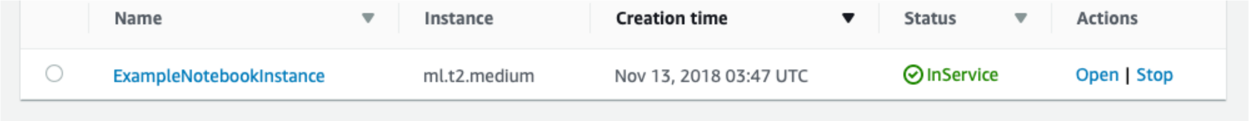
AWS Tutorial: AWS SageMaker

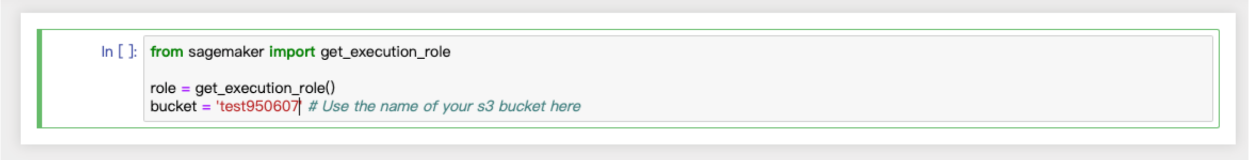
Train a Model with a Built-in Algorithm and Deploy It

Step 1: Create an Amazon SageMaker Notebook Instance



	Name	Instance	Creation time	Status	Actions
○	ExampleNotebookInstance	ml.t2.medium	Nov 13, 2018 03:47 UTC	InService	Open | Stop

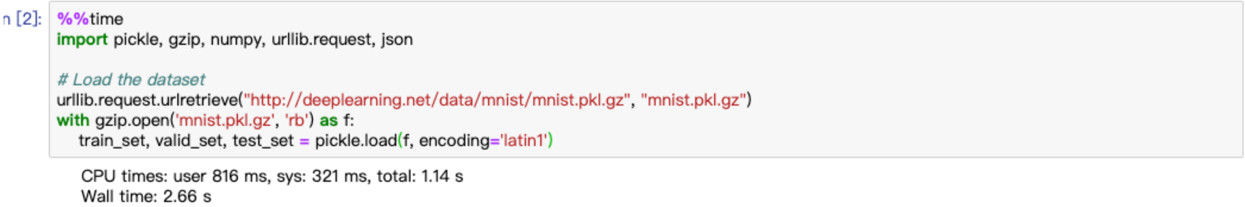
Step 2: Create a Jupyter Notebook and Initialize Variables



```
In [ ]: from sagemaker import get_execution_role

        role = get_execution_role()
        bucket = 'test950607' # Use the name of your s3 bucket here
```

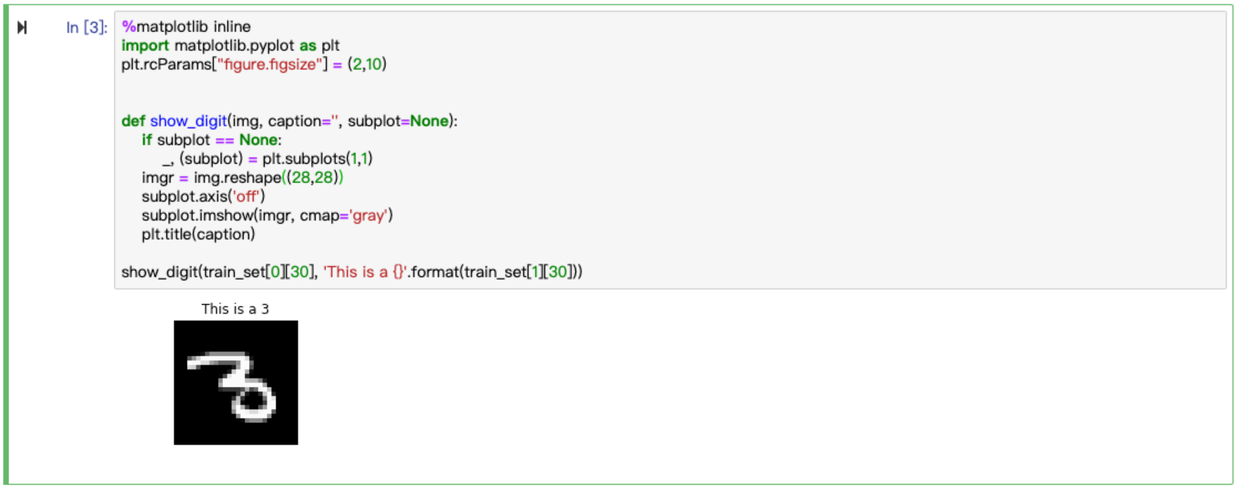
Step 3: Download, Explore, and Transform the Training Data

Download the MNIST Dataset



```
n [2]: %%time
       import pickle, gzip, numpy, urllib.request, json

       # Load the dataset
       urllib.request.urlretrieve("http://deeplearning.net/data/mnist/mnist.pkl.gz", "mnist.pkl.gz")
       with gzip.open('mnist.pkl.gz', 'rb') as f:
           train_set, valid_set, test_set = pickle.load(f, encoding='latin1')

         CPU times: user 816 ms, sys: 321 ms, total: 1.14 s
         Wall time: 2.66 s
```

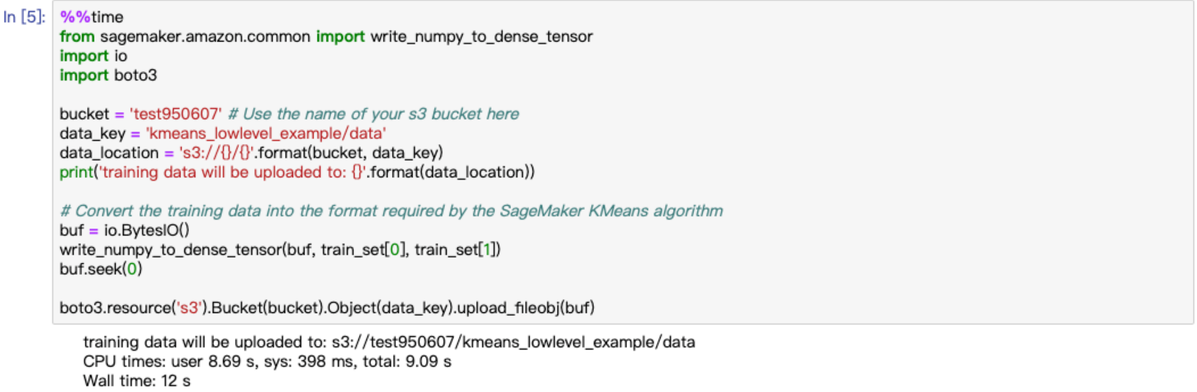
Explore the Training Dataset



```
In [3]: %matplotlib inline
        import matplotlib.pyplot as plt
        plt.rcParams["figure.figsize"] = (2,10)


        def show_digit(img, caption='', subplot=None):
            if subplot == None:
                _, (subplot) = plt.subplots(1,1)
            imgr = img.reshape((28,28))
            subplot.axis('off')
            subplot.imshow(imgr, cmap='gray')
            plt.title(caption)

        show_digit(train_set[0][30], 'This is a {}'.format(train_set[1][30]))
```

This is a 3

Transform the Training Dataset and Upload It to S3

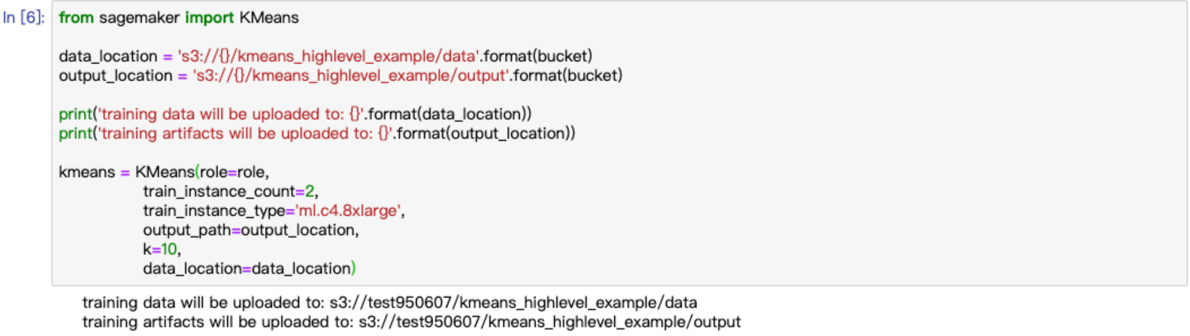


```
In [5]: %%time
        from sagemaker.amazon.common import write_numpy_to_dense_tensor
        import io
        import boto3

        bucket = 'test950607' # Use the name of your s3 bucket here
        data_key = 'kmeans_lowlevel_example/data'
        data_location = 's3://{}/{}'.format(bucket, data_key)
        print('training data will be uploaded to: {}'.format(data_location))

        # Convert the training data into the format required by the SageMaker KMeans algorithm
        buf = io.BytesIO()
        write_numpy_to_dense_tensor(buf, train_set[0], train_set[1])
        buf.seek(0)

        boto3.resource('s3').Bucket(bucket).Object(data_key).upload_fileobj(buf)
```

```
training data will be uploaded to: s3://test950607/kmeans_lowlevel_example/data
CPU times: user 8.69 s, sys: 398 ms, total: 9.09 s
Wall time: 12 s
```

Step 4: Train a Model

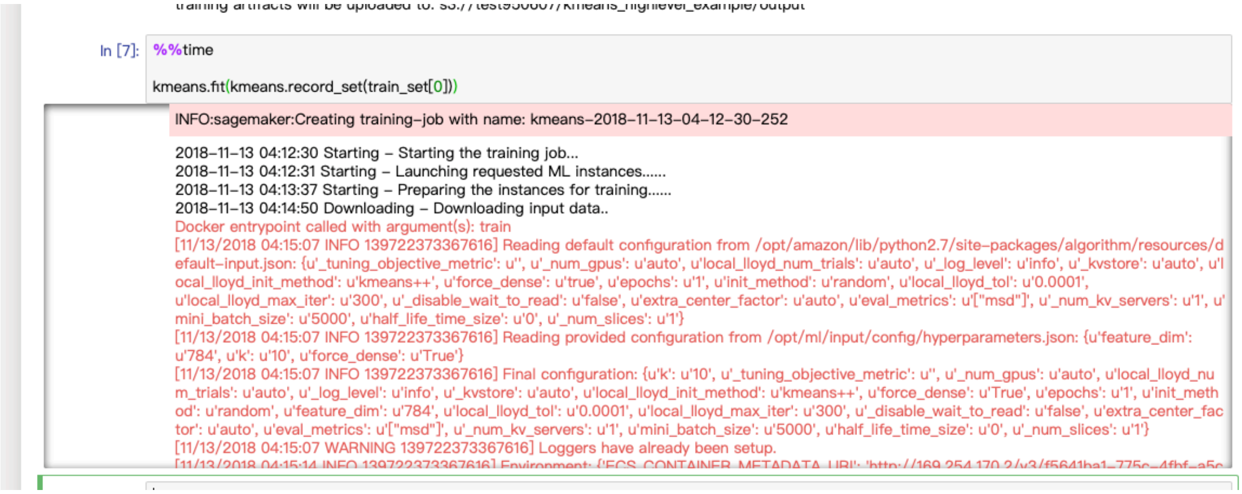
Create a Training Job



```
In [6]: from sagemaker import KMeans

        data_location = 's3://{}/kmeans_highlevel_example/data'.format(bucket)
        output_location = 's3://{}/kmeans_highlevel_example/output'.format(bucket)

        print('training data will be uploaded to: {}'.format(data_location))
        print('training artifacts will be uploaded to: {}'.format(output_location))

        kmeans = KMeans(role=role,
                        train_instance_count=2,
                        train_instance_type='ml.c4.8xlarge',
                        output_path=output_location,
                        k=10,
                        data_location=data_location)
```

```
training data will be uploaded to: s3://test950607/kmeans_highlevel_example/data
training artifacts will be uploaded to: s3://test950607/kmeans_highlevel_example/output
```

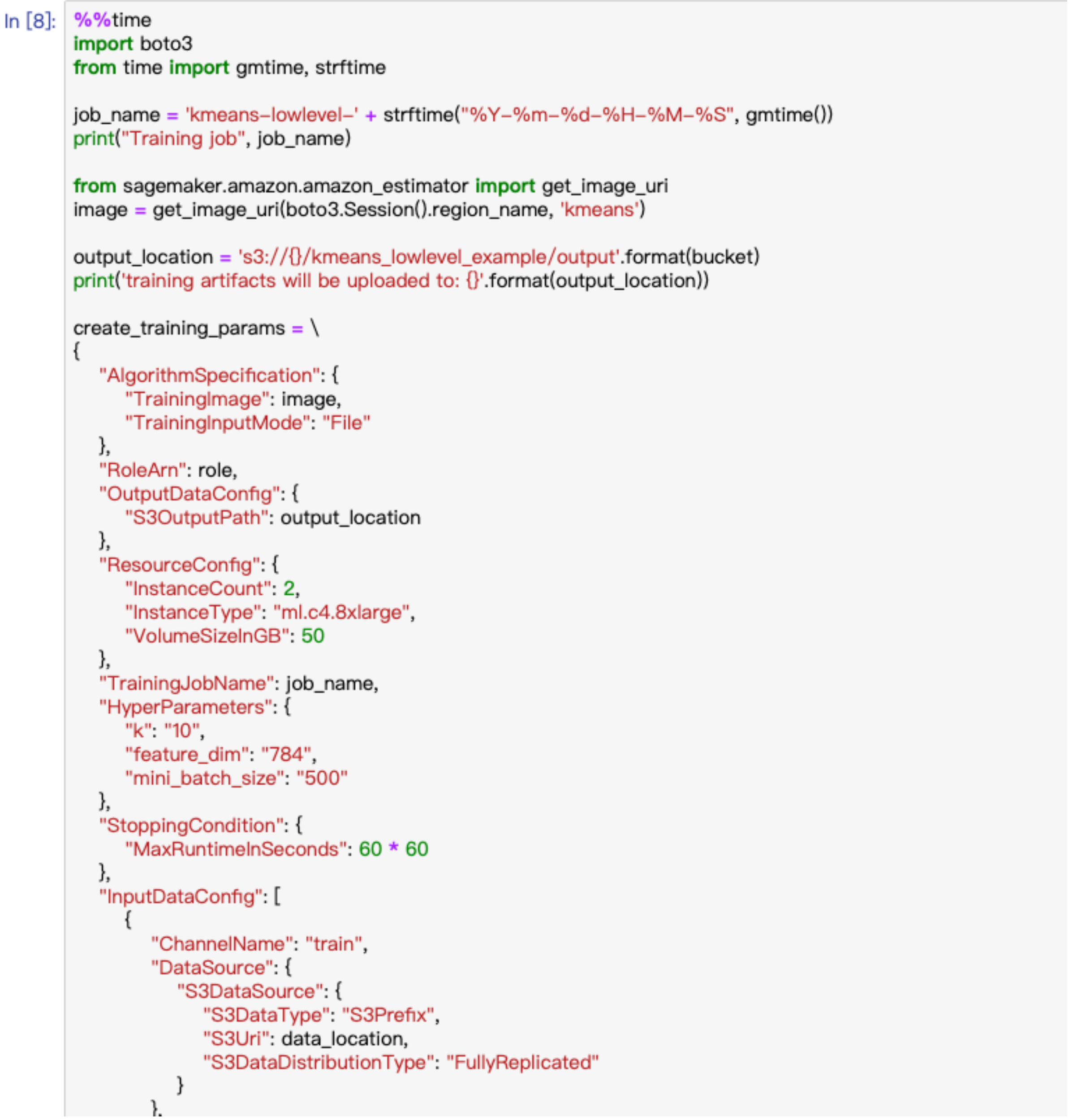
Start Train model



```
training artifacts will be uploaded to: s3://test950607/kmeans_highlevel_example/output

In [7]: %%time

        kmeans.fit(kmeans.record_set(train_set[0]))
```

```
INFO:sagemaker:Creating training-job with name: kmeans-2018-11-13-04-12-30-252

2018-11-13 04:12:30 Starting - Starting the training job...
2018-11-13 04:12:31 Starting - Launching requested ML instances......
2018-11-13 04:13:37 Starting - Preparing the instances for training......
2018-11-13 04:14:50 Downloading - Downloading input data..
Docker entrypoint called with argument(s): train
[11/13/2018 04:15:07 INFO 139722373367616] Reading default configuration from /opt/amazon/lib/python2.7/site-packages/algorithm/resources/d
efault-input.json: {u'_tuning_objective_metric': u'', u'_num_gpus': u'auto', u'local_lloyd_num_trials': u'auto', u'_log_level': u'info', u'_kvstore': u'auto', u'l
ocal_lloyd_init_method': u'kmeans++', u'force_dense': u'true', u'epochs': u'1', u'init_method': u'random', u'local_lloyd_tol': u'0.0001',
u'local_lloyd_max_iter': u'300', u'_disable_wait_to_read': u'false', u'extra_center_factor': u'auto', u'eval_metrics': u'["msd"]', u'_num_kv_servers': u'1', u'
mini_batch_size': u'5000', u'half_life_time_size': u'0', u'_num_slices': u'1'}
[11/13/2018 04:15:07 INFO 139722373367616] Reading provided configuration from /opt/ml/input/config/hyperparameters.json: {u'feature_dim':
u'784', u'k': u'10', u'force_dense': u'True'}
[11/13/2018 04:15:07 INFO 139722373367616] Final configuration: {u'k': u'10', u'_tuning_objective_metric': u'', u'_num_gpus': u'auto', u'local_lloyd_nu
m_trials': u'auto', u'_log_level': u'info', u'_kvstore': u'auto', u'local_lloyd_init_method': u'kmeans++', u'force_dense': u'True', u'epochs': u'1', u'init_meth
od': u'random', u'feature_dim': u'784', u'local_lloyd_tol': u'0.0001', u'local_lloyd_max_iter': u'300', u'_disable_wait_to_read': u'false', u'extra_center_fac
tor': u'auto', u'eval_metrics': u'["msd"]', u'_num_kv_servers': u'1', u'mini_batch_size': u'5000', u'half_life_time_size': u'0', u'_num_slices': u'1'}
[11/13/2018 04:15:07 WARNING 139722373367616] Loggers have already been setup.
[11/13/2018 04:15:14 INFO 139722373367616] Environment: {'ECS_CONTAINER_METADATA_URI': 'http://169.254.170.2/v3/f5641ba1-775c-4fbf-a5c
```

Use the SDK for Python



In [8]:

```
%%time
import boto3
from time import gmtime, strftime

job_name = 'kmeans-lowlevel-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print("Training job", job_name)

from sagemaker.amazon.amazon_estimator import get_image_uri
image = get_image_uri(boto3.Session().region_name, 'kmeans')

output_location = 's3://{}/kmeans_lowlevel_example/output'.format(bucket)
print('training artifacts will be uploaded to: {}'.format(output_location))

create_training_params = \
{
    "AlgorithmSpecification": {
        "TrainingImage": image,
        "TrainingInputMode": "File"
    },
    "RoleArn": role,
    "OutputDataConfig": {
        "S3OutputPath": output_location
    },
    "ResourceConfig": {
        "InstanceCount": 2,
        "InstanceType": "ml.c4.8xlarge",
        "VolumeSizeInGB": 50
    },
    "TrainingJobName": job_name,
    "HyperParameters": {
        "k": "10",
        "feature_dim": "784",
        "mini_batch_size": "500"
    },
    "StoppingCondition": {
        "MaxRuntimeInSeconds": 60 * 60
    },
    "InputDataConfig": [
        {
            "ChannelName": "train",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "S3Prefix",
                    "S3Uri": data_location,
                    "S3DataDistributionType": "FullyReplicated"
                }
            },
```

Step 5: Deploy the Model to Amazon SageMaker Hosting Services

```
In [9]: %%time

        kmeans_predictor = kmeans.deploy(initial_instance_count=1,
                                         instance_type='ml.m4.xlarge')
```

```
INFO:sagemaker:Creating model with name: kmeans-2018-11-13-04-46-33-734
INFO:sagemaker:Creating endpoint with name kmeans-2018-11-13-04-12-30-252
```

```
---------------------------------------------------------------!CPU times: user 305 ms, sys: 36.6 ms, total: 341 ms
Wall time: 5min 19s
```