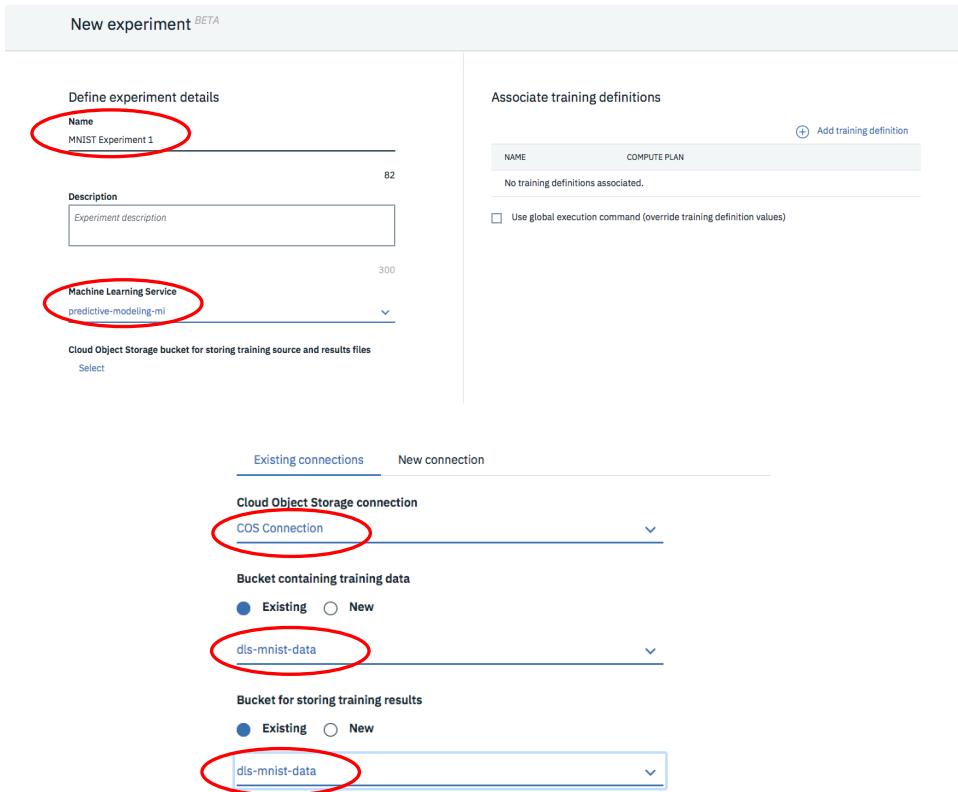
Step	Action												
	 <p>Storage / Cloud Object Storage-WolfpackLive Resource Group: default</p> <p>Buckets</p> <p>Type the first 3 characters of the bucket name(s)</p> <p>Create bucket +</p> <table border="1"> <thead> <tr> <th>NAME</th> <th>LOCATION</th> <th>CLASS</th> <th>ADVANCED</th> </tr> </thead> <tbody> <tr> <td>dl-lab</td> <td>us-geo</td> <td>Standard</td> <td>⋮</td> </tr> <tr> <td>dl-lab-training-results</td> <td>us-geo</td> <td>Standard</td> <td>⋮</td> </tr> </tbody> </table> <p>Create buckets Add objects to bucket Set permissions</p> <p>Learn more about your object storage in the documentation.</p>	NAME	LOCATION	CLASS	ADVANCED	dl-lab	us-geo	Standard	⋮	dl-lab-training-results	us-geo	Standard	⋮
NAME	LOCATION	CLASS	ADVANCED										
dl-lab	us-geo	Standard	⋮										
dl-lab-training-results	us-geo	Standard	⋮										
2	<p>Create a Model Flow</p> <p>a. From your project's overview page, click the "Add to Project" menu and select "Modeler Flow"</p>												

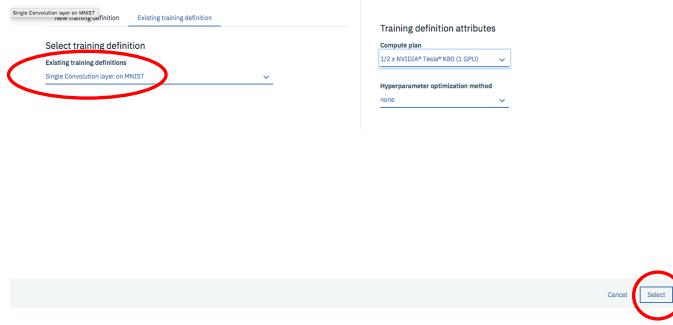
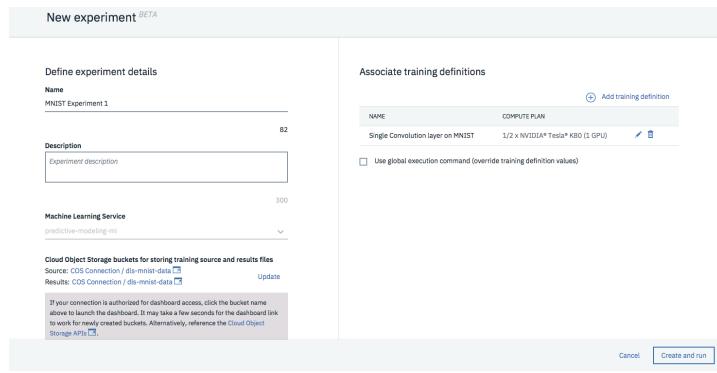
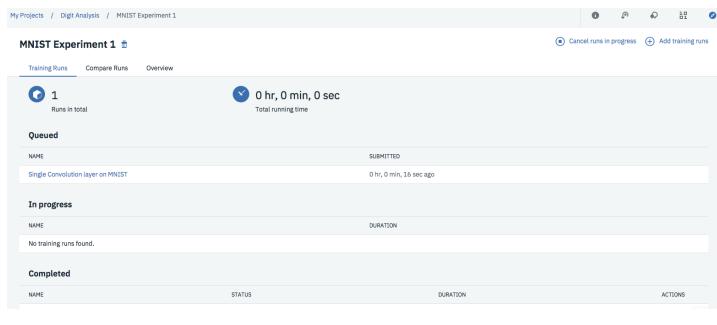
Step	Action
	 <p>In Modeler Flow creation page, select “From Example”. Select the “Single Convolution layer on MNIST” and click “Create”.</p>  <p>b. You will be brought to the model canvas that has been populated with our example model.</p>

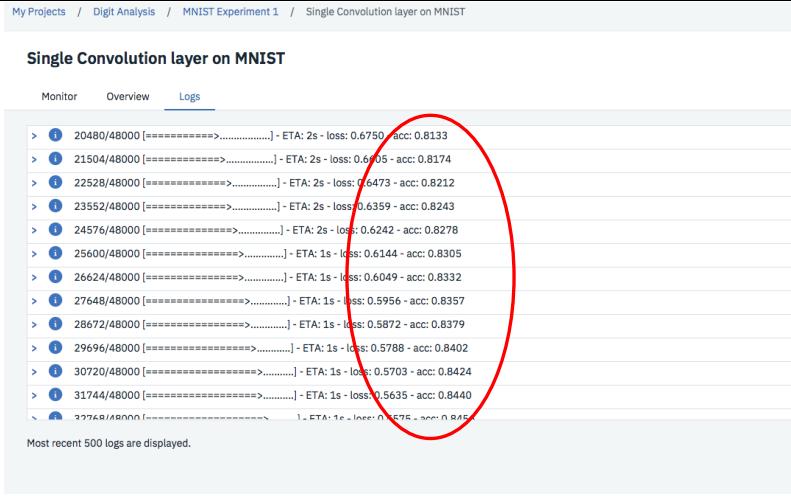
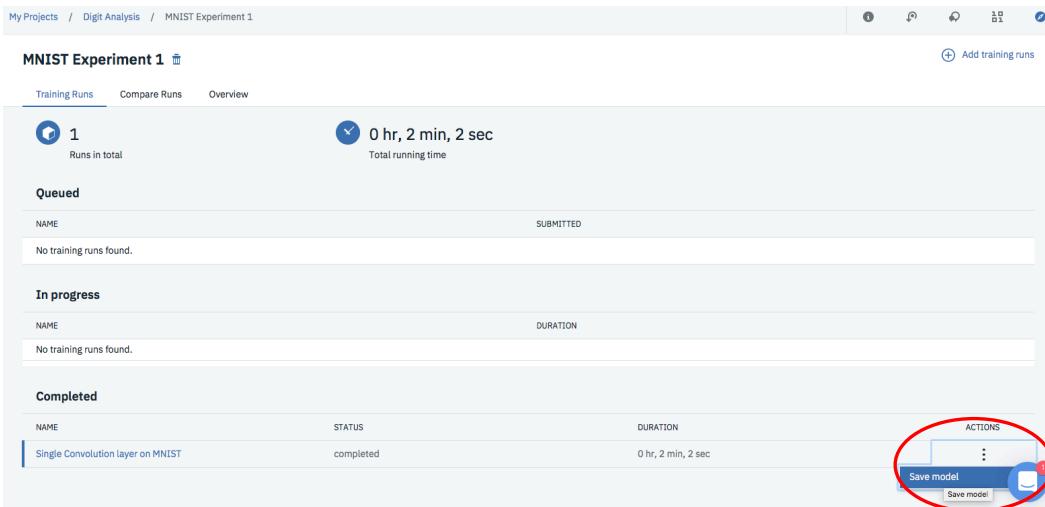
Step	Action
	 <p>c. In order to finalize this, we first need to point it to the data files the model will use for training and validation. Perform the following:</p> <ol style="list-style-type: none"> 1. Double-click the first node, labeled “Image Data” (this opens the node details side panel.) 2. In the details side panel, in the “DATA” section, in the “Data Connections” drop-down menu, select the Cloud Object Storage connection that you created in the previous step. 3. In the “Buckets” drop-down menu, select your training data bucket (“mnist-training”). 4. In the “Train data file” drop-down menu, select "mnist-train.pkl". 5. In the “Test data file” drop-down menu, select "mnist-test.pkl". 6. In the “Validation data file” drop-down menu, select "mnist-validate.pkl". 7. Click “Save”.

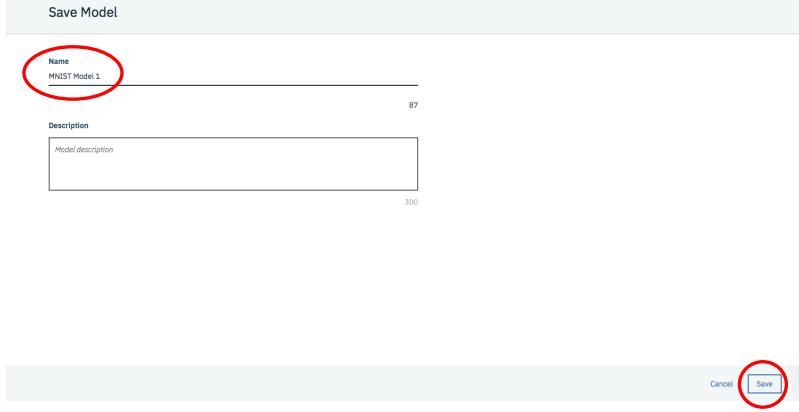
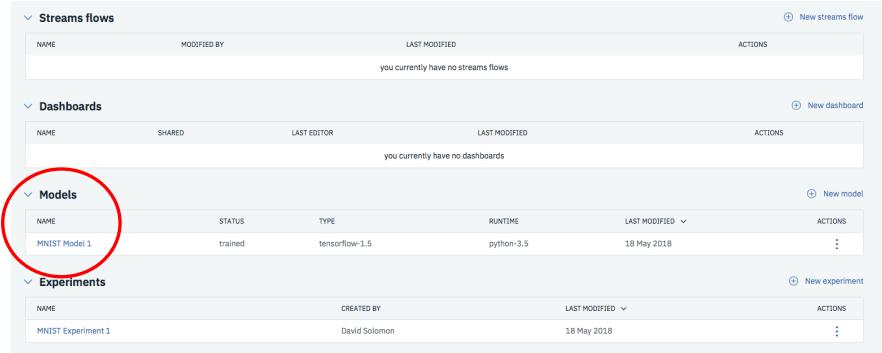
Step	Action
	 <p>d. Next, you will need to make some changes to the final node (SGD).</p> <ol style="list-style-type: none"> 1. Double-click the final node, labeled “SGD” (this opens the node details side panel.) 2. Set “Learning rate” to 0.001 3. Set “Decay” to 0 4. Click “Save”.  <p>e. You can now save our neural network design as a training definition.</p>

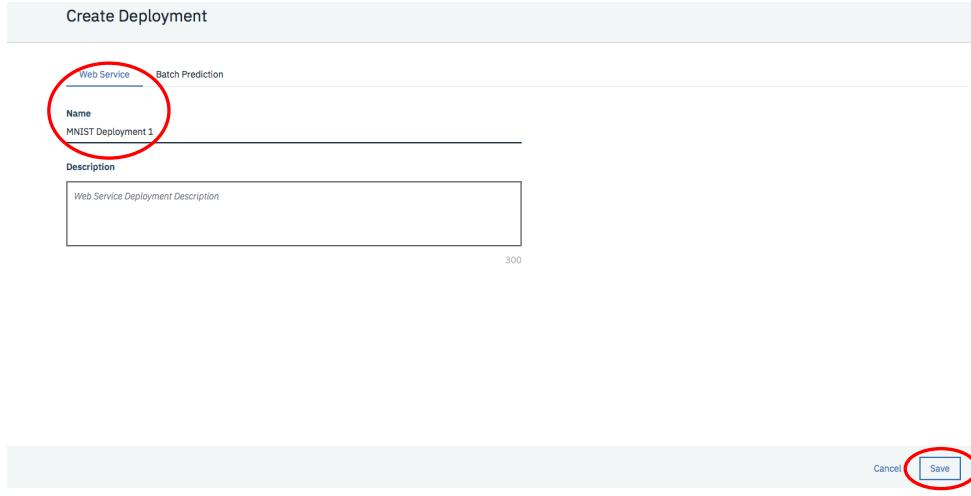
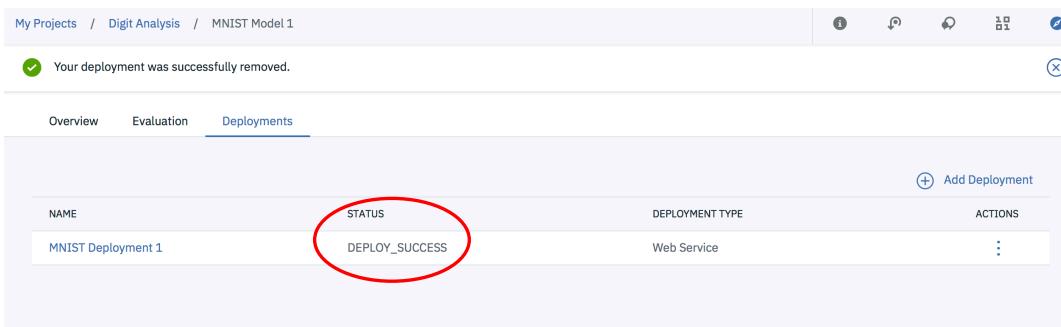
Step	Action
	<ol style="list-style-type: none"> 1. Click the publish icon (). 2. Specify a name for the training definition. 3. In the “Specify WML Instance” drop-down menu, select the Watson Machine Learning service instance that is associated with the project. 4. Click the “Publish” button. 
3	<p><u>Train your Model using the Experiment Builder</u></p> <p>a. From your project’s overview page, click the “Add to Project” menu and select “Experiment”</p>  <p>b. Complete the following in the new Experiment page:</p> <ol style="list-style-type: none"> 1. Name your experiment “MNIST Experiment 1”

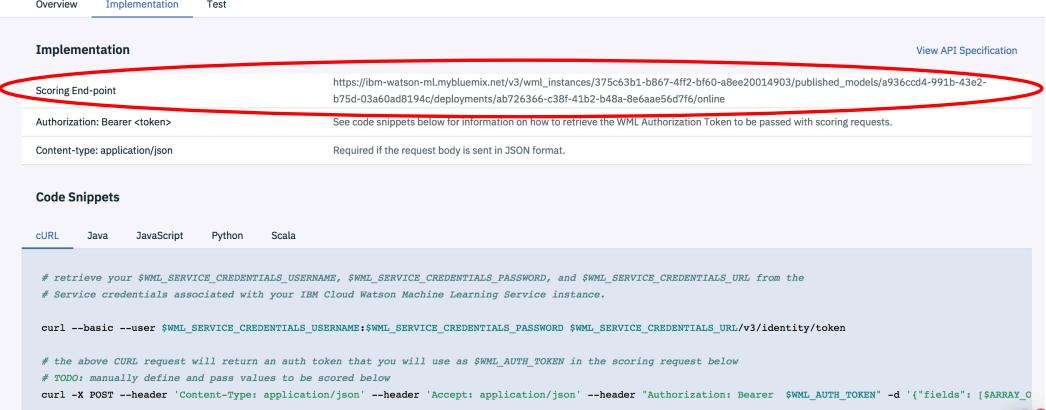
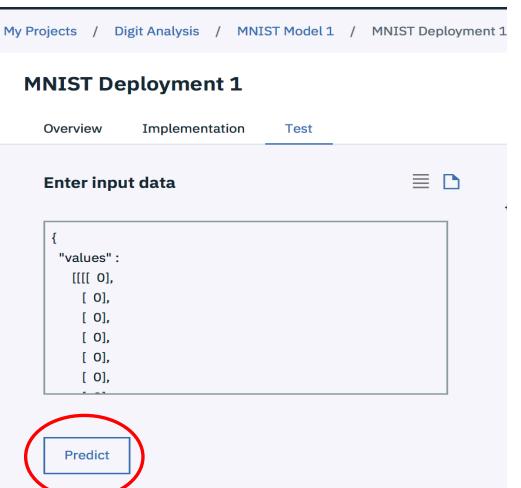
Step	Action
	<p>2. In the “Machine Learning Service” drop-down, select the Watson Machine Learning instance associated with your project</p> <p>3. In the area for “Cloud Object Storage” click “Select”.</p> <p>4. In the “Cloud Object Storage connection” drop-down list, select the Cloud Object Storage connection you created before.</p> <p>5. For the “Training Data Bucket”, select the training data bucket you created before.</p> <p>6. For the “Training Results Bucket”, select the results bucket you created before.</p> <p>7. Click “Create”</p>  <p>c. Add the training definition you created in the previous step.</p> <ol style="list-style-type: none"> 1. Click Add training definition. 2. Click the Existing training definitions tab. 3. From the drop-down menu, select the training definition you published from the flow editor. 4. Select "1/2 x NVIDIA Tesla K80 (1 GPU)" for the compute plan. 5. Select "none" for the hyperparameter optimization method. 6. Click “Select”

Step	Action
	 <p>d. Click Create and run</p>  <p>e. You can monitor the status of your training run.</p>  <p>f. Once the training run is completed, you can review the details of the training run results and other output by clicking on the training definition.</p>

Step	Action
	 <p>g. We will save the model, so it can be deployed into a Watson Machine Learning for testing and use by applications. To do, return to the Experiment page. Select the “Actions” menu for the training definition located in the “Completed” section of the page. From that menu, select “Save Model”. Save the model with the name “MNIST Model 1” and click “Save”.</p> 

Step	Action
	
4	<p><u>Deploy and Test your Model</u></p> <ol style="list-style-type: none"> We will now deploy the model into your Watson Machine Learning Service. Once the model is deployed, you can test it or have developers access it from their applications in production using a simple Rest URL. To deploy the model, return to the “Assets” page of your project. In the “Models” section, you should see your saved model listed. Click on the model.  <ol style="list-style-type: none"> When the model opens, go to the “Deployments” tab and click “Add Deployment”  <ol style="list-style-type: none"> On the create deployment page, select the “Web Service” tab, name your deployment “MNIST Deployment 1” and click “Save”.

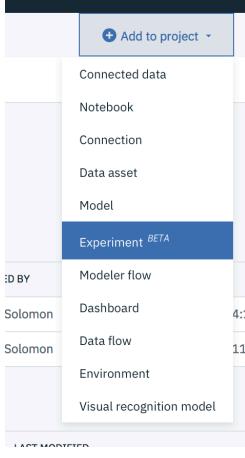
Step	Action
	 <p>d. Wait until you receive an indication that the deployment was successful.</p>  <p>e. Once deployed, we can either test the deployment or provide a URL to developers for use in applications. Click on the deployment name to view the details. Next, click on the "Implementation" tab. Notice the scoring URL, as well as the sample code for accessing it.</p>

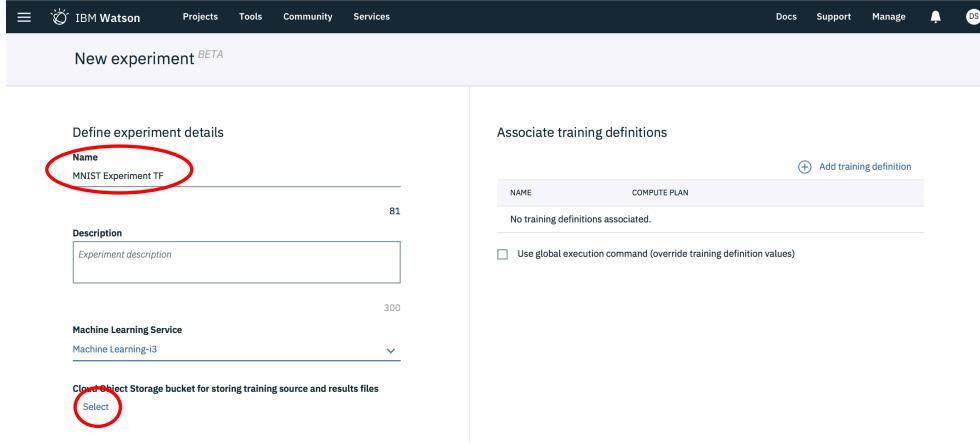
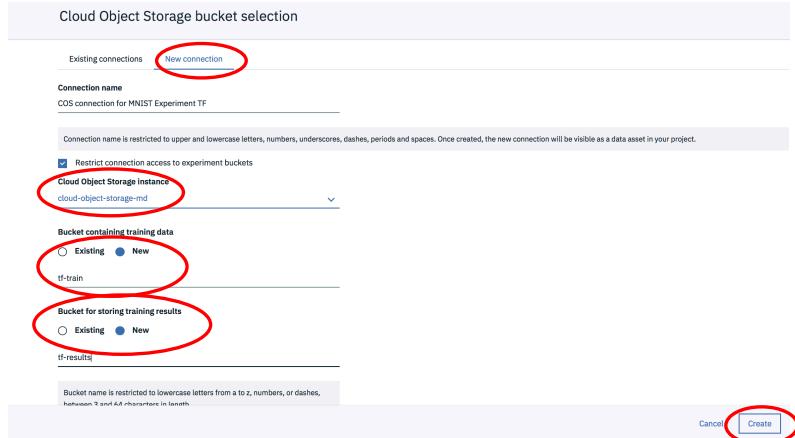
Step	Action
	 <p>The screenshot shows the "Implementation" tab of the MNIST Deployment 1 page. A red circle highlights the "Scoring End-point" field, which contains the URL: <code>https://ibm-watson-ml.mybluemix.net/v3/wml_instances/375c63b1-b867-4ff2-bf60-a8ee20014903/published_models/a936cccd4-991b-43e2-b75d-03a60ad8194c/deployments/ba726366-c38f-41b2-b48a-8e6aae56d7f6/online</code>. Another red circle highlights the "Authorization: Bearer <token>" field.</p>
f.	<p>Click on the “Test” tab. Copy and paste the contents of the “tf-mnist-test-payload-pkl.json” file you previously downloaded into the “Enter Input Data” field.</p> <p>g. Click “Predict”</p>  <p>The screenshot shows the "Test" tab of the MNIST Deployment 1 page. The "Enter input data" field contains a JSON array representing an image of the digit "7". The "Predict" button at the bottom of the field is highlighted with a red circle.</p>

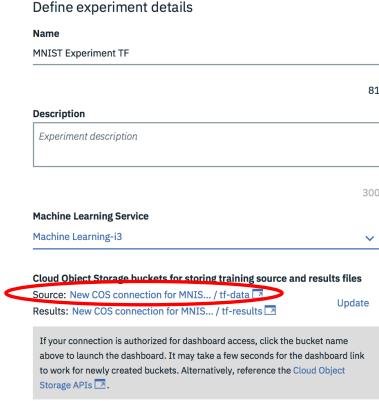
Step	Action
	<p>is labeled as “1” meaning that the prediction is that our image is a “7”.</p> <div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin-left: auto; margin-right: auto;"><pre>"fields": ["prediction"], "values": [[0, 0, 0, 0, 0, 0, 0, 1, // Circled in red 0, 0]]</pre></div>

Module 3: Building and Deploying a TensorFlow Model

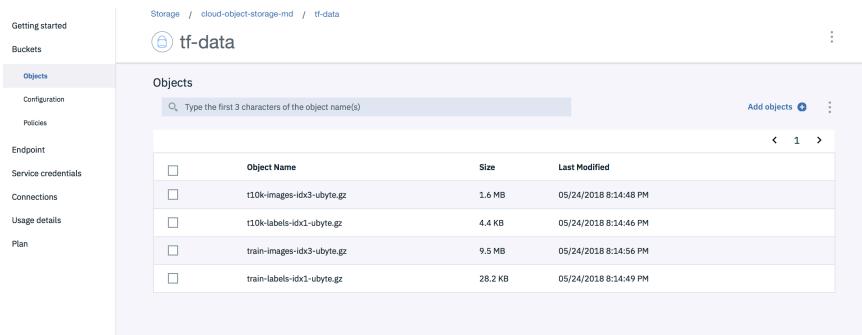
In this module, you will use the Experiment Builder to train and deploy an existing TensorFlow model. In addition, you will use hyperparameter optimizations as part of the training process. These optimizations can be used to refine the performance of your model.

Step	Action
1	<p><u>Download Sample Data and Model</u></p> <p>a. Download the 4 MNIST sample data files from the following URL:</p> <p style="text-align: center;">http://yann.lecun.com/exdb/mnist/</p> <p>b. Download the example TensorFlow sample code from the following URL</p> <p style="text-align: center;">http://ibm.biz/BdZNVi</p>
2	<p><u>Create and Run a New Experiment</u></p> <p>a. Go to the overview page of your project. Click on the “Add to Project” menu and select “Experiment”.</p>  <p>b. In the new experiment page, name your experiment “MNIST Experiment TF”.</p> <p>c. We will need to provide a location for the experiment to access training data and to store results. We use Cloud Object Storage for this purpose. Click on “Select” under the Cloud Object Storage Bucket area.</p>

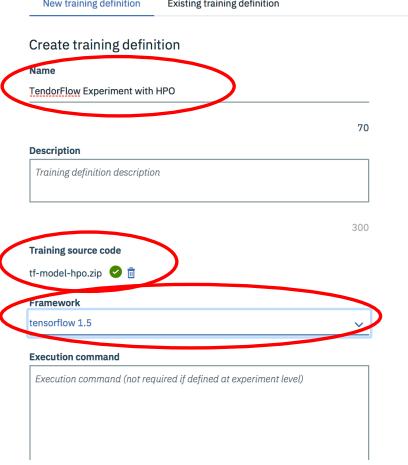
Step	Action
	 <p>d. In the Cloud Object Storage page, select “New Connection” to create a connection to your project’s default COS instance.</p> <p>e. From the list, select your existing COS instance.</p> <p>f. Click on “New” for both your training and results bucket selection.</p> <p>g. For you training data bucket name, enter “tf-data” and for training results enter “tf-results”</p> <p>h. Click “Create”</p>  <p>i. Once the experiment is created, click on the link to the “source” bucket, as shown below. This will open the COS bucket so you can add the training data.</p>

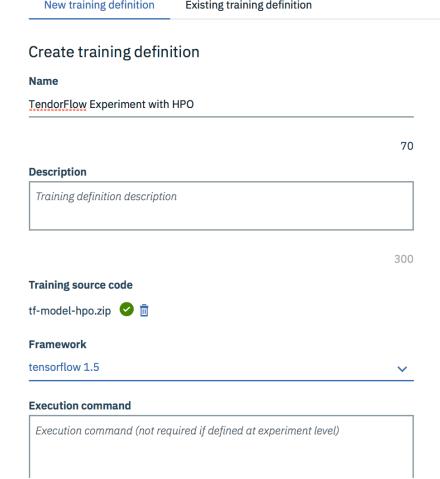
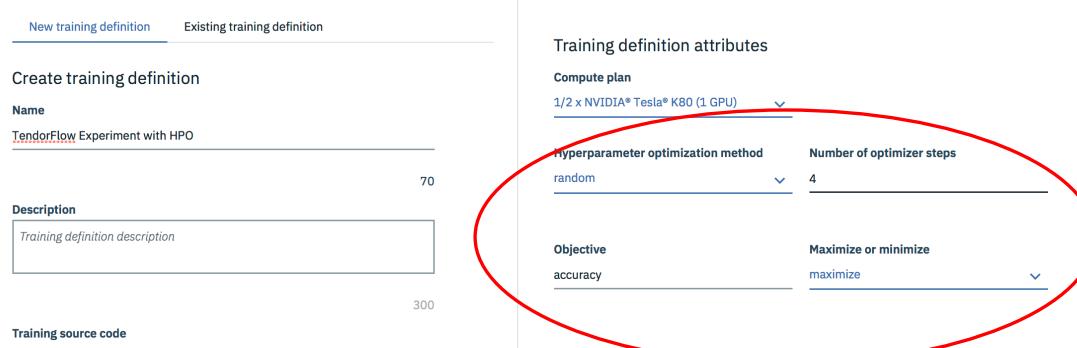
Step	Action					
	 <p>Define experiment details</p> <p>Name MNIST Experiment TF</p> <p>Description Experiment description</p> <p>Machine Learning Service Machine Learning-i3</p> <p>Cloud Object Storage buckets for storing training source and results files</p> <p>Source: New COS connection for MNIST... / tf-data </p> <p>Results: New COS connection for MNIST... / tf-results </p> <p>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 [?].</p>	 <p>Associate training definitions</p> <p>+ Add training definition</p> <table border="1"> <thead> <tr> <th>NAME</th> <th>COMPUTE PLAN</th> </tr> </thead> <tbody> <tr> <td>No training definitions associated.</td> <td></td> </tr> </tbody> </table> <p><input type="checkbox"/> Use global execution command (override training definition values)</p>	NAME	COMPUTE PLAN	No training definitions associated.	
NAME	COMPUTE PLAN					
No training definitions associated.						

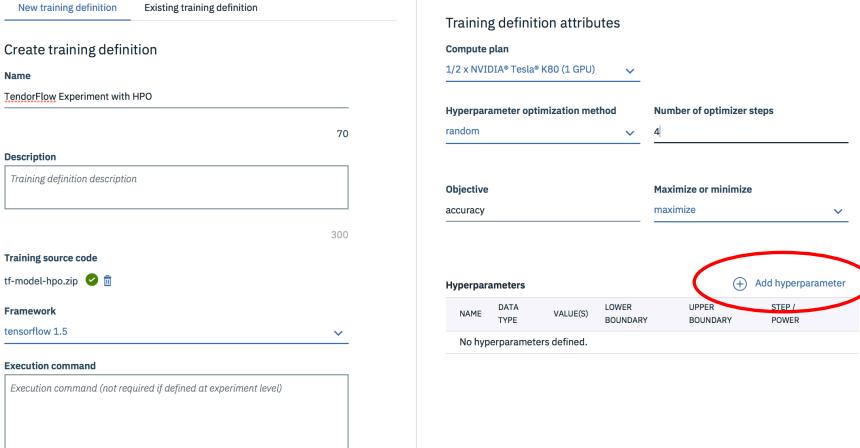
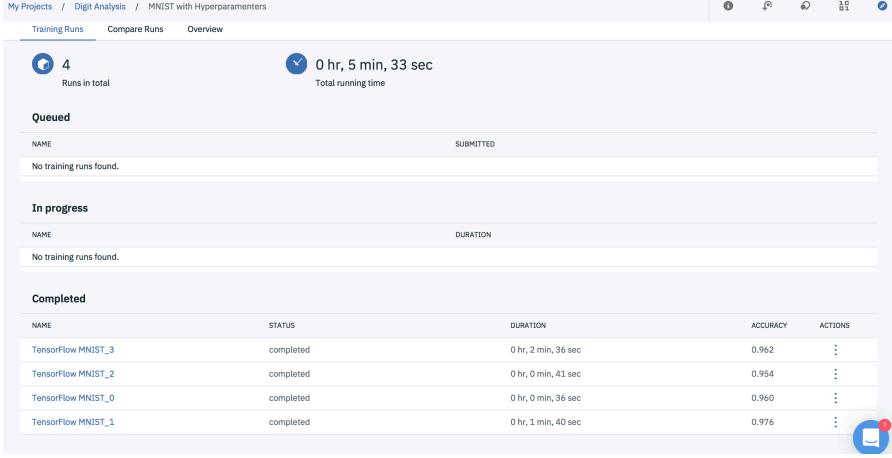
- j. Once the bucket is open, click on the “Add objects” link and select the MNIST files you downloaded earlier. The files will then be uploaded to the bucket.

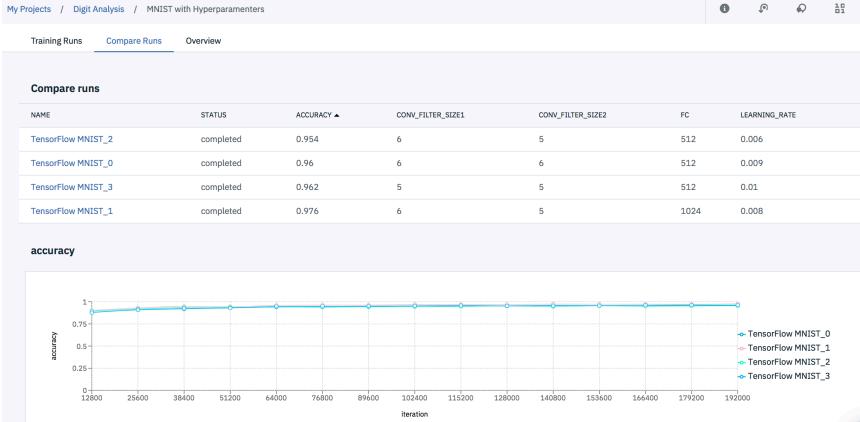
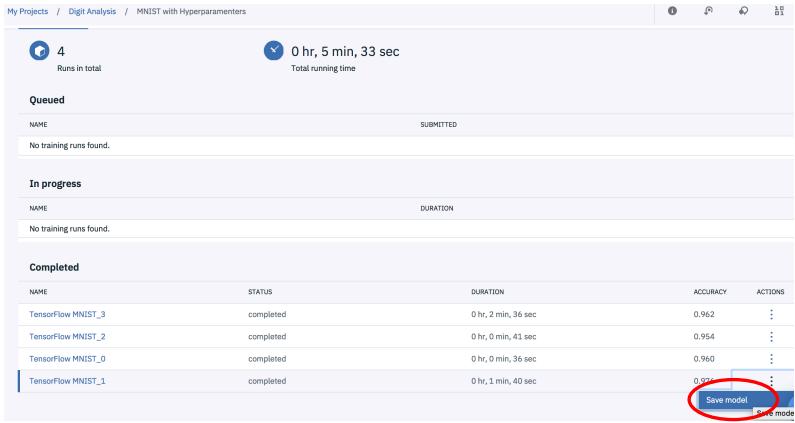


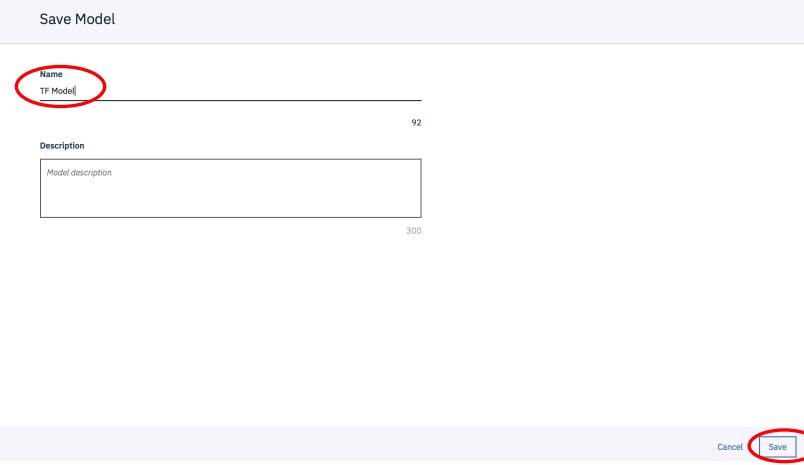
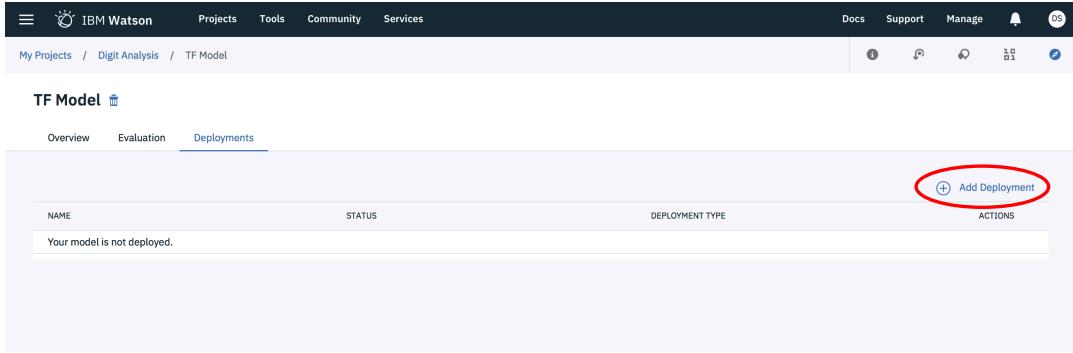
- k. We can now add our TensorFlow training definition code to our experiment. Return to the experiment page and click “Add Training Definition”. Enter “TensorFlow Experiment with HPO” as the name.
- l. Click “browse” and select the training definition file you previously downloaded.
- m. Select “TensorFlow 1.5” as the framework

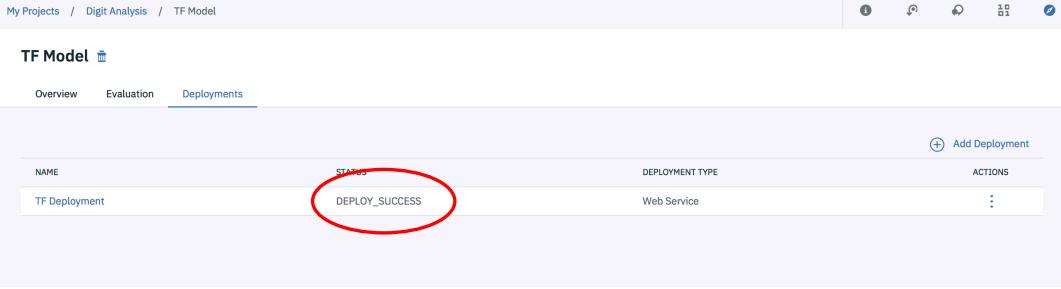
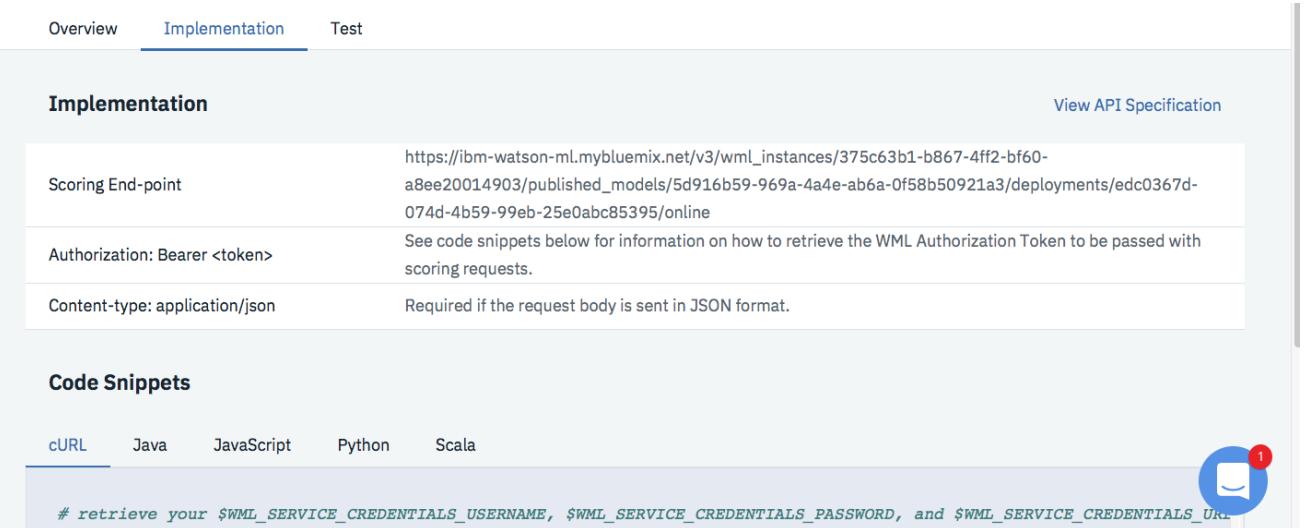
Step	Action
	 <p>n. For “Compute Plan”, select “1/2 x NVIDIA Tesla K80 (1 GPU)”.</p> <p>o. In the “Execution command” field, enter the following,</p> <pre data-bbox="218 882 1372 1072"> python3 convolutional_network.py --trainImagesFile \${DATA_DIR}/train-images-idx3-ubyte.gz --trainLabelsFile \${DATA_DIR}/train-labels-idx1-ubyte.gz --testImagesFile \${DATA_DIR}/t10k-images-idx3-ubyte.gz --testLabelsFile \${DATA_DIR}/t10k-labels-idx1-ubyte.gz --trainingIters 200000 </pre> <p>p. This training definition allows for the use of specific hyperparameter optimization (HPO) to enable us to assess the model, based on different conditions and assumptions. When the experiment is run, it can run multiple instances, each with a different HPO values to determine which combination of values provide the best performance and accuracy. For this lab, we will specify value ranges for 4 HPOs. To start, click on the “Hyperparameter optimization method” menu and select “Random”.</p>

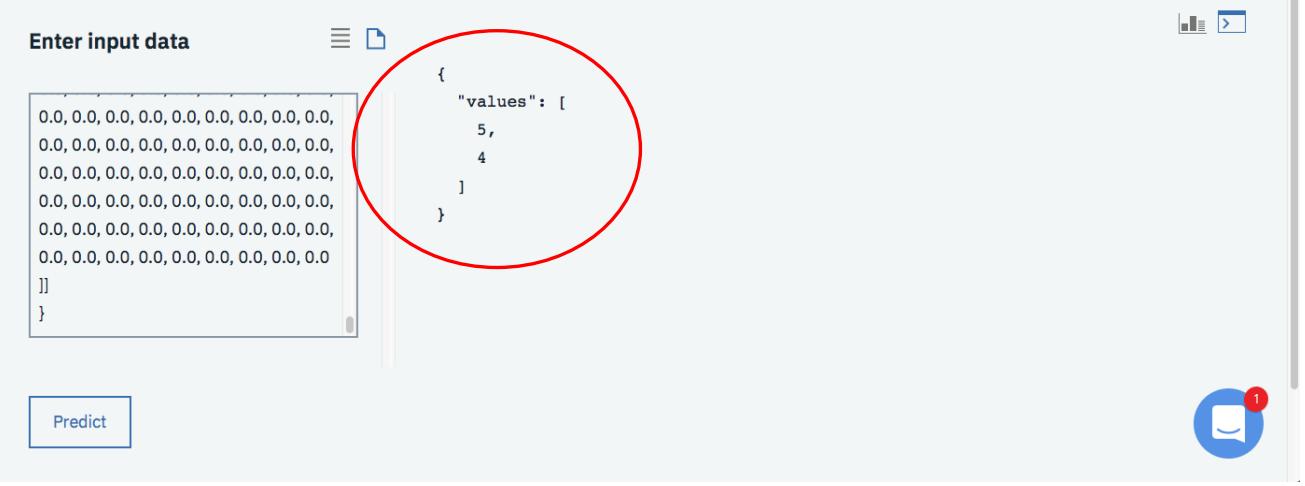
Step	Action
	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;">  <p>New training definition Existing training definition</p> <p>Create training definition</p> <p>Name <input type="text" value="TensorFlow Experiment with HPO"/></p> <p>Description <input type="text" value="Training definition description"/></p> <p>Training source code <input type="text" value="tf-model-hpo.zip"/>  </p> <p>Framework <input type="text" value="tensorflow 1.5"/></p> <p>Execution command <input type="text" value="Execution command (not required if defined at experiment level)"/></p> </div> <div style="width: 50%;"> <p>Training definition attributes</p> <p>Compute plan <input type="text" value="1/2 x NVIDIA® Tesla® K80 (1 GPU)"/></p> <p>Hyperparameter optimization method</p> <div style="border: 1px solid red; border-radius: 50%; width: 30px; height: 30px; margin-left: auto; margin-right: auto;"></div> <p>none</p> <p>RBFOpt</p> <p>random <input checked="" type="radio"/></p> </div> </div> <p>q. Additional options will now appear. Set the “number of optimizer steps” field to “4”, “Objective” to “Accuracy” and “Maximize or Minimize” to “maximize”.</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;">  <p>New training definition Existing training definition</p> <p>Create training definition</p> <p>Name <input type="text" value="TensorFlow Experiment with HPO"/></p> <p>Description <input type="text" value="Training definition description"/></p> <p>Training source code</p> </div> <div style="width: 50%;"> <p>Training definition attributes</p> <p>Compute plan <input type="text" value="1/2 x NVIDIA® Tesla® K80 (1 GPU)"/></p> <p>Hyperparameter optimization method <input type="text" value="random"/></p> <p>Number of optimizer steps <input type="text" value="4"/></p> <p>Objective accuracy <input type="text" value="maximize"/></p> <p>Maximize or minimize <input type="text" value="maximize"/></p> </div> </div> <p>r. We will now specify the HPO values that will be used for this experiment. To do this, click “add hyperparameter”.</p>

Step	Action																									
	 <p>s. Add the following hyperparameters, as specified in the following table.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Lower bound</th> <th>Upper bound</th> <th>Step/Power</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>learning_rate</td> <td>0.005</td> <td>0.01</td> <td>Step: 0.001</td> <td>Double</td> </tr> <tr> <td>conv_filter_size1</td> <td>5</td> <td>6</td> <td>Step: 1</td> <td>Integer</td> </tr> <tr> <td>conv_filter_size2</td> <td>5</td> <td>6</td> <td>Step: 1</td> <td>Integer</td> </tr> <tr> <td>fc</td> <td>9</td> <td>10</td> <td>Power: 2</td> <td>Integer</td> </tr> </tbody> </table> <p>t. Click “Create and Run” to run the experiment. The experiment control page will appear. Please note that it may take 5-10 minutes for your experiment to run.</p> 	Name	Lower bound	Upper bound	Step/Power	Data type	learning_rate	0.005	0.01	Step: 0.001	Double	conv_filter_size1	5	6	Step: 1	Integer	conv_filter_size2	5	6	Step: 1	Integer	fc	9	10	Power: 2	Integer
Name	Lower bound	Upper bound	Step/Power	Data type																						
learning_rate	0.005	0.01	Step: 0.001	Double																						
conv_filter_size1	5	6	Step: 1	Integer																						
conv_filter_size2	5	6	Step: 1	Integer																						
fc	9	10	Power: 2	Integer																						

Step	Action
3	<p><u>Review Experiment Results</u></p> <p>a. Once the experiment is complete, select the “Compare Runs” tab.</p>  <p>b. Note here that all four runs had very similar results in this case. Also note that each run used a different set of hyperparameter values, since we asked for a “random” allocation. We can also view the details of a specific run by clicking on its link.</p> <p>c. Click on the “Training Runs” tab. From here, we can create a model from any of these runs based on the results. Click on the “Actions” menu for the run with the highest accuracy and select “Save Model”. Name the model, “TF Model” and click “Save”.</p> 

Step	Action
	
4	<p>Deploy and Test the TensorFlow Model</p> <ol style="list-style-type: none"> We can now deploy the model using the same process as in Module 2. To do this, return to the “Assets” page of your project. Under the “Models” section, select the model you just saved. On the model overview page, select the “Deployments” tab. Click on “Add Deployment”.  <ol style="list-style-type: none"> On the deployment page, enter “TF Deployment” as the deployment name and click “Save”.

Step	Action
	<p>d. Your model will be deployed when the status changes to “DEPLOY_SUCCESS”</p> 
	<p>e. Click on the deployment name to view the details. Next, click on the “Implementation” tab. As with the previous lab exercise, we have generated a URL that developers can use in their applications to access the model, as well as example code.</p>  <p>f. Finally, we can test the model. Click on the “Test” tab. Paste the contents of the following file to the “Enter Input Data” field and click “Predict”.</p> <p style="text-align: center;">http://ibm.biz/BdZNWF</p> <p>g. In this case, the input data is the binary code for images of the numbers 4 and 5. Therefore, the output from this test should be as shown below.</p>

Step	Action
	<p>Overview Implementation Test</p> <p>Enter input data</p> <pre>{ "values": [5, 4] }</pre> <p>Predict</p> 

Conclusion

In this lab, we learned how Watson Studio can facilitate the development and deployment of Deep Learning models and applications in a highly collaborative and open environment. We welcome you continue to explore this other capabilities of Watson Studio.

Thank you for your time today!!

The IBM Wolfpack Team



