

Try IBM Event Streams in a demo environment

Built on open source Apache Kafka, IBM Event Streams is an event-streaming platform that helps you build smart applications that can react to events as they happen.

Tutorial

- Apache Kafka deployment that maximizes the spread of Kafka brokers across worker nodes of your Kubernetes cluster. This creates a highly-available configuration making the deployment resilient to many classes of failure with automatic restart of brokers included.
- Health check information and options to resolve issues with your clusters and brokers.
- Geo-replication of your topics between clusters to enable disaster recovery and scalability.
- UI for browsing messages to help view and filter hundreds of thousands of messages, including options to drill in and see message details from a set time.
- Encrypted communication between internal components and encrypted storage.
- Security with authentication and authorization.

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

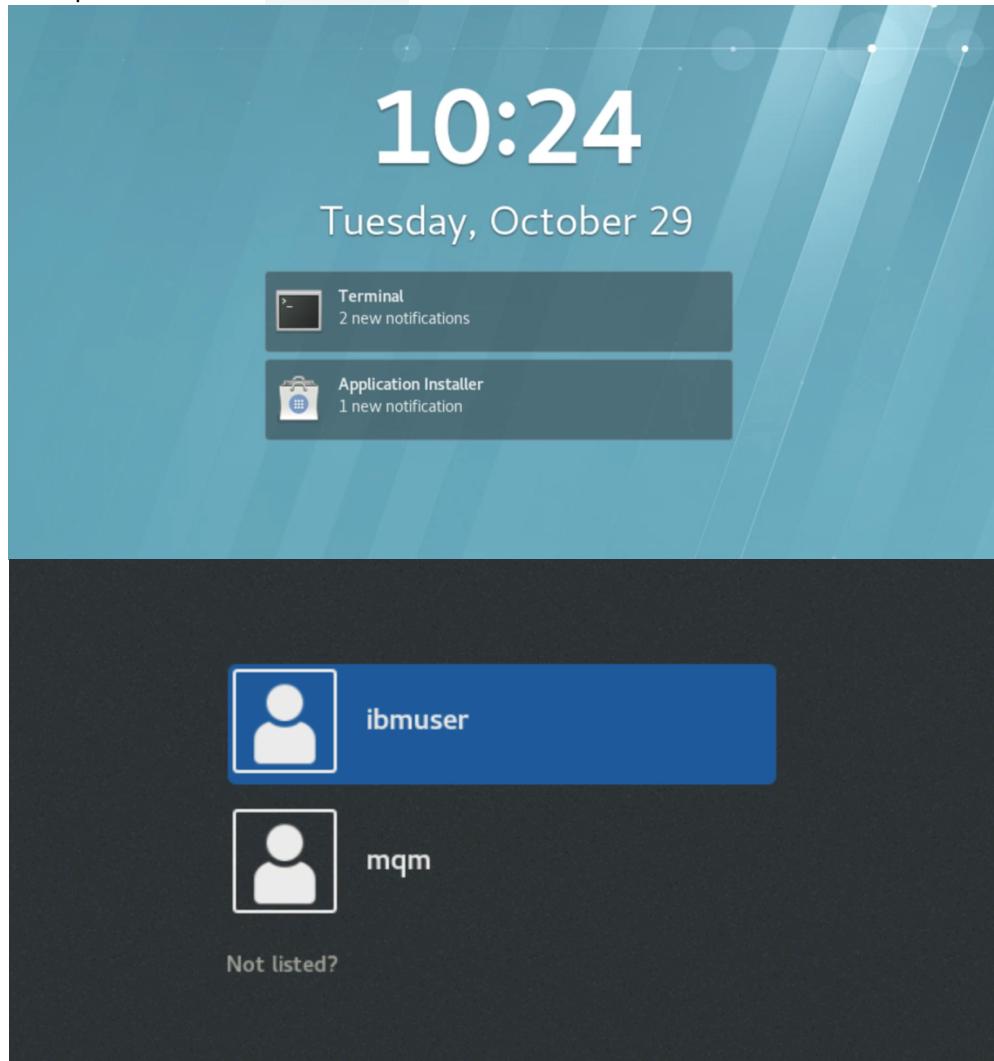
- Install an IBM Event Streams instance on the IBM Cloud Pak for Integration running on Red Hat OpenShift
- Test IBM Event Streams using the provided starter application
- Use Event Streams Toolbox

Task 1 -- Prepare IBM Cloud Pak for Integration for Installing Event Streams

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

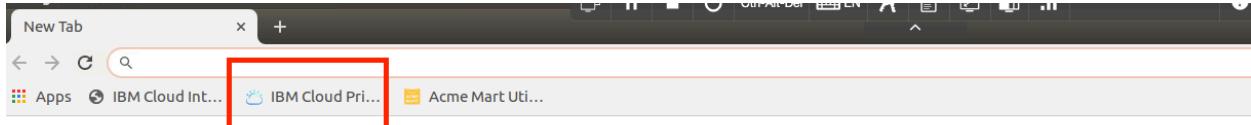
1. The next step is to check if the environment is done loading. Open a terminal console window by clicking the Terminal icon in the Desktop.
 - a. Type `./startup-scripts/oc-startup.sh`
 - b. Type `oc project eventstreams`
 - c. Type `oc get pods -n integration`

```
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 on the **IBM Cloud Private** bookmark.

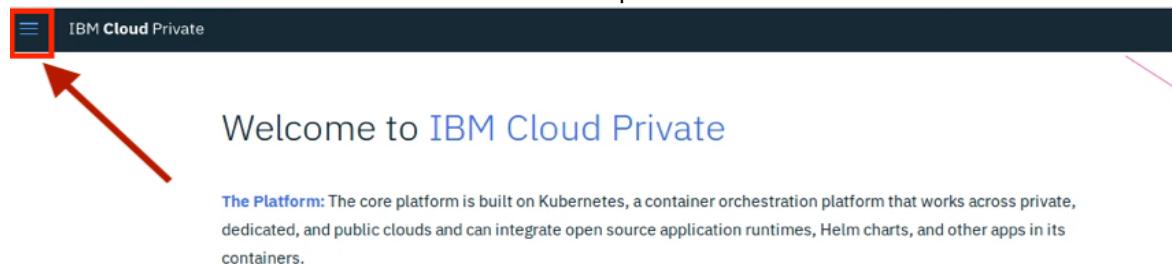


2. Log into IBM Cloud Private with the following credentials:

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

A screenshot of the IBM Cloud Private login page. At the top, there is a navigation bar with links to 'OpenShift Web Console', 'IBM Cloud Private' (which has a red arrow pointing to it from the browser's bookmark bar), and 'IBM Cloud Pak for Integ...'. Below the navigation bar, there is a large blue button labeled 'IBM Cloud Private'. Underneath the button, the text 'Fast. Flexible. Intelligent. Open. Enterprise-grade.' is displayed. Below this, there is a section titled 'Log in to your account' with fields for 'Username' (containing 'admin') and 'Password' (containing '*****'). At the bottom right of the form, there is a blue 'Log in' button, which is also highlighted with a red rectangular box.

3. Click the main menu icon at 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

Name 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 the 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". The "Sync" button is highlighted with a red circle and labeled with the number "1". In the background, the main table lists six repositories with their URLs. To the right of the table, there is a "Last Sync" column showing times like "24 minutes ago". A red circle labeled "2" is drawn around the "Last Sync" column header. At the bottom of the page, there are pagination controls and a status message "1 of 1 pages".

Name	Url	Last Sync
ibm-charts	https://	24 minutes ago
local-charts	https://	24 minutes ago
mgmt-charts	https://	24 minutes ago
ibm-charts-public	https://	24 minutes ago
ibm-community-charts	https://	24 minutes ago
ibm-entitled-charts	https://raw.githubusercontent.com/IBM/charts/master/repo/entitled/	Completed

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.

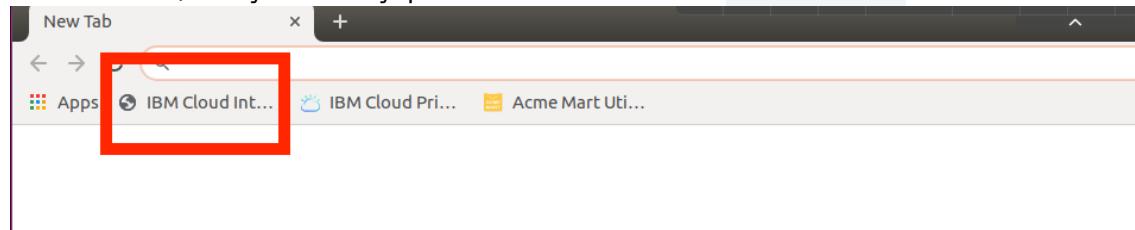
Unlock the power of your data and support the scale required for all of your integration and digital transformation initiatives.

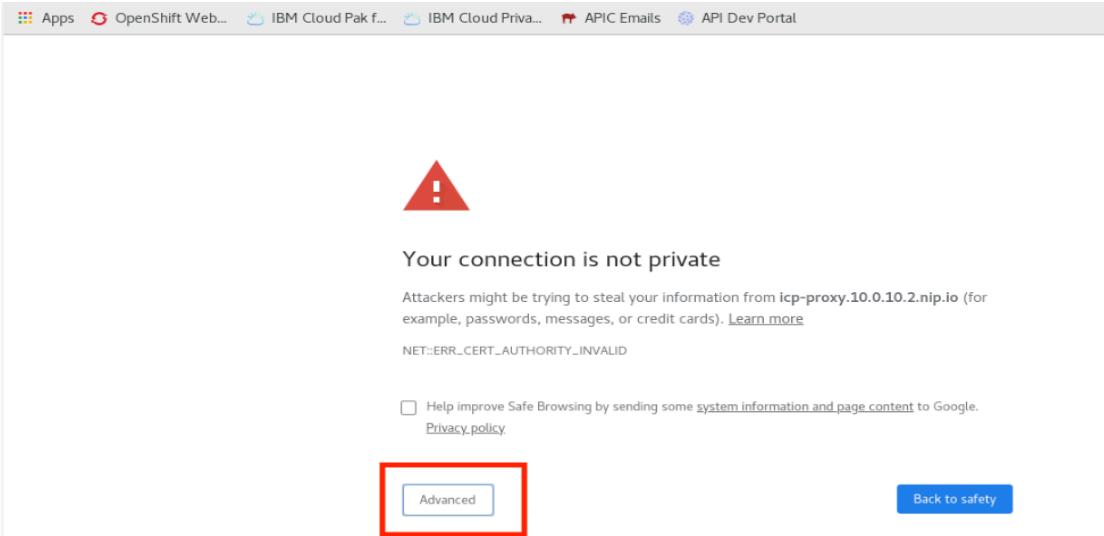
The home page of the **Cloud Pak for Integration** is referred to as the **Platform Navigator**. From the Platform Navigator you are able to navigate to all the integration and development technology contained within the platform.

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 Event Streams](#).

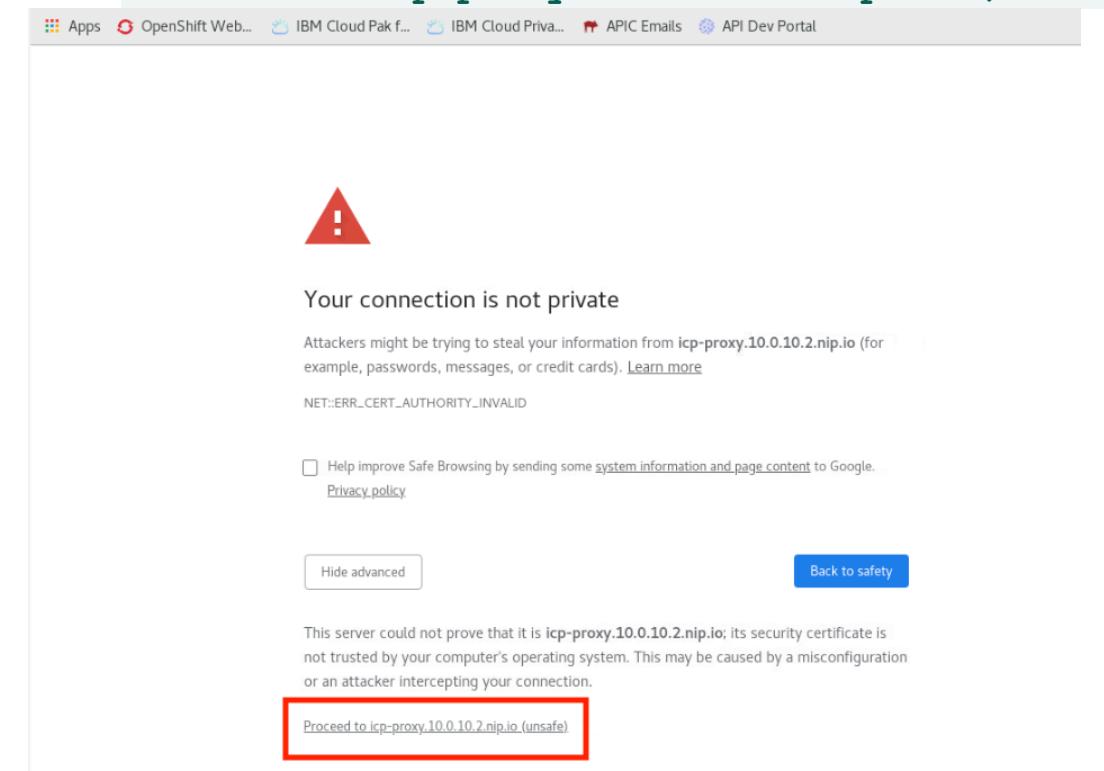
1. Click the **IBM Cloud Pak for Integration** bookmark in the bookmarks bar. If you see the login page, please continue to Task 2 (p. 9)

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 -- Install Event Streams

In this task, you'll learn how to install a new instance of Event Streams in IBM Cloud Pak for Integration.

1. Click on the bookmark for IBM Cloud Pak for Integration at the top of the page. Locate the Event Streams service and Click **Add new instance**.

The screenshot shows the IBM Cloud Pak for Integration Platform home page. It displays a grid of service icons and names. The 'Event Streams' service is located in the top right corner. Its icon is a blue circle with a white gear and a line. Below the icon, the name 'eventstreams' and the identifier 'es-1' are listed. To the right of the name is a small edit icon. Underneath the service name is a blue 'Add new instance' button. This button is highlighted with a red rectangular box. The other services visible in the grid are API Connect, App Connect, MQ, Aspera, and DataPower. Each service has its own icon, name, identifier, and an 'Add new instance' button below it.

2. Click **Continue**. Review the information provided about Event Streams on the overview page. Click **Configure** when you are ready to proceed.

The screenshot shows the overview page for the 'ibm-eventstreams-rhel-icp4i-prod V 1.3.1' Helm chart. At the top, there's a back arrow labeled 'View All'. Below it, the chart name 'ibm-eventstreams-rhel-icp4i-prod V 1.3.1' is displayed. There are two tabs: 'Overview' (which is selected) and 'Configuration'. The 'Overview' tab contains several sections:

- CERTIFIED IBM CLOUD PAK:** States 'IBM Event Streams based on Apache Kafka.' with a 'local-clients' link.
- CLOUD PAK VERSION:** Shows '1.3.1'.
- DETAILS & LINKS:** Includes 'Type: Certified IBM Cloud Pak', 'Published: October 22, 2019', 'App Version: 1.3.1', and links for 'Licenses', 'Release Notes', and 'Qualification'.
- SOURCE & TAR FILES:** A dropdown menu.

The main content area is titled 'IBM Event Streams' and describes the chart as a high-throughput, fault-tolerant, event streaming platform built on Apache Kafka. It mentions that the chart deploys Apache Kafka and supporting infrastructure like Apache ZooKeeper. Below this is the 'Introduction' section, which provides a brief overview of the chart's purpose and dependencies. The 'Chart Details' section lists what the Helm chart installs, including an Apache Kafka cluster, an Apache ZooKeeper ensemble, an administration user interface, and an administration server. At the bottom right of the main content area is a blue 'Configure' button, which is also highlighted with a red rectangular box.

3. The Helm chart creates a number of IBM Cloud Pak for Integration configuration objects that can be customized.

1. Helm release name, enter **eslab**
2. Target namespace, enter **eventstreams**
3. Check as Target cluster, **local-cluster**
4. Check the **License** agreement checkbox

⚠ Pod Security Warning Your ICP cluster is running all namespaces Unrestricted (ibm-anyuid-hostpath-psp) by default. This could pose a security risk.

Configuration

IBM Event Streams based on Apache Kafka. Edit these parameters for configuration.

Helm release name *
eslab 1

Target namespace *
eventstreams 2

Target Cluster *
local-cluster 3

License *
 I have read and agreed to the License agreement 4

4. Scroll down and expand the **Quick start** section.

1. Certificate Secret Name, enter **eslabTLS**
2. External hostname/IP address, enter **icp-proxy.10.0.10.2.nip.io**
3. Image Pull Secret, enter **deployer-dockercfg-7fkf1**

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.

Global install settings
Values that apply to all components of IBM Event Streams

Certificate Secret Name *
eslabTLS 1

Image pull secret *i
deployer-dockercfg-7fkf1 3

External access settings
Configuration for network connections to IBM Event Streams

External hostname/IP address *
icp-proxy.10.0.10.2.nip.io 2

5. Scroll down and expand the **All parameters** section.

1. Check the **Used in production** checkbox
2. Check the **Generate Certificate for Security** checkbox
3. Verify the Certificate Secret Name: **eslabTLS**.
4. Click **Install**

Global install settings
Values that apply to all components of IBM Event Streams

Used in production * 1

Generate Certificate for Security * 2

Certificate Secret Name * 3 Docker image registry *
eslabTLS docker-registry.default.svc:5000/eventstreams/

Image pull secret * Image pull policy *
deployer-dockercfg-7fkfl IfNotPresent

File system group ID Architecture *
Enter value amd64 platforms

4 Cancel Install

6. The installation process takes a few minutes (7-10 mins) to complete. Scroll to the top of the page. A notification will inform you that **Chart deployment is in progress**. To continue to the Event Streams interface, click **View all** to leave the configuration page.

IBM Cloud Pak for Integration CLUSTER mycluster

← View All

ibm-eventstreams-rhel-icp4i-prod V 1.3.1

Overview Configuration

⚠ Pod Security Warning Your ICP cluster is running all namespaces Unrestricted (ibm-anyuid-hostpath-psp) by default. This could pose a security risk.

ℹ Chart deployment is in progress and may take a few minutes. Go to the Helm Releases page to check on the status of your deployment.

Configuration
IBM Event Streams based on Apache Kafka. Edit these parameters for configuration.

Helm release name * eslab

Target namespace * eventstreams Target Cluster * local-cluster

7. Click **View all** to leave the configuration page and view the deployment. To verify the installation, go to IBM Cloud Private. Click the **Menu-->Workloads-->Helm Releases**.
8. You will see the Helm Releases. Type on search releases:
eslab and click on the **eslab** line.
9. Locate Pod services and check the Pod **Status:Running** and **Ready: 1/1**. The installation process takes a few minutes (7-10 mins) to complete.
Note: the Pod section will be similar but slightly different to the figure below

The screenshot shows the IBM Cloud Private interface with the 'mycluster' cluster selected. In the 'Helm Releases' section, a single entry for 'eslab' is listed. Below it, the 'Pod' section displays a table with one row. The table has columns: Name, READY, Status, RESTARTS, and Age. The 'READY' column for the pod 'orders1-ib-1352-0' contains the value '1/1', which is highlighted with a red box. The 'Status' column shows 'Running'. The 'RESTARTS' column shows the value '4'. The 'Age' column shows '5m36s'. The 'Role' section below shows a single entry for 'orders1-ibm-ace-server-icp4i-prod-role' with an age of '5m36s'.

Name	READY	Status	RESTARTS	Age
orders1-ib-1352-0	1/1	Running	4	5m36s

10. **Return** to the Cloud Pak for Integration browser. Click on **eslab** within the Cloud Pak for Integration interface to launch the interface in a new browser tab.

Note: If a pop-up window appears with the message below: eslab did not load correctly. Click **Open eslab**.



eslab did not load correctly.

This could be due to an issue signing in, a self-signed certificate being encountered, or a problem with eslab itself.

It may be possible to resolve the issue by opening eslab in a separate window and resolving any sign-in or certificate issues, then retrying.

[Open eslab](#)

[Try again](#)

Note: In the second popup-window click **Advanced**. Scroll down and click **Proceed to icp-proxy.10.0.10.2.nip.io (unsafe)** link.



Your connection is not private

Attackers might be trying to steal your information from **icp-proxy.10.0.10.2.nip.io** (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR_CERT_AUTHORITY_INVALID

Help improve Safe Browsing by sending some [system information and page content](#) to Google.
[Privacy policy](#)

[Advanced](#)

[Back to safety](#)

11. Use the **System is healthy** box to verify the health of each Event Streams component.

The screenshot shows the 'Getting started' page of the IBM Event Streams interface. At the top, there's a navigation bar with links for 'Getting started', 'Topics', 'Consumer groups', 'Schemas', 'Monitoring', and 'Toolbox'. To the right of the navigation is a button 'Connect to this cluster' with a gear icon. Below the navigation, a main heading says 'Welcome to IBM Event Streams, let's get you up and running...'. There are two main sections: 'Use a simulated topic' (with an icon of a tap and gears) 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' (icon of a cluster), 'Schema basics' (icon of a document with a gear), and 'Kafka Connect basics' (icon of a flow diagram). At the bottom right, a red-bordered box displays a green checkmark and the text 'System is healthy'.

Congratulations, you've installed IBM Event Streams on IBM Cloud Pak for Integration. In the next section, you'll learn how to manage topics, the core of Event Streams functionality. While you're at the Getting started page, take a moment to review Kafka Basics for in-product education.

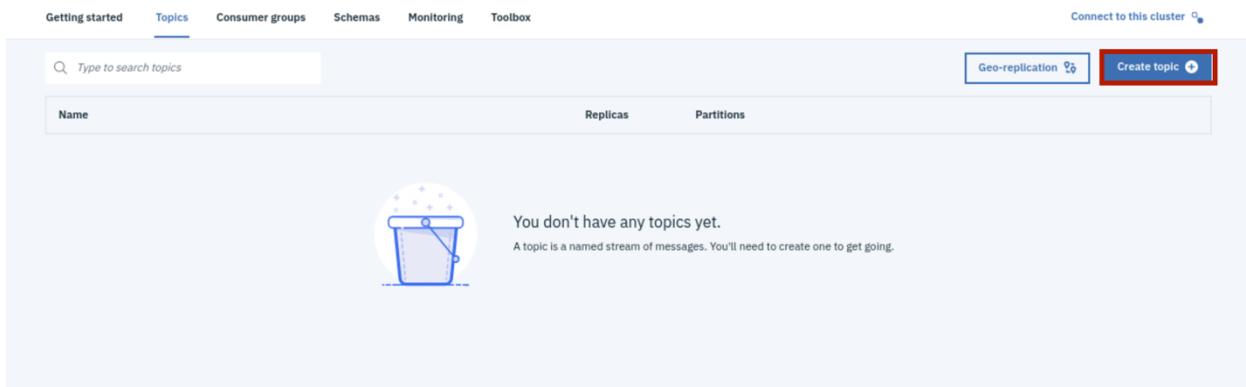
Task 3 -- Managing Event Streams Topics

Event Streams applications write to and read from topics, which are known groupings of related data. Applications connect to topics.

Topics are created and configured once only by the Event Streams administrator

1. You'll be guided through creating your topic. In the Event Streams interface, click **Topics**. Click on the **Create topic** button. At any time, use the **Advanced** toggle to view the full range of configuration options. In this tutorial, we'll only edit the core set.

eslab



2. The rest of this lab is preconfigured to connect to a specific topic. Enter **eslabtopic** as the topic name. Click **Next**.

A screenshot of a 'Create topic' dialog box. At the top left is the title 'Create topic'. Below it is an 'Advanced' toggle switch. To the right of the switch is a progress bar with four circles, the first of which is filled blue. The next section is labeled 'Topic name' with an input field containing the text 'eslabtopic'. To the right of the input field is a descriptive text: 'This is the unique name used to recognize your topic. It will also be used by your producers and consumers as part of the connection information, so make it something easy to recognize.' At the bottom right of the dialog are two buttons: 'Cancel' and 'Next', with 'Next' being highlighted with a red box.

3. Create 3 partitions. Click **Next**.

Create topic

Advanced

Partitions

3

1 partition is sufficient for getting started, but production systems often have more.

One or more partitions make up a topic. A partition is an ordered list of messages.

Partitions are distributed across the brokers in order to increase the scalability of your topic. You can also use them to distribute messages across the members of a consumer group.

Back **Next**

4. Define the message retention – 4 hours. Click **Next**

Create topic

Advanced

Message Retention

A day
 A week
 A month
 4

This is how long messages are retained before they are deleted.

If your messages are not read by a consumer within this time, they will be missed.

Back **Next**

5. Define the replicas of your topic. Select the default setting of

1. **Replication factor: 3**
2. **Minimum in-sync replicas: 2**
3. Click **Create topic**

The screenshot shows the 'Create topic' interface. At the top, there's a navigation bar with 'IBM Cloud Pak for Integration', 'Event Streams', and 'eventstreams | eslab'. Below it, a back arrow and 'Topics' are shown. The main title is 'Create topic'. There's an 'Advanced' toggle switch followed by a progress bar with four steps, all of which are checked. The first step is 'Replicas'. Under 'Replicas', two options are listed: 'Replication factor: 1 Minimum in-sync replicas: 1' (radio button is not selected) and 'Replication factor: 3 Minimum in-sync replicas: 2' (radio button is selected). To the right of this section, a tooltip explains: 'This is how many copies of a topic will be made for high availability. The partitions of each topic can be replicated across a configurable number of brokers.' Below 'Replicas', there's a 'Replication factor' dropdown set to '3'. Under 'Minimum in-sync replicas', there's a dropdown set to '2'. At the bottom right are 'Back' and 'Create topic' buttons, with 'Create topic' being highlighted with a red border.

You will be returned to the Topics page. Your new topic is now displayed along with a completion notification. You are ready to connect the starter application to Event Streams.

Task 4 -- Use a Starter application to send and receive data

The final steps generate and execute a starter application to represent the producing applications and consuming applications you might connect to a topic. Messages sent by the producer are viewed on the topic section in the Event Streams interface.

1. Event Streams includes several tools that may be used to test out the Event Streams installation as well as help with the development of Event Streams based applications. Click **Toolbox** to access these tools. When you are ready to continue, click on **Generate application** in the **Starter application** box.

The screenshot shows the 'eslab' workspace in the IBM Cloud Pak for Integration interface. The 'Toolbox' tab is selected. In the 'Applications' section, there are two main cards:

- Generate starter application**: This card features a Kafka icon and a brief description: "Download and install this application which you can connect to a topic. This application is a full project containing a Kafka producer and consumer for you to try out." It includes a blue "Generate application" button and a link "Just looking for the instructions?".
- Workload generation application**: This card features a Kafka icon and a brief description: "Want to quickly validate your IBM Event Streams instance? Download and run a pre-built tool to generate messages at a configurable rate, or download and modify the source to simulate your own workload." It includes a blue "View in GitHub" button.

In the 'Tools' section, there are two more cards:

- IBM Event Streams command-line interface**: This card features a terminal icon and a brief description: "Use the CLI to manage your IBM Event Streams instance from the command line." It includes a blue "Find out more" button.
- Apache Kafka Java client**: This card features a Kafka icon and a brief description: "Download the Apache Kafka Java client." It includes a blue "Find out more" button.

2. In the Application name, enter **eslabtester** (1). In the existing topic, select the **eslabtopic** topic you created earlier (2). Click **Generate**.

Generate starter application

Application name i

1

What do you want this application to be able to do?

Produce messages

Consume messages

Which topic shall we connect with?

You can choose to create a new topic, or you can choose an existing topic to use in this application

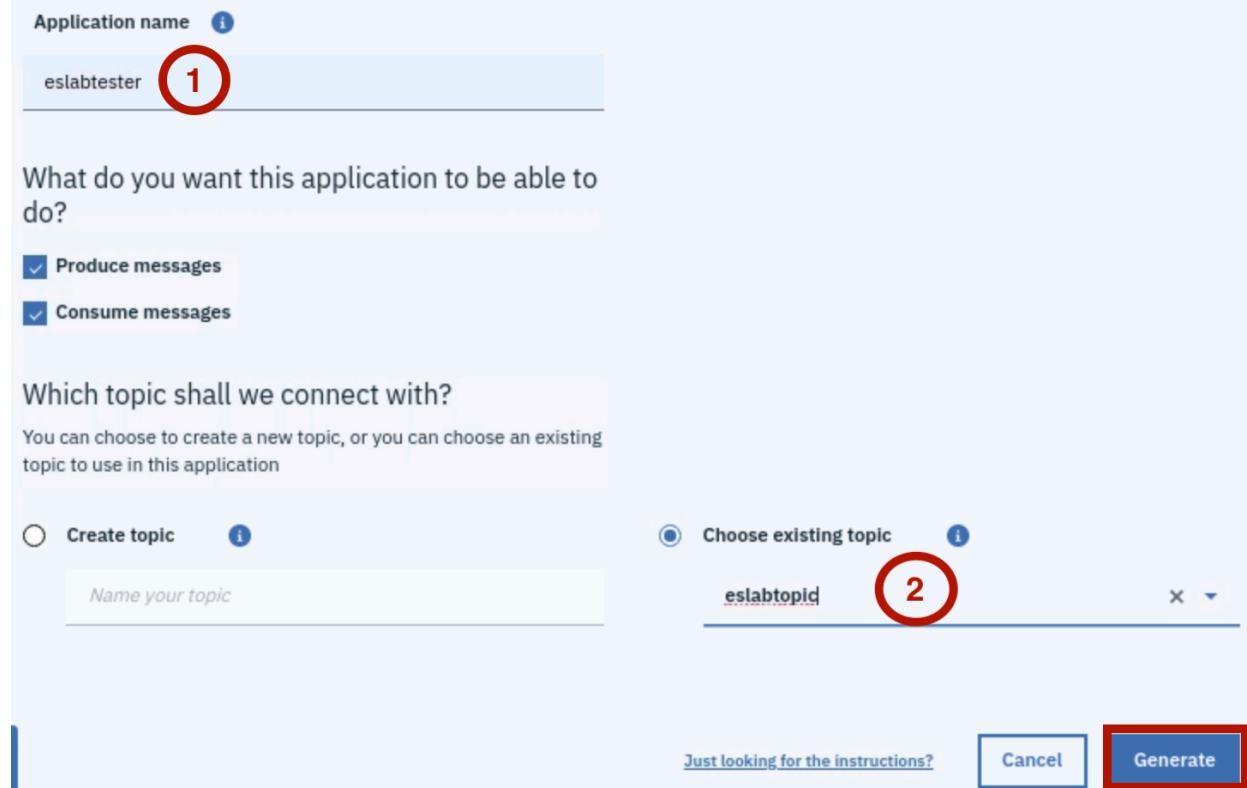
Create topic i

Choose existing topic i

2

[Just looking for the instructions?](#)

Cancel Generate



3. Once you see the message **The starter application has been generated** follow the guidance on the page.

Generate starter application

 The starter application has been generated

1. Download the starter application
Download the compressed file and extract the contents to your preferred location.
[Download](#) 
2. Navigate to the extracted file and run this command to build and deploy the application
Java version 8 and Maven are prerequisites to building and running the starter application
[Maven download repository](#)
 

Note: Detailed instructions on how to build and run the starter application can be found in the README.md file in the top level project directory.
3. Access the successfully deployed sample application using the following URL:
<http://localhost:9080>

 [Close and go to topics](#)

4. Download the Starter application to the Downloads directory.
Double click to **Download**

- a. Open a terminal window. Navigate to Downloads with the command: **cd /home/ibmuser/Downloads**
- b. Unzip the download with the command:
unzip IBMEventStreams_eslabtester.zip.
- c. If prompted to replace files, answer **A** for all.

- d. Use the pre-installed Maven build utility to build the sample application by running the command: **export**

JAVA_OPTIONS=Djdk.net.URLClassPath.disableClassPathURLCheck=true

```
[ibmuser@developer Downloads]$ export JAVA_OPTIONS=Djdk.net.URLClassPath.disableClassPathURLCheck=true
```

- e. Use the supplied maven command to run the starter application
mvn install liberty:run-server

- f. Once **The default Server server is ready to run a smarter planet** appears in the Command prompt window, the application is running.

```
File Edit View Search Terminal Help
[INFO] [AUDIT    ] CWWKZ0058I: Monitoring dropins for applications.
[INFO] [WARNING ] SRVE9967W: The manifest class path jaxb-api.jar can not be found in jar file
file:/home/ibmuser/Downloads/target/liberty/wlp/usr/servers/defaultServer/apps/expanded/eslabte
ster.war/WEB-INF/lib/jaxb-core-2.2.11.jar or its parent.
[INFO] [WARNING ] SRVE9967W: The manifest class path jaxb-core.jar can not be found in jar file
file:/home/ibmuser/Downloads/target/liberty/wlp/usr/servers/defaultServer/apps/expanded/eslabt
ester.war/WEB-INF/lib/jaxb-impl-2.2.11.jar or its parent.
[INFO] [AUDIT    ] CWWKT0016I: Web application available (default_host): http://developer:9080/m
etrics/
[INFO] [AUDIT    ] CWWKT0016I: Web application available (default_host): http://developer:9080/j
wt/
[INFO] [AUDIT    ] CWWKT0016I: Web application available (default_host): http://developer:9080/i
bm/api/
[INFO] [AUDIT    ] CWWKT0016I: Web application available (default_host): http://developer:9080/h
ealth/
[INFO] [WARNING ] SRVE9967W: The manifest class path jaxb-api.jar can not be found in jar file
file:/home/ibmuser/Downloads/target/liberty/wlp/usr/servers/defaultServer/apps/expanded/eslabte
ster.war/WEB-INF/lib/jaxb-core-2.2.11.jar or its parent.
[INFO] [WARNING ] SRVE9967W: The manifest class path jaxb-core.jar can not be found in jar file
file:/home/ibmuser/Downloads/target/liberty/wlp/usr/servers/defaultServer/apps/expanded/eslabt
ester.war/WEB-INF/lib/jaxb-impl-2.2.11.jar or its parent.
[INFO] [AUDIT    ] CWWKT0016I: Web application available (default_host): http://developer:9080/
[INFO] [AUDIT    ] CWWKZ0001I: Application eslabtester started in 1.782 seconds.
[INFO] [AUDIT    ] CWWKF0012I: The server installed the following features: [appSecurity-2.0, cd
i-1.2, concurrent-1.0, distributedMap-1.0, jaxrs-2.0, jaxrsClient-2.0, jndi-1.0, json-1.0, json
p-1.0, jwt-1.0, microProfile-1.2, mpConfig-1.1, mpFaultTolerance-1.0, mpHealth-1.0, mpJwt-1.0,
mpMetrics-1.0, servlet-3.1, ssl-1.0, websocket-1.1].
[INFO] [AUDIT    ] CWWKF0011I: The defaultServer server is ready to run a smarter planet. The de
faultServer server started in 4.817 seconds.
```

- g. Follow the URL <http://localhost:9080/eslabtester> to the Starter application's interface.

The screenshot shows a split-screen interface for the IBM Event Streams Starter Application. The left panel, titled 'Starter Application', displays a counter of '00' messages produced on topic 'eslabtopic'. It includes a 'Custom payload string (optional)' input field with 'Add custom payload (Hello World)' and a 'play' button. The right panel, also titled 'Starter Application', shows a counter of '00' messages consumed on topic 'eslabtopic'. It has tabs for 'Most recent messages', 'Partition', and 'Offset', with a note 'No messages consumed yet..'. A 'Stop listening for messages' button is at the top right of the right panel.

2. This page represents the producing application on the left of the screen, and the consuming application on the right. Type a message in the **Custom payload string** field (1). Click **play** on the producer, and notice messages appearing in the consuming application (2) as they are consumed from the Event Streams topic.

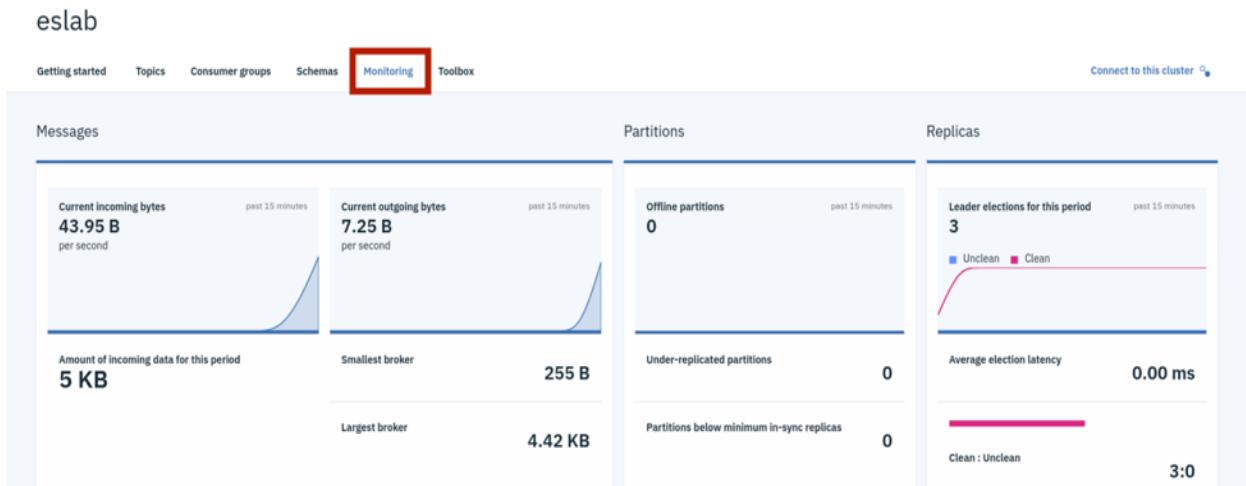
This screenshot shows the same split-screen interface after sending 10 messages. The left panel now shows '10' messages produced. The right panel shows '10' messages consumed. A red circle labeled '1' is over the 'play' button in the producer's message input field. A red circle labeled '2' is over the 'Start listening for messages' button in the consumer's header. The consumer panel also shows a table of 10 messages with columns for Message, Partition, Offset, and Consumed timestamp.

Message	Partition	Offset	Consumed at
Message 10	0	3	Consumed at 5:40:54 PM
Message 9	1	2	Consumed at 5:40:52 PM
Message 8	2	2	Consumed at 5:40:50 PM
Message 7	0	2	Consumed at 5:40:48 PM
Message 6	1	1	Consumed at 5:40:46 PM
Message 5	2	1	Consumed at 5:40:44 PM

3. Use the Event Streams interface to select the **eslabtopic** topic and see the messages it has received from the Starter application. Click **Messages**, select a message from the table, and use the message browser to step through each **Offset**.

Indexed timestamp	Partition	Offset
10/9/2019, 5:40:38 PM	2	0
10/9/2019, 5:40:35 PM	0	0
10/9/2019, 5:40:40 PM	1	0
10/9/2019, 5:40:42 PM	0	1
10/9/2019, 5:40:44 PM	2	1
10/9/2019, 5:40:46 PM	1	1

4. Return to the list of topics and click the Monitoring tab to view the rate of incoming and outgoing data.



Summary

Congratulations! You've set up a topic and connected an application to test the flow of data through Event Streams.