Azure Discovery Days 2019

Data Analytics & Near Real Time Intelligence with Azure - Hands-On Lab Guide

Lab 1: Ingest and Store

Summary

In this hands-on lab, you will:

1. Set up an Azure storage account and a blob container
2. Ingest source file-based data to your Azure storage account
3. Deploy an Azure Databricks cluster
4. On Databricks, run a Jupyter notebook to load, transform, and emit data for use in the next lab.

About the Dataset

You will use the publicly available NYC Taxi data set: see http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. In this lab, you will use "Yellow" and "Green" taxi ride data. For Yellow, you'll use trips from 2010-2018, and for Green, you'll use trips from 2013-2018.

Each taxi company's data set format changed over the years. That is a real-world challenge; one of our tasks as data engineers working with real data over time is to handle changes in the data's structure. Additionally, we want to analyze all taxi ride data across companies, so we also have to merge their respective data, which involves further data structure work. This lab addresses both those challenges to prepare the data for analysis in later labs/tasks.

As this is a brief lab, you will be using a sampled-down dataset (see details below). This dataset was reduced in size so that relevant tasks below complete in minutes, not hours. In testing, every one of the data engineering tasks in this lab completed in a few minutes, in every case under ten minutes.

References

- Create an Azure Storage Account: https://docs.microsoft.com/azure/storage/common/storage-quickstart-create-account
- AzCopy on Linux: https://docs.microsoft.com/azure/storage/common/storage-use-azcopy-linux
- Spark and Databricks documentation
    - https://docs.azuredatabricks.net/spark/latest/data-sources/azure/azure-storage.html
    - https://docs.azuredatabricks.net/user-guide/dbfs-databricks-file-system.html
    - https://docs.azuredatabricks.net/user-guide/importing-data.html

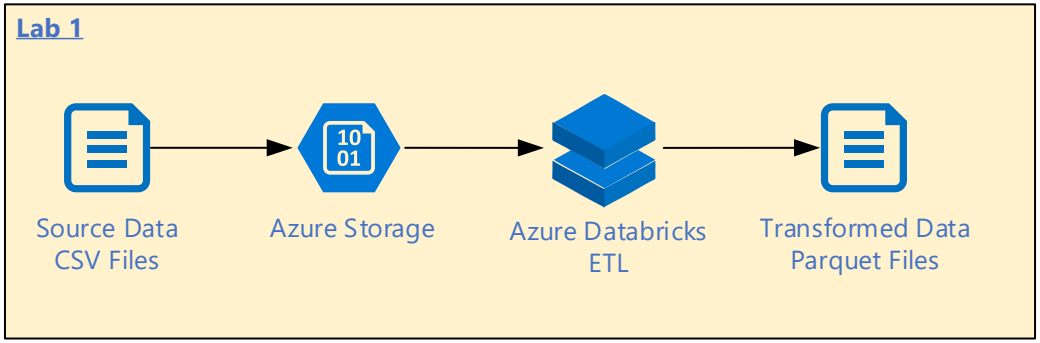
    - https://docs.azuredatabricks.net/spark/latest/faq/join-two-dataframes-duplicated-column.html
    - https://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame
- Jupyter Notebooks on Databricks: https://docs.azuredatabricks.net/user-guide/notebooks/index.html

General Notes

- IMPORTANT. Some browsers may not work correctly in the Azure or Databricks portals. If functionality is not working as shown in this lab document, please try a hard refresh or use a different browser.

Architecture for this Lab

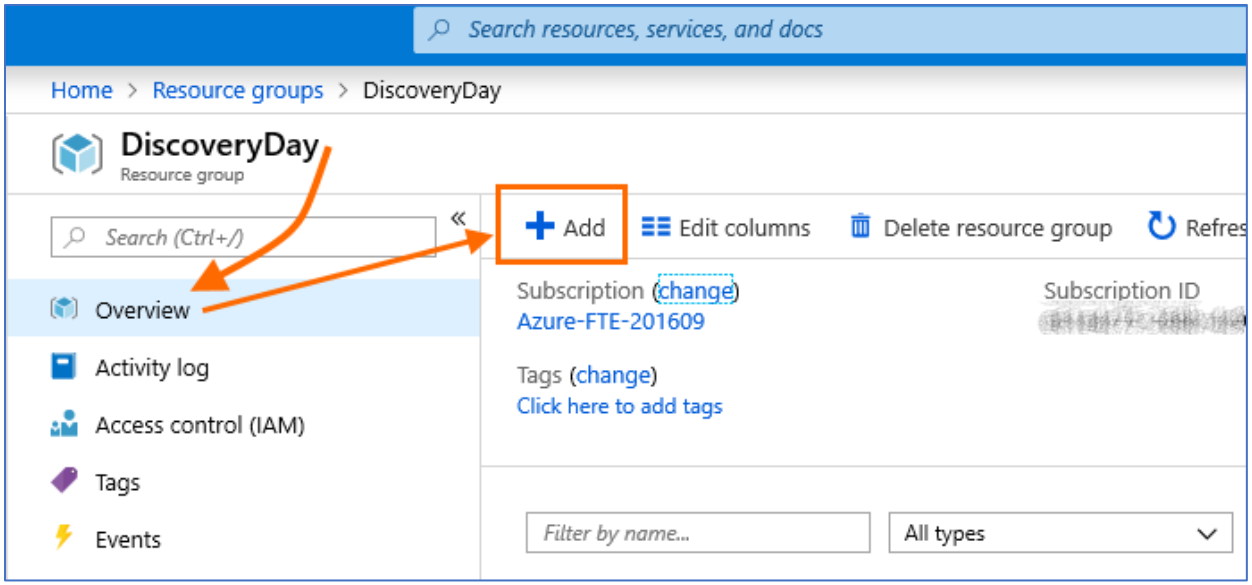
The tasks in this lab cover the following components of the overall architecture.



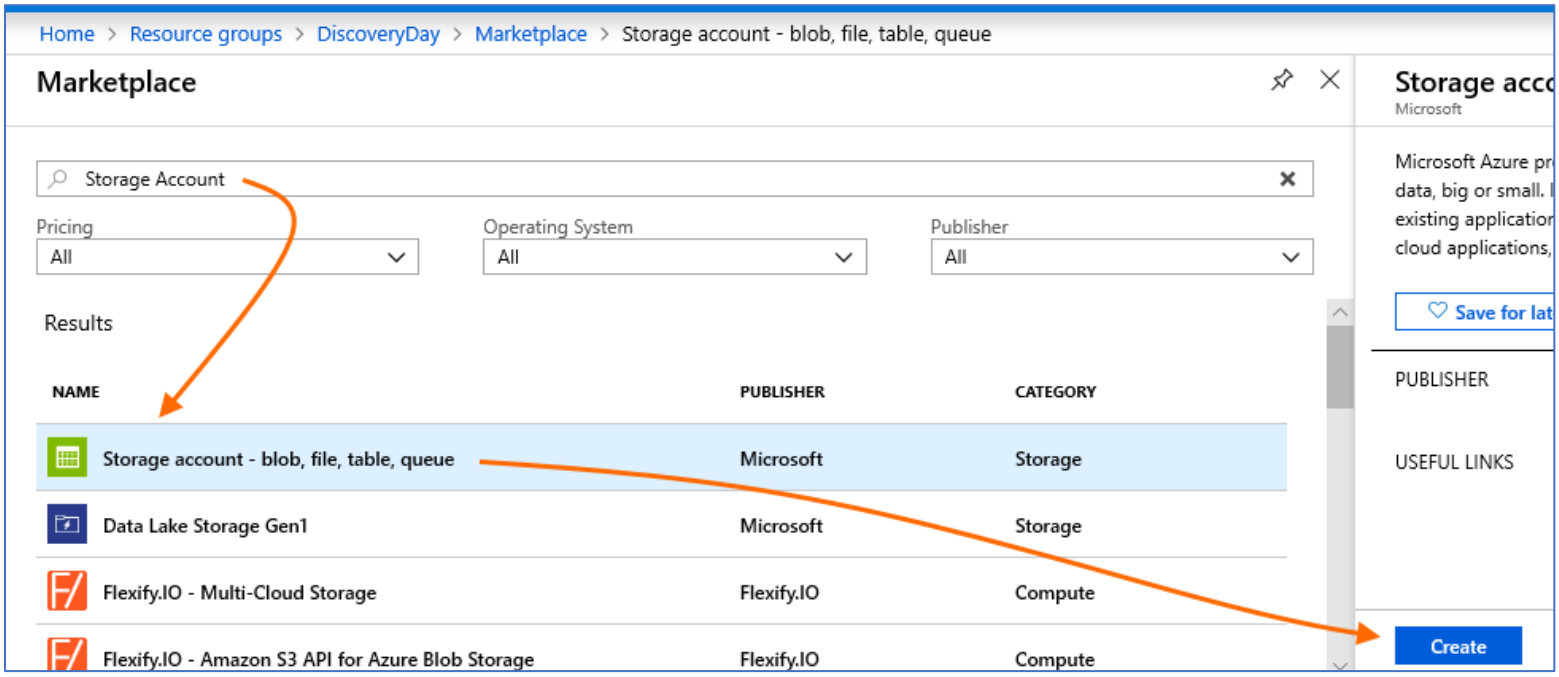
Task 1 – Set up an Azure storage account and a blob container

First you will create an Azure storage account in your resource group. This is where you will copy the source data files we will use later in this lab.

Start in the Azure portal, in the Resource Group you created in lab 0. Ensure you are on the “Overview” blade.

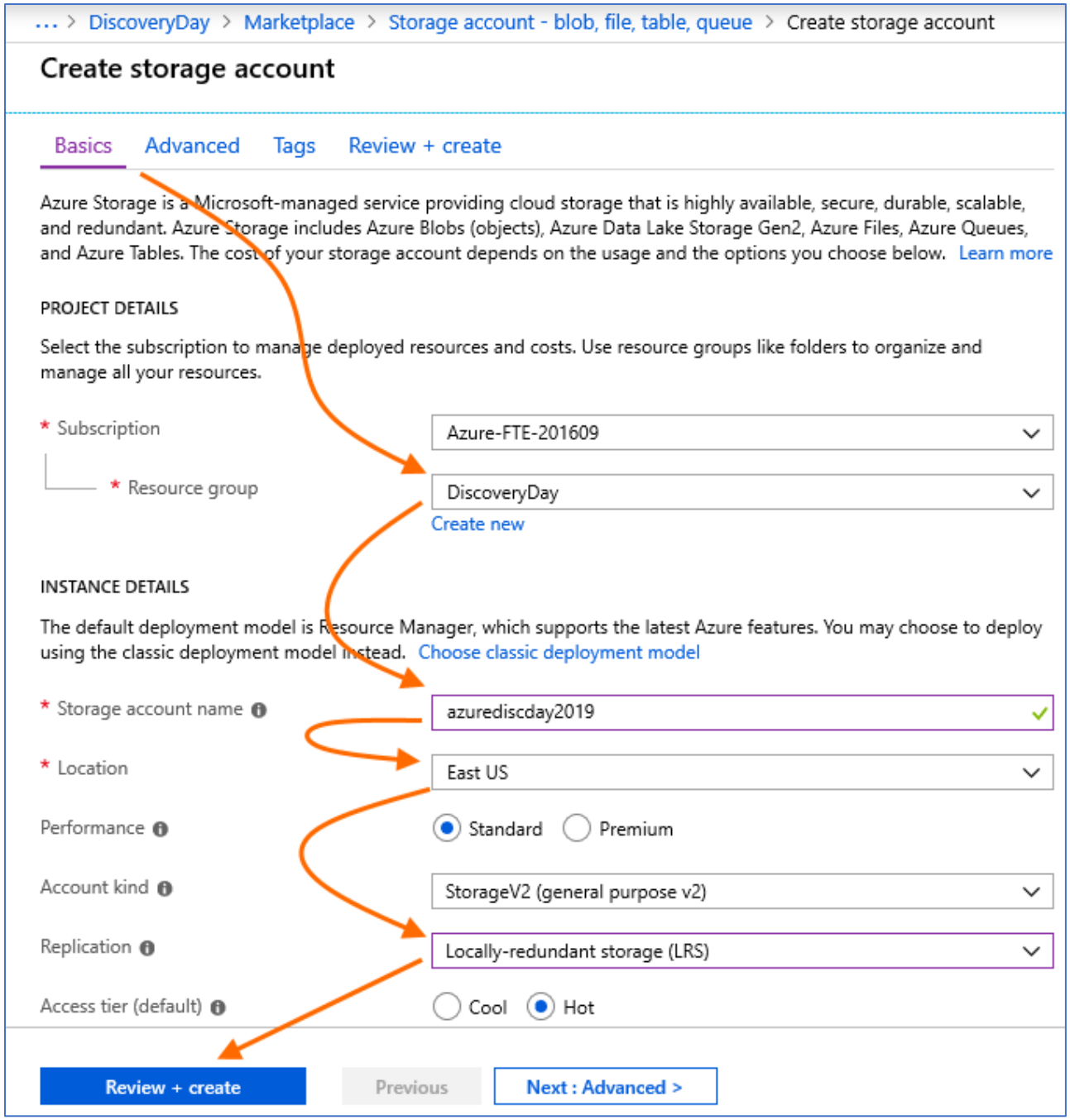


In the Search box, type “Storage Account” and hit Enter. From the results, click on “Storage account – blob, file, table, queue”. On the next blade, click “Create”.

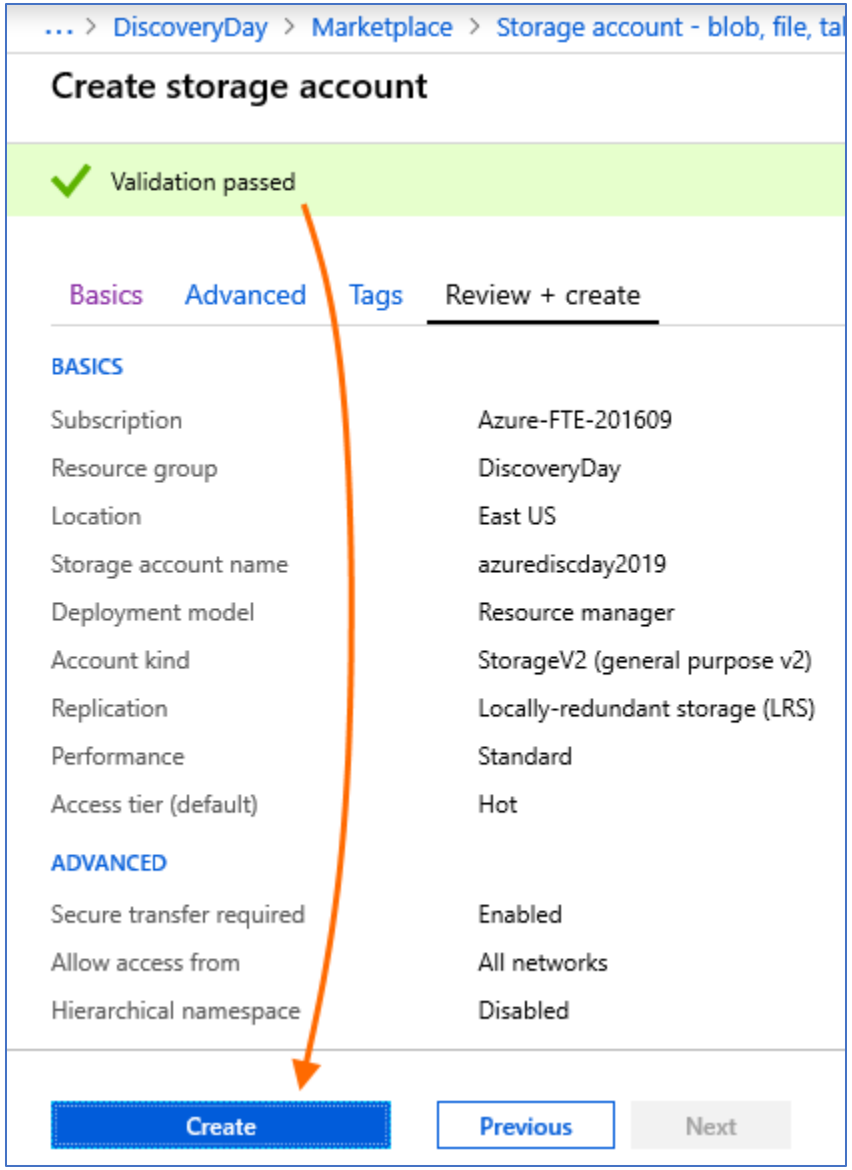


The create flow is divided into several tabs: “Basics”, “Advanced”, “Tags”, and “Review + create”. For this lab, we will only use “Basics” and “Review + create”, but please feel free to explore the other tabs.

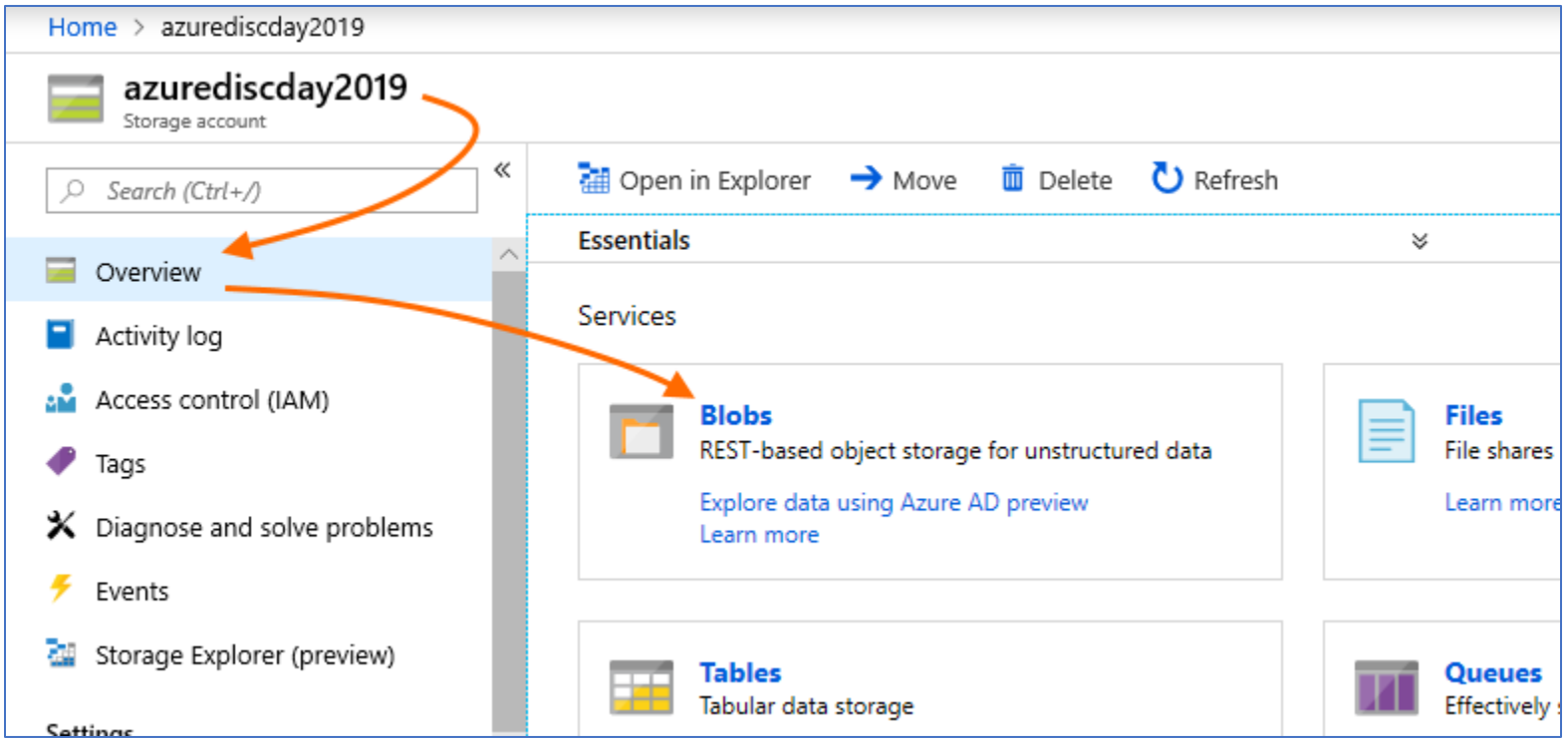
On the “Basics” tab, ensure your Resource Group is selected. Then provide a storage account name; this must be globally unique. Set the Azure region (Location) and set Replication to “Locally-redundant storage (LRS)” (this does not provide cross-region DR but is enough for this lab). Leave other settings at their defaults. Then click “Review + create”.


... > DiscoveryDay > Marketplace > Storage account - blob, file, table, queue > Create storage account
Create storage account
Basics
Advanced
Tags
Review + create
Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more
PROJECT DETAILS
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.
* Subscription
Azure-FTE-201609
* Resource group
DiscoveryDay
Create new
INSTANCE DETAILS
The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model
* Storage account name
azurediscday2019
* Location
East US
Performance
Standard
Premium
Account kind
StorageV2 (general purpose v2)
Replication
Locally-redundant storage (LRS)
Access tier (default)
Cool
Hot
Review + create
Previous
Next : Advanced >

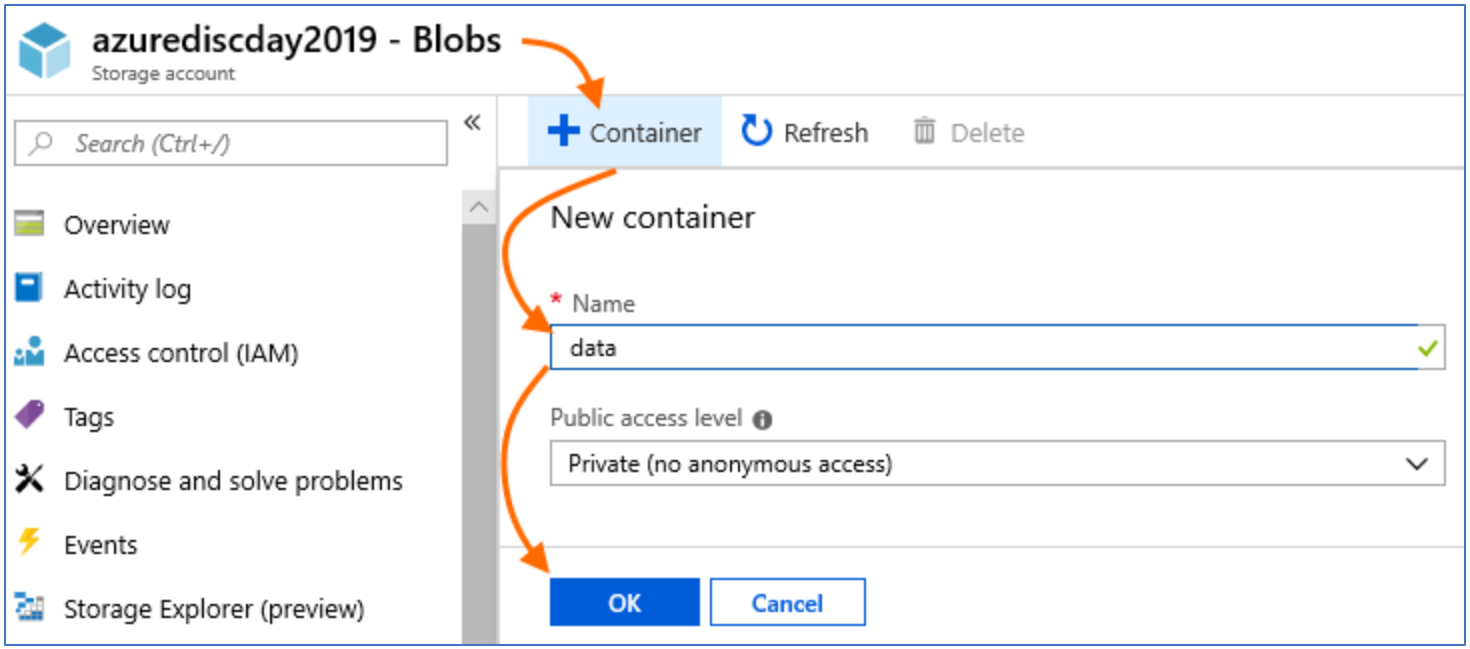
On the validation screen, click “Create”.



Wait for the deployment to complete (see Notifications, as shown in lab 0). Then go to your new storage account's Overview blade. Under Services, click "Blobs".



On the Blobs page, click "+ Container" and create a new blob container. The screenshot shows a container name of "data", but (subject to naming rules) please feel free to use a different name; however, be sure to adjust accordingly in later tasks and labs. Leave the public access level at its most secure default, "Private (no anonymous access)". Then click "OK".



After the container is successfully created and you can see it on the blob container page, this task is complete.

NOTE: for later tasks and labs, note the following information (e.g. in a OneNote page):

- Storage Account name
- Storage account key
- Blob container name

You can retrieve the key from the storage account's "Access keys" blade. You can use either the primary or secondary key.

Task 2 – Ingest source file-based data to your Azure storage account

In this task, you will copy the data files used for these labs. You will copy them from their canonical location, provided for this event, to the storage account you created in task 1. You will use your copy of the files, in your storage account, for all further tasks and labs.

You will retrieve two data folders: reference data, and a reduced set of NYC Taxi rides (the full data set is too large for this lab/event, so we have provided a sampled-down data set that preserves structure but significantly reduces data volume.)

The source data is located in the East US Azure region. If you are creating Azure resources in this region, your copy operations below will be faster than if you are working in a different Azure region.

Please note: you can work on Task 3 while your copy operations are processing!

Option 1: Copy files using AzCopy

You will use the AzCopy utility at an Azure CLI prompt. You can do this either in the Azure portal's Cloud Shell with no additional installations, or locally if you have downloaded and installed the Azure CLI and AzCopy tools. This task assumes you are using the Azure portal Cloud Shell, but the commands are the same if you are working locally.

You will perform two separate copy operations.

First, you will get some reference data. Start with the following azcopy command. Replace the three tokens ({YOUR STORAGE ACCOUNT NAME}, {YOUR CONTAINER NAME}, {YOUR STORAGE ACCOUNT KEY}) with the information you noted at the end of task 1. Next, type or paste the command with your info at the CLI and hit Enter. This command should take less than a minute to complete (in our tests copying within the Azure East US region, ~30 seconds but your times may vary).

```
azcopy --source "https://pzpubliceus.blob.core.windows.net/nyctaxi/reference-data/" --source-sas "?sv=2018-03-28&si=nyctaxi-public&sr=c&sig=f4%2ByhX8g9kngpufkftAgepsAt2WVC6D8xRLQEjjyF04%3D" --destination "https://{YOUR STORAGE ACCOUNT NAME}.blob.core.windows.net/{YOUR CONTAINER NAME}/reference-data/" --dest-key "{YOUR STORAGE ACCOUNT KEY}" --recursive
```

Second, you will get actual transaction data. Start with the following azcopy command, replace the same three tokens, paste the command with your info to the CLI, and hit Enter. This command should take several minutes to complete (in our tests copying within the Azure East US region, ~7 minutes but your times may vary).

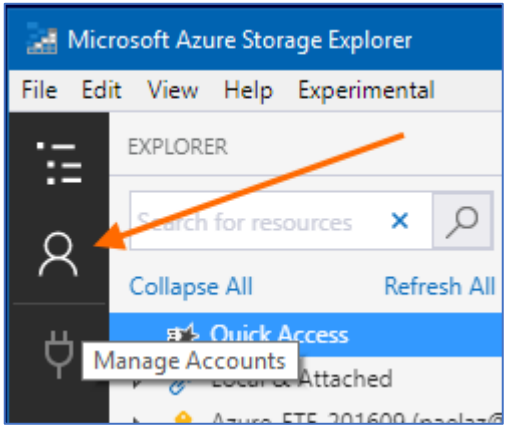
```
azcopy --source "https://pzpubliceus.blob.core.windows.net/nyctaxi/transactional-data-small/" --source-sas "?sv=2018-03-28&si=nyctaxi-public&sr=c&sig=f4%2ByhX8g9kngpufkftAgepsAt2WVC6D8xRLQEjjyF04%3D" --destination "https://{YOUR STORAGE ACCOUNT NAME}.blob.core.windows.net/{YOUR CONTAINER NAME}/transactional-data-small/" --dest-key "{YOUR STORAGE ACCOUNT KEY}" --recursive
```

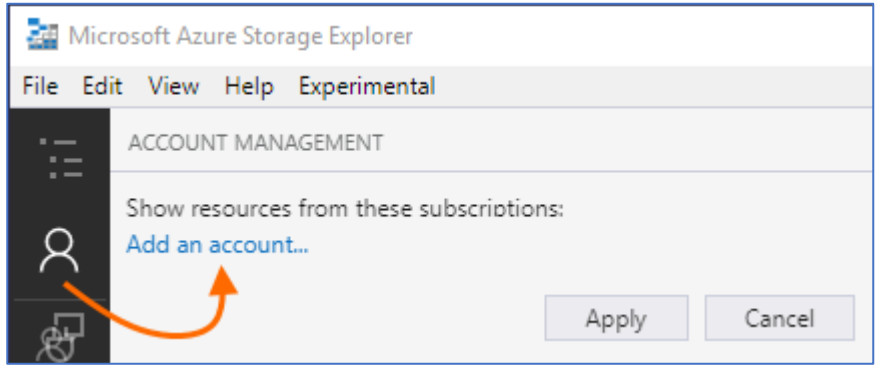
When the second command completes successfully, this task is complete.

Option 2: Copy files using Azure Storage Explorer

If you prefer to use a graphical tool, you can download and install Azure Storage Explorer from https://storageexplorer.com. After installation, start Storage Explorer.

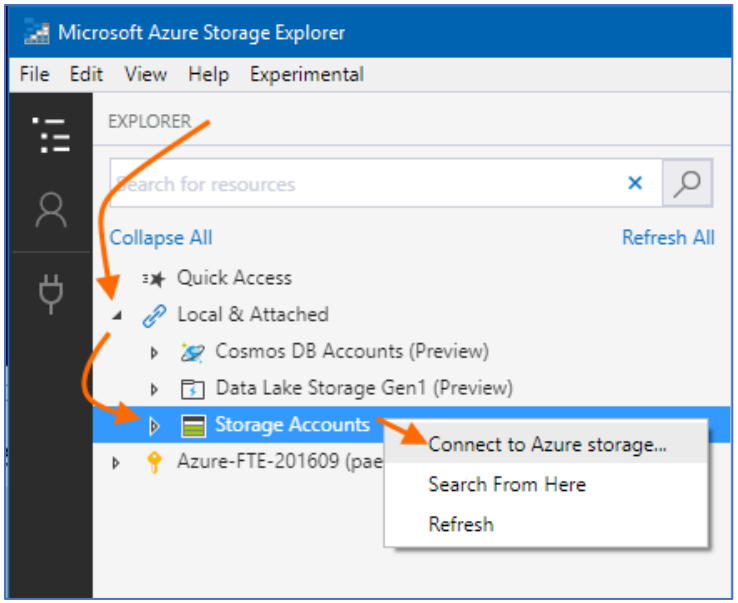
First, sign in to your Azure account. Click “Manage Accounts”, then “Add an account…” to sign into your Azure subscription. When the sign-in process is complete, you should see your Azure account and subscription in the “ACCOUNT MANAGEMENT” list. Click “Apply”, and you should see your subscription and storage account in the EXPLORER list.



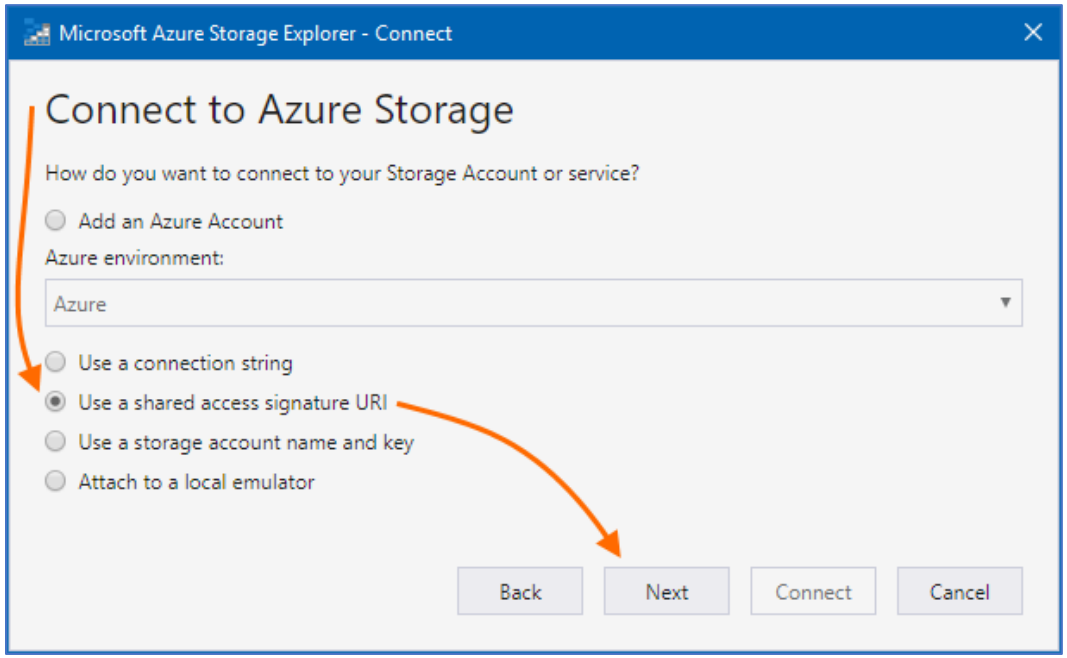


You have now connected to your new storage account, which is the destination to which you will copy the source data files. Now, you need to connect Storage Explorer to the file source, so that you can perform the copy operations.

In the EXPLORER view, open “Local & Attached”. Right-click “Storage Accounts” and click “Connect to Azure storage…”.

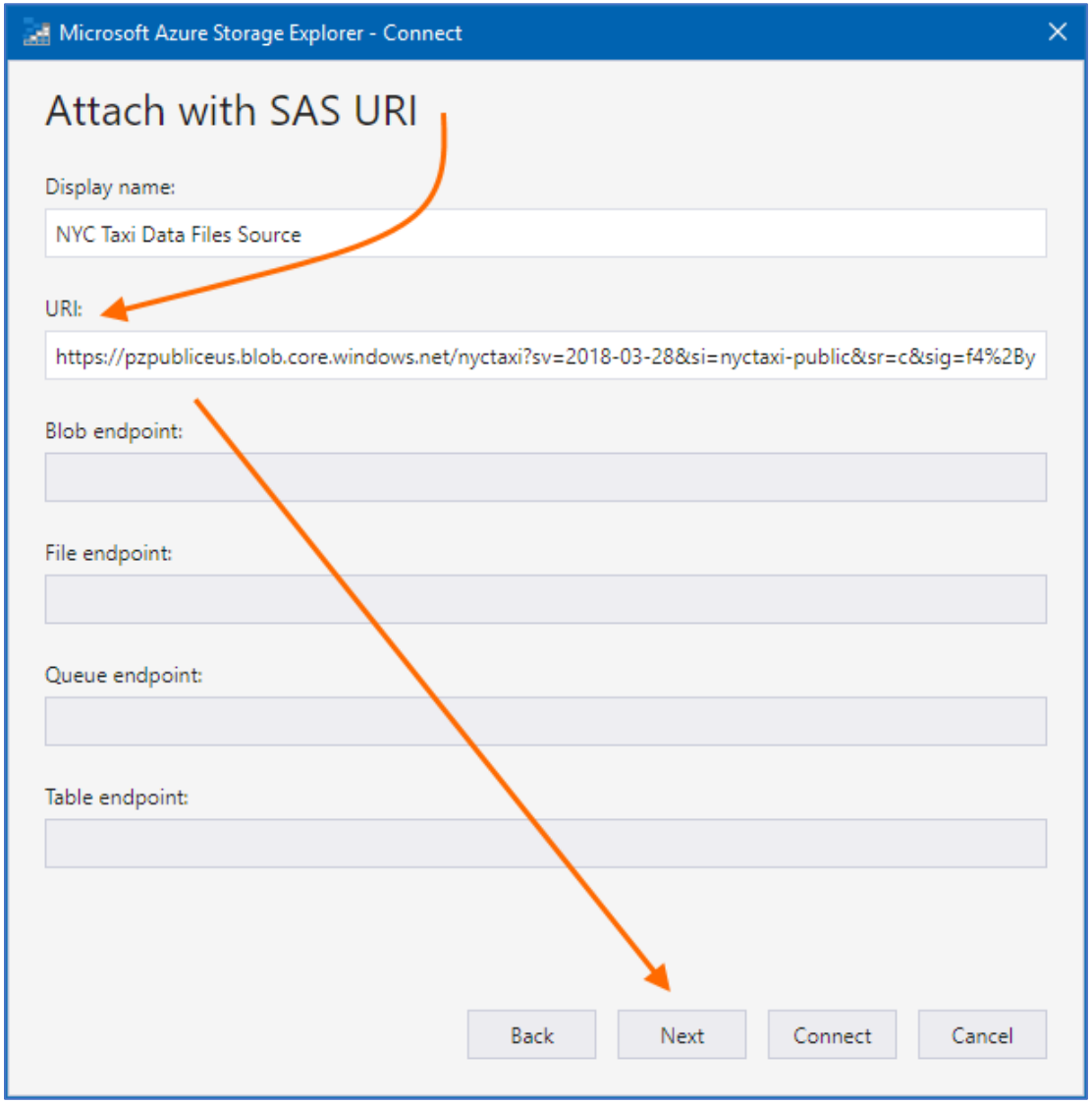


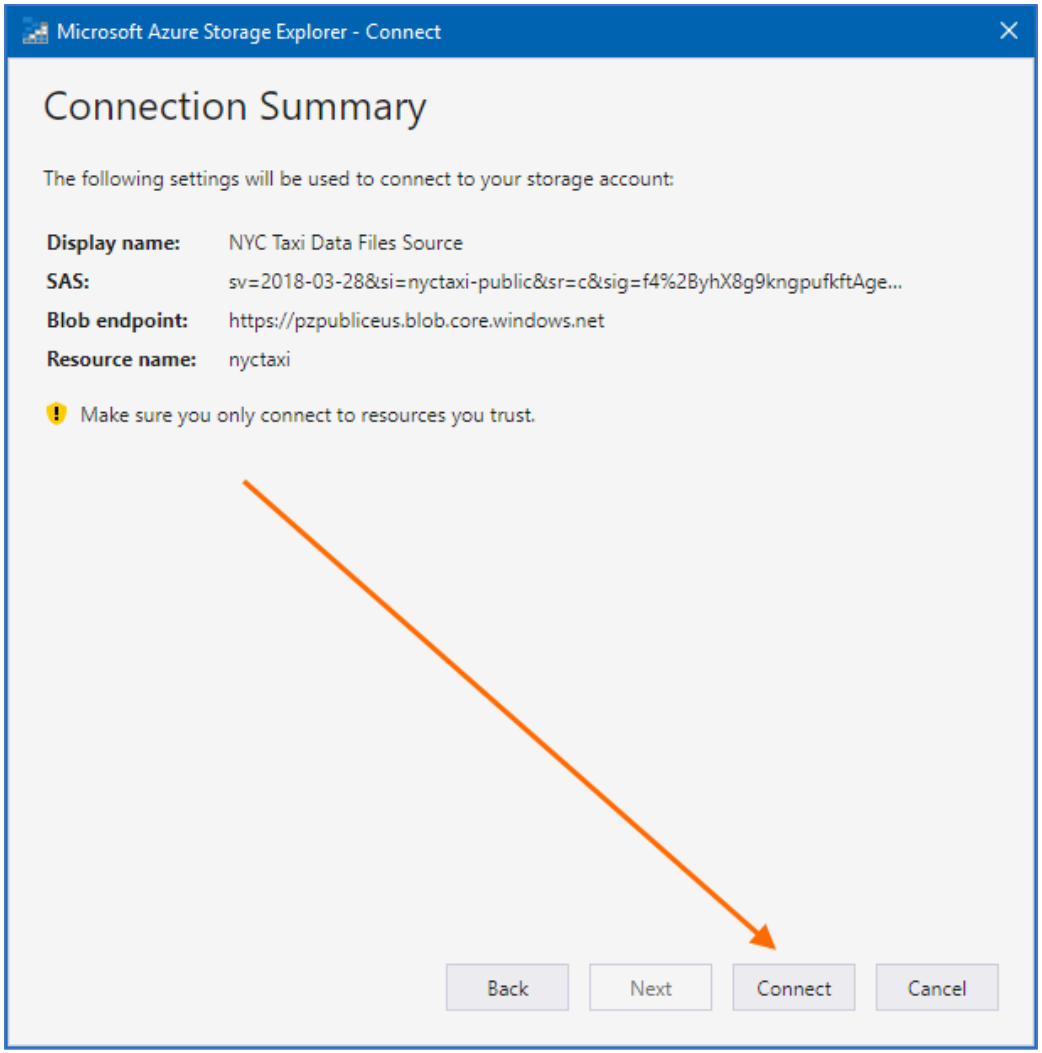
Select “Use a shared access signature URI” and click “Next”.



Paste the following URI into the “URI” text box. Feel free to adjust the Display Name – this is for visual purposes only and has no functional impact. Then click “Next”, then “Connect”.

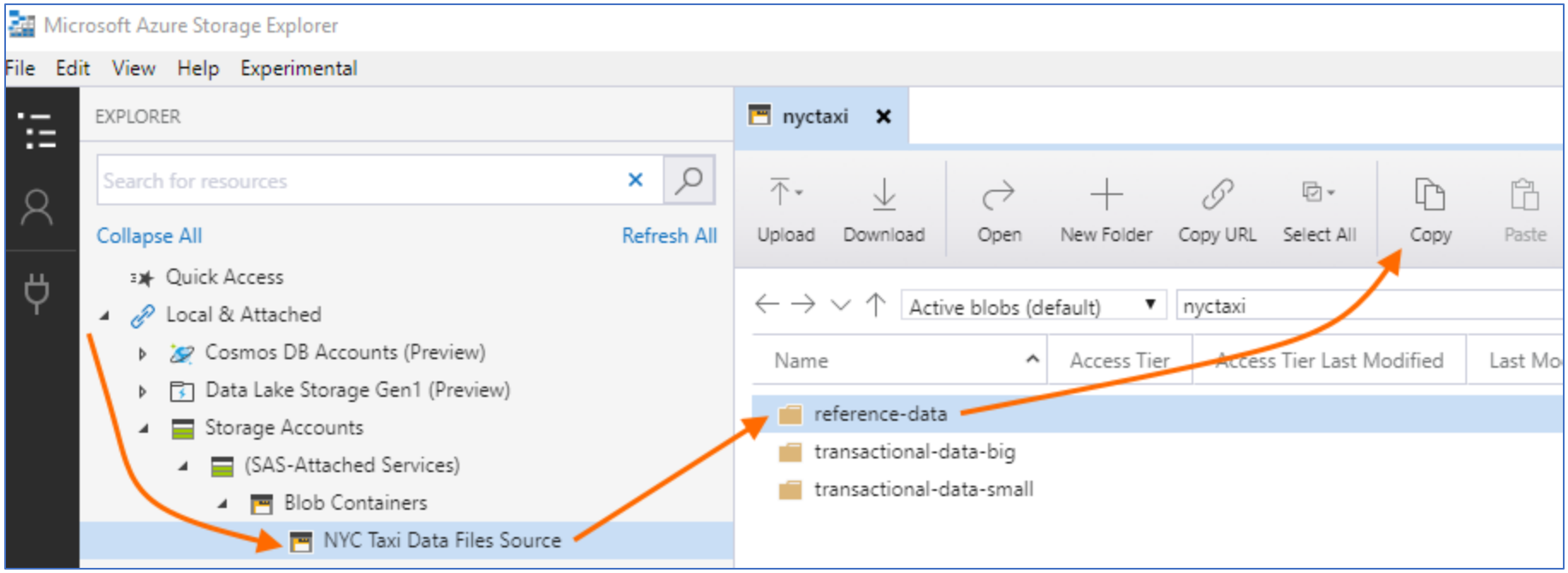
https://pzpubliceus.blob.core.windows.net/nyctaxi?sv=2018-03-28&si=nyctaxi-public&sr=c&sig=f4%2ByhX8g9kngpufkftAgepsAt2WVC6D8xRLQEjjyF04%3D



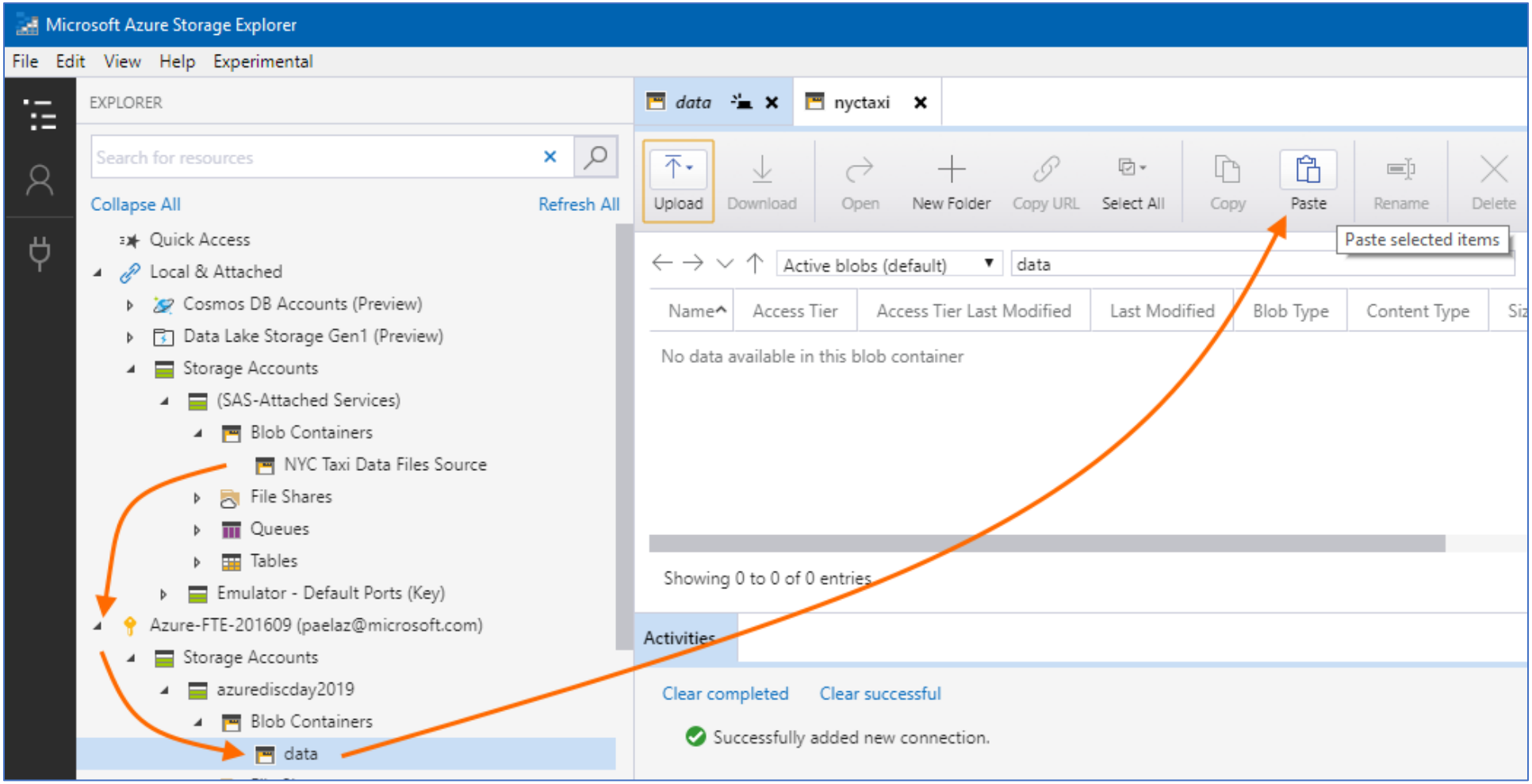


You should now see the attached source storage account in the EXPLORER list. Select it.

Next, select the “reference-data” folder, then click “Copy” in the toolbar.



Next, open your subscription, your new storage account, and select the new container you created in task 1. Click “Paste” in the toolbar. This will copy the “reference-data” files into your storage container.

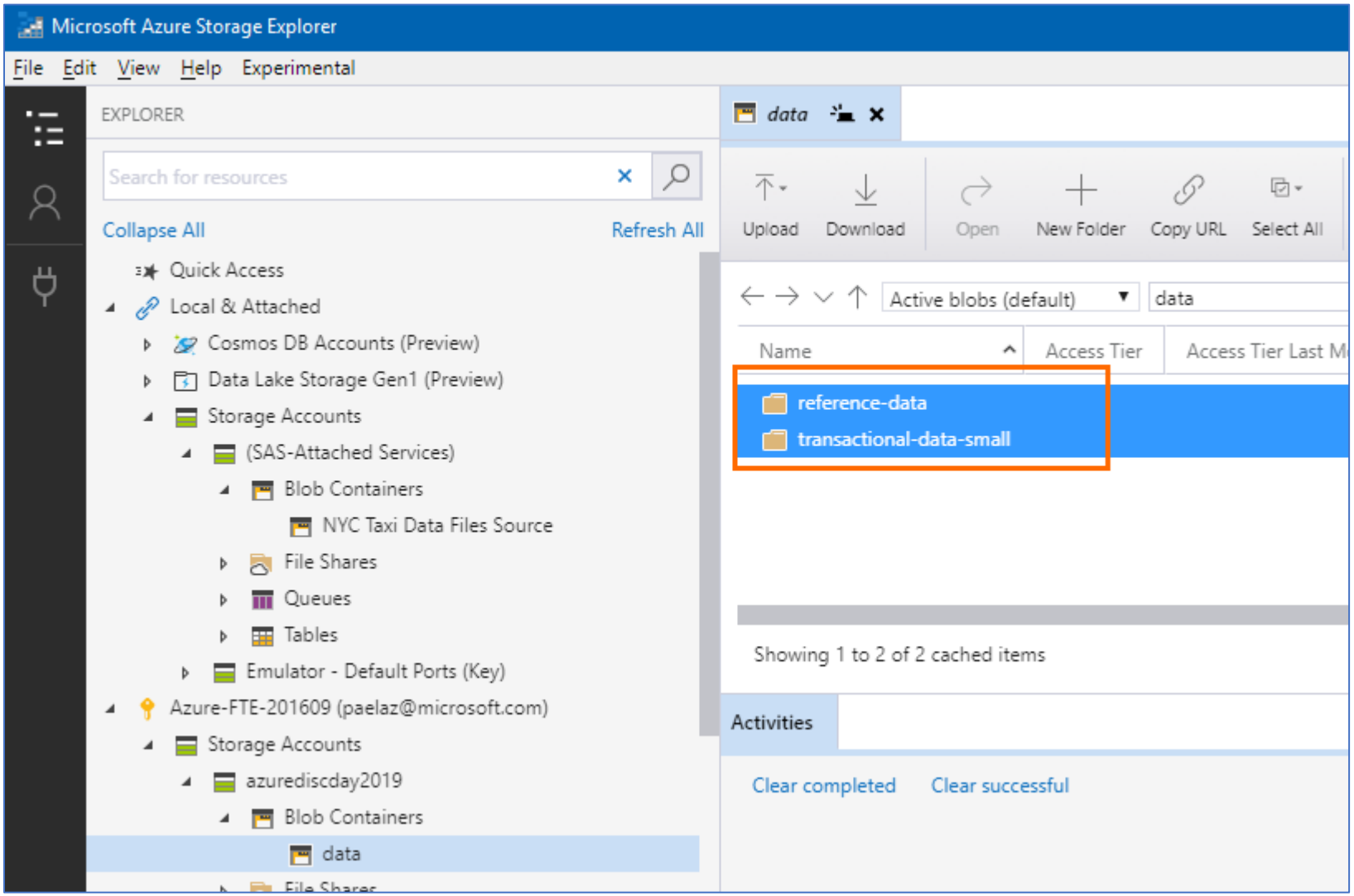


When this operation completes, you should see the “reference-data” folder in your storage container. You can right-click “reference-data” and then click “Selection statistics…”, and do the same on the source “reference-data” folder, to verify that file counts and total file size are identical (as you would expect after a successful copy).

Repeat the above process to copy the “transaction-data-small” folder from the source to your destination storage account.

IMPORTANT: do **NOT** copy the “transaction-data-big” folder! This is the raw, full-size data set and is too large for the tasks in this and later labs to complete in a reasonable amount of time!

When both operations have completed, you should see both folders in your storage account.

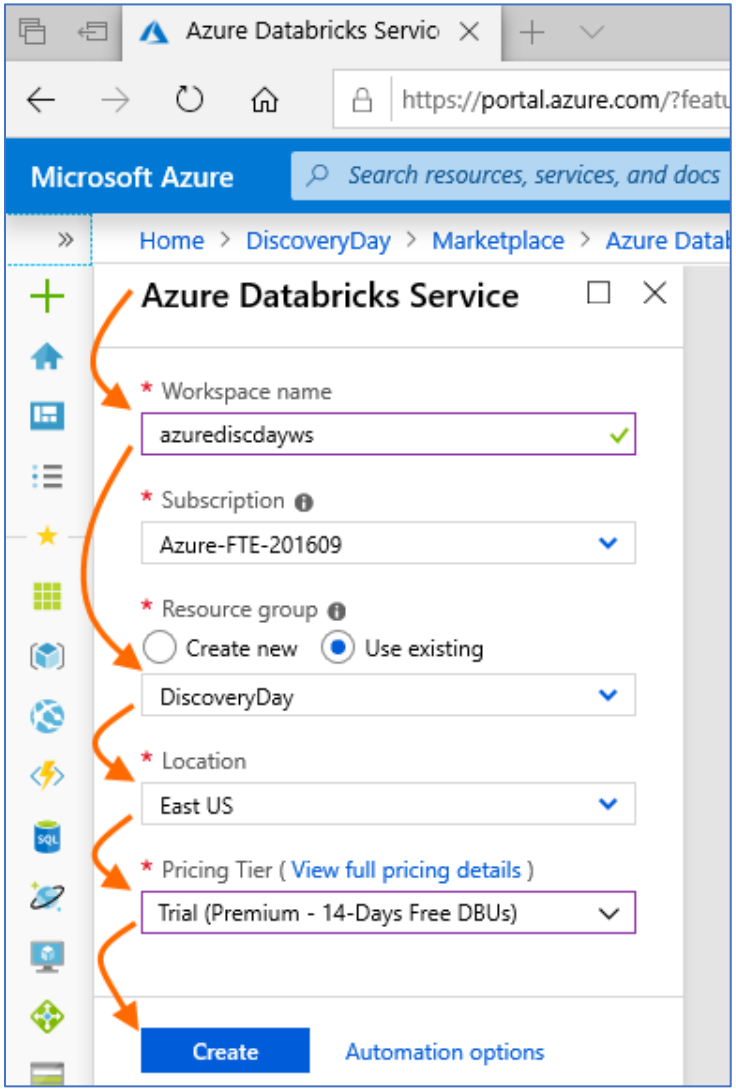


Optionally, you can now disconnect from the *source* storage account (where you copied *from*) by right-clicking it and selecting “Detach”. Now that you have the data files in your storage account, you no longer need the source storage account for further tasks or labs.

When you can see both “reference-data” and “transactional-data-small” folders in your storage account, this task is complete.

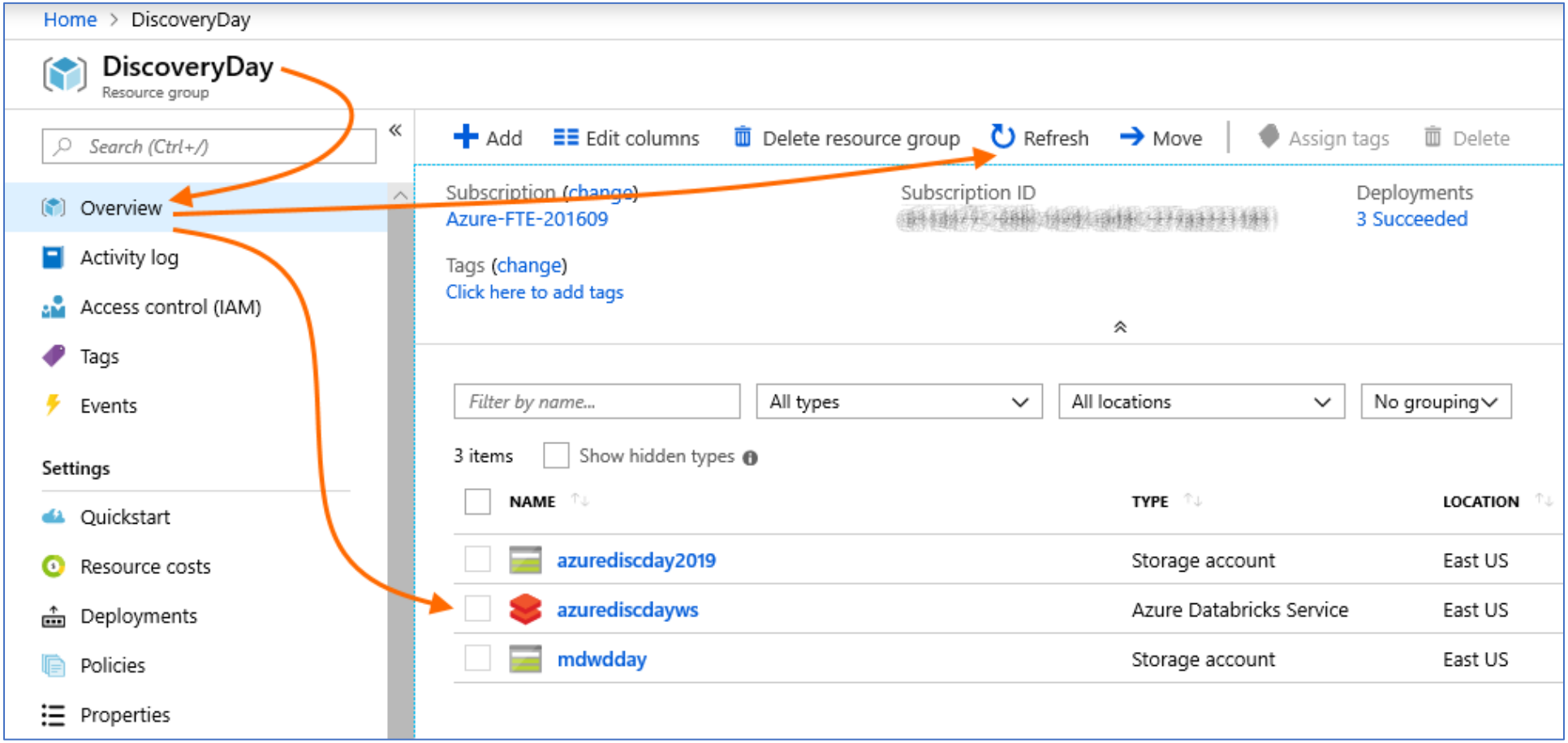
Task 3 – Deploy an Azure Databricks cluster

In the Azure portal, navigate to your Resource Group's Overview blade. Click "+Add". Type "Azure Databricks" in the search box and select it from the results. Click "Create" on its product blade. On the next blade, provide a workspace name; select the Resource Group you created in lab 0; select the Azure region you decided to use in lab 0. If available, select the Trial pricing tier, otherwise select the Standard tier. Then click "Create".

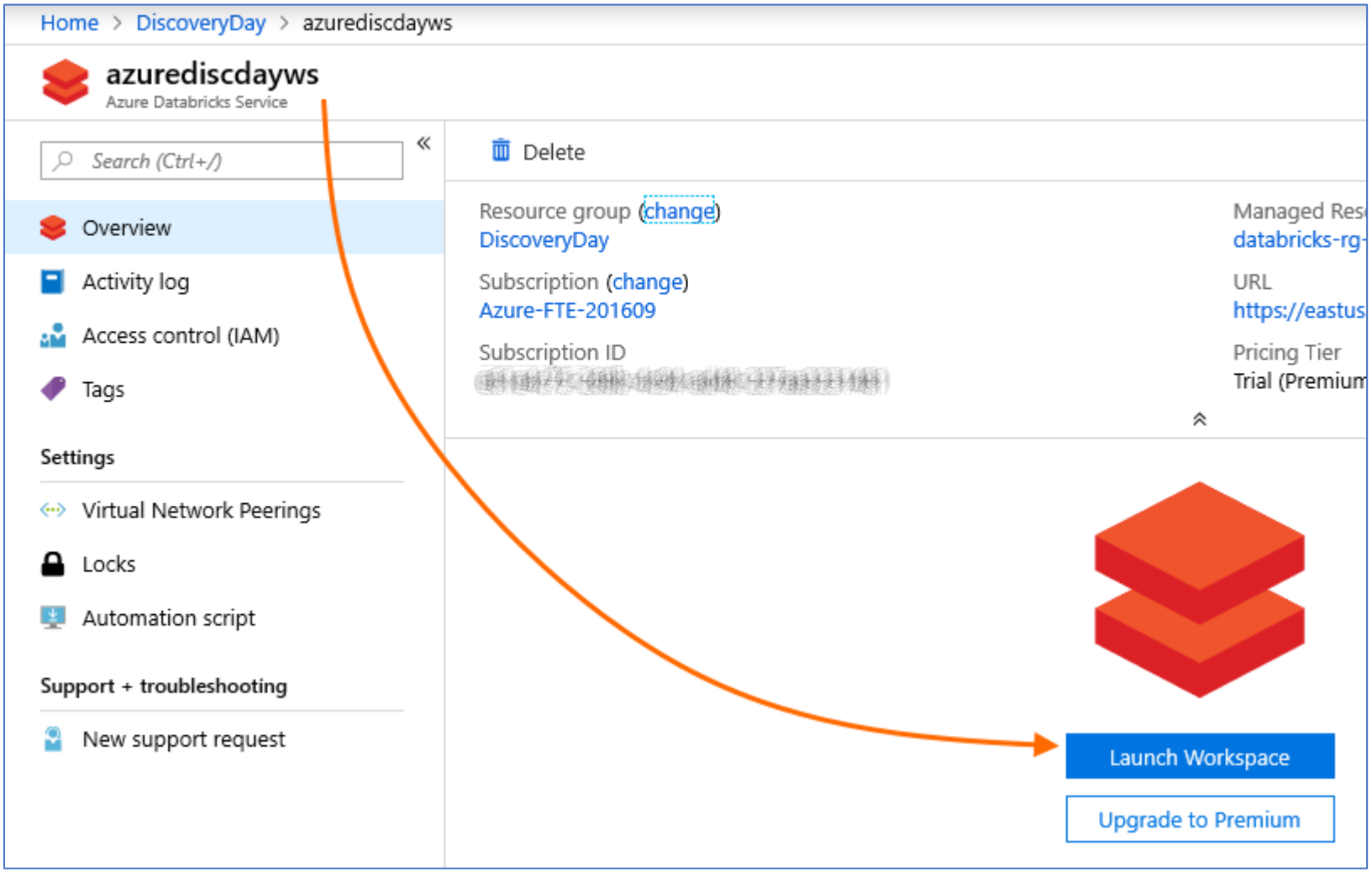


After clicking “Create”, the deployment of the Databricks workspace will proceed. This is not the actual *cluster* on which we will process data yet; this is the *environment* into which we will then deploy a *cluster*. Creation of a Databricks workspace will take a few minutes (in our tests in the East US region, this took ~2 minutes but your times may vary).

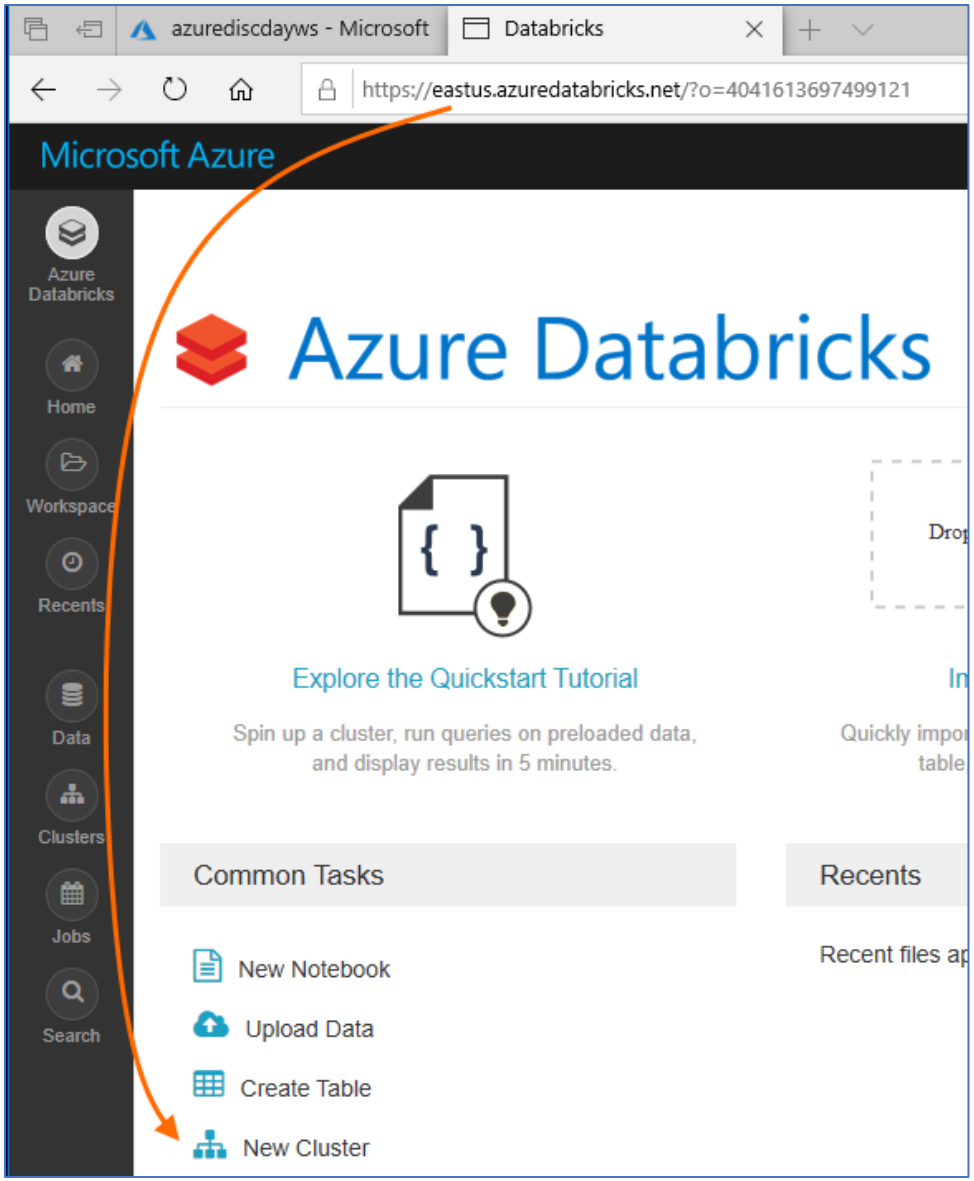
After workspace deployment completes (see Notifications as in previous labs/tasks), navigate to your Resource Group’s Overview blade. You may need to click the “Refresh” button in the portal. You should now see the Databricks workspace.



Next, click on the new Databricks workspace resource which you just deployed. On that page, click on “Launch Workspace”. That will bring up a new web site specific to your Azure Databricks workspace, which will sign you in automatically with your current Azure credentials.

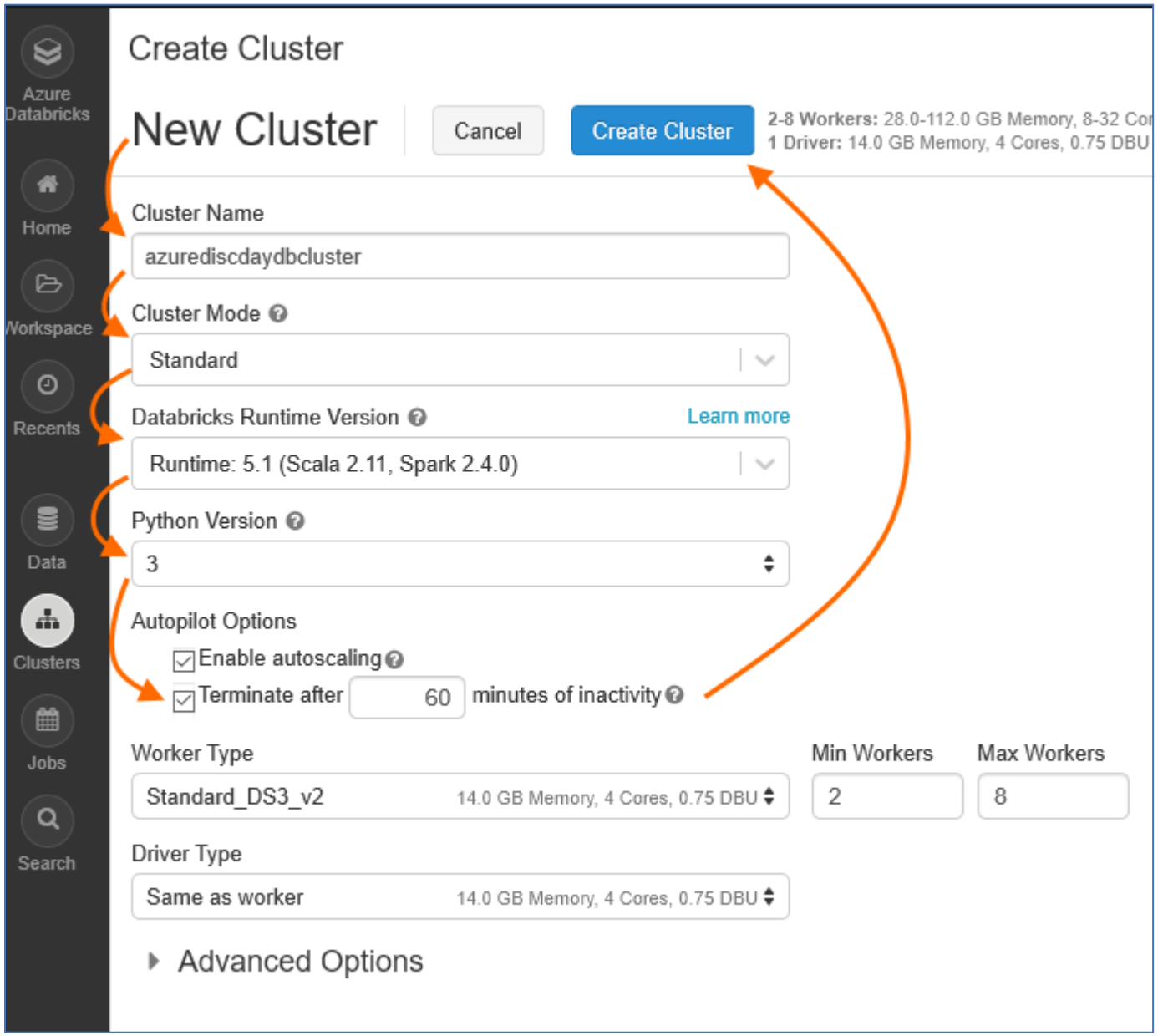


Once you are on the workspace home screen, click “New Cluster”.

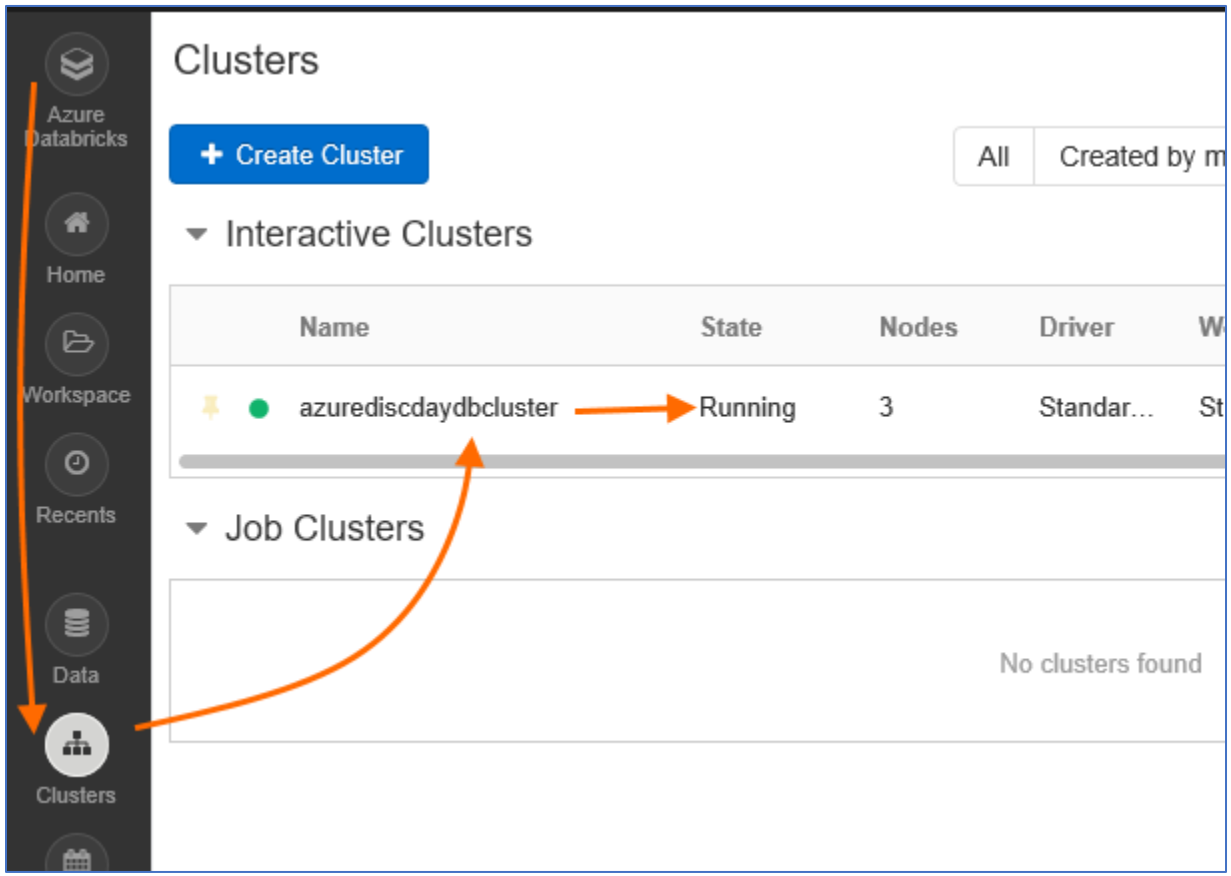


On the Create Cluster page, provide required information. For this lab: provide a name for your cluster; select “Standard” cluster mode; select the most recent, production (non-preview), non-GPU Databricks Runtime Version (at the time of this writing, Databricks 5.1); select Python 3; reduce inactivity timeout to 60

minutes; and leave the remaining options at their defaults unless you would like to experiment. Then click “Create Cluster”. In our tests, cluster deployment with the parameters shown here took a few minutes, but well under ten minutes.



When the cluster completes deployment, you should see it in the Clusters view with State “Running”.

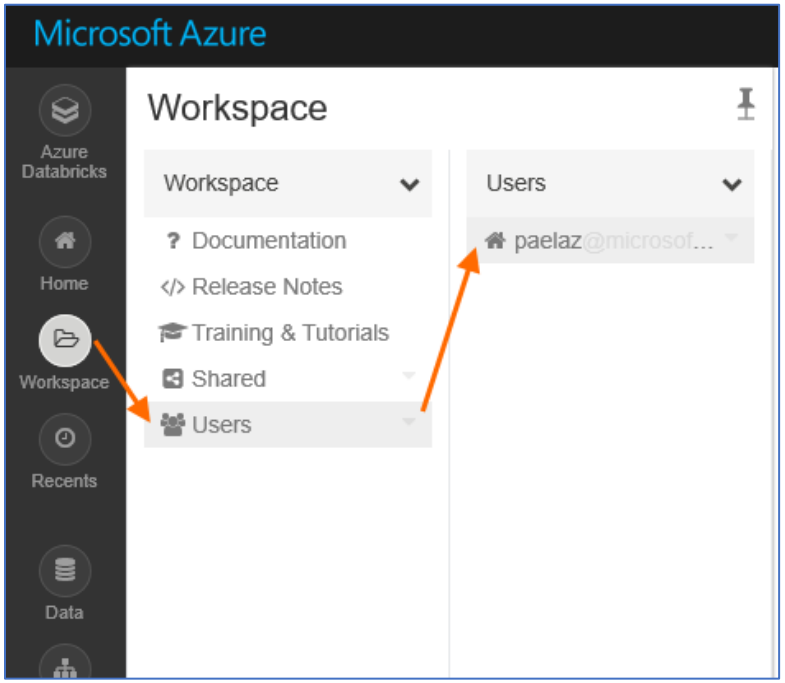


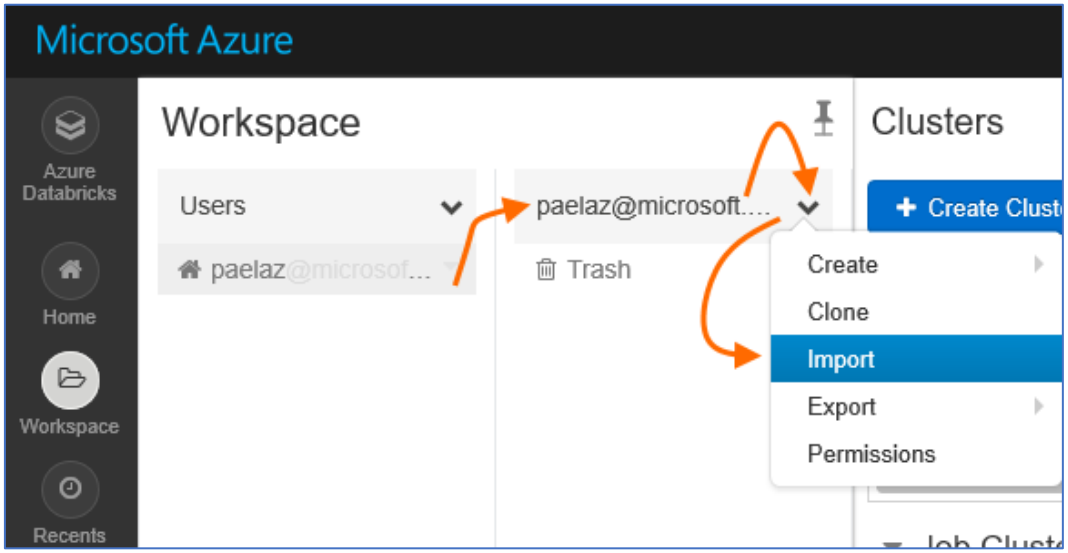
When the deployed cluster is shown as Running, this task is complete.

Task 4 – Run a Jupyter notebook on Databricks to ingest, transform, and emit data

Click on “Workspace” in the left nav bar. Navigate through “Users” and your username until you see a dropdown arrow next to your username.

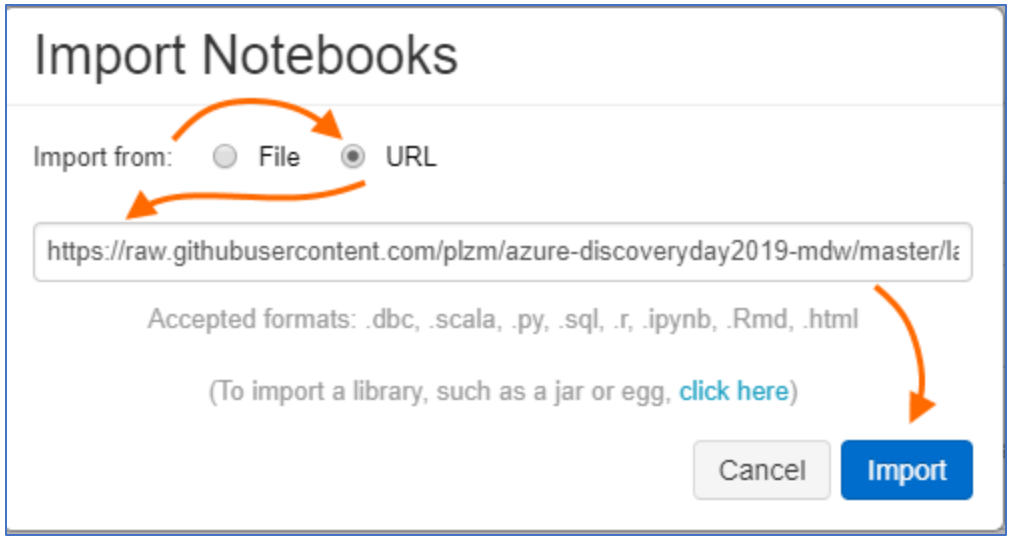
For this lab, you will import and adapt an existing Jupyter notebook. To create your own notebooks from scratch, you would click “Create” – but for this lab, you will use an existing notebook. Click the dropdown and select “Import”.



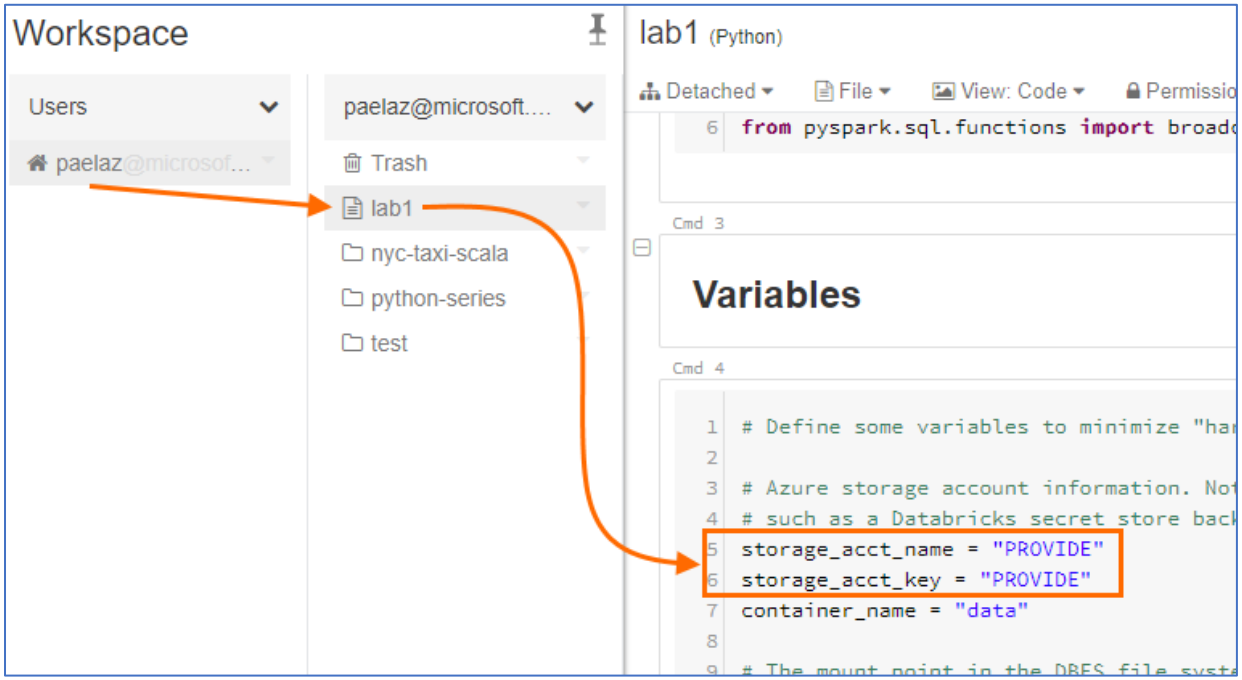


In the “Import Notebooks” dialog box, set “Import From” to URL, then paste the following URL into the URL text box:

https://raw.githubusercontent.com/plzm/azure-discoveryday2019-mdw/master/labs/lab1/lab1.ipynb



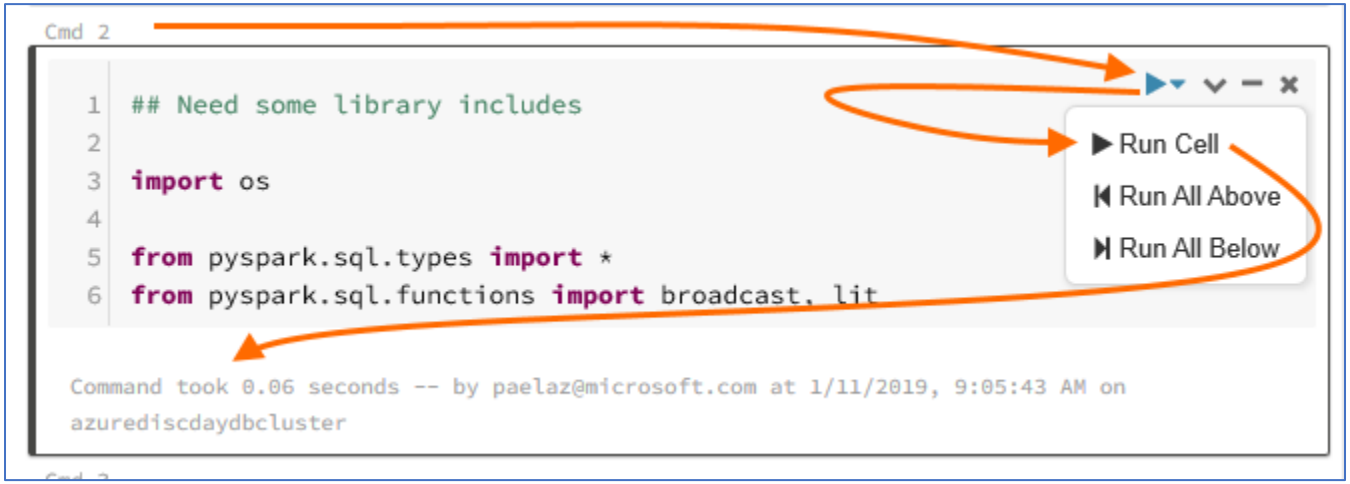
After clicking “Import”, the notebook should appear in your Databricks workspace. Before running the notebook, you must replace the placeholder “PROVIDE” with your storage account name and key in Cmd 4.



Once you have provided these values, you can step through the notebook. The notebook is fully functional as provided, but we strongly suggest you step through each command to understand what is being done, how, and why. Feel free to experiment and change the notebook; remember, you can always re-import it from the source link provided above.

If you are not familiar with Jupyter notebooks, please use the documentation provided in the References section at the start of this document, "Jupyter Notebooks on Databricks", for the basic key strokes and commands to step through notebook cells one by one.

Briefly: you can step through each notebook cell on its own, in order or out of order, by clicking in the cell and then either using the Shift+Enter keyboard shortcut to run the cell and step into the next one, or by using the cell's Run dropdown and selecting "Run Cell". The cell will then run and, when complete, will provide a status message and focus will move to the next cell.



In summary, this notebook ingests the raw data files from your storage account; performs schema transformations to conform the data set structures to a canonical structure; and emits the data in Parquet format. Trip data is emitted both in taxi company/year/month format as well as single merged dataset format.

Conclusion

Once you have successfully run this notebook through the tasks required for this lab and emitted Parquet data to your storage account, this task and this lab are complete! You can verify the contents of your storage account in the desktop Storage Explorer or in the Azure portal, in your storage account under "Blobs".

While your Databricks cluster will automatically shut down after the inactivity period specified when you created it, you can also terminate your cluster manually at this point, as it will not be used in other tasks or labs.