

Using the new Kafka Nodes in IBM Integration Bus 10.0.0.7

SanjayNagchowdhury

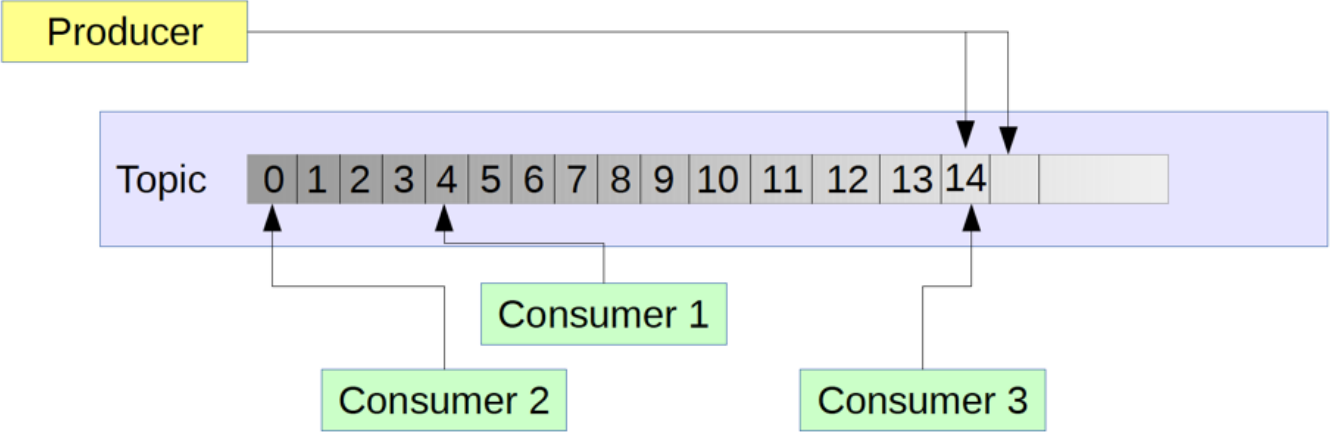
Published on November 25, 2016 / Updated on January 4, 2017

Introduction

In this article, I will describe the new KafkaProducer and KafkaConsumer nodes that have been provided in IBM Integration Bus 10.0.0.7. I will show how you can publish messages to a topic on IBM Message Hub and consume messages from that topic.

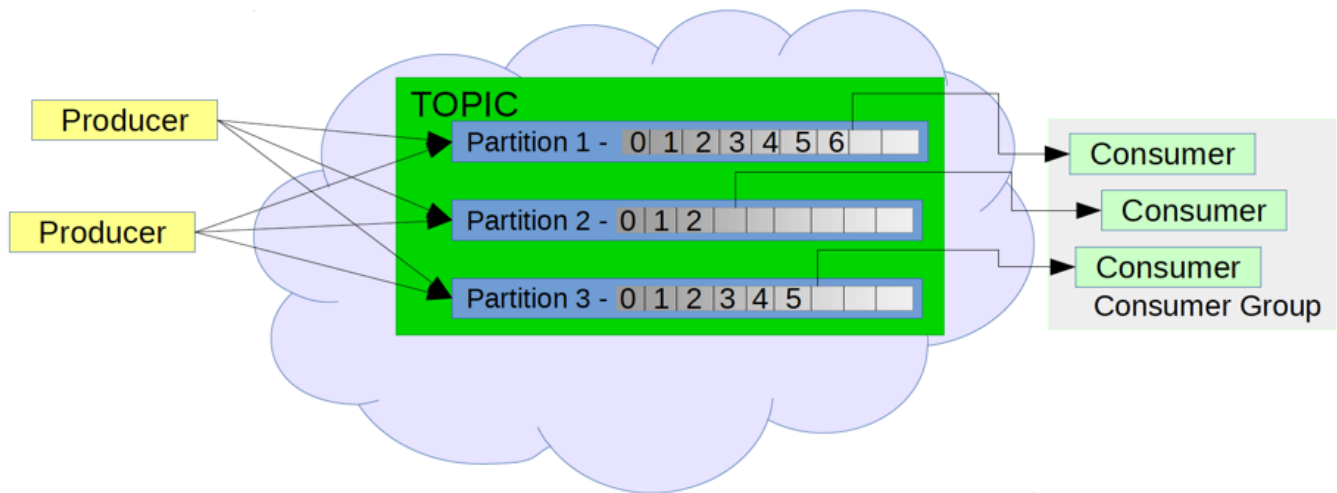
Overview of Apache Kafka

Apache Kafka is a distributed streaming platform which allows applications to publish and subscribe to streams of records. Kafka is architected as a cluster of one or more servers. A stream of records is called a 'topic'. A 'Producer' can publish messages to a topic. A 'Consumer' can subscribe to a topic and consume the messages that were published.



The above diagram shows that when messages are published on a Kafka topic, the messages are added to the tail of the log. As each message is published on the topic, it is identified by the 'offset' of the message in the log. All consumers retrieve messages from the same log, and as messages are consumed they are not destroyed but remain in the log for the pre-defined retention period. The offset is used by consumers to identify their position when receiving messages. The retention period can be defined by time, or by log size.

When a consumer starts, it must tell Kafka from where in the commit log it wishes to start receiving messages. This can either be the earliest available message or the latest message. When a consumer disconnects and then re-connects and it wishes to see all messages published while it was disconnected, then it must remember the offset of the last message it consumed.



To balance load and achieve scalability, Kafka allows topics to be divided into ‘partitions’. Each partition is an ordered, immutable sequence of records that is continually appended to in a structured commit log.

By defining a topic to have multiple partitions, messages published on the topic will be distributed equally amongst the partitions. Consumers in a consumer group will also split equally across the partitions. Kafka manages the distribution of messages and the assignment of consumers.

For more information, see the [Apache Kafka](#) documentation.

IBM Message Hub

IBM Message Hub is a scalable, distributed, high throughput messaging system, built on Apache Kafka. This can be used to stream data to analytics to realize powerful insights. IBM Message Hub uses a set of credentials which Producer and Consumer applications must use to publish or consume messages from a topic. For more information see [IBM Message Hub](#).

New Kafka Nodes

In IBM Integration Bus 10.0.0.7, we have provided 2 new Kafka nodes which can be used for integration solutions which require interactions with topics on a Kafka Cluster. The nodes are in a new Kafka drawer in the toolkit.



The KafkaProducer node allows you to publish messages to a topic on a Kafka server.



The KafkaConsumer node allows you to consume messages from a topic on a Kafka server.

KafkaProducer Node

The Kafka Producer node can be used to publish messages from a message flow to a Kafka topic. The node publishes messages non-transactionally to the Kafka topic. The effect of this on a message flow is that if an error occurs downstream in your message flow which causes the message to be rolled back, the publication of the message to the Kafka topic is not rolled back. You can use an acks property to specify if confirmation is needed from the Kafka server for the message that was published. If you wish, you can set acks to 0 which means that the KafkaProducer node does not wait for any acknowledgement.

KafkaConsumer Node

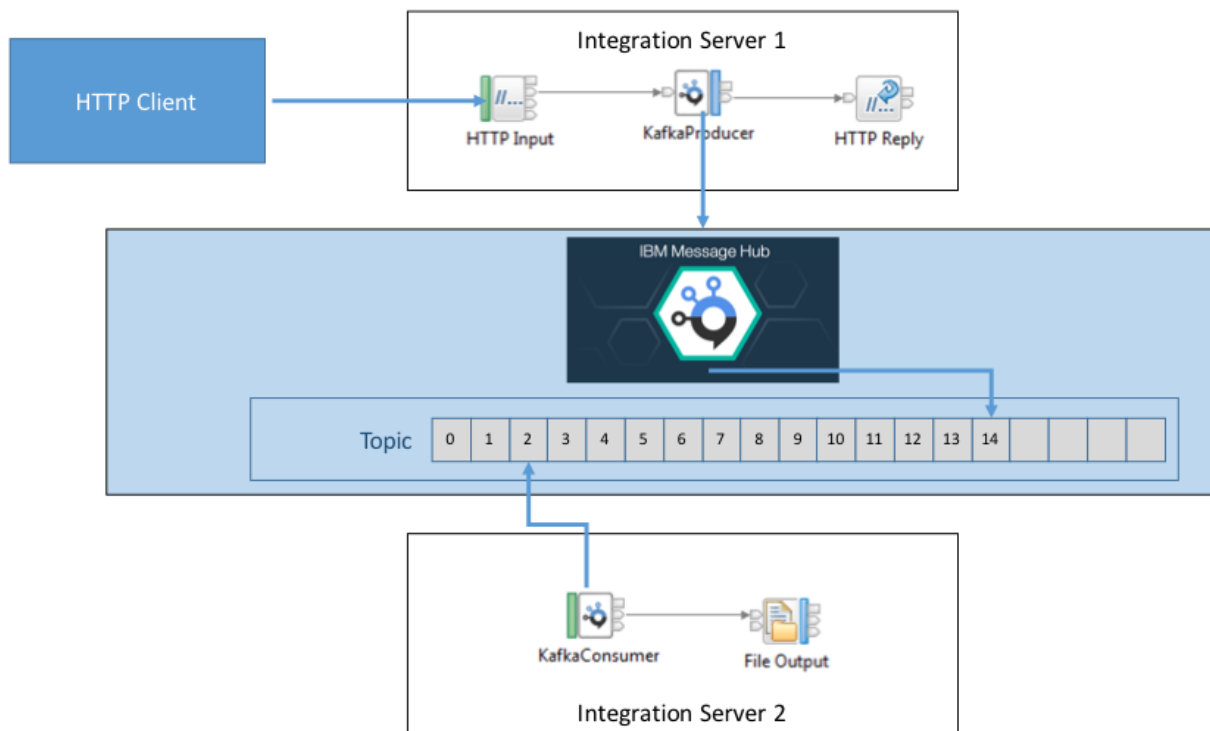
The KafkaConsumer node can be used to subscribe to a Kafka topic so that messages can be consumed and processed by the message flow. A KafkaConsumer node can subscribe to only a single Kafka topic. If the topic contains multiple partitions, then the KafkaConsumer node will receive messages from any of those partitions. The KafkaConsumer node can be defined to be part of a Consumer Group. This allows you to increase concurrency. If you deploy multiple KafkaConsumer nodes that share the same Group ID, Kafka will ensure that messages that are published on the topic are shared across the consumer group.

Security

All Kafka nodes that are deployed to the same integration server must use the same set of credentials to authenticate to the Kafka cluster. To save the credentials that the Kafka nodes will use to connect to the Kafka cluster, you use the `mqsisetdbparms` command to configure the resource name in the form `kafka::KAFKA::integrationServerName`.

Example using Kafka nodes with IBM Message Hub

The diagram below shows the interaction between two IIB message flows and IBM Message Hub using the KafkaProducer and KafkaConsumer nodes.



The first message flow will receive HTTP messages and publish it to a topic on Message Hub using the KafkaProducer node. The second message flow will subscribe to the same topic on Message Hub and consume messages from it and write them out to a file on the file system

The following steps will take you through how to construct the scenario.

Flow Development Steps

Step 1: Add the Message Hub service in Bluemix

Open up Bluemix: <https://console.ng.bluemix.net/>

Search for Message Hub and add it.

Application Services
Deliver new web and mobile apps.

- Blockchain**
Utilize IBM's Blockchain Technology within Bluemix
IBM
- Business Rules**
Enables developers to spend less time recoding and testing when the business
IBM
- Data Cache**
Improve the performance and user experience of web applications by retrieval
IBM **Deprecated**
- Message Hub**
IBM Message Hub is a scalable, distributed, high throughput message bus to unite you
IBM
- Session Cache**
Improve application resiliency by storing session state information across many HTTP
IBM
- WebSphere Application Server**
Allows you to quickly get up and running on a pre-configured WebSphere Application
IBM
- Workload Scheduler**
Automate your tasks to run one time or on recurring schedules. Far beyond Cron, ex
IBM
- APIs from Pitney Bowes**
Add enterprise-class geodata and commerce technology your application
Third Party
- box**
Powering Content and data for your application. Whether you are building a lin
Third Party

Step 2: Create a topic in Message Hub

Open the Message Hub service by clicking on it in your list of services.

Create a topic in Message Hub by clicking the button shown below:

Message Hub-6x

[Manage](#) [Service Credentials](#) [Connections](#)

Topics

Filter Topics...

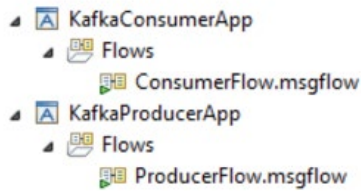
Partitions: 0 used / 100 maximum

Topic Names	Partitions	Retention (hours)
No Topics Found		

Topic Names	Partitions	Retention (hours)
Topic 'MyTopic' created.		
<input type="checkbox"/> MyTopic	1	24

Step 5: Create the message flows

Create two separate applications called KafkaProducerApp and KafkaConsumerApp

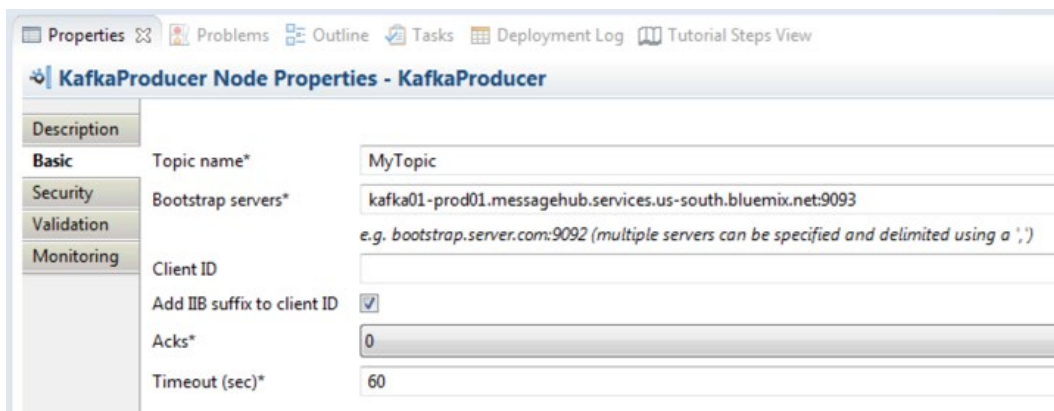


In the KafkaProducerApp, add a message flow called ProducerFlow and add these nodes:

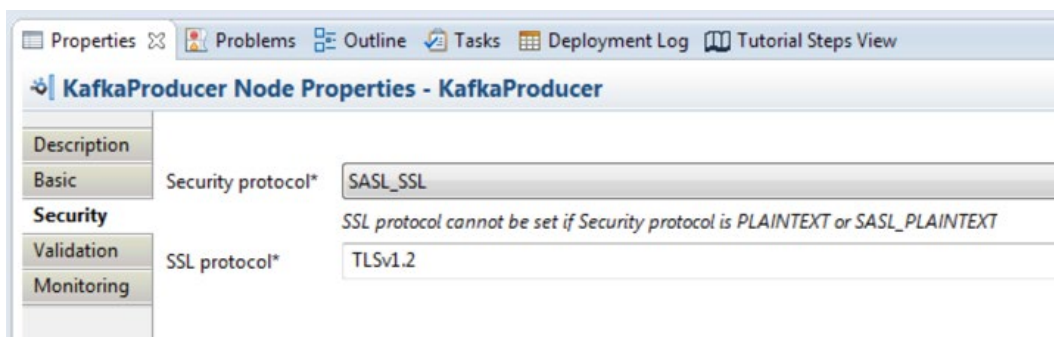


Configure the KafkaProducer node as follows:

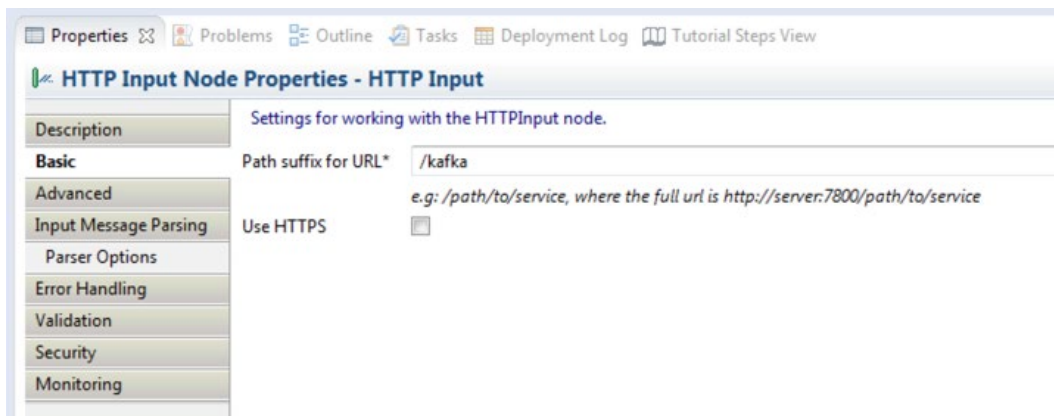
On the Basic Tab, specify the topic name and bootstrap server, using the value of the server that you copied from the Message Hub service credentials.



On the Security Tab, specify 'SASL_SSL' as the Security protocol and leave the default value of 'TLSv1.2' for the SSL protocol.



Specify a Path suffix for the URL for the HTTP Input node.

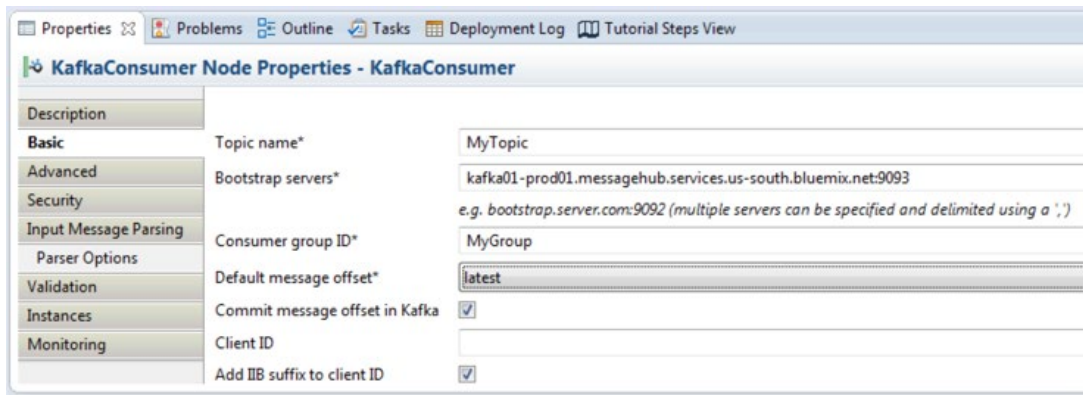


In the KafkaConsumerApp, add a message flow called ConsumerFlow and add these nodes:

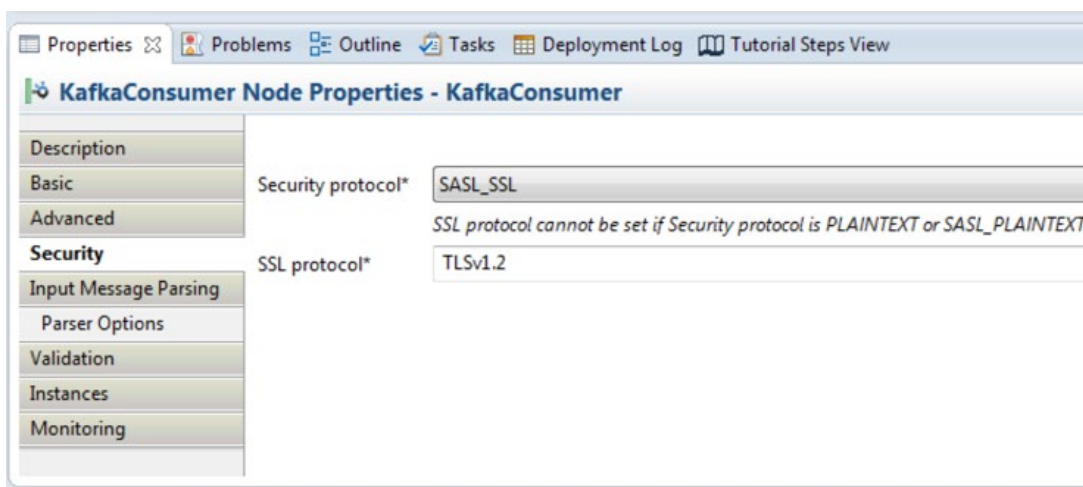


Configure the KafkaConsumer node as follows:

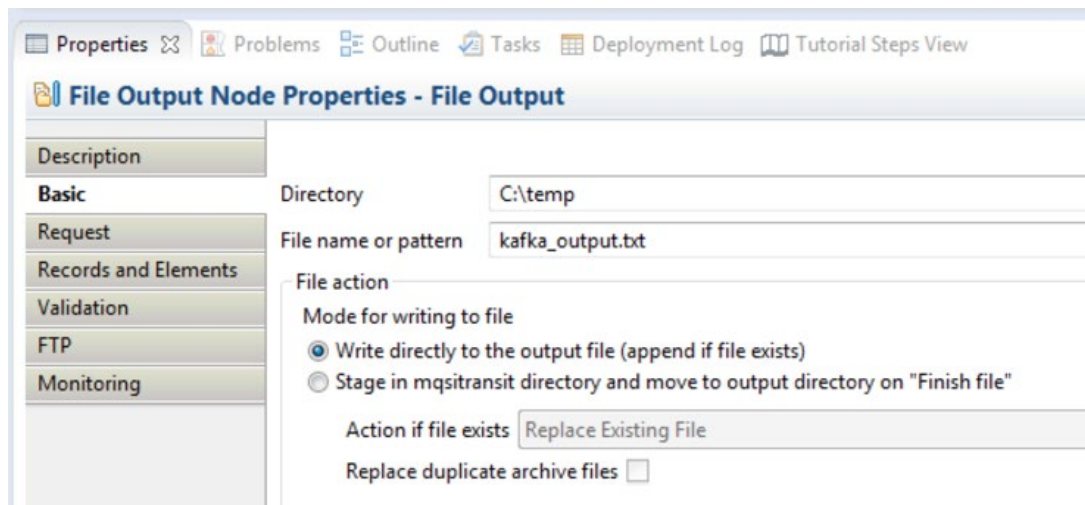
On the Basic Tab, specify the topic name, bootstrap server and a value for the Consumer group ID. Specify 'latest' as the Default message offset.



On the Security Tab, specify 'SASL_SSL' as the Security protocol and leave the default value of 'TLSv1.2' for the SSL protocol.



Specify a directory and file name on the FileOutput node.

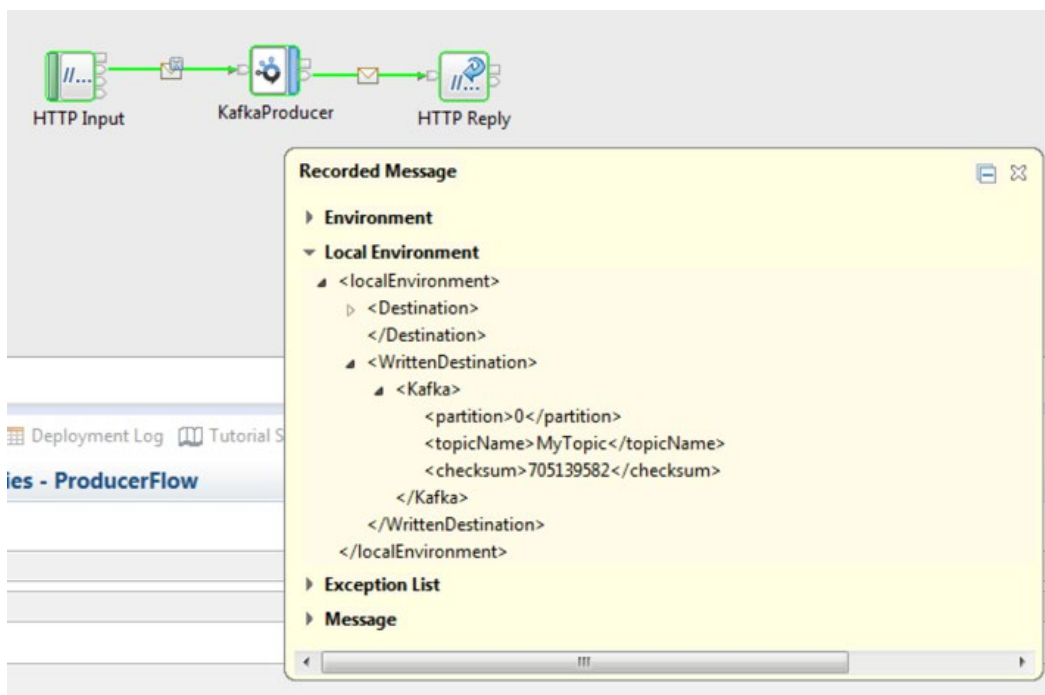


Step 6: Deploy the applications

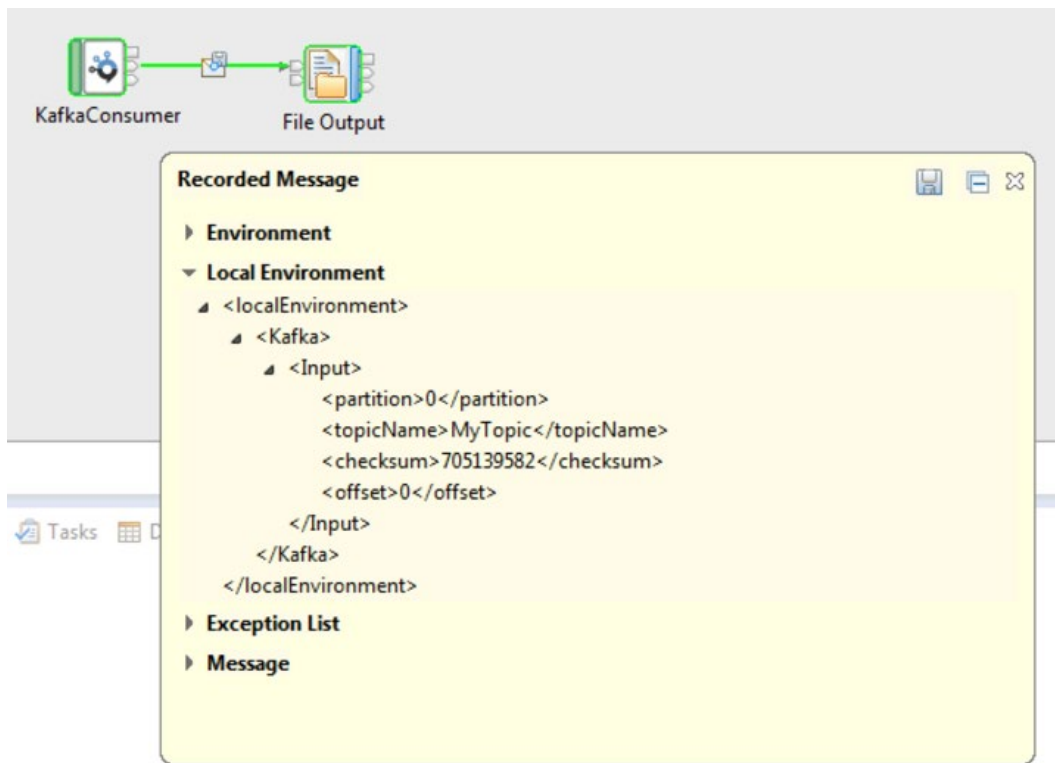
Deploy the ProducerFlow to Server1 and the ConsumerFlow to Server2 using the Flow Exerciser.

Send in a message.

If you click on the point after the KafkaProducer node, you will see details for where the message was published to in the WrittenDestination section of the LocalEnvironment.



If you click on the point after the KafkaConsumer node, you will see details for where the message was consumed from in the Kafka.Input section of the LocalEnvironment.



Notice that the offset is 0, indicating that this is the first message that has been consumed.

You will see that the file that you specified in the FileOutput node has been created with the text that you sent to the Producer flow.

Step 7: Check Activity Log in WebUI

Open the IIB Web User Interface.

Select the ProducerFlow and click on the Activity Log tab.

You will see an entry for the message that was published.

ProducerFlow - Message Flow

Overview | Statistics | Operational Policy | **Activity Log**

Total: 4 Clear Filter Refresh

Timestamp	Message	Thread ID	Node	Resource Manager	Message Detail
2016-11-24 00:29:05.000 GMT St	BIP13069I	1505308	KafkaProducer	Kafka	Published a message to the Kafka server.
2016-11-24 00:29:05.000 GMT St	BIP12064I	1505308	KafkaProducer	Kafka	Sent data to 'Kafka'.
2016-11-24 00:29:05.000 GMT St	BIP11506I	1505308	HTTP Input		Committed a local transaction.
2016-11-24 00:29:05.000 GMT St	BIP11302I	1505308	HTTP Input	Parsers	There are '15' parsers cached on this thread.

Published a message to the Kafka server.

Topic name: 'MyTopic'
 Partition: '0'
 Offset: ''
 Client ID: 'Server1-TESTNODE_sanjayn'

Now select the ConsumerFlow and click on the Activity Log tab.

You will see an entry for the message that was consumed.

ConsumerFlow - Message Flow

Overview | Statistics | Operational Policy | Activity Log

Total: 7 Clear Filter Refresh

Timestamp	Message	Message Detail	Tags
2016-11-24 00:29:05.000 GMT St	BIP13068I	Received a message from the Kafka server.	NODETYPE=INPUT,C
2016-11-24 00:29:05.000 GMT St	BIP12065I	Received data from 'Kafka'.	NODETYPE=INPUT
2016-11-24 00:29:05.000 GMT St	BIP13011I	Appended a record directly to the output file 'kafka_output.xml'.	NODETYPE=OUTPUT
2016-11-24 00:29:05.000 GMT St	BIP11506I	Committed a local transaction.	
2016-11-24 00:29:05.000 GMT St	BIP11302I	There are '9' parsers cached on this thread.	
2016-11-24 00:29:05.000 GMT St	BIP12066I	Waiting for data from 'Kafka'.	NODETYPE=INPUT
2016-11-24 00:28:26.000 GMT St	BIP12066I	Waiting for data from 'Kafka'.	NODETYPE=INPUT

Received a message from the Kafka server.

Topic name: 'MyTopic'
 Partition: '0'
 Offset: '0'
 Client ID: 'Server2-TESTNODE_sanjayn'
 Group ID: 'MyGroup'

Summary

In this article, I have described an overview of Apache Kafka and shown how to use the new KafkaProducer and KafkaConsumer nodes to publish and consume messages from a topic on IBM Message Hub. The Kafka nodes can also be used with any Kafka Server implementation. All Kafka nodes that are deployed to the same integration server must use the same set of credentials to authenticate to the Kafka cluster. IBM Message Hub uses SASL_SSL as the Security Protocol. For more information, see the [IBM Integration Bus v10 Knowledge Center](#).

Ben Thompson's recent article [Explore the new features of IBM Integration Bus 10.0.0.7](#) provides information about further enhancements that have been provided.

You may wish to watch my video, 'Introducing the new KafkaConsumer and KafkaProducer nodes' which demonstrates a scenario to show how they can be used with IBM Integration Bus.

TAGS KAFKA

SanjayNagchowdhury

14 comments on "Using the new Kafka Nodes in IBM Integration Bus 10.0.0.7"

Sarath March 27, 2020

Hi SanjayNagchowdhury,

Sorry my previous message was cryptic .Let me explain our scenarios.

- We are using IIB Kafka producer node to sending data to kafka
- Intermittently due a network issue we are getting timeout errors connecting kafka from IIB kafka producer node.
- By default, we don't have any property on IIB kafka producer node for doing retries

Where as in native kafka library has these additional properties for producer are used handle transient errors.

o retries

o delivery.timeout.ms

o request.timeout.ms

- Based on this article <https://www.ibm.com/support/pages/apar/IT23442>

We should be able configure additional properties supported by native library.

But we don't see this working for below properties.

o retries

o delivery.timeout.ms

o request.timeout.ms.

- Other properties like sasl.mechanism and others are getting picked up when we follow the instructions in above link.
- Is there any way to configure retries on transient or recoverable errors?

[Reply \(Edit\)](#)

Sarath March 19, 2020

we are trying to configure the following properties in producer.properties. But looks like these properties are considered in IIB kafka producer node.

retries

delivery.timeout.ms

request.timeout.ms

[Reply \(Edit\)](#)

SanjayNagchowdhury March 23, 2020

Hi,

I'm not quite sure what your question is. Could you re-phrase it please.

Thanks

Sanjay

[Reply \(Edit\)](#)

Raghavendra Channappa October 10, 2019

Hello,

I do not see an option in the producer node configuration to 'batch' multiple messages into one write to a Kafka topic. Unless I am missing something, producer node can write only one message at a time to Kafka topic. Can you please confirm.

[Reply \(Edit\)](#)

BenThompsonIBM October 25, 2019

Hi Raghavendra, That's correct there is no "batching" facility with KafkaProducer message flow node ... Generally the use case patterns we see for Kafka is more aligned to a larger volume of smaller messages .. and of course you can use the PROPAGATE function to send multiple tree propagations into the node. If there is a facility or particular property of the underlying Kafka java client (v2.2.0 is the level we are using now in the most recent fix packs of both IIB and ACE) which you'd like to see the node expose then please let us know! Cheers, Ben

[Reply \(Edit\)](#)

Abhinav August 03, 2018

Hi,

How can we dynamically configure the bootstrap servers and topic name instead of hardcoding it in KafkaProducer/Consumer node.

[Reply \(Edit\)](#)

BenThompsonIBM January 25, 2019

Hi Abhinav,

Although (from memory) the overrides were not all available when the Kafka nodes were initially released in IIBv10.0.0.7, some Kafka properties were subsequently made dynamically overrideable in a later fix pack, including the topic name used by the KafkaProducer node. More details in the Knowledge Center

here: https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.mft.doc/bz91060_.htm

Cheers,

Ben

[Reply \(Edit\)](#)

Vijay August 29, 2017

Hi,

When running the producer flow with the above configuration I get the below error

Error sending request to http "http://localhost:7800/publish"

SOAP-ENV:Server

BIP3113E: Exception detected in message flow KafkaProducer (integration node TESTNODE_root)

<http://localhost:7800/publishTopic>

Exception. BIP2230E: Error detected whilst processing a message in node 'KafkaProducer.KafkaProducer'. :

F:\build\S1000_slot1\S1000_P\src\DataFlowEngine\TemplateNodes\ImbOutputTemplateNode.cpp: 303:

ImbOutputTemplateNode::processMessageAssemblyToFailure: ComIbmOutputNode: KafkaProducer#FCMComposite_1_1

BIP3895E: Failed to initialise Kafka output connector. Reason "Failed to construct kafka producer". : ContainerServices.java: 138: throwException: :

What could have gone wrong???

[Reply \(Edit\)](#)

SanjayNagchowdhury September 01, 2017

Hi Vijay,

There isn't enough information in your comment for us to give you direct assistance. I suggest that you check the Event Log to see if there is further information which helps you diagnose where you have gone wrong. Otherwise, I advise you to raise a PMR and work with our Service team to resolve the issue.

Thanks

Sanjay

[Reply \(Edit\)](#)

Vijay September 08, 2017

Hi Sanjay

Thanks for taking time to respond to my query. Let me provide more details which I have been trying now.

Basically I wanted to send a message to my local q – make it transfer to message hub – use a consumer flow to pick it up

- 1) Secure gateway is successfully established between my laptop to Bluemix (running successfully with no errors)
- 2) Created a topic called MQLight in message hub
- 3) Created a bridge to my Local Q using the topic created above
- 4) Successfully the bridge is connected to my local q, I could see input q count as 1 in my explore
- 5) Created a Kafka consumer to listen to the MQLight topic (I am not sure if this will work, since the bridge says unidirectional, let me know if it has to be done otherwise..nevertheless there is connection issue to my message hub)
- 6) Copied the server, user and password from credentials for message hub and used in my Kafka consumer node
- 7) Also ran mqsisetdbparm to set the Kafka credentials to my integration server, the reportproperties also confirms the same (as per the article, but the error shows something like "KAFKA'. : login")

```
mqsisetdbparms MYNODE -n kafka::KAFKA::default -u 9kVhxRHGNiuFxxOf -p XXXXXXXXX
```

```
C:\Program Files\IBM\IIB\10.0.0.9>mqsisetdbparms MYNODE -n *
```

```
BIP8180I: The resource name 'jdbc::HRDB' has userID 'db2admin'.
```

```
BIP8180I: The resource name 'jdbc::HRDB_SecurityId' has userID 'db2admin'.
```

```
BIP8180I: The resource name 'kafka::KAFKA::default' has userID '9kVhxRHGNiuFxxOf'.
```

Below error is shown in event log

```
(MYNODE.default) Java node error: '[BIPmsgs:3883]BIP3883E: The Kafka client failed to connect to the Kafka server as the security credentials 'KAFKA' have not been configured for the Integration Server.
```

The Kafka client requested a userid and password in order to authenticate the connection to the Kafka server but no Kafka security credentials have been configured for the Integration Server.

Use the mqsisetdbparms command to configure the security credentials for the Integration server using the resource name 'KAFKA'. : login'.

Error message generated by user Java node.

Contact the node provider for further details.

[Reply \(Edit\)](#)

Lidia T. May 07, 2018

Hi. Did you could solve the problem? the same thing is happening to me.
Thank for your reply.

Regards.

[Reply \(Edit\)](#)

Trevor May 09, 2018

Hi, Assuming you are using IIB v10 the most common causes for this error are
– when the mqsisetdbparms command is run, the value of the '-n' parameter was specified incorrectly. For IIB v10 this is
mqsisetdbparms -n kafka::KAFKA:: -u -p
There is no validation on the name of the IntegrationServer specified so it is easy to make a mistake.
– And after the mqsisetdbparms command is run, the IntegrationServer must be restarted in order to pick up the changes.
Thanks, Trevor

[\(Edit\)](#)

Allen Schmutzler July 27, 2017

Sanjay... are there any performance considerations/implications or testing that has been done

[Reply \(Edit\)](#)

SanjayNagchowdhury July 31, 2017

Hi Allen,

We do not have performance benchmarks for the Kafka nodes. The maximum message size on IBM Message Hub (public service) is 1Mb. Kafka can handle bigger messages but requires Kafka broker and Kafka client configuration to support messages > 1Mb.

This knowledge center topic may be useful to you on how to process large input messages in IIB:

https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.mft.doc/bc23802_.htm

Thanks

Sanjay

[Reply \(Edit\)](#)