Data engineering on Azure Databricks - workshop

Anagha Khanolkar
Azure Cloud Solution Architects – Data and AI

About the lab

Learn how to…

- Provision Azure resources – blob storage, Azure SQL DB, Databricks
- Learn how to copy a public dataset into Azure
- Learn how to build a data engineering pipeline in Azure Databricks
- Learn how to automate a report jobset that integrates reports generated in Databricks into an RDBMS

The datasets

NYC taxi data

Transactional data

Yellow taxi trips (675 million) | 2009 - 2017
Green taxi trips (59 million) | 2013 - 2017
CSV
Schema varies between taxi types
Schema varies across years
~30GB raw

Reference data

Trip months
Payment type
Rate code
Taxi zone
Trip type
Vendor

The Azure services

Utilities:
Azure Cloud Shell – for provisioning storage and copying data
Azure SQL Database Query Explorer – cloud IDE for Azure SQL DB

The lab modules

- **Data copy** – Copy datasets using Azure cloud shell into your storage account
- **Data load** – Read CSV from blob storage, save raw data as partitioned parquet
- **Transform** – Apply data transformations to homogenize schemas across taxi type over years
- **Create materialized views** – Homogenize and merge curated yellow and green taxi data into a single table
- **Run reports & visualize** – Run some reports
  Create some charts
- **Batch job automation** – Create and schedule and automated Databricks batch job, write to RDBMS

Data copy

With Azure Cloud Shell - Bash

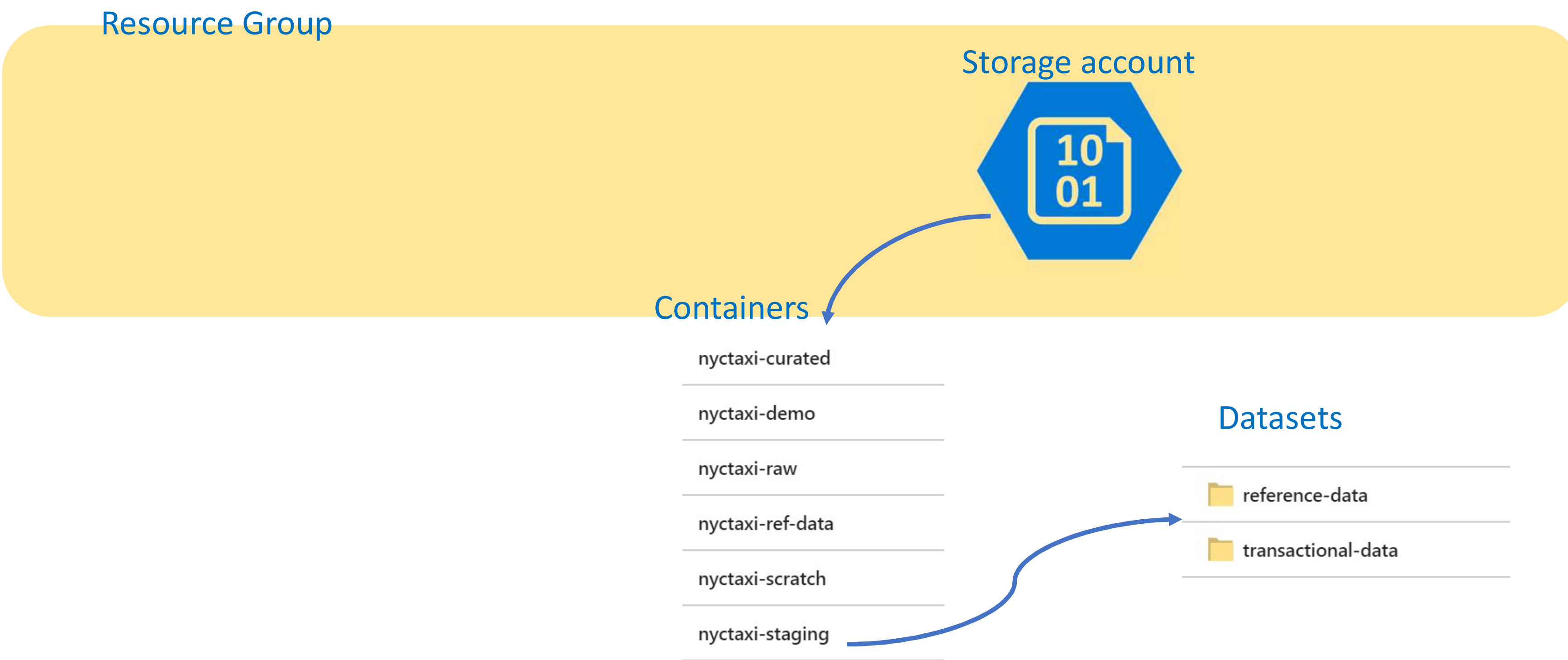
Data load TODOs

- ❑ Launch cloud shell
- ❑ Create a resource group in US East 2
- ❑ Create a storage account
- ❑ Create blob storage containers
- ❑ Copy data for the workshop (details in doc)
- ❑ Upload a subset of the workshop data into staging directory

Instructions:
https://aka.ms/DEW-Setup
Copying the entire dataset (30 GB) look 2 hours USE2-USE2
For the lab- we will use a subset

By now…

You should have the following set up in your subscription:



Azure services

Provisioning, configuration

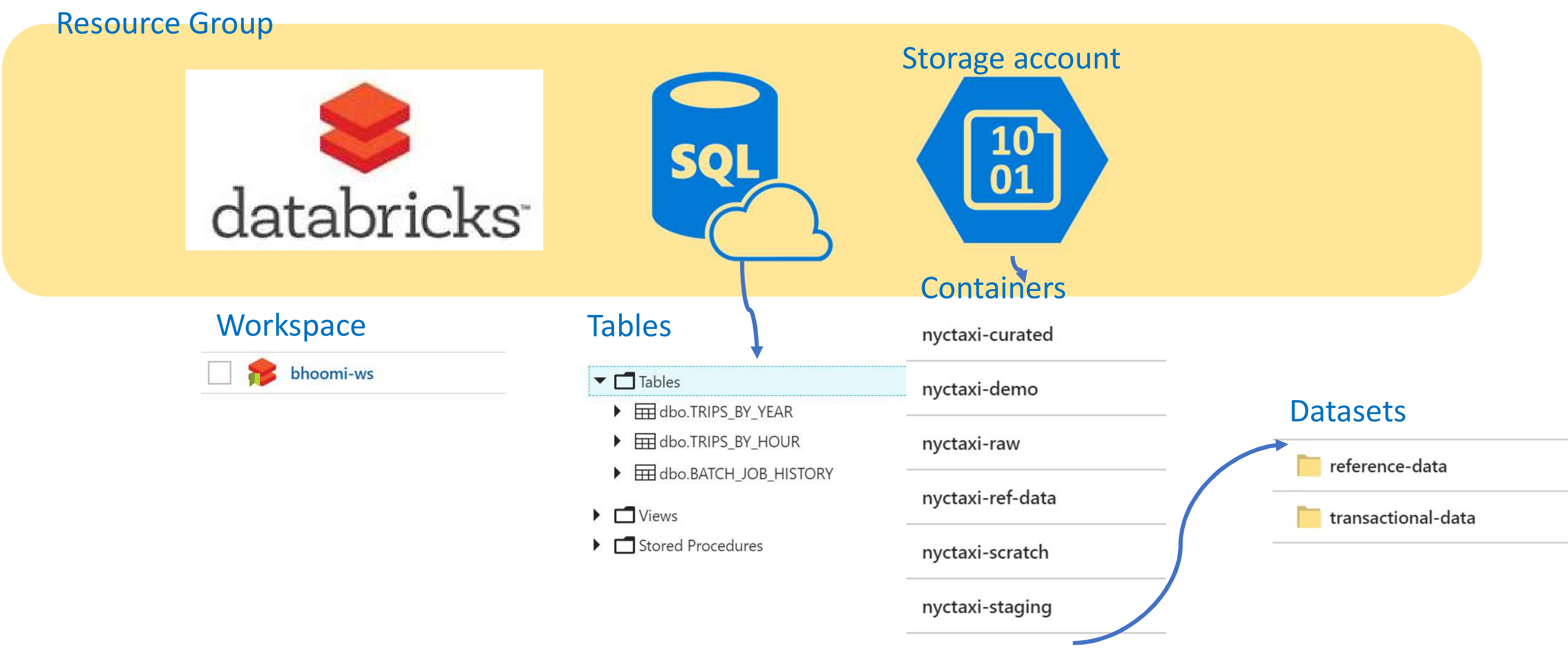
Provisioning TODOs

- ❑ Provision Azure SQL Database in the resource group (100 DTU)
- ❑ Create tables in the Azure SQL Database – for batch job history & reports
- ❑ Provision Azure Databricks workspace in the resource group
- ❑ Launch workspace

Instructions:
https://aka.ms/DEW-Setup

By now…

You should have the following set up in your subscription:



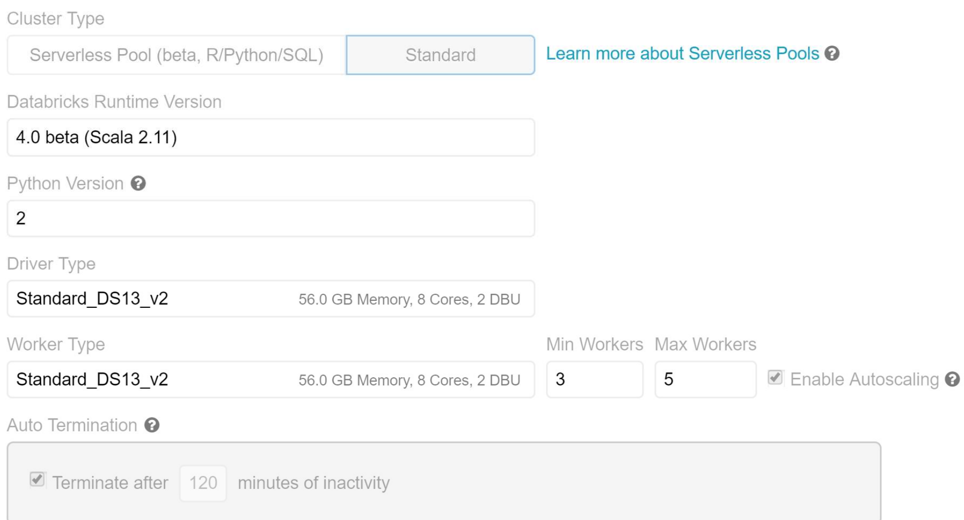
Azure Databricks

Provisioning, configuration

Provision cluster

Azure Databricks – setup

Initialize workspace and create a cluster with the following specifications:

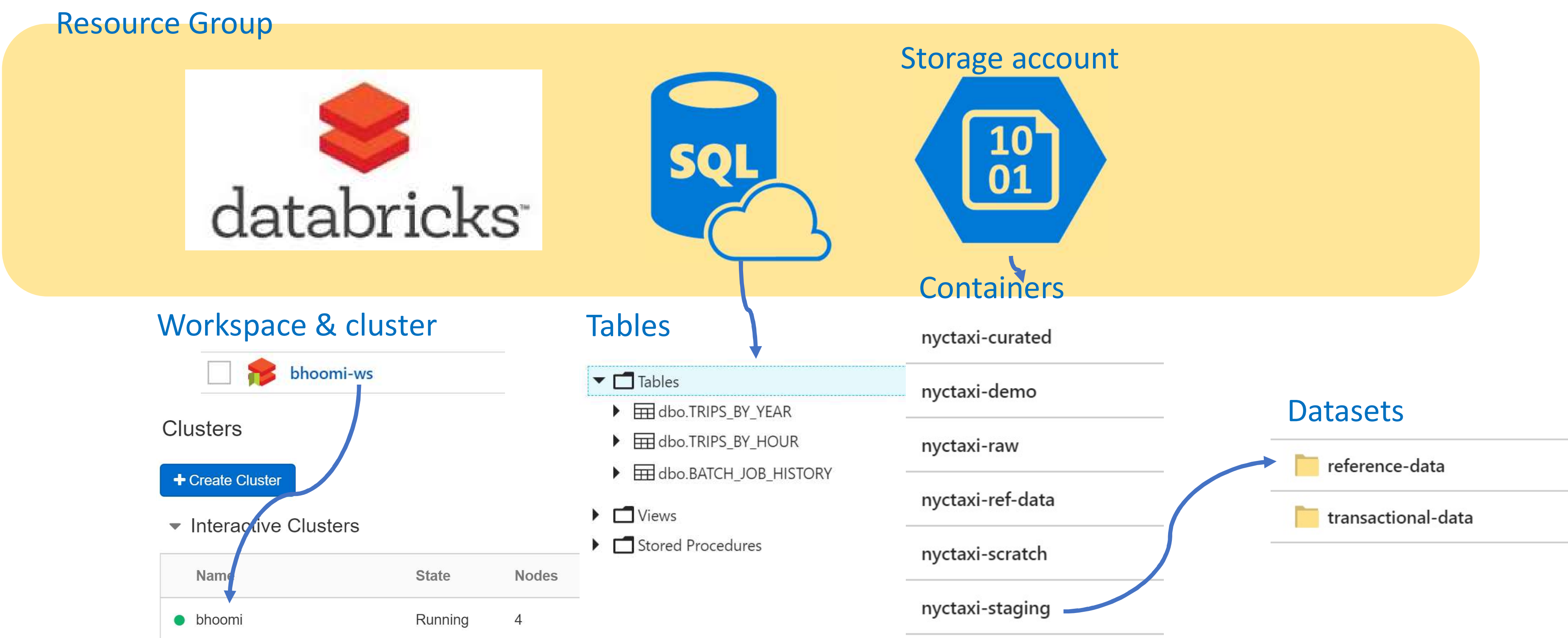


In the Spark config section, enter your storage account credentials-

spark.hadoop.fs.azure.account.key.**<storageaAccount>.**blob.core.windows.net **<key>**

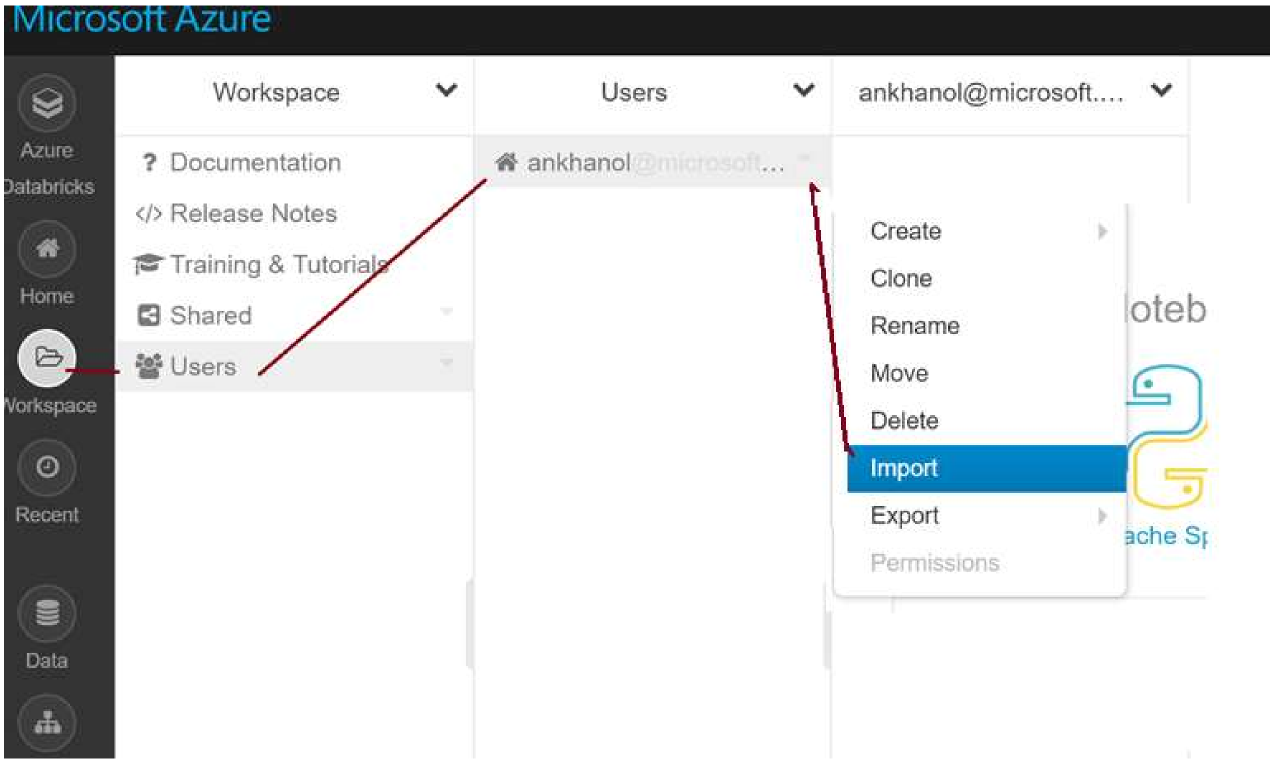
By now…

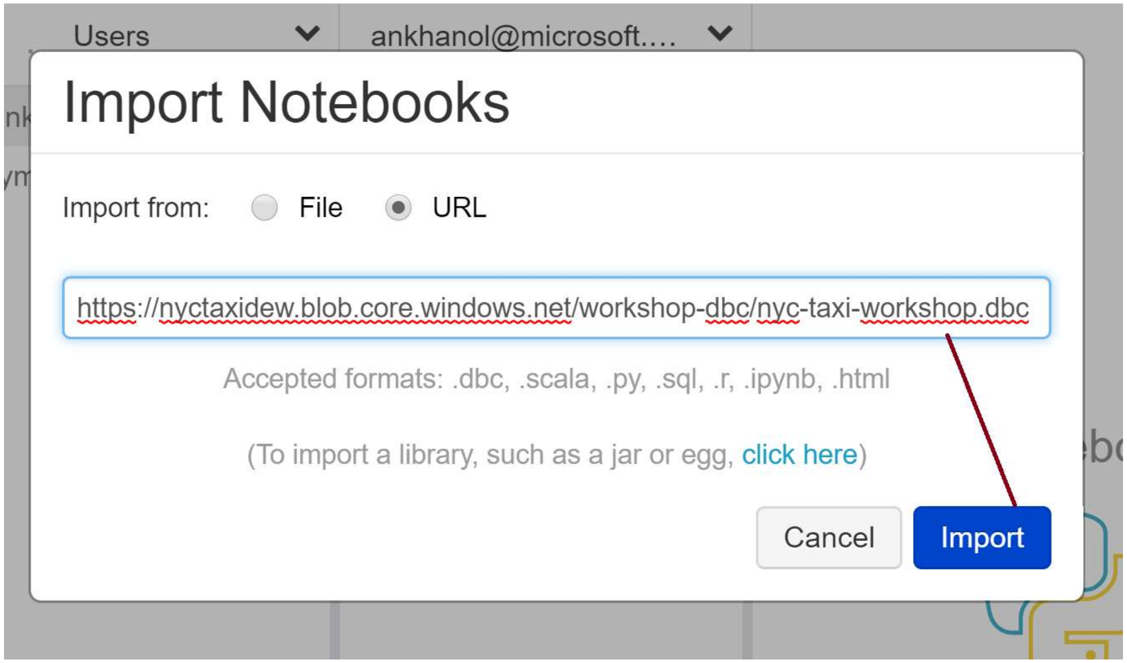
You should have the following set up in your subscription:



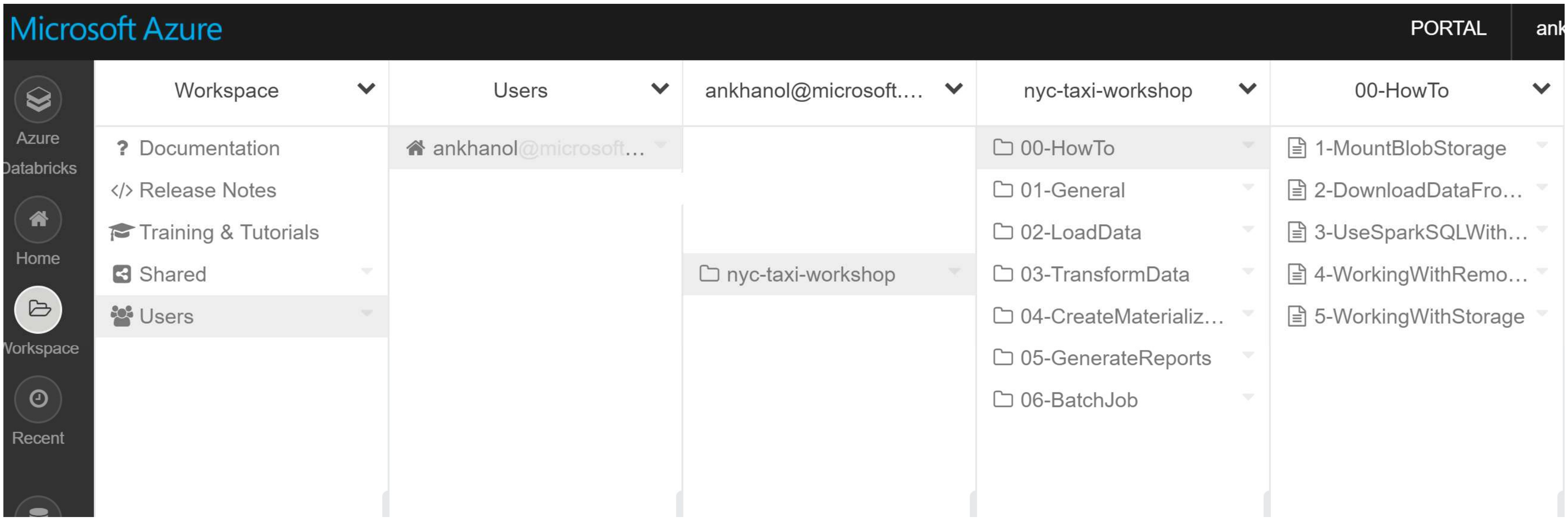
Azure Databricks
Import workshop notebooks

Import workshop notebooks



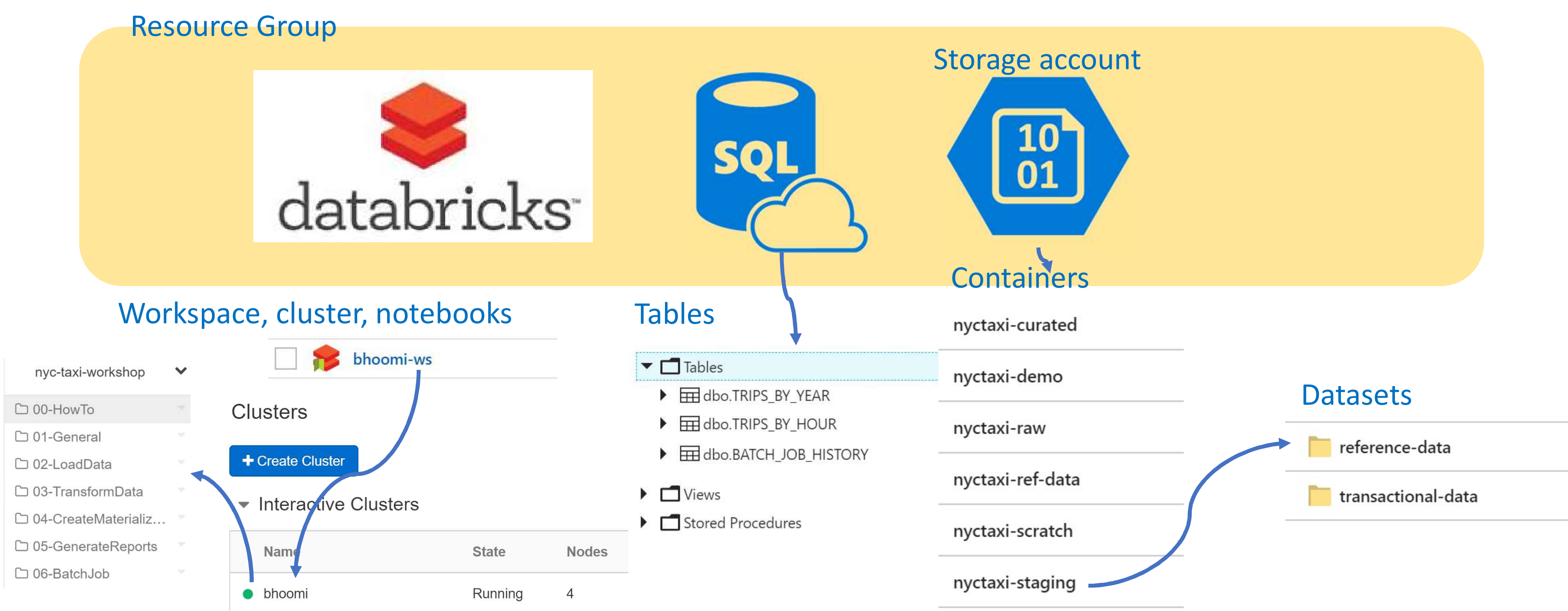


https://nyctaxidew.blob.core.windows.net/workshop-dbc/nyc-taxi-workshop.dbc


Microsoft Azure
PORTAL
Azure Databricks
Home
Workspace
Recent
Workspace
Users
ankhanol@microsoft....
nyc-taxi-workshop
00-HowTo
Documentation
Release Notes
Training & Tutorials
Shared
Users
ankhanol@microsoft...
nyc-taxi-workshop
00-HowTo
01-General
02-LoadData
03-TransformData
04-CreateMaterializ...
05-GenerateReports
06-BatchJob
1-MountBlobStorage
2-DownloadDataFro...
3-UseSparkSQLWith...
4-WorkingWithRemo...
5-WorkingWithStorage

You are done with setup!

You should have the following set up in your subscription:



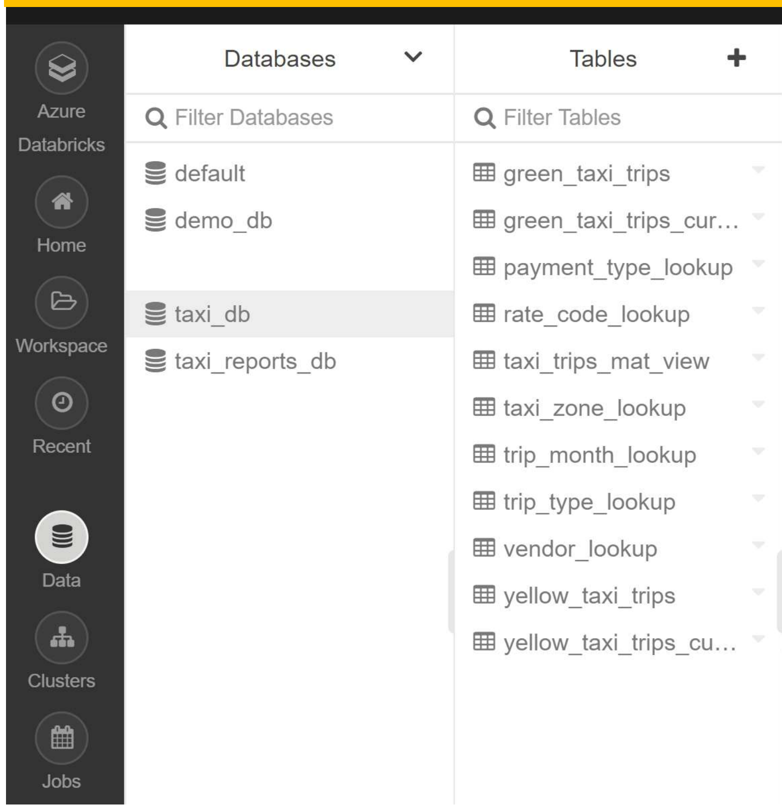
Password for demo database

The password for the RDBMS demodbserver is:
d@t@br1ck$

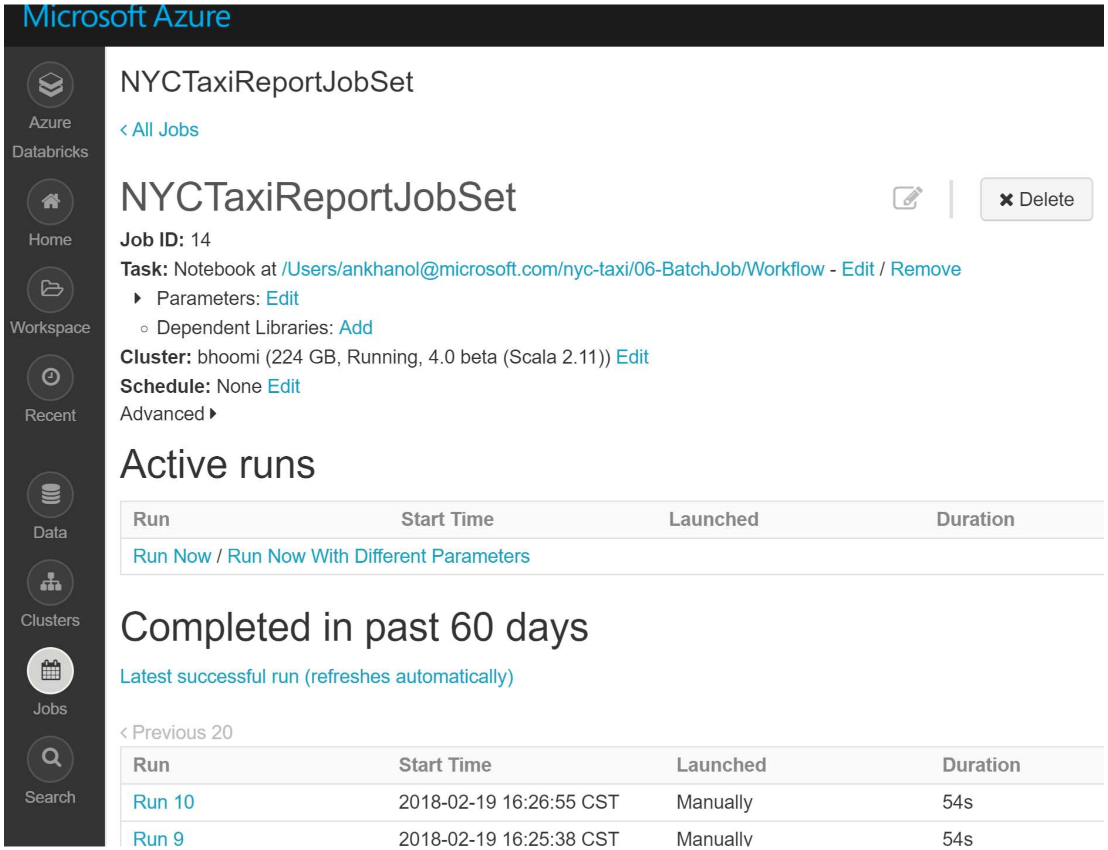
Azure Databricks

Preview of what's you'll be creating

Query-able Hive tables



On-demand/scheduled batch jobset



Hands on lab - Module 1

<u>HOW TO</u>

work with Hive, with DBFS, with remote databases

1.1. Mount blob storage

Why?

What’s easier?

```
Option 1: wasbs URI
wasbs://storageContainer@storageAccount. blob.core.windows.net/<myDirectory>

Option 2: Mount point
/mnt/data/<myDirectory>
```

Mounting blob storage is an option with Azure Databricks 3.5 and above, and a general best practice.

It simplifies and secures accessing storage

1.1. Mount blob storage
How to

Lets review the notebook –
nyc-taxi-workshop/00-HowTo/00-MountBlobStorage.scala

In this notebook – we will mount a storage container, and access it using the mountpoint. We will also learn to refresh mountpoints and to unmount.

1.2. Downloading the original Yellow Taxi data

How to

<u>This is informational only</u>

Lets review the notebook –
nyc-taxi-workshop/00-HowTo/2-DownloadDataFromInternet.scala

1.3. Working with Hive & SparkSQL

How to

Lets review the notebook –
nyc-taxi-workshop/00-HowTo/3-UseSparkSQLWithHive.scala

In this notebook – we will create a text file locally (bash), load it to DBFS, create a hive table on it, and run queries using SQL

1.4. Working with remote databases/JDBC
How to

Lets review the notebook –
nyc-taxi-workshop/00-HowTo/4-WorkingWithRemoteDatabases.scala

In this notebook – we will query a remote Azure SQL database like it were a table created in Azure Databricks

1.5. Working with storage
How to

Lets review the notebook –
nyc-taxi-workshop/00-HowTo/5-WorkingWithStorage.scala

In this notebook, we will work with DBFS file system commands

Recap of module

We learned how to -

1) Mount blob storage
2) Work with Hive and SparkSQL
3) Work with a remote SQL database
4) Work with Databricks File System (DBFS)

Hands on lab - Module 2

<u>General setup</u>

Mount blob storage

Create database

Common functions

2.1. Mount blob storage
Workshop - setup

Lets review this notebook-
nyc-taxi-workshop/01-General/1-MountBlobStorage.scala

In this notebook, we create a function to mount blob storage and call it to mount several containers

2.2. Create database objects

Workshop - setup

Lets review this notebook-
nyc-taxi-workshop/01-General/2-CreateDatabaseObjects.scala

In this notebook, -
(1) We create a database taxi_db
(2) We create a JDBC Hive table against the Azure SQL database table you set up during provisioning – batch_job_history

2.3. Define common functions

Focus - reusability

Lets review this notebook-
nyc-taxi-workshop/01-General/3-CommonFunctions.scala

In this notebook, -
We create a set of commonly used functions in a notebook for future use in the workshop

Recap of module

We completed-

1) Mounting blob storage
2) Creating a Hive database
3) Creating a Hive table definition for a remote Azure SQL database
4) Creating a set of commonly used functions in a notebook for future use in the workshop

Hands on lab - Module 3

Load dataset

3.1. Load reference data

Lets review this notebook-
nyc-taxi-workshop/02-LoadData/1-LoadReferenceData.scala

In this notebook, -
(1) We load all the reference datasets – read from stagingDir, save as parquet to refDataDir
(2) Create Hive tables on top of them
(3) Compute statistics

Trip months
Payment type
Rate code
Taxi zone
Trip type
Vendor

3.2. Load transactional data – Yellow Taxi

Lets review this notebook-
nyc-taxi-workshop/02-LoadData/2-LoadData-YellowTaxi.scala

In this notebook, -
(1) We load yellow taxi data over several years– read from stagingDir, save as parquet to refDataDir
(2) Create Hive tables on top of them
(3) Compute statistics

To note: the schema of the dataset differs over years. We homogenize the schema in this notebook

3.3. Load transactional data – Green Taxi

Lets review this notebook-
nyc-taxi-workshop/02-LoadData/3-LoadData-GreenTaxi.scala

In this notebook, -
(1) We load green taxi data over several years– read from stagingDir, save as parquet to refDataDir
(2) Create Hive tables on top of them
(3) Compute statistics

To note: the schema of the dataset differs over years. We homogenize the schema in this notebook

Recap of module

We completed-

1) Loading reference data and saving as parquet, with hive tables & stats
2) Loading yellow taxi and green taxi data, saving as parquet, with hive tables & stats

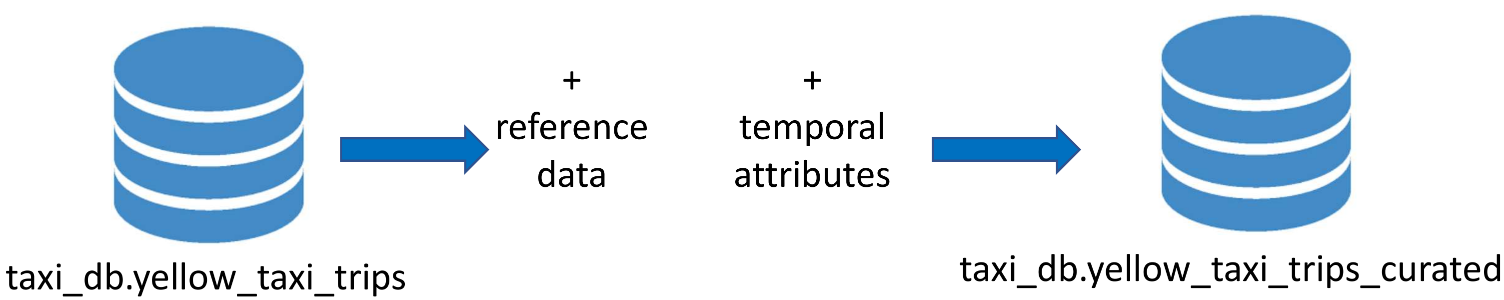
Hands on lab - Module 4

Transform data

4.1. Transform yellow taxi data

Lets review this notebook-
nyc-taxi-workshop/03-TransformData/1-TransformData-YellowTaxi.scala

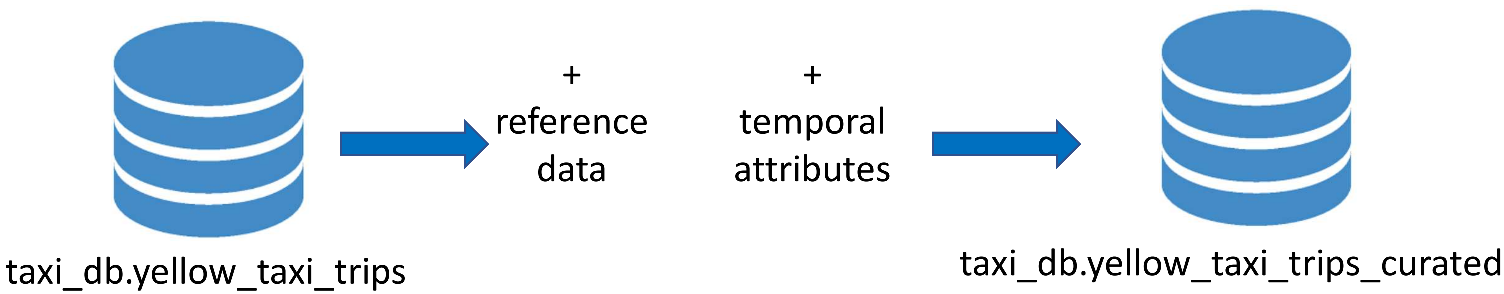
In this notebook, -
(1) We execute the notebook with common functions
(2) Denormalize - join yellow taxi with all reference datasets and persist to DBFS
(3) Define Hive table for the transformed dataset (curated), create partitions (year, month)
(4) Compute statistics



4.2. Transform green taxi data

Lets review this notebook-
nyc-taxi-workshop/03-TransformData/2-TransformData-GreenTaxi.scala

In this notebook, -
(1) We execute the notebook with common functions
(2) Denormalize - join yellow taxi with all reference datasets and persist to DBFS
(3) Define Hive table for the transformed dataset (curated), create partitions (year, month)
(4) Compute statistics



Recap of module

We completed-

Denormalizing yellow and green trip taxi data – joining with reference data, and saving as parquet, and ran hive tables & stats

Hands on lab - Module 5

Create materialized view

5.1. Create materialized view

Lets review this notebook-
nyc-taxi-workshop/04-CreateMaterializedView/1-CreateMaterializedViews.scala

In this notebook, -
(1) We execute the notebook with common functions
(2) We add columns to yellow taxi data to match green taxi data
(3) Do a union all on yellow and green taxi curated datasets and persist to DBFS
(4) Define Hive table for the transformed dataset (curated), create partitions (taxi type, year, month)
(5) Compute statistics

Hands on lab - Module 6

Generate a report with visualization

6.1. Generate report

Lets review this notebook-
nyc-taxi-workshop/05-GenerateReports/Report-1.scala

In this notebook, we will run several reports and visualize-

1. Trip count by taxi type
2. Revenue including tips by taxi type
3. Revenue share by taxi type
4. Trip count trend between 2013 and 2016
5. Trip count trend by month, by taxi type, for 2016
6. Average trip distance by taxi type

6.1. Generate report

7. Average trip amount by taxi type
8. Trips with no tip, by taxi type
9. Trips with no charge, by taxi type
10. Trips by payment type
11. Trip trend by pickup hour for yellow taxi in 2016
12. Top 3 yellow taxi pickup-dropoff zones for 2016

Hands on lab - Module 7

Batch job automation

7.1. Global vars and functions

Lets review the notebook-
nyc-taxi-workshop/06-BatchJob/GlobalVarsAndMethods.scala

In this notebook-

(1) We define JDBC credentials
(2) We create a function to generate a batch_ID by querying the batch_job_history table in Azure SQL database

7.2. Report 1

Lets review the notebook-
nyc-taxi-workshop/06-BatchJob/Report-1.scala

In this notebook-

(1) We create a dataframe – a simple report – trips by year
(2) Persist the data to Azure SQL database table trips_by_year

7.3. Report 2

Lets review the notebook-
nyc-taxi-workshop/06-BatchJob/Report-2.scala

In this notebook-

(1) We create a dataframe – a simple report – trips by hour
(2) Persist the data to Azure SQL database table trips_by_hour

7.4. Workflow

Lets review the notebook-
nyc-taxi-workshop/06-BatchJob/Workflow.scala

This is a notebook workflow spec and in this we –

(1) Execute the GlobalVarsAndFunctions notebook

(2) We generate batch ID by calling a method from #1

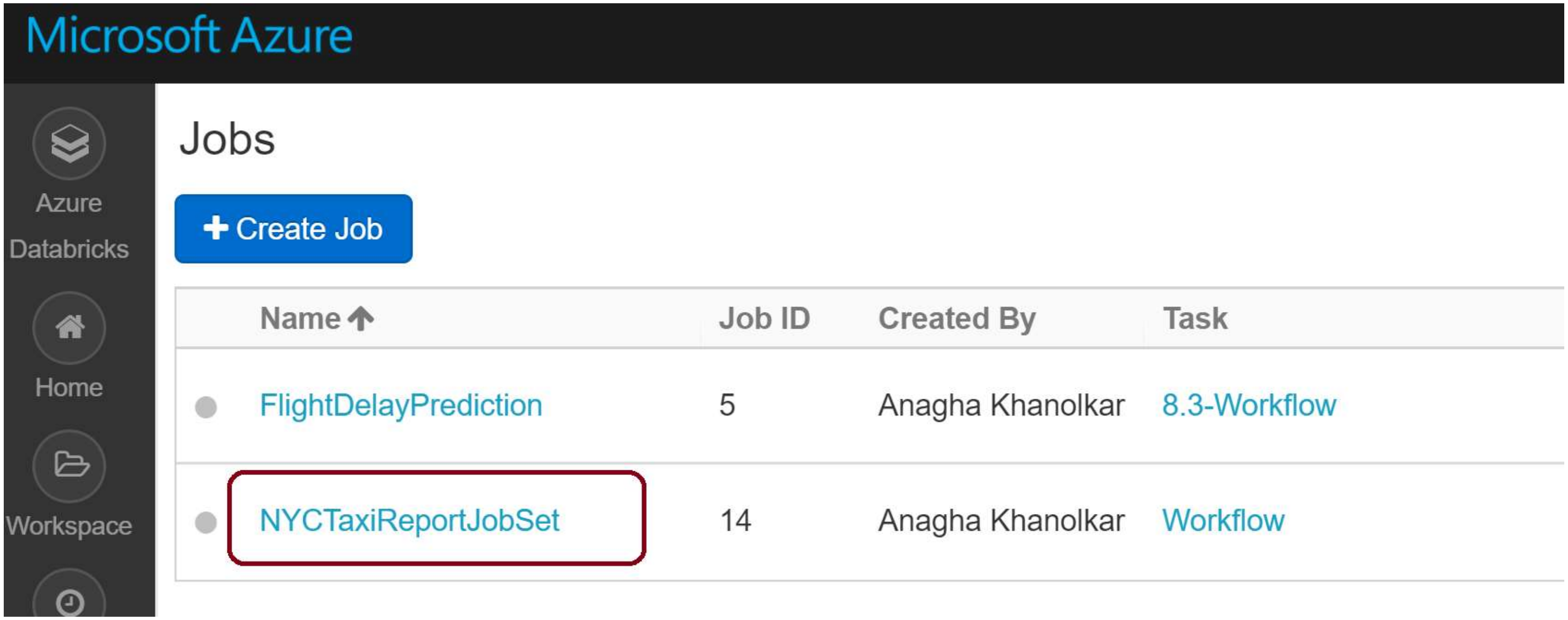
(3) We insert start time into batch_job_history, execute notebook Report-1, then insert completion time into the RDBMS table

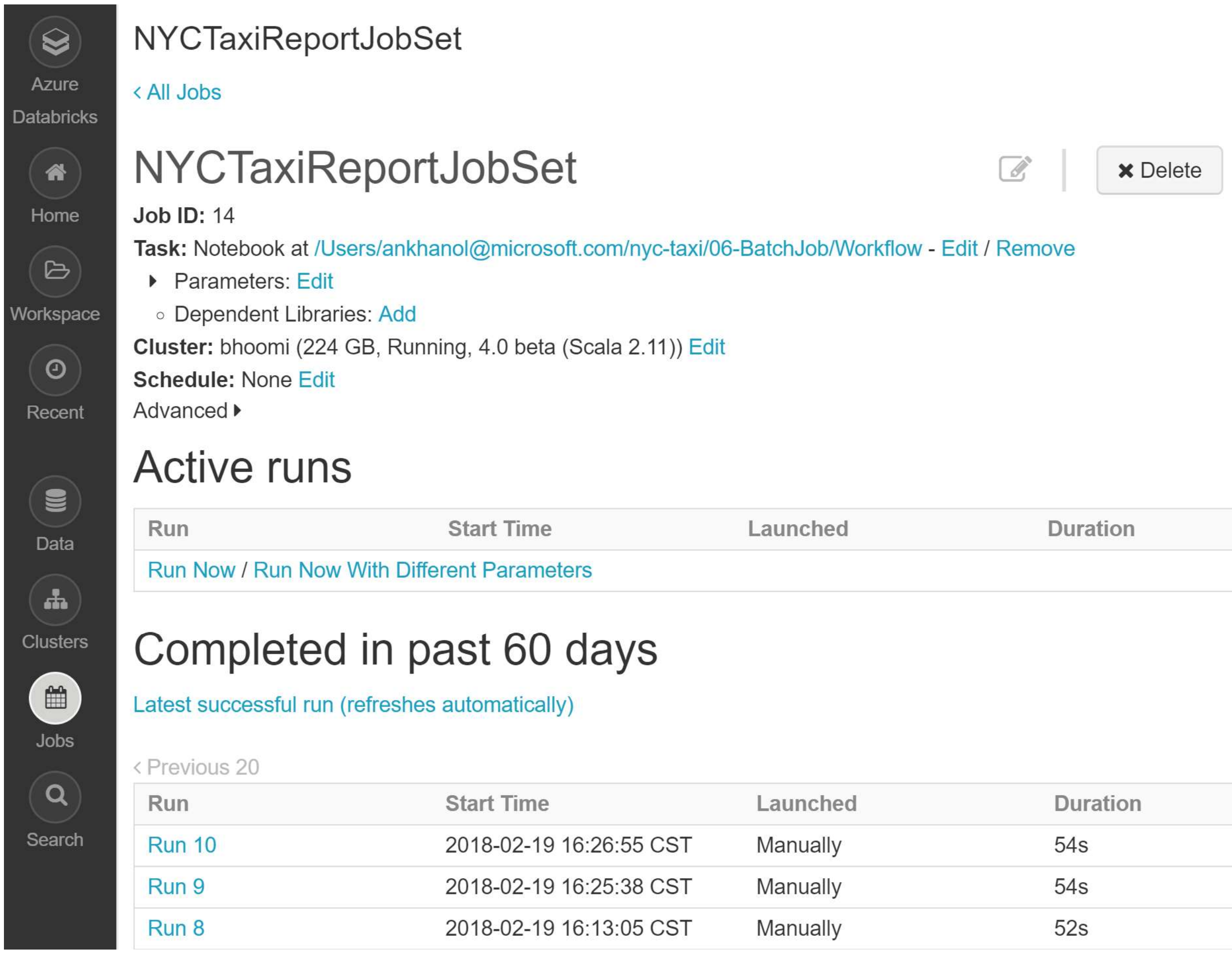
7.4. Workflow

(4) If #3 completed successfully, we repeat #3 for Report-2

(5) If #4 completed successfully, we exit with a pass status

(6) We create a batch job in which we call workflow.scala and do multiple ad-hoc executions


Microsoft Azure
Azure Databricks
Home
Workspace
Jobs
Create Job
Name
Job ID
Created By
Task
FlightDelayPrediction
5
Anagha Khanolkar
8.3-Workflow
NYCTaxiReportJobSet
14
Anagha Khanolkar
Workflow



Azure Databricks
Home
Workspace
Recent
Data
Clusters
Jobs
Search
NYCTaxiReportJobSet
‹ All Jobs
NYCTaxiReportJobSet
Delete
Job ID: 14
Task: Notebook at /Users/ankhanol@microsoft.com/nyc-taxi/06-BatchJob/Workflow - Edit / Remove
Parameters: Edit
Dependent Libraries: Add
Cluster: bhoomi (224 GB, Running, 4.0 beta (Scala 2.11)) Edit
Schedule: None Edit
Advanced
Active runs
Run
Start Time
Launched
Duration
Run Now / Run Now With Different Parameters
Completed in past 60 days
Latest successful run (refreshes automatically)
‹ Previous 20
Run
Start Time
Launched
Duration
Run 10
2018-02-19 16:26:55 CST
Manually
54s
Run 9
2018-02-19 16:25:38 CST
Manually
54s
Run 8
2018-02-19 16:13:05 CST
Manually
52s

Wrap up

Machine Learning Workshop - preview

Lecture/discussion:

Data science process
Data science lexicon – level set

Lab:

Use case: Flight delay prediction

Technologies/Services:
(1) Azure Machine Learning (AML) Studio – we will learn how to predict flight delays using AML – ingest, cleanse, dedupe, train, test, operational model, batch score

(2) Spark ML – we will repeat the exact same experiment in Spark ML – dataframe API for a more scalable solution

We will discuss how we can operationalize a Spark model trained in Databricks on the Azure ML platform

Q & A

1. Terminate your cluster
2. Any questions?

   Note: you will use the same cluster next week for the machine learning on Databricks workshop