Microsoft Azure - Starter Kits for Partners

Hands on Lab

Intelligent Apps Scenario

Building Predictive Pipelines Incorporating Azure Data Lake and Azure Machine Learning

Last Update: Aug 2016





**MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.**

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The descriptions of other companies’ products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Microsoft. Microsoft cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

© 2016 Microsoft Corporation. All rights reserved. Any use or distribution of these materials without express authorization of Microsoft Corp. is strictly prohibited.

Microsoft and Windows are either registered trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

[Overview 4](#_Toc459905828)

[Scenario: 4](#_Toc459905829)

[Prerequisites 6](#_Toc459905830)

[Architecture 6](#_Toc459905831)

[Deployment instructions 7](#_Toc459905832)

[Exercise 1: Create Azure SQL Data Warehouse tables 7](#_Toc459905833)

[Exercise 2: Create the Azure Machine Learning service 7](#_Toc459905834)

[Exercise 3: Edit and start the Azure Stream Analytics job 8](#_Toc459905835)

[Exercise 4: Deploy the data generator as a Webjob 8](#_Toc459905836)

[Exercise 5: Upload U-SQL script to Azure Blob Storage 8](#_Toc459905837)

[Exercise 6: Create Azure Data Factory 8](#_Toc459905838)

[Exercise 7: Create the Power BI dashboard 8](#_Toc459905839)

[Summary 9](#_Toc459905840)

# Overview

The objective of this tutorial is to demonstrate techniques for the movement of data between an external data source, an Azure Data Lake Store and Azure SQL Data Warehouse while demonstrating the use of U-SQL for processing information in a Data Lake Store and performing advanced analytics using Azure Machine Learning (AML). This tutorial will be developed in reference to a use case described in the following section.

An Azure Data Lake Store is a flexible, scalable repository for any type of data. It provides unlimited storage with high frequency, low latency throughput capabilities and provides immediate read and analysis capabilities over your data. Once data is captured in the Data Lake, advanced transformation and processing of the data can be performed using Microsoft's extendable and scalable U-SQL language, integrated with Azure Data Lake Analytics, Azure Machine Learning, or any HDFS compliant project, such as Hive running in HD Insight cluster.

Some of the principal benefits of an Azure Data Lake Store include:

* Unlimited storage space
* High-throughput read/write
* Security through integration with Active Directory
* Automatic data replication
* Compatibility with the Hadoop Distributed File System (HDFS).
* Compatibility with HDFS compliant project (e.g. Hive, HBase, Storm, etc.)

**Estimated time** to complete this lab: **240 minutes**.

**Audience**: IT Pro, Architect, Application Owners and Developers

# Scenario:

Switch based telephone companies, both land line and cellular, produce very large volumes of information, principally in the form of call detail records. Each telecom switch records information on the calling and called numbers, incoming and outgoing trunks, and information of the time of the call along with a number of other features.

The duration that telephone companies keep their data has varied between land line and cellular companies from one to many years. Various legislation (e.g. the USA Freedom Act) is being considered that will require telecommunication companies to hold the data for a longer period of time. The amount of data can be extremely large. If you consider a modestly sized telecom carrier with 10M customers can readily produce over 1 billion description messages per day, including call detail records (CDR) at a size close to 1 TB per day. In shortly over ½ of a year this amount of data could begin to surpass the maximum capacity of an Azure Storage account (500TB).

In scenarios such as this, the integrated SQL and C# capabilities of U-SQL, the unlimited data storage capacity, and the ability for high velocity data ingestion of the Azure Data Lake Store makes it an ideal technical solution for the persistence and management of telephony call data.

Telecommunication network optimization techniques can hugely benefit from getting switch overload or malfunction predictions ahead of time. Such predictions help maintain SLA and overall network health by allowing for mitigating actions to be taken proactively, such as possibly rerouting calls and avoid call drops and perhaps an eventual switch shutdown. Microsoft’s capability to manage unlimited volumes of data within an Azure Data Lake Store combined with the powerful means for interacting with the Data Lake Store through U-SQL and the predictive modeling capabilities of Azure Machine Learning (AML) readily address all of the challenges with storage compliance and provide a seamless means for impactful analysis suitable for network optimization and other integration.

The intent of this tutorial is to provide the engineering steps necessary to capture and reproduce completely the scenario described above.

The tutorial will include:

* The generation and ingestion of CDR Data using an Azure Event Hub and Azure Streaming Analytics.
* The creation of an Azure Data Lake Store (ADLS) to meet long term CDR management requirements.
* Using Azure Data Lake Analytics (ADLA) and Microsoft’s U-SQL to interact with the Data Lake. The ADLA U-SQL job generates an aggregate view over the ingested CDR data that is stored in ADLS.
* Creation and integration of staging store for storing analytics results from U-SQL and predictions from Azure Machine Learning (AML). This staging store is implemented using Azure SQL Data Warehouse (SQL DW) and provides a backend for Power BI dashboards.
* AML model which predicts the switch overload

The focus of this tutorial is on the architecture, data transformation, and the movement of data between the different storage architectures and the Azure Machine Learning (AML) environment. While this example demonstrates techniques for integrating AML into the solution architecture, the focus is not on machine learning. Machine learning can be used in telecommunication industry for many purposes such as predictive maintenance, load and capacity forecast, failure prediction, effective marketing campaign, reducing infrastructure cost, etc.

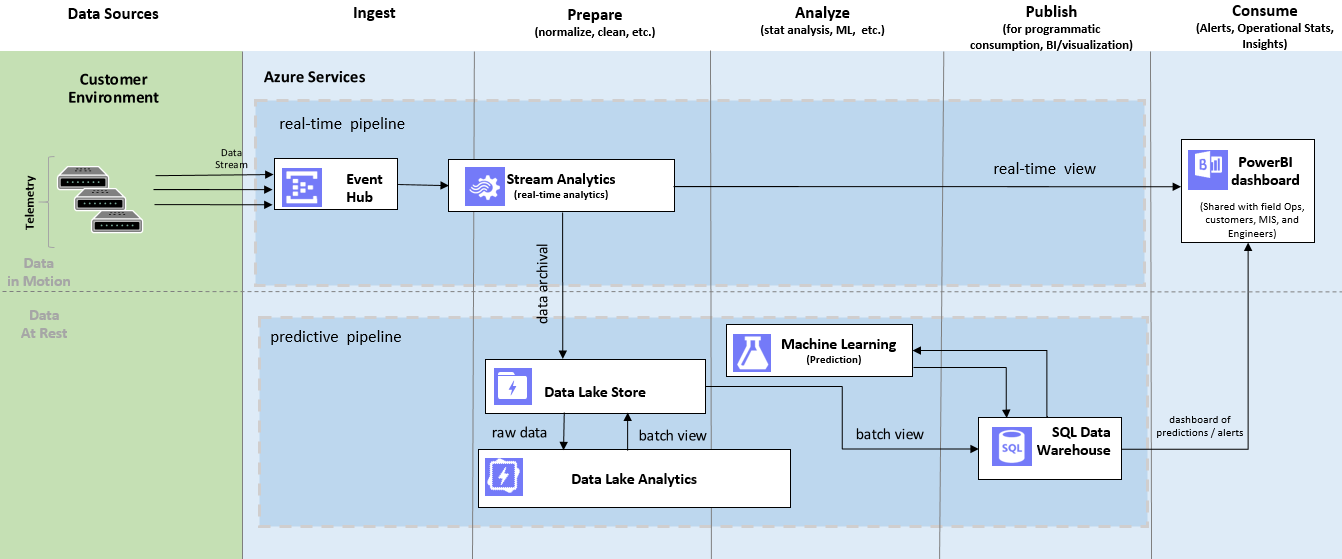
# Prerequisites

The steps described later in this tutorial requires the following prerequisites:

1. Azure subscription with login credentials (<https://azure.microsoft.com/en-us/>)
2. Azure Machine Learning Studio subscription (<https://azure.microsoft.com/en-us/services/machine-learning/>)
3. A Microsoft Power BI account (<https://powerbi.microsoft.com/en-us/>)
4. Power BI Desktop installation (<https://powerbi.microsoft.com/en-us/desktop/?gated=0&number=0>)
5. Microsoft Azure Storage Explorer (<http://storageexplorer.com/>)
6. A local installation of [Visual Studio with SQL Server Data Tools (SSDT)](https://azure.microsoft.com/en-us/documentation/articles/sql-data-warehouse-install-visual-studio/)

# Architecture

Figure 1 illustrates the Azure architecture developed in this sample.

[](https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/media/architecture.png)Figure 1: Architecture

Call detail record (CDR) data is generated via a data generator which simulates a phone switch and is deployed as an Azure Web Job. The CDR data is sent to an Event Hub. Azure Stream Analytics (ASA) takes in the CDR data flowed through Event Hub, processes the data by using ASA SQL and sends the processed data to a) Power BI for real time visualization and b) Azure Data Lake Store for storage. Azure Data Lake Analytics runs a U-SQL job to pre-process the data before sending it to SQL Data Warehouse (staging and publishing store) for Azure Machine Learning to run predictive analytics.

Predictive analytics is done by using the batch endpoint of an experiment published as a web service in the Azure Machine Learning Studio. The AML web service imports data (dropped call aggregates) from SQL Data Warehouse and exports the prediction, i.e. the scoring results back to SQL Data Warehouse. We use Azure Data Factory to orchestrate 1) U-SQL job in Azure Data Lake 2) Copy the results of the U-SQL job to SQL Data Warehouse 3) Apply predictive analytics in AML. The machine learning model here is used as an example experiment. You can use field knowledge and combine the available datasets to build more advanced model to meet your business requirements.

# Deployment instructions



This section describes the common instructions that will be applied to the deployment process, you need to following this instruction to create all Azure resources for this lab:

1. Service Bus,
2. Event Hub,
3. Stream Analytics Job
4. SQL Server, SQL Data Warehouse,
5. Azure Storage Account
6. Azure Data Lake Store Account
7. Azure Data Lake Analytics Account

**Instructions guide:** <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#deploy>

## Exercise 1: Create Azure SQL Data Warehouse tables

We need to create the matching tables in the SQL Data Warehouse. Follow this guide to create SQL Data Warehouse table:

Instructions guide: <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#create-azure-sql-data-warehouse-tables>

## Exercise 2: Create the Azure Machine Learning service

In this lab, Azure Machine Learning reads the data from Azure SQL Data Warehouse and sends the predictive results back to the Azure DW. Follow this guide to create the Azure Machine Learning service:

Instructions guide: <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#create-the-aml-service>

## Exercise 3: Edit and start the Azure Stream Analytics job

In this lab, Azure Stream Analytics reads data from Event Hub and run a per-minute aggregation job and send the result to the Power BI for visualization in real-time. Besides, Azure Stream Analytics store all the data to Azure Data Lake Store. Follow this guide to edit and start the Azure Stream Analytics job:

Instructions guide: <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#edit-and-start-the-asa-job>

## Exercise 4: Deploy the data generator as a Webjob

This lab uses a data generator to simulate the Call Detail Records coming from telephony switch and ingests the data to Azure Event Hub, follow this guide to deploy the data generator app as an Azure Webjob application:

Instructions guide: <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#deploy-the-data-generator-as-a-web-job>

## Exercise 5: Upload U-SQL script to Azure Blob Storage

In this lab, An Azure Data Lake Analytic will run a U-SQL job to generate device-based time aggregates. Follow this guide to upload the script to Azure Blob Storage:

Instructions guide: <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#upload-u-sql-script-to-azure-blob-storage>

## Exercise 6: Create Azure Data Factory

This lab uses Azure Data Factory to compose all the data storage, movement and analytics processing into a pipeline. Follow this guide to configure linked services, datasets and pipelines in the Azure Data Factory:

Instructions guide: <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#create-data-factory>

## Exercise 7: Create the Power BI dashboard

This lab uses Power BI dashboard to show real time data updates and to run queries on data in SQL Data Warehouse to generate advanced visualization. Follow this guide to create the real-time view and the predictive view in Power BI:

Instructions guide: <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#create-the-pbi-dashboard>

# Summary

In this lab, a sample application with real time and predictive pipelines is built to showcase the power of Azure Data Lake Store and its integration with Azure Machine Learning and many of the other Azure services.