Prerequisites:

1. Apache NiFi Installed: Ensure you have Apache NiFi installed and running.
2. Access to ActiveMQ: Ensure you have access to your ActiveMQ instance and the necessary credentials.
3. Azure Event Hub: Ensure you have an Azure Event Hub created with the necessary credentials.

Steps to Configure Apache NiFi:

1. Install and Launch Apache NiFi:

* Download and install Apache NiFi from the official website.
* Start NiFi by running the nifi.sh start (Linux) or nifi.bat start (Windows) script.
* Access the NiFi UI at <http://localhost:8080/nifi>.

2. Set Up ActiveMQ Consumer in NiFi:

1. Add a JMS Connection Factory Controller Service:

* In the NiFi UI, go to the Controller Settings (gear icon on the top right).
* Under the Controller Services tab, add a new JMSConnectionFactoryProvider service.
* Configure it with your ActiveMQ connection details:

json

Copy code

Broker URI: `tcp://broker-url:61616`   
JMS Client Library: `path-to-activemq-client.jar` 

1. Add a ConsumeJMS Processor:

* Drag and drop a ConsumeJMS processor onto the NiFi canvas.
* Configure the processor to use the JMSConnectionFactoryProvider created earlier.
* Set the destination name (queue or topic) to the ActiveMQ destination from which you want to consume messages.
* Configure other properties like Destination Type, Client ID, Subscription Name as needed.

3. Set Up Azure Event Hub Publisher in NiFi:

1. Add an Azure Event Hub Connection Factory Controller Service:

* Under the Controller Services tab, add a new AzureEventHubConnectionFactory service.
* Configure it with your Azure Event Hub connection string and Event Hub name.

1. Add a PublishAzureEventHub Processor:

* Drag and drop a PublishAzureEventHub processor onto the NiFi canvas.
* Configure the processor to use the AzureEventHubConnectionFactory created earlier.
* Set the Event Hub Name and other properties like Message Body to be sent.

4. Link the Processors:

* Connect the ConsumeJMS processor to the PublishAzureEventHub processor.
* Configure the connection with appropriate relationships (e.g., success).

5. Configure Error Handling (Optional but Recommended):

* Add a LogMessage processor to handle failures.
* Connect the failure relationship from ConsumeJMS and PublishAzureEventHub to the LogMessage processor.

6. Start the Data Flow:

* Enable the JMSConnectionFactoryProvider and AzureEventHubConnectionFactory controller services.
* Start the ConsumeJMS and PublishAzureEventHub processors.
* Monitor the data flow through the NiFi UI.

Detailed Configuration Example:

JMSConnectionFactoryProvider:

* Broker URI: tcp://your-activemq-broker:61616
* JMS Client Library: /path/to/activemq-all-x.x.x.jar

ConsumeJMS Processor:

* JMS Connection Factory Controller Service: JMSConnectionFactoryProvider
* Destination Name: yourQueueOrTopicName
* Destination Type: QUEUE or TOPIC
* Client ID: (if using durable subscriptions for topics)
* Subscription Name: (if using durable subscriptions for topics)

AzureEventHubConnectionFactory:

* Connection String: Endpoint=sb://your-event-hub.servicebus.windows.net/;SharedAccessKeyName=yourKeyName;SharedAccessKey=yourKeyValue;EntityPath=yourEventHub
* Event Hub Name: yourEventHubName

PublishAzureEventHub Processor:

* Event Hub Name: yourEventHubName
* Message Body: ${field\_containing\_message\_body} (or use FlowFile Content if entire message is in content)

Monitoring and Maintenance:

* Use the NiFi UI to monitor data flow, check for errors, and adjust configurations as needed.
* Regularly check and update connection details and security credentials.

This configuration should help you achieve a seamless data transfer from ActiveMQ to Azure Event Hub with minimal coding and efficient handling of potentially high data volumes.