Apache Kafka

- Conceptos Fundamentales de Apache Kafka
- Instalación y Configuración
- Gestión del Clúster de Kafka
- Proyectos con Cliente Python
- Estructuras de Datos
- Visualización de Datos Real-Time en Power BI

¿Qué es Apache Kafka?

- Apache Kafka es una plataforma de transmisión de datos de código abierto diseñada para **gestionar flujos masivos de información en tiempo real**.
- Es utilizado por miles de **empresas**, incluidas más del 80% de las Fortune 100.
- Existe una gran variedad de **casos de uso empresariales** en los que adoptar Kafka en los que adoptar Kafka como solución de ingesta y procesamiento de eventos en tiempo real.

Casos de uso

- Procesamiento de registros en tiempo real. (Real-Time Data Streaming).
- Integración de sistemas (Event-Driven Architectures).
- Monitoreo y analítica en tiempo real.
- Gestión de datos de IoT.
- Pipeline de datos para Data Lakes y Data Warehouses.
- Sistemas de recomendación.
- Procesamiento de pagos y transacciones.

Análisis de datos en tiempo real

- Los **datos** en tiempo real son aquellos **que se generan, procesan y analizan de forma instantánea y sin necesidad de almacenarlos previamente**. Estos datos permiten obtener **información** valiosa **sobre el comportamiento de los clientes, rendimiento, los procesos, tendencias de mercado y oportunidades de negocio** casi **al mismo tiempo que se obtienen los datos**.

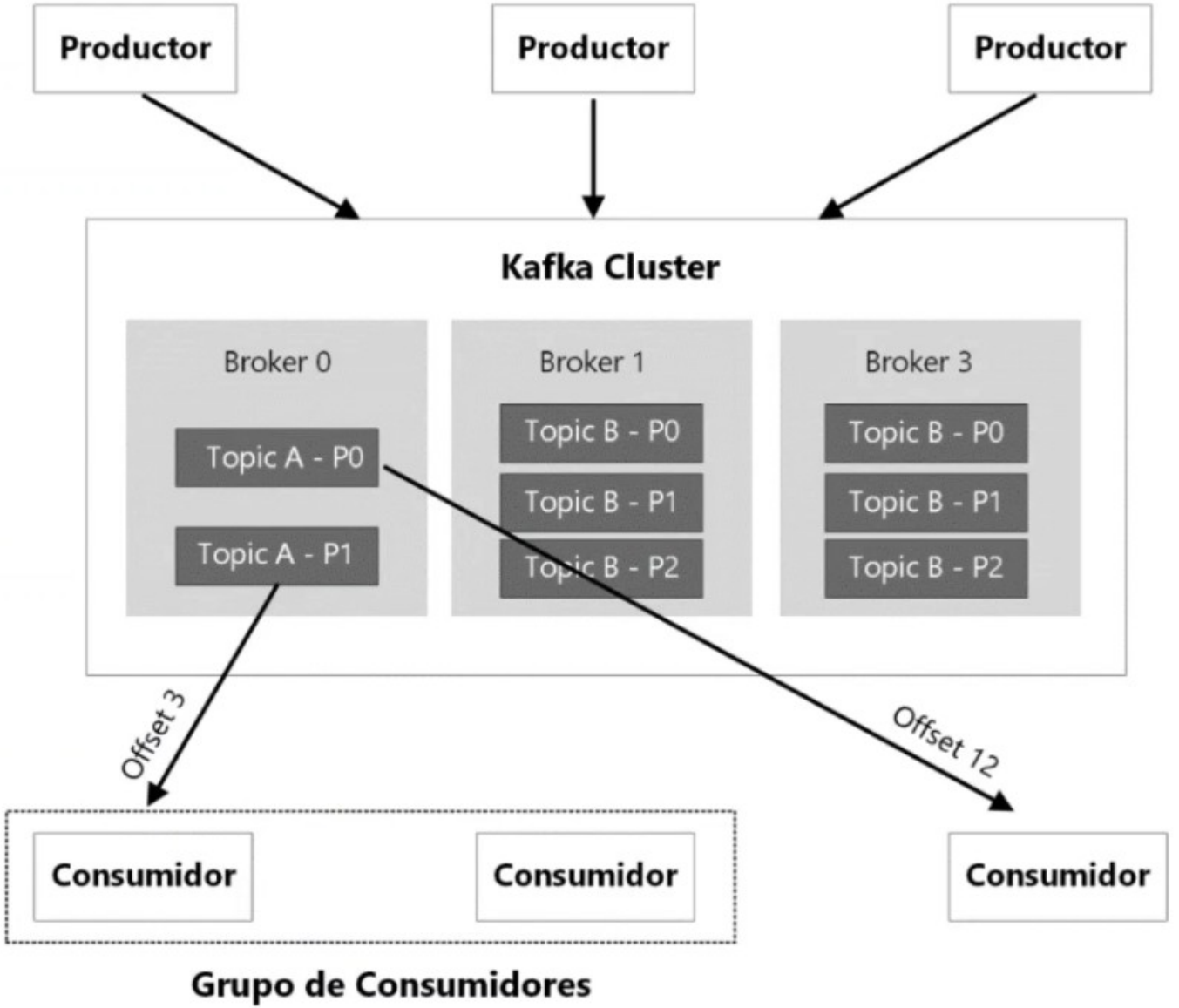
Beneficios del análisis de datos en *real-time*

- Satisfacción del cliente.
- Agilidad en la toma de decisiones.
- Optimizar procesos y recursos.
- Descubrir nuevas oportunidades.

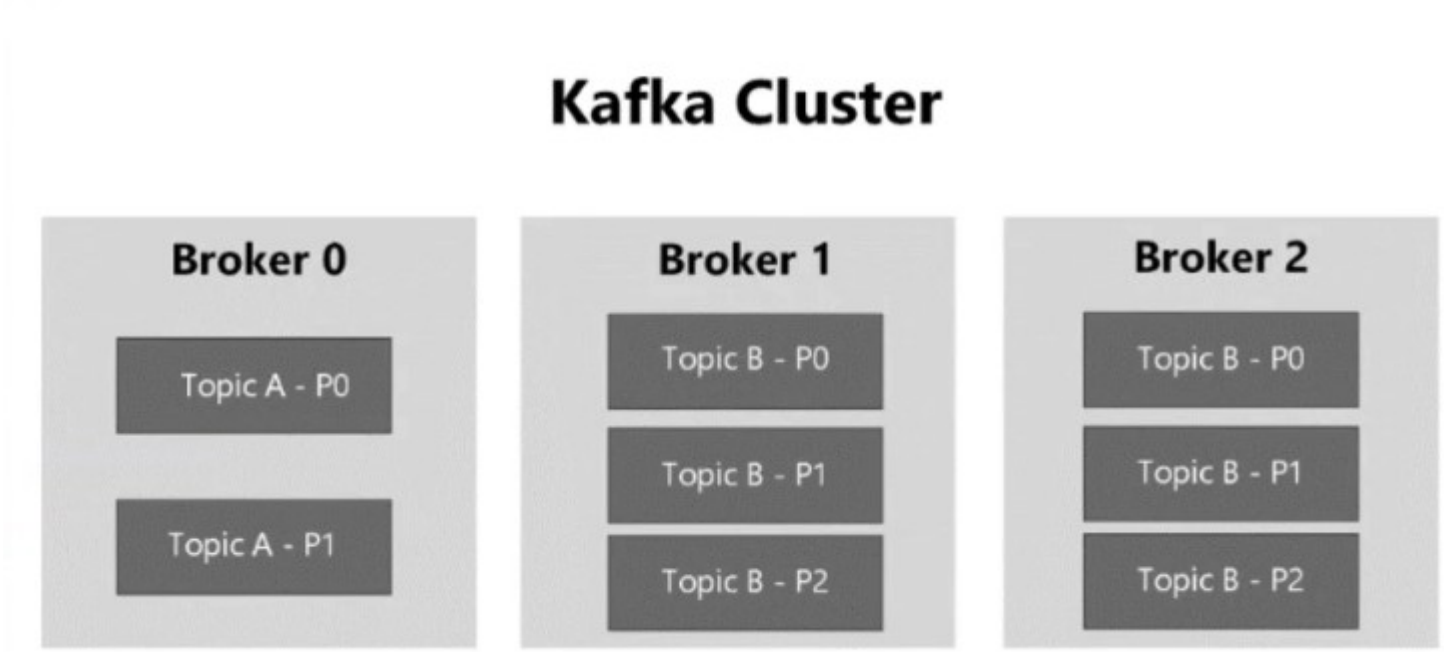
Características principales

- Arquitectura distribuida.
- Escalabilidad.
- Latencia.
- Durabilidad y Retención de Datos. Data replication.
- Data Streaming. Kafka stream => Real time.
- Ecosistema Extensible con otras tecnologías.
- Monitorización y Administración del propio Kafka.

- Arquitectura
- ~~Zookeeper~~
- KRaft



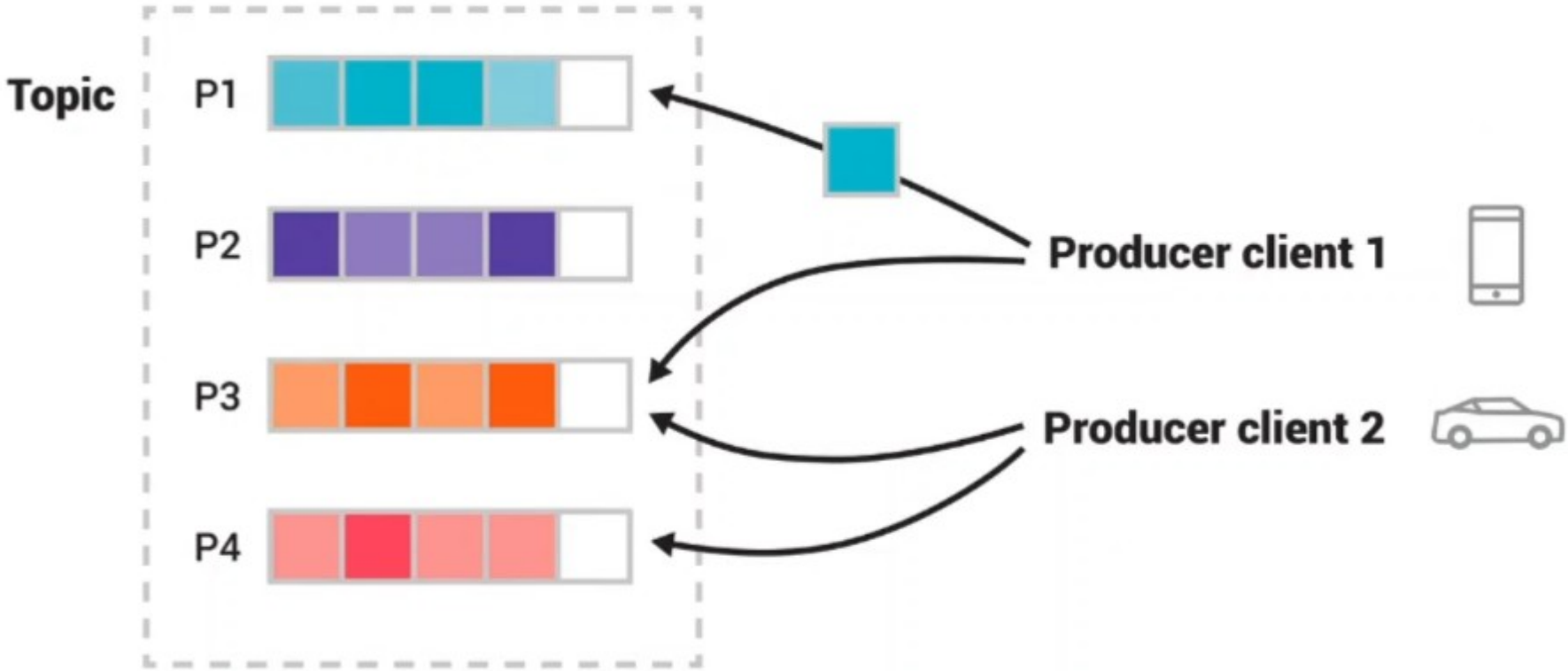
Cluster, Brokers y Servidores



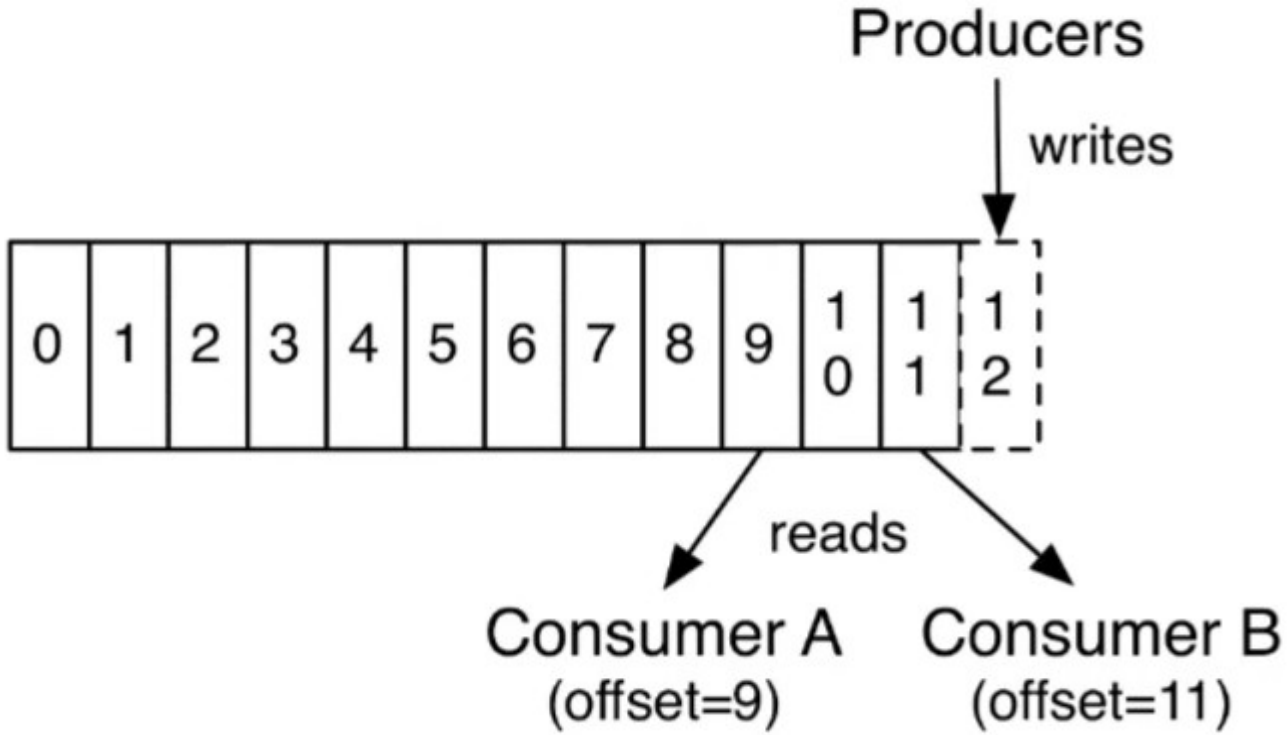
Topics, Particiones y Offsets

- ***Topic***: Es una categoría o canal de datos que permite la organización y clasificación de los mensajes.
- **Partición**: Cada partición es tratada como un registro de *log* ordenado y proporciona una grado adicional de paralelismo al permitir que los *brokers* gestionen diferentes fragmentos de flujo de datos de un *topic*.
- ***Offset***: Identificador único asignado a cada mensaje dentro de la partición de un *topic*.

Topics, Particiones y Offsets

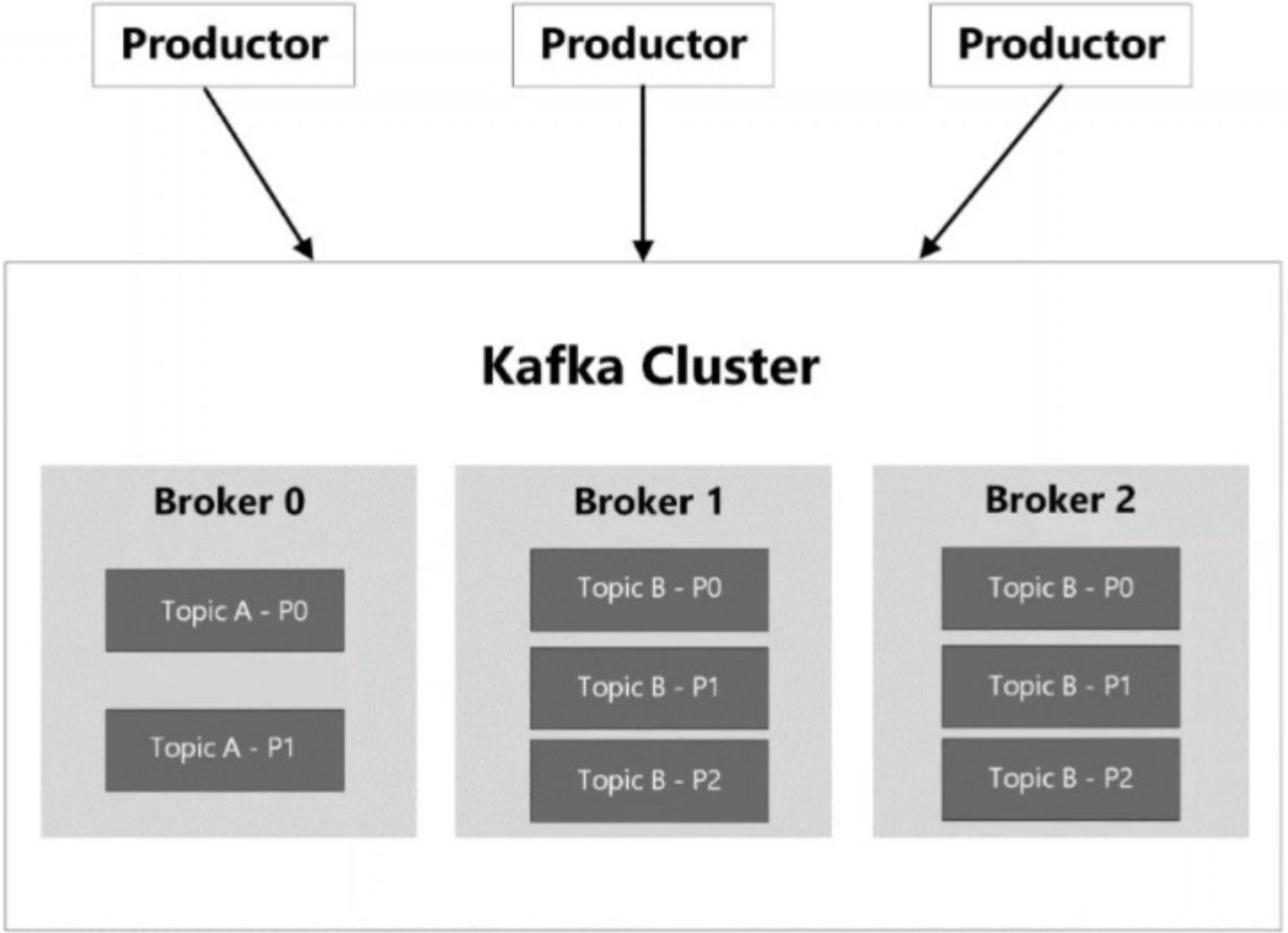


Topics, Particiones y Offsets



Productores

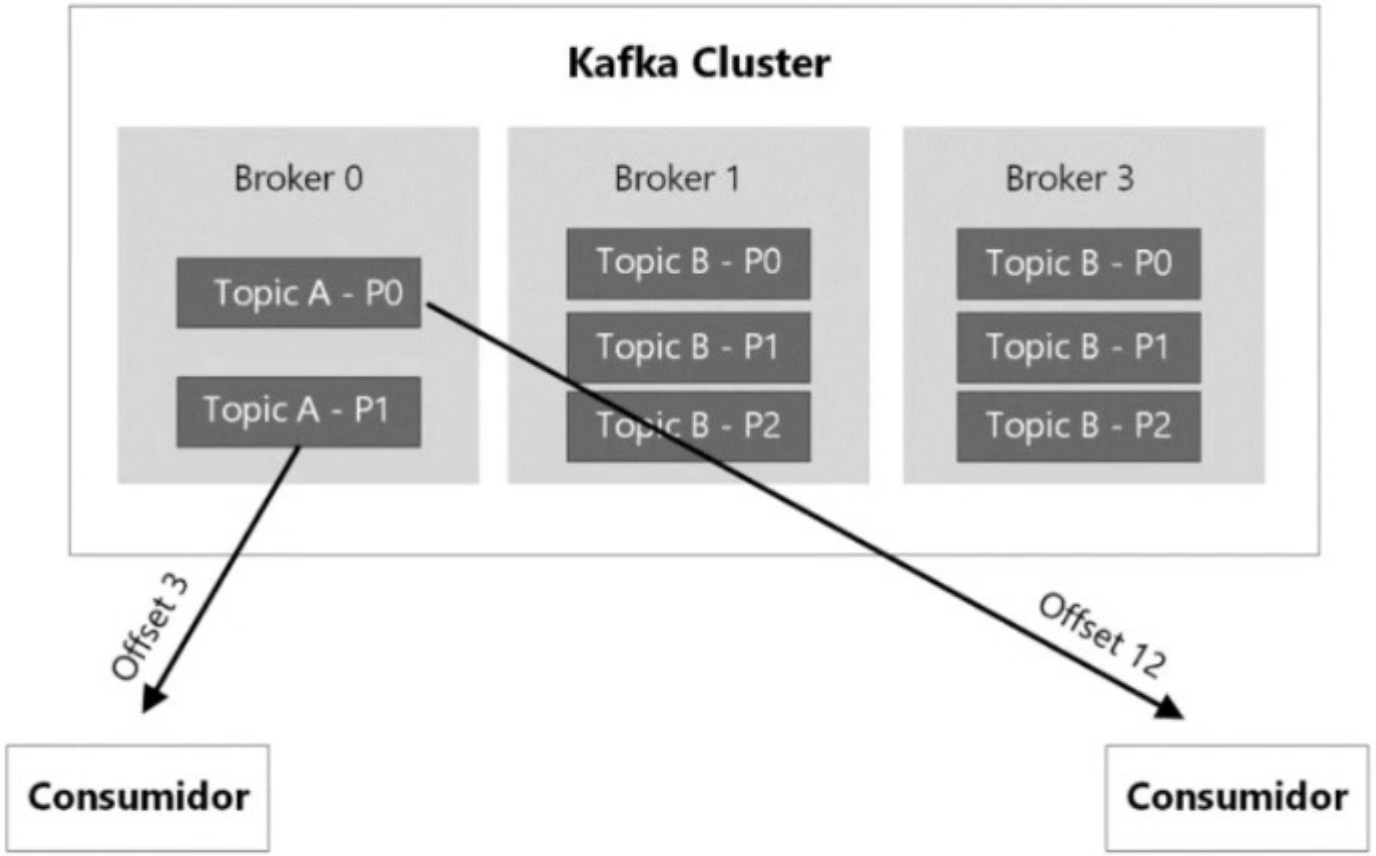
- **Transfieren mensajes** a los *topics* que correspondan del ***cluster* de Kafka**.
- La flexibilidad de los productores en cuanto a la diversidad de datos que pueden manejar es clave para su **versatilidad** en distintos escenarios de uso.


Productor
Productor
Productor
Kafka Cluster
Broker 0
Topic A - P0
Topic A - P1
Broker 1
Topic B - P0
Topic B - P1
Topic B - P2
Broker 2
Topic B - P0
Topic B - P1
Topic B - P2

Consumidores

- La función principal de los consumidores radica en suscribirse a uno o varios *topics* específicos y **procesar los mensajes** que fluyen a través de ellos.
- A medida que los productores generan mensajes y los envían a los *topics* correspondientes, los consumidores están a la espera, listos para recibir y utilizar la **información** de manera **inmediata**.
- Esta dinámica de suscripción y procesamiento en tiempo real es crucial para escenarios donde la **latencia mínima** es esencial.

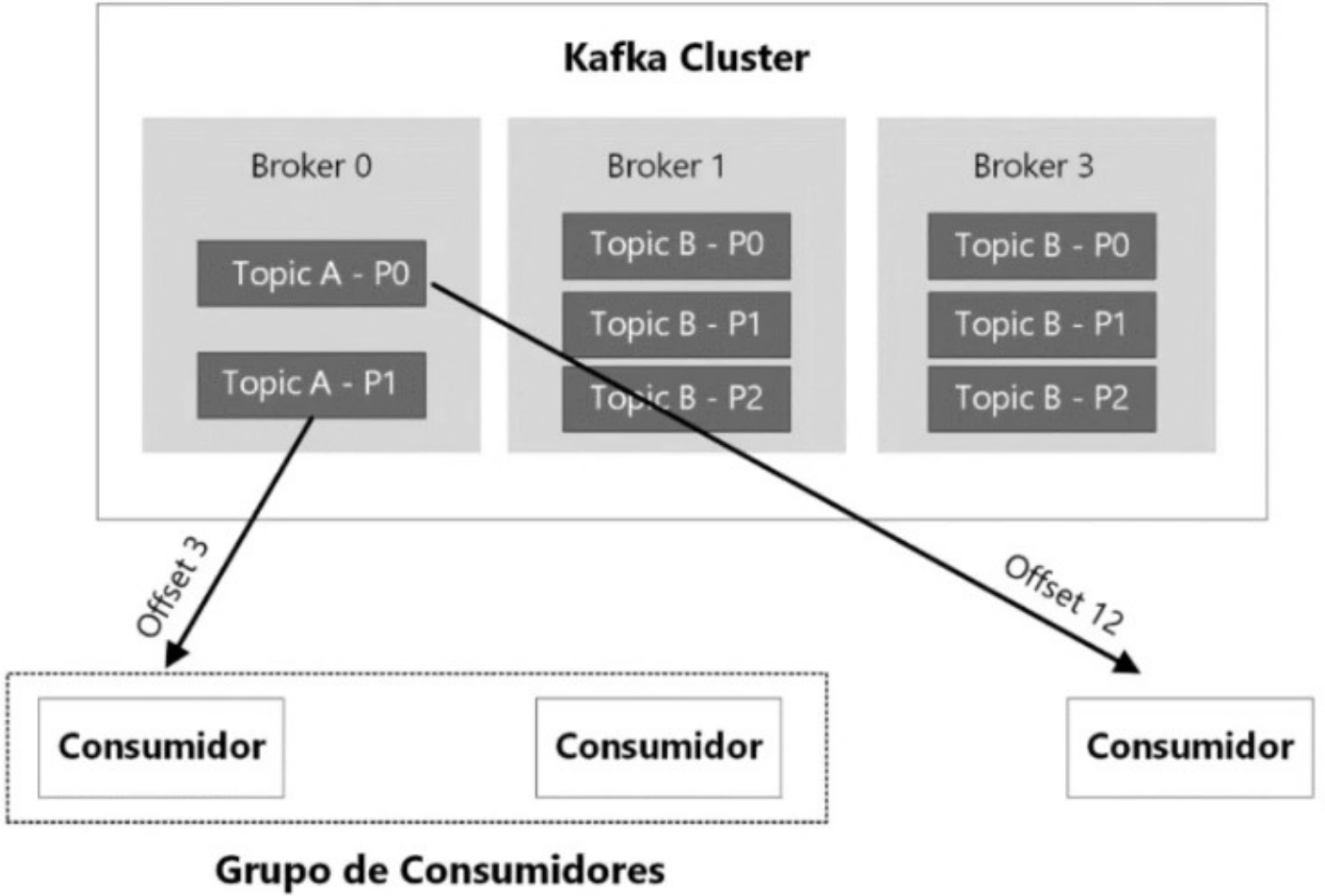
Consumidores



Grupos de Consumidores

- Múltiples consumidores se agrupan para conseguir una mayor **escalabilidad y una gestión eficiente** de la carga.
- Los topics pueden ser consumidos por **varios grupos simultáneamente**, permitiendo una distribución paralela de la carga de procesamiento.
- Proporcionan mayor **flexibilidad** y adaptación a errores puntuales del entorno.

Grupos de Consumidores



Instalación

- https://kafka.apache.org/downloads
- Desactivar IPv6
  - sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
  - sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
- Instalar Java
  - sudo apt update; apt install -y default-jdk
  - java --version
- Descargar Kafka, extraerlo y mover al directorio raíz:
  - wget https://dlcdn.apache.org/kafka/3.9.0/kafka_2.13-3.9.0.tgz
  - tar -xzvf kafka_2.13-3.9.0.tgz
  - mv kafka_2.13-3.9.0 /
- Configurar PATH en el usuario NO root. Editar .bashrc y añadir al final el PATH
  - nano ~/.bashrc
  - PATH=”$PATH:/kafka_2.13-3.9.0/bin”

Kraft vs Zookeeper

- Diferencias Fundamentales:
  - KRaft: Kafka Raft, Protocolo replicación en incorporado en Kafka.
  - ZooKeeper: Servicio externo para estado y coordinación.
- Ventajas de KRaft:
  - Simplifica configuración.
  - Elimina dependencia de ZooKeeper.
  - Enfoque especifico para Kafka.
- Práctico y Eficiente:
  - Configuración sencilla con KRaft.
  - Inicio directo de Kafka sin ZooKeper.
  - Mejora en escalabilidad y tolerancia a fallos.
- Decisión Estratégica:
  - Transición a KRaft desde la versión 2.8.
  - KRaft listo para producción en versión 3.3.1
  - Facilita la gestión y prepara para el futuro.