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

Technologies showcased: PowerShell, ARM templates, Azure Portal

## 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:

* Running a PowerShell file to configure and deploy our Azure resources needed. This includes
  + Azure storage account
  + Azure SQL Server
  + Azure Logic App
  + Azure Data warehouse
  + Office365 API connection
* Configure the Office365 Authentication for sending email in future lab modules (optional but recommended)

Note: X amount of MBs of text, CSV, and bacpac files will be uploaded as part of this deployment script. Upload times for this content will depend on your internet upload speed.

## 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/>)
* Web browser (Edge/Chrome recommended)

## 

## Scenario

|  |  |  |
| --- | --- | --- |
| Part 1 – Configure the LabDeploy PowerShell script from Github | | |
| **Scenario** | | |
| First, we are going to configure the PowerShell variables for setting up configurations needed for server resources. | | |
| **Commentary / Notes** | **Click Steps & ‘Bits’** | **Screenshots** |
| This file should be in the Deployment folder from the resources downloaded from the GitHub link here: <https://github.com/imcuteani/adflab>.  Lab is assuming using the Windows PowerShell ISE to edit and run the deployment script. You could use another text editor to edit and then run the PowerShell through the command line or in Explorer. | 1. Locate and run the Windows PowerShell ISE. |  |
|  | 1. Open the LabDeploy.ps1 in the Deployment folder of the Lab contents. |  |
| The PowerShell file has descriptions on what each configuration variable represents. | 1. Review the configuration section and change values to desired values. |  |
| Note the following resources will be deployed to the Resource Group configured:   * Azure Storage account * Azure SQL Server * Azure Logic App * Office365 API connection   The following other changes will also be performed by the script:   * Create containers needed on the Azure Storage account * Upload txt, csv, and database backups to the Azure Storage account * Restore bacpacs from the Azure Storage account to the Azure SQL Server * Create Schema on the Azure SQL DW DB | 1. Click the Save icon. 2. Click the Run Script icon (play button) or hit F5 to run the script. Note that depending on internet connection this may take anywhere from 10-45 minutes to run. |  |
|  | 1. You will be prompted to enter your credentials to an account with an Azure subscription. 2. Fill in your Azure account email. Click Next. 3. Fill in your Azure account password. Click Sign-In. |  |
| 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. The PowerShell console output should update with various messages as the resources are deployed. When the database restore is complete you should receive a Lab Deployment Completed message. |  |

|  |  |  |
| --- | --- | --- |
| Part 2 – Verify Deployment and Configure Office365 API connection | | |
| **Scenario** | | |
| We are going to login to the Azure portal to verify our server resources deployed configure the Office365 API connection. | | |
| **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. |  |
| Note the SQL Server and Storage Account name use a 4-character hash of the resource ID at the end of the name to try to ensure a unique name. | 1. Verify you have the following resources: Logic App API Connection SQL Server SQL data warehouse SQL databases (OLTP and ODS) Storage Account |  |
| Now we are going to configure the O365 API connection.  Note this step is optional but highly recommended to receive emails in future lab modules. | 1. Click the API Connection name. 2. You should see the API Connection blade and a message that the connection is not authenticated. 3. Click the Edit API Connection menu option. |  |
|  | 1. Verify the Display Name is the email you want to authorize with Office 365 to send emails from the Logic App. 2. Click Authorize. |  |
|  | 1. You will be prompted to login with your Office365 account. 2. Click the proper account (or sign-in with Use another account). |  |
|  | 1. You should see a message that the authorization was successful. 2. Click the Save button. |  |

|  |  |  |
| --- | --- | --- |
| Part 3 – 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. For the lab we used adflab-adf. 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. |  |

|  |  |  |
| --- | --- | --- |
| Part 4.2 – Create an Application account(Optional) | | |
| **Scenario** | | |
| We are going to use the portal to create an application account that we can use to spin up an On-Demand HDInsight cluster in a future lab module. This application needs contributor access to your resource group and we will configure this. | | |
| **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 Azure Active Directory Icon in the left menu |  |
|  | 1. Click the App Registrations menu item in the Azure Active directory menu options. |  |
|  | 1. Click the +New application registration button. |  |
|  | 1. Fill out the name of the Create pane properties: Name: Choose a name for your application. In our case we chose adflab-app Sign-on URL: Even though we won’t be using this as a web application a URL is still required. In our case we used the URL of our corporate organization. 2. Click the Create button. |  |
|  | 1. When you return to the app registrations pane click on the name of your Application. 2. Note down the Application ID. This will be needed in Lab Module 5 as the service principal ID. |  |
|  | 1. Click the Settings button. 2. Click the Keys menu item. |  |
|  | 1. Under the Password sections you should see an input box for the Key description. For the description put in a descriptive key name such as HDInsight. Duration you can set to an appropriate time value. 2. Click Save. |  |
| This key will used as the service principal key in Lab Module 5. | 1. This will generate a Key under the value column. This key must be copied down to a secure location as you will not be able to return to it later. |  |
|  | 1. Click on the Resource Group menu in the left to view your resource groups. 2. Click on your Resource Group name created in Lab Module 1. |  |
|  | 1. Click on the Access control (IAM) menu item in the Resource Group blade. |  |
|  | 1. Click the +Add button. |  |
|  | 1. Fill out the following information in the Add Permissions pane: Role: Contributor Select: Search for or type in the name of the application you created in the previous steps. 2. Click Save. |  |
|  |  |  |

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