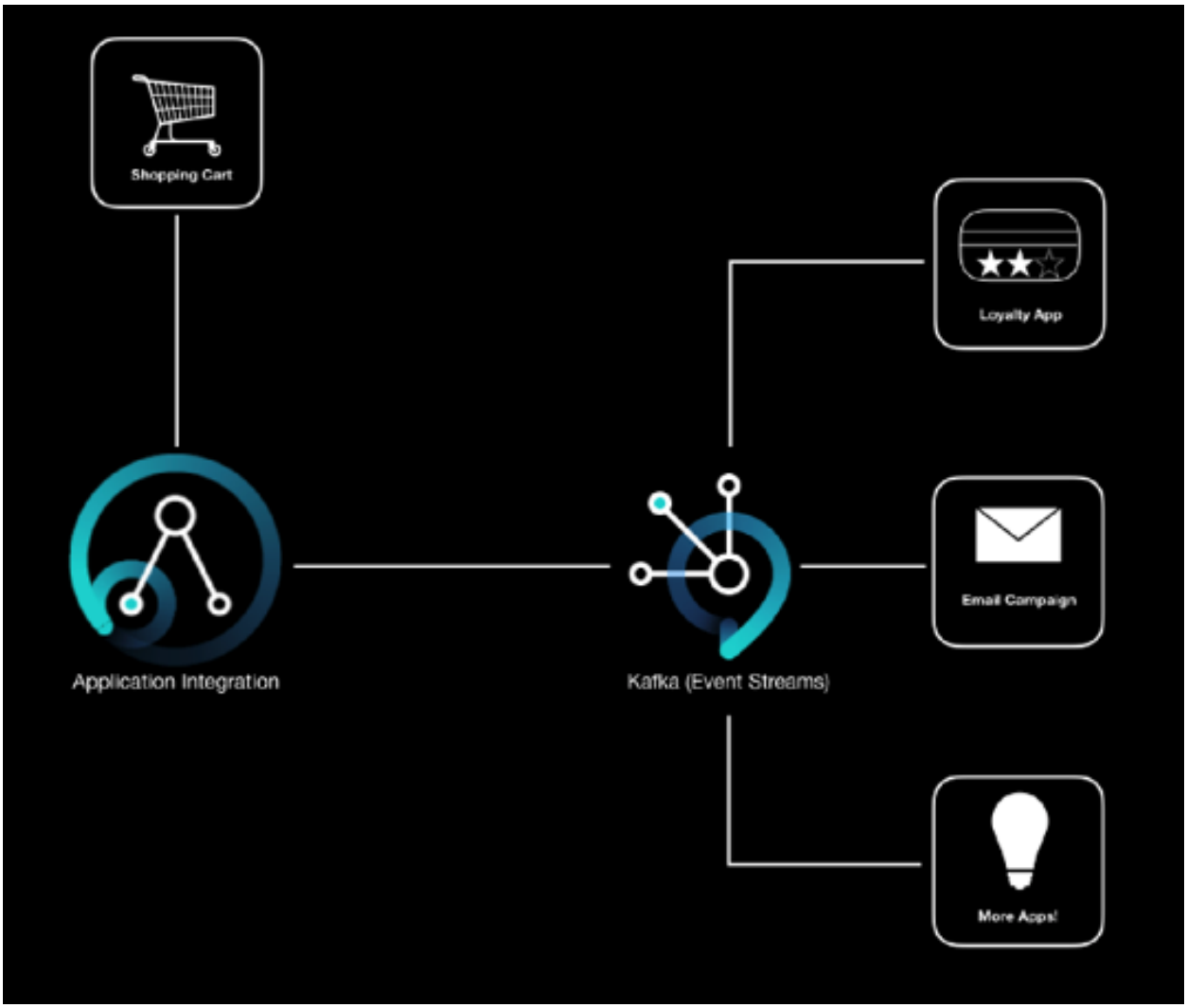
Augment existing business functions with new applications using Kafka

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

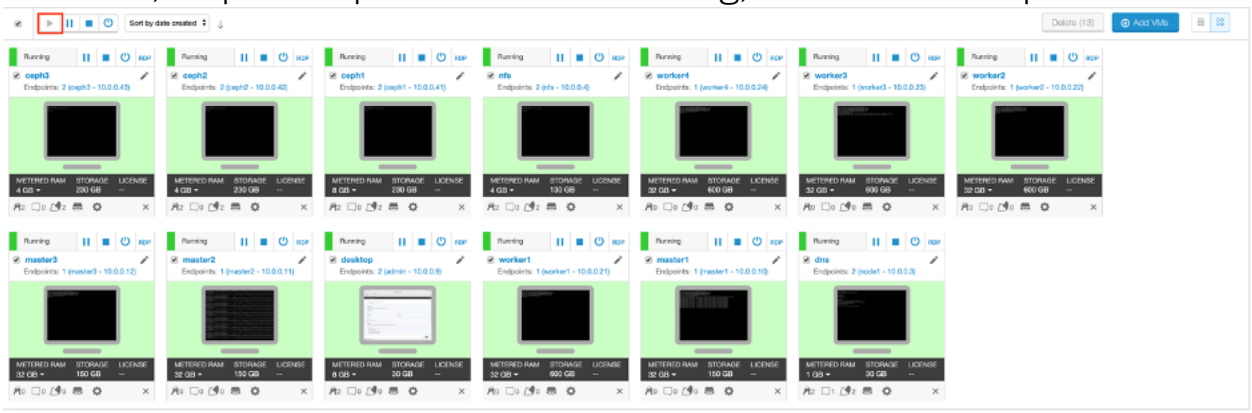
- Checking the message using Operations Dashboard (tracing).

  Task 1 - Preparing IBM Cloud Pak for Integration Environment

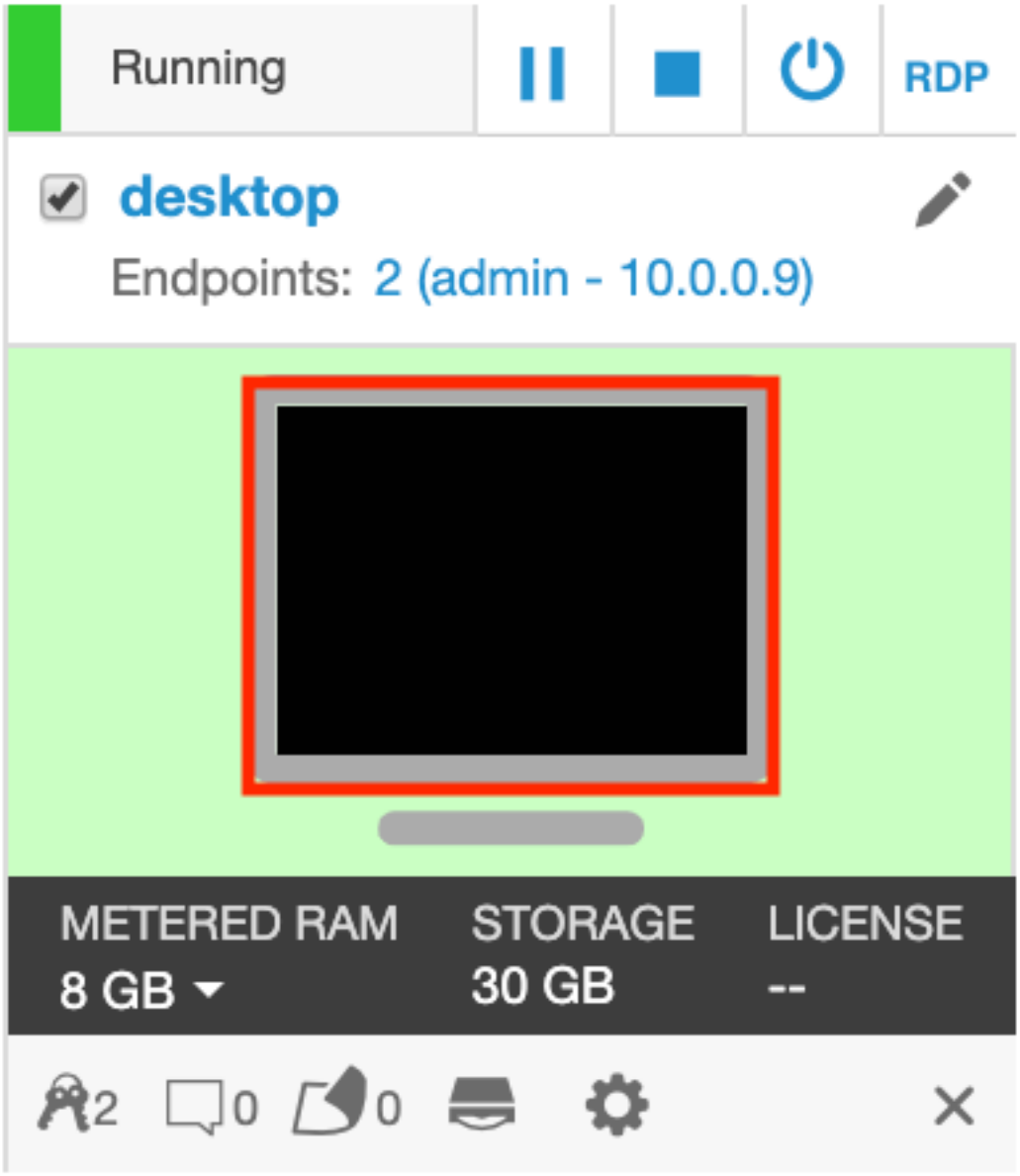
  The demo runs on a virtual machine that is provided by IBM Demos. To get access, reserve an instance in the **Before you start** section on this tutorial page. You then receive an email with the URL and password to access the virtual machine.

  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.

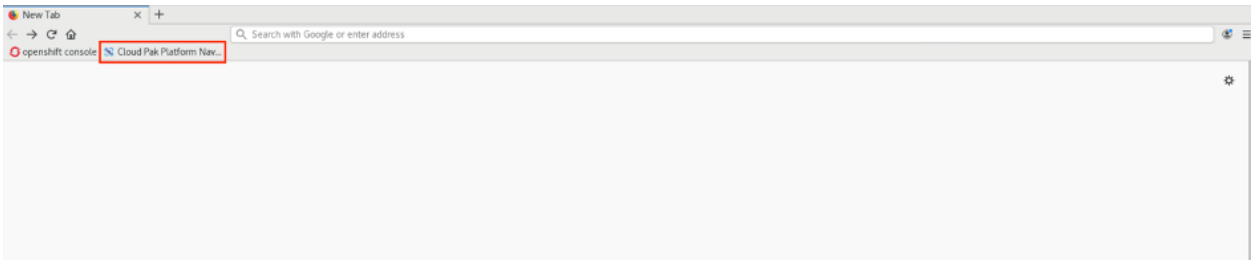


2. Once the virtual machines have started, start IBM Cloud Pack. Click the **Desktop** Machine screen image to start your lab exercise.

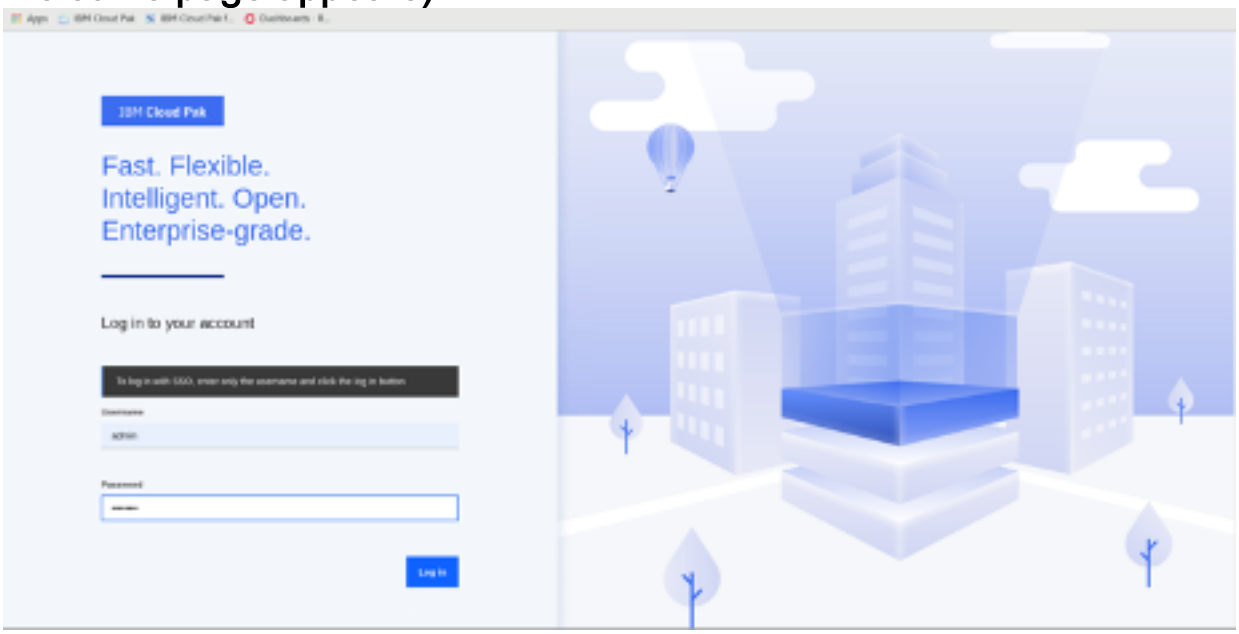


3. To access the Developer machine, log in: userid: **ibmuser** and password: **engageibm.**

4. Open a browser and click IBM Cloud Pak for Integration Bookmark bar.



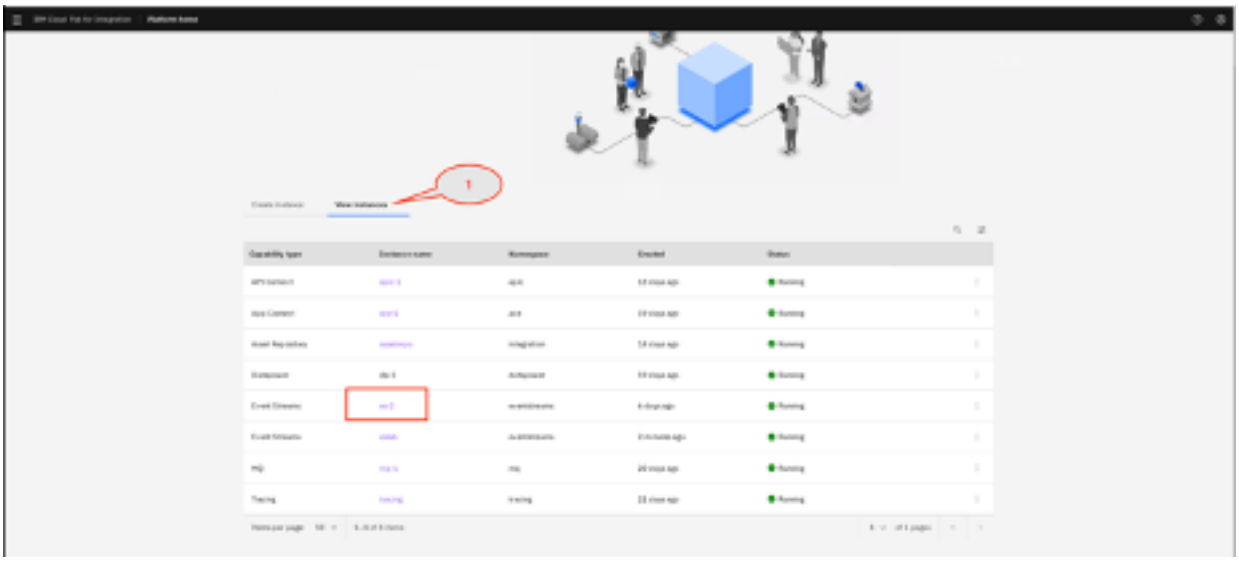
5. **Log in to IBM Cloud Pak for Integration.** The username and password are already filled in. If not, ensure username: **admin** and password: **passw0rd** is entered. Click **Log in** and then click **Skip Welcome (if Welcome page appears).**



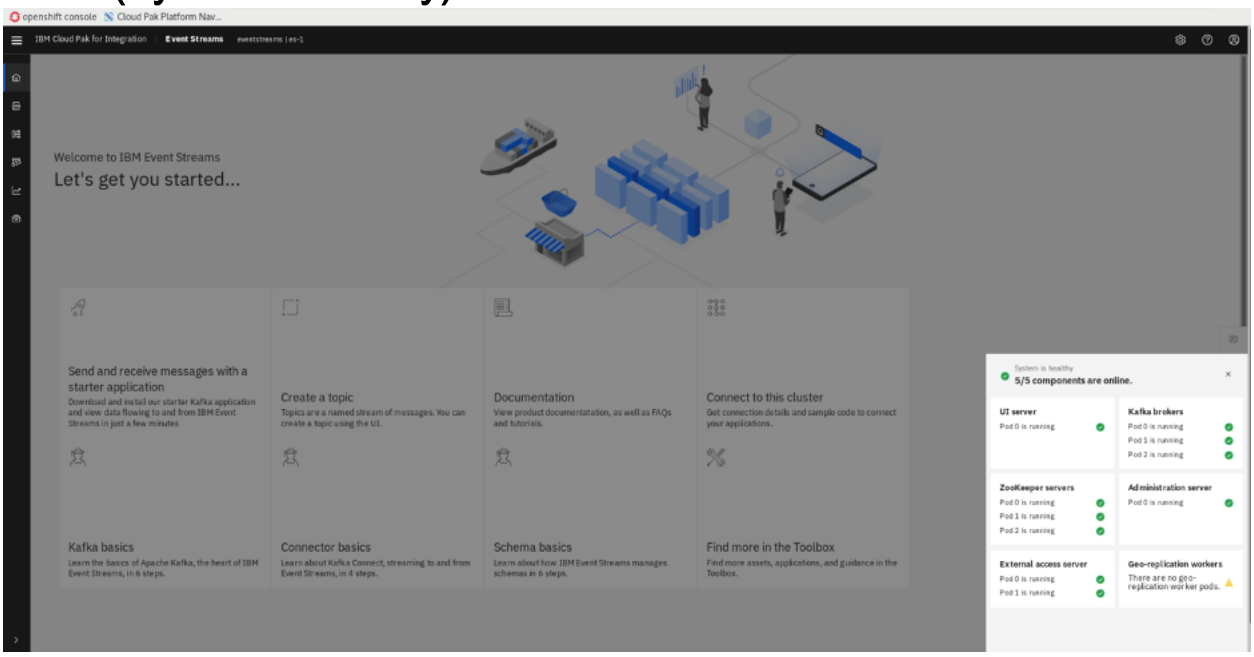
Task 2 – Creating and configuring an Event Streams Topic

Creating an Event Streams topic in the existing Event Streams instance.

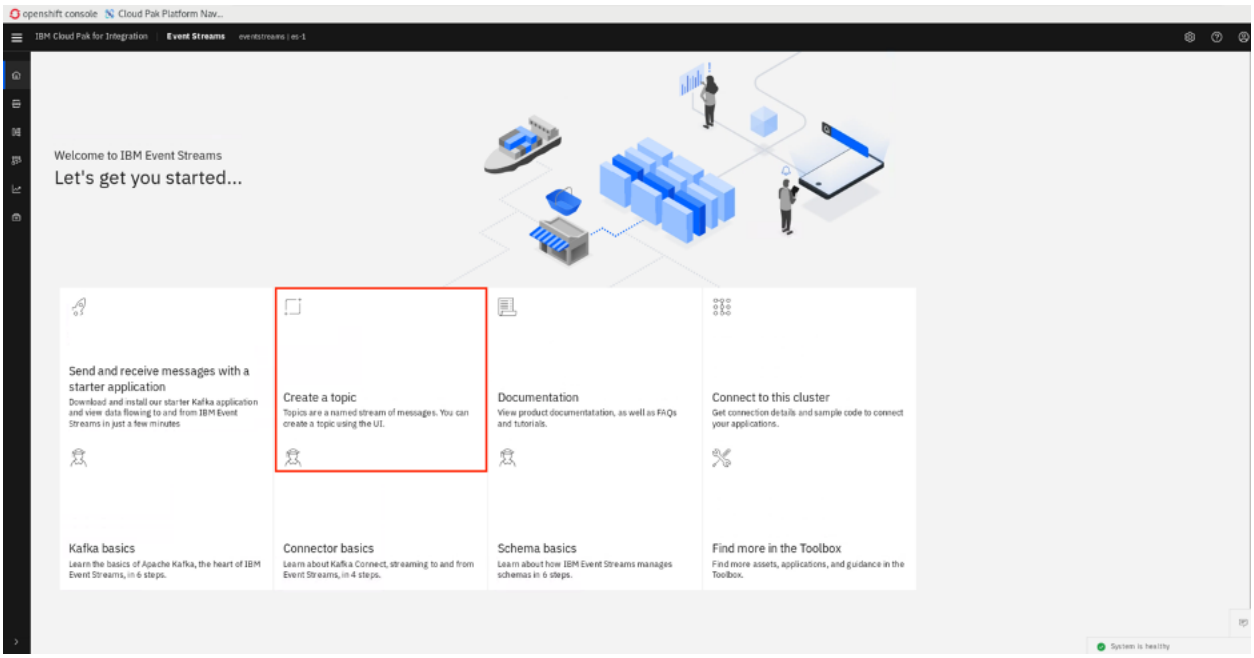
1. On the **IBM Cloud Pak for Integration** main page, click **View Instances. The list of i**ntegration products deployed is displayed (API Connect, App Connect, Asset Repository, DataPower, Event Streams, MQ, and Tracing).
2. Click **es-1** to use the existing Event Streams instance.



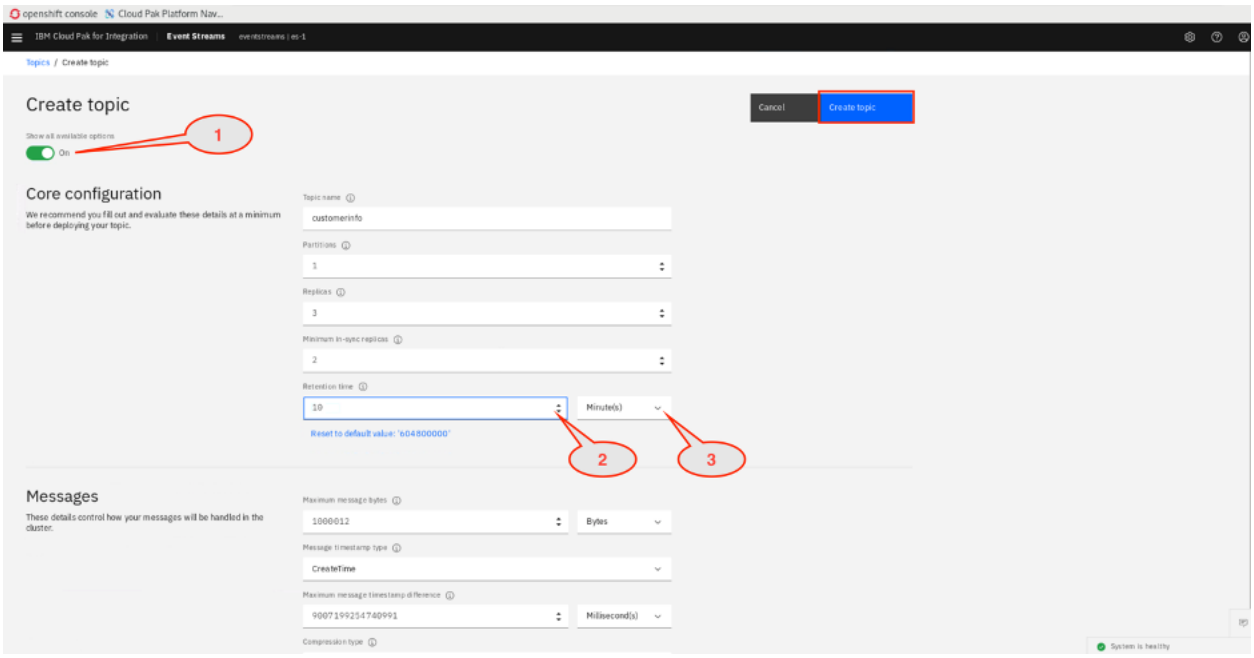
3. On the IBM Event Streams main page, check the status of the instance **(System is healthy).**



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

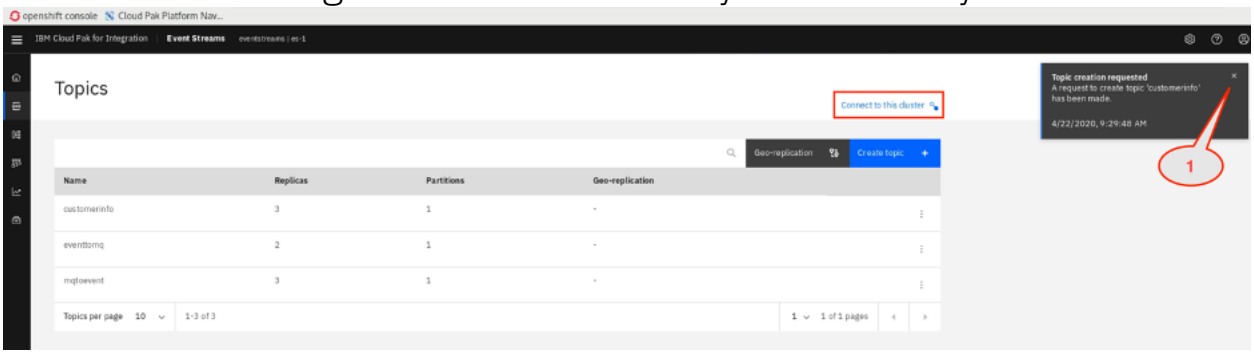


5. View the full range of configuration options by setting the **Show all available options** to **on**.
   Create the topic as follow***s:***
1. In **Show all available options**, change to **On.**
2. Topic name: **customerinfo** .
3. Set **Retention time:** ***10 Minutes. (***This is how long messages are retained before they are deleted).
4. Click **Create topic.**

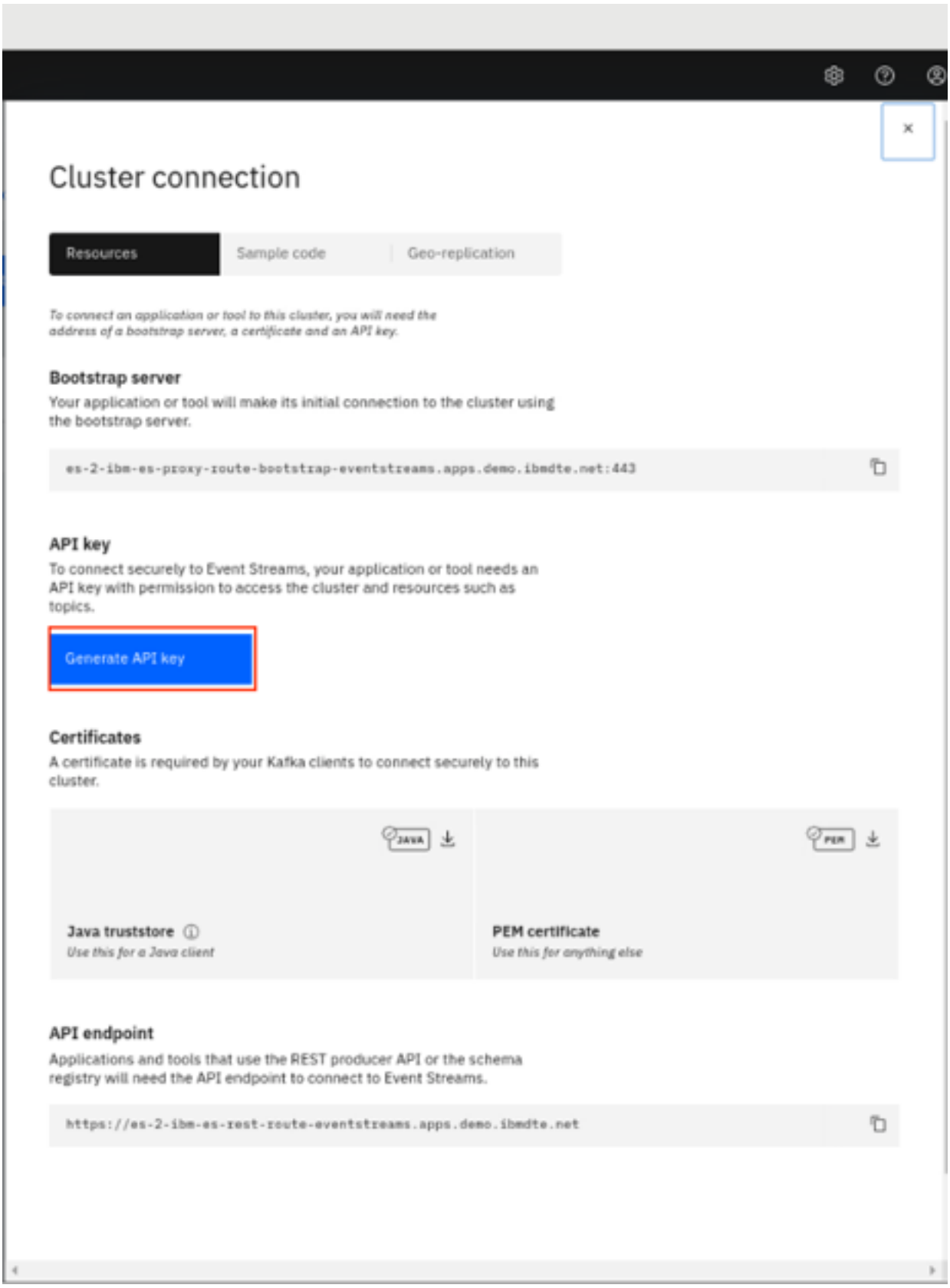


The Topics page is displayed**.** Your new topic is displayed along with a completion notification.

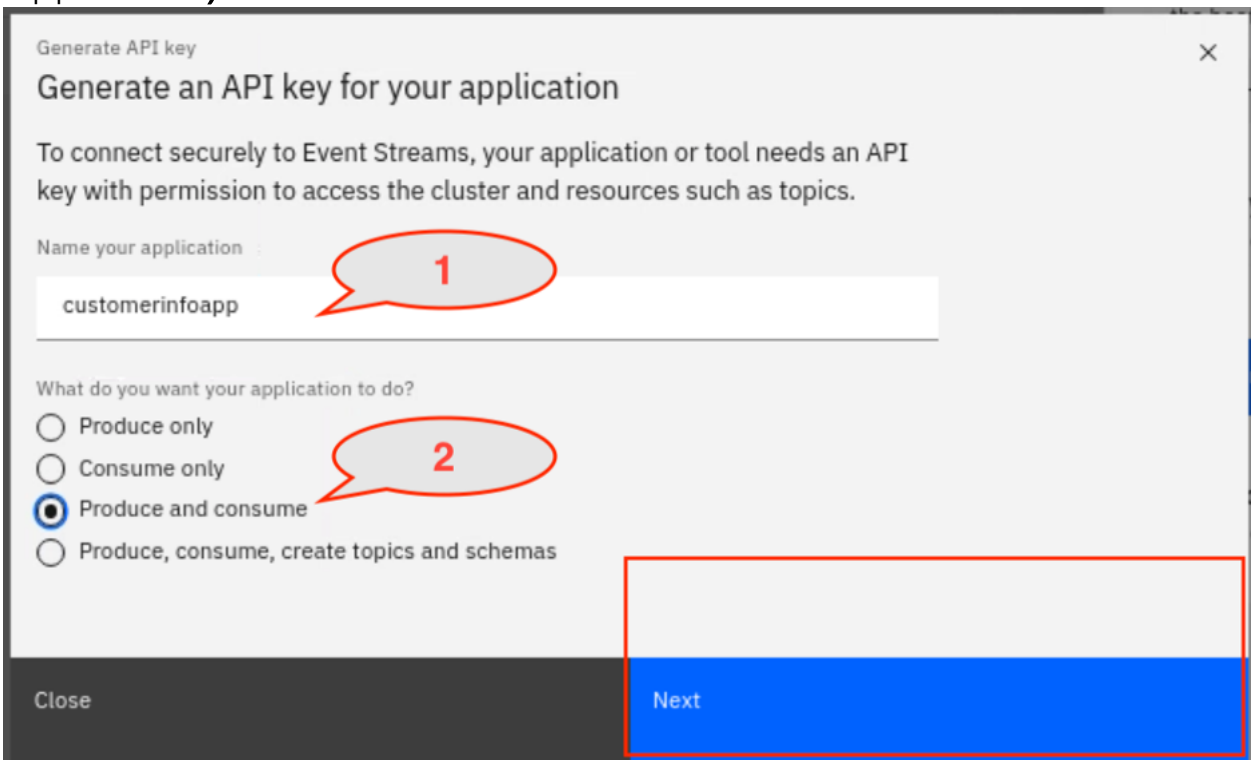
6. The topic has created, close the pop-up window and click to **connect to this cluster**, you will need an **API key** that allows access to the topic, the Bootstrap server address to connect to the cluster, and the **PEM certificate** for securing the connection from your client to your cluster.



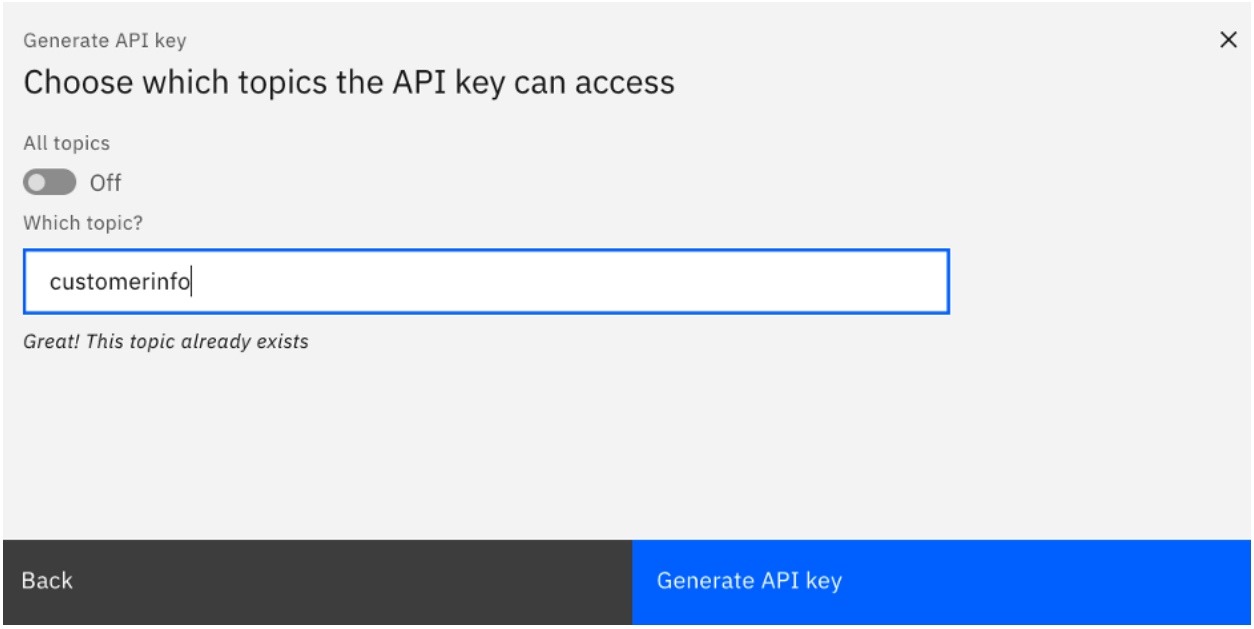
1. Click **Generate API Key** to create an API key.


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

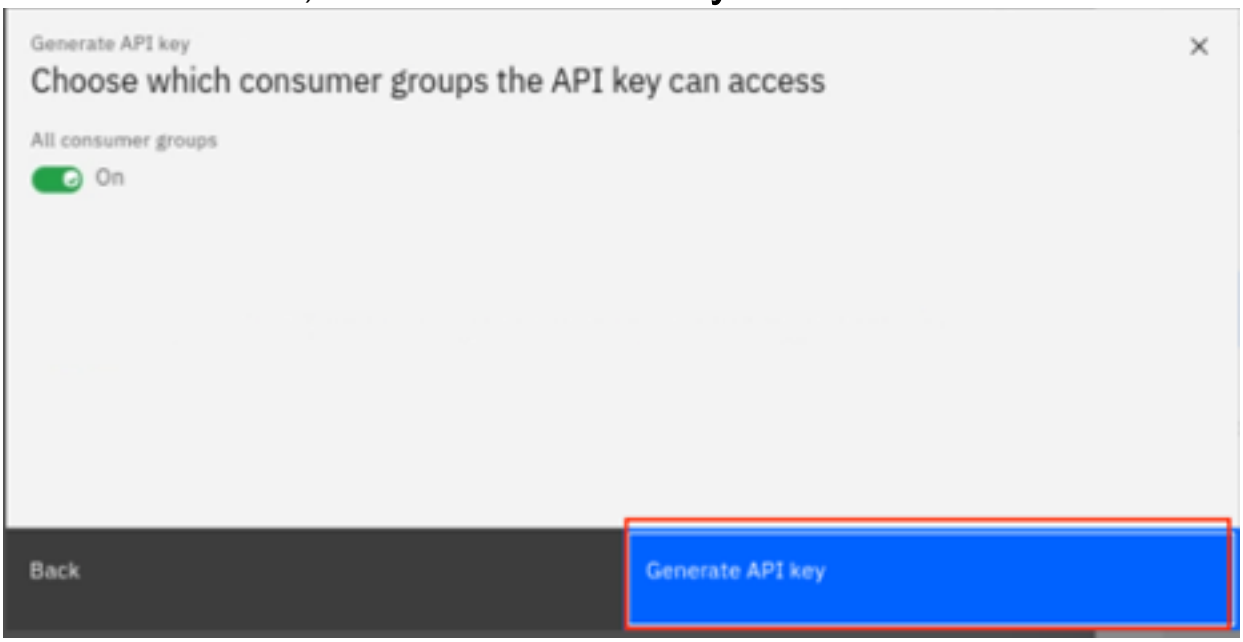
2. Enter the name of your application on the window (that is, **customerinfoapp**). Select **Produce and consume** for **What do you want your application to do ? (**you can use Produce if you send a message from an Application**)** and click **Next**. .



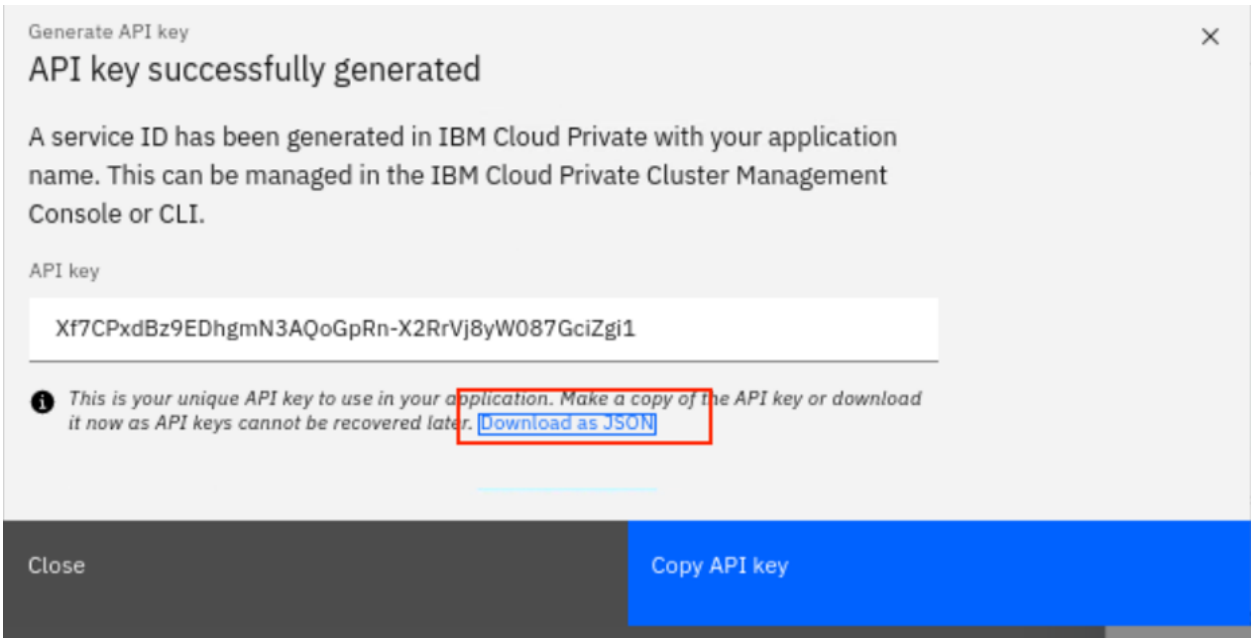
3. Enter the name of the topic **customerinfo**, and click **Generate API key**. Event Streams generates the API key.



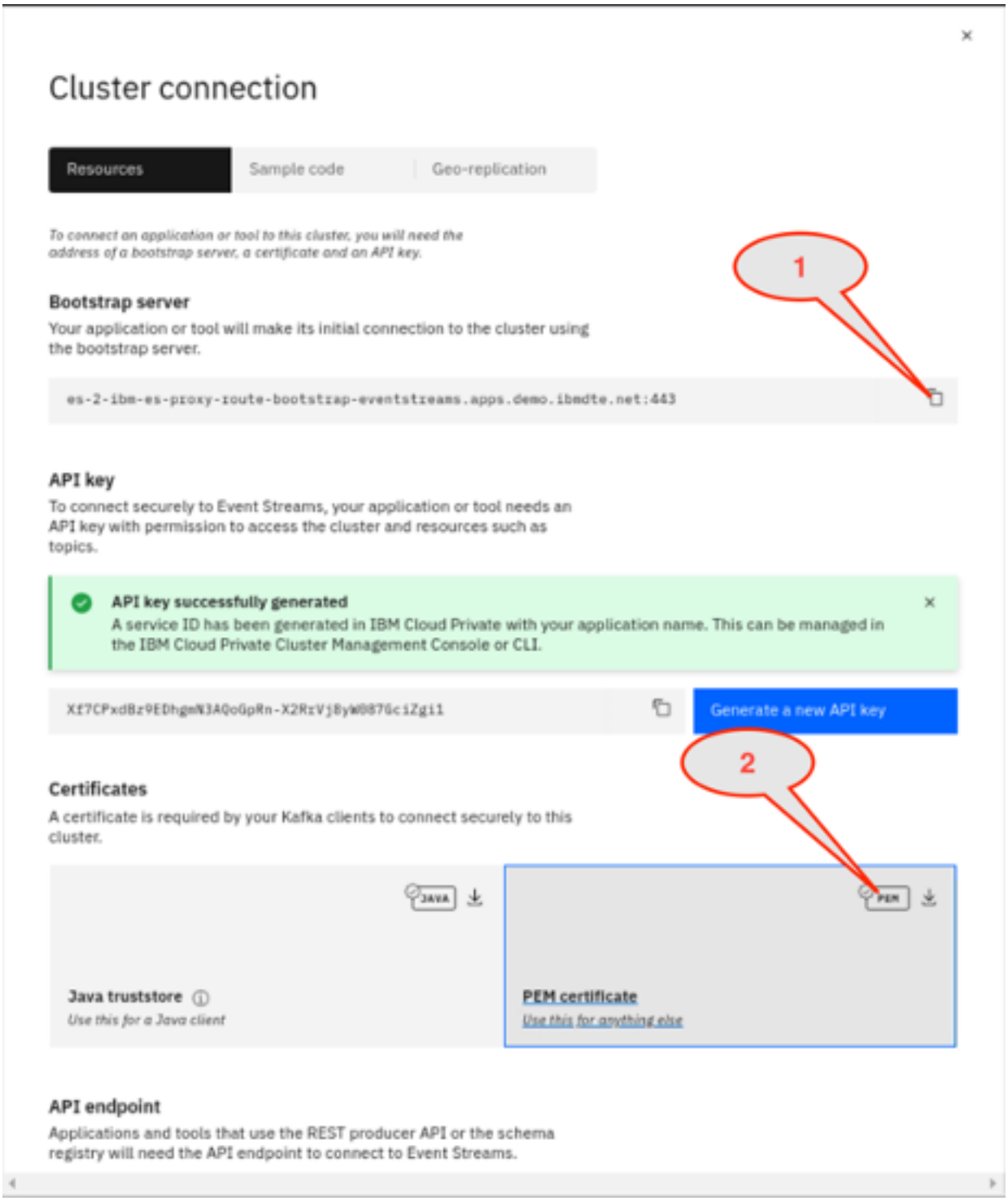
4. On the next window, click **Generate API key.**



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).

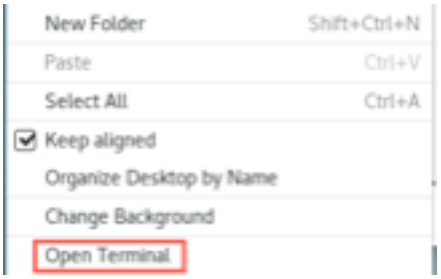


Task 3 - Configuring App Connect Enterprise flow using App Connect Enterprise Toolkit

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

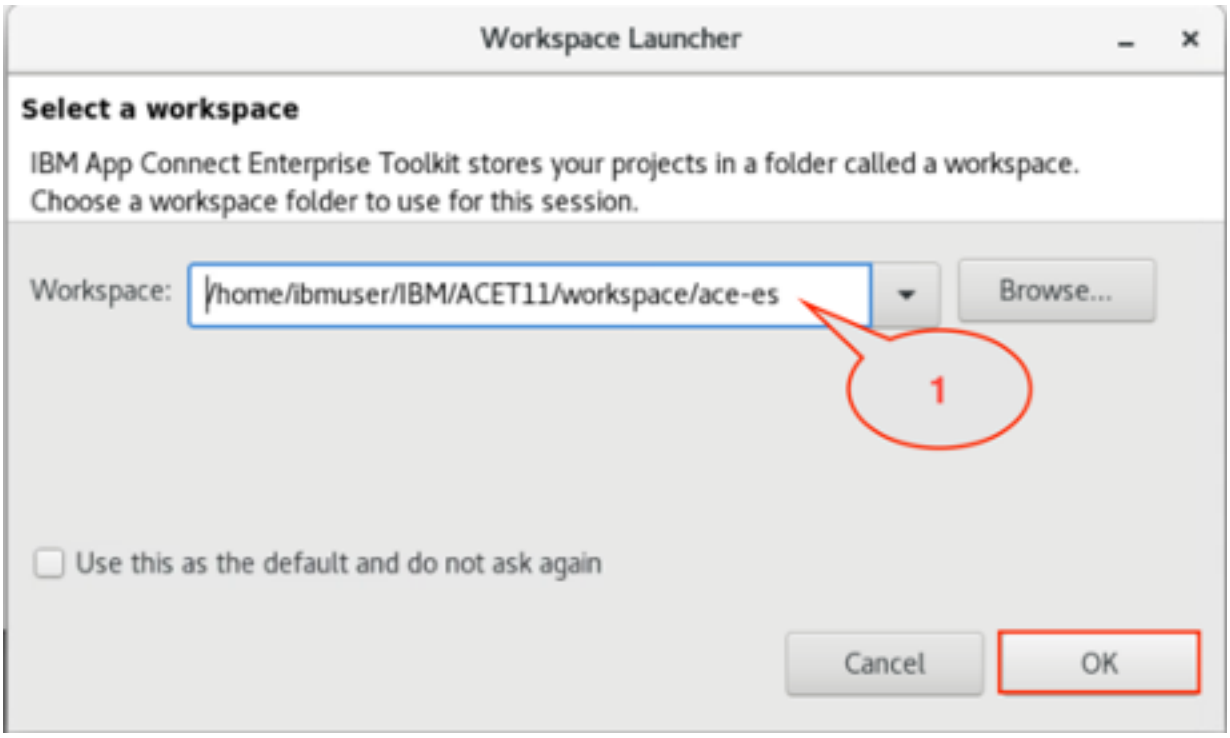
flow and generate a BAR file to deploy in the App Connect Enterprise Dashboard.

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

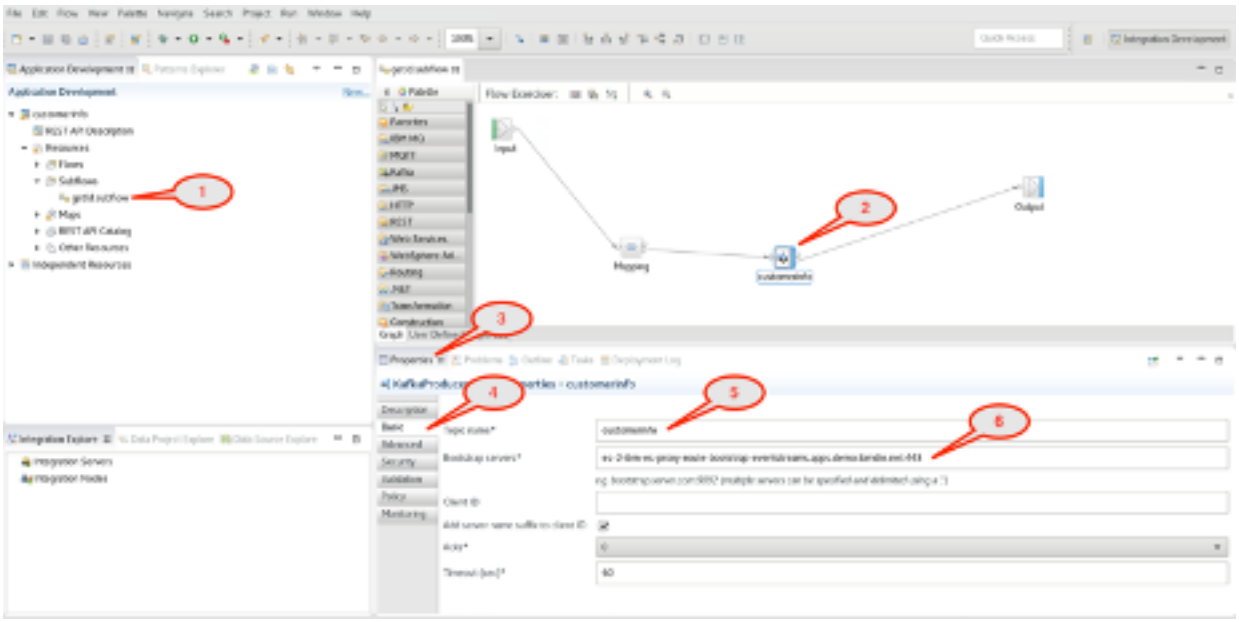


2. You have configured Event Streams configured. Open a terminal window (Developer Machine). Enter **ace toolkit .**

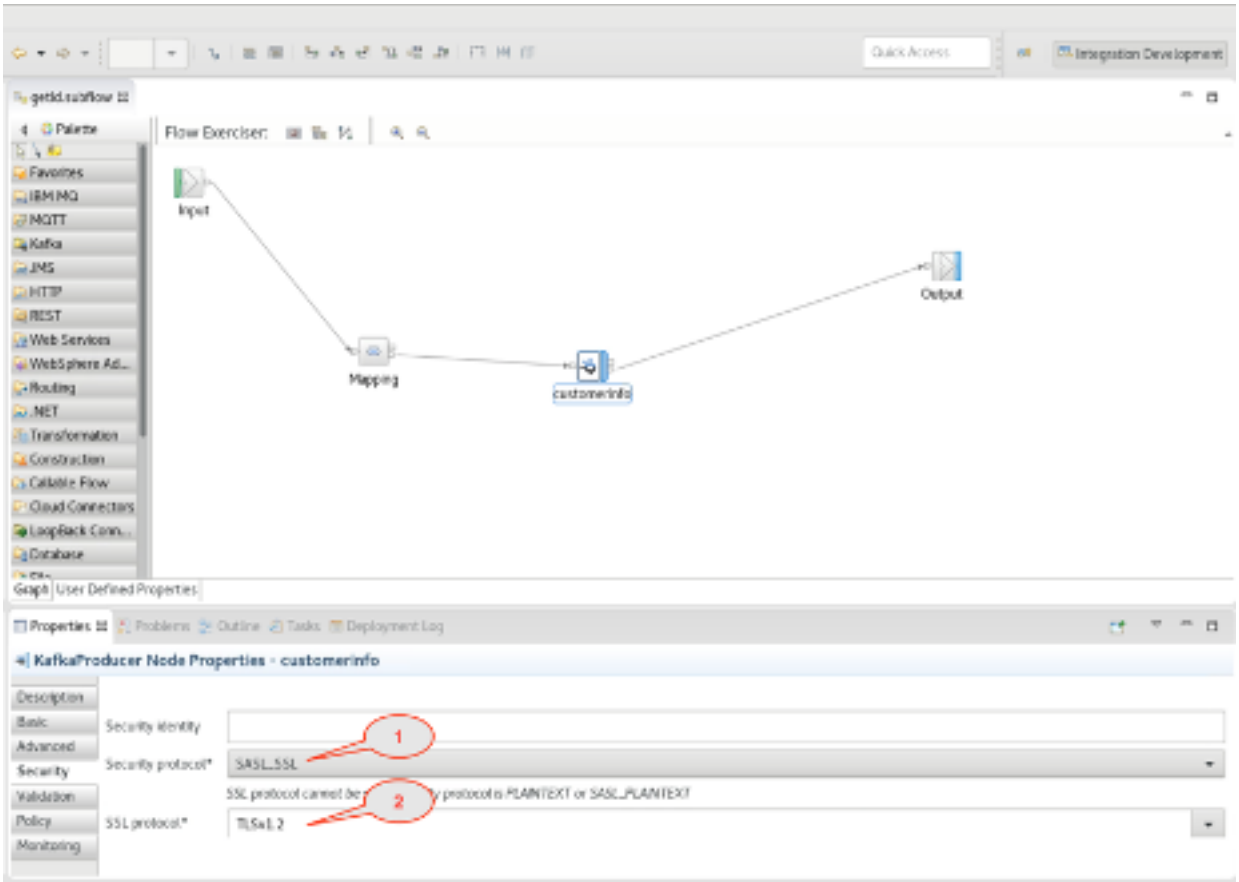
3. Check the App Connect Enterprise workspace directory routes to the ace-es folder (/home/ibmuser/IBM/ACET11/workspace/ace-es). Click **OK** to open App Connect Enterprise toolkit.



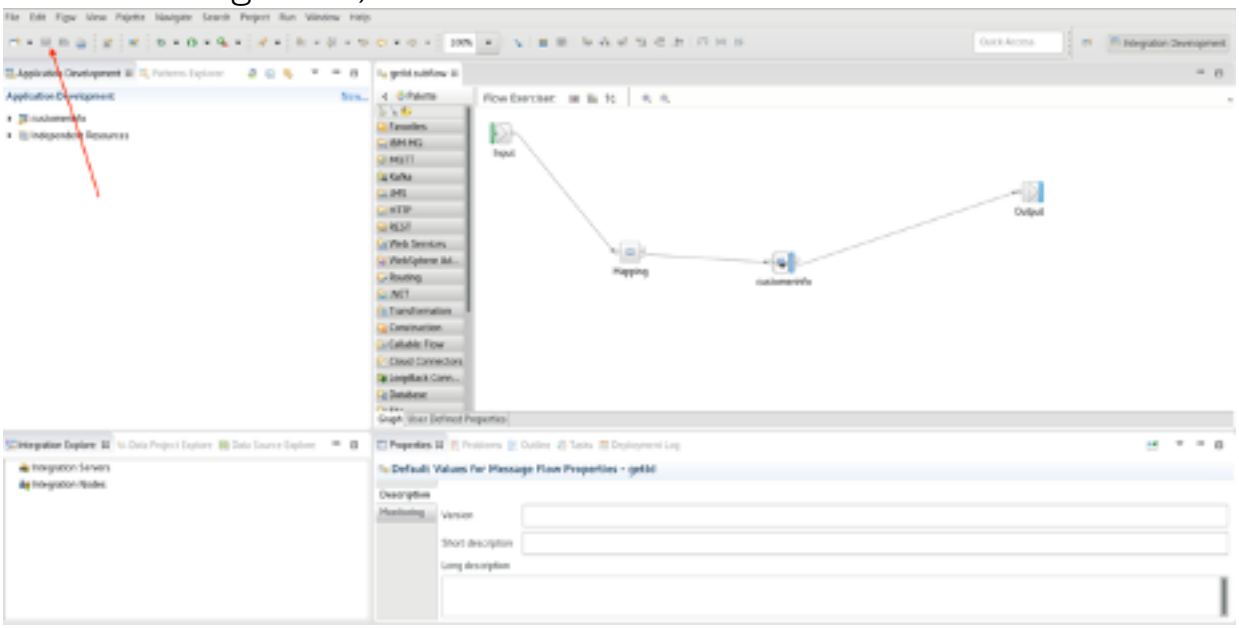
4. In the Application Developer on the left bar, select **customerinfo --> Resources --> Subflows** and click **getid.subflow.** Some errors might show up, you will fix this after you complete and save message flow.
1. Select **customerinfo** node (Kafka Producer node)
2. Click **Properties.**
3. Select **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 C**onnect to this cluster->Cluster connection**)



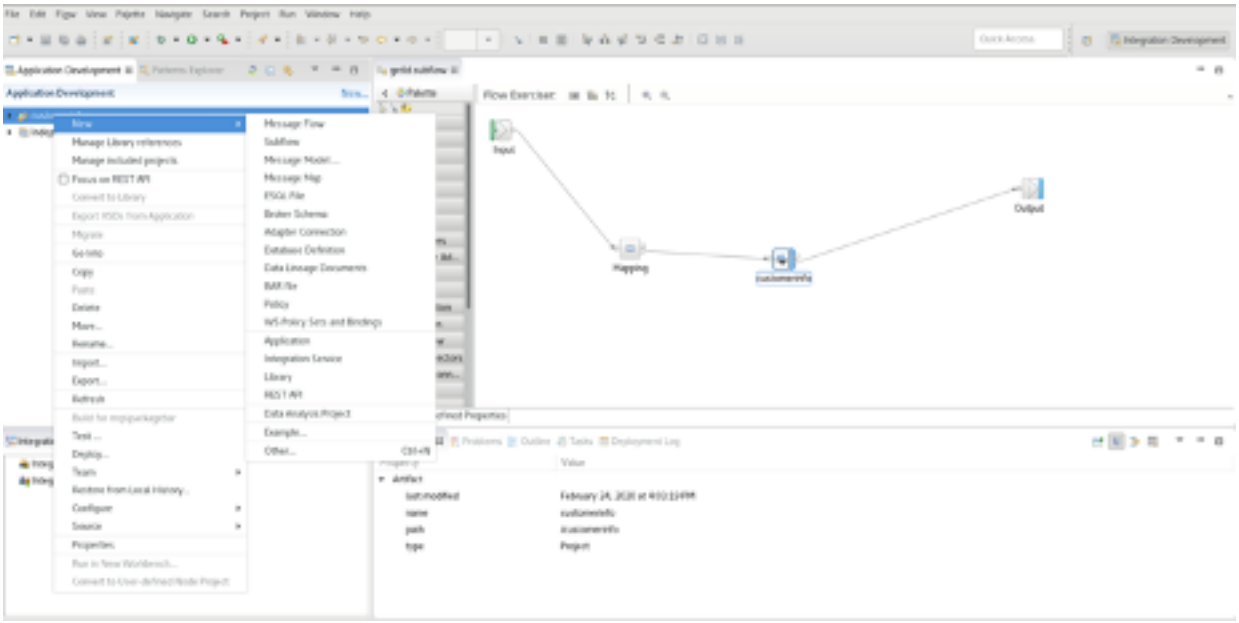
5. In the Security Tab, set the Security Protocol to **SASL SSL** and SSL protocol to **TLSv1.2 .**



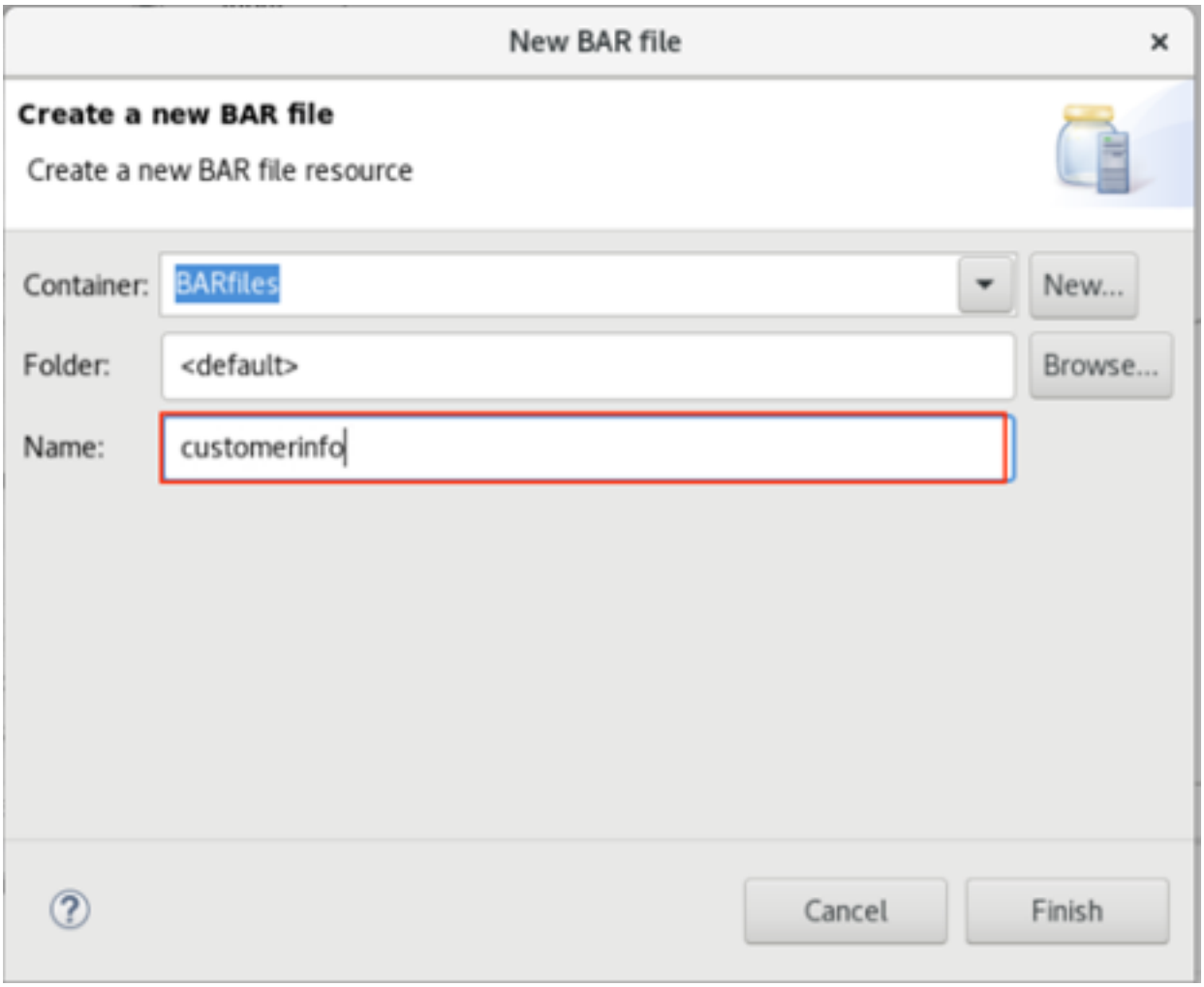
6. Save the message flow, click the **Save** button.



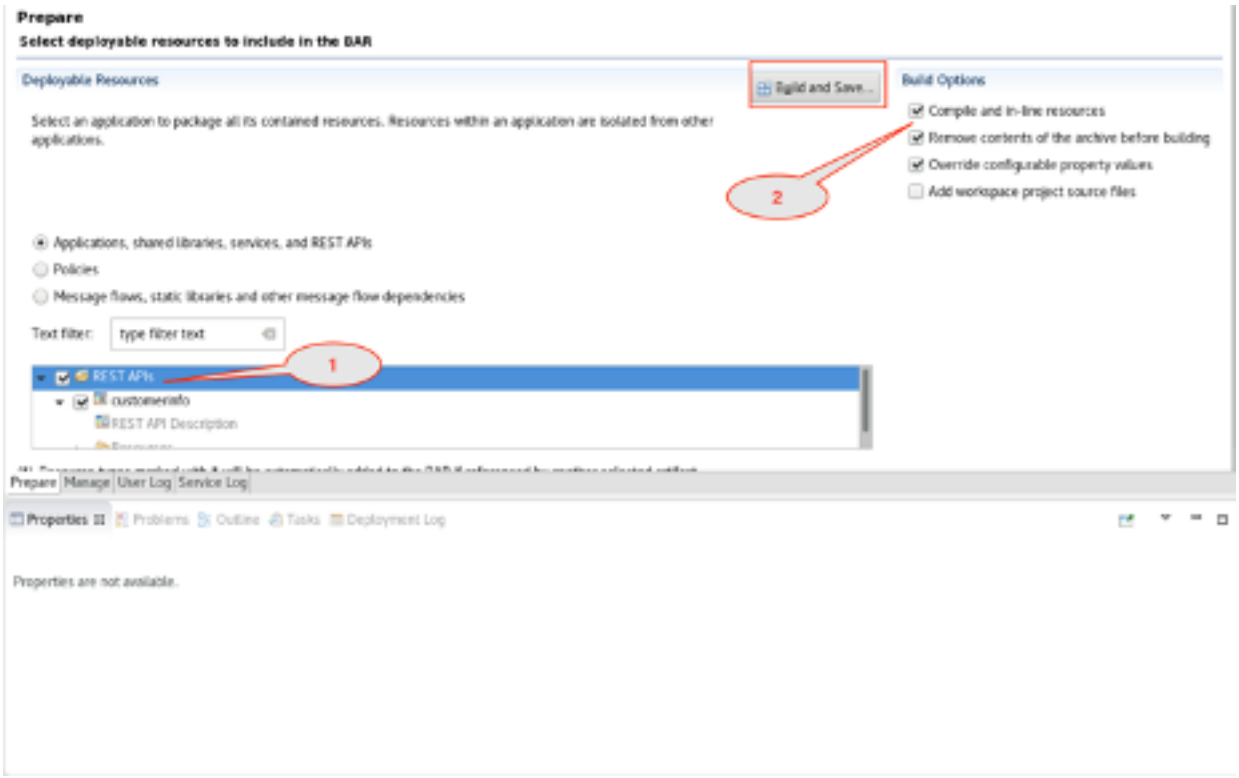
7. You need to deploy the customerinfo application in App Connect Enterprise server. Select **the customerinfo application**. Click **File -> New -> BAR file**.



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



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**.

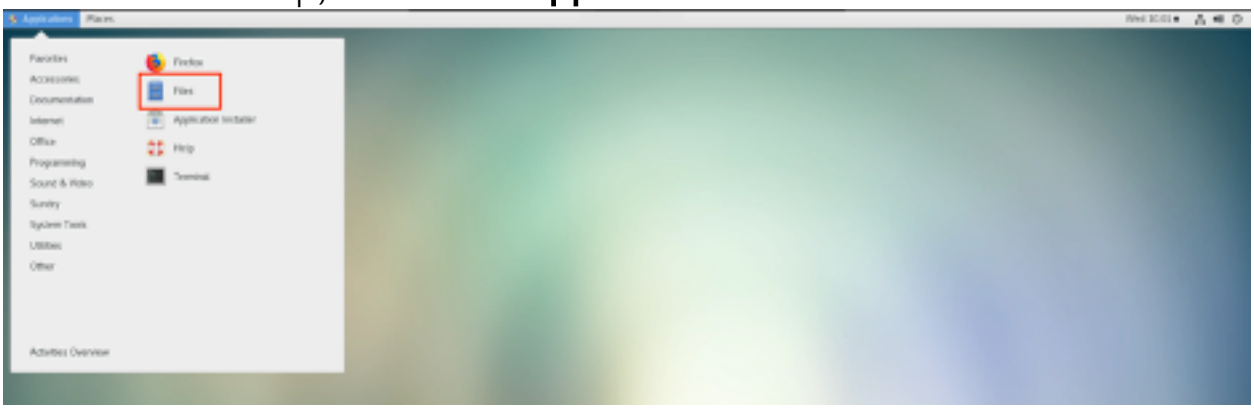


10. A pop-up window display the message “Operation completed successfully.” Click **Ok** to confirm and close the App Connect Enterprise

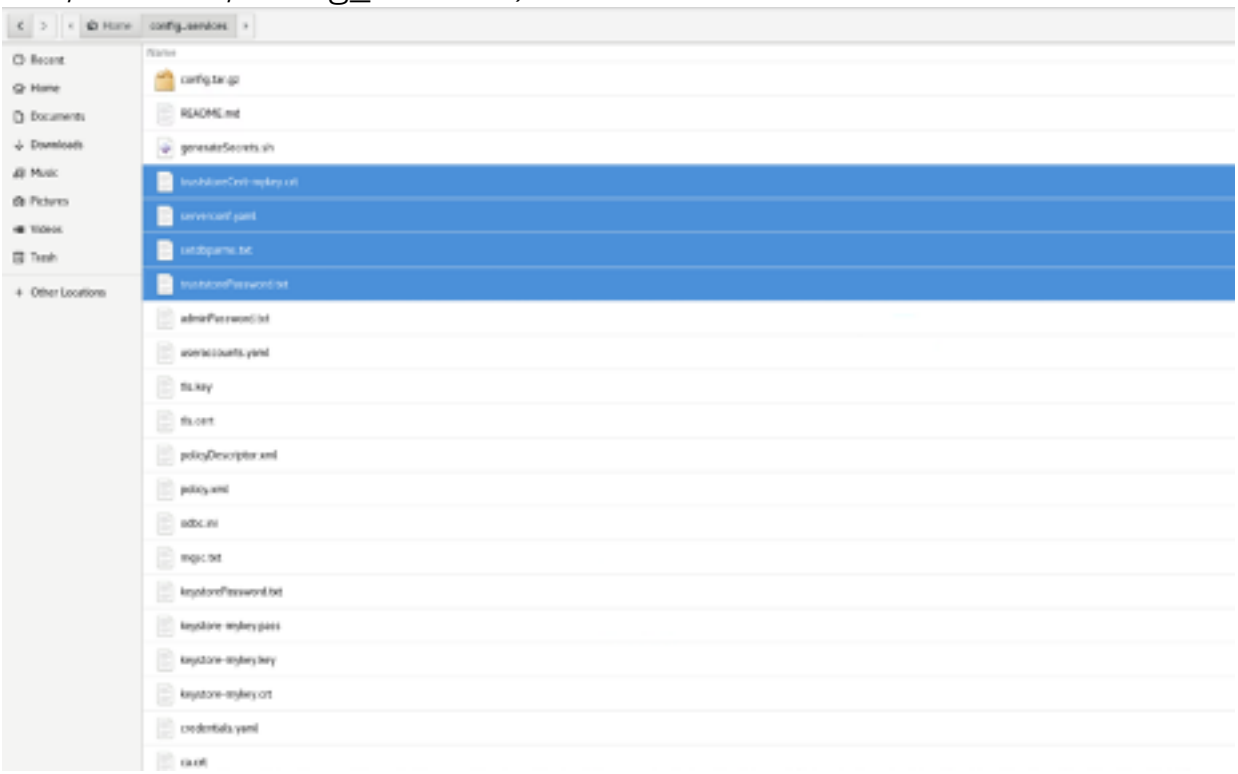
Task 4 - Configuring App Connect Enterprise Services on Cloud Pak for Integration

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**.

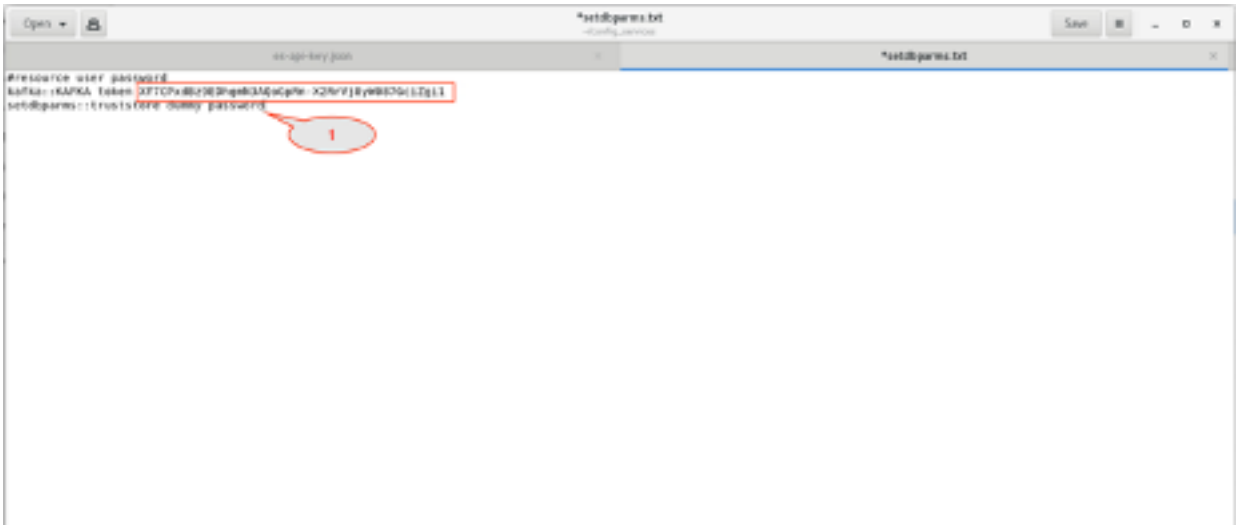


2. Open the folder **/home/ibmuser/config_services**, App Connect Enterprise needs the configuration parameters to access Event Streams. You configure four files: **setdbparms.txt, serverconf.yaml, truststorePassword and truststore-Cert-mykey.crt**. (These files are in /home/ibmuser/config_services)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

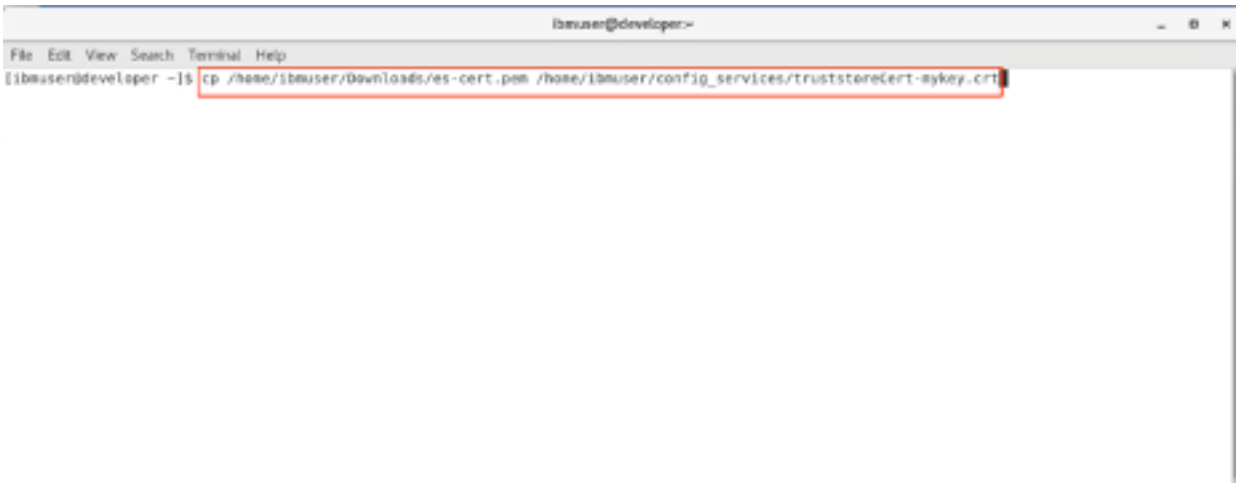


3. You need to edit the Event Streams API Key. Go to the /home/ibmuser/Downloads directory, locate and edit **es-api-key.json.** Copy api key between the quotation marks .

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**.

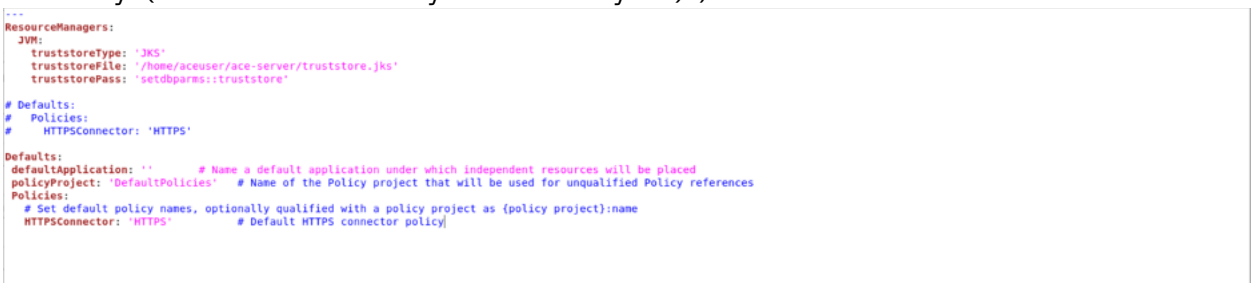


5. Event Streams (Kafka) requires a Certificate. You use the PEM certificate (es-cert.pem) that you downloaded when you configured the Event Streams connection. Open a terminal window and go to /home/ibmuser/Downloads and copy **es-cert.pem** as **truststoreCert-mykey.crt**.



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

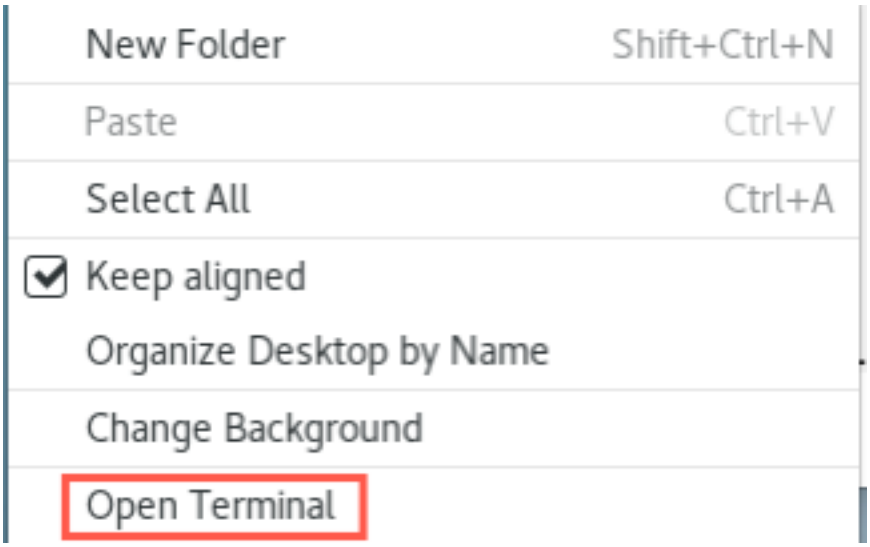
7. Go to /home/ibmuser/config_services, edit **server.conf.yaml,** and if necessary (We have already done for you) , enter these lines.



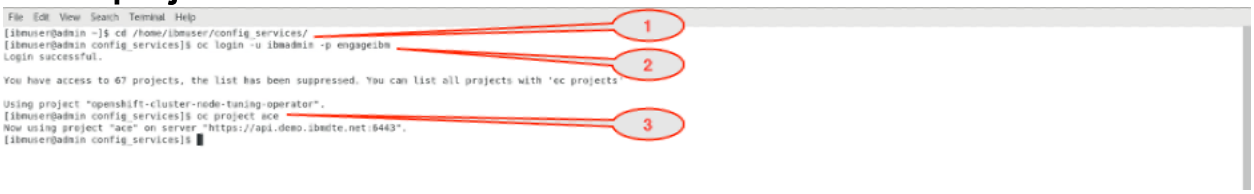
8. Make sure these files are saved and the name is correct.

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.**

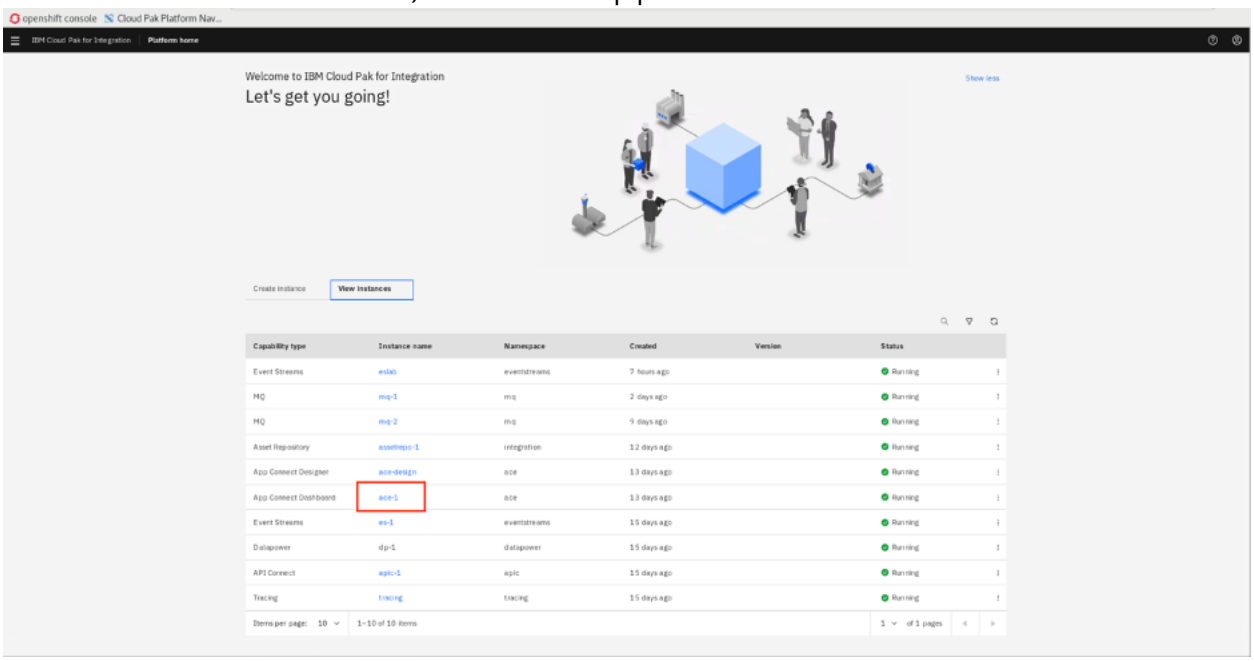


10. Verify you're in the /home/ibmuser/config_services directory, enter **./generateSecrets.sh my-secret**. You have created an App Connect Enterprise configuration.

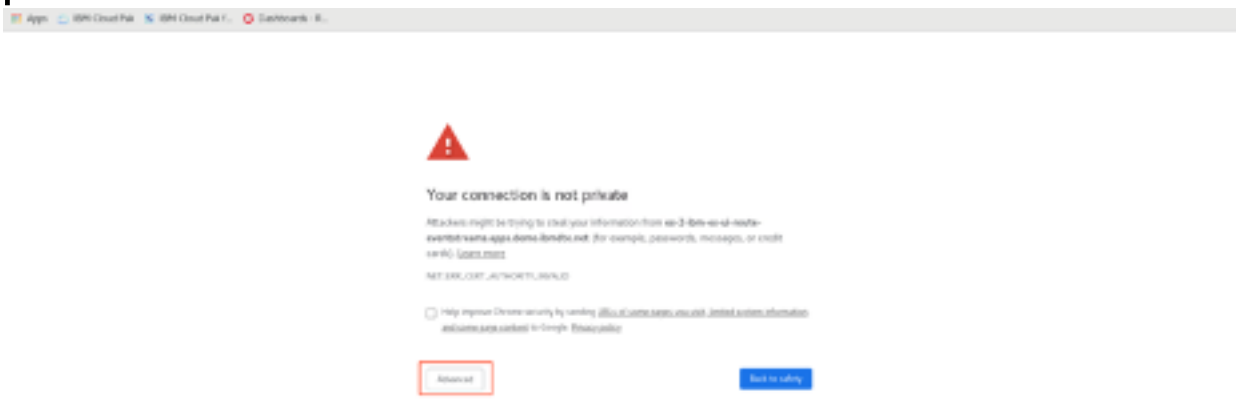
Task 5 - Deploy App Connect BAR file on App Connect Enterprise Server

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

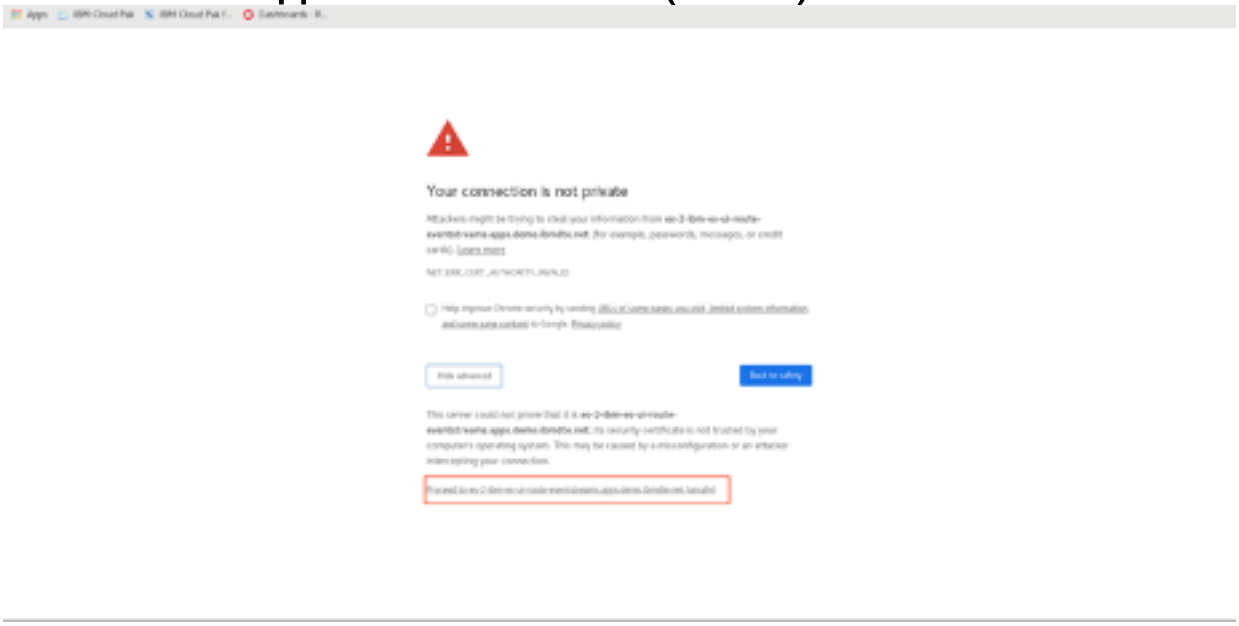
1. Open a new tab and click the **IBM Cloud Pak for Integration** bookmark Bar. Under view instances, click the App Connect link: **ace-1**.



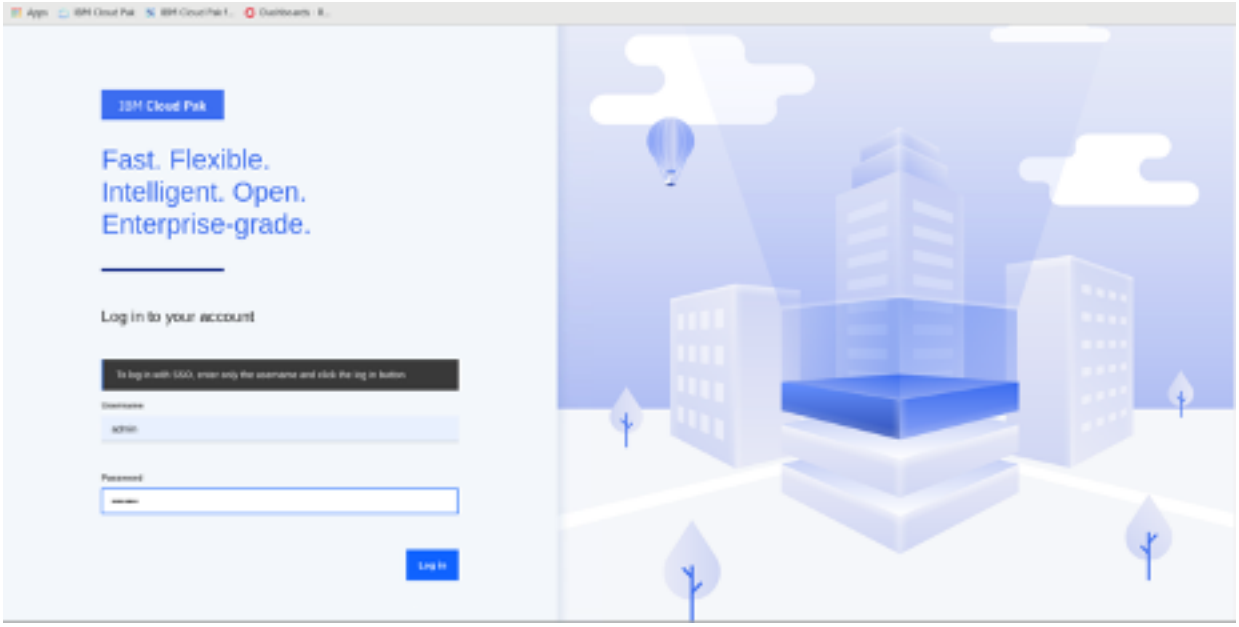
You might receive a page **Your connection is not private.** Click **Advanced.**



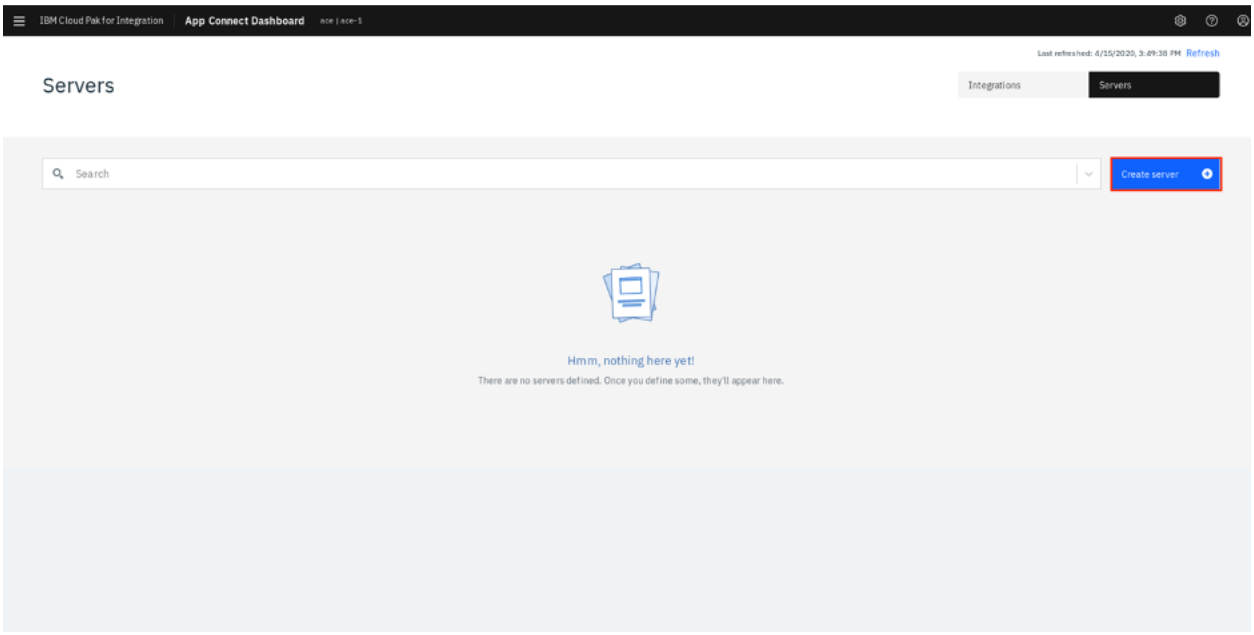
Click the link **Proceed to es-1ibm-es-ui-route-eventstreams.apps.demo.ibmdte.net (unsafe).**



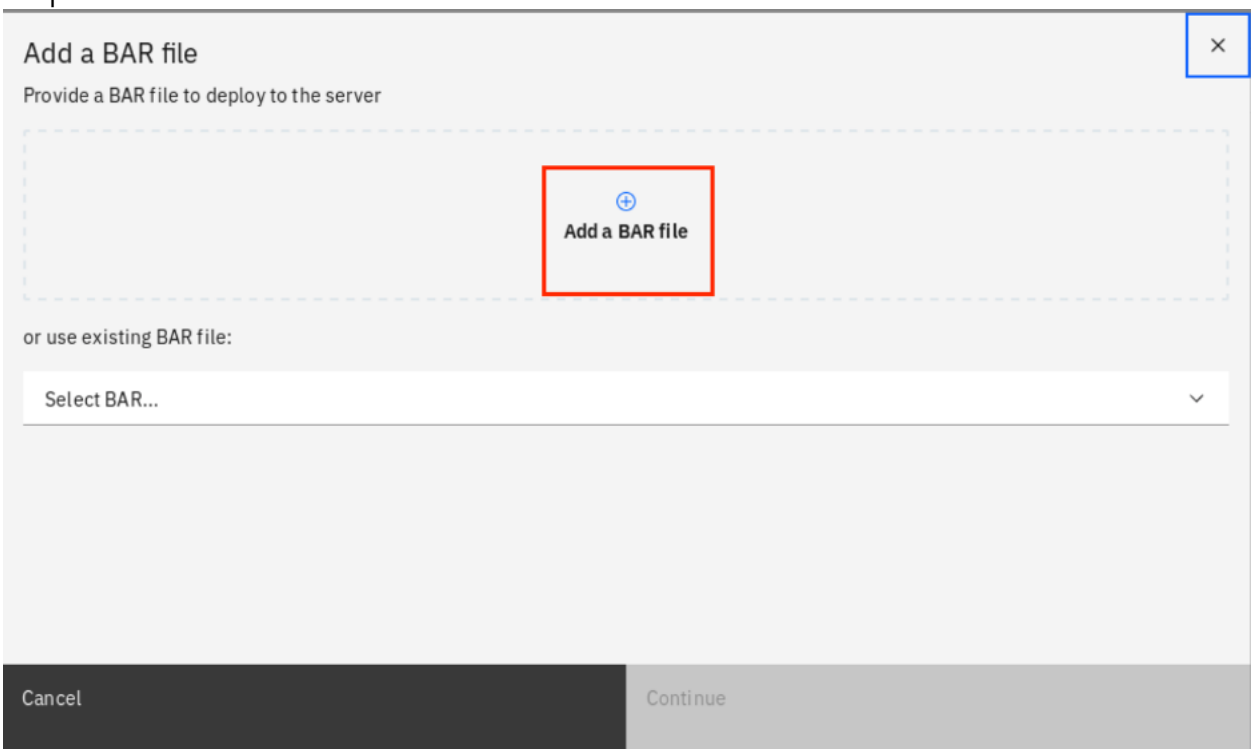
2. If you receive the login page, make the login username: **admin** and Password: **passw0rd**.



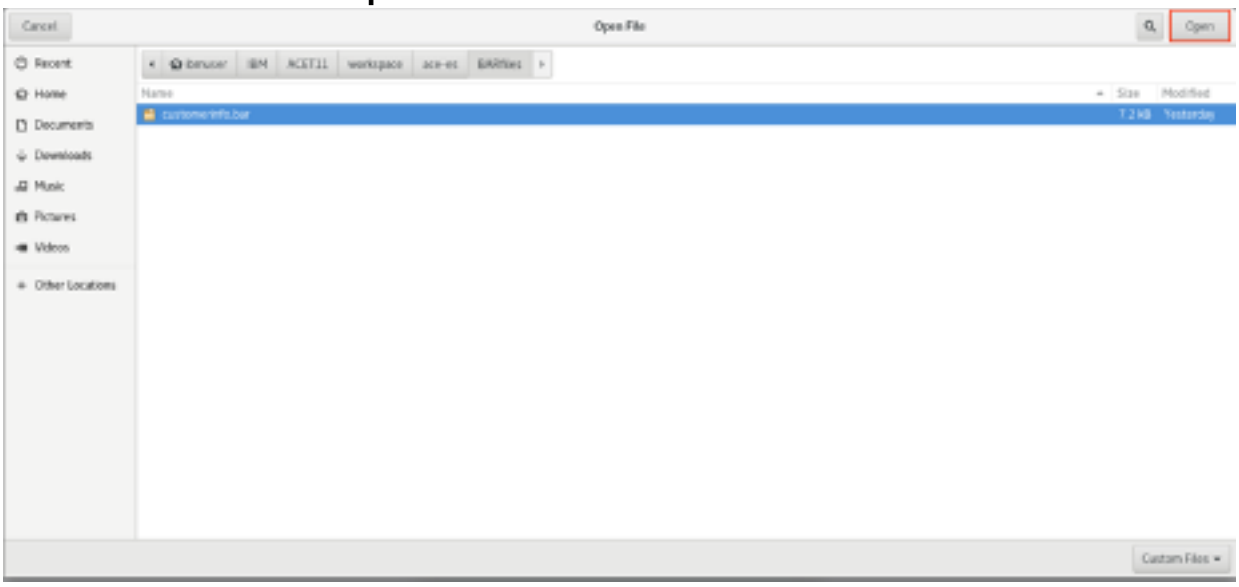
3. You deploy the BAR file that you saved in the App Connect Enterprise toolkit. This the App Connect Enterpriser Dashboard. All deployed Integration servers will be here. Click **Create Server.**



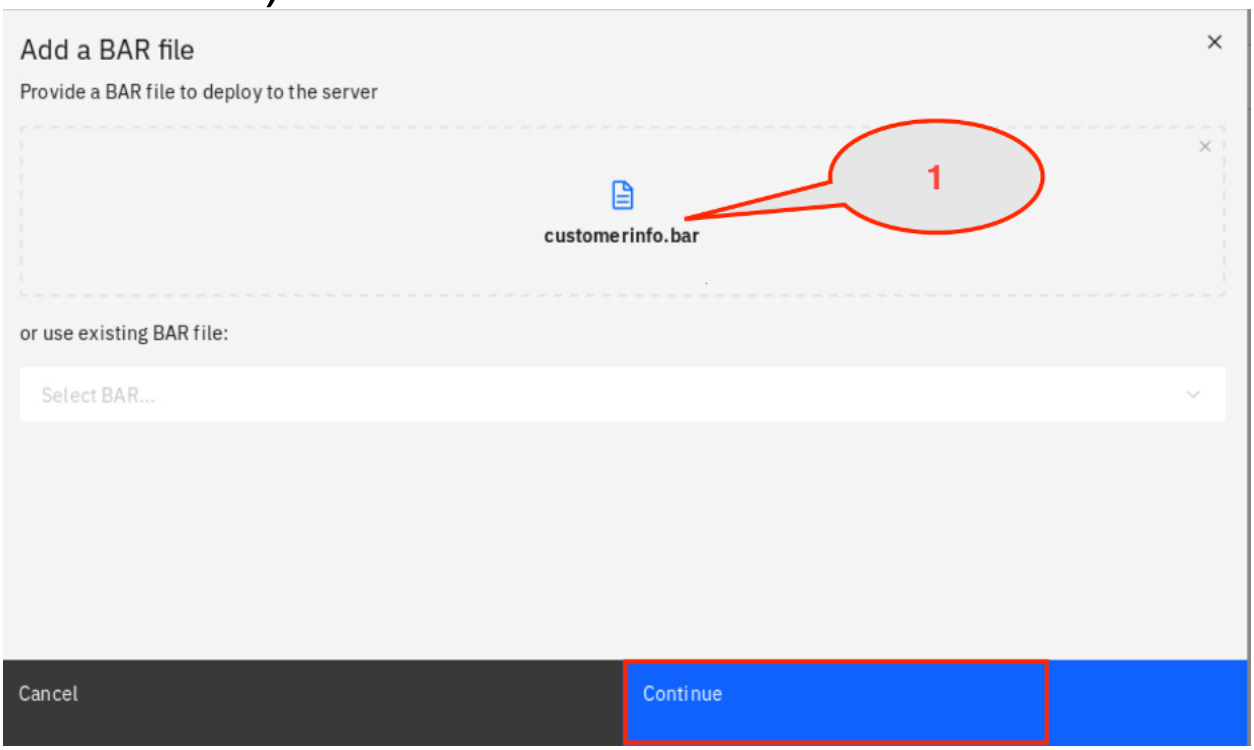
4. On the pop-up, **Add a Bar file**, add the BAR file you saved in App Connect Enterprise toolkit. click **Add a BAR file.**



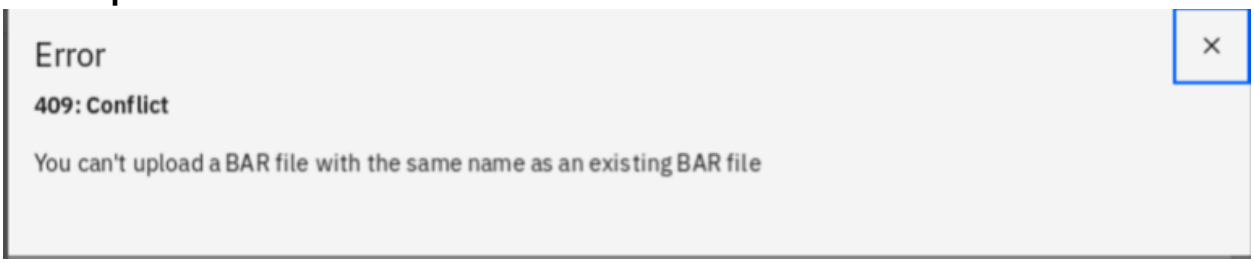
5. The deployment process opens **Open File** and locate **customerinfo.bar** in **/home/ibmuser/IBM/workspace/ace-es/BARfiles** and click **Open.**



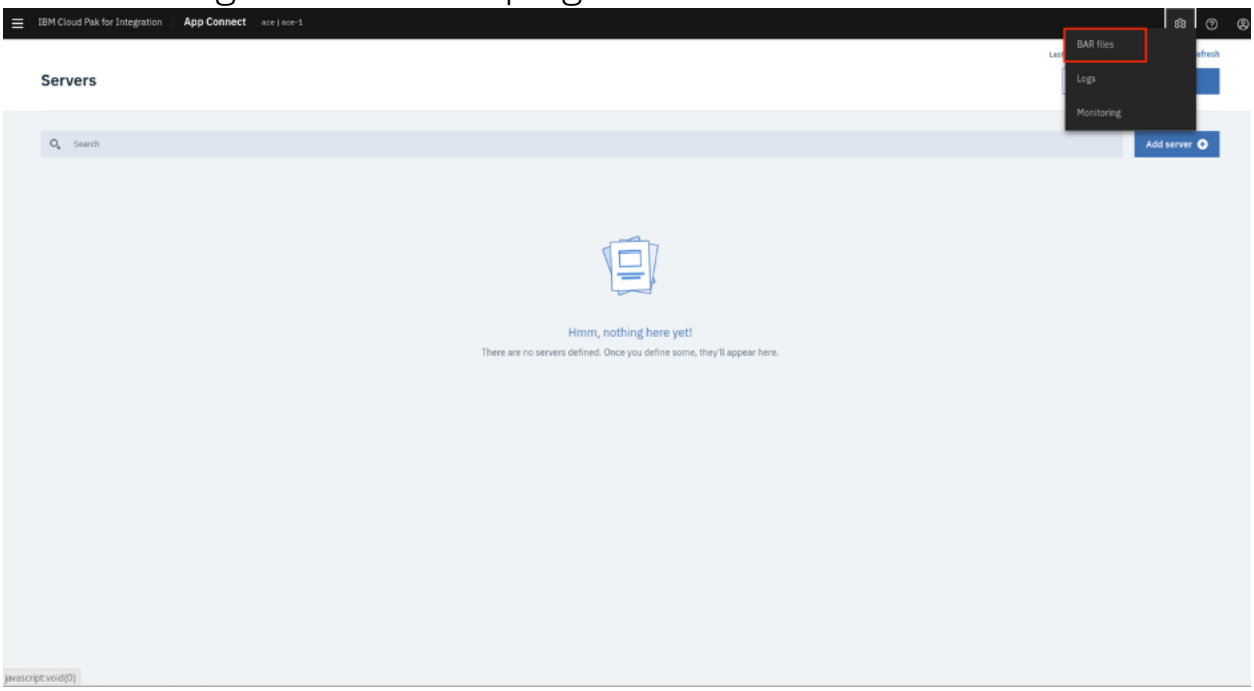
6. App Connect Enterprise shows the BAR file that you selected (**customerinfo.bar)**. Click **Continue.**



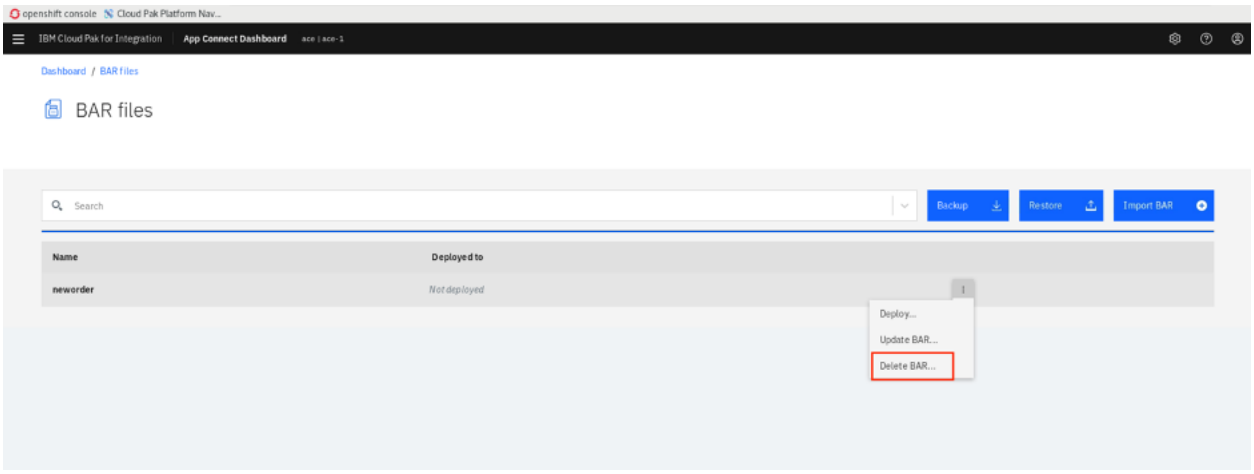
7. You might receive an Error message (**Conflict**). Close the Window. If not go to **Step 12 .**



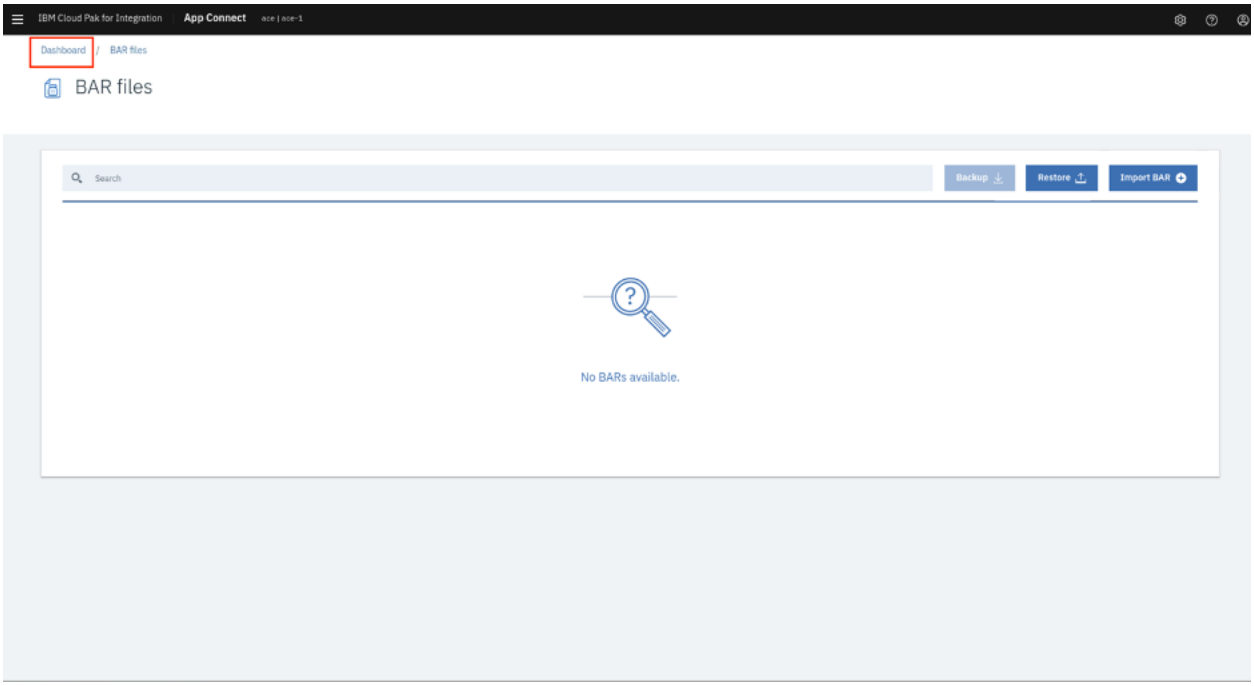
8. Click the settings icon on the top right and select **BAR files.**



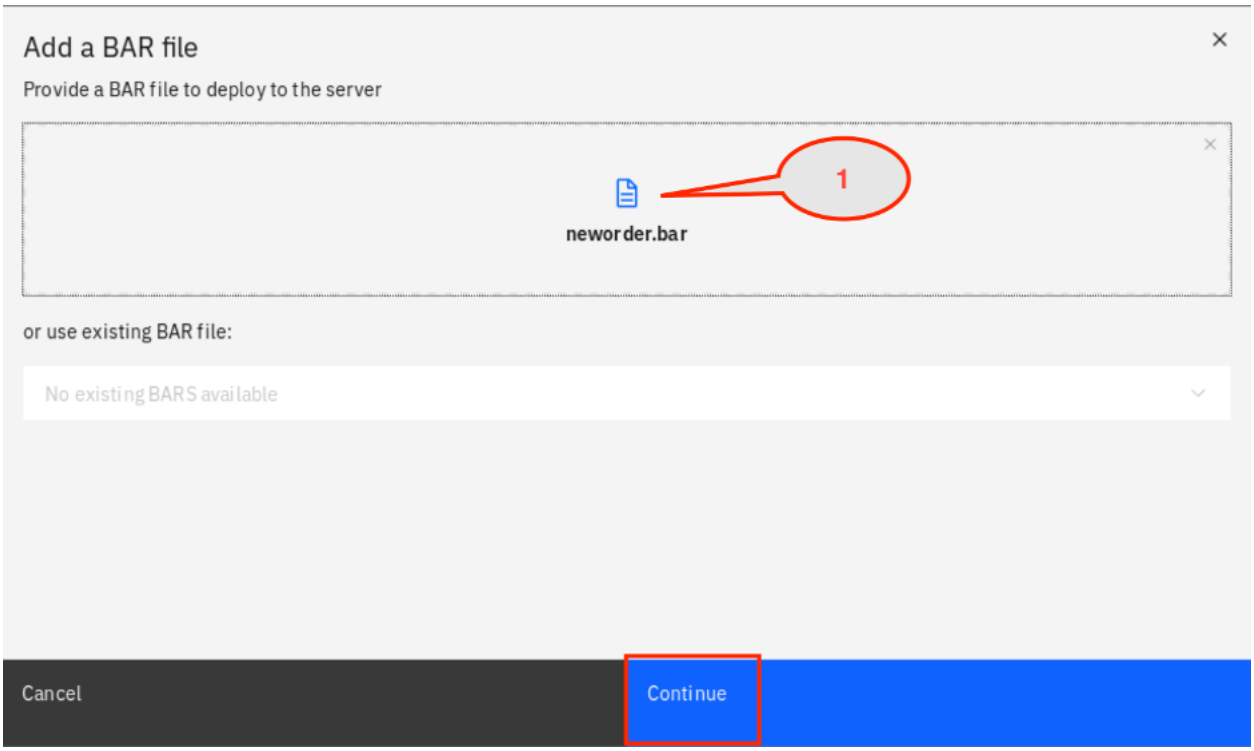
9. You see the **Not Deployed** BAR file, click (**...**) and select **Delete BAR.**)You can deploy, delete or update the Bar file. For this lab just Delete).



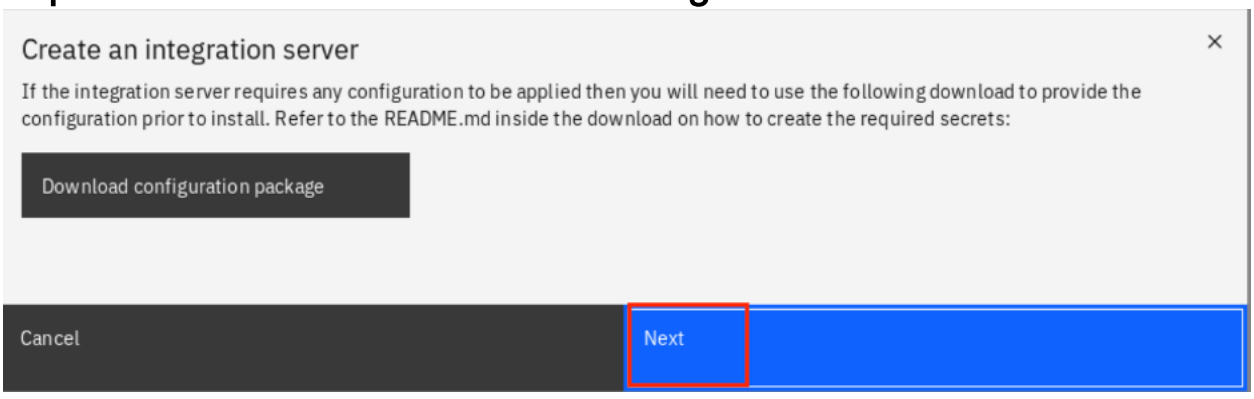
10. Click **Dashboard** link on the top of the page. To deploy the BAR file.



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.**

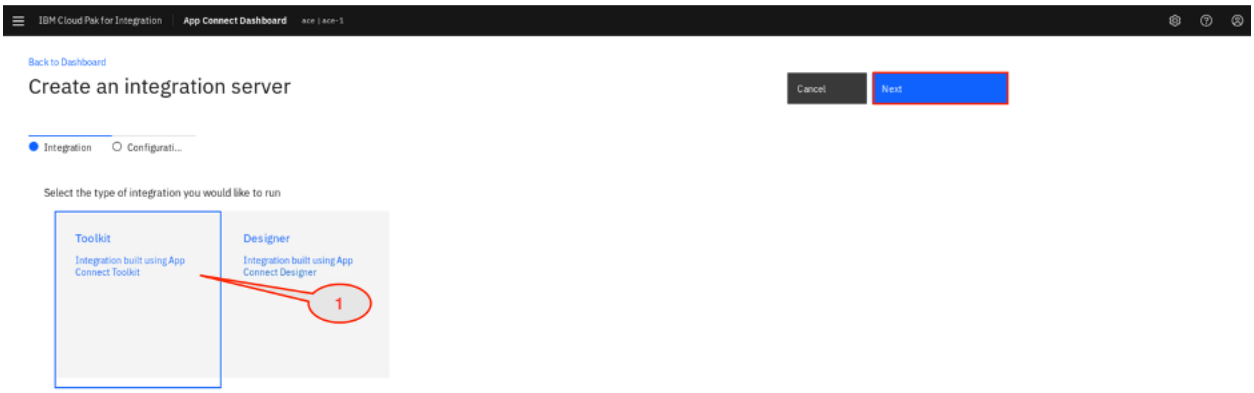


12. You don't need to download configuration package., you already have configured the App Connect Enterprise server (when you run **generate-secret script**). Click **Next**. (You might notice an icon **Download configuration package**. You can skip this because you have already set this setting in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



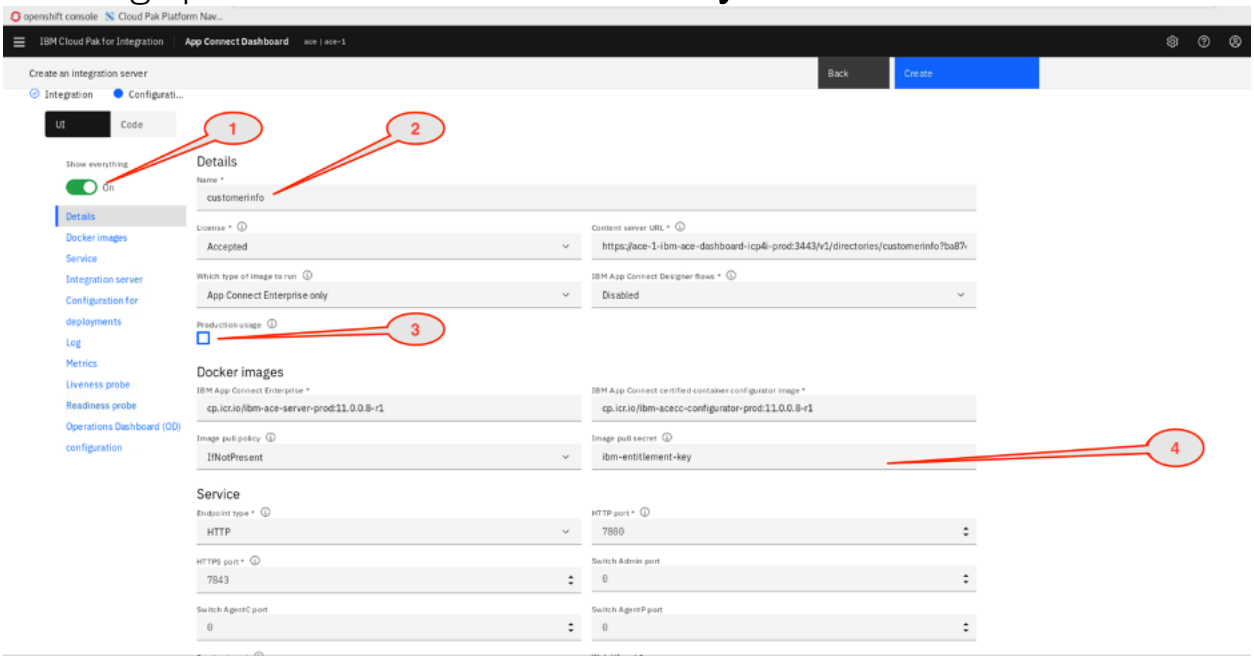
13. In the **Create an integration server** page. You have two option to deploy a BAR file. Deploy a BAR file from **App Connect Toolkit** or a BAR

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:

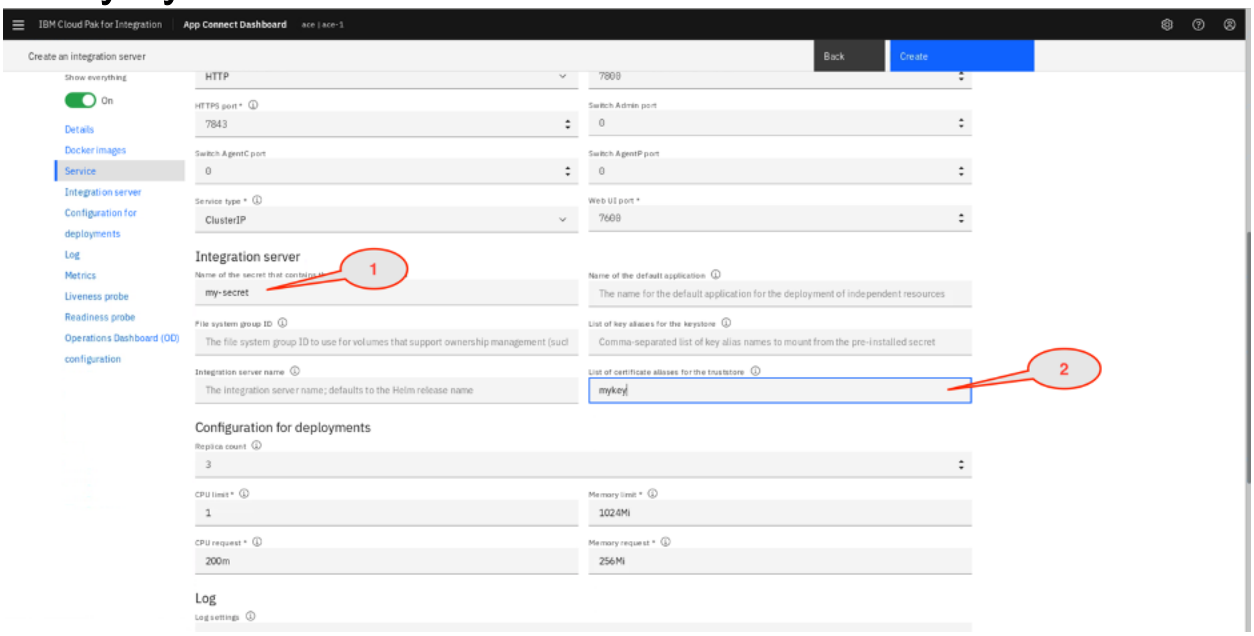
1. Change **On** to **Show Everything**
2. Enter the deployment name: **customerinfo**
3. Uncheck **Production usage** box.
4. Enter Image pull secret: **ibm-entitlement-key .**



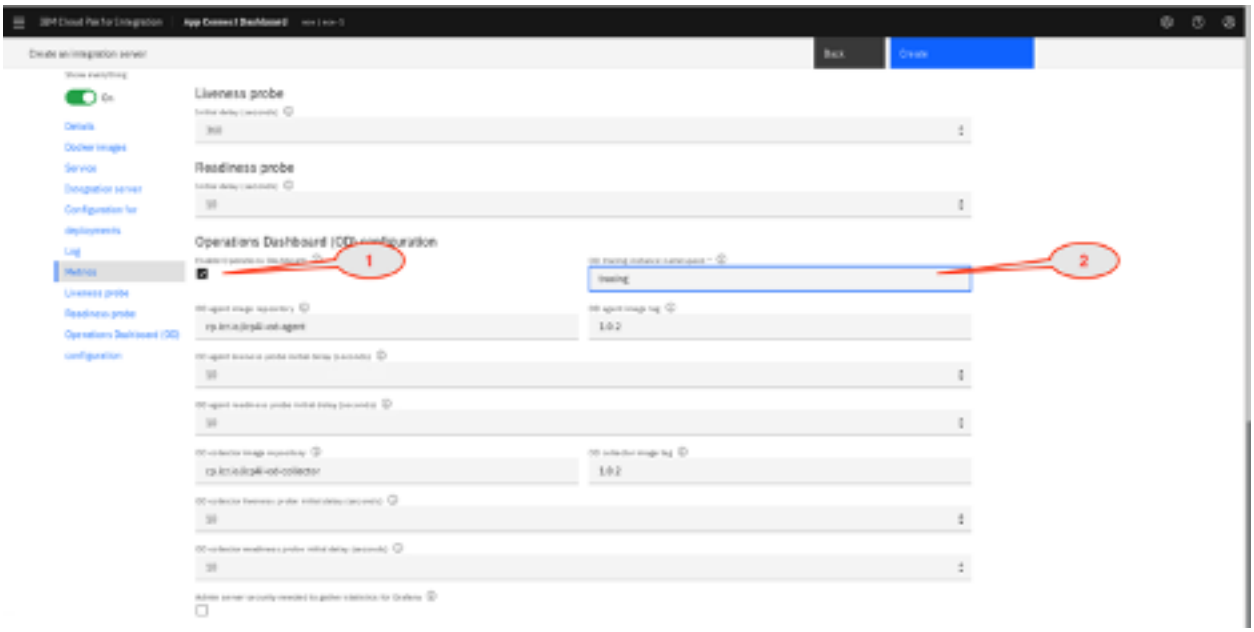
15. Scroll and locate Integration Server parameters:

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

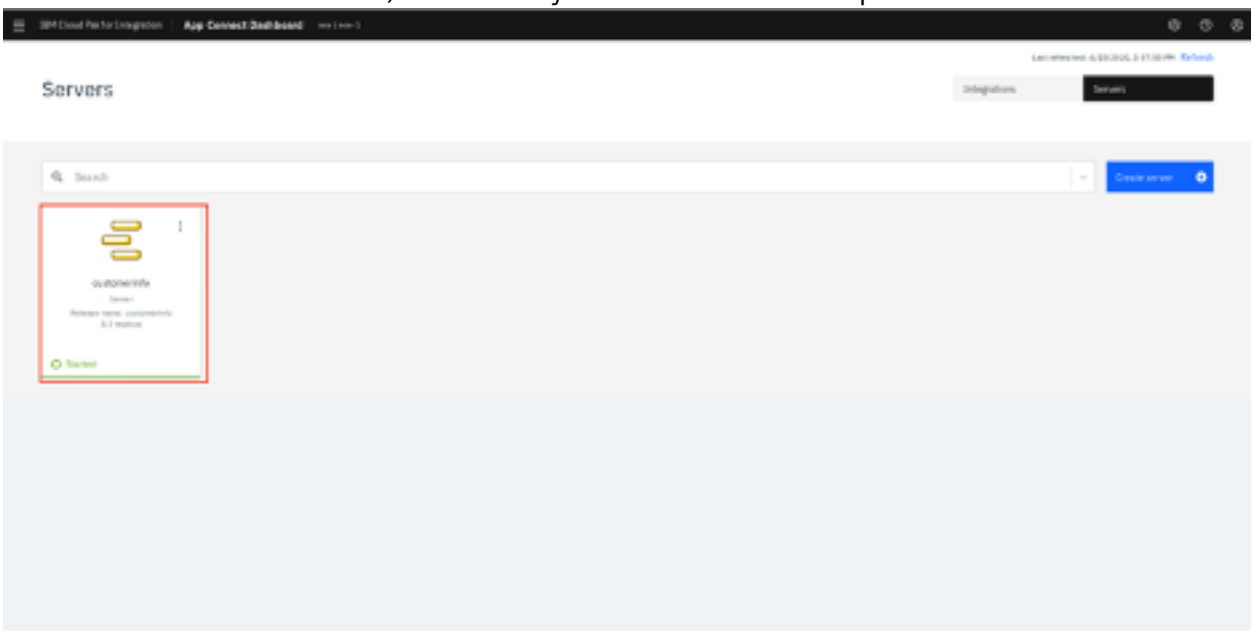
2. Enter **mykey** as the **List of certificate aliases for the the truststore**.



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.**



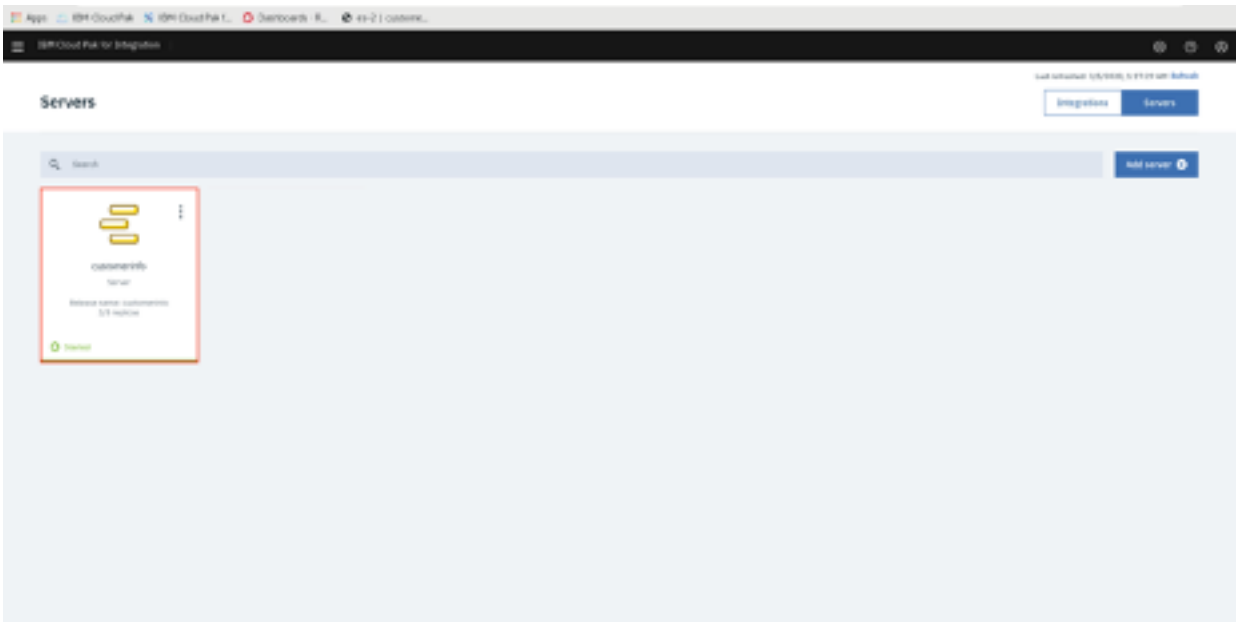
17. After the BAR deployment, the process guides you back to App Connect Dashboard. The deployed application will be on **Started** status, if the status is **Unavailable**, **refresh** your browser to update.



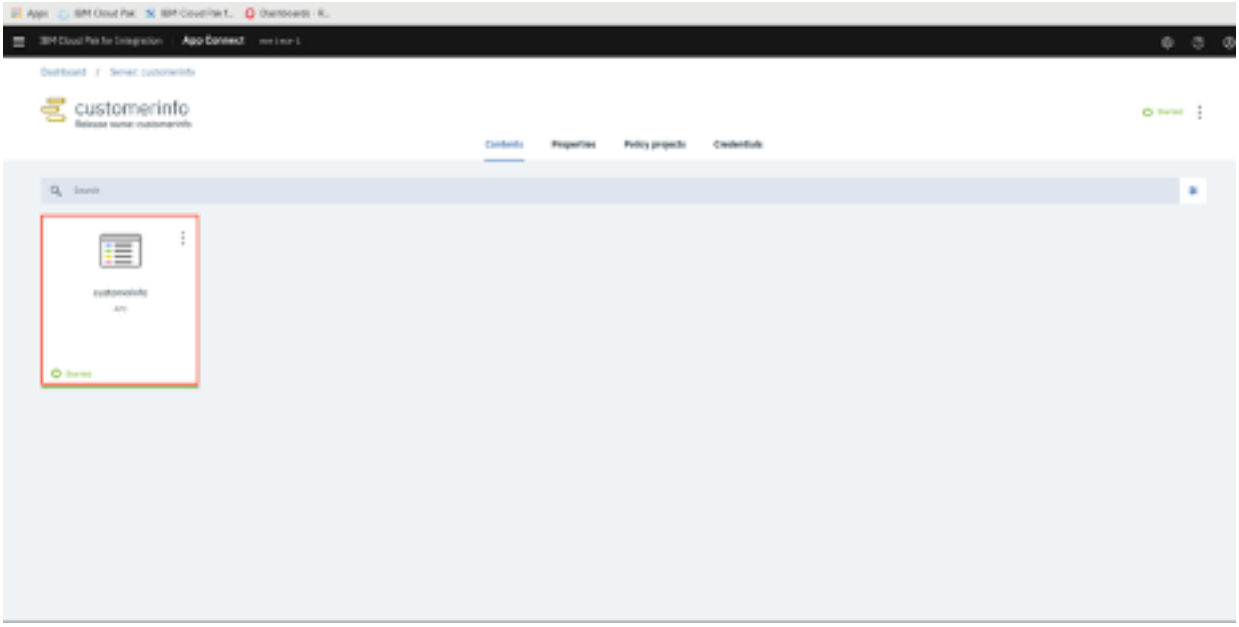
Task 6 - Testing App Connect Enterprise API sending a message to Event Streams

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

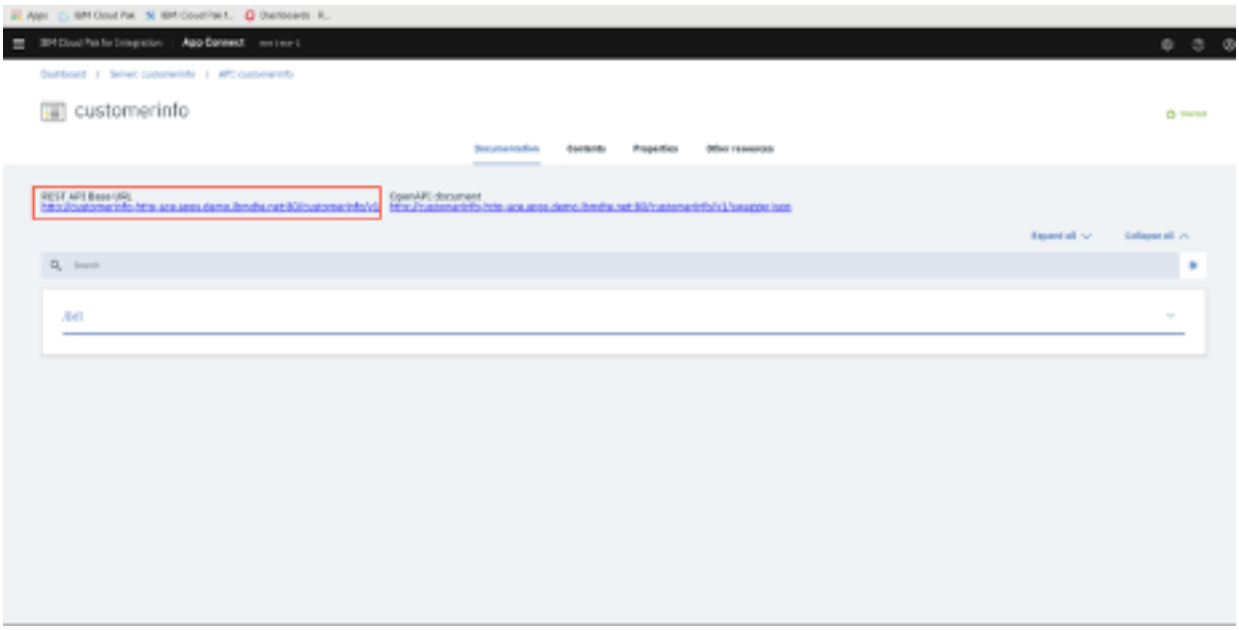
1. You see App Connect Enterprise Server running.
   Click the **customerinfo** server icon.



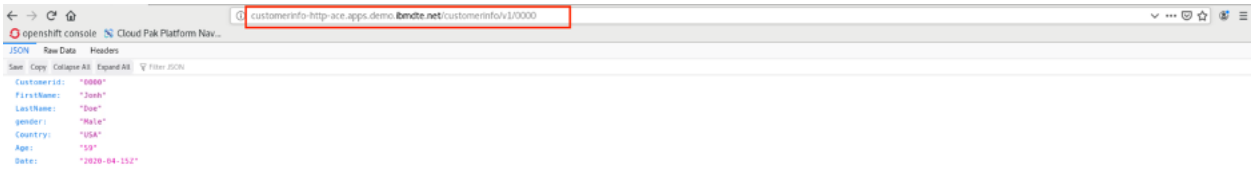
2. Next, click **customerinfo API** and then **Open.**



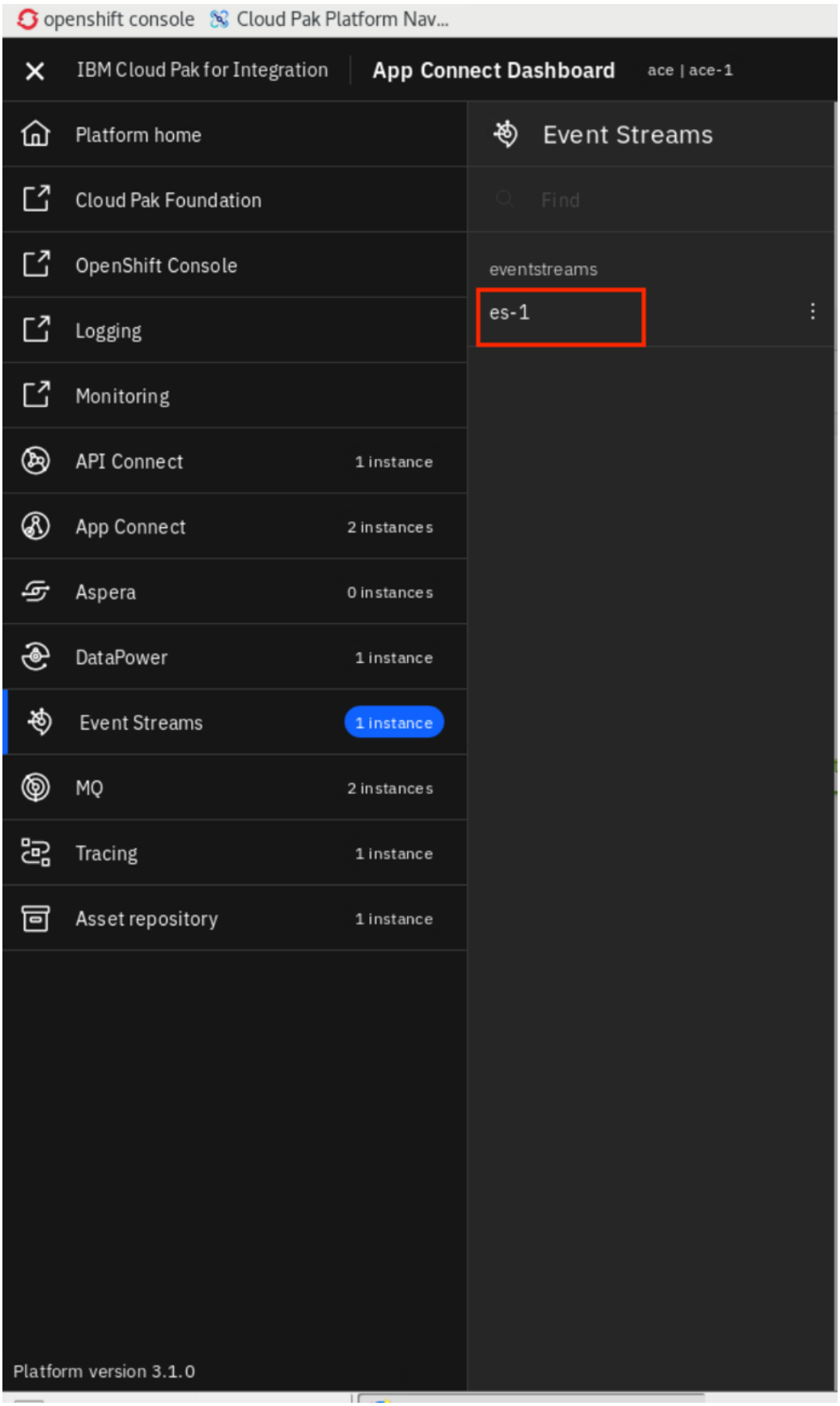
3. App Connect Enterprise created a **REST API base URL** and **OpenAPI document**.



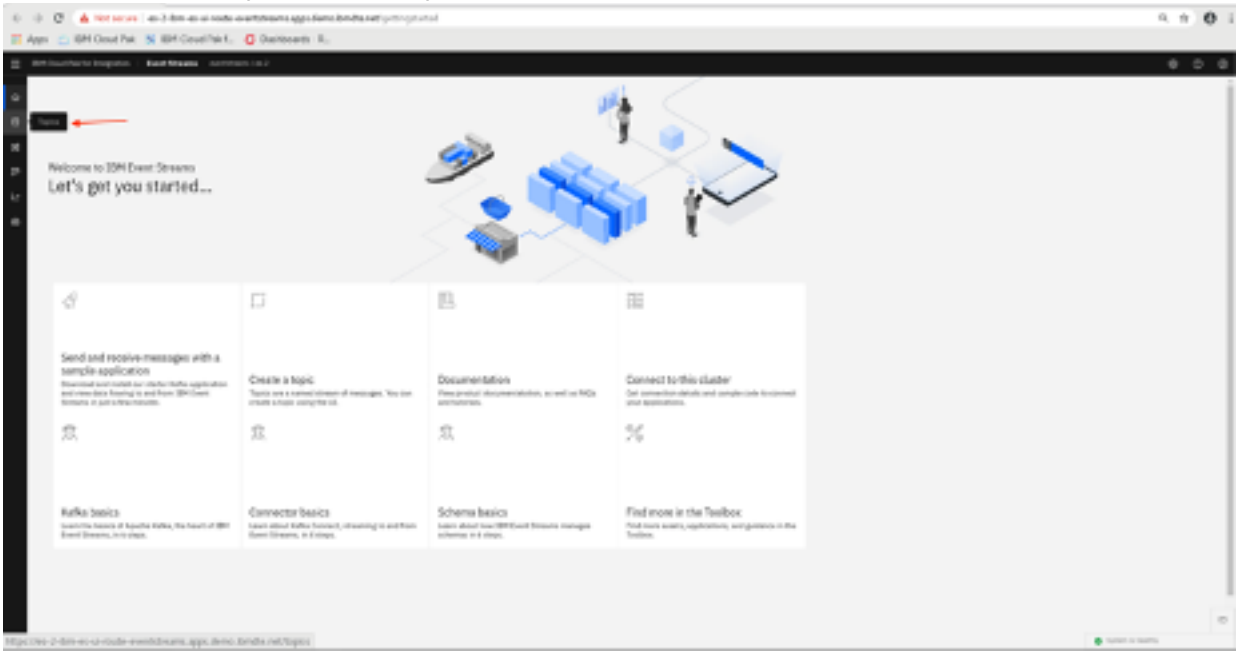
4. Click the RESTAPI base URL. You receive a 404 error. Complete the Rest API on the URL browser: **customerinfo-http-ace.apps.demo.ibmdte.net/customerinfo/v1/0000** and see the results. Note the customer information and Customerid: **0000**.



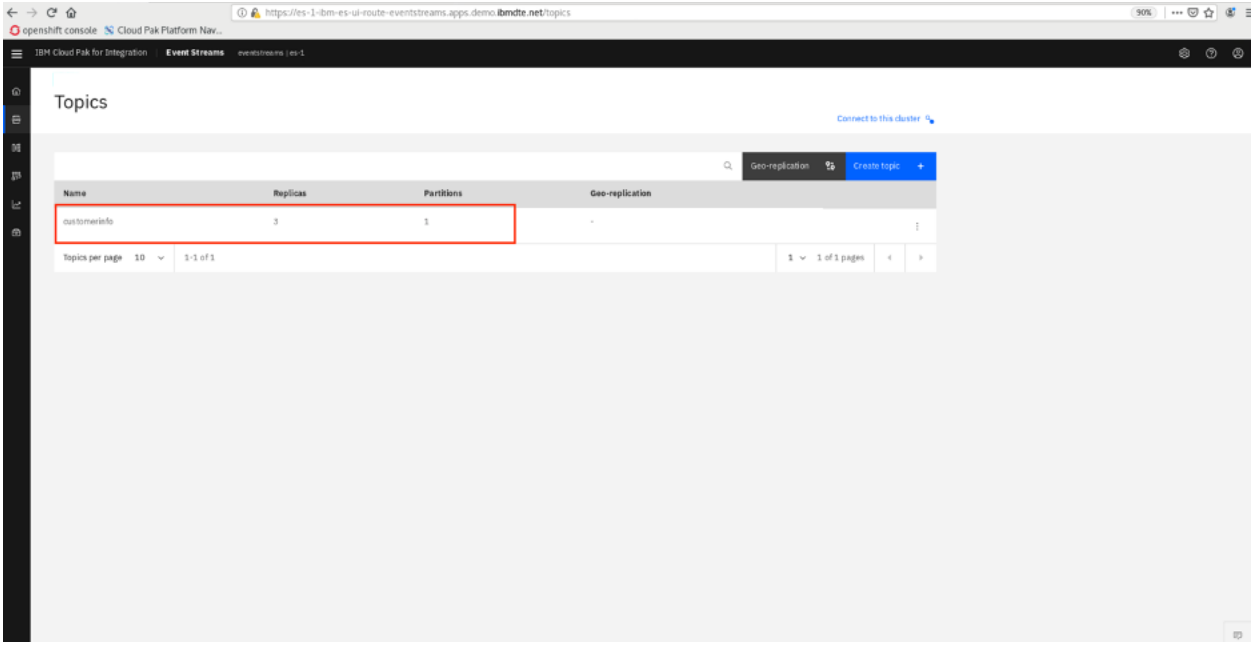
5. A message was sent from App Connect Enterprise to IBM Event Streams. Go to I**BM Pak Cloud for Integration**. **. Y**ou can use **Cloud Pak Menu, click the** Hamburger menu on the top and Click **Event Streams application -> es-1 instance .**


openshift console
Cloud Pak Platform Nav...
IBM Cloud Pak for Integration
App Connect Dashboard
ace | ace-1
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
Platform version 3.1.0
Event Streams
Find
eventstreams
es-1

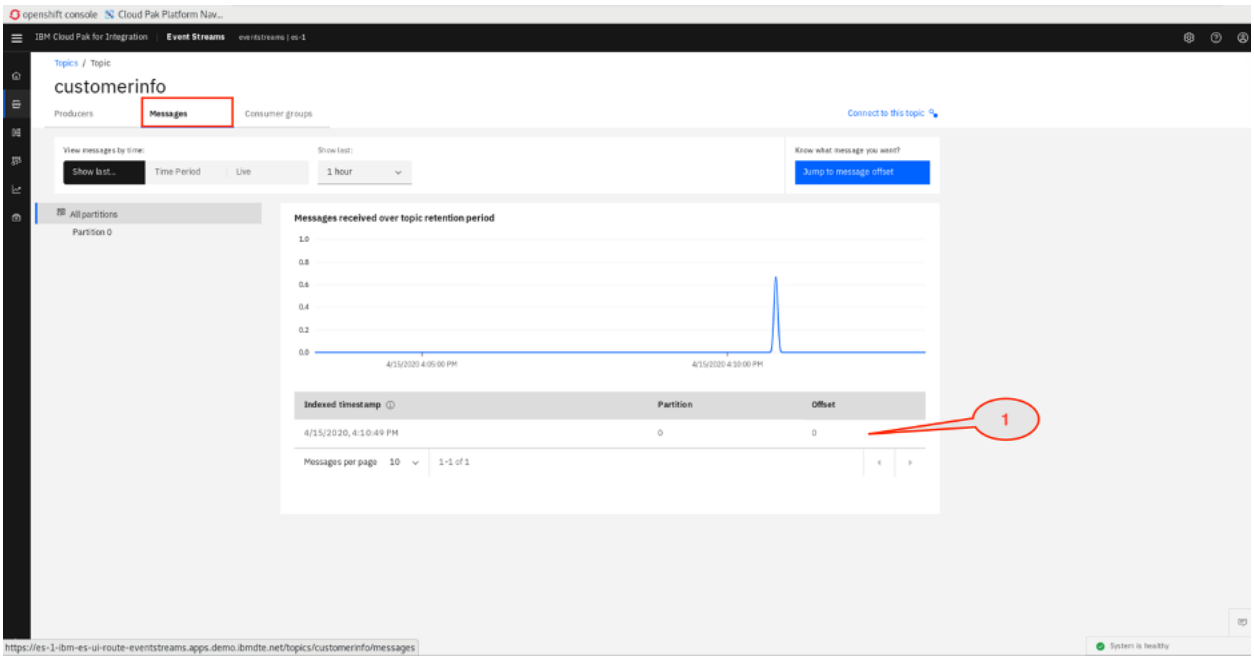
6. Click **Topics** to open the topics list of this Event Streams instance.



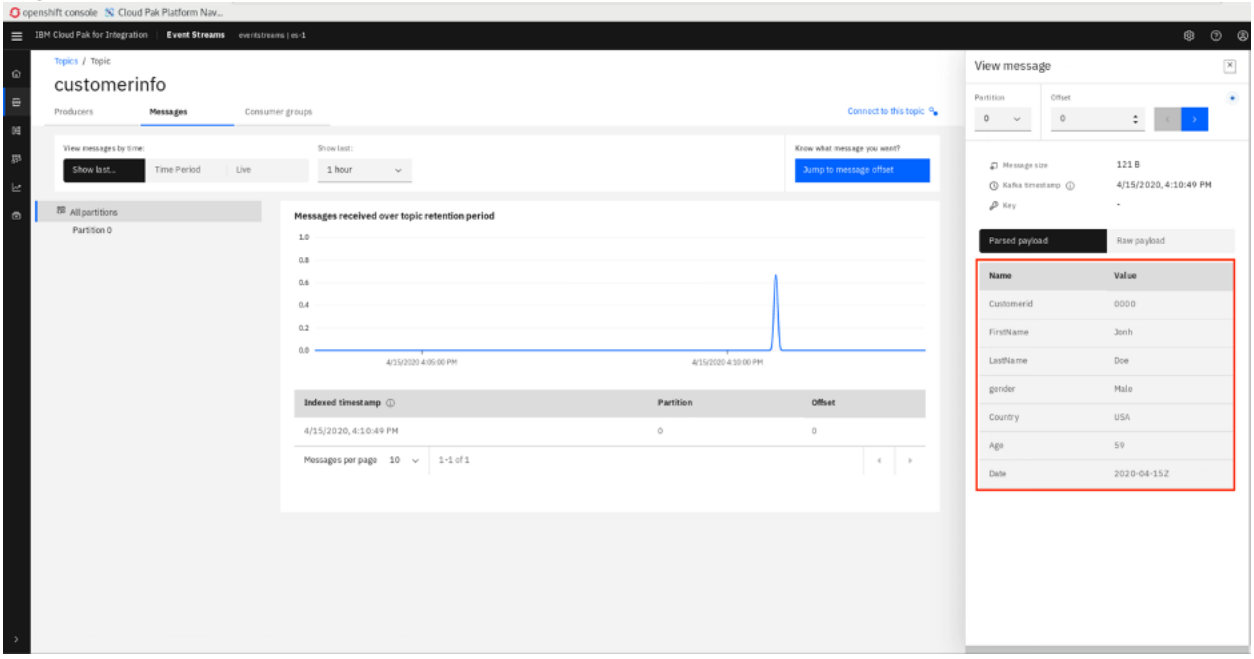
7. In the Topics page, click the topic **customerinfo.**to open the topic page.



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)

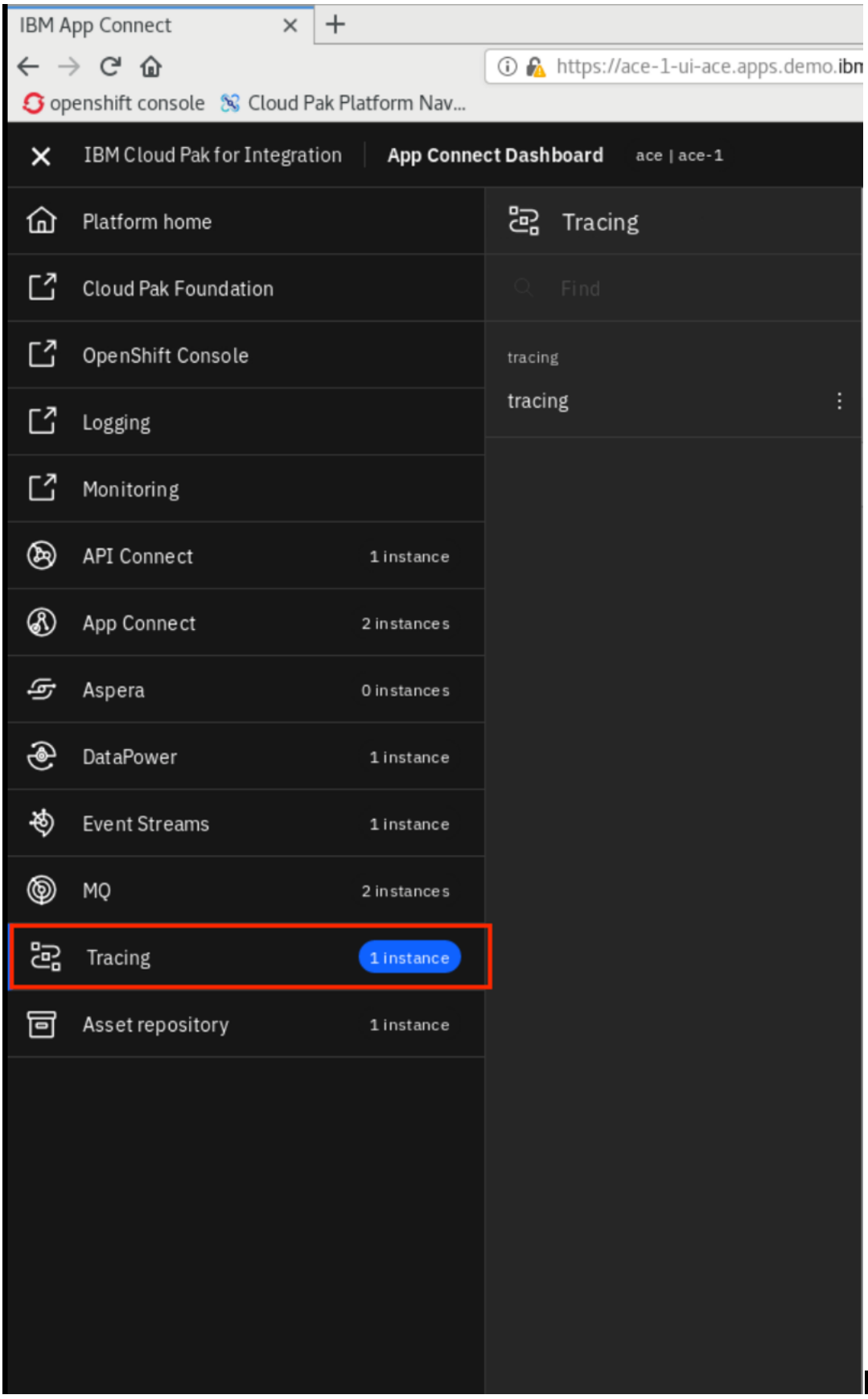
The Operations Dashboard collects data from all the registered capabilities (such as MQ) in real time. By default, and for this lab, 10 percent of traffic is sampled.

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.

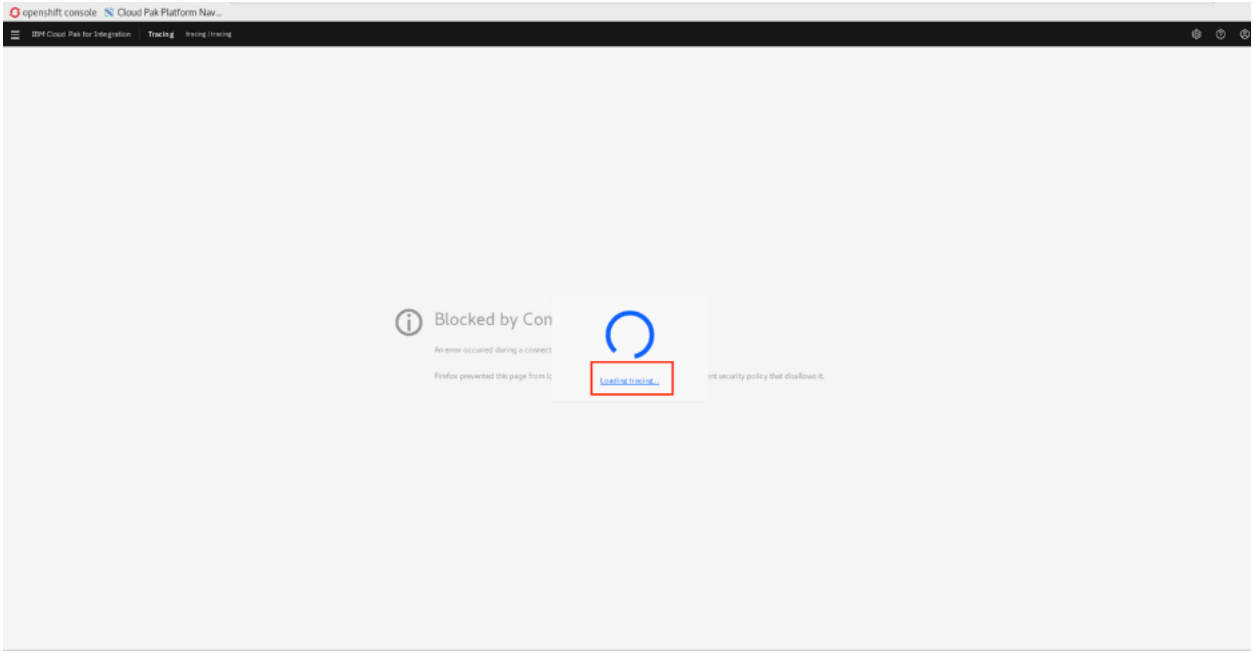
- **Trace:** The description of a transaction as it moves through IBM Cloud Pak for Integration platform.
- **Span:** A named, timed operation representing a piece of the workflow (e.g. calling an API, invoking a message flow or placing a message in a queue or a topic).
- **Span context:** Trace information 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 Hambuger Menu and select **tracing application -> tracing** instance to open the Operations Dashboard instance.

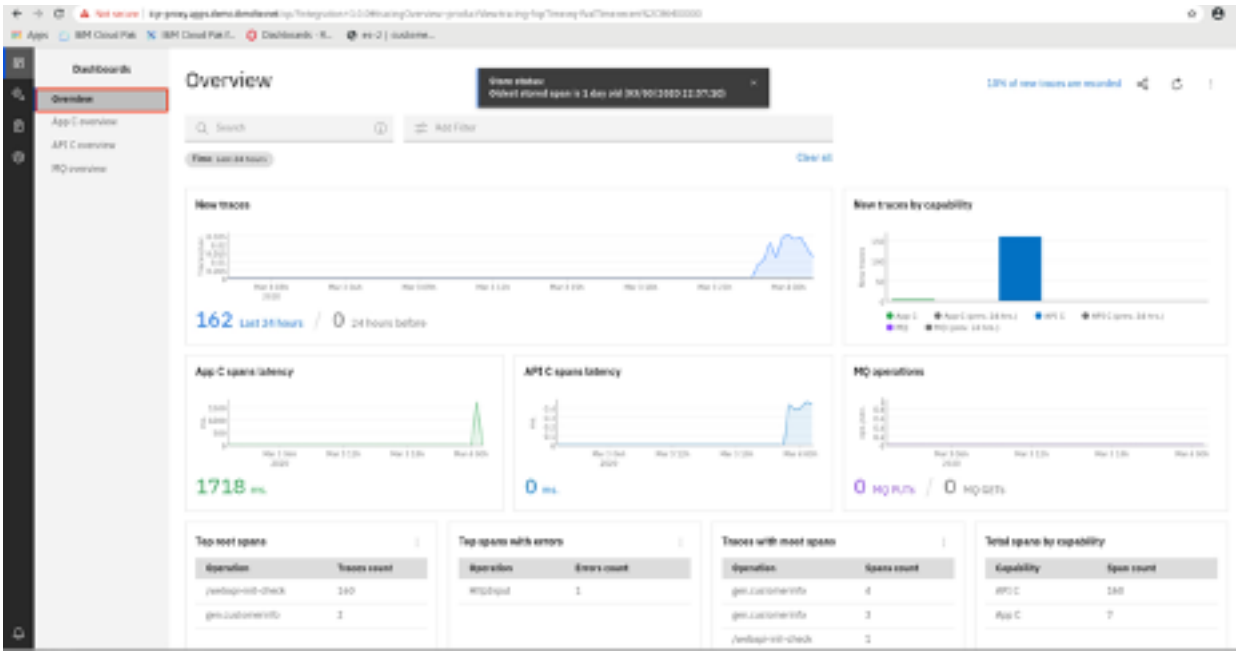

IBM App Connect
https://ace-1-ui-ace.apps.demo.ibn
openshift console
Cloud Pak Platform Nav...
IBM Cloud Pak for Integration
App Connect Dashboard
ace | ace-1
Platform home
Cloud Pak Foundation
OpenShift Console
Logging
Monitoring
API Connect
1 instance
App Connect
2 instances
Aspera
0 instances
DataPower
1 instance
Event Streams
1 instance
MQ
2 instances
Tracing
1 instance
Asset repository
1 instance
Tracing
Find
tracing
tracing

Note:

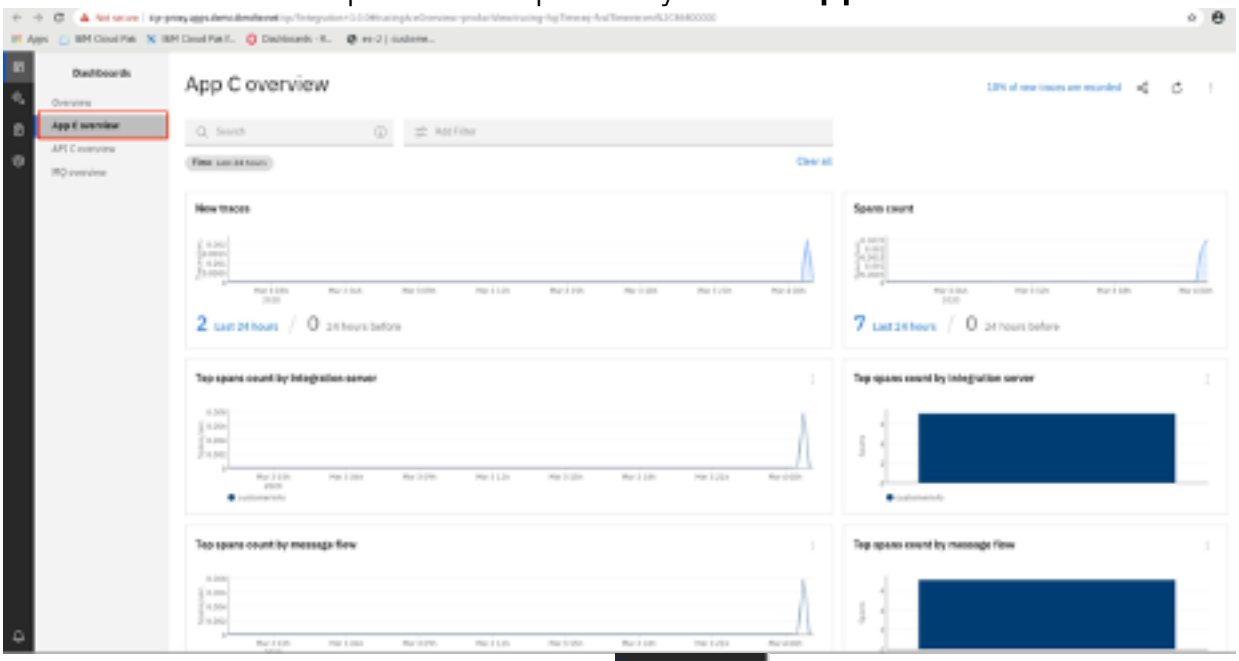
might receive this page. Click **loading tracing.**

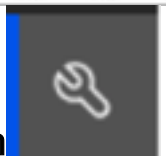


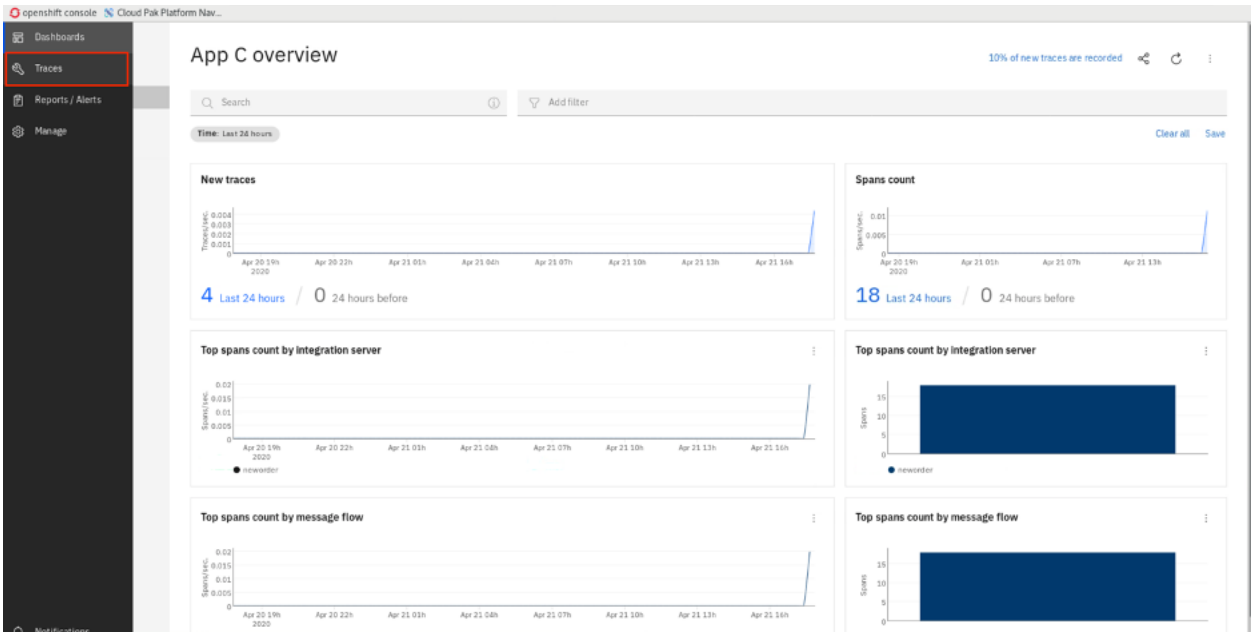
2. In the Tracing page, check the Overview page. You see all the products that you can use this tool: APIC (including DataPower), APP Connect and MQ. You see all the tracing of **MQ, App Connect and APIC** (You see how to configure tracing in APIC lab). Operations Dashboard Add-on is based on Jaeger 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.



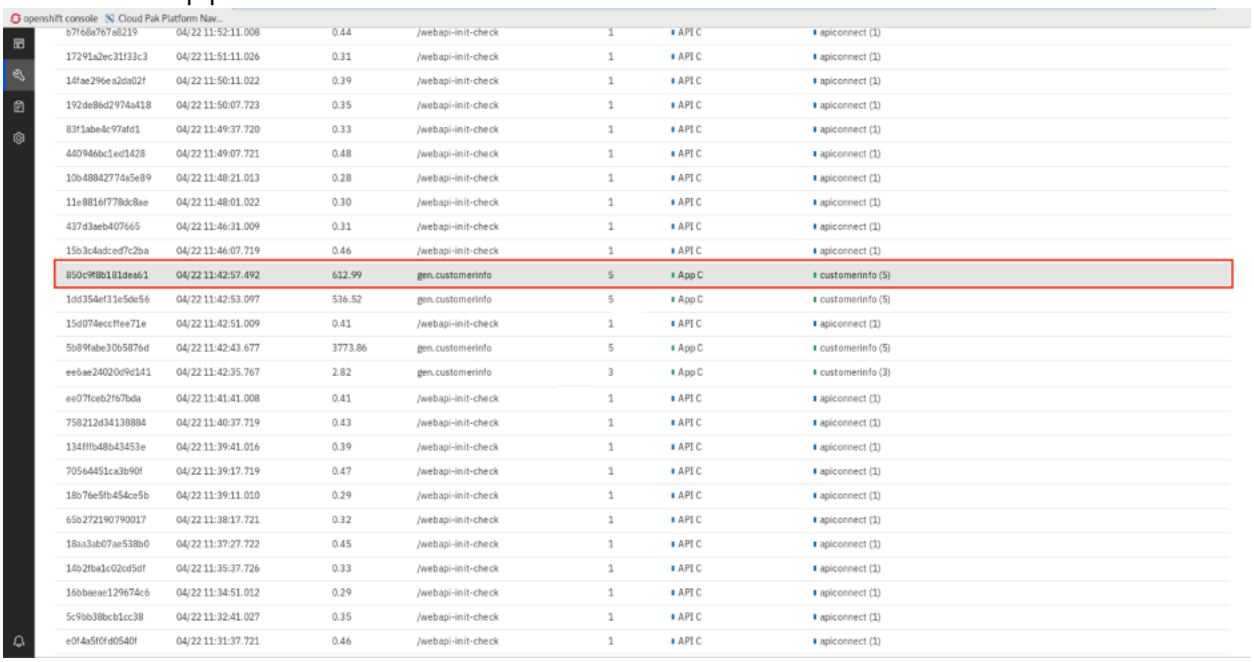
3. You can monitor each product separately. Click **App C overview**.



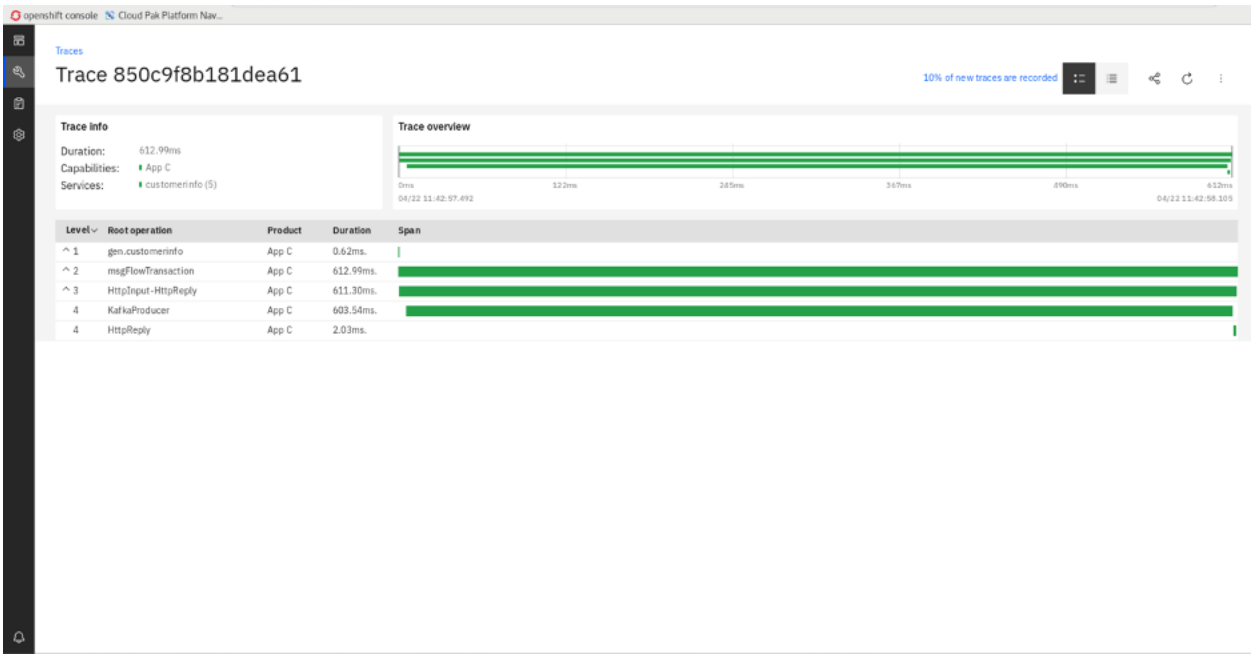
4. In the tracing page, select **traces icon**  the menu on the left.



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



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.