

Step 6 Consume Data from IoT Hub in Azure

IoT Hub supports several ways to consume data.

IoT Hub Client SDK

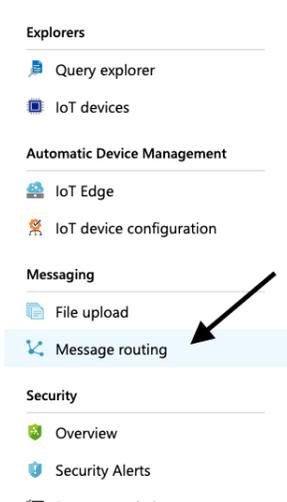
IoT Hub provides you with a client libraries for all major languages to push and pull data from IoT Hub see the link for more information: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-sdks>

Azure Cloud Services & BloB Storage

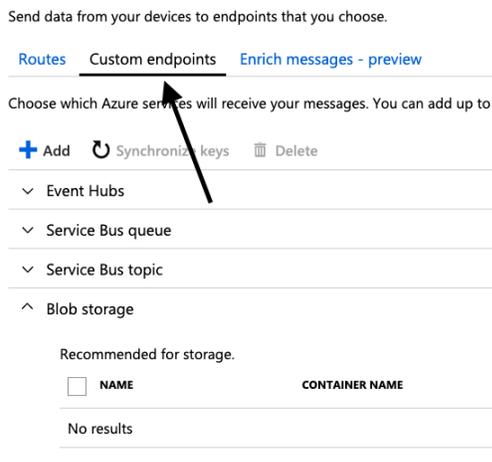
IoT Hub has full integration with all major Azure Cloud services, in this section we will focus on **BloB Storage**.

AzureBlobStorage is a scalable, cost-effective cloud storage for all your unstructured data. Pay only for what you use, and save money compared with on-premises storage options. Choose from among four storage tiers, depending on how often you'll access the data. Store performance-sensitive data in Premium, frequently accessed data in Hot, infrequently accessed data in Cool, and rarely accessed data in Archive.

1. Go to your Azure Portal and click on on your IoT Hub.
2. In the left click on **Message Routing**.



3. Then on the Custom endpoints tab.



4. Click Add button to add a new custom endpoint (where the messages will be routed).
5. And choose Blob Storage.

Choose which Azure services will receive your messages. You can a

+ Add Synchronize keys Delete

- Event hubs
- Service bus queue
- Service bus topic
- Blob storage** ←

Recommended for storage.

6. Give a unique name to this endpoint.

Add a storage endpoint

Route your telemetry and device messages to Azure Storage as blobs.

* Endpoint name ⓘ

niagara-events-endpoint ←

Azure Storage account and container

Create a new container, or choose an existing one that shares a subscription with this IoT hub.

Azure Storage container

[Pick a container](#)

Batch frequency ⓘ



Chunk size window ⓘ



Encoding ⓘ

AVRO JSON

* Blob file name format ⓘ

{iothub}/{partition}/{YYYY}/{MM}/{DD}/{HH}/{mm}

The format must contain {iothub}, {partition}, {YYYY}, {MM}, {DD}, {HH} and {mm} in any order.

If multiple files are created within the same minute, the filename format would be btbiothub/0/2019/07/10/10/47-01.

7. Pick a storage container.

Add a storage endpoint

Route your telemetry and device messages to Azure Storage as blobs.

* Endpoint name ⓘ

niagara-events-endpoint

Azure Storage account and container

Create a new container, or choose an existing one that shares a subscription with this IoT hub.

Azure Storage container

Pick a container

Batch frequency ⓘ

Chunk size window ⓘ

Encoding ⓘ

AVRO

JSON

* Blob file name format ⓘ

{iothub}/{partition}/{YYYY}/{MM}/{DD}/{HH}/{mm}

The format must contain {iothub}, {partition}, {YYYY}, {MM}, {DD}, {HH} and {mm} in any order.

If multiple files are created within the same minute, the filename format would be btibiothub/0/2019/07/10/10/47-01.

8. Click on + Storage Account.

Storage accounts

+ Storage account

 Refresh

Search storage accounts

NAME

No storage account found.

9. Give it a unique name and click OK.

* Name
btibniagarablob  
.core.windows.net

Account kind ⓘ
Storage (general purpose v1) ▼

Performance ⓘ
Standard Premium

Replication ⓘ
Locally-redundant storage (LRS) ▼

* Location
(US) East US 

10. Then chose the storage account you created.

Storage accounts

 Storage account  Refresh

Search storage accounts

NAME

btibniagara 

11. And Create a container for data.

Containers
btibniagara

+ Container Refresh

New container

* Name
niagara-events-data

Public access level ⓘ
Private (no anonymous access)

OK Cancel

12. Choose the container then hit Select.

Containers
btibniagara

+ Container Refresh

Search containers by prefix

NAME
niagara-events-data

Select

13. Choose the data format and click on create.

Add a storage endpoint

Route your telemetry and device messages to Azure Storage as blobs.

* Endpoint name 

[Azure Storage account and container](#)

Create a new container, or choose an existing one that shares a subscription with this IoT hub.

Azure Storage container

<https://btibniagara.blob.core.windows.net/niagara-events-data>

[Pick a container](#)

Batch frequency 


Chunk size window 


Encoding 
 AVRO JSON 

* Blob file name format 

The format must contain {iothub}, {partition}, {YYYY}, {MM}, {DD}, {HH} and {mm} in any order.

If multiple files are created within the same minute, the filename format would be btbiothub/0/

[Create](#) 

14. Your endpoint is created.

Send data from your devices to endpoints that you choose.

[Routes](#) [Custom endpoints](#) [Enrich messages - preview](#)

Choose which Azure services will receive your messages. You can add up to 10 endpoints to an IoT hub.

[+ Add](#) [Synchronize keys](#) [Delete](#)

Event Hubs

Service Bus queue

Service Bus topic

^ Blob storage

Recommended for storage.

<input type="checkbox"/>	NAME	CONTAINER NAME	ENCODING FORMAT
<input checked="" type="checkbox"/>	niagara-events-endpoint	niagara-events-data	JSON



15. Now we will redirect all events to this endpoint.

16. Go to the route tab and click add.

Send data from your devices to endpoints that you choose.

Routes Custom endpoints Enrich messages - preview

Create an endpoint, and then add a route (you can add up to 100 from each I

Disable fallback route

+ Add Test all routes Delete

NAME DATA SOURCE

No results

17. Give your route a name. choose the endpoint we created and choose **Device Telemetry Message** then hit **Save**.

Add a route

* Name niagara-events-route

* Endpoint niagara-events-endpoint

* Data source Device Telemetry Messages

* Enable route Enable Disable

Create a query to filter messages before data is routed to an endpoint. [Learn more](#)

Routing query

1 true

Test

Save

18. Congratulation now, all the telemetry data are stored on the Azure Blob Storage.

19. For devices/points tags follow the same procedure and in the data source option choose **Device Twin Change Events**.

Name
niagara-events-route

* Endpoint ⓘ

niagara-events-endpoint

* Data source ⓘ

- Device Twin Change Events
- Device Telemetry Messages
- Device Twin Change Events 
- Device Lifecycle Events

Create a query to filter messages before data is routed to an endpoint. [Learn more](#)

Routing query ⓘ

```
1 true
```