Lab 9: Provisioning of Azure Data Factory job & implementation of activity /pipeline

Technologies showcased: PowerShell, ARM templates, Azure Portal

## Pre-requisites

* Azure Subscription with rights to use/deploy Azure services, and X of Azure credit
* Azure PowerShell (<https://docs.microsoft.com/en-us/powershell/azure/install-azurerm-ps?view=azurermps-5.1.1>)
* Deployment files for this Lab located at [GitHub link], downloaded to a local folder
* SQL Server Management Studio (<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>)
* (optional) Microsoft Azure Storage Explorer (<https://azure.microsoft.com/en-us/features/storage-explorer/>)
* Provision Azure SQL Database with Sample AdventureWorks database. (AdventureWorks DB)( <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-single-database-get-started>)
* Web browser (Edge/Chrome recommended)

## 

## Scenario

|  |  |  |
| --- | --- | --- |
| Part 1 – Create an Azure Data Factory | | |
| **Scenario** | | |
| We are going to use the portal to create the Azure Data Factory we will be using in our future lab modules. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
|  | 1. Navigate to the Azure portal within your web browser and navigate to <https://portal.azure.com>. 2. Click the Resource Group icon in the left menu. 3. Click the Resource group name you configured in the PowerShell script. |  |
|  | 1. Click the Add button in the right pane. |  |
|  | 1. Type Data Factory in the search box. 2. Click Data Factory. |  |
|  | 1. Click Create. |  |
| After the ADF is deployed we are ready to start the lab. | 1. Name your Data Factory. 2. The rest of the information should be filled out since you created this from the Resource Group pane. Verify the version is V2. 3. Click Create. |  |

## Part 4.1 Copy Data from Azure SQL DB to Blob Storage(v2)

|  |  |
| --- | --- |
| Open Azure Data Factory v2 ‘Author & Monitor’ section from Portal & select ‘Copy data’ section.   * Provide the task name & share the ‘Task cadence’ or ‘ Task schedule’ * Next ‘under Azure’ provide your Azure SQL DB connection details. * Provide the respective Azure SQL DB single instance details(e.g. Server name, DB name, Connection string/Azure Key vault) * Test the connection before proceeding & once it’s successful, proceed to Next step. * Select ‘existing tables’ or write custom query to move data from Azure SQL DB. * Next, in the destination source, select ‘Azure Blob storage’ & provide the respective storage account & its access details. Test the connection & finish the data destination pipeline. * In the next screen, provide the folder of data path(i.e. Blob container details in this case) you may provide any customized name, file type extension(default .txt) & choose any compression type(bzip2, gzip, deflate, zipdeflate). * Finally share the destination file format details for moved data file format details, column delimiter, row delimiter, line count, header to the file etc. & click on next. * Next step, you can configure fault tolerance settings & performance setting by enabling staging if needed (for e.g. SQL db to SQL DW) staging via azure blob storage. * In the finalized summary screen, check source to destination copy activity details from Azure SQL db to Azure Blob storage(in this demo) & click on next to finish the ETL pipeline. |  |

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