

# Technical guide to the Cortana Analytics Solution Template for Demand Forecast in Energy

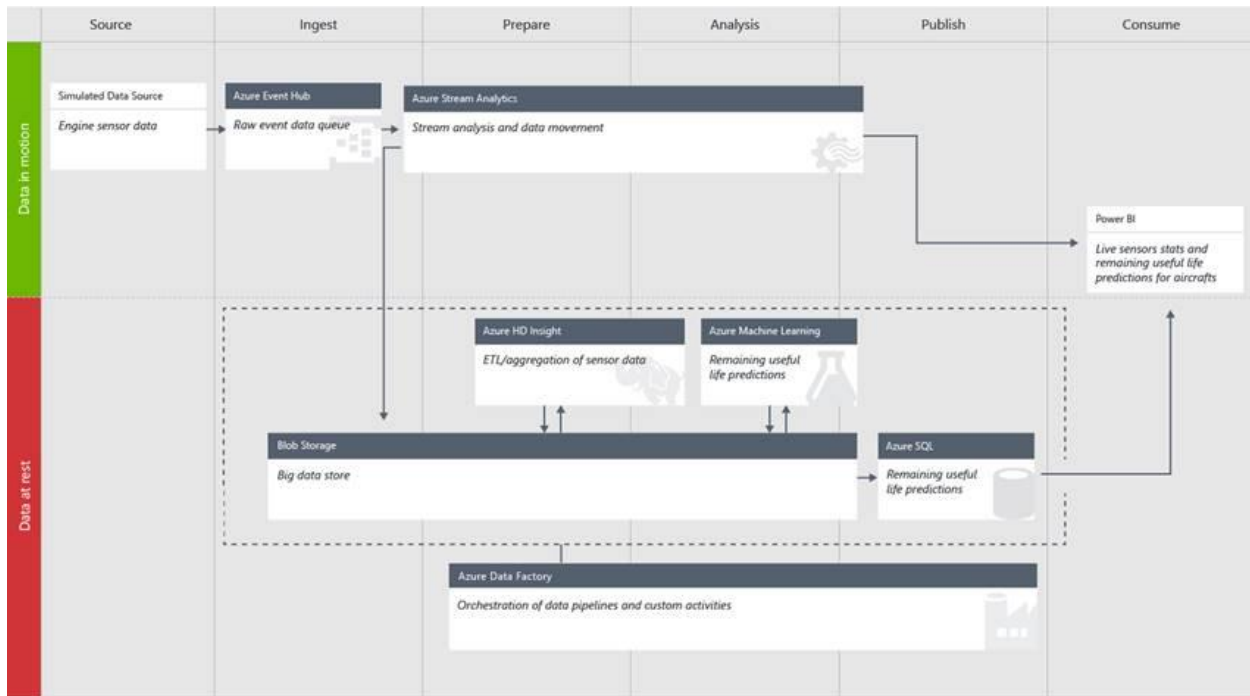
## Overview

*Solution Templates are designed to accelerate the process of building an E2E demo on top of Cortana Analytics Suite. A deployed template will provision your subscription with necessary Cortana Analytics component and build the relationships between. It also seeds the data pipeline with sample data getting generated from a data simulation application. Download the data simulator from the link provided and install it on your local machine, refer to the readme.txt file for instruction on using the simulator. Data generated from the simulator will hydrate the data pipeline and start generating machine learning prediction which can then be visualized on the PowerBI dashboard.*

*The deployment process will guide you through several steps to set up your solution credentials. Make sure you record these credentials such as solution name, username, and password you provide during the deployment.*

*The goal of this document is to explain the reference architecture and different components provisioned in your subscription as part of this Solution Template. The document also talks about how to replace the sample data, with real data of your own to be able to see insights/predictions from you won data. Additionally, the document talks about the parts of the Solution Template that would need to be modified if you want to customize the solution with your own data. Instructions on how to build the PowerBI dashboard for this Solution Template are provided at the end.*

## Big picture



## Architecture Explained

When the solution is deployed, various Azure services within Cortana Analytics Suite are activated (i.e. Event Hub, Stream Analytics, HDInsight, Data Factory, Machine Learning, etc.). The architecture diagram above shows, at a high level, how the Demand Forecasting for Energy Solution Template is constructed from end-to-end. You will be able to investigate these services by clicking on them on the solution template diagram created with the deployment of the solution. You can download a [full-size version of the diagram](#). The following sections describe each piece.

## Data Source and Ingestion

### Synthetic Data Source

For this template the data source used is generated from a desktop application that you will download and run locally after successful deployment. You will find the instructions to download and install this application in the properties bar when you select the first node called Demand Forecasting Data Simulator on the solution template diagram. This application feeds the [Azure Event Hub](#) service with data points, or events, that will be used in the rest of the solution flow.

The event generation application will populate the Azure Event Hub only while it is executing on your computer.

### Azure Event Hub

The [Azure Event Hub](#) service is the recipient of the input provided by the Synthetic Data Source described above.

## Data Preparation and Analysis

### Azure Stream Analytics

The [Azure Stream Analytics](#) service is used to provide near real-time analytics on the input stream from the [Azure Event Hub](#) service and publish results onto a [Power BI](#) dashboard as well as archiving all raw incoming events to [Azure Storage](#) service for later processing by the [Azure Data Factory](#) service.

### HD Insights Custom Aggregation

The [Azure HD Insights](#) service is used to run [Hive](#) scripts (orchestrated by Azure Data Factory) to provide aggregations on the raw events that were archived using the Azure Stream Analytics service.

### Azure Machine Learning

The [Azure Machine Learning](#) service is used (orchestrated by Azure Data Factory) to make forecast on future power consumption of a particular region given the inputs received.

## Data Publishing

### Azure SQL Database Service

The [Azure SQL Database](#) service is used to store (managed by Azure Data Factory) the predictions received by the Azure Machine Learning service that will be consumed in the [Power BI](#) dashboard.

## Data Consumption

### Power BI

The [Power BI](#) service is used to show a dashboard that contains aggregations provided by the [Azure Stream Analytics](#) service as well as demand forecast results stored in [Azure SQL Database](#) that were produced using the [Azure Machine Learning](#) service. For Instructions on how to build the PowerBI dashboard for this Solution Template, refer to the section below.

## How to bring in your own data

This section describes how to bring your own data to Azure, and what areas would require changes for the data you bring into this architecture.

It is unlikely that any dataset you bring would match the dataset used for this solution template. Understanding your data and the requirements will be crucial in how you modify this template to work with your own data. If this is your first exposure to the [Azure Machine Learning](#) service, you can get an introduction to it by using the example in [How to create your first experiment](#).

The following sections will discuss the sections of the template that will require modifications when a new dataset is introduced.

### Azure Event Hub

The [Azure Event Hub](#) service is very generic, such that data can be posted to the hub in either CSV or JSON format. No special processing occurs in the Azure Event Hub, but it is important you understand the data that is fed into it.


This document does not describe how to ingest your data, but one can easily send events or data to an Azure Event Hub, using the [Event Hub API](#).

### Azure Stream Analytics

The [Azure Stream Analytics](#) service is used to provide near real-time analytics by reading from data streams and outputting data to any number of sources.

For the Demand Forecasting for Energy Solution Template, the Azure Stream Analytics query consists of two sub-queries, each consuming events from the Azure Event Hub service as inputs and having outputs to two distinct locations. These outputs consist of one Power BI dataset and one Azure Storage location.

The [Azure Stream Analytics](#) query can be found by:

- Logging into the [Azure Portal](#)
- Locating the two stream analytics jobs  that were generated when the solution was deployed. One is for pushing data to blob storage (e.g. mytest1streaming432822asablob) and the other one is for pushing data to PowerBI (e.g.mytest1streaming432822asapbi).
- Selecting
  - o **INPUTS** to view the query input
  - o **QUERY** to view the query itself
  - o **OUTPUTS** to view the different outputs

Information about [Azure Stream Analytics](#) query construction can be found in the [Stream Analytics Query Reference](#) on [MSDN](#).

In this solution, the Azure Stream Analytics job which outputs dataset with near real-time analytics information about the incoming data stream to a Power BI dashboard is provided as part of this solution template. Because there's implicit knowledge about the incoming data format, these queries would need to be altered based on your data format.

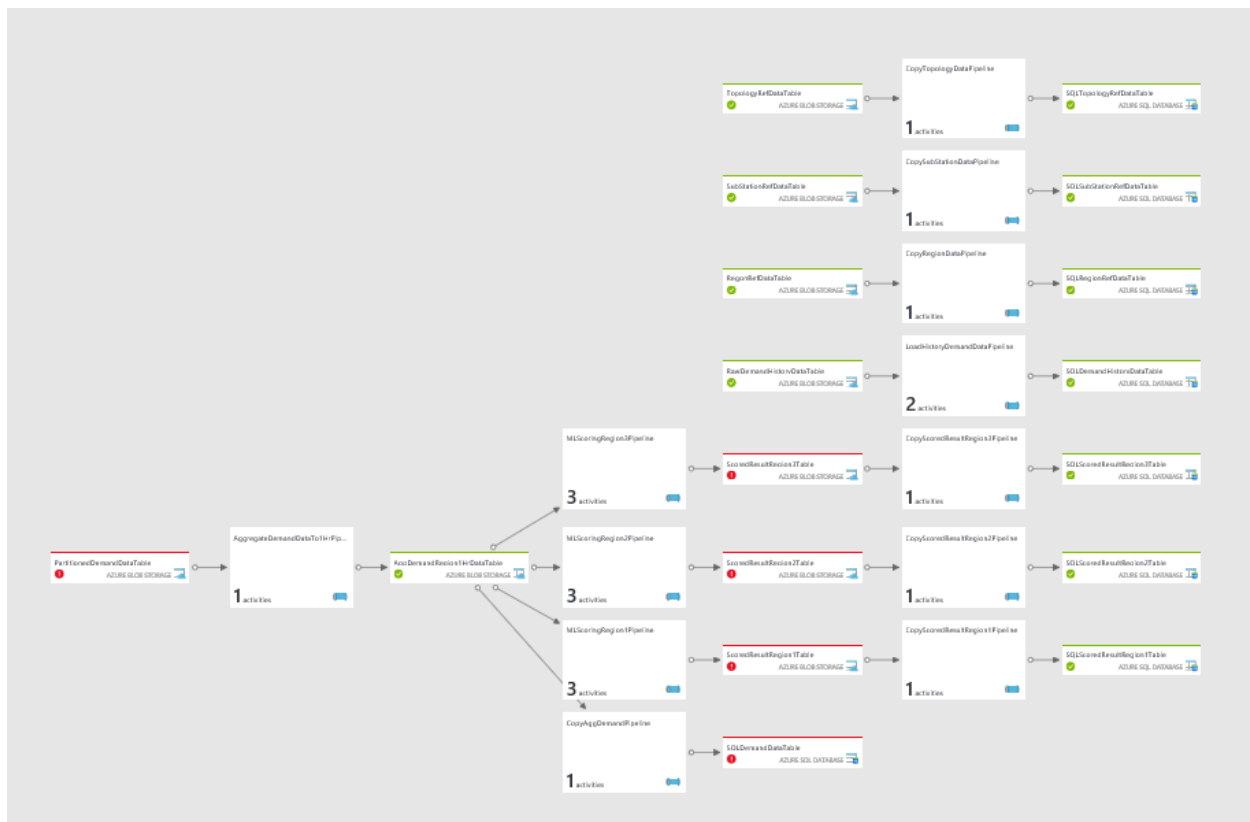
The other Azure Stream Analytics job outputs all [Event Hub](#) events to [Azure Storage](#) and hence requires no alteration regardless of your data format as the full event information is streamed to storage.

### Azure Data Factory

The [Azure Data Factory](#) service orchestrates the movement and processing of data. In the Demand Forecasting for Energy Solution Template the data factory is made up of twelve [pipelines](#) that move and process the data using various technologies.

You can access your data factory by opening the Data Factory node at the bottom of the solution template diagram created with the deployment of the solution. This will take you to the data factory on your Azure portal. If you see errors under your datasets, you can ignore those as they are due to data factory being deployed before the data generator was started. Those errors do not prevent your data factory from functioning.

This section discusses the necessary [pipelines](#) and [activities](#) contained in the [Azure Data Factory](#). Below is the diagram view of the solution.



Five of the pipelines of this factory contain [Hive](#) scripts that are used to partition and aggregate the data. When noted, the scripts will be located in the [Azure Storage](#) account created during setup. Their location will be: demandforecasting\\script\\hive\\ (or https://[Your solution name].blob.core.windows.net/ demandforecasting)

Similar to the [Azure Stream Analytics](#) queries, the [Hive](#) scripts have implicit knowledge about the incoming data format, these queries would need to be altered based on your data format and [feature engineering](#) requirements.

#### *AggregateDemandDataTo1HrPipeline*

This [pipeline](#) contains a single activity - an [HDInsightHive](#) activity using a [HDInsightLinkedService](#) that runs a [Hive](#) script to aggregate the every 10 seconds streamed in demand data in substation level to hourly region level and put in [Azure Storage](#) through the [Azure Stream Analytics](#) job.

The [Hive](#) script for this aggregation task is ***AggregateDemandRegion1Hr.hql***

#### *LoadHistoryDemandDataPipeline*

This [pipeline](#) contains two activities:

- [HDInsightHive](#) activity using a [HDInsightLinkedService](#) that runs a [Hive](#) script to aggregate the hourly history demand data in substation level to hourly region level and put in Azure Storage during the Azure Stream Analytics job
- [Copy](#) activity that moves the aggregated data from Azure Storage blob to the Azure SQL Database that was provisioned as part of the solution template installation.

The [Hive](#) script for this task is ***AggregateDemandHistoryRegion.hql***

#### *MLScoringRegionXPipeline*

These [pipelines](#) contain several activities and whose end result is the scored predictions from the [Azure Machine Learning](#) experiment associated with this solution template. They are almost identical except each of them only handles the different region which is being done by different RegionId passed in the ADF pipeline and the hive script for each region.

The activities contained in this are:

- [HDInsightHive](#) activity using a [HDInsightLinkedService](#) that runs a [Hive](#) script to perform aggregations and feature engineering necessary for the [Azure Machine Learning](#) experiment. The [Hive](#) scripts for this task are respective ***PrepareMLInputRegionX.hql***.
- [Copy](#) activity that moves the results from the [HDInsightHive](#) activity to a single [Azure Storage](#) blob that can be access by the [AzureMLBatchScoring](#) activity.
- [AzureMLBatchScoring](#) activity that calls the [Azure Machine Learning](#) experiment which results in the results being put in a single [Azure Storage](#) blob.

#### *CopyScoredResultRegionXPipeline*

These [pipelines](#) contain a single activity - a [Copy](#) activity that moves the results of the [Azure Machine Learning](#) experiment from the respective ***MLScoringRegionXPipeline*** to the [Azure SQL Database](#) that was provisioned as part of the solution template installation.

#### *CopyAggDemandPipeline*

This [pipeline](#) contain a single activity - a [Copy](#) activity that moves the aggregated ongoing demand data from ***LoadHistoryDemandDataPipeline*** to the [Azure SQL Database](#) that was provisioned as part of the solution template installation.

#### *CopyRegionDataPipeline, CopySubstationDataPipeline, CopyTopologyDataPipeline*

These [pipeline](#) contain a single activity - a [Copy](#) activity that moves the reference data of Region/Substation/Topologygeo that are uploaded to [Azure Storage](#) blob as part of the solution template installation to the [Azure SQL Database](#) that was provisioned as part of the solution template installation.

#### Azure Machine Learning

The [Azure Machine Learning](#) experiment used for this solution template provides the prediction of demand of region. The experiment is specific to the data set consumed and therefore will require modification or replacement specific to the data that is brought in.

#### Power BI Dashboard


##### Overview

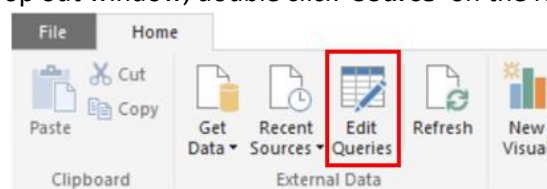
*This section describes how to set up Power BI dashboard to visualize your real time data from Azure stream analytics (hot path), as well as forecast results from Azure machine learning (cold path).*

## Setup Cold Path Dashboard

In cold path data pipeline, the essential goal is to get the demand forecast of each region. The forecast result is updated every 1 hour.

Power BI connects to an Azure SQL database as its data source, where the prediction results are stored. Please note: 1) Upon deploying your solution, real prediction will show up in the database about 1 hour. 2) In this step, the prerequisite is to download and install the free software [Power BI desktop](#).

1. Get the database credentials.
  - You will need **database server name, database name, user name and password** before moving to next steps. Here are the steps to guide you how to find them.
  - Record the solution name, username and password you provided during the deployment. Your database username and password are the same as the username and password previously recorded. Your database name is the same as your solution name (e.g. **mytest1db** for predictive maintenance solution) .
  - Once '**Azure SQL Database**' on your diagram turn into green, click it and then click '**Open**'.
  - You will see a new browser tab/window which shows the [Azure portal](#) page. Click '**Resource groups**' on the left panel.
  - Select the subscription you are using for deploying the solution, and then select '**YourSolutionName\_ResourceGroup**'.
  - In the new pop out panel, click  to access your database, and then you can find the **database server name**. (The database server name should look like **YourSoutionname.database.windows.net** )
  - Your database **username** and **password** are the same as the username and password previously recorded during deployment of the solution.
2. Update the data source of the cold path report file with [Power BI Desktop](#).
  - In the folder on your PC where you downloaded and unzipped the Generator file, double click the '**PowerBI\demoprediction.pbix**' file. Once you open it, on the top of the file, click '**Edit Queries**'. In the pop out window, double click '**Source**' on the right panel.





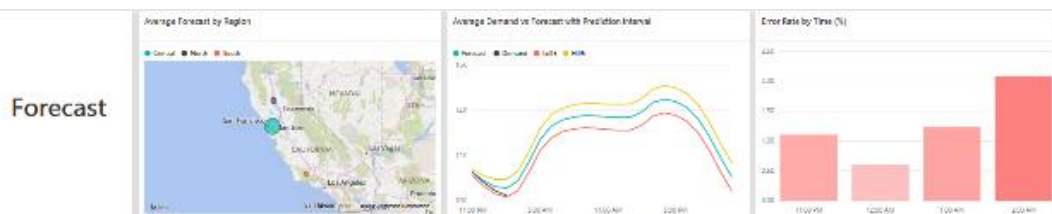
- In the pop out window, replace '**Server**' and '**Database**' with your own database credentials, click '**OK**'. In the server name please make sure you specify the port 1433 (YourSoutionname.database.windows.net, 1433). Ignore the warning messages that appear on the screen.
- In the next pop out window, you will see two options on the left pane (**Windows** and **Database**). Click '**Database**', fill in your '**Username**' and '**Password**'. (This is the username and password you entered when you first deployed the solution and created an Azure SQL database). In '*Select which level to apply these settings to*', check database level option. Then click '**Connect**'.

- Once you are guided back to the previous page, close the window. A message on the right will pop out and click **'Apply'**. Last, close the Power BI desktop file and click **'Save'** button to save the changes. Your PowerBI file has now established connection to the server. The visualizations should refresh to reflect new data now.


3. (Optional) Publish the cold path dashboard to [Power BI online](#).

Note that this step needs Power BI account (or [Office 365 account](#)).

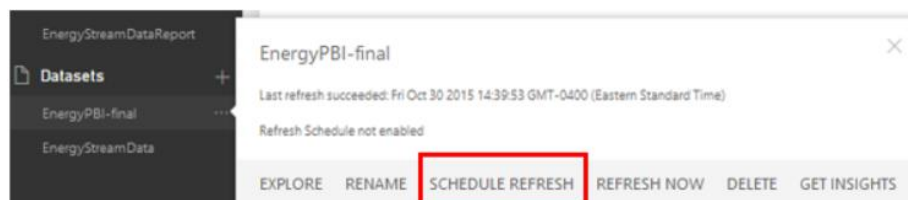
- Click **'Publish'** and few seconds later, a window appears and shows "Publishing to Power BI Success!" with a green check mark. Click the link below "Open demoprediction.pbix in Power BI". You can find detailed instructions [here](#).
- To create a new dashboard: click + sign next to the **Dashboards** section on the left pane to create a new dashboard. Enter the name "Predictive Maintenance Demo" for this new dashboard.
- Once you open the report, click  to pin all the visualizations to your dashboard. Go back to the dashboard page and adjust the size and location of your visualizations. You can find detailed instructions [here](#). This is the final view you will see.
- Once you open the report, click  to pin all the visualizations to your dashboard. To find detailed instructions, see [Pin a tile to a Power BI dashboard from a report](#). Go to the dashboard page and adjust the size and location of your visualizations and edit their titles. To find detailed instructions on how to edit your tiles, see [Edit a tile -- resize, move, rename, pin, delete, add hyperlink](#). Here is an example dashboard with some cold path visualizations pinned to it.



4. (Optional) Schedule refresh of the data source.

- Hover your mouse on the 'EnergyPBI-Final' dataset, click  and then choose **'Schedule Refresh'**.

Note: If you see a warning message, please click **'Edit Credentials'** and make sure your database credentials are the same as those described in step 1.




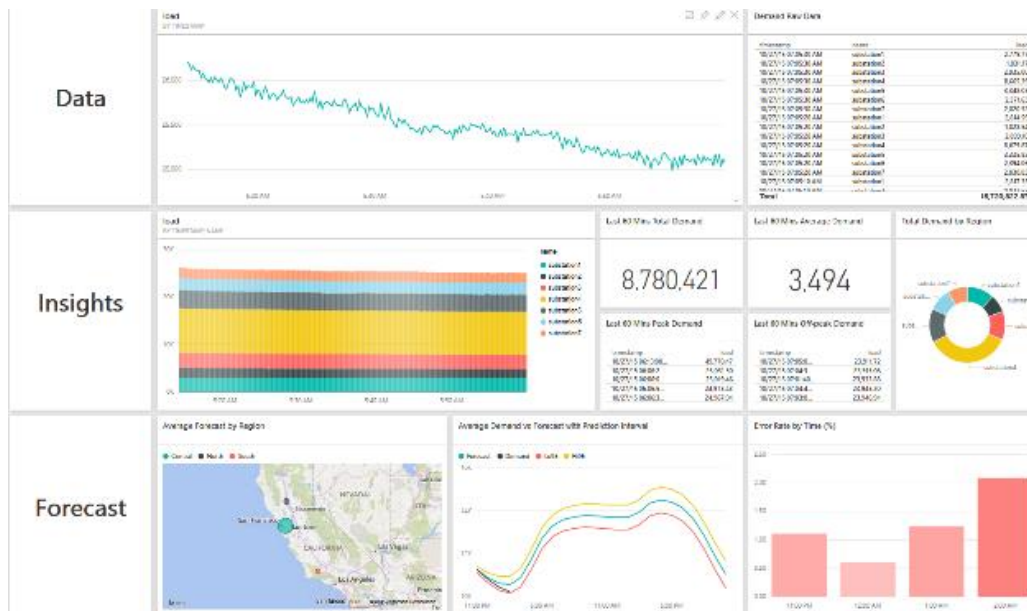
- Expand 'Schedule Refresh' session. Turn on 'keep your data up-to-date'.
- Schedule the refresh based on your needs. To find more information, see [Data refresh in Power BI](#).



## Setup Hot Path Dashboard

The following steps will guide you how to visualize real time data output from Stream Analytics jobs that were generated at the time of solution deployment. A [Power BI online](#) account is required to perform the following steps. If you don't have an account, you can [create one](#).

1. Add Power BI output in Azure Stream Analytics (ASA).
  - You will need to follow the instructions in [Azure Stream Analytics & Power BI: A real-time analytics dashboard for real-time visibility of streaming data](#) to set up the output of your Azure Stream Analytics job as your Power BI dashboard.
  - Locate the stream analytics job **energyforecastasapbi** in the [Azure Portal](#).
  - Setup the output of the ASA query which is **PBIoutput**. Make sure the **Output Alias** is the same as in your query. You can name your **Dataset Name** and **Table Name** as '**EnergyStreamData**'. Once you have added all three output tables and started the Stream Analytics job, you should get a confirmation message (e.g., "Starting stream analytics job energyforecastasapbi succeeded").
2. Log in [Power BI online](#)
  - On the left panel Datasets section in My Workspace, you should be able to see a new dataset showing on the left panel of PowerBI. This is the streaming data you pushed from Azure Stream Analytics in the previous step.
  - Make sure the **Visualizations** pane is open and is shown on the right side of the screen.
3. Create the "Demand by Timestamp" tile:
  - Click dataset 'EnergyStreamData' on the left panel Datasets section.
  - Click "Line Chart" icon .
  - Click 'EnergyStreamData' in **Fields** panel.
  - Click "Timestamp" and "Load" so that they both shows under "Values".
  - Click **SAVE** on the top and name the report as "EnergyStreamDataReport". The report named "EnergyStreamDataReport" will be shown in **Reports** section in the **Navigator** pane on left.
  - Click "**Pin Visual**" icon on top right corner of this line chart, a "Pin to Dashboard" window may show up for you to choose a dashboard. Please select "EnergyStreamDataReport", then click "Pin".
  - Hover the mouse over this tile on the dashboard, click "edit" icon on top right corner to change its title as "Demand by Timestamp"
4. Create other dashboard tiles based on appropriate datasets. The final dashboard view is shown below.



## Cost Estimation Tools

There are two tools to help you better understand the total costs involved in running the Demand Forecasting for Energy Solution Template in your subscription

- [Microsoft Azure Cost Estimator Tool \(online\)](#)
- [Microsoft Azure Cost Estimator Tool \(desktop\)](#)