

Mastering Strimzi For Enterprise-Grade Kafka Deployments

Best Practices And Strategies For Scalable And Reliable Kafka Solutions With Strimzi

Executive Summary

This whitepaper guides deploying and managing Apache Kafka using Strimzi in an enterprise environment. Apache Kafka is a powerful distributed streaming platform, but its deployment and management can be complex and challenging, especially at scale. Strimzi simplifies these tasks by offering Kubernetes-native solutions for Kafka operations. This whitepaper addresses enterprises' common challenges with Kafka deployments and presents best practices for leveraging Strimzi to achieve scalable and reliable Kafka solutions.

Key Findings

Complexity of Kafka Deployments:

Deploying and managing Apache Kafka at scale presents significant challenges, including complex configurations, extensive resource management, and high operational costs. Traditional setups are sensitive to misconfigurations and system instability, particularly during scaling and high-availability management.

Strimzi as a Solution:

Strimzi offers a Kubernetes-native approach that simplifies Kafka management. It provides automated setup, scaling, and failover handling, which reduces operational complexity and improves reliability. This integration leverages Kubernetes' container orchestration capabilities to streamline Kafka operations.

Automated Scaling and Management:

Strimzi's dynamic scaling feature adjusts Kafka clusters based on real-time usage, ensuring efficient resource use without manual intervention. Centralized monitoring and logging further enhance operational efficiency, making it easier to manage Kafka clusters.

Enhanced Security and Reliability:

Strimzi includes security features such as TLS encryption and role-based access control, which are crucial for protecting sensitive data. Its high availability and fault tolerance mechanisms, including automatic failover and data replication, ensure continuous operation and data integrity.

Best Practices for Implementation:

Successful Kafka deployments with Strimzi require adherence to best practices, including careful planning of the Kafka setup, regular monitoring, and leveraging Strimzi's automated features for scaling and security. A phased implementation approach, starting with a pilot project, is recommended to validate Strimzi's effectiveness in a controlled environment.

Real-World Use Case:

The case study of [Enexis](#) demonstrates Strimzi's practical benefits, such as simplifying the management of smart meter data streams and improving operational efficiency. Strimzi's integration with Axual's platform has enabled Enexis to scale its data streaming capabilities effectively.

Table Of Contents

04	Introduction – Strimzi
06	Benefits – Strimzi
08	How to implement Strimzi
09	Market Analysis 09 Industry Overview 11 Enterprise Adoption 12 Target audience
14	Competitor Analysis
19	Use Case - ENEXIS
20	Conclusion
22	References
23	Terms and Definitions

Introduction – Strimzi

Background

Apache Kafka is a widely adopted distributed event-streaming platform for building real-time data pipelines and streaming applications. It enables enterprises to process and analyze large volumes of data in real-time, which is critical for modern data-driven decision-making. However, deploying and managing Kafka at scale presents significant challenges. Traditionally, Kafka deployments involve complex configurations, extensive resource management, and ongoing operational maintenance.

Strimzi, an open-source project, offers a Kubernetes-native solution for deploying and managing Kafka clusters. By leveraging Kubernetes' container orchestration capabilities, Strimzi simplifies the Kafka deployment process, providing features like automated scaling, failover handling, and centralized management. This integration addresses many of the operational complexities associated with Kafka, making it a valuable tool for enterprises seeking to optimize their infrastructure.

Challenges and solutions in deploying and managing Kafka clusters

Despite Kafka's benefits, enterprises often face substantial difficulties deploying and managing Kafka clusters. The manual configuration required for scaling, high availability, and operational management can lead to increased complexity, higher operational costs, and potential system instability. Additionally, traditional Kafka setups can be inconvenient to adapt and maintain, especially in dynamic environments that require rapid scaling and high reliability.

The challenge is to find an effective solution that simplifies the deployment and management of Kafka while ensuring the system remains scalable, reliable, and easy to operate. Strimzi presents a promising solution, but organizations need clear guidance on how to implement and leverage it effectively for enterprise-grade deployments.

Introduction – Strimzi

Objectives

The primary objectives of this whitepaper are to:

1. **Identify Common Challenges:** We will highlight the key issues and complexities associated with traditional Kafka deployments, such as scalability, high availability, and operational management.
2. **Demonstrate Strimzi Benefits:** We will explain how Strimzi addresses these challenges by providing a Kubernetes-native approach to Kafka management. And will detail the advantages of using Strimzi, including automated scaling, simplified operations, and enhanced reliability.
3. **Provide Best Practices:** Here are actionable recommendations and best practices for deploying and managing Kafka clusters with Strimzi. This includes strategies for scaling, ensuring high availability, securing Kafka environments, and maintaining operational efficiency.
4. **Guide Successful Implementation:** We will give you an example outline of an implementation roadmap for enterprises looking to adopt Strimzi for their Kafka deployments.
5. **Enhance Decision-Making:** We will share our knowledge on evaluating Strimzi as a solution for your Kafka needs and provide insights on leveraging it to optimize your data streaming infrastructure.

Complexities of Deploying and Managing Apache Kafka in Enterprise Environments

Deploying and managing Apache Kafka in an enterprise environment is inherently complex due to the following reasons:

Initial setup

Setting up a Kafka cluster involves configuring multiple components, such as brokers, zookeepers, and various Kafka settings. This configuration can be intricate and error-prone, especially for large-scale deployments.

Scaling

As data volumes grow, scaling Kafka clusters requires careful adjustment of configuration parameters, resource allocations, and broker load balance. Manual scaling processes can lead to misconfigurations which go hand in hand with downtime.

Benefits

In the fast-paced world of enterprise technology, managing large-scale data systems can often feel like navigating a labyrinth. For many companies, Apache Kafka is at the heart of this challenge.

Managing large-scale data systems poses a significant challenge in the ever-evolving world of enterprise technology. Many organizations rely on Apache Kafka for real-time data processing, but deploying and managing Kafka clusters can be complex and demanding.

Strimzi

Enter Strimzi, an open-source project designed to simplify the deployment and management of Kafka within Kubernetes environments. Strimzi is transforming how organizations handle their Kafka infrastructure, offering several key benefits that address common challenges. Red Hat first released Strimzi in 2017, for the company's OpenShift Kubernetes distribution, as the open source counterpart of AMQ Streams. The Operator pattern, created by CoreOS, provided the framework. CNCF took it on as a sandbox project in 2019. Now Strimzi is officially part of CNCF incubator since 2024.

Automate Kafka setup

One of Strimzi's standout features is its ability to automate Kafka setup. Configuring Kafka traditionally involves a meticulous process with numerous components, each requiring precise settings. Strimzi automates this process, easily configuring Kafka brokers, zookeepers, and related components. This automation reduces the risk of misconfigurations and saves valuable time, allowing IT teams to focus on more strategic tasks.

Kubernetes

Strimzi integrates seamlessly with Kubernetes, a powerful platform for managing containerized applications. Using Kubernetes, you can easily move these containers across different environments, from a developer's laptop to a company's servers. It provides a consistent way to manage these containers, no matter where they run. This integration means managing Kafka clusters becomes part of the existing Kubernetes environment. Organizations benefit from Kubernetes' powerful orchestration capabilities, making Kafka management more streamlined and efficient.

Benefits

Autoscaling

A major advantage of Strimzi is its support for auto-scaling, especially when run on Kubernetes. Organizations often face varying data loads throughout the day, with peak periods requiring additional resources. Strimzi's dynamic scaling feature, coupled with Kubernetes' orchestration capabilities, adjusts Kafka clusters automatically based on real-time usage and traffic patterns. This ensures that the system scales up to handle high demand and scales down during quieter times, optimizing resource use without manual intervention. Kubernetes provides the underlying infrastructure that enhances Strimzi's scalability and resilience, making it an ideal solution for managing Kafka in a cloud-native environment.

Safety first

High availability and fault tolerance are crucial for any data-driven organization. Strimzi addresses these needs with built-in redundancy and failover mechanisms. Strimzi ensures that Kafka clusters remain operational through automatic failover and data replication. Its disaster recovery features also help organizations quickly recover from significant failures, maintaining data integrity and minimizing disruption. Security is a top priority for any organization handling sensitive data. Strimzi enhances security with features like TLS encryption, SASL authentication, and role-based access control. These measures protect data in transit and at rest, ensuring compliance with regulatory requirements and safeguarding against potential breaches.

More efficient

Operational efficiency is greatly improved with Strimzi. The platform provides centralized monitoring and logging, making it easier to track Kafka cluster performance and health. The integration with Kubernetes means that monitoring tools used for other parts of the infrastructure can also be applied to Kafka, offering a unified view of the entire system.

How to implement Strimzi

In short you can follow these steps in the order provided to install Strimzi, and start sending and receiving messages from a topic:

1 Ensure you have the required prerequisites.

Minikube must be installed and you need to be able to access the Strimzi GitHub.

2 Install and start Minikube.

Minikube is local Kubernetes, focusing on making it easy to learn and develop for Kubernetes. All you need is Docker to start.

3 Install Strimzi.

Install Strimzi with the Custom Resource Definitions (CRDs) and RBAC configuration required for deployment.

4 Create a Kafka cluster.

With Strimzi installed, you create a Kafka cluster, then a topic within the cluster. When you create a cluster, the Cluster Operator you deployed when installing Strimzi watches for new Kafka resources.

5 Access the Kafka cluster to send and receive messages.

You can test your Strimzi installation by sending and receiving messages to my-topic from outside the cluster. Use a terminal to run a Kafka producer and consumer on a local machine.

Strimzi Quickstart Guide. online Available at: <https://strimzi.io/docs/quickstart/0.16.2/>

Market Analysis - Industry Overview

To understand the dynamic landscape of data streaming and real-time analytics, it is essential to explore the key trends and data driving the industry's growth and innovation.

Industry Overview

The data streaming and real-time analytics industry is rapidly evolving as organizations increasingly rely on real-time data to drive decision-making, enhance customer experiences, and gain competitive advantages. This overview highlights key trends and data relevant to the industry:

1. Growing Demand for Real-Time Data Processing

Adoption Rate: Businesses across various sectors are embracing real-time data processing to handle the surge in data generated from IoT devices, social media, and online transactions. According to a MarketsandMarkets report, the global real-time streaming analytics market size was valued at \$15.4 billion in 2021 and is projected to grow \$ to 50.1 Billion at a compound annual growth rate (CAGR) of 26.5% from 2021 to 2026. ([Streaming Analytics Market - Global Forecast to 2027](#))

2. Emergence of Event-Streaming Platforms

Event Streaming: Platforms like Apache Kafka have gained prominence for their ability to handle high-throughput, low-latency data streams. The adoption of event-streaming architectures is becoming common in industries such as finance, healthcare, and retail.

Market Growth: The event-streaming market, driven by Apache Kafka and similar technologies, is expected to expand significantly. As per a report by Grand View Research, the market for event stream processing was valued at \$7.8 billion in 2022 and is anticipated to reach \$26.8 billion by 2030, growing at a CAGR of 16.8%.

3. Kubernetes and Container Orchestration

Integration with Kafka: Kubernetes has become a critical component in managing containerized applications, including Kafka clusters. The integration of Kafka with Kubernetes, facilitated by projects like Strimzi, simplifies the deployment, scaling, and management of Kafka clusters.

Market Adoption: According to the Cloud Native Computing Foundation (CNCF), Kubernetes adoption has grown dramatically, with over 80% of enterprises using Kubernetes for container orchestration as of 2024. This growth is driving the need for solutions that integrate seamlessly with Kubernetes.

Market Analysis - Industry Overview

4. Focus on Automation and Efficiency

Automation Trends: There is a strong emphasis on automating the management of data infrastructure to reduce manual intervention and errors. Solutions like Strimzi automate Kafka setup and scaling, addressing the operational challenges associated with managing large-scale data systems.

Operational Efficiency: Companies are investing in automation tools to enhance operational efficiency and reduce the total cost of ownership. The rise of Infrastructure as Code (IaC) and continuous integration/continuous deployment (CI/CD) practices is a testament to this trend.

5. Emphasis on Security and Compliance

Data Security: As data breaches and cyber threats become more prevalent, ensuring data security and compliance is a top priority. Solutions like Strimzi provide built-in security features such as TLS encryption and role-based access control to safeguard data.

Regulatory Compliance: Organizations are increasingly required to comply with stringent data protection regulations such as GDPR and CCPA. The need for secure and compliant data management solutions is driving the adoption of advanced security features in data streaming platforms.

6. Competitive Landscape

Key Players: The market is characterized by a mix of established players like Confluent, which offers a comprehensive Kafka-based platform, and newer entrants like Axual, which leverages Strimzi for Kubernetes-based deployments. Aiven is also notable for its fully managed open-source data infrastructure.

Differentiation: Companies differentiate themselves through features such as ease of use, integration capabilities, customer support, and advanced functionality. For instance, Axual emphasizes user experience and ease of use, while Confluent focuses on advanced features and a rich ecosystem.

Market Analysis - Enterprise Adoption

Real-world use cases demonstrate the value of data streaming solutions. For example, Enexis successfully implemented Axual's platform with Strimzi to streamline smart metering data processing, showcasing the benefits of real-time data management in practical scenarios.

In summary, the real-time data streaming industry is growing rapidly, driven by the need for timely data processing, the integration of event-streaming platforms with Kubernetes, and a focus on automation and security. The competitive landscape features a mix of established and emerging players, each offering unique solutions to address the evolving needs of businesses in various sectors.



Competitor Analysis

Several key players dominate the Kafka market. Here, we focus on Aiven, Confluent and Axual due to their relevance.

Aiven

Aiven provides fully managed open-source data infrastructure as a service, supporting technologies such as Apache Kafka, PostgreSQL, and OpenSearch. While Aiven does not use Strimzi, it offers a managed Kafka service that abstracts away the complexities of Kafka management, allowing users to leverage Kafka's capabilities without needing to manage the underlying infrastructure directly.

Confluent

Confluent is a major player in the event streaming space, primarily built around Apache Kafka. It focuses on enabling real-time data streaming and integration across various systems. Unlike Axual, Confluent does not use Strimzi; instead, it offers its own managed Kafka service, incorporating a significant amount of proprietary code to deliver enterprise-grade features and capabilities. This approach includes an ecosystem designed to meet the needs of complex, large-scale deployments in the cloud.

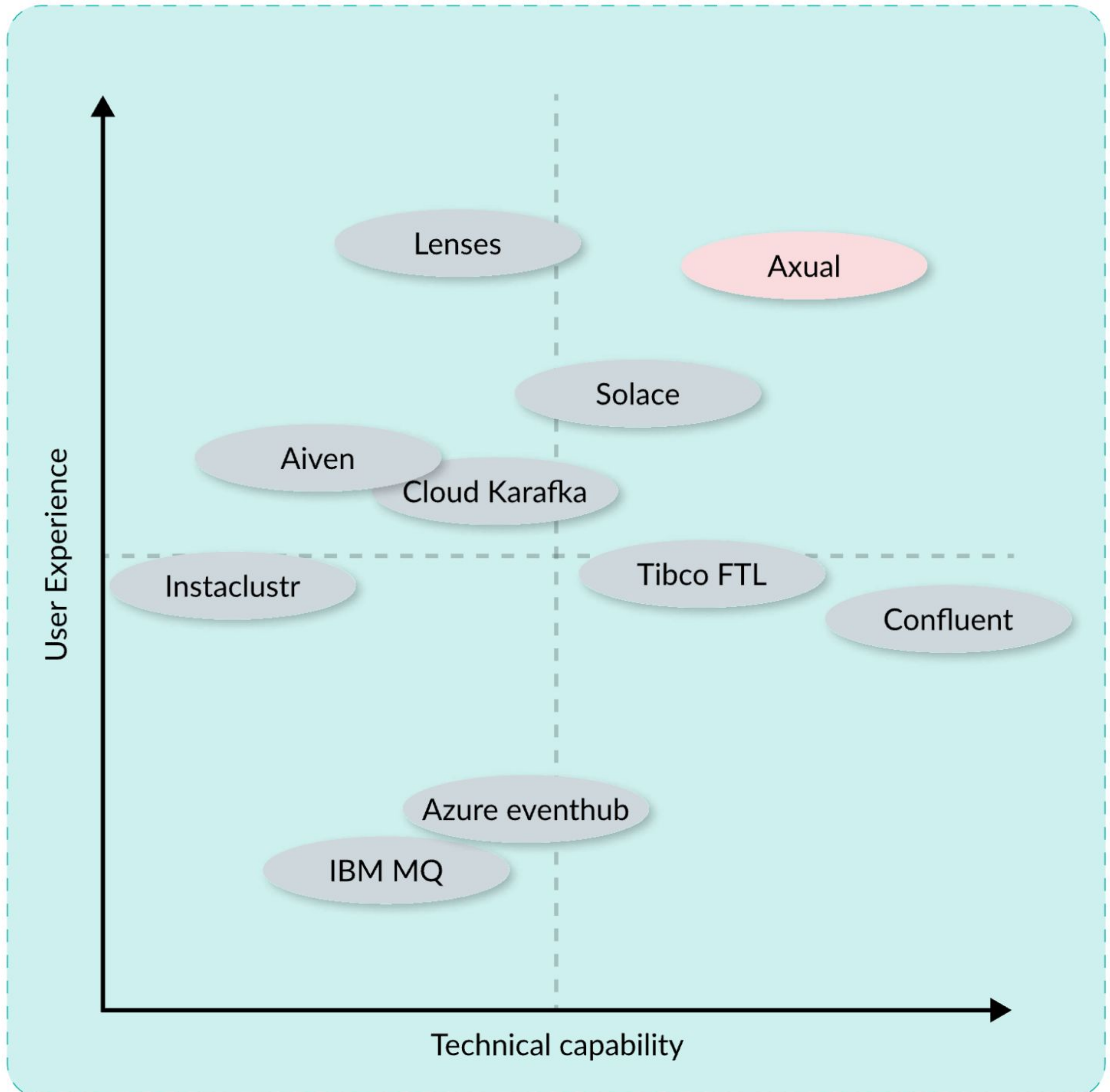
Axual

Axual offers an out-of-the-box Kafka solution. The platform combines the data streaming capabilities of Strimzi, an open-source Kafka on Kubernetes framework, with a user-centric interface that enables large organizations to utilize Apache Kafka with enterprise features such as access control, data governance and a visual way to see where data is flowing. The goal of the platform is to take the complexity out of Kafka so organizations can focus on building out their use case instead of maintaining Apache Kafka.

Competitor Analysis

A simplified comparison

This graph, designed by Axual, provides a clear overview of the system's performance and scaling capabilities.



Jeroen van Disseldorp, Axual. (2023). Graph of Kafka Performance Metrics.

Competitor Analysis

Feature/aspect	Aiven	Confluent	Axual
Main products	Platform to Stream, Store and Serve	Confluent Cloud, Confluent Platform	Axual Governance, Axual Streaming Available on On-premise, Cloud and Hybrid
Market Positioning	Positioned as a flexible provider of fully managed open-source data infrastructure, supporting a range of technologies including Apache Kafka, with a focus on multi cloud deployments. Focussing more on user experience and ease of use.	Fully managed Apache Kafka platform with advanced features for real-time data processing. Confluent scores high on technical capability.	Axual specializes in real-time data streaming with a customer-centric approach, providing managed Apache Kafka services and leveraging Strimzi for Kubernetes-based deployments. With a focus on larger organizations.
Customer base	Serves a diverse customer base, ranging from startups to large enterprises.	Has a wide-ranging customer base with a strong presence among large enterprises and tech companies.	Axual focuses on a customer base that includes medium to large enterprises looking for real-time data streaming solutions.
Strengths	Designed for simplicity, with a user-friendly interface and automation features.	Leadership in event streaming with a rich ecosystem and integrations.	Rich architecture, with enterprise comes with support and governance on top of existing Strimzi environments.
Weaknesses	Limitations in use, because some parts are not as customizable and easy.	Although built on Apache Kafka, Confluent's proprietary extensions and features could create a lock-in effect, making it harder for customers to switch to other solutions or revert to open-source Kafka.	While Axual excels in providing a User-friendly platform that simplifies real-time data streaming and integration, details regarding configuration are abstracted away for the end-user.

Methodology

This comparison of Aiven, Confluent, Axual is based on extracting detailed information from company websites to understand their services, features, and market positions. Key data sources included industry reports, expert insights, and customer reviews on G2, which were analyzed to gauge user satisfaction and service effectiveness. By combining these approaches, we aimed to deliver a thorough evaluation of each company's offerings and their standing in the real-time data streaming and management market.

Competitor Analysis - Results

Findings

Aiven excels in providing fully managed open-source data infrastructure with multicloud support, focusing on technologies like Apache Kafka, PostgreSQL, and OpenSearch.

Confluent stands out as a leader in event streaming with its fully managed platform, offering advanced features and a rich ecosystem for enterprise-grade data processing.

Axual differentiates itself with a customer-centric approach, leveraging Strimzi for Kubernetes-based deployments and offering managed Kafka services designed for ease of use and scalability.

Implications

These findings suggest that organizations should carefully consider their specific needs, such as the desired level of management, and existing infrastructure, when choosing between these solutions. Aiven may be ideal for those valuing open-source flexibility and multicloud support, Confluent for enterprises requiring comprehensive, advanced capabilities, and Axual for those prioritizing ease of use and customer support with Kubernetes integration where Strimzi serves as a valuable tool for Kubernetes-centric environments.

Use Case - ENEXIS

Enexis, a Dutch Distribution System Operator (DSO), manages the electricity and gas grids for over 3 million households and businesses. In 2019, several projects at Enexis required data streaming capabilities, prompting them to explore solutions. They chose Axual's platform to implement real-time data streaming, starting with a use case centered on smart metering. Axual's out-of-the-box Kafka solution, combined with Strimzi, simplifies the use of Apache Kafka with features like access control and data governance. This allowed Enexis to manage the lifecycle of smart meters, process data, and connect with various technologies without needing deep technical knowledge.

Previously, collecting data from smart meters was labor-intensive and lacked standardization, causing inefficiencies. The Axual platform provided a scalable, controllable, and cost-effective data streaming solution. It enabled efficient data collection, processing, and distribution to relevant teams. Axual also supported Enexis by providing training and infrastructure setup to facilitate the adoption of Kafka. As a result, Enexis has scaled the platform internally, supporting over 70 streams, 60 applications, and 150 developers. This collaboration has improved Enexis's data management and operational efficiency significantly.

Read the full use cas



Conclusion

Recap of the Problem

Deploying and managing Apache Kafka in enterprise environments can be complex, involving setups, manual scaling, and ongoing maintenance. These challenges can lead to increased operational costs, misconfigurations, and system instability.

Solution

Strimzi offers a Kubernetes-native solution that simplifies Kafka deployments by automating setup, scaling, and management processes. This whitepaper outlines how Strimzi addresses these challenges, providing best practices for achieving scalable, reliable, and efficient Kafka solutions.

Benefits

Using Strimzi with Kubernetes, enterprises can automate Kafka configurations, enhance resource optimization through auto-scaling, ensure high availability and fault tolerance, and improve security and operational efficiency. This enables IT teams to focus on strategic tasks while maintaining a data streaming infrastructure.

Next Steps

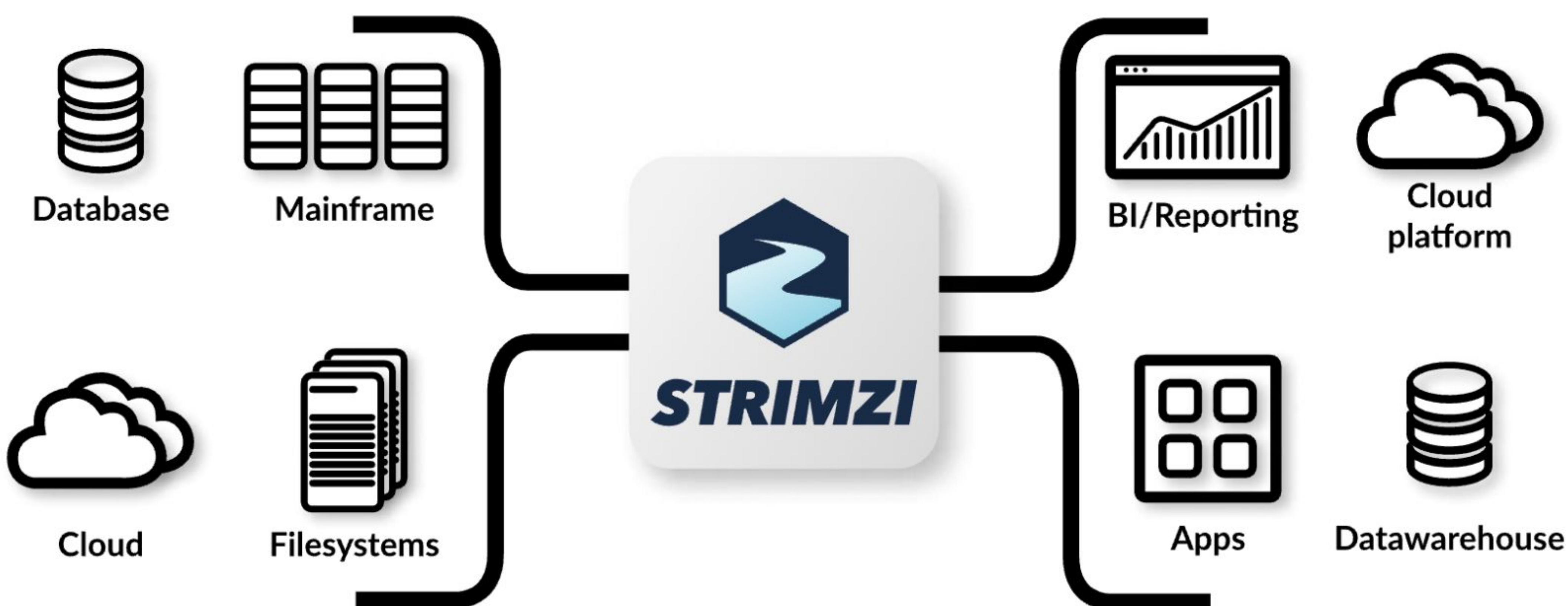
This is how you can embark on an exciting journey to upgrade your Kafka.

1. Evaluate your current Kafka setup: Assess the complexity and challenges your organization faces with its existing Kafka deployments.
2. Explore Strimzi: Learn more about how Strimzi can simplify Kafka management by integrating with Kubernetes.
3. Implement best practices: Follow the best practices outlined in this whitepaper to deploy and manage Kafka clusters using Strimzi.
4. Plan a pilot project: Initiate a pilot project to test Strimzi in a controlled environment before a full-scale rollout.
5. Leverage Strimzi features: Utilize automated scaling, enhanced security, and centralized monitoring to optimize your Kafka operations.
6. Engage with experts: Consider engaging with Strimzi and Kubernetes experts to guide the implementation process and provide training for your team.

We from Axual have a solution for enterprises seeking to optimize their real-time data streaming solutions with Kafka on Kubernetes. Want to know more about it? Get in contact with our Strimzi experts.

Conclusion

In conclusion, mastering Strimzi for enterprise-grade Kafka deployments presents a transformative approach for managing large-scale data systems within Kubernetes environments. This whitepaper has illustrated how Strimzi simplifies the complexities of deploying and managing Apache Kafka, addressing critical challenges such as scalability, high availability, and operational efficiency. By leveraging Strimzi's automated setup, dynamic scaling, and security features, enterprises can significantly reduce the operational burden and enhance the reliability of their Kafka clusters. Additionally, the integration with Kubernetes allows organizations to streamline their infrastructure management, aligning Kafka operations with modern container orchestration practices. The successful implementation and benefits observed in real-world use cases, such as that of Enexis, underscore Strimzi's potential to optimize data streaming infrastructures. Enterprises looking to adopt or improve their Kafka deployments can think of using Strimzi to lead their business to more scalable, reliable, and cost-effective solutions.



[Request a Demo](#)

References

Research Methods

Company Websites: Information was extracted from the official websites of Aiven, Confluent, Axual to gather details on their services, features, and market positions.

Data Sources

Customer Reviews and Testimonials: Reviews and testimonials from current and former users of Aiven, Confluent, and Axual were analyzed to assess user satisfaction and service effectiveness.

List of references

Axual. (n.d.). Real-Time Data Streaming with Axual. Retrieved from <https://axual.com/>

CNCF. (n.d.). Strimzi: Kafka on Kubernetes. Retrieved from <https://www.cncf.io/projects/strimzi>

CoreOS. (n.d.). Operator Framework. Retrieved from <https://coreos.com/operators>

Enexis. (n.d.). Enexis Case Study: Real-Time Data Streaming with Axual and Strimzi. Retrieved from <https://axual.com/case-studies/enexis>

Minikube. (n.d.). Minikube: Run Kubernetes Locally. Retrieved from <https://minikube.sigs.k8s.io/docs/>

Red Hat. (n.d.). OpenShift: Kubernetes Platform for the Enterprise. Retrieved from <https://www.openshift.com>

MarketsandMarkets, n.d. Streaming Analytics Market - Global Forecast to 2027. Retrieved from https://www.marketsandmarkets.com/Market-Reports/streaming-analytics-market-64196229.html?gad_source=1&gclid=CjwKCAjwko21BhAPEiwAwfaQCKzKuJMbw6lBmJ50bc6tMe0qxFAMv6tloRLG0ckX8slunTYSI1hX5RoCYXEQAvD_BwE

Strimzi. (n.d.). Strimzi Quickstart Guide. Retrieved from <https://strimzi.io/docs/quickstart/0.16.2/>

The New Stack. (n.d.). CNCF-Approved Strimzi Eases Apache Kafka onto K8s Clusters. Retrieved from <https://thenewstack.io/>

Aiven. (n.d.). Managed Open-Source Data Infrastructure. Retrieved from <https://aiven.io/>

G2. (n.d.). Aiven Reviews. Retrieved from <https://www.g2.com/products/aiven/reviews>

G2. (n.d.). Confluent Reviews. Retrieved from <https://www.g2.com/products/confluent/reviews>

G2. (n.d.). Axual Reviews. Retrieved from <https://www.g2.com/products/axual/reviews>

Confluent. (n.d.). Confluent: Real-Time Data Streaming Platform. Retrieved from <https://www.confluent.io/>

Terms and Definitions

Apache Kafka

A distributed streaming platform used for building real-time data pipelines and streaming applications, enabling enterprises to process and analyze large volumes of data in real-time.

Auto-scaling

The capability of a system to automatically adjust its resources based on current demand, ensuring optimal performance and resource utilization without manual intervention.

Broker

A Kafka server that stores and serves data to clients. It handles the messages sent to Kafka topics and is responsible for maintaining the data stream.

Centralized Monitoring

A system that provides a unified view of the performance and health of various components within an infrastructure, allowing for easier management and troubleshooting.

Cluster

A group of servers or nodes working together as a single system to ensure high availability and scalability.

Container Orchestration

The automated process of managing and coordinating the deployment, scaling, and operation of application containers.

CoreOS

A company known for developing technologies for managing containerized applications, including the Operator pattern used by Strimzi.

Data Governance

The management of data availability, usability, integrity, and security within an organization, ensuring that data is handled properly and meets compliance requirements.

DevOps

A set of practices that combine software development (Dev) and IT operations (Ops), aiming to shorten the development lifecycle and deliver high-quality software continuously.

Disaster Recovery

Strategies and mechanisms for recovering data and maintaining system functionality after a catastrophic event.

Distributed Streaming Platform

A system that allows for the continuous ingestion, processing, and analysis of real-time data across distributed components.

Enexis

A Dutch Distribution System Operator (DSO) that manages electricity and gas grids, utilizing Axual's platform for real-time data streaming to manage smart meters.

Failover Handling

Mechanisms ensuring that a system continues to operate by switching to a standby system when the primary system fails.

High Availability (HA)

A system design approach ensuring operational performance and availability at a high level, even during failures.

Kubernetes

An open-source platform for managing containerized applications across multiple hosts, providing automated deployment, scaling, and operations.

Minikube

A tool that runs a single-node Kubernetes cluster inside a virtual machine on a local machine, used for development and testing.

Operator Pattern

A Kubernetes-native approach to managing complex applications by extending the Kubernetes API and providing application-specific controllers.

Red Hat OpenShift

A Kubernetes distribution from Red Hat that provides an enterprise-grade platform for building and managing containerized applications.

Role-Based Access Control (RBAC)

A method of regulating access to computer or network resources based on the roles of individual users within an enterprise.

Smart Metering

The use of digital meters that record energy consumption and communicate the information to the energy provider for monitoring and billing.

Strimzi

An open-source project providing a Kubernetes-native solution for deploying and managing Apache Kafka clusters, simplifying operations with features like automated scaling and failover handling.

Zookeeper

A centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services, used by Kafka for cluster management.