Azure SQL MI PowerBI

March 2019

Contents

Introduction 1

Understanding Integration Runtime in Azure Data Factory 1

SQL Managed Instance – Power BI 2

Exercise 0: Provisioning a self-service runtime environment for Managed Instance data movement. 2

Exercise 1: Use the Copy Data Tool to Create and Run Data Pipeline 10

Overview 10

Exercise 2: Provisioning an Azure-SSIS Integration Runtime in Azure Data Factory 20

Overview 20

Exercise 3: Deploy and Run SSIS Packages from Azure Data Factory SSIS IR 28

Overview 28

Connect to SSISDB 28

Exercise 4: Power BI 34

Overview 34

Conclusion 40

Introduction

In this lab, you will learn key concepts in data integration centered on orchestrating and operationalizing data movement and data transformation activities, as well as cloud-based SSIS package execution via Azure Data Factory's Integration Runtime feature. You will establish connections with both on-premises SQL Server Databases, as well as Managed Instances in the cloud, to allow the creation of data pipelines that run from your sources, all the way to your destination. Then, you will lift-and-shift native SSIS packages onto the cloud to allow for more seamless control and monitoring during executions. Finally, you'll complete the journey of your data by establishing a connection to a live Power BI Desktop report, where you will be able to visualize directly the various components and insights contained in your data.

Understanding Integration Runtime in Azure Data Factory

Integration Runtime (IR) *is the* **compute infrastructure** *used by Azure Data Factory to provide the following data integration capabilities across different network environments:*

- **Data Movement:** Moving data between data stores in public and private networks.
- **Activity Dispatch:** Monitoring transformation activities running on a variety of compute services.
- **SSIS Package Execution:** Natively execute SQL Server Integration Services (SSIS) packages in a managed Azure compute environment.

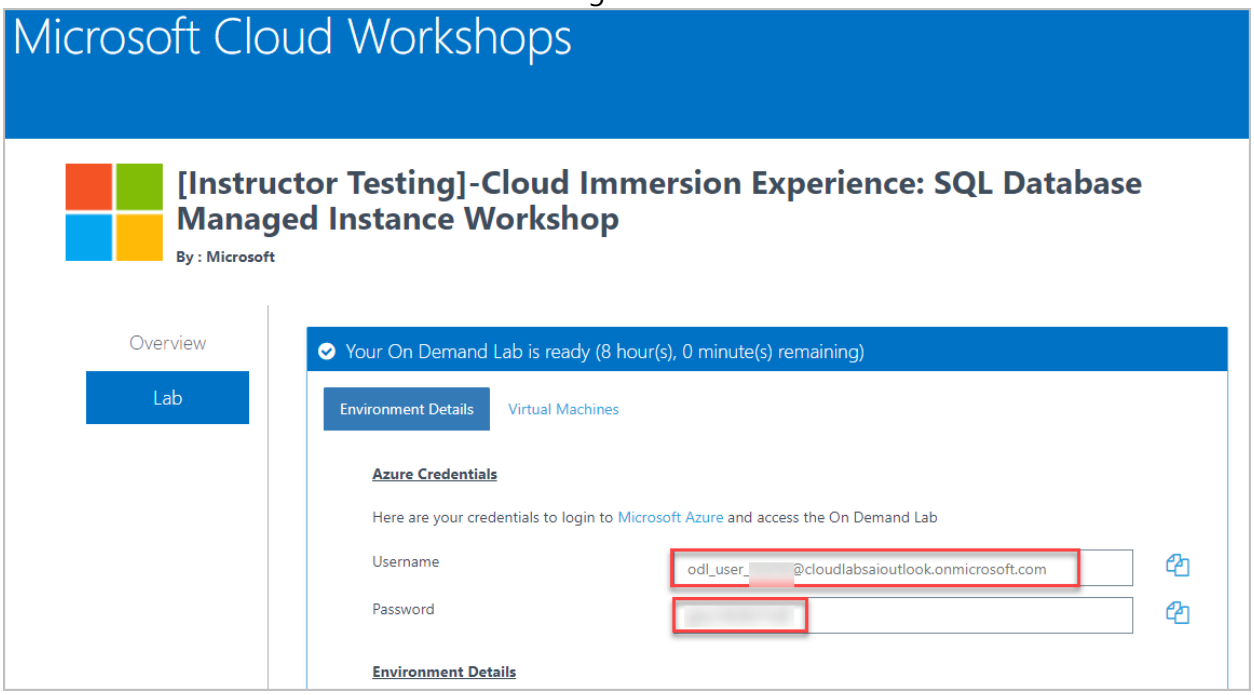
Azure Data Factory *offers three types of Integration Runtimes that serve specific data integration capabilities; Azure, Self-hosted, and* ***Azure-SSIS****. The last type supports the previously mentioned* ***SSIS Package Execution****, which is what we'll be using in this section of the lab.*

SQL Managed Instance – Power BI

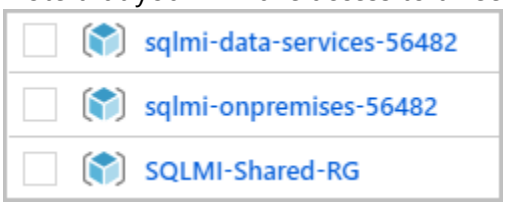
Exercise 0: Provisioning a self-service runtime environment for Managed Instance data movement.

Create a Self-Hosted Integration Runtime for Managed Instance:

1. Launch a browser and navigate to https://portal.azure.com. Once prompted, login with the Azure Credentials from the Lab Details Page.

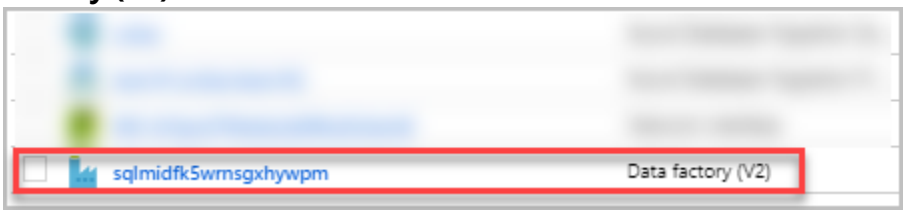


2. In the Stay signed in? pop-up window, click No
3. In the Welcome to Microsoft Azure pop-up window, click Maybe Later
   Note: If you receive a pop-up for Azure Advisor, click the X in the top right corner of the pop-up to close it.
4. You will be directed to the dashboard.
5. From the left side of the Page, select **Resource Groups**
6. Note that you will have access to three Resource groups:

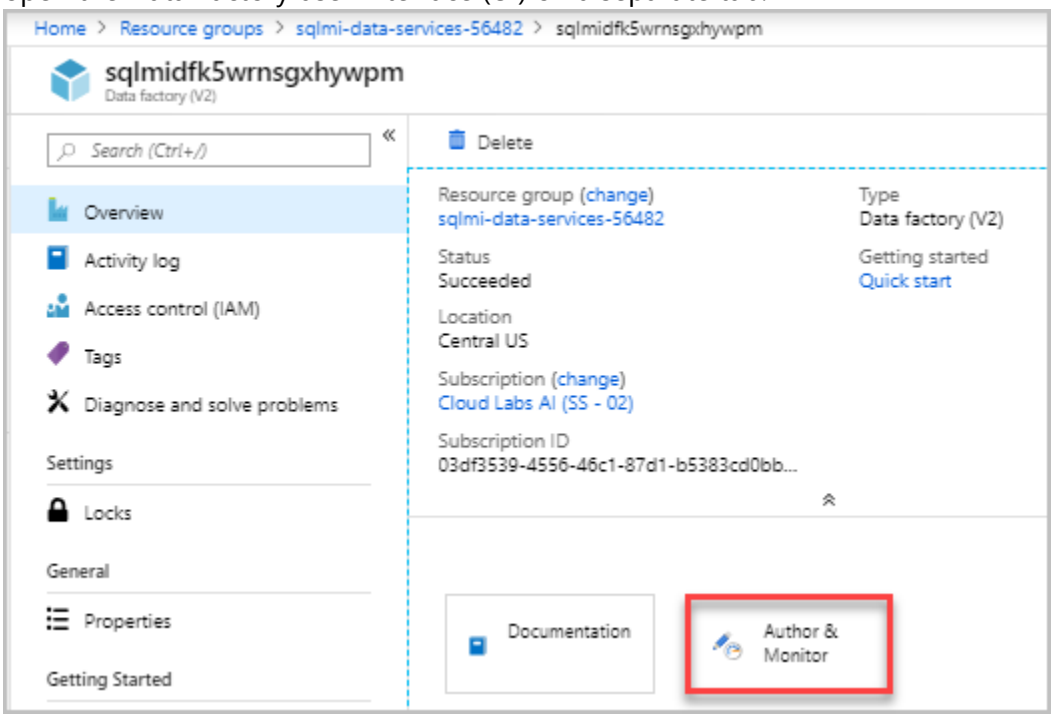


7. Select resource group starting with **sqlmi-data-services-**

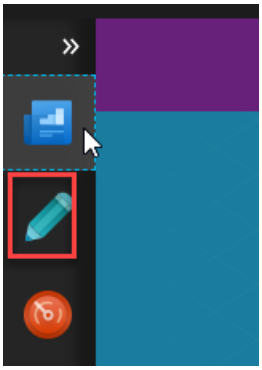
8. From the overview blade of the Resource group, **select** the resource with type **Data Factory (V2)** from the resources:



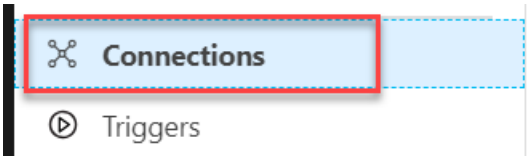
9. From the overview blade of the Data Factory, navigate to the Author & Monitor tile to open the Data Factory user Interface (UI) on a separate tab.



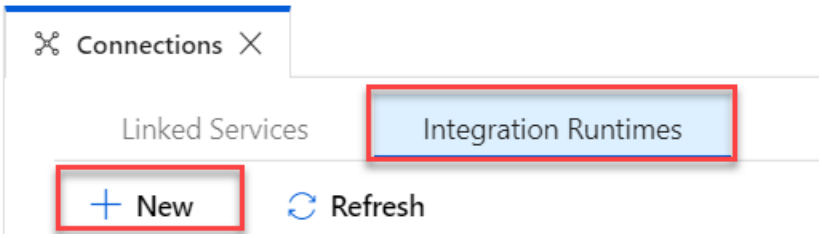
10. Click on Author(Pencil Icon)

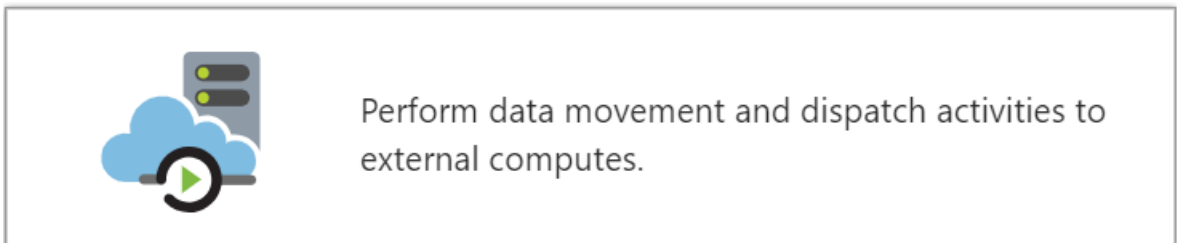
11. 

12. Click on Connections

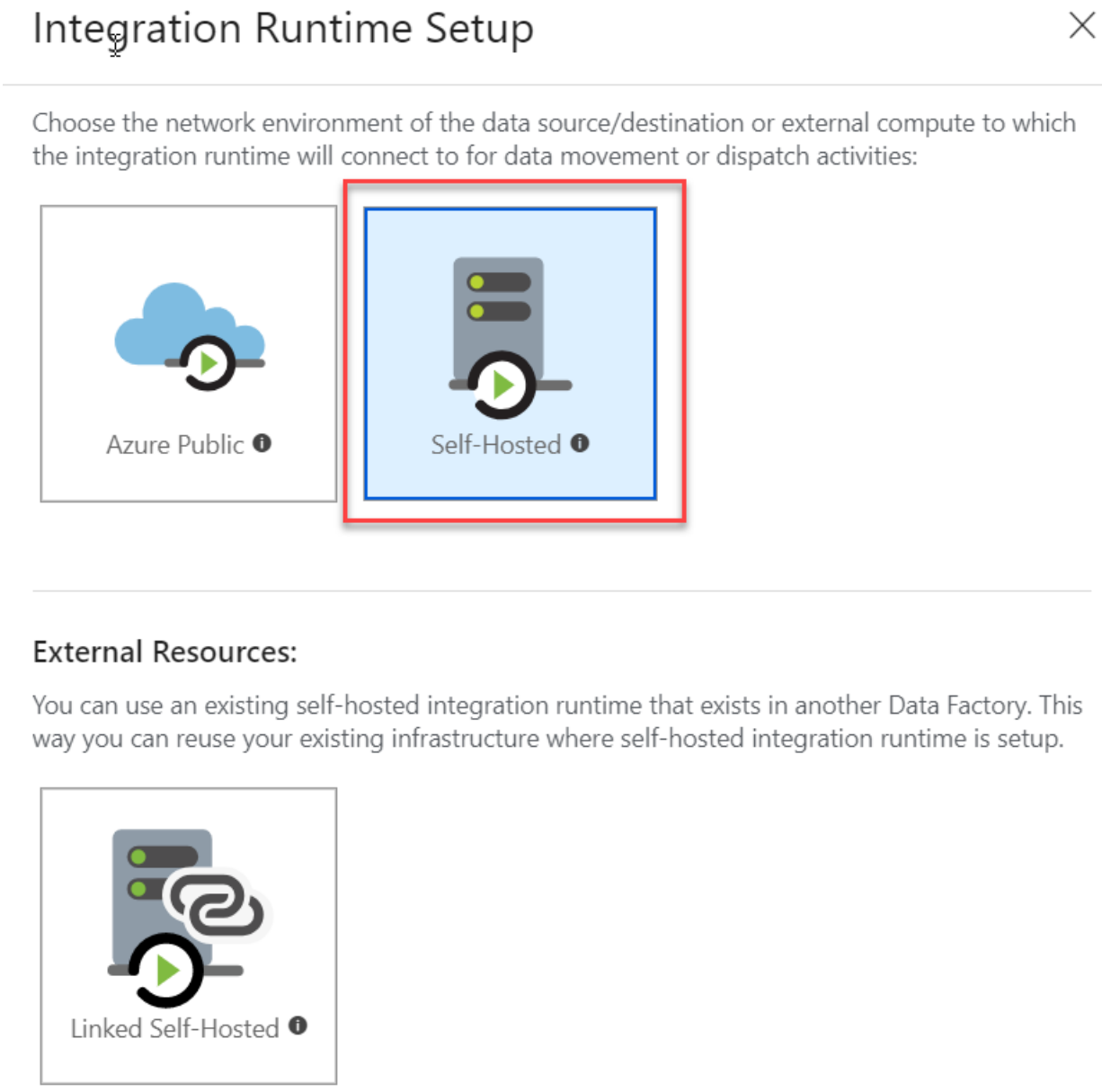
13. 

14. Select Integration Runtimes and Click on +New

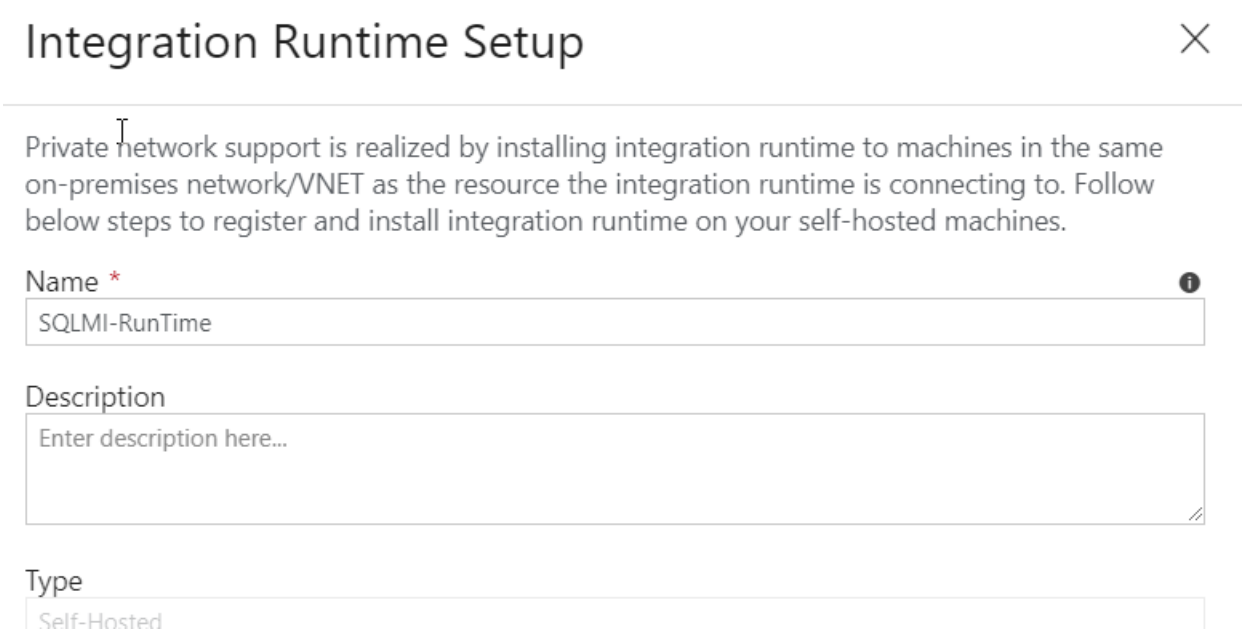
15. 

16.
17. Select Perform data movement and dispatch activities to external computers
18. 

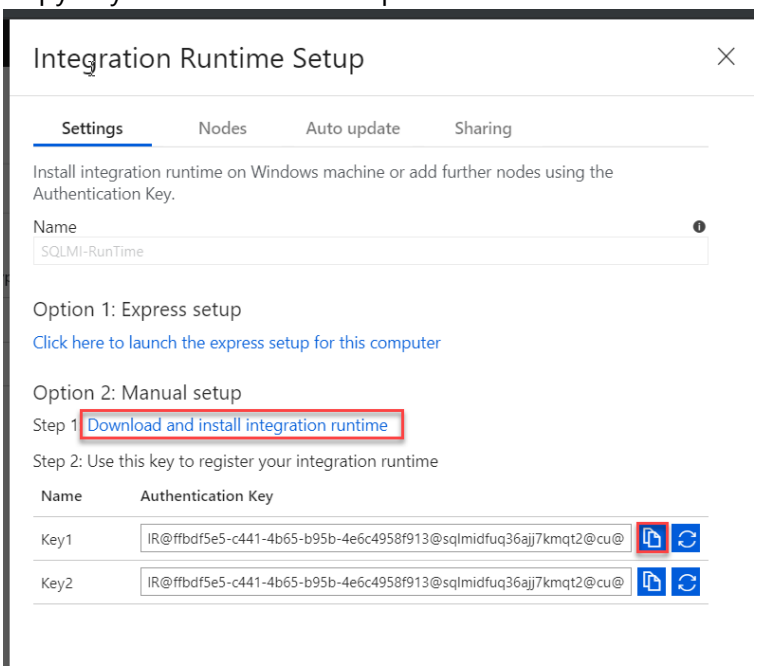
19.
20. Select Self-Hosted and Click Next.



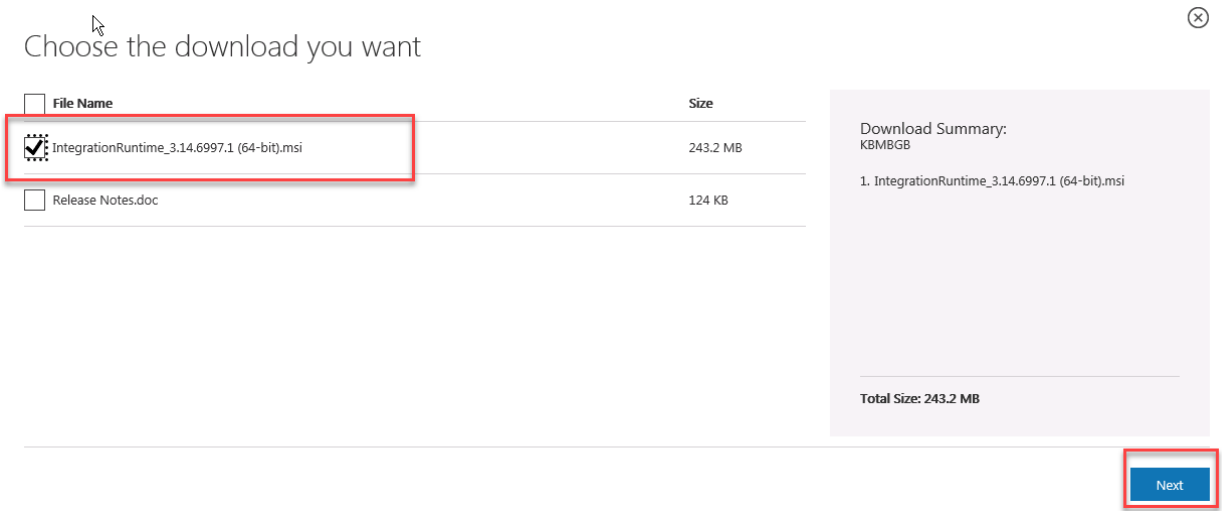
21.
22. Provide SQLMI-RunTime as name and click next



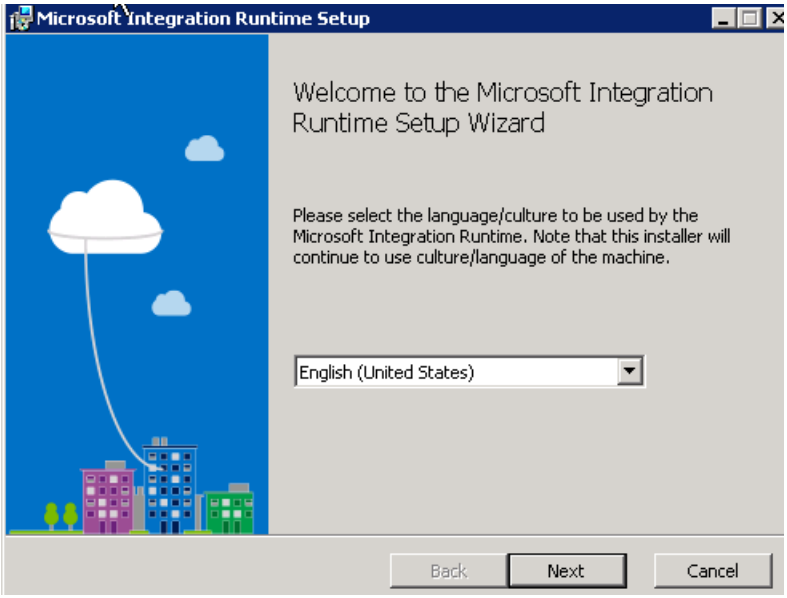
23.
24. Copy Keys and save in a safe place.



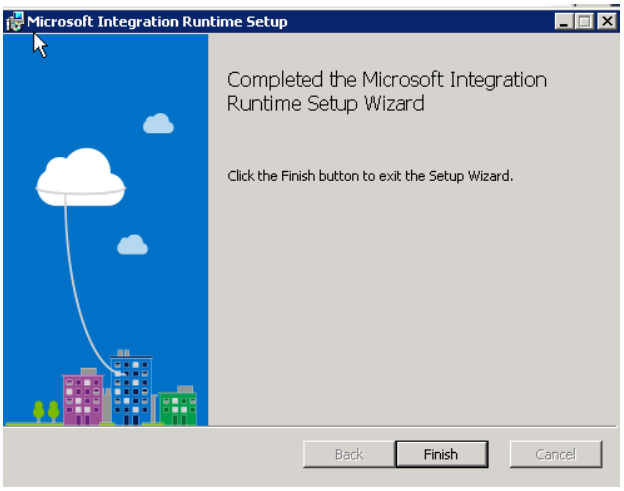
25.
26. Now, we'll install the runtime binaries on the SQL Server VM.
27. Login to SQL Server Virtual Machine using Remote Desktop and Launch a browser window.
28. Download the manual setup file form https://www.microsoft.com/en-us/download/details.aspx?id=39717



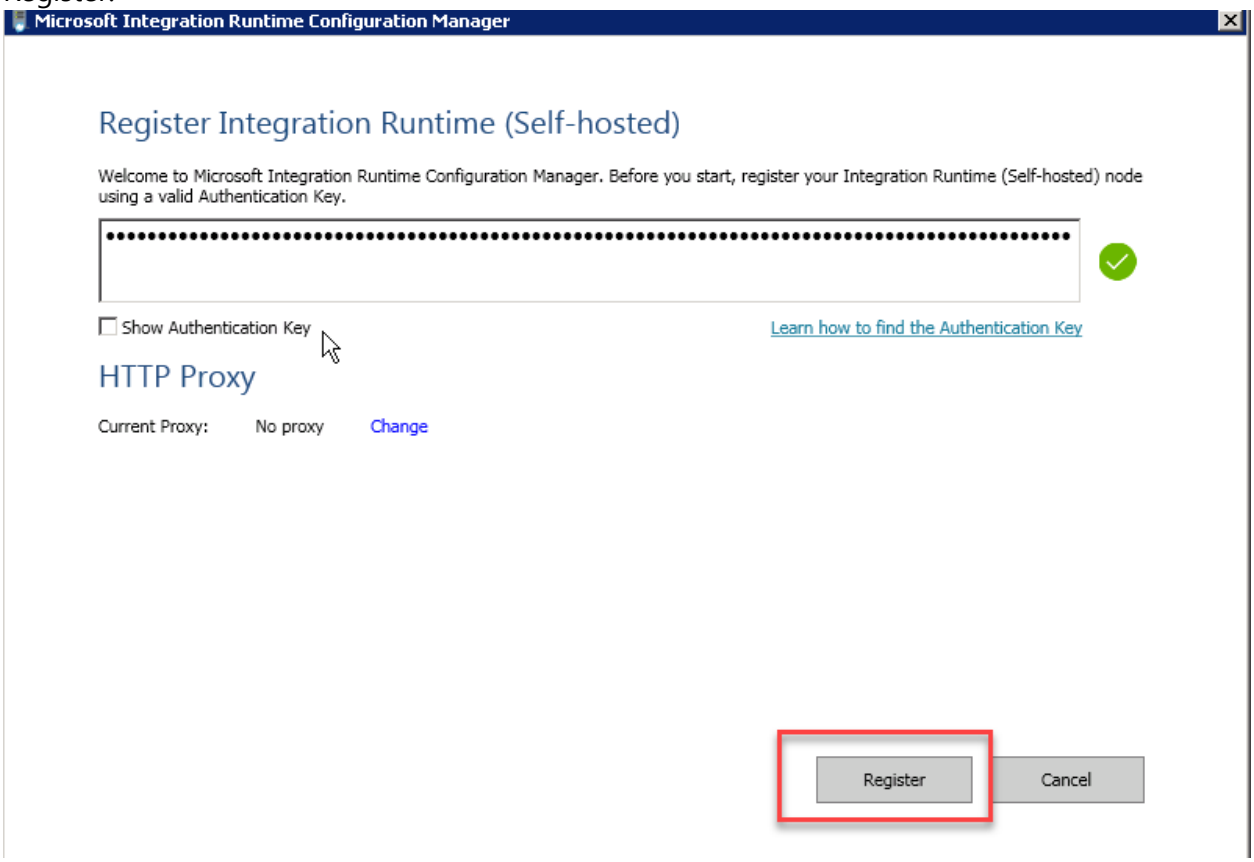
29. Once Downloaded, run the executable.



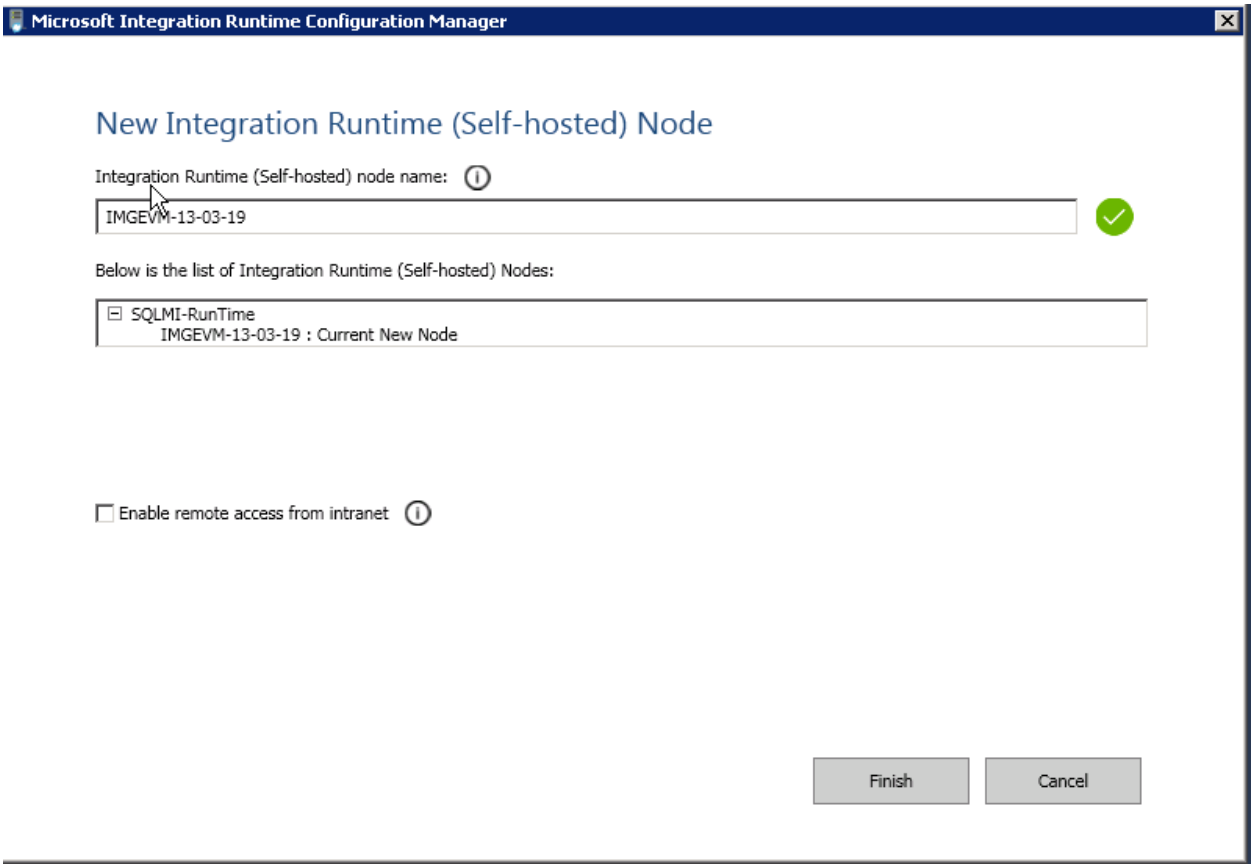
30. Click **Next**
31. Accept the terms and conditions and **proceed**.
32. Accept Default Location and Proceed further for installation
33. Click **Finish** once completed.



34. Upon Finishing, it'll launch the Configuration Wizard. Enter the key you copied and click Register.

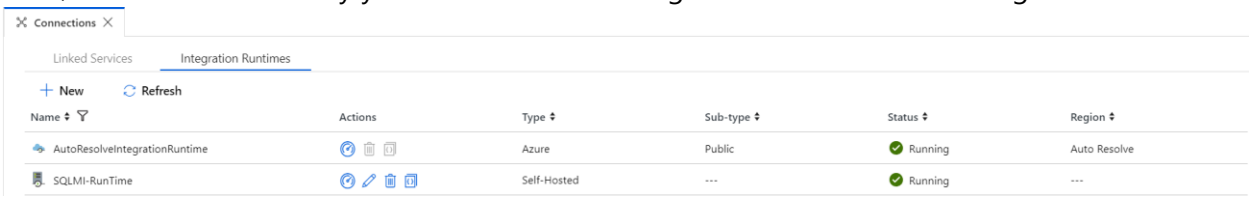


35. Review the settings and click Finish



36. Click Close once configuration is completed.

37. Now, In Azure Data Factory you should see this Integration Runtime as Running.



38. Self-Hosted Integration run time setup for SQL managed Instance is now completed.

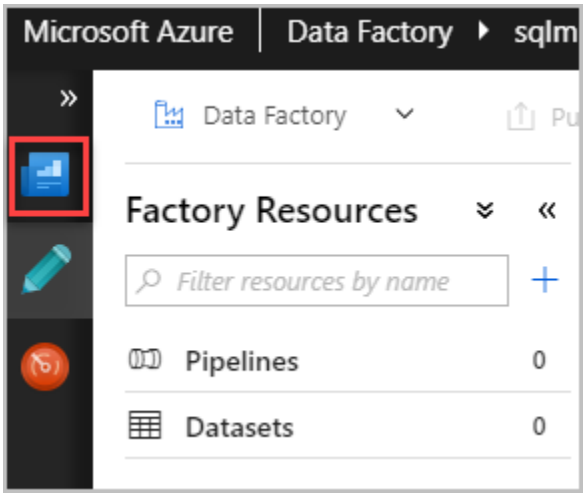
Exercise 1: Use the Copy Data Tool to Create and Run Data Pipeline

Overview

In this exercise, we'll use copy data tool to copy data from on-prem SQL Server to SQL MI.

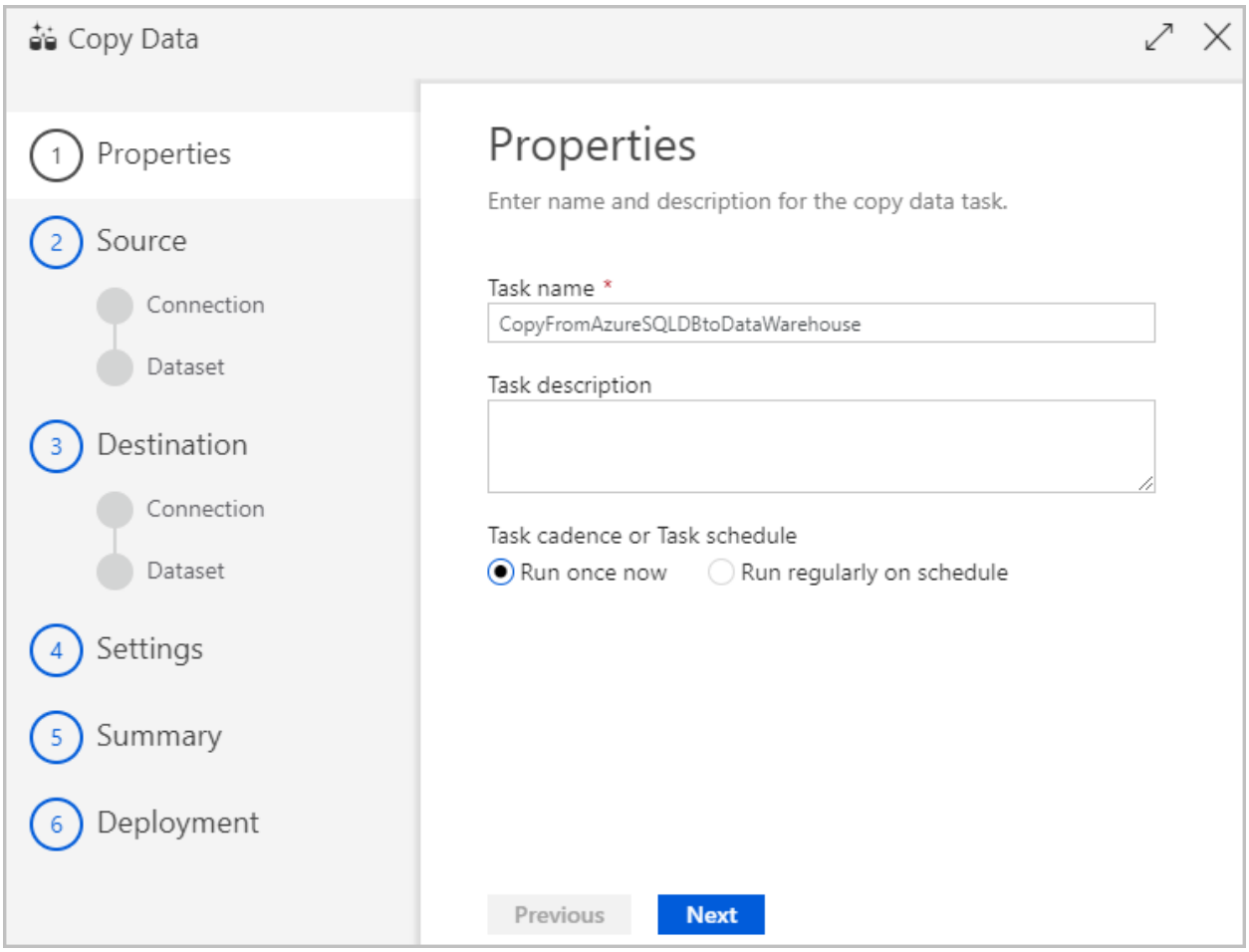
Task 1:

1. On the left-hand column of the screen, click the blue **Data Factory** icon to return to the home screen

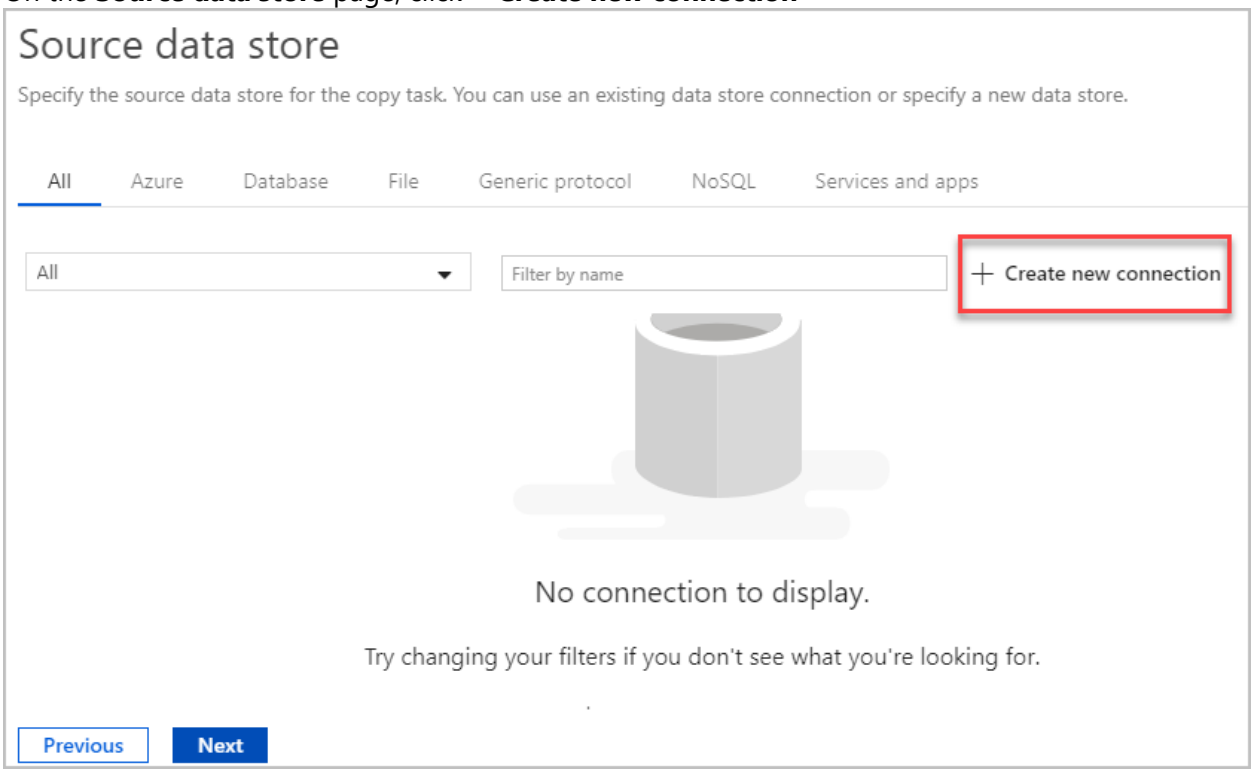


2. On the **Let's get started** page, select the **Copy Data** tile to launch the **Copy Data tool**

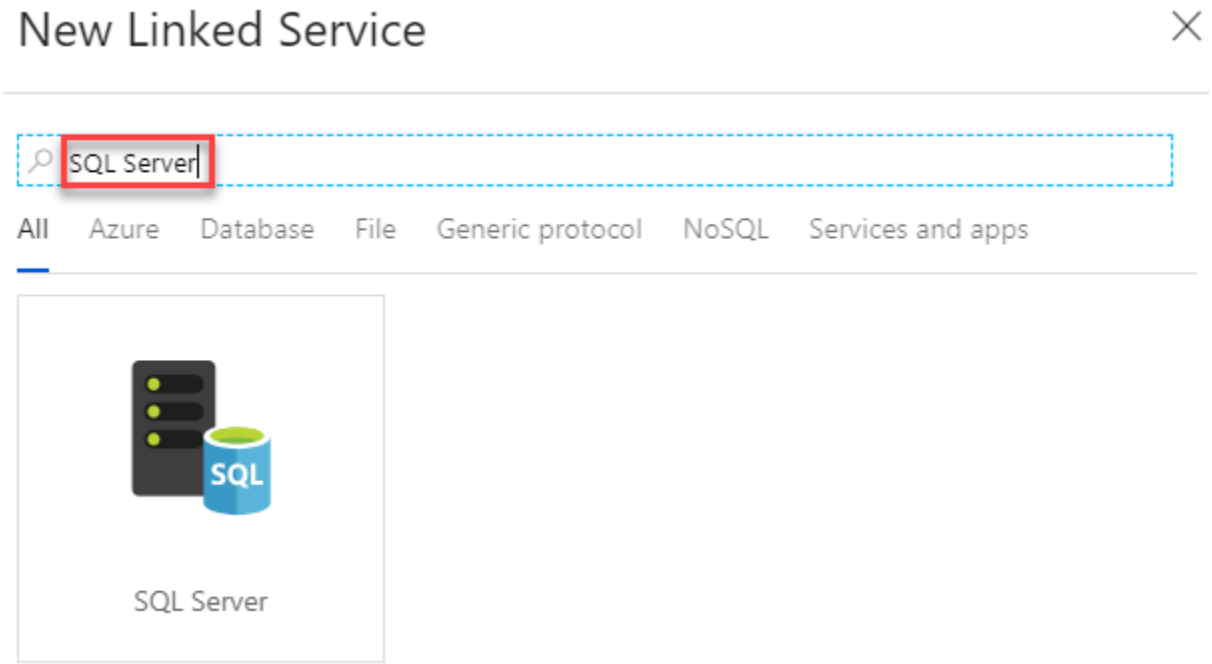
3. On the **Properties** page, under **Task name**, enter **CopyFromAzureSQLDBtoMI,** then select **Next**



4. On the **Source data store** page, click **+ Create new connection**

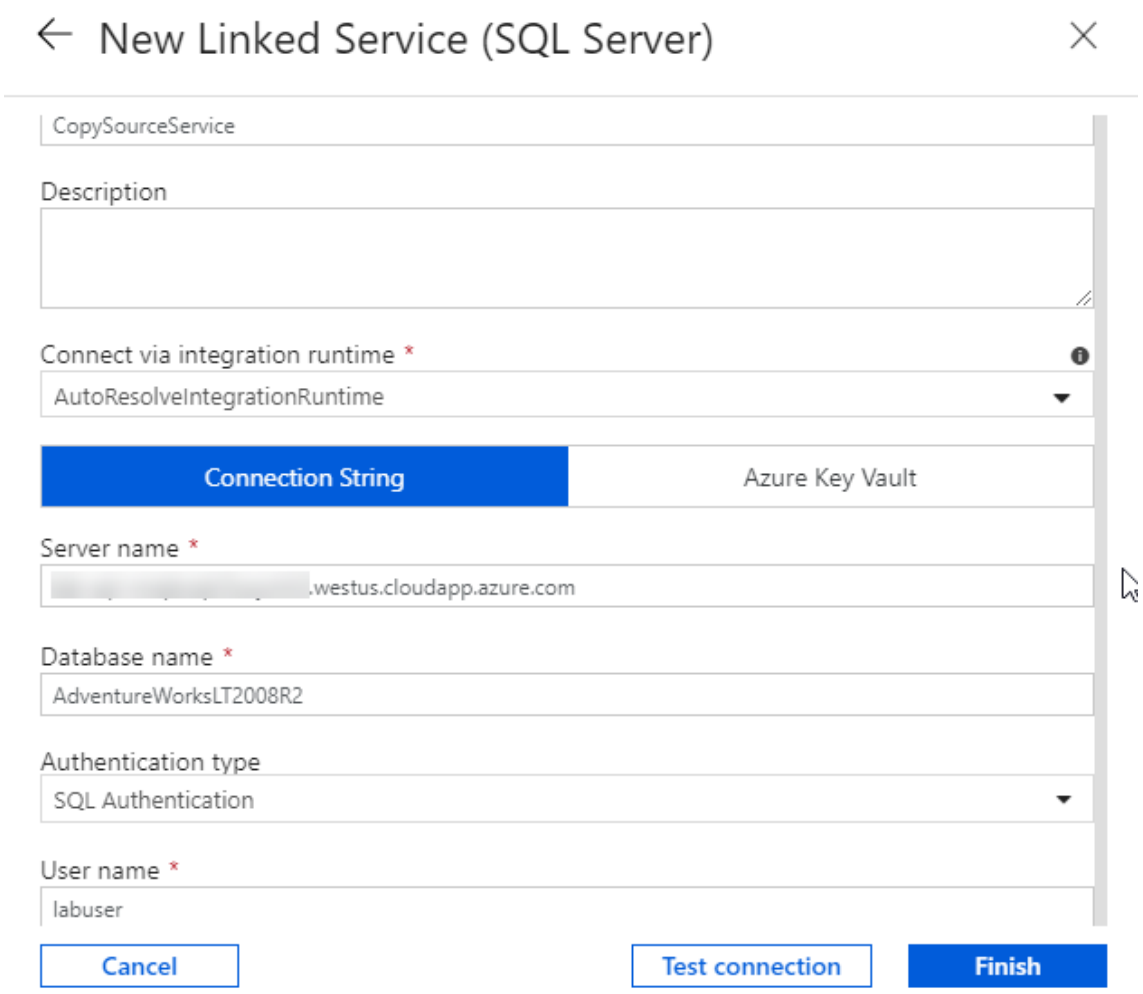


5. Search **SQL Server** and select **SQL Server** from the gallery, and then select **Continue**



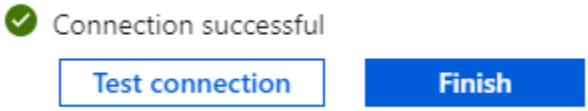
6. On the New Linked Service page, enter the following information:
    a. **Name:** Enter **CopySourceService**
    b. **Description:** Leave blank
    c. **Connection via Integration Runtime:** Select **AutoResolveIntegrationRuntime**

d. **Connection String or Azure Key Vault:** Select **Connection String**
e. **Server name:** Provide SQL Server VM DNS Name
f. **Database Name:** Provide **the database name of the database**
g. **Authentication Type:** Select **SQL Authentication**
h. **User name:** labuser
i. **Password:** demopassword1!

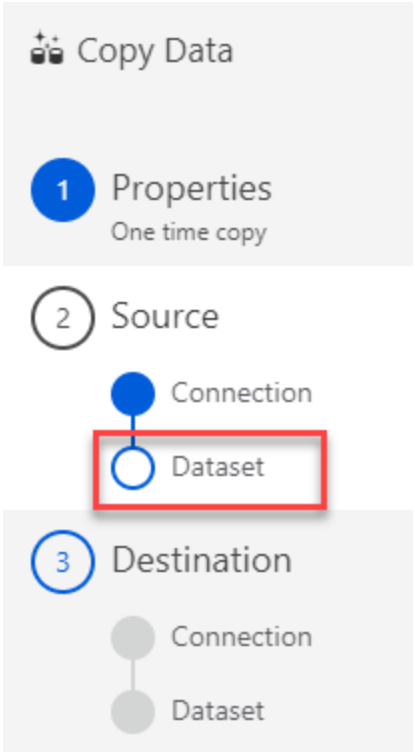


j. **Click on Test connection**

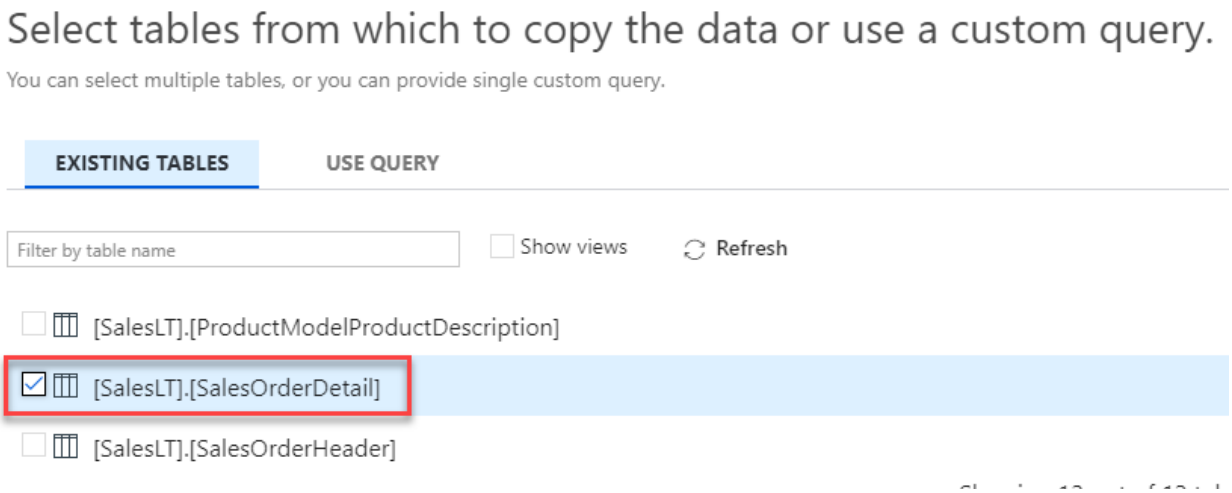
k. Once connection is successful, Click on **Finish**



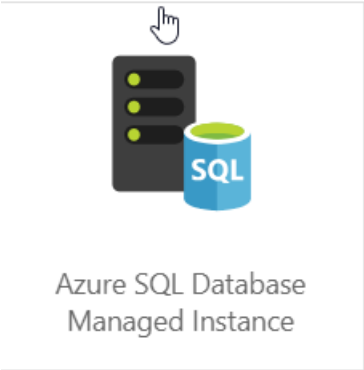
7. Select the newly created linked service as source, then click **on Dataset**



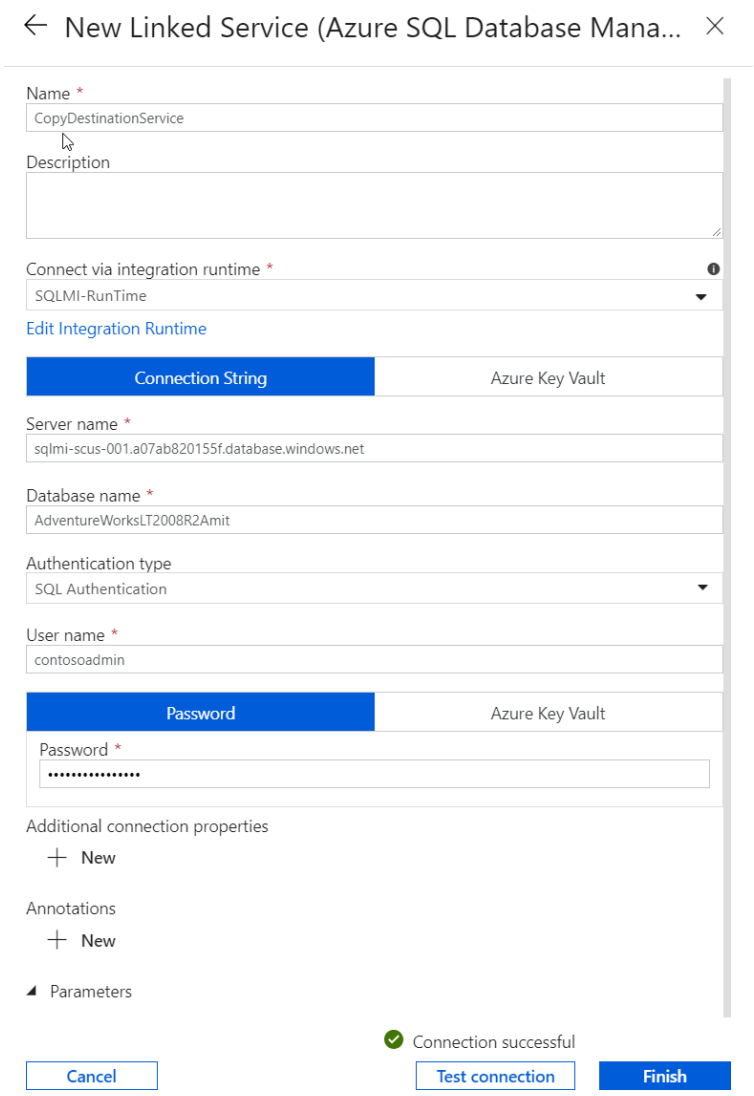
8. On the **Select tables from which to copy the data or use a custom query** page, select the Table **[**[SalesLT].[SalesOrderDetail] then click **next**



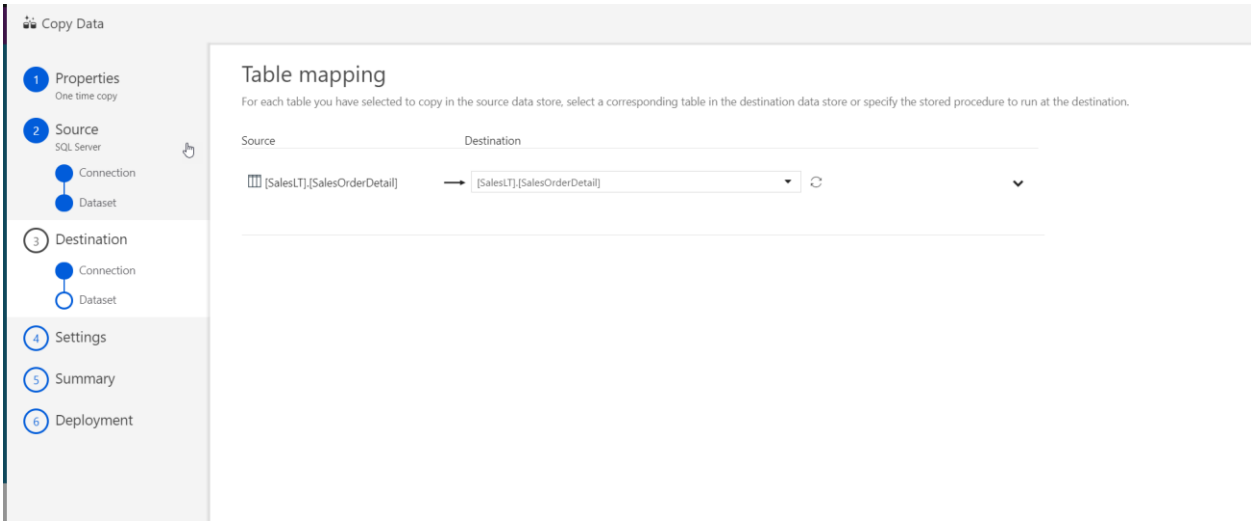
9. On the **Destination data store** page, click **+ Create new connection** to add a connection
10. Select **Azure SQL Database Managed Instance** from the gallery, and then select **Continue**



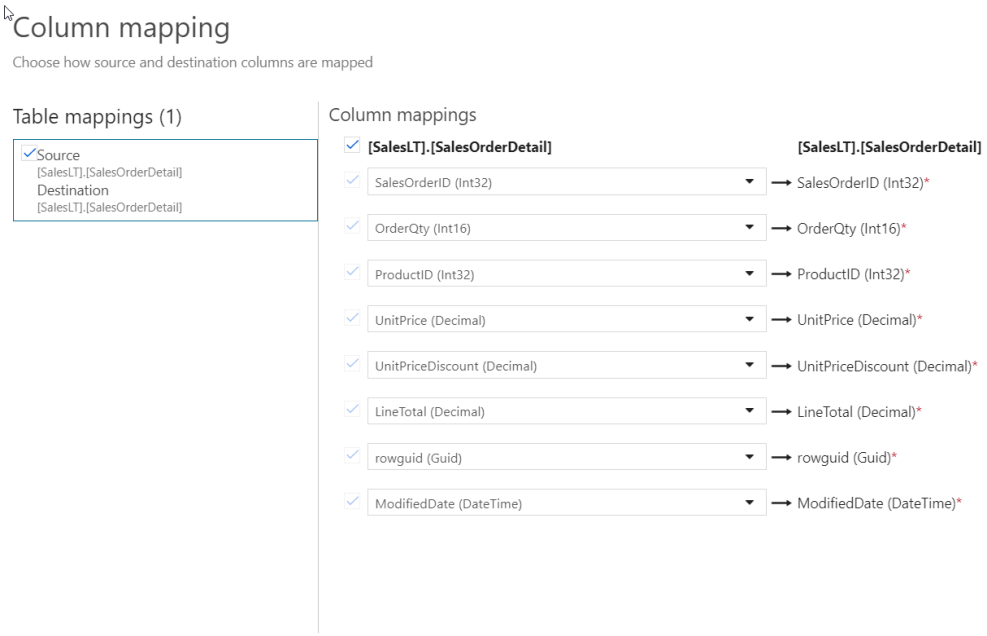
11. On the **New Linked Service** page, enter the following information:
    a. **Name:** Enter **CopyDestinationService**
    b. **Description:** Leave blank
    c. **Connection via Integration Runtime:** Select **SQLMI-Runtime (Your previously created runtime)**
    d. **Connection String or Azure Key Vault:** Select **Connection String**
    e. **Server name:** Provide SQL MI Server Name
    f. **Database Name:** Enter the database you created earlier for this.
    g. **Authentication Type:** Select **SQL Authentication**
    h. **User name:** Enter **contosoadmin**
    i. **Password:** Enter **IAE5fAijit0w^rDM**
    j. Click **Finish**



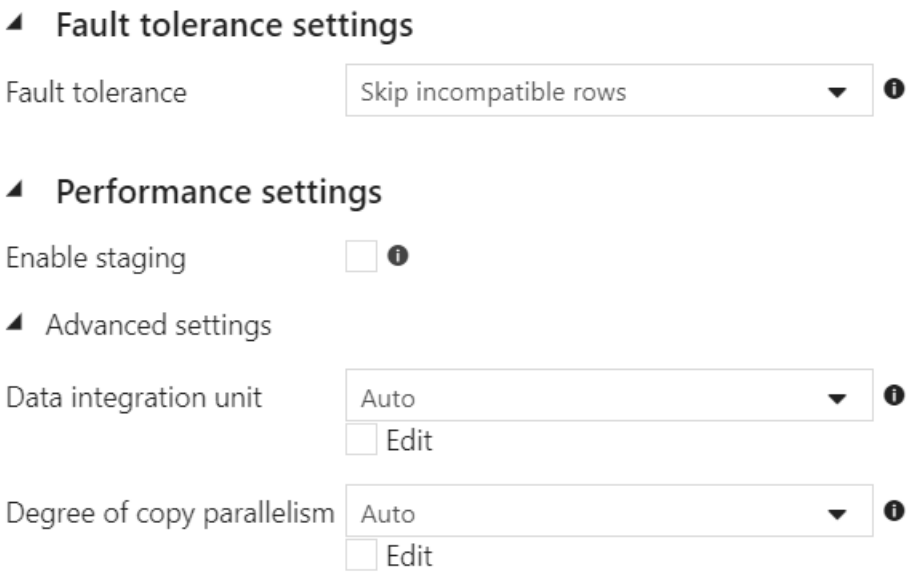
12. Select the newly created linked service as sink, then click **Next**
13. On the **Table mapping** page,Select the corresponding table and select **Next**



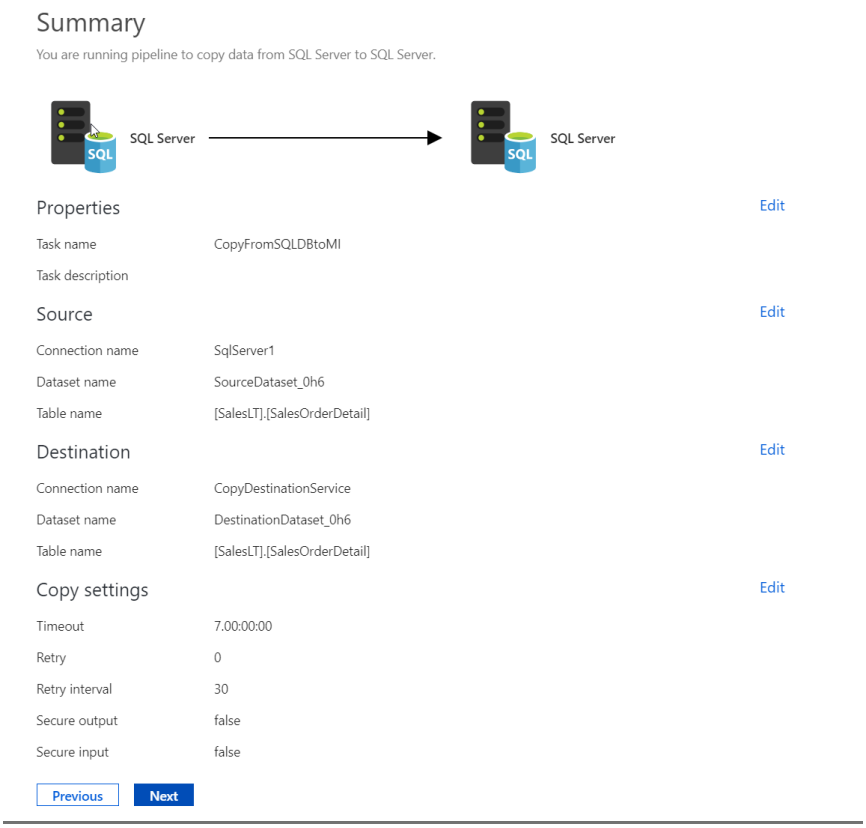
14. On the **Column mapping** page, select **Next**



15. On the **Settings** page, Change fault tolerance to Skip Incompatible rows, Uncheck “Enable Staging” and click Next

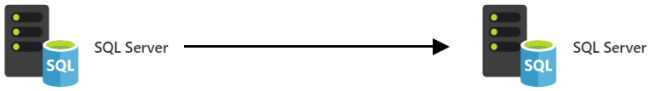


16. On the **Summary** page, review the settings, and then select **Next**



17. On the **Deployment** page, select **Monitor** to monitor the pipeline (task)

    NOTE: Notice that the Monitor tab on the left is automatically selected. The Actions column includes links to view activity run details and to rerun the pipeline



Deployment complete

- Creating Datasets
- Creating Pipelines
- Running Pipelines

Datasets and pipelines have been created. You can now monitor and edit the copy pipelines or click finish to close the copy wizard.

Edit Pipeline Monitor

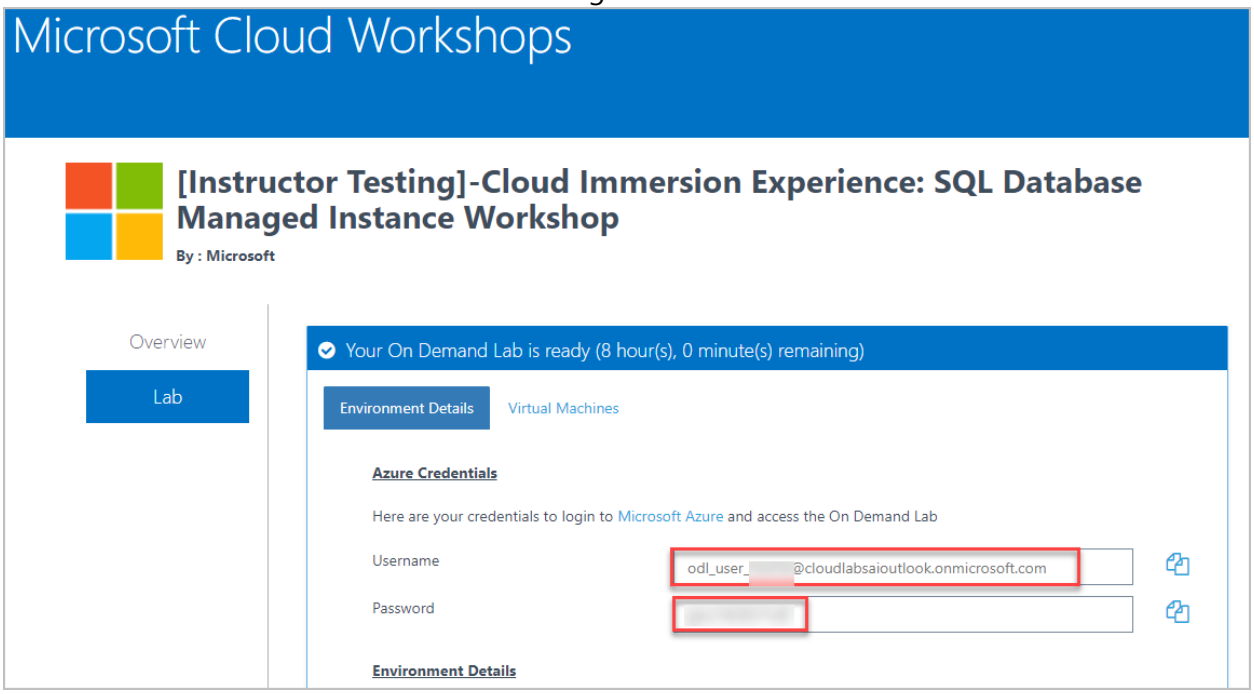
Exercise 2: Provisioning an Azure-SSIS Integration Runtime in Azure Data Factory

Overview

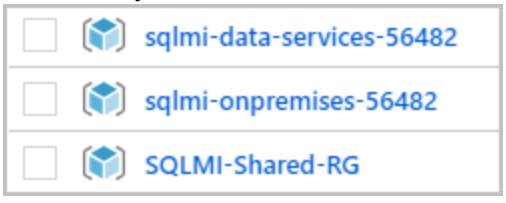
This section will provide you with the ability to use the Azure Portal to create the foundation needed for the Integration Runtime's infrastructure to effectively host and manage the executions of SSIS packages residing virtually anywhere in the world.

Task 1: Configure Integration Runtime Setup

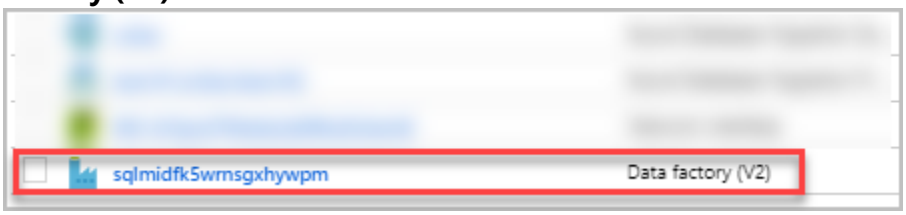
1. Launch a browser and navigate to https://portal.azure.com. Once prompted, login with the Azure Credentials from the Lab Details Page.



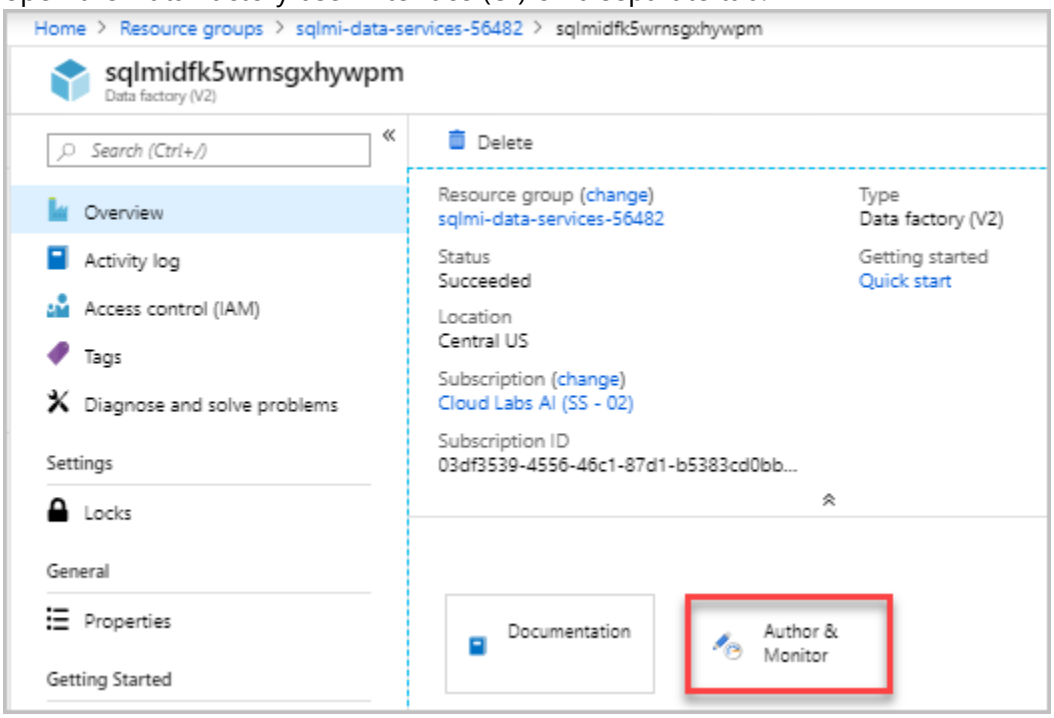
2. In the Stay signed in? pop-up window, click No
3. In the Welcome to Microsoft Azure pop-up window, click Maybe Later
   Note: If you receive a pop-up for Azure Advisor, click the X in the top right corner of the pop-up to close it.
4. You will be directed to the dashboard.
5. From the left side of the Page, select **Resource Groups**
6. Note that you will have access to three Resource groups:



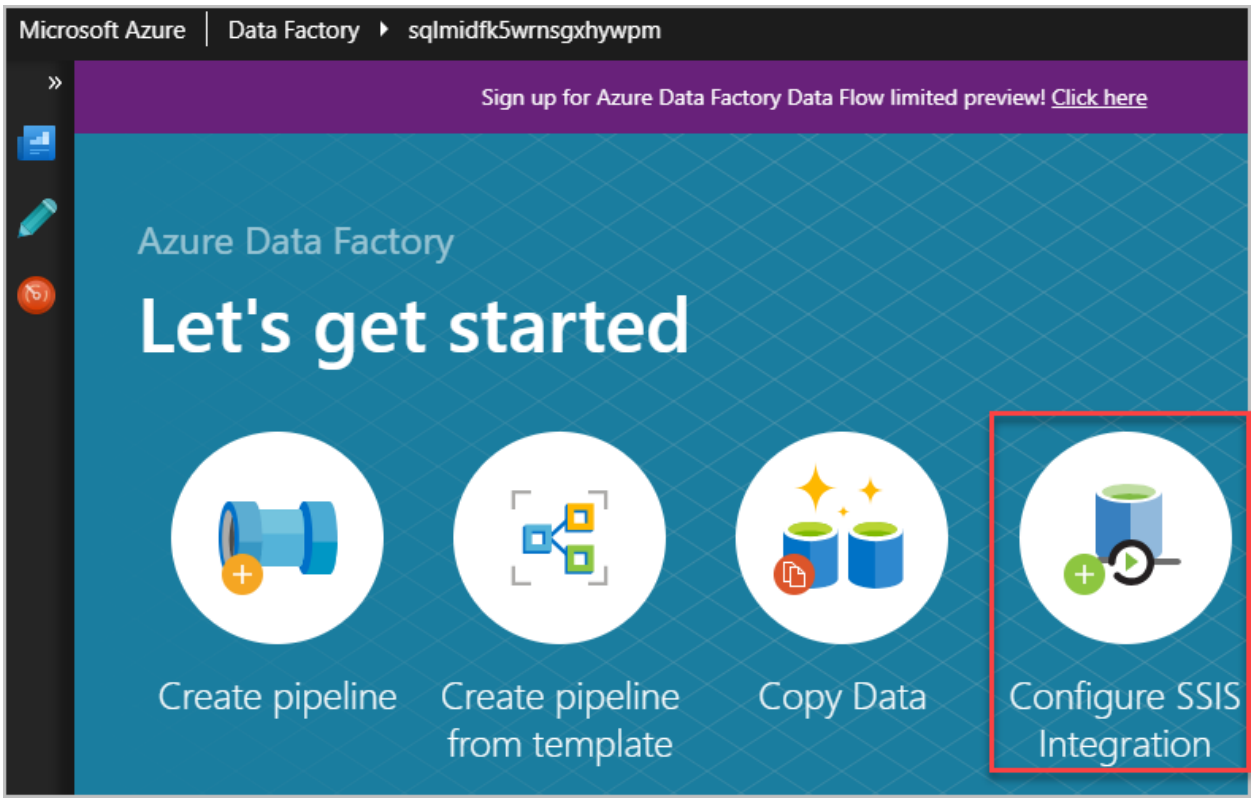
7. Select resource group starting with **sqlmi-data-services-**
8. From the overview blade of the Resource group, **select** the resource with type **Data Factory (V2)** from the resources:



9. From the overview blade of the Data Factory, navigate to the Author & Monitor tile to open the Data Factory user Interface (UI) on a separate tab.



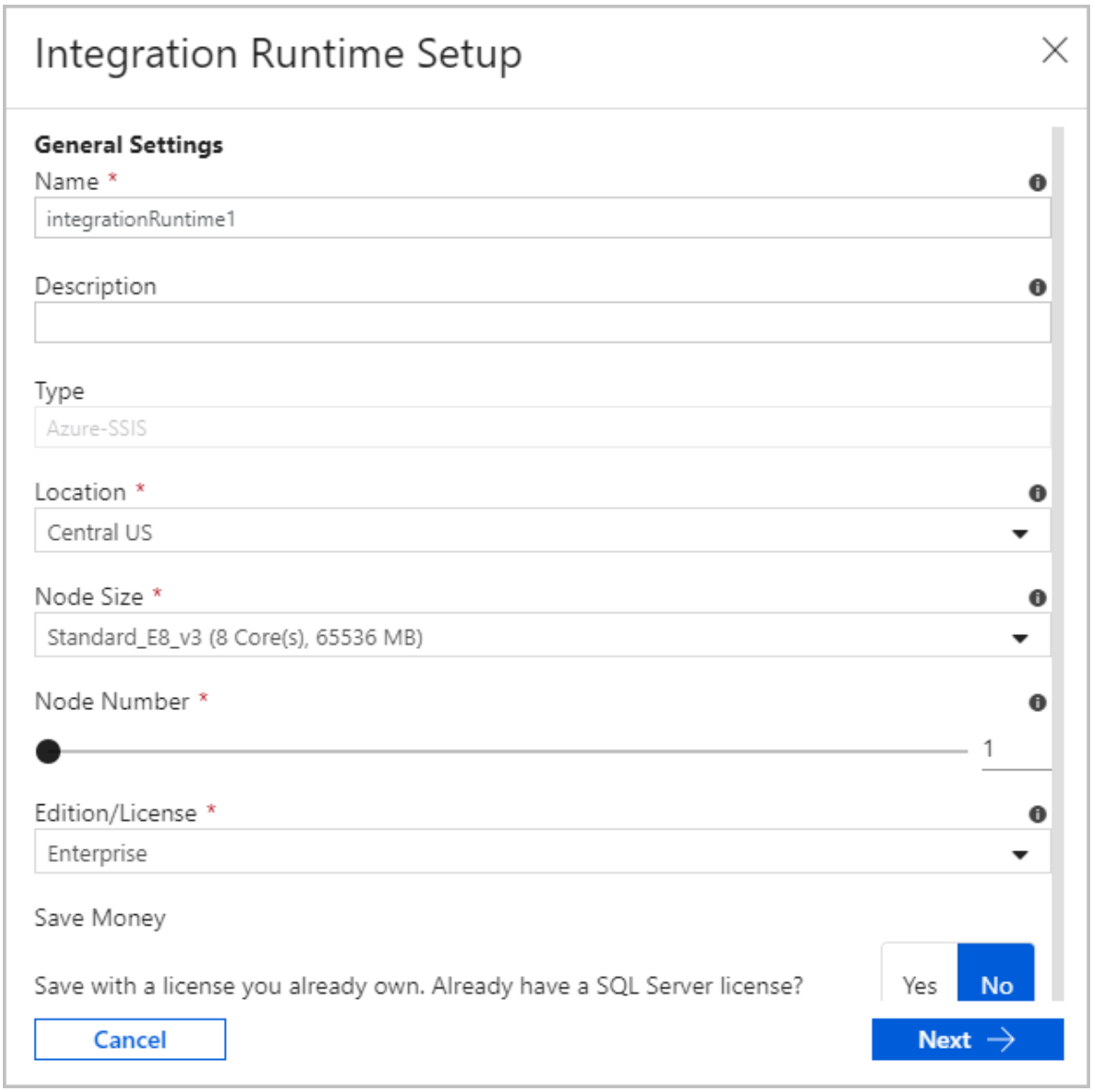
10. On the **Let's get started** page, click the **Configure SSIS Integration Runtime** tile



11. On the **General Settings** form of Integration Runtime Setup page, enter the following information:
    a. **Name:** integrationRuntime1
    b. **Description:** Leave blank
    c. **Location:** Select **location of the resource group** from the dropdown

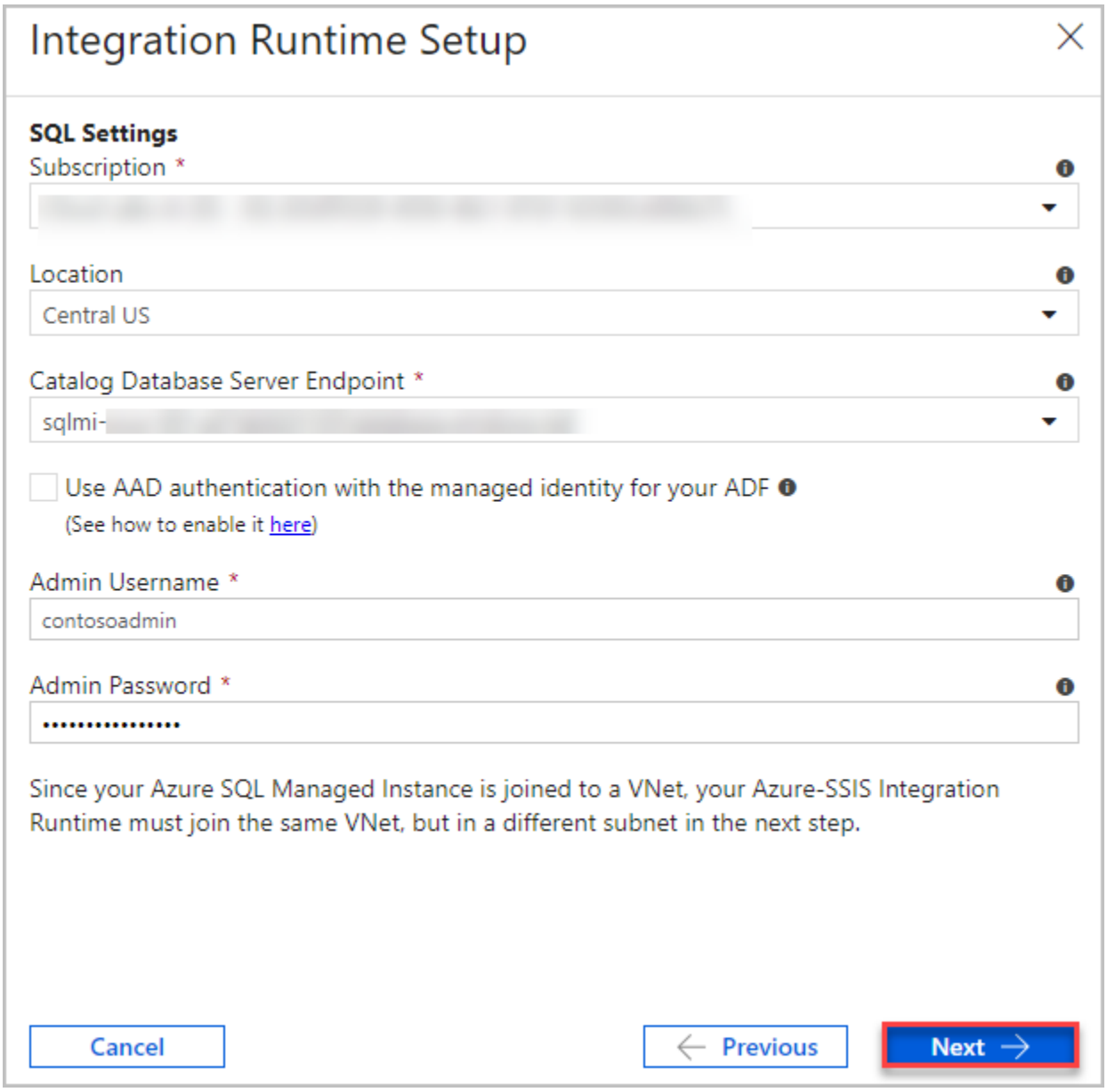
NOTE: The location does not need to be that of your data factory's, but it should be the same as your Azure SQL Database/Managed Instance server's where your **SSISDB** is to be hosted, allowing easy access without incurring excessive traffic.

    d. **Node Size:** Select **Standard_E8_v3** from the dropdown *This node's high memory-to-CPU ratio results in superior performance when working with **large database servers** - which this lab will. Another case where selecting a large node size would be beneficial is when you anticipate running **compute/memory-intensive SSIS packages**
    e. **Node Number:** Select **1** by using the slider. *The number of nodes you select determines the cluster size of your IR, which impacts performance when running parallel SSIS packages.* ***Select a large cluster if you plan on running many packages in parallel***
    f. **Edition/License:** Select **Enterprise** from the dropdown, as it will allow the use of advanced/premium features on your integration runtime, showcasing various features during this lab
    g. **Save Money:** Click **No**



12. Click **Next** to bring up the **SQL Settings** portion of Integration Runtime Setup page, and provide the following information:
13. **Subscription:** Select your default subscription
14. **Location:** Select your resource group location from the dropdown, as it is recommended that you select the same location of your integration runtime
15. **Catalog Database Server Endpoint:** Select the available Database Server Endpoint from the Dropdown.
16. **Do not** check **Use AAD authentication with your ADF MSI**
17. Admin Username**: contosoadmin**

18. Admin Password**: IAE5fAijit0w^rDM**

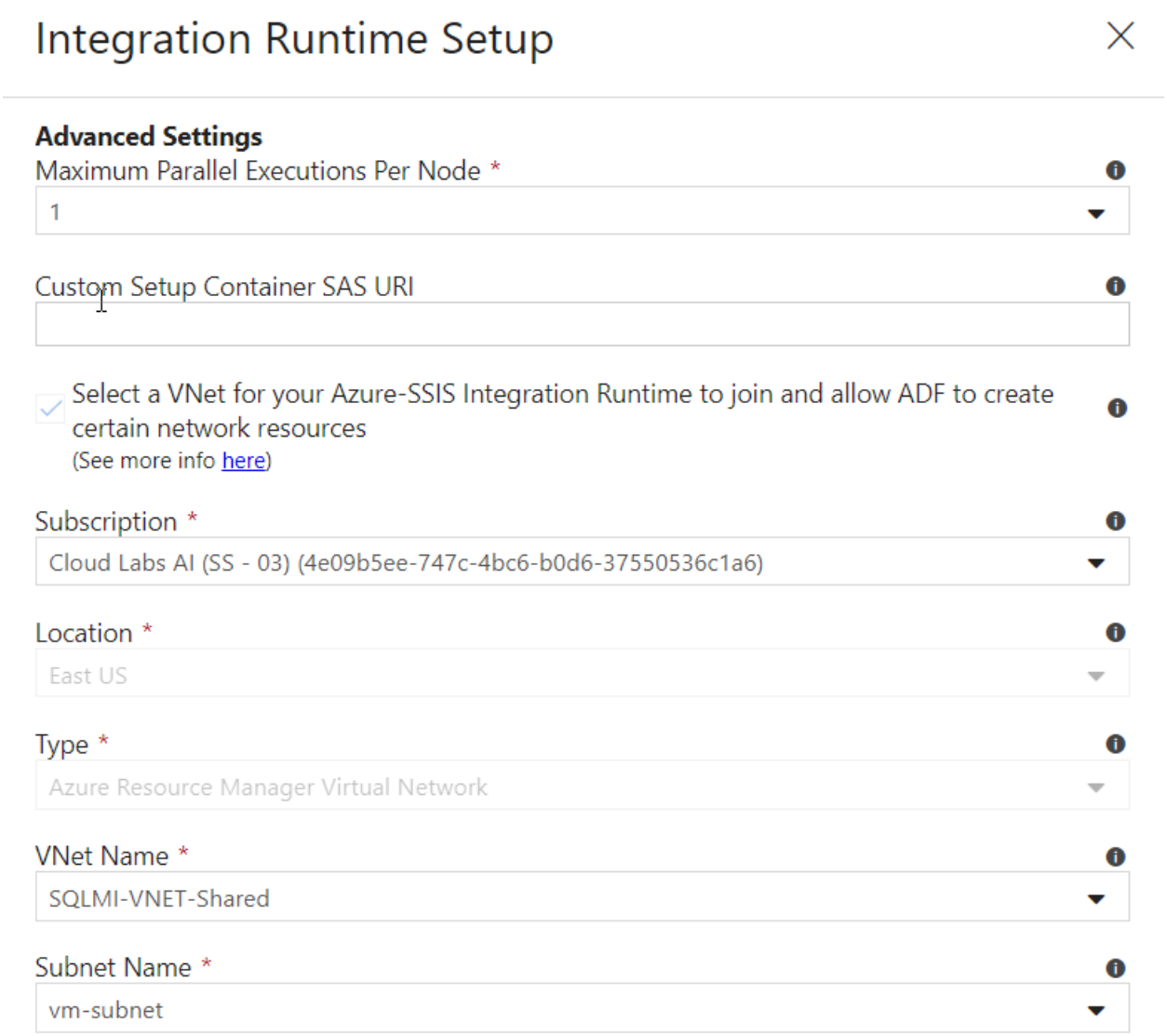


19. **Maximum Parallel Executions Per Node:** select **1** from the dropdown box

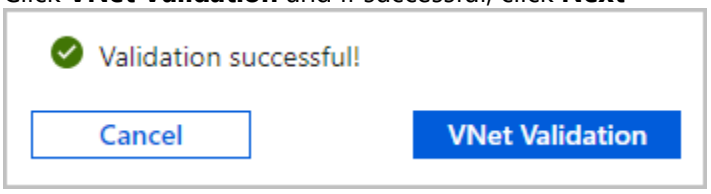
    NOTE: the maximum number of packages to execute concurrently per node in your integration runtime cluster. Only supported package numbers are displayed. Select a low number, if you want to use more than one cores to run a single large/heavy-weight package that is compute/memory -intensive. Select a high number, if you want to run one or more small/light-weight packages in a single core.

20. **Custom Setup Container SAS URI:** Leave blank
21. **Select a VNet…:** Leave checked
22. **Subnet:** Choose vm-subnet

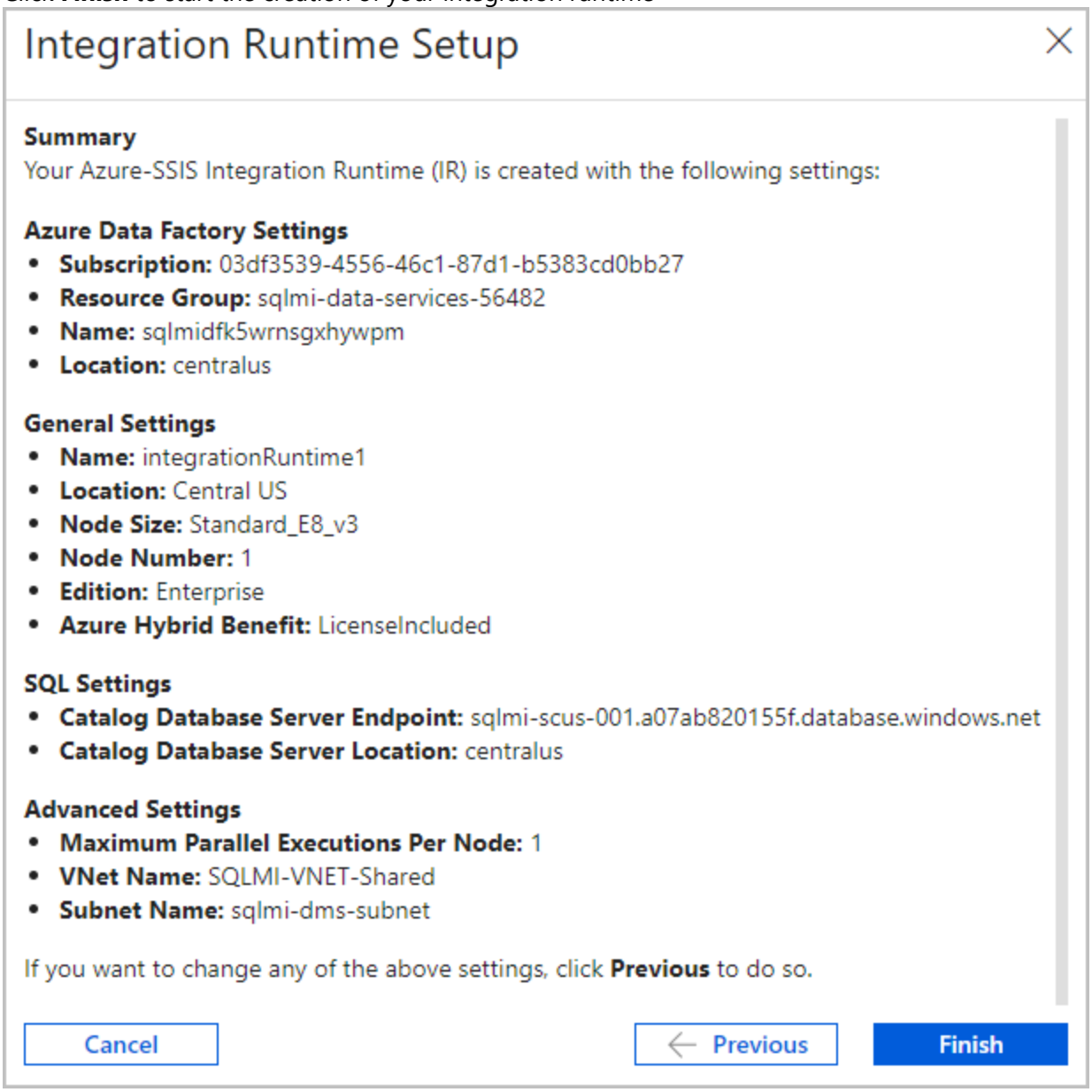
23. Select the subscription from the dropdown.



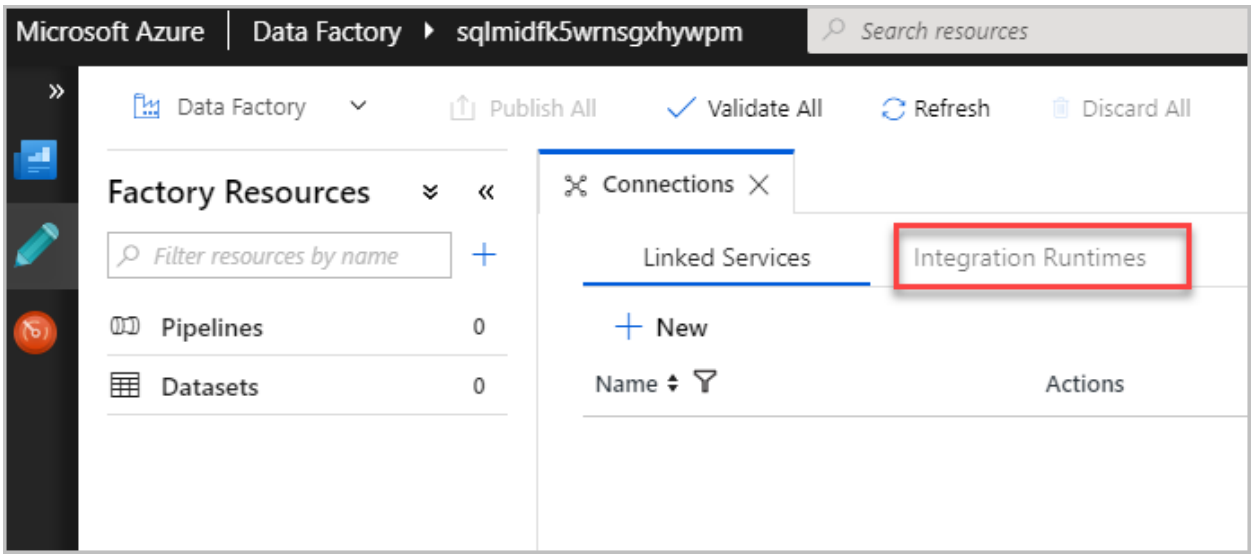
24. Click **VNet Validation** and if successful, click **Next**

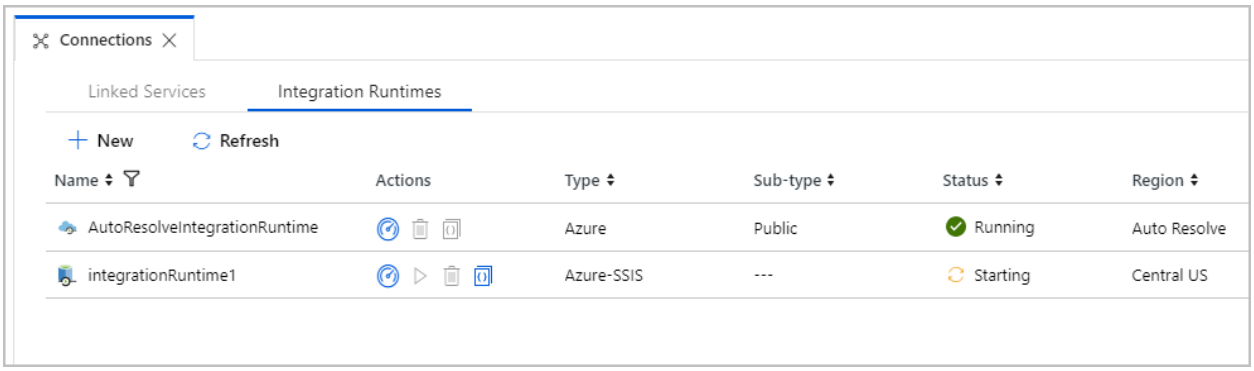


25. Click **Finish** to start the creation of your integration runtime



26. On the **Connections** tab, switch to **Integration Runtimes** if needed. Select **Refresh** to refresh the status



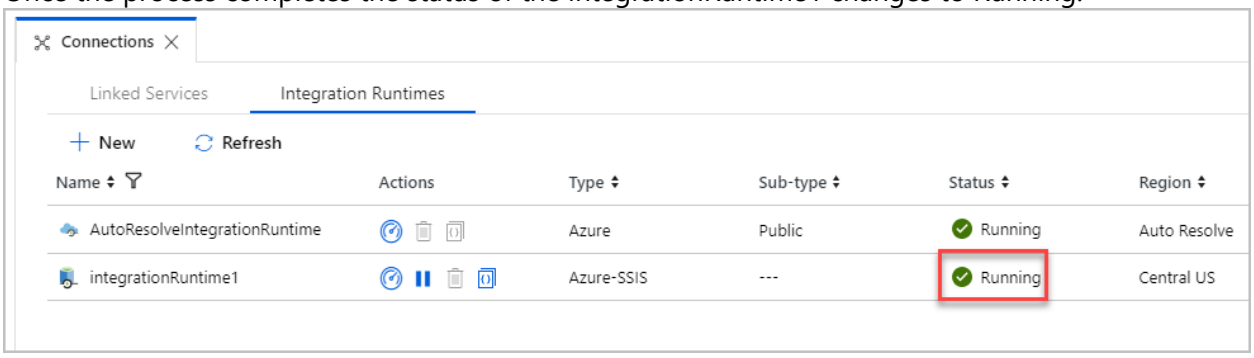


NOTE: This process takes approximately **20 to 30 minutes** to complete due to the Azure Feature Pack for SSIS and the Access Redistributable installations. This is taking place while the Data Factory service connects to your Azure SQL Database server to prepare the **SSIS Catalog** (SSISDB database).

NOTE: Use the links in the **Actions** column to stop/start, edit, or delete the integration runtime. Use the last link to view JSON code for the integration runtime. The edit and delete buttons are enabled only when the IR is stopped.

27. Please continue with the lab while the integration runtime installs

28. Once the process completes the status of the integrationRuntime1 changes to Running.



Exercise 3: Deploy and Run SSIS Packages from Azure Data Factory SSIS IR

Overview

In this exercise, you'll deploy and run SSIS package from Azure Data Factory SSIS IR

Task 1:

Verify the Integration Runtime you created earlier in the lab has completed.

1. On the monitor page, click the **Integration Runtimes** tab near the top of the page
2. Verify that **IntegrationRunTime1** a Status of **Running**, If its status is **Starting** then wait for the Runtime to complete before continuing.

   NOTE: Click the refresh button to update the status

   NOTE: This process takes approximately **20 to 30 minutes**

Connect to SSISDB

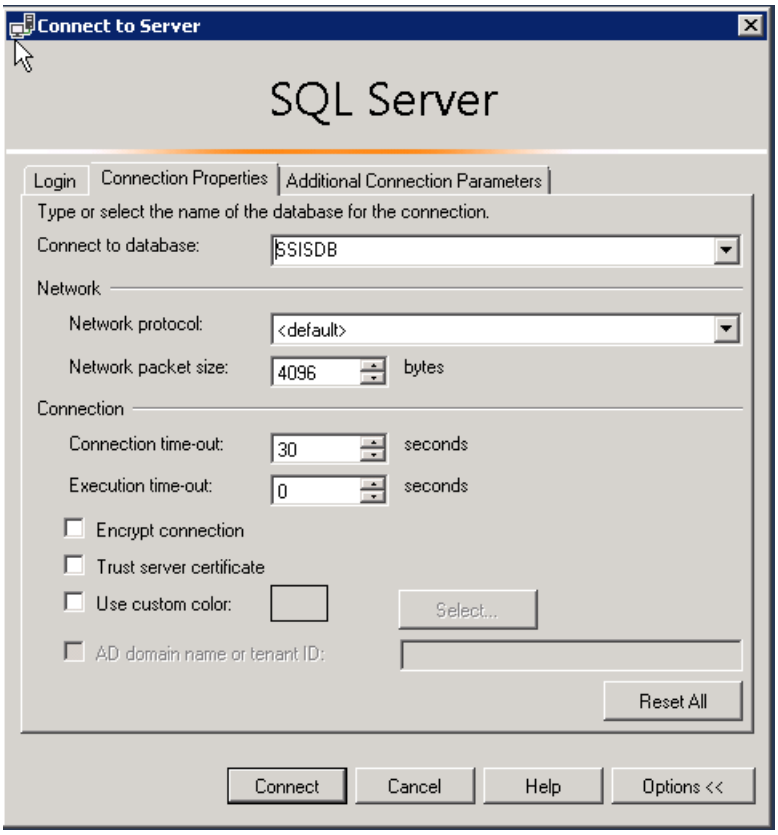
To deploy and then run the package on Azure SQL Database, you'll need to first **connect** to the SSIS Catalog database (SSISDB). To do so, complete the following steps:

1. Login to the SQL Server Virtual Machine via Remote desktop.
2. Double click **SSMS** icon on desktop to launch **SQL Server Management Studio**
3. In the **Connect to Server** dialog box, enter the following information:
4. **Server name:** Hostname of your SQL Managed Instance
5. **Authentication:** Select **SQL Server Authentication**
    1. **User name:** Enter **contosoadmin**
    2. **Password:** Enter **IAE5fAijit0w^rDM**

6. Click the **Options** button
7. Click on the **Connection Properties** tab
8. Click on the dropdown for selecting a **database**

   NOTE: If a dialogue box appears and asks to connect first, click **Yes**.

9. Click **SSISDB** and make sure it's highlighted



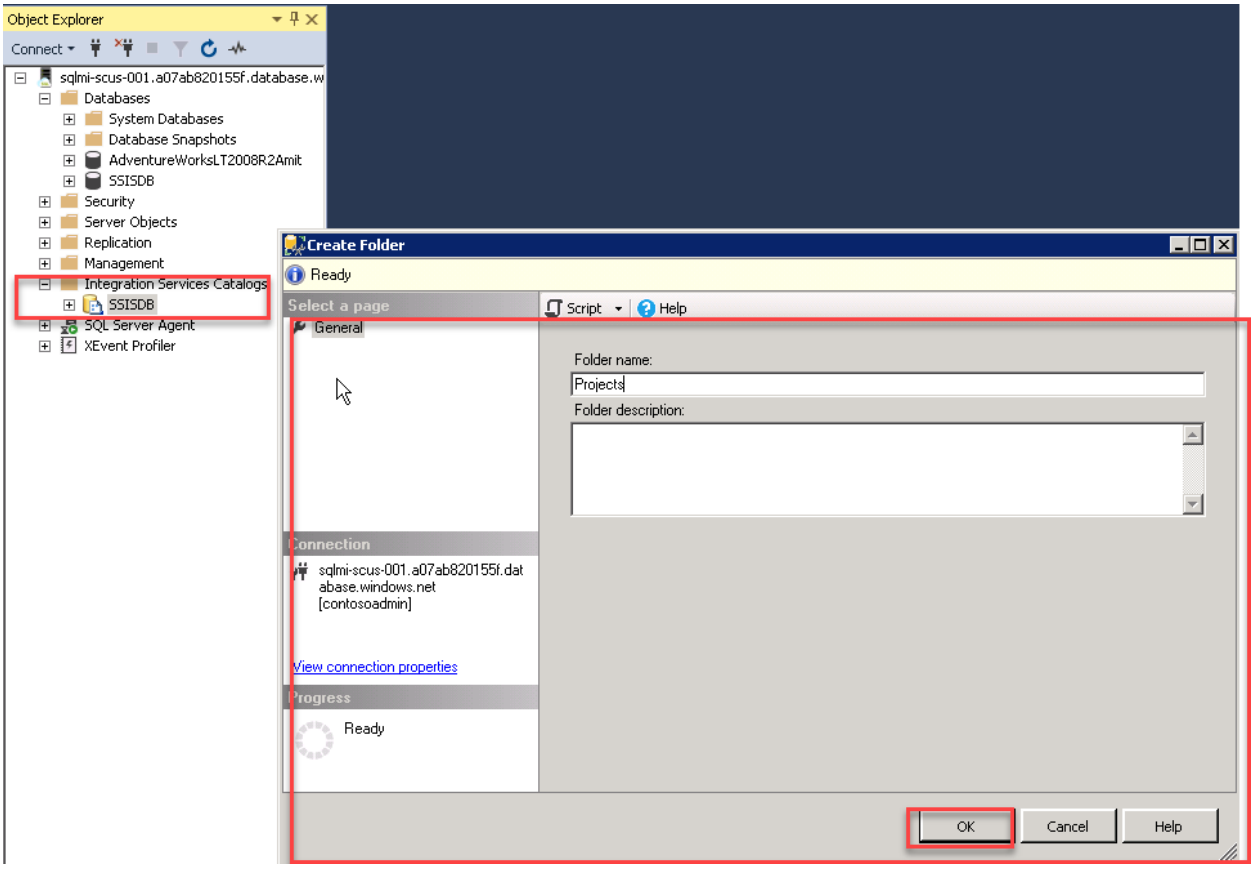
10. Click **Ok**
11. In the **Object Window** on your left, navigate to **Integration Services Catalogs** and click the **Expand Object** icon (plus sign "+"), then expand the **SSISDB** object as well

Deploying a Project/Package

14. Under Object Explorer, Browse through Integration Service Catalogue

    Right click on **SSISDB**, select **Create Folder**

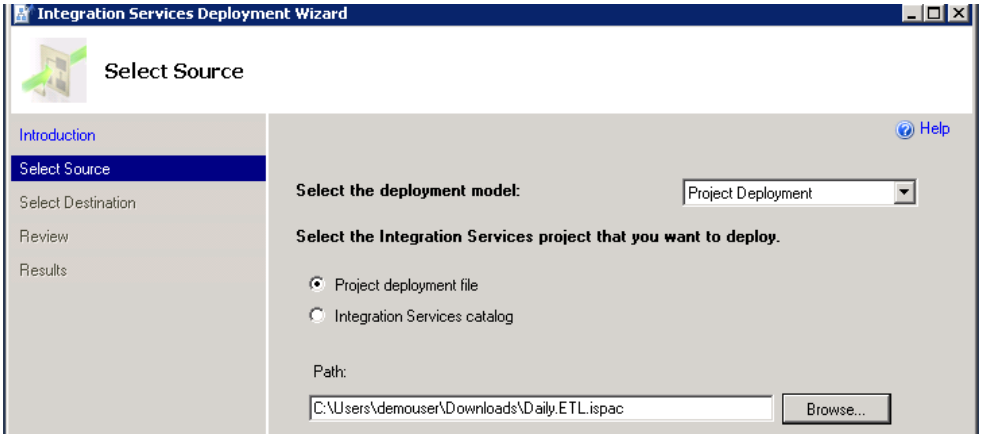
15. Name the folder **Myproject**, click **ok**



16. Expand My Project Click on the **Projects** node
17. Right-click on the **Projects** node and select **Deploy project** to launch the **Services Deployment Wizard**

    NOTE: You can deploy a project from the current catalog **or** from the file system.

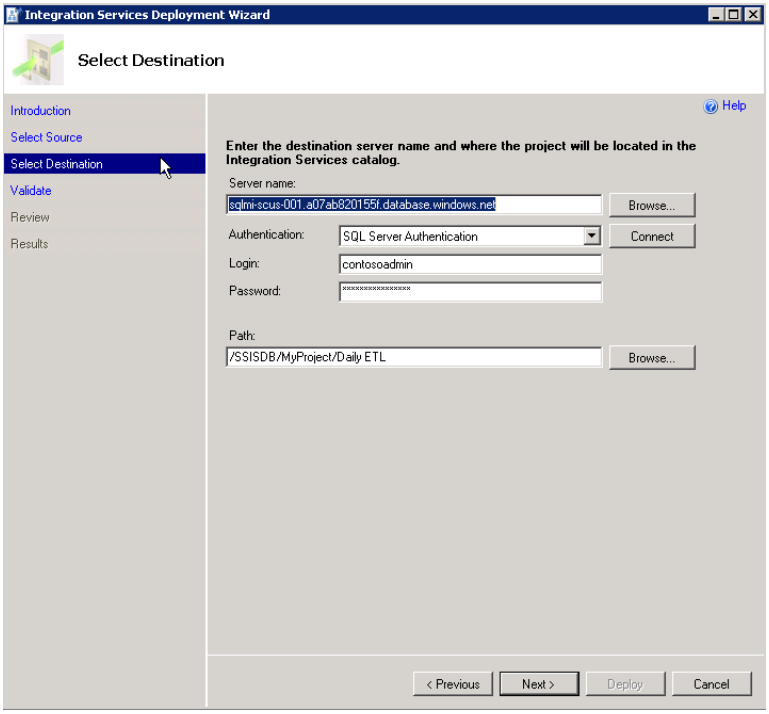
18. On the **Introduction** page of the wizard, review the introduction, then click **Next** to open the **Select Source** page
19. Download Daily.ETC.ispac from https://github.com/Microsoft/sql-server-samples/releases/download/wide-world-importers-v1.0/Daily.ETL.ispac
20. On the **Select Source** page, select **Project deployment file** and browse through file explorer to select latest download Daily.ETC.ispac file.



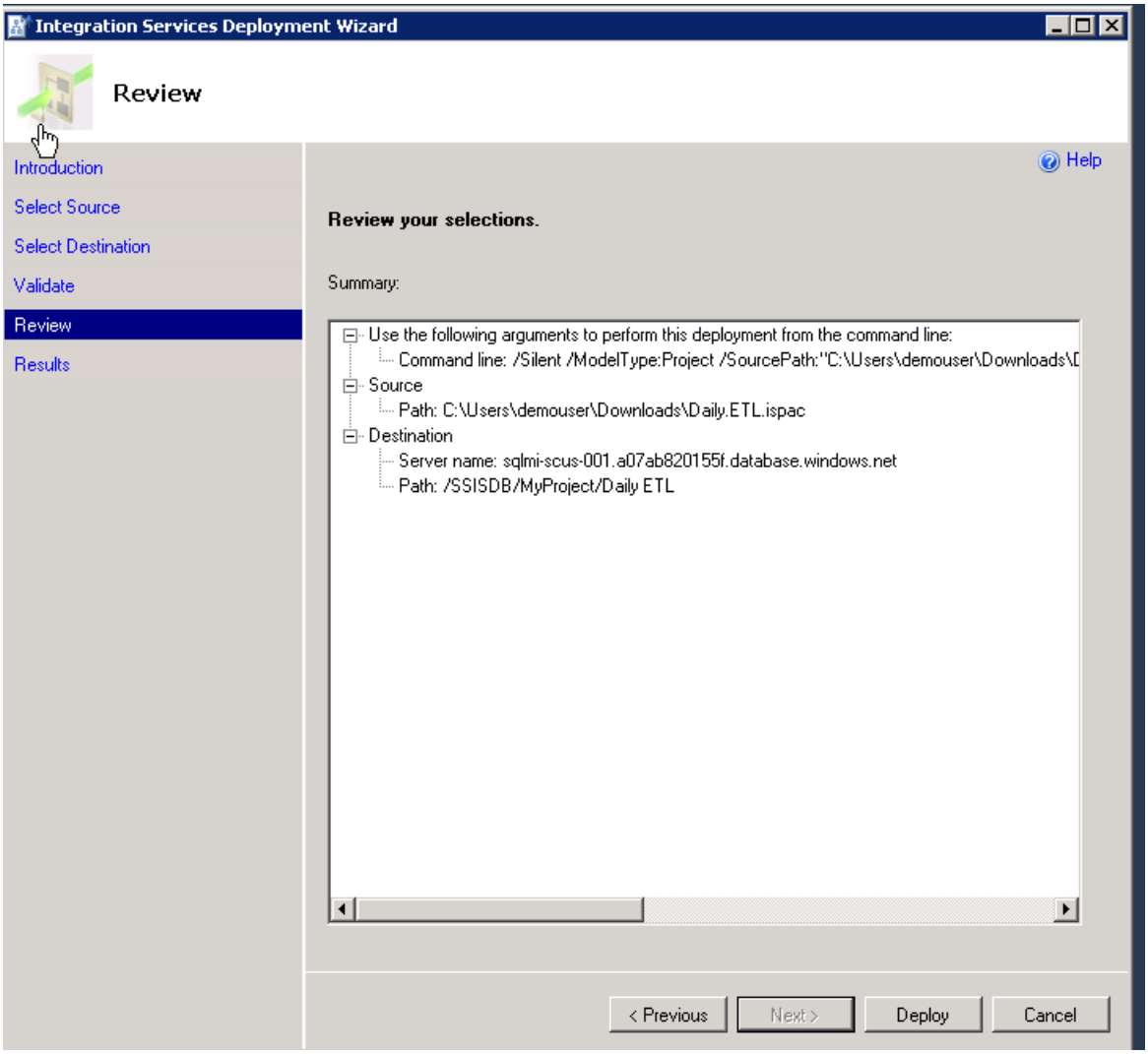
NOTE: To deploy a project that is already deployed to an SSIS catalog database, select **Integration Services** catalog, and then enter the server name and the path to the project in the catalog.

21. On the **Select Destination** page, select

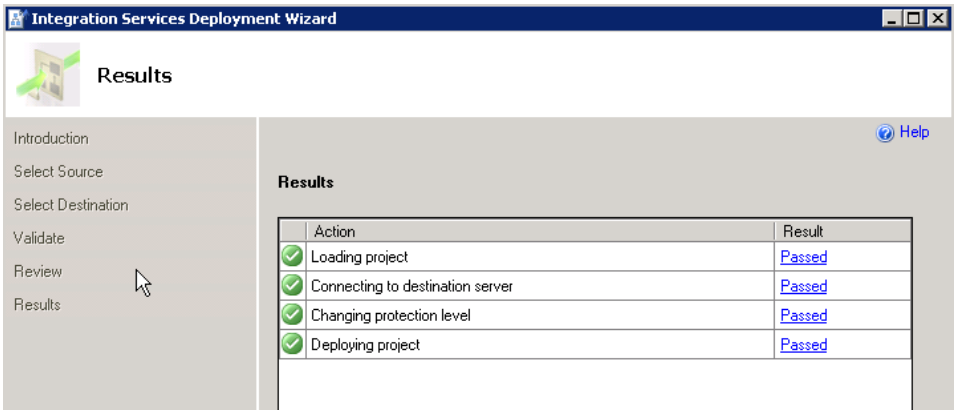
22. Server Name – **Keep Default (SQL MI Hostname)**
23. Authentication: **SQL**
24. **User name:** Enter **contosoadmin**
25. **Password:** Enter **IAE5fAijit0w^rDM**
26. Click **Connect**
27. Click **Next** after you **Connect**



28. On **Validate** page, click Next
29. On the **Review** page, review the settings you selected
30. Click **Deploy** to start the deployment process

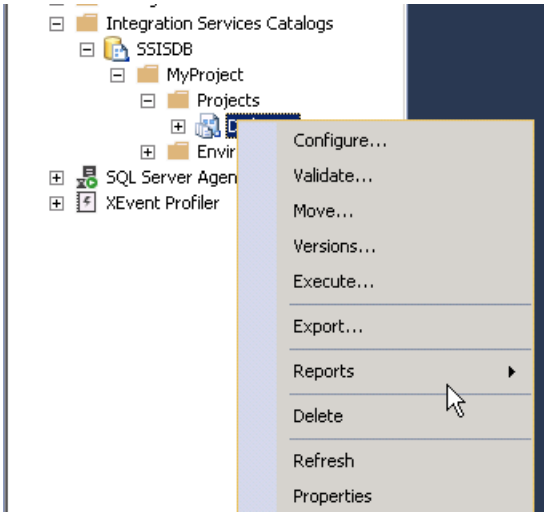


31. Once complete, the **Results** page will display successes and/or failures of any actions. Once complete, click **Close** to exit the wizard

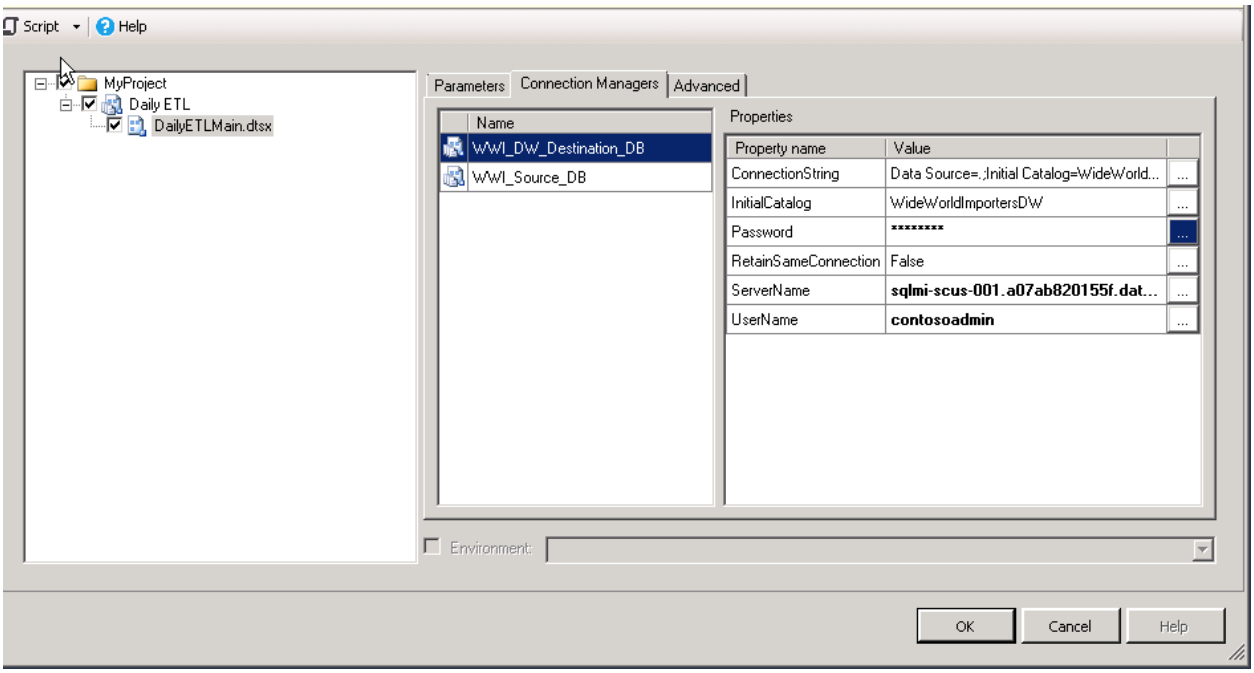


Running a Package

32. Refresh the **Projects** folder in SSMS
33. Select **Daily ETL**
34. Right-click and select **Execute**



35. After the **Execute Package** dialog box opens, click the **Connection Manager** tab
36. Here you'll need to change Source and Destination configuration. Be sure to use SQL Authentication, not Windows Authentication.
37. Enter SQL Server Managed Instance Hostname and Credentials for destination.



38. Click **OK** to run the package
39. Click **yes** to open **Execution Report** to review

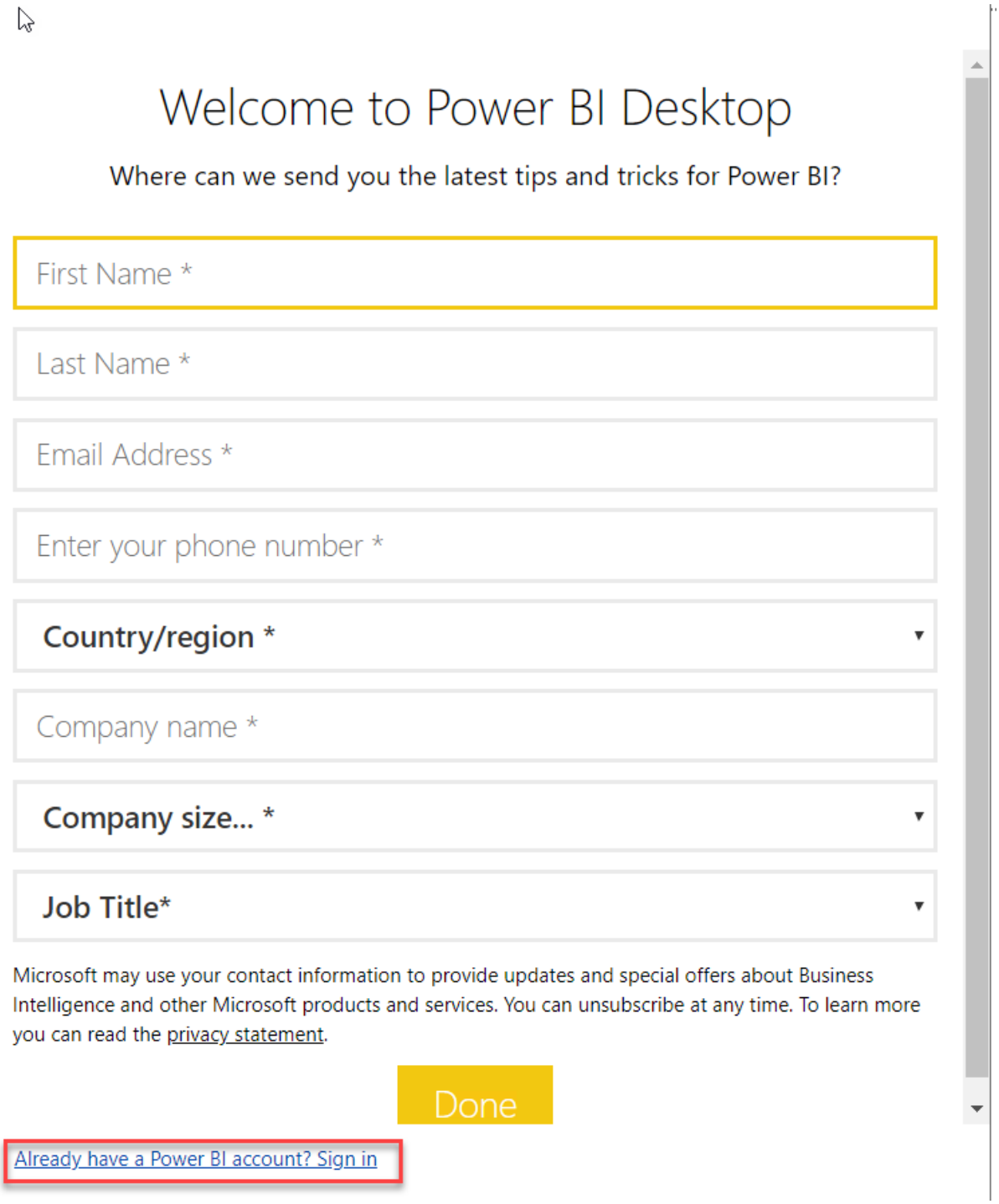
Exercise 4: Power BI

Overview

Task 1:

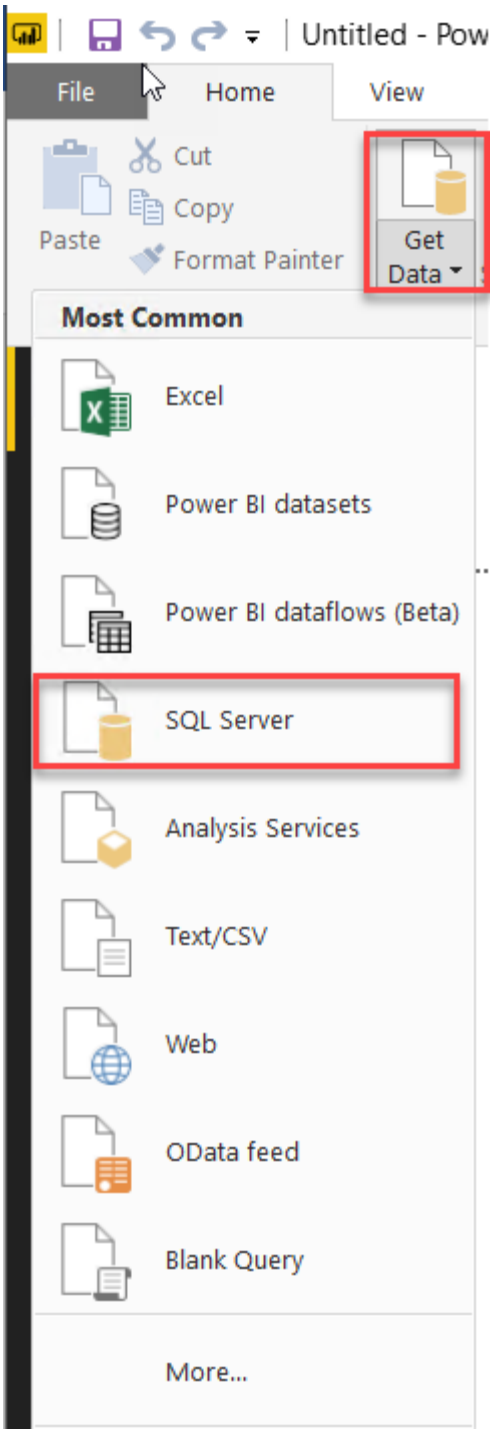
1. Connect to lab-jump-vm using RDP, You can copy the VM DNS Name using Lab details page/email or through azure portal.

2. Double click **Power BI Desktop** icon on desktop to launch **Power BI Desktop**
3. Note: If you have PowerBI desktop installed in your local machine, you can also use that optionally to complete this exercise
4. Upon opening the window, a **Welcome to Power BI Desktop** pop-up will appear. Click on Sign in at the bottom.



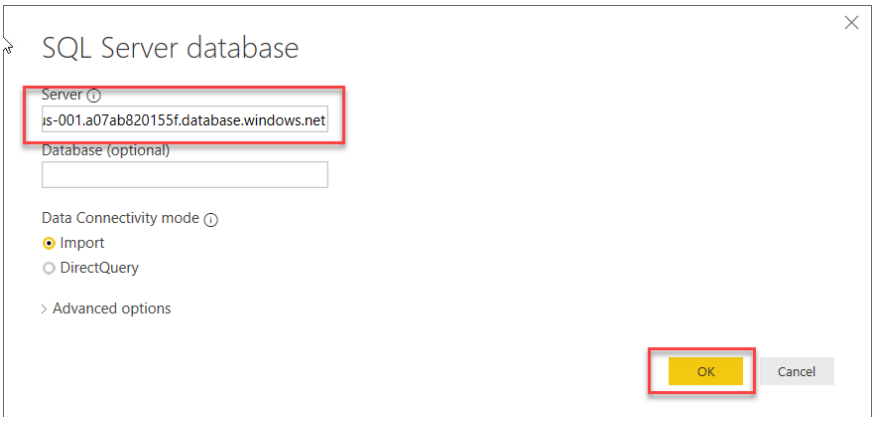
Note: If Power BI Desktop doesn't display a screen showing **Success!** within 30 seconds, you may need to click **Turn on protected mode** on the bottom of your application browser.

5. Login with the Azure Username and Password you received upon signup.
6. On the opening screen for Power BI Desktop, click **Get data**,
7. On the **Get Data** blade, click on **SQL Server database**

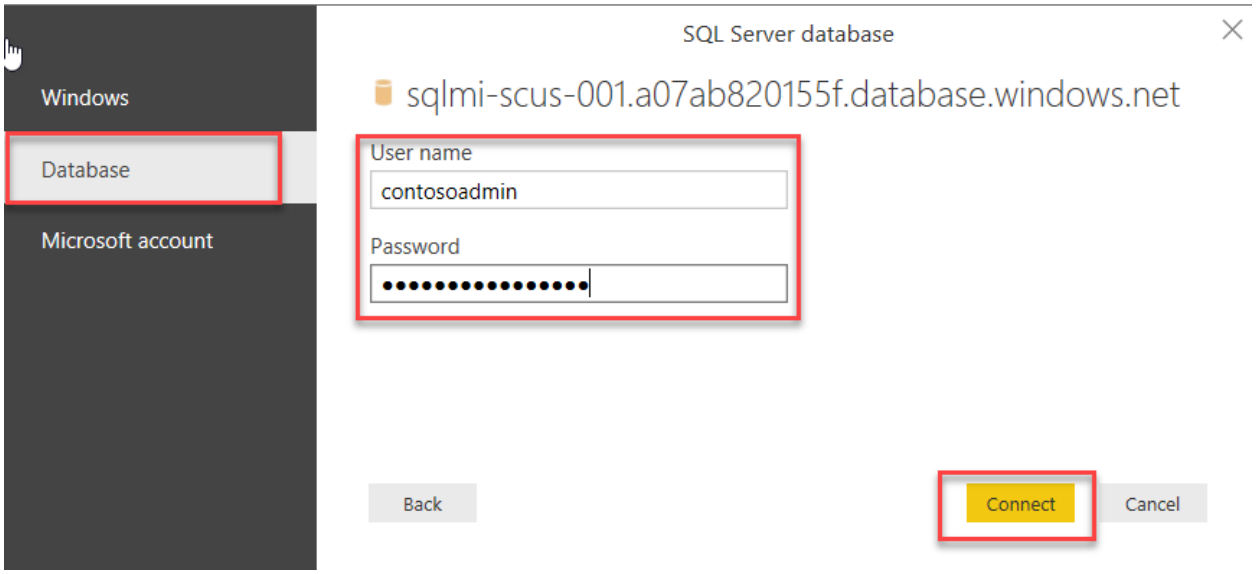


Note: This mimics the connection requirements for our datawarehouse to be able to communicate with Power BI the same way a SQL Server database would.

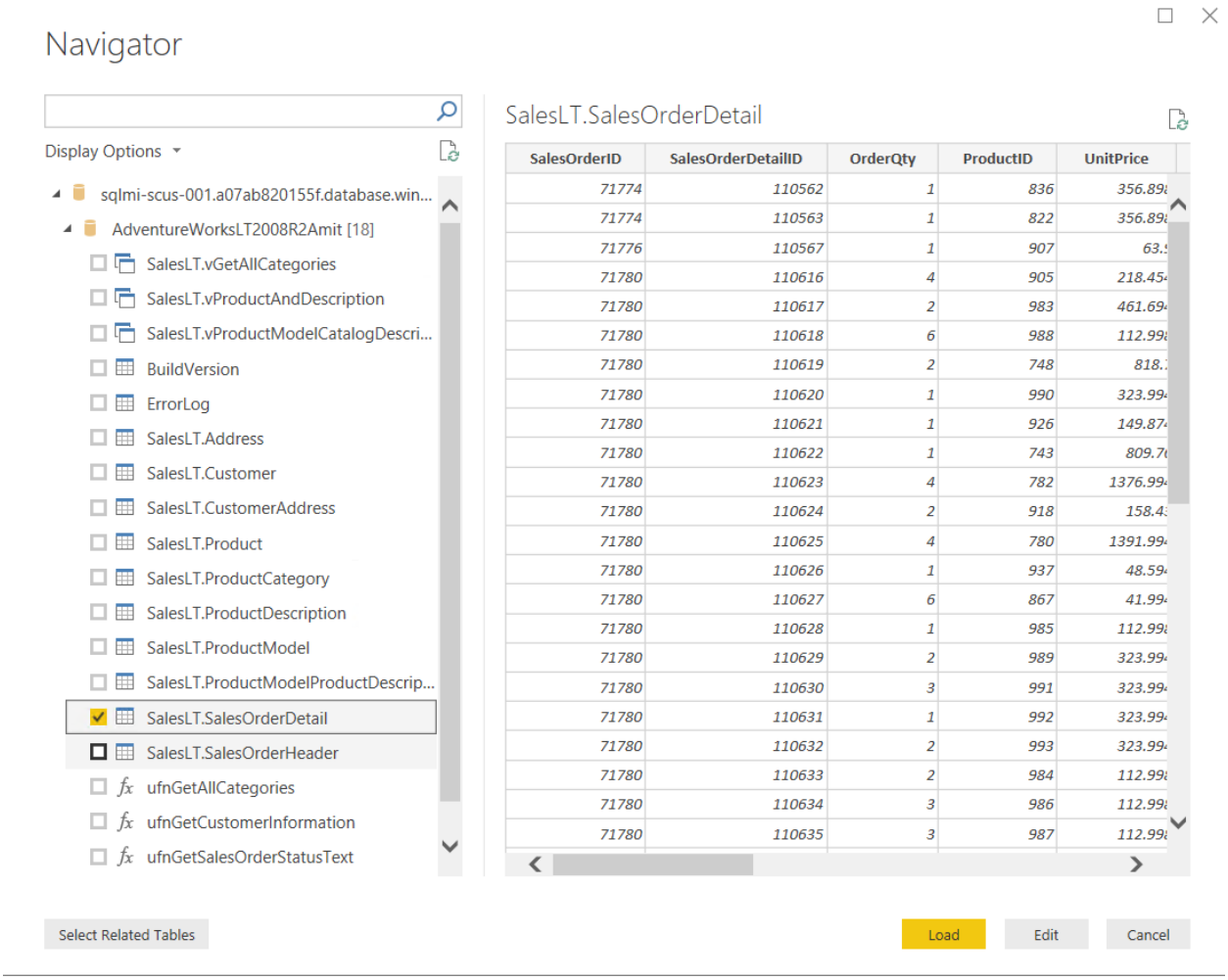
8. Click **Database**, then **SQL Server database**
9. Enter the Hostname of your SQL MI Instance under server
10. Make sure **Import** is selected under **Data Connectivity mode**
11. Click **OK**



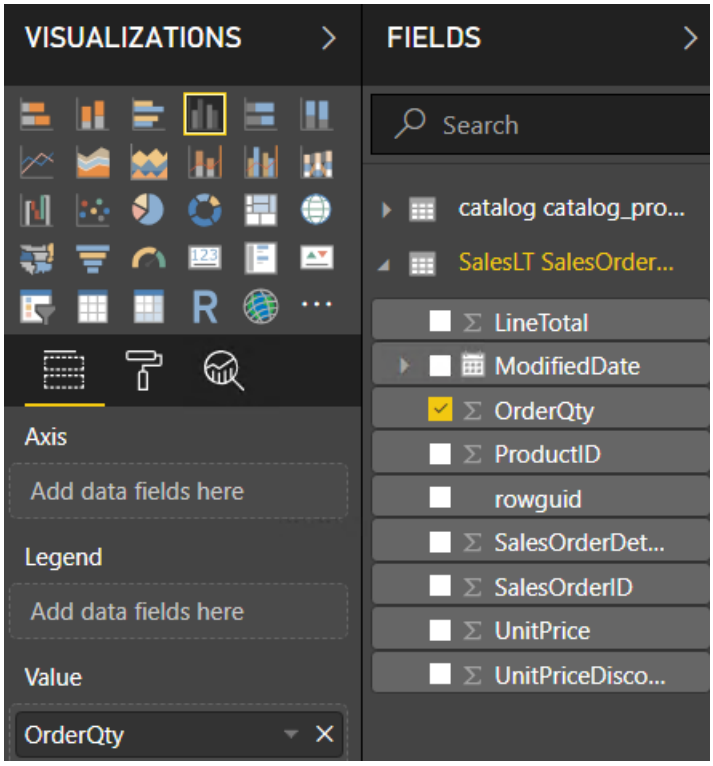
12. On the left-hand side of the window, click the **Database** tab
    1. **User name:** Enter **contosoadmin**
    2. **Password:** Enter **IAE5fAijit0w^rDM**



13. Under Databases, You can select AdventureWorksDB
14. Select the SalesOrderDetails table and move Click Load.



15. Data is now loaded, You can use PowerBI to design the reports and dashboards.



Stop the Azure-SSIS Integration Runtime

1. Login to Azure Portal with your Lab Credentails.
2. Navigate to your Azure Data Factory: **DataFactory**
3. Click the **Author & Monitor** tile to open the Data Factory user interface (UI) on a separate tab
4. On the left hand side navigation bar, click the pen icon (Author)
5. On the Factory Resources page, near the bottom, click the **Connections** button
6. When the Connections tab is displayed, click the **Integration Runtimes** tab
7. Click the Pause\Stop button in the Actions section of the **MyFirstIntegrationRuntime** to stop the service
8. When asked "Are you sure you want to Stop integration runtime", click **Stop**

Conclusion

In this lab you've learned how to create and configure Integration Runtimes in Azure Data Factory to facilitate cloud-based execution, modification, and monitoring of natively hosted SSIS packages. Then, you've learned how to create a simple yet robust data pipeline to create a reliable data copying strategy. Finally, you learned how the entire process comes together for the end-user by connecting the output data to Power BI Desktop, where you can create and visualize various reports and dashboards to gain insight into your data.