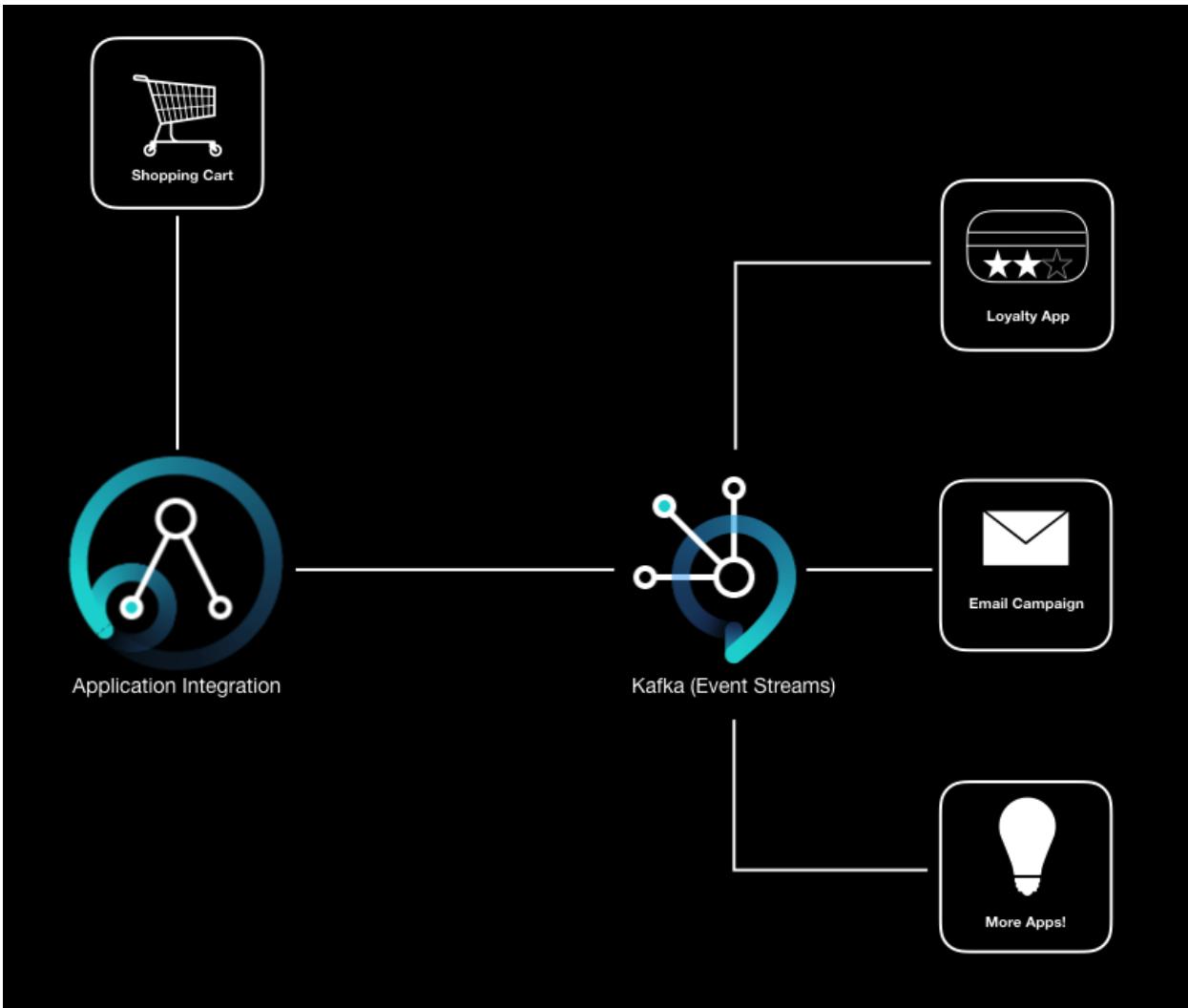


Integrate Kafka with business applications to create new responsive experiences

Integrate Kafka with business applications to create new responsive experiences

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'll be taking 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 that when a customer places an order, we want to provide a real-time response -- reward the customer with points in customer loyalty app or a gamification experience or sign them up for a certain email nurture program. To do that, we need each order to emit an event. The Cloud Pak for Integration combines integration capabilities with Kafka based event streaming to make the data available for cloud-native applications to subscribe to and use for a variety of business purposes.



In this tutorial, you will create a topic in Kafka, modify an integration flow to call an API, and emit an event onto the topic.

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

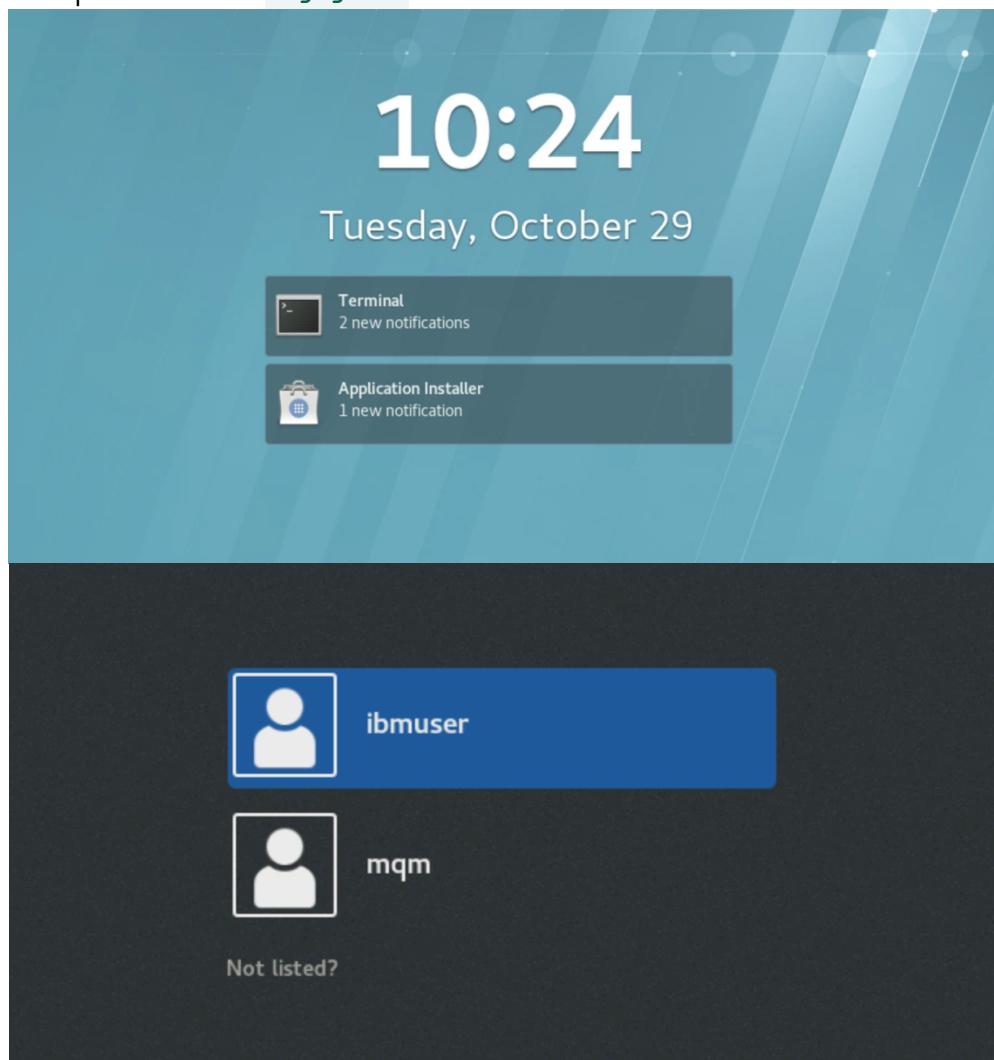
- Start IBM Cloud Pak for Integration Environment
- Creating and Configuring an Event Streams Topic
- Configure App Connect Enterprise flow using App Connect Enterprise
- Configuring App Connect Enterprise on Cloud Pak for Integration
- Deploy App Connect BAR file on App Connect Enterprise Server
- Testing ACE API sending a message to Event Streams

Task 1 - Start IBM Cloud Pak for Integration Environment

As this is a new deployment of the IBM Cloud Pak for Integration, you must execute some steps to prepare the environment. Initial setup steps are only needed for a fresh installation of the platform. They do not need to be repeated.

Start the Environment

1. Log into the Linux desktop with the following steps and credentials
 - a. Click **ENTER** to view the Linux desktop.
 - b. userid: **ibmuser**
 - c. password: **engageibm**



Confirm the Cloud Pak for Integration is up and running

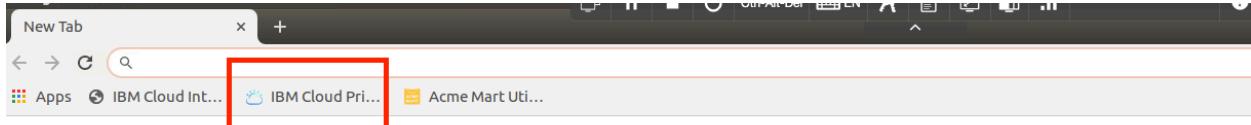
1. The next step is to check if the environment is done loading. Open a terminal console window by clicking the Terminal icon in the Desktop.
 - a. Type `./startup-scripts/oc-startup.sh`
 - b. Type `oc project eventstreams`
 - c. Type `oc get pods -n integration`

```
File Edit View Search Terminal Help
statefulset.apps/aspera-1-redis-ha-sentinel scaled
statefulset.apps/aspera-1-redis-ha-server scaled
deployment.extensions/assetrepo-1-asset-files-api scaled
deployment.extensions/assetrepo-1-catalog-api scaled
deployment.extensions/assetrepo-1-clt-haproxy scaled
deployment.extensions/assetrepo-1-dc-main scaled
deployment.extensions/assetrepo-1-portal-catalog scaled
deployment.extensions/assetrepo-1-portal-common-api scaled
deployment.extensions/assetrepo-1-redis-ha-sentinel scaled
deployment.extensions/assetrepo-1-redis-ha-server scaled
statefulset.apps/assetrepo-1-clt-db scaled
deployment.extensions/es-1-ibm-es-access-controller-deploy scaled
deployment.extensions/es-1-ibm-es-collector-deploy scaled
deployment.extensions/es-1-ibm-es-indexmgr-deploy scaled
deployment.extensions/es-1-ibm-es-proxy-deploy scaled
deployment.extensions/es-1-ibm-es-rest-deploy scaled
deployment.extensions/es-1-ibm-es-rest-producer-deploy scaled
deployment.extensions/es-1-ibm-es-rest-proxy-deploy scaled
deployment.extensions/es-1-ibm-es-ui-deploy scaled
statefulset.apps/es-1-ibm-es-elastic-sts scaled
statefulset.apps/es-1-ibm-es-kafka-sts scaled
statefulset.apps/es-1-ibm-es-schemaregistry-sts scaled
statefulset.apps/es-1-ibm-es-zookeeper-sts scaled
[ibmuser@developer ~]$ oc project eventstreams
Now using project "eventstreams" on server "https://master.ibm.demo:8443".
[ibmuser@developer ~]$ oc get pods -n integration
NAME                               READY   STATUS    RESTARTS   AGE
assetrepo-1-asset-files-api-9946b849d-wwlg   1/1     Running   0          3m
assetrepo-1-catalog-api-76cf9d474f-wpn62   1/1     Running   0          3m
assetrepo-1-clt-db-0                      2/2     Running   0          3m
assetrepo-1-clt-haproxy-7856dfb8d6-2jp4r   2/2     Running   0          3m
assetrepo-1-dc-main-5c467f68f7-z62p8       1/1     Running   0          3m
assetrepo-1-portal-catalog-867f9594c5-dvwr5   1/1     Running   0          3m
assetrepo-1-portal-common-api-66498db67b-9gwd2 1/1     Running   0          3m
assetrepo-1-redis-ha-sentinel-86ddcb88dd-ll2l2 1/1     Running   0          3m
assetrepo-1-redis-ha-server-b5cccd65c8-z54ds   1/1     Running   0          3m
ibm-icp4i-prod-ibm-icp4i-prod-6d5648d965-pqksh 2/2     Running   8          37d
```

Note: The third command ensure the Cloud Pak is running. After the third command, once all the pods show 1/1 or equivalent proceed to the next step.

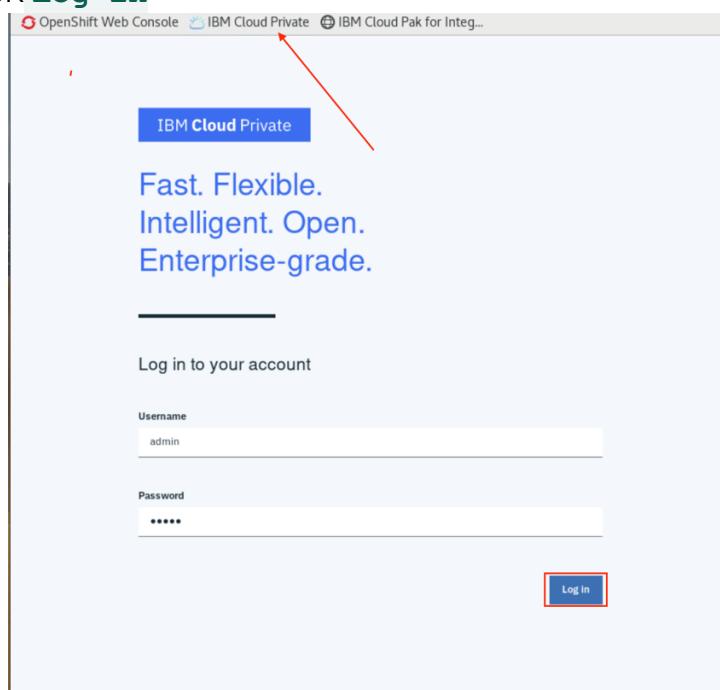
Sync Helm Repositories

1. The Helm repositories must be resynchronized between the repository and the server. Click the **IBM Cloud Private** bookmark.

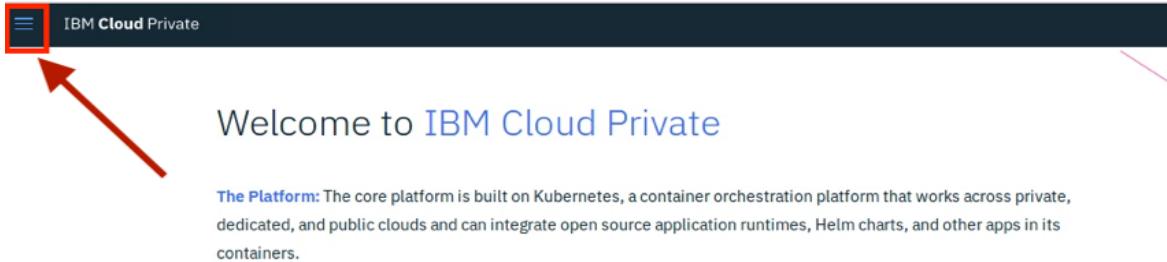


2. Log into IBM Cloud Private with the following credentials:

- a. username: **admin**
- b. password **admin**
- c. Click **Log in**



3. Click the main menu icon in the top left.



4. Click **Manage -> Helm Repositories**.

X IBM Cloud Private

- Dashboard
- Container Images
- Workloads
- Network Access
- Configuration
- Platform
- Manage
 - Identity & Access
 - Resource Security
 - Service Brokers
 - Helm Repositories
- Command Line Tools
- Getting started

5. Click **Sync all** and click **Sync** in the new window to confirm.

The screenshot shows the Helm Repositories page in the IBM Cloud Private interface. A modal dialog box is open, asking "Sync all Helm Repositories?". The dialog contains a message: "Are you sure you want to sync all repositories (6 items)? Only charts that are not syncing will be added to the queue. The process might take a few seconds to sync all of the charts." At the bottom of the dialog are two buttons: "Cancel" and "Sync", with "Sync" highlighted by a red circle labeled "1". In the background, the main table lists six repositories with their URLs. To the right of the table, there is a "Last Sync" section showing five entries, each with a timestamp of "24 minutes ago". A red circle labeled "2" is drawn around the "Last Sync" section. At the bottom of the page, there are pagination controls: "1 of 1 pages", a left arrow, a page number "1", and a right arrow.

Name	Url
ibm-charts	https://
local-charts	https://
mgmt-charts	https://
ibm-charts-public	https://
ibm-community-charts	https://
ibm-entitled-charts	https://raw.githubusercontent.com/IBM/charts/master/repo/entitled/

Items per page: 20 | 1-6 of 6 items

Last Sync

Time Ago
24 minutes ago

1 of 1 pages < 1 >

Explore the platform capabilities.

IBM Cloud Pak for Integration provides a single solution for all of your enterprise integration needs. The platform provides a comprehensive set of industry-leading **capabilities**. Combine the powerful integration capabilities to create, manage, and monitor all of your integrations across applications, messaging, events, APIs, and more.

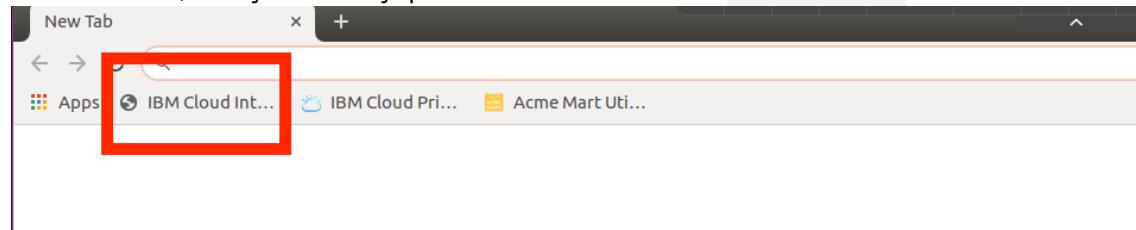
Unlock the power of your data and support the scale required for all of your integration and digital transformation initiatives.

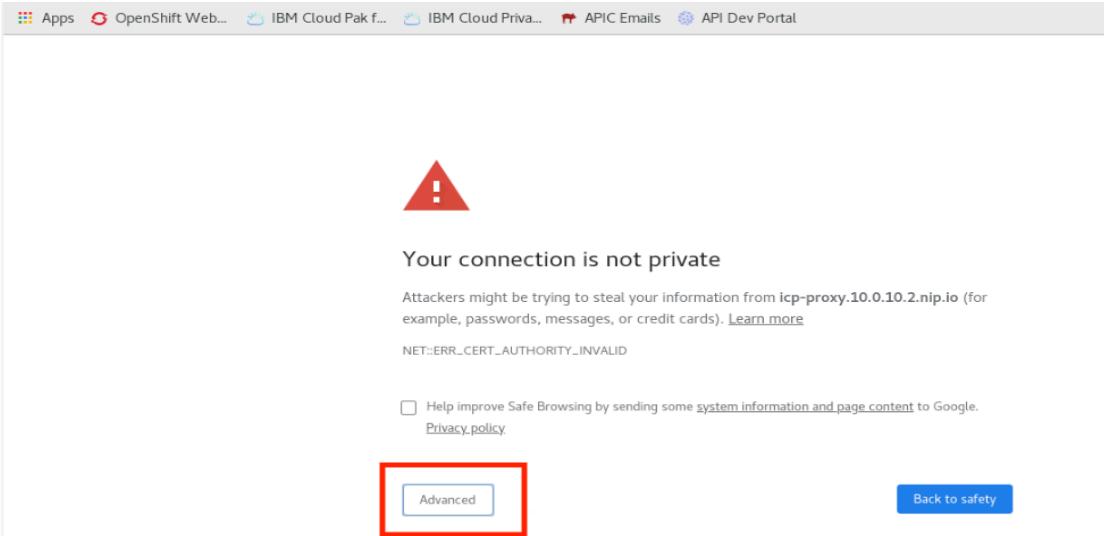
The home page of the **Cloud Pak for Integration** is referred to as the **Platform Navigator**. From the Platform Navigator you are able to navigate to all the integration and development technology contained within the platform.

As of today, the technology included is API management, application integration, message queues, and Kafka event streams. For this lab, we will work with IBM App Connect and IBM Event Streams.

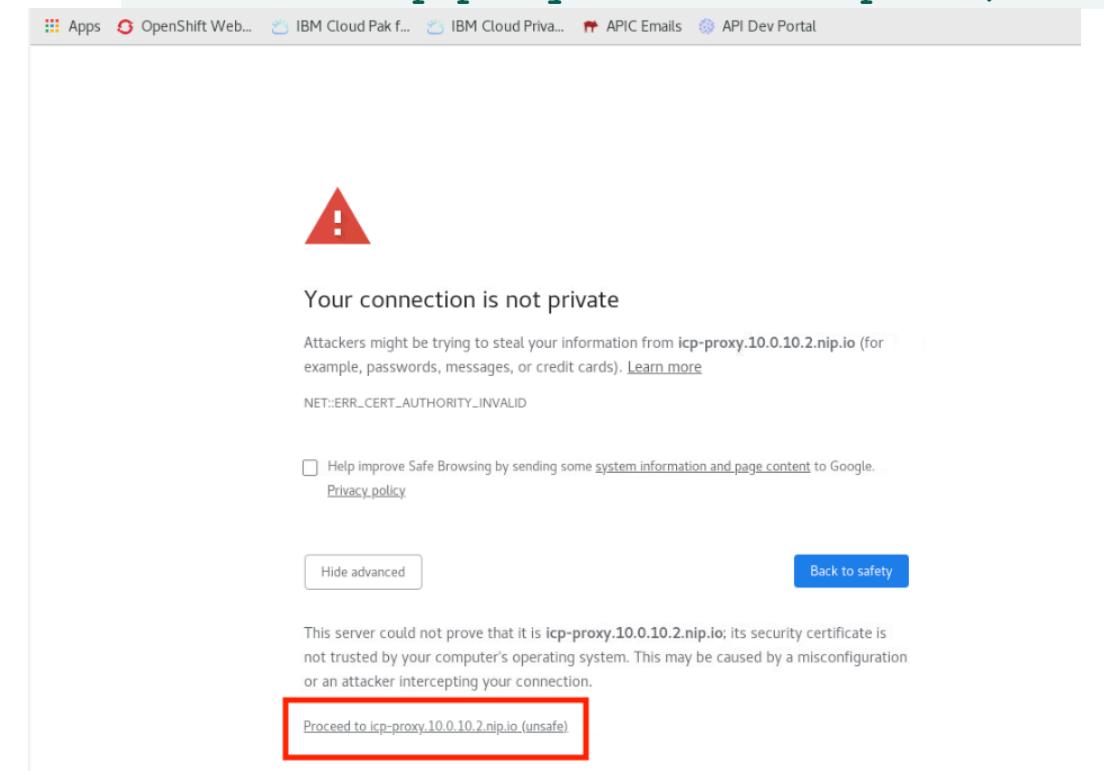
1. Click the **IBM Cloud Pak for Integration** bookmark in the bookmarks bar. If you see the login page, please continue to Task 2 (p. 10)

If instead you see the **Your connection is not private** page, there is an issue reading a certificate. The certificate will not affect this lab, so you may proceed and click **Advanced**.





2. Click **Proceed to icp-proxy.10.0.10.2.nip.io (unsafe)**.



Task 2 – Creating and Configuring an Event Streams Topic

1. You will see the Main Page of the **IBM Cloud Pak**. You will use an existing Event Streams instance. Click **es-1** in the Event Streams services.

The screenshot shows the main dashboard of the IBM Cloud Pak for Integration platform. It displays six service categories in a grid:

- API Connect**: Instances: apic, apic-1. Buttons: Add new instance.
- App Connect**: Instances: ace, ace-1. Buttons: Add new instance.
- MQ**: Instances: mq, mq-1. Buttons: Add new instance.
- Event Streams**: Instances: eventstreams, es-1. The 'es-1' instance is highlighted with a red box. Buttons: Add new instance.
- Aspera**: Instances: aspera, aspera-1. Buttons: Add new instance.
- DataPower**: Instances: datapower, dp-1. Buttons: Add new instance.

Note:

If you see the message "**es-1 did not load correctly**", click **open es-1**. This issue occurs when using a self-signed certificate such as in this demo environment. This will load es-1 in a new tab. Accept the self-sign certificate. Afterwards come back to the original tab and click **Try again**.

2. On the main page of the IBM Event Streams instance **es-1**, check the status of the instance in the bottom right corner of the page. If system is healthy, click the “**Topics**” tab.

The screenshot shows the main interface of the IBM Event Streams instance 'es-1'. At the top, there's a navigation bar with tabs: 'Getting started' (selected), 'Topics' (highlighted with a red box), 'Consumer groups', 'Schemas', 'Monitoring', and 'Toolbox'. To the right of the tabs is a 'Connect to this cluster' button. Below the navigation is a welcome message: 'Welcome to IBM Event Streams, let's get you up and running...'. There are two main cards: 'Use a simulated topic' (with an icon of gears and a wrench) and 'Generate a starter application' (with an icon of a coffee machine). To the right, there's a sidebar titled 'Learn more...' with three items: 'Kafka basics' (with an icon of a cluster), 'Schema basics' (with an icon of a document), and 'Kafka Connect basics' (with an icon of data flow). At the bottom right, there's a green box indicating 'System is healthy' with a checkmark icon.

3. Click on **Create Topic**.

The screenshot shows the 'Topics' page of the IBM Event Streams instance 'es-1'. The 'Topics' tab is selected. At the top, there's a search bar with the placeholder 'Type to search topics'. Below the search bar is a table with columns: 'Name', 'Replicas', and 'Partitions'. In the top right corner, there are two buttons: 'Geo-replication' and 'Create topic' (which is highlighted with a red box). The 'Create topic' button has a small circular icon next to it.

4. Click “**Advanced**” switch and type:

1. **Topic name: customerinfo**
2. Keep Partitions: **1**
3. Keep Replicas: **3**
4. Keep Minimum in-sync replicas: **2**
5. Change **Retention time: 10 minutes**
6. Scroll down and click **Create topic** icon.

Advanced

Core configuration

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

Topic name <small>i</small>	customerinfo
Partitions <small>i</small>	1
Replicas <small>i</small>	3
Minimum in-sync replicas <small>i</small>	2
Retention time <small>i</small>	10 Minute(s)

System is healthy

Note:

- A partition is an ordered list of messages.
- In order to improve availability, each topic can be replicated onto multiple brokers. Hence the number of replicas.
- In-sync replicas determine the reliability achievable for this topic.
- Retention time is how long messages are retained before they are deleted.

[Topics](#)

Create topic

Advanced

Core configuration

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

Topic name <small>i</small>	customerinfo 1
Partitions <small>i</small>	1 2
Replicas <small>i</small>	3 3
Minimum in-sync replicas <small>i</small>	2 4
Retention time <small>i</small>	10 Minute(s) 5

5. You created a topic. Now you need to configure the connection for this topic. Click on **Connect to this cluster**.

The screenshot shows the 'Topics' section of the Event Streams UI. A single topic named 'customerinfo' is listed. It has 3 replicas and 1 partition. In the top right corner of the interface, there is a red box highlighting the 'Connect to this cluster' button.

6. To connect to this cluster, create a token - **API Key**. Insert the name of your application. You can type **customerinfoapp**. Click on **Produce and consume**.

This means this API Key can be used as producer and consumer.

The screenshot shows the 'Cluster connection' dialog. It includes sections for 'Bootstrap server' (with a text input field containing 'tcp://proxy.10.0.10.2.nip.io:31707'), 'Certificates' (with options for Java truststore and PEM certificate), and 'API key' (with a description and a text input field where 'customerinfoapp' is typed). A red circle highlights the 'customerinfoapp' input field.

Cluster connection

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

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

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.

1

Name your application
customerinfoapp

What do you want it to do?

- Produce only →
- Consume only →
- Produce and consume** → (This option is highlighted with a red border)
- Produce, consume, create topics and schemas →

7. Type the name of topic that you will access under **Which topic?**

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.

Which topic?

All topics

customerinfo 1

Great! This topic already exists

Next >

8. Click the **Generate API Key** icon. All external applications will use this as their userid.

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.

Which consumer group?

All

Enter your consumer group name

Generate API key

9. Within the Cluster connection console in Event Streams

1. **API Key:** Click the download icon
2. **PEM Certificate:** Certificate of the Cluster
3. Click the **Bootstrap Server** IP Address to copy

For reference

- a. The **es-api-key** will be on /home/ibmuser/Downloads directory
- b. The **es-cert.pem** will be on /home/ibmuser/Downloads directory

Cluster connection



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.

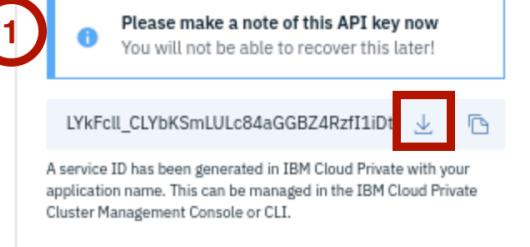
icp-proxy.10.0.10.2.nip.io:32639  

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.

Successful API key generation!

The following API key will allow an application to produce to and consume from topic customerinfo using all consumer groups. Take a copy of it or download it now.



[Create a new API key](#) 

Connecting your tool or application will generate a service ID using your application name, a service policy and an API key in IBM Cloud Private. Additional API keys can be generated in the IBM Cloud Private Cluster Management Console or CLI.

[IBM Cloud Private Cluster Management Console](#)

Certificates

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

Java truststore

Use this for a Java client



Truststore password: password

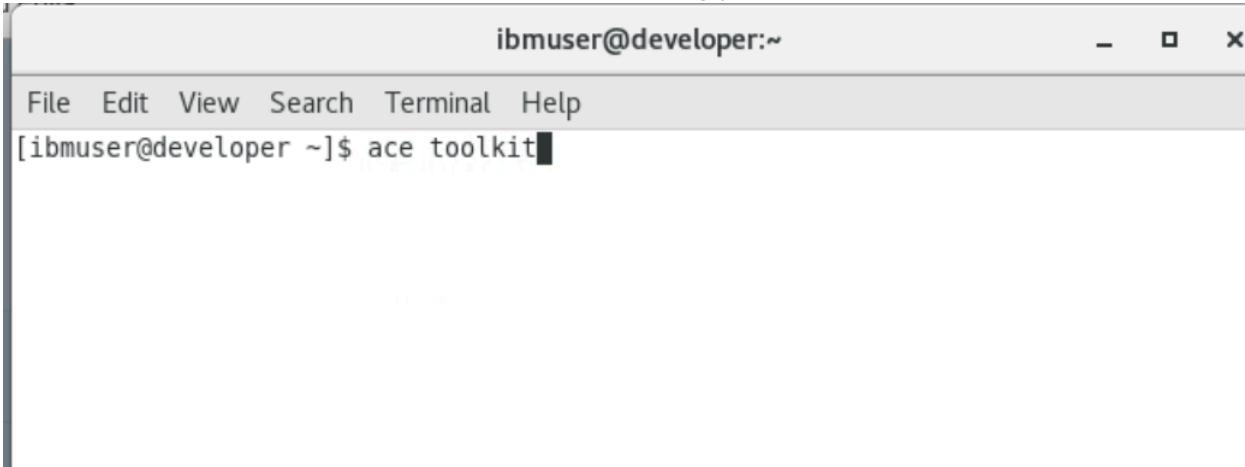
PEM certificate

Use this for anything else



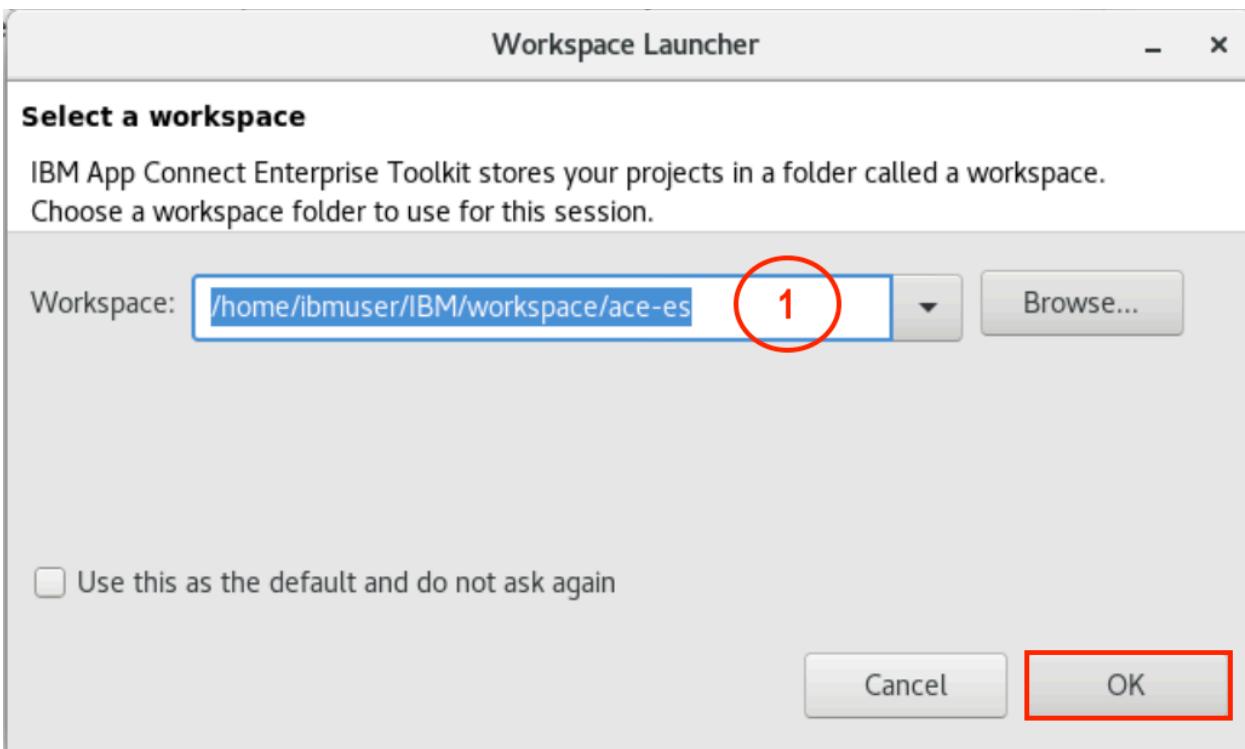
Task 3 - Configure App Connect Enterprise flow using App Connect Enterprise Toolkit

1. You have configured Event Streams. Let's configure App Connect Enterprise. Open a terminal window. Go to the ACE directory **/home/ibmuser/ace-11.0.0.5** and type **ace toolkit**



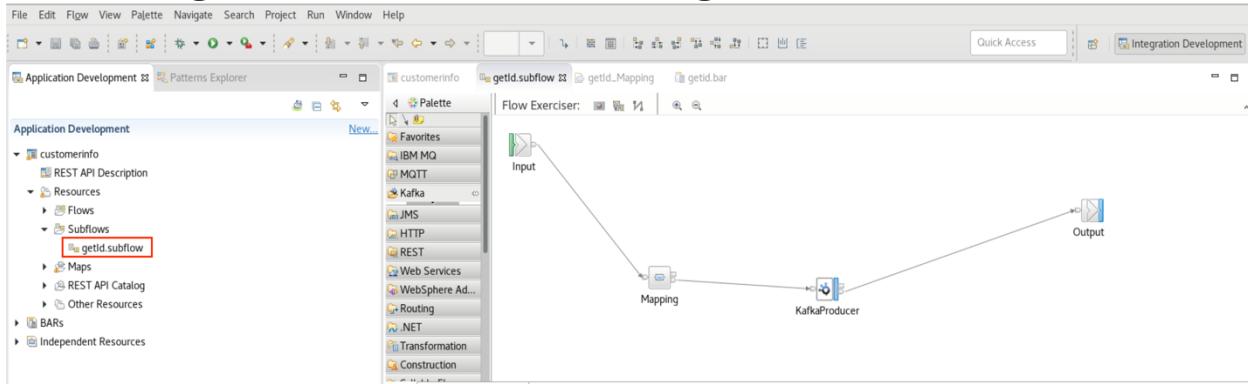
A screenshot of a terminal window titled "ibmuser@developer:~". The window has a standard OS X-style title bar with minimize, maximize, and close buttons. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane shows the command "[ibmuser@developer ~]\$ ace toolkit" being typed into the terminal. The background of the window is light gray.

2. Check the ACE workspace directory and select the ace-es folder. **/home/ibmuser/IBM/workspace/ace-es**. Click **OK** to open the ACE toolkit.

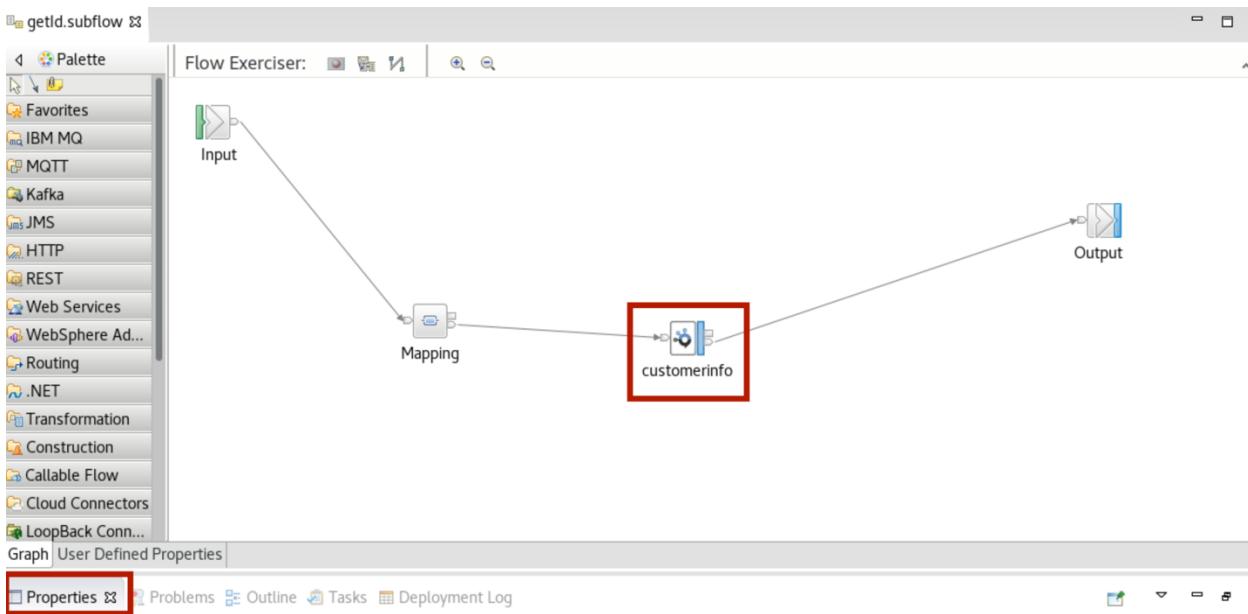


3. You will work with an existing application. Select **customerinfo--> Resources --> Subflows --> getid.subflow**

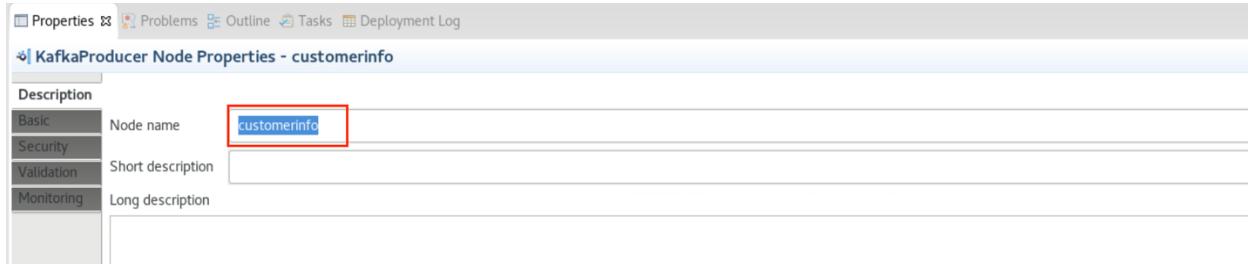
Some errors may show up, you will fix this after you complete the configuration and save the message flow.



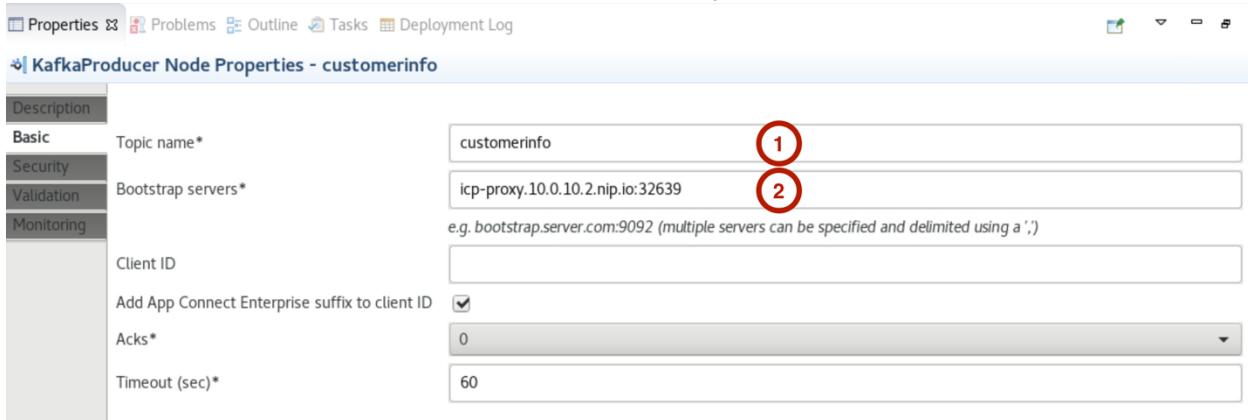
4. Select **Kafka Producer: customerinfo**. Click **Properties** on the lower half of the screen.



5. Click **Description** to change the node name to the suggested name: **customerinfo**.



6. Select **Basic** and type the topic name as: **customerinfo**. This is the topic name you created in the Event Streams configuration. Paste the Bootstrap server value from the Cluster connection console in Event Streams: **tcp-proxy.10.0.10.2.nip.io:32639**.

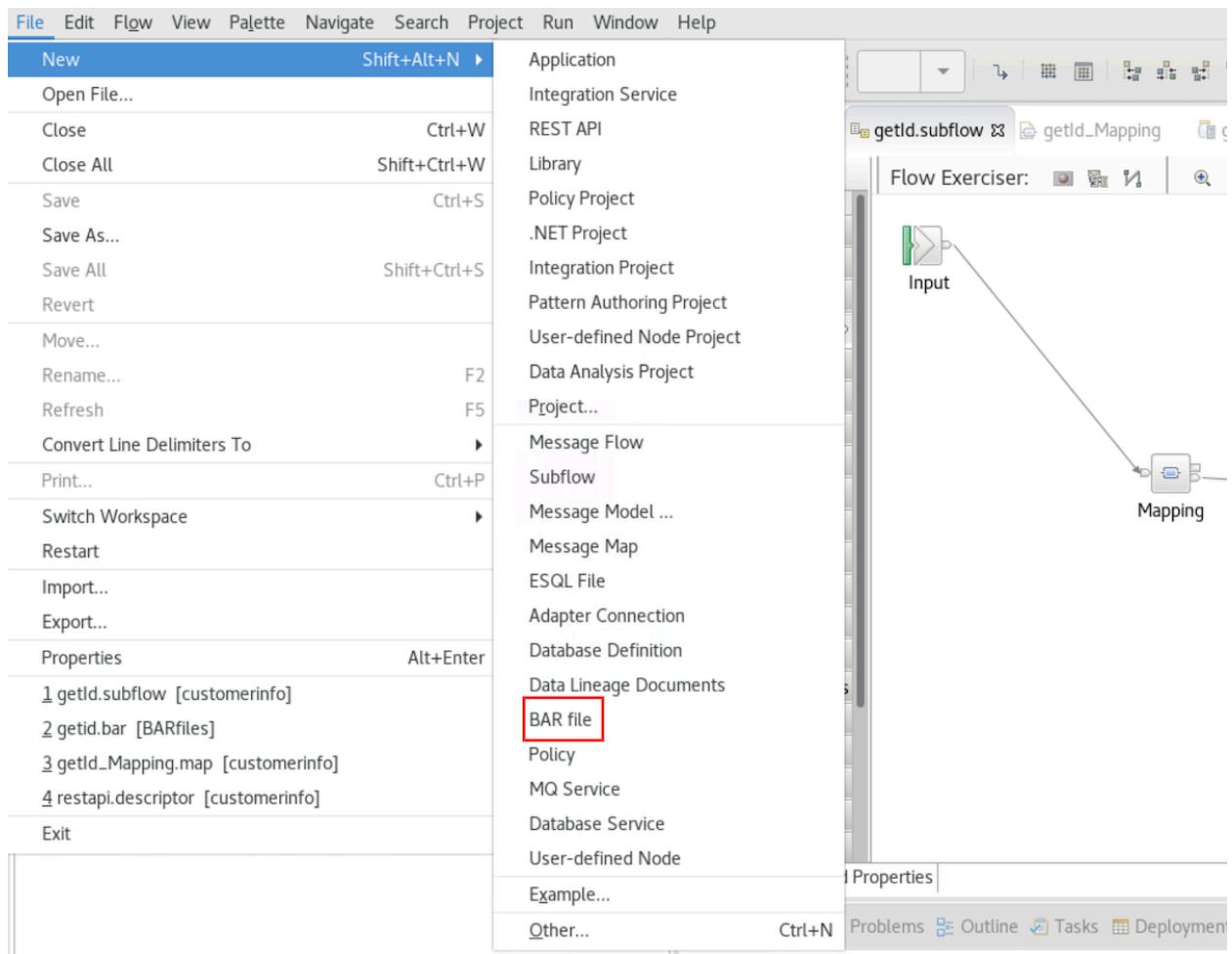


7. In the Security Tab set the Security Protocol to **SASL SSL** and SSL protocol to **TLSv1.2**

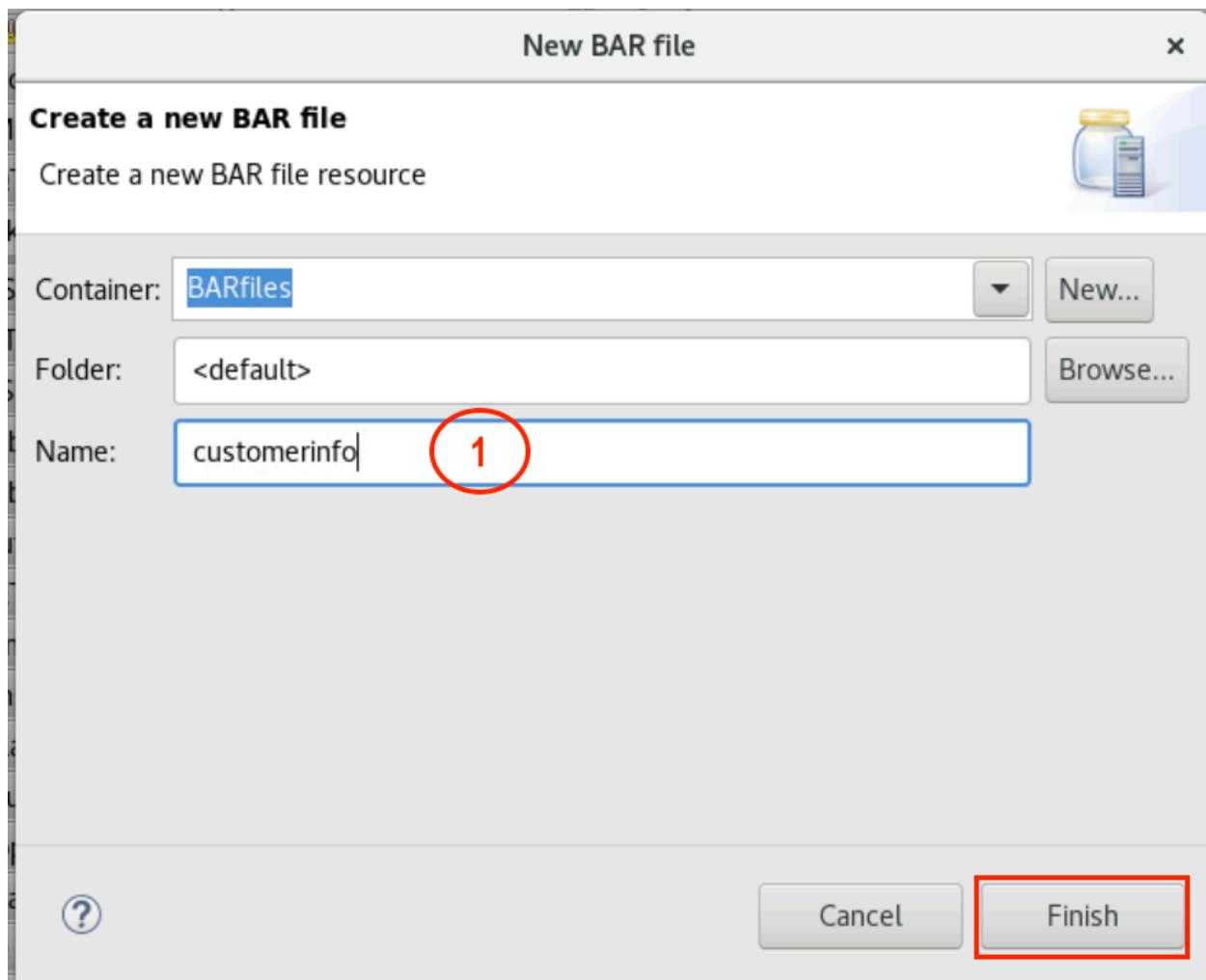
8. Save the message flow and click the **Save** button.



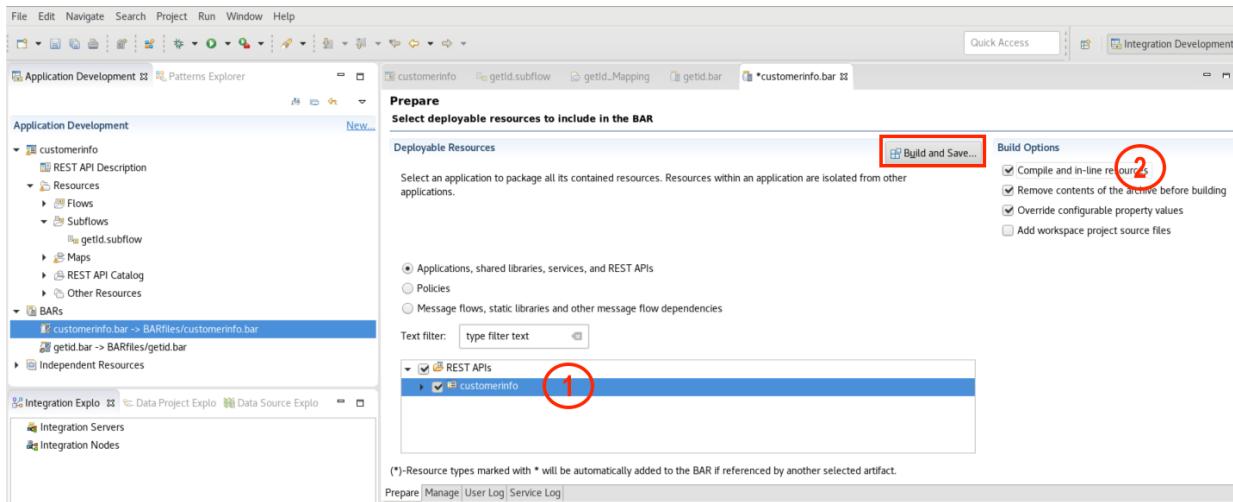
9. Click File --> New --> BAR file.



10. Enter the name of BAR file with the suggested name:
customerinfo. Click **Finish**.



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



12. A pop-up window will show Operation completed successfully. Click **Ok** to confirm and close the App Connect Enterprise toolkit.

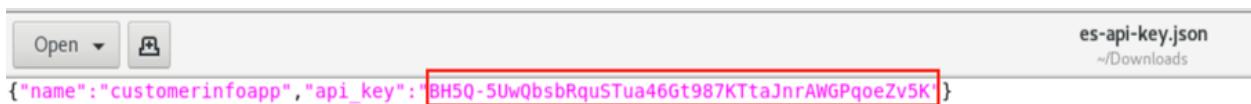
Task 4 - Configuring App Connect Enterprise on Cloud Pak for Integration

In this task, you will configure the App Connect Enterprise parameters for accessing IBM Event Streams.

1. Open the Home folder and go to the config_services folder. The path is **/home/ibmuser/IBM/config_services**. ACE needs the configuration parameters to access Event Streams. You will configure 2 files: setdbparms.txt and truststore-Cert-mykey.crt.

Name	Size	Modified
generateSecrets.sh	3.8 kB	4 Jun
README.md	3.5 kB	7 May
serverconf.yaml	636 bytes	10 Jun
setdbparms.txt	128 bytes	12:16
adminusers.txt	12 bytes	4 Jun
truststorePassword.txt	9 bytes	12:16
truststore-Cert-mykey.crt	0 bytes	12:18
policyDescriptor.xml	0 bytes	7 May
policy.xml	0 bytes	7 May
odbc.ini	0 bytes	7 May
mqsc.txt	0 bytes	7 May
keystorePassword.txt	0 bytes	7 May
keystore-mykey.pass	0 bytes	7 May

2. You need to edit the Event Streams API Key. Go to /home/ibmuser/Downloads directory. Click to edit the API Key file: **es-api-key.json**. Copy the API key between the quotes.



The screenshot shows a file editor window with the file name "es-api-key.json" and the path "~/Downloads". The file content is a JSON object:

```
{"name": "customerinfoapp", "api_key": "8H5Q-5UwQbsbRqusTua46Gt987KTtaJnrAWGPqoeZv5K"}
```

3. Go to /home/ibmuser/config_services and edit **setdbparms.txt**.

Paste the API Key from the previous step over the **API-KEY** section as shown below. ACE will use this parameter to start Kafka services and use the token to access Event Streams. Click **Save**.

```
# resource user password
kafka::KAFKA token BH5Q-5UwQbsbRquSTua46Gt987KTtaJnrAWGPqoeZv5K
setdbparms::truststore dummy password
```

4. Event Streams (Kafka) requires a certificate. You will use the PEM certificate that you downloaded when you configured the Event Streams connection. Open a terminal window and go to /home/ibmuser/Downloads. Move the file es-cert.pem to the directory /home/ibmuser/config_services.

Command: mv es-cert.pem /home/ibmuser/config_services

```
[ibmuser@developer Downloads]$ mv es-cert.pem /home/ibmuser/config_services
[ibmuser@developer Downloads]$
```

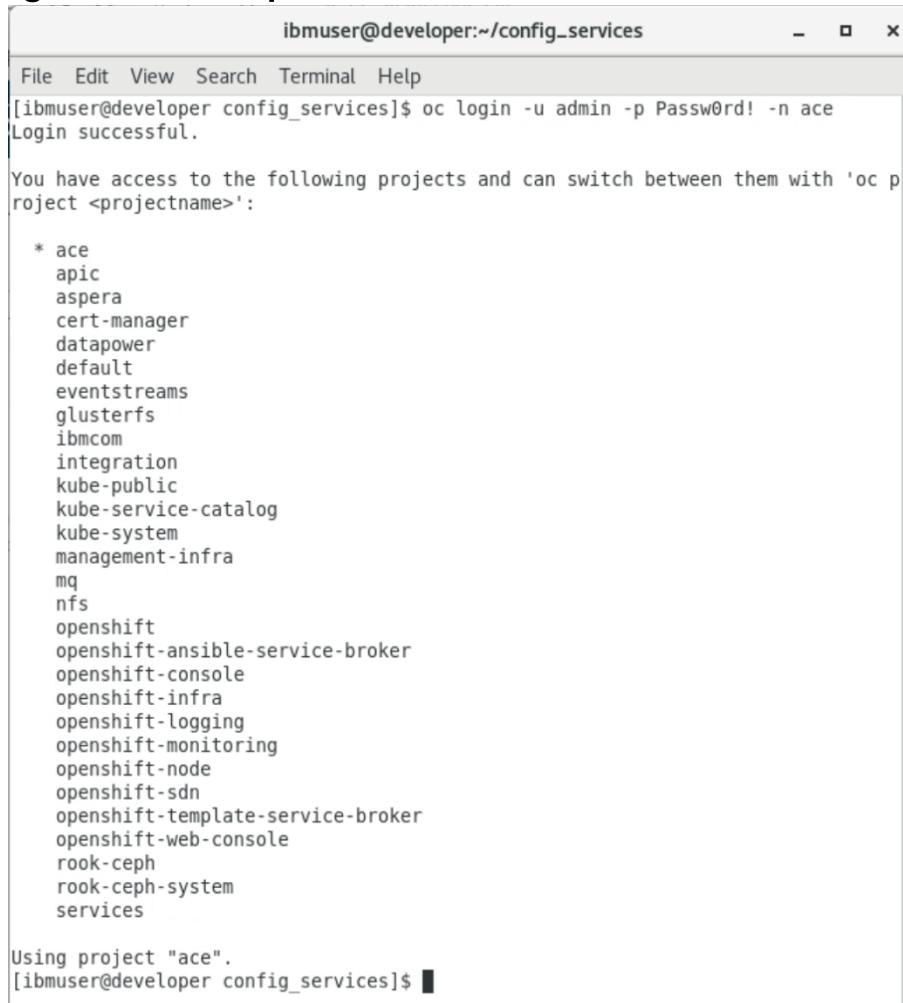
5. Open a terminal and go to /home/ibmuser/config_services. Copy es-cert.pem to the truststoreCert-mykey.crt. Make sure these files are saved and the naming is correct.

Command: mv /home/ibmuser/config_services/es-cert.pem
/home/ibmuser/config_services/truststoreCert-mykey.crt

```
[ibmuser@developer config_services]$ ls
adminPassword.txt      keystore-mykey.key      policy.xml
appPassword.txt        keystore-mykey.pass     README.md
config.tar.gz          keystorePassword.txt   serverconf.yaml
es-cert.pem            mqsc.txt                setdbparms.txt
generateSecrets.sh     odbc.ini               truststoreCert-mykey.crt
keystore-mykey.crt    policyDescriptor.xml   truststorePassword.txt
[ibmuser@developer config_services]$ mv /home/ibmuser/config_services/es-cert.pem
/home/ibmuser/config_services/truststoreCert-mykey.crt
```

6. Now you will create an ACE service. Administrator access is required to create the service. Open a terminal window and go to /home/ibmuser/config_services directory. Type

```
oc login -u admin -p Passw0rd! -n ace
```



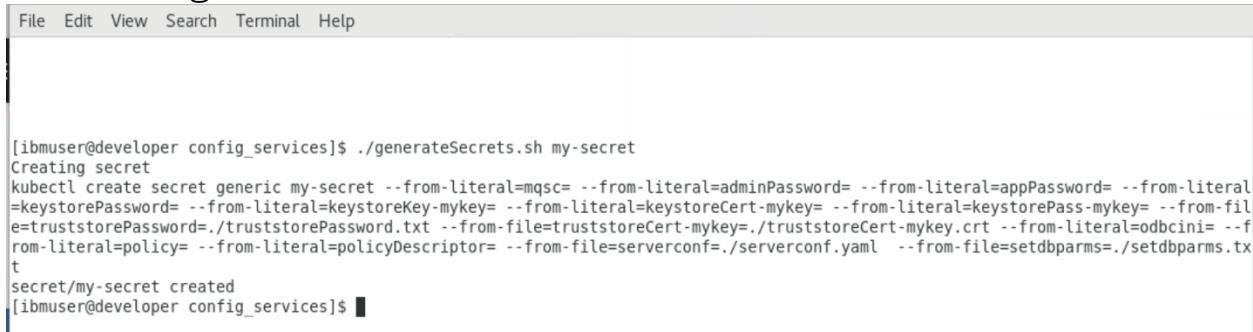
The screenshot shows a terminal window titled "ibmuser@developer:~/config_services". The window contains the following text:

```
ibmuser@developer:~/config_services
File Edit View Search Terminal Help
[ibmuser@developer config_services]$ oc login -u admin -p Passw0rd! -n ace
Login successful.

You have access to the following projects and can switch between them with 'oc project <projectname>':
* ace
  apic
  aspera
  cert-manager
  datapower
  default
  eventstreams
  glusterfs
  ibmcom
  integration
  kube-public
  kube-service-catalog
  kube-system
  management-infra
  mq
  nfs
  openshift
  openshift-ansible-service-broker
  openshift-console
  openshift-infra
  openshift-logging
  openshift-monitoring
  openshift-node
  openshift-sdn
  openshift-template-service-broker
  openshift-web-console
  rook-ceph
  rook-ceph-system
  services

Using project "ace".
[ibmuser@developer config_services]$
```

7. Enter **./generateSecrets.sh my-secret**. You have created an ACE configuration.



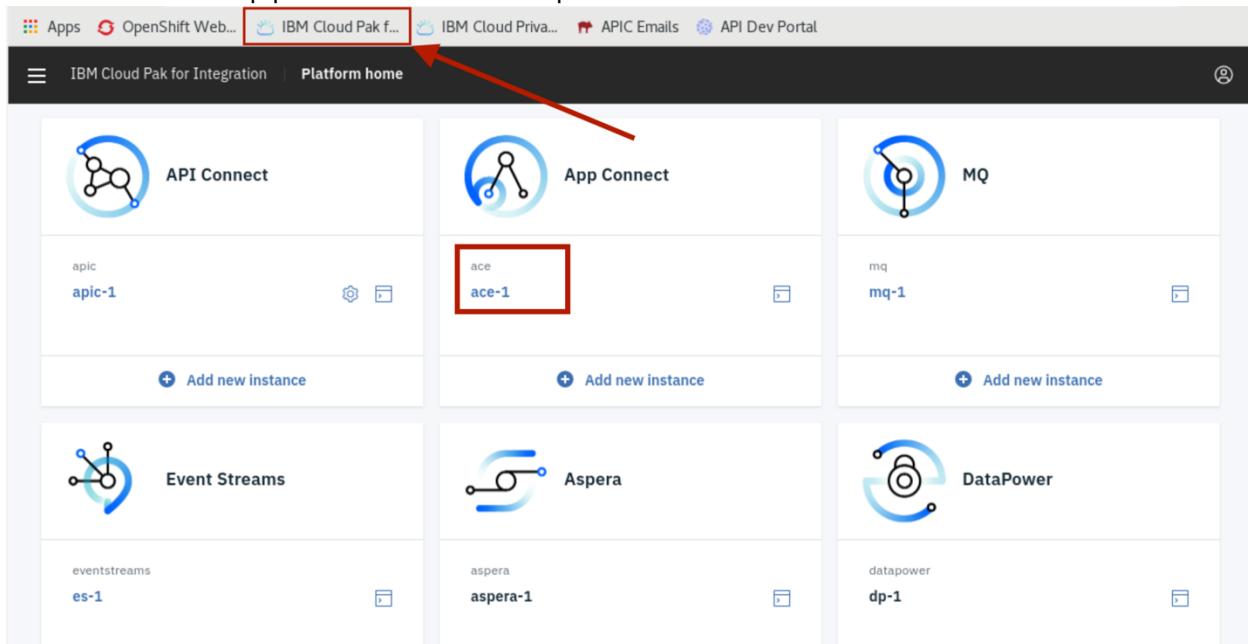
The screenshot shows a terminal window with the following text:

```
File Edit View Search Terminal Help
[ibmuser@developer config_services]$ ./generateSecrets.sh my-secret
Creating secret
kubectl create secret generic my-secret --from-literal=mqsc= --from-literal=adminPassword= --from-literal=appPassword= --from-literal=keystorePassword= --from-literal=keystoreKey-mykey= --from-literal=keystoreCert-mykey= --from-literal=keystorePass-mykey= --from-file=truststorePassword=./truststorePassword.txt --from-file=truststoreCert-mykey=./truststoreCert-mykey.crt --from-literal=dbcini= --from-literal=policy= --from-literal=policyDescriptor= --from-file=serverconf=./serverconf.yaml --from-file=setdbparms=./setdbparms.txt
secret/my-secret created
[ibmuser@developer config_services]$
```

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 the information to run an ACE instance.

1. Open a new tab and click on the **IBM Cloud Pak** bookmark. Click on the App Connect Enterprise instance: **ace-1**.



The screenshot shows the IBM Cloud Pak for Integration Platform home page. The top navigation bar includes links for Apps, OpenShift Web..., IBM Cloud Pak f..., IBM Cloud Priv..., API Emails, and API Dev Portal. The main content area is titled "Platform home". It features a grid of service icons and instances. The "App Connect" section contains one instance named "ace" with the sub-instance "ace-1" highlighted by a red box. Other visible instances include "api" (apic-1), "MQ" (mq-1), "Event Streams" (eventstreams-es-1), "Aspera" (aspera-aspera-1), and "DataPower" (datapower-dp-1). Each service row includes an "Add new instance" button.

Note:

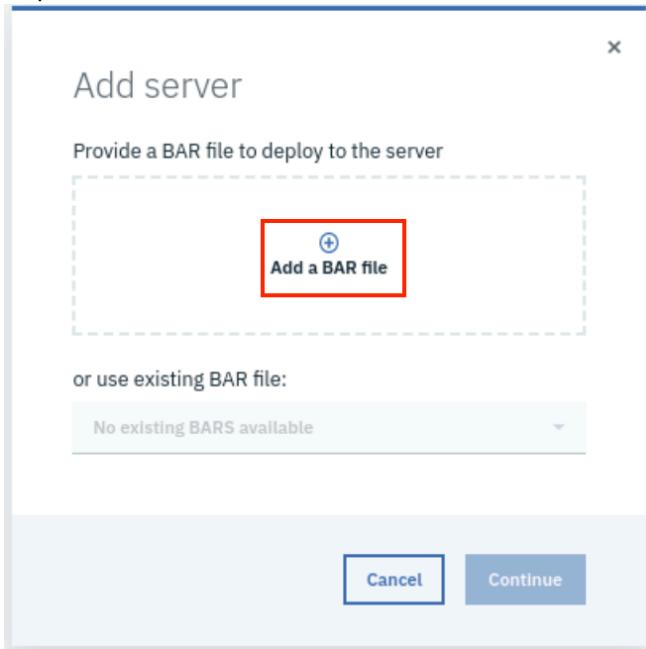
If you see the message "**ace-1 did not load correctly**", click **Open ace-1**. This issue occurs when using a self-signed certificate such as in this demo environment. This will load ace-1 in a new tab. Accept the self-sign certificate. Afterwards come back to the original tab and click **Try again**.

2. You will deploy the BAR file that you saved in the App Connect Enterprise toolkit. Click **Add Server**.

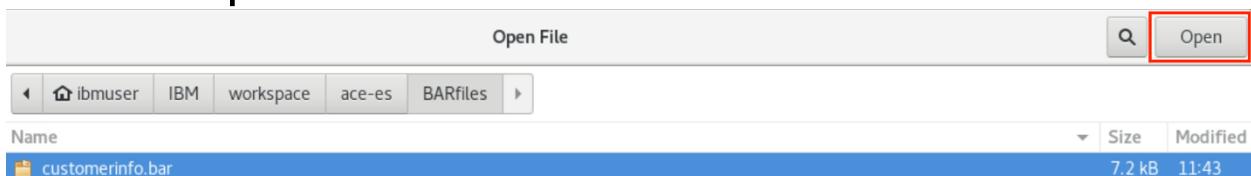


The screenshot shows the "Servers" tab in the App Connect Enterprise interface. At the top, there are tabs for "Integrations" and "Servers", with "Servers" being the active tab. A status message at the top right indicates "Last refreshed: 10/30/2019, 12:22:12 PM Refresh". Below the tabs is a search bar with the placeholder "Search". On the far right of the search bar is a blue "Add server" button with a white plus sign icon, which is also highlighted with a red box.

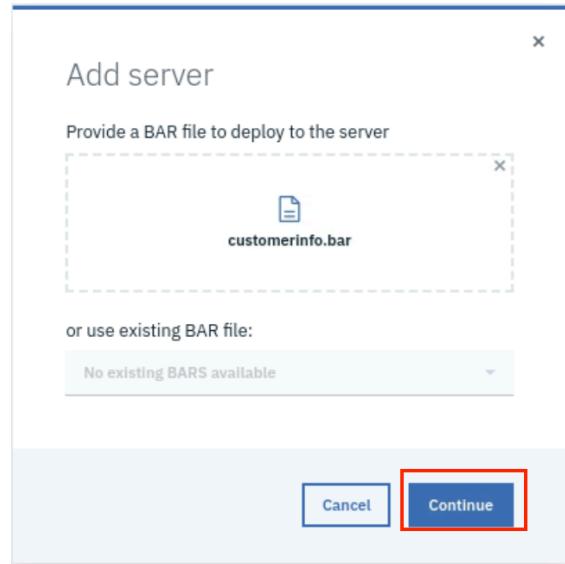
3. On the pop-up window, click **Add a BAR file**.



4. Locate the **customerinfo.bar** within the ace-es folder. Navigate to the /home/ibmuser/IBM/workspace/ace-es/BARfiles. Click **Open**.

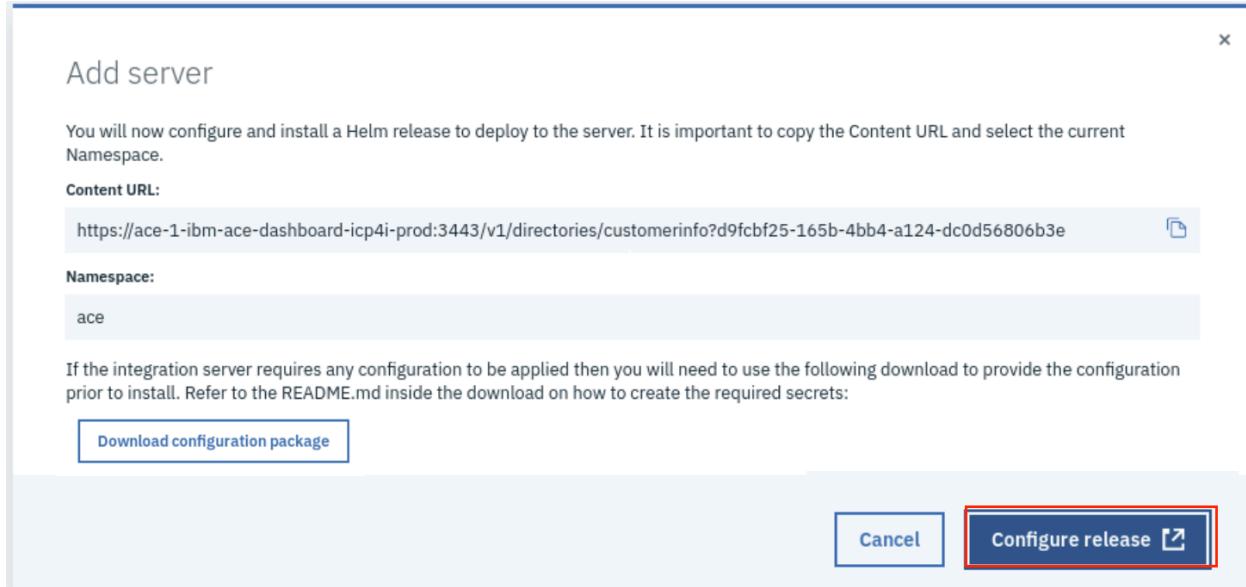


5. ACE will show the BAR file that you selected: **customerinfo.bar**.
Click **Continue**.



6. A pop up with a content URL will appear. Copy the **Content URL** and click **Configure Release**. You may notice the icon Download configuration package. You can skip this because you already completed this.

Note: Your URL will be different than the one displayed below.



7. This is the ACE installation server main page. Verify the ACE version and click **Configure** in the bottom right corner or in the upper left corner.

The screenshot shows the IBM Cloud Private interface with the URL [ibm-ace-server-icp4i-prod V 2.0.0](#). The 'Configuration' tab is selected. The page displays information about the IBM APP CONNECT ENTERPRISE helm chart, including its introduction, chart details, prerequisites, and a 'Configure' button in the bottom right corner.

8. Type the parameters for ACE server:

- 1) Helm release name: **customerinfo**
- 2) Target namespace: **ace**
- 3) Check as **Target Cluster: local-cluster**
- 4) Check **License agreement** checkbox

The screenshot shows the configuration form for the ACE helm chart. The fields are highlighted with red circles:

- Helm release name ***: customerinfo (1)
- Target namespace ***: ace (2)
- Target Cluster ***: local-cluster (3)
- License ***: I have read and agreed to the [License agreement](#) (4)

9. Scroll down and expand the **Quick start** section.
- 1) Content Server URL space, paste the Content URL from Step 6
 - 2) In the Proxy Node IP, enter **icp-proxy.10.0.10.2.nip.io**
You will use this address when you invoke the API from ACE.

Parameters

To install this chart, additional configuration is needed in Quick start. To customize installation, view and edit All parameters.

Quick start
Required and recommended parameters to view and edit.

Content Server URL *
https://ace-1-ibm-ace-dashboard-icp4i-prod:3443/v1/directories/customerinfo?d9fcfb25-165b-4bb4-a124-dc0d56806b3e

Service
Service settings

Proxy Node IP or FQDN * ⓘ
icp-proxy.10.0.10.2.nip.io

10. Scroll down and expand **All parameters:**
- 1) Check the Production usage box checkbox
 - 2) Check the **Local Default Queue Manager** checkbox

↙ **All parameters**
Other required, optional, and read-only parameters to view and edit.

Content Server URL *
https://ace-1-ibm-ace-dashboard-icp4i-prod:3443/v1/directories/orders?d3e5cfce-8dce-4467-8755-76f48c71b837

Production usage

Local default Queue Manager

Architecture scheduling preference ⓘ
amd64

Enable IBM App Connect Designer flows

11. Scroll down until Docker Image settings:

- 1) Change Image pull policy: **Always**
- 2) Type Image pull secret: **deployer-dockercfg-bnjx2**

Docker image
Specify the image to run. The image is selected based on whether MQ is enabled.

Docker image for ACE only *	Docker image for ACE with MQ *
docker-registry.default.svc:5000/ace/ibm-ace-server-prod	docker-registry.default.svc:5000/ace/ibm-ace-mq-server-prod
Configurator docker image *	Docker image for ACE with Designer flows *
docker-registry.default.svc:5000/ace/ibm-ace-icp-configurator-prod	docker-registry.default.svc:5000/ace/ibm-ace-designer-flows-prod
Image tag *	Image pull policy
11.0.0.5-amd64	Always 1
Image pull secret ?	2
<code>deployer-dockercfg-bnjx2</code>	

12. Scroll down to the Integration Server

- 1) In List of certificate aliases for the truststore, enter **mykey**
- 2) In The name of the secret to create or to use that contains the server configuration, enter **my-secret**

Integration Server
Define configuration for the Integration Server

Integration Server name	List of key aliases for the keystore
Enter value	Enter value
List of certificate aliases for the truststore	Name of the default application
mykey 1	Enter value
The name of the secret to create or to use that contains the server configuration ?	File system group ID
my-secret 2	Enter value

13. Scroll down and set **Replica Count** to 1

Configuration for ACE (without MQ) deployments
Configuration settings for specifying required resources when running ACE without MQ

CPU request *	Memory request *
1	1024Mi
CPU limit *	Memory limit * ?
1	1024Mi
Replica count	
1	

14. Scroll down to Persistence settings

- 1) Uncheck **Enable persistence**
- 2) Uncheck **Use dynamic provisioning**. Click **Install**.

The screenshot shows the IBM Cloud Private interface with a cluster named 'mycluster'. It displays CPU and Memory requests and limits. Below these, under 'Persistence settings', there are two checkboxes: 'Enable persistence' and 'Use dynamic provisioning'. Both checkboxes are currently unchecked. Red circles with numbers 1 and 2 are overlaid on these checkboxes to indicate they need to be unselected.

15. View the progress of the install via Helm Releases as prompted. Click **View Helm Release**.

A modal window is displayed with the following content:
Installation started. For progress view your Helm release.
[View Helm Release](#)
[Return to Catalog](#)

Task 6 - Testing ACE API sending a message to Event Streams

1. The ACE installation server may take some time to deploy.
(Approximately 5 mins) To verify the installation, go to IBM Cloud Private. Click the **Menu-->Workloads-->Helm Releases**.
2. Search for **customerinfo**. Click on the customerinfo line. Locate Pods within the Helm release and verify the pod is set to **Ready (1/1)** and **Status Running**.

The screenshot shows two parts of the IBM Cloud Private interface. The top part is a table titled 'customerinfo' showing the release details. It has columns for Name, Namespace, Status, Chart name, Current version, Available version, and Updated. One row is shown: 'customerinfo' in namespace 'ace' is 'Deployed' with chart 'ibm-ace-server-icp4i-prod', current version '2.0.0', available version '2.2.0', and updated on 'October 31, 2019 07:43pm'. The bottom part is a table titled 'Pod' showing the status of pods within the release. It has columns for Name, READY, Status, RESTARTS, and Age. One pod is listed: 'customerinfo-ib-1352-0' is '1/1' ready, 'Running', has '0' restarts, and is '3m28s' old.

3. You can open the ACE server that was installed. Go to the Cloud Pak for Integration tab (it is already opened). Click **Done**.

Add server

You will now configure and install a Helm release to deploy to the server. It is important to copy the Content URL and select the current Namespace.

Content URL:

<https://ace-1-ibm-ace-dashboard-icp4i-prod:3443/v1/directories/customerinfo?d9fcfb25-165b-4bb4-a124-dc0d56806b3e>



Namespace:

ace

If the integration server requires any configuration to be applied then you will need to use the following download to provide the configuration prior to install. Refer to the README.md inside the download on how to create the required secrets:

Done

4. You see the ACE Server running. Select the **customerinfo** server.

Servers

Integrations Servers

customerinfo
Server (with MQ)
Release name: customerinfo
1/1 replicas
Started

5. Click the **customerinfo API**.

customerinfo
Release name: customerinfo
Contents Properties Policies

customerinfo API
Started

6. ACE created a REST API base URL and OpenAPI document.

Click on the **OpenAPI** document.

IBM Cloud Pak for Integration App Connect ace | ace-1

Dashboard / Server: customerinfo / API: customerinfo

customerinfo Started

Documentation Contents Properties Other resources

REST API Base URL
<http://icp-proxy.10.0.10.2.nip.io:31938/customerinfo/v1>

OpenAPI document
<http://icp-proxy.10.0.10.2.nip.io:31938/customerinfo/v1/swagger.json>

Expand all ▾ Collapse all ▲

Search /{id}

7. You can check the json document created.

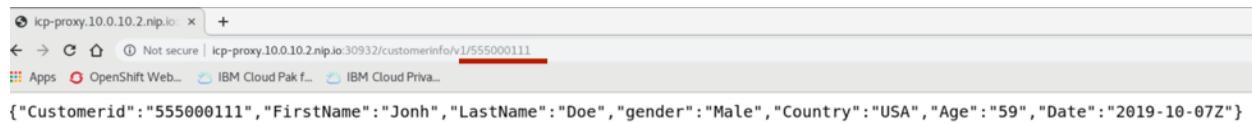
```
{ "swagger" : "2.0",
  "info" : {
    "title" : "customerinfo",
    "version" : "1.0.0",
    "description" : "customerinfo"
  },
  "paths" : {
    "/{id}" : {
      "get" : {
        "operationId" : "getId",
        "responses" : {
          "200" : {
            "description" : "The operation was successful.",
            "schema" : {
              "type" : "string"
            }
          }
        },
        "produces" : [ "application/json" ],
        "description" : "Retrieve id",
        "parameters" : [ {
          "required" : true,
          "name" : "id",
          "in" : "path",
          "type" : "string"
        } ]
      }
    }
  },
  "basePath" : "/customerinfo/v1",
  "definitions" : {
    "Customer" : {
      "type" : "object",
      "properties" : {
        "Customerid" : {
          "type" : "string"
        },
        "FirstName" : {
          "type" : "string"
        },
        "LastName" : {
          "type" : "string"
        },
        "gender" : {
          "type" : "string"
        },
        "Country" : {
          "type" : "string"
        },
        "Age" : {
          "type" : "string"
        },
        "Date" : {
          "type" : "string"
        }
      }
    }
  },
  "schemes" : [ "http" ],
  "host" : "icp-proxy.10.0.10.2.nip.io:31571"
}
```

8. Back to the **customerinfo** API. Click on **REST API base URL**.

The screenshot shows the IBM Cloud Pak for Integration interface. At the top, there is a navigation bar with the following items: three horizontal bars icon, IBM Cloud Pak for Integration, App Connect, ace | ace-1, a gear icon, and a help icon. Below the navigation bar, the page title is "Dashboard / Server: customerinfo / API: customerinfo". The main content area has a card titled "customerinfo". Inside the card, there are tabs: Documentation (which is selected), Contents, Properties, and Other resources. Under the Documentation tab, there is a section with the heading "REST API Base URL" followed by a blue link: "http://icp-proxy.10.0.10.2.nip.io:31938/customerinfo/v1". This link is highlighted with a red rectangular box. To the right of this link, there is another link: "OpenAPI document http://icp-proxy.10.0.10.2.nip.io:31938/customerinfo/v1/swagger.json". At the bottom of the card, there is a search bar with the placeholder "Search" and a "Collapse all" button.

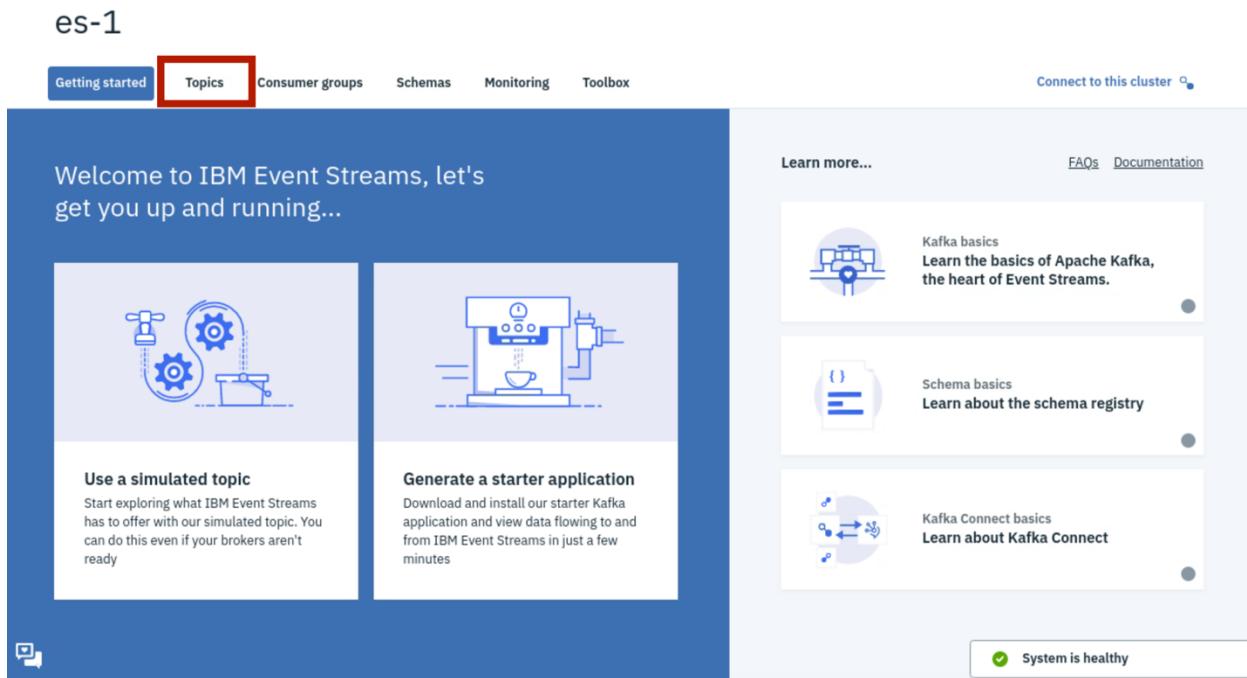
9. Complete the URL by appending: 555000111 after v1. See the results. You sent a message from App Connect Enterprise to IBM Event Streams.

<http://icp-proxy.10.0.10.2.nip.io:30932/customerinfo/v1/555000111>



```
{"Customerid": "555000111", "FirstName": "Jonh", "LastName": "Doe", "gender": "Male", "Country": "USA", "Age": "59", "Date": "2019-10-07Z"}
```

10. Let's verify the message was received by Event Streams. Open a new browser and click on **IBM Cloud Pak**. In the main page click on the Event Streams instance **es-1**. Click **Topics** to open a list of topics of this Event Streams instance.



Welcome to IBM Event Streams, let's get you up and running...

Use a simulated topic
Start exploring what IBM Event Streams has to offer with our simulated topic. You can do this even if your brokers aren't ready

Generate a starter application
Download and install our starter Kafka application and view data flowing to and from IBM Event Streams in just a few minutes

Learn more... [FAQs](#) [Documentation](#)

Kafka basics
Learn the basics of Apache Kafka, the heart of Event Streams.

Schema basics
Learn about the schema registry

Kafka Connect basics
Learn about Kafka Connect

System is healthy

11. Select the **customerinfo** topic. Click **Messages** to check if the message from ACE has arrived.

Note: To view more messages, repeat Step 9 multiple times.
Messages will appear after each App Connect Enterprise to Event Streams connection.

Indexed timestamp	Partition	Offset
10/7/2019, 4:41:56 PM	0	0
10/7/2019, 4:47:16 PM	0	1
10/7/2019, 4:48:18 PM	0	2

12. You will see the list of messages that are stored on the Event Streams topic. Click on a **message** and view the information from the message on the right hand panel.

Name	Value
Customerid	555000111
FirstName	John
LastName	Doe
gender	Male
Country	USA
Age	59
Date	2019-10-07Z

Summary

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

- Deploy a back-end integration to containers that are readily available as a scalable web service.
- Secure access to the back-end integration by creating a secure, governed API using the OpenAPI definition of the integration.
- Use API Test and Monitor to execute quick and easy functional testing of your new APIs.

Now that you've made your back-end integrations ready for external distribution, your developer community will be able to access the APIs via a developer portal. The developer portal is included in the platform and provides a full-featured out-of-the-box experience to onboard and nurture your API consumers. For more information about Cloud Pak for Integration, go to <https://www.ibm.com/cloud/cloud-pak-for-integration>.