

Red Hat  
**Summit**

**Connect**

# Managed Kafka jak zacząć?

Andrzej Kowalczyk

# Cloud Services Managed by Red Hat

Managed OpenShift + Application Services + Data Services



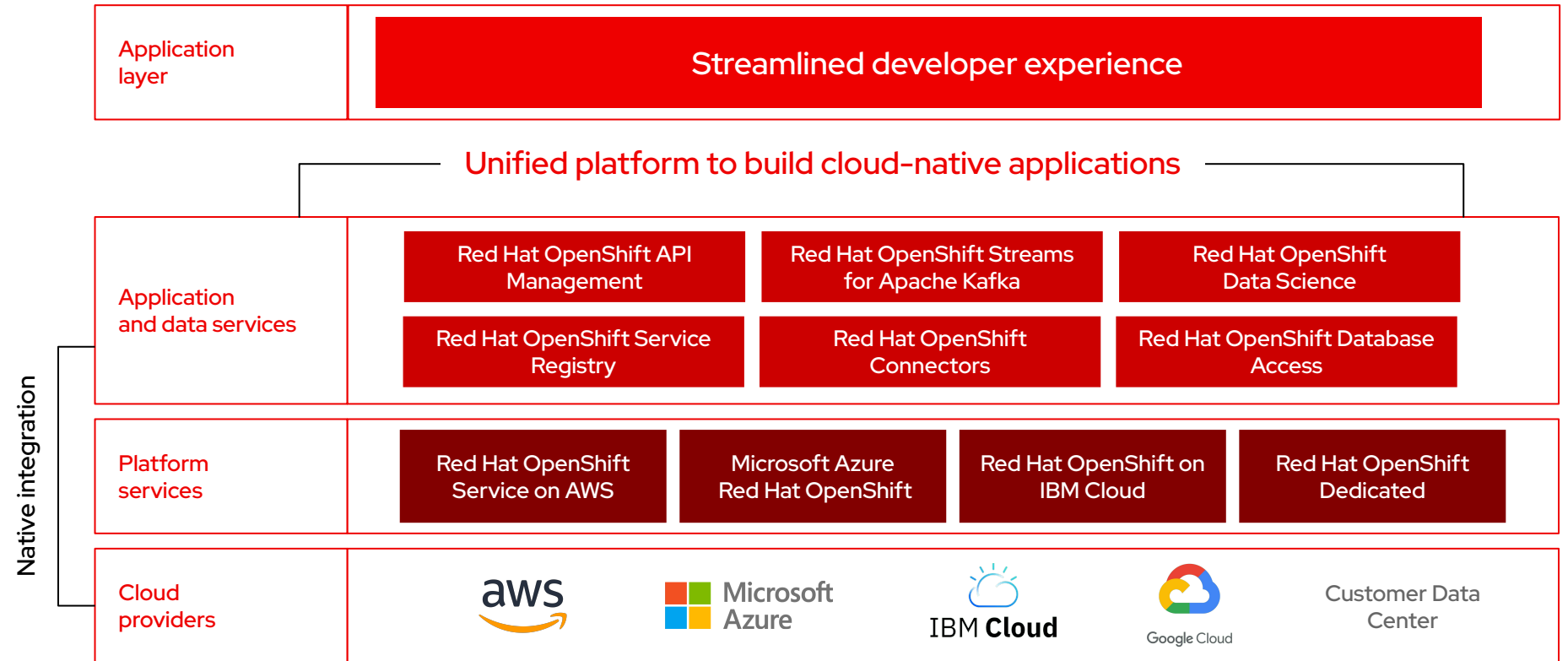
Full stack management and unified experience



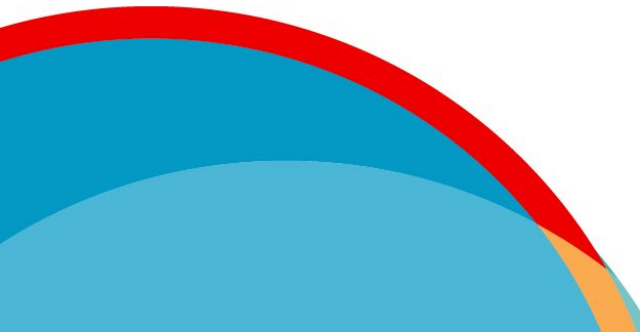
Maximize full value of Red Hat OpenShift



Hybrid cloud flexibility



# Red Hat OpenShift Streams for Apache Kafka



# Choosing your Streams

Self Managed

Red Hat  
Integration (AMQ  
Streams)



Fully Managed by Red Hat

Red Hat  
OpenShift Streams  
for Apache Kafka



# The value of Red Hat OpenShift Streams for Apache Kafka



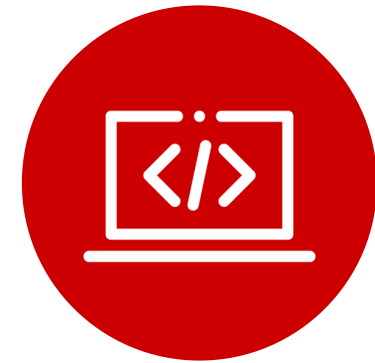
## Faster application velocity

Begin developing immediately and continuously respond to change



## Unified experience across all clouds

Seamlessly connects applications across public and private clouds



## Kafka ecosystem for streams-based applications

Delivers a curated set of cloud services to simplify delivery of stream-based applications

# Red Hat OpenShift Streams for Apache Kafka - Key features

Complete solution for stream-based applications



## **Real-time, streaming data broker** -

Dedicated Apache Kafka cluster delivered as a service in the cloud and location of choice



**Schema registry** - OpenShift Service Registry is included, making it easy for dev teams to publish, communicate and discover any streaming data topics.



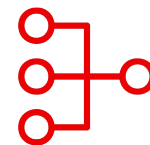
## **Delivered as a service, managed by Red Hat** -

Reduce operational overhead and complexity with pre-built connectors managed by Red Hat.



## **Streamlined developer experience** -

developer-first, consistent experience that shields the user from administrative tasks



**Connectors** - securely connect to distributed services to consume and share streaming data between apps, enterprise systems, and cloud provider services.

# Red Hat OpenShift Streams for Apache Kafka

Red Hat managed solution for stream-based applications

**Streamlined developer experience:** a curated solution with a developer-first, consistent experience

**Delivered as a service, managed by Red Hat SRE** - 24x7 global support and a 99.95% service-level agreement (SLA)

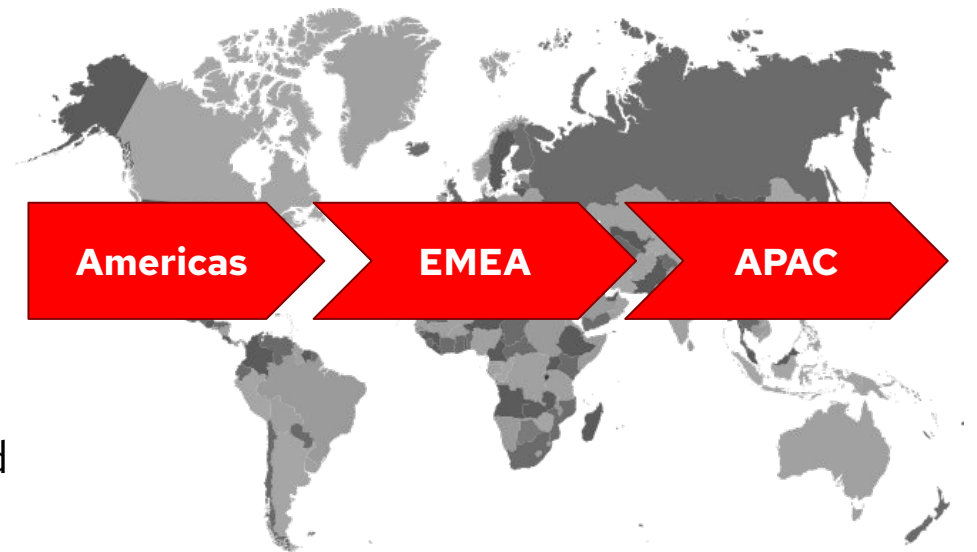
**Real-time, streaming data broker** - Dedicated Apache Kafka cluster delivered as a service in the cloud and location of choice

- ▶ Access to Kafka brokers, topics, and partitions
- ▶ Managed ZooKeeper
- ▶ Metrics and monitoring
- ▶ Integrated identity & access management

# Hosted & managed service offering

Red Hat cloud services are managed and operated by Red Hat's Site Reliability Engineers

- ▶ SREs serve as the cloud provider account owner and cluster administrator owning the **SLA (99.95%) and uptime**
- ▶ Responsible for the **24x7 support** for all managed and hosted environments
  - Including for building, installing, upgrading, **managing and maintaining every cluster**
- ▶ SRE teams are distributed **across 3 regions**: APAC, EMEA and Americas
- ▶ The team ensures **open communications channels** centralized around the dedicated customer portal





# Red Hat Management

We deliver premium support and 99.95% uptime



## Planned & Emergency Maintenance Events

Included and performed by Red Hat Service Reliability Engineering (RH SRE)



## Monitoring, Logging and Alerting

Available for RH SRE teams to track the performance of the technology stack installed by Red Hat



## CVEs

Performed by RH-SRE where Red Hat Product Security dictates



## Backups and DR

Performed by RH-SRE where Red Hat Product Security dictates

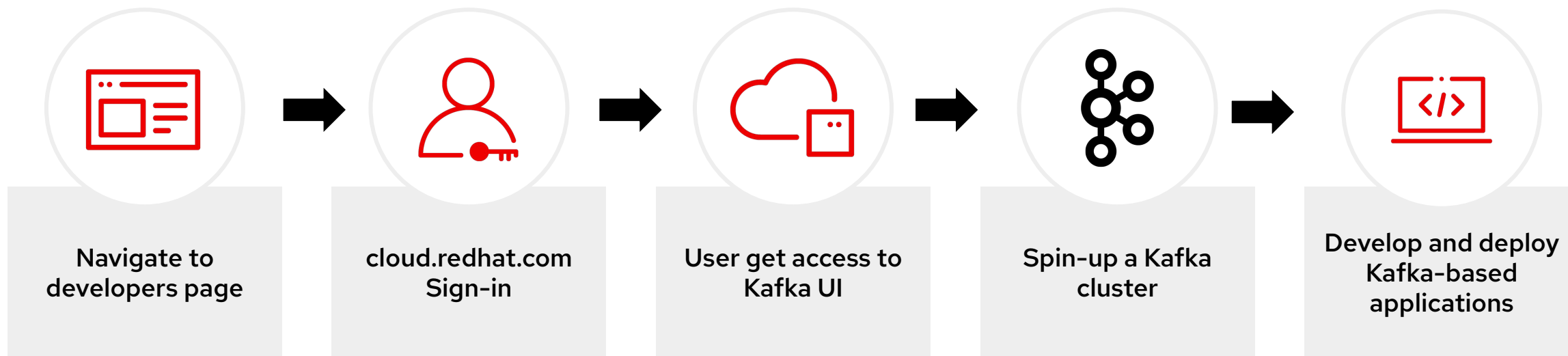
# Red Hat OpenShift Streams for Apache Kafka

Components pre-configured to optimize for high availability and reliability

- ▶ OpenShift Streams for Apache Kafka provides **dedicated Kafka instances**
- ▶ Service was designed following **experts recommended configurations**
  - Each Kafka cluster consists of **three brokers**, with each broker in a different AZ of the target region
  - **All topics have a replication factor of 3**, resulting in a partition replica placed in each AZ
  - All topics have an **in-sync replica count of 2**
- ▶ The Zookeeper environment supporting Kafka instances is completely managed by Red Hat and not accessible to end users

# Red Hat OpenShift Streams for Apache Kafka

How to get started?



# Moje doświadczenia

# console.redhat.com - tu się zaczyna

Home | console.redhat.com

Red Hat Hybrid Cloud Console | All apps and services

Home

- Application Services
- OpenShift
- Red Hat Enterprise Linux
- Ansible Automation Platform

App services

- 2 Kafka Instances

RHEL

- 799 Connected systems
- 183 Stale systems

Ansible Automation Platform

- 4 Ansible Platform C...
- 118 Collections
- 56 Partners
- 4 Platforms
- 0 Portfolios

Red Hat Insights

Gain increased visibility into your hybrid cloud deployments so you can improve performance and increase security.

RHEL 2

- View 8 incidents  
Insights has identified 8 incidents affecting your systems.
- Create a remediation playbook  
Fix issues identified by Insights on your systems.

Recommendations

- Get started with RHOSAK  
Learn how to create and use a Kafka instance.
- Download and install the Application Services CLI  
Get started using your managed Kafka instances.
- Get started with Quarkus apps and Streams for Apache Kafka  
Learn how to connect to your Kafka instance from a Quarkus application.
- Try Red Hat OpenShift API Management  
Deploy, monitor, and control APIs throughout their entire life cycle.
- Get started with Red Hat OpenShift API Management  
Deploy, monitor, and control APIs throughout their entire life cycle.
- View Automation Services Catalog approvals  
Govern content with approval processes.
- Create an Ansible Playbook



# Pokażę jak działa wersja trial (instancja będzie usunięta po 48h)

Overview | Red Hat OpenShift x +

console.redhat.com/application-services/overview

Red Hat Hybrid Cloud Console

All apps and services

Application Services

Overview

API Management

Data Science

Service Accounts

Service Registry

Streams for Apache Kafka

Insights

Subscriptions

Database Access

Learning Resources

## Get started with Red Hat OpenShift Application Services

Cloud services for managed OpenShift

Red Hat OpenShift Application Services deliver a streamlined developer experience for building, deploying, and scaling cloud-native applications.

Don't miss your chance to preview one of our new application services.

[Try OpenShift Streams for Apache Kafka](#)

Reduce operational costs and complexity for application development teams with hosted and managed OpenShift Application Services.

- Red Hat OpenShift API Management**  
API Management supports application development teams that want to take an API-first approach to building microservices-based applications.  
This service is hosted and managed by Red Hat as an add-on for managed OpenShift.
- Red Hat OpenShift Data Science**  
Field trial  
Data Science provides a fully-supported sandbox to rapidly develop, train, and test containerized machine learning models in the public cloud before deploying in production.  
This service is hosted and managed by Red Hat as an add-on for managed OpenShift.
- Red Hat OpenShift Service Registry**  
Service Registry provides a metadata layer for developers and applications to publish, discover and evolve schema definitions and API designs.  
This service is fully hosted and managed by Red Hat and offered at no cost to Streams for Apache Kafka and API Management users.
- Red Hat OpenShift Streams for Apache Kafka**  
Streams for Apache Kafka enables IT development teams to capture, process, and stream real-time data across hybrid cloud environments.  
This service is fully hosted and managed by Red Hat for stream-based applications.

# Można użyć już istniejącego broker (twój) lub stworzyć nowy

The screenshot shows the Red Hat Hybrid Cloud Console interface for managing Kafka instances. The left sidebar contains navigation options: Application Services, Overview, API Management, Data Science, Service Accounts, Service Registry, Streams for Apache Kafka (selected), Documentation, Insights, Subscriptions, Database Access, and Learning Resources. The main content area is titled 'Kafka Instances' and features a search bar with a 'Create Kafka instance' button. Below this is a table with two instances:

Name	Cloud provider	Region	Owner	Status	Time created
samurai-pizza-kafkas	Amazon Web Services	US East, N. Virginia	[Redacted]	Ready	about 1 hour ago 48 hours duration
ak	Amazon Web Services	US East, N. Virginia	akowalc@redhat.com	Ready	2 days ago 48 hours duration

At the bottom right of the console, there is a 'Feedback' button and a notification icon showing '8'.

# Kopiuję szczegóły połączenia i tworzę service account

The screenshot displays the Red Hat Hybrid Cloud Console interface. On the left, a sidebar lists navigation options: Application Services, Overview, API Management, Data Science, Service Accounts, Service Registry, Streams for Apache Kafka (selected), Kafka Instances, Documentation, Insights, Subscriptions, Database Access, and Learning Resources. The main content area is titled 'Kafka Instances' and features a table with columns for Name, Cloud provider, and Region. Two instances are listed: 'samurai-pizza-kafkas' and 'ak'. The 'ak' instance is selected, and its details are shown in a modal window on the right. The modal has two tabs: 'Details' and 'Connection'. The 'Connection' tab is active, showing the bootstrap server URL: 'ak-c--usk-rg-lbvru-ou-g.bf2.kafka.rhcloud.com:443'. Below this, the 'Service accounts' section is highlighted with a red circle and contains a 'Create service account' button. The 'Authentication method' section shows 'SASL/OAUTHBEARER' as the recommended method. The 'Token endpoint URL' field is also highlighted with a red circle and contains the value 'https://api.redhat.com/oauth/authorize?response\_type=token&scope=openid-connect/token'. The 'SASL/PLAIN' method is also visible at the bottom of the modal.

Name	Cloud provider	Region
samurai-pizza-kafkas	Amazon Web Services	US East, N. Virginia
ak	Amazon Web Services	US East, N. Virginia

**Connection details for instance 'ak':**

- Bootstrap server:** ak-c--usk-rg-lbvru-ou-g.bf2.kafka.rhcloud.com:443
- Service accounts:** Create a service account to generate credentials. Manage the service accounts you create on the [Service Accounts](#) page. [Create service account](#)
- Authentication method:** SASL/OAUTHBEARER (Recommended)
- Token endpoint URL:** https://api.redhat.com/oauth/authorize?response\_type=token&scope=openid-connect/token
- SASL/PLAIN:** Use your service account credentials (Client ID and Client secret) as the user name and password to authenticate your client with the Kafka instance.



# Tworzę tematy (topics)

The screenshot shows the Red Hat OpenShift Streams console interface. The browser address bar displays the URL: `console.redhat.com/application-services/streams/kafkas/c94usk0rg6lbvru2ou2g/topics`. The user is logged in as Andrzej Kowalczyk. The left sidebar shows the navigation menu with 'Streams for Apache Kafka' expanded to 'Kafka Instances'. The main content area is titled 'ak' and has tabs for 'Dashboard', 'Topics', 'Consumer groups', and 'Access'. The 'Topics' tab is active, showing a search bar, a 'Create topic' button (circled in red), and a table of existing topics. The table has columns for Name, Partitions, Retention time, and Retention size. There are three topics listed: 'full\_names', 'movies', and 'prices', each with 1 partition and a retention time of 604800000 ms (7 days). The retention size for all is 'Unlimited'. The table is paginated to show 1-3 of 3 items.

Name	Partitions	Retention time	Retention size
<a href="#">full_names</a>	1	604800000 ms (7 days)	Unlimited
<a href="#">movies</a>	1	604800000 ms (7 days)	Unlimited
<a href="#">prices</a>	1	604800000 ms (7 days)	Unlimited

# Dodaję uprawnienia (connect/read/write)

The screenshot shows the Red Hat OpenShift Streams console interface. A 'Manage access' dialog is open, displaying a table of permissions for a selected Kafka instance. The dialog includes a 'Save' button and a 'Cancel' button. The background shows the console navigation menu and a list of Kafka instances.

Resource	Permission	Account
KI Kafka instance	Allow Describe	All accounts
G Consumer group is "**"	Allow Describe	All accounts
T Topic is "**"	Allow Describe	All accounts
T Topic is "**"	Allow Describe configs	All accounts
KI Kafka instance	Allow Alter	[Trash]
G Consumer group starts with "**"	Allow Read	[Trash]
Add permission	Allow Read	[Trash]
Task-based permissions	Allow Create	[Trash]
Consume from a topic	Allow Describe	[Trash]
Produce to a topic	Allow Read	[Trash]
Manage access		

# Jak przetestować czy działa?

The screenshot shows the Red Hat OpenShift console interface. The left sidebar contains navigation options, with 'Kafka Instances' highlighted. The main content area displays the 'Access' tab for a Kafka instance named 'ak'. A table lists various permissions granted to different accounts.

Account	Permission	Resource
<input type="checkbox"/> All accounts	Allow Describe	KI Kafka instance
<input type="checkbox"/> All accounts	Allow Describe	G Consumer group is "**"
<input type="checkbox"/> All accounts	Allow Describe	T Topic is "**"
<input type="checkbox"/> All accounts	Allow Describe configs	T Topic is "**"
<input type="checkbox"/> svc-acct-6ae0076c-b5b6-405d-alf3-865b0b2280b8	Allow Alter	KI Kafka instance
<input type="checkbox"/> svc-acct-6ae0076c-b5b6-405d-alf3-865b0b2280b8	Allow Read	G Consumer group starts with "**"
<input type="checkbox"/> svc-acct-6ae0076c-b5b6-405d-alf3-865b0b2280b8	Allow Read	G Consumer group is "quarkus-kafka-quickstart"
<input type="checkbox"/> svc-acct-6ae0076c-b5b6-405d-alf3-865b0b2280b8	Allow Create	T Topic is "**"
<input type="checkbox"/> svc-acct-6ae0076c-b5b6-405d-alf3-865b0b2280b8	Allow Describe	T Topic is "**"
<input type="checkbox"/> svc-acct-6ae0076c-b5b6-405d-alf3-865b0b2280b8	Allow Read	T Topic is "**"

# Najprościej uruchomić przykład z dokumentacji

Red Hat OpenShift Streams for Apache Kafka

Product Documentation for Red Hat OpenShift Streams for Apache Kafka

access.redhat.com/documentation/en-us/red\_hat\_openshift\_streams\_for\_apache\_kafka/1

[Configuring topics in Red Hat OpenShift Streams for Apache Kafka](#)  
Verify or edit your Kafka topic properties. Check topics for matching schemas in Red Hat OpenShift Service Registry.

[Managing account access in Red Hat OpenShift Streams for Apache Kafka](#)  
Manage the level of access that other user accounts and service accounts have to your instance.

**Use Kafka instances**

[Binding OpenShift applications to Red Hat OpenShift Streams for Apache Kafka](#)  
Use the Service Binding Operator to automatically configure applications running on OpenShift to connect to Kafka instances.

[Using Node.js applications with Kafka instances in Red Hat OpenShift Streams for Apache Kafka](#)  
Connect Node.js applications to Kafka instances.

[Using Quarkus applications with Kafka instances in Red Hat OpenShift Streams for Apache Kafka](#)  
Connect Quarkus applications to Kafka instances.

[Configuring and connecting Kafka scripts with Red Hat OpenShift Streams for Apache Kafka](#)  
Use Kafka scripts to produce and consume Kafka messages.

[Configuring and connecting Kafkacat with Red Hat OpenShift Streams for Apache Kafka](#)  
Use Kafkacat to produce and consume Kafka messages.

[Monitoring metrics in Red Hat OpenShift Streams for Apache Kafka](#)  
View metrics to visualize the performance and data usage for Kafka instances and topics that you have access to.

# Podążam za dokumentacją...

The screenshot displays the Red Hat CodeReady Studio IDE interface. The main editor shows the `PriceConverter.java` file with the following code:

```
1 package org.acme.kafka;
2
3 import javax.enterprise.context.ApplicationScoped;
4
5 /**
6  * A bean consuming data from the "prices" Kafka topic and applying some conversion.
7  * The result is pushed to the "my-data-stream" stream which is an in-memory stream.
8  */
9
10 @ApplicationScoped
11 public class PriceConverter {
12     private static final double CONVERSION_RATE = 0.88;
13
14     // Consume from the 'prices' channel and produce to the 'my-data-stream' channel
15     @Incoming("prices")
16     @Outgoing("my-data-stream")
17     // Send to all subscribers
18     @Broadcast
19     // Acknowledge the messages before calling this method.
20     @Acknowledgment(Acknowledgment.Strategy.PRE_PROCESSING)
21     public double process(int priceInUsd) {
22         return priceInUsd * CONVERSION_RATE;
23     }
24 }
```

The left sidebar shows the Project Explorer with the following structure:

- quarkus-kafka-quickstart [app-services-guid]
  - src/main/java
    - org.acme.kafka
      - PriceConverter.java
      - PriceGenerator.java
      - PriceResource.java
  - src/main/resources
    - META-INF
    - application.properties
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - mvnw
  - mvnw.cmd
  - pom.xml
  - README.md

The bottom console shows the following output:

```
<terminated> quarkus-kafka-quickstart [Maven Build] /usr/lib/jvm/java-11-openjdk-11.0.14.1-2.el8_5.x86_64/bin/java (Apr 5, 2022, 3:32:39 PM - 3:35:32 PM)
ssl.trustmanager.algorithm = PKIX
ssl.truststore.certificates = null
ssl.truststore.location = null
ssl.truststore.password = null
ssl.truststore.type = JKS
transaction.timeout.ms = 60000
transactional.id = null
value.serializer = class org.apache.kafka.common.serialization.IntegerSerializer

2022-04-05 15:32:48,805 INFO [org.apa.kaf.com.uti.AppInfoParser] (Quarkus Main Thread) Kafka version: 2.8.0
2022-04-05 15:32:48,806 INFO [org.apa.kaf.com.uti.AppInfoParser] (Quarkus Main Thread) Kafka commitId: ebb1d6e21cc92130
2022-04-05 15:32:48,806 INFO [org.apa.kaf.com.uti.AppInfoParser] (Quarkus Main Thread) Kafka startTimeMs: 1649165568805
2022-04-05 15:32:48,959 INFO [io.quarkus] (Quarkus Main Thread) quarkus-kafka-quickstart 1.0-SNAPSHOT on JVM (powered by Quarkus 2.2.1.Final) started in 4.215s. Listening on: h
2022-04-05 15:32:48,959 INFO [io.quarkus] (Quarkus Main Thread) Profile dev activated. Live Coding activated.
```

The right sidebar shows the Properties view for the `PriceConverter` class, with the `CONVERSION` property set to `process(int)`.

Hmm... jest błąd

```
-left: 10px;  
-right: 10px;  
padding-left: 10px;  
padding-right: 10px;  
font-weight: bold;  
font-size: auto;  
  
<div id="login_form" style="color: red; font-weight: bold;">  
<div class="dError1">Authentication Failed</div>  
<div class="dError1">Please contact the administrator for further  
assistance.</div>  
</div>  
  
<saml-auth-status>-1</saml-auth-status>  
  
<script>window.top.location='/php/login.php';return false;</script>
```

# Tak, zapomniałem utworzyć consumer group i dodać uprawnienia

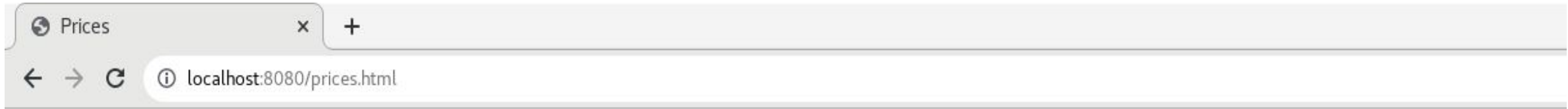
The screenshot shows the Red Hat OpenShift Streams console interface. The browser address bar indicates the URL: `console.redhat.com/application-services/streams/kafkas/c94usk0rg6lbvru2ou2g/consumer-groups`. The console header includes the Red Hat Hybrid Cloud Console logo, navigation options, and the user name "Andrzej Kowalczyk".

The main content area displays the "Consumer groups" tab for a Kafka instance named "ak". The "Consumer groups" tab is circled in red. Below the search bar, a table lists the consumer groups. The "quarkus-kafka-quickstart" group is circled in red. The table has the following columns: "Consumer group ID", "Active members", "Partitions with lag", and "State".

Consumer group ID	Active members	Partitions with lag	State
quarkus-kafka-quickstart	1	0	Empty

At the bottom right of the console, there is a "Feedback" button and a notification icon with the number "8".

I w końcu po 20 minutach ... :)



Last price

The last price is **20.24 €**.



Wszystko trwało 20 minut do uruchomienia kodu.

I ponad godzinę zrobienie zrzutów z ekranu i przygotowanie slajdów :)

# Cloud Services Managed by Red Hat

Managed OpenShift + Application Services + Data Services



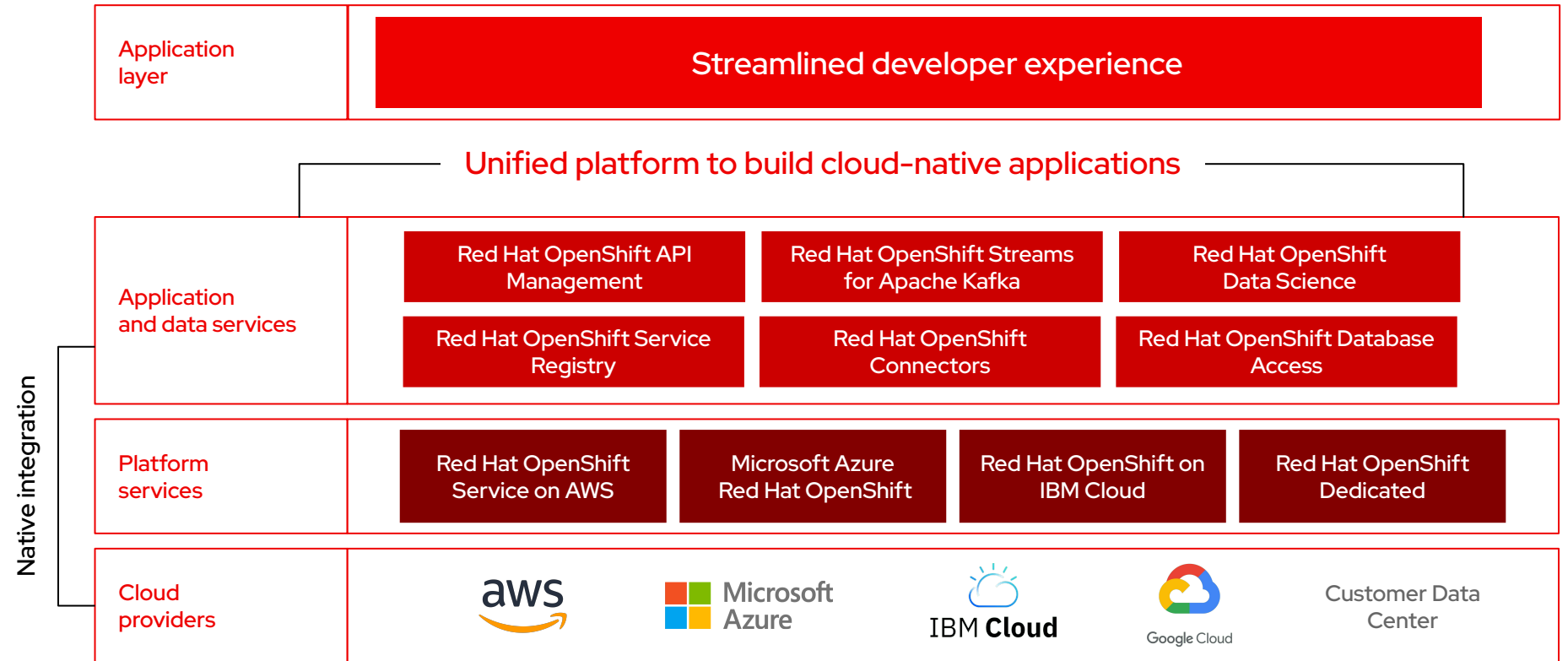
Full stack management and unified experience



Maximize full value of Red Hat OpenShift



Hybrid cloud flexibility





# Red Hat OpenShift Streams for Apache Kafka

**Try Kafka!**

**No cost - no strings attached**

**[red.ht/TryKafka](https://red.ht/TryKafka)**

## Managed Kafka cluster

- Spin up your own Kafka cluster
- Create your topics and its partitions
- Connect your producers and consumers
- Get started with the quick starts
- Integrate your apps to the service

## Time and resource limited

- Access for 48 hours
- Limited number of topics & brokers

## Sign-up

- Go to: [red.ht/TryKafka](https://red.ht/TryKafka)
- Create your own Red Hat account
- Sign-in to try the service

Red Hat  
**Summit**

**Connect**

Thank you



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[twitter.com/RedHat](https://twitter.com/RedHat)