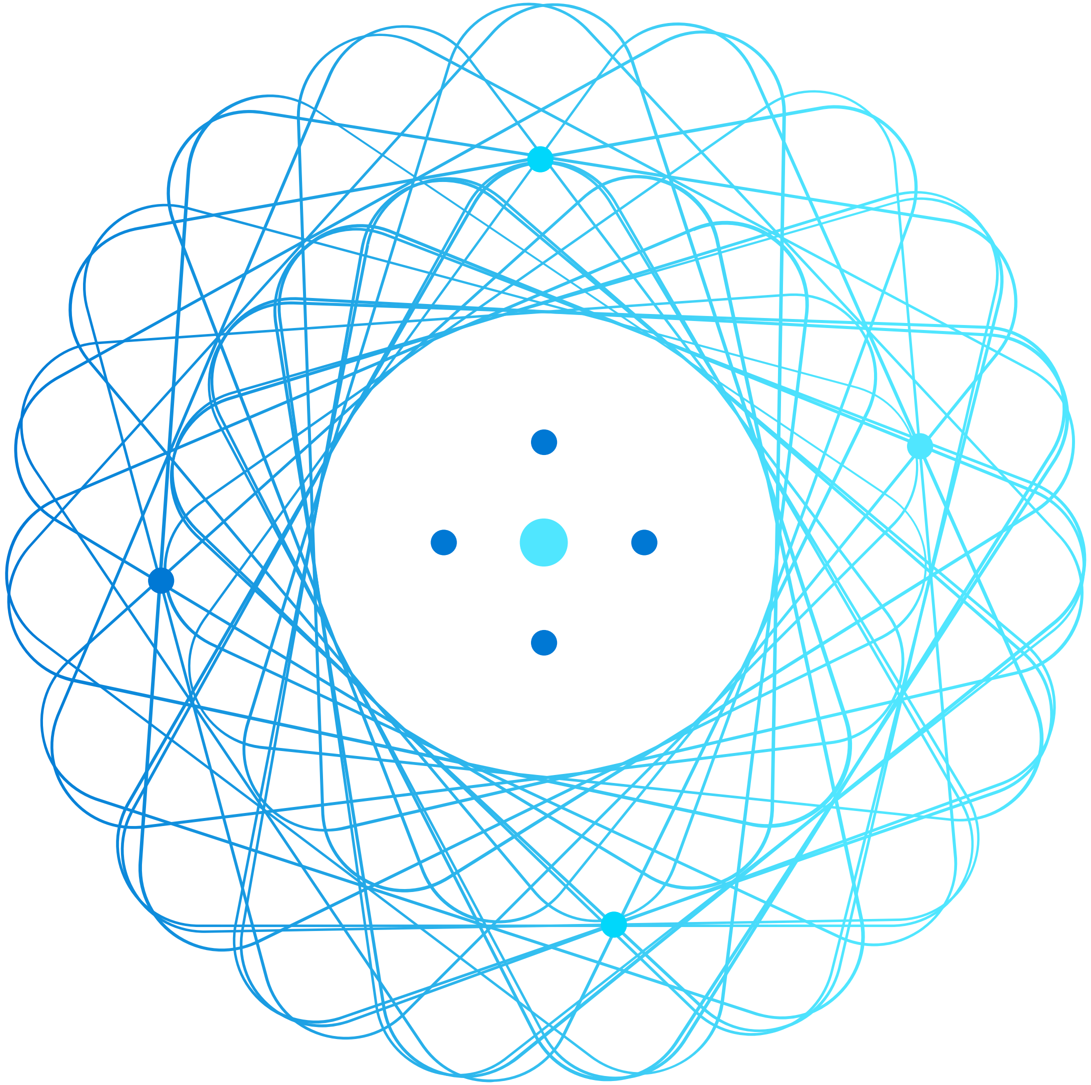
Microsoft Azure

Module 4: Working with Data

Mohammed Arif



Agenda

- Working with Datastores
- Working with Datasets

Working with Datastores

What are Datastores?

Abstractions for cloud data sources

- Azure Storage
- Azure Data Lake
- Azure SQL Database
- Azure Databricks File System
- Others

Built-in Datastores

- workspaceblobstore (default)
- workspacefilestore
- azureml_globaldatasets*

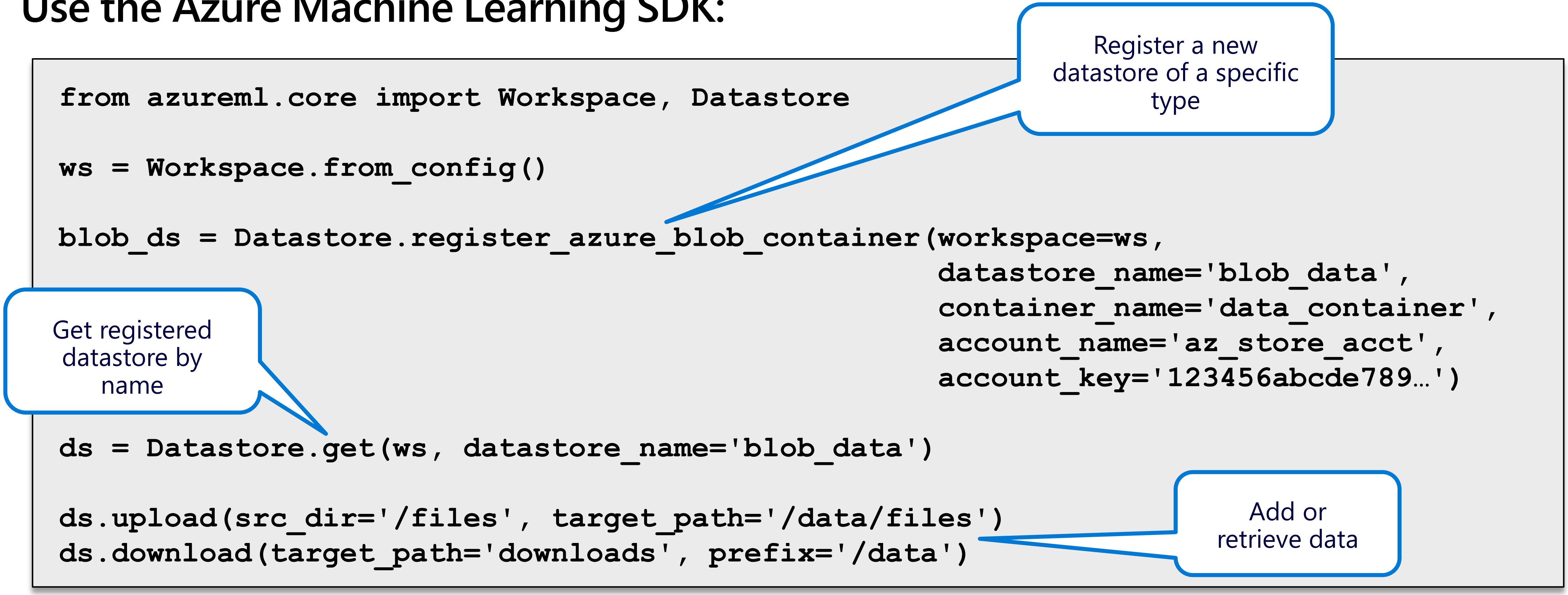
* Added when open datasets are used

Working with Datastores

Add a datastore in Azure Machine Learning studio

or

Use the Azure Machine Learning SDK:



```
from azureml.core import Workspace, Datastore

ws = Workspace.from_config()

blob_ds = Datastore.register_azure_blob_container(workspace=ws,
                                                  datastore_name='blob_data',
                                                  container_name='data_container',
                                                  account_name='az_store_acct',
                                                  account_key='123456abcde789…')

ds = Datastore.get(ws, datastore_name='blob_data')

ds.upload(src_dir='/files', target_path='/data/files')
ds.download(target_path='downloads', prefix='/data')
```

Register a new datastore of a specific type

Get registered datastore by name

Add or retrieve data

Considerations for Datastores

✓ Configure blob storage performance type and replication for your needs

✓ *Parquet* file format generally performs better than *CSV*

✓ You can manage the default datastore using the SDK

```
ws.set_default_datastore(my_datastore)
...
ds = ws.get_default_datastore()
```

Working with Datasets

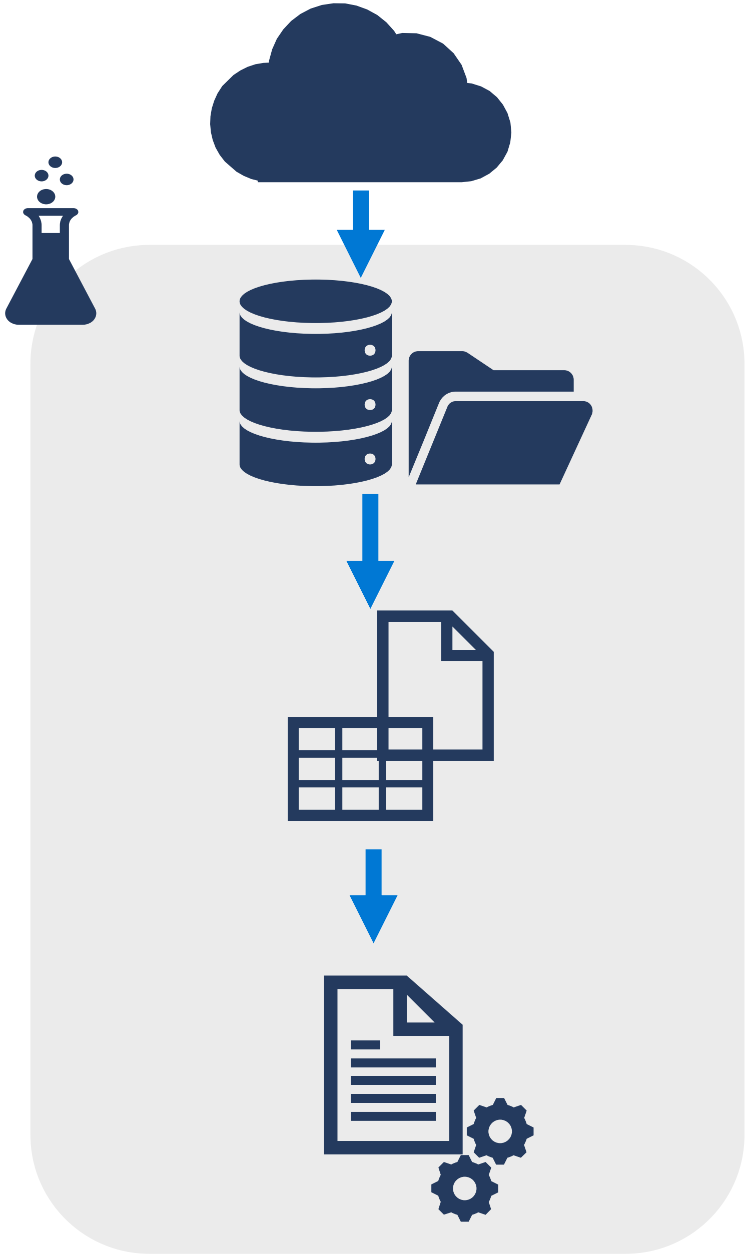
What are Datasets?

Versioned data objects for experiments

Usually based on datastore contents

Two types:

- *Tabular* datasets: Easy conversion to Pandas dataframe format for structured data files
- *File* datasets: Collection of file references for structured or unstructured data



Creating and Registering Datasets

Add a dataset in Azure Machine Learning studio

or

Use the Dataset object in the SDK

```
from azureml.core import Dataset

csv_paths = [(blob_ds, 'data/files/current_data.csv'),(blob_ds, 'data/files/archive/*.csv')]
tab_ds = Dataset.Tabular.from_delimited_files(path=csv_paths)
tab_ds = tab_ds.register(workspace=ws, name='csv_table')

csv_ds = ws.datasets['csv_table']
```

Create tabular dataset

Register in workspace

Retrieve (in this case from workspace **datasets** collection)

```
from azureml.core import Dataset

file_ds = Dataset.File.from_files(path=(blob_ds, 'data/files/images/*.jpg'))
file_ds = file_ds.register(workspace=ws, name='img_files')

img_ds = Dataset.get_by_name(ws, 'img_files')
```

Create file dataset

Register in workspace

Retrieve (in this case from **Dataset** class by name)

Working with Tabular Datasets

Pass a dataset as a script argument

ScriptRunConfig:

```
env = Environment('my_env')
packages = CondaDependencies.create(
     pip_packages=[...,'azureml-dataprep[pandas]'])
env.python.conda_dependencies = packages

sc = ScriptRunConfig(source_directory='my_dir',
                     script='script.py',
                     arguments=['--ds', tab_ds],
                     environment=env)
```

Required to work with datasets in script

Pass dataset object as script argument

Script:

```
from azureml.core import Run, Dataset

parser.add_argument('--ds', type=str, dest='ds_id')
args = parser.parse_args()

run = Run.get_context()
ws = run.experiment.workspace
dataset = Dataset.get_by_id(ws, id=args.ds_id)
data = dataset.to_pandas_dataframe()
```

Argument contains dataset ID

Get dataset by ID

Convert to dataframe

or

Pass a dataset as a named input

ScriptRunConfig:

```
env = Environment('my_env')
packages = CondaDependencies.create(
     pip_packages=[...,'azureml-dataprep[pandas]'])
env.python.conda_dependencies = packages

sc = ScriptRunConfig(source_directory='my_dir',
        script='script.py',
        arguments=['--ds', tab_ds.as_named_input('my_ds')],
        environment=env)
```

Required to work with datasets in script

Pass dataset as named input

Script:

```
from azureml.core import Run

parser.add_argument('--ds', type=str, dest='ds_id')
args = parser.parse_args()

run = Run.get_context()
dataset = run.input_datasets['my_ds']
data = dataset.to_pandas_dataframe()
```

Argument still required!

Retrieve named dataset from input_datasets

Convert to dataframe

Working with File Datasets

Pass a dataset as a script argument

ScriptRunConfig:

Required to work with datasets in script

```
env = Environment('my_env')
packages = CondaDependencies.create(
     pip_packages=[...,'azureml-dataprep[pandas]'])
env.python.conda_dependencies = packages

sc = ScriptRunConfig(source_directory='my_dir',
            script='script.py',
            arguments=['--ds', file_ds.as_download()],
            environment=env)
```

Pass dataset object as *download* or *mount*

Script:

Argument contains data reference

```
from azureml.core import Run
import glob

parser.add_argument('--ds', type=str, dest='ds_ref')
args = parser.parse_args()
run = Run.get_context()

imgs = glob.glob(ds_ref + "/*.jpg")
```

Get file paths from data reference

or

Pass a dataset as a named input

ScriptRunConfig:

```
env = Environment('my_env')
packages = CondaDependencies.create(
     pip_packages=[...,'azureml-dataprep[pandas]'])
env.python.conda_dependencies = packages

sc = ScriptRunConfig(source_directory='my_dir',
        script='script.py',
        arguments=['--ds',
            file_ds.as_named_input('my_ds').as_download()],
        environment=env)
```

Pass dataset as named input

Script:

Argument still required!

```
from azureml.core import Run
import glob

parser.add_argument('--ds', type=str, dest='ds_ref')
args = parser.parse_args()
run = Run.get_context()

dataset = run.input_datasets['my_ds']
imgs= glob.glob(dataset + "/*.jpg")
```

Retrieve named dataset from input_datasets

Get file paths from data reference

Dataset Versioning

Create a new version of an existing dataset

```
# add .png files to dataset definition
img_paths = [(blob_ds, 'data/files/images/*.jpg'),(blob_ds, 'data/files/images/*.png')]
file_ds = Dataset.File.from_files(path=img_paths)
file_ds = file_ds.register(workspace=ws, name='img_files', create_new_version=True)
```

Auto-increments version if a dataset of the same name exists

Specify a version to retrieve

```
ds = Dataset.get_by_name(workspace=ws, name='img_files', version=2)
```

Version number

Lab: Work with Data

1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Work with data** exercise

Knowledge check

You have a reference to a Workspace named *ws*.
Which code retrieves the default datastore for the workspace?

☐ `default_ds = Datastore.get(ws, 'default')`

☐ `default_ds = ws.Datastores[0]`

☑ `default_ds = ws.get_default_datastore()`

A datastore contains a CSV file of structured data that you want to use as a Pandas dataframe.
Which kind of dataset should you create to make it easy to do this?

☐ A file dataset

☑ A tabular dataset

You want a script to stream data directly from a file dataset. Which mode should you use?

☑ `as_mount()`

☐ `as_download()`

☐ `as_upload()`

References

Microsoft Learn: Work with Data in Azure Machine Learning
https://docs.microsoft.com/learn/modules/work-with-data-in-aml/

Azure Machine Learning data documentation
https://docs.microsoft.com/azure/machine-learning/concept-data

Microsoft Azure