

Migrate EDW to Azure SQL Data Warehouse

Friday, September 15, 2017 5:25 PM

Migrate an existing on-premises SQL Server data warehouse to Azure SQL Data Warehouse.

Replace the existing functionality of an on-premises system: setup and configure an Azure SQL Data Warehouse, validate and migrate the existing data warehouse schema and data to Azure, configure an Azure Data Factory pipeline to move the data, and configure Power BI to access the data.

What You Will Learn

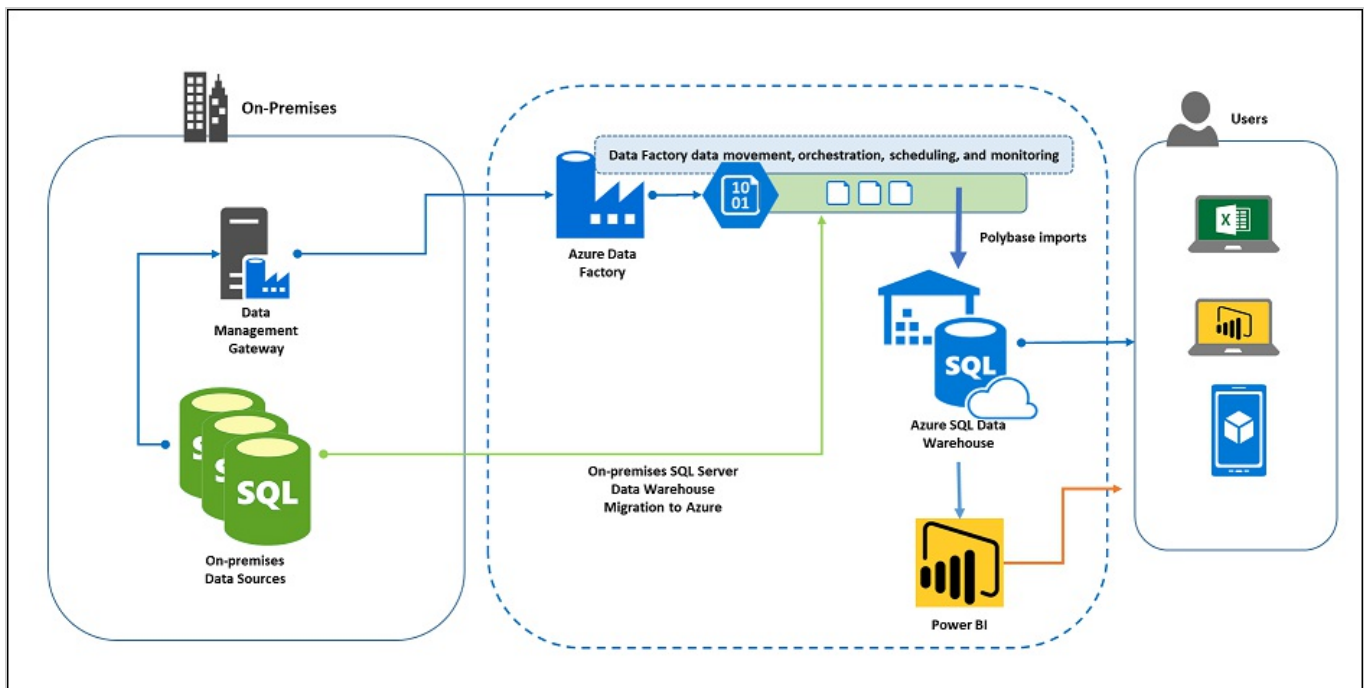
- Azure Data Platform
- Azure SQL Data Warehouse
- Azure Data Factory
- SQL Server 2016
- Power BI

Ideal Audience

- CIOs
- IT Managers
- Data Architects and DBAs
- Database Developers

Overview

Coho has asked you to migrate an existing on-premises SQL Server data warehouse to Azure SQL Data Warehouse. To build out a viable solution that can replace the existing functionality of the on-premises system you will need to setup and configure an Azure SQL Data Warehouse, validate and migrate the existing data warehouse schema and data to Azure SQL Data Warehouse, configure an Azure Data Factory pipeline to move data from an on-premises system to Azure SQL Data Warehouse and configure Power BI to access the Azure SQL Data Warehouse.



Time Estimate: 5.0 hours

Requirements

Setup Requirements

- Microsoft Azure subscription. To setup, click [next](#)
- Local machine or a virtual machine configured with **Visual Studio 2017 Community Edition**

Azure Registration

Azure

Direct your browser to <https://azure.microsoft.com/en-us/free/> and begin by clicking on the green button that reads **Start free**.

1. In the first section, complete the form in its entirety. Make sure you use your *real* email address for the important notifications.
2. In the second section, enter a *real* mobile phone number to receive a text verification number. Click send message and re-type the received code.
3. Enter a valid credit card number. **NOTE:** You will *not* be charged. This is for verification of identity only in order to comply with federal regulations. Your account statement may see a temporary hold of \$1.00 from Microsoft, but, again, this is for verification only and will "fall off" your account within 2-3 banking days.
4. Agree to Microsoft's Terms and Conditions and click **Sign Up**.

This may take a minute or two, but you should see a welcome screen informing you that your subscription is ready. The Azure subscription is good for up to \$200 of resources for 30 days. After 30 days, your subscription (and resources) will be suspended unless you convert your trial subscription to a paid one. And, should you choose to do so, you can elect to use a different credit card than the one you just entered.

Congratulations! You've now created an Azure tenant and subscription!

Environment Setup

Exercise 0: Environment setup

Overview: In this exercise you will create a lab machine if necessary and deploy the source environment for this workshop.

Prerequisites:







- Microsoft Azure subscription: [Azure Registration](#)
- Client computer with Windows 7 or later with Visual Studio 2017

Task 1: Setup a Development Environment

Create VM

If you do not have a machine setup with Visual Studio 2017 Community and Azure SDK 2.9+ you will need to complete this task.

1. Create a virtual machine in Azure using the latest release of Visual Studio Community 2017.
 - It is highly recommended to use a DS2_V2 or D2_V2 instance size for this VM

Visual Studio Community			
Results			
NAME	PUBLISHER	CATEGORY	
 Visual Studio Community 2017 on Windows 10 Enterprise N (x64)	Microsoft	Compute	
 Visual Studio Community 2017 on Windows Server 2016 (x64)	Microsoft	Compute	
 Visual Studio Community 2017 (version 15.2) on Windows Server 2016 (x64)	Microsoft	Compute	
 Visual Studio Community 2015 Update 3 with Azure SDK 2.9 on Windows Server 2012 R2	Microsoft	Compute	
 Visual Studio Community 2017 (version 15.2) on Windows 10 Enterprise N (x64)	Microsoft	Compute	
 Visual Studio Community 2015 Update 3 with Universal Windows Tools and Azure SDK 2.9 on Windows 10 Enterprise N (x64)	Microsoft	Compute	

Task 2: Disable IE Enhanced Security

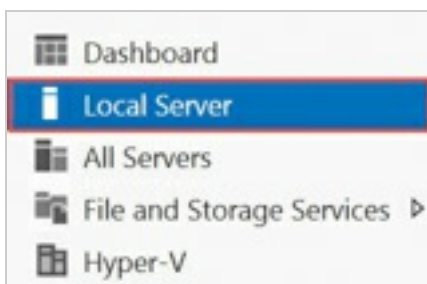
Note

Sometimes this image has IE ESC disabled, sometimes it does not.

1. On the new VM you just created click the Server Manager icon.



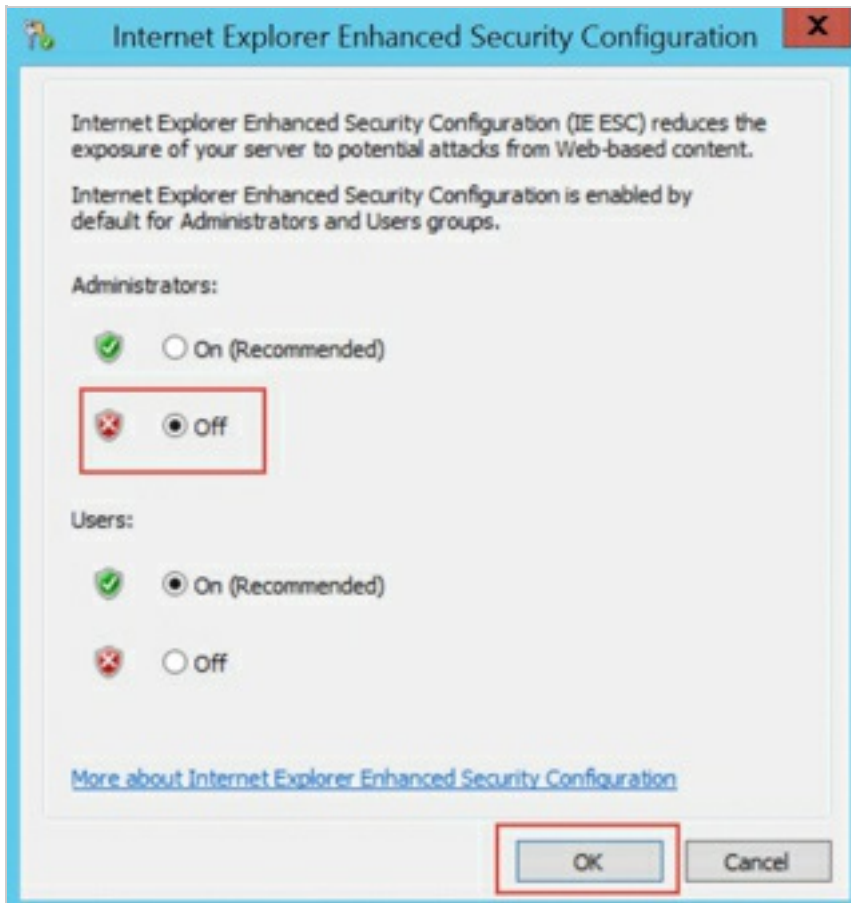
2. Click Local Server



3. On the right side of the pane, click **On** by IE Enhanced Security Configuration.

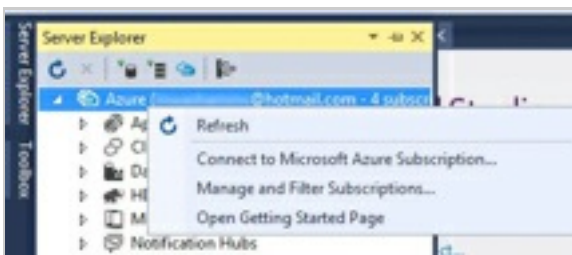


4. Change to **Off** for Administrators and click **OK**.



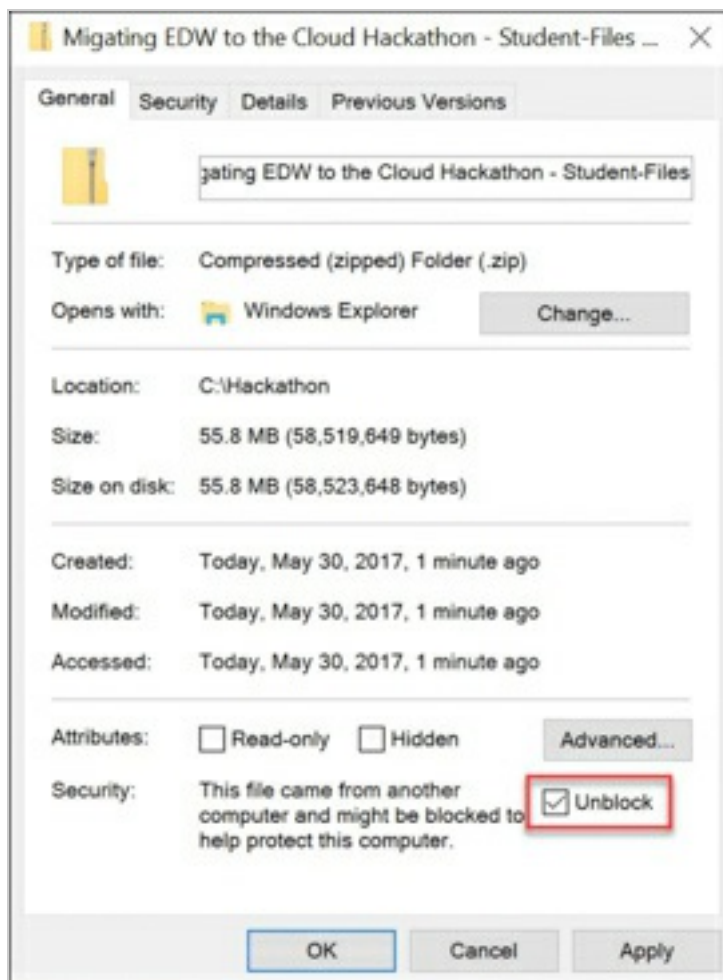
Task 3: Validate Connectivity to Azure

1. From within the virtual machine, Launch Visual Studio 2017 and validate that you can login with your Microsoft Account when prompted.
2. Validate connectivity to your Azure subscription. Launch Visual Studio, open Server Explorer from the View menu, and ensure that you can connect to your Azure subscription.

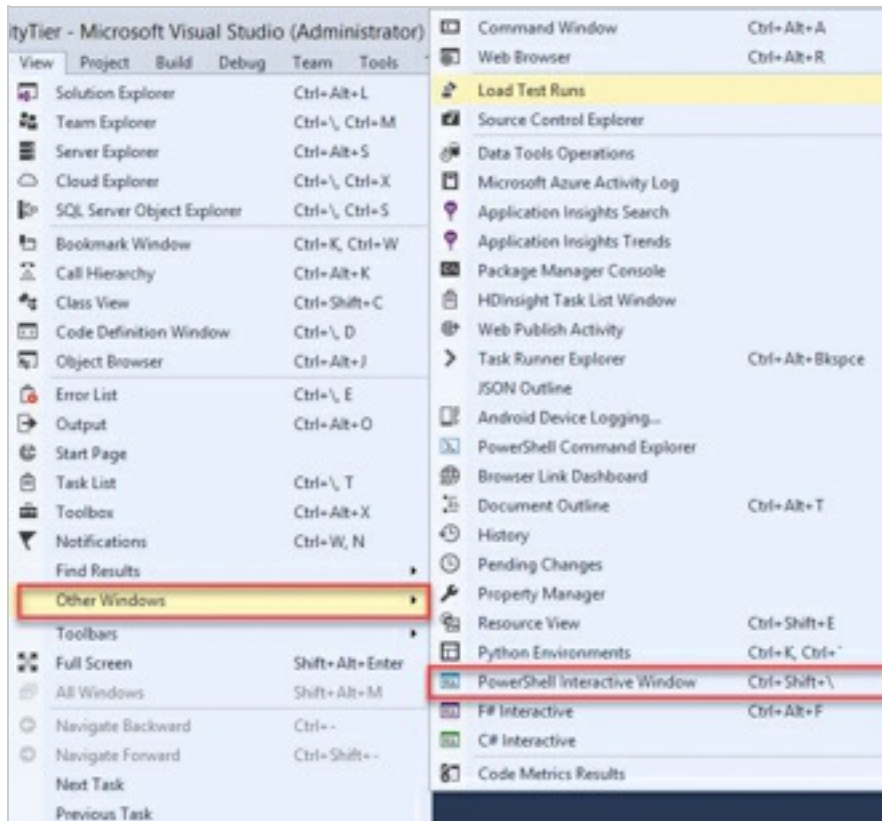


Task 4: Deploy the ARM template

1. From within your lab machine, create a new folder named **C:\workshop**
2. Download the ARM template from <https://cloudworkshop.blob.core.windows.net/migrate-edw/Migating%20EDW%20to%20the%20Cloud%20workshop%20-%20Student-Files-6-2017.zip> and extract to C:\workshop.
3. Right-click the zip file, select properties, check the **Unblock** checkbox and click **OK**.

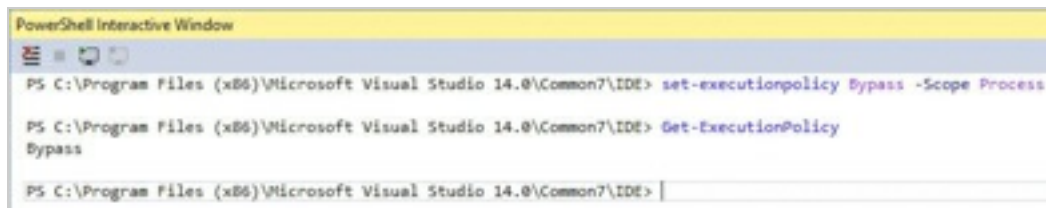


4. Open the C:\Hackathon\Migating EDW to the Cloud workshop - Student-Files\Templates\EnvironmentSetup\EnvironmentSetup.sln in Visual Studio
5. Open **PowerShell Interactive Window** within Visual Studio.



6. Run below commands to set PowerShell execution policy to bypass

```
set-executionpolicy Bypass -Scope Process
```



7. In the Solution Explorer pane, under the EnvironmentSetup solution, right-click the EnvironmentSetup project, navigate to **Deploy -> New...**

8. In the Deploy to Resource Group window, update the following:

1. Subscription: **Choose the subscription you are using for your workshop**
2. Resource group: **Create a new resource group called DWEnvironment and select a location that is near you.** Be aware that there is a limit to the number of cores per region available to you. Consider using another region if you already have resources deployed in your nearest region.
3. Deployment template: **azuredeploy.json**
4. Template parameters file: **azuredeploy.parameters.json**

5. Click the **Edit Parameters** button.

- AdminUserName: **demouser**
- AdminPassword: **demo@pass123**
- dcipDnsName: **create a unique name**
- SQLsalesIPDnsName: **create a unique name**
- SQLwarehouseIPDnsName: **create a unique name**

6. Artifact storage account: **<Automatically create a storage account>**

Deploy to Resource Group

Microsoft account
paul@opsgility.com

Subscription:
Visual Studio Enterprise (paul@opsgility.com)

Resource group:
DWEEnvironment (South Central US)

Deployment template:
azuredeploy.json

Template parameters file:
azuredeploy.parameters.json

Artifact storage account: ⓘ
stagef2857e77ee7b4305951 (South Central US)

[How do I deploy project artifacts with an Azure deployment template?](#)

Deploy Cancel

Edit Parameters

The following parameter values will be used for this deployment:

Parameter Name	Value
vmstorageType	Premium_LRS
adDomain	coho.com
DCName	cohoDC
AdminUserName	demouser
AdminPassword
DCWindowsOSVersion	2012-R2-Datacenter
_artifactsLocation	<Auto-generated>
_artifactsLocationSasToken	<Auto-generated>
SQLwarehouseName	SQLCohoDW
SQLImageSKU	SQLDEV
SQLsalesName	SQLCohoSales
SQLwarehouseDSCUpdateTagVersion	1.0
SQLsalesDSCUpdateTagVersion	1.0
dcipDnsName	opsgilitydc05
SQLsalesIPDnsName	opssales05
SQLwarehouseIPDnsName	opsdw05

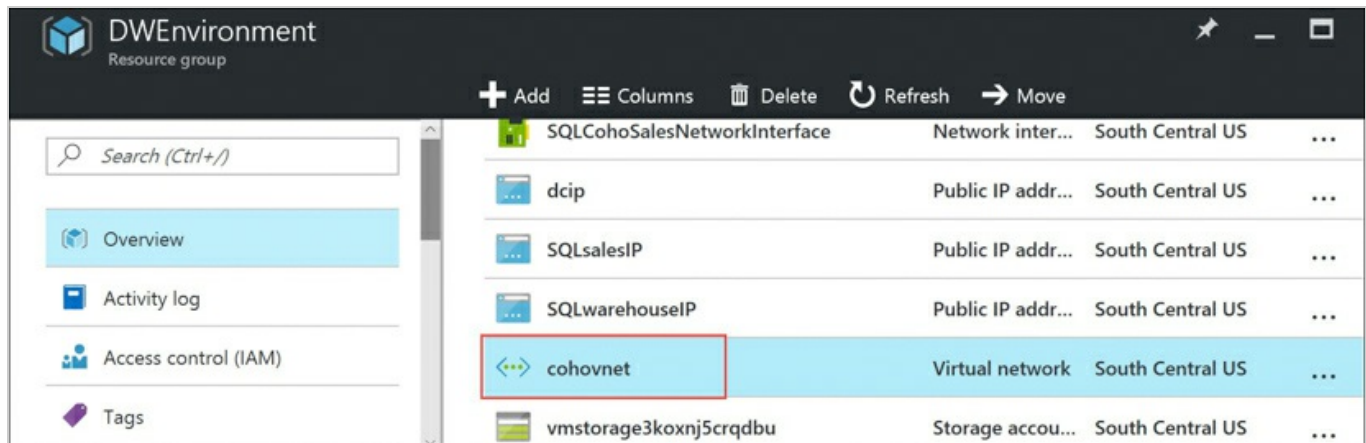
☒ Save passwords as plain text in the parameters file

Save Cancel

9. Click on **OK**.

Task 5: Configure DNS and join the domain

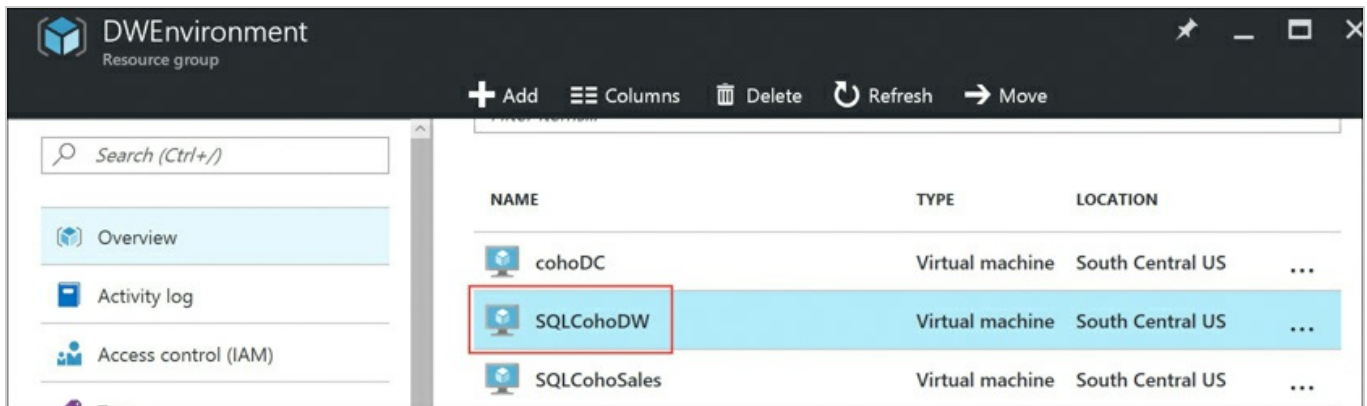
1. Browse to the Azure Portal and authenticate at <https://portal.azure.com/>
2. Navigate to the **DWEnvironment** resource group and select the **cohovnet** virtual network.



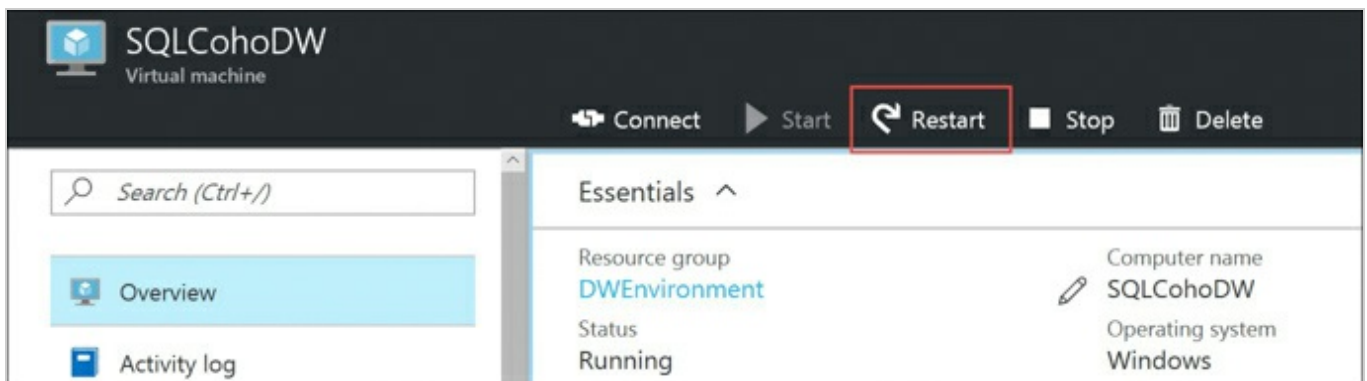
3. On the settings pane select **DNS servers**, then select **Custom DNS** on the DNS servers blade. Type **10.0.0.10** for the Primary DNS server then click the **Save** button.



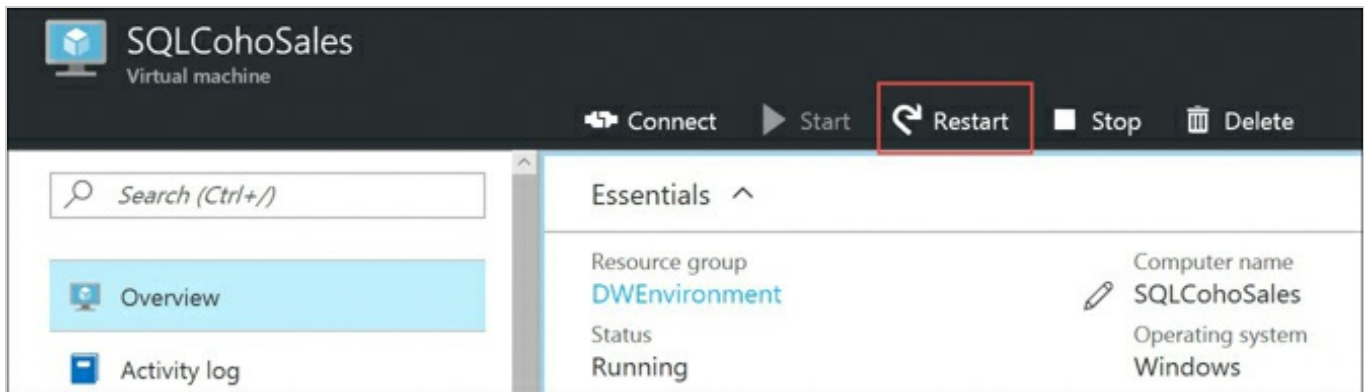
4. Wait for the configuration change to complete before continuing.
5. Navigate back to your **DWEnvironment** resource group and select the **SQLCohoDW** virtual machine.



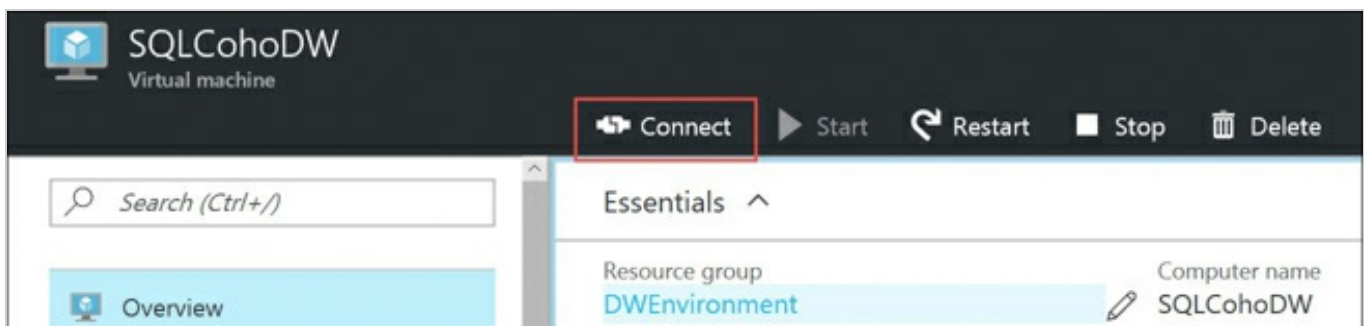
6. On the **SQLCohoDW** virtual machine blade, click the **restart** button, then click **yes** to restart the virtual machine.



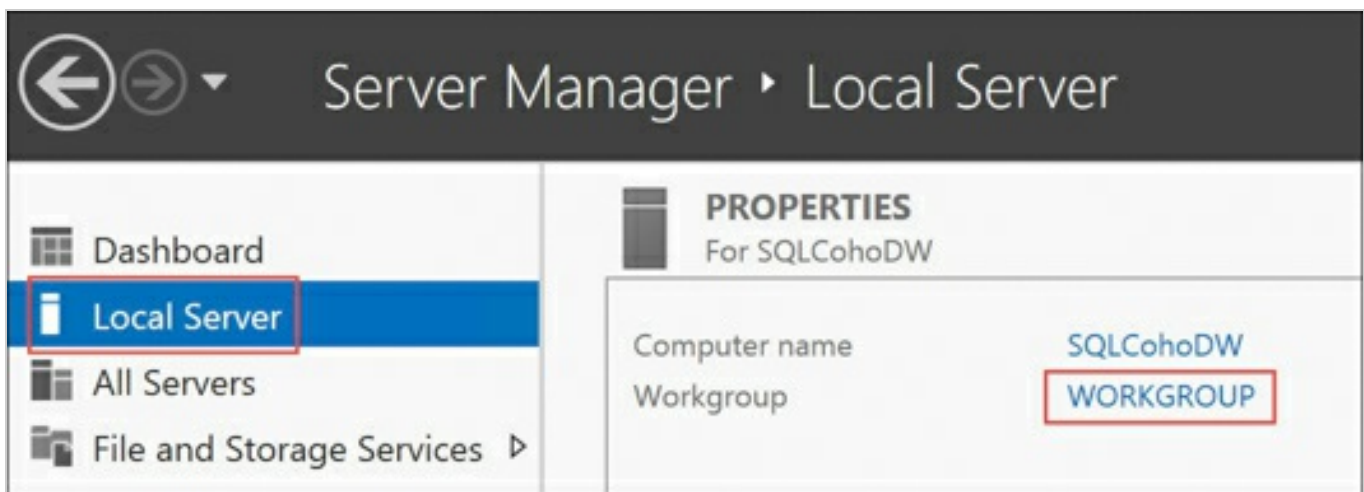
7. Repeat steps 4 and 5 to restart the **SQLCohoSales** virtual machine.



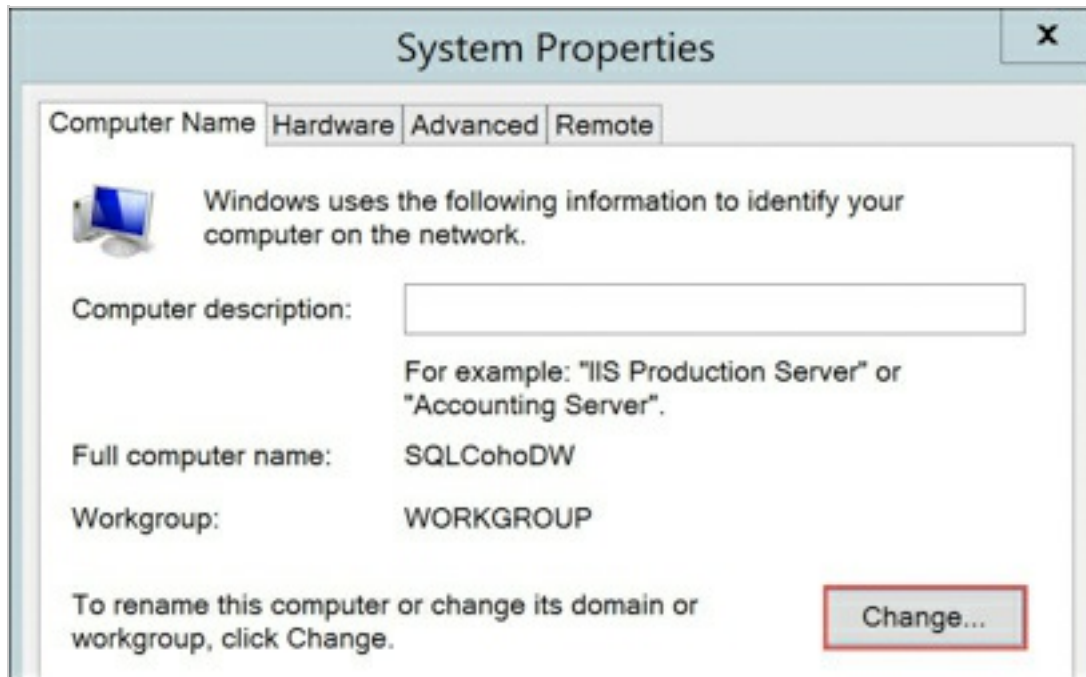
8. Navigate back to the **SQLCohoDW** virtual machine and once **SQLCohoDW** has finished rebooting, connect to it using your **demouser** admin account.



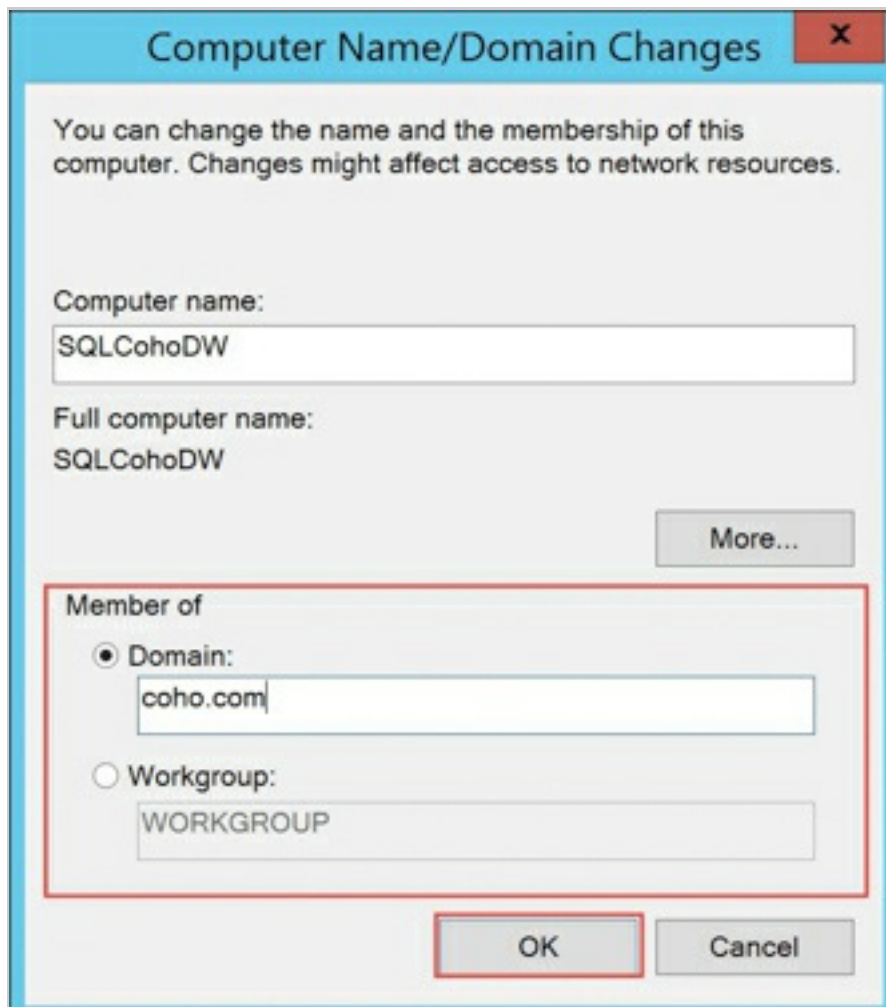
9. Once logged into **SQLCohoDW**, navigate to Server Manager, then click on **Local Server**, then click **WORKGROUP**.



10. On the Computer Name tab, click the **Change** button.



11. Make this virtual machine a member of the **coho.com** domain then click OK.



12. Use **demouser** and **demo@pass123** for the domain administrator account, then click **OK**, **Close**, **Restart Now** to reboot the computer.
13. Repeat steps 8 through 12 on the **SQLCohoSales** virtual machine to join **SQLCohoSales** to your domain.

Configure Azure Services

Exercise 1: Configure Azure services

Overview: In this exercise, you will create and configure an Azure SQL Data Warehouse, an Azure Storage Account, and a Power BI Account.

Task1: Create an Azure Storage Account

1. Browse to the Azure Portal and authenticate at <https://portal.azure.com/>
2. Click **+New** and type **Storage account** in the search box. Choose **Storage account** from the results.



3. Click **Create** on the Storage account blade. Specify the following information and click **Create**.
 - Name: **specify a unique DNS name**
 - Deployment model: **Resource manager**
 - Account kind: **General purpose**
 - Performance: **Standard**
 - Replication: **Locally-redundant storage (LRS)**
 - Storage encryption: **Disabled**
 - Secure transfer required: **Disabled**
 - Resource group: **Create new - EDWmigrationStor**
 - Location: **Location near you**

Create storage account

The cost of your storage account depends on the usage and the options [Learn more](#)

* Name ✓
 .core.windows.net

Deployment model ☒ Resource manager ☐ Classic

Account kind

Performance ☒ Standard ☐ Premium

Replication

* Storage service encryption ☒ Disabled ☐ Enabled

* Subscription

* Resource group ☒ Create new ☐ Use existing
 ✓

* Location

☐ Pin to dashboard

[Automation options](#)

4. Navigate to the new storage account and click **Blobs**.

edwstor0
Storage account

Delete

Search (Ctrl+F)

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

SETTINGS
Access keys
Configuration

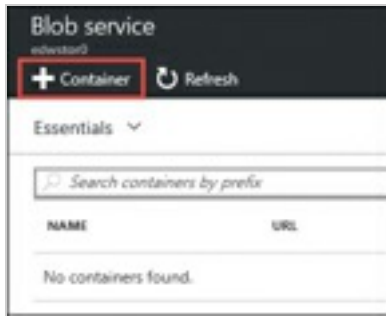
Essentials

Resource group: **edwmigrationstor**
 Status: Primary; Available
 Location: South Central US
 Subscription name: **Visual Studio Enterprise**
 Subscription ID: f2857e77-ee7b-4305-9510-f509fa2e036f

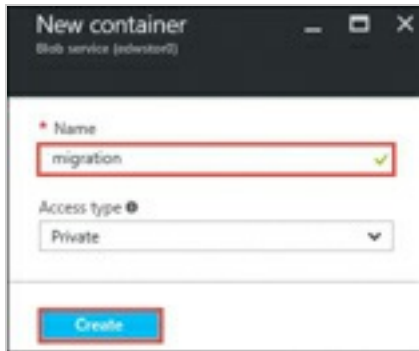
Performance: Standard
 Replication: Locally-redundant storage (LRS)

Services

5. On the Blob service blade Click the **+Container** button.



- On the New container blade type **migration** for the name and then click **Create**.



Task 2: Create an Azure SQL Data Warehouse

- Click **+New** and type **SQL Data Warehouse** in the search box. Choose **SQL Data Warehouse** from the results.



- Click **Create** on the SQL Data Warehouse blade. Specify the following information and then click the **Server** tile.
 - Name: **CohoDW**
 - Resource group: **Create new - CohoDWRG**
 - Performance: **100**

SQL Data Warehouse

SQL Data Warehouse is moving to a new pricing model starting on September 1st, 2016.

* Database name
CohoDW ✓

* Subscription
Visual Studio Enterprise ▼

* Resource group ⓘ
☒ Create new ☐ Use existing
CohoDWRG ✓

* Select source ⓘ
Blank database ▼

* Server
Configure required settings >

* Collation ⓘ
SQL_Latin1_General_CP1_CI_AS

Performance ⓘ
100

100 DWU X 0.0121 = **1.21** USD / HOUR
[Learn more about pricing. »](#)

☐ Pin to dashboard

[Create](#) [Automation options](#)

3. On the Server blade, select **Create a new server**, specify the following options and click **Select**.
- Server name: **Choose a unique name**
 - Server admin login: **demouser**
 - Password: **demo@pass123**
 - Location: **Same location as your source**
 - Allow azure services: **checked**

New server

* Server name

cohodw

✓

.database.windows.net

* Server admin login

demouser

✓

* Password

●●●●●●●●

✓

* Confirm password

●●●●●●●●

✓

* Location

South Central US

▼

☒ Allow azure services to access server ⓘ

Select

4. On the SQL Data Warehouse blade click **Create**.

Server
cohodw (South Central US)

Collation
SQL_Latin1_General_CP1_CI_AS

Performance
100

100 DWU X 0.0151 = **1.51** USD / HOUR

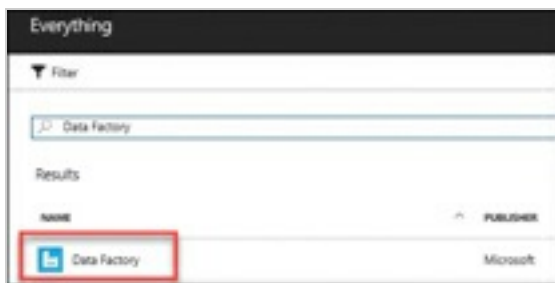
[Learn more about pricing.](#)

☐ Pin to dashboard

Create [Automation options](#)

Task 3: Create an Azure Data Factory

1. From the Azure Portal, click new and type Data Factory into the search box.
2. Choose Data Factory from the search results.



3. Click **create** on the information blade.
4. On the new data factory blade, give your data factory a unique name, create a new resource group called **CohoDF**, and choose the location nearest to you.

New data factory

Name
cohodatafactory

Subscription
Visual Studio Enterprise

Resource Group
☒ Create new ☐ Use existing
 CohoDF

Location
West Central US

Create

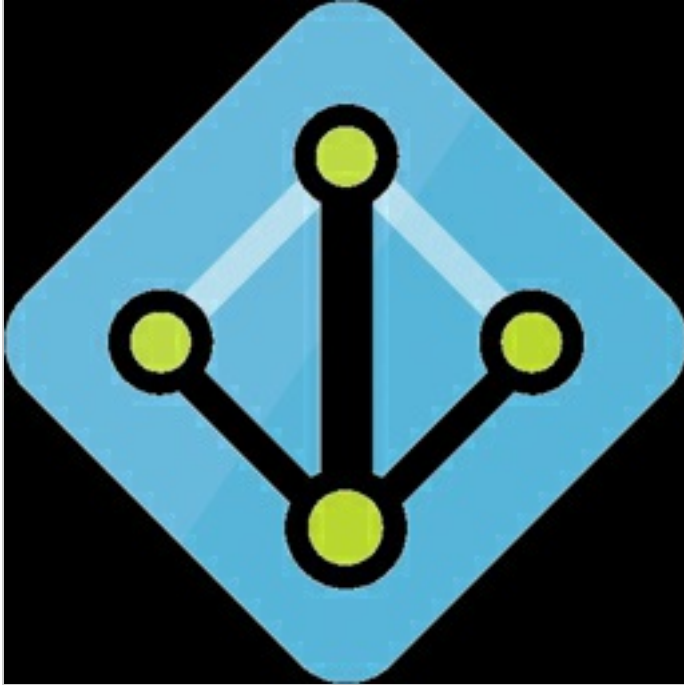
5. Click **Create** to deploy your Data Factory.

Task 4: Create a Power BI Pro account

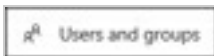
1. Note that these steps describe creating an account that can be enabled for a Power BI Pro trial from within an Azure Pass subscription or MSDN subscription. If you are using an organizational account for this hackathon some of these steps may be unnecessary or

impossible depending on your level of access. If your existing account is already enabled for Power BI Pro then you may skip this task.

2. Open a browser and navigate to the Azure Portal at <http://portal.azure.com>
3. Navigate to **Azure Active Directory**.



4. If you have multiple subscriptions, you will need to choose the correct Azure Active Directory Tenant. By default, your tenant is called **default directory**.
5. Once you have determined the correct AD tenant, click on **Users and groups**.



6. Click on the **All users** tile.

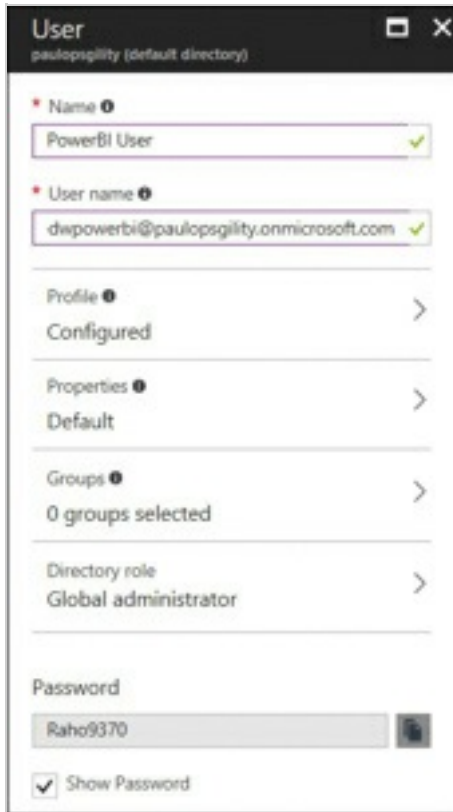


7. At the bottom of the users screen, click on the **+New user** button.



8. Fill in the configurations per the below instructions and then click **Create**.
 - o Name: PowerBI User
 - o User name: Choose a username, this will require you to specify the domain. By default this domain will be **<name of your subscription>.onmicrosoft.com**
 - o Profile: Configure the profile
 - o Properties: Leave the default
 - o Groups: Leave the default
 - o Directory Role: **Global administrator**

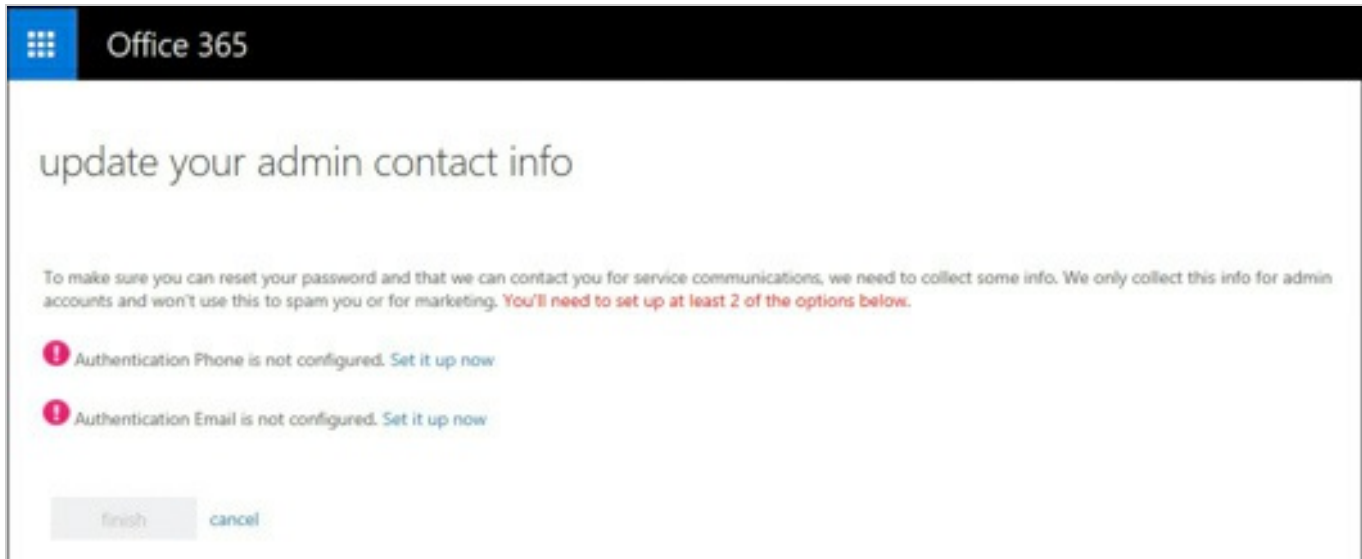
- Password: **Copy the password**, you will login and change this later.



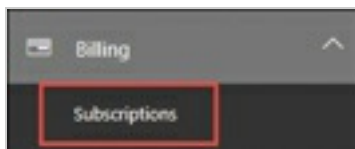
9. Open a new **InPrivate** browser and login to <http://portal.azure.com>. You will be prompted to change your password.



10. After changing your password, navigate to the Office 365 admin center at <https://portal.office.com/admin/default.aspx>
11. If required, **update your admin contact info**.



12. On the navigation pane, expand **Billing** and click **Subscriptions**.



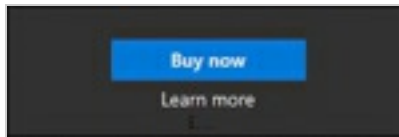
13. Click the **+ Add subscriptions** button.



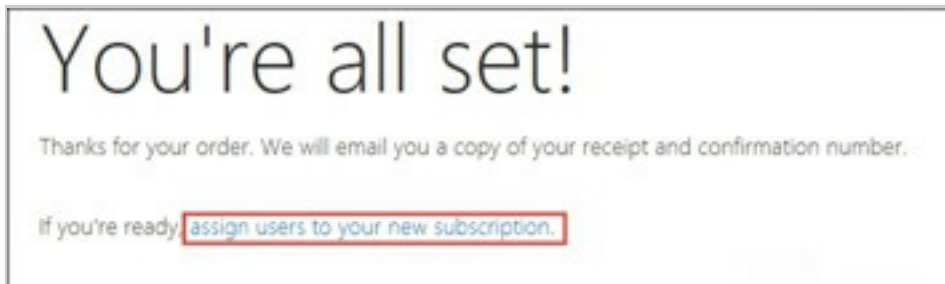
14. Scroll down to Other Plans, and find **Power BI (free)** and hover over the **ellipses** at the bottom.



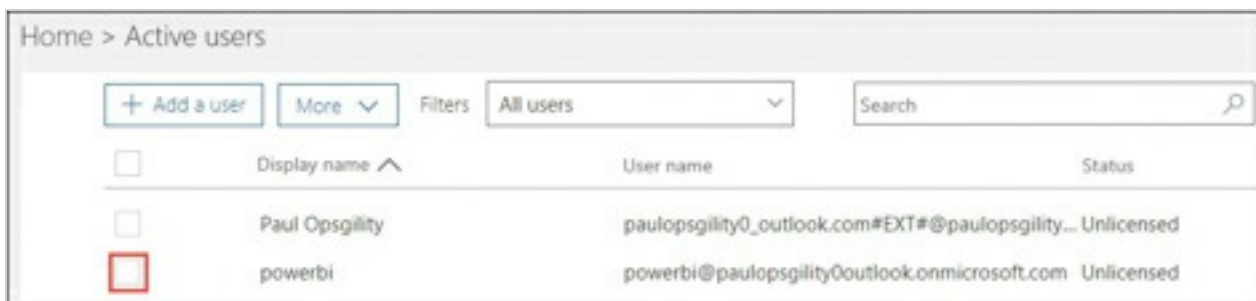
15. Click the **Buy now** button.



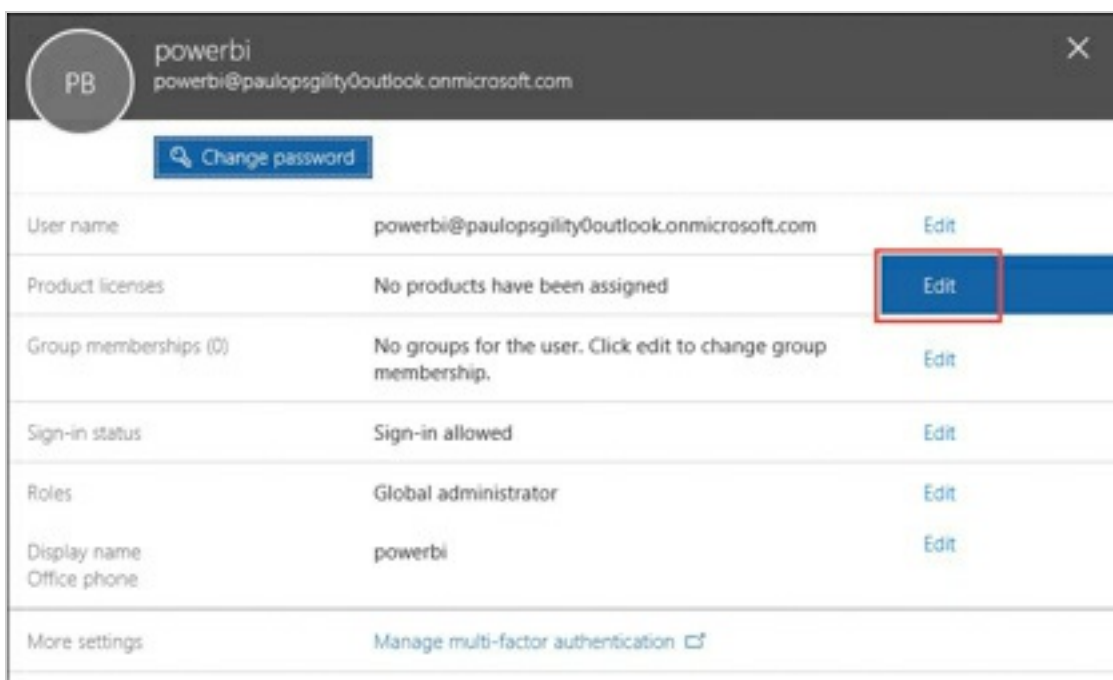
- 16. Click on **checkout**, accept the defaults and fill out the remaining billing information and **choose to be invoiced for the bill**. The total should be \$0.00.
- 17. Click on **assign users to your new subscription**.



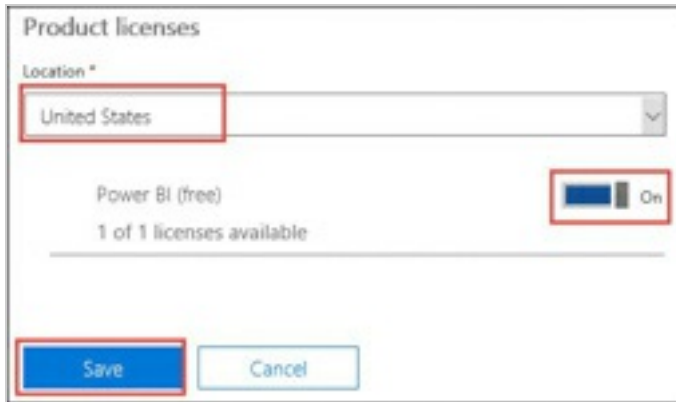
- 18. Check the box next to the account you created.



- 19. Click **Edit** Product licenses.



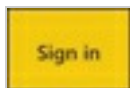
20. Set the **location** to your location, set the Power BI (free) switch to **On**, then click **Save**.



The screenshot shows the 'Product licenses' dialog box. At the top, it says 'Product licenses'. Below that is a 'Location *' dropdown menu with 'United States' selected. Underneath is the 'Power BI (free)' section, which includes a toggle switch that is currently turned 'On' and the text '1 of 1 licenses available'. At the bottom of the dialog are two buttons: 'Save' and 'Cancel'. The 'Save' button is highlighted with a red border.

21. Close all of the account editor windows.

22. Navigate to <http://powerbi.com> and click **Sign in**. If you are prompted for additional sign-in then enter your sign-in if necessary. If you are prompted to invite more people, choose **skip**. This should bring you to the Power BI Service. Later in this hackathon you will enable the trial for Power BI Pro.



Summary

In this exercise, you configured your Azure services in preparation for your warehouse environment migration. You configured an Azure Storage account to temporarily store your data before importing, you then created an Azure SQL Data Warehouse which you will use as a migration target and finally you created a Power BI account which you will use to integrate and modernize your data warehouse environment.

Data and Schema Preparation

Exercise 2: Data and schema preparation

Overview: Coho is relying on you to migrate the data warehouse to Azure SQL Data Warehouse. One of the most important steps is preparing the data and schema. During this phase you will need to verify compatibility of the schema and data and make any necessary changes required for a successful migration.

Task 1: Check compatibility with the Data Warehouse Migration Utility (Preview)

1. In the Azure portal, navigate to your **DWEnvironment** resource group, then connect to the **SQLCohoDW** virtual machine.
2. Download the [Data Warehouse Migration Utility](#) (Preview) and extract the contents.



Note

If you get an internet explorer message 'Your current security settings do not allow this file to be downloaded', Go to Internet explorer Menu -> Internet Options -> Security Tab -> Click on Internet -> Select Custom Level -> Select 'Enable' in Downloads\File Download

3. Run the Windows Installer file and follow the steps to install the Data Warehouse Migration Utility.
4. Open the Data Warehouse Migration Utility by double-clicking the icon on your desktop.



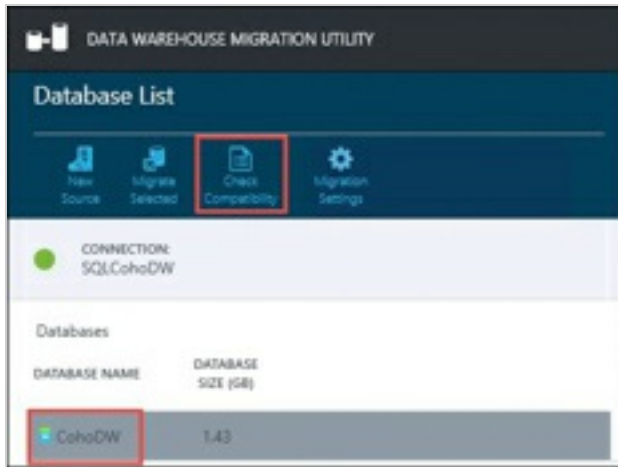
5. The Data Warehouse Migration Utility will open on the source and destination selection window. Review the source and destination. Notice that the Data Warehouse Migration Utility supports migrations from SQL Server and Azure SQL Database. Make sure that Source Type is set to **SQL Server** then click **Next**.

A screenshot of the "DATA WAREHOUSE MIGRATION UTILITY" window. The window has a dark blue background and a white title bar with a close button. At the top, there is a white icon of two cylinders connected by an arrow. Below the icon, the text "DATA WAREHOUSE MIGRATION UTILITY" is displayed. The window contains two dropdown menus: "SOURCE TYPE" with "SQL Server" selected, and "DESTINATION TYPE" with "Azure SQL Data Warehouse" selected. At the bottom, there are two buttons: "CANCEL" and "NEXT". The "NEXT" button is highlighted with a red border.

6. On the Migration Source window, change the Server Name to **SQLCohoDW**, set Authentication to Windows, and then click **Connect**.

A screenshot of the "DATA WAREHOUSE MIGRATION UTILITY" window, specifically the "MIGRATION SOURCE" tab. The window has a dark blue background and a white title bar with a close button. At the top, there is a white icon of two cylinders connected by an arrow. Below the icon, the text "DATA WAREHOUSE MIGRATION UTILITY" and "MIGRATION SOURCE" are displayed. The window contains four input fields: "Server Name" with "SQLCohoDW" entered, "Authentication" with "Windows" selected, "User Name" (empty), and "Password" (empty). At the bottom, there are two buttons: "BACK" and "CONNECT". The "CONNECT" button is highlighted with a red border.

7. Select the **CohoDW** database and then click the Check Compatibility button.



8. Choose an output location for your compatibility report and click the **Save** button. In the File Saved Successfully popup choose **No**.

NOTE

You will not be able to open the compatibility report on the server as it does not have Excel installed.

9. Open the file location that your report was saved (by default it will be in your My Documents folder) and copy and paste the file to your local machine.
10. Open the file on your local machine to review the output (your output may be different than what you see below).

CohoDW - DatabaseCompatibilityReport.xlsx - Excel				
File Home Insert Draw Page Layout Formulas Data Review View Add-Ins LOAD TEST Power Pivot Team Tell me what you want to do				
Clipboard Font Alignment Number Styles				
A1 DataBase Name				
1	DataBase Name	Impact Name	Rule Name	Instance Count
2	CohoDW	HIGH	Detect the existence of foreign key co Object	62
3	CohoDW	MEDIUM	Detect the existence of primary key co Object	30
4	CohoDW	MEDIUM	Detect the existence of unique NONCI Object	13
5	CohoDW	MEDIUM	Detect the existence of unique CLUSTI Object	28
6	CohoDW	MEDIUM	Detect the existence of identity colurn Object	22
7	CohoDW	LOW	Detect the existence of unique key co Object	7
8	CohoDW	LOW	Detect the use of user defined datatyp Object	5
9	CohoDW	LOW	Detect the existence of columns with Object	6
10	CohoDW	LOW	Detect if full-text is enabled for the di Object	1
11	CohoDW	LOW	Detect a row length greater than 806i Object	6
12	CohoDW	LOW	Detect the datatype varchar or nvarchar Object	7
13	CohoDW	LOW	Detect RETURN keyword Code	3
14	CohoDW	LOW	Detect CREATE TRIGGER keywords Code	1
15	CohoDW	LOW	Detect EVENTDATA function Code	1
16	CohoDW	LOW	Detect CURRENT_USER function Code	1
17	CohoDW	LOW	Detect SET NOCOUNT keywords Code	1
18	CohoDW	LOW	Detect PRINT keyword Code	3
19	CohoDW	LOW	Generated Rule : Check for use of SQL Code	2
20	CohoDW	LOW	Generated Rule : Check for use of SQL Code	2

NOTE

The Data Warehouse Migration Utility is currently in preview and the feature-set is incomplete and may have additional issues such as false negatives. It can be used to easily automate the migration of smaller databases but be aware that it does not compress files, move data to Azure storage or use Polybase for import.

Task 2: Validate schema and data

1. In the Azure portal, navigate to your **DWEnvironment** resource group, then connect to the **SQLCohoDW** virtual machine.
2. Launch SQL Server Management Studio and open a **New Query** window.



3. Run the following query to check for data incompatibility and potential data length issues.

```

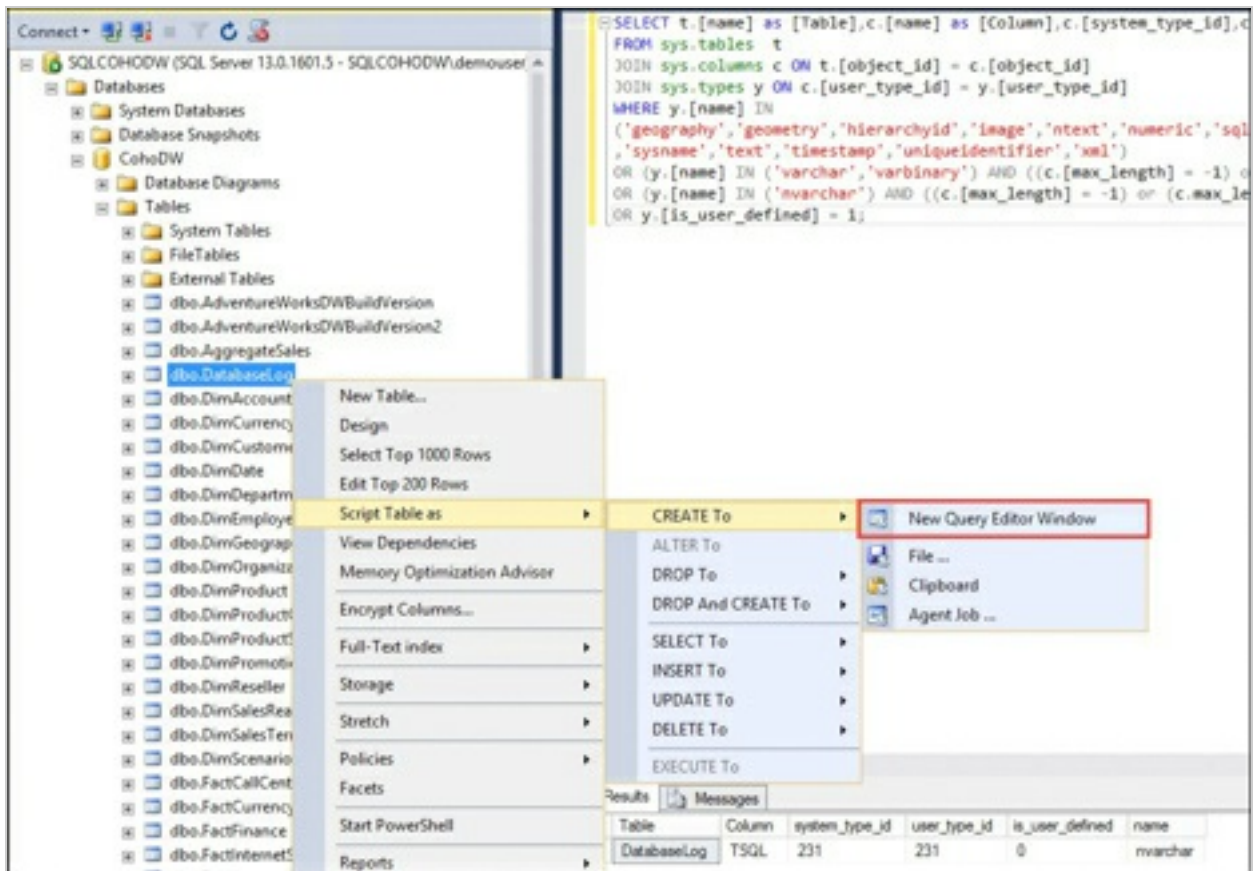
USE CohoDW
GO
SELECT t.[name] as [Table],
       c.[name] as [Column],
       c.[system_type_id],
       c.[user_type_id],
       y.[is_user_defined],
       y.[name]
FROM sys.tables t JOIN sys.columns c ON t.[object_id] = c.[object_id]
                JOIN sys.types y ON c.[user_type_id] = y.[user_type_id]
WHERE y.[name] IN ('geography', 'geometry', 'hierarchyid', 'image', 'ntext',
'numeric', 'sql_variant', 'sysname', 'text', 'timestamp', 'uniqueidentifier', 'xml')
      OR (y.[name] IN ('varchar', 'varbinary') AND ((c.[max_length] = -1) or
                                                    (c.max_length > 8000)))
      OR (y.[name] IN ('nvarchar') AND ((c.[max_length] = -1) or
                                         (c.max_length > 4000))) OR y.[is_user_defined] =
1;

```

NOTE

A full list of incompatible table features and data types can be found in the migration documentation at <https://azure.microsoft.com/en-us/documentation/articles/sql-data-warehouse-overview-migrate/>

4. The output of the query shows the table and column but not the reason for the incompatibility. To gain more insight into the reason you can script the table out by expanding the CohoDW database in Object Explorer, right-click the table, select Script Table as -> CREATE To -> New Query Editor Window.



- From the script of the table you can see that the 'TSQL' column of the 'DataLog' table has a data type nvarchar(4000) which is equivalent to 8000 bytes which means that the data may potential exceed the maximum data size.

```
CREATE TABLE [dbo].[DatabaseLog](
    [DatabaseLogID] [int] NOT NULL,
    [PostTime] [datetime] NOT NULL,
    [DatabaseUser] [nvarchar](128) NOT NULL,
    [Event] [nvarchar](128) NOT NULL,
    [Schema] [nvarchar](128) NULL,
    [Object] [nvarchar](128) NULL,
    [TSQL] [nvarchar](4000) NOT NULL
) ON [PRIMARY]
GO
```

- Before we fix this column we must validate that none of the data would be truncated. Check the maximum actual data size with the following query.

```
SELECT MAX(DATALENGTH([TSQL]))
FROM DatabaseLog
```

* The result is 3034 which means that our longest value is 3034 bytes or 1517 characters, leaving us plenty of space to modify the column with no loss of data.

7. Modify the column by executing the following query:

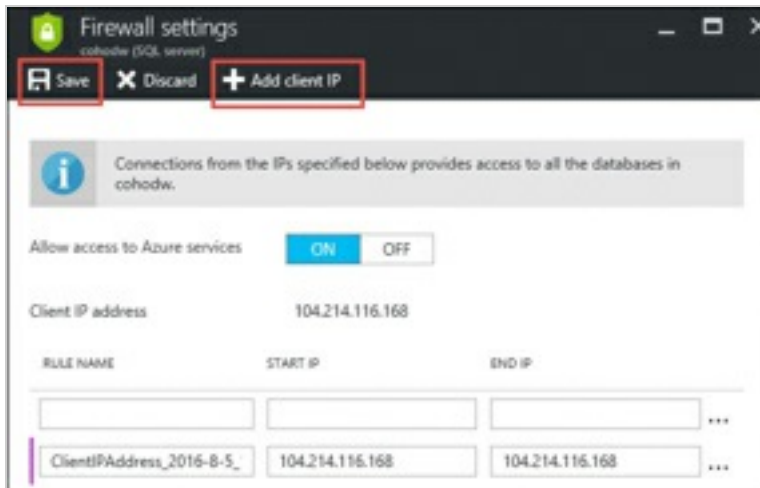
```
ALTER TABLE dbo.DatabaseLog ALTER COLUMN [TSQL] nvarchar(2000)
```

Task 3: Prepare Azure SQL Data Warehouse and migrate schema

1. In the Azure portal, navigate to your **DWEnvironment** group, then connect to the **SQLCohoDW** virtual machine.
2. Open Internet Explorer and connect to the Azure Portal.
3. Navigate to your CohoDWRG resource group, then click on the cohodw logical SQL Server that hosts your Azure SQL Data Warehouse.

NAME	TYPE	LOCATION	
 cohodw	SQL server	South Central US	...
 CohoDW	SQL data ware...	South Central US	...

4. In the settings blade, click on **Firewall**.
5. In the cohodw - Firewall blade, click the **+Add client IP** button, then click the **Save** button.



Firewall settings
cohodw (SQL server)

Save Discard + Add client IP

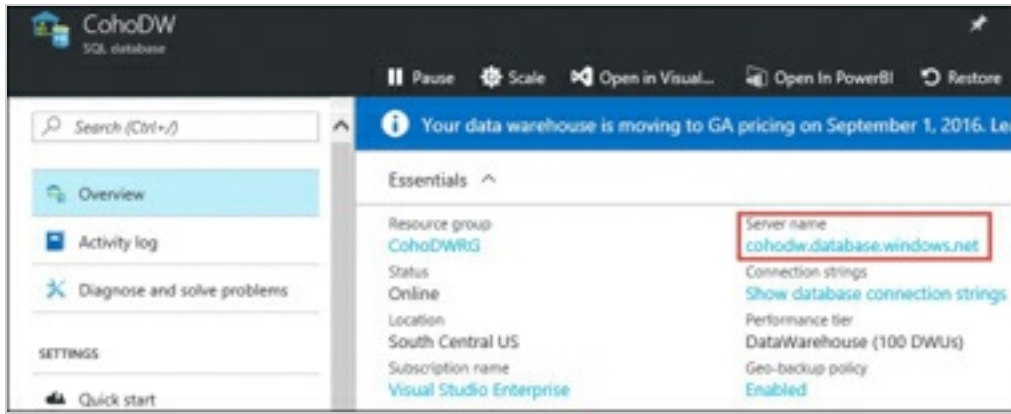
Connections from the IPs specified below provides access to all the databases in cohodw.

Allow access to Azure services ☒ ON ☐ OFF

Client IP address 104.214.116.168

RULE NAME	START IP	END IP	
			...
ClientIPAddress_2016-8-5_	104.214.116.168	104.214.116.168	...

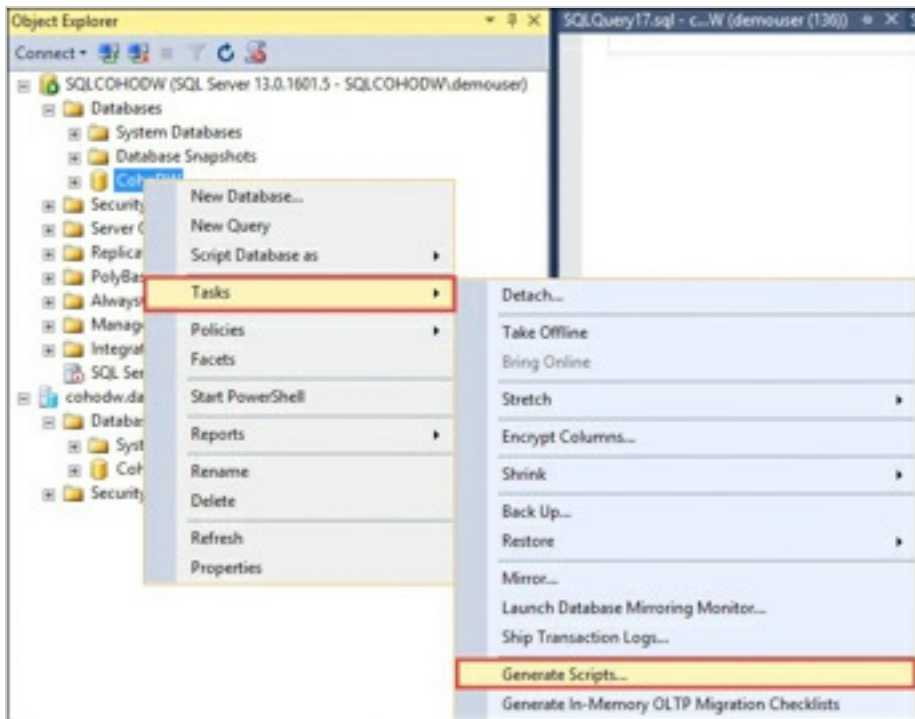
6. Back in the **CohoDWRG** resource group, select the **CohoDW** data warehouse and copy the server name.



7. Open SQL Server Management studio, click the connect button in Object Explorer and connect to your SQL Data Warehouse using the **demouser** account and password to verify connectivity.

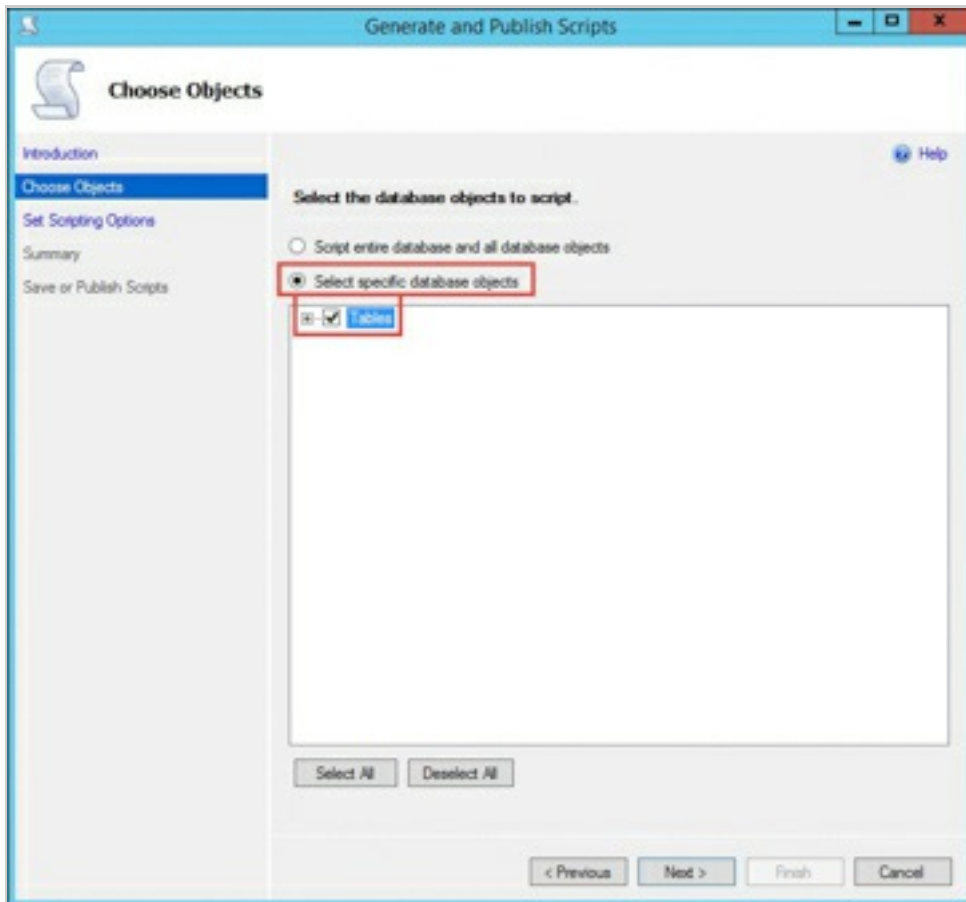


8. On your Azure SQL Data Warehouse, expand databases, select the CohoDW database then click the **New Query** button.
9. Connect to the local SQLCohoDW instance of SQL Server, right click your local copy of CohoDW, and select **Tasks -> Generate Scripts** to launch the Generate and Publish Scripts wizard.

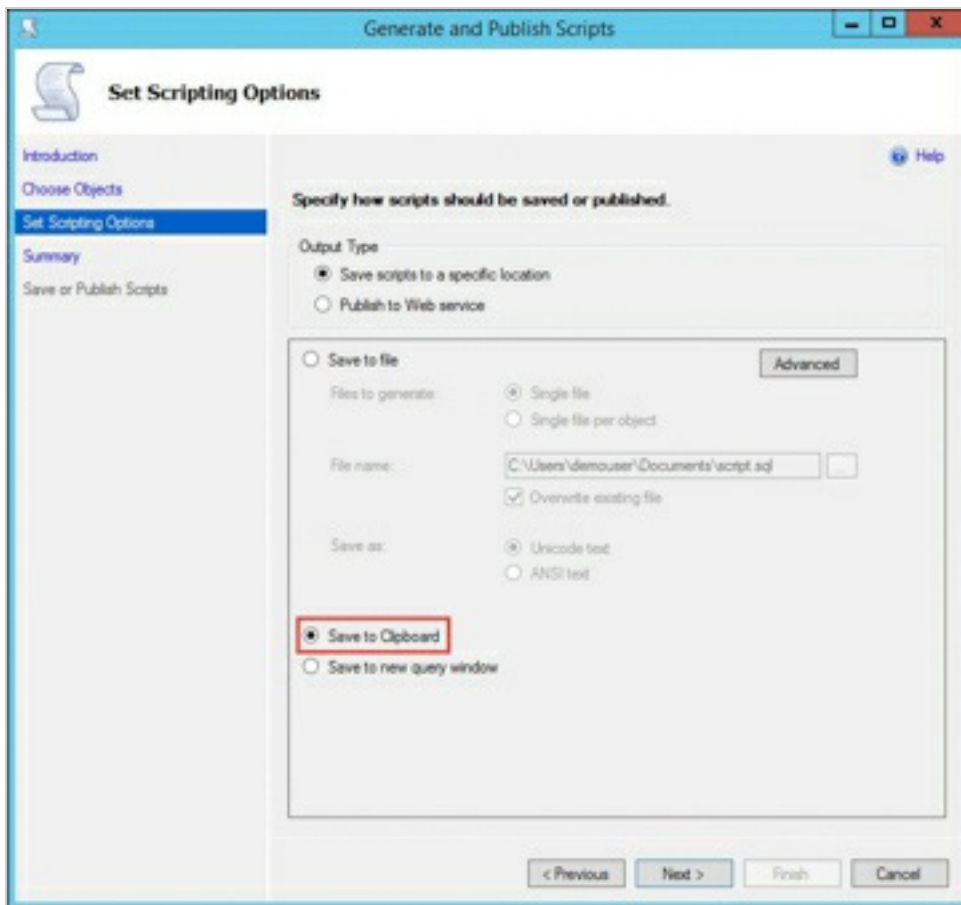


10. Click Next on the Introduction screen.

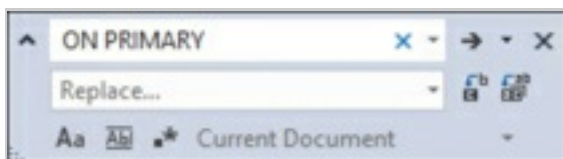
11. On the Choose Objects screen, **Select the Select specific database objects** radio button and check **Tables**, then click Next.



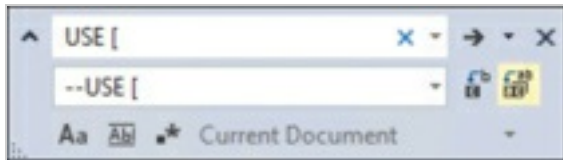
12. On the Set Scripting Options screen. Select the **Save to Clipboard** radio button and click Next.



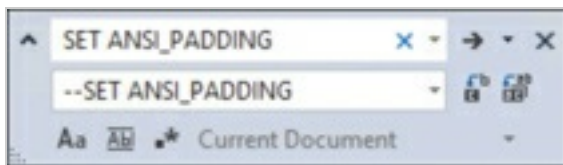
13. Accept the defaults for the remaining screens and click **Finish**.
14. Paste the results into the Query windows connected to your Azure SQL Data Warehouse.
15. This script still needs to be modified before it will run correctly in Azure SQL Data Warehouse because some T-SQL syntax is not supported in Azure SQL Data Warehouse. Make the following updates to the script:
 - Execute a Find and Replace on your script to replace all occurrences of "ON [PRIMARY]" with "" to remove them from the script.



* Execute a Find and Replace on your script to replace all occurrences of "USE [" with "--USE [" to comment out those lines.



* Execute a Find and Replace on your script to replace all occurrences of "SET ANSI_PADDING" with "--SET ANSI_PADDING" to comment out those lines.



16. Run the script by clicking the Execute button. This will use the default options to create tables, Clustered Columnstore Indexing and ROUNDROBIN distribution.



17. Execute the following query to verify that your tables were created. There should be 33 rows returned.

```
SELECT * FROM sys.tables
```

Summary

In this exercise, you prepared and verified your schema with a combination of the Data Warehouse Migration Utility and T-SQL scripts to analyze the schema and data. You then scripted your database, made the necessary changes and applied your script to Azure SQL Data Warehouse.

Migrate to ADW

Exercise 3: Migrate the data to Azure SQL Data Warehouse

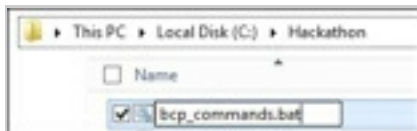
Overview: This exercise is focused on migrating the data from your existing data warehouse into SQL Data Warehouse. We will be pulling the data and then uploading it to an Azure storage account. We will then import the data via Polybase.

Task 1: Exporting data from your current data warehouse.

1. Connect to your **SQLCohoDW** virtual machine.
2. Open the **C:\Hackathon\bcp_commands.txt** file. These are the bcp commands for each of the tables you need to migrate. The line below is an example. Notice the bcp commands all use the -C 65001 parameter. This indicates that the output will be in UTF-8 which is required by Polybase. This code page is only an option with bcp.exe that ships with SQL Server 2016 tools. If you are using an older version of bcp you will have an additional step to convert to UTF-8.

```
bcp "select [ScenarioKey],REPLACE([ScenarioName],',','|') from [CohoDW].[dbo].[DimScenario]" queryout "C:\Migration\dbo.DimScenario.txt" -q -c -C 65001 -t "|" -r "\n" -S SQLCohoDW -T
```

3. Close the file after you are done reviewing it. Change the file name to **bcp_commands.bat**.

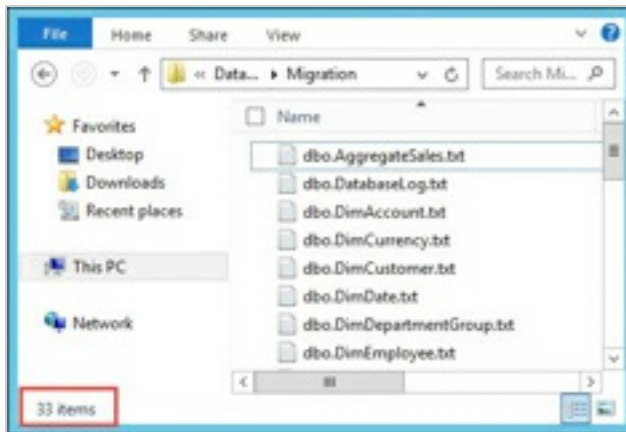


4. Open a command prompt and execute **C:\Hackathon\bcp_commands.bat**

Note

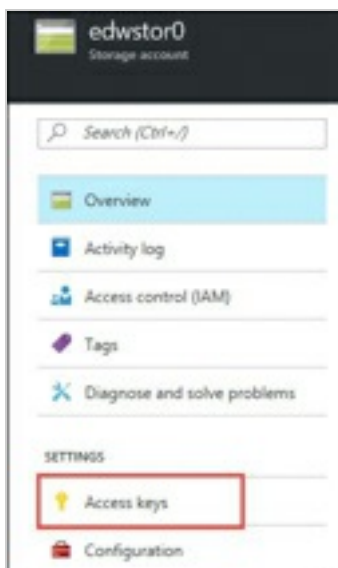
In a production environment you would likely make some effort to parallelize the execution of the various bcp commands. For larger tables, you also might parallelize the export from a single table.

5. Navigate to the **C:\Migration** folder. If the commands completed successfully you will have **33 files**.

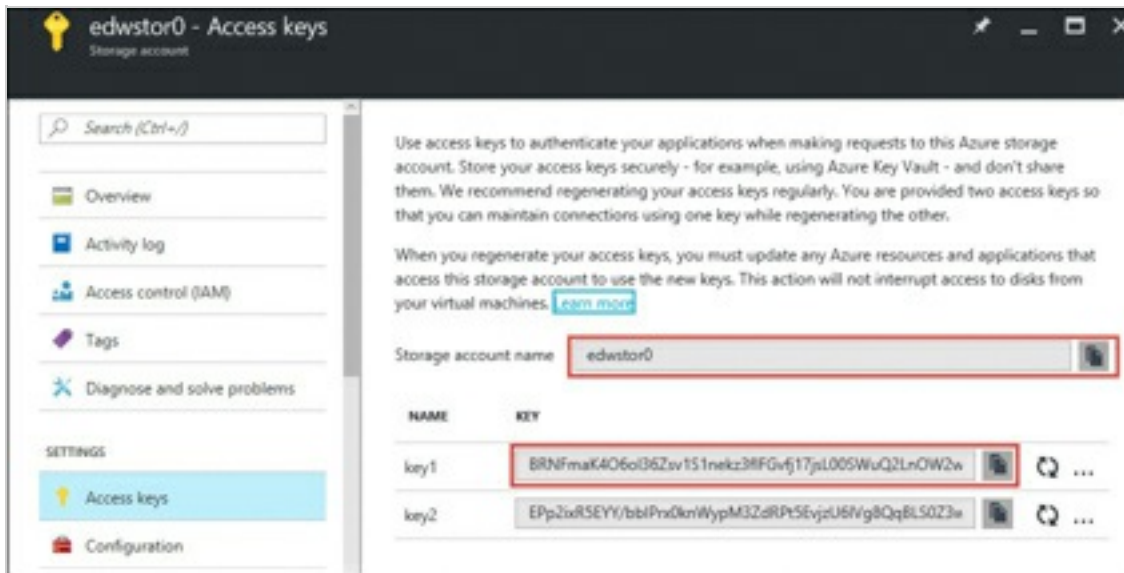


Task 2: Transfer your data to Azure

1. From your SQLCohoDW virtual machine navigate to you C:\Hackathon folder and double-click **MicrosoftAzureStorageTools.msi** to install AzCopy.
2. In the Azure Portal navigate to your **EDWmigrationStor** resource group and click on your storage account.
3. In the Storage account blade, under settings, click on **Access keys**



4. Copy the storage account name and access key1 and paste into notepad for later use.



5. Open a command prompt and navigate to the **C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy** folder.
6. Update the following command with your storage account name and key then execute it to begin copying your data files to Azure (all of the text is a single command)

```
AzCopy /Source:"C:\Migration" /Dest:https://<YourStorageAccount>.blob.core.windows.net/migration /DestKey:<YourStorageAccountKey> /pattern:*.txt /NC:2
```

7. Confirm that all 33 files were transferred successfully.

```
C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>AzCopy /
1xPFJxoh8mc57PrFlguFMRRcosW28yA== /pattern:*.txt /NC:2
Finished 33 of total 33 file(s).
[2016/09/16 21:47:47] Transfer summary:
Total files transferred: 33
Transfer successfully: 33
Transfer skipped: 0
Transfer failed: 0
Elapsed time: 00:00:00:03
C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>
```

* Confirm that the files are in the correct storage container by navigating to your storage account, clicking on blobs, then clicking on your container.

migration			
Container			
Refresh Delete container Properties Access policy			
Location: migration			
Search blobs by prefix (case-sensitive)			
NAME	MODIFIED	BLOB TYPE	SIZE
dbo.AggregateSales.txt	9/16/2016 4:47:43 PM	Block blob	2.24 MB
dbo.DatabaseLog.txt	9/16/2016 4:47:43 PM	Block blob	38.8 KB
dbo.DimAccount.txt	9/16/2016 4:47:43 PM	Block blob	5.64 KB
dbo.DimCurrency.txt	9/16/2016 4:47:43 PM	Block blob	2.21 KB
dbo.DimCustomer.txt	9/16/2016 4:47:43 PM	Block blob	4.48 MB
dbo.DimDate.txt	9/16/2016 4:47:43 PM	Block blob	116.27 KB
dbo.DimDepartmentGroup.txt	9/16/2016 4:47:43 PM	Block blob	179 B

8. Open SQL Server Management Studio and connect to your SQL Data Warehouse.
9. Open a new query window and execute the following command to create a master key for your database.

```
CREATE MASTER KEY
```

10. Execute the following to create a database scoped credential that you will use to store the access key to the migration storage account.

```
CREATE DATABASE SCOPED CREDENTIAL MigrationCredential
WITH IDENTITY = '<YourStorageAccountName>' , SECRET = '<YourStorageAccountKey>'
```

11. Create an external file format by executing the following query. The external file format defines the external storage and its layout.

```
CREATE EXTERNAL DATA SOURCE MigrationStor WITH (TYPE = HADOOP ,
LOCATION=
'wasbs://<YourStorageContainerName>@<YourStorageAccountName>.blob.core.windows.
net',
CREDENTIAL = MigrationCredential);
```

12. Create an external file format by executing the following query. The external file format defines the external storage and its layout.

```
CREATE EXTERNAL FILE FORMAT MigrationFiles WITH(FORMAT_TYPE = DelimitedText,  
FORMAT_OPTIONS (FIELD_TERMINATOR = '|'));
```

13. Open the C:\Hackathon\CreateExternalTables.sql file in SQL Server Management Studio.
14. This file contains all of the external table definitions for our tables and directly leverages the external data source and external file format we created above. Click Execute to create the external tables.



15. Run the following code to verify that 33 tables were created.

```
SELECT * FROM SYS.TABLES WHERE is_external = 1
```

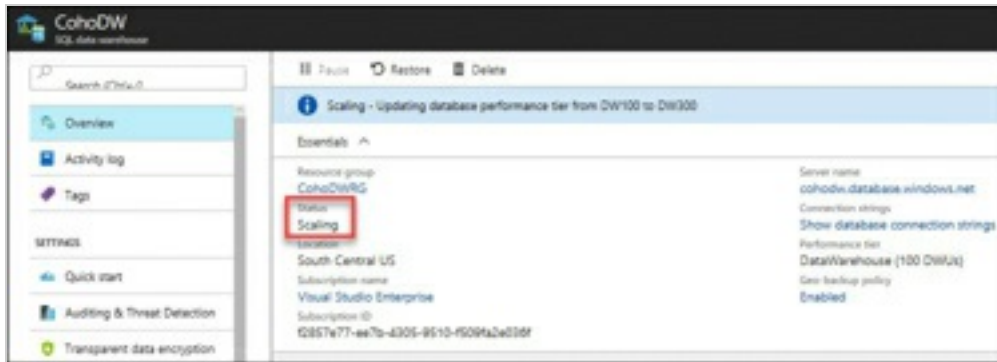
16. We are about to load the warehouse. We would like to maximize performance of the load. To do this we will adjust the performance of the Azure SQL Data Warehouse by adjusting the scale slider in the Azure Portal. Navigate to your CohoDW SQL Data Warehouse and click on the Scale button.



17. In the Scale blade, adjust the Performance slider to 300 and click the Save button.



18. Wait until the scaling process is complete to move on to the next step. You can view the SQL Data Warehouse status in the portal. The status will change from "Scaling" to "Online" and the Performance tier will reflect 300 DWUs when scaling is complete.



19. From your **SQLCohoDW** virtual machine, open the C:\Hackathon\LoadData.sql file in SQL Server Management Studio.
20. The commands in this file insert data extracted directly from the data files stored in Azure Storage via the external tables we defined in the previous steps. Click execute to begin the data load.



21. After your data is uploaded you can select data from any of the tables to verify success. In production environments, you would go through a much more thorough data validation process.

IMPORTANT!

****Scale your SQL Data Warehouse back down to 100 to prevent excessive cost.****

Integrare with ADF

Exercise 4: Integrate Azure SQL Data Warehouse and On-Premises Data Sources with Azure Data Factory and Power BI

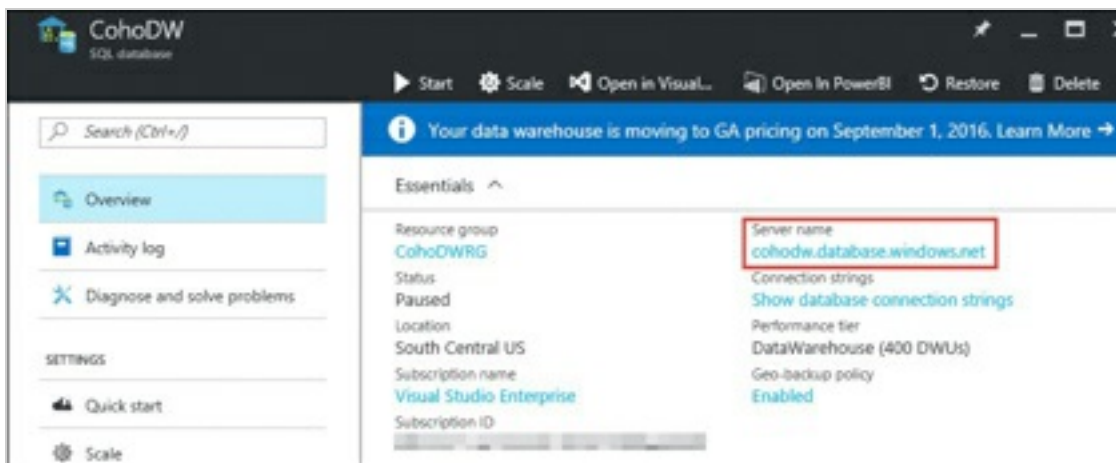
Overview: In this portion of the exercise, you will setup integration with Azure Data Factory and Power BI.

Note

To connect Power BI dashboards and reports directly to Azure SQL Data Warehouse requires a Power BI Pro subscription. Please see the instructions earlier in this hackathon to configure an account for Power BI and enable a Power BI Pro trial if you do not already have one.

Task 1: Connect Power BI to Azure SQL Data Warehouse

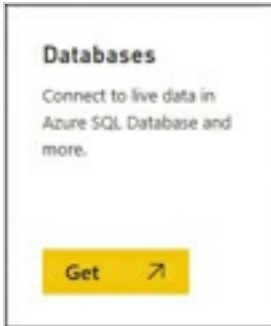
1. Connect to the Azure Portal and to your Azure SQL Data Warehouse.
2. Make note of your server name to use in your data source configuration later in this task.



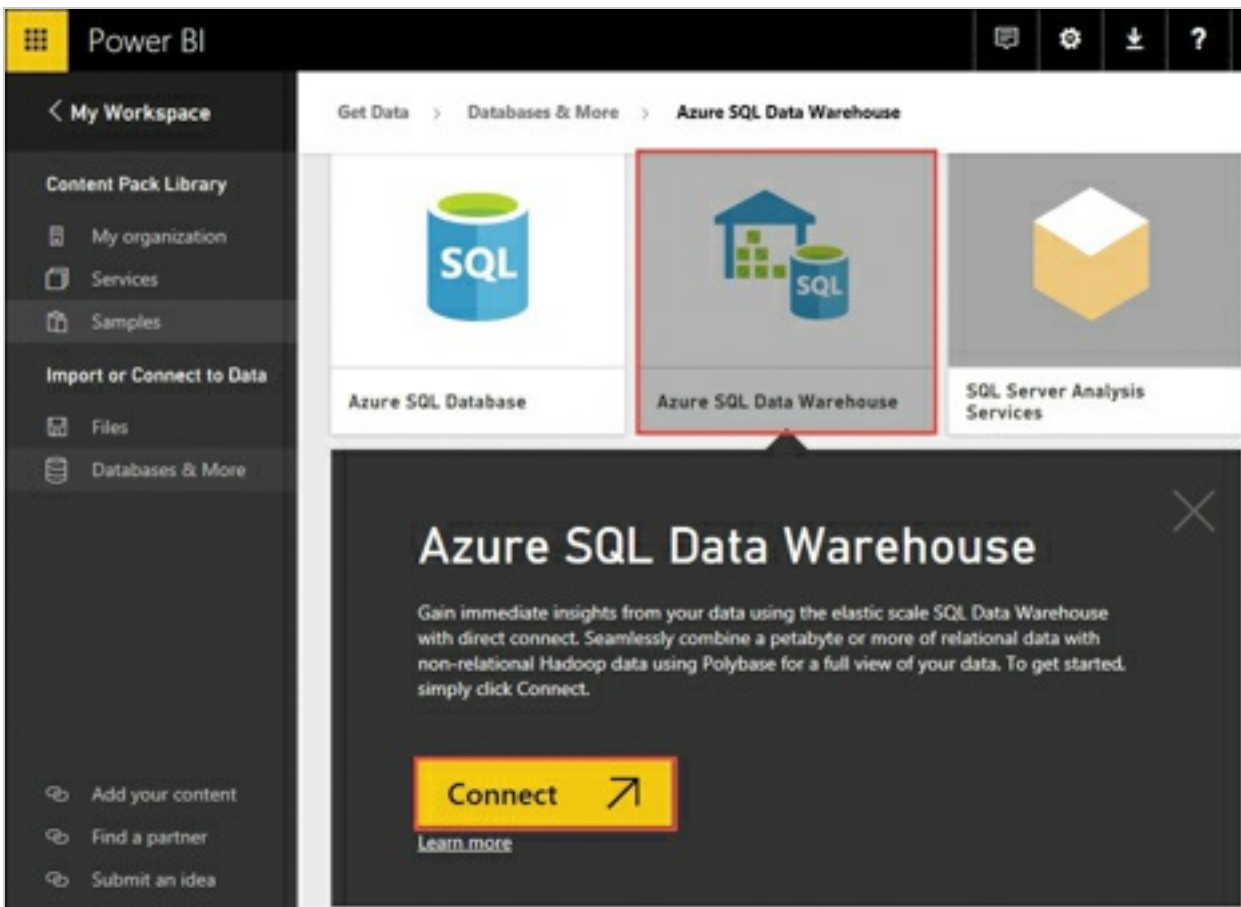
3. Open a web browser and navigate to **powerbi.com** and click **Sign in** in the upper right corner of the screen.



4. Sign in with the organizational account that you are using to access Power BI Pro.
5. Click on the **Get Databases** button.



6. On the **Get Data > Databases & More** screen, click on **Azure SQL Data Warehouse** then click **Connect**. If you are presented with a message to indicating that you need Power BI Pro, click the option to enable the trial and a Power BI Pro trial will be started for this account.



7. Supply your Azure Data Warehouse server details:

Connect to Azure SQL Data Warehouse

To get started with Azure SQL Data Warehouse, we need some information to connect you to your database.
Need help connecting? [Learn more](#)

Server

Database

☐ Enable Advanced Options

Next Cancel

8. Provide the Azure SQL Data Warehouse credentials: **demouser** and **demo@pass123**

Connect to Azure SQL Data Warehouse

To get started with Azure SQL Data Warehouse, we need so information to connect you to your database.
Need help connecting? [Learn more](#)

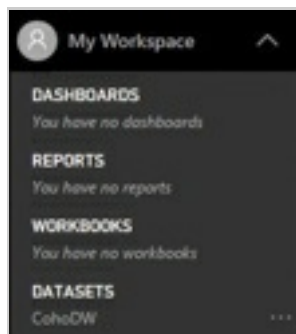
Username

Password

9. It will take 1 to 2 minutes to load the data.

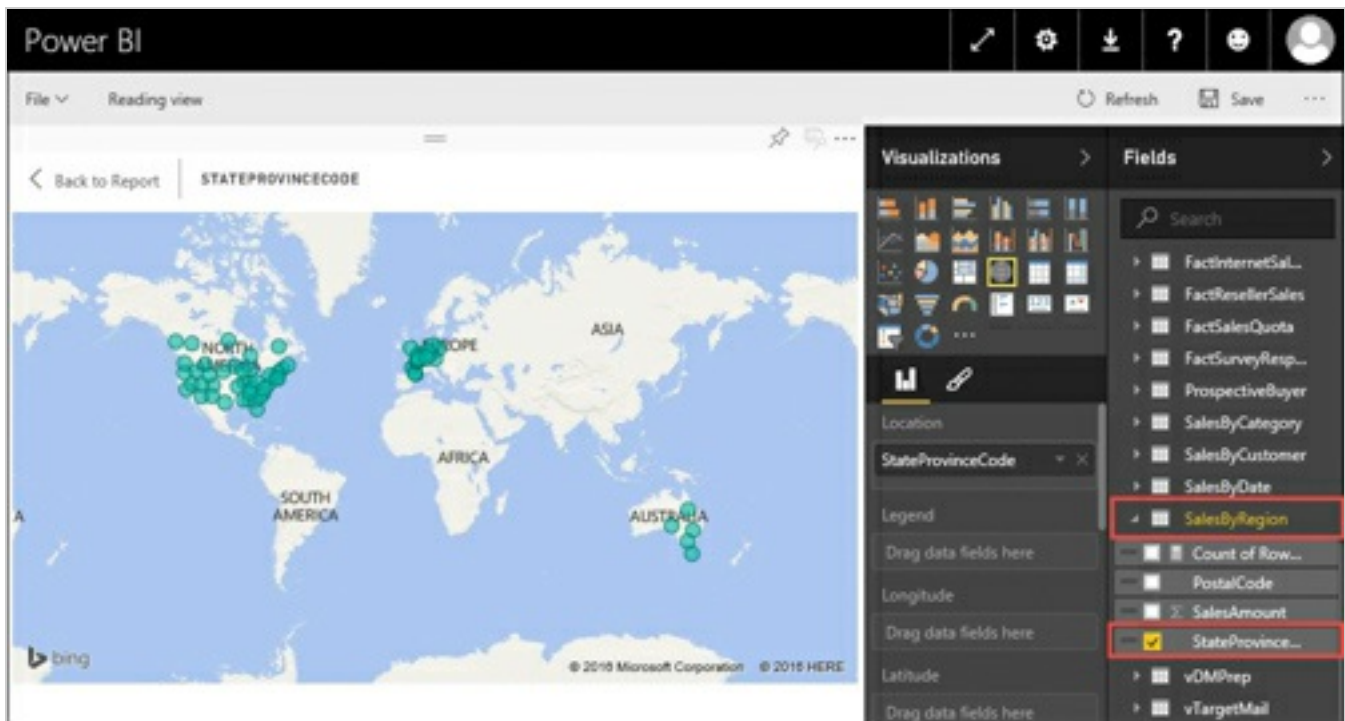


10. Once added the CohoDW SQL Data Warehouse will be shown under My Workspace > Datasets. Click on the CohoDW dataset.

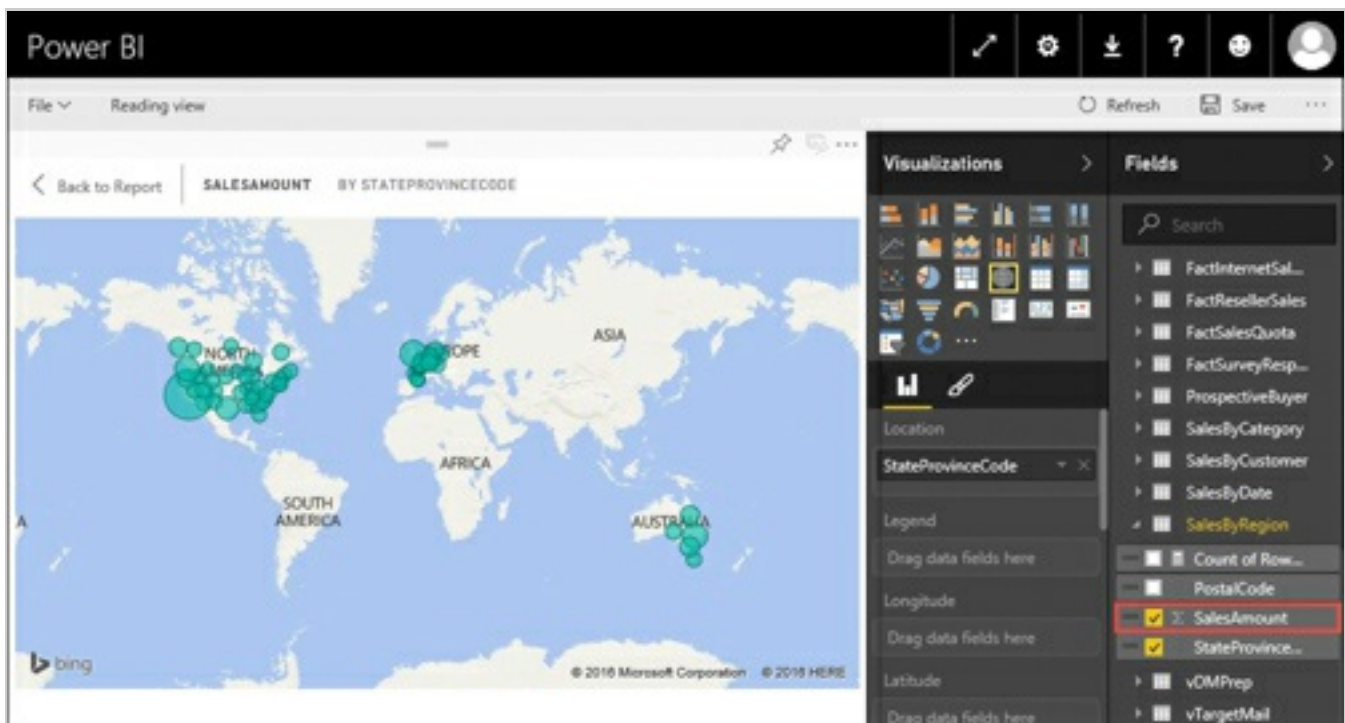


11. In PowerBI, in the Fields blade, expand the **SalesByRegion** table and check the box next to

StateProvinceCode. This will automatically launch the map visualization because PowerBI is smart enough to understand that this is geographic data.



12. The circles that PowerBI adds to the map are simply every state or province in which Coho had sales. Let's add the sales amount to this to make the map a little more interesting. Add the **SalesAmount** from the **SalesByRegion** table by putting a check next to it. The circles on the map will change in size to reflect the sum of all sales in that particular State/Province.



13. Click the **Save** button in the top right of your screen, name your report **Sales by state** and click **Save**.

Save your report

Enter a name for your report:

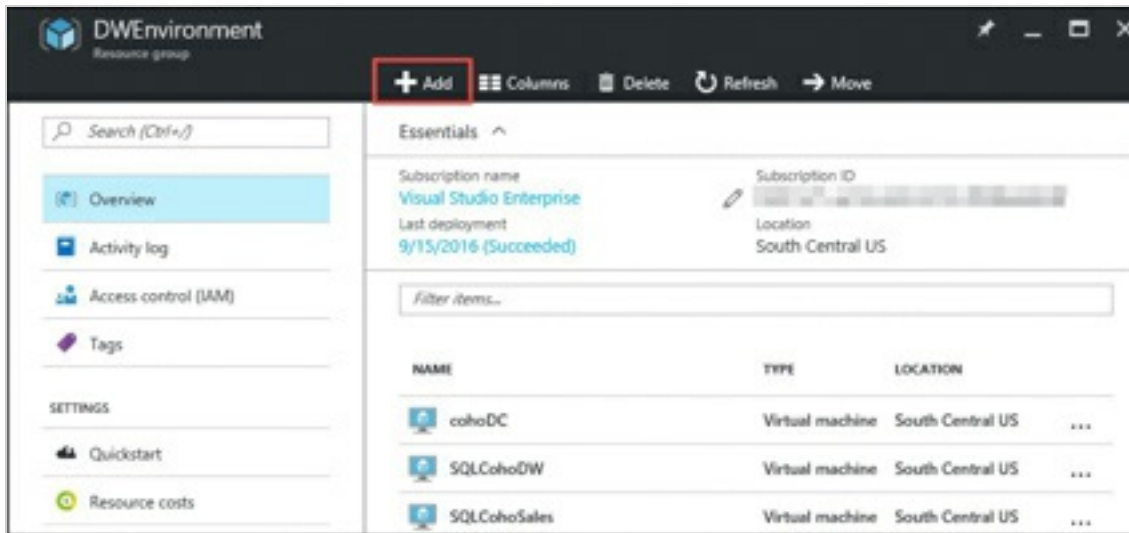
Sales by state

Save

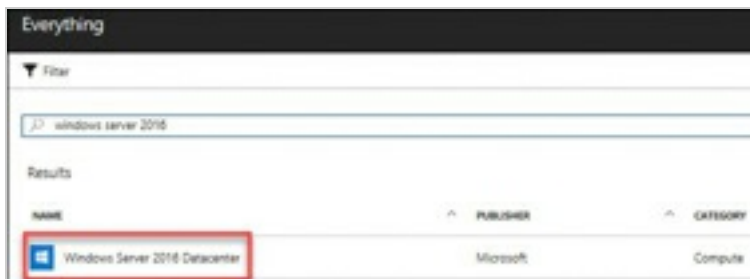
Cancel

Task 2: Configure a Data Management Gateway machine to enable on-premises data sources with Data Factory.

1. Open the Azure Portal and navigate to your DWEnvironment resource group where you deployed your “on-premises” environment.
2. In the DWEnvironment blade, click the **Add+** button.



3. In the search field type **Windows Server 2016** and hit enter, then choose **Windows Server 2016 Datacenter** from the results.



4. On the Windows Server 2016 Datacenter blade click **Create**.
5. On the Create virtual machine Basics blade, fill in the following information and then click **OK**:
 - Name: **DataGateway**
 - VM disk type: **SSD**
 - Username: **demouser**

- Password: **demo@pass123**
- Resource Group: **Use existing - DWEnvironment**
- Location: **Same location you used for this deployment**

Create virtual machine Basics

1 Basics
Configure basic settings

2 Size
Choose virtual machine size

3 Settings
Configure optional features

4 Summary
Windows Server 2018 Datacenter...

Name: DataGateway ✓

VM disk type: SSD

User name: demouser ✓

Password: demo@pass123 ✓

Confirm password: demo@pass123 ✓

Subscription: Visual Studio Enterprise

Resource group: ☐ Create new ☒ Use existing

DWEnvironment

Location: South Central US

OK

6. On the Create virtual machine, Choose a size blade, select **DS2_v2** and click **Select**.

Create virtual machine Choose a size

Browse the available sizes and their features

1 Basics
Done ✓

2 Size
Choose virtual machine size

3 Settings
Configure optional features

4 Summary
Windows Server 2012 R2 Datacenter...

Prices presented below are estimates in your local currency that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. Recommended sizes are determined by the publisher of the selected image based on hardware and software requirements.

★ Recommended | [View all](#)

DS1_V2 Standard ★	DS2_V2 Standard ★	DS11_V2 Standard ★
1 Core	2 Cores	2 Cores
3.5 GB	7 GB	14 GB
2 Data disks	4 Data disks	4 Data disks
3200 Max IOPS	6400 Max IOPS	6400 Max IOPS
7 GB Local SSD	14 GB Local SSD	28 GB Local SSD
Load balancing	Load balancing	Load balancing
Premium disk support	Premium disk support	Premium disk support
47.62 USD/MONTH (ESTIMATED)	94.49 USD/MONTH (ESTIMATED)	123.50 USD/MONTH (ESTIMATED)

Select

Note

If the DS2_v2 size is not visible you may need to select View All to see it.

7. On the Create virtual machine Settings blade, configure the following settings, take the defaults for any settings not defined here, then click **OK**:
 - Virtual network: **cohovnet**
 - Subnet: **Identity**

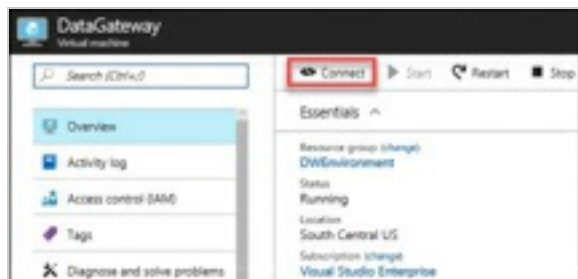
The screenshot displays the 'Create virtual machine' Settings blade. The left sidebar indicates the progress through four steps: 1 Basics (Done), 2 Size (Done), 3 Settings (Configure optional features), and 4 Summary (Windows Server 2012 R2 Data...). The main configuration area is titled 'Settings' and includes the following sections:

- Storage:** Storage account (labvmdisks345)
- Network:** Virtual network (cohovnet), Subnet (Identity (10.0.0.0/24))
- Public IP address:** (new) PowerBiGateway-ip
- Network security group (firewall):** (new) PowerBiGateway-nsg
- Extensions:** No extensions
- High availability:** Availability set (None)
- Monitoring:** Diagnostics (Enabled), Diagnostics storage account (edwstor0)

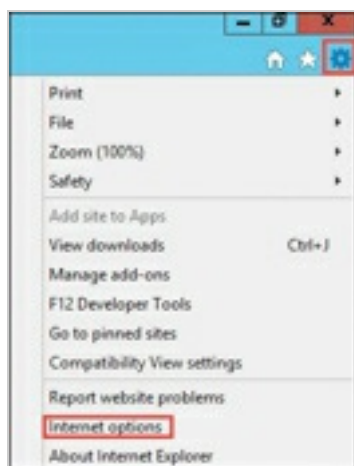
The 'OK' button at the bottom right is highlighted with a red box.

8. On the Create virtual machine Summary blade, click **OK** to begin the deployment of your Data Management Gateway machine.
9. Wait for the virtual machine to deploy before proceeding.

10. Once the virtual machine deployment has succeeded, connect to **DataGateway** using your **demouser** admin account.



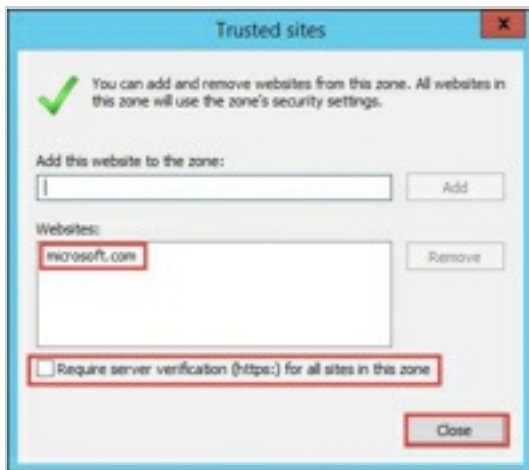
11. Once logged into **DataGateway**, open Internet Explorer, click **Settings**, then choose **Internet Options**.



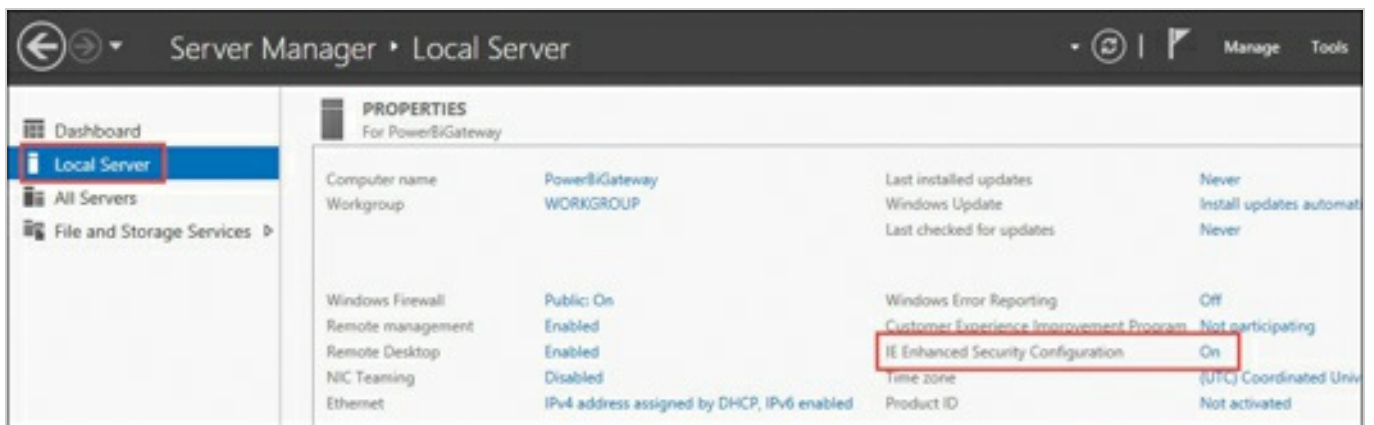
12. Click the Security tab, click on **Trusted sites**, and then click the **Sites** button.



13. On the Trusted sites window, uncheck the **Require server verification (https:) for all sites in this zone** checkbox, then add **microsoft.com** to the trusted site zone, then click **Close**.



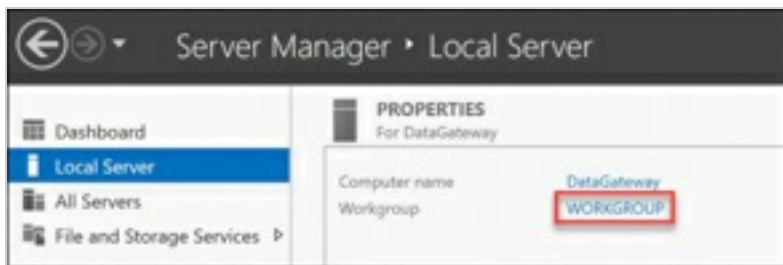
14. Click **OK**, then close Internet Explorer.
15. Navigate to Server Manager, click on **Local Server**, then click on **IE Enhanced Security Configuration**.



16. Set IE Enhanced Security Configuration to **Off** for Administrators, then click **OK**.



17. From within Server Manager, navigate to **Local Server**, then click **WORKGROUP**.



18. On the Computer Name tab, click the **Change** button.



19. Make this virtual machine a member of the **coho.com** domain then click **OK**.



20. Use **COHO\demouser** and **demo@pass123** for the domain administrator account, then click **OK, Close, Restart Now** to reboot the computer.
21. After the computer has rebooted, connect to ****DataGateway** using your **demouser** admin account.
22. Open Internet Explorer, and navigate to the Azure Portal.
23. Navigate to your Azure Data Factory that you created earlier.
24. Click the **Author and deploy** tile.



25. On the Data Factory Editor, click **...More**, then select **New data gateway**.



26. Give your gateway a unique name and click **OK**.



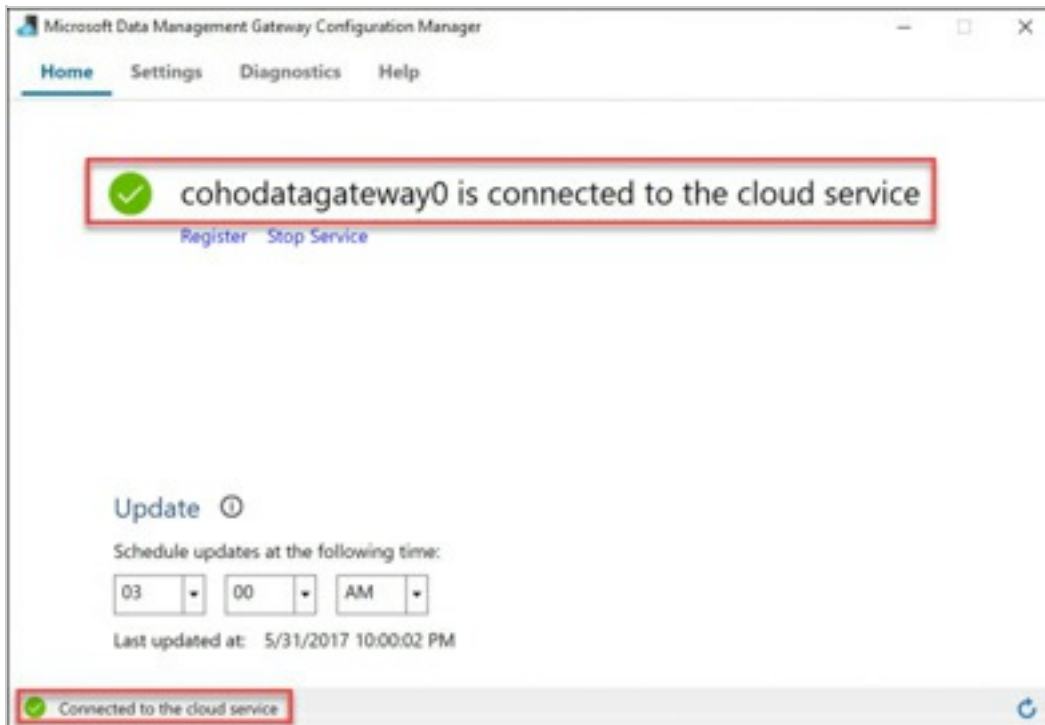
27. Click **Install directly on this computer** under Express Setup. This will launch the ClickOnce installation. Note that if you are using a browser other than Internet Explorer the install may not work correctly. If you are using Google Chrome you can install the ClickOnce plugin from the Chrome web store. Alternatively you may use the manual installation and register your Gateway with the key provided.



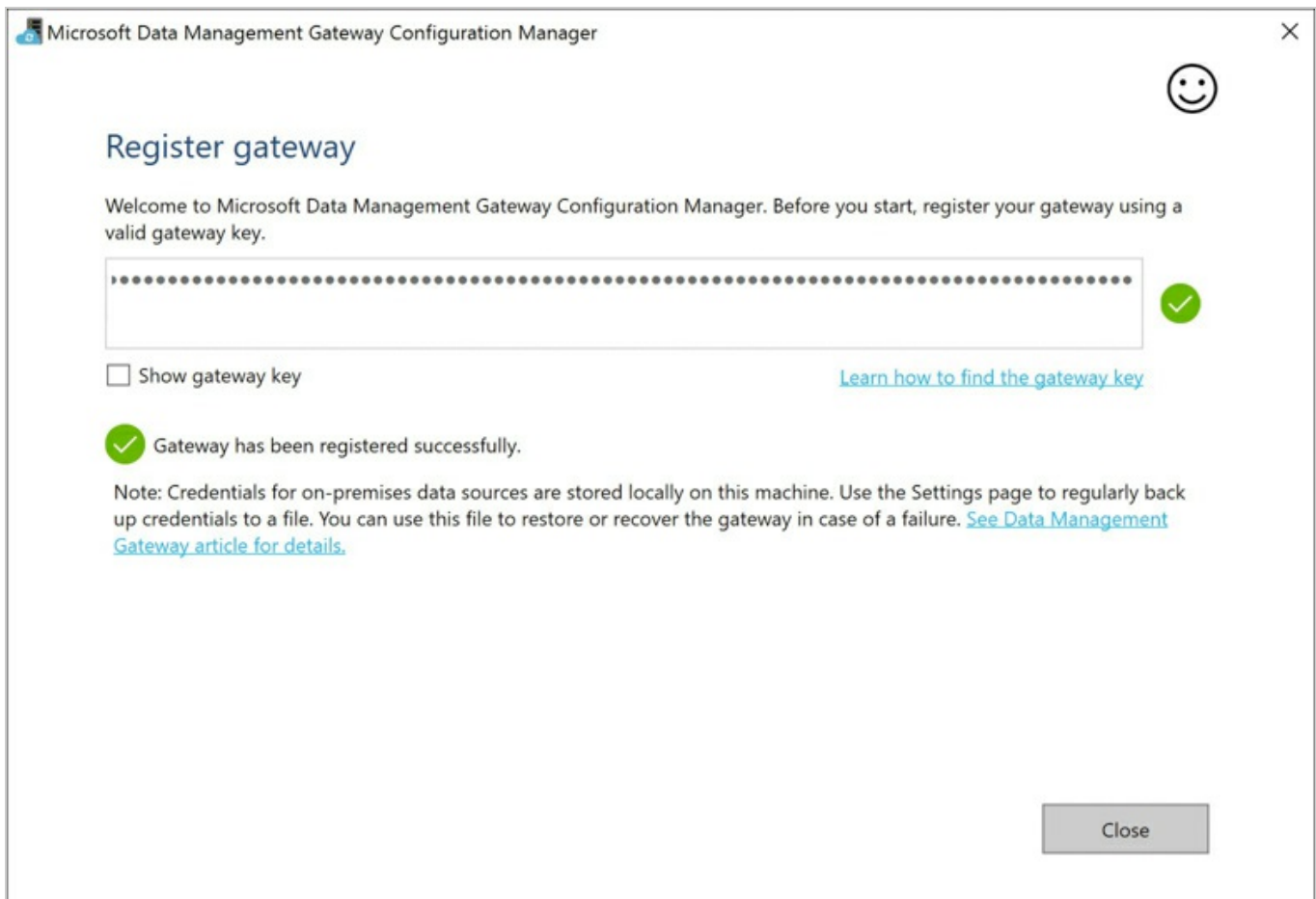
28. After the install completes, copy the **authentication key** and then click **OK**.



29. From your DataGateway machine, launch the **Microsoft Data Management Gateway Configuration Manager** and verify that your data gateway is connected to your cloud service.

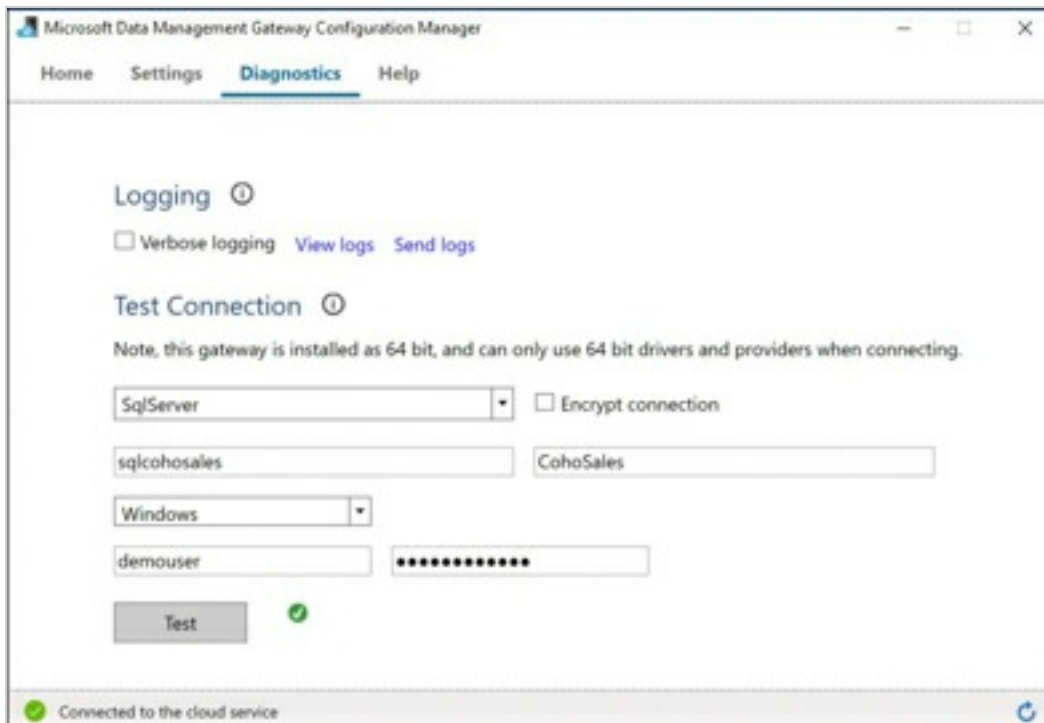


30. Register your gateway by clicking the Register link. Paste the data gateway authentication key into the box and click **Register**.



31. Test your data management gateway by clicking on the Diagnostics tab, choose **SqlServer**

for the data source type, **sqlcohosales** for the server, **CohoSales** for the database, **Windows** for the authentication mode, **demouser** for the user name and **demo@pass123** for the password.



32. From your DataGateway machine, go back to the Azure Portal and navigate to your Data Factory Editor.
33. Click the **New data store** button and choose **SQL Server** from the list.



34. Modify the existing JSON, replacing the placeholder values with the appropriate **ConnectionString**, **gatewayName**, **username** and **password**. You should set Integrated Security to **True**. Your JSON should look similar to the following:

```
Drafts/Draft-1

Add activity Encrypt Clone Discard Deploy

{
  "name": "SqlServerLinkedService",
  "properties": {
    "type": "OnPremisesSqlServer",
    "description": "",
    "typeProperties": {
      "connectionString": "Data Source=sqlcohosales;Initial Catalog=cohosales;Integrated Security=True;User ID=demouser;Password=demo@pass123;",
      "gatewayName": "cohodatagateway0",
      "userName": "demouser",
      "password": "demo@pass123"
    }
  }
}
```

35. Click the **Encrypt** button to launch the Credentials Manager application and encrypt the Credentials. If you are using the Chrome browser this may not work without a ClickOnce extension installed.



36. After the Credentials Manager pops up, change the authentication to **Windows Authentication**, set the username to **demouser** and the password to **demo@pass123** then click **OK**.

Setting Credentials

SqlServerLinkedService

AUTHENTICATION

Windows Authentication

USERNAME

demouser

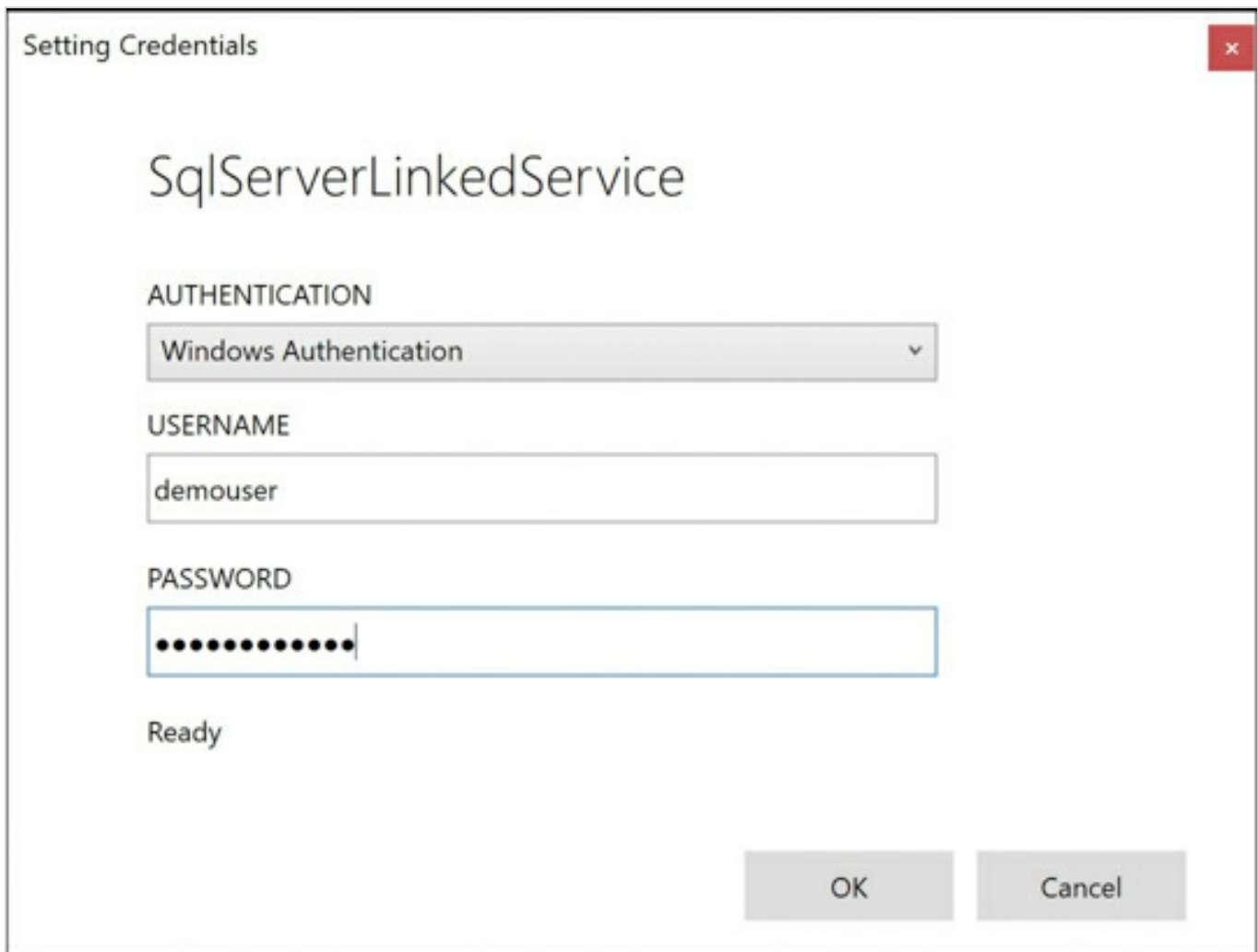
PASSWORD

.....

Ready

OK Cancel

37. Credentials Manager will encrypt your credentials, the result should look similar to this in your JSON document.



The image shows a 'Setting Credentials' dialog box for a 'SqlServerLinkedService'. It has a title bar with a close button. The main content area is titled 'SqlServerLinkedService'. Below this, there is an 'AUTHENTICATION' section with a dropdown menu set to 'Windows Authentication'. Below that is a 'USERNAME' field containing 'demouser'. Below that is a 'PASSWORD' field filled with dots. At the bottom left, it says 'Ready'. At the bottom right, there are 'OK' and 'Cancel' buttons.

38. Click the Deploy button to deploy your SQL Server Linked Server.



39. Go back to the Data Factory Overview and click the **Copy data** tile to launch the copy data wizard.



40. On the Properties page, set the Task name to **CohoSalesPipeline**, and set the schedule to **Run once now**.

Copy Data (cohodatafactory)

1 Properties

2 Source

3 Destination

4 Summary

5 Deployment

Properties

Enter name and description for the copy data task and specify how often you want to run the task.

Task name (required) ⓘ
CohoSalesPipeline

Task description
Enter description here

Task cadence or Task schedule

☒ Run once now
☐ Run regularly on schedule

Expiration time (required) ⓘ
3.00:00:00

Note

In a production scenario, you would schedule this pipeline to run on a recurring schedule based on business requirements but for our purposes a one-time copy will be sufficient.

41. On the Source data store page, click the From Existing Connections tab, then select the SqlServerLinkedService.

Copy Data (cohodatafactory)

1 Properties
Recurring copy

2 Source

3 Destination

4 Summary

5 Deployment

Source data store

Specify the source data store for the copy task. You can use an existing data store connection or specify a new data store. Click [HERE](#) to suggest new copy sources or give comments.

FROM EXISTING CONNECTIONS

CONNECT TO A DATA STORE

All ▼ Filter by name

SqlServerLinkedService

42. Click **Next** on the Connection properties page.

43. On the Select tables page, select **Sales.SalesOrderHeader** and **Sales.SalesTerritory** tables and click **Next**.

Copy Data (cohodatafactory)

1 Properties
Recurring copy

2 Source
Connection
Dataset

3 Destination

4 Summary

5 Deployment

Select tables from which to copy the data or use a custom query.
You can select multiple tables, or you can provide single custom query

EXISTING TABLES USE QUERY

Filter by table name ☐ Show views

- ☒ [Sales].[SalesOrderHeader]
- ☐ [Sales].[SalesOrderHeader_inmem]
- ☐ [Sales].[SalesOrderHeader_ondisk]
- ☐ [Sales].[SalesOrderHeaderSalesReason]
- ☐ [Sales].[SalesPerson]
- ☐ [Sales].[SalesPersonQuotaHistory]
- ☐ [Sales].[SalesReason]
- ☐ [Sales].[SalesTaxRate]
- ☒ [Sales].[SalesTerritory]

44. On the Destination data store page, select **Azure SQL Data Warehouse** and click **Next**.



45. Specify the Azure SQL Data Warehouse configuration for your Azure SQL Data Warehouse.

Specify the Azure SQL Data Warehouse

Connection name *(required)* ⓘ

Server / database selection method *(required)* ⓘ

Azure subscription *(required)* ⓘ

Server name *(required)*

Database name *(required)*

User name *(required)*

Password *(required)*

46. Accept the defaults on the Table mapping page and click **Next**.

Copy Data (cohodatafactory)

1 Properties
Recurring copy

2 Source
SQL Server

3 Destination
Connection
Dataset

4 Performance
Parallel copy, Polybase, Staging

5 Summary

6 Deployment

Table mapping

For each table you have selected to copy in the source data store, select a corresponding table in the destination data store or specify the stored procedure to run at the destination.

Source	Destination
<input checked="" type="checkbox"/> [Sales].[SalesOrderHeader]	→ [Sales].[SalesOrderHeader] (auto-create) Use existing table
<input checked="" type="checkbox"/> [Sales].[SalesTerritory]	→ [Sales].[SalesTerritory] (auto-create) Use existing table

47. Accept the defaults on the Schema mapping page and click **Next**.

Copy Data (cohodatafactory)

1 Properties

One time copy

2 Source

SQL Server

3 Destination

Connection

Dataset

4 Performance

Parallel copy, Polybase, Staging

5 Summary

6 Deployment

Schema mapping

Choose how source and destination columns are mapped

Source: [Sales].[SalesOrderHeader]

Destination: [Sales].[SalesOrderHeader]

Source: [Sales].[SalesTerritory]

Destination: [Sales].[SalesTerritory]

[Sales].[SalesOrderHeader]		[Sales].[SalesOrderHeader]	Include this column
SalesOrderID (Int32)	→	SalesOrderID (Int32)	<input checked="" type="checkbox"/>
RevisionNumber (Int16)	→	RevisionNumber (Int16)	<input checked="" type="checkbox"/>
OrderDate (DateTime)	→	OrderDate (DateTime)	<input checked="" type="checkbox"/>
DueDate (DateTime)	→	DueDate (DateTime)	<input checked="" type="checkbox"/>
ShipDate (DateTime)	→	ShipDate (DateTime)	<input checked="" type="checkbox"/>
Status (Int16)	→	Status (Int16)	<input checked="" type="checkbox"/>
OnlineOrderFlag (Boolean)	→	OnlineOrderFlag (Boolean)	<input checked="" type="checkbox"/>
SalesOrderNumber (String)	→	SalesOrderNumber (String)	<input checked="" type="checkbox"/>
PurchaseOrderNumber (String)	→	PurchaseOrderNumber (String)	<input checked="" type="checkbox"/>
AccountNumber (String)	→	AccountNumber (String)	<input checked="" type="checkbox"/>

Repeatability settings

Method

None

48. On the Performance settings page, you will specify a storage account that will be used to stage your data before Data Factory copies the data using Polybase. Specify a **New connection**, change the connection name to **CohoStagingStorage**, and choose the storage account you used during your migration for the storage account name.

Copy Data (cohodatafactory)

- 1 Properties
Recurring copy
- 2 Source
SQL Server
- 3 Destination
Azure SQL Data Warehouse
- 4 Performance
- 5 Summary
- 6 Deployment

Performance settings

Performance improvement options

Staging storage account ⓘ

Use staging account

New connection ▼

Connection name (required) ⓘ
CohoStagingStorage ✕

Account selection method (required) ⓘ
From Azure subscriptions ▼

Azure subscription (required) ⓘ
Visual Studio Enterprise (f2857e77-ee7b-4305-5 ▼

Storage account name (required)
edwstor01 ▼

Advanced settings

49. The summary page will give you a diagram and the configuration of what you are about to build. Click **Next** to initiate the deployment.

Copy Data (cohodatafactory)

- 1 Properties
Recurring copy
- 2 Source
SQL Server
- 3 Destination
Azure SQL Data Warehouse
- 4 Performance
Parallel copy, Polybase, Staging
- 5 Summary
- 6 Deployment

Summary

You are running scheduled pipeline to copy data from SQL Server to Azure SQL Data Warehouse.
2 tables will be copied

```

graph LR
    SQL[SQL Server  
2 table(s)  
Region: On Premises] -- "Copy Run Time Region: On Premises" --> Blob[Blob (staging)  
Region: South Central US]
    Blob -- "Copy Run Time Region: South Central US" --> Azure[Azure SQL Data Warehouse  
2 table(s)  
Region: South Central US]
  
```

Properties

[Edit](#)

Task name: CohoSalesPipeline

Task description: <no description has been provided>

Task cadence: Minute, every 15 minutes between Tue, 01 Jan 2008 15:41:00 GMT and Thu, 31 Dec 2099 00:00:00 GMT

Source

[Edit](#)

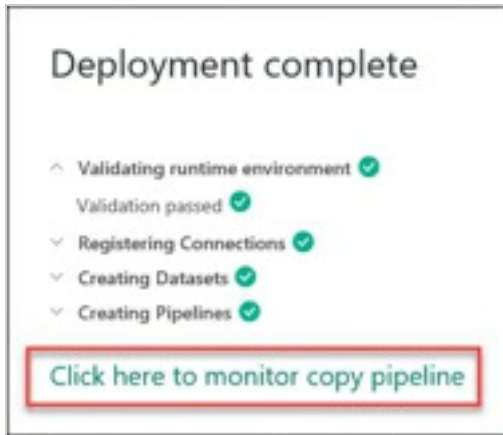
Existing Connection: SqlServerLinkedService

Dataset name prefix: InputDatasets-crx

Table: 2 table(s) ⓘ

Region: On Premises

50. After the deployment completes, wait about 5 minutes then click on the **Click here to monitor copy pipeline** link.



- Click on each of the pipelines in the lower pane of dashboard. Scroll down on the right pane to see status of the pipeline and the metrics from execution.

Data factory

RESOURCE EXPLORER

- Data Factories
 - cohodatafactory
 - Pipelines
 - CohoSalesPipeline
 - Datasets
 - Linked services
 - CohoStagingStorage
 - SQLDW
 - SqServerLinkedService
 - Gateways
 - cohodatagateway

cohodatafactory / CohoSalesPipeline

Start time (UTC): 05/29/2017 04:58 pm End time (UTC): 06/01/2017 04:58 pm Apply

No scheduled runs for this pipeline.

This pipeline has no content to display.

ACTIVITY WINDOWS

1 filter applied

Pipeline	Activity	Window Start	Window End	Status	Type	Last Attempt	Last Attempt	Duration	Retry Attempts
CohoSalesPp...	Activity-1 (Sal...	05/29/2017 12...	05/30/2017 12...	Ready	Copy	06/01/2017 4...	06/01/2017 4...	00:01:07	1
CohoSalesPp...	Activity-0 (Sal...	05/29/2017 12...	05/30/2017 12...	Ready	Copy	06/01/2017 4...	06/01/2017 4...	00:01:07	1

Activity Window

Start & End Time
05/29/2017 12:00 AM UTC - 05/30/2017 12:00 AM UTC

Status
Ready

Activity
Activity-1-[Sales]-[SalesTerritory]->[Sales]-[SalesTerritory]

Pipeline
CohoSalesPipeline

Input Datasets
InputDatasets-n32-[Sales]-[SalesTerritory]

Output Datasets
OutputDatasets-n32-[Sales]-[SalesTerritory]

Attempts

06/01/2017 4:56 PM UTC

Succeeded(100% complete)

Run ID
15308f84-ec73-4b0b-893f-d4784e06067b_636316128000000000_636316992000000000_OutputDatasets-n32-[Sales]-[SalesTerritory]

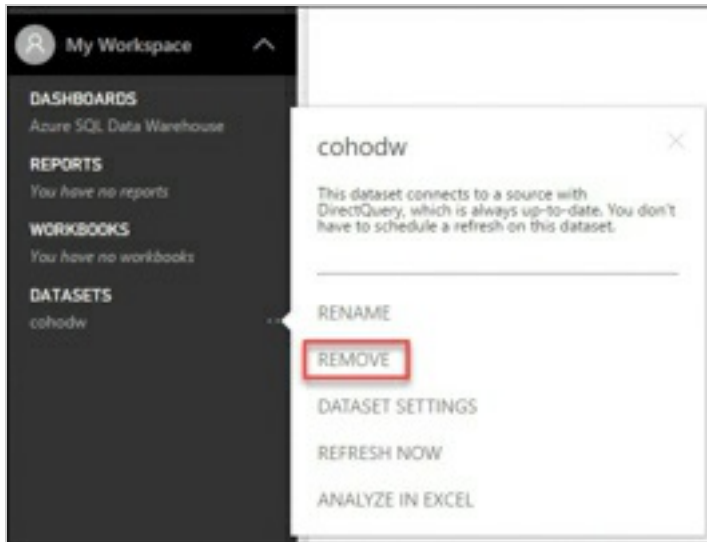
Data read: 1.30 KB
Data written: 1.46 KB
Rows: 10
Throughput: 106.44 Bytes/s
Copy duration: 00:00:12
Billed duration for data movement: 00:00:06

Source to staging blob runtime region: On premises, gateway name: cohodatagateway
Staging blob to destination runtime region: South Central US
SQL DW Polybase: True
Gateway node: DATAGATEWAY

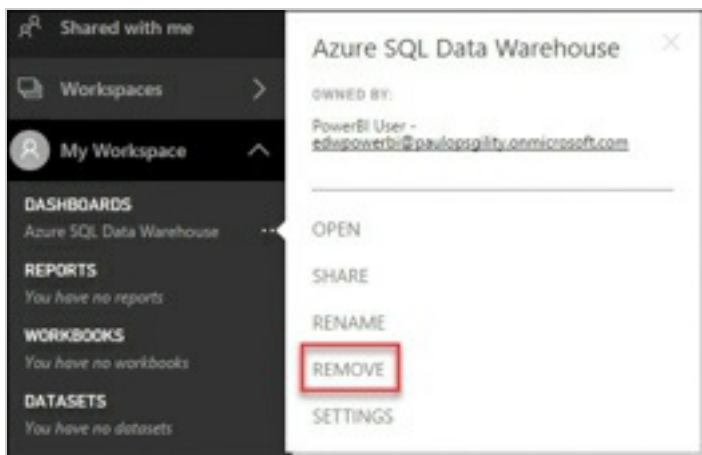
- Open Internet Explorer and navigate to **powerbi.com** and click **Sign in** in the upper right corner of the screen.



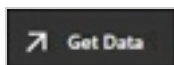
- Sign in with the organizational account that you are using to access Power Bi Pro.
- Expand My Workspace, click the ellipses next to your dataset and choose remove.



55. Click the ellipses next to your Azure SQL Data Warehouse dashboard and remove it.



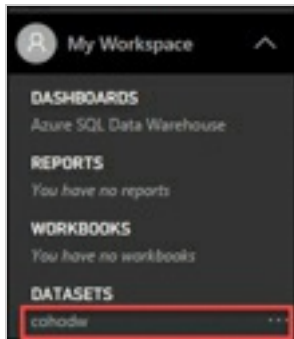
56. Click on the Get Data link in the lower left corner of the screen.



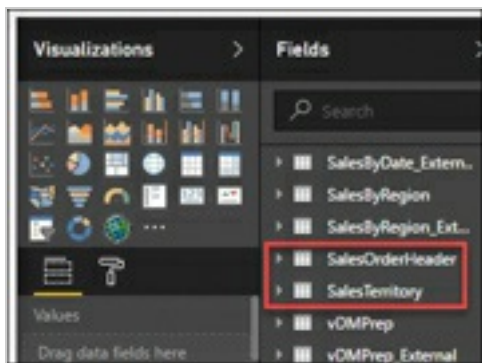
57. Setup a new connection to your Azure SQL Data Warehouse.

58. If your new connection and dataset do not show up after the import, you may need to refresh your browser.

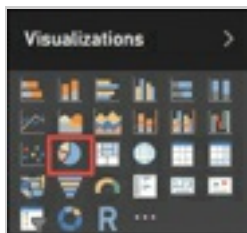
59. Expand My Workspace and click on your dataset.



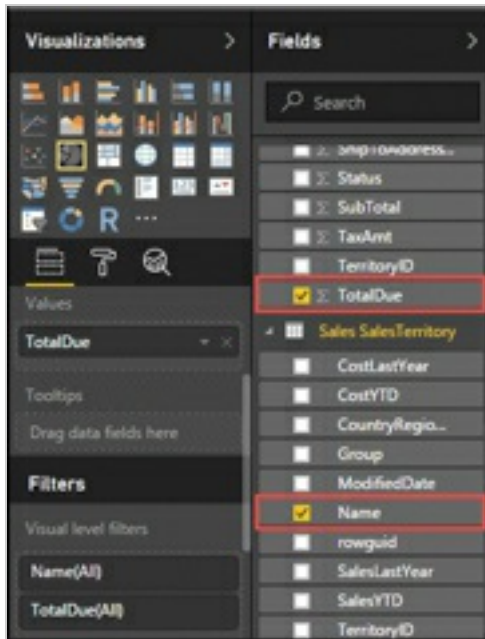
60. Scroll down through your list of fields and you should now see the **SalesOrderHeader** and **SalesTerritory** tables that we imported using Azure Data Factory.



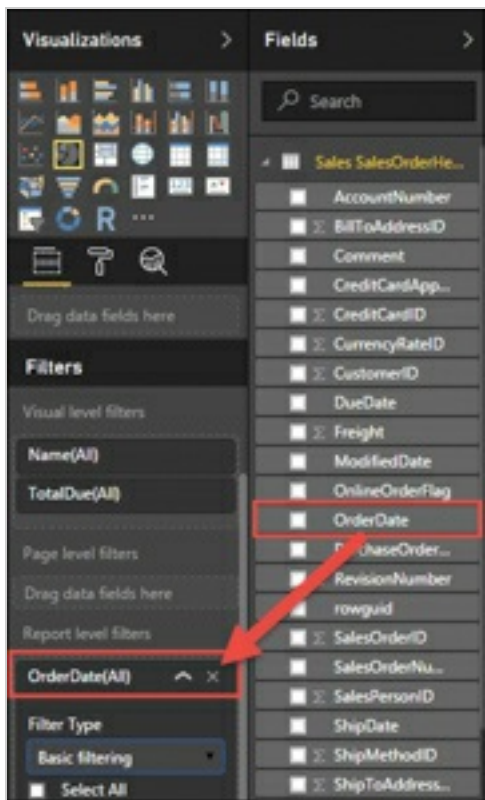
61. On the visualizations blade click the **pie-chart** icon.



62. On the Fields blade, expand the SalesOrderHeader table and put a check next to **Σ TotalDue**, expand SalesTerritory and put a check next to **Name**.

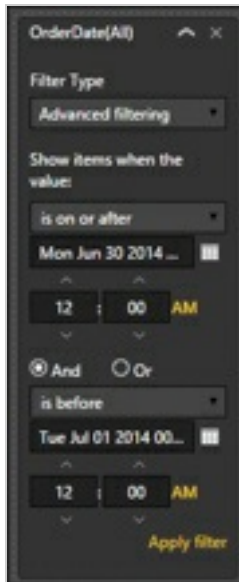


63. Drag the **OrderDate** field from the SalesOrderHeader table to the Report level filters.



64. Change the Filter Type to **Advanced filtering**, set the following values and then click **Apply filter**:

- Show items when the value: **is on or after**
- Date: **2014-06-30**
- Time: **12:00AM**
- And: **is before**
- Date: **2014-07-01**
- Time: **12:00AM**



65. Click **save** and save your report with the name **Sales by Region 6-30-2014**

Summary

In this exercise, you have configured Power BI to connect directly to an Azure SQL Data Warehouse. You then deployed an Data Management Gateway to enable access to on-premises data sources from Azure Data Factory. You created a Copy Data pipeline in Azure Data Factory to update data in Azure SQL Data Warehouse. Finally, you created a new Power BI report to leverage the new data in your Azure SQL Data Warehouse.