Anomaly Detection in IoT Telemetry

Azure IoT Workshop

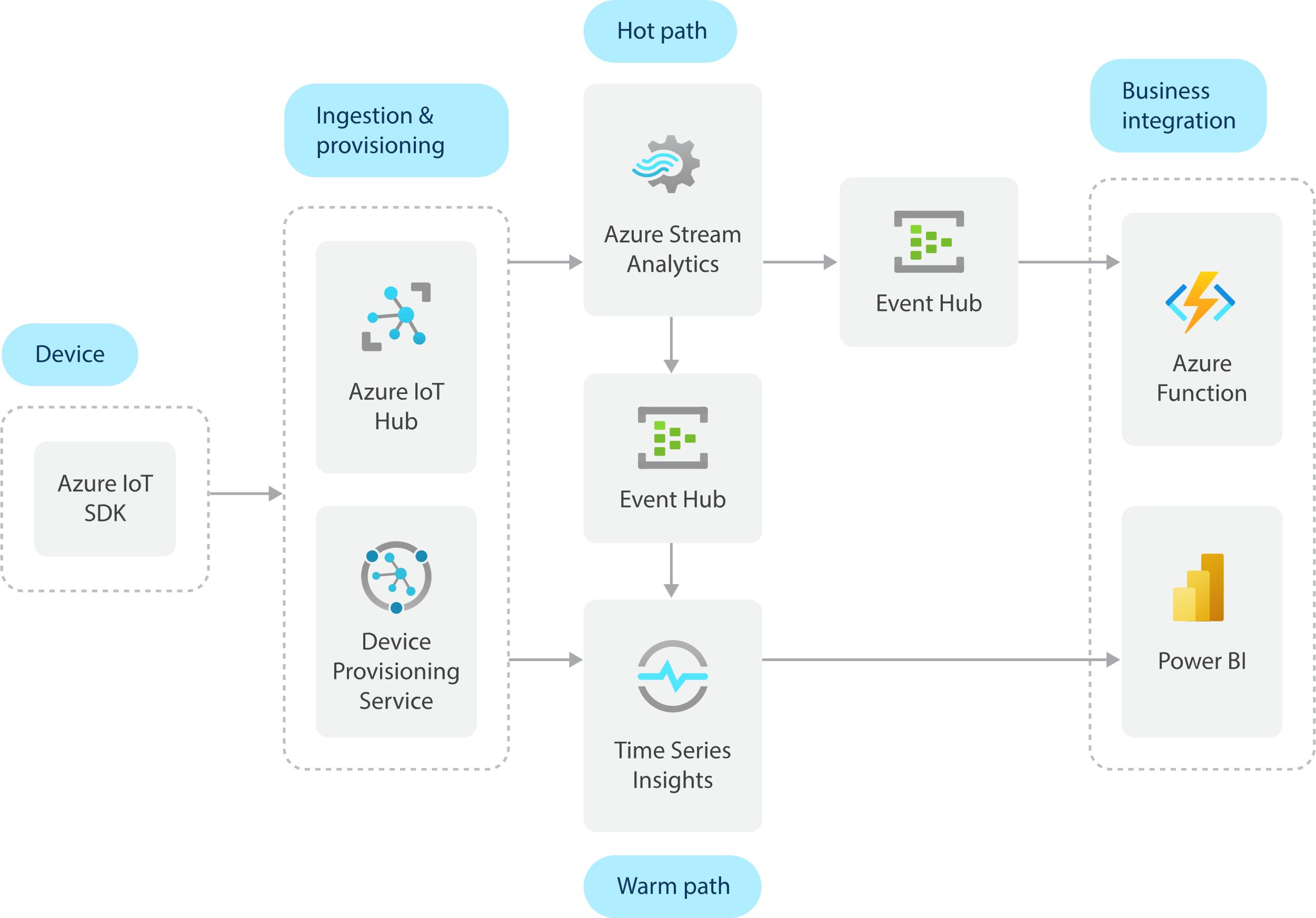
By Jevgenijs Pankovs

# Introduction

The goal of this workshop is to show you in minute details how to implement a typical IoT solution, fulfilling the following requirements:

* An IoT device, sending telemetry data, shall be connected to an Azure IoT Hub
* The telemetry data shall be processed with Azure Stream Analytics service to detect anomalies – spikes and dips or change points
* Visualize telemetry data and anomalies with Azure Time Series Insights
* Visualize telemetry and anomalies with Power BI
* Egress Stream Analytics output to an Azure Function to process anomalies.

The following diagram reflects different Azure components that will be used to architect our IoT solution:



*Figure 1. Diagram of the architecture*

First, we will deploy all the components to the Cloud manually. Then, we will be using a deployment script, creating the same components, to simplify and speed up our work.

Though, we could use a real IoT device (like a Raspberry Pi), we will create a device simulator using Visual Studio Code, Azure IoT SDK and Python.

Here is our plan:

1. Create a resource group in Azure portal
2. Create an IoT hub
3. Create a Device Provisioning service, and link it to the IoT hub
4. Create a Stream Analytics job, and configure its inputs, outputs, and queries
5. Create a Time Series Insights environment, and configure its event sources
6. Create an IoT device simulator, connect it to the IoT hub, and send telemetry data
7. Use Azure IoT Explorer to view the telemetry the device is sending
8. Visualize telemetry data in Time Series Insights
9. Visualize data from Time Series Insights in Power BI
10. Create an Azure Function and trigger it in response to Stream Analytics events.

Enough with the introduction, let’s start.

# Hands-on Lab

## 1. Create a resource group

Let us first create an Azure Resource Group, which will be a logical container for all our components. All the resources in a resource group will have the same life cycle. That means when a resource group is deleted, all resources in the group are deleted as well. The group needs a location to specify where the metadata will be stored, which is necessary for some compliance policies.

1. Open Azure portal on the browser by navigating to <http://portal.azure.com>
2. From the left top side of the portal home page click **+ Create a resource** and select **Resource group** from the features options
3. Select your **Subscription**, enter ***anomaly-detection-rg*** as a **Resource group** name, select ***North Europe*** **Region**, and then select **Review + Create**
4. On the **Review + Create** page, select **Create**.

Graphical user interface, text, application, email

Description automatically generated

Once the deployment is complete, select **Go to resource** to open the new resource group. Click **+ Create** to create a new resource:

Graphical user interface, text, application

Description automatically generated

## 2. Create an IoT hub

Now, we will create an IoT hub, acting as a central message hub for communication between our IoT applications and its attached devices.

Graphical user interface, text, application, email

Description automatically generated

1. From the list of resources select **IoT Hub**.
2. Enter ***anomaly-detection-hub*** as a name of the IoT hub, and then select ***North Europe*** **Region**
3. Press **Next: Networking**, and then select **Next: Management** to continue creating your hub.

### Graphical user interface, text, application, email Description automatically generated

1. Leave **S1: Standard tier** as **Pricing and scale tier**
2. Select **Review + Create** and click **Create**.

Once the IoT hub is created, select **Go to resource** to open the hub.

IoT Hub exposes a built-in endpoint to read the device-to-cloud messages received by your hub. IoT Hub enables you to create consumer groups on the endpoint, enabling multiple consuming applications to read the stream independently.

We are going to create a new consumer group for Azure Stream Analytics service, reading telemetry data from our IoT hub.

1. Go to **Built-in endpoints** tab
2. Enter ***asaconsumergroup*** as a new **Consumer Group**.

## Graphical user interface, text, application Description automatically generated

## 3. Create a Device Provisioning Service

Azure IoT Service offers many ways to connect an IoT device to your IoT hub. For example, you can register your devices in the IoT hub manually and connect them using symmetric keys or X.509 certificates. But if you need to provision thousands (or even millions) of devices, that’s not the case.

We need to realize a scenario when a connected IoT device is automatically registered in our IoT hub. That is why we will use Device Provisioning Service (DPS), supporting IoT Plug and Play architecture. Once a device starts communicating using Plug and Play protocol, it is automatically registered (if it was not) in the IoT hub. That is what we need. But first, let’s create a DPS.

1. In the Azure portal, select **+ Create a resource**, and select **IoT Hub Device Provisioning Service**
2. Select your **Subscription**, select ***anomaly-detection-rg*** as a **Resource group**, and ***North Europe*** as a **Region**
3. Enter ***anomaly-detection-dps*** as a **Name** and then select **Review + Create**
4. On the **Review + Create** page, select **Create**.

Graphical user interface, text, application, email

Description automatically generated

1. Once the DPS instance is created, open it, and go to **Linked IoT hubs** tab
2. Click **+ Add** button to link our previously created IoT hub

Graphical user interface, text, application

Description automatically generated

1. On the **Add link to IoT hub** panel, select your **Subscription**
2. Select ***anomaly-detection-hub*** as a name of the **IoT hub**
3. Select ***iothubowner*** as **Access Policy**, and then click **Save**.

Graphical user interface, text, application, email

Description automatically generated

We’ve just linked our IoT hub to the DPS. If you do not see your hub in the list of linked hubs, click **Refresh**.

Now, we need to create a DPS Enrollment Group. This enrollment group will have a set of credentials for all IoT devices in the same group.

Select **Manage enrollments** tab and click **+ Add enrollment group**.

Graphical user interface, text, application

Description automatically generated

1. On **Add Enrollment Group**, enter ***anomaly-detection-group*** as a **Group name**
2. Select ***Symmetric Key*** as **Attestation Type**
3. Check **Auto-generate keys** checkbox and click **Save**.

Graphical user interface, text, application, email

Description automatically generated

Once you saved your enrollment group, open the enrollment, copy, and save the value of your generated **Primary Key** for future usage. This key is your master group key – we will need it in our IoT device simulator.

Graphical user interface, application, Teams

Description automatically generated

Another important parameter, needed for our device simulator is DPS ID Scope. Go to **Overview** tab and find **ID Scope** value:

Graphical user interface, application

Description automatically generated

Copy and save it somewhere in your notes as **DPS ID Scope** value.

## 4. Create a Stream Analytics job

We are going to receive telemetry stream (temperature and humidity) and detect in real time sudden temperature spikes or dips, using Stream Analytics Query Language.

Azure Stream Analytics offers built-in machine learning based **AnomalyDetection\_SpikeAndDip** function (see [Anomaly detection in Azure Stream Analytics](https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-machine-learning-anomaly-detection)) which returns two columns:

**IsAnomaly** - a BIGINT (0 or 1) indicating if the event was anomalous or not, and  
**Score** - the computed p-value score (float) indicating how anomalous an event is.

We will output the results into an Event Hub so the events corresponding to anomalies can be used by other components: Time Series Insights, Power BI, and Azure Functions.

Now, we will set up a Stream Analytics job with this anomaly detection function to read telemetry from our IoT Hub and detect anomalies.

1. In the Azure portal, select **+ Create a resource**, and select **Stream Analytics job**
2. Enter ***streamjob*** as a **Job name**
3. Select your **Subscription**, select ***anomaly-detection-rg*** as a **Resource group**, and ***North Europe*** as a **Region**
4. Click **Create**.

Graphical user interface, text, application

Description automatically generated

Once our Stream Analytics job is created, we need to configure it – create an **Input**, **Output**, and a **Query**.

### 4.1 Configure Stream Analytics Input

Our next step is to define an input source for the job to read data from the IoT hub.

1. Select **Inputs** tab
2. Click **+ Add stream input** and select **IoT Hub.** IoT Hub panel will be opened (see figure below)
3. Enter ***input*** as **Input alias**
4. Select your **Subscription**
5. Select ***anomaly-detection-hub*** as an **IoT Hub**
6. Select ***asaconsumergroup*** as a **Consumer group**
7. Select ***iothubowner*** as a **Shared access policy name**
8. Click **Save**

Graphical user interface, application

Description automatically generated

We want to detect anomalies in temperature and feed these anomalous “events” into other systems or services: Time Series Insights and Azure Functions. Stream Analytics supports many types of outputs: Azure Data Lake, Azure SQL Database, Azure Synapse Analytics, Blob and Table storages, Event Hub, Service Bus, Power BI, Cosmos DB, and Azure Functions. We will use two Event Hubs as outputs for our job. One – for Time Series Insights, and second – for an Azure Function.

### 4.2 Configure Stream Analytics Outputs

First, we need to create a new Event Hub namespace in which we create our event hubs.

1. Go to **anomaly-detection-rg** resource group, click **+ Create**, and select **Event Hubs**
2. Select your **Subscription**, select ***anomaly-detection-rg*** as a **Resource group**, and select ***North Europe*** as **Location**
3. Enter a unique name for the namespace, and then select **Review + Create**
4. On the **Review + Create** page, select **Create**.

Graphical user interface, text, application, email

Description automatically generated

Once the deployment is complete, select **Go to resource** to open the Event Hubs Namespace. We will create two event hubs: ***eventhub1*** as a data source for Time Series Insights, and ***eventhub2*** as a data source for Azure Functions.

1. Select **Event Hubs** tab and click **+ Event Hub**
2. Enter ***eventhub1*** as a Name and click **Create.**

Graphical user interface, text, application, email

Description automatically generated

1. Select **Event Hubs** tab and click **+ Event Hub**
2. Enter ***eventhub2*** as a Name and click **Create**.

Graphical user interface, text, application, email

Description automatically generated

In order to provide access to the Time Series Insights, we will need to create a Shared Access Policy.

1. Select **Shared access policies** tab and click **+ Add**
2. Enter ***eventHubSharedAccessPolicy*** as a **Policy name**
3. Check **Manage** checkbox and click **Create**.

Graphical user interface, text, application, chat or text message

Description automatically generated

Now we have two event hubs, which will be used as Stream Analytics job outputs.

Graphical user interface, application

Description automatically generated

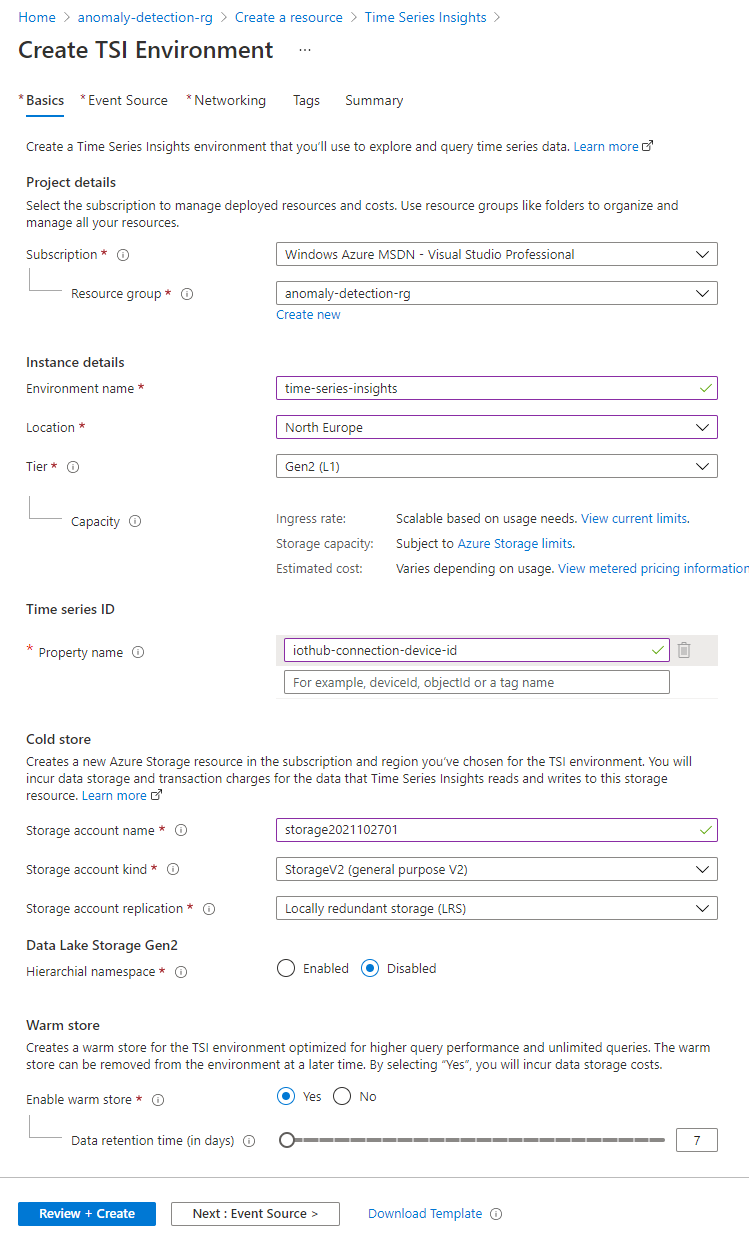
Graphical user interface, application

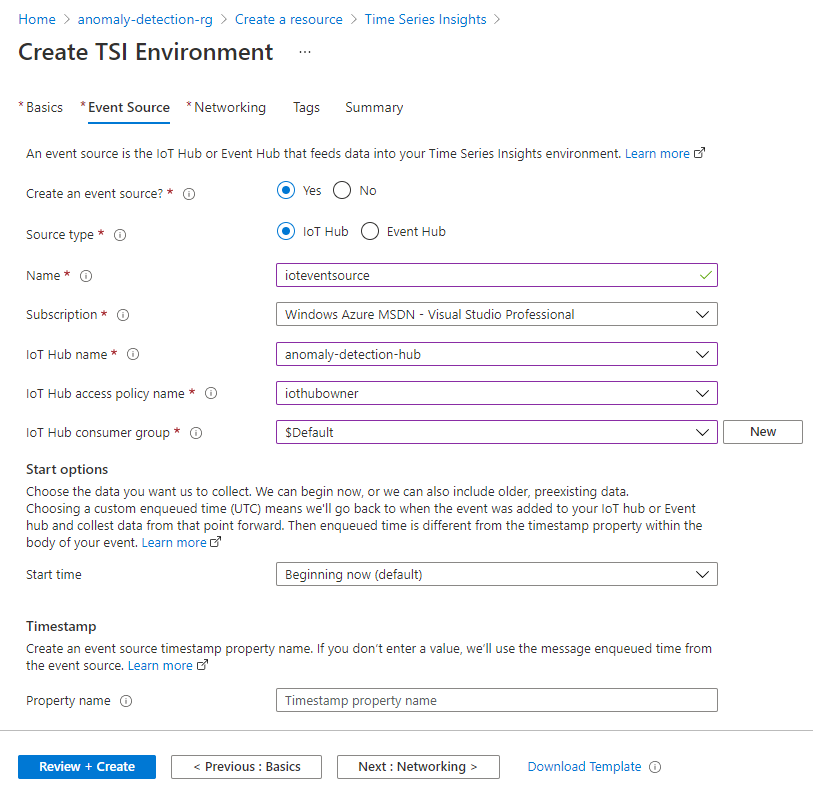
Description automatically generated

### Set-up for send-telemetry project

python --version  
cd it-workshop-anomaly-detection\source\send-telemetry  
python -m Ven env  
.\env\Scripts\activate  
# source env/bin/activate  
python -m pip install --upgrade pip  
pip3 install azure-ion-device  
pip3 install asyncio

## Create Time Series Insights





## Create Stream Analytics job

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

# Send telemetry data

cd it-workshop-anomaly-detection\source\send-telemetry  
.\env\Scripts\activate  
SET DPS\_ID\_SCOPE=0ne004269B3  
SET DEVICE\_ID=device001  
SET DPS\_PRIMARY\_KEY=cwB2AGMAXgBbAC8ASQA8AG8AJQBsAHwAeQBoACQAKQA3ADAAOABlADsAZQBpACsAQwBpAFAAdABfAEkAVQByAA==  
python telemetry.py

# References

1. [Quickstart: Set up the IoT Hub Device Provisioning Service with the Azure portal](https://docs.microsoft.com/en-us/azure/iot-dps/quick-setup-auto-provision)
2. [What is Azure IoT Hub Device Provisioning Service?](https://docs.microsoft.com/en-us/azure/iot-dps/about-iot-dps)
3. [How to provision devices using symmetric key enrollment groups](https://docs.microsoft.com/en-us/azure/iot-dps/how-to-legacy-device-symm-key?tabs=windows)
4. [Quickstart: Create a Stream Analytics job by using the Azure portal](https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-quick-create-portal)
5. [Anomaly detection in Azure Stream Analytics](https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-machine-learning-anomaly-detection)
6. [AnomalyDetection\_SpikeAndDip Azure Stream Analytics function](https://docs.microsoft.com/en-us/stream-analytics-query/anomalydetection-spikeanddip-azure-stream-analytics)
7. [Stream Analytics Query Language Reference](https://docs.microsoft.com/en-us/stream-analytics-query/stream-analytics-query-language-reference)
8. [Create an Azure Time Series Insights Gen2 environment using the Azure portal](https://docs.microsoft.com/en-us/azure/time-series-insights/how-to-create-environment-using-portal)