Azure Discovery Days 2019

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

Lab 2: Copy processed data from Azure Blob storage to Azure SQL DB

Summary

In this hands-on lab, you will:

1. Create an Azure SQL DB.
2. Create an Azure Data Factory.
3. Create an Azure Data Factory pipeline using the Copy Data wizard.
4. Create Power BI Reports using Power BI desktop and Azure SQL DB as data source.
5. Deploy Power BI reports to Power BI Online and create Dashboards.

About this Lab

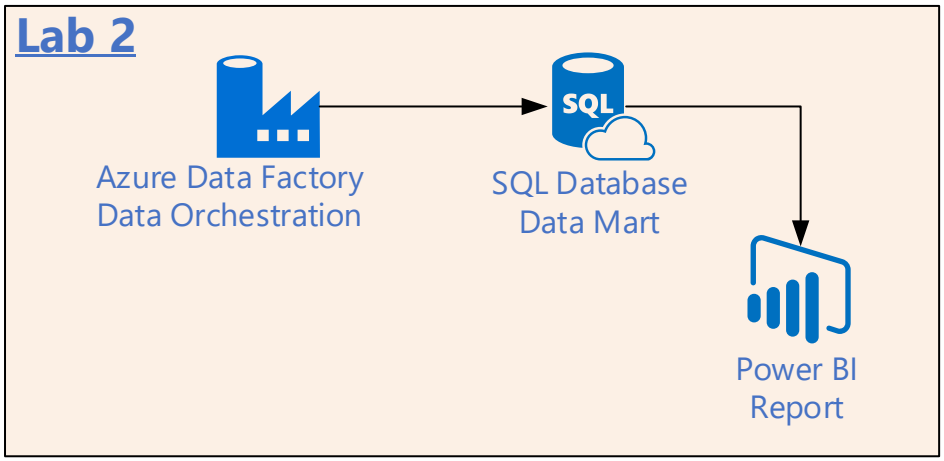
Copy data from Azure Blob storage (Parquet format) to Azure SQL Database using copy recursion option. And then create Power BI reports & dashboards using Azure SQL Database as a source.

References

https://docs.microsoft.com/en-us/azure/data-factory/tutorial-copy-data-tool

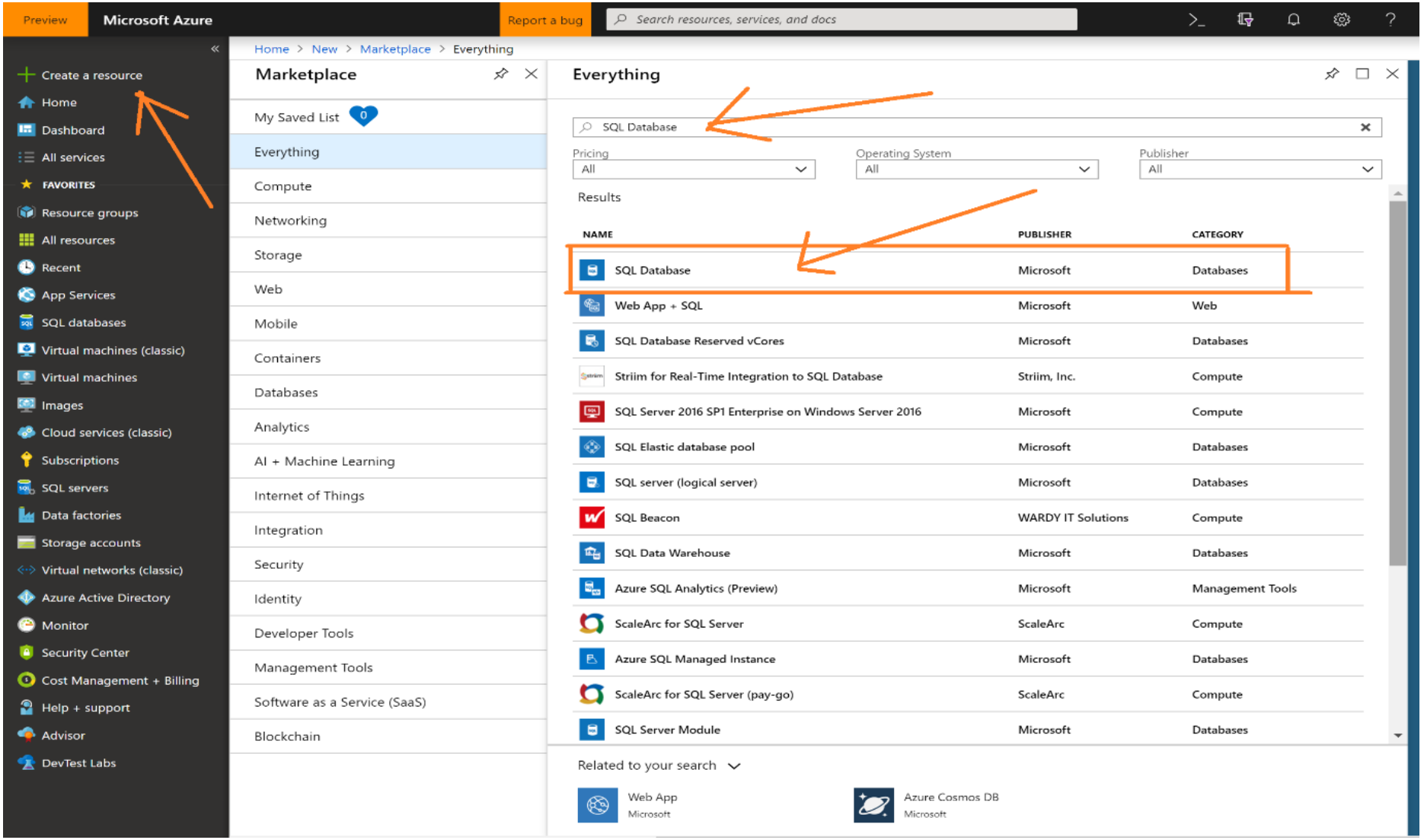
Architecture for this Lab

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

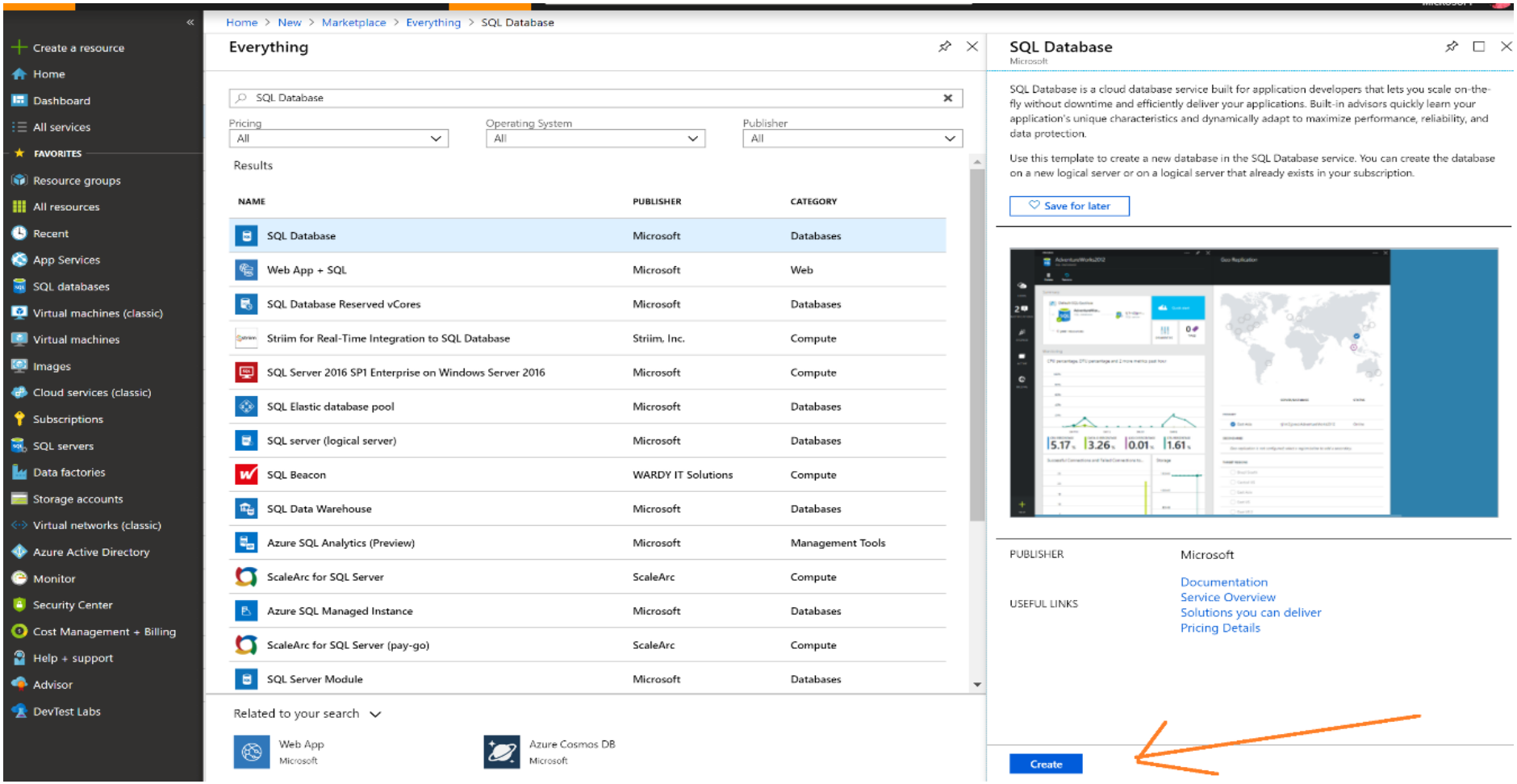


Task 1 – Create Azure SQL Database

Go to Azure portal and then select 'Create a resource' → Search for SQL database → Select SQL Database as shown:



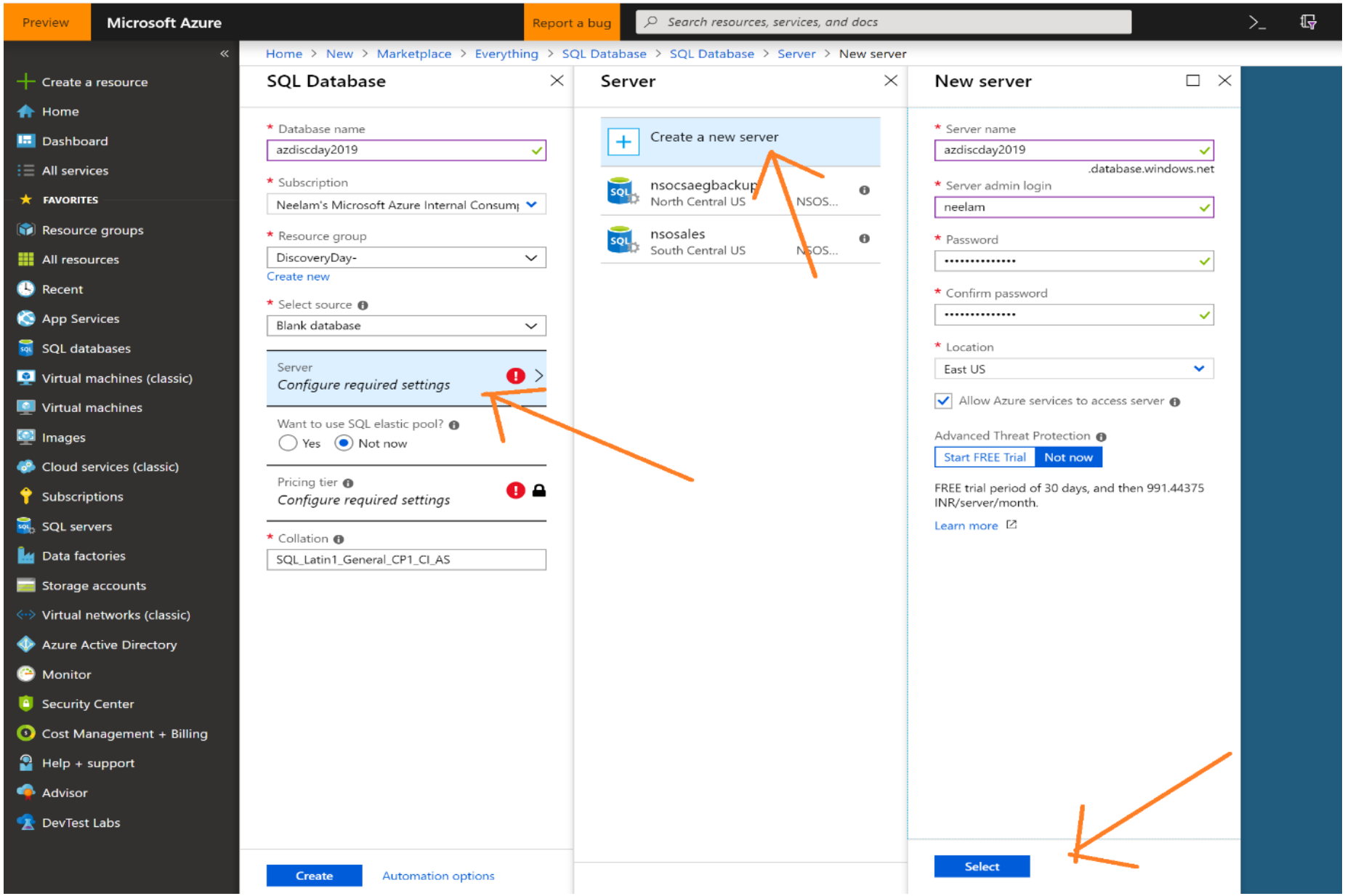
Please click create.



Enter Database name, select Azure subscription and select your Resource Group.

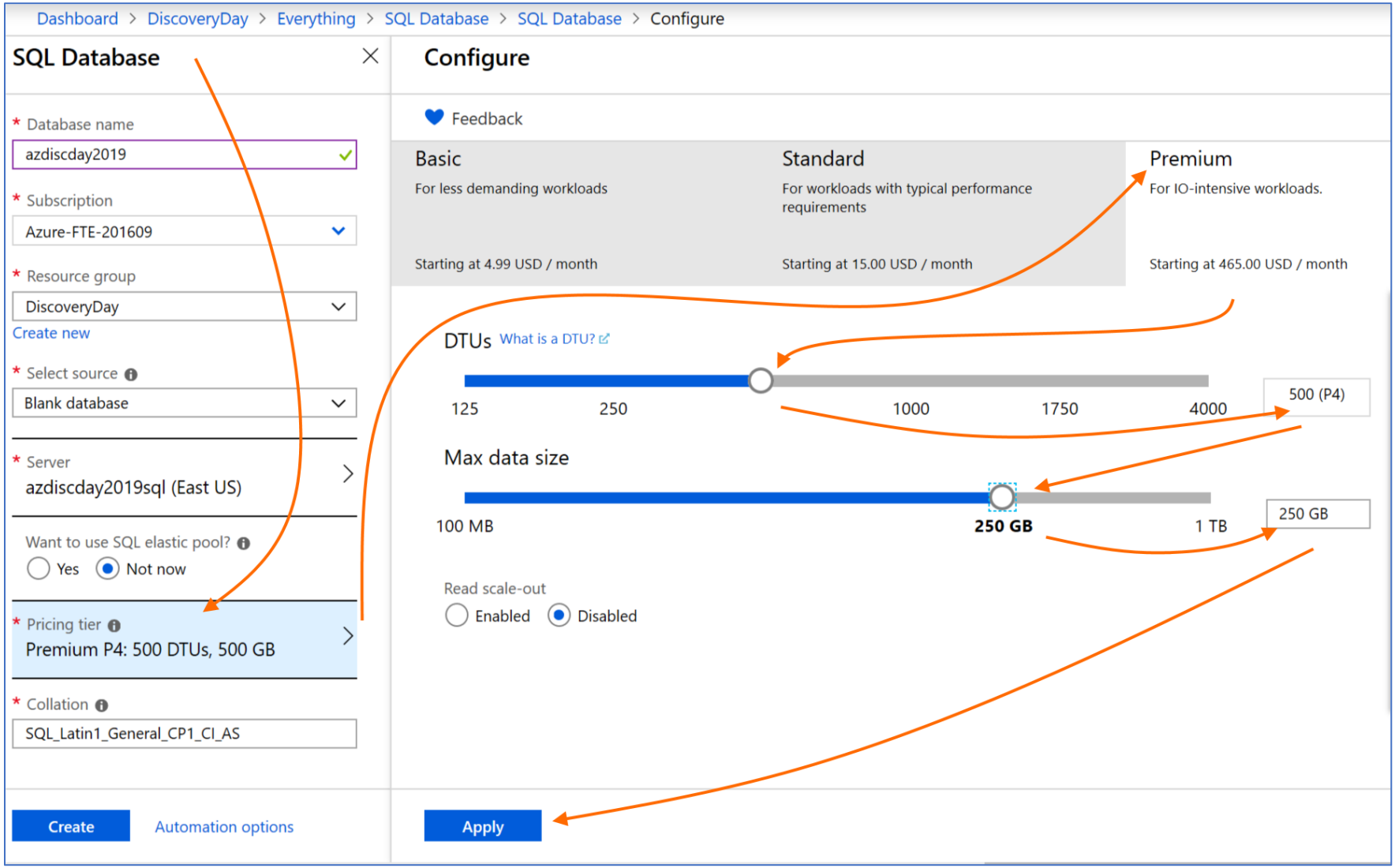
Select source as Blank Database. Then click on Server to create a new logical server.

Enter details as shown and please take note of admin user name and password you entered.

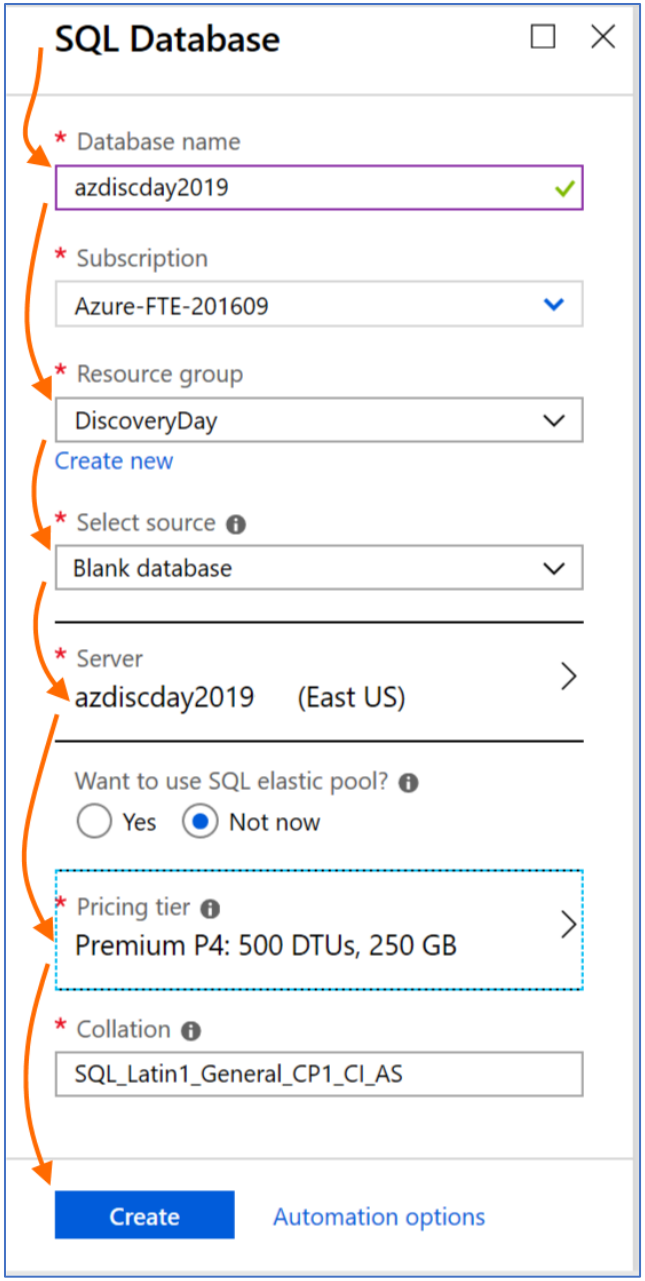

Preview
Microsoft Azure
Report a bug
Search resources, services, and docs
Home > New > Marketplace > Everything > SQL Database > SQL Database > Server > New server
Create a resource
Home
Dashboard
All services
FAVORITES
Resource groups
All resources
Recent
App Services
SQL databases
Virtual machines (classic)
Virtual machines
Images
Cloud services (classic)
Subscriptions
SQL servers
Data factories
Storage accounts
Virtual networks (classic)
Azure Active Directory
Monitor
Security Center
Cost Management + Billing
Help + support
Advisor
DevTest Labs
SQL Database
* Database name
azdiscday2019
* Subscription
Neelam's Microsoft Azure Internal Consum
* Resource group
DiscoveryDay-
Create new
* Select source
Blank database
Server
Configure required settings
Want to use SQL elastic pool?
Yes
Not now
Pricing tier
Configure required settings
* Collation
SQL_Latin1_General_CP1_CI_AS
Create
Automation options
Server
Create a new server
nsocsaegbackup
North Central US
NSOS...
nsosales
South Central US
NSOS...
New server
* Server name
azdiscday2019
.database.windows.net
* Server admin login
neelam
* Password
* Confirm password
* Location
East US
Allow Azure services to access server
Advanced Threat Protection
Start FREE Trial
Not now
FREE trial period of 30 days, and then 991.44375 INR/server/month.
Learn more
Select

Click on the pricing tier. While “S0” is the default, change the tier to “Premium” and slide the “DTUs” slider to **500 (P4)**. You can leave “Max data size” as is, or lower it to 250GB. Then click “Apply”.

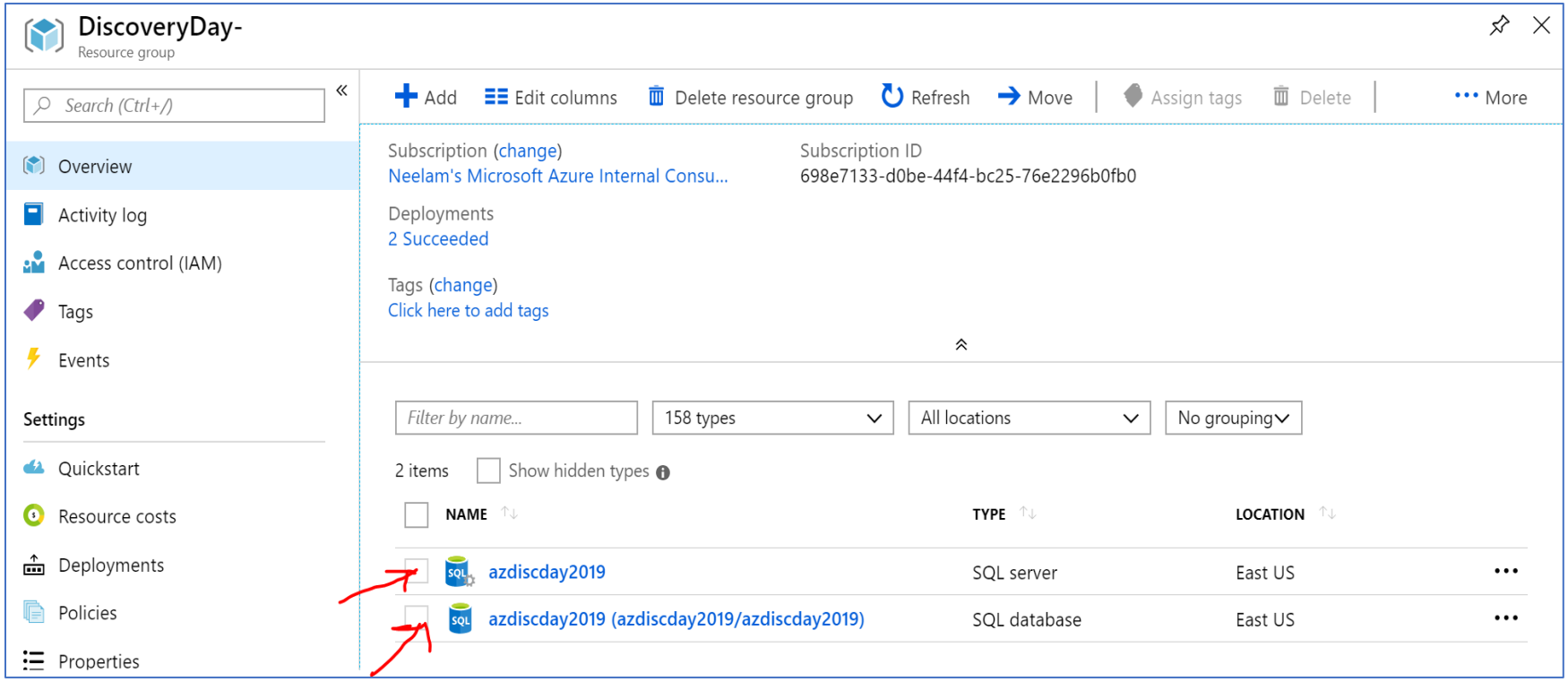
Do not accept the initial default of “S0”. A higher performance level, such as P4 shown here, is critical to completing high-volume, high-IO data ingestion jobs such as this in a reasonable amount of time (like part of a lab).



Please review the details and then “Create”. Remember – please use a higher performance tier, such as P4.

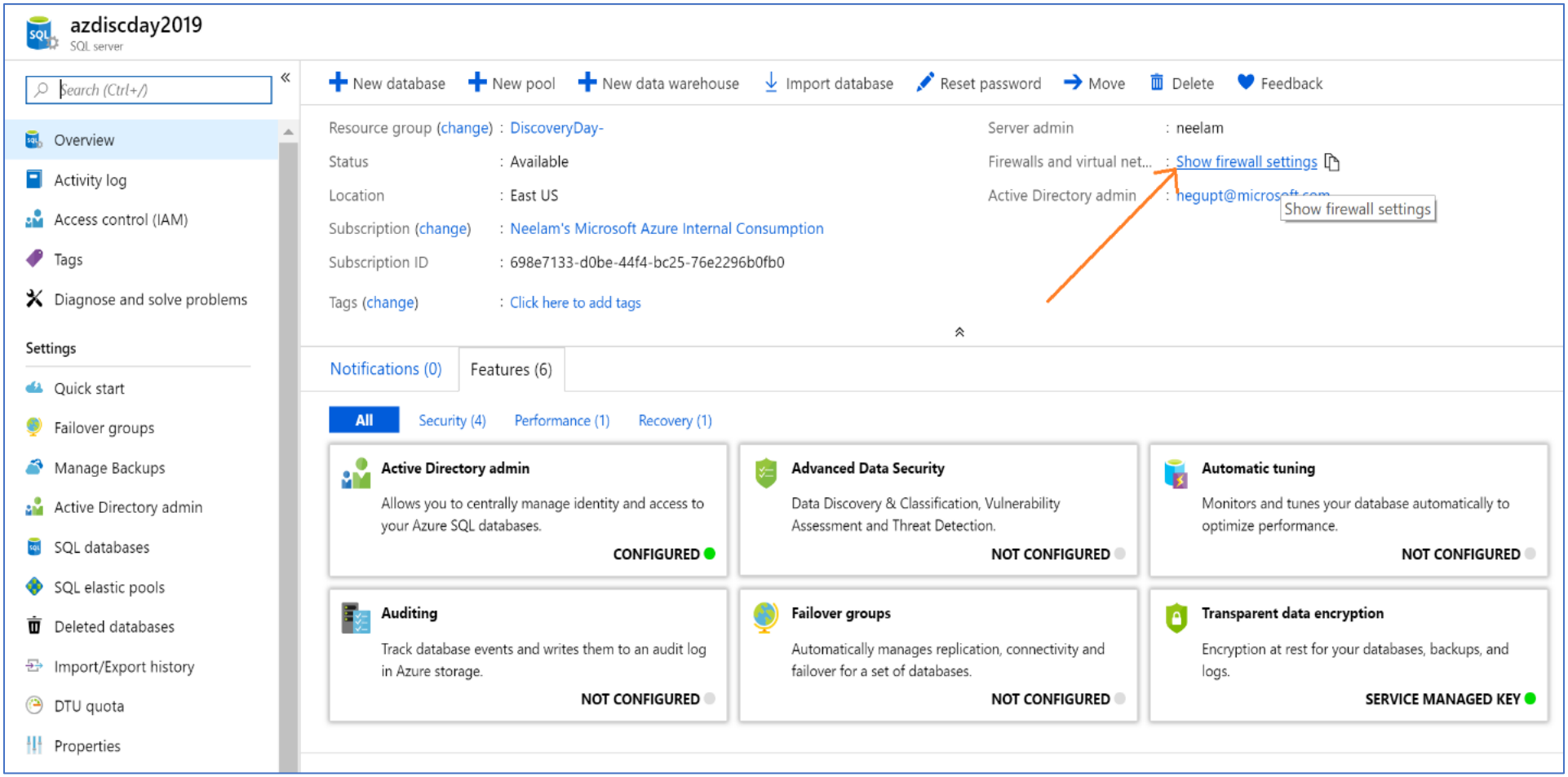


Go to your Resource Group and you will see two new components created under your resource group: Azure SQL Database and its logical SQL Server.

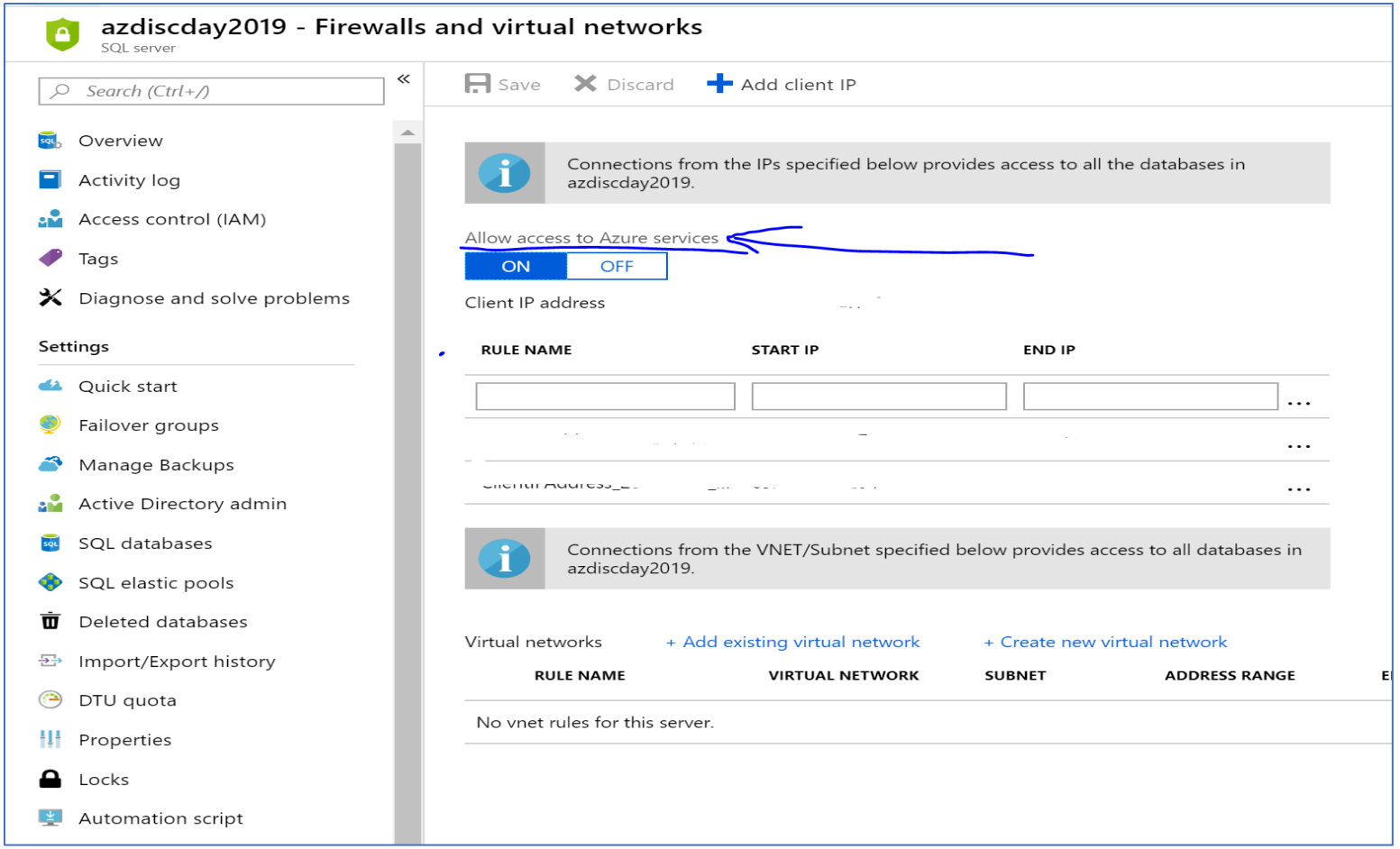


By default, all other Azure services like Azure Data Factory and Power BI etc can connect to Azure SQL Server.

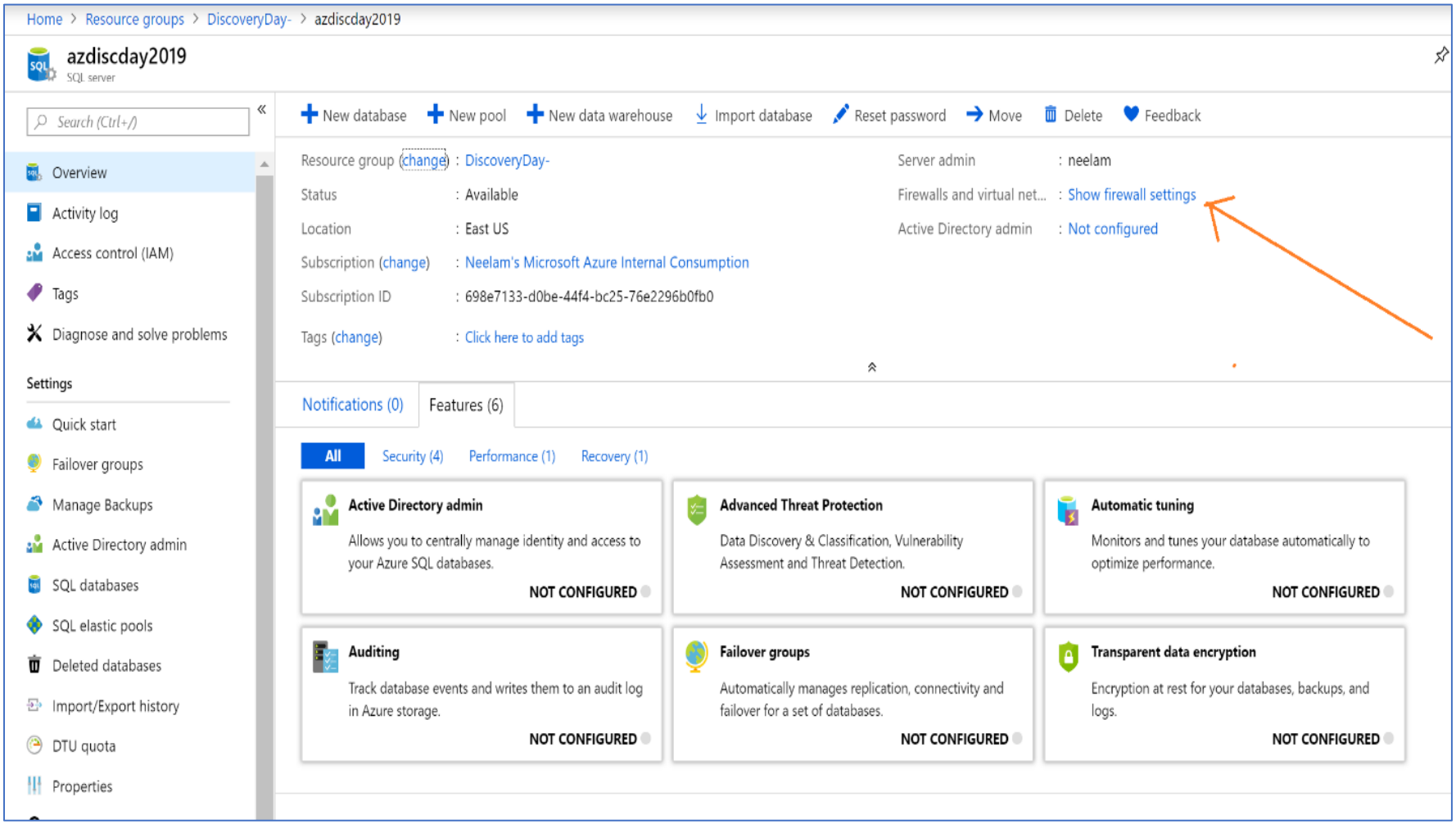
Go to Server and then click “Show firewall settings” as shown –



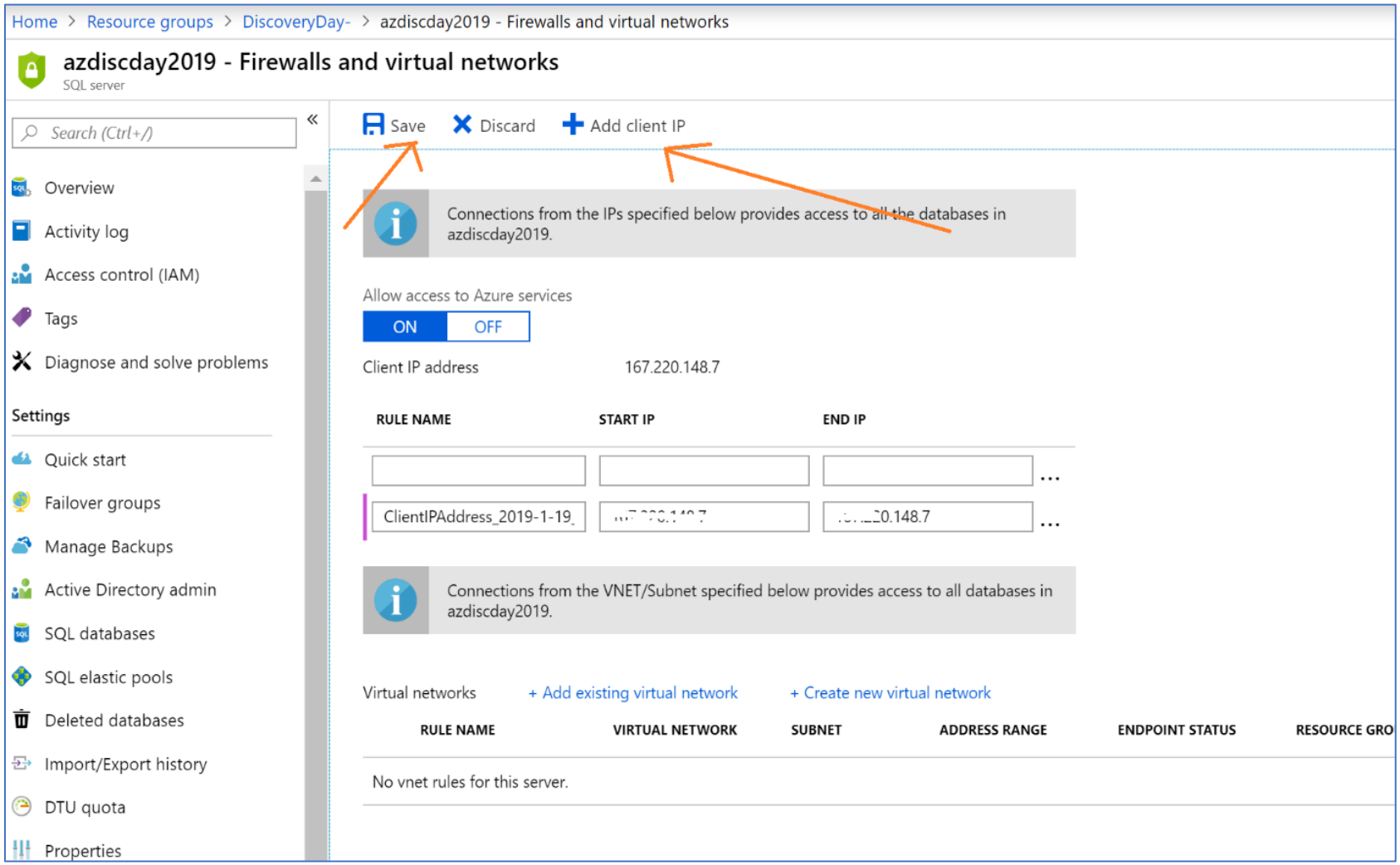
Allow access to Azure Services option is ON by default.



If you want to connect from your laptop or VM using tools like SQL Server Management Studio or Power BI Desktop, then you will need to whitelist all required IP addresses.



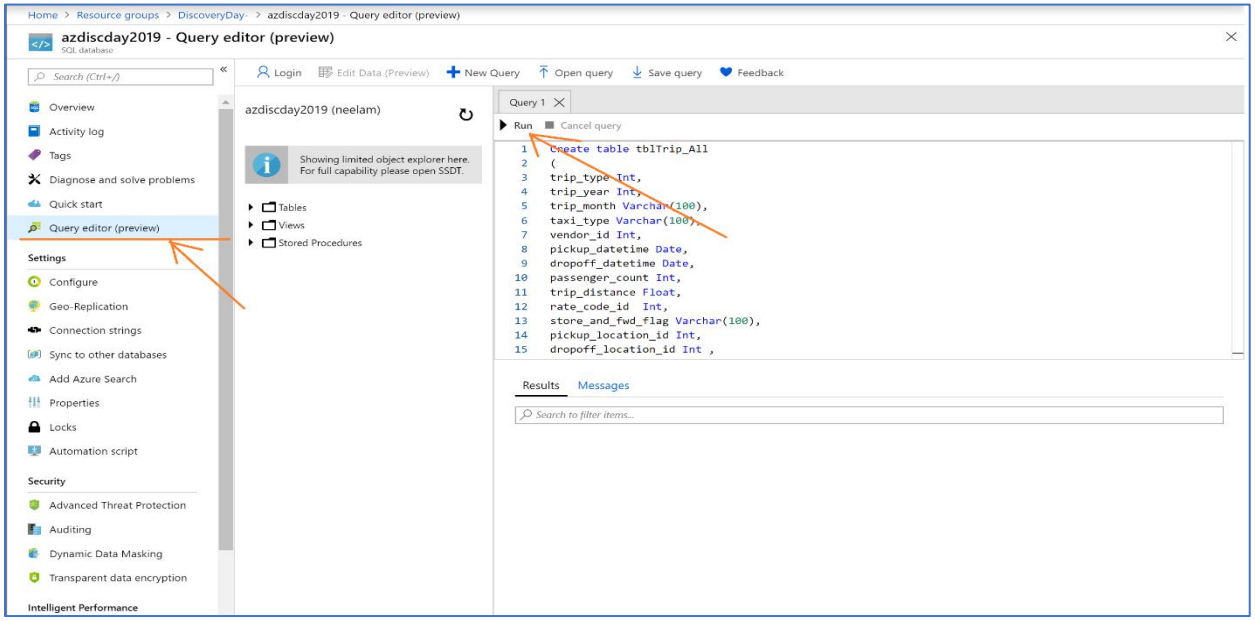
Click on Add Client IP and then click save. It will automatically whitelist your IP to use Azure SQL DB. It may take up to 5 minutes for this change to take effect. (Make sure you do this part from the machine – i.e. your laptop or the lab VM – from where you will also run SSMS or Power BI.)



Now we will connect to Azure SQL Database to execute DDL. There are two options to connect to Azure SQL Database –

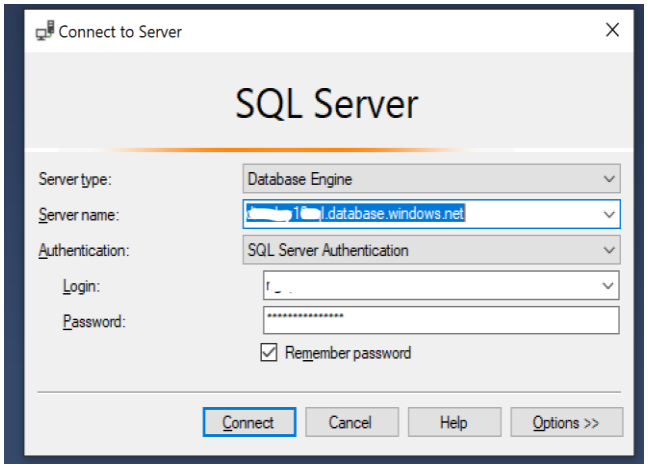
1. Using SQL query editor on the Azure portal
2. Using SQL Server Management Studio

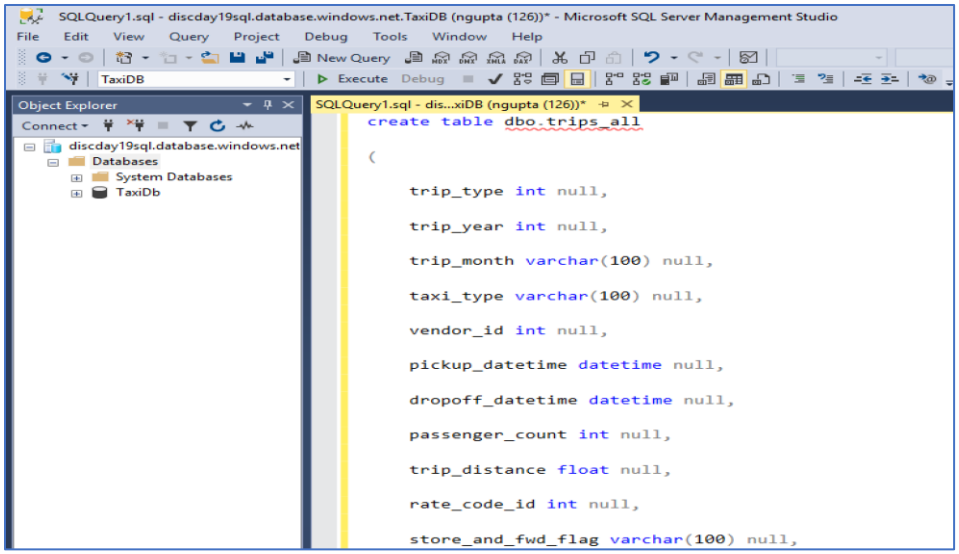
Option 1- Go to Azure SQL Database query editor and execute sql statements to create schema for the tables and view.



Option -2 - Go to SQL Server Management Studio client tool and connect to Azure SQL Database as shown –

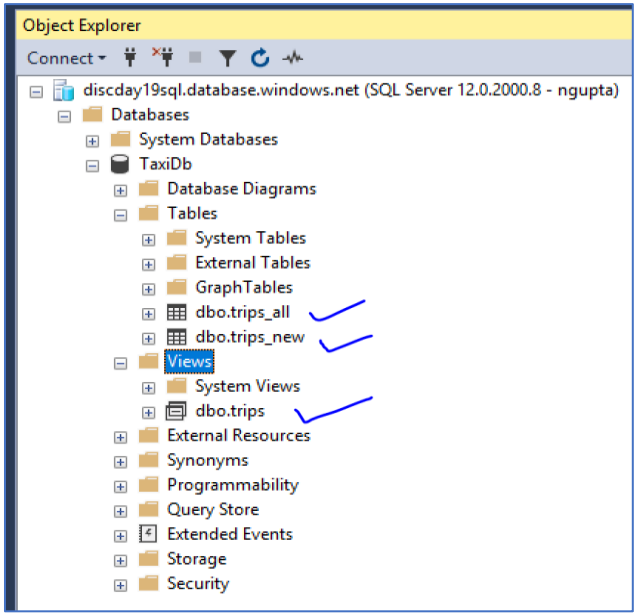
Enter Azure SQL Server Name, Login and Password.





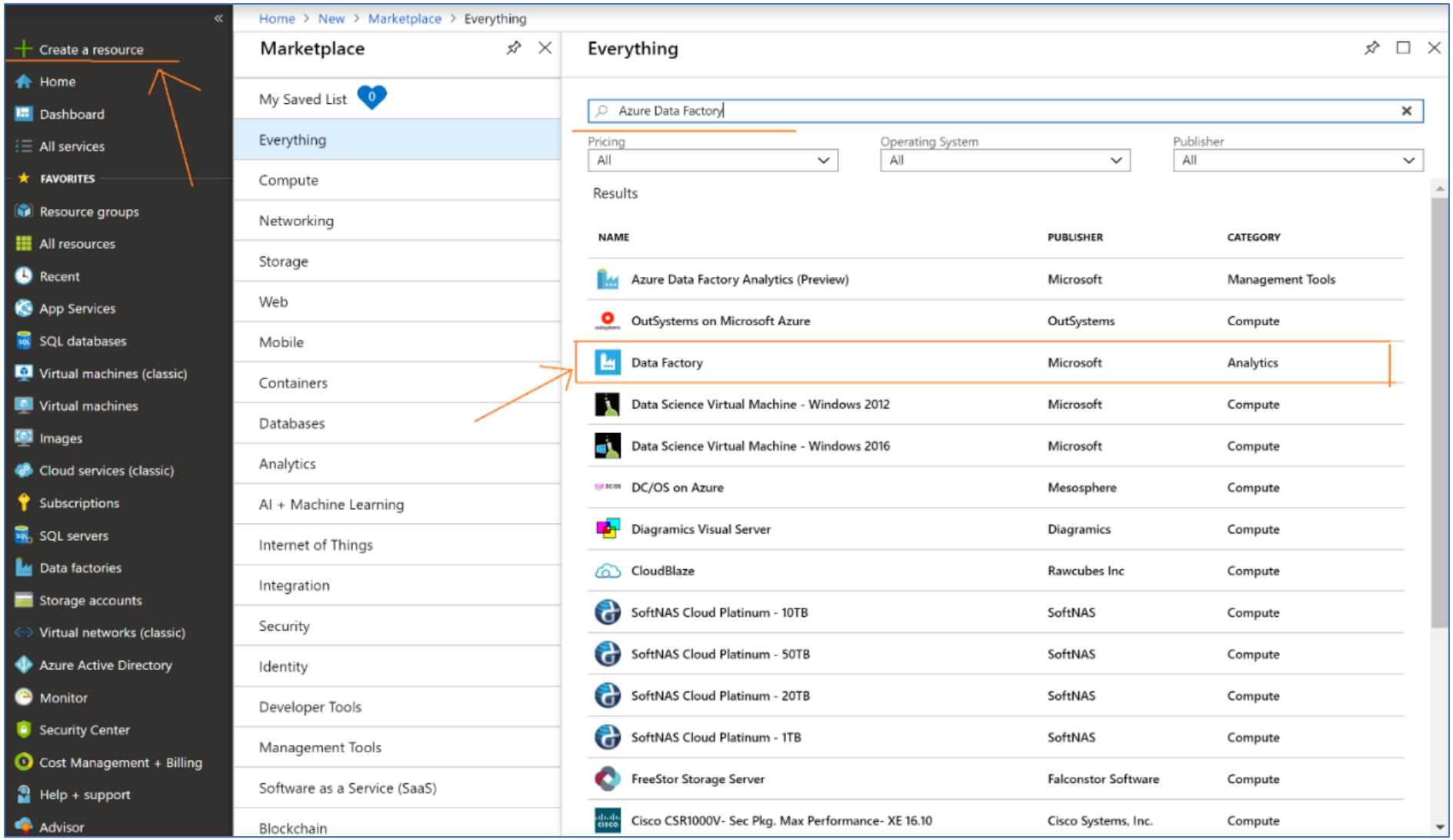
The script to create tables and a view for the taxi data is at: https://raw.githubusercontent.com/plzm/azure-discoveryday2019-mdw/master/labs/lab2/lab2.sql

When the script gets executed successfully, right-click the “Databases” node and select “Refresh”. You will see three objects under your database-



Task 2 – Create Azure Data Factory Project

In the Azure portal, click “+ Create a resource” and search for “Azure Data Factory”. (You can also click “+ Add” in your Resource Group.)



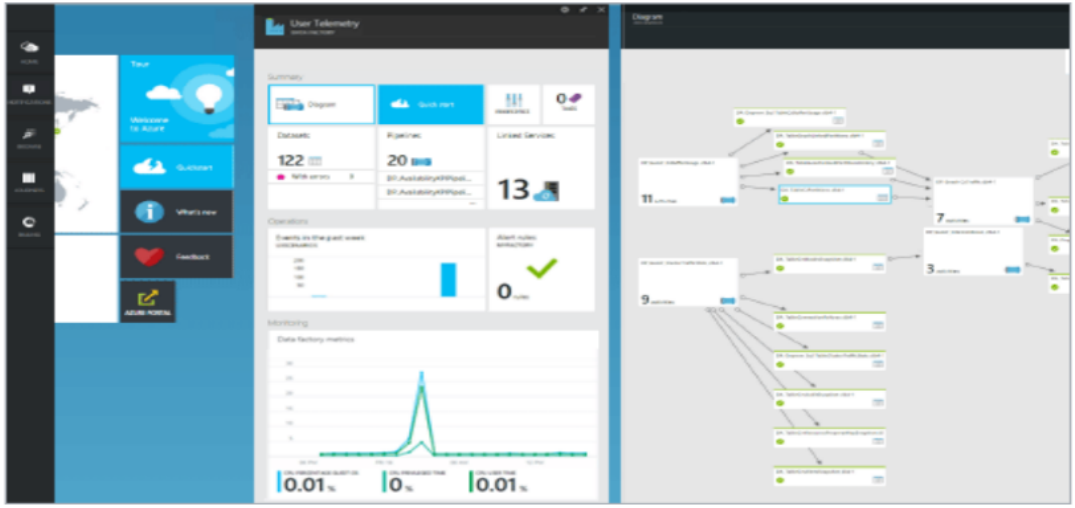
Data Factory

Microsoft

Microsoft Azure Data Factory is a cloud-based data integration service that automates the movement and transformation of data. You can quickly create, deploy, schedule, and monitor highly-available, fault tolerant data flow pipelines. Move and transform data of all shapes and sizes, and deliver the results to a range of destination storage services. Monitor all of your data pipelines and service health at a glance with a rich visual experience. Easily consume the data produced with BI, analytics tools, and other applications to drive key business insights and decisions.

- Compose data storage, movement and processing services into data flow pipelines
- Enhanced HDInsight integration including HCAT and on-demand cluster management
- Schedule data pipelines with fine-tuned control
- New data connectors for on-premises and cloud data sources
- Integration with Azure Machine Learning and Azure Batch
- Globally deployed data movement as a service
- Create, edit and deploy data pipelines with a Visual Studio plug-in

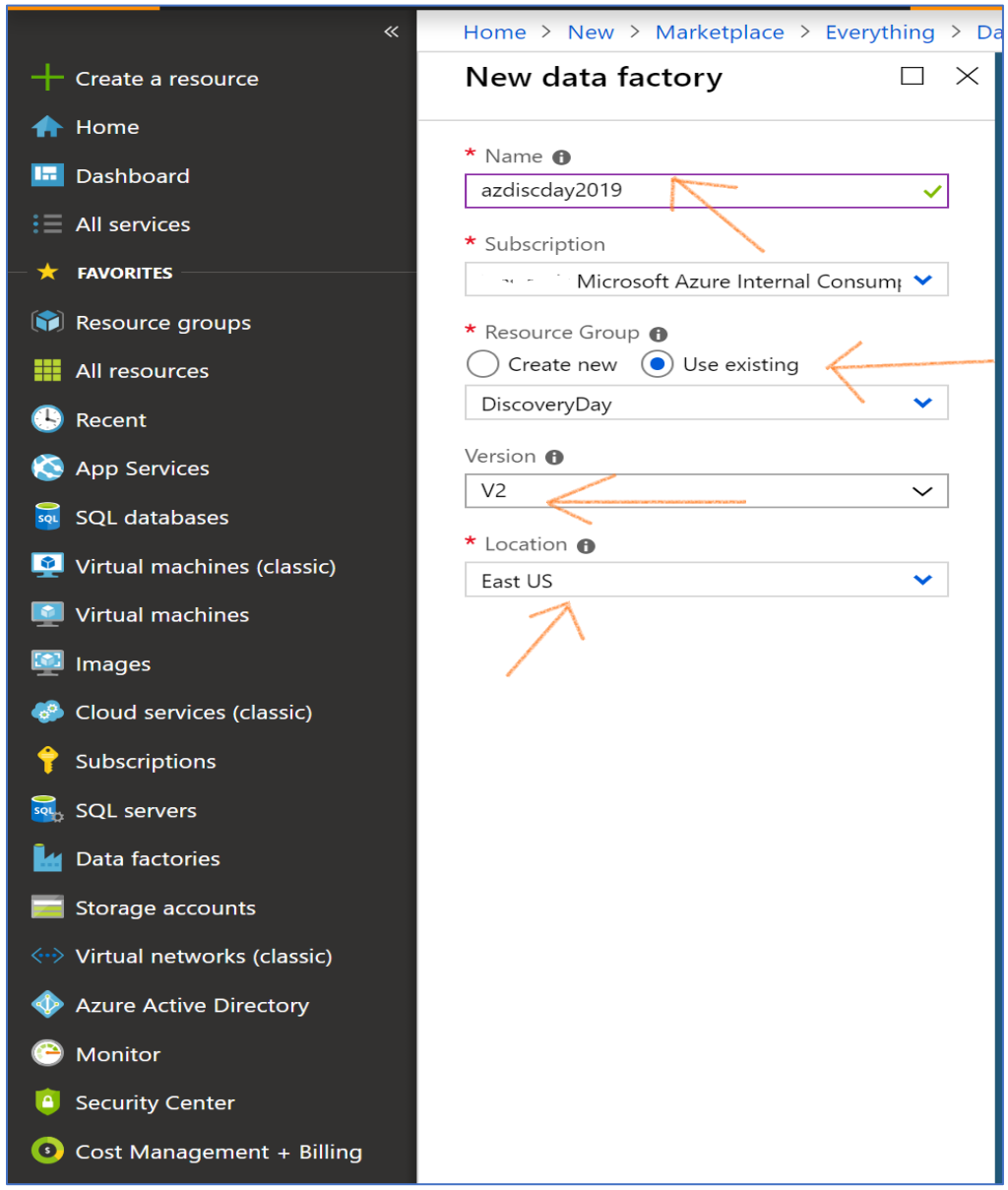
Save for later

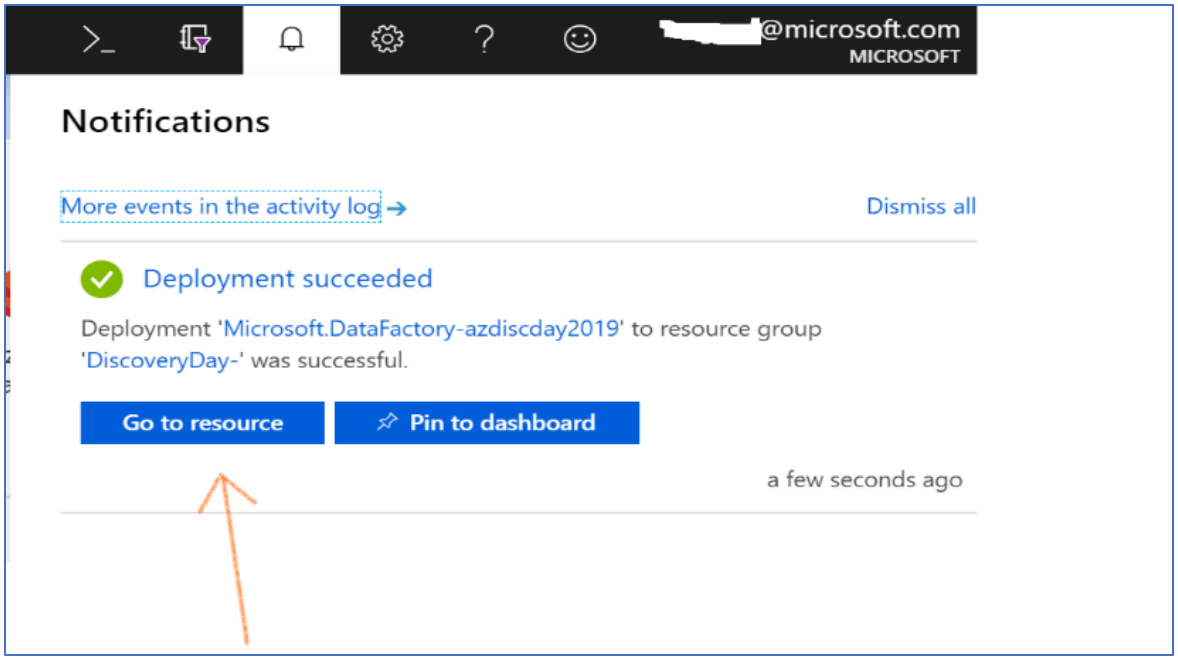


PUBLISHER	Microsoft
USEFUL LINKS	Documentation Service overview Pricing details

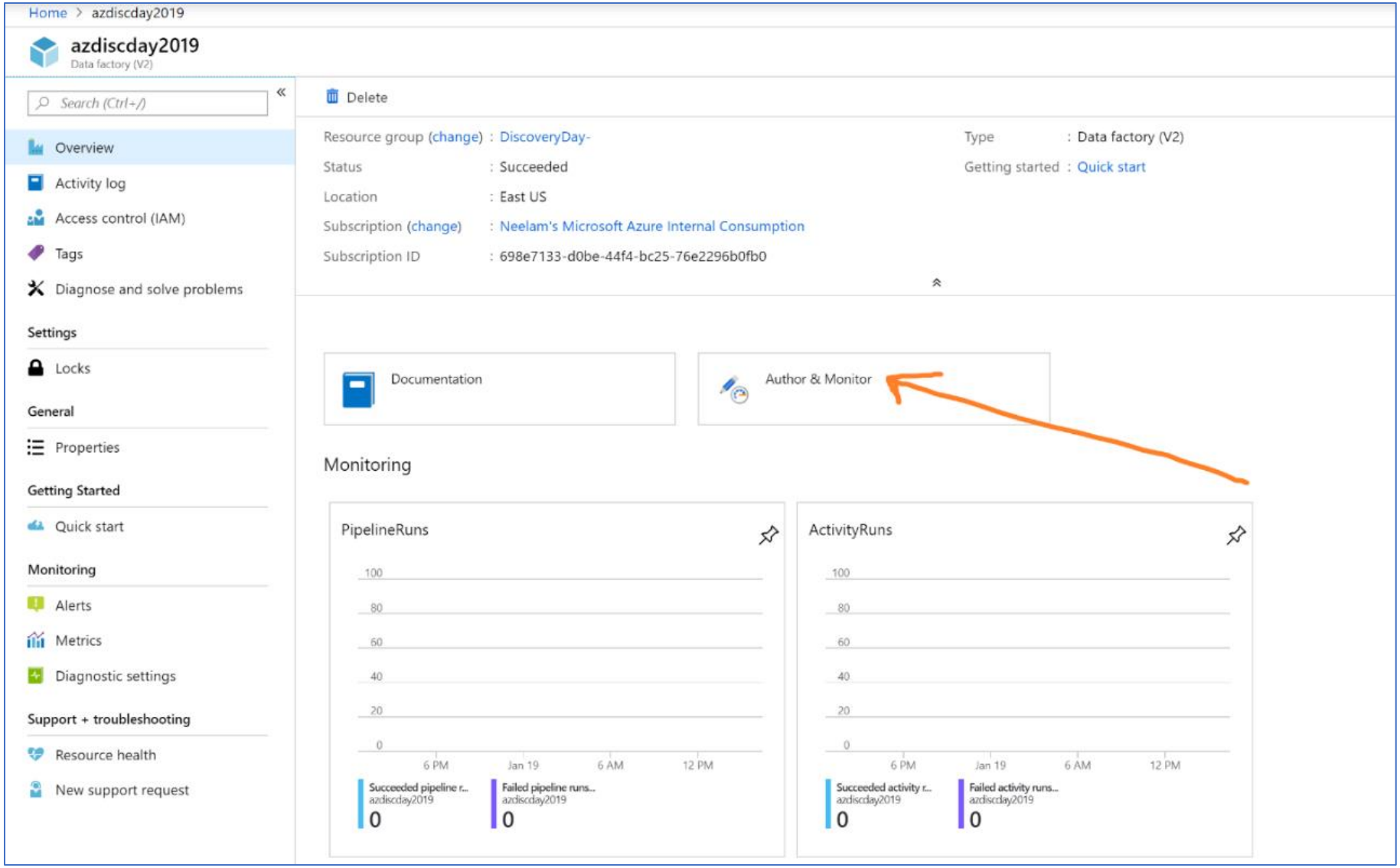
Create

Please enter Data Factory Name, Select Azure subscription and then select your Resource Group. Set “Version” to “V2”, and select the same Azure region you have been using so far. Then click “Create” and wait for the success notification.



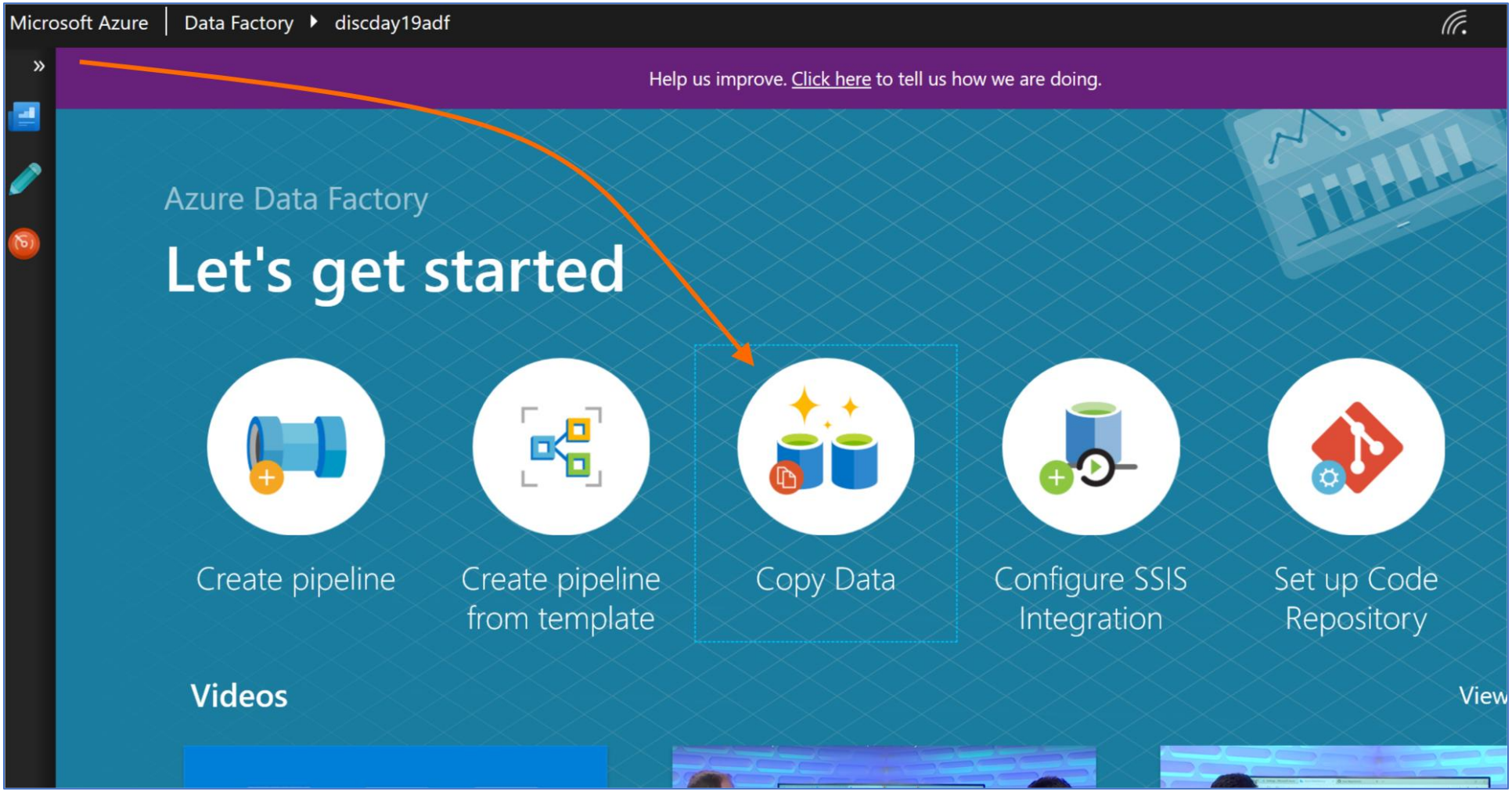

[redacted]@microsoft.com
MICROSOFT
Notifications
More events in the activity log
Dismiss all
Deployment succeeded
Deployment 'Microsoft.DataFactory-azdiscday2019' to resource group 'DiscoveryDay-' was successful.
Go to resource
Pin to dashboard
a few seconds ago

When you click “Go to resource”, it will take you to the Data Factory project as shown – on this screen, click Author & Monitor.



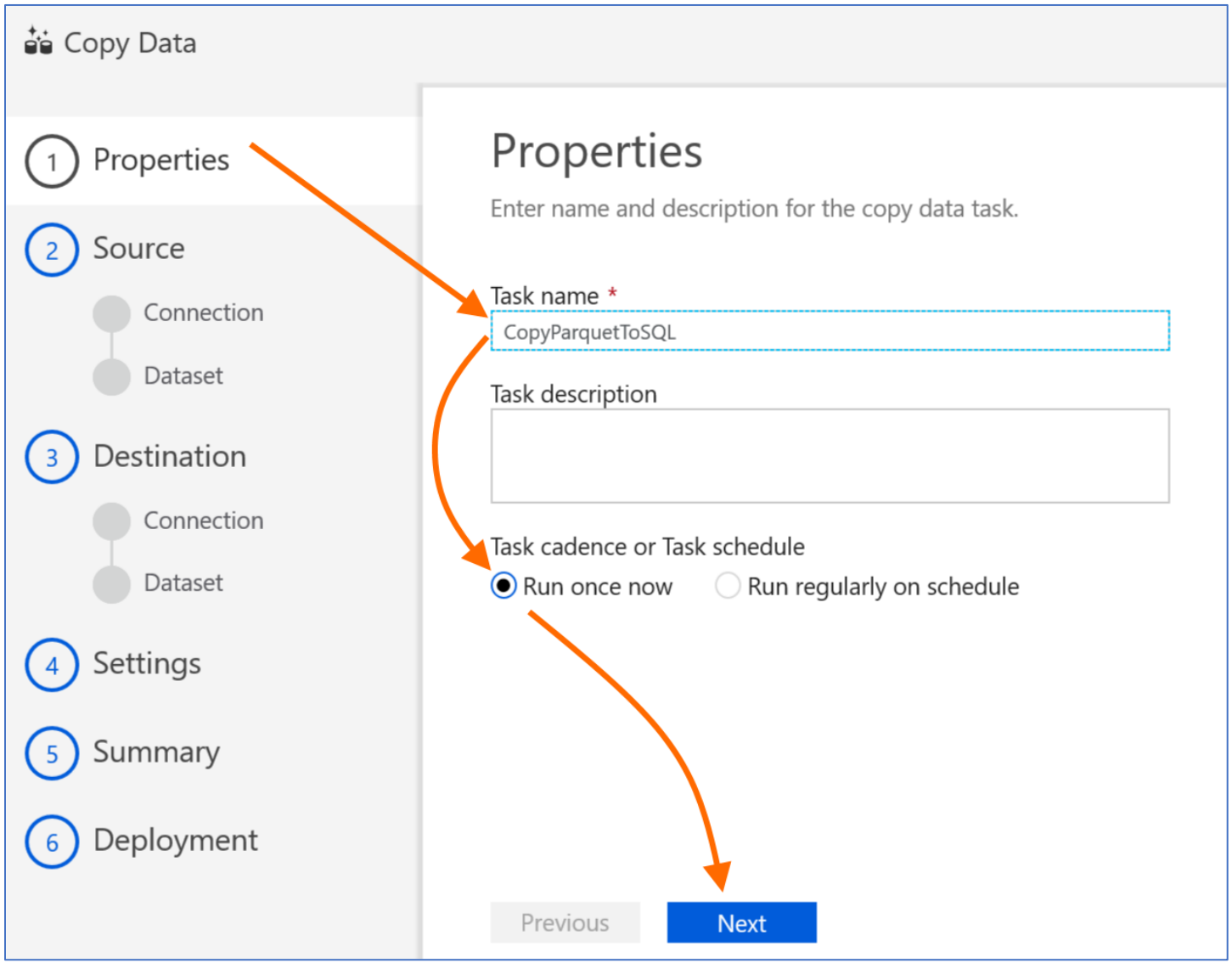
There are two ways to create and edit Azure Data Factory pipelines. There's a "Copy Data" wizard, which is very effective to build simple data copying pipelines, and this is what you will use for this lab. The other options ("Create pipeline" and "Create pipeline from template") are more powerful and flexible, and you would use those to build orchestrations more complex than the kind of file copying you'll do in this lab.

To get started, click "Copy Data".



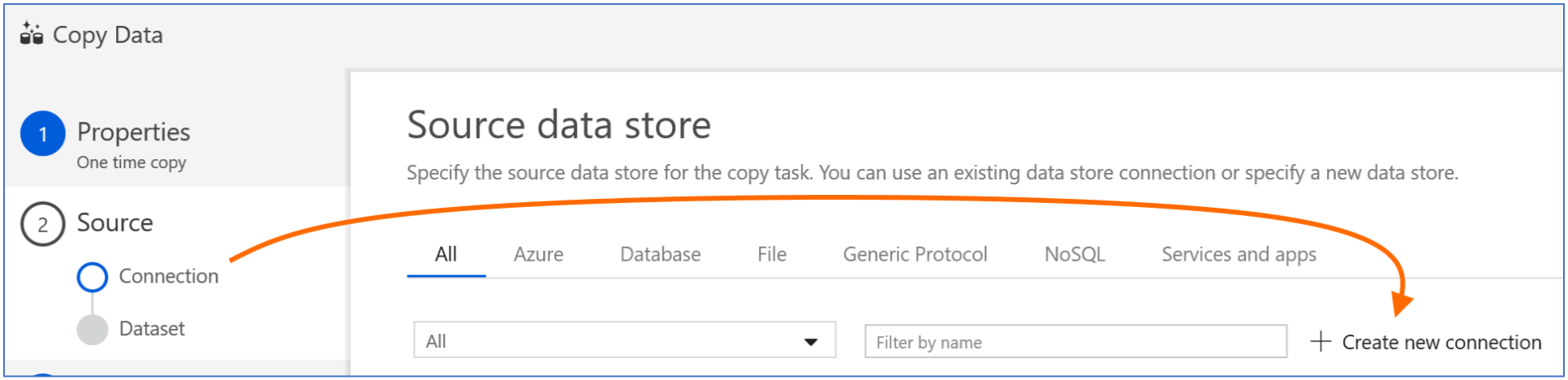
Enter Task Name and then select “Run once now” for this lab. Then click “Next”.

(There is also an option to create a recurring run schedule, but for this lab we’ll run the pipeline just once. You can still manually re-run the pipeline if needed.

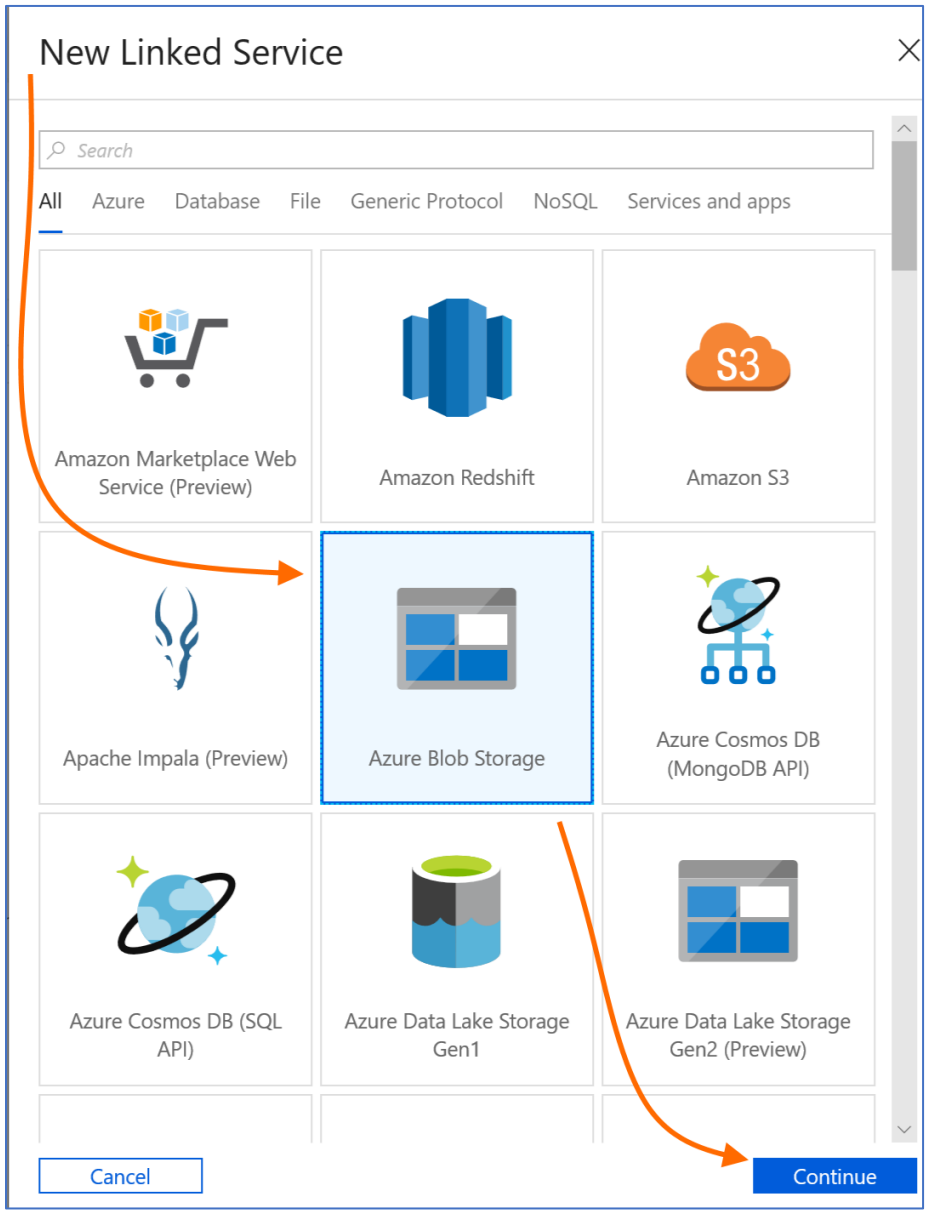


In this lab, we are going to copy data from Azure blob storage (the Parquet data you created at the end of lab 1) to Azure SQL DB.

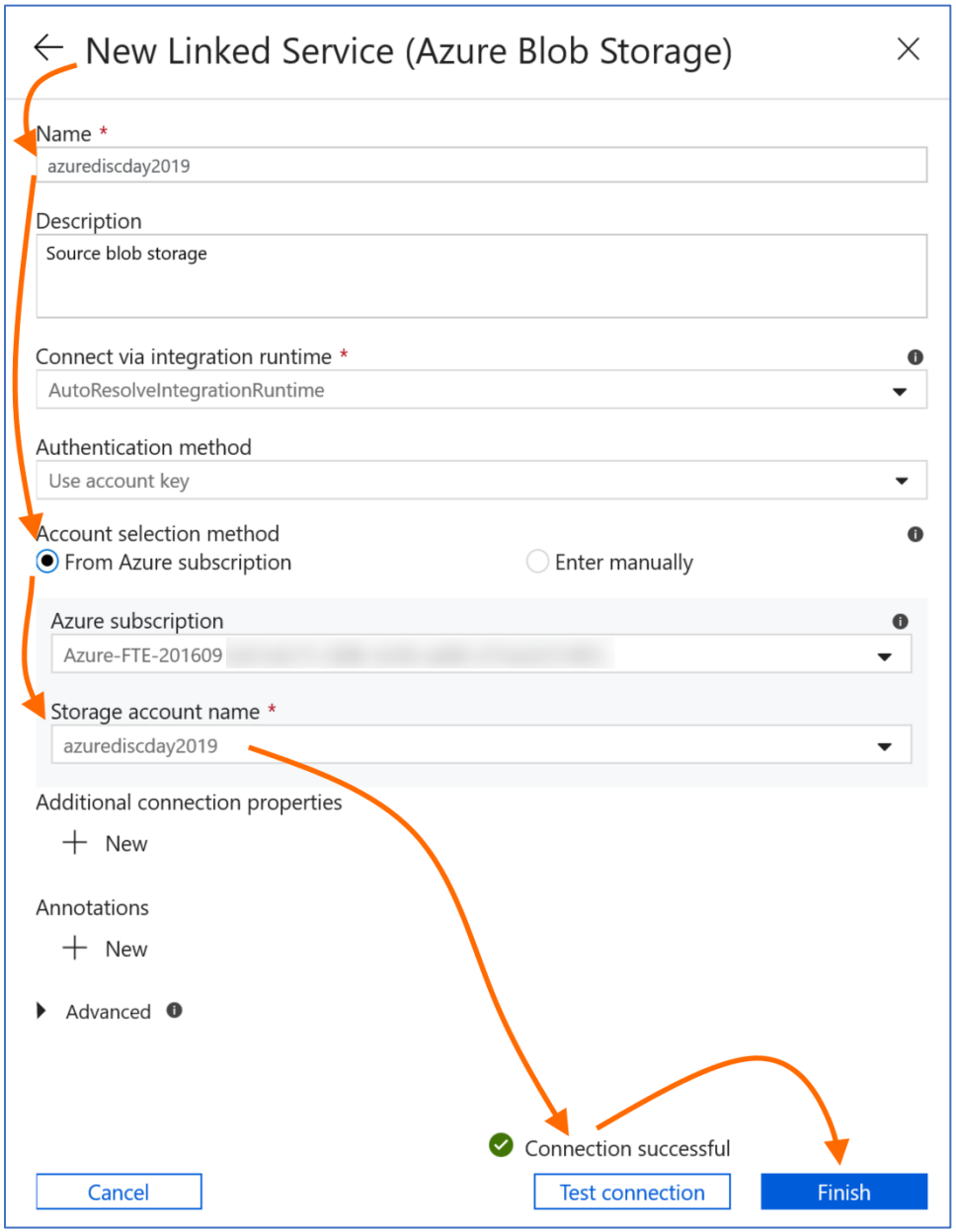
First, let's create a connection to our source data. Click "+ Create new connection".



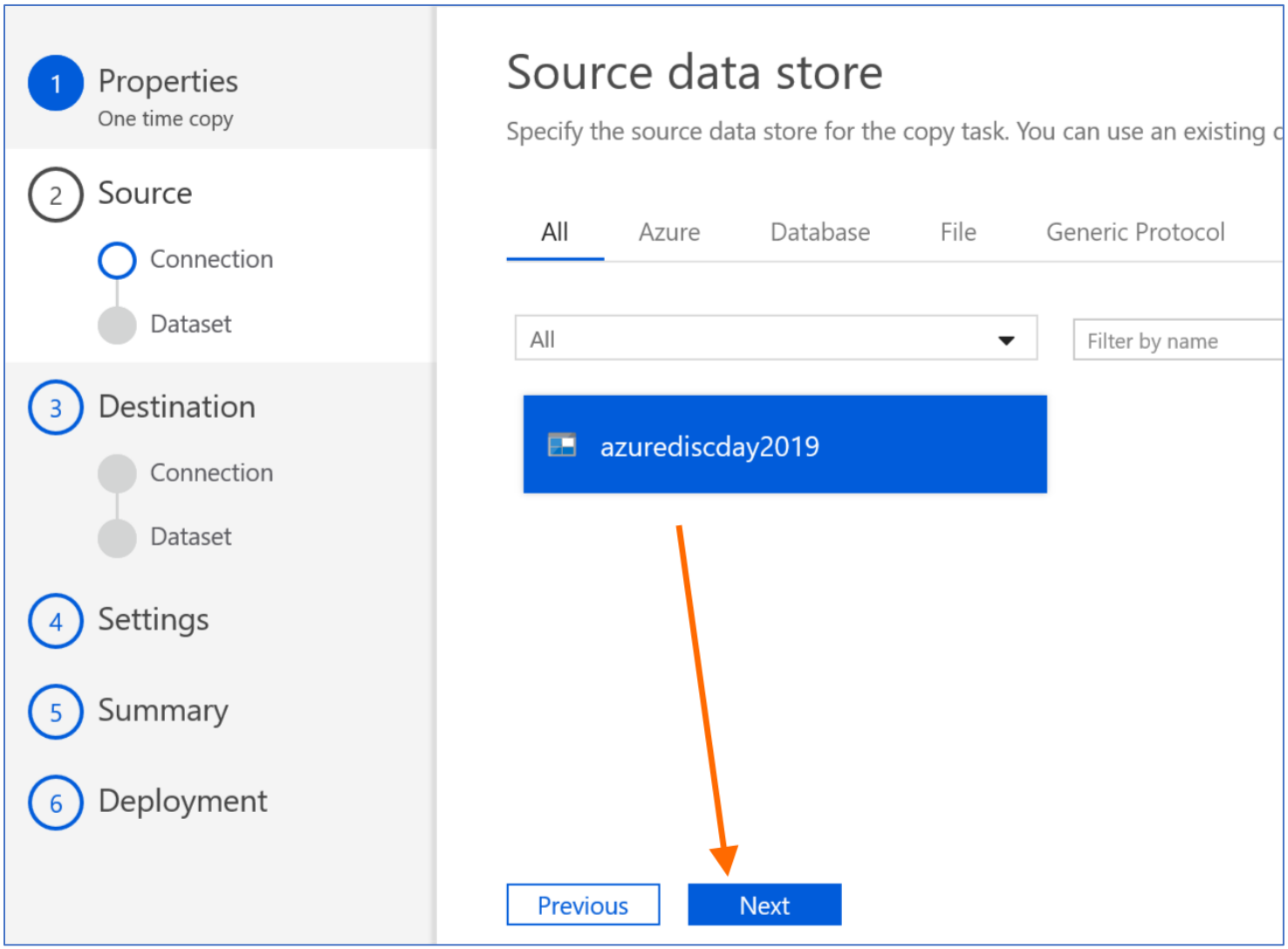
Next, select “Azure Blob Storage”, then click “Continue”.



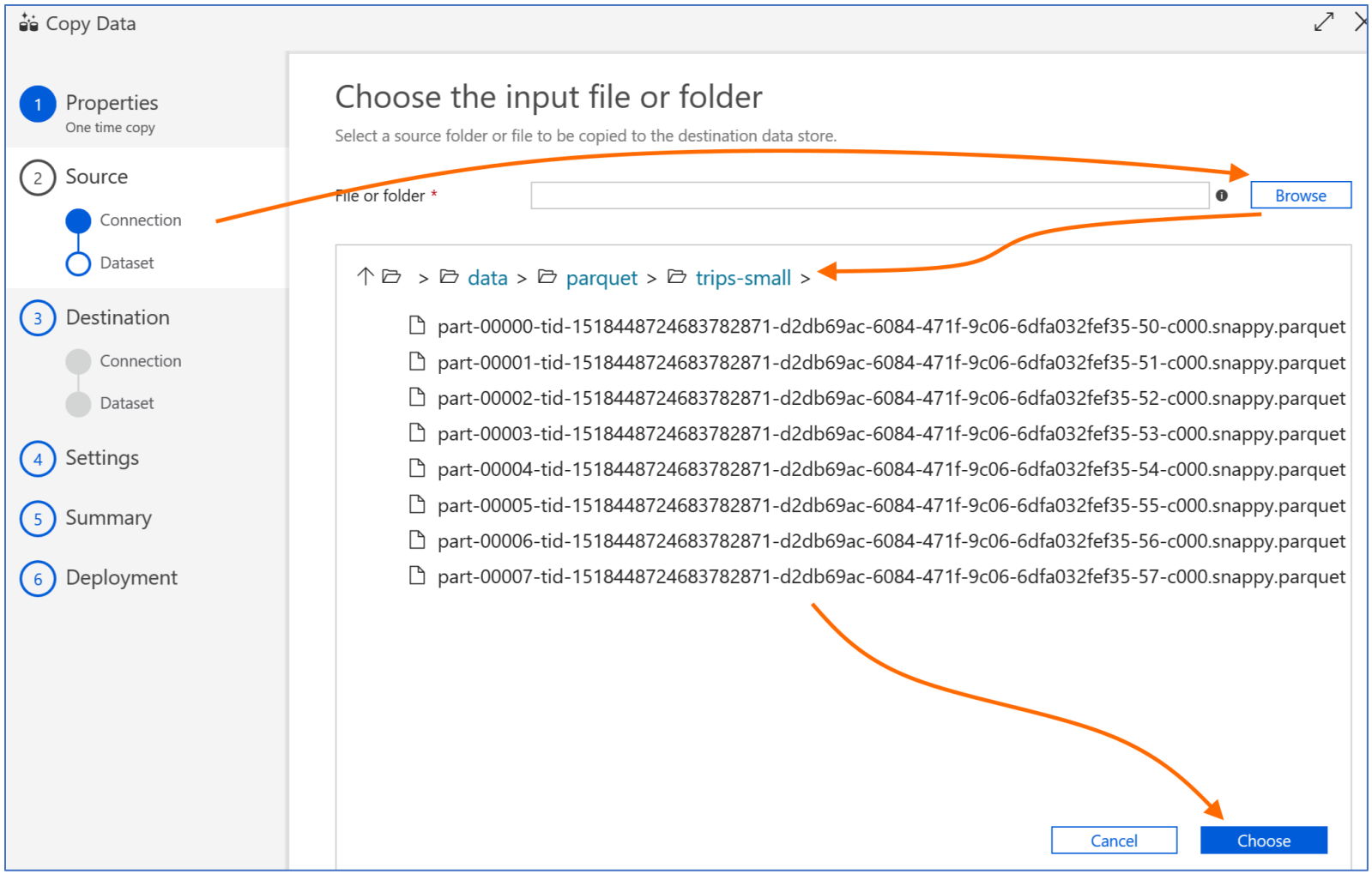
Name the data source (be specific, so you'll know what it is in a list of data sources – here, we're using the storage account name). Select your Azure subscription and storage account. Optionally, click "Test connection" and ensure the connection is successful. Then click "Finish".



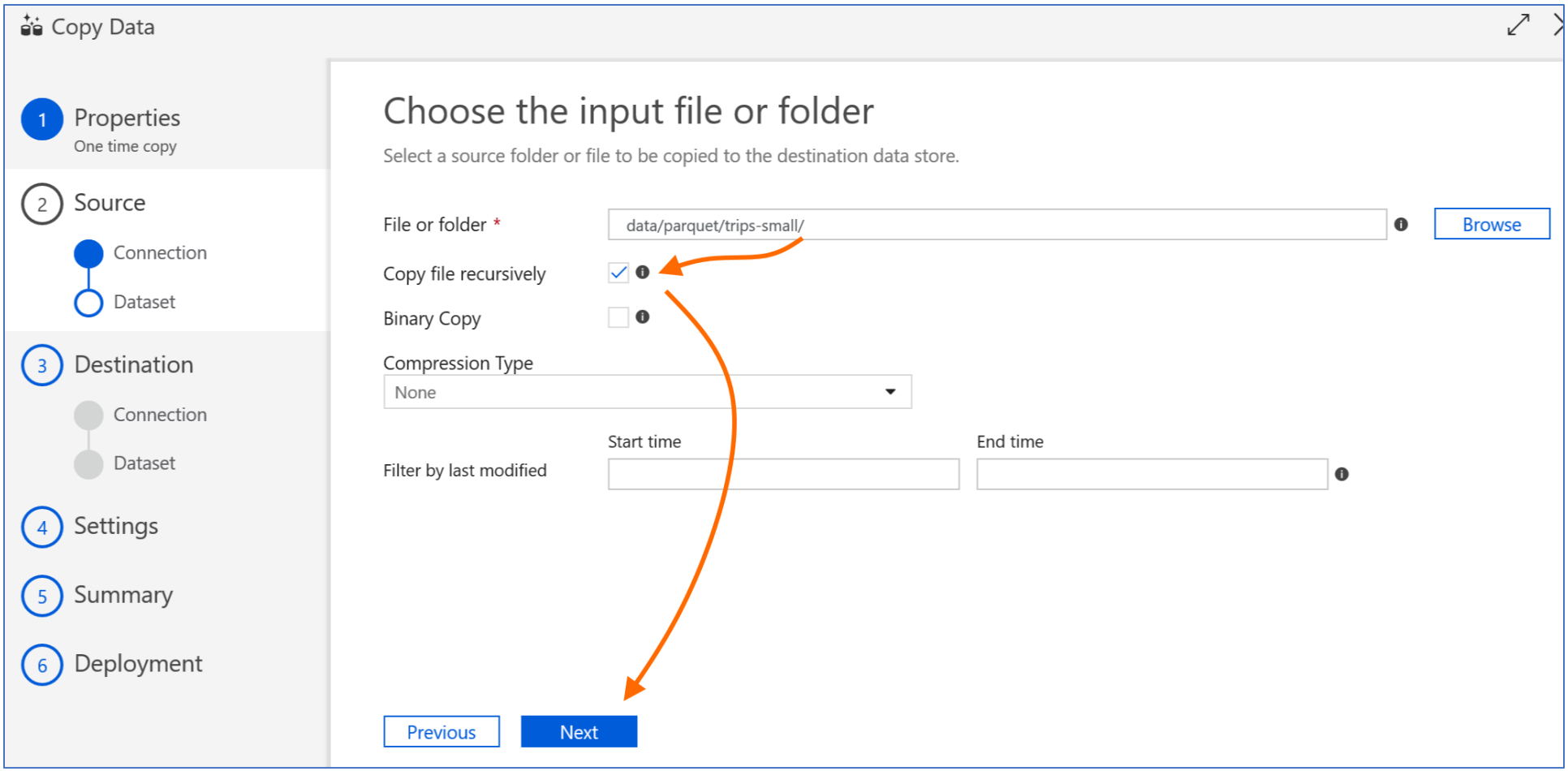
Now that we have our Azure blob storage source, we need to select the specific files. Click “Next” to continue in the ADF Copy wizard.



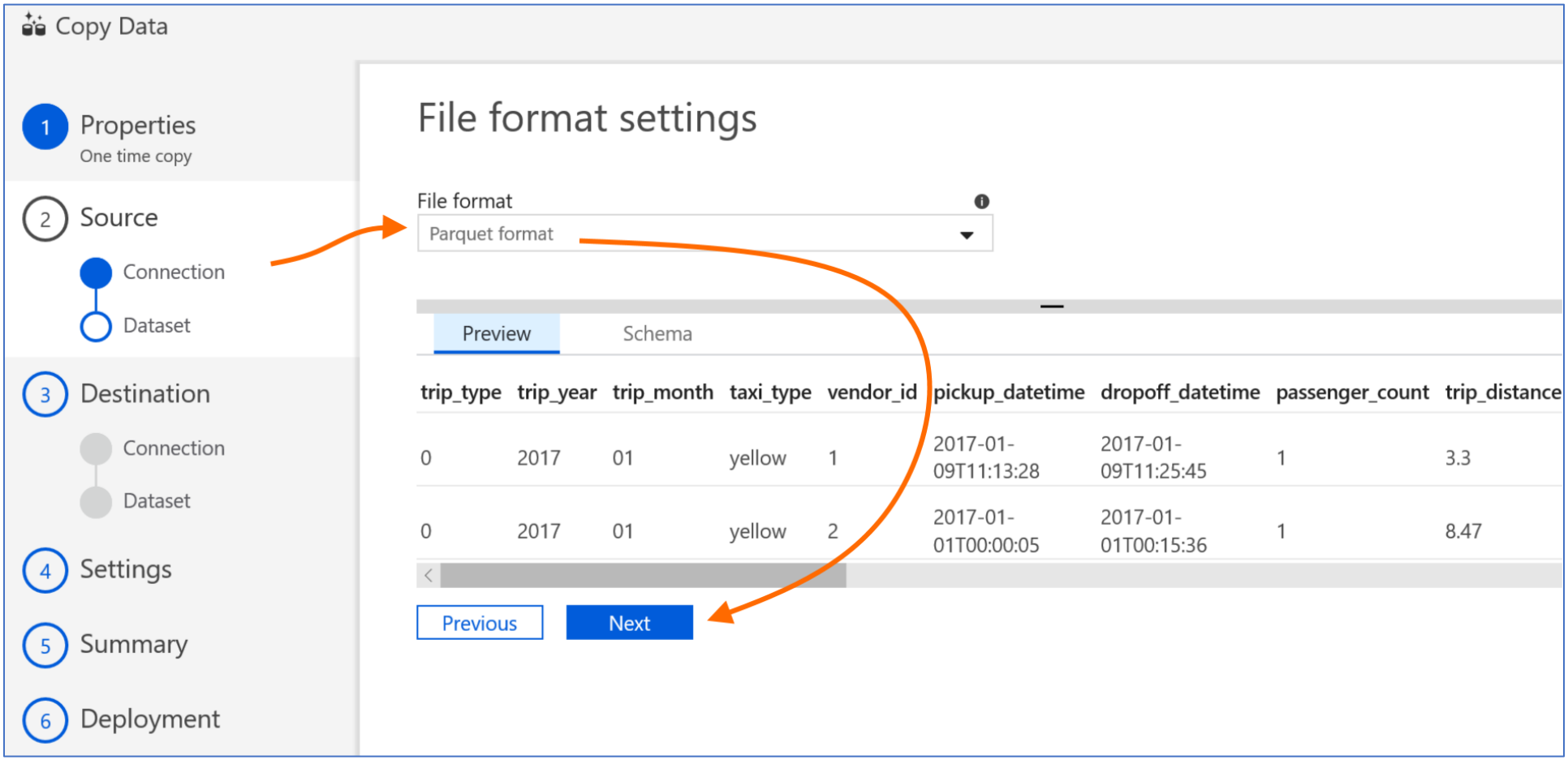
Click “Browse”, then use the folder list to navigate into the folder where your Parquet files were written at the end of lab 1. Then click “Choose”.



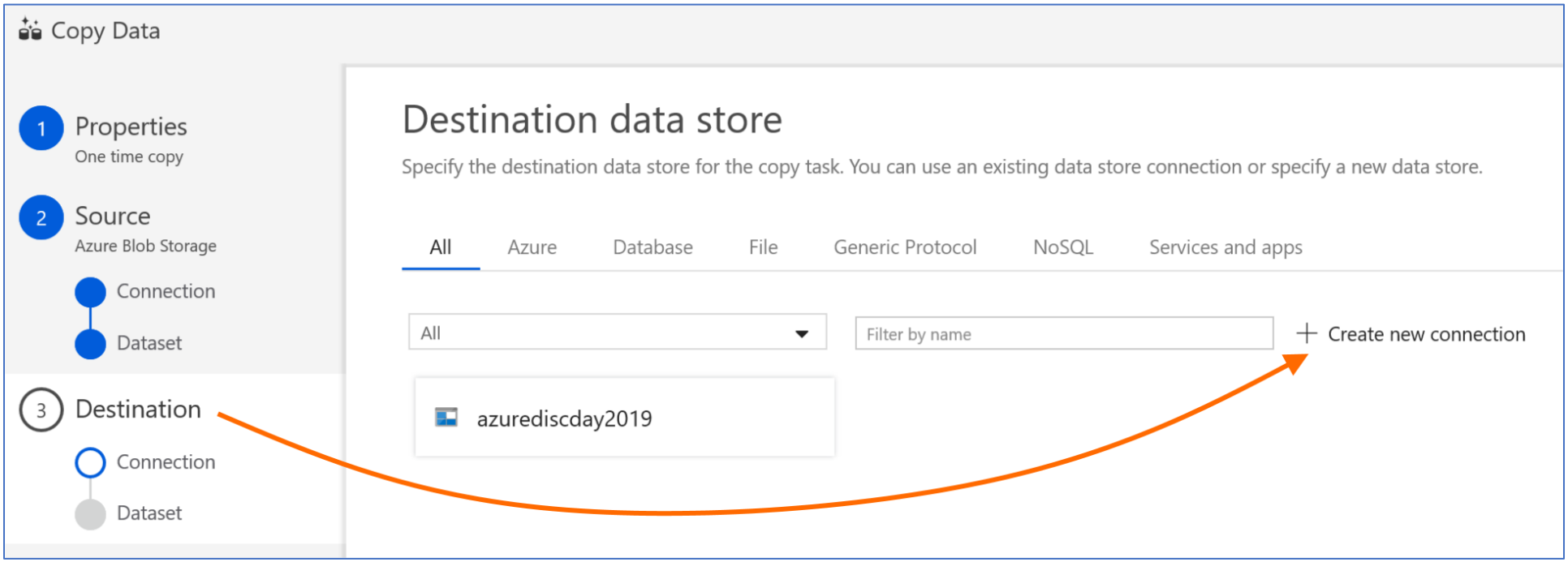
Check “Copy file recursively” (since our Parquet data is spread across multiple files and we want to read all files in the selected folder), then click “Next”.



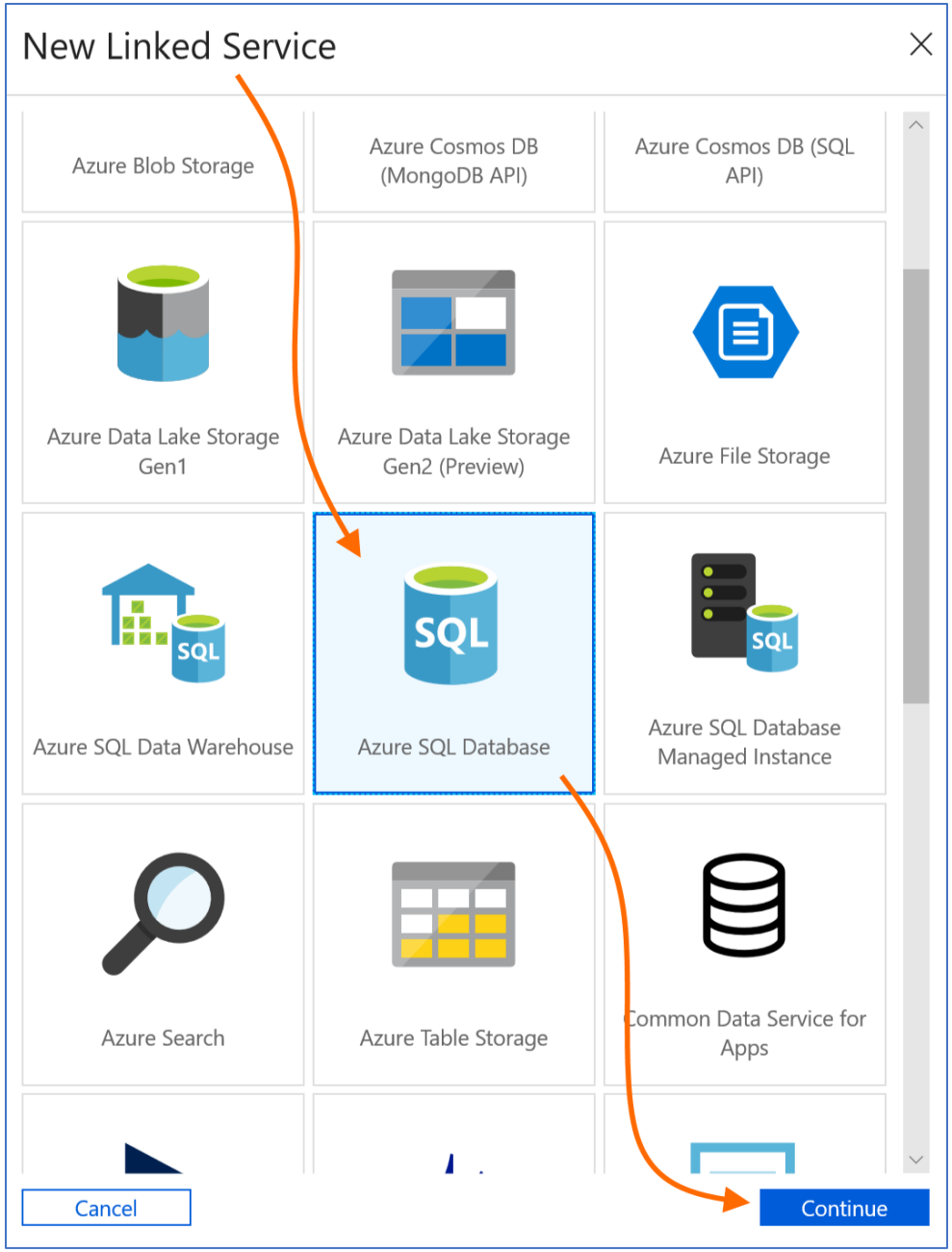
Set “File format” to “Parquet format”. The Preview tab automatically shows a preview of the data for a quick visual check. Optionally, you can also click the “Schema” tab to validate the file schema compared to the files you worked with in lab 1. Then, click “Next”.



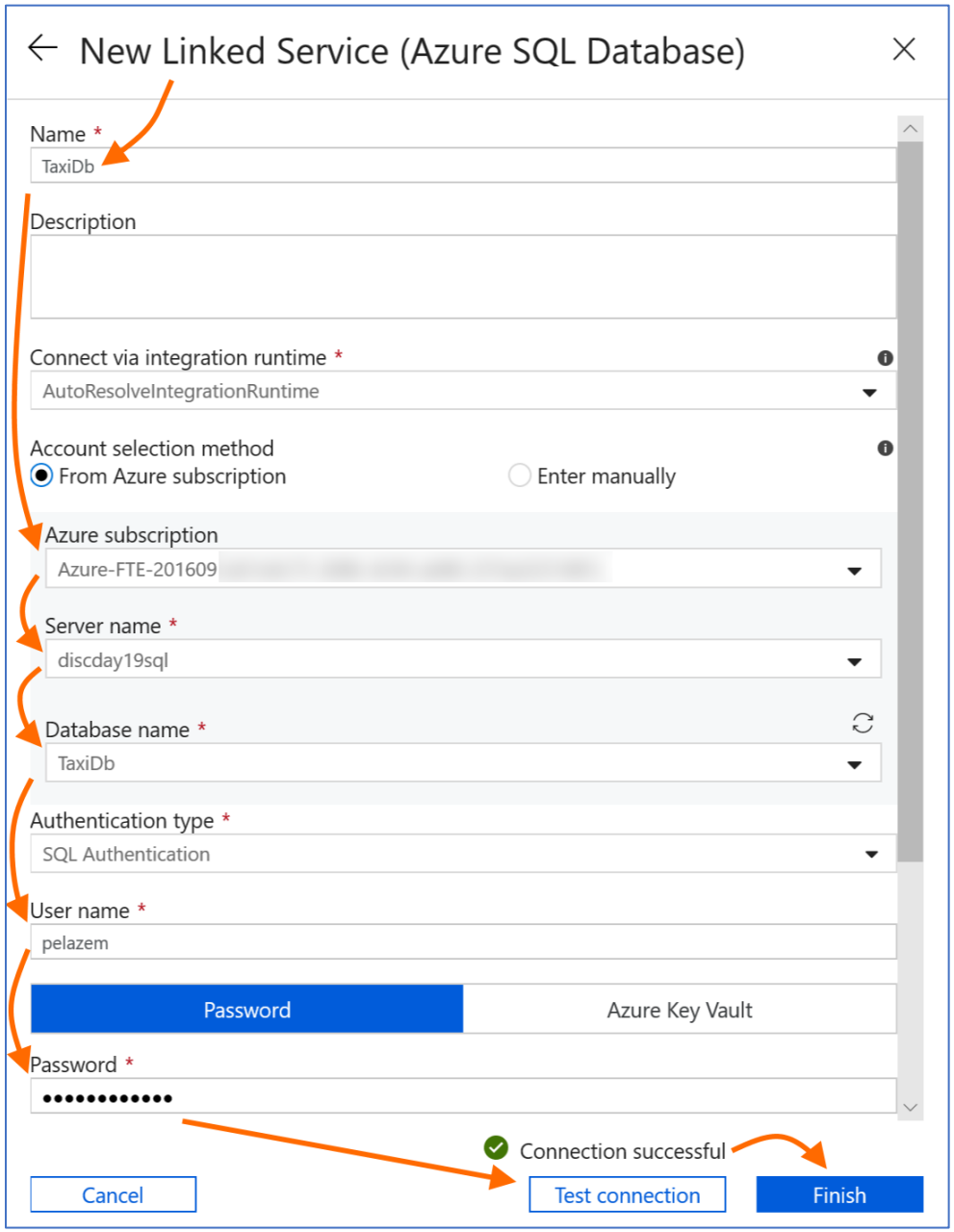
Next, you need to configure a destination, which is the Azure SQL Database you created earlier in this lab. In the “Destination” stage, click “+ Create new connection”, similarly to how you created a source data connection previously.



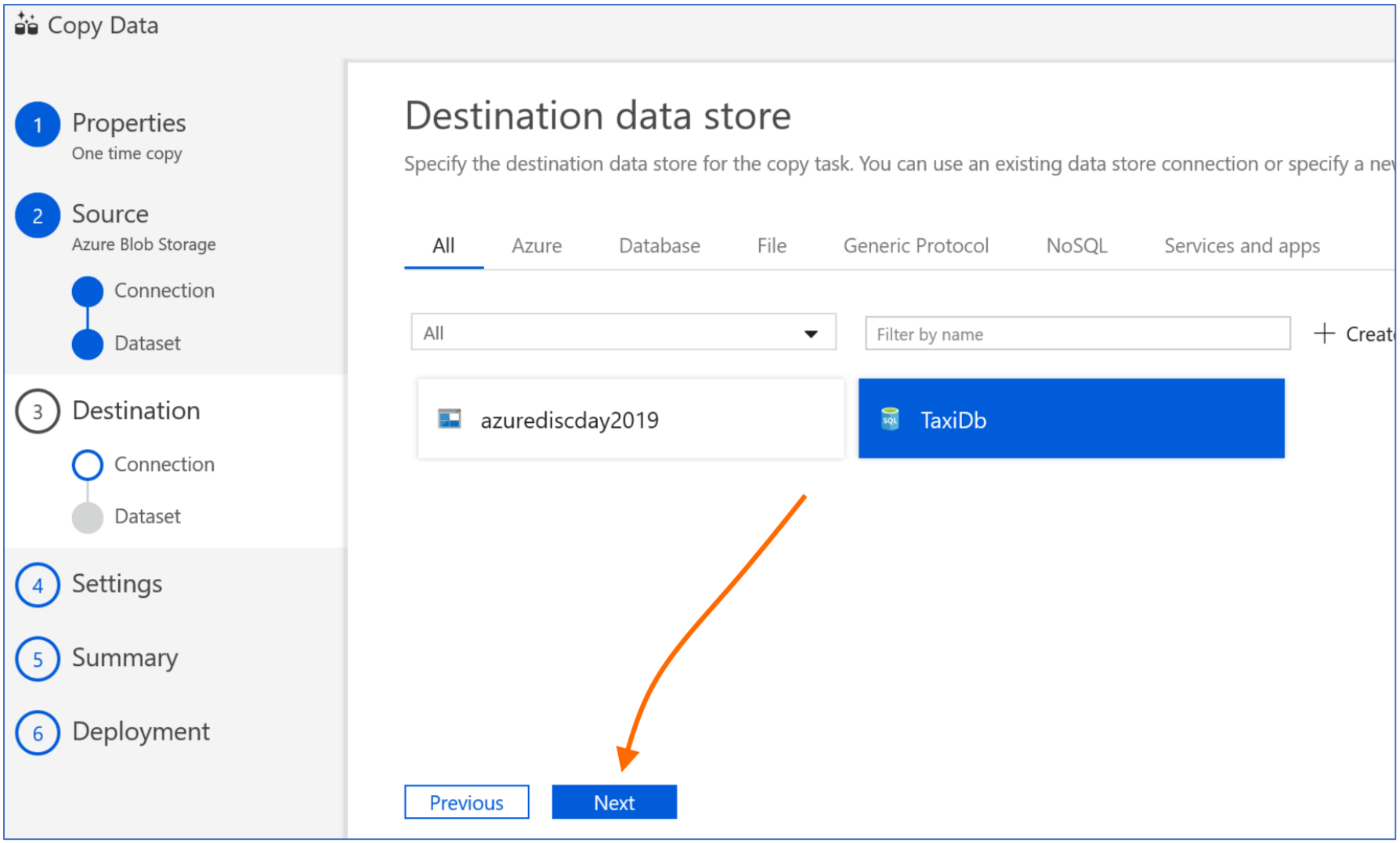
Select “Azure SQL Database”, then click “Continue”.



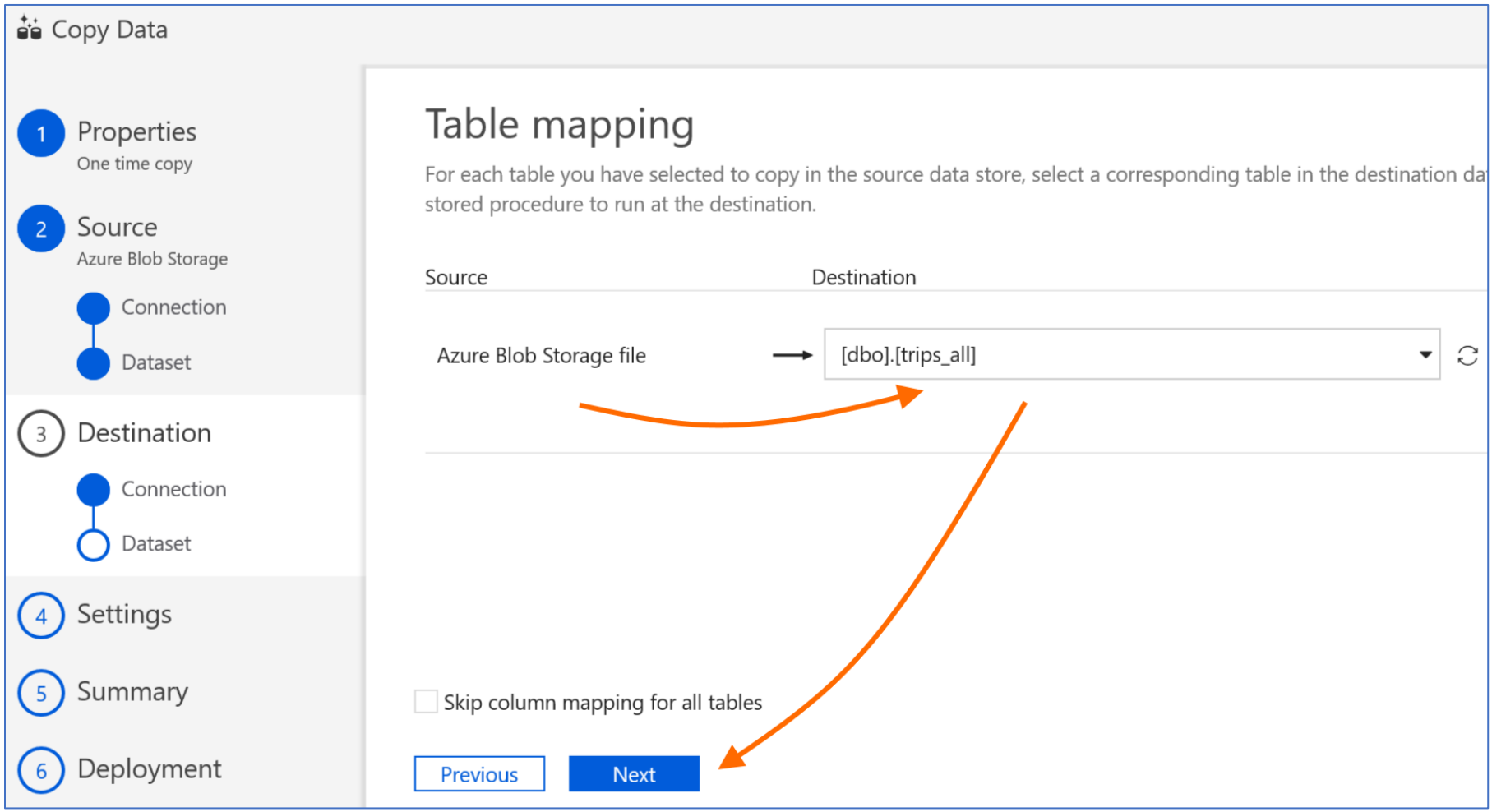
Name the data source. Select your Azure subscription, then the SQL server and Azure SQL DB database name. Enter the credentials (username and password) you specified when creating the database earlier in this lab. Optionally, click “Test connection” and verify that it is successful, then click “Finish”.



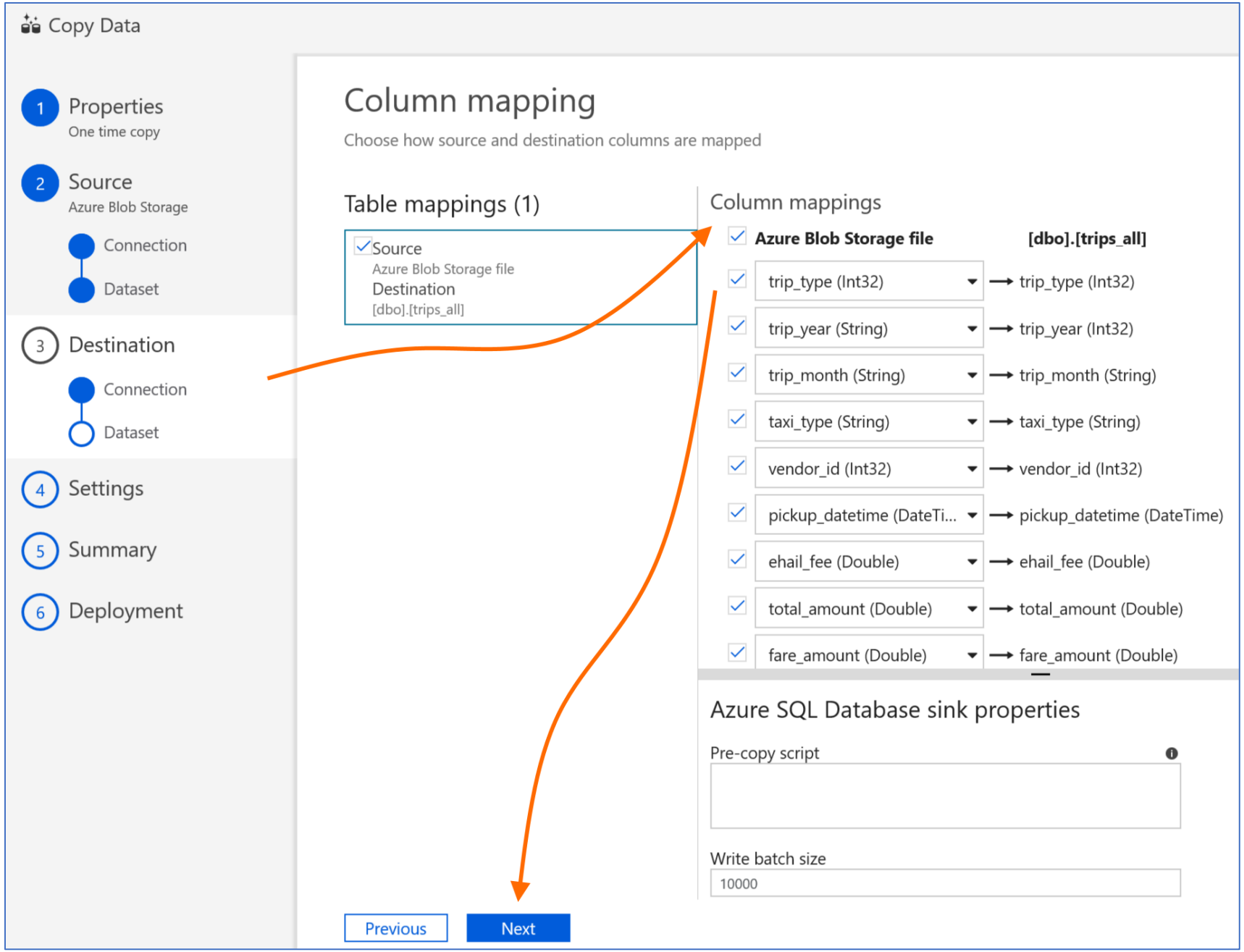
You have now created a source (Parquet files from lab 1) and a destination data set (your Azure SQL database). Click “Next” to move to schema mapping.



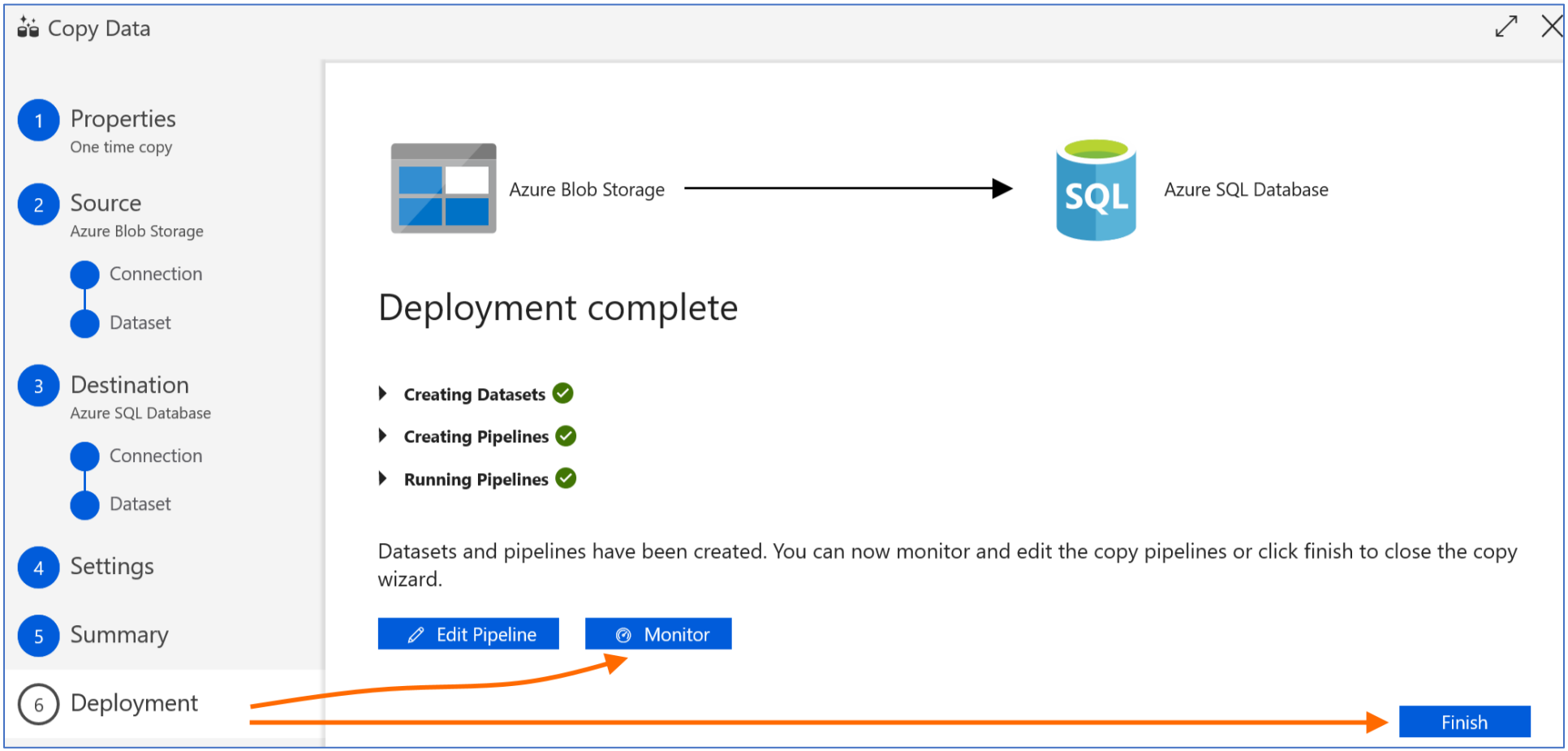
Select the destination table in your Azure SQL DB, then click “Next”.



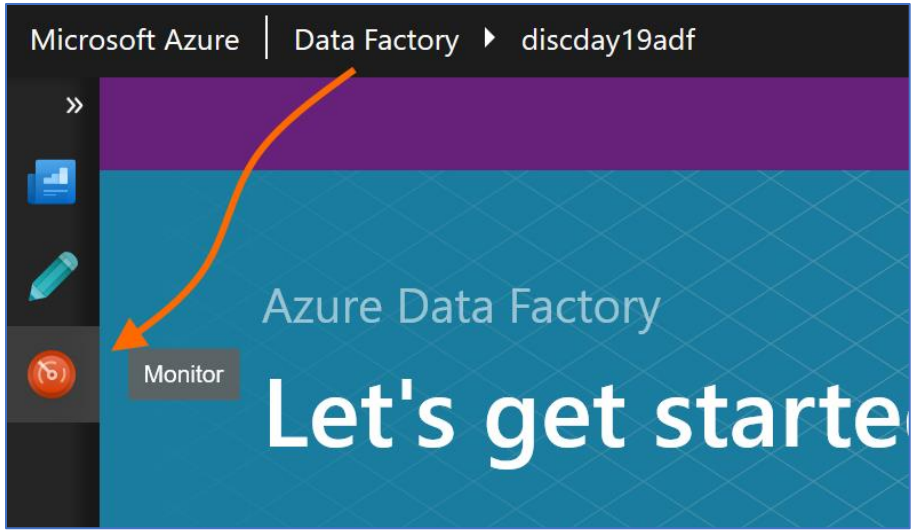
Review the column mappings – no changes should be needed here – then click “Next”.



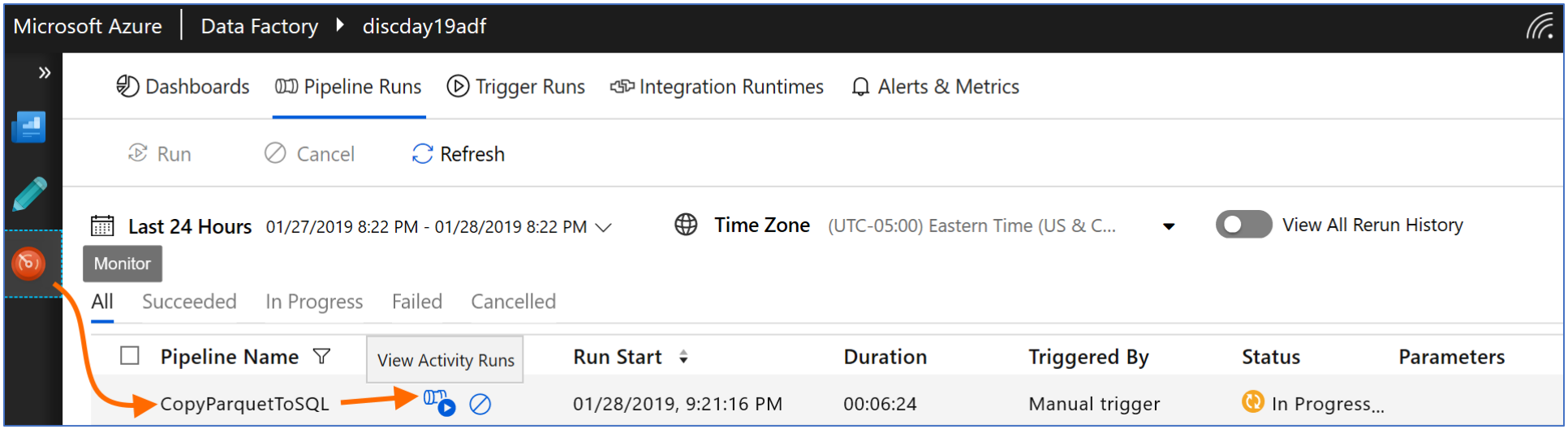
Next, on the tabs for “4 – Settings” and “5 – Summary”, accept the defaults and click “Next” until the “Deployment” tab is reached and the copy activity runs. You can monitor its progress by clicking “Monitor”, or close the copy wizard by clicking “Finish”.



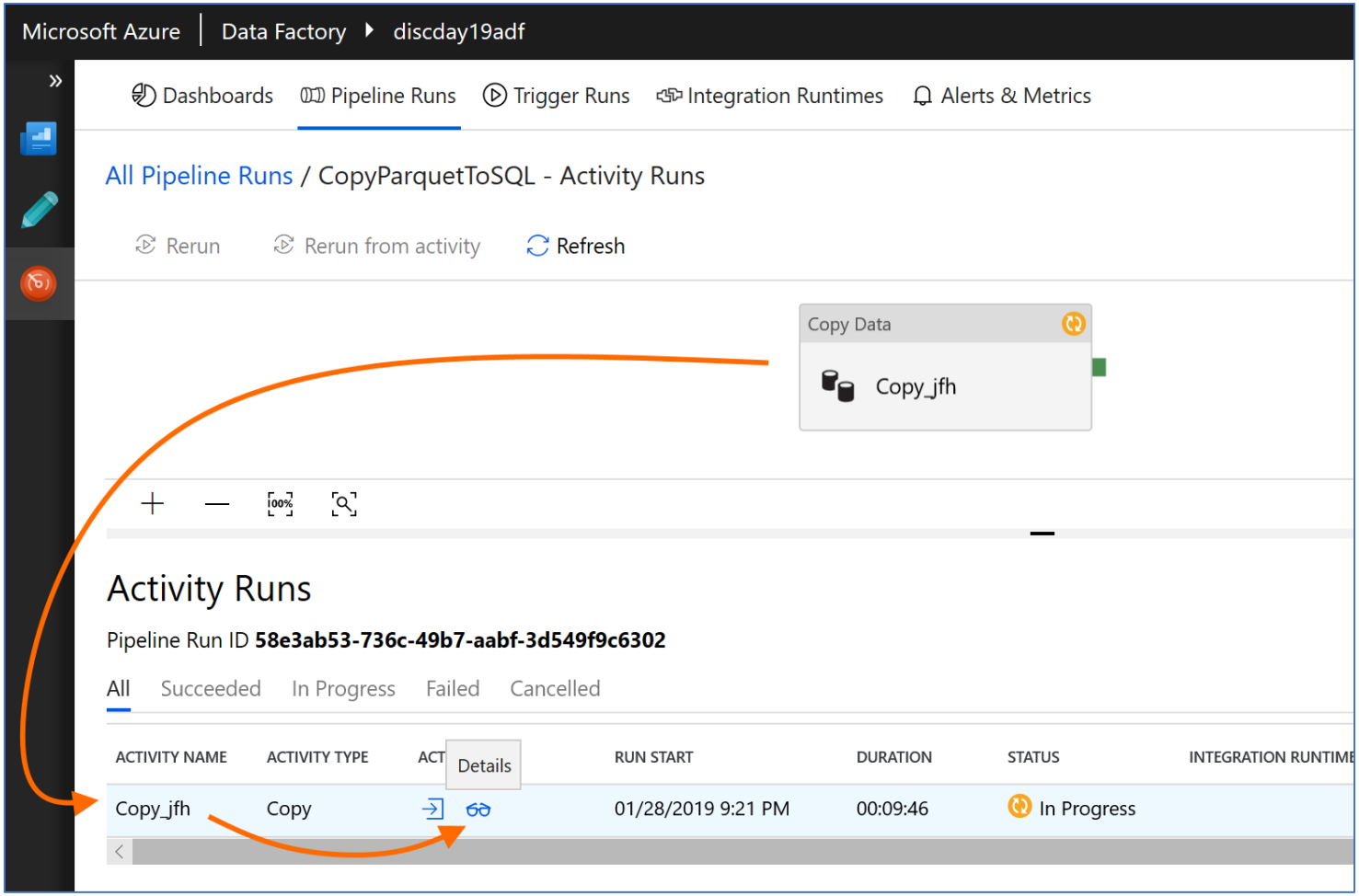
You can monitor pipeline progress anytime by clicking “Monitor”.



In the Monitor view, you can drill into a pipeline’s run history (current and past) by clicking the “View Activity Runs” icon next to its name.



You can then drill in even further to an activity run's details by clicking its "Details" icon.



Details Refresh

Performance tuning tips:
Sink Azure SQL Database: The DTU utilization was high during the copy activity run. To achieve better performance, you are suggested to scale the database to a higher tier than the current 500 DTUs. Refer to this document.

Learn more on copy performance details from here.

Azure Blob Storage

Succeeded

Azure SQL Database

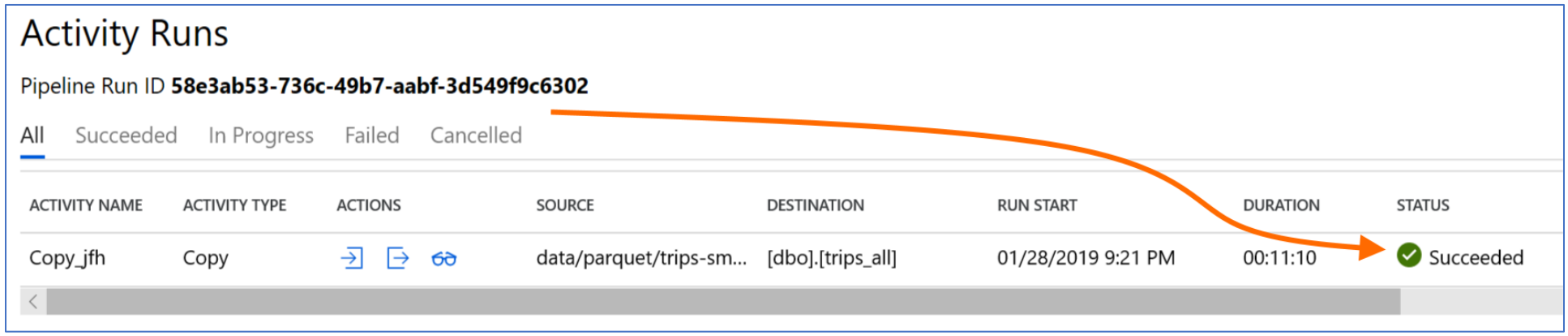
Data read: 483.641 MB
Files read: 8
Rows read: 22,234,960

Data written: 3.146 GB
Rows written: 22,234,960
Throughput: 712.377 KB/s

Copy duration 00:11:03

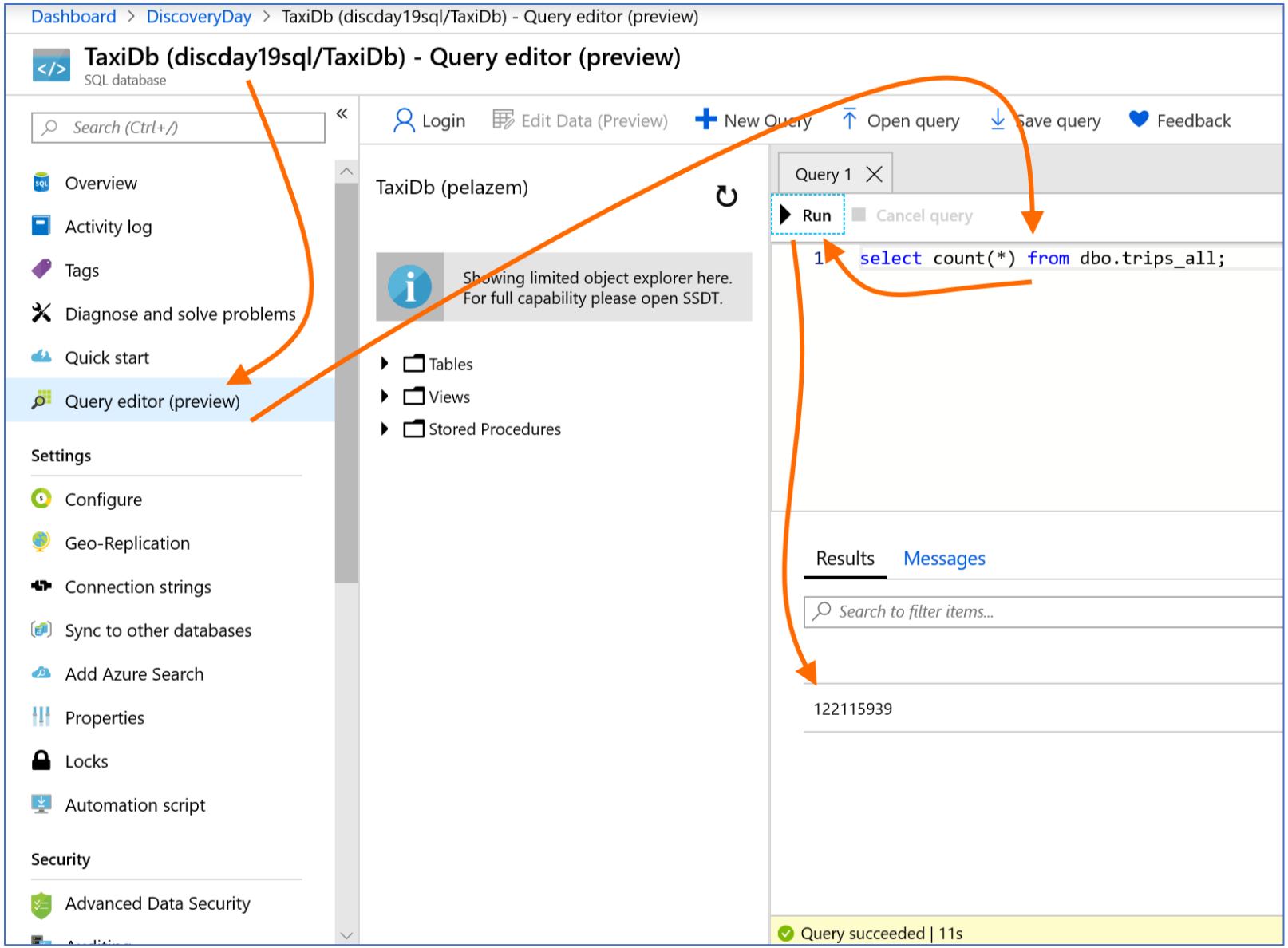
▸ Azure Blob Storage → Azure SQL Database Queue 00:00:03 | Transfer 00:10:59

When the activity completes, it will show status of “Succeeded”.



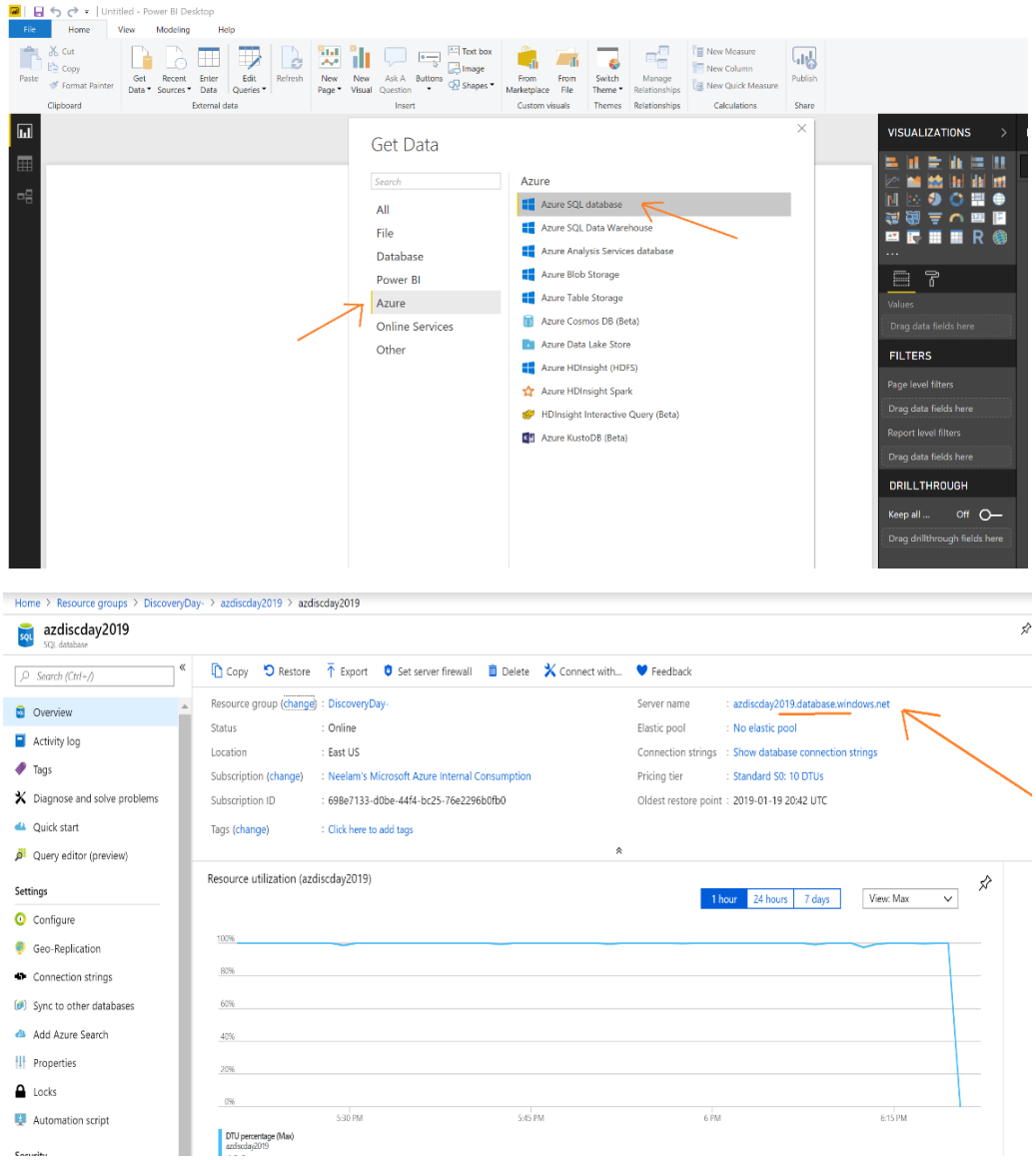
After the copy activity has succeeded, you can go back to your Resource Group, into your Azure SQL DB, and to the “Query editor” tab. Alternately, you can connect to your database in SQL Server Management Studio. Either way, run a simple query like the following to verify that data has been copied into your database.

```
select count(*) from dbo.trips_all;
```

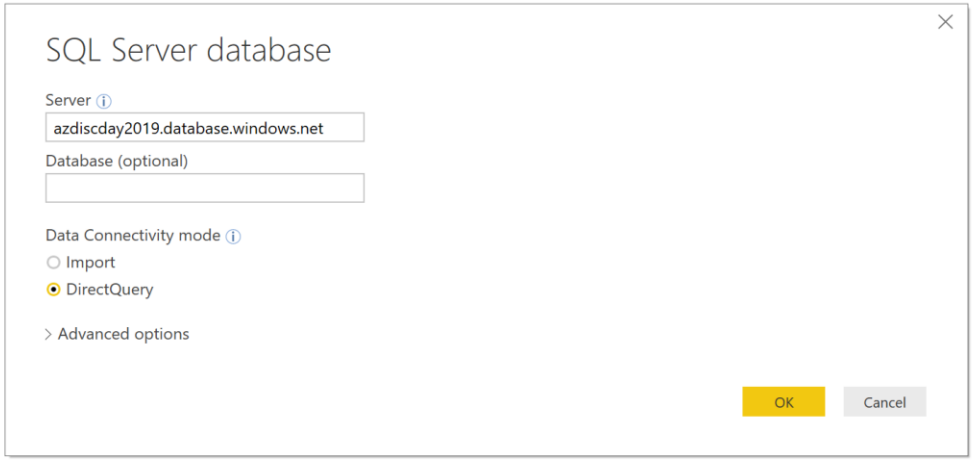


Task 3 – Create Power BI Reports

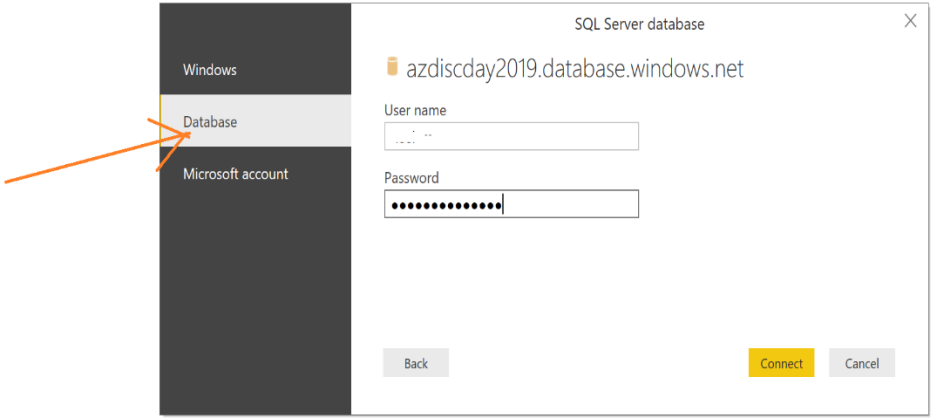
There are two options to create Open Power BI desktop application and connect to Azure sql database.



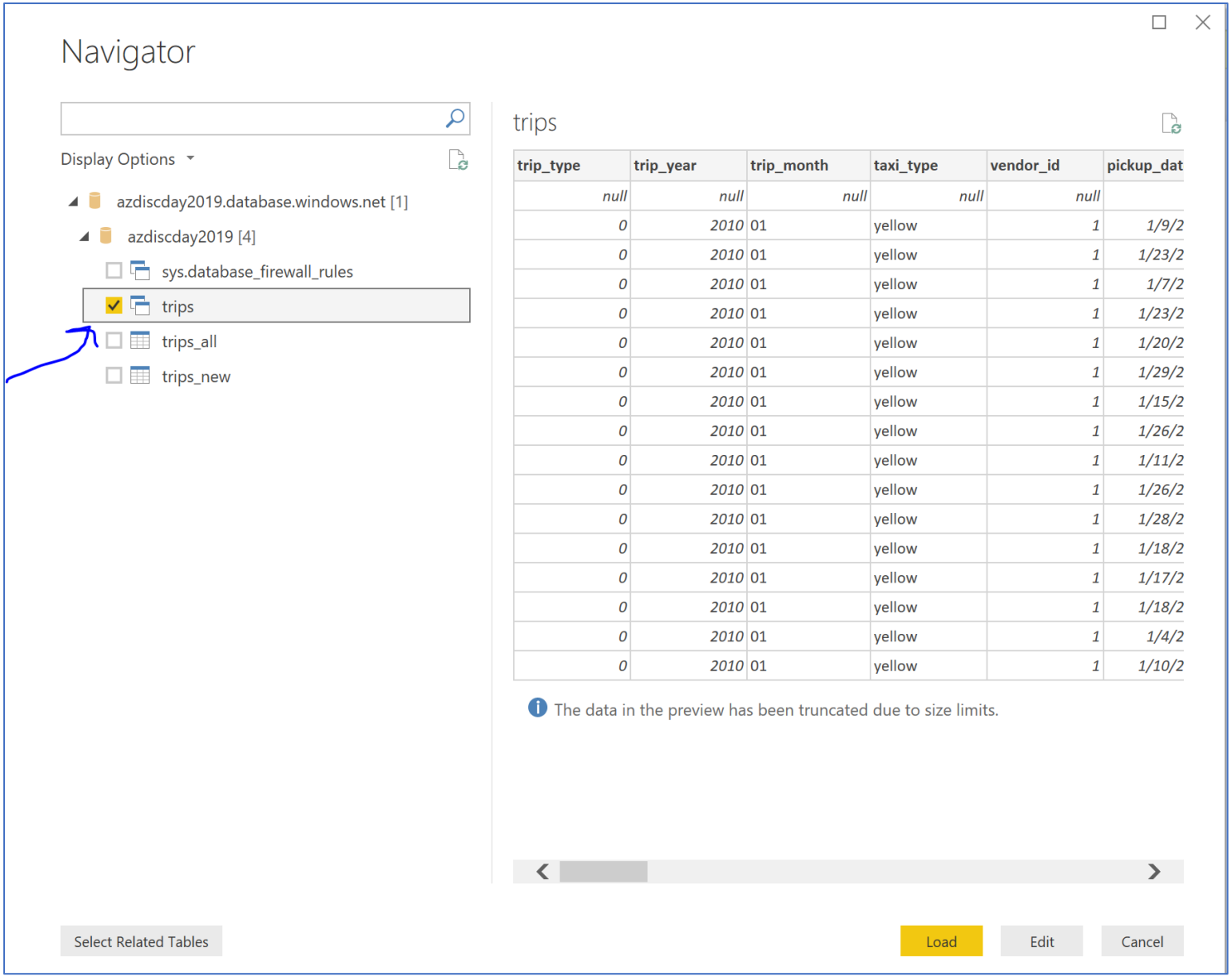
Enter Server name and select direct query and click ok

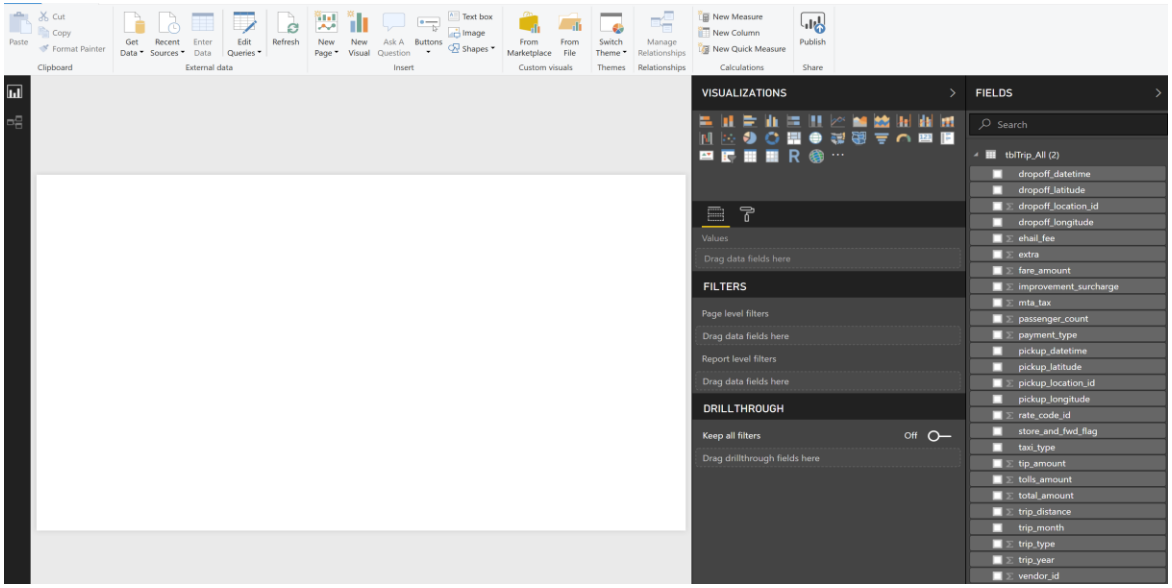


Select Database as the option and enter credentials



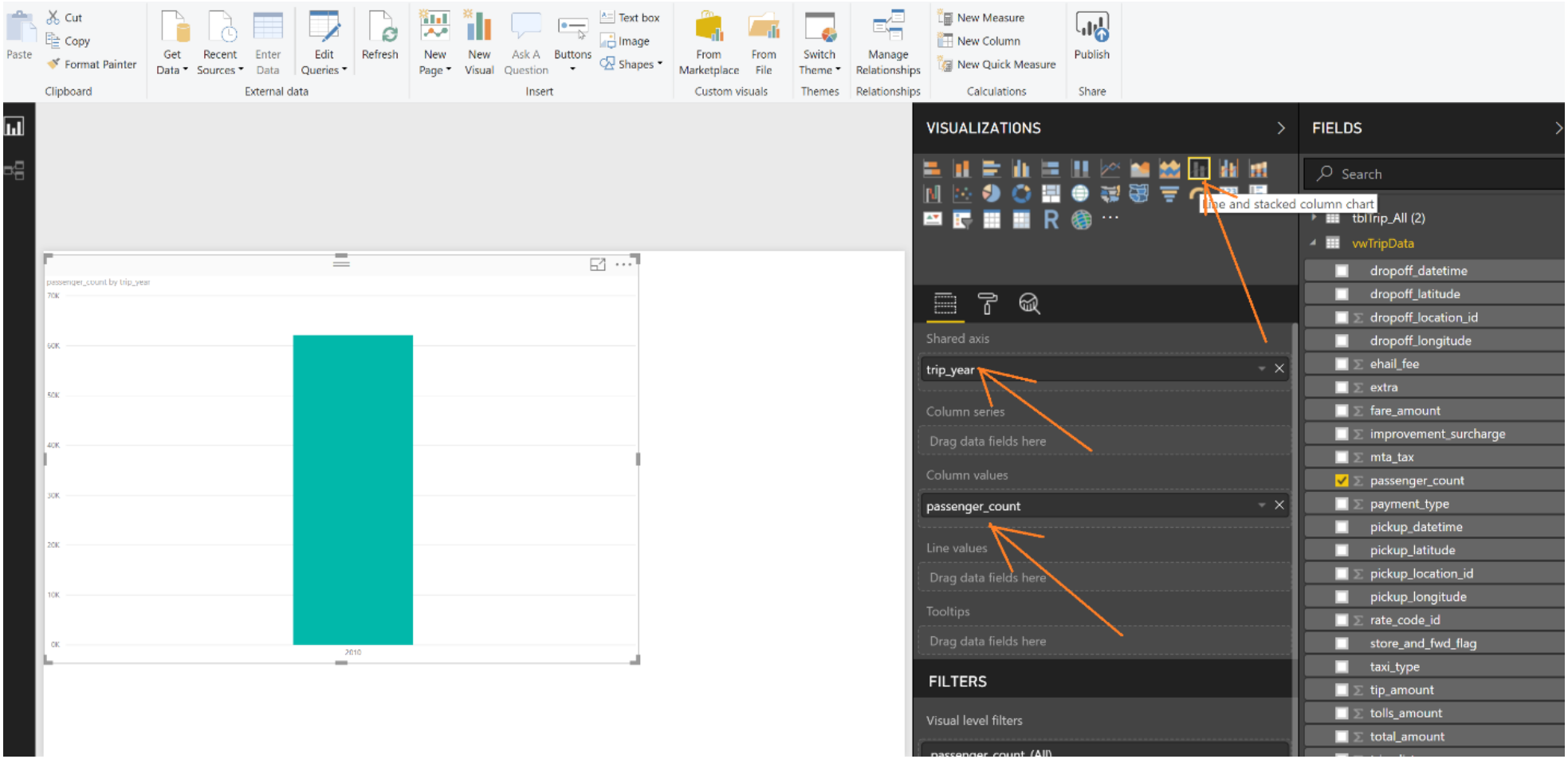
Select the view you created (which unions the batch and streaming data tables) and click “Load”.



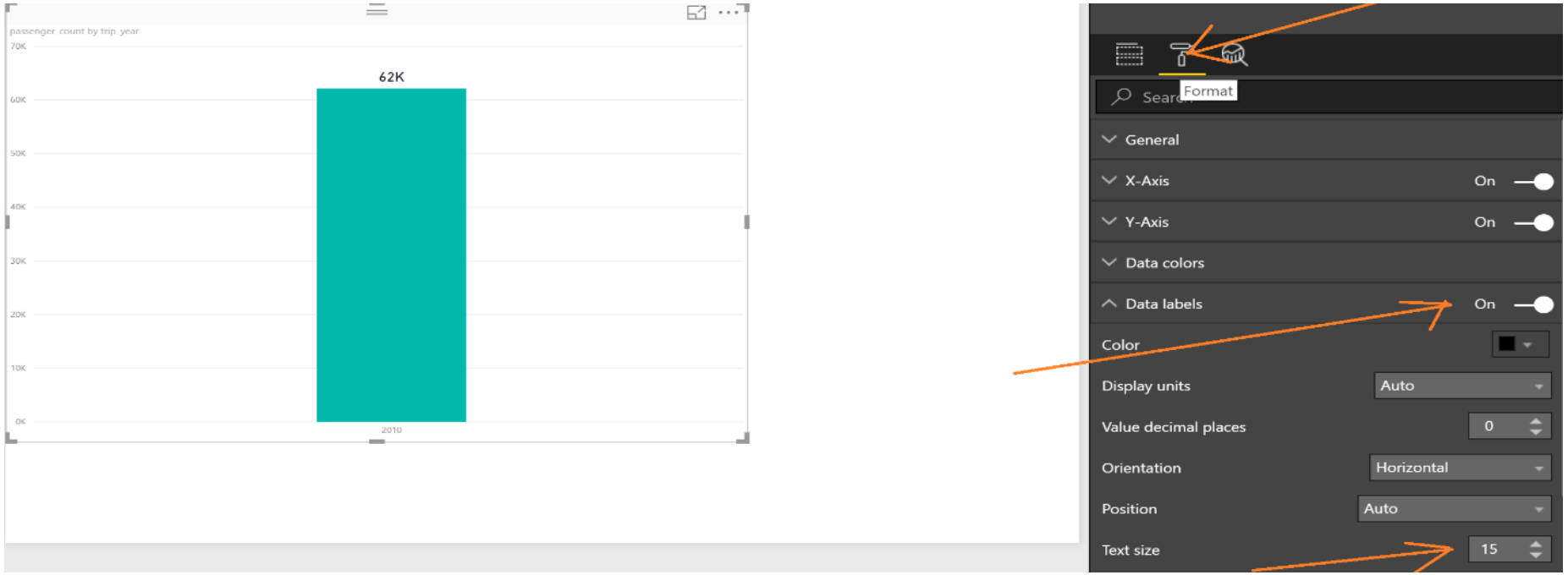


Create a Power BI Chart → Total passengers by year.

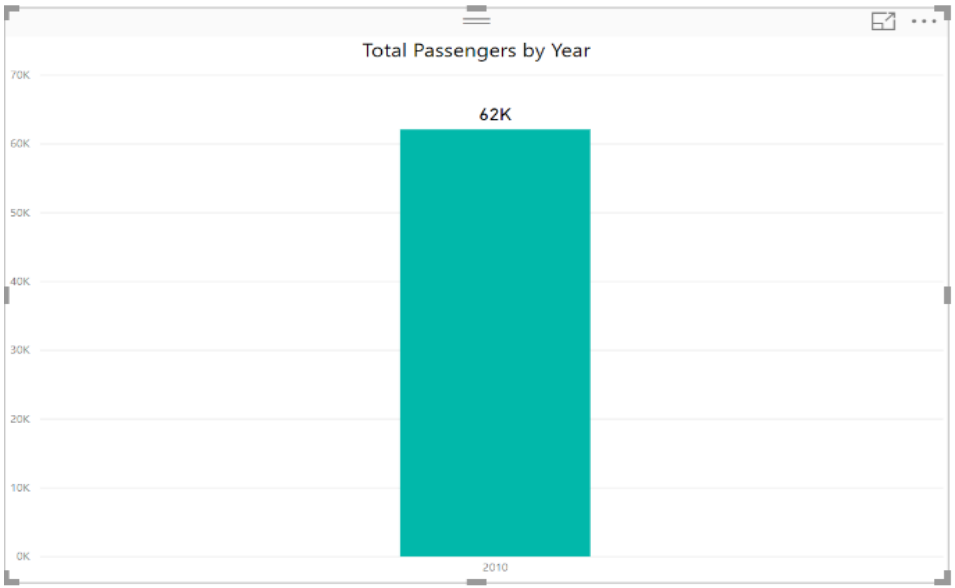
Drag year into Axis and passengers count in values.

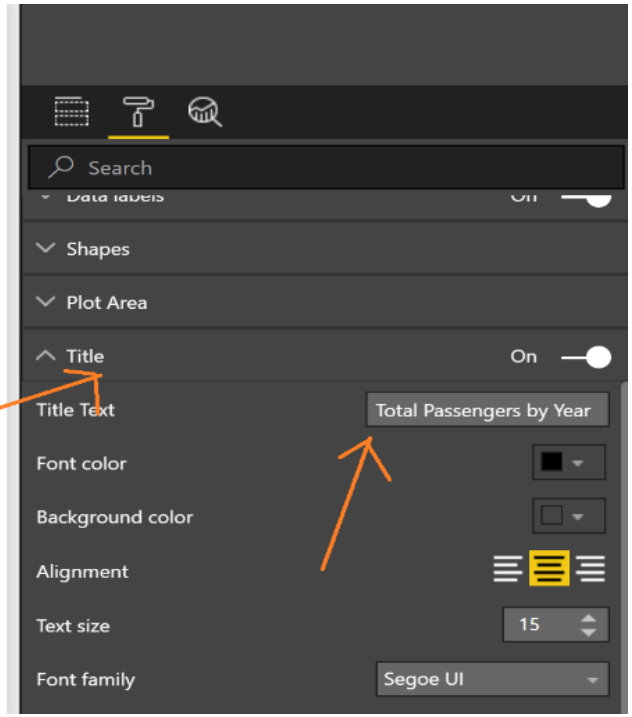


Please format labels using below options –

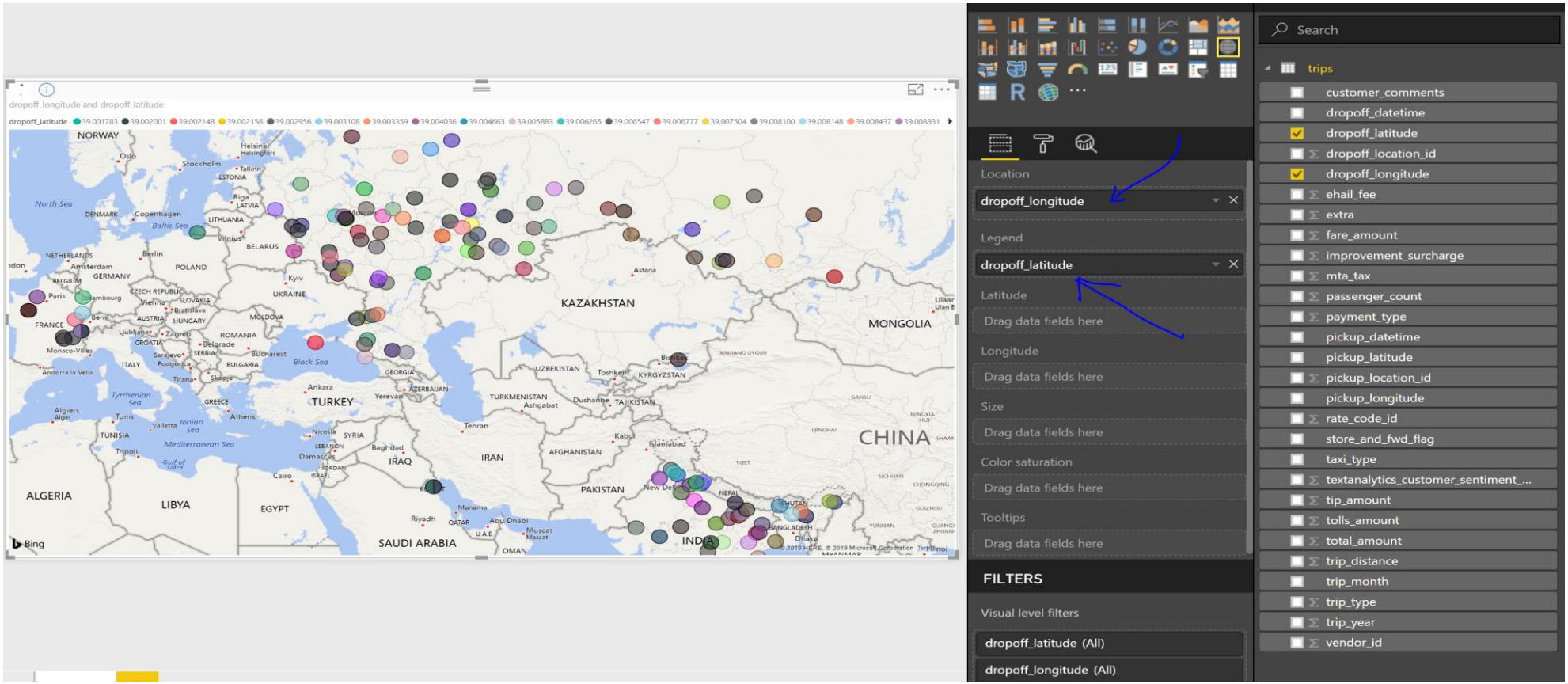


Rename chart title –



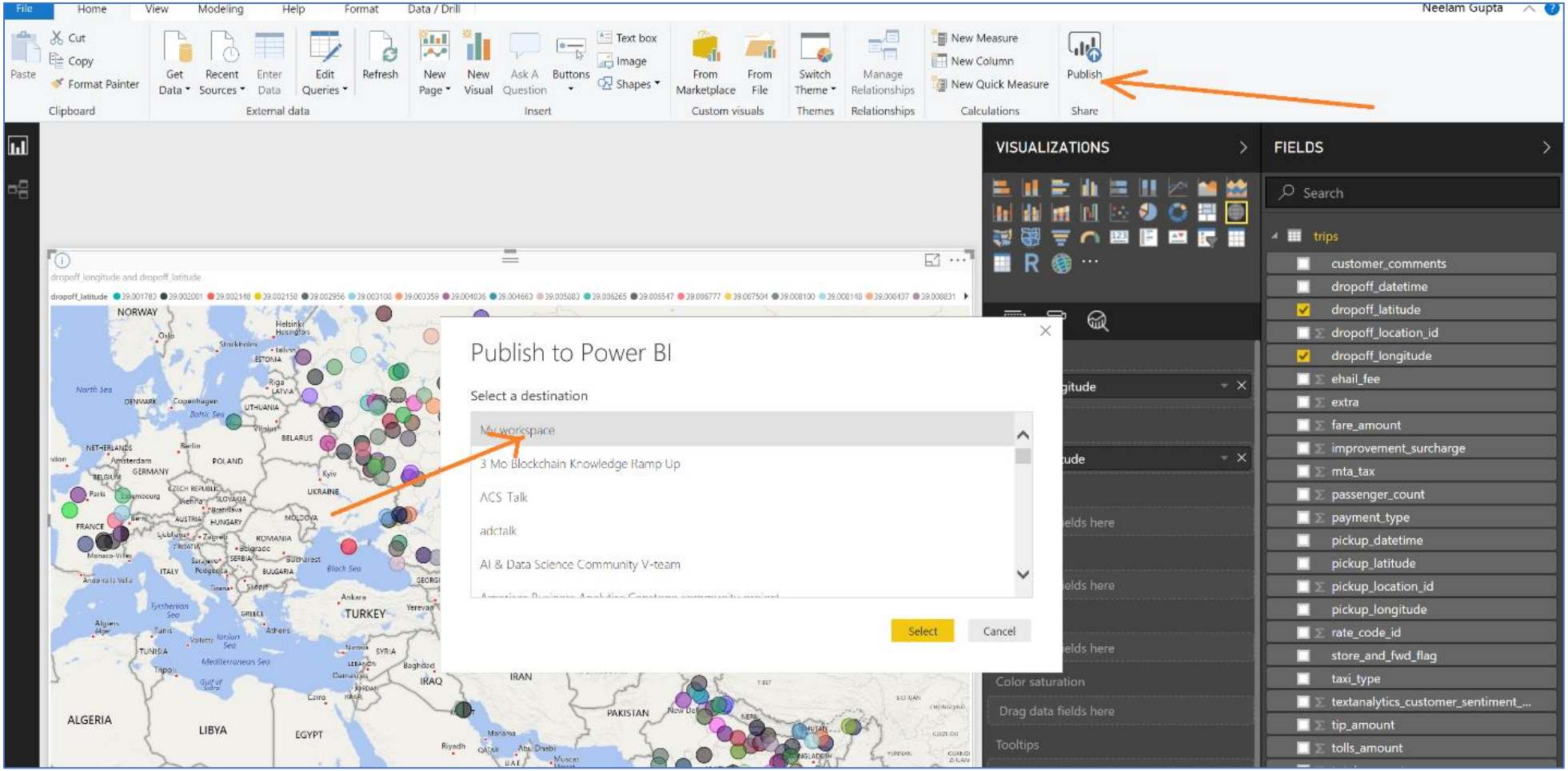


Create Map chart.



After creating all reports, deploy to App.PowerBI.com

Click Publish on the ribbon and then select workspace under which you want to deploy reports.



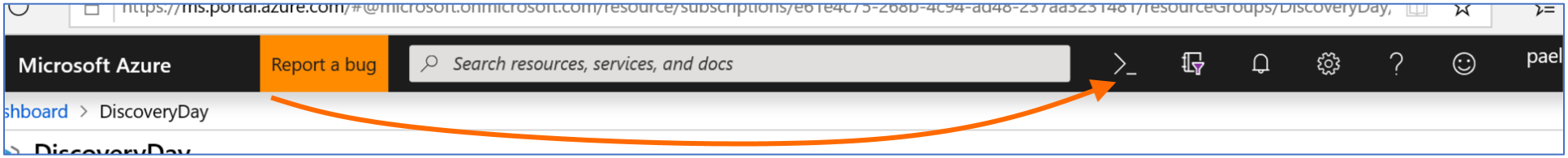
Conclusion

You have now completed lab 2. Great job! Now, it’s time to switch to the streaming data path in labs 3 and 4.

Appendix

In case you encounter difficulties copying the Parquet data into Azure SQL DB, this is an alternate approach to provisioning your Azure SQL DB. We have provided a .bacpac export of a small database (2018 data only).

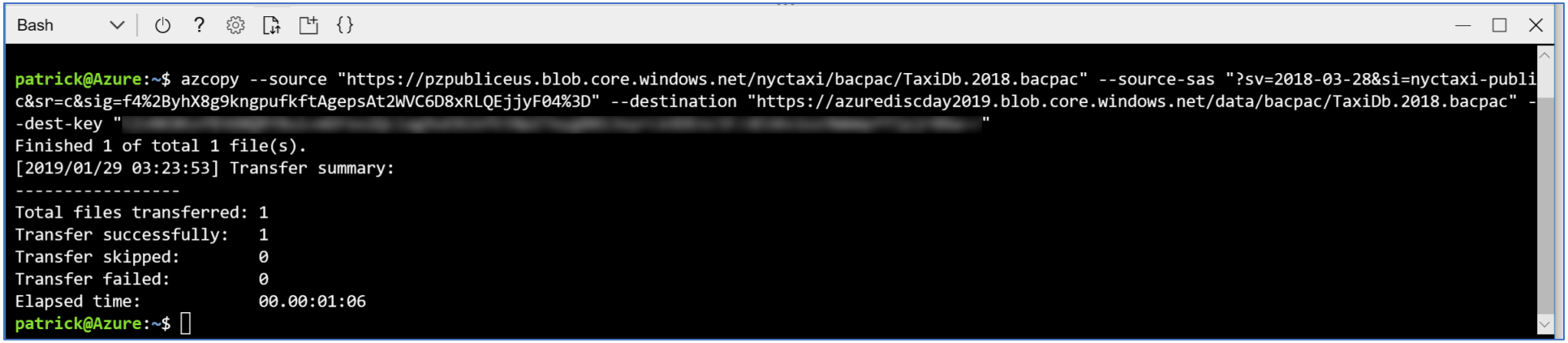
In your Resource Group, click the Cloud Shell icon in the top bar.



In the following command, substitute your Azure storage account name and key for the two tokens "{YOUR STORAGE ACCOUNT NAME}" and "{YOUR STORAGE ACCOUNT KEY}". Substitute your storage container name for the token "{YOUR CONTAINER NAME}".

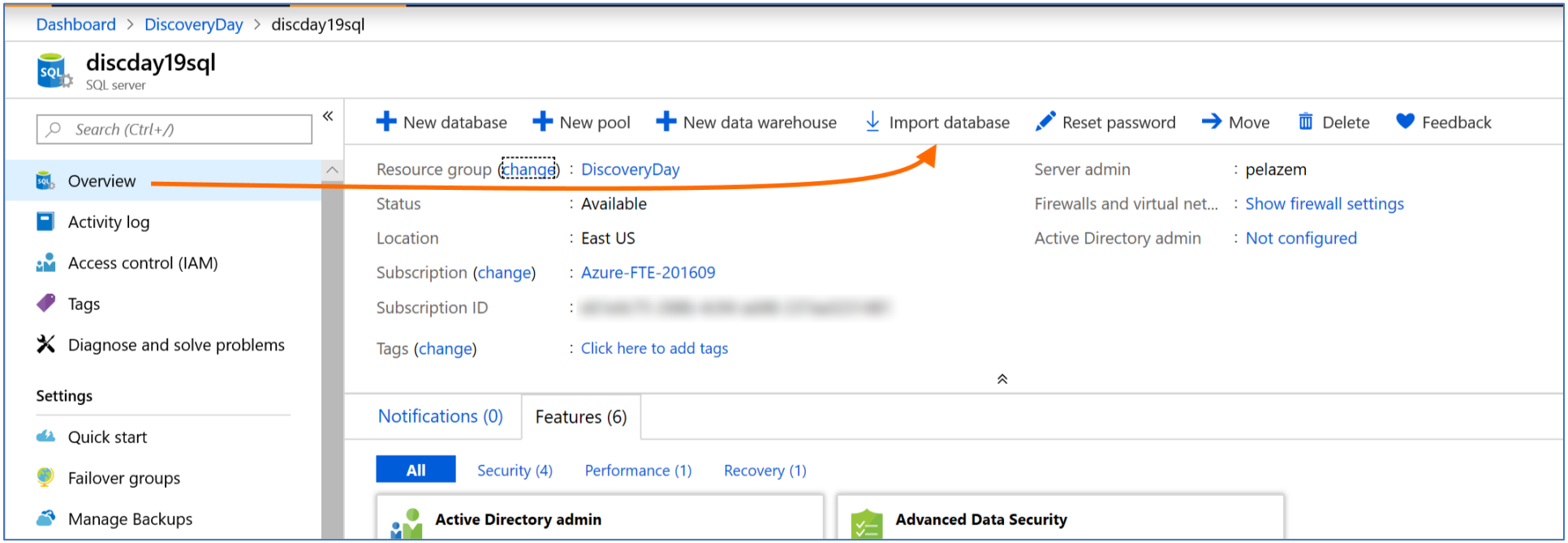
```
azcopy --source "https://pzpubliceus.blob.core.windows.net/nyctaxi/bacpac/TaxiDb.2018.bacpac" --source-sas
"?sv=2018-03-28&si=nyctaxi-public&sr=c&sig=f4%2ByhX8g9kngpufkftAgepsAt2WVC6D8xRLQEjjyF04%3D" --destination
"https://{YOUR STORAGE ACCOUNT NAME}.blob.core.windows.net/{YOUR CONTAINER NAME}/bacpac/TaxiDb.2018.bacpac"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

Copy the updated command and paste it onto the cloud shell prompt. Hit Enter and wait for the copy to complete.



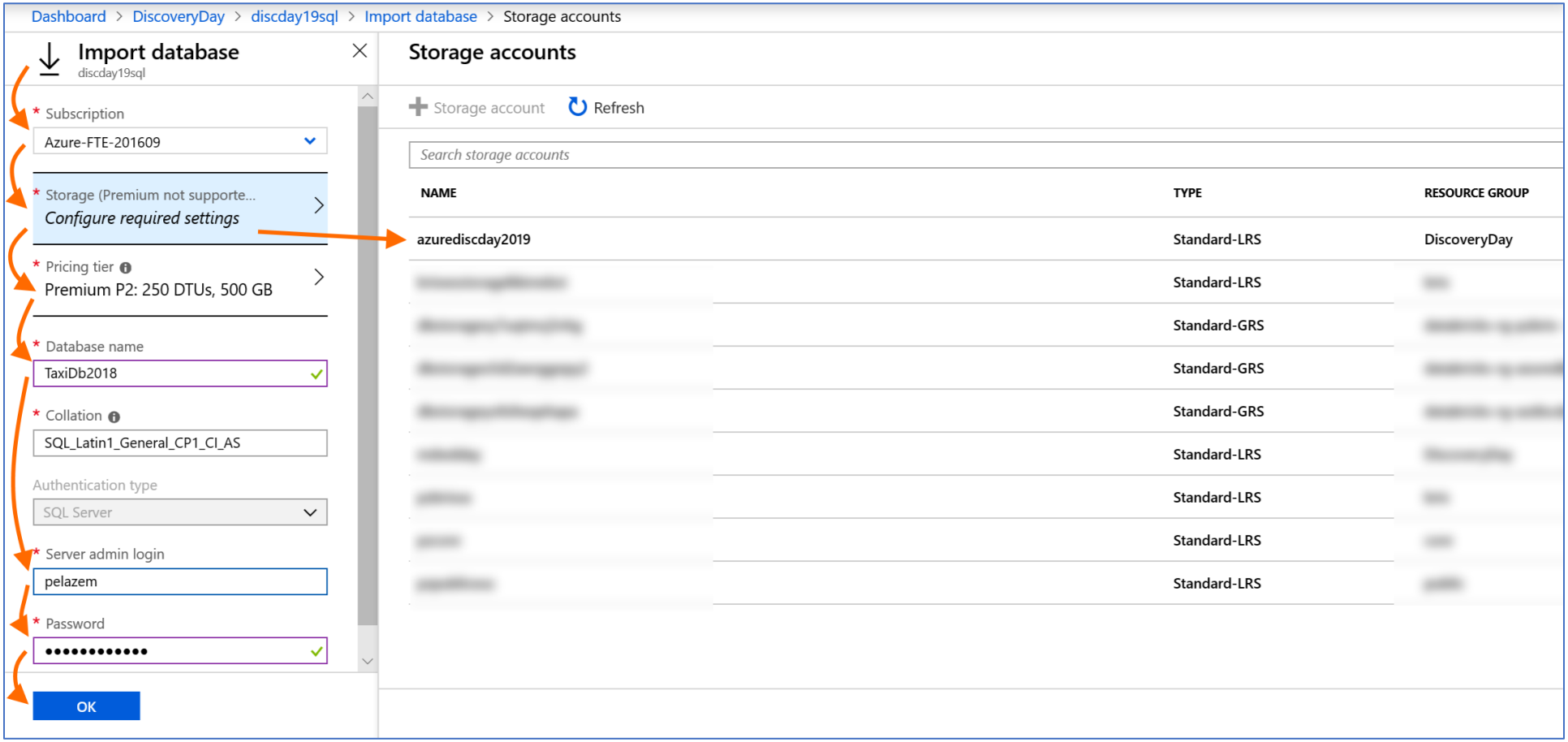
You now have a .bacpac file in your storage container, in a new bacpac folder.

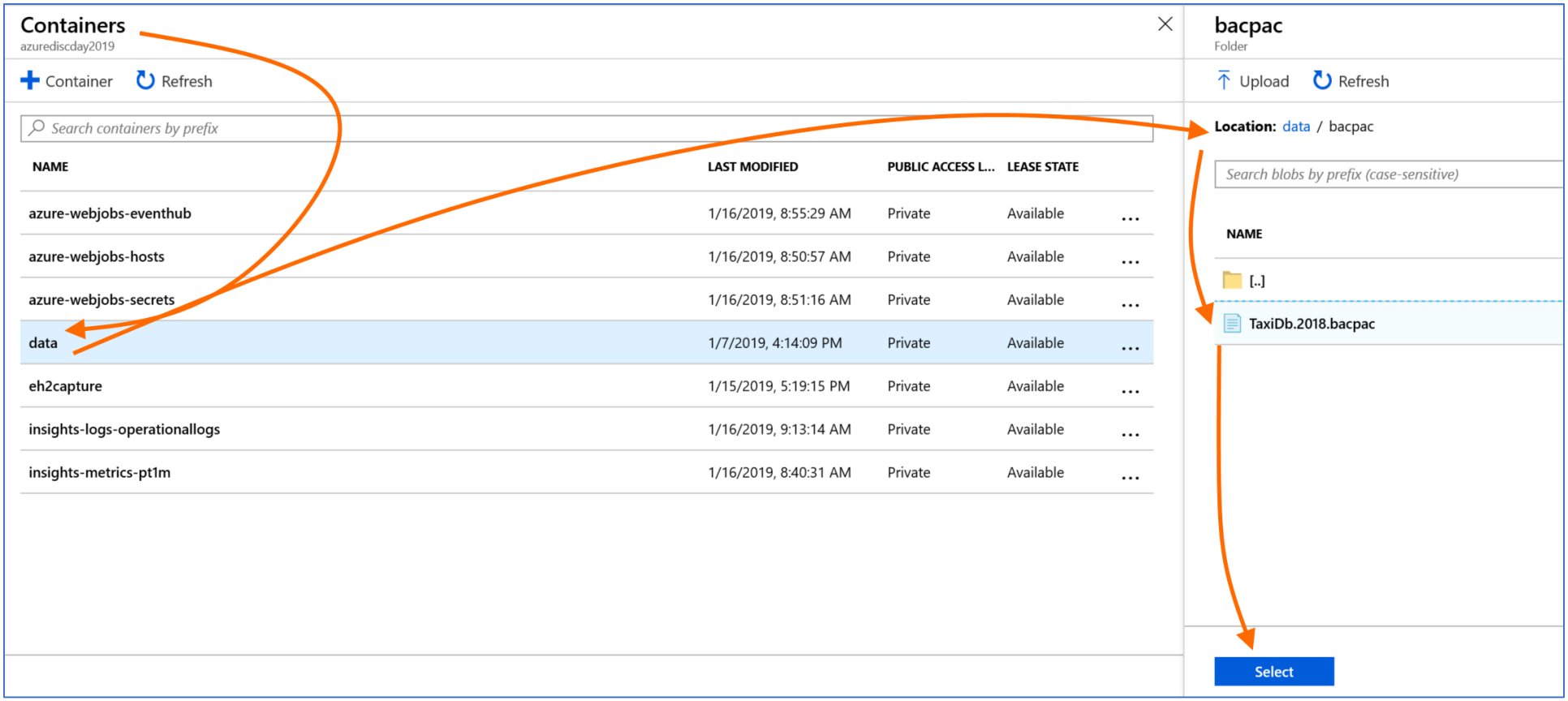
Next, return to your Resource Group. Click on the SQL Server resource that was created when you created an Azure SQL DB. In its “Overview” view click “Import database”.



On the Import Database view, select your subscription. Set an appropriate performance level for the database; since you'll be querying it in Power BI but will not be doing bulk import, P2 should be enough (and you can always scale Azure SQL DB down or up without downtime).

After you click on your storage account, you will need to further navigate into the container and to the folder where you copied the .bacpac file above, then click on the .bacpac file and click "Select" in order to complete this Import database screen.





Click “OK” and wait until the import is complete (see Notifications). You should now be able to query the imported database in the Azure portal query editor or in SQL Server Management Studio, as shown previously.