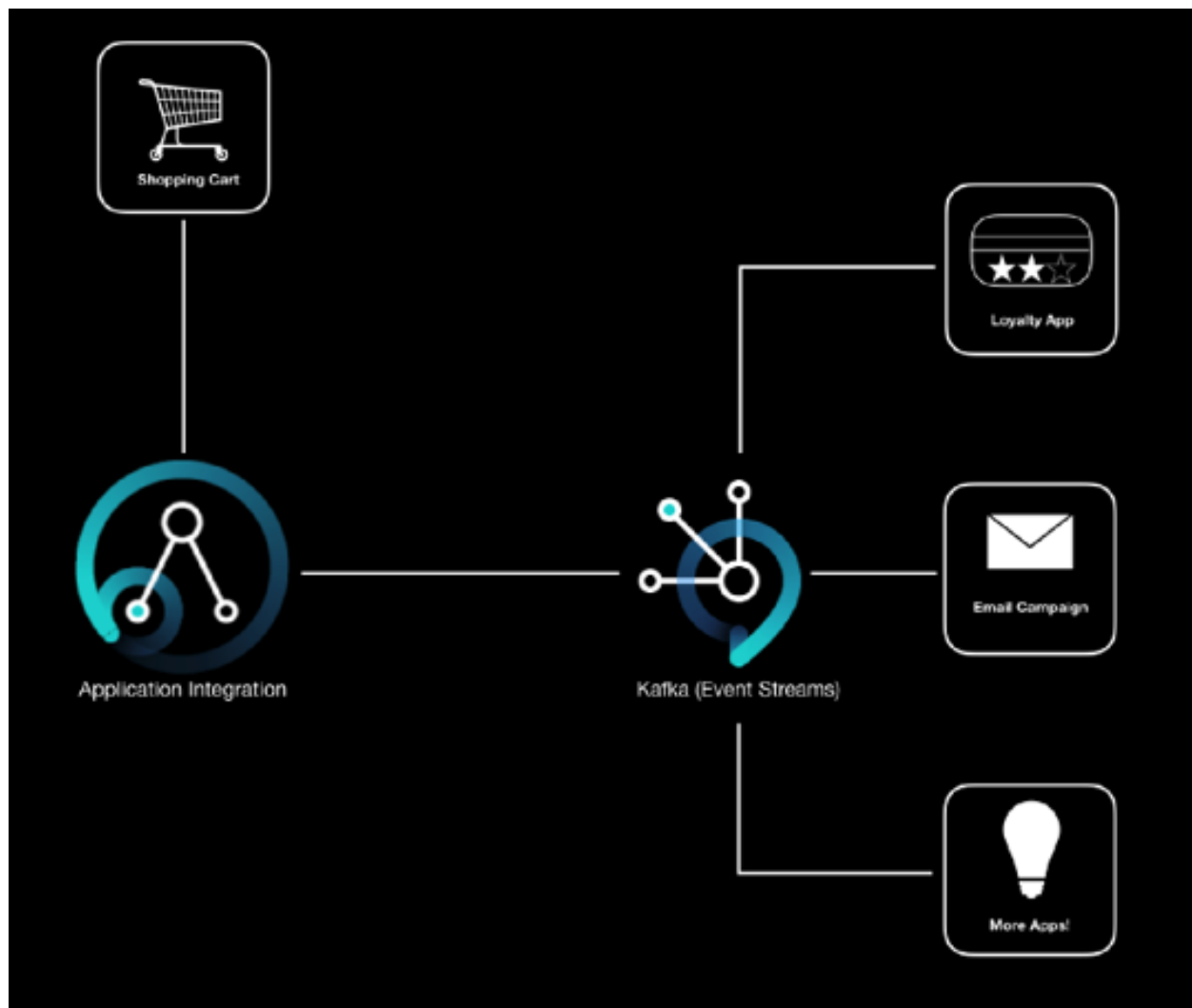


## **Augment existing business functions with new applications using Kafka**

The most interesting and impactful new applications in an enterprise are those that provide interactive experiences by reacting to existing systems carrying out a business function. In this tutorial, we take a look at an example from the retail industry starting with an existing API orchestrating the business function to "place an order". Let's say we have an existing API orchestrating the business function to "place an order", and when a customer places an order, we want to provide a real-time response - for example, reward the customer with points in the customer loyalty app or sign them up for an email nurture program or gamification experience. To do any of those actions, we need each order to emit an event. IBM Cloud Pak for Integration combines integration capabilities with Kafka-based IBM Event Streams to make the data available to cloud-native applications that can subscribe to the data and use it for a variety of business purposes



In this tutorial, you create a topic in IBM Event Streams, modify an integration flow to call an API, produce an event to a topic, and use a tracing tool to verify the message from App Connect Enterprise to Event Streams.

In this tutorial, you will explore the following key capabilities:

- Starting IBM Cloud Pak for Integration Environment
- Creating and configuring an Event Streams topic
- Configure App Connect Enterprise message flow using App Connect Enterprise toolkit
- Configuring App Connect Enterprise service
- Deploying App Connect BAR file on App Connect Enterprise Server
- Testing App Connect Enterprise API sending a message to Event Streams

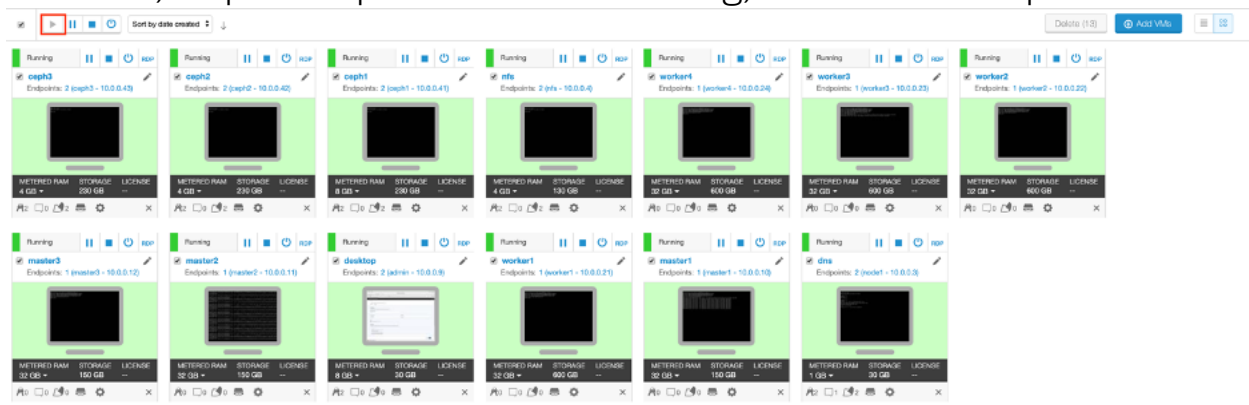
- Checking the message using Operations Dashboard (tracing).

## Task 1 - Preparing IBM Cloud Pak for Integration Environment

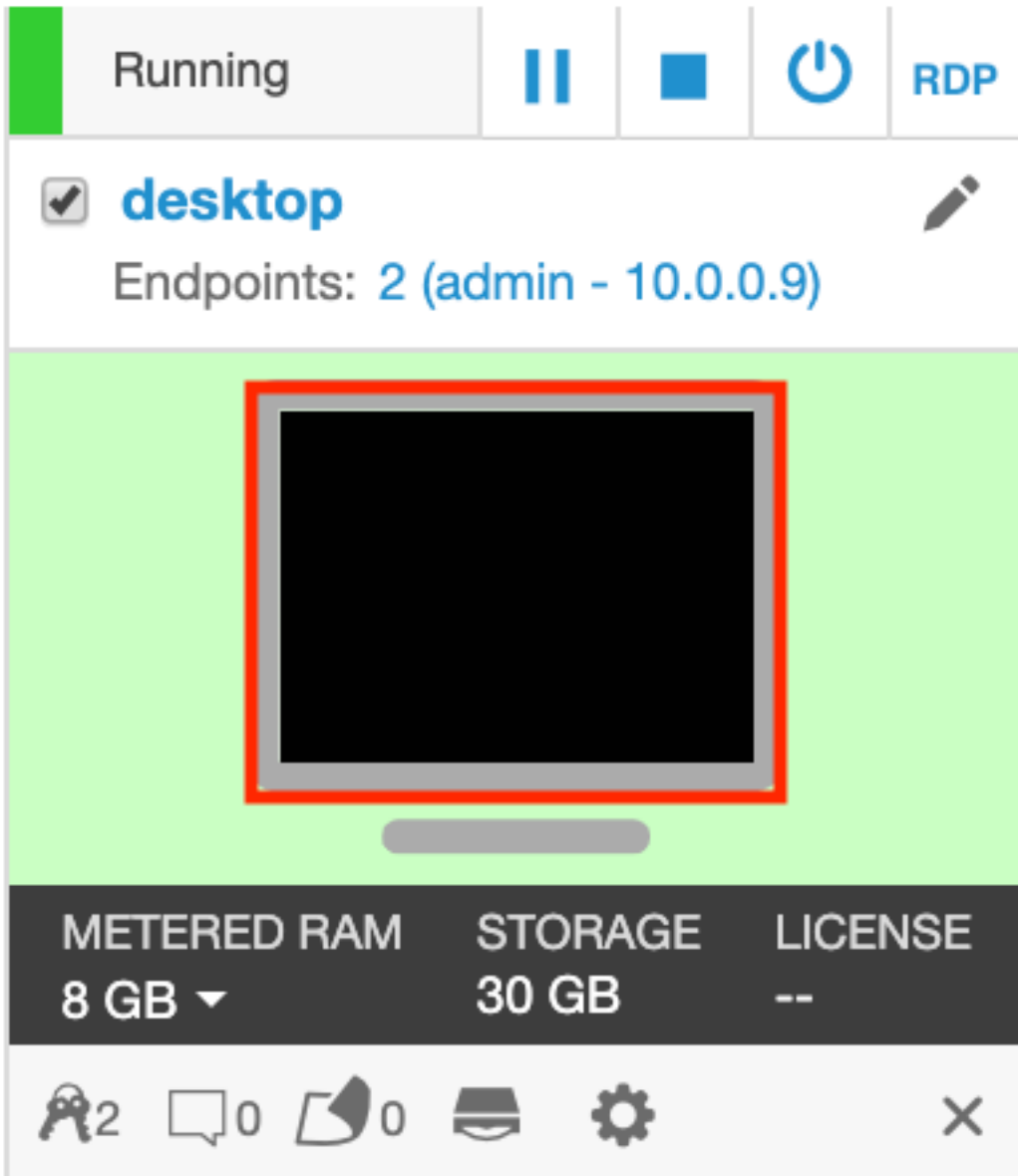
The demo runs on a virtual machine that is provided by IBM Demos. To get access, reserve an instance in the **Before you start** section on this tutorial page. You then receive an email with the URL and password to access the virtual machine.

Navigate to the URL in the email, which opens the IBM Demonstration Portal. Enter the password also included in the email. The IBM Demonstration Portal presents several Linux virtual machines configured in an IBM Cloud Pak for Integration cluster on Red Hat OpenShift.

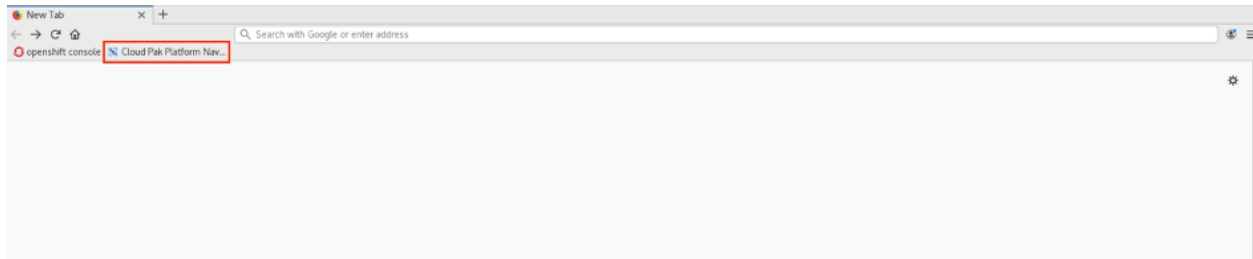
1. If the environment is already up and running when you open your reservation, skip to step 3. If it is not running, continue to step 2.



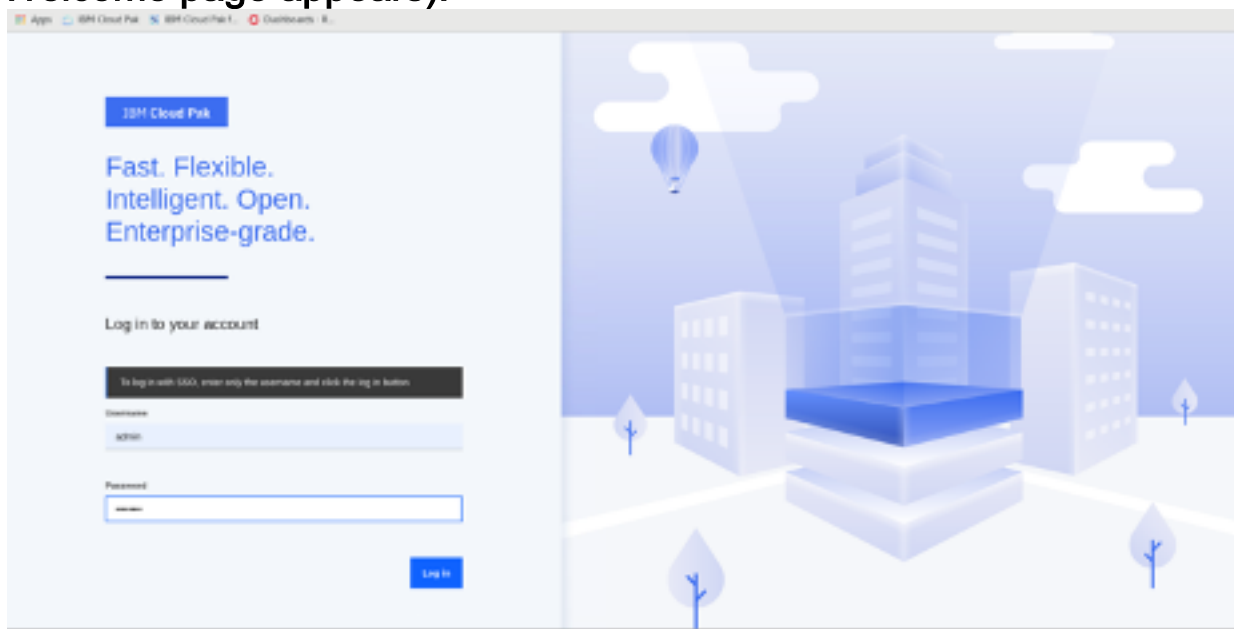
2. Once the virtual machines have started, start IBM Cloud Pack. Click the **Desktop** Machine screen image to start your lab exercise.



3. To access the Developer machine, log in: userid: **ibmuser** and password: **engageibm**.
4. Open a browser and click IBM Cloud Pak for Integration Bookmark bar.

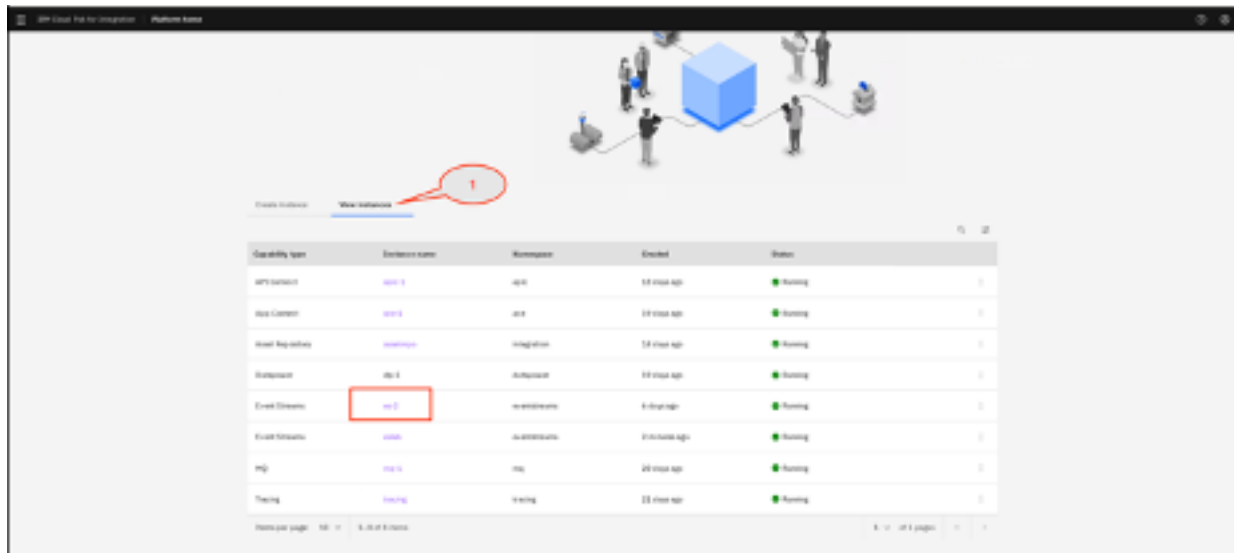


5. **Log in to IBM Cloud Pak for Integration.** The username and password are already filled in. If not, ensure username: **admin** and password: **passw0rd** is entered. Click **Log in** and then click **Skip Welcome** (if Welcome page appears).

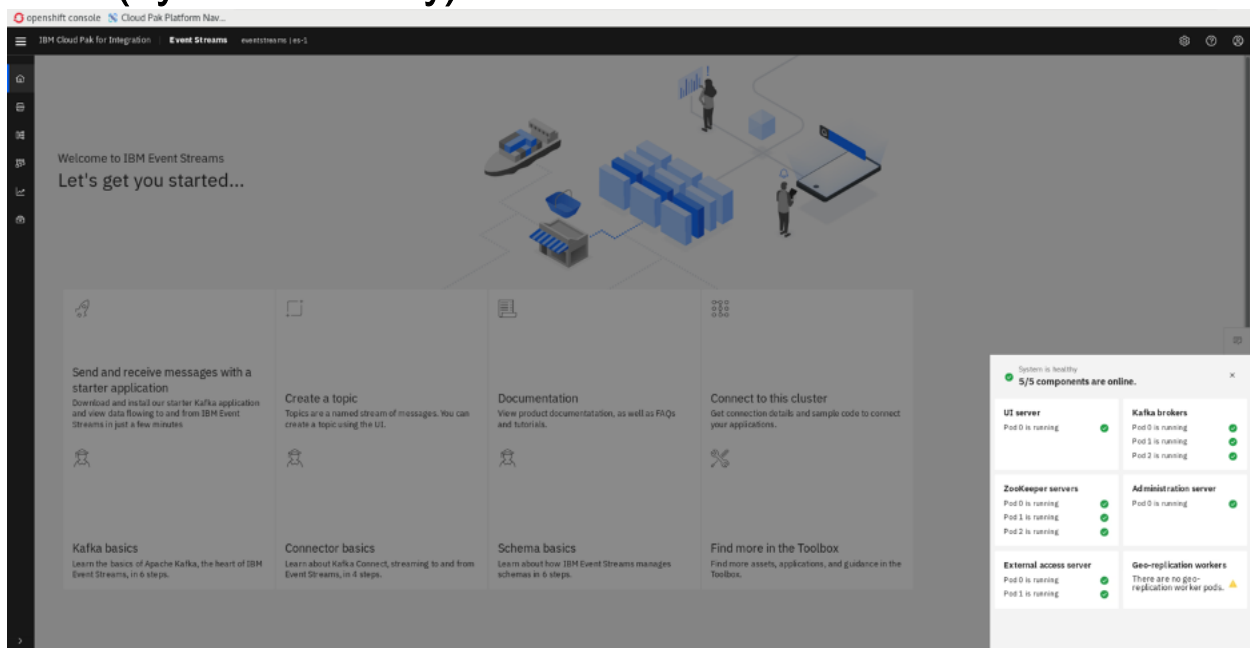


Task 2 – Creating and configuring an Event Streams Topic  
Creating an Event Streams topic in the existing Event Streams instance.

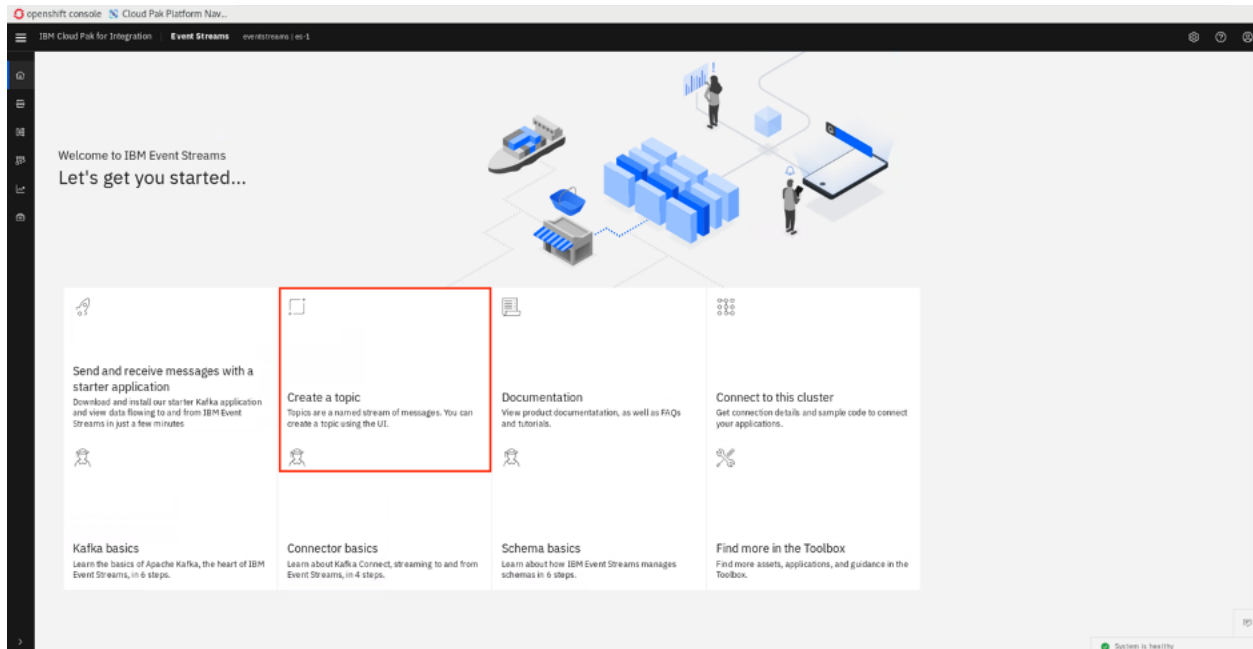
1. On the **IBM Cloud Pak for Integration** main page, click **View Instances**.  
**The list of** integration products deployed is displayed (API Connect, App Connect, Asset Repository, DataPower, Event Streams, MQ, and Tracing).
2. Click **es-1** to use the existing Event Streams instance.



3. On the IBM Event Streams main page, check the status of the instance (**System is healthy**).



4. Click **Create a topic** to configure a topic.



5. View the full range of configuration options by setting the **Show all available options** to **on**.

Create the topic as follows:

1. In **Show all available options**, change to **On**.
2. Topic name: **customerinfo** .
3. Set **Retention time: 10 Minutes**. (This is how long messages are retained before they are deleted).
4. Click **Create topic**.

openshift console Cloud Pak Platform Nav...

Topics / Create topic

Create topic

Show all available options **1**

Core configuration

We recommend you fill out and evaluate these details at a minimum before deploying your topic.

Topic name

Partitions

Replicas

Maximum in-sync replicas

Retention time  Minutes(s) **2** **3**

Reset to default value: '604800000'

Messages

These details control how your messages will be handled in the cluster.

Maximum message bytes  Bytes

Message timestamp type

Maximum message timestamp difference  Milliseconds

Compression type

Cancel Create topic

System is healthy

The Topics page is displayed. Your new topic is displayed along with a completion notification.

- The topic has created, close the pop-up window and click to **connect to this cluster**, you will need an **API key** that allows access to the topic, the Bootstrap server address to connect to the cluster, and the **PEM certificate** for securing the connection from your client to your cluster.

openshift console Cloud Pak Platform Nav...

Topics

Connect to this cluster




Name	Replicas	Partitions	Geo-replication
customerinfo	3	1	-
eventingmq	2	1	-
mqtoevent	3	1	-


Topics per page 10 1-3 of 3 1 1 of 1 pages

Topic creation requested  
A request to create topic 'customerinfo' has been made.  
4/22/2020, 9:29:48 AM **1**

- Click **Generate API Key** to create an API key.







## Cluster connection

Resources


Sample code

Geo-replication

To connect an application or tool to this cluster, you will need the address of a bootstrap server, a certificate and an API key.

### Bootstrap server

Your application or tool will make its initial connection to the cluster using the bootstrap server.

`es-2-ibm-es-proxy-route-bootstrap-eventstreams.apps.demo.ibmde.net:443`



### API key



To connect securely to Event Streams, your application or tool needs an API key with permission to access the cluster and resources such as topics.

Generate API key

### Certificates


A certificate is required by your Kafka clients to connect securely to this cluster.

  
**Java truststore** ⓘ  
Use this for a Java client

  
**PEM certificate**  
Use this for anything else

### API endpoint

Applications and tools that use the REST producer API or the schema registry will need the API endpoint to connect to Event Streams.

`https://es-2-ibm-es-rest-route-eventstreams.apps.demo.ibmde.net`

2. Enter the name of your application on the window (that is, **customerinfoapp**). Select **Produce and consume** for **What do you want your application to do ?** (you can use Produce if you send a message from an Application) and click **Next** .

Generate API key

Generate an API key for your application

To connect securely to Event Streams, your application or tool needs an API key with permission to access the cluster and resources such as topics.

Name your application :

What do you want your application to do?

☐ Produce only

☐ Consume only

☒ Produce and consume

☐ Produce, consume, create topics and schemas

Close Next

3. Enter the name of the topic **customerinfo**, and click **Generate API key**. Event Streams generates the API key.

Generate API key ×

Choose which topics the API key can access

All topics  
☐ Off

Which topic?

customerinfo

*Great! This topic already exists*

Back Generate API key

4. On the next window, click **Generate API key**.

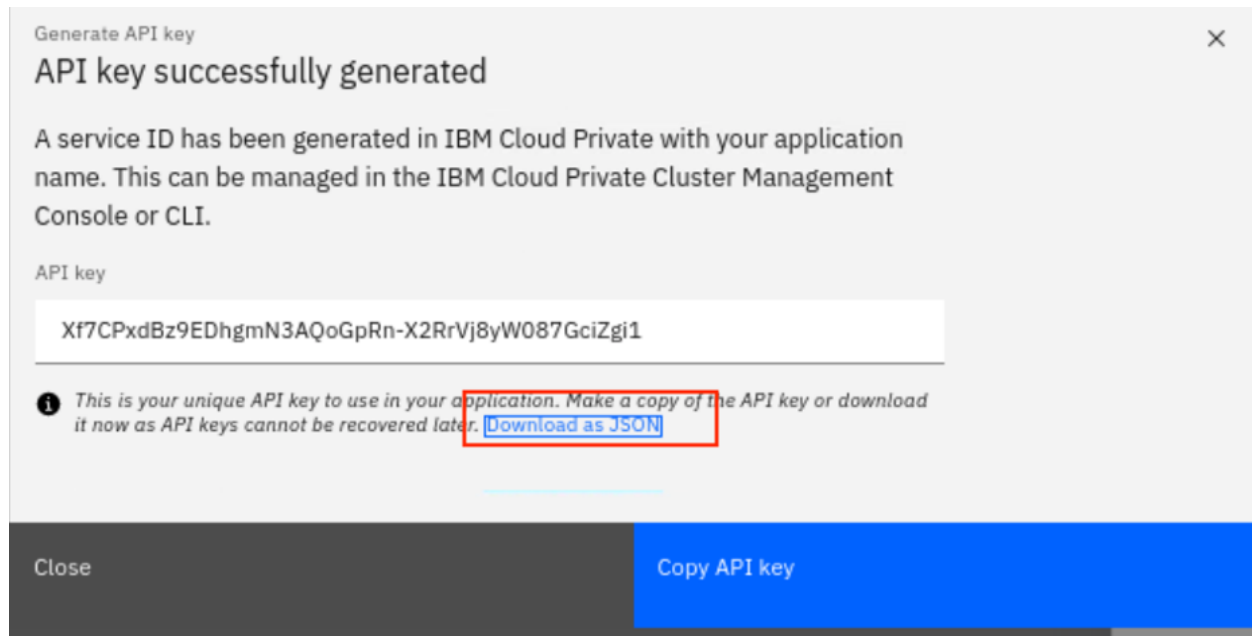
Generate API key ×

Choose which consumer groups the API key can access

All consumer groups  
☒ On

Back Generate API key

5. Event Streams generates an API key. Click **Download as JSON** link (the json file **es-api-key.json** is located in /home/ibmuser/Downloads directory) or copy to a clipboard. Click **Close**.



7. In the Cluster connection page. Copy **Bootstrap server** address and download the **PEM certificate** (the **es-cert.pem** is located in the /home/ibmuser/Downloads directory).

Cluster connection

Resources

Sample code

Geo-replication

To connect an application or tool to this cluster, you will need the address of a bootstrap server, a certificate and an API key.

### Bootstrap server

Your application or tool will make its initial connection to the cluster using the bootstrap server.

`es-2-ibm-es-proxy-route-bootstrap-eventstreams.apps.demo.ibmde.net:443`

### API key

To connect securely to Event Streams, your application or tool needs an API key with permission to access the cluster and resources such as topics.

API key successfully generated

A service ID has been generated in IBM Cloud Private with your application name. This can be managed in the IBM Cloud Private Cluster Management Console or CLI.

`Xt7CPxdBz9EDhgmN3AQo0pRn-X2RzVj8Yw0876ciZgi1`

Generate a new API key

### Certificates

A certificate is required by your Kafka clients to connect securely to this cluster.

JAVA

Java truststore

Use this for a Java client

PEM

PEM certificate

Use this for anything else

### API endpoint

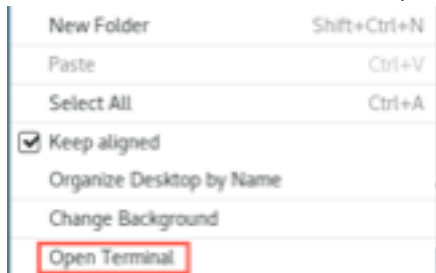
Applications and tools that use the REST producer API or the schema registry will need the API endpoint to connect to Event Streams.

## Task 3 - Configuring App Connect Enterprise flow using App Connect Enterprise Toolkit

You have created a topic in Event Streams created. App Connect Enterprise produces a message and send it to the Event Streams topic. In this task, you will configure an App Connect Enterprise message

flow and generate a BAR file to deploy in the App Connect Enterprise Dashboard.

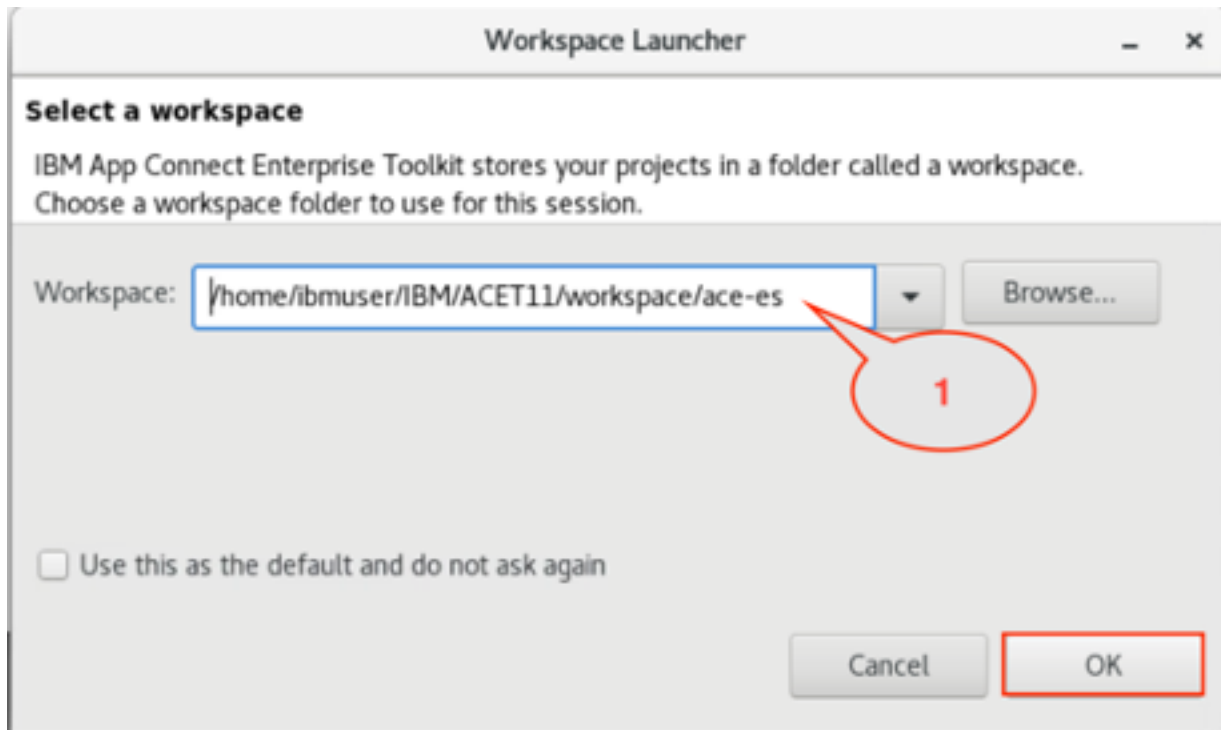
1. You have configured Event Streams. In the Linux desktop, right mouse to open a terminal window (Desktop Machine).



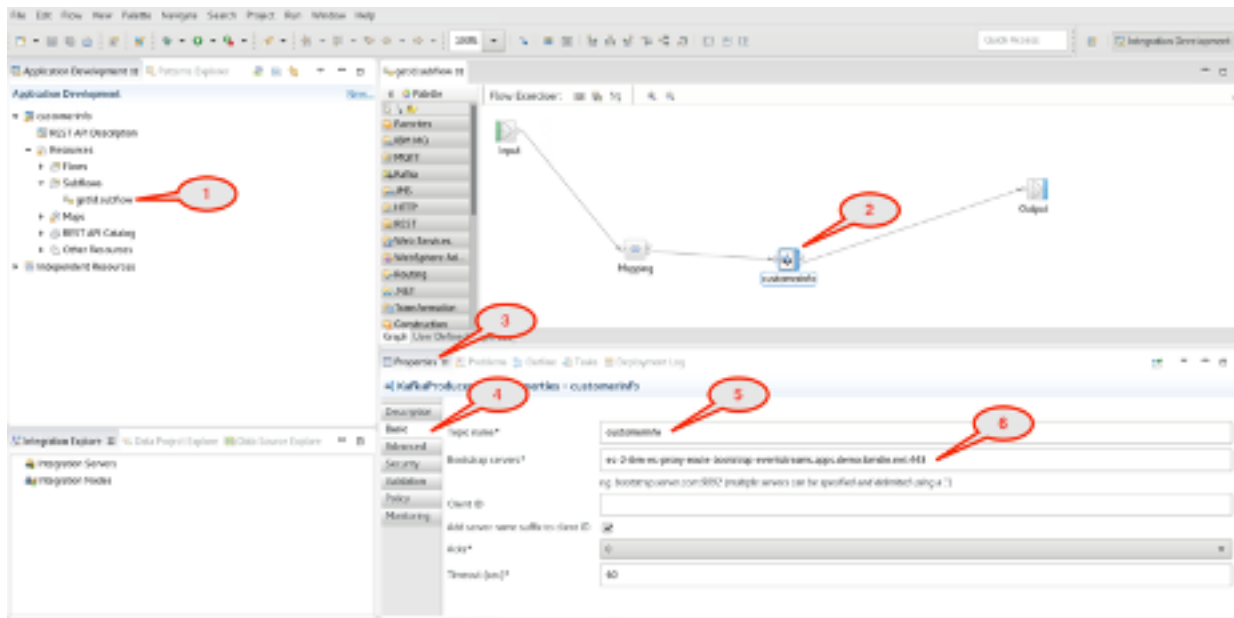
2. You have configured Event Streams configured. Open a terminal window (Developer Machine). Enter **ace toolkit** .



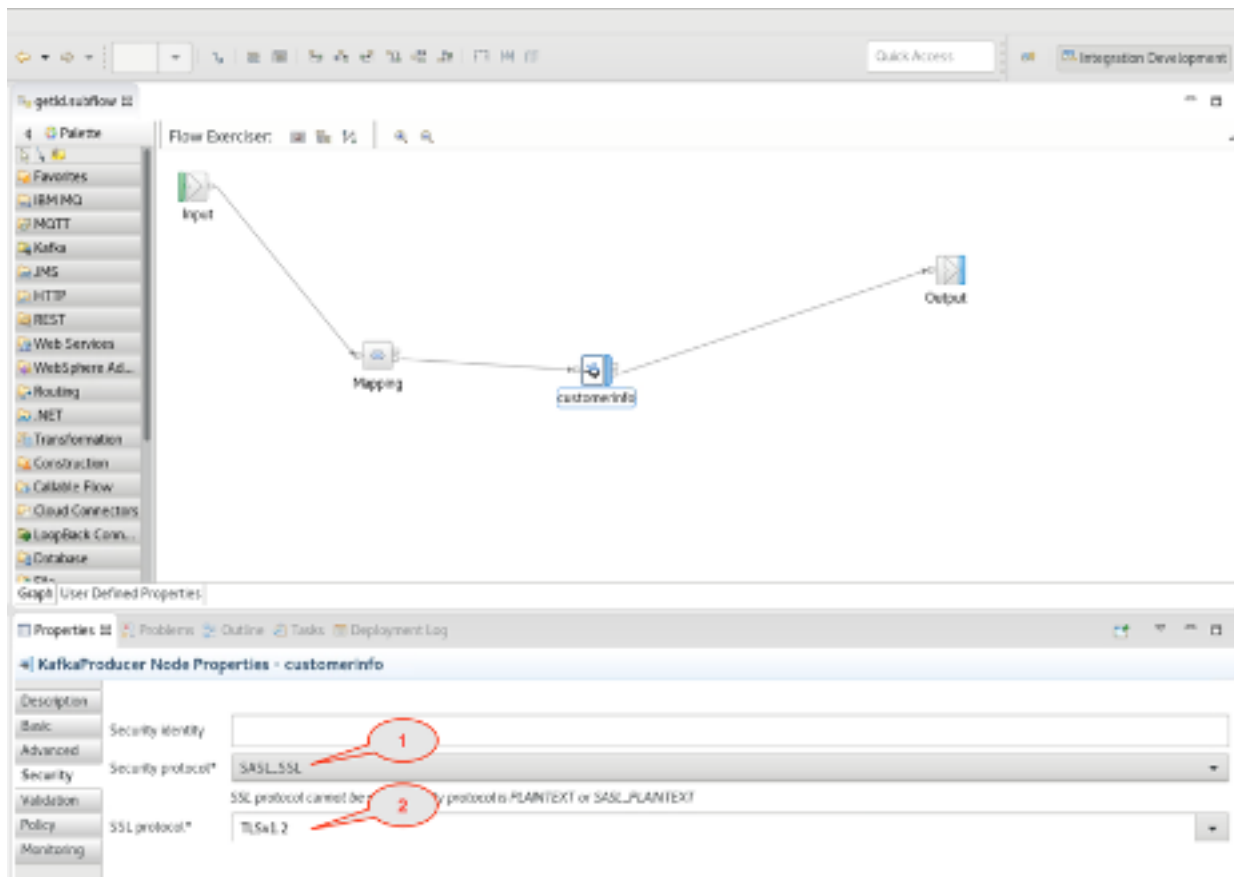
3. Check the App Connect Enterprise workspace directory routes to the ace-es folder (/home/ibmuser/IBM/ACET11/workspace/ace-es). Click **OK** to open App Connect Enterprise toolkit.



4. In the Application Developer on the left bar, select **customerinfo** --> **Resources** --> **Subflows** and click **getid.subflow**. Some errors might show up, you will fix this after you complete and save message flow.
  1. Select **customerinfo** node (Kafka Producer node)
  2. Click **Properties**.
  3. Select **Basic** properties
  4. Check topic name: **customerinfo** (the topic name that you created in Event Streams).
  5. Paste the Bootstrap servers address (the address is found in Event Streams, under **Connect to this cluster->Cluster connection**)

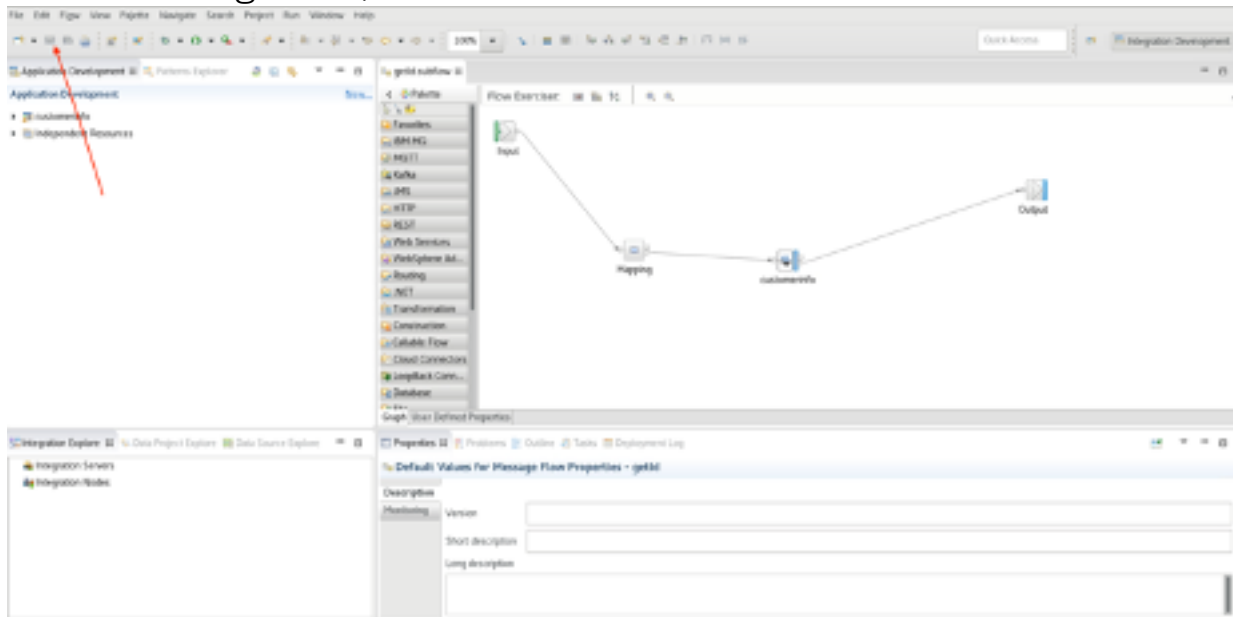


5. In the Security Tab, set the Security Protocol to **SASL SSL** and SSL protocol to **TLSv1.2**.

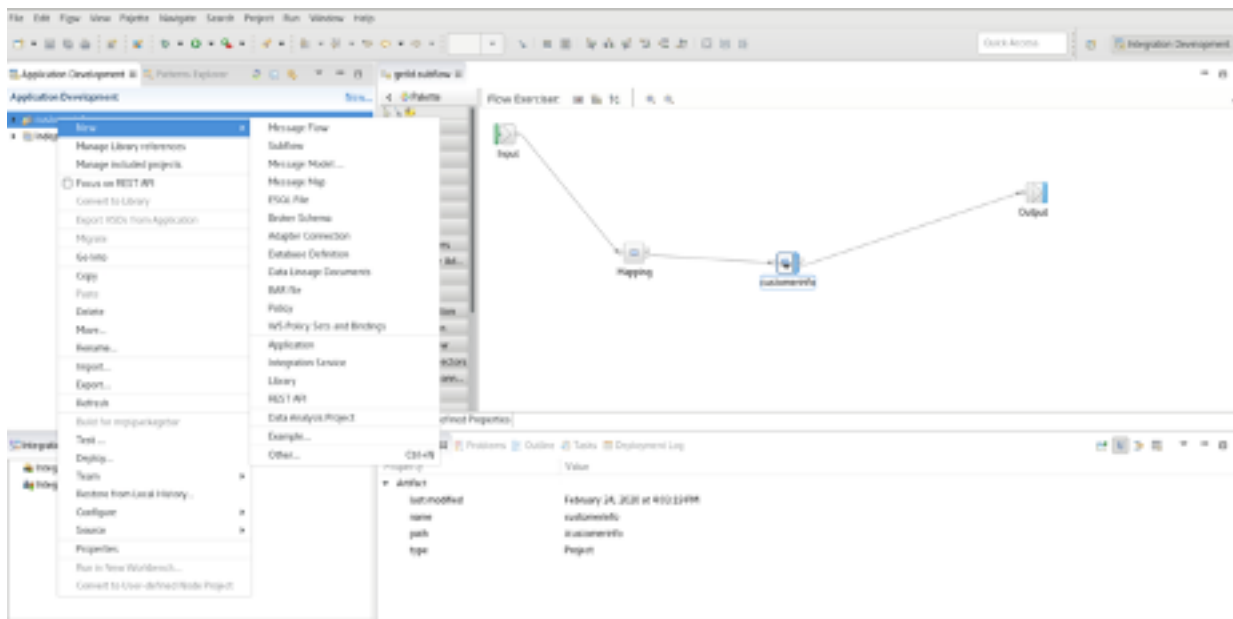




6. Save the message flow, click the **Save** button.




7. You need to deploy the customerinfo application in App Connect Enterprise server. Select **the customerinfo application**. Click **File -> New -> BAR file**.



8. Enter the suggested BAR file name: **customerinfo** and click **Finish**.


**New BAR file** [X]

**Create a new BAR file**  
Create a new BAR file resource 

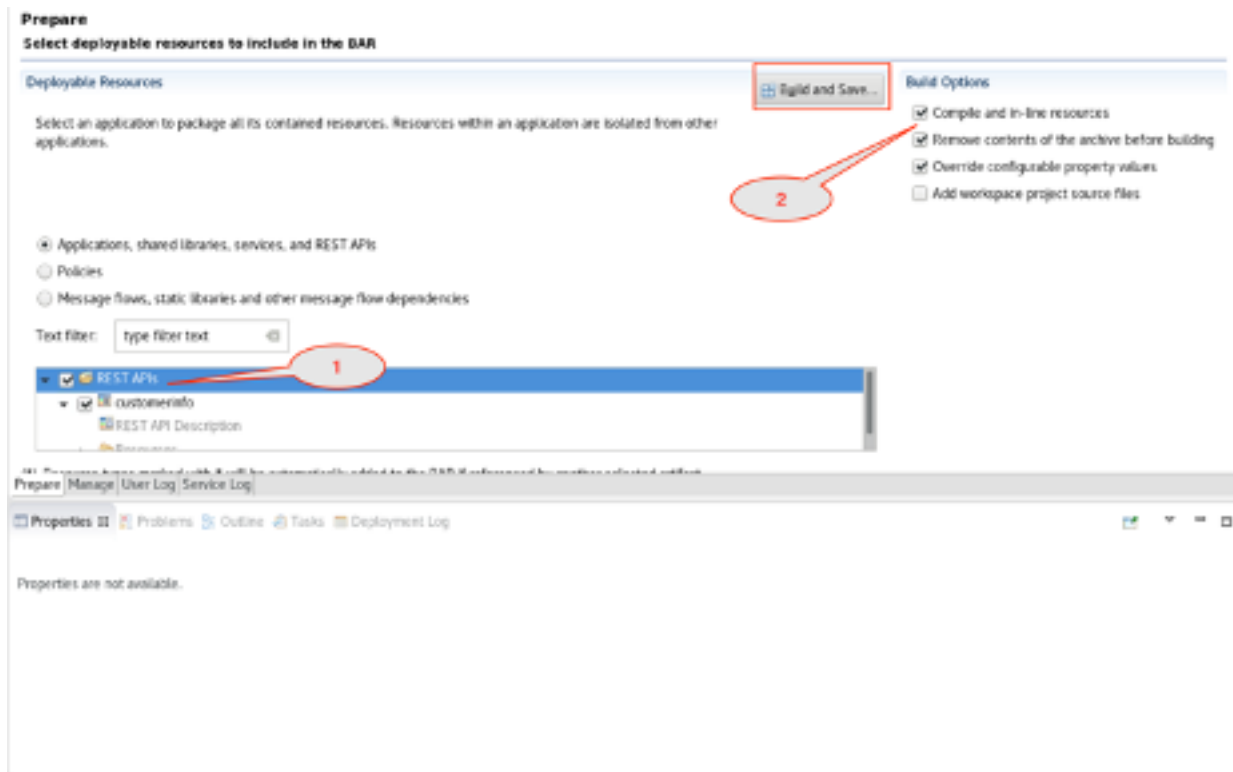
Container:  [v]

Folder:

Name:

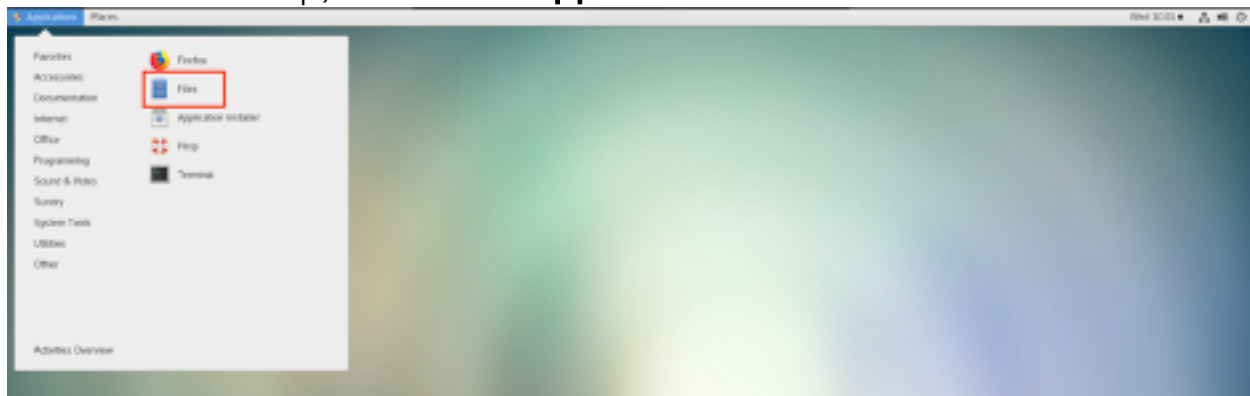


9. Check **customerinfo** application box on the REST API tree. If necessary scroll right to check **Compile and in-line resource** and Click **Build and Save**.

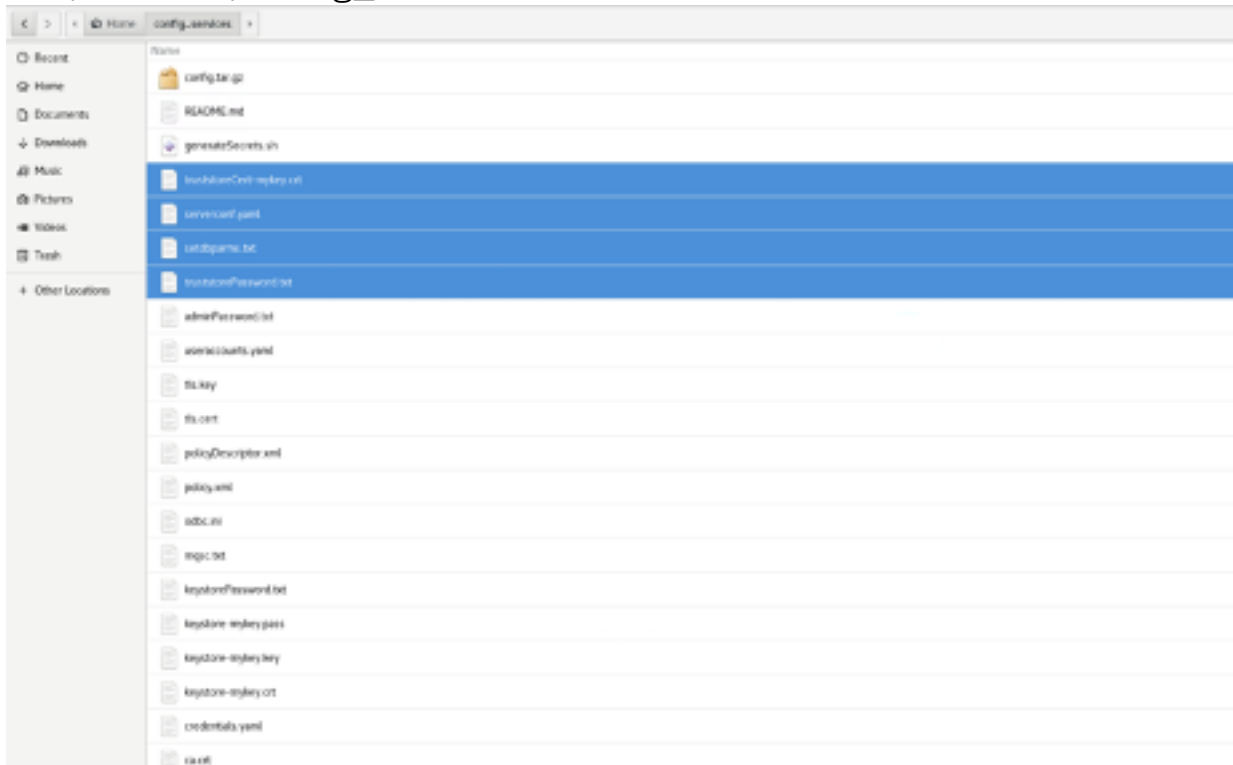


10. A pop-up window display the message “Operation completed successfully.” Click **Ok** to confirm and close the App Connect Enterprise Task 4 - Configuring App Connect Enterprise Services on Cloud Pak for Integration  
In this task you will configure the App Connect Enterprise parameters to access IBM Event Streams.

1. In the Linux Desktop, locate the **Applications** Menu and click **Files**.



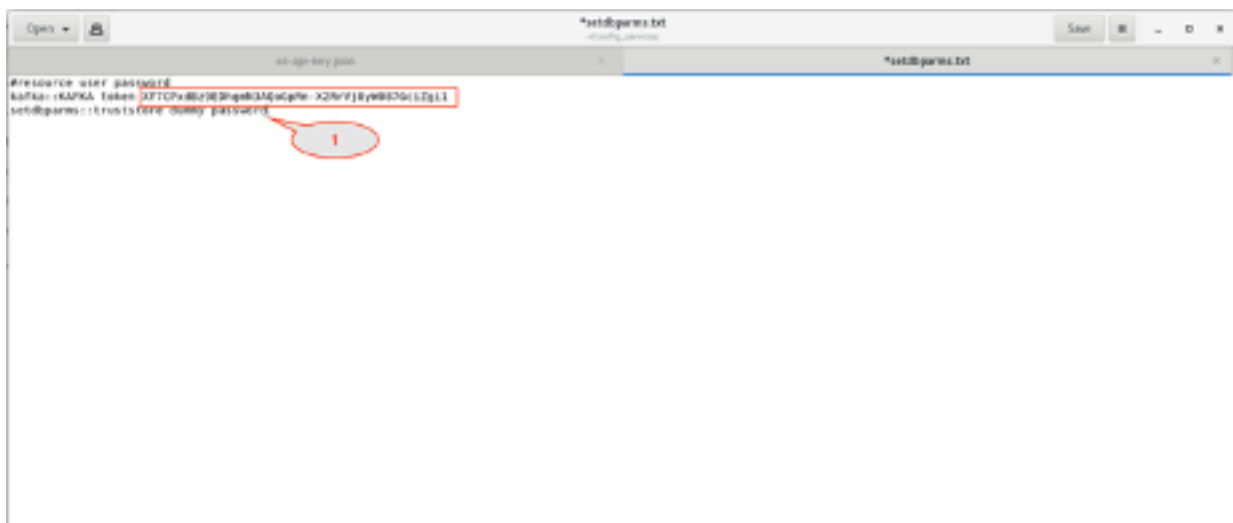
2. Open the folder **/home/ibmuser/config\_services**, App Connect Enterprise needs the configuration parameters to access Event Streams. You configure four files: **setdbparms.txt**, **serverconf.yaml**, **truststorePassword** and **truststore-Cert-mykey.crt**. (These files are in **/home/ibmuser/config\_services**) .



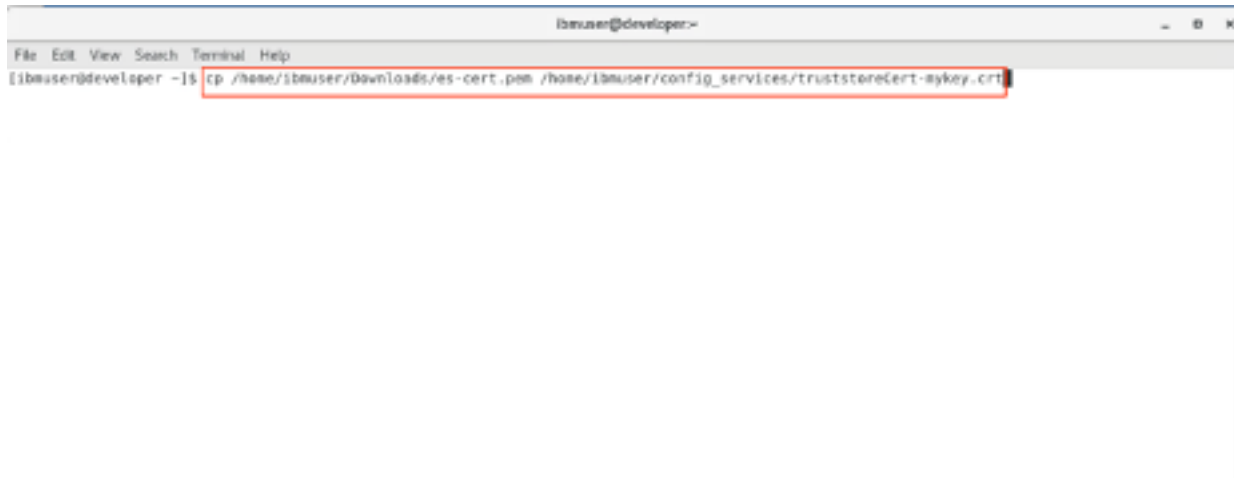
3. You need to edit the Event Streams API Key. Go to the **/home/ibmuser/Downloads** directory, locate and edit **es-api-key.json**. Copy api key between the quotation marks .



4. Open the directory and go to /home/ibmuser/config\_services and edit **setdbparms.txt** by pasting **es-api-key**. App Connect Enterprise uses this parameter to start the Kafka services and use the token to access Event Streams. Click **Save**.



5. Event Streams (Kafka) requires a Certificate. You use the PEM certificate (es-cert.pem) that you downloaded when you configured the Event Streams connection. Open a terminal window and go to /home/ibmuser/Downloads and copy **es-cert.pem** as **truststoreCert-mykey.crt**.



6. Go to `/home/ibmuser/config_services`, edit **trustorePassword.txt** and if necessary, enter **password** in line 1.



7. Go to `/home/ibmuser/config_services`, edit **server.conf.yaml**, and if necessary (We have already done for you) , enter these lines.

```

ResourceManagers:
  JVM:
    truststoreType: 'JKS'
    truststoreFile: '/home/aceuser/ace-server/truststore.jks'
    truststorePass: 'setdbparms:truststore'

# Defaults:
# Policies:
#   HTTPSConnector: 'HTTPS'

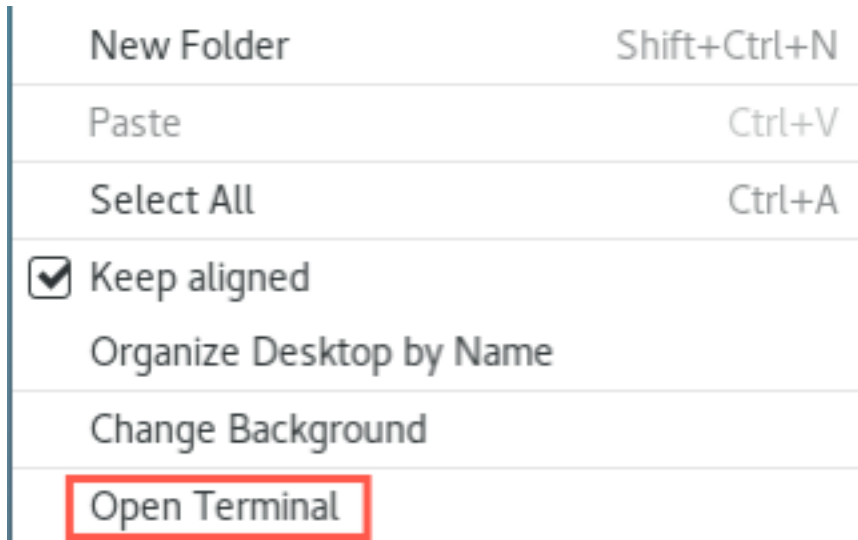
Defaults:
defaultApplication: '' # Name a default application under which independent resources will be placed
policyProject: 'DefaultPolicies' # Name of the Policy project that will be used for unqualified Policy references
Policies:
  # Set default policy names, optionally qualified with a policy project as {policy project}:name
  # HTTPSConnector: 'HTTPS' # Default HTTPS connector policy

```

8. Make sure these files are saved and the name is correct.

9. Create for App Connect Enterprise a secret.

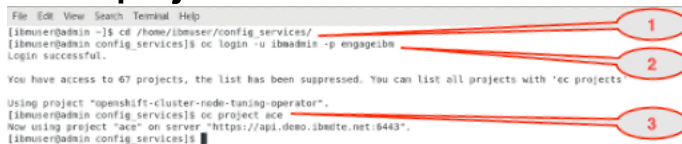
1. Open a terminal window.



2. Enter **cd /home/ibmuser/config\_services**

3. Enter **oc login -u ibmadmin -p engageibm**

4. Enter **oc project ace**.



10. Verify you're in the /home/ibmuser/config\_services directory, enter **./generateSecrets.sh my-secret**. You have created an App Connect Enterprise configuration.



## Task 5 - Deploy App Connect BAR file on App Connect Enterprise Server

The App Connect Enterprise toolkit generated a BAR file. The BAR file has all information to run an App Connect Enterprise instance.

1. Open a new tab and click the **IBM Cloud Pak for Integration** bookmark Bar. Under view instances, click the App Connect link: **ace-1**.

Welcome to IBM Cloud Pak for Integration  
Let's get you going!

Create Instance [View Instances](#)

Capability type	Instance name	Namespace	Created	Version	Status
Event Streams	es-0	eventstreams	7 hours ago		Running
HQ	mq-1	mq	2 days ago		Running
HQ	mq-2	mq	9 days ago		Running
Asset Repository	asurmpo-1	integration	12 days ago		Running
App Connect Designer	acd-design	ace	13 days ago		Running
App Connect Dashboard	acd-1	ace	13 days ago		Running
Event Streams	es-1	eventstreams	15 days ago		Running
Datapower	dp-1	datapower	15 days ago		Running
API Connect	apic-1	apic	15 days ago		Running
Tracing	tracing	tracing	15 days ago		Running

Items per page: 10 1-10 of 10 items 1 of 1 pages

You might receive a page **Your connection is not private**. Click **Advanced**.

**Your connection is not private**

Attackers might be trying to steal your information from [es-1-dat-ent-1-master-eventstreams-apps.demo.ibmcloud.net](#) (for example, passwords, messages, or credit cards).

[View details](#)

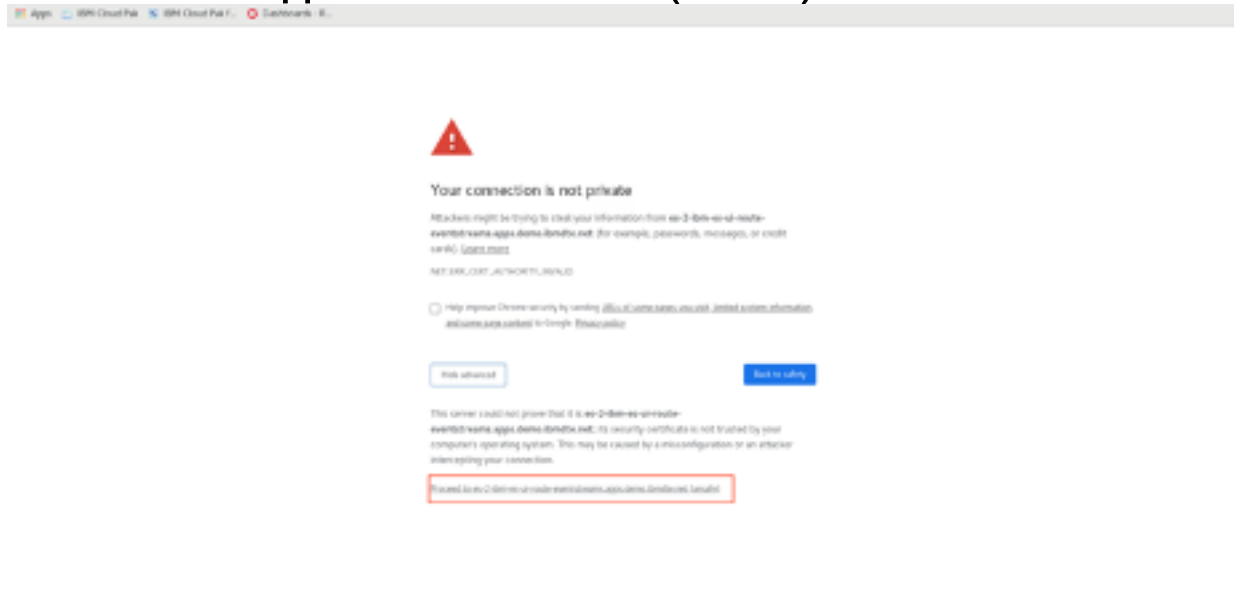
[NET::ERR\\_CERT\\_AUTHORITY\\_INVALID](#)

☐ Help improve Chrome security by sending [diagnostics](#) (you will send limited system information, and some data will be sent to Google). [Privacy policy](#)

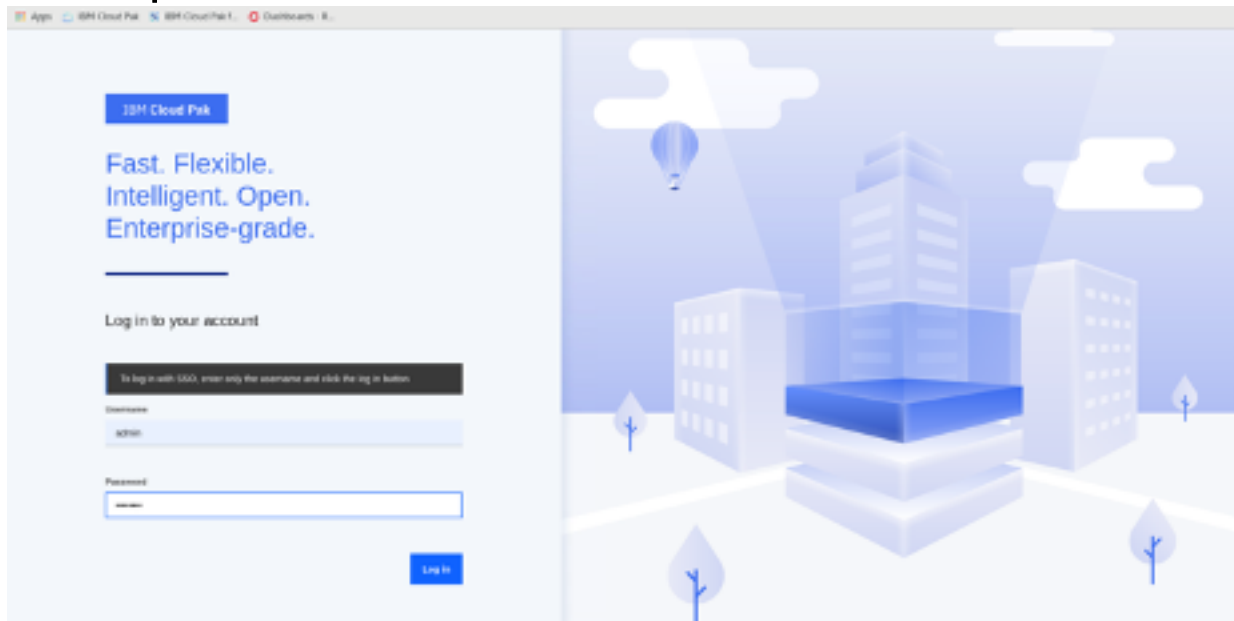
[Advanced](#) [Back to safety](#)



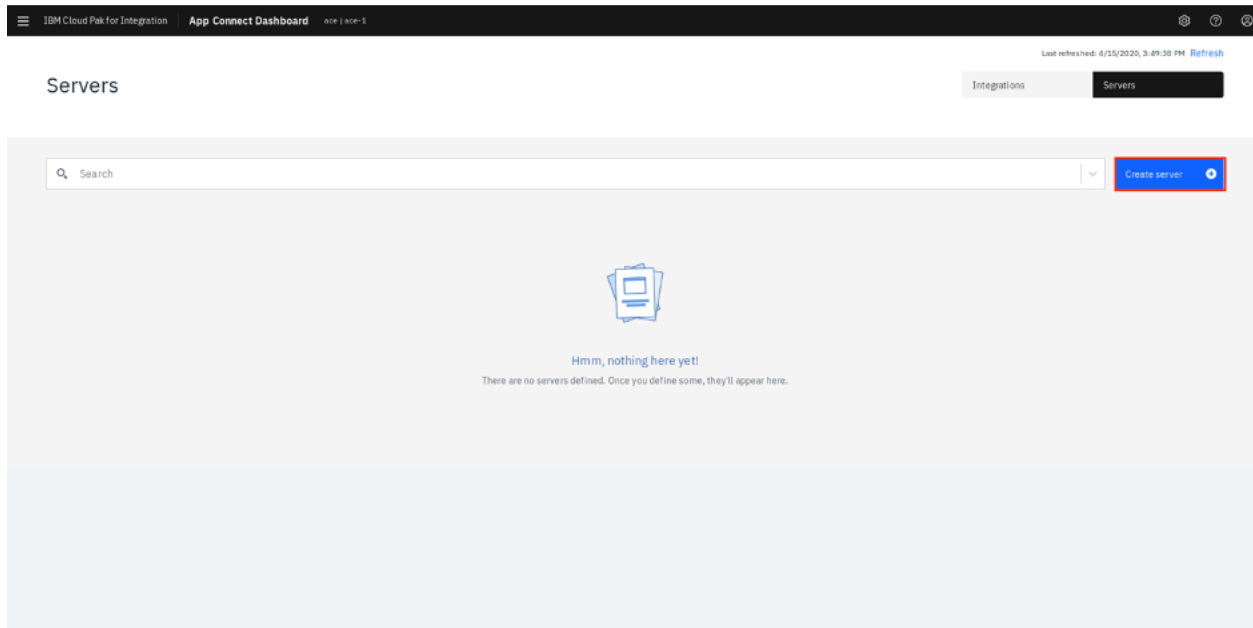
Click the link **Proceed to es-1ibm-es-ui-route-eventstreams.apps.demo.ibmde.net (unsafe).**



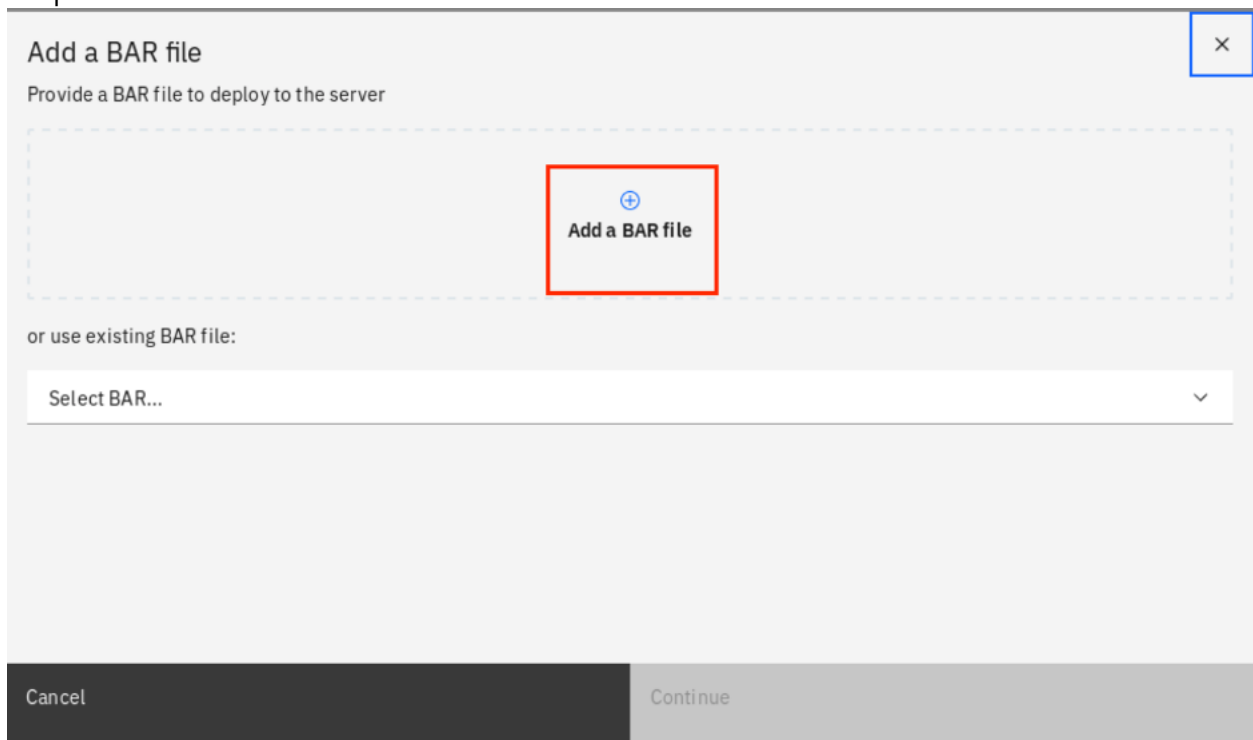
2. If you receive the login page, make the login username: **admin** and Password: **passw0rd**.



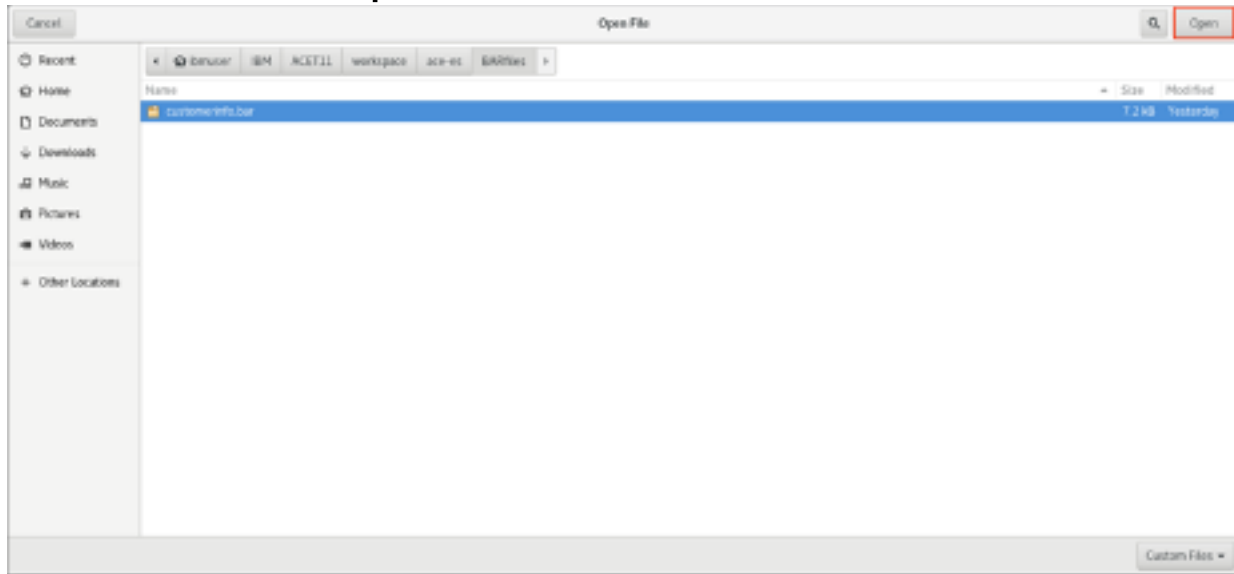
3. You deploy the BAR file that you saved in the App Connect Enterprise toolkit. This is the App Connect Enterprise Dashboard. All deployed Integration servers will be here. Click **Create Server**.



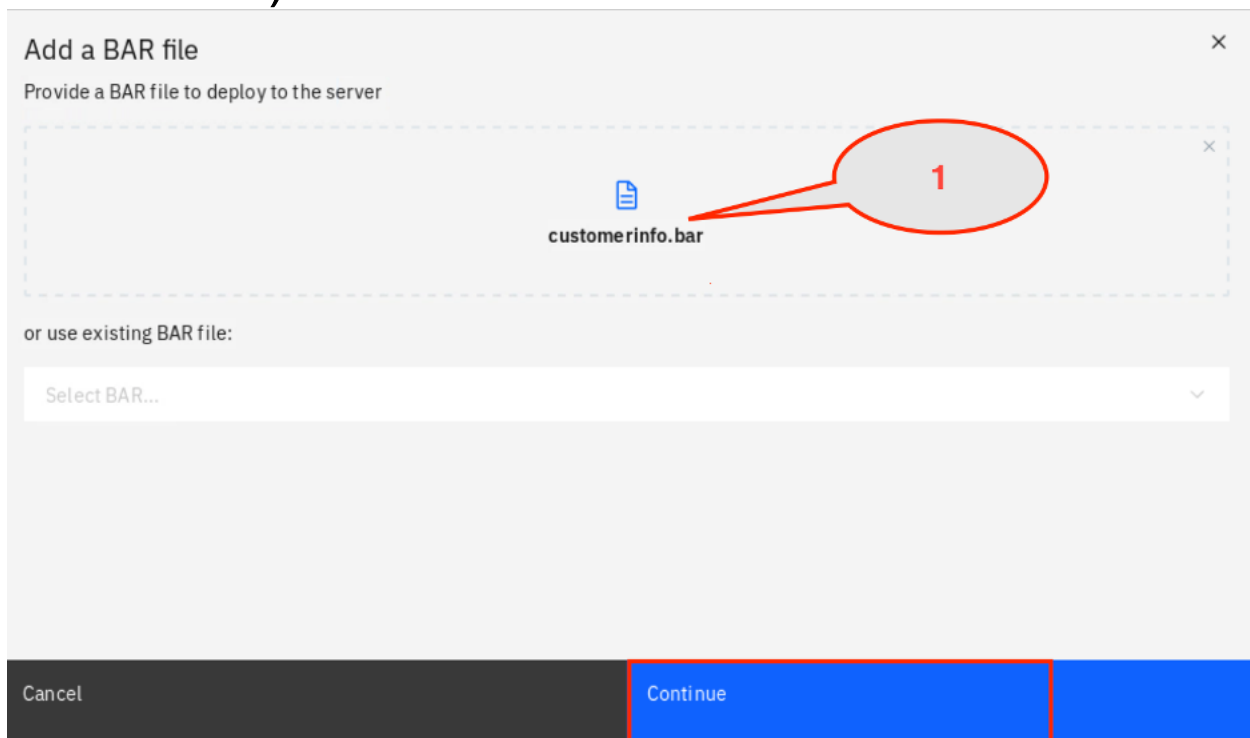
4. On the pop-up, **Add a Bar file**, add the BAR file you saved in App Connect Enterprise toolkit. click **Add a BAR file**.



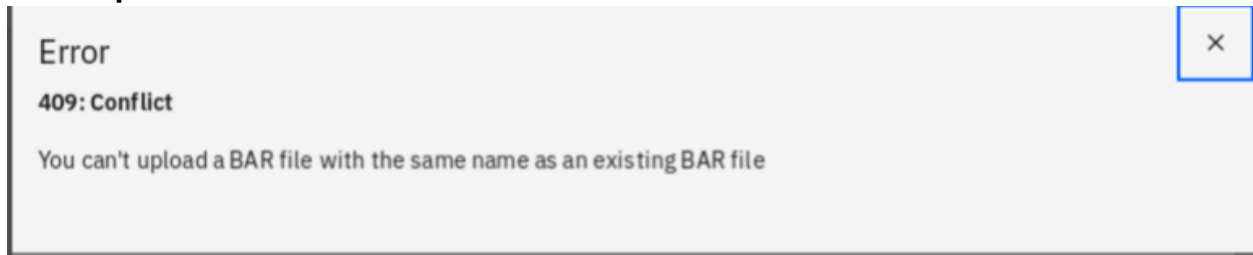
5. The deployment process opens **Open File** and locate **customerinfo.bar** in **/home/ibmuser/IBM/workspace/aces/BARfiles** and click **Open**.



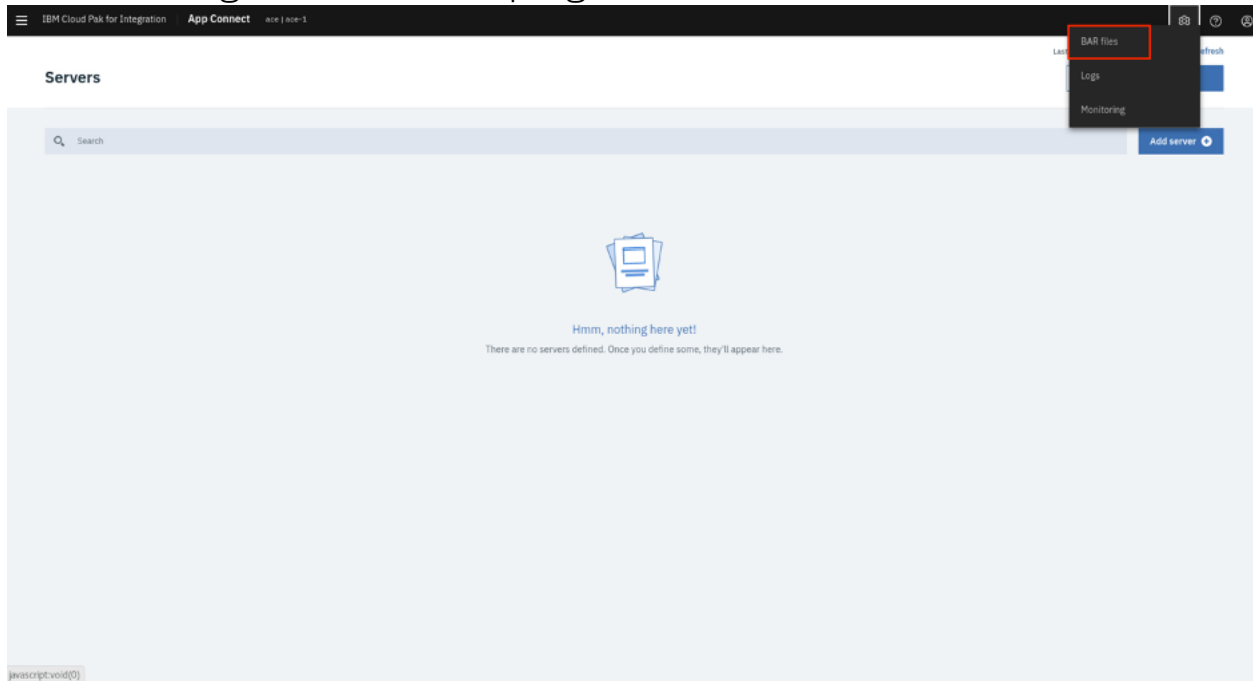
6. App Connect Enterprise shows the BAR file that you selected (**customerinfo.bar**). Click **Continue**.



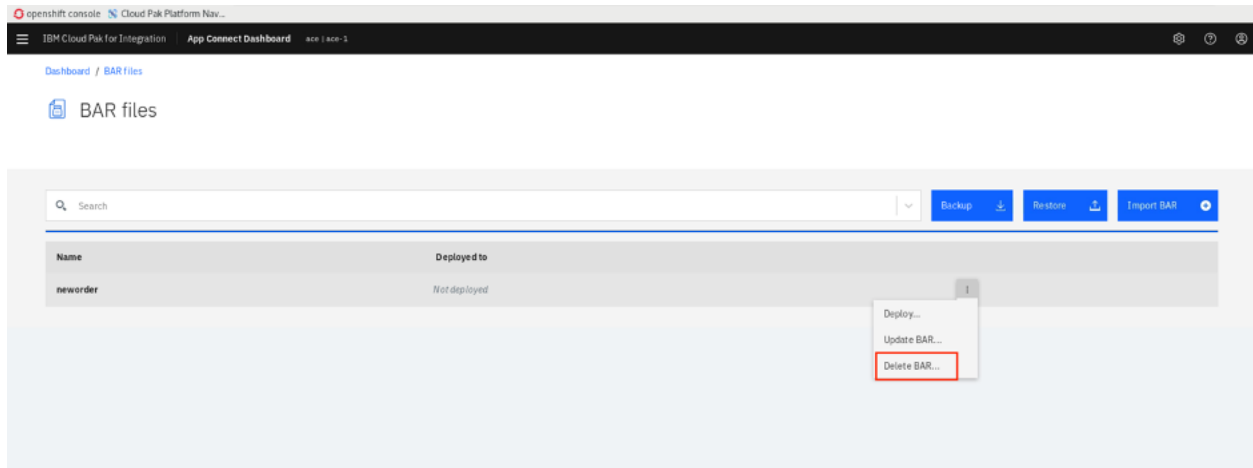
7. You might receive an Error message (**Conflict**). Close the Window. If not go to **Step 12** .



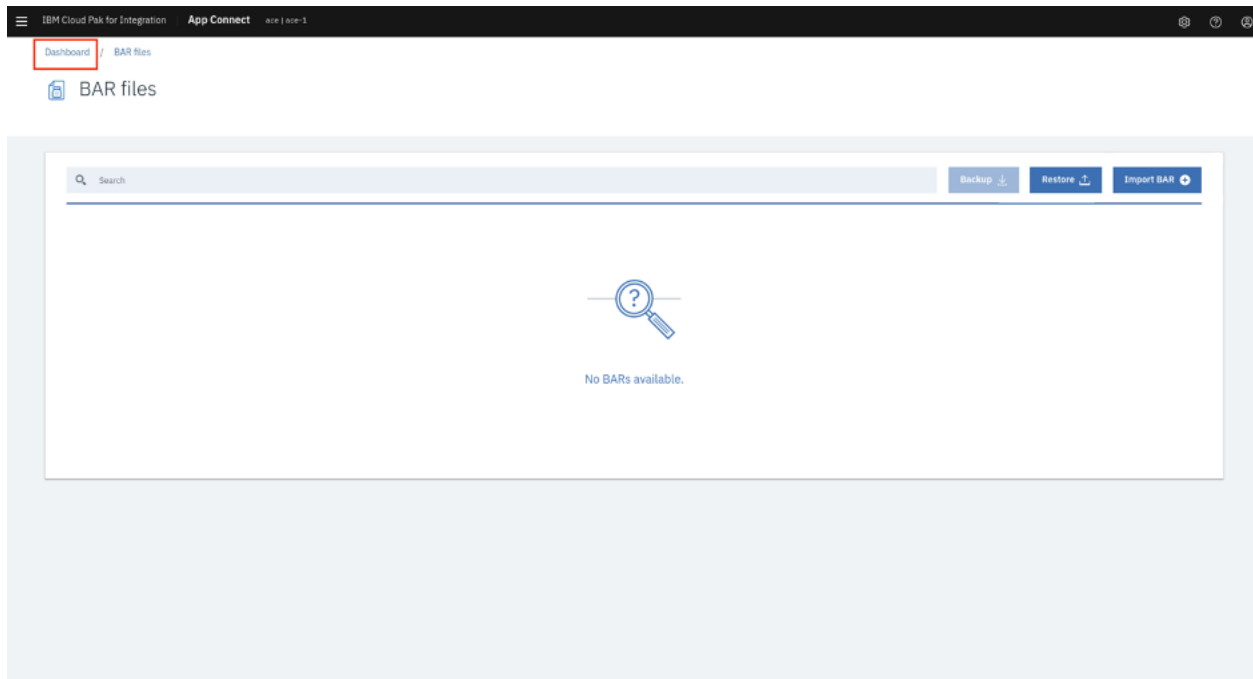
8. Click the settings icon on the top right and select **BAR files**.



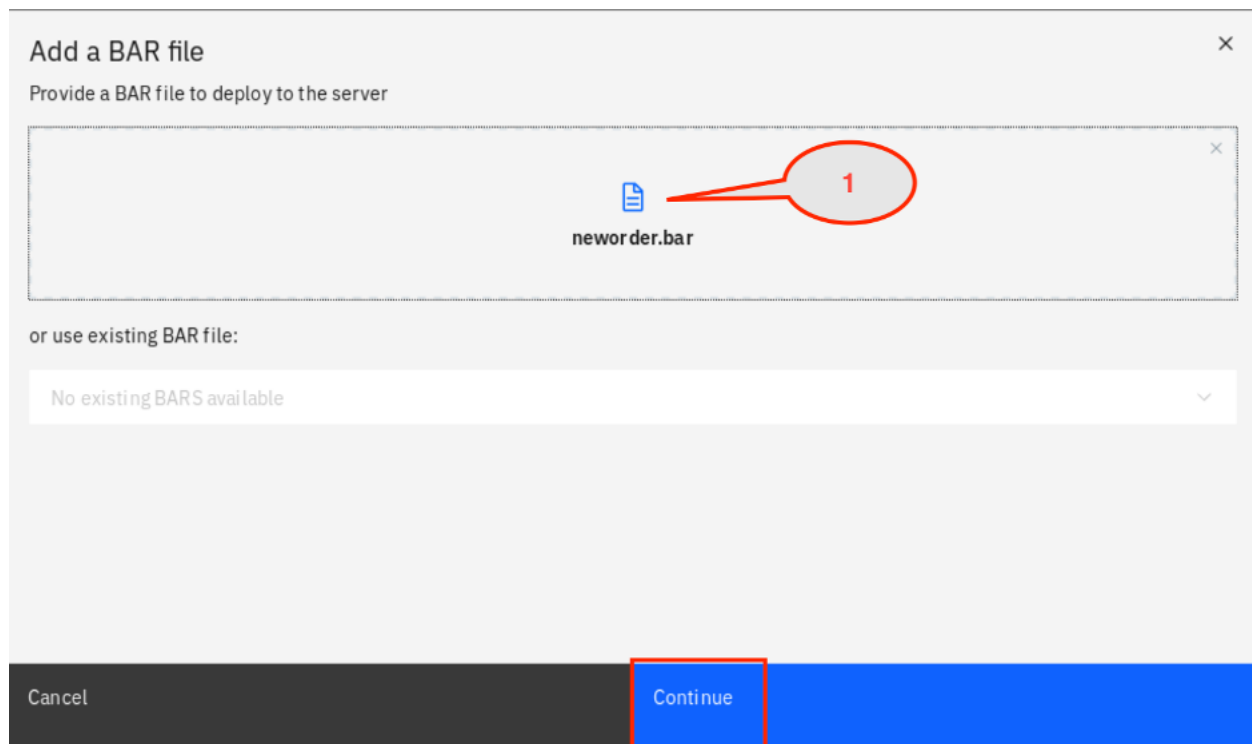
9. You see the **Not Deployed** BAR file, click ( ... ) and select **Delete BAR**. )You can deploy, delete or update the Bar file. For this lab just Delete).



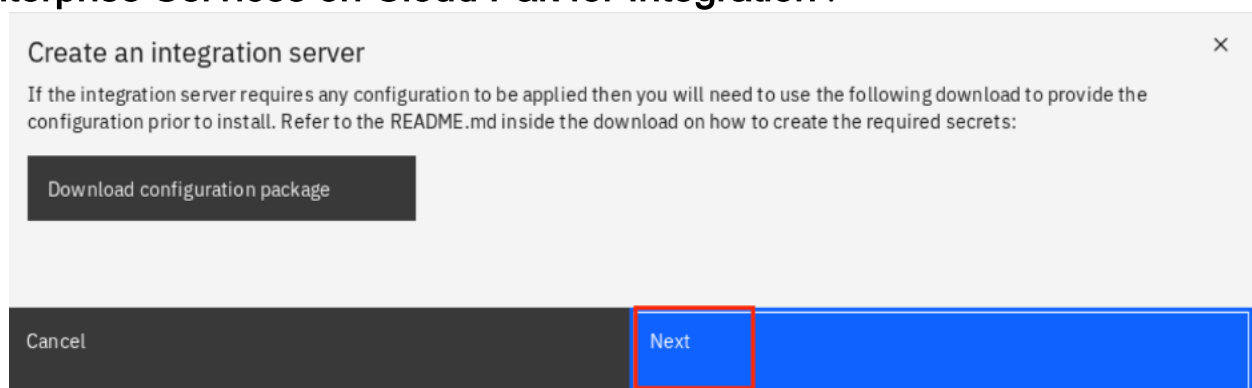
10. Click **Dashboard** link on the top of the page. To deploy the BAR file.



11. Repeat steps 3, 4, 5 and 6 and then In the **Add a BAR File**. Check the BAR file name: **neworder.bar** and then click **Continue**.



12. You don't need to download configuration package., you already have configured the App Connect Enterprise server (when you run **generate-secret script**). Click **Next**. (You might notice an icon **Download configuration package**. You can skip this because you have already set this setting in **Task 4 Configuring App Connect Enterprise Services on Cloud Pak for Integration** .



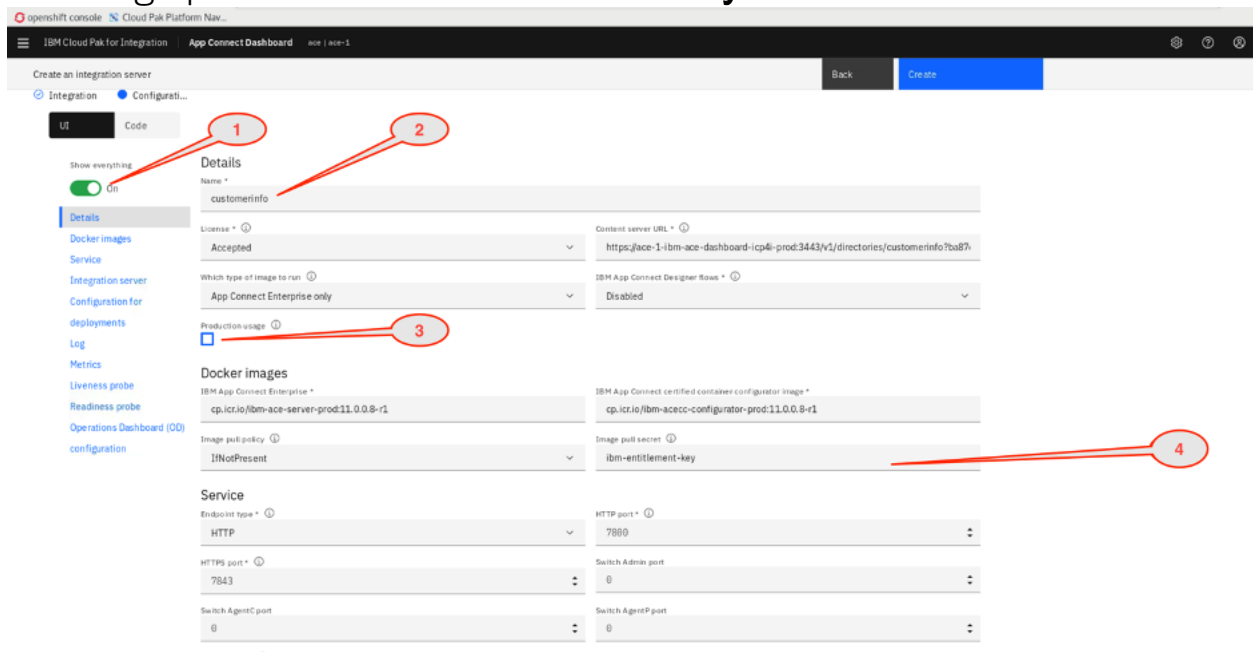
13. In the **Create an integration server** page. You have two option to deploy a BAR file. Deploy a BAR file from **App Connect Toolkit** or a BAR

file from **App Connect Designer**. In this lab you deploy BAR file from App Connect Toolkit. Select **Toolkit** link and then click **NEXT**.



14. In Create Integration server page, you configure the parameters:

1. Change **On** to **Show Everything**
2. Enter the deployment name: **customerinfo**
3. Uncheck **Production usage** box.
4. Enter Image pull secret: **ibm-entitlement-key**.



15. Scroll and locate Integration Server parameters:

1. Enter **my-secret** as the **Name of the secret that contains the server configuration.**

2. Enter **mykey** as the **List of certificate aliases for the the truststore**.

IBM Cloud Pak for Integration | App Connect Dashboard | new | page 1

Create an integration server

Show everything

On

Details

Docker images

Service

Integration server

Configuration for deployments

Log

Metrics

Liveness probe

Readiness probe

Operations Dashboard (OD) configuration

Integration server

HTTP

7800

HTTPS port \*

7843

Switch Admin port

0

Switch Agent port

0

Web UI port \*

7600

Service type \*

ClusterIP

Integration server

Name of the secret that contains the key

my-secret

Name of the default application

The name of the default application for the deployment of independent resources

File system group ID

The file system group ID to use for volumes that support ownership management (sudo)

List of key aliases for the keystore

Comma-separated list of key alias names to mount from the pre-installed secret

List of certificate aliases for the truststore

mykey

Configuration for deployments

Replica count

3

CPU limit \*

1

Memory limit \*

1024Mi

CPU request \*

200m

Memory request \*

256Mi

Log

Log settings

16. Scroll to **Operations Dashboard configuration**. Check **Enable Operations Dashboard** box. Enter the name of the Operations Dashboard namespace: **tracing** in field OD tracing instance namespace and click **Create**.

IBM Cloud Pak for Integration | App Connect Dashboard | new | page 1

Create an integration server

Show everything

On

Details

Docker images

Service

Integration server

Configuration for deployments

Log

Metrics

Liveness probe

Readiness probe

Operations Dashboard (OD) configuration

Enable Operations Dashboard

OD tracing instance namespace

tracing

OD agent image repository

cp.km.vsp4i-od-agent

OD agent image tag

1.0.2

OD agent broker image repository

cp.km.vsp4i-od-agent

OD agent broker image tag

1.0.2

OD agent readiness probe initial delay (seconds)

30

OD collector image repository

cp.km.vsp4i-od-collector

OD collector image tag

1.0.2

OD collector readiness probe initial delay (seconds)

30

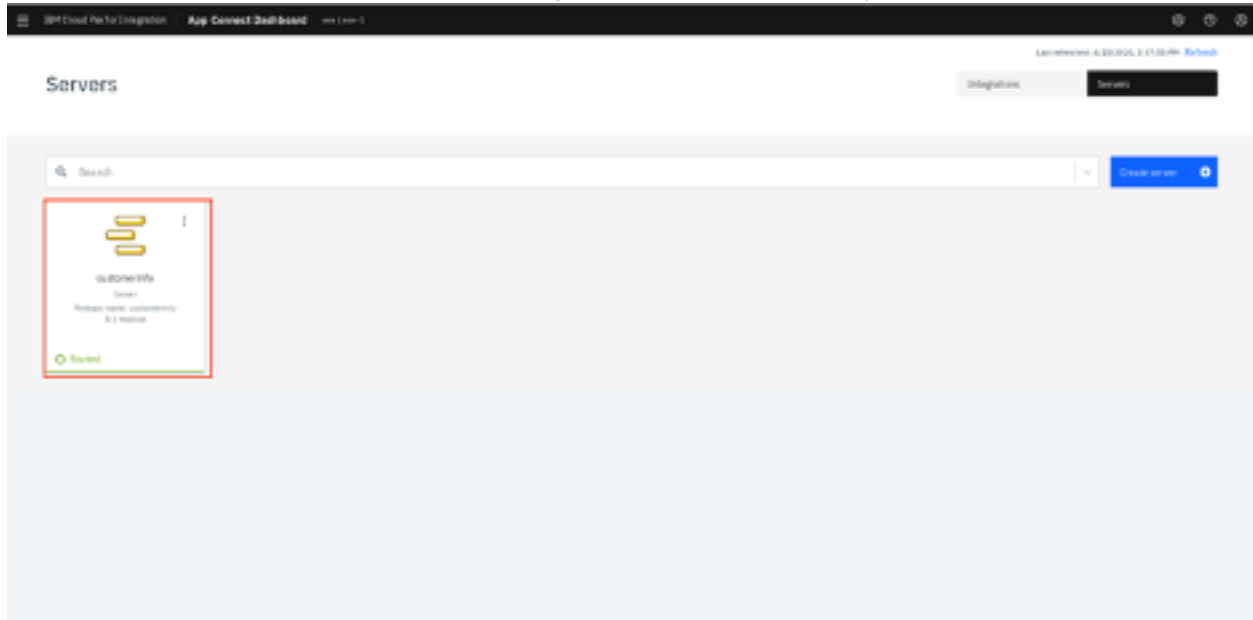
OD collector readiness probe initial delay (seconds)

30

Admin server security needed to gather statistics for Grafana



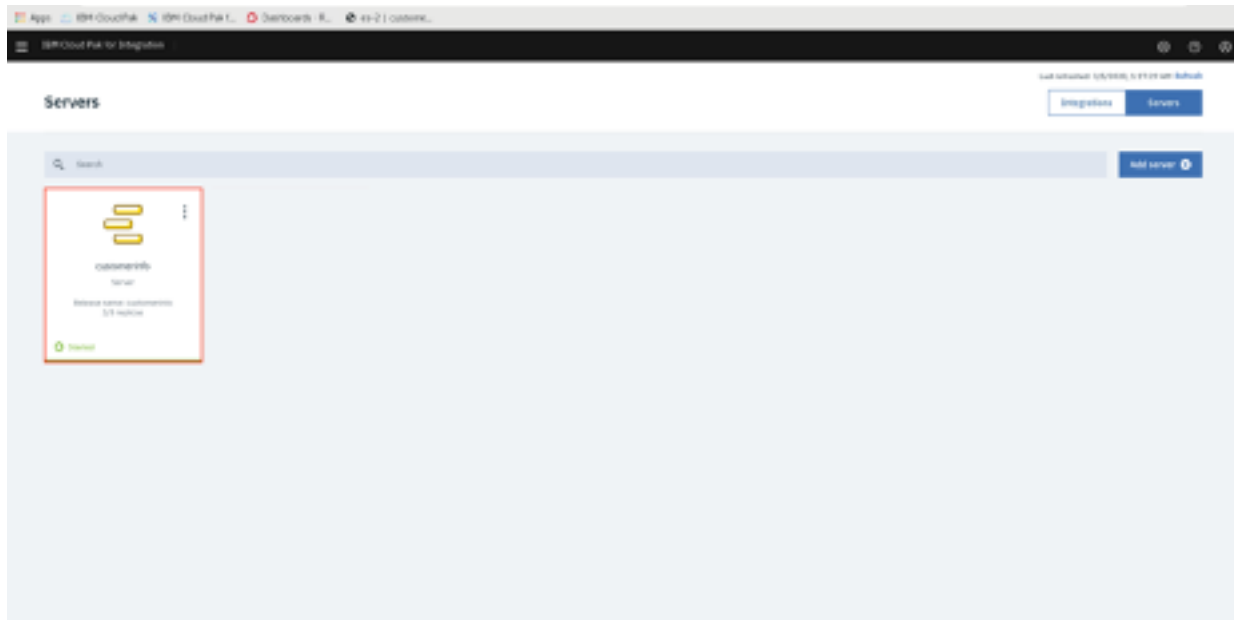
17. After the BAR deployment, the process guides you back to App Connect Dashboard. The deployed application will be on **Started** status, if the status is **Unavailable**, **refresh** your browser to update.



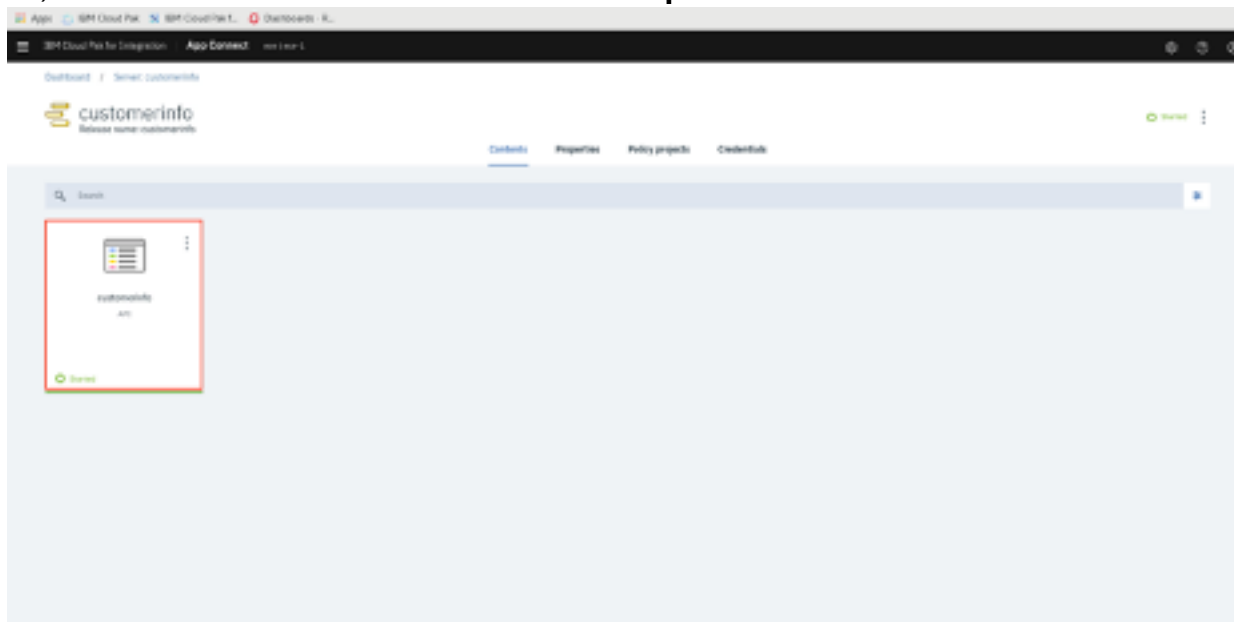
Task 6 - Testing App Connect Enterprise API sending a message to Event Streams

In this task, test an API created in App Connect Enterprise that sends to Event Streams.

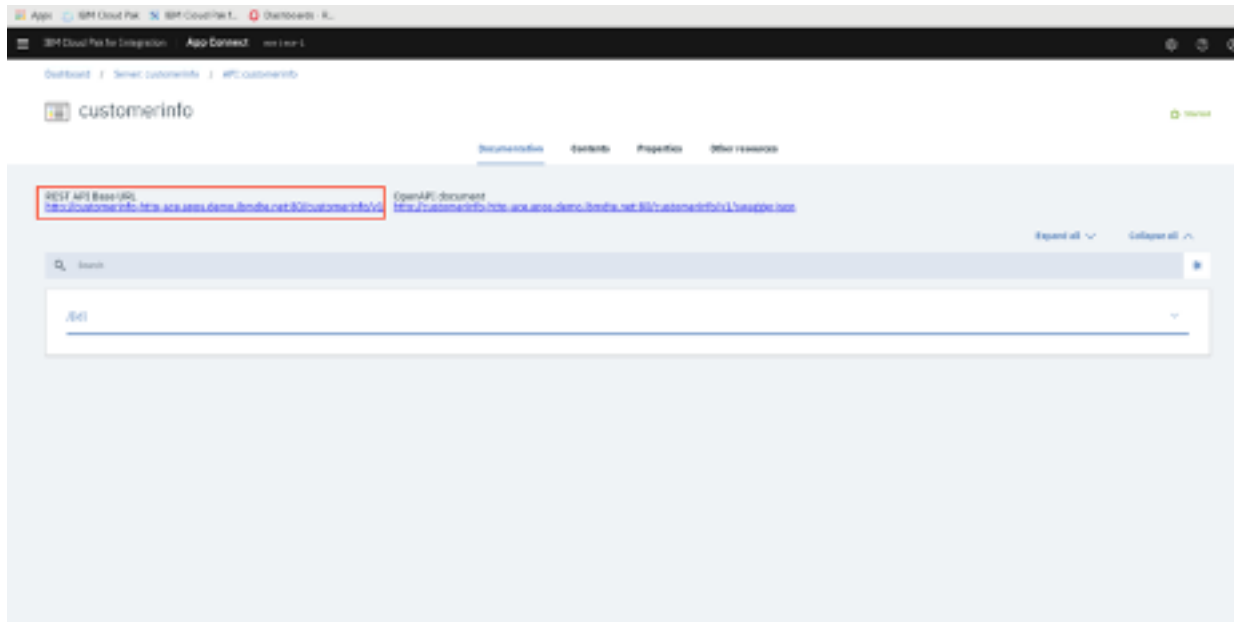
1. You see App Connect Enterprise Server running.  
Click the **customerinfo** server icon.



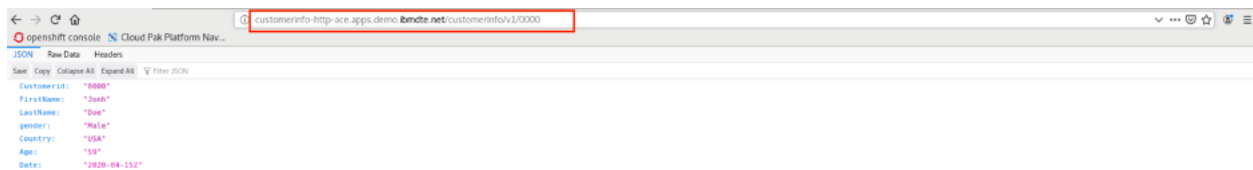
2. Next, click **customerinfo API** and then **Open**.



3. App Connect Enterprise created a **REST API base URL** and **OpenAPI document**.



4. Click the RESTAPI base URL. You receive a 404 error. Complete the Rest API on the URL browser: **customerinfo-http-ace.apps.demo.ibmmdte.net/customerinfo/v1/0000** and see the results. Note the customer information and Customerid: **0000**.



5. A message was sent from App Connect Enterprise to IBM Event Streams. Go to **IBM Pak Cloud for Integration**. You can use **Cloud Pak Menu**, click the Hamburger menu on the top and Click **Event Streams** application -> **es-1** instance .

Platform home

Cloud Pak Foundation

OpenShift Console

Logging

Monitoring

API Connect 1 instance

App Connect 2 instances

Aspera 0 instances

DataPower 1 instance

Event Streams 1 instance

MQ 2 instances

Tracing 1 instance

Asset repository 1 instance

Event Streams

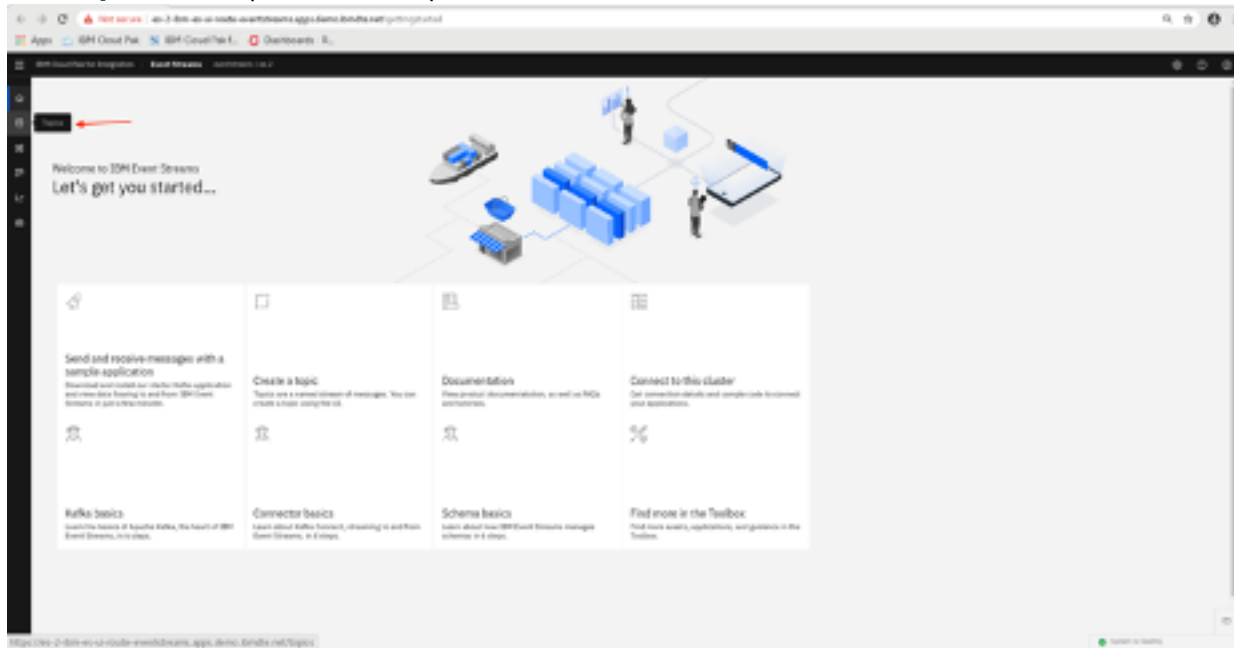
Find

eventstreams

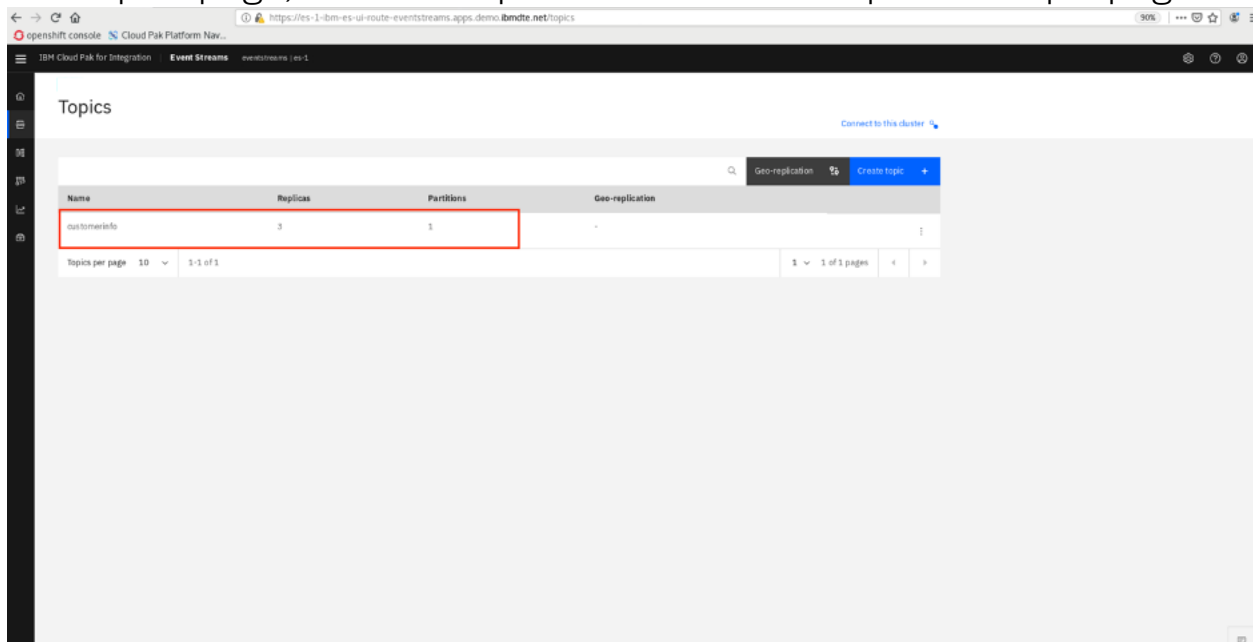
es-1

⋮

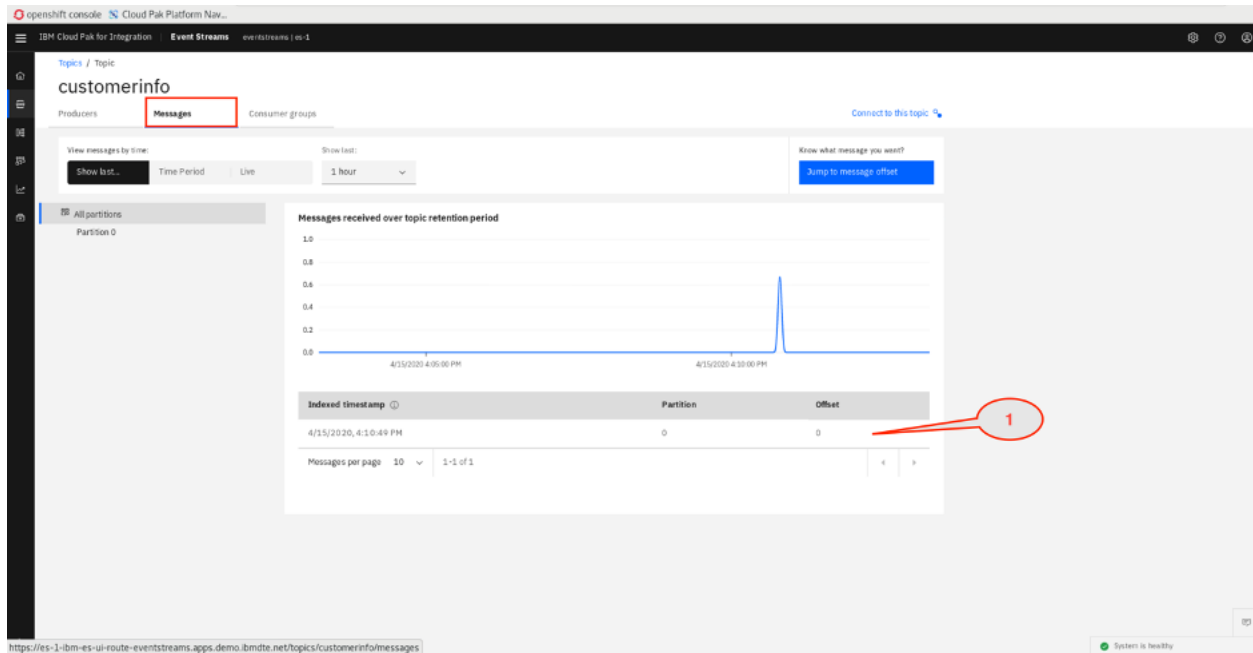
6. Click **Topics** to open the topics list of this Event Streams instance.



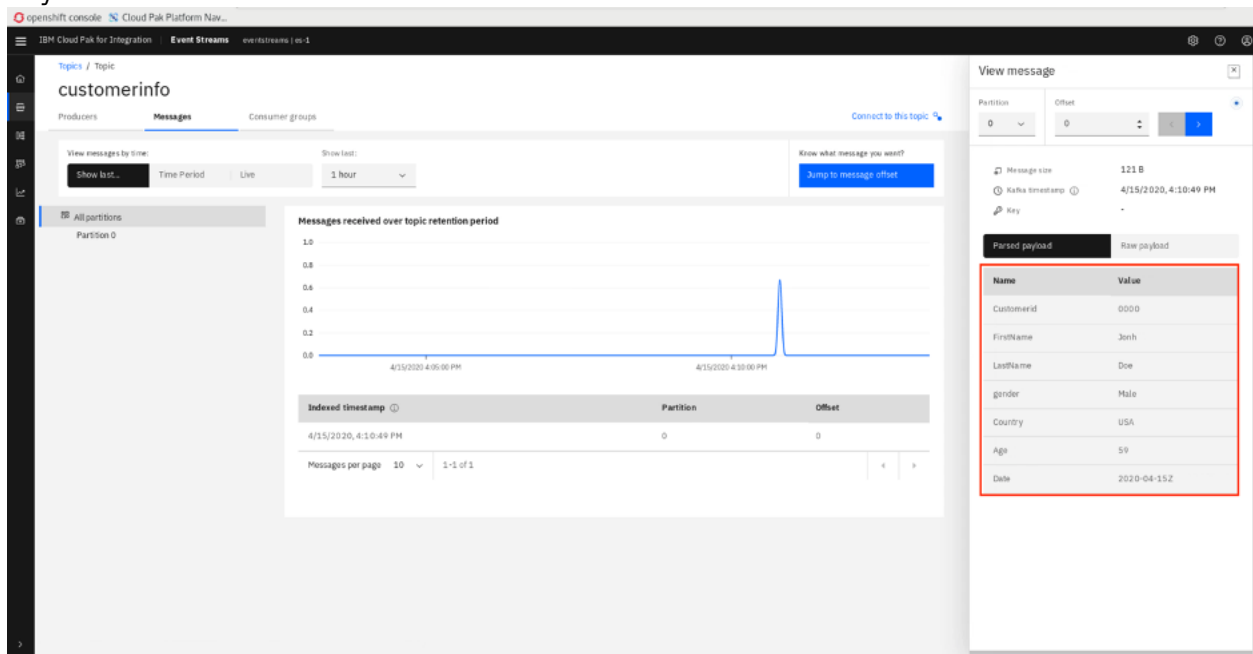
7. In the Topics page, click the topic **customerinfo** to open the topic page.



8. Click **Messages** to check if the message from App Connect Enterprise has arrived. You see the list of messages that are stored on the Event Streams topic. Take time to look at the monitor to explore the information.



9. Click the message and Verify the message on the **customerinfo** topic. Verify the Customerid: **0000**.



## Task 7 - Using Operations Dashboard (tracing)

The Operations Dashboard collects data from all the registered capabilities (such as MQ) in real time. By default, and for this lab, 10 percent of traffic is sampled.

IBM Cloud Pak for Integration Operations Dashboard has adopted [OpenTracing](#) API specification for collecting tracing data.

OpenTracing is comprised of an API specification for distributed tracing, frameworks and libraries that have implemented the specification and documentation.

- **Trace:** The description of a transaction as it moves through IBM Cloud Pak for Integration platform.
  - **Span:** A named, timed operation representing a piece of the workflow (e.g. calling an API, invoking a message flow or placing a message in a queue or a topic).
  - **Span context:** Trace information that accompanies the distributed transaction, including when it passes the service to service over the network or through a message bus.
1. Go to the **IBM Pak Cloud Integration**. Click the Hamburger Menu and select **tracing application -> tracing** instance to open the Operations Dashboard instance.

IBM App Connect

← → ↻ 🏠

🔒 <https://ace-1-ui-ace.apps.demo.ibm>

🔗 openshift console 🔗 Cloud Pak Platform Nav...

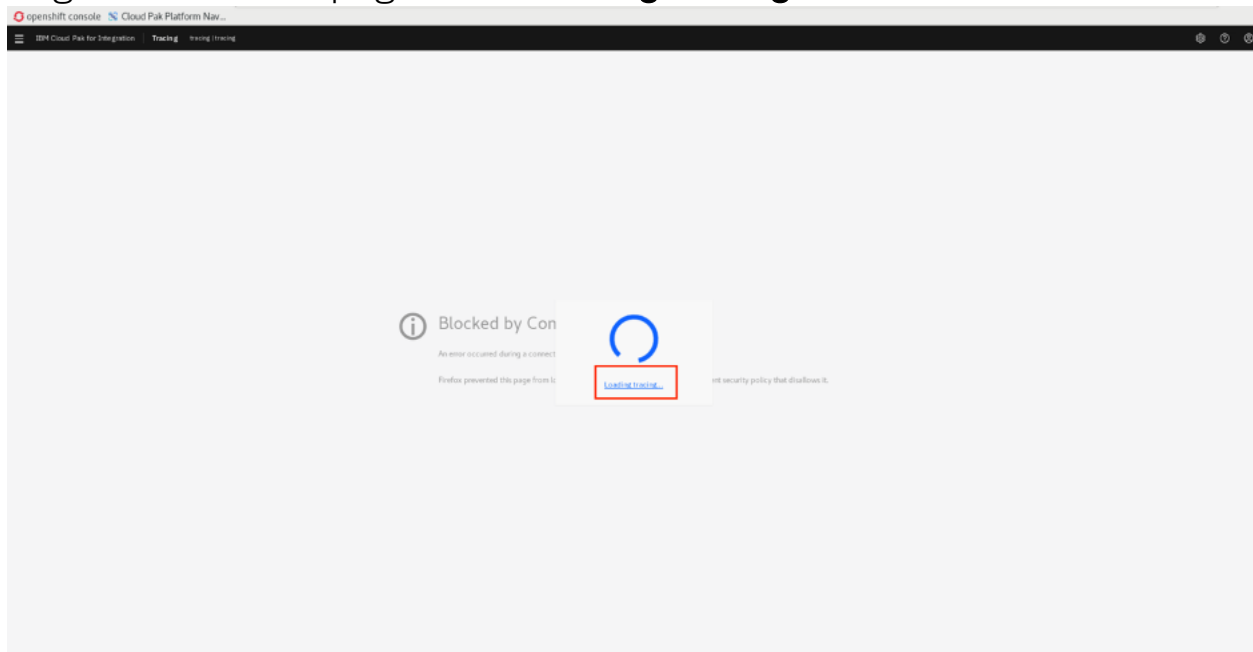
✕ IBM Cloud Pak for Integration | **App Connect Dashboard** ace | ace-1

🏠 Platform home	🔍 Tracing
🔗 Cloud Pak Foundation	🔍 Find
🔗 OpenShift Console	tracing
🔗 Logging	tracing ⋮
🔗 Monitoring	
🔗 API Connect 1 instance	
🔗 App Connect 2 instances	
🔗 Aspera 0 instances	
🔗 DataPower 1 instance	
🔗 Event Streams 1 instance	
🔗 MQ 2 instances	
🔗 Tracing 1 instance	
🔗 Asset repository 1 instance	

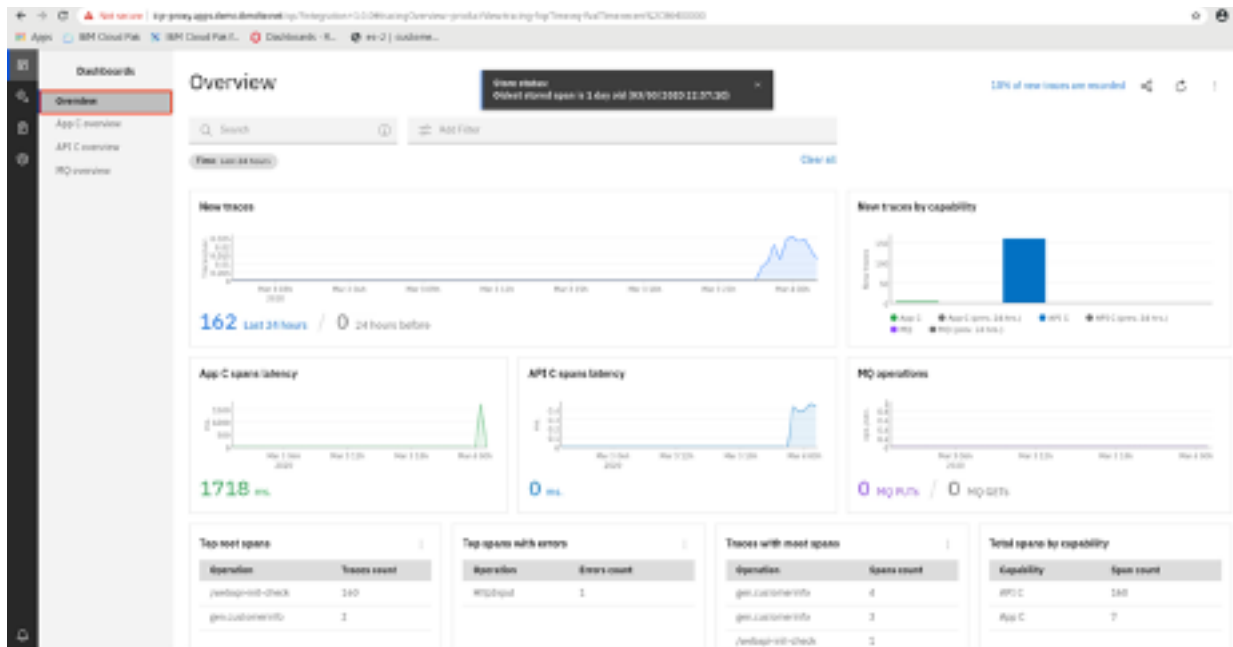
Note:



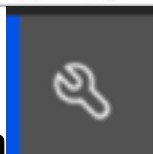
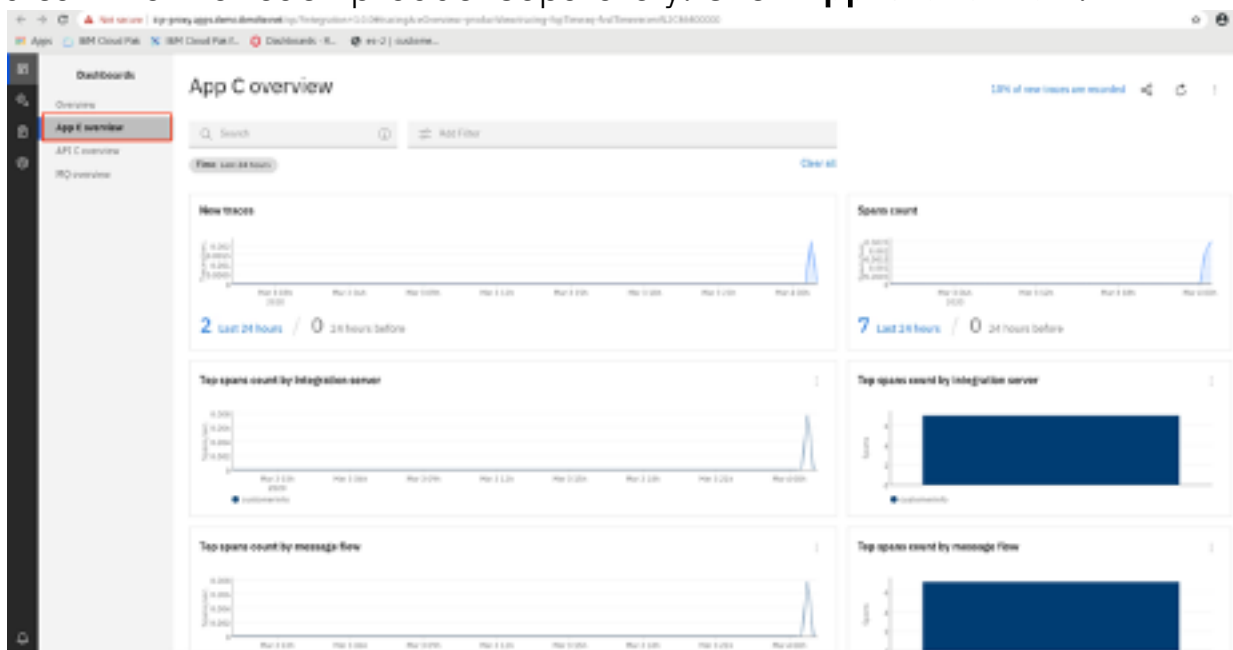
might receive this page. Click **loading tracing**.



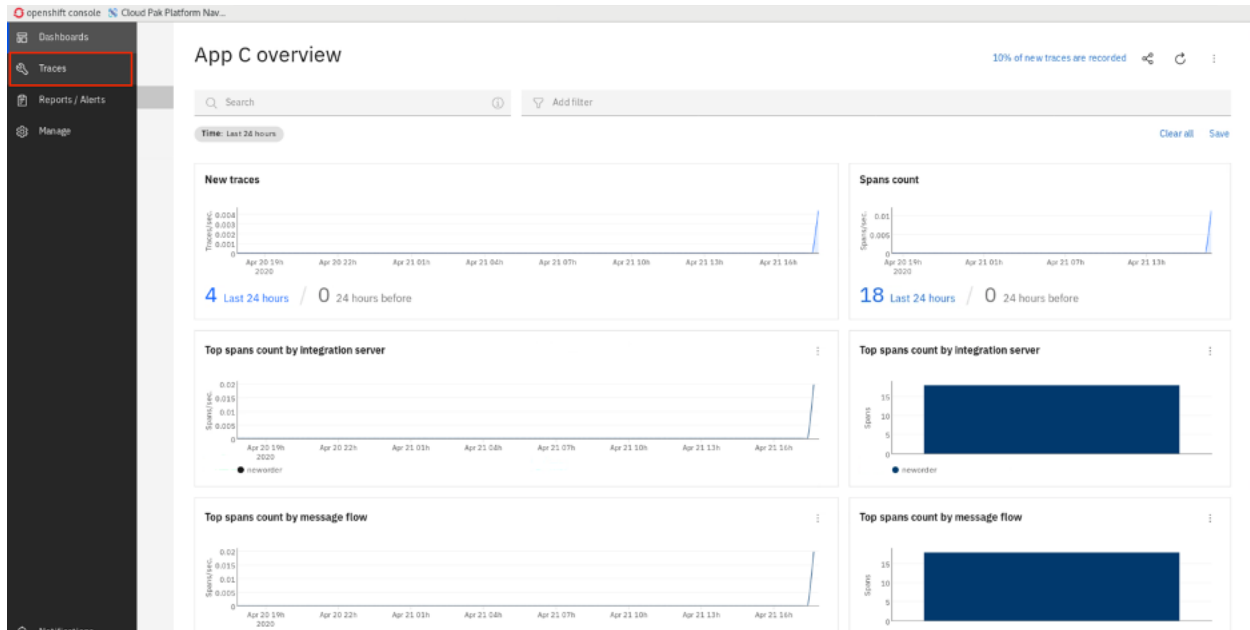
2. In the Tracing page, check the Overview page. You see all the products that you can use this tool: APIC (including DataPower), APP Connect and MQ. You see all the tracing of **MQ, App Connect and APIC** (You see how to configure tracing in APIC lab). Operations Dashboard Add-on is based on Jaeger open source project and the OpenTracing standard to monitor and troubleshoot microservices-based distributed systems. Operations Dashboard can distinguish call paths and latencies. DevOps personnel, developers, and performance engineers now have one tool to visualize throughput and latency across integration components that run on Cloud Pak for Integration. Cloud Pak for Integration - Operations Dashboard Add-on is designed to help organizations that need to meet and ensure maximum service availability and react quickly to any variations in their systems.



3. You can monitor each product separately. Click **App C overview**.



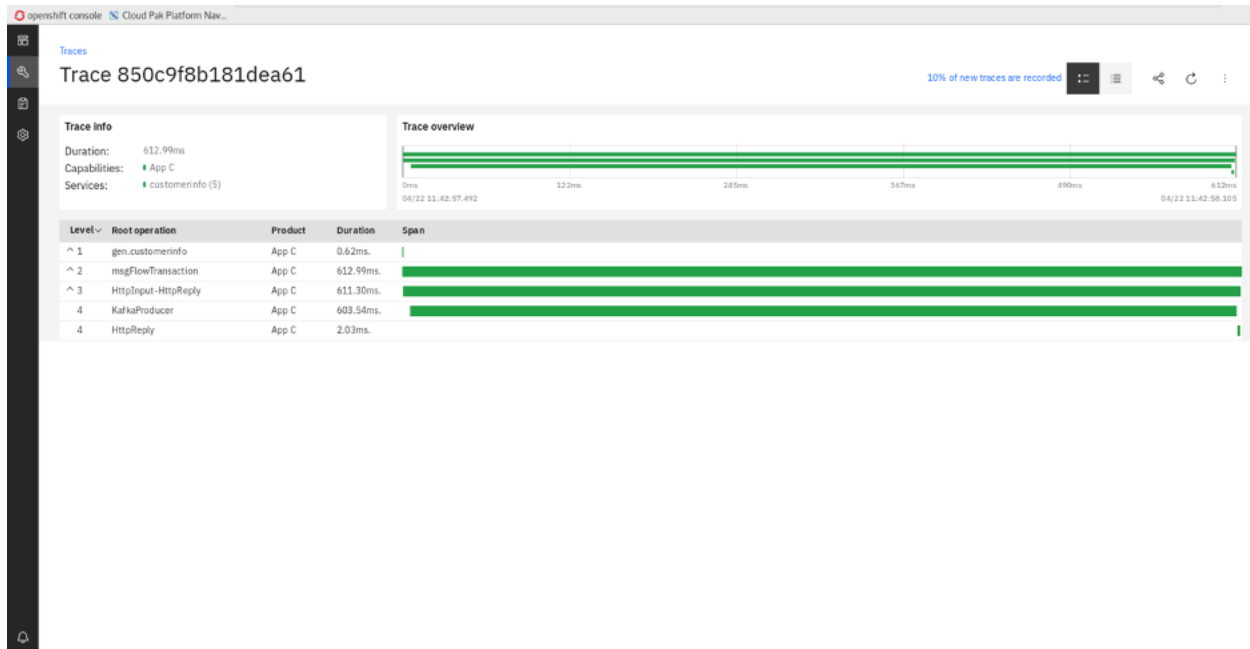
4. In the tracing page, select **traces icon** the menu on the left.



5. You see the list of tracing, select the **customerinfo** line to analyze the trace of the application **customerinfo**.

67f66a767a6219	04/22 11:52:11.008	0.44	/webapi-init-check	1	API C	apiconnect (1)
17291a2ec31f33c3	04/22 11:51:11.026	0.31	/webapi-init-check	1	API C	apiconnect (1)
16fae296e2c0d02f	04/22 11:50:11.022	0.39	/webapi-init-check	1	API C	apiconnect (1)
192de86d2974a418	04/22 11:50:07.723	0.35	/webapi-init-check	1	API C	apiconnect (1)
83f1abedc97af01	04/22 11:49:37.720	0.33	/webapi-init-check	1	API C	apiconnect (1)
440946bc1ed1428	04/22 11:49:07.721	0.48	/webapi-init-check	1	API C	apiconnect (1)
10b48842774d5e89	04/22 11:48:21.013	0.28	/webapi-init-check	1	API C	apiconnect (1)
11e8816f778d83ee	04/22 11:48:01.022	0.30	/webapi-init-check	1	API C	apiconnect (1)
437d3aeb407665	04/22 11:46:31.009	0.31	/webapi-init-check	1	API C	apiconnect (1)
15b3c4dcded7c2ba	04/22 11:46:07.719	0.46	/webapi-init-check	1	API C	apiconnect (1)
850c98b181dea61	04/22 11:42:57.492	612.99	gen.customerinfo	5	App C	customerinfo (5)
1dd354ef31e5de56	04/22 11:42:53.097	536.52	gen.customerinfo	5	App C	customerinfo (5)
15d074ecffef71e	04/22 11:42:51.009	0.41	/webapi-init-check	1	API C	apiconnect (1)
5b89abe30b5876d	04/22 11:42:43.677	3773.86	gen.customerinfo	5	App C	customerinfo (5)
ee6ae24020c9d141	04/22 11:42:35.767	2.82	gen.customerinfo	3	App C	customerinfo (3)
ee07fceb2f67bda	04/22 11:41:41.008	0.41	/webapi-init-check	1	API C	apiconnect (1)
758212d34138884	04/22 11:40:37.719	0.43	/webapi-init-check	1	API C	apiconnect (1)
134fffb48b43453e	04/22 11:39:41.016	0.39	/webapi-init-check	1	API C	apiconnect (1)
70564451ca3b90f	04/22 11:39:17.719	0.47	/webapi-init-check	1	API C	apiconnect (1)
18b76e5fb454ce5b	04/22 11:39:11.010	0.29	/webapi-init-check	1	API C	apiconnect (1)
65b272190790017	04/22 11:38:17.721	0.32	/webapi-init-check	1	API C	apiconnect (1)
18aa3ab07ae538b0	04/22 11:37:27.722	0.45	/webapi-init-check	1	API C	apiconnect (1)
14b2fb41c020d5df	04/22 11:35:37.726	0.33	/webapi-init-check	1	API C	apiconnect (1)
16bbaee129674c6	04/22 11:34:51.012	0.29	/webapi-init-check	1	API C	apiconnect (1)
5c9bb38bc1cc38	04/22 11:32:41.027	0.35	/webapi-init-check	1	API C	apiconnect (1)
e0f4a5f0f0d0540f	04/22 11:31:37.721	0.46	/webapi-init-check	1	API C	apiconnect (1)

6. Observe the trace results.



## Summary

You have successfully completed this lab. In this lab you learned how to:

- Create a topic in Event Streams
- Create an integration between an API service and Event Streams.
- Deploy the new integration as containers in Kubernetes.
- Use Operations Dashboard tool

Now that you've created a topic in Kafka (Event Streams), applications are able to subscribe and received data. To try out more labs, go to Cloud Pak for Integration Demos. For more information about the Cloud Pak for Integration, go to <https://www.ibm.com/cloud/cloud-pak-for-integration>.