TECH mahindra

Whitepaper

Serverless Development Architecture to Handle Streaming Event on Microsoft Azure Cloud Platform

Abstract

Traditionally most of the organization try to handle streaming live data with different datatypes and high volume. Analyse Ingested data for valuable insights that are not available through traditional data storage and analysis. Confluent Kafka on Azure emanates finest technology for in stream processing platform and fault tolerant messaging system to roar. Primary goal is to provide confluent Kafka on Azure as well as to deploy and manage Kafka clusters more easily with a fully managed and secure service.

Key Takeaways

The key areas covered in this whitepaper as follows.

01

Realtime Metric Collection

02

Cloud-Based Event Streaming

03

Streamlined Resource Management

04

Seamless Network Integration

05

Public Cloud Kafka Services

06

Private Link Kafka

07

Unified Sign-On

08

Topic Specific Metrics

09

Enhanced Security and Compliance

Introduction

Every company uses heterogeneous data sources that are in different data formats. So, when we plan data integration, we combine data from these different sources into an integrated and unified repository. However, due to their enormous interdependence and complex connectivity, these different integrations can pose a real and unique challenge for us.

Confluent Kafka as a service business model. The customer runs applications in multiple data centres in Europe, Asia, and the United States. Applications want lower latency and higher throughput when interacting with Kafka clusters. To achieve high availability and low latency, set up a separate cluster in each location and use Confluent Replicator technology to keep clusters synchronised for selected topics for each application. Applications can connect to a closer cluster to achieve low latency. In the event of a disaster, administrators can switch the application to the other cluster.

Confluent Kafka with Azure not only promises to help us decouple these differentiated data streams and systems, but it also allows us the connectivity we need with the Microsoft cloud, pleasantly solving this complex conundrum..

Trends

There are many business sectors using event streaming with Confluent Kafka in the automotive, banking, manufacturing, and logistics industries. Logistics ecosystems are expected to prioritize operational efficiency and significant impact on the movement of goods. Infrastructure and deployment will vary by use case. Confluent Kafka can effectively process analytical and transactional data in motion. The following are trends in event data.

- **Cloud native**
- **Data sharing in real time**
- **Advanced data governance and policies**

Challenges

Data Integration: Integrating event streaming into an existing infrastructure can be complex. To ensure that data from multiple sources, such as databases, microservices, and external systems can be efficiently ingested into event streaming platform.

Data Loss Prevention: Guaranteeing data integrity and preventing data loss is crucial - Azure Kafka's replication and fault tolerance features ensure data durability and resilience against failures.

Data Decoupling: Many applications suffer from tight coupling with data producers and consumers. Azure Kafka decouples these components reducing dependencies and allowing for more modular and flexible systems.

Data Security: Securing data in transit and at rest is paramount.

Data Serialization: Managing data serialization formats like Avro, JSON, and handling schema evolution can be challenging. As your data schemas evolve over time, you need to ensure backward and forward. Compatibility to avoid breaking existing consumers.

Message Order: It can be difficult to maintain the order of messages within a stream, especially in a distributed environment. It is not always possible to guarantee a strict order for all events.

Duplication of Events: Events can be duplicated due to network problems, retries, or system failures. Implementing mechanisms to detect and deduplicate events is critical to ensuring data consistency.

Solutions and Benefits

Data motion is a new paradigm of data infrastructure. We have many events processing engine is in the market like Rabbit MQ, Logstash, Boomi, Cloudera, IBM MQ, Amazon Kinesis, Snowflake, Apache Kafka, Apache stream, and so on.

We provided a solution for Confluent Kafka on Azure because it is a truly cloud-native service and a complete stack for leveraging data in motion that goes far beyond Apache Kafka.

Confluent Kafka on Azure is designed to enable new data infrastructures. It is a fully managed cloud service running on Azure cloud. The Confluent platform can be deployed as self-managed software on premise or in the cloud (private/public) and can be deployed on Kubernetes.

Confluent Kafka on Azure provides encryption, authentication, and authorization mechanisms to safeguard sensitive data.

Logistics companies have many global applications to process shipment data and the applications run in multiple data centres. The applications want lower latency and higher throughput when interacting with Kafka. We have provided an automation framework to set up a dedicated cluster and use replicator technology to keep clusters in sync for selected topics for each application.

SaaS has a special offering for users.

- **Pay-per-use pricing model.**
    - **Subscription-based cost component**
    - **Usage-based cost component**
- **Capacity management**
- **No downtime for upgrades**

System Architecture

This app modernization solution has been designed and built following Microsoft best practice guidelines and Microsoft Azure well-architected framework standards. The main goal of this service is to provide Kafka services in the public cloud – simplifying the deployment and management of Kafka clusters with a fully managed and secure Kafka service. Availability zones are a high availability offering from Microsoft that protects applications and data from data centre outages. Zones are unique physical locations within an Azure region. Each zone consists of one or more data centres equipped with independent power, cooling, and networking. To ensure resiliency, there is always more than one zone in all zoned regions. By physically separating the availability zones within a region, applications and data are protected from data centre failures. CKC clusters can be deployed in multiple availability zones.

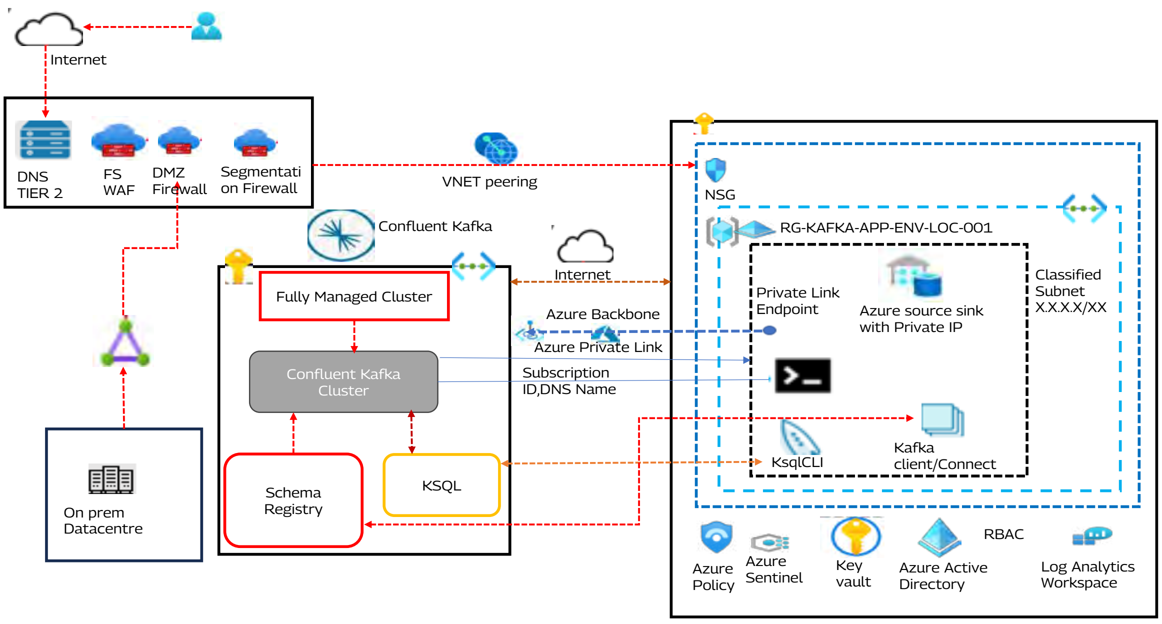

Figure 1: System Architecture

Confluent Kafka on Azure clusters in production would only be accessible through the customer's Azure network. If an on-site server needs to access the Confluent clusters, additional network routing and firewall configuration may be required. Confluent cloud via the Confluent portal, the DNS name is provided by Confluent and should be added via the Azure portal subscription.

The following ports are required.

kafka - 9092

HTTPS endpoints – SR

KSQL – 443

Description of the Architecture

Schema registry: Schema registry is linked to the environment. All clusters in the same environments use the same schema registry. Note that the schema registry is a regional resource. so, if the customer chooses a region for a particular schema registration, all clusters. at all sites in the same environment use the region-based schema registration. Latency should not be an issue in most use cases.

Clusters: Each cluster is a logical Apache Kafka cluster. The clusters have their own topics. Each cluster can have 0 or more ksqlDB clusters. Each cluster can have 0 or more connectors. Cluster is a geolocated resource.

ksqlDB: ksqlDB runs SQL-like queries on the stream of events to extract data.

Connector: The connector integrates with upstream applications (source connector) and downstream applications (sink connector). The source connector transfers data from external systems into Kafka and the sink connector transfers data from Kafka to external systems.

Message: A message is an application data unit. Each message may contain headers, metadata, and user data in arbitrary bytes. Messages can be encoded and compressed in various formats. compressed by using different compression codecs.

Partition: Partition is a logical FIFO queue for messages. Each partition can contain a number of messages.

Topic: The topic is the logical stream of events. Each topic can have one or more partitions.

Application Integration

In a shared cluster, each application should have its own private prefix. The prefix is identified by the application prefix. For example, Tech01 may have a namespace with the prefix “Tech01-”. Similarly, consumer group names and transaction names ID used by Tech01 have the same prefix.

An example table illustrating the access matrix by namespace (prefix) is as follows:

Type	Application	Resource
TOPIC	Tech01	Tech01-TOPIC1 Tech01-TOPIC2
Consumer Group	Tech01	Tech01-Consumer-Group-01 Tech01-Consumer-Group-02 Tech01-Consumer-Group-03
Transactional ID	Tech01	Tech01-Transaction-ID1 Tech01-Transaction-ID2
TOPIC	Tech02	Tech02-Topic
Consumer Group	Tech02	Tech02-Consumer-Group-01
Transactional ID	Tech02	Tech02-Transaction-ID1

The above configuration can be easily managed by granting ACL for service accounts with prefixes. Granting can be done at the application deployment stage in the self-service portal after proper approval. After successful onboarding, each application engineer/devOps will receive a common set of configuration parameters for each environment as listed below:

Configuration Parameter	Java Config Key	Comments
Bootstrap Server	bootstrap. Servers	Endpoint for Kafka cluster
Kafka Access Key	Username	Authentication credential (key)
Kafka Access Key Secret	Password	Authentication credential (secret)
Schema Registry URL	schema.registry.url	Endpoint to the schema registry
Schema Registry Access Key	basic.auth.user.info	HTTPS basic authentication credential to schema registry
Schema Registry Access Key Secret	basic.auth.user.info	HTTPS basic authentication credential to schema registry
Topic Management	-	Topics to be created, updated, deleted in self-service portal. Topic configuration might be changed in the self-service portal or by admin upon request.
Transactional ID	transactional.id	Self-decided with the specified prefix
Consumer Group ID	group.id	Self-decided with the specified prefix

With these parameters, clients can produce into and consume from the shared Kafka cluster. Typically, it is not necessary for an application engineer to connect to the Confluent cloud console to perform application integration.

In certain circumstances, an application (Tech03) may need to access a topic that is not in scope (without a prefix of "APP03-"). When APP03 attempts to access the topic's data using the Tech03 credentials, the application receives an error message indicating that authorization failed. In this case, a specific ACL may be granted on an ad hoc basis. The ACL is managed by the end-user administrator in their own dedicated cluster.

If the customer wishes to deploy Kafka clusters dedicated to specific applications only, they can deploy such clusters in the appropriate environment and assign cluster administrators to the application owner. API keys management and connection parameters query can be done directly by the cluster administrators.

Single Sign on (SSO) via Azure to Confluent Cloud

Confluent Kafka on Azure allows for single sign-on (SSO) using an existing SAML-based identity provider. With SSO, enterprise users can log in to multiple, unrelated systems using a single user ID and password, which means enterprises no longer must store and manage passwords in Confluent Kafka on Azure platform. SSO also improves security and decreases service and troubleshooting issues associated with individual logins.

SSO can only be configured by organization admins. SSO configuration is a onetime task and requires configuring both Confluent Kafka on Azure settings and Idp (e.g., Azure active directory application) for federation settings.

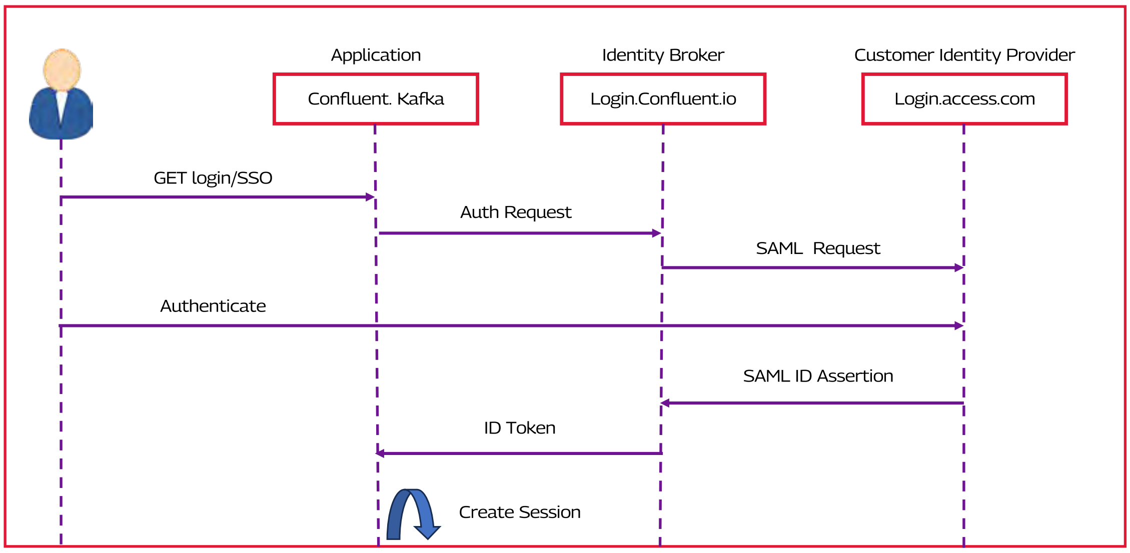

Figure 2: Confluent Kafka on Azure SSO Flow

When an employee joins or leaves the company, credentials are immediately provisioned or revoked according to AD user management. However, role access management is still required to enable the user to perform all necessary administrative tasks in the Confluent Kafka on Azure

RBAC access Matrix

Enforce granular controls over application access and management of topics and consumer groups with Kafka ACLS

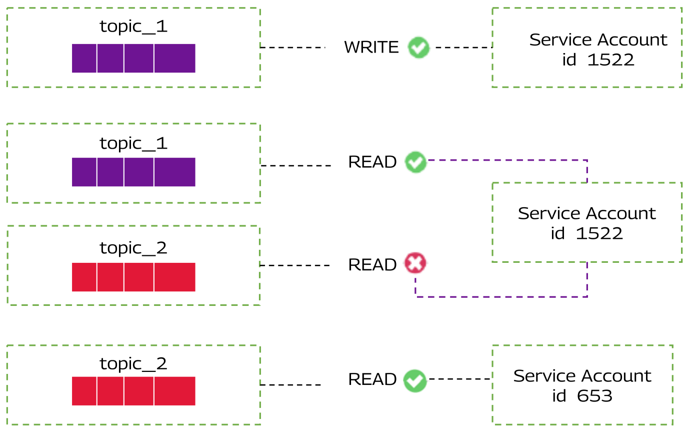

Figure 3: Management of Topics

From the above picture, service account ID 1522 does not have read access for topic_2 and can access only topic_1. RBAC roles, will be able to provide granular access to specific topics with Confluent cloud RBAC.

Audit Event Management

Audit logs to detect abnormal behaviour and meet compliance such as capturing logs in dedicated topics and offload to external systems. Audit logs can be viewed through the Kafka-console-consumer or via imported to the running Kafka cluster and then exported to external systems. The audit events are JSON documents that can be parsed, understood, and monitored with the SPLUNK.

There are some types of events that may be relevant for audit logging to monitor.

Successful authentication to a Kafka cluster with an API key. The user:abc123 is managed with an API key, which could be a service account, or global access key.

Monitoring and Alert using Splunk

Confluent Kafka offers monitoring metrics and can configure threshold for alerting for different metrics.

Metrics	Comments
io.Confluent.kafka.server/received_bytes	To monitor topic level ingress
io.Confluent.kafka.server/sent_bytes	To monitor topic level egress
io.Confluent.kafka.server/retained_bytes	To monitor topic level storage.
io.Confluent.kafka.server/partition_count	To monitor Partition count by cluster
io.Confluent.kafka.ksql/streaming_unit_count	To monitor Streaming unit count
io.Confluent.kafka.schema_registry/schema_count	To monitor Number of schemas registered

Conclusion

There are many challenges when using streaming data, especially when handling offsets. If an error occurs in the middle of the process, the data is lost. The offset must be explicitly maintained in our code or a separate table.

But now Confluent Kafka provides all the facilities, and the retention period is maintained for the theme so that data loss can be avoided.

Confluent Kafka offers multiple methods of data recovery, and the infrastructure is maintained by both the cloud provider and the Confluent cloud team and is therefore considered a highly available and replicated system.

About The Authors

Guru Prasad C P

Group Practice Head, Microsoft Business Unit, Tech Mahindra

Guru Prasad C P has an experience of over 24 years with more than 8 years specifically in the public cloud working in Asia, ANZ, Europe, and the US. His experience includes, setting up practice teams aligned to industry verticals and horizontals, analyst interactions for positioning the offerings, hiring the right talent, involving in strategic exercise mergers and acquisitions, organization building, creating frameworks, and IP's.

At Tech Mahindra he is responsible for practice and competency development which includes alignment with OEMs for solutions, offerings and adoption of new technologies, customer interfacing where he acts as a trusted advisor in providing unbiased views/opinions and aligning with organization goals at the same time, value creation, developing practice areas deal making, solution support for large deals, and carve out deals from azure and hybrid cloud perspective.

M Rajashekar Reddy

Solution Architect, Cloud (App Mod, Integrations, Data&AI),
Microsoft Business Unit, Tech Mahindra

M Rajashekar Reddy is a seasoned Multi-Cloud Architect with 17 years of expertise in product management, pre-sales, and technical architecture. He has worked with clients in diverse industries like aerospace, healthcare, insurance, energy, oil and gas, telecom, semiconductors, geospatial, and transportation. His skills encompass Azure, AWS, GCP, OCI, integrations, data, ai, automation, technical program management, delivery leadership, product management, pre-sales, proposal management, and agile project management.

At Tech Mahindra, he collaborates with alliances to bring technology best practices, show-casing the benefits of cloud architecture through cost reduction, increased productivity, faster turnaround times, high availability, scalability, resilience, performance, and security. He focuses on developing in-house capabilities for supporting new offerings, implementing cloud integration and cloud performance engineering best practices, and providing advisory services for cloud native and non-native approaches. He also develops business cases and joint GTM plans with partners, creating collaterals and reusable artefacts across various domains and infrastructures.

Jegatha Antony

Solution Architect (Data&AI), Microsoft Business Unit, Tech Mahindra

Jegatha Antony has 15 years of experience in product development and architecture of projects in banking, logistics and telecommunications. Her expertise extends to Big Data technologies like Spark, Kafka. She is currently working on solutions that leverage the public cloud AWS and Azure.

At Tech Mahindra, her primary focus is on helping customers implement and scale new services using Azure AI, data platform, ML and develop POCs, document best practices. She is very instrumental in the implementation of comprehensive data governance frameworks for reengineering initiatives.

TECH mahindra

2023 Brand Finance® Awards TOP 10 STRONGEST IT SERVICES BRAND

2023 Brand Finance® Awards FASTEST-GROWING IT SERVICES BRAND IN BRAND VALUE RANK