

พอสด

สามารถใช้ service bus แทนได้

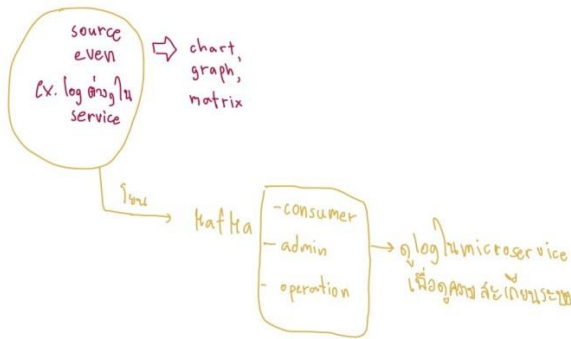
Kafka - message broker  
Java, scala ฯลฯ

- message, even, data = ข้อมูลที่ส่งต่อ
- Kafka เป็น Broker = รับส่งหรือกระจายข้อมูล (ตัวกลาง)
- Architect สามารถทำ high to put, low latency = จัด data เข้าไปได้โดยอัตโนมัติความถี่สูง
- Kafka สามารถ scale ให้รองรับการเก็บได้บน storage ได้
- Kafka สามารถทำมิม redundancy เก็บหลาย copy ได้ ถ้าจุดไหนหนึ่งสามารถหา data จากอีกจุดแทนได้ ทำให้ data ปลอดภัย
- เหมาะสำหรับการรับ ปริมาณข้อมูลที่มีแนวโน้มที่มียุคมากขึ้นตาม module ได้

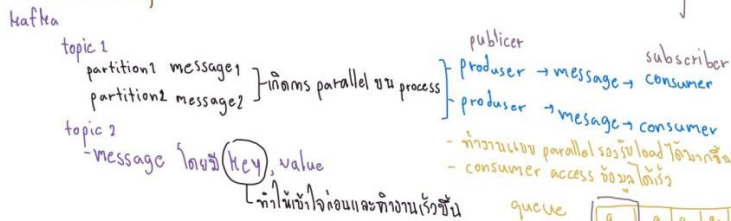
ปกติโครงสร้างของมททำงาน



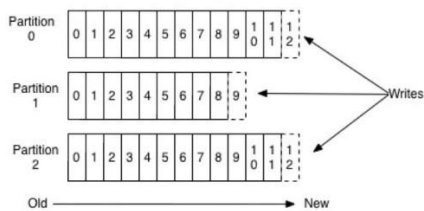
\* even เกิดมาต่อเนื่องกันเป็นลำดับ = even streaming  
 - ทุกสิ่งที่ user ทำ  
 - ex. website realtime activity tracking



โครงสร้างคร่าวๆ



Anatomy of a Topic



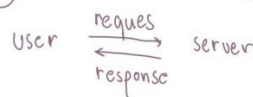
ข้อดีของ Kafka

- Kafka ไม่ขึ้นกับ software แบบ sequential
- Kafka ไม่ขึ้นกับ even deviant แบบคลัสเตอร์อื่น  
ex. 1 even ที่สามารถเกิดพร้อมกันได้
- ms design ไม่เหมือน even แบบอื่น เช่น swift ไม่ทำงานพร้อมกัน
- ขึ้นอยู่กับ rabbit mq ใช้สเกลในเซิร์ฟเวอร์  
ถ้าเกิด low latency, too put สูง, handle realtime data ได้

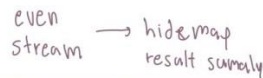
1. มี even ex. มรคณิก, การเข้าข้อมูล

2. ปริมาณข้อมูลทั้งหมด

ปกติขอเรียกใช้ function



Kafka ขอเรียกใช้ function



Kafka ทำงานเป็น cluster ของ kafka broker รวมกัน

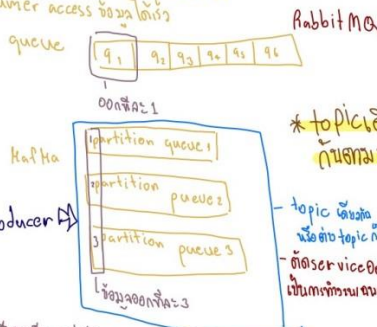
- ตัวที่ทำงานเรียกว่า Kafka broker เก็บข้อมูลแบบกระจายโดย ms copy

cluster 1 A, B, C  
 cluster 2 B, D, E  
 เป็นมิมของกันมทของ Broker

topic เก็บข้อมูลแบบที่ตัวส่งกับ โดยไป topic จะรับรอบคต

message = Data ก่อนข้อมูล ที่จะไปเข้า topic

มีมททำงาน คือ producer -> topic -> consumer



\* topic เดียวกันทำใน 1 ms partition queue ทำงานพร้อมกันตามลำดับที่ขอมูลเอง

topic ต้องมี  
 ข้อดีของ topic คือ  
 - ตัว service บนคลัสเตอร์  
 เป็นมททำงานแบบ

ข้อดี

1. ทำงานเร็ว, Data ปลอดภัย
2. จัดแนว flow ได้ชัดเจนและแบ่งแต่ละ flow (topic) ทำหน้าที่ของตัวเองได้  
ซึ่งเป็น message broker

Kafka เป็น lib ในภาษา set up ได้

เป็น open source ใช้กับ aws ใช้ manage service มี MSK

Amazon managed streaming kafka -> Apache Kafka