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

[Introduction 5](#_Toc459305340)

[Prerequisites 7](#_Toc459305341)

[Architecture 7](#_Toc459305342)

[Deploy 8](#_Toc459305343)

[Deploy Multiple Resources, including: 8](#_Toc459305344)

[Exercise 1: Create Azure SQL Data Warehouse tables 11](#_Toc459305345)

[Exercise 2: Create the AML service 11](#_Toc459305346)

[Exercise 3: Edit and start the ASA job 12](#_Toc459305347)

[Exercise 4: Deploy the data generator as a Web Job 13](#_Toc459305348)

[Exercise 5: Upload U-SQL script to Azure Blob Storage 14](#_Toc459305349)

[Exercise 6: Create Data Factory 14](#_Toc459305350)

[Exercise 7: Create the PBI dashboard 19](#_Toc459305351)

[Summary 21](#_Toc459305352)

[Undeploy 21](#_Toc459305353)

[Appendix: 22](#_Toc459305354)

# Introduction

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

**Use Case**

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.

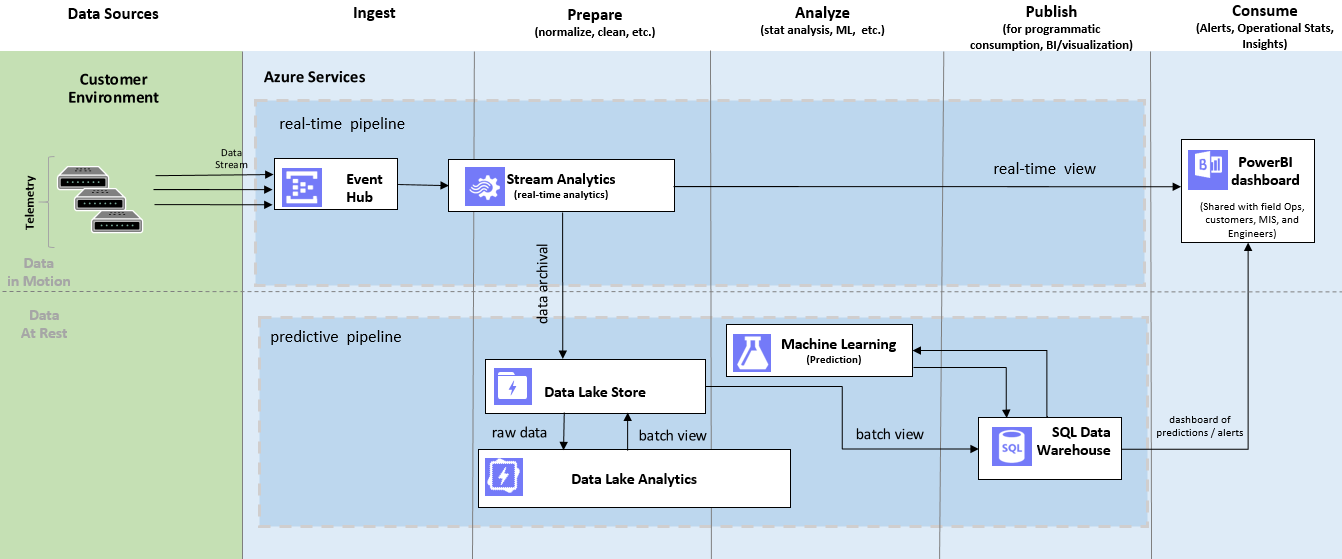
# Prerequisites

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.

# Deploy

Below are the steps to deploy the use case into your Azure subscription. Note that to condense the steps somewhat, > is used between repeated actions. For example:

Click: **Button A**

Click: **Button B**

is written as

Click: **Button A** > **Button B**

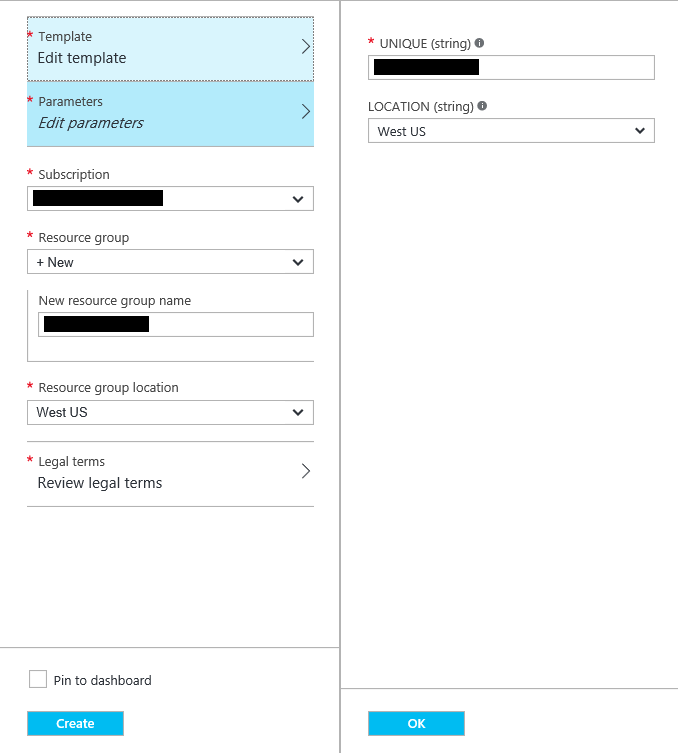
## Deploy Multiple Resources, including:

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

You will need a unique string to identify your deployment. We suggest you use only letters and numbers in this string and the length should not be greater than 9. Please open your memo file and write down "unique :[unique]" with "[unique]" replaced with your actual unique string. To get started, click the below button.

[https://camo.githubusercontent.com/9285dd3998997a0835869065bb15e5d500475034/687474703a2f2f617a7572656465706c6f792e6e65742f6465706c6f79627574746f6e2e706e67](https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure%2FCortana-Intelligence-Gallery-Content%2Fmaster%2FTutorials%2FData-Lake%2Fazuredeploy_part1.json)

This will create a new "blade" in the Azure portal([https://ms.portal.azure.com](https://ms.portal.azure.com/)).

[](https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/media/arm1.PNG)

1. Parameters
   1. Type: UNIQUE (string): **[*UNIQUE*]** (You need to select a globally unique string)
   2. Select: LOCATION: **[*LOCATION*]** (The region where everything will be deployed)
   3. Click: **OK**
2. Select: Subscription: **[*SUBSCRIPTION*]** (The Azure subscription you want to use)
3. Resource group
   1. Select: **New**
   2. Type: New resource group name: **[*UNIQUE*]** (Same as above)
4. Select: Resource group location: **[*LOCATION*]** (Same as above)
5. Click: **Review legal terms** > **Create**
6. Check: **Pin to dashboard** (If you want it on your dashboard)
7. Click: **Create**

The resource group will serve as an organizational framework for the associated Azure services.

After deployment, in the Azure Portal [https://ms.portal.azure.com](https://ms.portal.azure.com/) , select the “Resource Groups” option from the menu, use the **[*UNIQUE*]** to find the resource group you just created and you will find all the resources that have just been deployed. The following table lists some important account information and also information that you need to use in walking through this tutorial. Please note that '[unique]' should be replaced with your own unique string and '[' and ']' should not in your final information.

| **Item** | **Value** |
| --- | --- |
| service bus name space | adlservicebus[unique] |
| event hub name | adleventhub[unique] |
| stream analytic job output power bi | adlstreamanalytics[unique]]powerbi |
| stream analytic job output data lake | adlstreamanalytics[unique]]datalake |
| storage account name | storage[unique] |
| Data Lake Store Account Name | adls[unique] |
| Data Lake Analytic Account Name | adla[unique] |
| SQL Server name | adl-[unique].database.windows.net |
| SQL Server user name | adluser |
| SQL Server user password | pass@word1 |
| SQL Database Name | adlDB |

Table 1: Resources

## Exercise 1: Create Azure SQL Data Warehouse tables

Next you need to create the matching tables in the SQL Data Warehouse. You can do this by following these steps:

1. Start Visual Studio. Note that you must have installed the SQL Server Data Tools.
2. Select: View: **SQL Server Object Explorer**
3. Right click: **SQL Server**
4. Click: **Add SQL Server...**
5. Type: Server Name: **adl-[*UNIQUE*].database.windows.net**
6. Select: Authentication: **Sql Server Authentication**
7. Type: User name: **adluser**
8. Type: Password: **pass@word1** #This password is used in this tutorial as an example. You should avoid using it and other common passwords in production environment. Follow [Microsoft SQL Server Strong Password Requirements](https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/Microsoft%20SQL%20Server%20Strong%20Password%20Requirements) to create strong passwords.
9. Select: Database Name: **adlDB**
10. Click: **Connect**
11. Right click: **adllDB**
12. Select: **New Query...**
13. Copy and paste:

CREATE TABLE [dbo].[SwitchCallInfo] (

[Time] datetime NOT NULL,

[Switch] varchar(100) COLLATE Latin1\_General\_100\_CI\_AS\_KS\_WS NOT NULL,

[CallCount] bigint NOT NULL,

[CallFailure] bigint NOT NULL

)

WITH (CLUSTERED COLUMNSTORE INDEX, DISTRIBUTION = HASH([Time]));

CREATE TABLE [dbo].[ForcastCallFailure] (

[Time] datetime NOT NULL,

[CallFailure] bigint NOT NULL

)

WITH (CLUSTERED COLUMNSTORE INDEX, DISTRIBUTION = HASH([Time]));

1. Click: **Execute**

## Exercise 2: Create the AML service

1. Browse: <http://gallery.cortanaintelligence.com/Experiment/CDR-Call-Failure-Prediction-Azure-Data-Lake-1> # You will copy this experiment from the gallery
2. Click: **Open in Studio**
3. Select: REGION: **[*REGION*]** (Up to you)
4. Select: WORKSPACE: **[*WORKSPACE*]** (Your workspace)
5. Click: **Import Data**
6. Type: Database server name: **adl-[*UNIQUE*].database.windows.net**
7. Type: Password: **pass@word1**
8. Click: **Export Data**
9. Type: Database server name: **adl-[*UNIQUE*].database.windows.net**
10. Type: Server user account password: **pass@word1**
11. Click: **RUN** > **DEPLOY WEB SERVICE**

## Exercise 3: Edit and start the ASA job

Browse: [https://manage.windowsazure.com](https://manage.windowsazure.com/)

**To edit the input of the job that outputs to Power BI**

1. Click: **STREAM ANALYTICS** > **adlstreamanalytics[*unique*]powerbi**
2. Click: **INPUTS**> **cdreventhubinput** >
3. Type: EVENT HUB CONSUMER GROUP: **powerbi**
4. Click: **SAVE** > **Yes**

**To edit the output of the job that outputs to Power BI**

1. Click: **STREAM ANALYTICS** > **adlstreamanalytics[*unique*]powerbi**>**OUTPUTS**
2. Click: **DELETE** > **Yes**
3. Click: **ADD OUTPUT**
4. Select: **Power BI**
5. Click: **Next** > **Authorize Now** (Login with your credentials)
6. Type: OUTPUT ALIAS: **callinfoperminute**
7. Type: DATASET NAME: **callinfoperminute** (This dataset will be overwritten in PBI should it already exist)
8. Type: TABLE NAME: **callinfoperminute**
9. Select: WORKSPACE: **My Workspace** (Default)
10. Click: **Finish** > **Start** > **Finish** (You do not need to specify a custom time)

Browse: [https://manage.windowsazure.com](https://manage.windowsazure.com/)

**To edit the input of the job that outputs to Data Lake**

1. Click: **STREAM ANALYTICS** > **adlstreamanalytics[*unique*]datalake**
2. Click: **INPUTS**> **cdreventhubinput** >
3. Type: EVENT HUB CONSUMER GROUP: **datalake**
4. Click: **SAVE** > **Yes**

**To edit the output of the job that output to Data Lake**

1. Click: **STREAM ANALYTICS** > **adlstreamanalytics[*unique*]datalake**>**OUTPUTS**
2. Click: **DELETE** > **Yes**
3. Click: **ADD OUTPUT**
4. Select: **Data Lake Store**
5. Click: **Next** > **Authorize Now** (Login with your credentials)
6. Click: **Next**
7. Type: OUTPUT ALIAS: **datalakestoreoutput**
8. Select DATA LAKE STORE ACCOUNT: **adls[unique]**
9. Type: PATH PREFIX PATTERN: **/cdrdata/input/{date}/{time}**
10. Select DATE FORMAT: **YYYY/MM/DD**
11. Select TIME FORMAT: **HH**
12. Click: **Next**
13. Select OUTPUT Format: **CSV**
14. Click: **Finish** > **Start** > **Finish** (You do not need to specify a custom time)

## Exercise 4: Deploy the data generator as a Web Job

1. Download data generator: <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/datagenerator.zip>, Click raw and the file will be downloaded
2. Unzip: **datagenerator.zip**
3. Edit: **cdr-gen.exe.config**
4. Replace: EVENTHUBNAME: With: **adleventhub[*UNIQUE*]**
5. Get CONNECTION STRING
   1. Browse: [https://manage.windowsazure.com](https://manage.windowsazure.com/) (Get the endpoint)
   2. Click: SERVICE BUS
   3. Select: **adlservicebus[*UNIQUE*]**
   4. Click: CONNECTION INFORMATION
   5. Copy: CONNECTION STRING
6. Find: key Microsoft.ServiceBus.ConnectionString : replace its value With: CONNECTION STRING;TransportType=Amqp
7. Zip: **datagenerator.zip**
8. Browse: [https://manage.windowsazure.com](https://manage.windowsazure.com/)
9. Click: **NEW** > **COMPUTE** > **WEB APP** > **QUICK CREATE**
10. Type: URL: **adl[*UNIQUE*]**
11. Select: APP SERVICE PLAN: From your subscription
12. Click: **adl[*UNIQUE*]** > **WEBJOBS** > **ADD A JOB**
13. Type: NAME: **adl[*UNIQUE*]**
14. Browse: **datagenerator.zip**
15. Select: HOW TO RUN: **Run continuously**
16. Click: **Finish**

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

Download the script from <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/script/cdrSummary.txt>, and save it to a folder with name "script"

Download Microsoft Azure Storage Explorer, login with your credentials, and

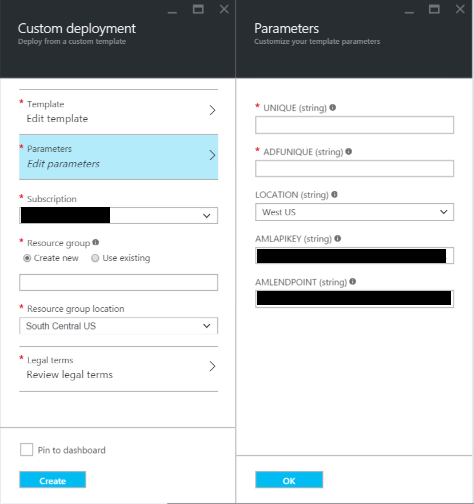
1. Select the storage account:**storage[*unique*]**
2. Right Click **"Create Blob container"**
3. Type: **cdrdata**
4. Right click **cdrdata**
5. Select **Open Blob Container Editor**
6. On the top of the right panel, Click **Upload**, Select **Upload Folder** and upload the script folder

## Exercise 6: Create Data Factory

To get started, click the below button.

[https://camo.githubusercontent.com/9285dd3998997a0835869065bb15e5d500475034/687474703a2f2f617a7572656465706c6f792e6e65742f6465706c6f79627574746f6e2e706e67](https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure%2FCortana-Intelligence-Gallery-Content%2Fmaster%2FTutorials%2FData-Lake%2Fazuredeploy_part2.json)

This will create a new "blade" in the Azure portal([https://ms.portal.azure.com](https://ms.portal.azure.com/)).

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

1. Parameters
   1. Type: UNIQUE (string): **[*UNIQUE*]** (Use the one previously entered)
   2. Type: ADFUNIQUE (string): **Azure Data Factory(ADF) Identifier** (Use a number)
   3. Select: LOCATION: **[*LOCATION*]** (Use the one previously selected)
   4. Type: AZUREMLAPIKEY: **[*AZUREMLAPIKEY*]**
      1. Browse: [https://studio.azureml.net](https://studio.azureml.net/)
      2. Click: **WEB SERVICES** > **CDR Call Failure Prediction (Azure Data Lake)**
      3. Click: Copy: **API key**
   5. Type: AZUREMLENDPOINT: **[*AZUREMLENDPOINT*]**
      1. Browse: [https://studio.azureml.net](https://studio.azureml.net/)
      2. Click: **WEB SERVICES** > **CDR Call Failure Prediction (Azure Data Lake)** > **BATCH EXECUTION**
      3. Copy: POST: **REQUEST URI** (Everything from "https" up to and including "jobs")
   6. Click: **OK**
2. Select: Subscription: **[*SUBSCRIPTION*]** (Use the one previously selected)
3. Select: Resource group: **[*UNIQUE*]** (Use the one previously selected)
4. Click: **Review legal terms** > **Create**
5. Check: **Pin to dashboard** (If you want it on your dashboard)
6. Click: **Create**

This will deploy a data factory with

* linked service: AzureStorageLinkedService, MLLinkedService
* dataset: AzureSqlDWAggregated,AzureSqlDWScored
* pipeline: MLPipeline

We need the following manual steps to complete the data factory.

**Add Azure Data Lake Store Linked Service**

1. Browse: [https://portal.azure.com](https://portal.azure.com/)
2. Click: **Data factories** > **dataFactory[*UNIQUE*][*ADFUNIQUE*]** > **Author and deploy**
3. Hover mouse over the icon, stop at **New Data Store**, Click *New Data Store*\*, Select "Azure Data Lake Store"
4. Copy

{

"name": "AzureDataLakeStoreLinkedService",

"properties": {

"type": "AzureDataLakeStore",

"description": "",

"typeProperties": {

"authorization": "<Click 'Authorize' to allow this data factory and the activities it runs to access this Data Lake Store with your access rights>",

"dataLakeStoreUri": "https://adls[unique].azuredatalakestore.net/webhdfs/v1",

"sessionId": "<OAuth session id from the OAuth authorization session. Each session id is unique and may only be used once>"

}

}

}

To the Editor, replace [unique] with your unique string

1. Click **Authorize**, input your credentials
2. Click: **Deploy**

**Add Azure Data Lake Analytic Linked Service**

1. Hover mouse over the icon, stop at **New Compute**, Click *New Compute*\*, Select "Azure Data Lake Analytics"
2. Copy

{

"name": "AzureDataLakeAnalyticsLinkedService",

"properties": {

"type": "AzureDataLakeAnalytics",

"description": "",

"typeProperties": {

"authorization": "<Authorization code is automatically retrieved after clicking 'Authorize' and completing the OAuth login>",

"accountName": "adla[unique]",

"sessionId": "<OAuth session id from the OAuth authorization session. Each session id is unique and may only be used once>"

}

}

}

To the Editor, replace [**unique**] with your unique string

1. Click **Authorize**, input your credentials
2. Click: **Deploy**

Wait until the upper two linked service are deployed by using the portal to check provision state.

**Add Azure Data Lake Data Sets**

1. Hover mouse over the icon, stop at **New Data Set**, Click **New Data Set**, Select "Azure Data Lake Store"
2. Copy the content from <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/dataset/DataLakeTable.json> into the Editor
3. Click: **Deploy**
4. Hover mouse over the icon, stop at **New Data Set**, Click **New Data Set**, Select "Azure Data Lake Store"
5. Copy the content from <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/dataset/DataLakeCDRAggregateTable.json> into the Editor
6. Click: **Deploy**

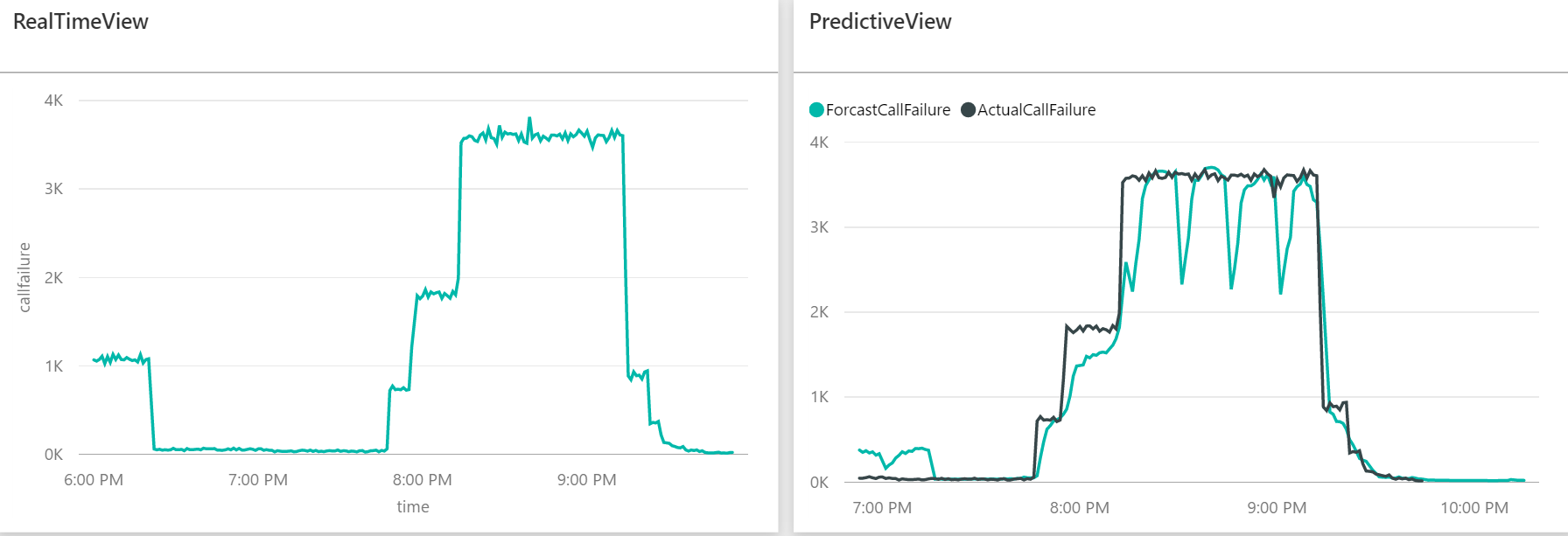
**Add Data Pipelines**

1. Hover mouse over the icon, stop at **New Pipeline**
2. Copy the content from <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/pipeline/DataLakeCDRSummary.json> into the Editor
3. Edit: start: **2016-05-12T00:00:00Z**: to: Your current time in UTC 24 hour clock (for example <http://www.timeanddate.com/worldclock/timezone/utc>)
4. Edit: end: **2016-05-12T00:00:00Z**: to: Your current time in UTC 24 hour clock plus three hours (for example <http://www.timeanddate.com/worldclock/timezone/utc>)
5. Edit: **"isPaused": true** : to **"isPaused": false**
6. Click: **Deploy**
7. Hover mouse over the icon, stop at **New Pipeline**
8. Copy the content from <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/pipeline/DataLakeStoreToSqlDW.json> into the Editor
9. Edit: start: **2016-05-12T00:00:00Z**: to: Your current time in UTC 24 hour clock (for example <http://www.timeanddate.com/worldclock/timezone/utc>)
10. Edit: end: **2016-05-12T00:00:00Z**: to: Your current time in UTC 24 hour clock plus three hours (for example <http://www.timeanddate.com/worldclock/timezone/utc>)
11. Edit: **"isPaused": true** : to **"isPaused": false**
12. Click: **Deploy**

**Start Machine Learning Pipeline**

1. Expand: **Pipelines**
2. Select: **MLPipeline**
3. Edit: start: **2016-05-12T00:00:00Z**: to: Your current time in UTC 24 hour clock (for example <http://www.timeanddate.com/worldclock/timezone/utc>)
4. Edit: end: **2016-05-12T00:00:00Z**: to: Your current time in UTC 24 hour clock plus three hours (for example <http://www.timeanddate.com/worldclock/timezone/utc>)
5. Edit: **"isPaused": true** : to **"isPaused": false**
6. Click: **Deploy**

## Exercise 7: Create the PBI dashboard

At the end of this section, you will have a dashboard which looks like the following: [](https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/media/result_dashboard.png)

**Realtime visualization**

1. Browse: [https://powerbi.microsoft.com](https://powerbi.microsoft.com/)
2. Click: **Sign in** (Login with your credentials)
3. Show: The navigation pane,
4. Scroll to the bottom to the section of **Datasets**
5. Click: **callinfoperminute** > **Line chart** Under Visualizations
6. Select: **Time**
7. Select: **CallFailure**
8. Click: the icon under Visualizations with tooltip **Format**
9. Click: **X-Axis**, Toggle **Title** to **On**
10. Click: **Y-Axis**, Toggle **Title** to **On**
11. Click: **Title**
12. Type **RealTimeView** to Title Text
13. Click **Pin visual** (pin icon on upper-right)
14. Type **RealTimeView** for the report name
15. Click **Save and Continue**
16. Select: **New dashboard**
17. Type: Name: **CallInfoDashBoard**
18. Click: **Pin**

**Predictive visualization**

**Power BI Desktop Part**

1. Download the Power BI Desktop application (<https://powerbi.microsoft.com/en-us/desktop>)
2. Download the Power BI template file <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/blob/master/Tutorials/Data-Lake/PowerBI/DataLakeCDRPredictive.pbix> (Click Raw to start downloading) and open it with Power BI application
3. On the application ribbon menu, choose Edit Queries
4. Go to *Query Settings* on the right pane, double click *Source*

In the SQL Server Database dialog

1. Type: Server Name: **adl-[*UNIQUE*].database.windows.net**
2. Type: Database Name: **adlDB**
3. Click: OK
4. Choose: **Database** for Authentication method
5. Input: **adluser** for username and **pass@word1** for password
6. On the application ribbon menu, click "Close and Apply"
7. Once data is loaded, On the application ribbon menu, click "Publish"
8. When prompt with dialog windows, click "Save"

**Power BI Website Part**

1. Browse: [https://powerbi.microsoft.com](https://powerbi.microsoft.com/)
2. Click: **Sign in** (Login with your credentials)
3. Show: The navigation pane,
4. Scroll to the bottom to the section of **Datasets**
5. Right Click: **DataLakeCDRPredictive**
6. Click: *Dataset Settings*
7. Click: Edit credentials
8. Input: *adluser* as user name and *pass@word1* as password
9. Click: **DataLakeCDRPredictive** > **Line chart** Under Visualizations
10. Select: **Time**
11. Select: **ForcastCallFailure**
12. Select: **ActualCallFailure**
13. Click: **X-Axis**, Toggle **Title** to **On**
14. Click: the icon under Visualizations with tooltip **Format**
15. Click: **Title**
16. Type **PredictiveView** to Title Text
17. Click **Pin visual** (pin icon on upper-right)
18. Type *PredictiveView* for the report name
19. Click *Save and Continue*\*
20. Select: **Existing dashboard**
21. Choose: Name: **CallInfoDashBoard**
22. Click: **Pin**

# Summary

Congratulations! If you made it to this point, you should have a running sample with real time and predictive pipelines showcasing the power of Azure Data Lake Store and its integration with Azure Machine Learning and many of the other Azure services. The next section lists the steps to tear things down when you are done.

# Undeploy

1. Delete Resources (Service Bus, Event Hub, SQL Data Warehouse, Data Factories)
   1. Browse: [https://portal.azure.com](https://portal.azure.com/)
   2. Click: **Resource groups**
   3. Right click: **[*UNIQUE*]** (your resource group)
   4. Select: **Delete**
2. Delete WebApp (data generator)
   1. Browse: [https://manage.windowsazure.com](https://manage.windowsazure.com/)
   2. Click: **WEB APPS**
   3. Select: **[*UNIQUE*]datagenerator** (Your web app)
   4. Click: **DELETE**
3. Delete AML Service
   1. Browse: [https://studio.azureml.net](https://studio.azureml.net/)
   2. Click: **WEB SERVICES**
   3. Select: **CDR Call Failure Prediction (Azure Data Lake)**
   4. Click: **DELETE** > **EXPERIMENTS**
   5. Select: **CDR Call Failure Prediction (Azure Data Lake)**
   6. Click: **DELETE**
4. Delete PBI dashboard
   1. Browse: [https://powerbi.microsoft.com](https://powerbi.microsoft.com/)
   2. Select: **Dashboards**
   3. Right click: **CallInfoDashBoard**
   4. Select: **REMOVE**
   5. Scroll to **Reports**
   6. Right click: **PredictiveView**
   7. Select: **REMOVE**
   8. Scroll to **Reports**
   9. Right click: **RealTimeView**
   10. Select: **REMOVE**
   11. Scroll to **Datasets**
   12. Right click: **DataLakeCDRPredictive**
   13. Select: **REMOVE**
   14. Scroll to **Datasets**
   15. Right click: **callinfoperminute**
   16. Select: **REMOVE**

# Appendix:

* 1. <https://github.com/Azure/Cortana-Intelligence-Gallery-Content/tree/master/Tutorials/Data-Lake#building-predictive-pipelines-incorporating-azure-data-lake-and-azure-machine-learning>