Azure Databricks Intro Workshop

Lab 1: Setting up the Lab

Technologies showcased: Azure Portal

Author: Mike Dobing Data Platform Solution Architect

Date: 11/6/2018

Table of Contents

[Summary 3](#_Toc519584237)

[Pre-requisites 3](#_Toc519584238)

[Scenario 3](#_Toc519584239)

[Part 1 – Create A SQL Datawarehouse 3](#_Toc519584240)

[Part 2 – Create A Server Level Firewall Rule 9](#_Toc519584241)

[Part 3 – Get Fully Qualified Server name 11](#_Toc519584242)

[Part 4 – Connect to the server as server admin 11](#_Toc519584243)

[Part 5 – Create Database Master Key 14](#_Toc519584244)

[Part 6 – Create a Blob Storage Account 15](#_Toc519584245)

## Summary

This lab module will walk through necessary steps to deploy and configure all the resources need for the rest of the lab. We will be performing the following tasks:

* Create a SQL Datawarehouse Instance using the Azure Portal
* Create a database master key for the Azure SQL Data Warehouse
* Create an Azure Blob storage account, and a container within it. Also, retrieve the access key to access the storage account

## Pre-requisites

* Azure Subscription with rights to use/deploy Azure services, and X of Azure credit
* SQL Server Management Studio (<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>)
* Web browser (Edge/Chrome recommended)

## 

## Scenario

|  |  |  |
| --- | --- | --- |
| Part 1 – Create A SQL Datawarehouse | | |
| **Scenario** | | |
| First, we’re going to provision a SQL Data Warehouse instance to be used for writing from Databricks | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
| Here we use the Azure portal to create the relevant services | 1. Click **Create a resource** in the upper left-hand corner of the Azure portal. 2. Select **Databases** from the **New** page, and select **SQL Data Warehouse** under **Featured** on the **New** page. |  |
|  | 1. Fill out the SQL Data Warehouse form with the following information:  | Setting | Suggested value | Description | | --- | --- | --- | | **Database name** | mySampleDataWarehouse | For valid database names, see [Database Identifiers](https://docs.microsoft.com/en-us/sql/relational-databases/databases/database-identifiers). Note, a data warehouse is a type of database. | | **Subscription** | Your subscription | For details about your subscriptions, see [Subscriptions](https://account.windowsazure.com/Subscriptions). | | **Resource group** | Your Resource Group | For valid resource group names, see [Naming rules and restrictions](https://docs.microsoft.com/azure/architecture/best-practices/naming-conventions). | | **Select source** | Sample | Specifies to load a sample database. Note, a data warehouse is one type of database. | | **Select sample** | AdventureWorksDW | Specifies to load the AdventureWorksDW sample database. | |  |
|  | 1. Click **Server** to create and configure a new server for your new database. Fill out the **New server form** with the following information:  | Setting | Suggested value | | --- | --- | | **Server name** | Any globally unique name | | **Server admin login** | Any valid name | | **Password** | Any valid password | | **Location** | Nearest Location To You |   Note the values you used for username and password. |  |
|  | 1. Click Select 2. Click Performance level to specify the performance configuration for the data warehouse. 3. For this tutorial, select Gen2. The slider, by default, is set to DW1000c. Try moving it up and down to see how it works. 4. Click Apply |  |
|  | 1. Now that you have completed the SQL Data Warehouse form, click Create to provision the database. Provisioning takes a few minutes. |  |
| Note depending on upload speed this process may take 10-45 minutes. The OLTP database restore is also a significant portion of this time. | 1. On the toolbar, click Notifications to monitor the deployment process. |  |

|  |  |  |
| --- | --- | --- |
| Part 2 – Create A Server Level Firewall Rule | | |
| **Scenario** | | |
| The SQL Data Warehouse service creates a firewall at the server-level that prevents external applications and tools from connecting to the server or any databases on the server. To enable connectivity, you can add firewall rules that enable connectivity for specific IP addresses. Follow these steps to create a server-level firewall rule for your client's IP address. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
|  | 1. After the deployment completes, click **SQL data warehouses** from the left-hand menu and then click **mySampleDatabase** on the **SQL data warehouses** page. The overview page for your database opens, showing you the fully qualified server name (such as **mynewserver-20180430.database.windows.net**) and provides options for further configuration. 2. Copy this fully qualified server name for use to connect to your server and its databases in subsequent quick starts. To open server settings, click the server name. |  |
|  | 1. To open server settings, 2. click the server name. 3. Click Show firewall settings. The Firewall settings page for the SQL Database server opens. |  |
| This step adds your IP address to the firewall allowing you to connect to the DW | 1. To add your current IP address to a new firewall rule, click Add client IP on the toolbar. A firewall rule can open port 1433 for a single IP address or a range of IP addresses. 2. Click Save. A server-level firewall rule is created for your current IP address opening port 1433 on the logical server. 3. Click OK and then close the Firewall settings page. |  |

|  |  |  |
| --- | --- | --- |
| Part 3 – Get Fully Qualified Server name | | |
| **Scenario** | | |
| Get the fully qualified server name for your SQL server in the Azure portal. Later you use the fully qualified name when connecting to the server. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
|  | 1. Select SQL Data warehouses from the left-hand menu, and click your data warehouse on the SQL datawarehouses page. 2. In the Essentials pane in the Azure portal page for your database, locate and then copy the Server name. In this example, the fully qualified name is mynewserver-20180430.database.windows.net. |  |

|  |  |  |
| --- | --- | --- |
| Part 4 – Connect to the server as server admin | | |
| **Scenario** | | |
| This section uses SQL Server Management Studio (SSMS) to establish a connection to your Azure SQL server. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
|  | 1. Open SQL Server Management Studio.  | Setting | Suggested value | Description | | --- | --- | --- | | Server type | Database engine | This value is required | | Server name | The fully qualified server name | Here's an example: **mynewserver-20180430.database.windows.net**. | | Authentication | SQL Server Authentication | SQL Authentication is the only authentication type that is configured in this tutorial. | | Login | The server admin account | This is the account that you specified when you created the server. | | Password | The password for your server admin account | This is the password that you specified when you created the server. |  1. In the **Connect to Server** dialog box, enter the following information: |  |
|  | 1. Click **Connect**. The Object Explorer window opens in SSMS. 2. In Object Explorer, expand **Databases**. Then expand **mySampleDatabase** to view the objects in your new database. |  |
|  |  |  |
| Part 5 – Create Database Master Key | | |
| **Scenario** | | |
| This topic describes how to create a database master key in SQL Server 2017 by using Transact-SQL. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
|  | 1. Choose a password for encrypting the copy of the master key that will be stored in the database. 2. In **Object Explorer**, connect to an instance of Database Engine. 3. On the Standard bar, click **New Query**. 4. Copy and paste the following example into the query window and click **Execute**. |  |
|  | Open a new query to your newly created database  Run the code opposite | CREATE MASTER KEY ENCRYPTION BY PASSWORD = ‘<YourOwnPassword>’; |

|  |  |  |
| --- | --- | --- |
| Part 6 – Create a Blob Storage Account | | |
| **Scenario** | | |
| First, create a new general-purpose storage account to use for this lab. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
| Create a general storage account using the azure portal and obtain the access key for later use | 1. Go to the [Azure portal](https://portal.azure.com/#create/Microsoft.StorageAccount-ARM) and sign in by using your Azure account. 2. Create a Storage Account 3. Use StorageV2 4. Enter a unique name for your storage account. Keep these rules in mind for naming your storage account:    * The name must be 3 to 24 characters in length.    * The name can contain numbers and lowercase letters only. 5. Select your subscription. 6. Create a new **Resource group** and give it a unique name. 7. Select the **Location** to use for your storage account. 8. Leave other fields set to their default values. 9. Select **Pin to dashboard** and select **Create** to create your storage account.   After your storage account is created, it's pinned to the dashboard of the Azure portal. Select the storage account to open it. Under **Settings**, select **Access keys**. Select the primary account access key and click the **Copy** button to copy the associated connection string to the clipboard. Then paste the string into a text editor for later use. |  |
| Create a container in the account | 1. Navigate to your new storage account in the Azure portal. 2. In the left menu for the storage account, scroll to the **Blob Service** section, then select **Browse Blobs**. 3. Click the **Add Container** button. 4. Enter a name for your new container. The container name must be lowercase, must start with a letter or numbers and can contain only letters, numbers, and the dash (-) character. See [Naming and Referencing Containers, Blobs, and Metadata](https://docs.microsoft.com/rest/api/storageservices/naming-and-referencing-containers--blobs--and-metadata) for more information about container and blob names. 5. Set the level of public access to the container. The default level is **Private (no anonymous access)**. 6. Click **OK** to create the container. |  |

**IMPORTANT: AVOID INCURRING EXTRA CHARGES BY PAUSING YOUR SUBSCRIPTION RESOURCES**