Configure dataflow endpoints

Article • 11/19/2024

> ### ⓘ Important
>
> This page includes instructions for managing Azure IoT Operations components using Kubernetes deployment manifests, which is in **preview**. This feature is provided with [several limitations](#), and shouldn't be used for production workloads.
>
> See the [Supplemental Terms of Use for Microsoft Azure Previews](#) for legal terms that apply to Azure features that are in beta, preview, or otherwise not yet released into general availability.

To get started with dataflows, first create dataflow endpoints. A dataflow endpoint is the connection point for the dataflow. You can use an endpoint as a source or destination for the dataflow. Some endpoint types can be used as both sources and destinations, while others are for destinations only. A dataflow needs at least one source endpoint and one destination endpoint.

Use the following table to choose the endpoint type to configure:

⌞⌝ **Expand table**

Endpoint type	Description	Can be used as a source	Can be used as a destination
MQTT	For bi-directional messaging with MQTT brokers, including the one built-in to Azure IoT Operations and Event Grid.	Yes	Yes
Kafka	For bi-directional messaging with Kafka brokers, including Azure Event Hubs.	Yes	Yes
Data Lake	For uploading data to Azure Data Lake Gen2 storage accounts.	No	Yes
Microsoft Fabric OneLake	For uploading data to Microsoft Fabric OneLake lakehouses.	No	Yes

Endpoint type	Description	Can be used as a source	Can be used as a destination
Azure Data Explorer	For uploading data to Azure Data Explorer databases.	No	Yes
Local storage	For sending data to a locally available persistent volume, through which you can upload data via Azure Container Storage enabled by Azure Arc edge volumes.	No	Yes

> ⓘ **Important**
>
> Storage endpoints require a **schema for serialization**. To use dataflow with Microsoft Fabric OneLake, Azure Data Lake Storage, Azure Data Explorer, or Local Storage, you must **specify a schema reference**.
>
> To generate the schema from a sample data file, use the **Schema Gen Helper** .

Dataflows must use local MQTT broker endpoint

When you create a dataflow, you specify the source and destination endpoints. The dataflow moves data from the source endpoint to the destination endpoint. You can use the same endpoint for multiple dataflows, and you can use the same endpoint as both the source and destination in a dataflow.

However, using custom endpoints as both the source and destination in a dataflow isn't supported. This restriction means the built-in MQTT broker in Azure IoT Operations must be at least one endpoint. It can be either the source, destination, or both. To avoid dataflow deployment failures, use the default MQTT dataflow endpoint as either the source or destination for every dataflow.

The specific requirement is each dataflow must have either the source or destination configured with an MQTT endpoint that has the host `aio-broker`. So it's not strictly required to use the default endpoint, and you can create additional dataflow endpoints pointing to the local MQTT broker as long as the host is `aio-broker`. However, to avoid confusion and manageability issues, the default endpoint is the recommended approach.

The following table shows the supported scenarios:

⌞⌝ **Expand table**

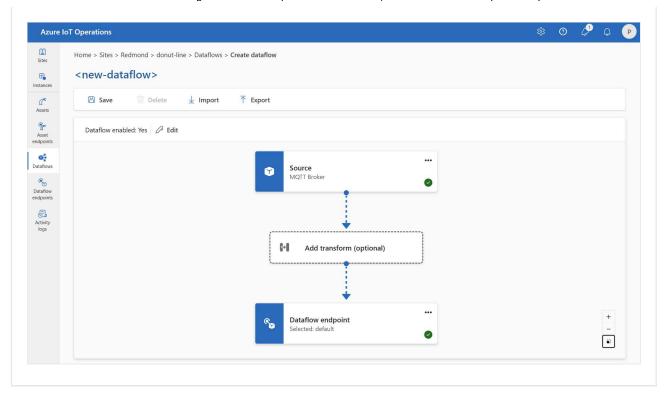| Scenario	Supported
Default endpoint as source	Yes
Default endpoint as destination	Yes
Custom endpoint as source	Yes, if destination is default endpoint or an MQTT endpoint with host `aio-broker`
Custom endpoint as destination	Yes, if source is default endpoint or an MQTT endpoint with host `aio-broker`
Custom endpoint as source and destination	No, unless one of them is an MQTT endpoints with host `aio-broker`

Reuse endpoints

Think of each dataflow endpoint as a bundle of configuration settings that contains where the data should come from or go to (the `host` value), how to authenticate with the endpoint, and other settings like TLS configuration or batching preference. So you just need to create it once and then you can reuse it in multiple dataflows where these settings would be the same.
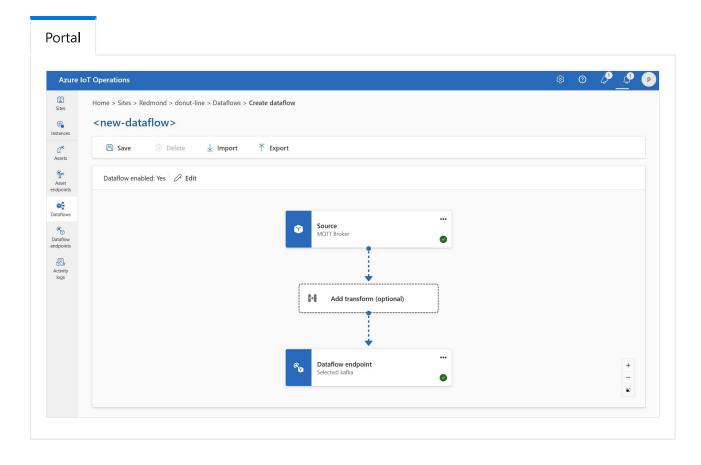
To make it easier to reuse endpoints, the MQTT or Kafka topic filter isn't part of the endpoint configuration. Instead, you specify the topic filter in the dataflow configuration. This means you can use the same endpoint for multiple dataflows that use different topic filters.

For example, you can use the default MQTT broker dataflow endpoint. You can use it for both the source and destination with different topic filters:

Portal

Similarly, you can create multiple dataflows that use the same MQTT endpoint for other endpoints and topics. For example, you can use the same MQTT endpoint for a dataflow that sends data to an Event Hubs endpoint.

Portal

Similar to the MQTT example, you can create multiple dataflows that use the same Kafka endpoint for different topics, or the same Data Lake endpoint for different tables.

Next steps

Create a dataflow endpoint:

- MQTT or Event Grid
- Kafka or Event Hubs
- Data Lake
- Microsoft Fabric OneLake
- Local storage

Feedback

Was this page helpful?   👍 Yes   👎 No

Provide product feedback | Get help at Microsoft Q&A