Getting Started with IBM Event Streams

Building an event-driven architecture with IBM Event Steams allows organizations to transition from traditional monolith systems and silos to more modern microservices and event streaming applications that increase their agility and accelerate their time to innovation.
IBM Event Streams builds on top of open-source Apache Kafka ® to offer enterprise-grade event streaming capabilities. The following features are included as part of IBM Event Streams:

- Identity and Access Management (IAM), fine-grain security controls to manage the access that you want to grant each user for Kafka clusters, Topics, Consumer Groups, Producers and more.
- Geo-replication to deploy multiple instances of Event Streams in different locations and then synchronize data between your clusters to improve service availability.
- Visually driven management and monitoring experience with the Event Streams dashboard that displays metrics collected from the cluster, Kafka brokers, messages, consumers, and producers to provide health check information and options to resolve issues.
- Encrypted communication between internal components and encrypted storage.

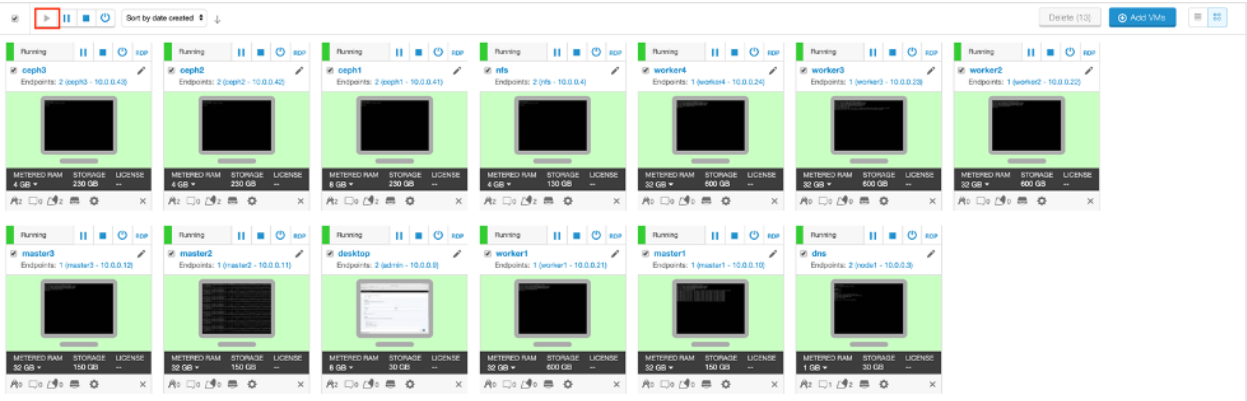
In this tutorial, you will explore the following key capabilities:

- Prepare IBM Cloud Pak for Integration 2020.1.1 running on Red Hat OpenShift 4.3
- Install an IBM Event Streams 2019.4.2 instance in IBM Cloud Pak
- Create and manage Event Streams topics
- Use a Starter application to send and receive data

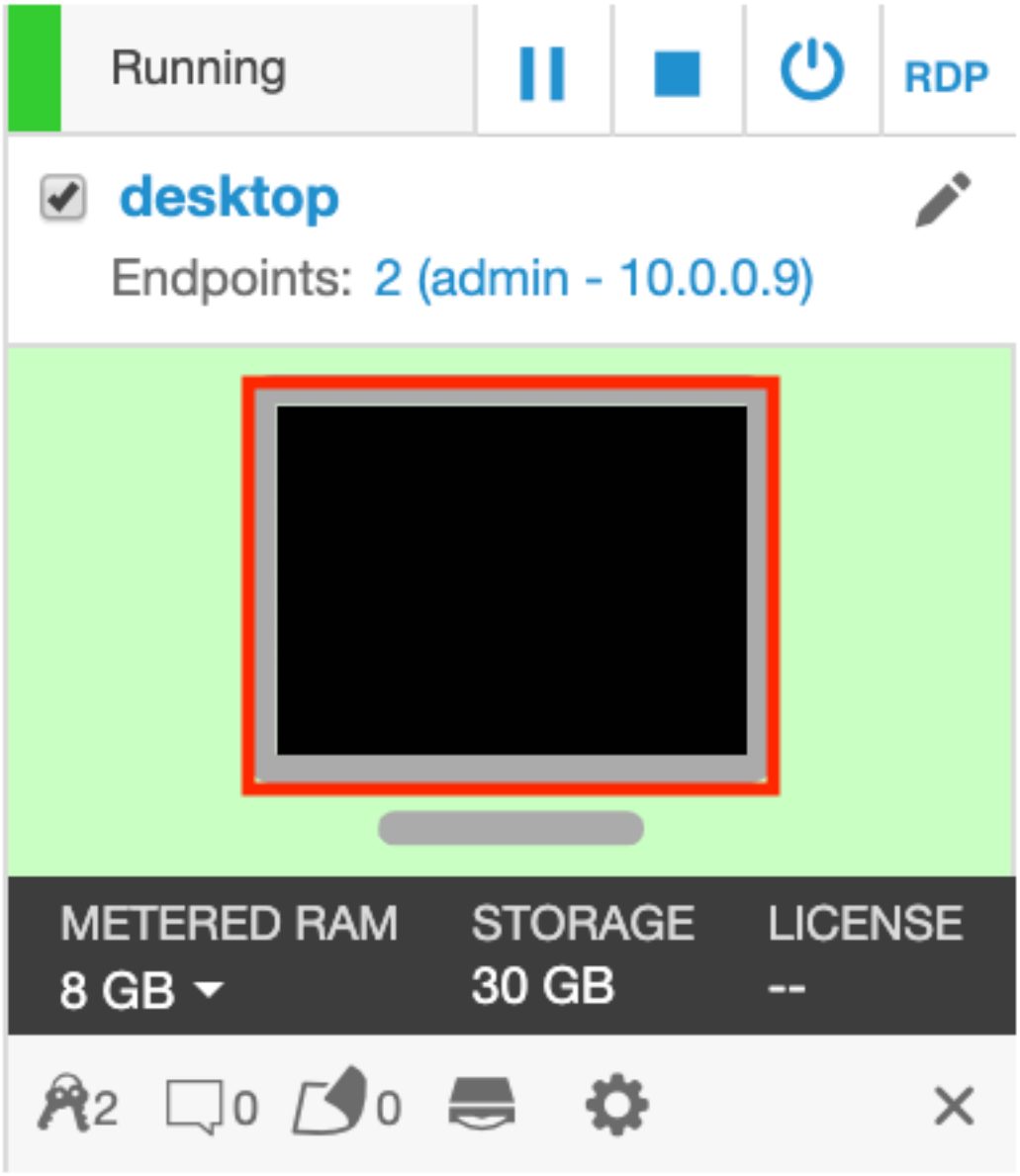
Task 1 - Preparing IBM Cloud Pak for Integration
Set up your IBM Cloud Pak for Integration environment before installing Event Streams
The demo runs on a virtual machine that is provided by IBM Demos.
To get access, click **reserve** link to create an instance in the **Before you**

start section on this tutorial page. You'll 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 2020.1.1 cluster on Red Hat OpenShift 4.3.

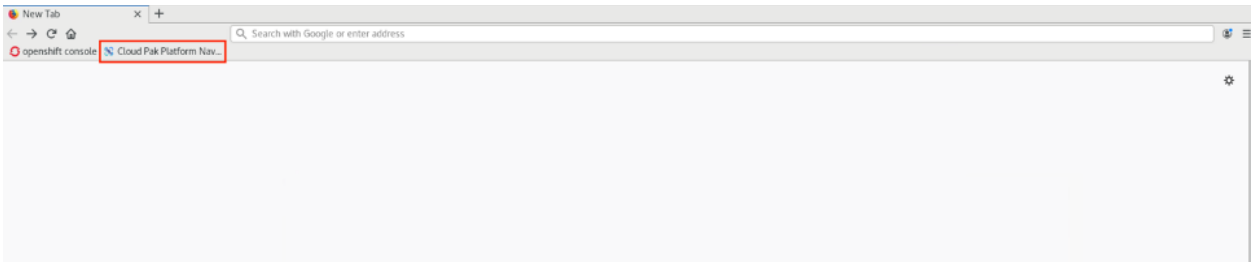
1. If needed, click the **run button** to start the virtual machines.



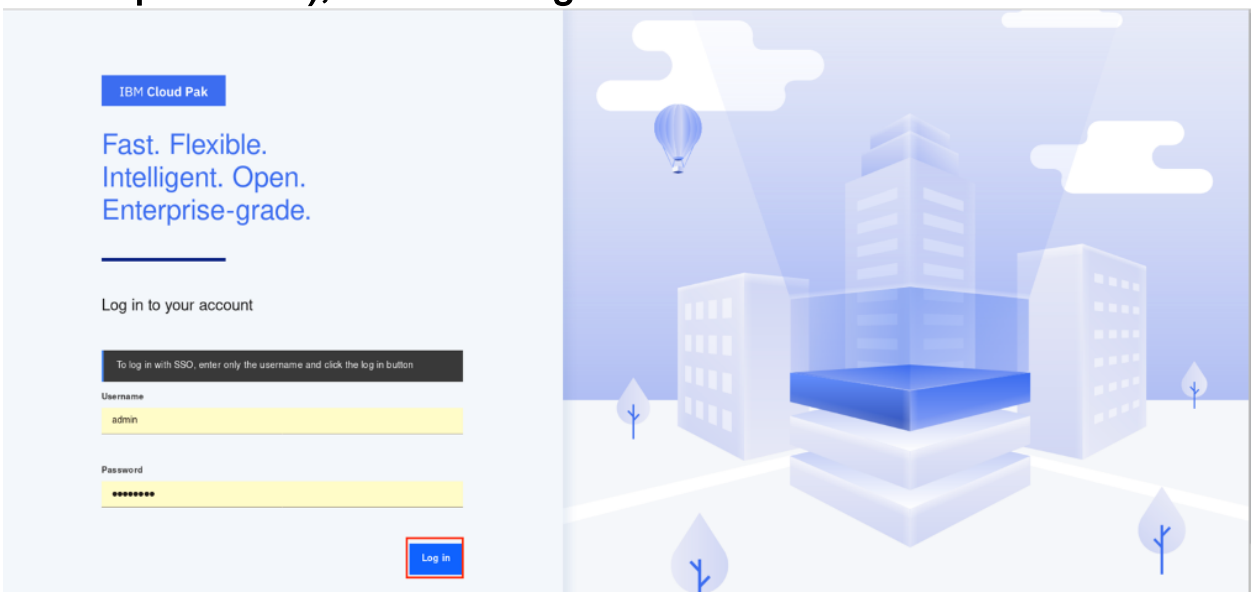
2. Once the virtual machines display **Running** as their status, click the **desktop node** image to launch the lab machine.



3. Log in to the Linux desktop with User ID **ibmuser** and Password **engageibm .**
4. Open a browser and navigate to IBM Cloud Pak for Integration by using the bookmark toolbar and select **Cloud Pak Platform Navigator**. (Preferred browser: Firefox)



5. Log in to IBM Cloud Pak if a login screen is presented.You may not need enter the username and password (username **admin** and Password **passw0rd),** and click **Log in.**

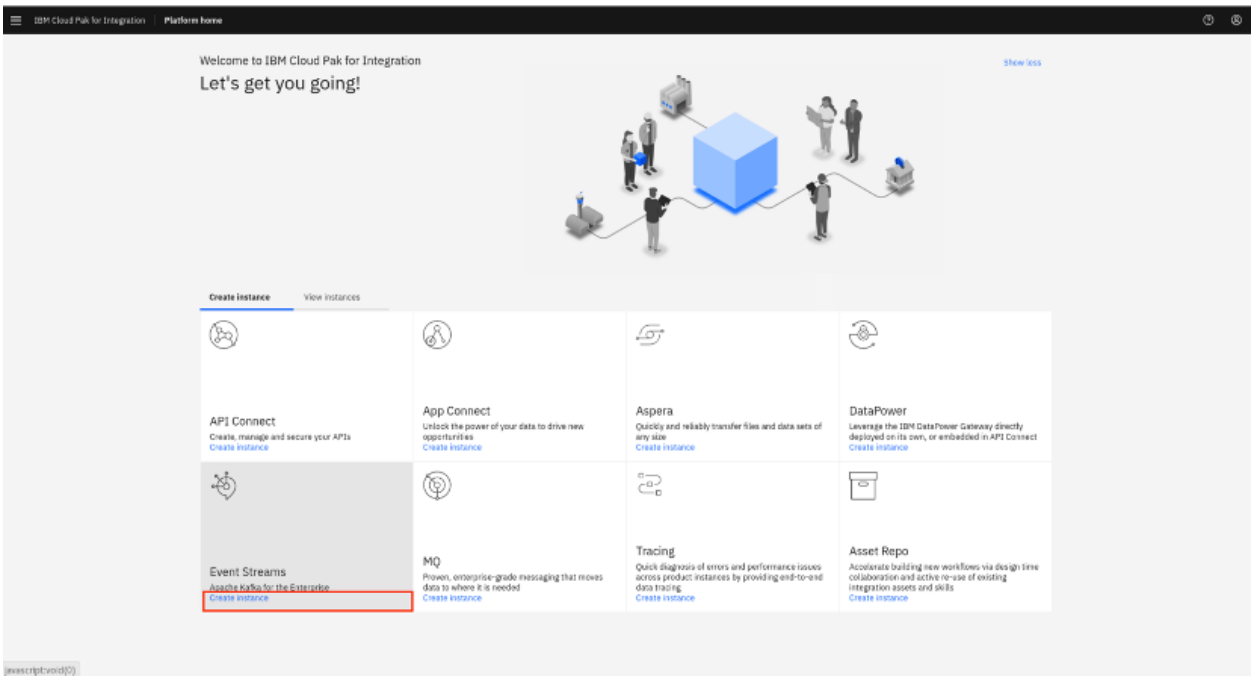


Task 2 - Installing an Event Streams instance

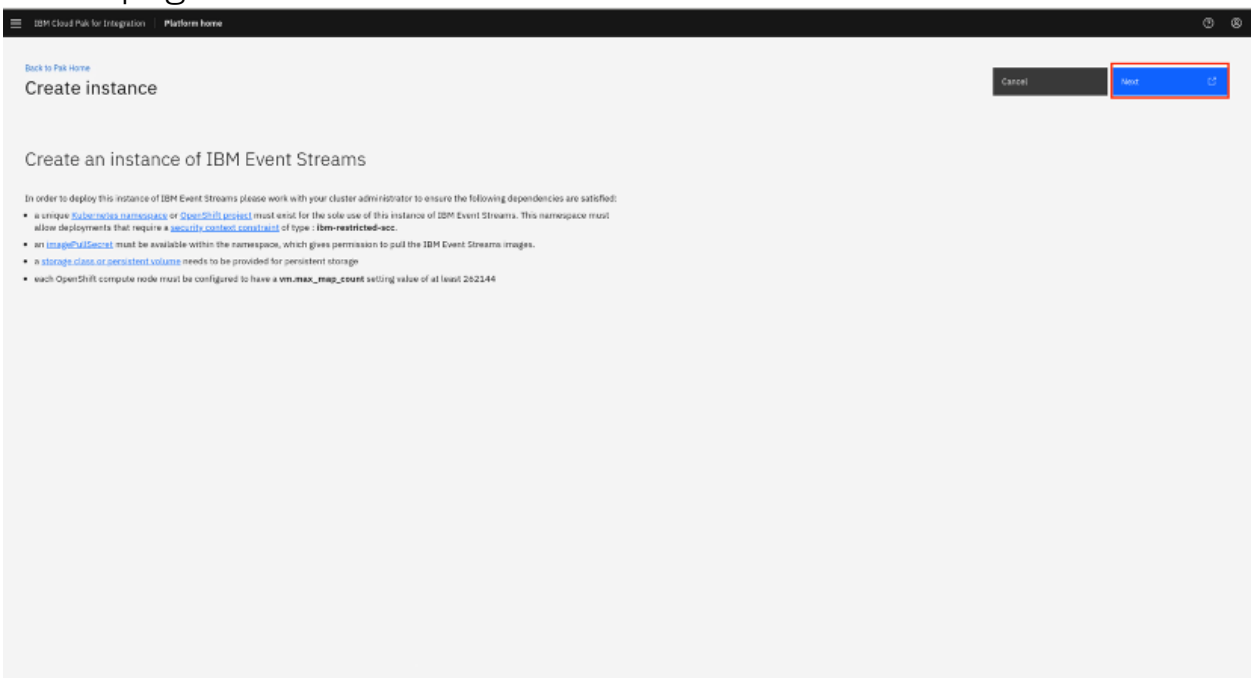
IBM Cloud Pak for Integration offers a single, unified platform for all your enterprise integration needs. It deploys integration capabilities into the Red Hat OpenShift managed container environment and uses the monitoring, logging, and security systems of OpenShift to ensure consistency across all integration solutions

Install a new instance of Event Streams in IBM Cloud Pak for Integration.

1. Using the virtual machine, click the bookmark for IBM Cloud Pak for Integration. Locate Event Streams and click **Create instance.**

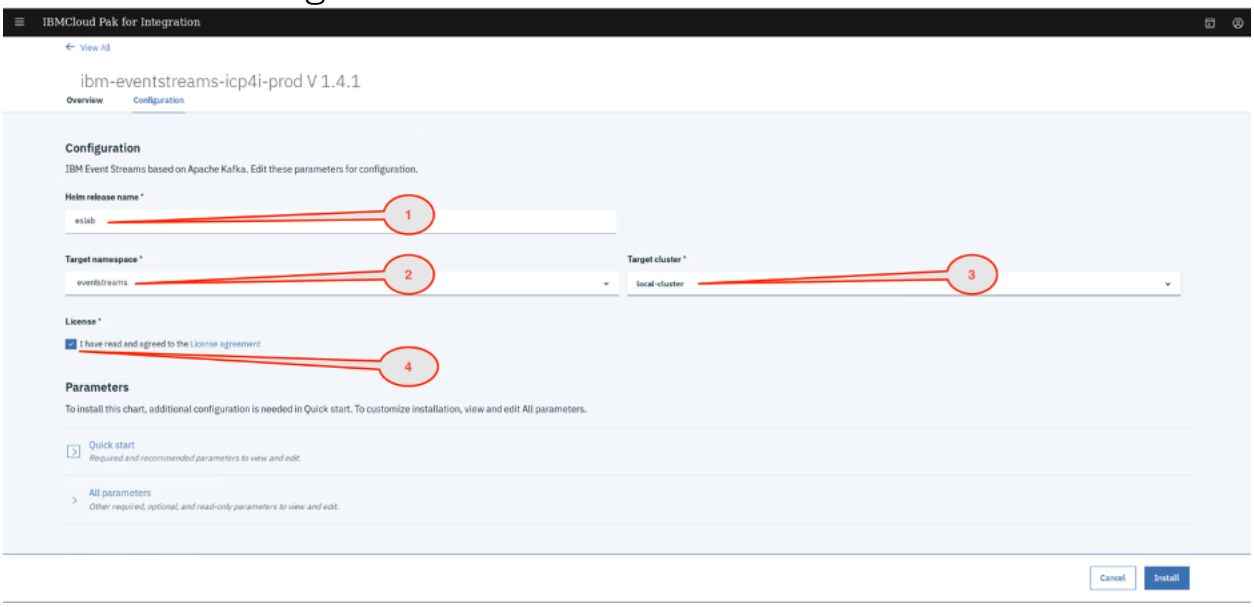


2. Click **Next** .Review the information provided about Event Streams on the overview page.

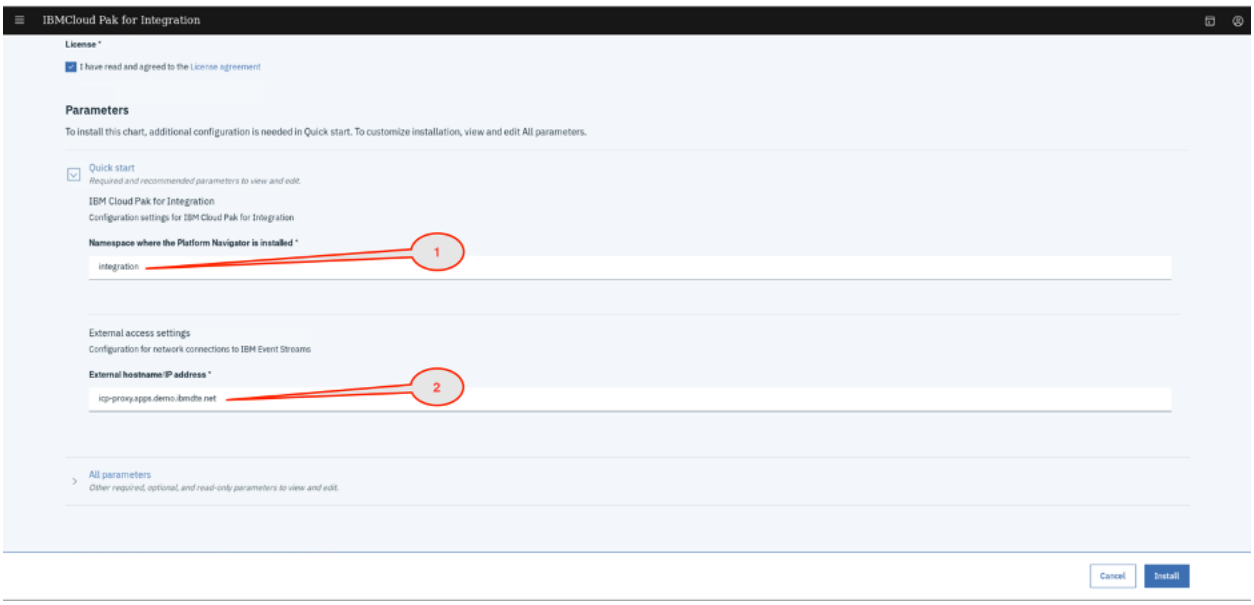


3. Configure the Event Streams chart as follows. The helm chart creates a number of IBM Cloud Pak for Integration configuration objects that can be customized.

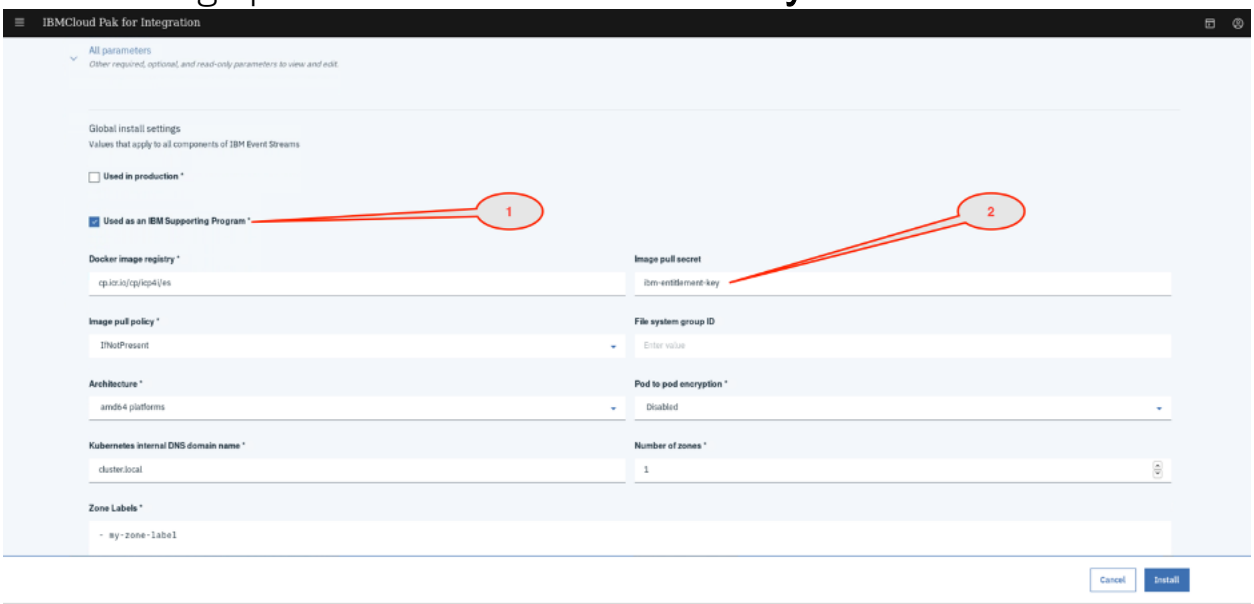
1. In the Helm release name, enter **eslab**.
2. As the Target namespace, enter **eventstreams**.
3. Select **local-cluster** as the Target cluster
4. Select the License agreement checkbox.



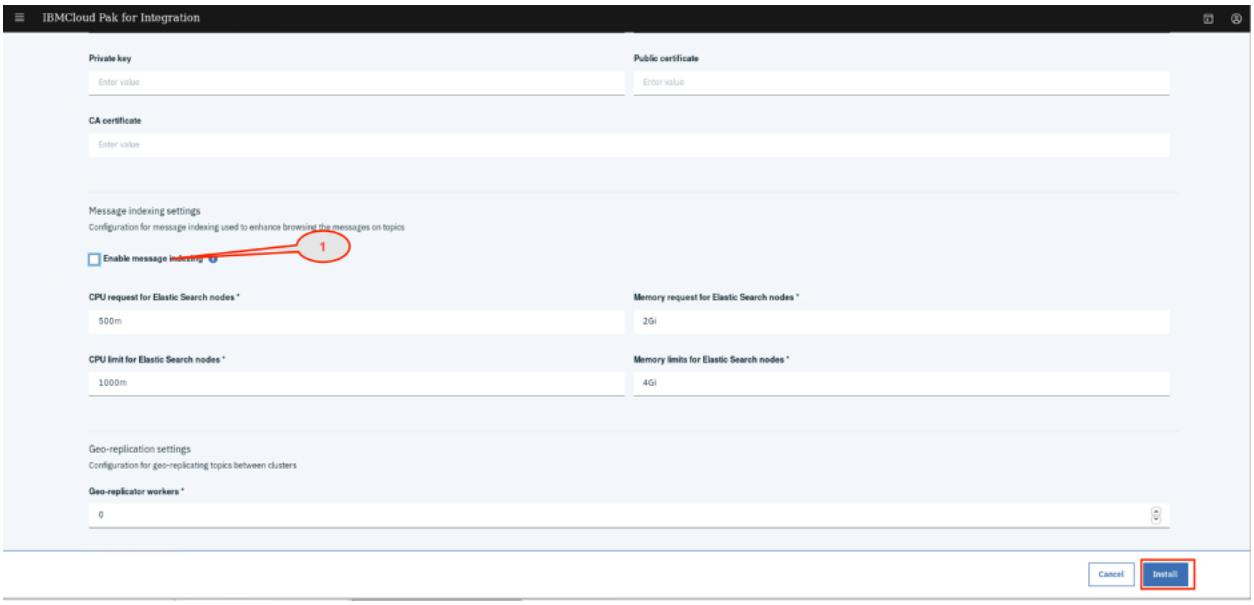
4. Expand the **Quick start** section.
1. Enter **integration** as the Namespace where the Platform Navigator is installed.
2. Enter the External hostname/IP address: **icp-proxy.apps.demo.ibmdte.net.**



5. Expand the **All parameters** section.
1. Check that the **Used as an IBM Supporting Program checkbox.**
2. Enter the Image pull secret: **ibm-entitlement-key**

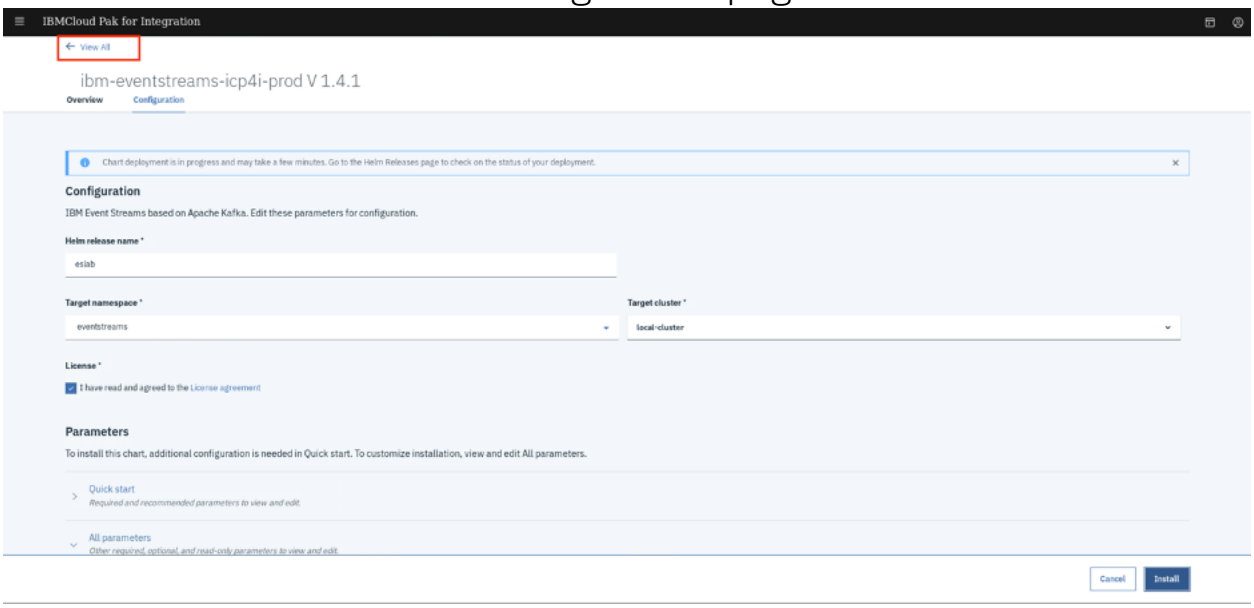


6. For this lab, scroll and Uncheck **Enable message Indexing** and then click **Install .**
   The installation process takes a few minutes to complete. A notification at the top of the page informs you that **Chart deployment is in progress**.
   Note: For this lab, Keep as default the rest of parameters.

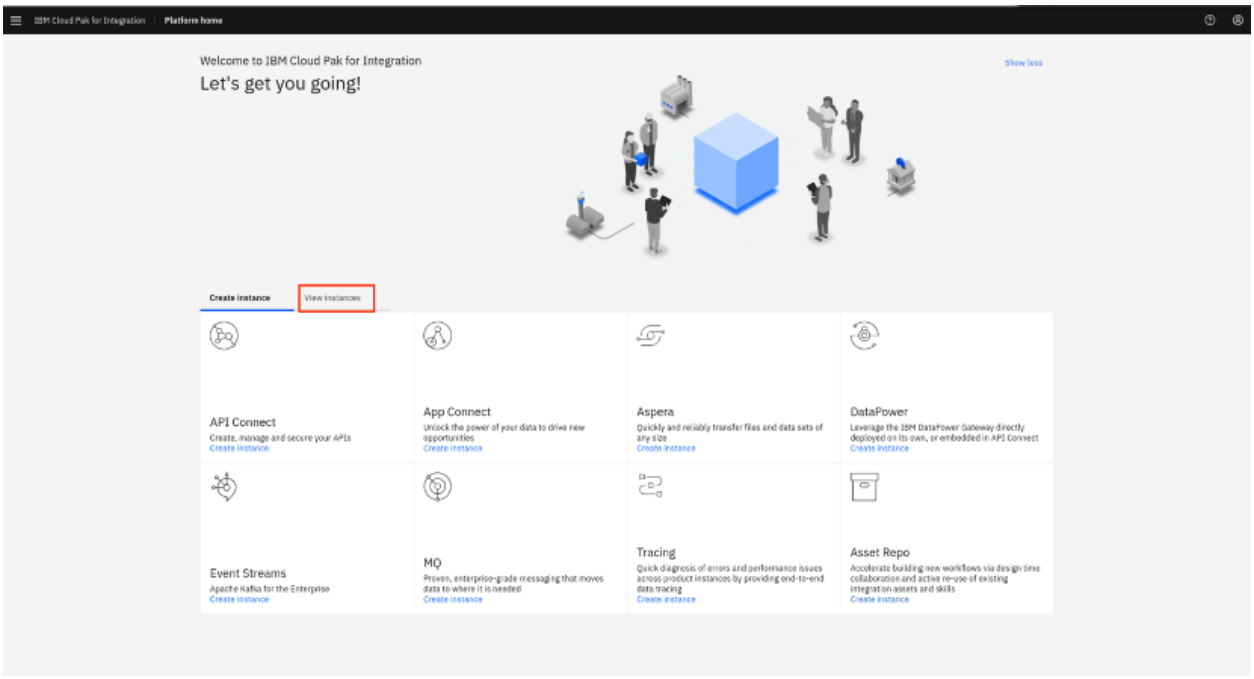


The installation process takes a few minutes to complete. A notification at the top of the page informs you that **Chart deployment is in progress.**

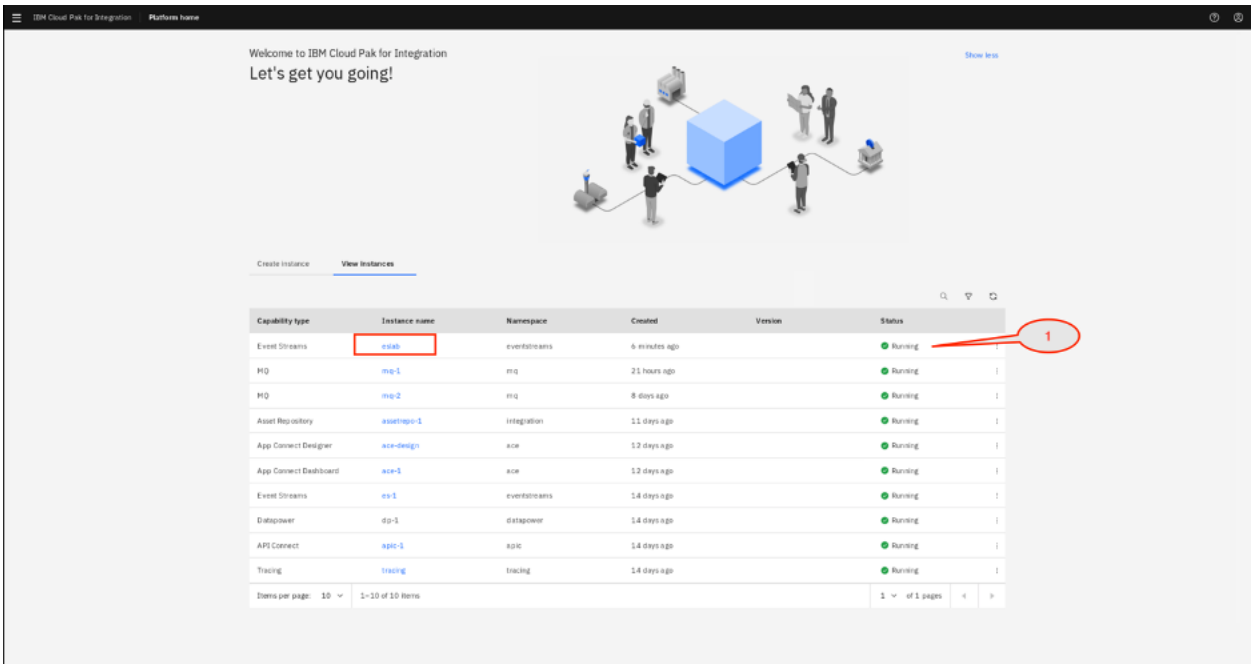
7. To continue to the Event Streams interface, scroll to the top of the page and click **View all** to leave the configuration page.



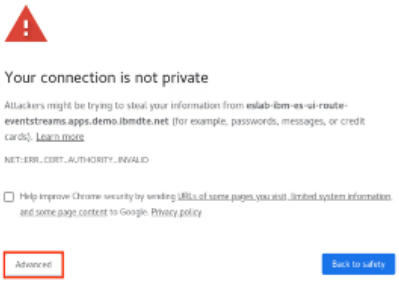
8. Click **View Instances.** All installed instances are displayed.



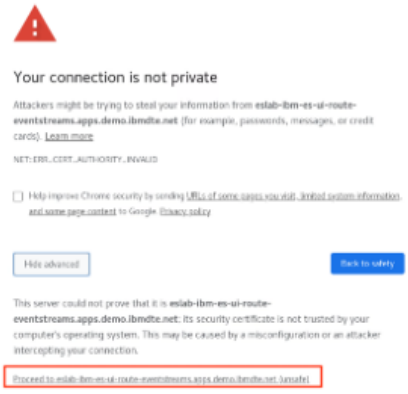
9. Verify if Event Streams instance **eslab** is running. Click instance name: **eslab.**



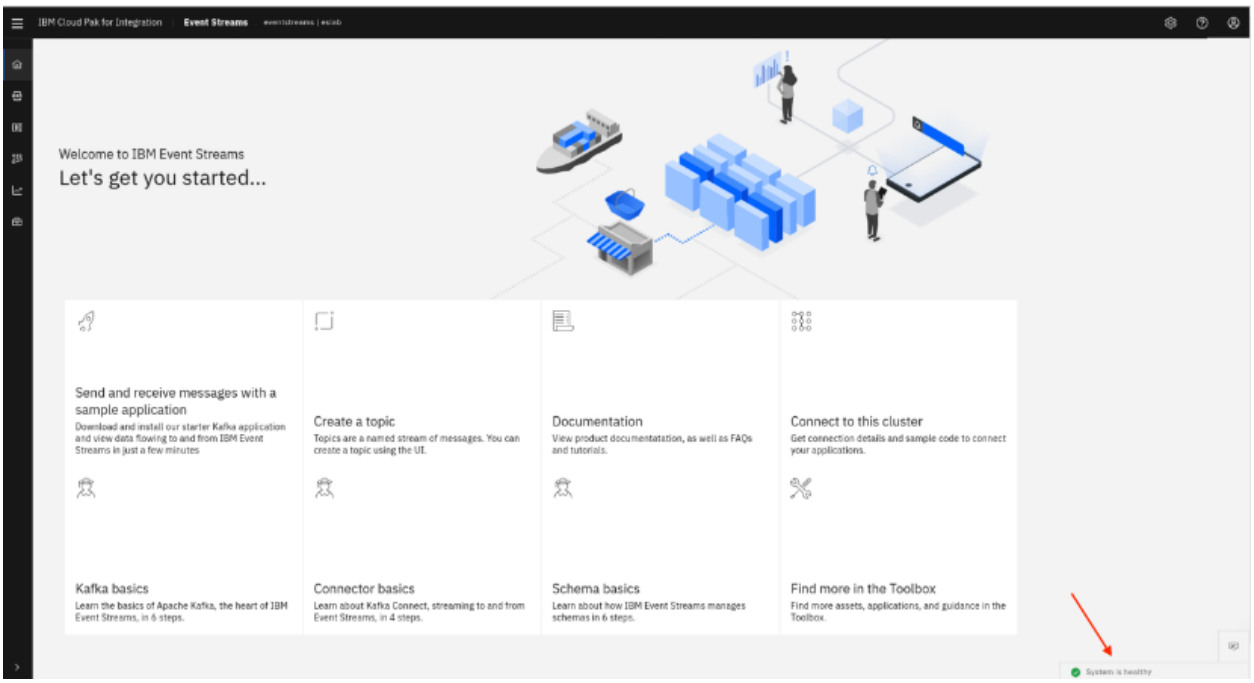
Note : You might receive a pop-up window appears with the message **Your connection is not private**, click **Advanced .**

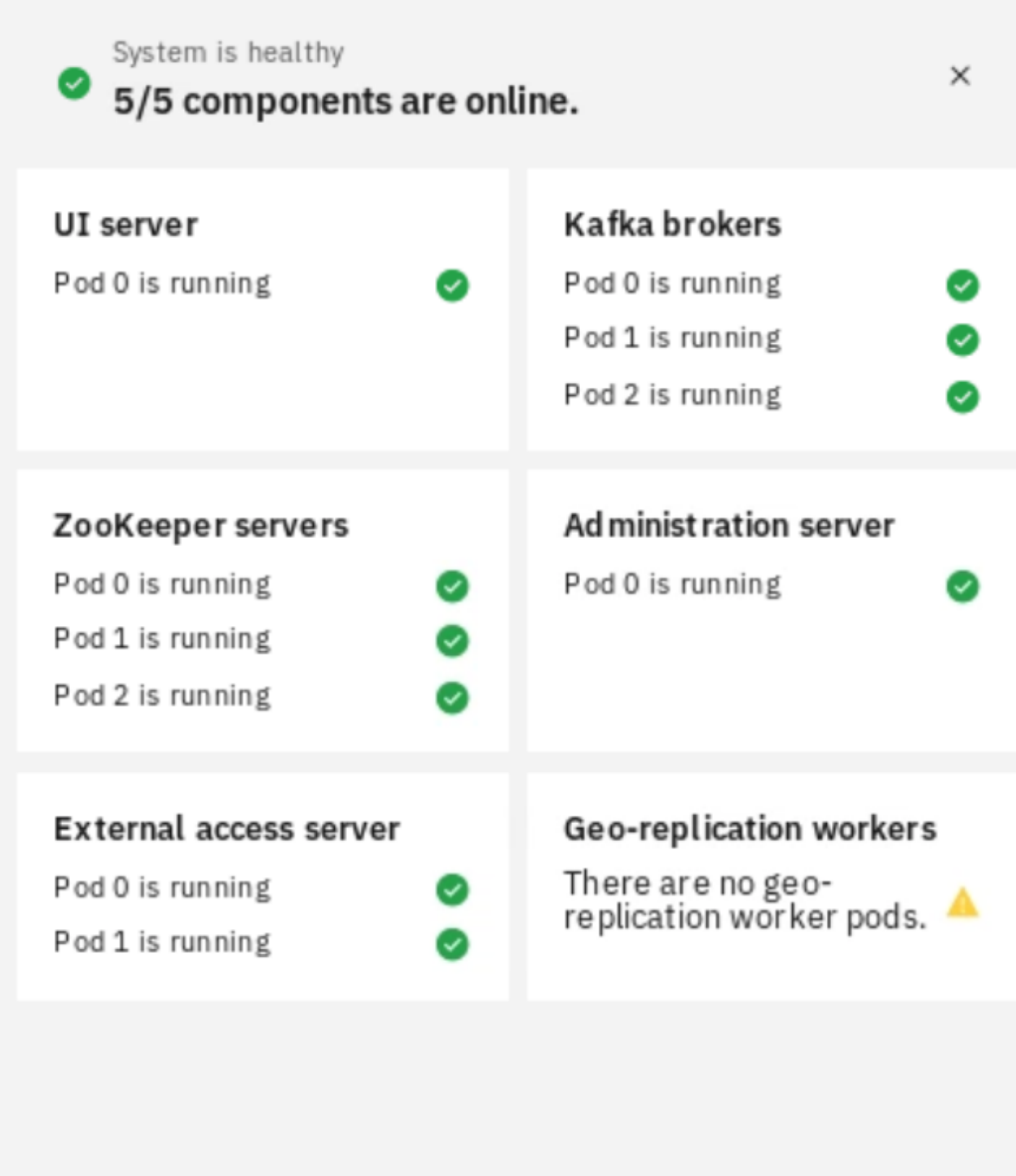


In the second.popup-window. Scroll down and click **Proceed to eslab-ibm-es-ui-route-eventstreams.apps.demo.ibmdte.net (unsafe)** link.



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


IBM Cloud Pak for Integration
Event Streams
Welcome to IBM Event Streams
Let's get you started...
Send and receive messages with a sample application
Download and install our starter Kafka application and view data flowing to and from IBM Event Streams in just a few minutes
Create a topic
Topics are a named stream of messages. You can create a topic using the UI.
Documentation
View product documentation, as well as FAQs and tutorials.
Connect to this cluster
Get connection details and sample code to connect your applications.
Kafka basics
Learn the basics of Apache Kafka, the heart of IBM Event Streams, in 6 steps.
Connector basics
Learn about Kafka Connect, streaming to and from Event Streams, in 4 steps.
Schema basics
Learn about how IBM Event Streams manages schemas in 6 steps.
Find more in the Toolbox
Find more assets, applications, and guidance in the Toolbox.
System is healthy

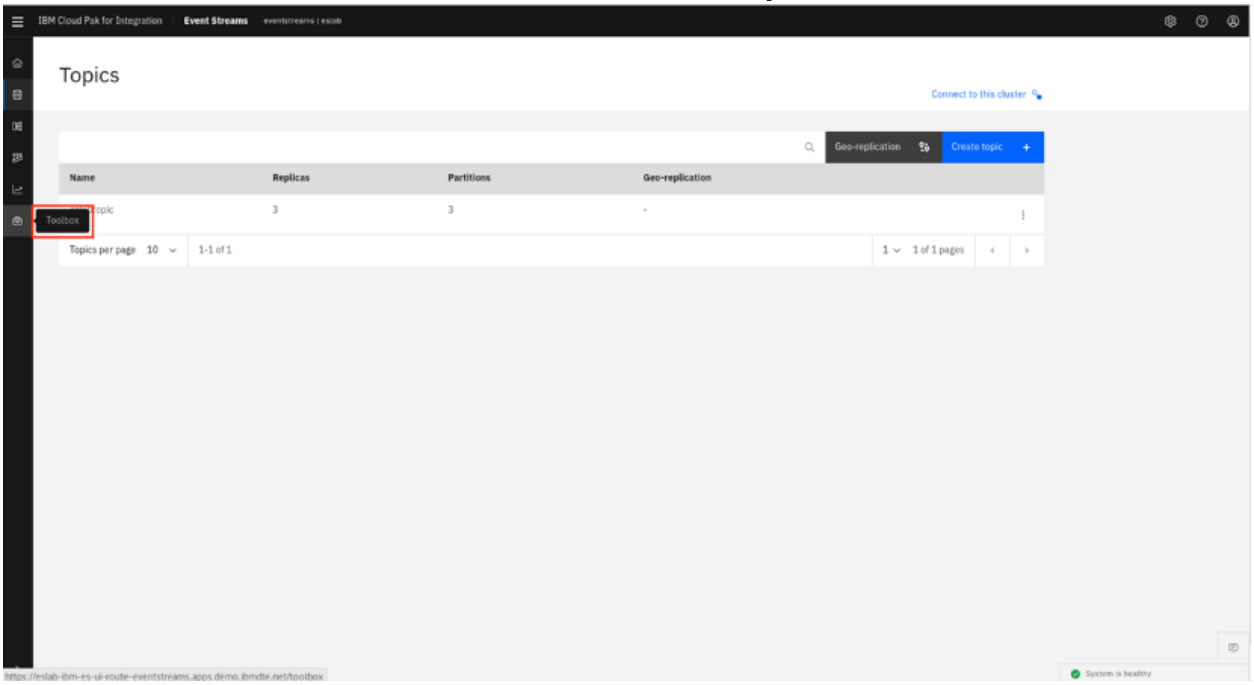


11. You've installed IBM Event Streams on IBM Cloud Pak for Integration. In the next section, learn how to manage topics, the core of Event Streams functions. While you're at the **Getting started** page, take a moment to review **Kafka Basics**, some of the in-product education.

Task 3 – Creating and managing Event Streams topic

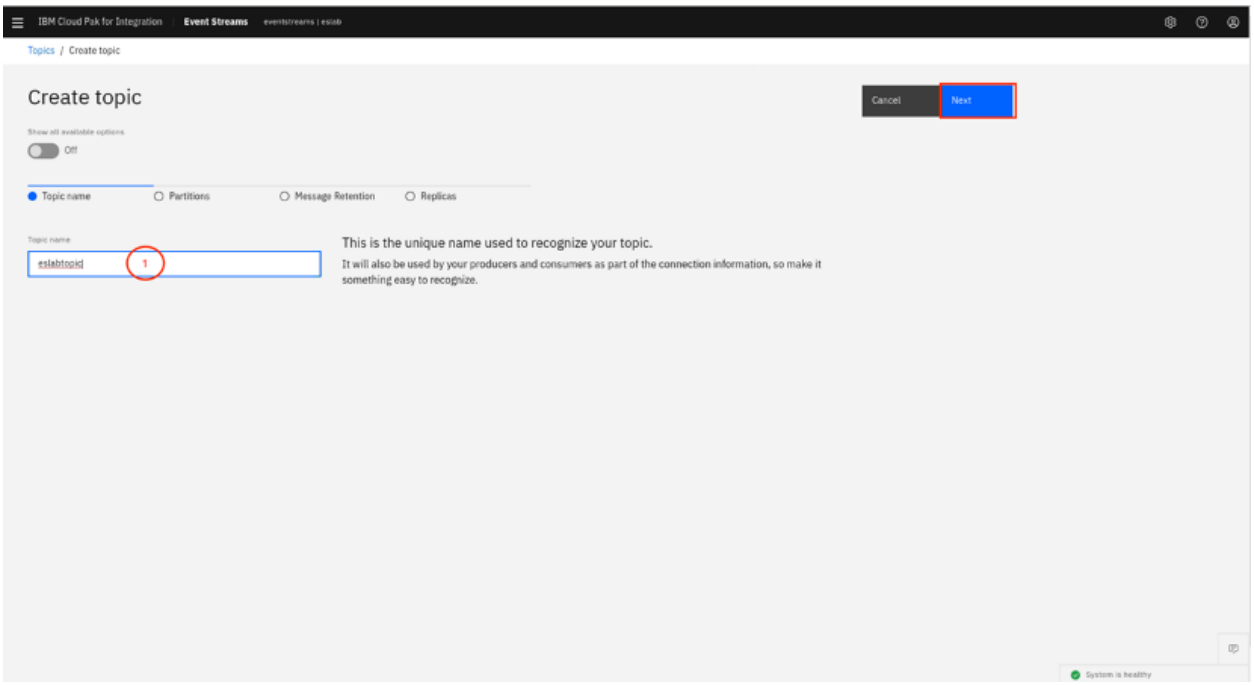
Applications connect to Event Streams topics and write to and read from them. Topics are known groupings of related data. Topics are created and configured by the Event Streams administrator.

1. In the Event Streams interface, click **Create topic**.

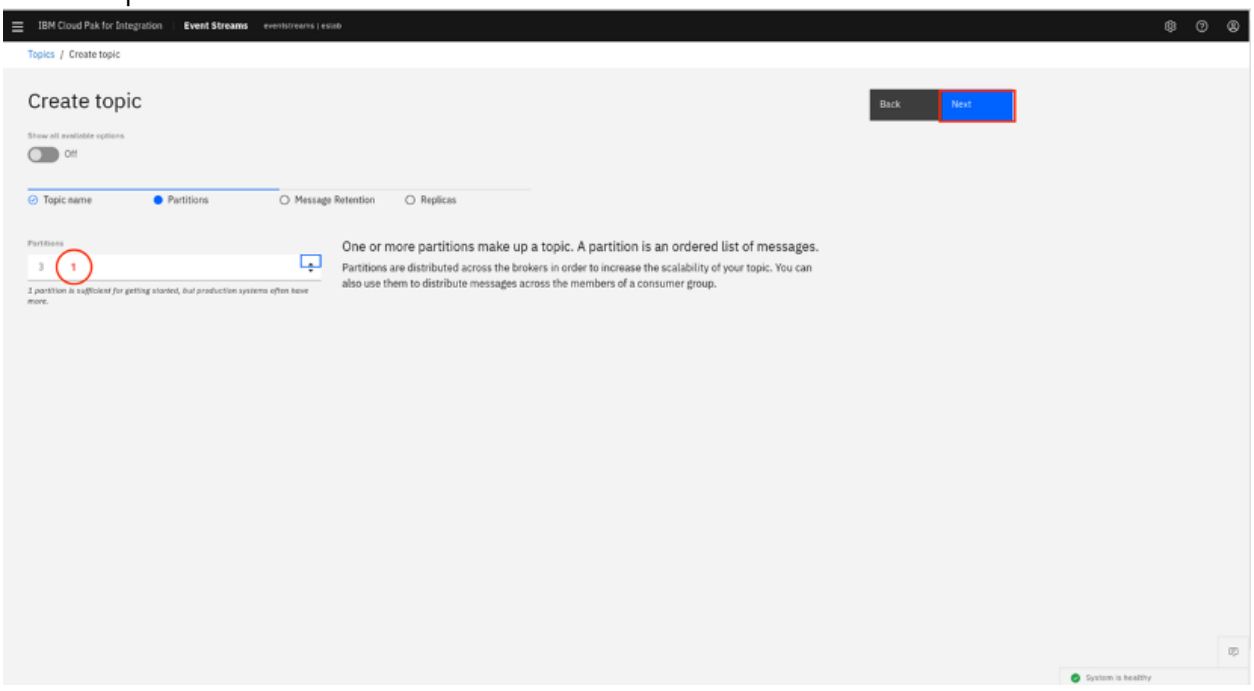


2. Enter **eslabtopic** as the topic name and click **Next**.

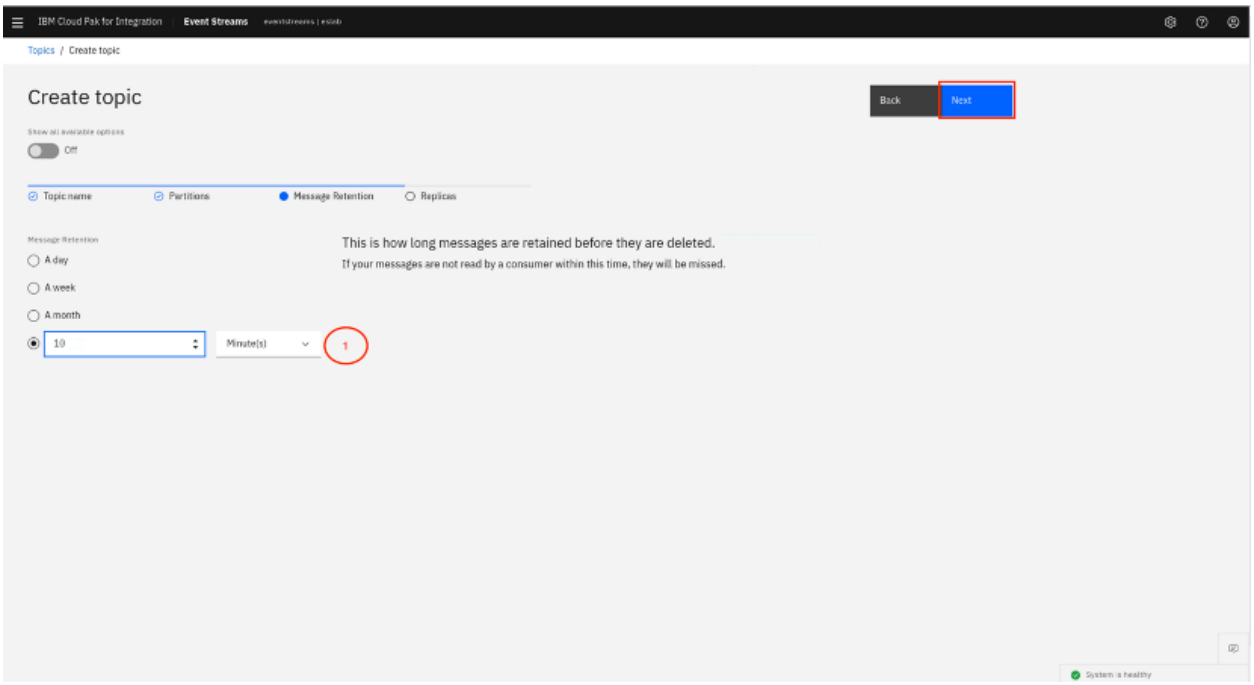
   Note: This lab is preconfigured to connect to that specific topic. You can view the full range of configuration options by setting the **Show all available options** to on. However, this tutorial only focuses on the core set.



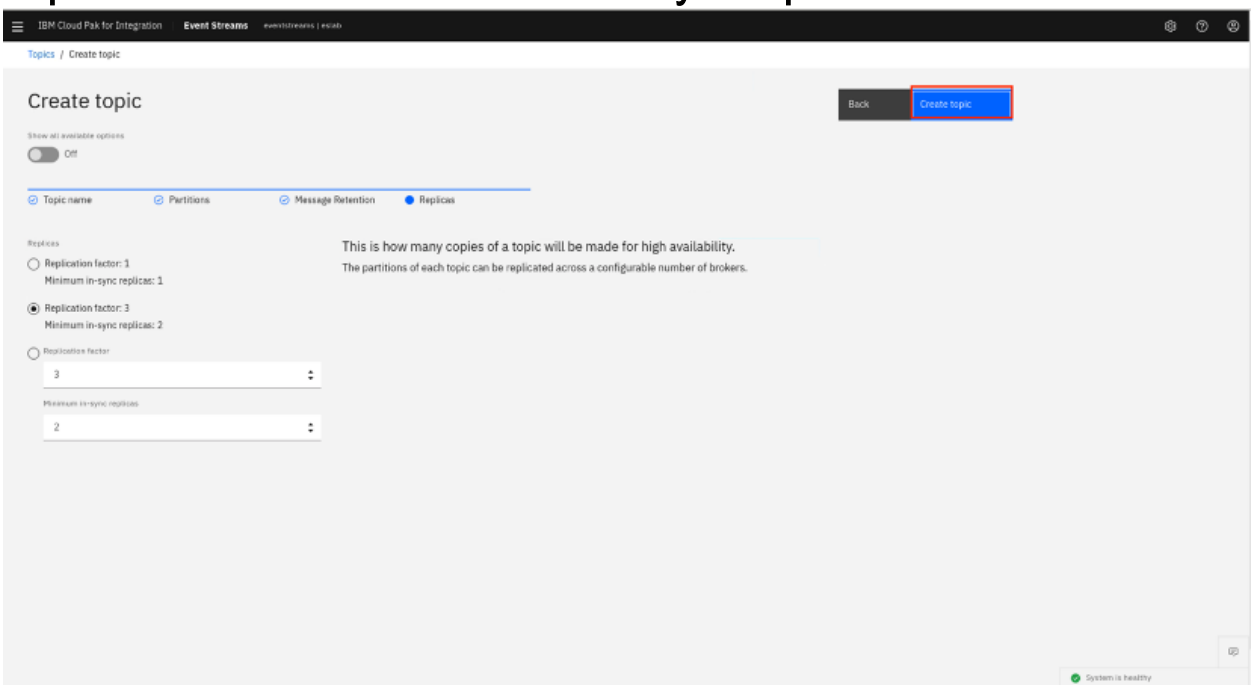
3. Set three partitions and click **Next**.



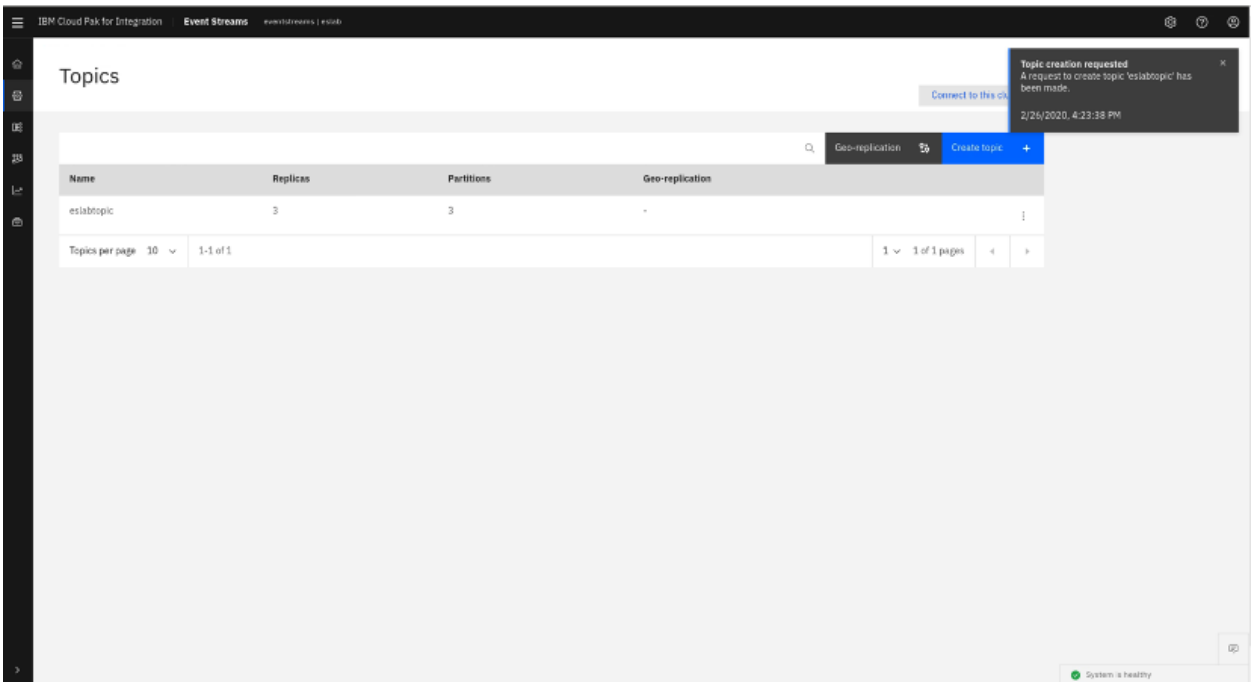
4. Define the message retention time. Set it to 10 minutes for this lab. Click **Next.**



5. Define the number of replicas for your topic. Select the default setting of **Replication factor: 3** and **Minimum in-sync replicas: 2**.



6. Click **Create topic.** The Topics page is displayed**.** Your new topic is displayed along with a completion notification. You can now connect the starter application to Event Streams .

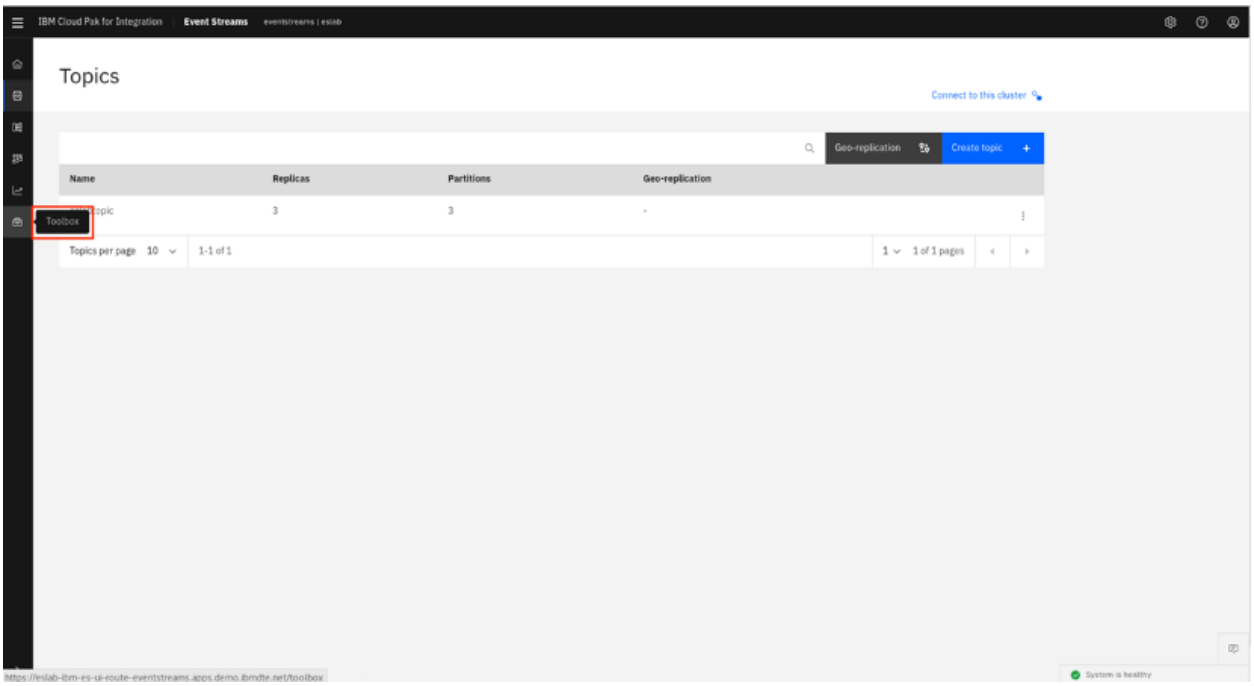


Task 4 - Use a Starter application to send and receive data

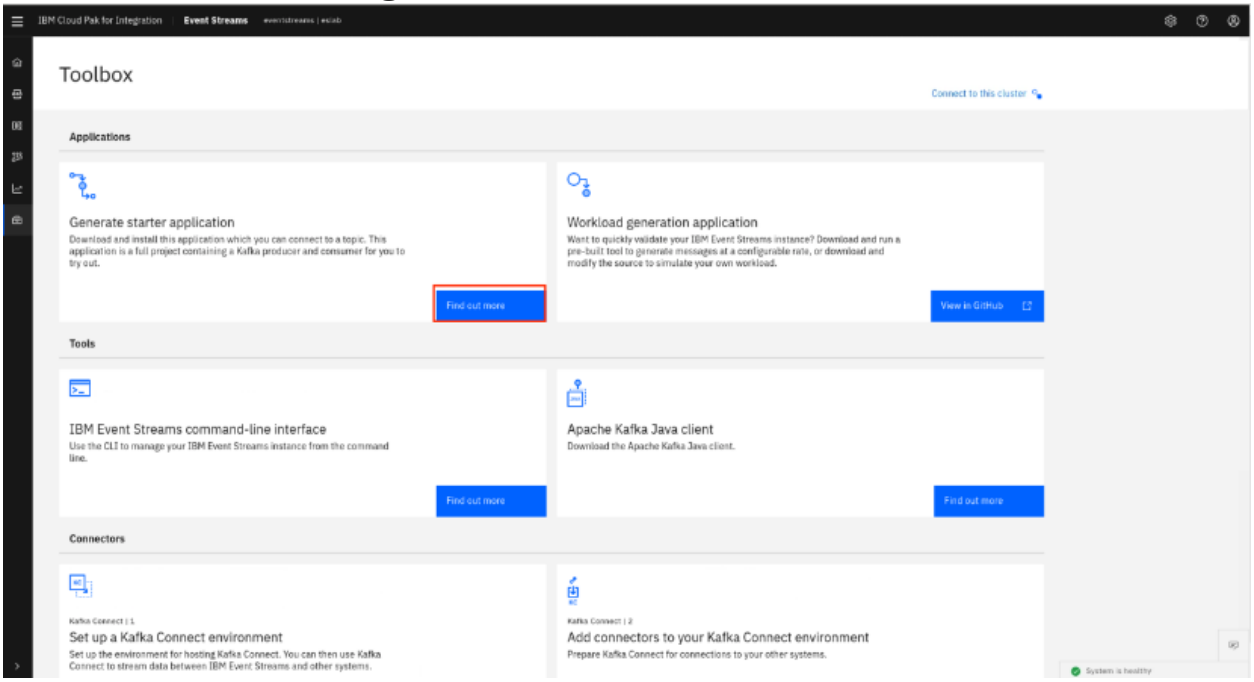
The final task in this tutorial shows you how to generate and run a starter application. Using the starter application, you can see how producing and consuming applications connect to a topic and send messages (a message is a unit of data in Kafka). Data sent by the producer can be viewed in the topic in Event Streams. You can then view the same data in the consuming application.

Event Streams includes several tools that can be used to test Event Streams and help with the development of Event Steams-based applications.

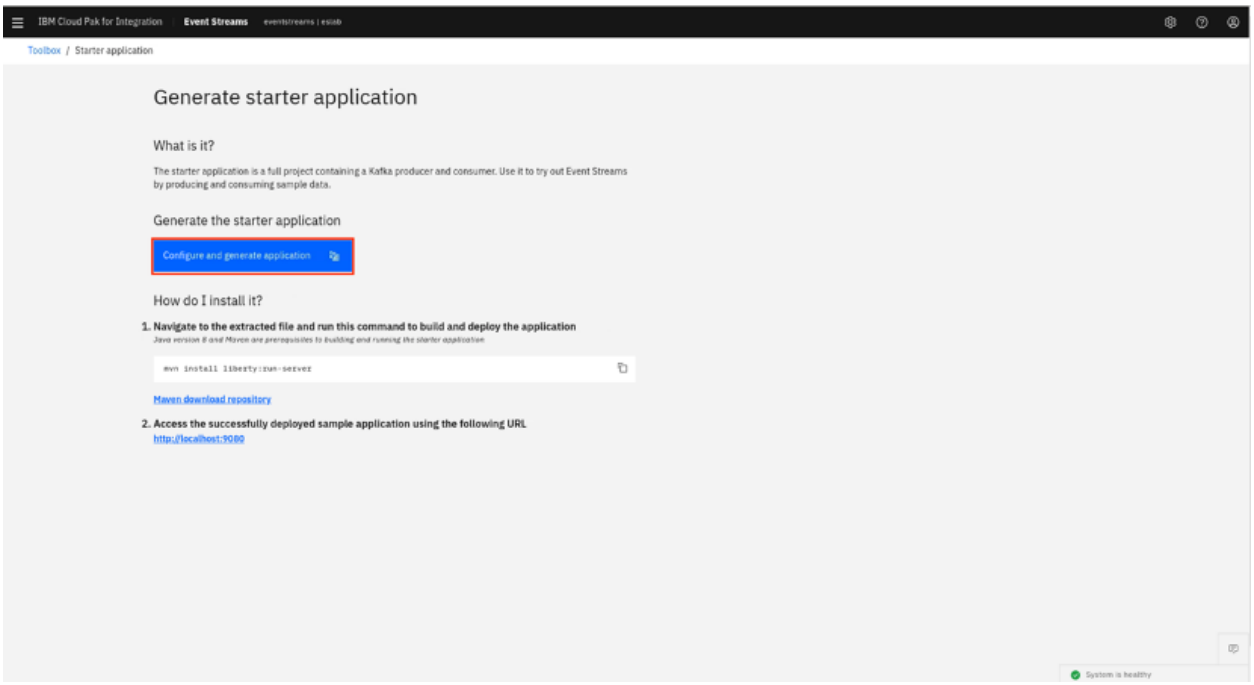
1. Click **Toolbox** in the primary navigation on the left.



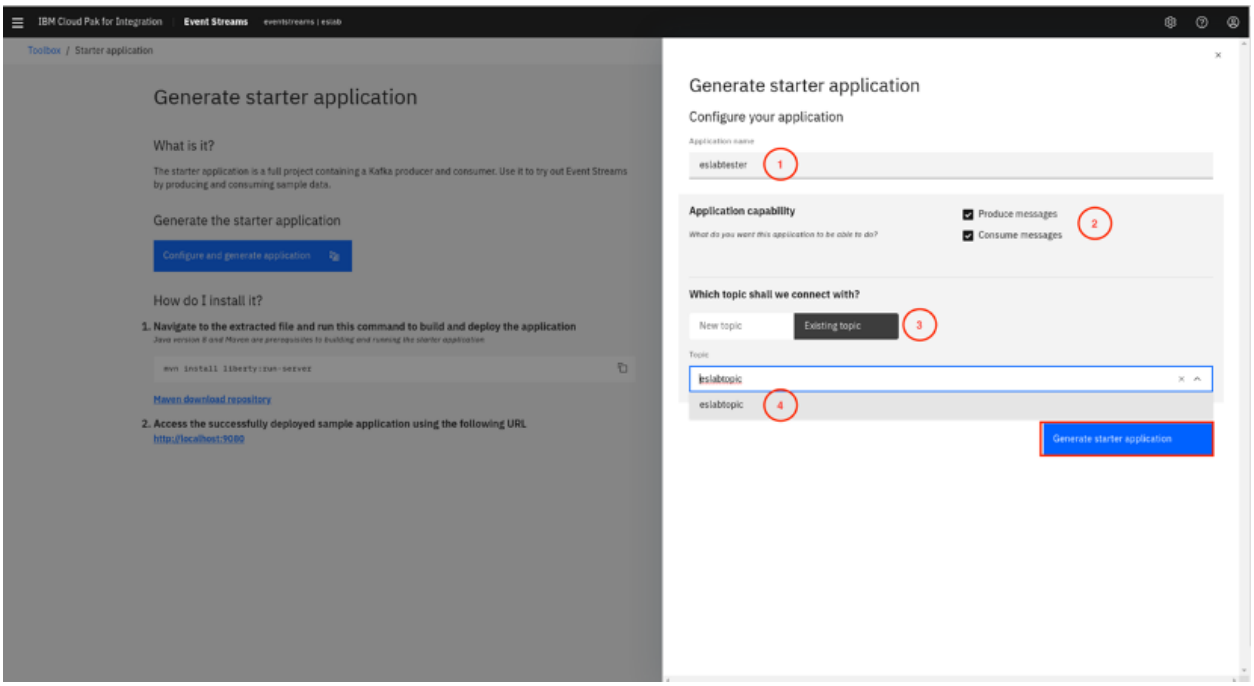
2. Go to the **Generate starter application** section and click **Find out more**. The **Generate starter application** page is displayed where you can configure and generate an application, run a liberty profile server, and send and receive messages (data).



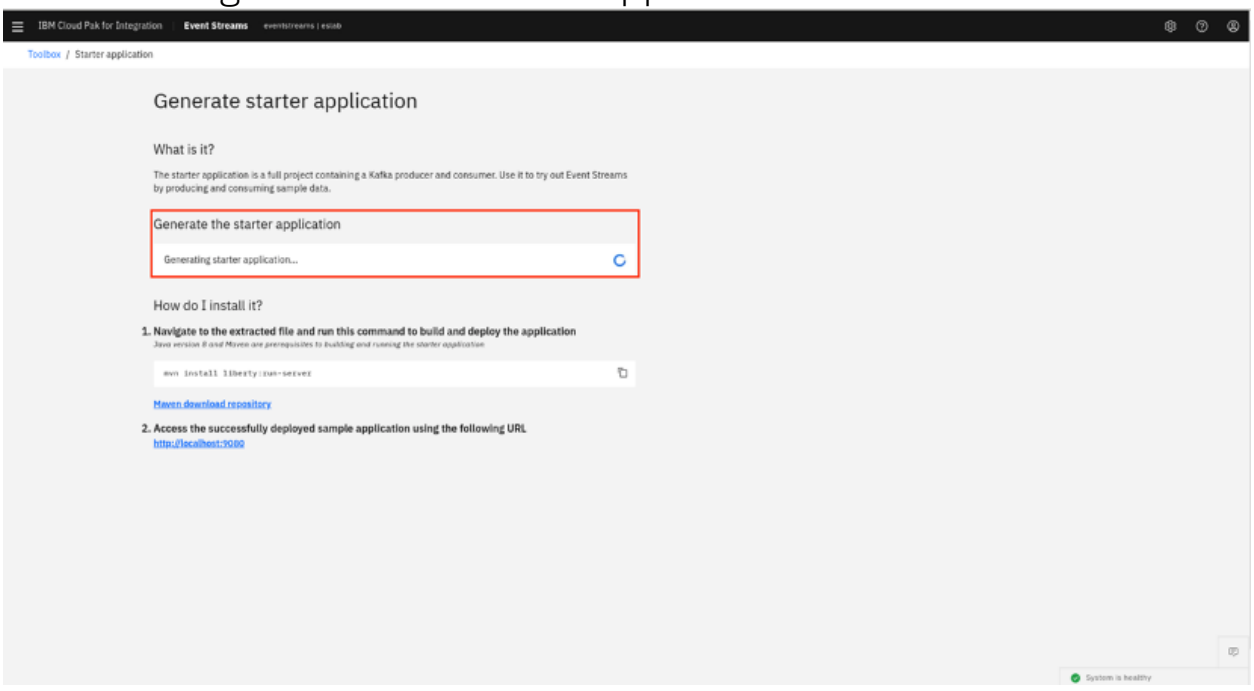
3. Click **Configure and generate application** to open the configuration panel for the starter application.



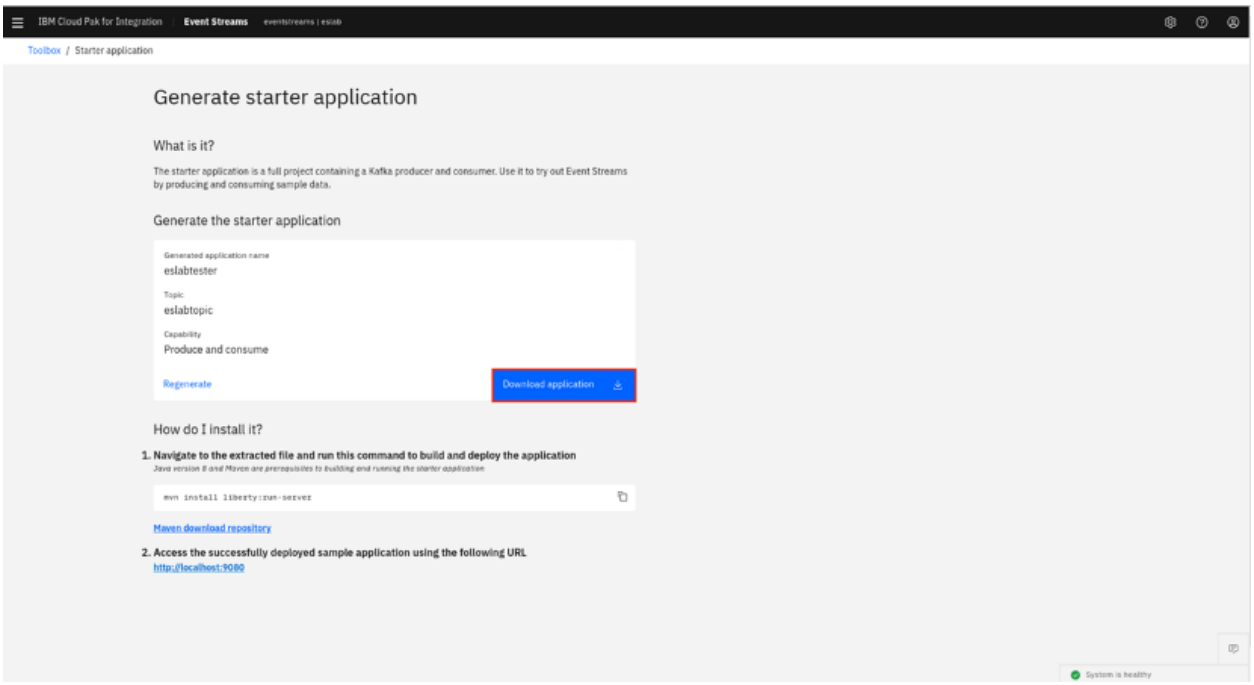
4. Configure the application as follows:
1. Enter the application name: **eslabtester**.
2. Ensure both **Produce message** and **Consume messages** are selected.
3. Select **Existing topic**.
4. Select **eslabtopic**.
5. Click **Generate starter application**.



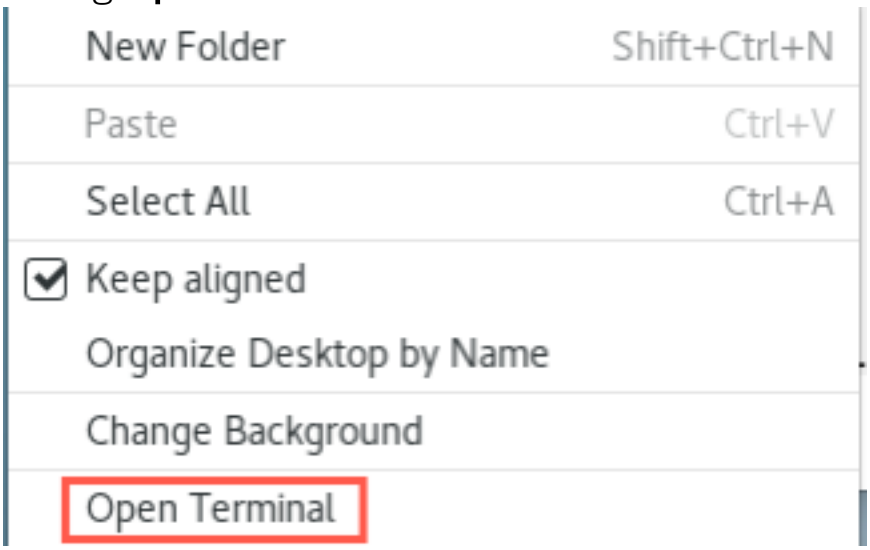
5. Event Streams generates the starter application.



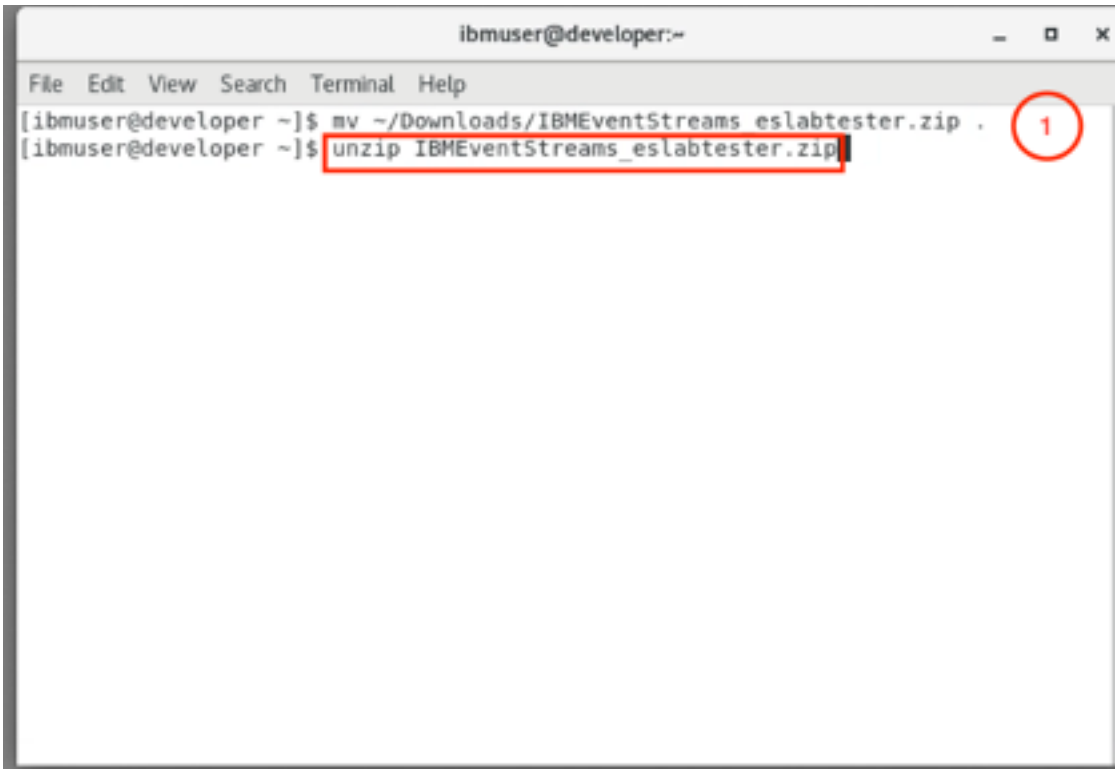
6. Click **Download application** to download the Starter application. In the popup window, check the **Save file** radio button and click **OK**. (this tutorial uses the **Downloads** directory).



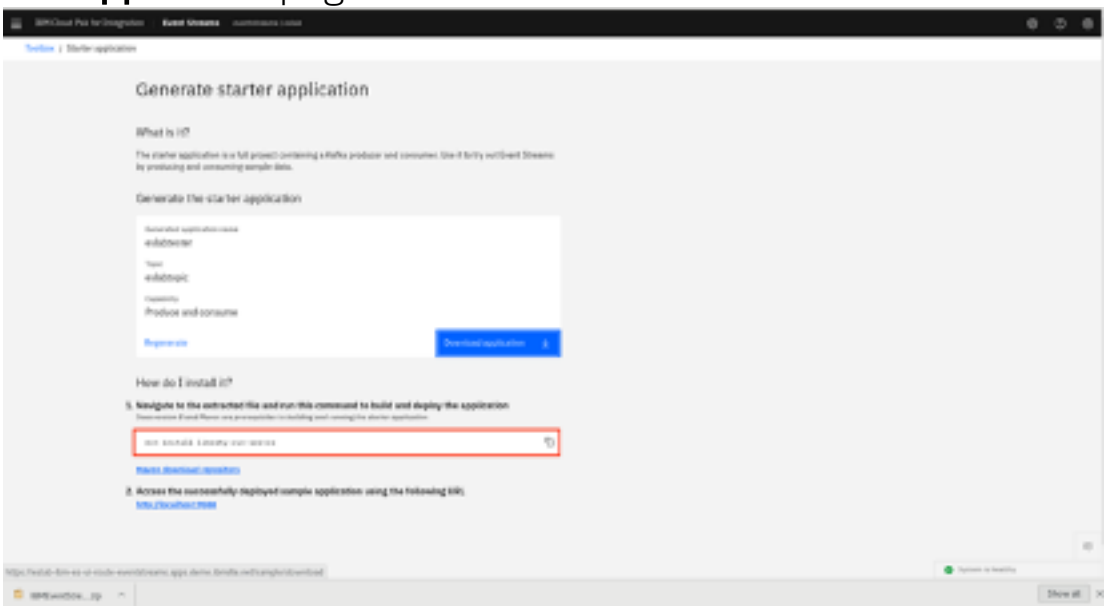
7. Open a terminal window by right clicking on the desktop and selecting **Open Terminal**



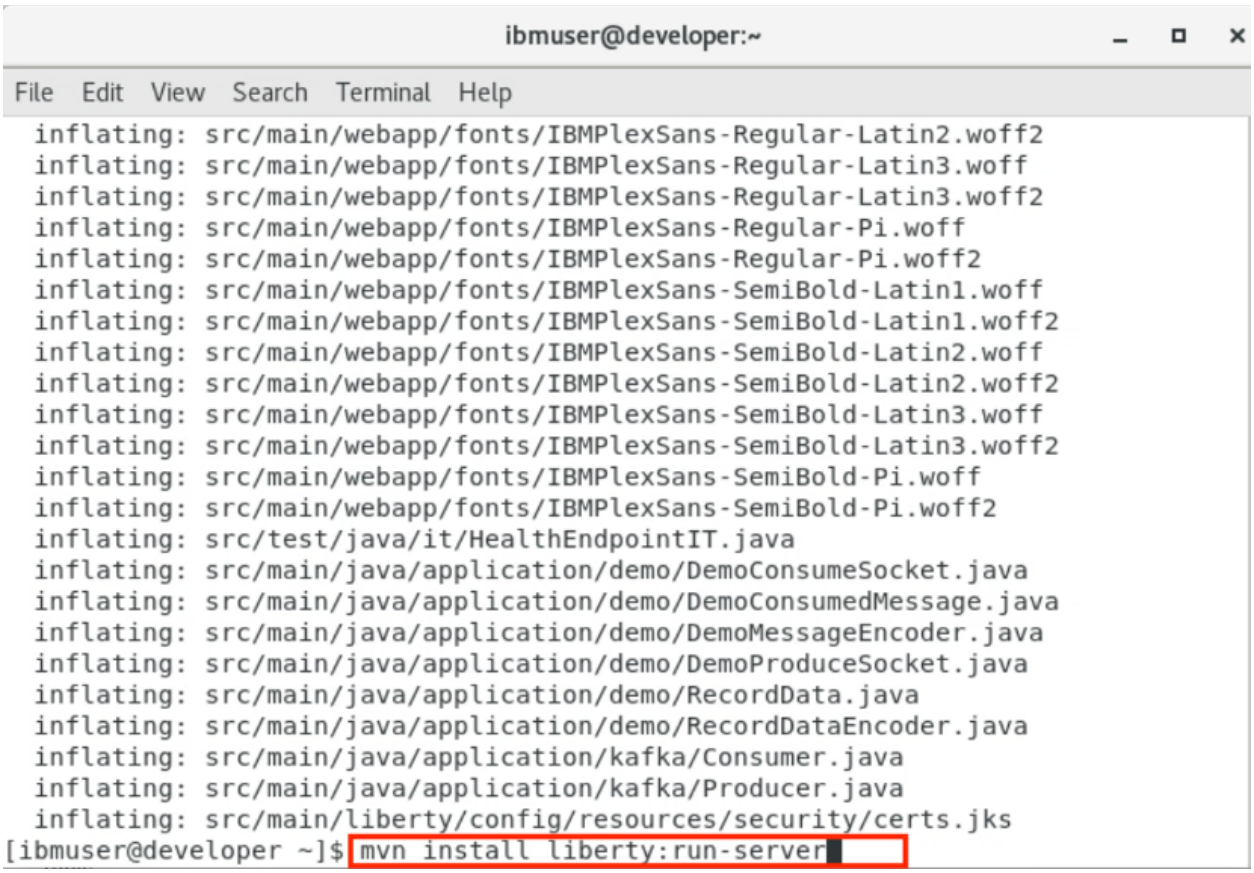
8. Move the **IBMEventStreams_eslabtester.zip** file to **home/ibmuser/** . Then, extract the file: **unzip IBMEventStreams_eslabtester.zip.**



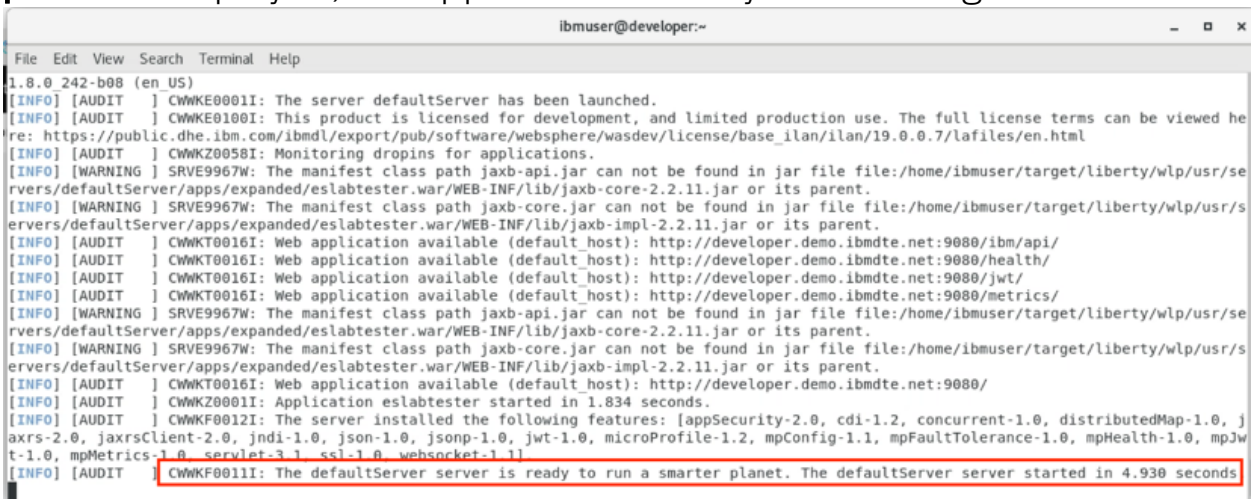
9. Click **Copy** icon **mvn install liberty:run-server** command from **Generate starter application** page.



10. **Paste** and execute the maven command to run the starter application .

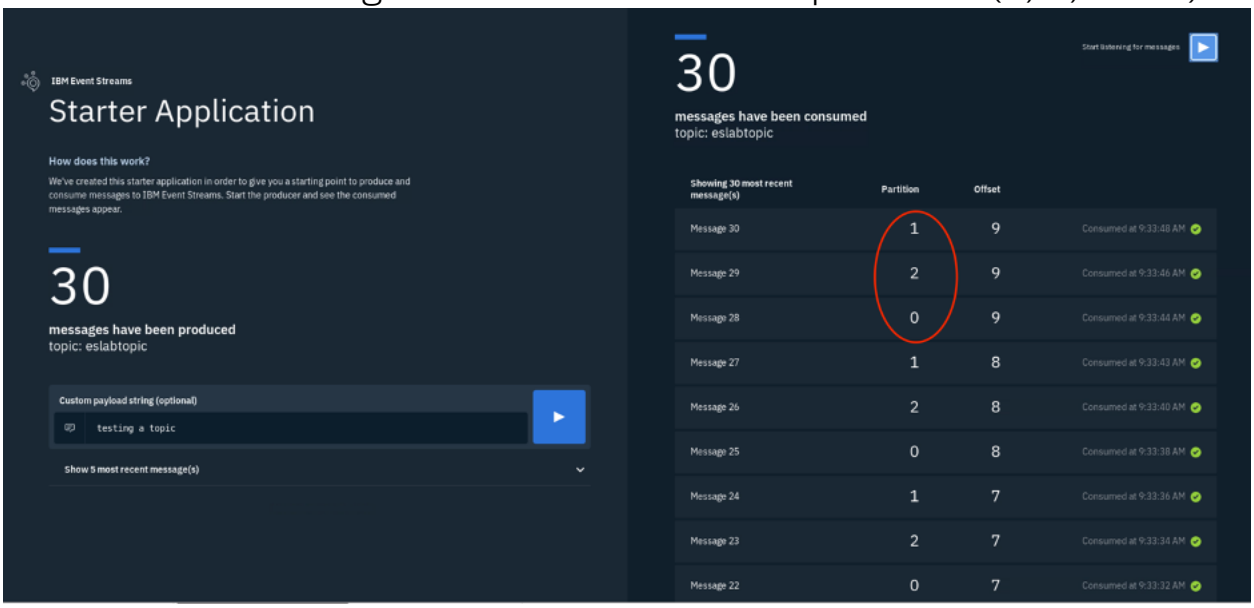


When the message **The server defaultServer is ready to run a smarter planet** is displayed, the application is ready and running.

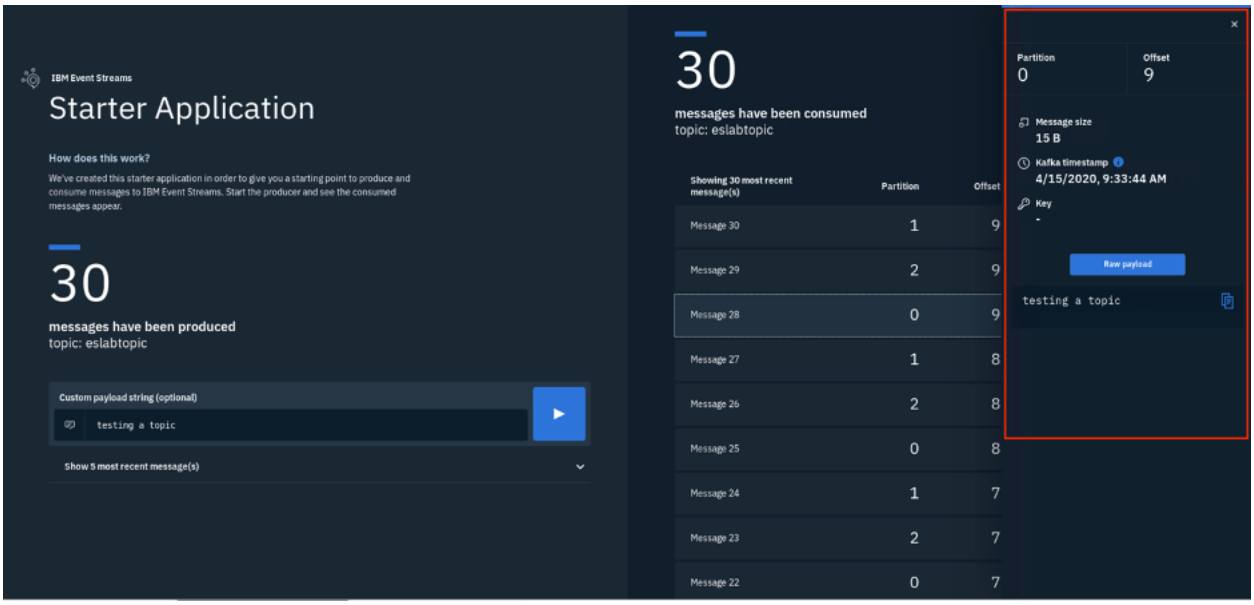


11. Open the starter application by using **http://localhost:9080** (the link is also available on the **Generate starter application** page). This page represents the producing application on the left of the screen and the consuming application on the right.

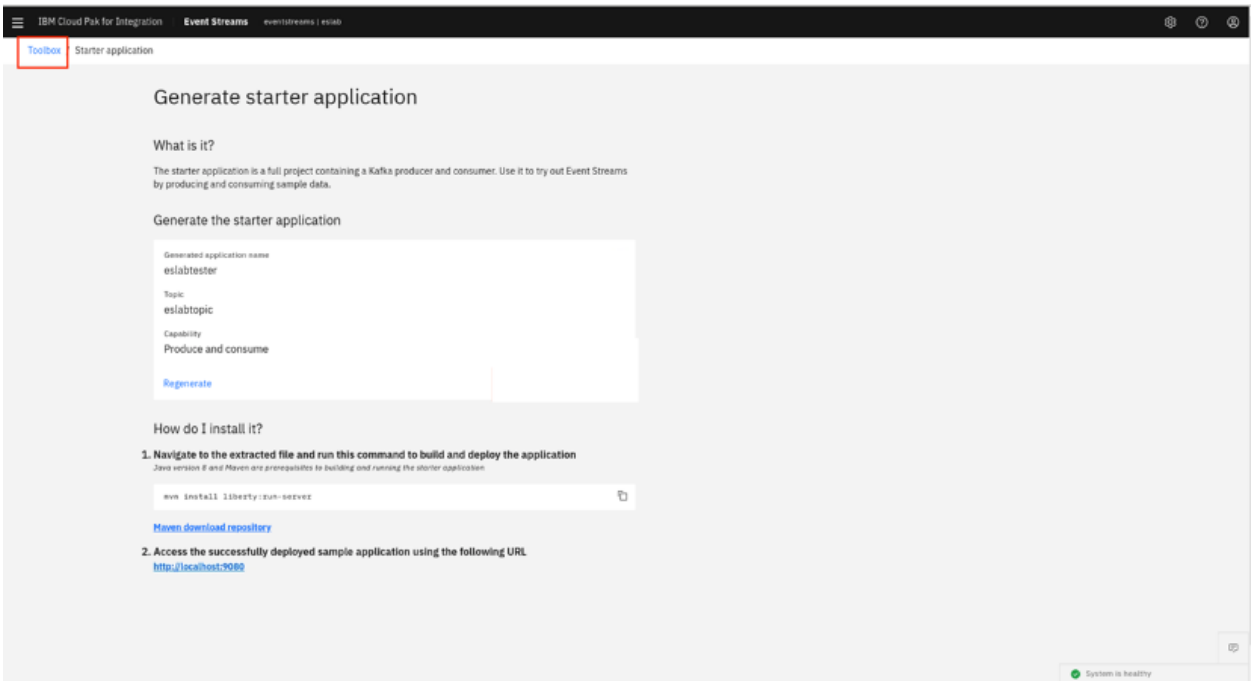
12. Type a message in the **Custom payload string** field (1). Click **play** on the **producer,** and see messages appearing in the consuming application as they are consumed from the Event Streams topic.
Look at **Offset. Stop the Producer by clicking the start/pause button (play button when not active, and spinning blue circle when running).** Also, click the blue icon in the **Consumer** to stop listening for messages. You can see that the message is distributed on three partitions (0, 1, and 2).



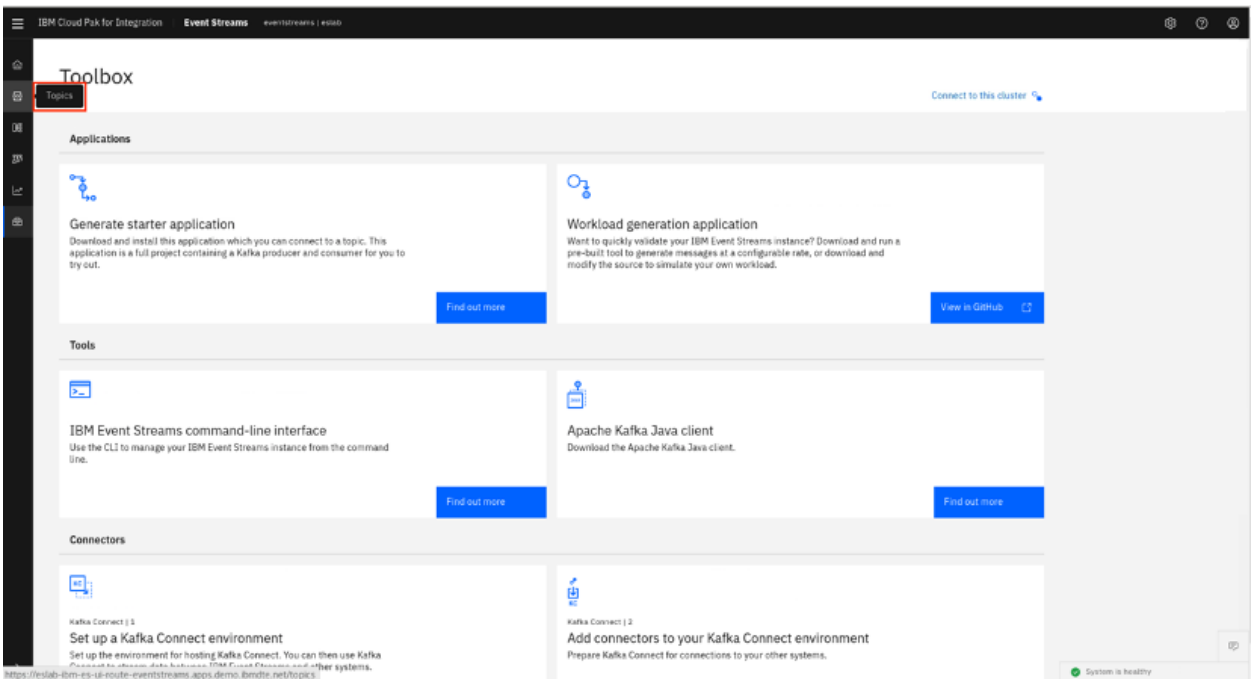
13. View the message being produced (1), and click any message being consumed (2) to view all available information about the message.



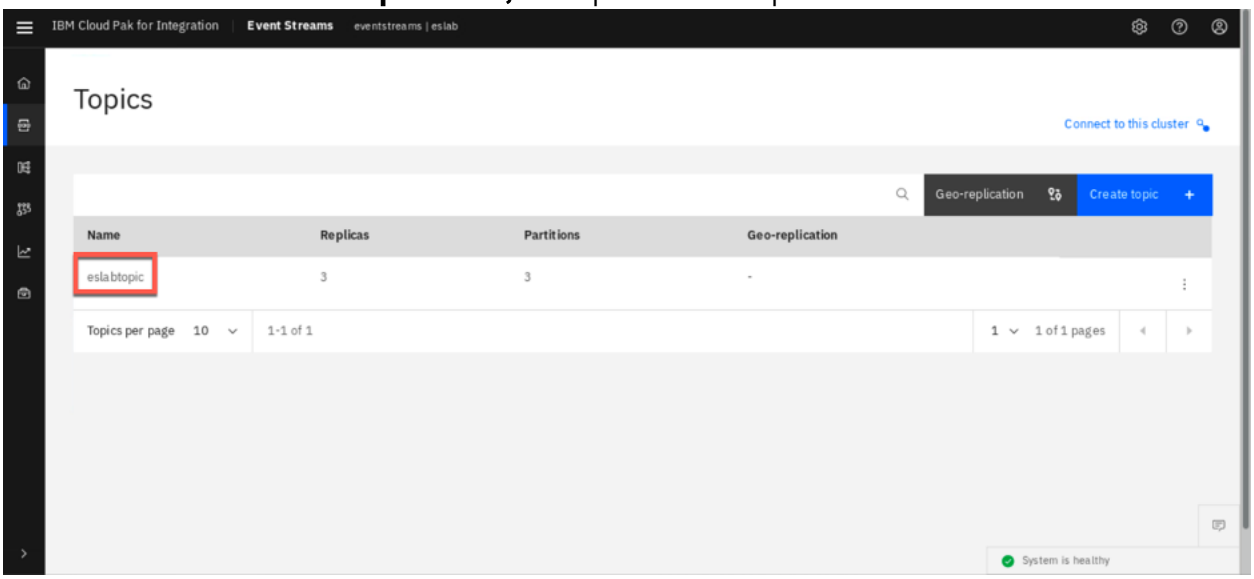
14. Return to the **Generate starter application** page and click **Toolbox.**



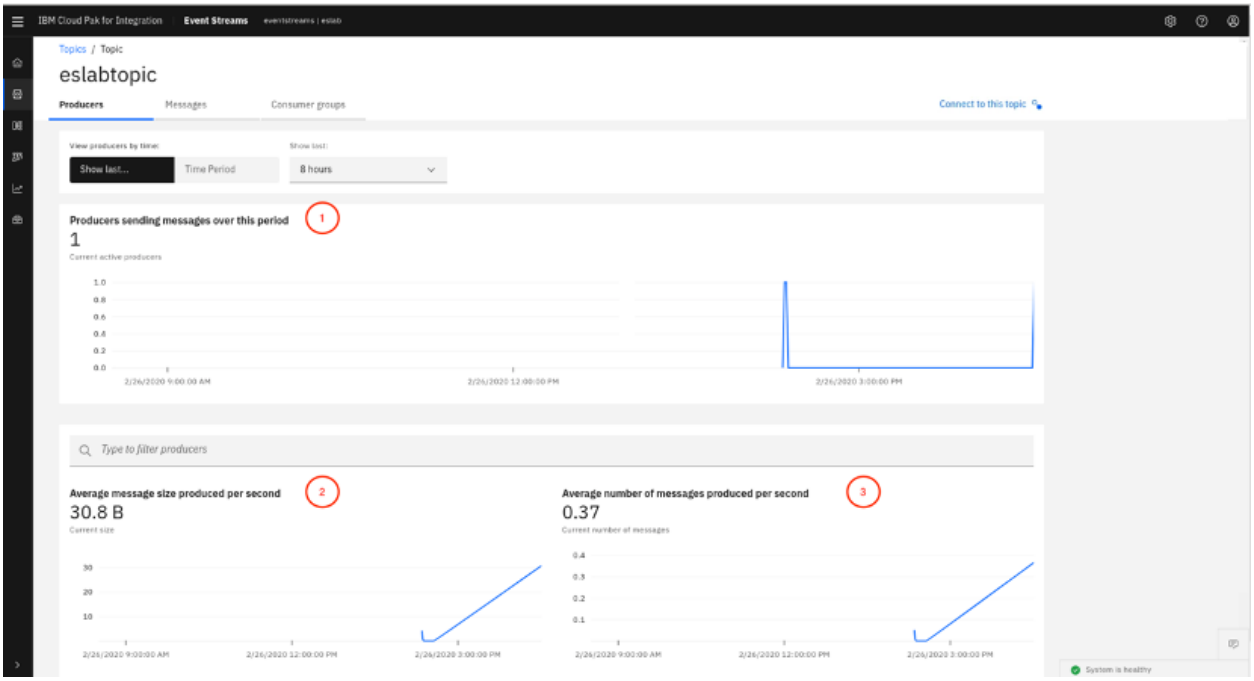
15. Click **Topics** in the primary navigation on the left.



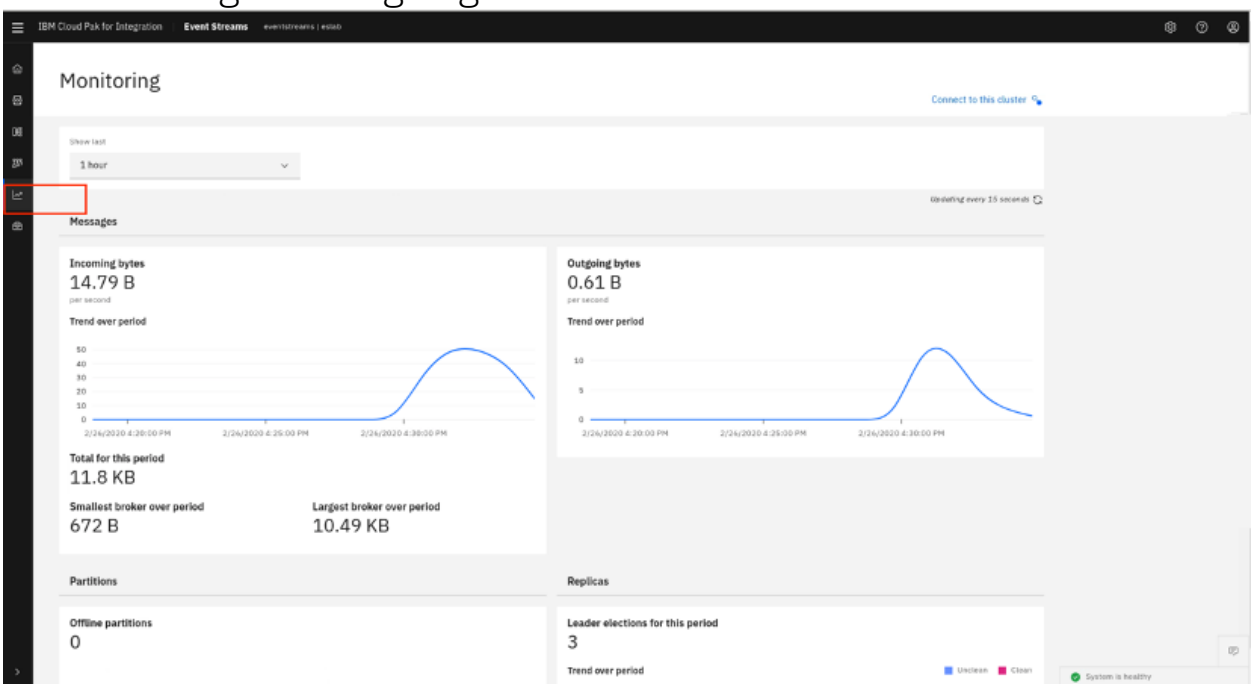
16. Click on **eslabtopic line,** to open the topic.



17. Use the Event Streams interface to evaluate the messages produced, for example, information such as the **Average message size produced per second** and the **Average number of messages produced per second**.



18. Click **Monitoring** in the primary navigation on the left to view the rate of incoming and outgoing data.



Summary

You have completed this Tutorial and you've learned how to:

- Install an Even Streams Instance

- Set up a topic and connected an application to test the flow of data through Event Streams. 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