Introduction to IoT development based on Azure infrastructure.

Project overview.

This project covers such topics as device simulation, connectivity, storage, processing and visualization. After completing the project the student will be able to use such Azure resources as IoT hub, Azure Stream Analytics, Azure SQL database, Power BI and basics of Azure IoT hub SDK.

Project's tasks.

The main aim of this project is to develop an end to end system starting from data collection and up to data utilization, be it either visualizations using Power Bi or analytics with Azure Machine Learning studio (optional).

The project is advised to be completed in the following order:

1. Create and configure IoT hub instance.
2. Set up simulated device to send telemetry to IoT hub.
3. Create and configure SQL database to store telemetry data from IoT hub.
4. Create and configure Stream Analytics job to read data going to IoT hub and save them to SQL database.
5. Connect Power Bi to SQL to visualize collected data.

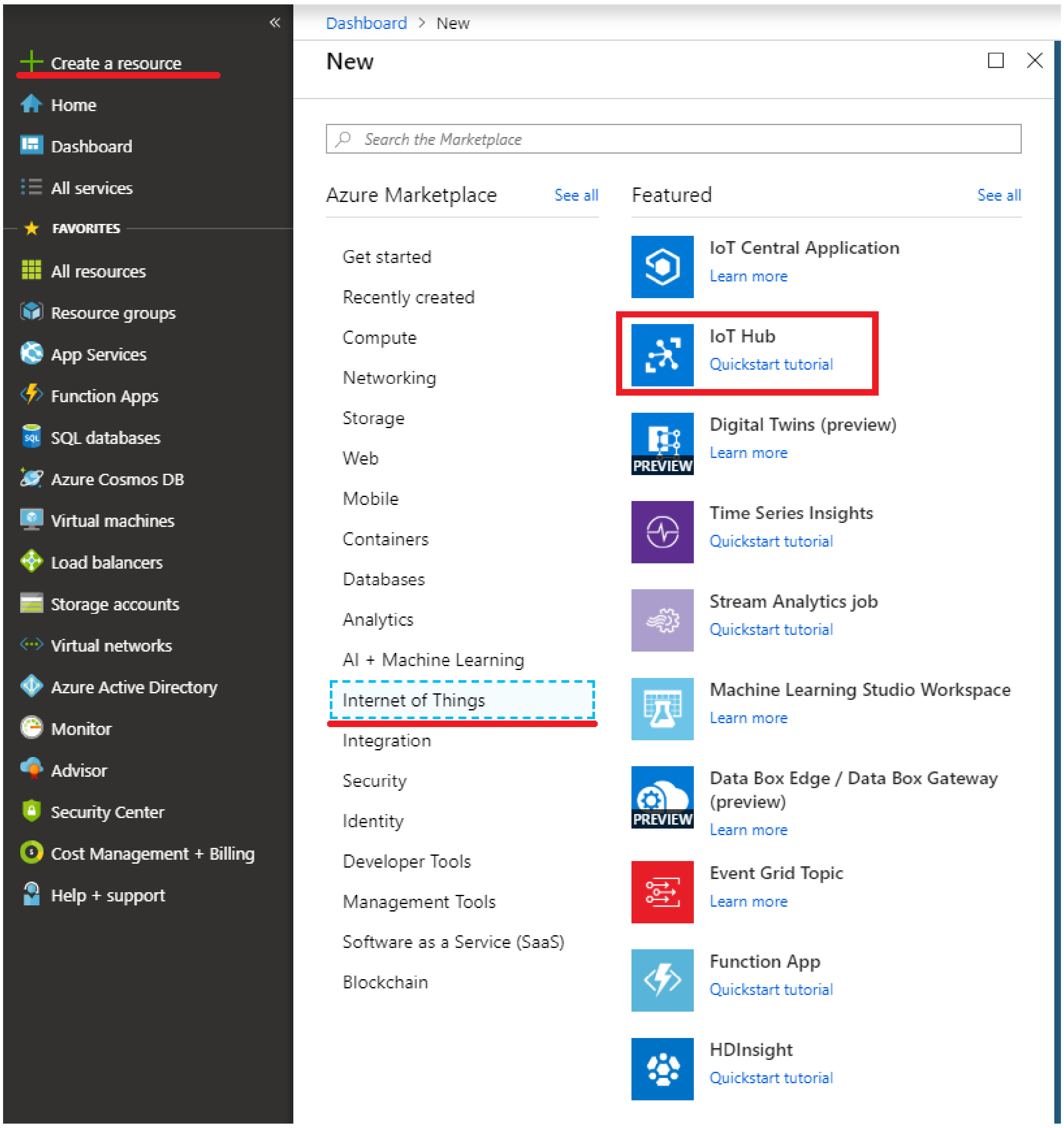
1. Create and configure IoT Hub instance.

IoT hub is an Azure platform used for managing big numbers of IoT assets, allowing to manage connections, communication and routing. IoT Hub does not provide storage or processing power by itself, it is only used as an entry point for all the data.

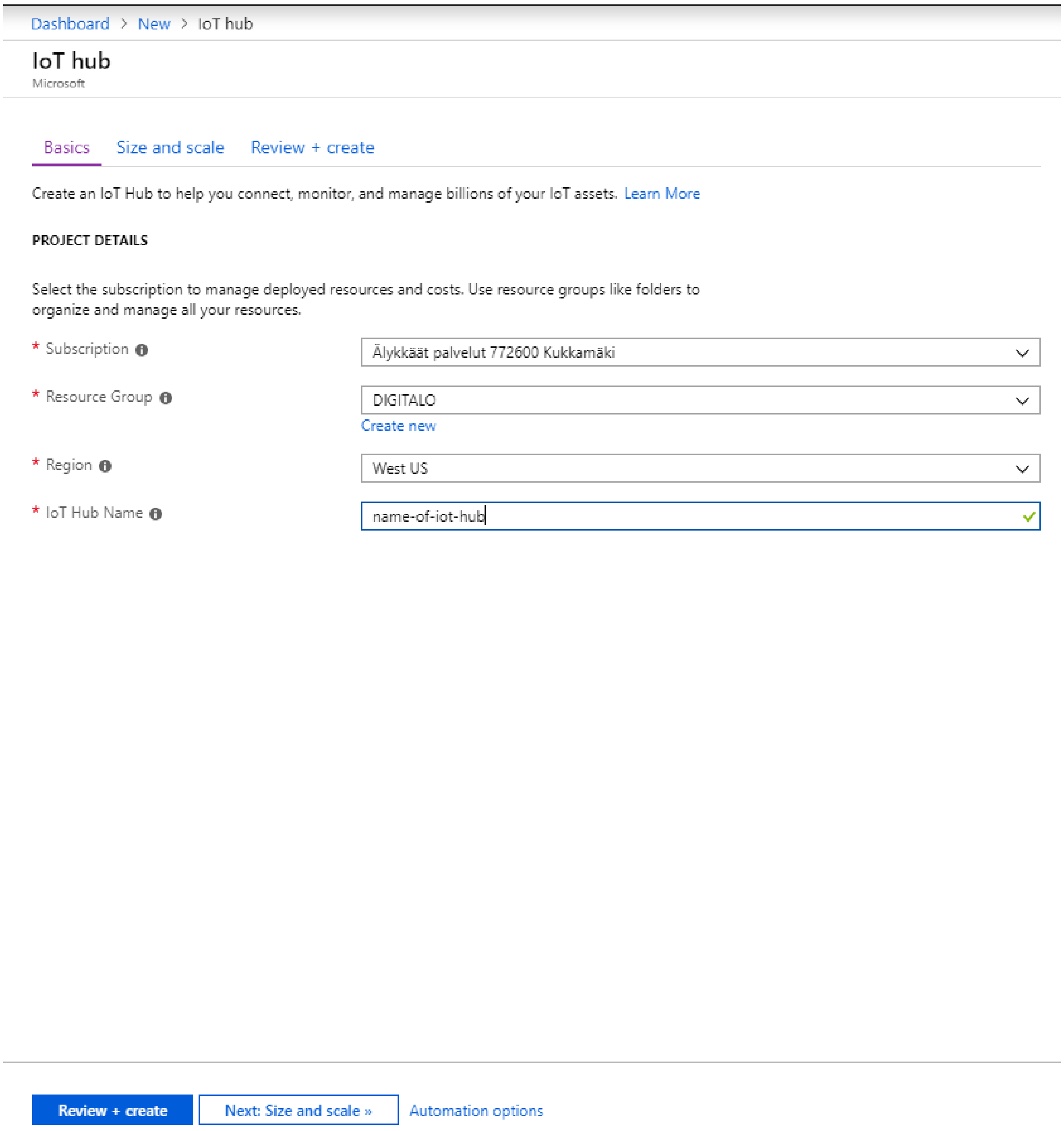
The main reason to use IoT Hub is scalability. It is perfectly fine to avoid using it for a project of a small case (e.g. this project does not really requires IoT hub), but when the number of the devices and messages increases there is no way to avoid it.

Instruction.

1. Create a new instance of IoT Hub.
   Start by creating a new resource and selecting IoT Hub in "Internet of Things" folder.



2. Continue with configuring new IoT Hub properties. Select the subscription (you should be provided with one for the course), resource group (should be provided by the course instructor), region (North Europe, but anything is fine) and the name of your choice. Your values can be and will be different from those of the screenshot.


Dashboard > New > IoT hub

IoT hub
Microsoft

Basics | Size and scale | Review + create

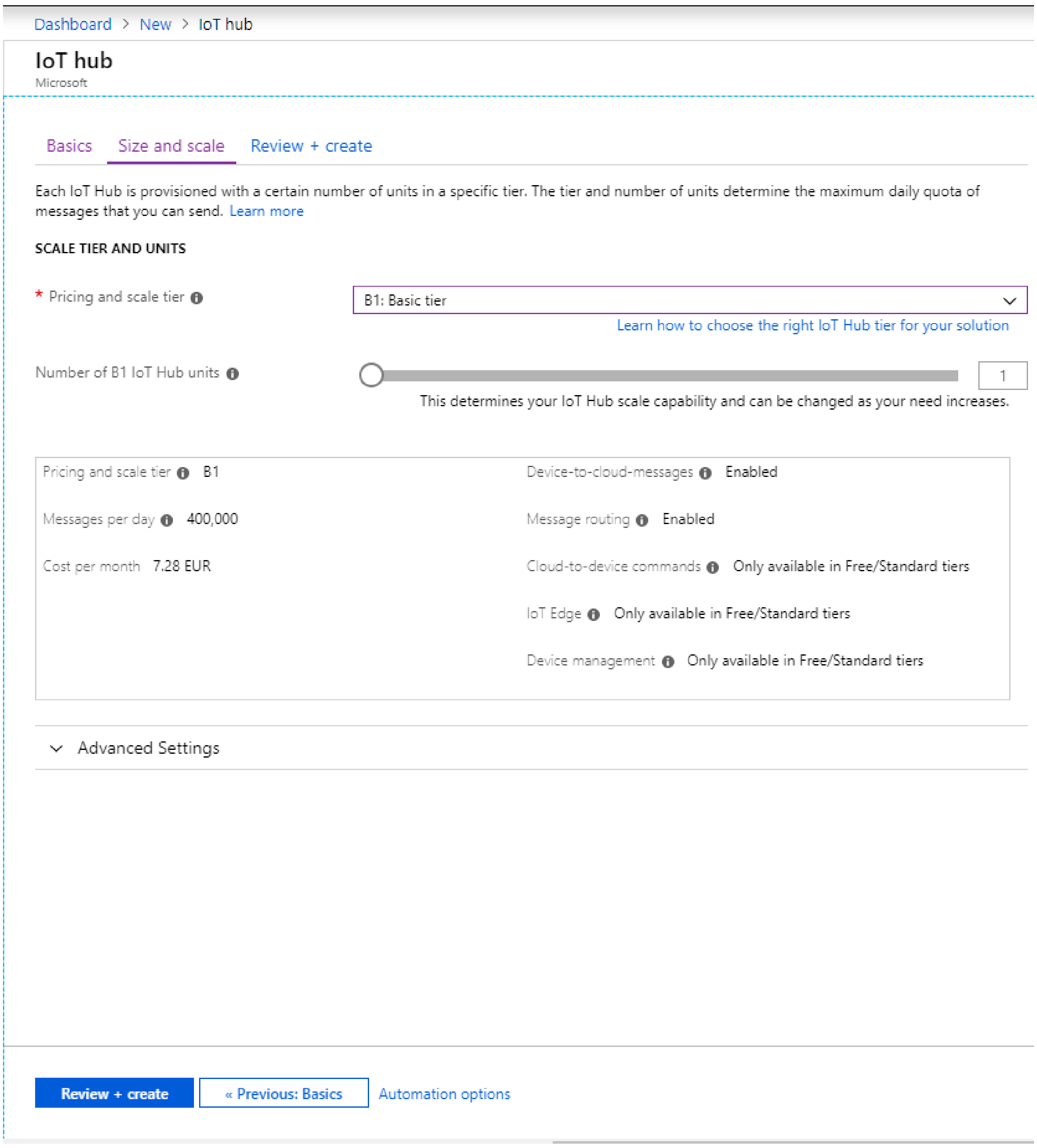
Create an IoT Hub to help you connect, monitor, and manage billions of your IoT assets. Learn More

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription	Älykkäät palvelut 772600 Kukkamäki
* Resource Group	DIGITALO
	Create new
* Region	West US
* IoT Hub Name	name-of-iot-hub

3. Next step is to configure the size and scale. In “Pricing and scale tier” select “B1: Basic tier”, this setting determines amount of handled messages and provided processing power and it is the main factor in the price of the resource. Set the number of “IoT Hub units” to 1. These settings are enough for 400000 messages a day.


Dashboard > New > IoT hub

IoT hub
Microsoft

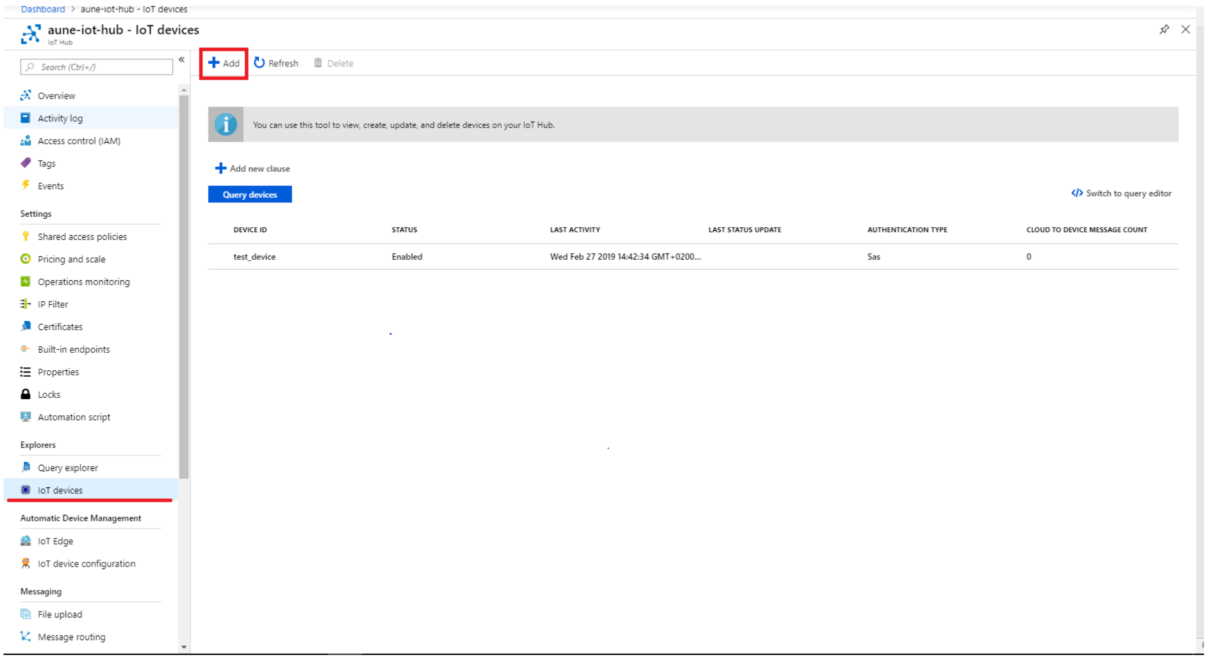
Each IoT Hub is provisioned with a certain number of units in a specific tier. The tier and number of units determine the maximum daily quota of messages that you can send. Learn more

SCALE TIER AND UNITS

* Pricing and scale tier	B1: Basic tier
	Learn how to choose the right IoT Hub tier for your solution
Number of B1 IoT Hub units	1
	This determines your IoT Hub scale capability and can be changed as your need increases.

Pricing and scale tier	B1	Device-to-cloud-messages	Enabled
Messages per day	400,000	Message routing	Enabled
Cost per month	7.28 EUR	Cloud-to-device commands	Only available in Free/Standard tiers
		IoT Edge	Only available in Free/Standard tiers
		Device management	Only available in Free/Standard tiers

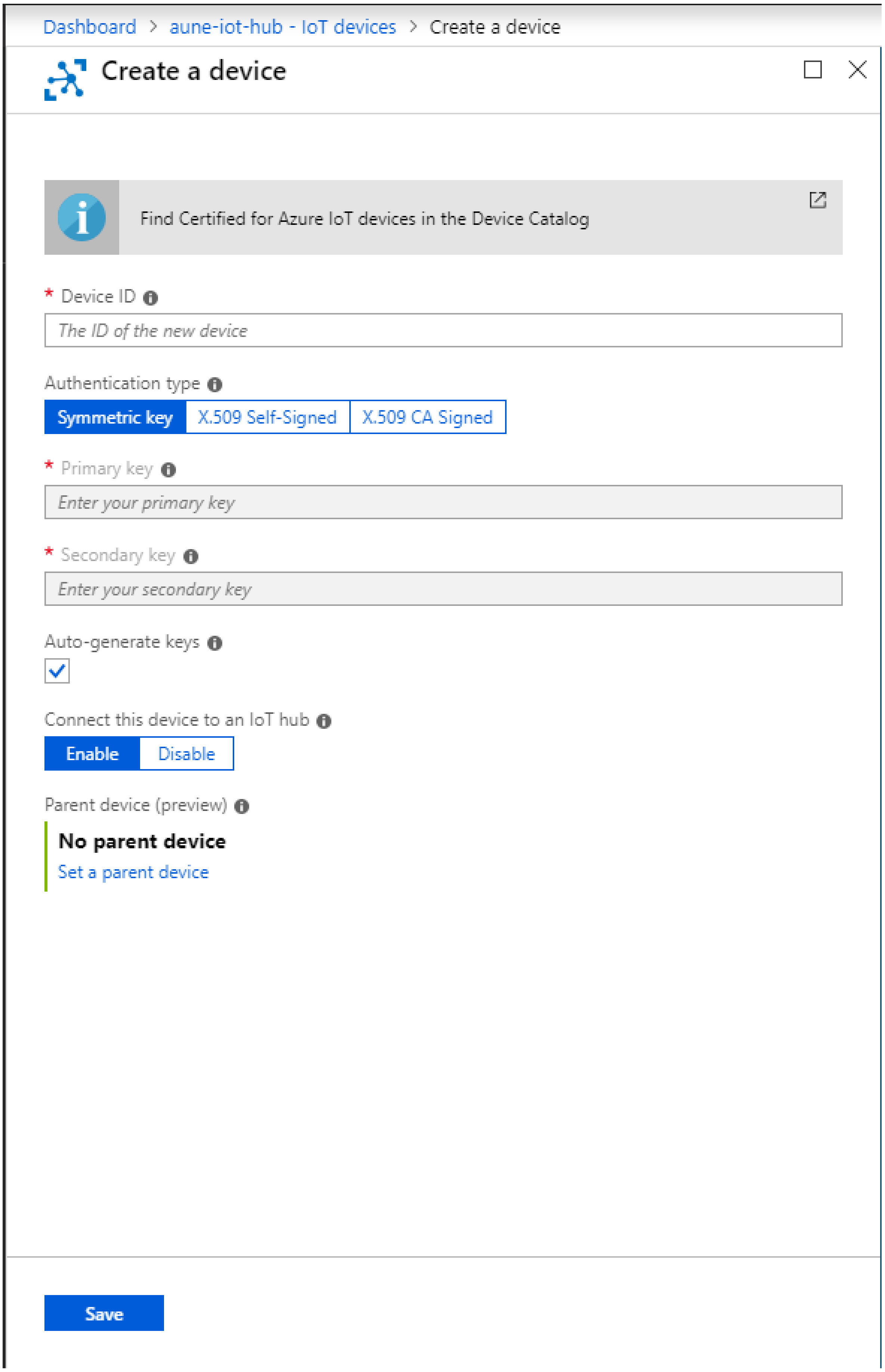
4. Review the settings and confirm. The creating of IoT Hub instance can take some time, you will be notified about the completion.
5. Open IoT Hub and create new IoT device.



You can use this tool to view, create, update, and delete devices on your IoT Hub.

DEVICE ID	STATUS	LAST ACTIVITY	LAST STATUS UPDATE	AUTHENTICATION TYPE	CLOUD TO DEVICE MESSAGE COUNT
test_device	Enabled	Wed Feb 27 2019 14:42:34 GMT+0200...		Sas	0

In a Create a device window provide “Device ID” (name of your choice), “Authentication type” to Symmetric key and “Auto-generate keys” to true. Don’t forget to check that the device is enabled.



Dashboard > aune-iot-hub - IoT devices > Create a device

Create a device

Find Certified for Azure IoT devices in the Device Catalog

* Device ID

The ID of the new device

Authentication type

[✓ Symmetric key] X.509 Self-Signed X.509 CA Signed

* Primary key

Enter your primary key

* Secondary key

Enter your secondary key

Auto-generate keys

✓

Connect this device to an IoT hub

[✓ Enable] Disable

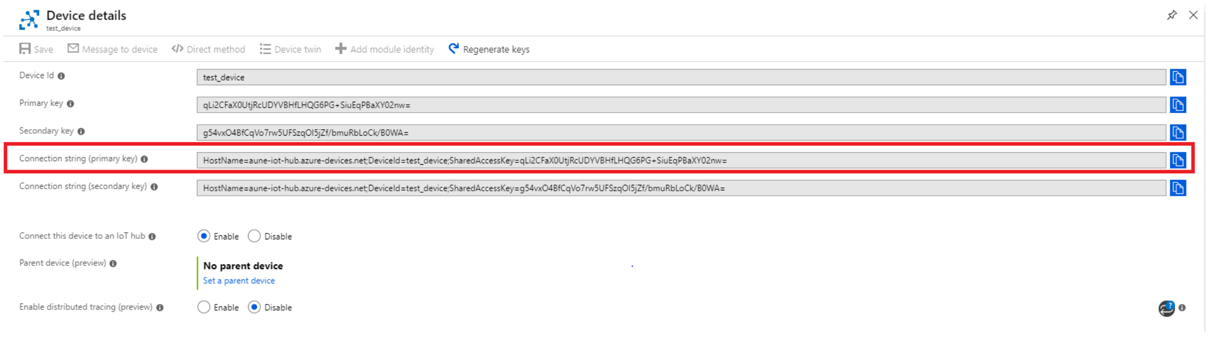
Parent device (preview)

No parent device

Set a parent device

Save

6. Get “Connection string” for the device.
   Click on the newly created IoT Device and take a note of “Connection string (primary key)”. This string will be needed to connect simulated device to the IoT Hub.



Device details
test_device

Device Id	test_device
Primary key	qLi2CFaX0UtjRcUDYVBHfLHQG6PG+SiuEqPBaXY02nw=
Secondary key	g54vxO4BfCqVo7rw5UFSzqOI5jZf/bmuRbLoCk/B0WA=
Connection string (primary key)	HostName=aune-iot-hub.azure-devices.net;DeviceId=test_device;SharedAccessKey=qLi2CFaX0UtjRcUDYVBHfLHQG6PG+SiuEqPBaXY02nw=
Connection string (secondary key)	HostName=aune-iot-hub.azure-devices.net;DeviceId=test_device;SharedAccessKey=g54vxO4BfCqVo7rw5UFSzqOI5jZf/bmuRbLoCk/B0WA=
Connect this device to an IoT hub	[✓ Enable] Disable
Parent device (preview)	No parent device Set a parent device
Enable distributed tracing (preview)	Enable [✓ Disable]

2. Set up a simulated device.

This project uses a program running on your computer to simulate real device sending temperature and humidity. The program requires node.js installed on the computer.

Instruction.

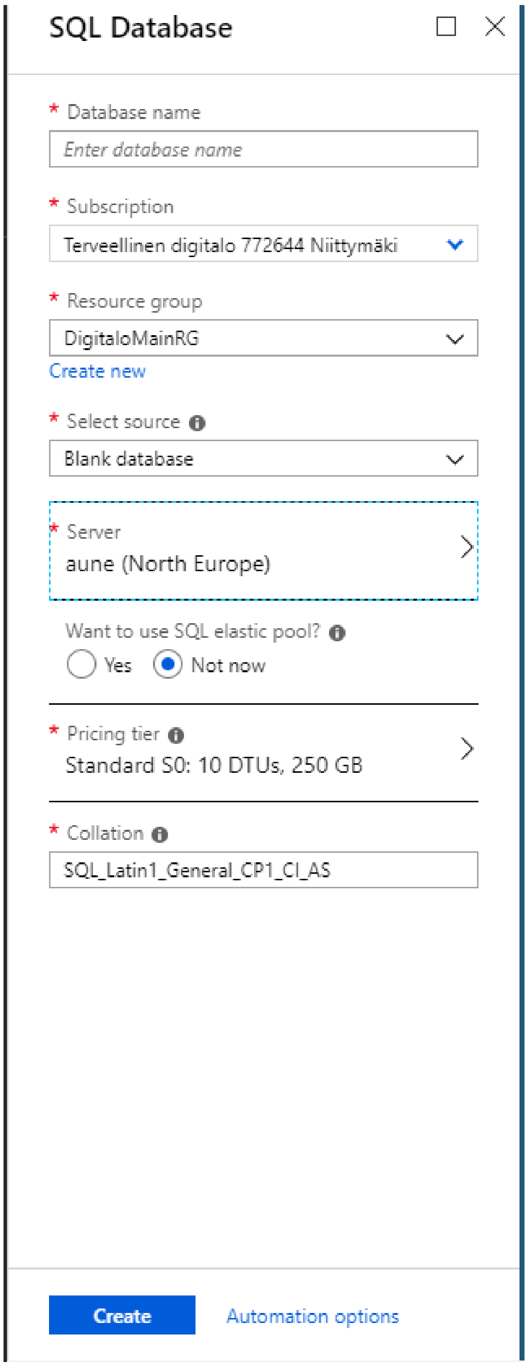
1. Download the sample code from https://github.com/Azure-Samples/azure-iot-samples-node/archive/master.zip.
2. Unzip the folder.
3. Navigate to “iot-hub\Quickstarts\simulated-device”.
4. Open SimulatedDevice.js in an editor of your choice and change **connectionString** variable to the one assigned to the previously created device in IoT hub (from step 6 in IoT Hub creation.
5. In the local terminal (open the folder from previous step in cmd.exe or powershell if you are on Windows or in bush if you are on Linux-based OS) type following commands:
   npm install
   node SimulatedDevice.js
6. The program now sends telemetry to the IoT hub. In case some questions occur , you can refer to the following manual: https://docs.microsoft.com/en-us/azure/iot-hub/quickstart-send-telemetry-node

3. Create and set up Azure SQL database.

This project uses Azure SQL database for storing telemetry from the simulated device. SQL databases are not a perfect tools for time series types of data (and telemetry is a time series data), however it works fine to demonstrate how to route and store data in Azure.

Instruction.

1. Create SQL Database.
2. Select name, subscription, resource group, pricing tier (basic tier with 1 DTU).



SQL Database

* Database name: Enter database name

* Subscription: Terveellinen digitalo 772644 Niittymäki

* Resource group: DigitaloMainRG

Create new

* Select source: Blank database

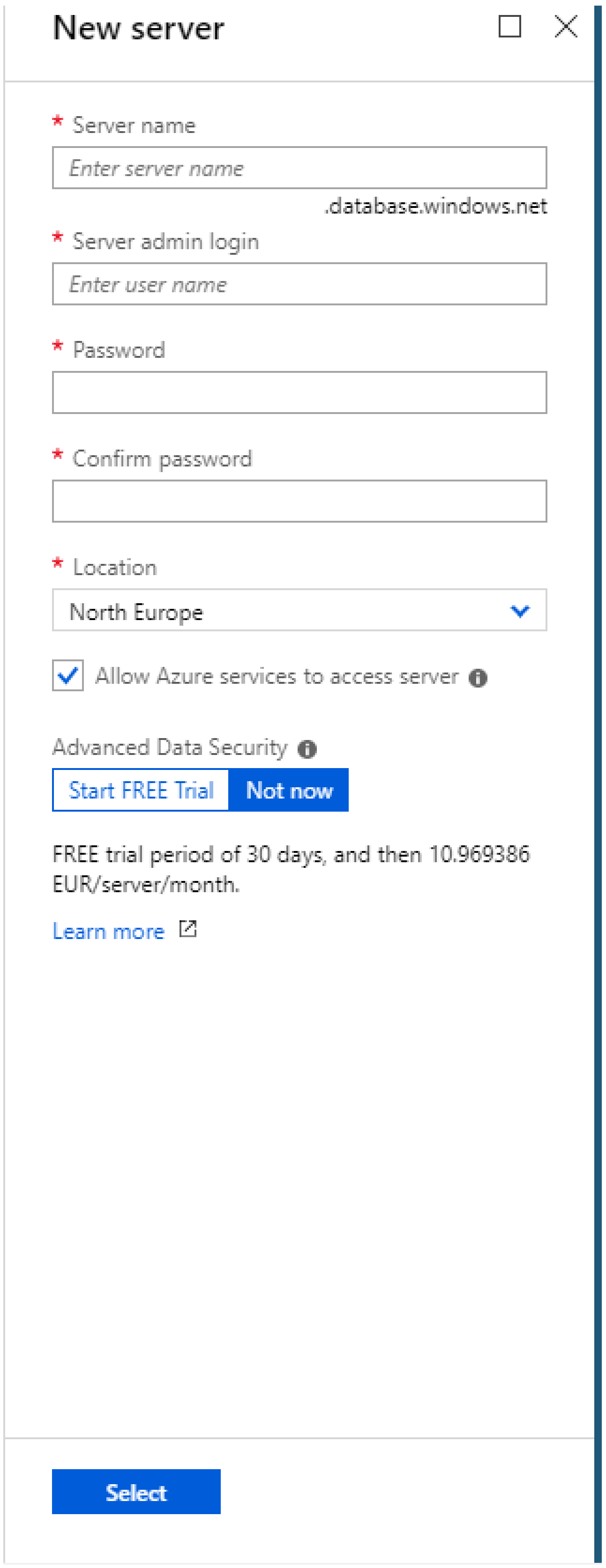
* Server: aune (North Europe)

Want to use SQL elastic pool? Yes [✓ Not now]

* Pricing tier: Standard S0: 10 DTUs, 250 GB

* Collation: SQL_Latin1_General_CP1_CI_AS

3. Create new Server for the database. Chose the name, server admin login and password. Make a note of login and password as you will use them later for authentication.


New server

* Server name

Enter server name

.database.windows.net

* Server admin login

Enter user name

* Password

* Confirm password

* Location

North Europe

[✓] Allow Azure services to access server

Advanced Data Security

Start FREE Trial [✓ Not now]

FREE trial period of 30 days, and then 10.969386 EUR/server/month.

Learn more

Select

4. Open newly created SQL database and go to Query editor (preview).
5. In Query editor create a new table to save the telemetry data. For that type the following query:

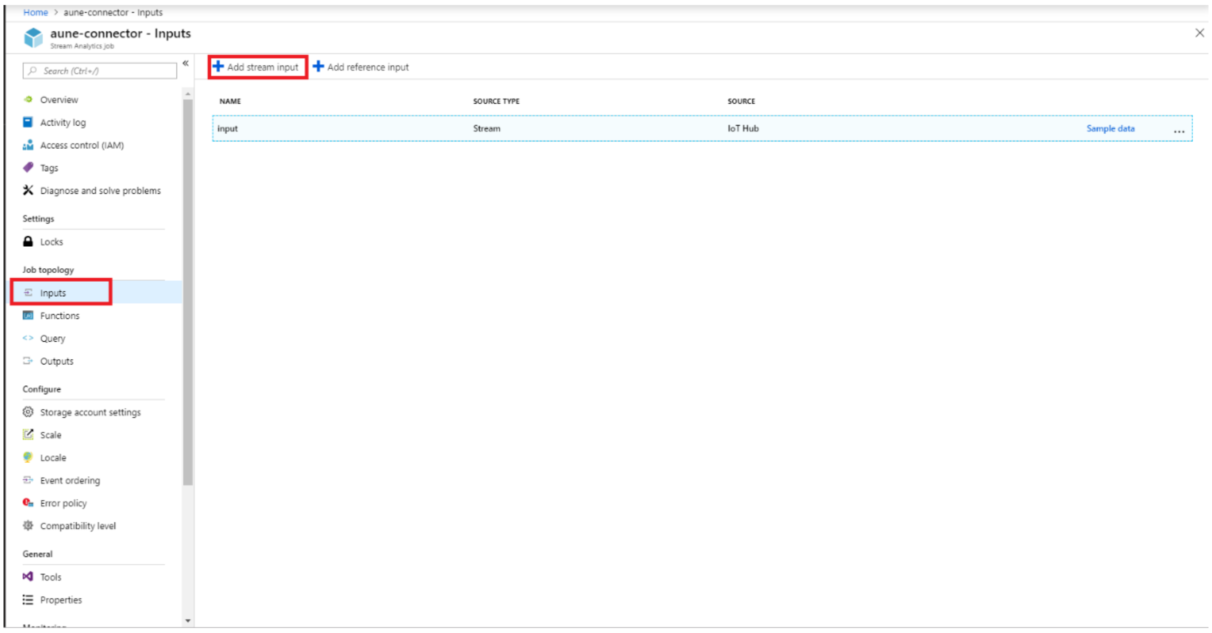
```
CREATE TABLE telemetry (
id PRIMARY KEY, INT, IDENTITY(1,1),
temperature FLOAT,
humidity FLOAT,
date_time DATETIME);
```

4. Create Stream Analytics job.

Stream analytics job allows to forward IoT messages to appropriate destinations.

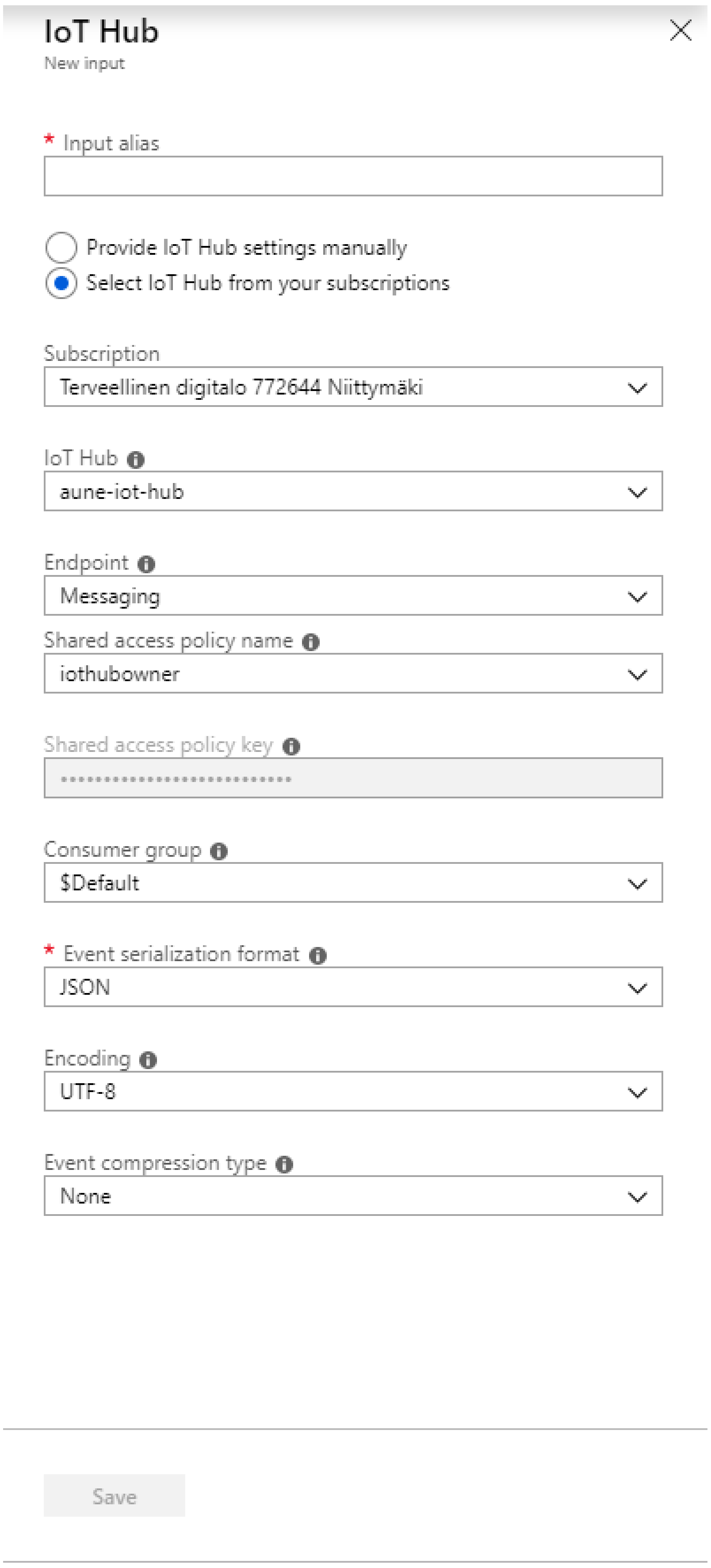
Instructions.

1. Create new Stream Analytics job. It can be found in Analytics folder.
2. Select the name of your choice, subscription, resource group, location, hosting environment (cloud) and streaming units (for such a simple task 1 is more than enough).
3. Open new Stream Analytics job and create new input and output.
4. For input, create new stream input for IoT Hub.



NAME	SOURCE TYPE	SOURCE	
input	Stream	IoT Hub	Sample data

5. Select previously created IoT hub and provide alias name. There is no need to change any of the default settings.



IoT Hub

New input

* Input alias

( ) Provide IoT Hub settings manually
[✓ Select IoT Hub from your subscriptions]

Subscription: Terveellinen digitalo 772644 Niittymäki

IoT Hub: aune-iot-hub

Endpoint: Messaging

Shared access policy name: iothubowner

Shared access policy key: ••••••••••••••••••••••••••••

Consumer group: $Default

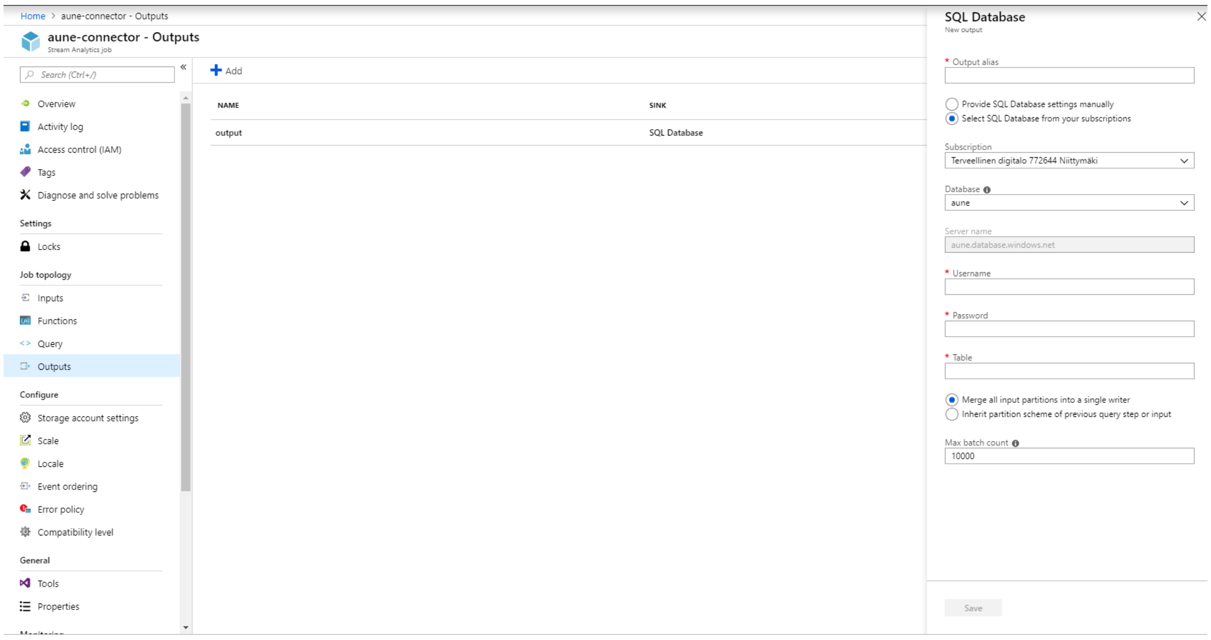
* Event serialization format: JSON

Encoding: UTF-8

Event compression type: None

Save

6. Create new output in a similar way. Select SQL Database as a type of the output and provide alias name, database server, login and password you provided while creating SQL server and table name.



NAME	SINK
output	SQL Database

SQL Database
New output

* Output alias:

( ) Provide SQL Database settings manually
(●) Select SQL Database from your subscriptions

Subscription: Terveellinen digitalo 772644 Niittymäki

Database: aune

Server name: aune.database.windows.net

* Username:

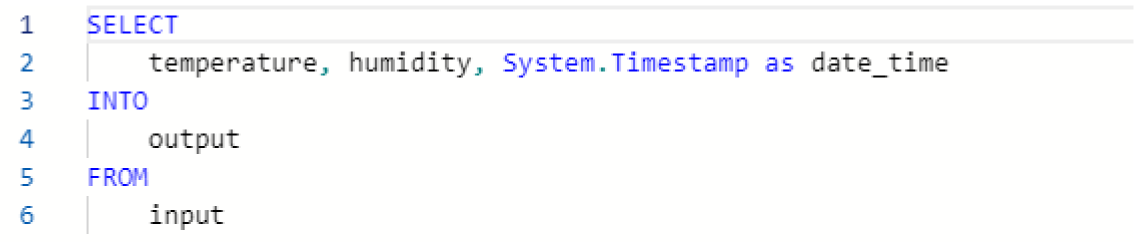
* Password:

* Table:

(●) Merge all input partitions into a single writer
( ) Inherit partition scheme of previous query step or input

Max batch count: 10000

7. Create the query for routing messages. Go to Query under Job topology section and modify the default query to the one provided on the screenshot below.



```
SELECT
    temperature, humidity, System.Timestamp as date_time
INTO
    output
FROM
    input
```

There is no need to provide id for our SQL database as it is automatically generated. The name of the property under SELECT must be same as a column name in the database table. In case it is initially different use AS.

8. Go to overview and start the Stream Analytic job. It can take quite some time. After completion, all the messages to the IoT ticket are going to be forwarded to SQL database.

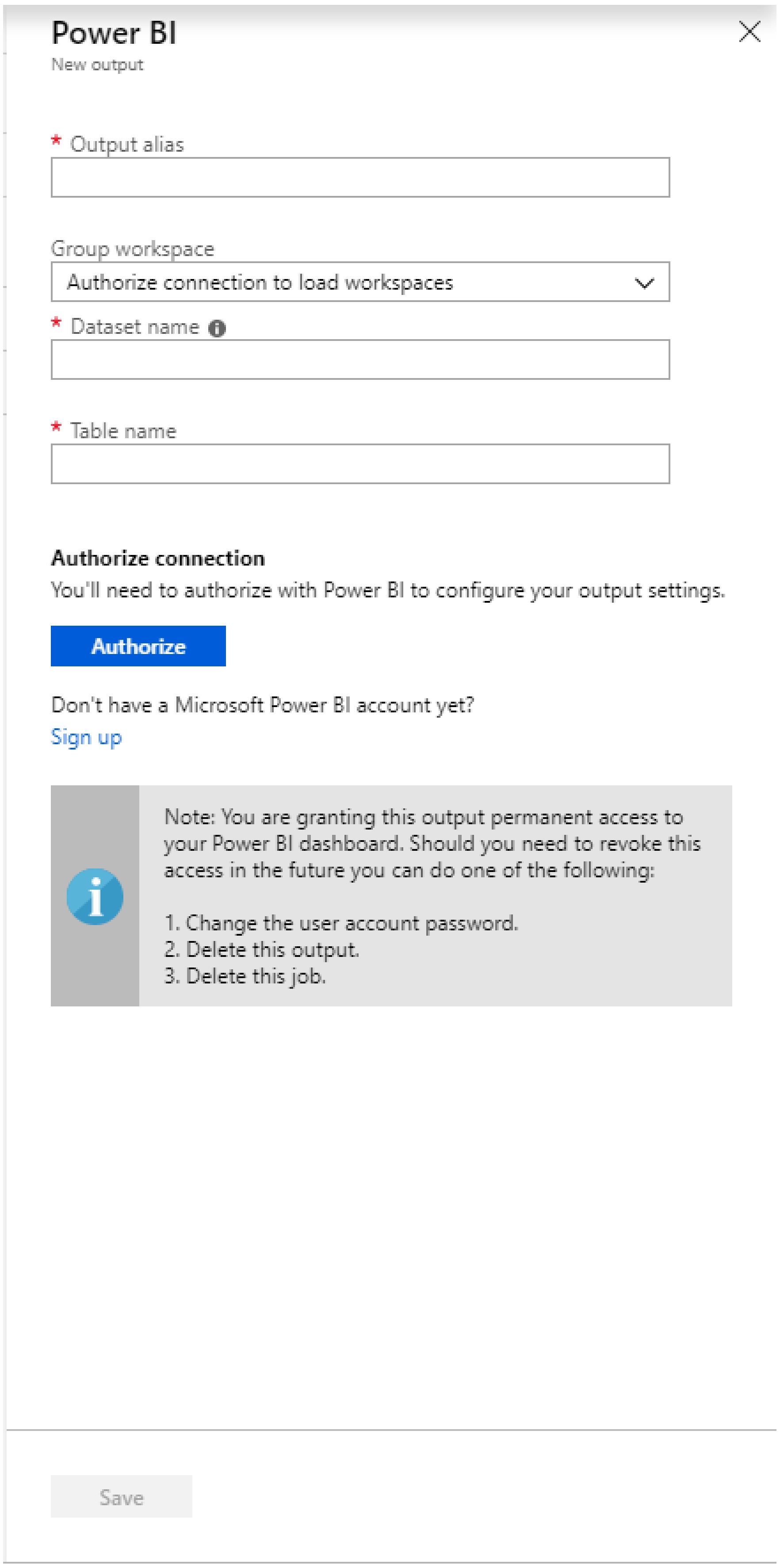
5. Visualization

This project relies on Power BI for visualizations.

Instruction.

1. Open Stream Analytics job you created previously for routing data to SQL database.

2. Create a new output for Power BI. Select alias, authorize with your student email account and select group workspace (my workspace).


Power BI
New output

* Output alias

Group workspace
Authorize connection to load workspaces

* Dataset name

* Table name

Authorize connection
You'll need to authorize with Power BI to configure your output settings.

Authorize

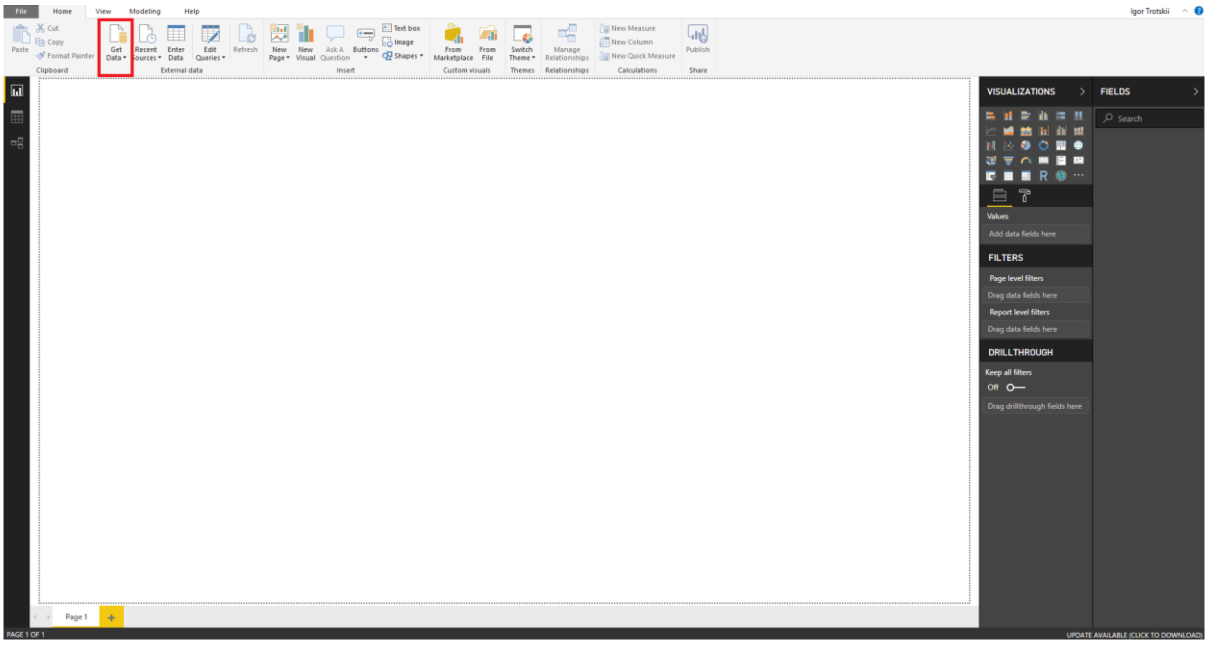
Don't have a Microsoft Power BI account yet?
Sign up

Note: You are granting this output permanent access to your Power BI dashboard. Should you need to revoke this access in the future you can do one of the following:

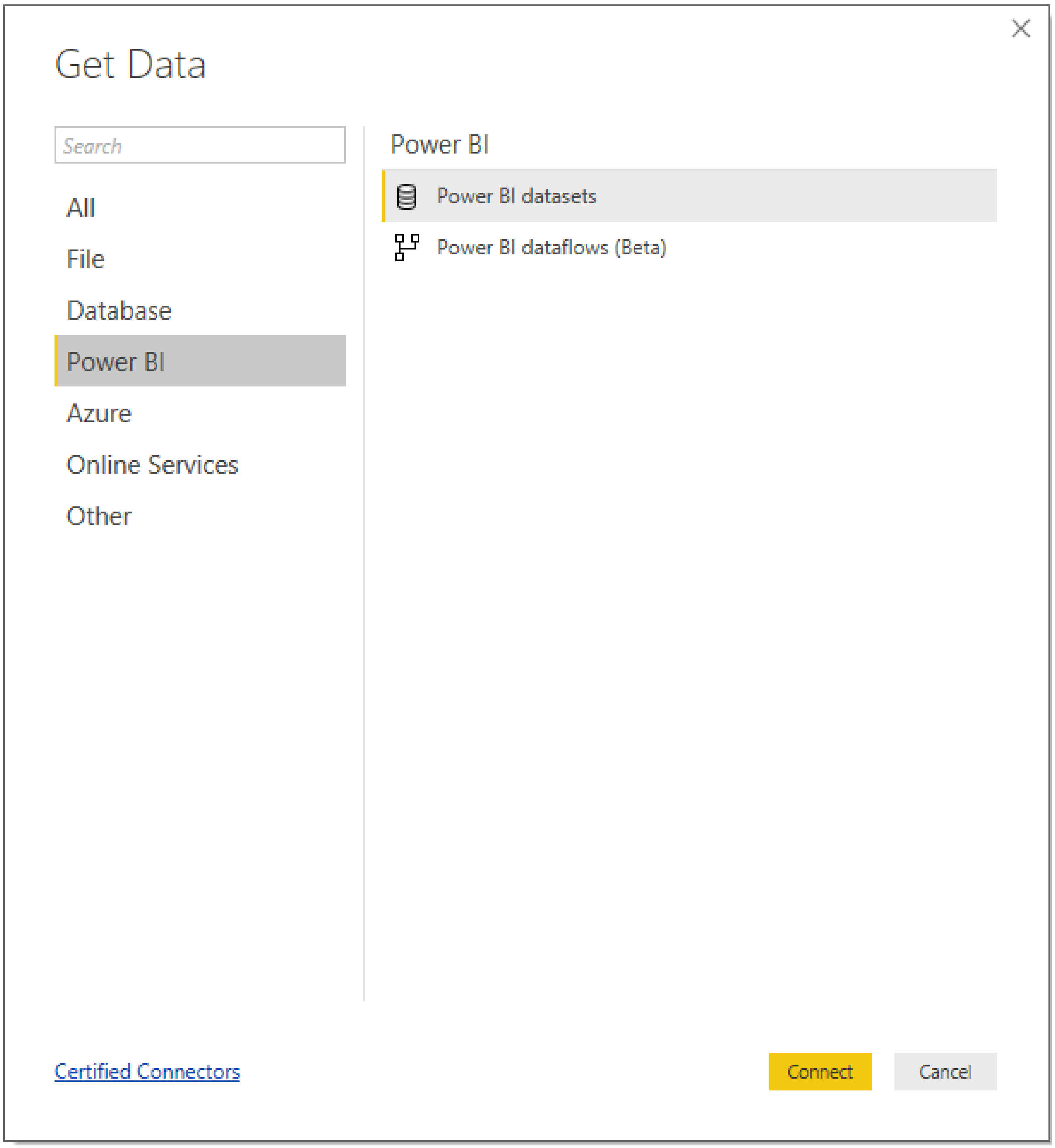
1. Change the user account password.
2. Delete this output.
3. Delete this job.

Save

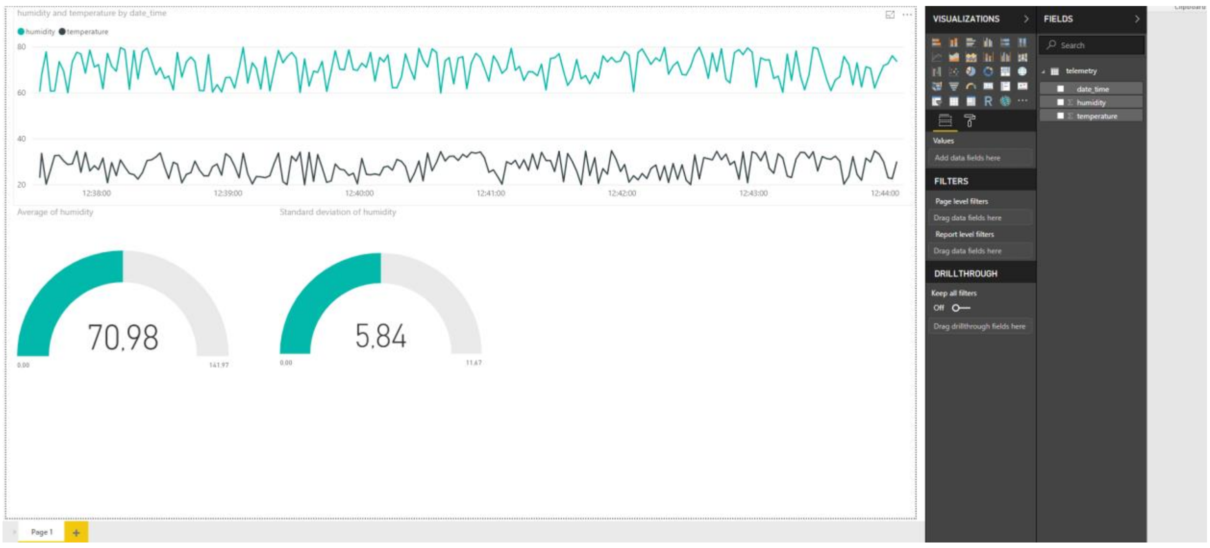
3. Change existing query to the one from the screenshot bellow and restart the job.
4. Open Power BI and press Get Data button.



5. Select Power BI -> Power Bi datasets as a data source.



6. Select workspace and dataset name you specified in the Stream Analytics.
7. Now you can visualize the data flowing into IoT hub.

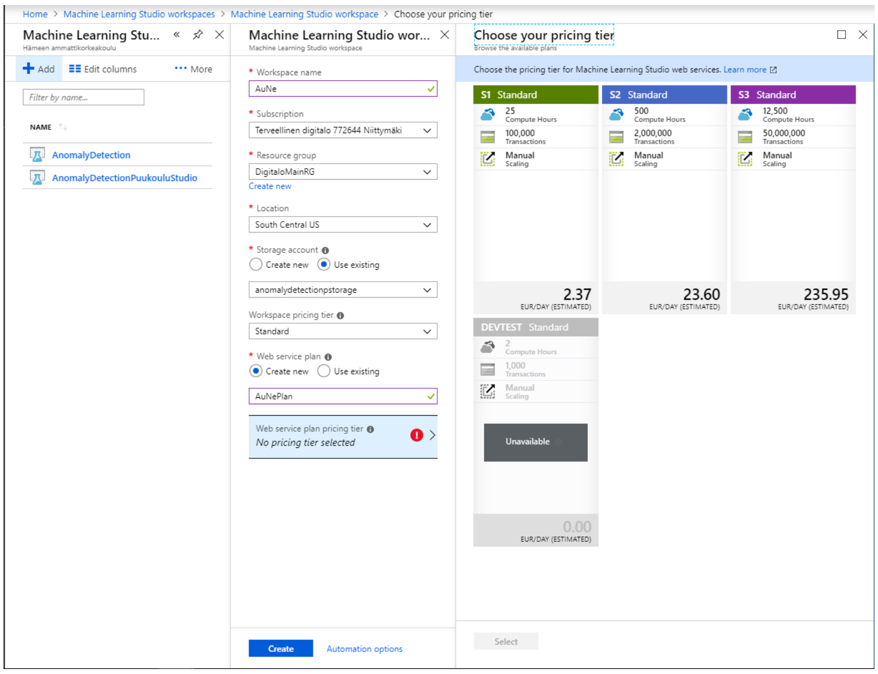


6. Machine learning studio

Azure provides you with extensive analytics capabilities. One of the most powerful and in the same time easiest tools to use is Azure Machine Learning studio. It provides user friendly graphical interface and a large amount of prepaid modules for machine learning and statistical analysis. For this project, you will learn how to use Azure Machine Learning studio to get basic statistical parameters of the simulated data.

Instruction.

1. Create a new resource: Machine Learning Studio workspace.
2. Fill required fields. Set Workspace pricing tier to Standard and Web Service plan to S1 Standard.



Home > Machine Learning Studio workspaces > Machine Learning Studio workspace > Choose your pricing tier

Machine Learning Stu...
Hämeen ammattikorkeakoulu

Add | Edit columns | More

Filter by name...

NAME
- AnomalyDetection
- AnomalyDetectionPuukouluStudio

Machine Learning Studio wor...
Machine Learning Studio workspace

* Workspace name: AuNe
* Subscription: Terveellinen digitalo 772644 Niittymäki
* Resource group: DigitaloMainRG
Create new
* Location: South Central US
* Storage account: ( ) Create new (●) Use existing
anomalydetectionpstorage
Workspace pricing tier: Standard
* Web service plan: (●) Create new ( ) Use existing
AuNePlan
Web service plan pricing tier
No pricing tier selected

Create | Automation options

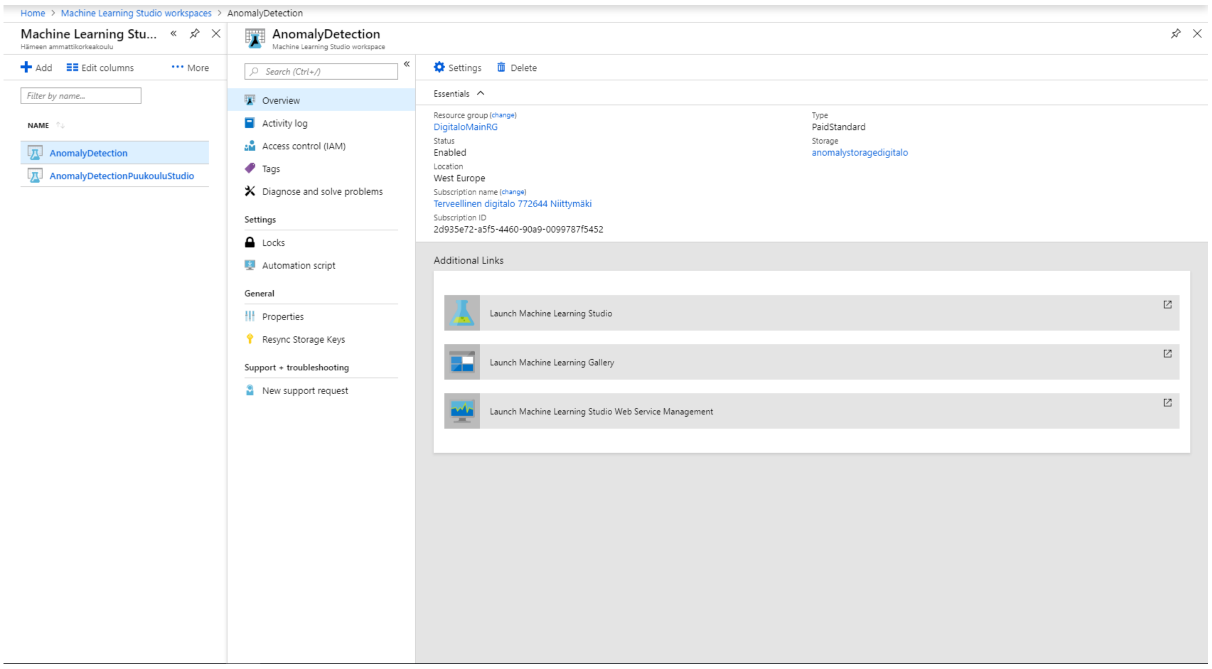
Choose your pricing tier
Browse the available plans

Choose the pricing tier for Machine Learning Studio web services. Learn more

S1 Standard	S2 Standard	S3 Standard	DEVTEST Standard
25 Compute Hours	500 Compute Hours	12,500 Compute Hours	2 Compute Hours
100,000 Transactions	2,000,000 Transactions	50,000,000 Transactions	1,000 Transactions
Manual Scaling	Manual Scaling	Manual Scaling	Manual Scaling
			Unavailable
2.37 EUR/DAY (ESTIMATED)	23.60 EUR/DAY (ESTIMATED)	235.95 EUR/DAY (ESTIMATED)	0.00 EUR/DAY (ESTIMATED)

Select

3. Open created resource and launch Machine Learning Studio.



Home > Machine Learning Studio workspaces > AnomalyDetection

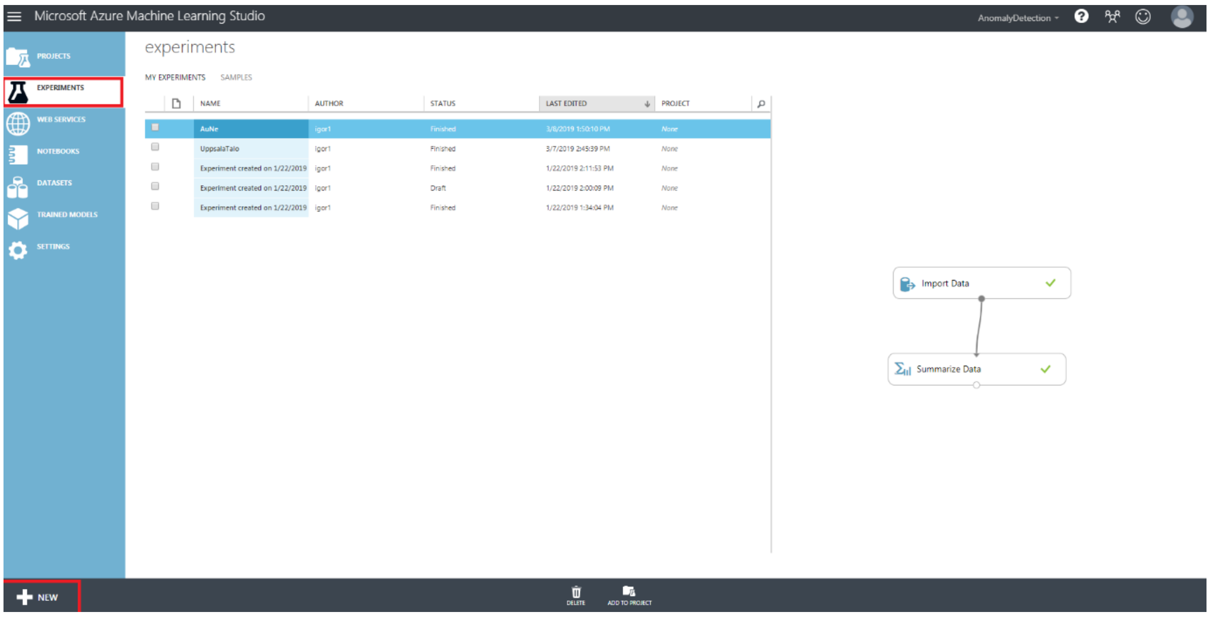
Machine Learning Stu...
Hämeen ammattikorkeakoulu

NAME
AnomalyDetection
AnomalyDetectionPuukouluStudio

AnomalyDetection
Machine Learning Studio workspace

Resource group (change)
DigitaloMainRG
Status
Enabled
Location
West Europe
Subscription name (change)
Terveellinen digitalo 772644 Niittymäki
Subscription ID
2d935e72-a5f5-4460-90a9-0099787f5452
Type
PaidStandard
Storage
anomalystoragedigitalo

4. Press Sign In button, if you are not sent directly to Machine Learning Studio.
5. Create new experiment.



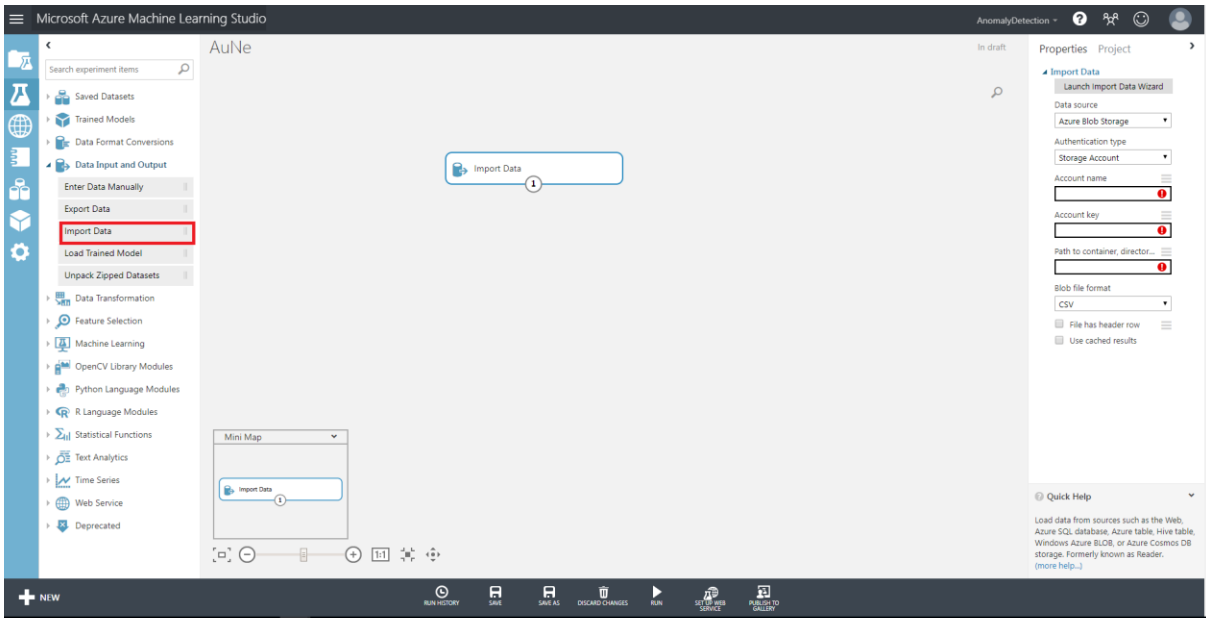
Microsoft Azure Machine Learning Studio

experiments

MY EXPERIMENTS SAMPLES

NAME	AUTHOR	STATUS	LAST EDITED	PROJECT
AuNe	igor1	Finished	3/8/2019 1:50:10 PM	None
UppsalaTalo	igor1	Finished	3/7/2019 2:45:39 PM	None
Experiment created on 1/22/2019	igor1	Finished	1/22/2019 2:11:53 PM	None
Experiment created on 1/22/2019	igor1	Draft	1/22/2019 2:00:09 PM	None
Experiment created on 1/22/2019	igor1	Finished	1/22/2019 1:34:04 PM	None

6. In the opened window go to Data Input and Output folder, select and drag Import Data to the workspace.



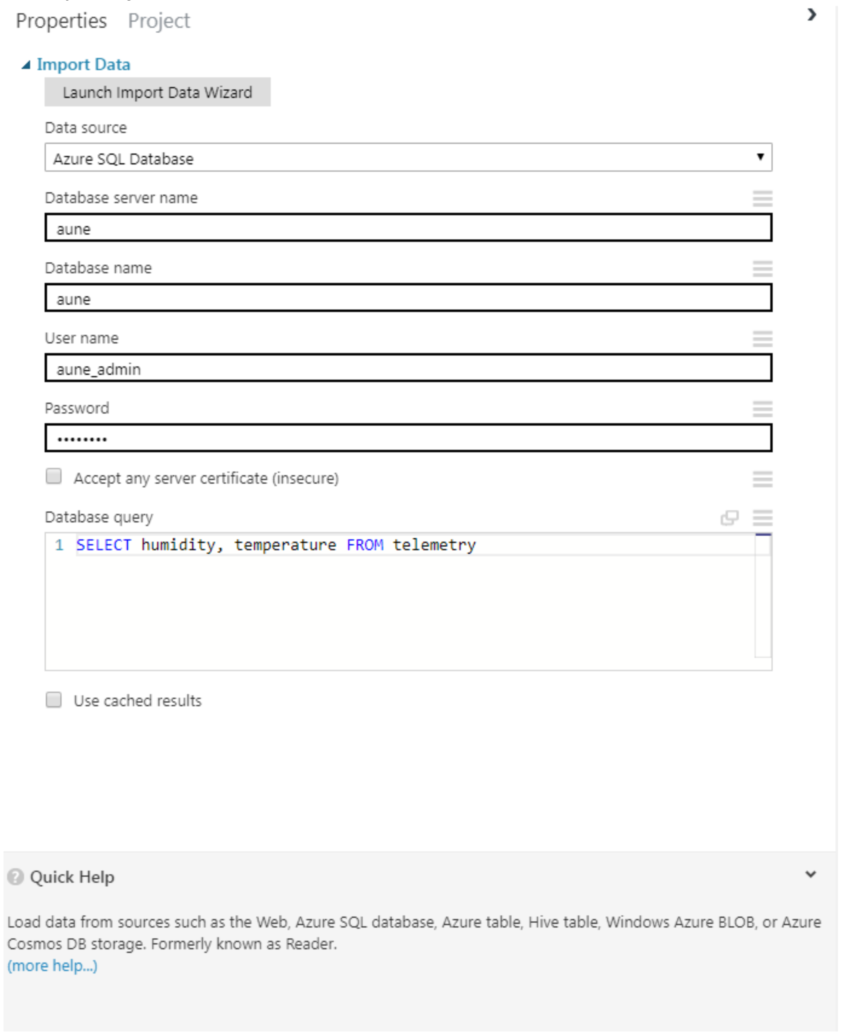
Data source: Azure Blob Storage

Authentication type: Storage Account

Blob file format: CSV

Load data from sources such as the Web, Azure SQL database, Azure table, Hive table, Windows Azure BLOB, or Azure Cosmos DB storage. Formerly known as Reader. (more help...)

7. Configure Input Data node to obtain data from the SQL database. You can use Launch Import Data Wizard button for that or do it manually. For database server, name, user name and password use same values you used during Stream Analytics job creation.



Data source: Azure SQL Database

Database server name: aune

Database name: aune

User name: aune_admin

Password: ••••••••

Accept any server certificate (insecure)

Database query

```
SELECT humidity, temperature FROM telemetry
```

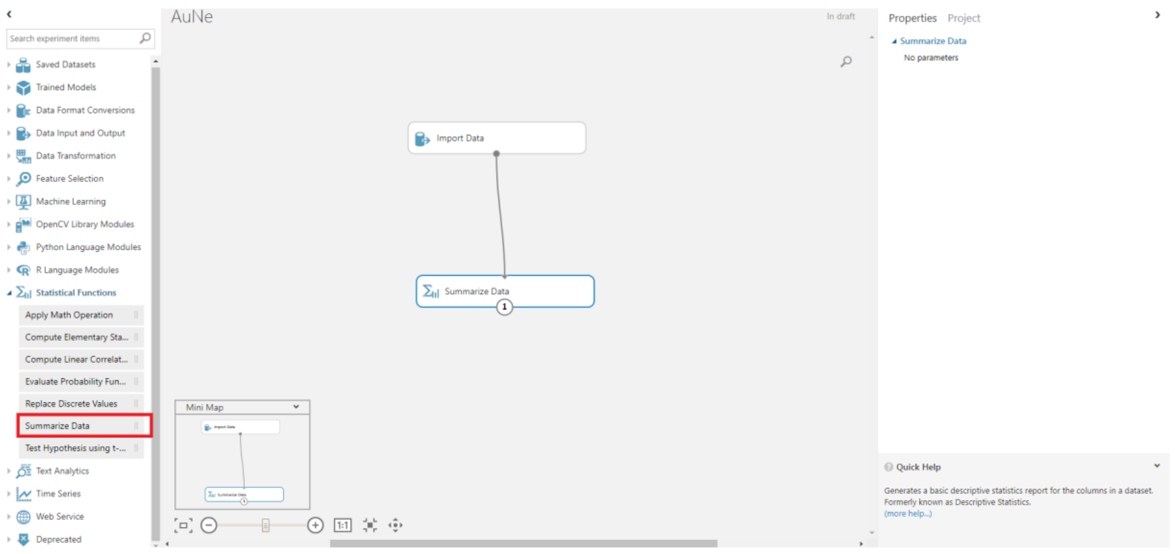
Use cached results

Load data from sources such as the Web, Azure SQL database, Azure table, Hive table, Windows Azure BLOB, or Azure Cosmos DB storage. Formerly known as Reader.
(more help...)

8. Create database query to get the data out of SQL database:

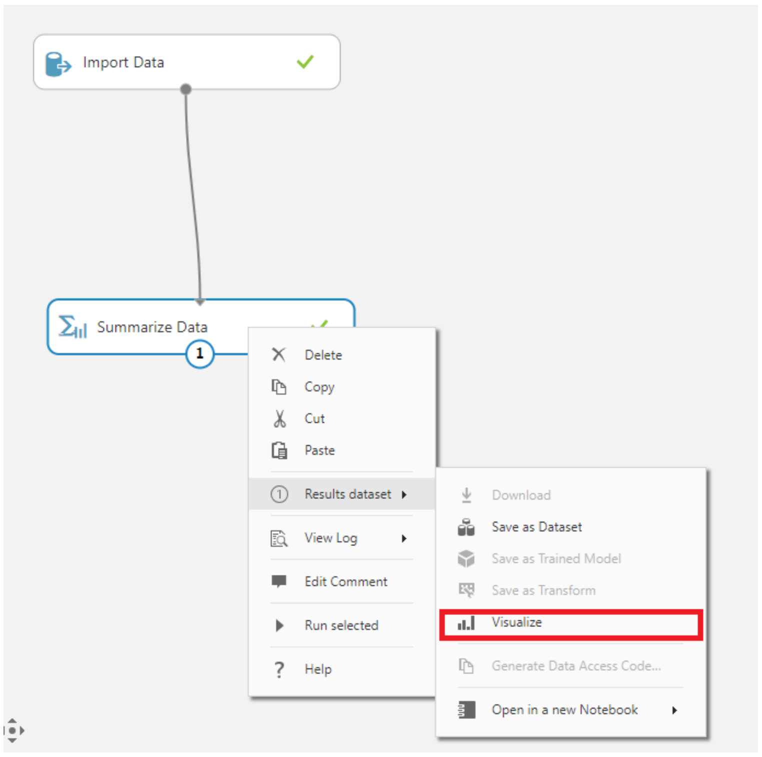
SELECT humidity, temperature FROM telemetry

9. From Statistical Functions drag and drop Summarize Data node and connect it to Import Data node.

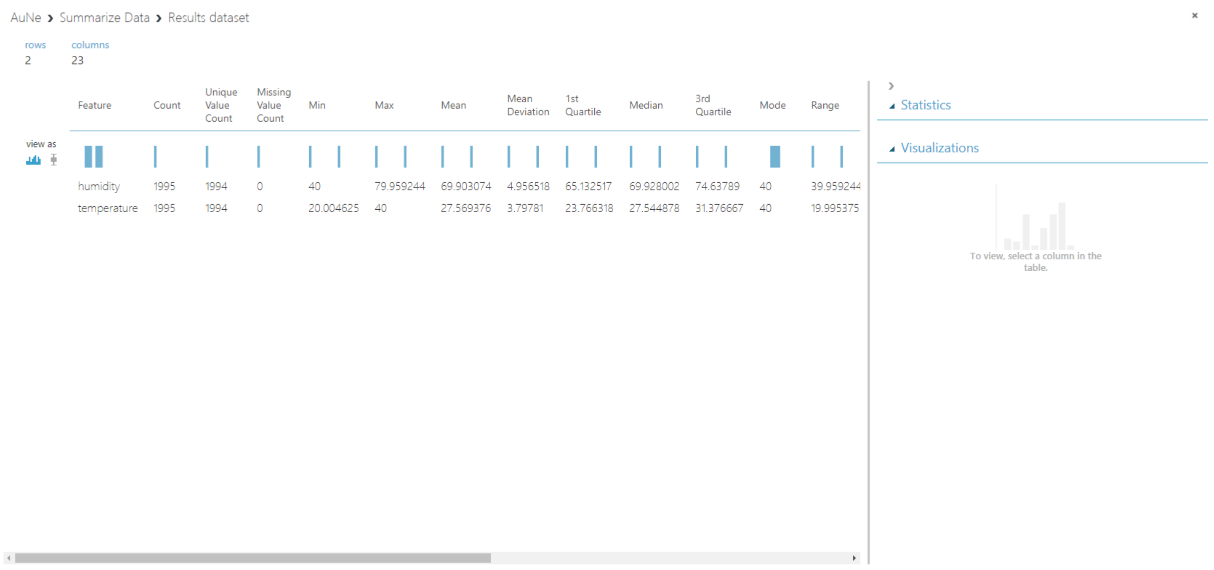


Generates a basic descriptive statistics report for the columns in a dataset. Formerly known as Descriptive Statistics.
(more help...)

10. Press Run button on the bottom of the page.
11. Check the results from Summarize Data node: Right click on the node->Result dataset->Visualize.



12. Now you can check most important statistical parameters of the generated data.



AuNe › Summarize Data › Results dataset

rows 2 columns 23

Feature	Count	Unique Value Count	Missing Value Count	Min	Max	Mean	Mean Deviation	1st Quartile	Median	3rd Quartile	Mode	Range
humidity	1995	1994	0	40	79.959244	69.903074	4.956518	65.132517	69.928002	74.63789	40	39.959244
temperature	1995	1994	0	20.004625	40	27.569376	3.79781	23.766318	27.544878	31.376667	40	19.995375

7. Other data analysis possibilities for machine learning studio.

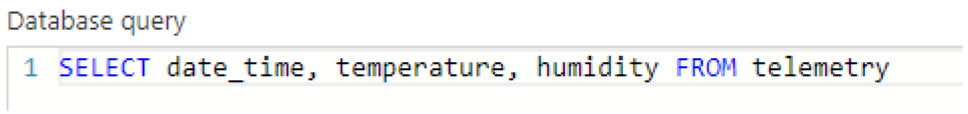
This chapter provides additional information about applying other techniques for data analysis. The section covers time series anomaly detection and k-Means clustering.

Time series anomaly detection.

Time series anomaly detection allows to find outliers in time series types of data, which can indicate different faults or anomalies in the system, which produced the measurements. The main difference from most of the usual algorithms is that you do not have many examples, which represent anomalies, therefore the algorithm has to learn from very few examples.

Instructions.

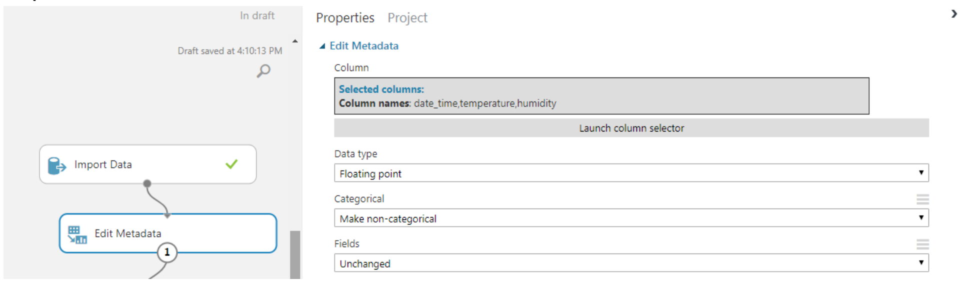
1. Open previously created machine learning studio instance and proceed to the previous experiment.
2. In the Import Data node change the query to include timestamp column.



Database query

```
SELECT date_time, temperature, humidity FROM telemetry
```

3. Add Edit Metadata node and select all available columns. Leave data type filed set to Unchanged and set categorical field to make non-categorical. This will ensure proper datatype for the next node. For all the columns to appear you might need to run the experiment at least once.



Edit Metadata

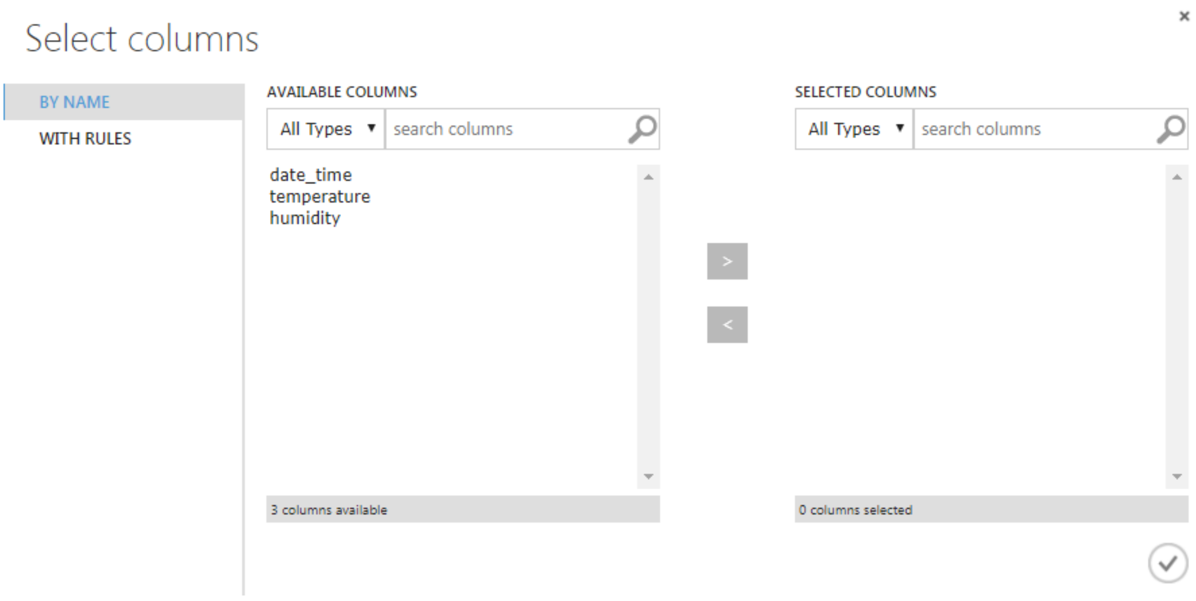
Column

Selected columns:
Column names: date_time,temperature,humidity

Data type: Floating point

Categorical: Make non-categorical

Fields: Unchanged

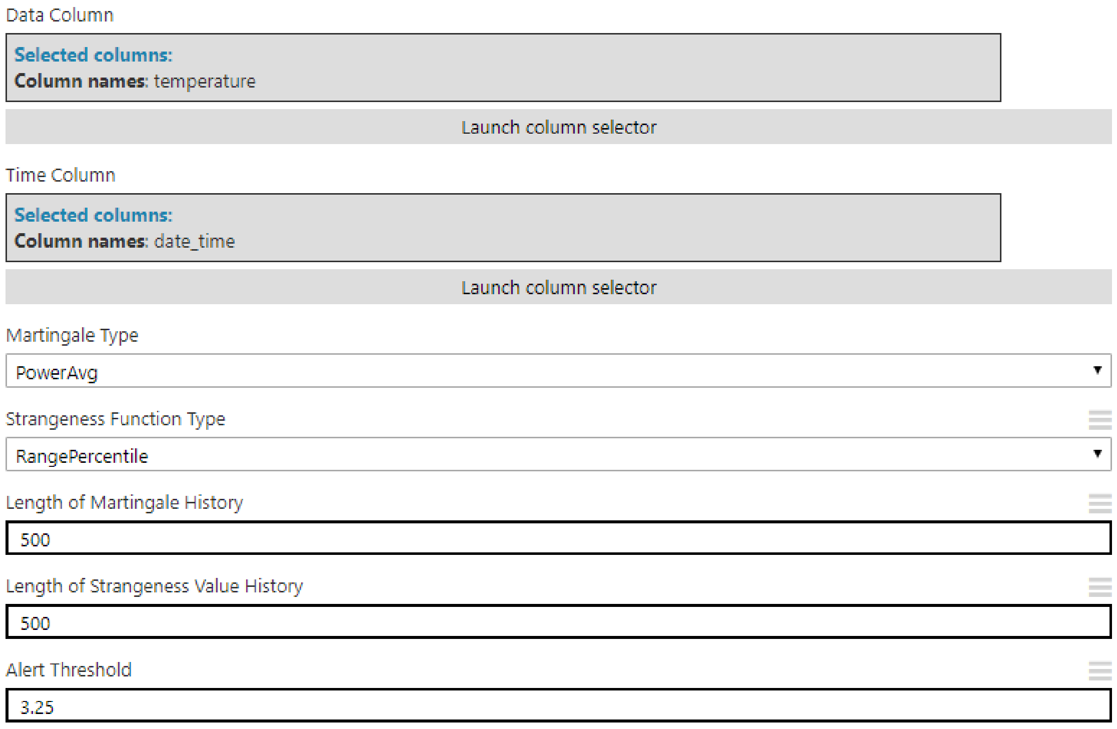

Select columns

BY NAME
WITH RULES

AVAILABLE COLUMNS
All Types ▾ search columns
date_time
temperature
humidity
3 columns available

SELECTED COLUMNS
All Types ▾ search columns
0 columns selected

4. Add Time Series Anomaly Detection node. Select data column (either humidity or temperature, not both) and time column (date_time if you followed the instructions). Change alert threshold to the value of your choice (the smaller it is the stricter the anomaly detection) Without this change you are unlikely to get any anomalies from uniformly distributed data set, we generated for this project.


Data Column
Selected columns:
Column names: temperature
Launch column selector

Time Column
Selected columns:
Column names: date_time
Launch column selector

Martingale Type
PowerAvg

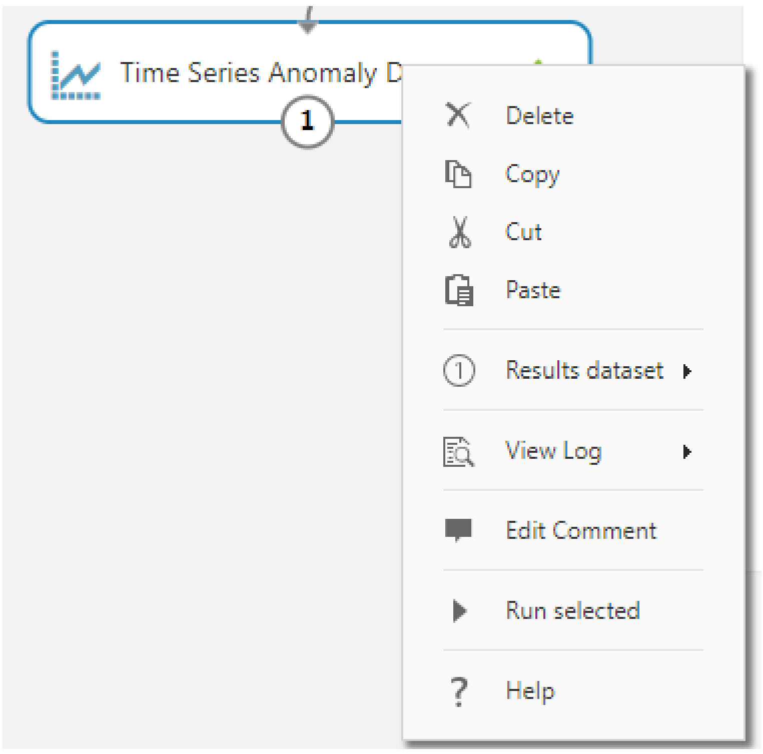
Strangeness Function Type
RangePercentile

Length of Martingale History
500

Length of Strangeness Value History
500

Alert Threshold
3.25

5. Run the experiment.
6. Check the results. You can either see default visualization, which covers only small part of the whole dataset or export dataset to see it fully.

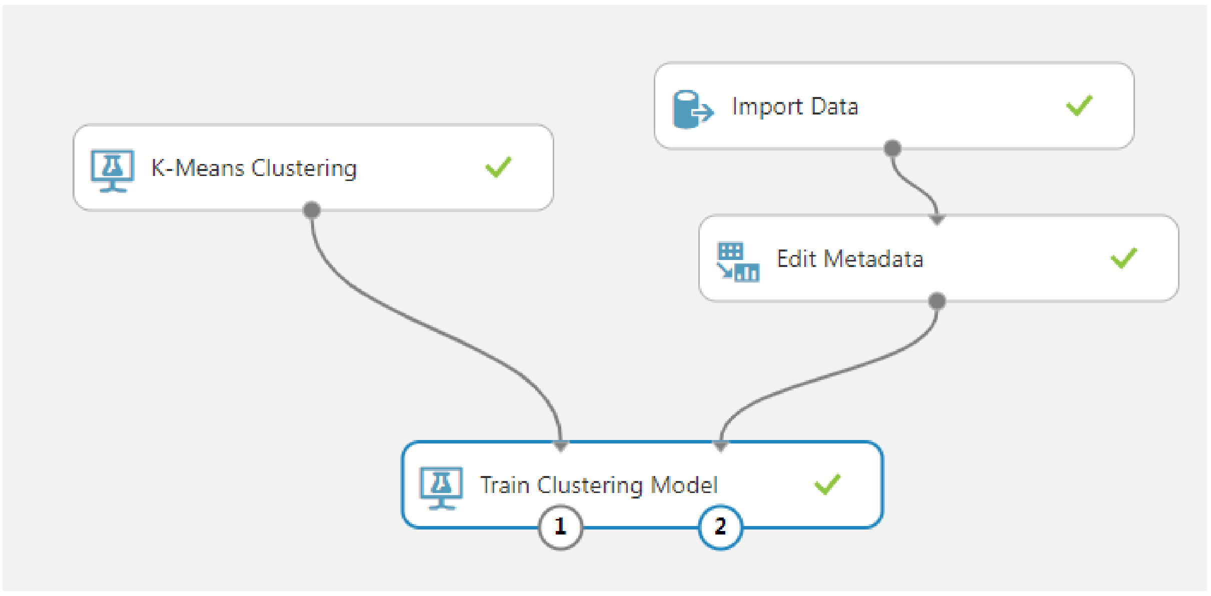


k-Means clustering.

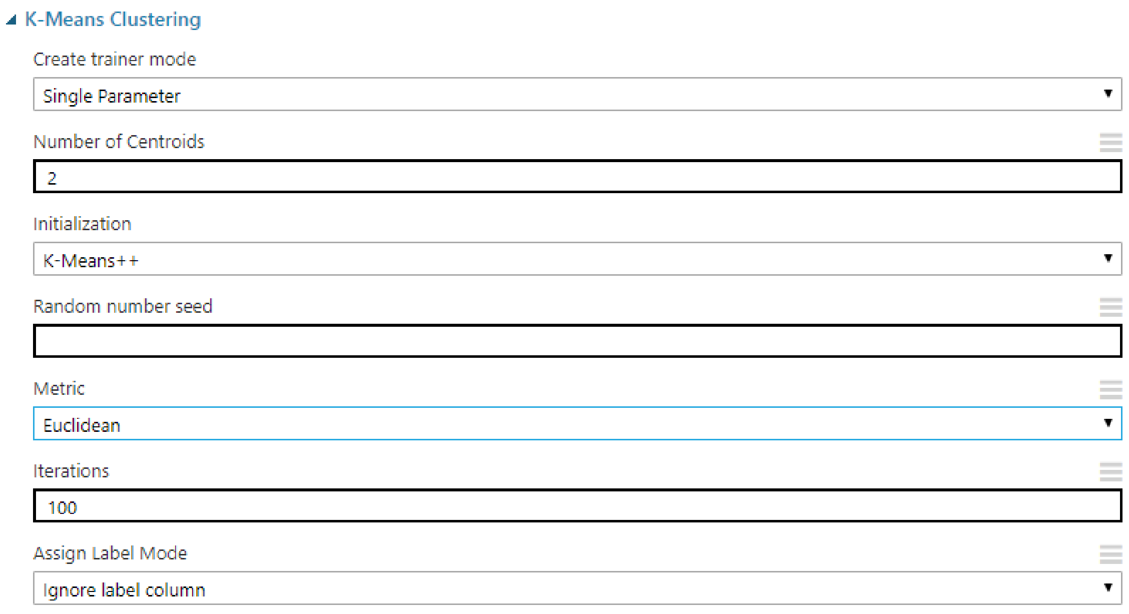
Clustering allows to automatically separate data into groups. It is an essential way to study unfamiliar data for some internal relationships. In our case clustering will be able to produce any sensible results, as we only have simulated data from uniform distribution, but if you, for example, had real temperature sensor, then you could find day/night separation.

Instructions.

1. Open previous experiment and remove Time Series Anomaly Detection node.
2. Add K-Means clustering and Train Clustering Model nodes to the experiment.
3. Connect all the nodes in the following way.



4. Configure k-Means clustering model.



K-Means Clustering

Create trainer mode: Single Parameter

Number of Centroids: 2

Initialization: K-Means++

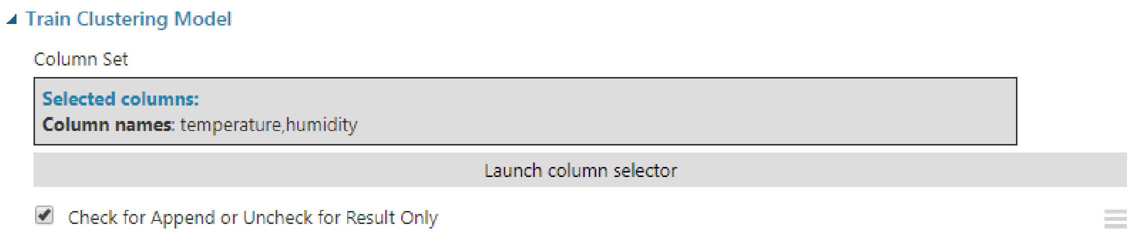
Random number seed:

Metric: Euclidean

Iterations: 100

Assign Label Mode: Ignore label column

5. Configure Train Clustering model node.



Train Clustering Model

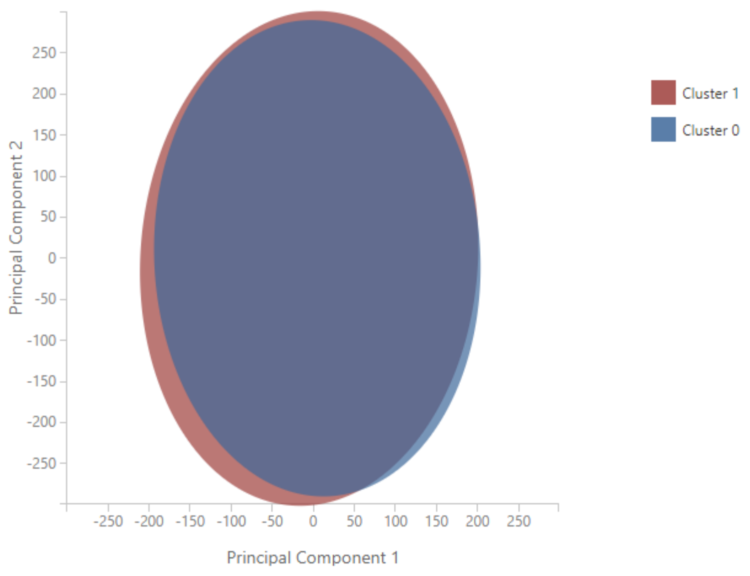
Column Set

Selected columns:
Column names: temperature,humidity

☑ Check for Append or Uncheck for Result Only

Select only temperature and humidity in column selector.

6. Run the experiment.
7. Check the results of the clustering by right clicking on the Train Clustering Model node and selecting visualize in the result dataset subsection.



The clustering algorithms was not able to separate our data into 2 different clusters. That was expected as we have data from the same uniform distributions, meaning we only have one cluster.

8. Conclusion.

After completing described steps you've got an understanding of basic components, which can be used in order to implement IoT system based on Azure cloud. Provided system can be used as a backbone for more complicated systems. Data stored in SQL database can easily be explored and modelled with the use of Azure Machine learning studio.