



# Level up in die Cloud: Wer streamt, gewinnt!

\* Evi Schneider, CEMEA Senior Marketing Managerin

\* Carsten Mütlitz, Solution Engineering Consultant

\* Jan Svoboda, Solution Engineering Consultant

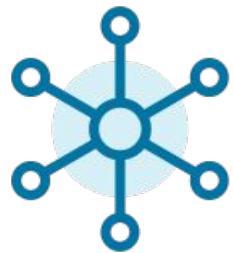
Suvad Sahovic, Solution Engineering Consultant



**Microservices**



**Cloud**



**Internet of  
Things**



**Machine  
Learning**

# Confluent Platform



DEVELOPER	OPERATOR	ARCHITECT	EXECUTIVE
<b>Unrestricted Developer Productivity</b>	<b>Efficient Operations at Scale</b>	<b>Production-stage Prerequisites</b>	<b>Partnership for Business Success</b>
<b>Multi-language Development</b> Non-Java Clients   REST Proxy Admin REST APIs	<b>GUI-driven Mgmt &amp; Monitoring</b> Control Center   Proactive Support	<b>Enterprise-grade Security</b> RBAC   Secrets   Audit Logs	<b>Complete Engagement Model</b>
<b>Rich Pre-built Ecosystem</b> Connectors   Hub   Schema Registry	<b>Flexible DevOps Automation</b> Operator   Ansible	<b>Data Compatibility</b> Schema Registry   Schema Validation	<b>Revenue / Cost / Risk Impact</b>
<b>Event Streaming Database</b> ksqldb	<b>Dynamic Performance &amp; Elasticity</b> Self-Balancing Clusters   Tiered Storage	<b>Global Resilience</b> Multi-Region Clusters   Replicator Cluster Linking	<b>TCO / ROI</b>

## Apache Kafka

Open Source | Community licensed



**Self-managed Software**

**Freedom of Choice**



**Fully Managed Cloud Service**



**Enterprise Support**



**Professional Services**

**Committer-driven Expertise**



**Training**



**Partners**

# Flexible Deployment Options - Today focus ccloud



Cloud-native SaaS

## Confluent Cloud

Apache Kafka Re-engineered  
for the Cloud



Available on the leading public clouds



Self-Managed Software

## Confluent Platform

The Enterprise Distribution of  
Apache Kafka



Deploy on any platform, on-prem or cloud

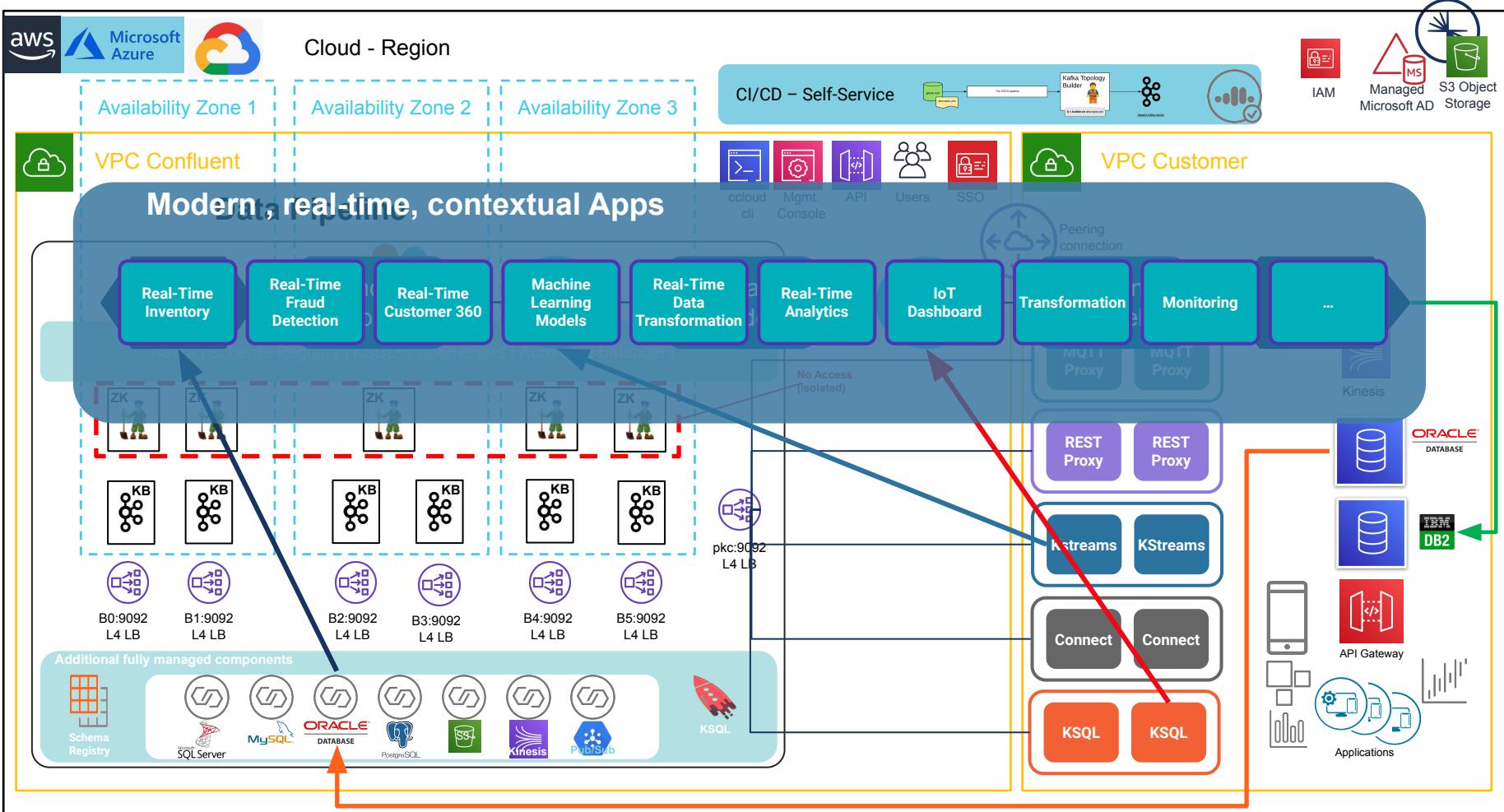


# Start streaming with Kafka within minutes



	<b>Basic</b>	<b>Standard</b>	<b>Dedicated</b>
Sizing	Get started with scale to \$0 pricing	Production-ready for most applications	Customizable for any application
Replication options	No sizing required Stream up to 100MBps Store up to 5TB	No sizing required Stream up to 100MBps Store up to 5TB	Limits based on provisioned capacity
Uptime SLAs	Single AZ	Single & Multi AZ	Single & Multi AZ
Private networking options	99.5%	99.95%	99.95%
<b>Ideal for</b>	Prototyping, early development, and early production use cases	Production use cases streaming below 100MBps	Mission-critical applications at any scale

**Mix and match any cluster type across your organization**



**Die Webinar Serie basiert auf**

<https://github.com/ora0600/apache-kafka-as-a-service-by-confluent>



Level 1:  Apache Kafka

# Tutorial: Apache Kafka as a Service - Der Einstieg

Carsten Mütlitz, Solution Engineering Consultant

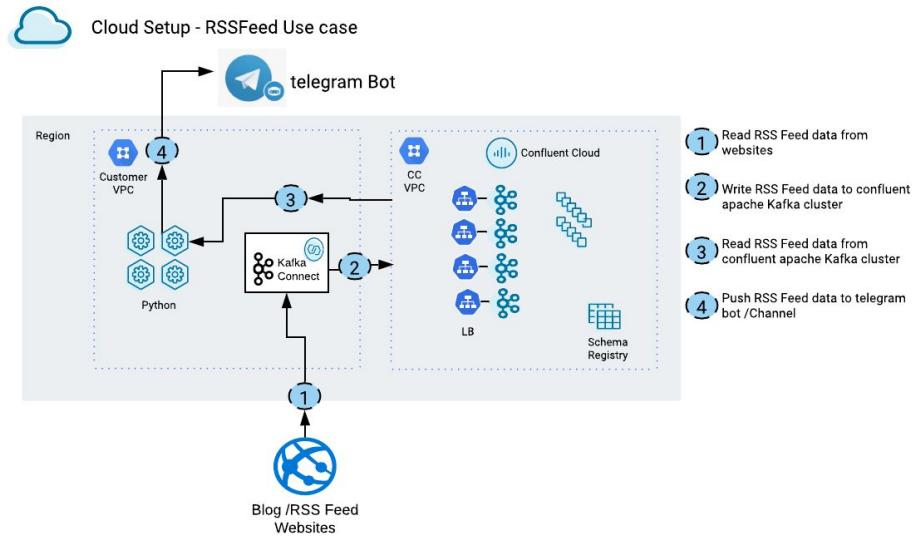
29 October 2020



# Level 1: Einführung

Im Tutorial erwarten euch:

- Erstellung eines Kafka-Clusters
- Monitoring- und Compliance-Möglichkeiten
- Vorstellung der Formate und der Schema Registry
- Erstellung von Topics (Message Browser, Producer, Consumer)
- Zugriffs- und Nutzungsmöglichkeiten auf Confluent-Cloud-Umgebung und innerhalb des Apache Kafka Clusters
- **ALLES LIVE UND IN FARBE**





CONFLUENT



Level 2:

Verbindungen schaffen - Kafka  
Connectors selbst-verwaltet  
vs. vollständig gemanaged

Carsten Mützlitz, Solution Engineering Consultant

Jan Svoboda, Solution Engineering Consultant

5 November 2020

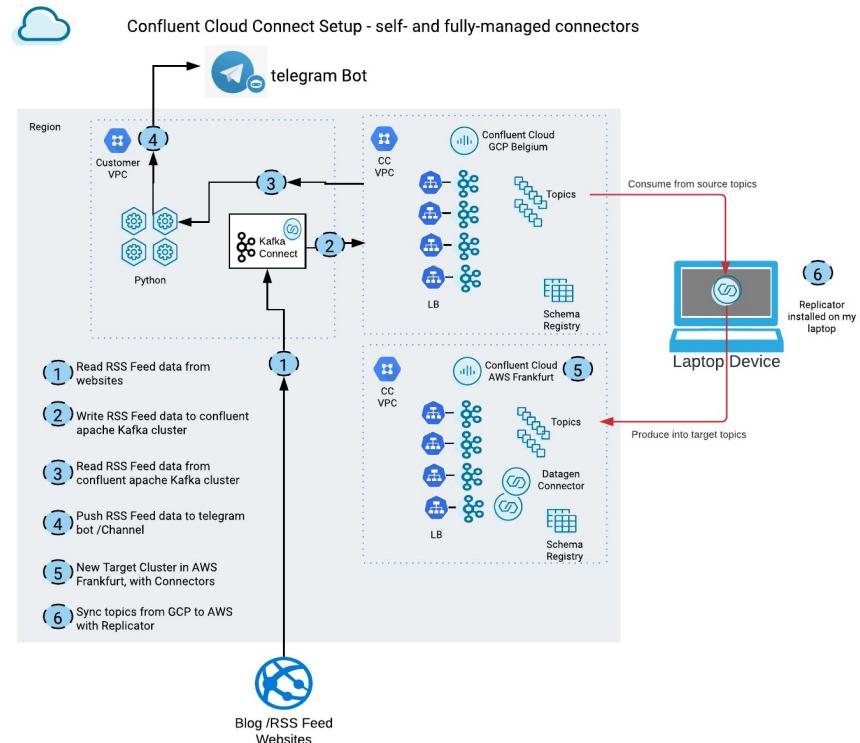




# Level 2: Connector fully-managed vs. self-managed

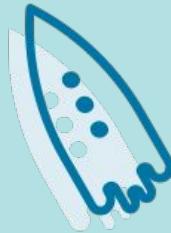
Heute erwartet euch:

- Wir nutzen verschiedene Connectoren
- 2 **datagen-connectors** **fully-managed** (users, pageviews)
- **RSSFeed Connector self-managed** in aws compute instance (from webinar 1)
- **Replicator** running on local machine (your laptop) für cmorders\_avro sync (GCP Belgium -> AWS FRA)
- **ALLES LIVE UND IN FARBE**





CONFLUENT



# Level 3: !Stream Processing für alle!

Carsten Mützlitz, Solution Engineering Consultant

Jan Svododa, Solution Engineering Consultant

10 November 2020

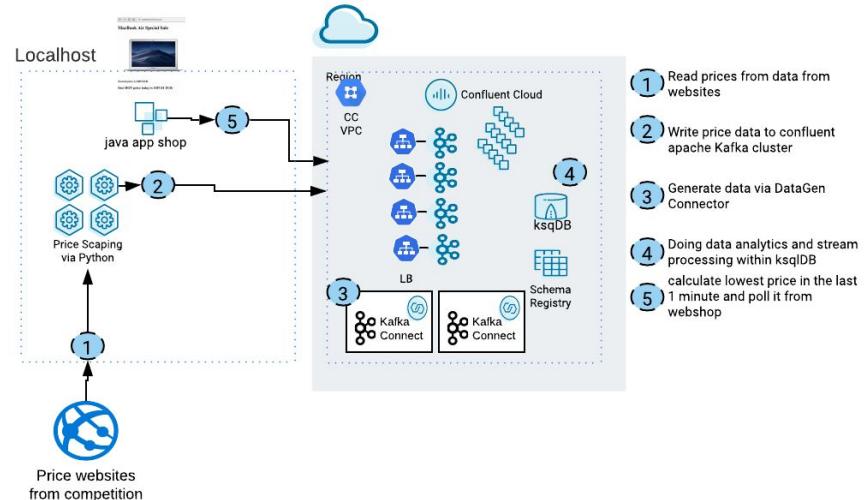


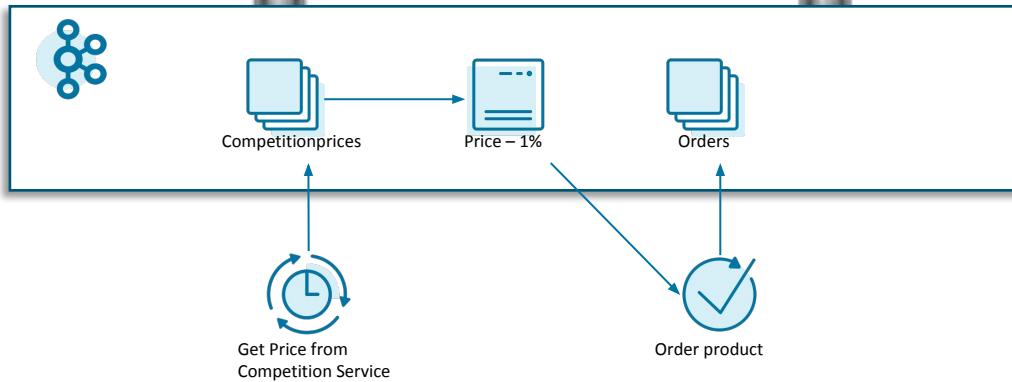
# Level 3: ksqlDB

Im Tutorial erwarten euch:

- Wir bauen einen kleinen Web-Shop  
“Der -1% günstiger als unsere Konkurrenz Webshop”
- Hier nutzen wir ksqlDB, um die Preise unserer Konkurrenz innerhalb eines 1 Minuten Fensters immer um 1% zu unterbieten
- **ALLES LIVE UND IN FARBE**

Cloud Setup - Pricescraping and ksqlDB analytics Use case





## Business Idea:

- Become a cost leader for high quality product
- we are scraping every 30 seconds the price of our competition
- Calculate our price -1% based on the min price of last 1 minutes of our competition
- Integrate the real-time price in our webshop

*Let's implement this simple solution and use mainly Confluent Cloud.*

# The -1% under competition webshop



The -1% under Competition Price Webshop

Shop now MacBook Air -1% cheaper than Competition >>> [Check Price in Real-time](#)

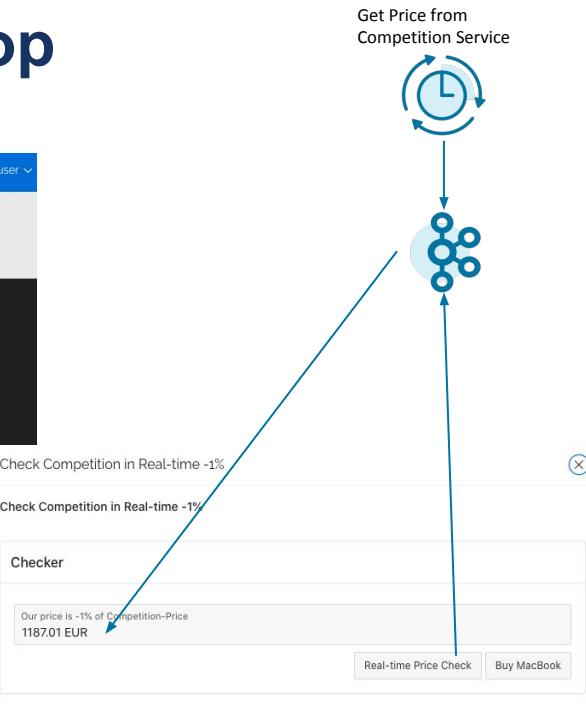


**Hot stuff**

Product: Apple MacBook Air  
Display: 13.3 Zoll (33,78 cm) Retina Display (2.560 x 1600)  
Prozessor: 1,1 GHz Dual-Core Intel Core i3 Prozessor  
Arbeitsspeicher: 8 GB LPDDR4X - 3733 RAM  
Speicher: 256 GB SSD  
Grafik: Intel Iris Plus

This MacBook Air is always available.

A single click is all you need.





# Let's install everything what we need

## Fully Managed Confluent Kafka

 Confluent Platform running in Confluent Cloud

 ksqlDB running in Confluent Cloud as fully managed service Including REST Interface

## Configure the cluster

- Already done according to best practice
- Only create two topics
  - Orders
  - competitionprices
- And some security
  - API Keys
  - Access Control Lists

## Webshop

- In my case I used Java Webshop by @Jan

*For Installation I used the ccloud cli in a shell script . /00\_create\_ccloudcluster.sh*

*A variety of laws may apply to unauthorized scraping, including contract, copyright and trespass to chattels laws. (“Trespass to chattels” protects against unauthorized use of someone's personal property, such as computer servers).*

*The fact that so many laws restrict scraping means it is legally dubious. Please be aware that this is only demo case. If you want do something similar, please ask for allowance at website provider.*

## DISCLAIMER



# What do we need in Kafka

- enable ksqlDB

- 3 SQL statements

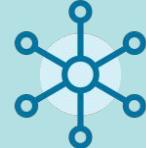
```
ksql> CREATE STREAM competitionprices (rowkey STRING KEY, shop VARCHAR, title VARCHAR, pricestr VARCHAR,
        pricefloat DOUBLE)
        WITH (KAFKA_TOPIC='competitionprices', VALUE_FORMAT='JSON');

ksql> CREATE TABLE competitionprices_table AS
        SELECT title as productname, shop,
               min(pricefloat) AS lowestprice_1minutes
        FROM competitionprices WINDOW TUMBLING (SIZE 1 MINUTES) GROUP BY title,shop EMIT CHANGES;
ksql> SELECT lowestprice_1minutes-(lowestprice_1minutes/100) as ourPrice
        from competitionprices_table emit changes limit 1;
```

- The last statement was a workaround. Since yesterday pull queries are allowed in confluent cloud fully managed ksqlDB APP



CONFLUENT



Level 4:

# Netzwerken leicht gemacht: VPC/VNet Peering & Private Link

Suvad Sahovic, Solution Engineering Consultant

Carsten Mützlitz, Solution Engineering Consultant

19 November 2020



# Confluent Cloud Network Connectivity



1. Internet Endpoint
2. VPC/VNET Peering
3. Private Link
4. AWS Transit Gateway



# VPC/VNET Peering

- NO Public Endpoint
-

# Private Link



.

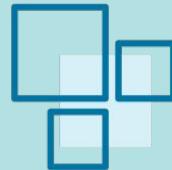
# AWS Transit Gateway



•



CONFLUENT



# Level 5: Ab auf Wolke 7 - Cloud-Native Microservices-Entwicklung

Jan Svoboda, Solution Engineering Consultant  
Suvad Sahovic, Solution Engineering Consultant



# Level 5: Cloud-Native Microservices-Entwicklung



Im Tutorial erwarten euch:

- Microservices Architektur mit Apache Kafka Einführung
- Best Practices und Beispiele für:
  - Microservice Decoupling
  - Database replacement / Event Sourcing
  - State Store Caching
  - Microservices Scaling
- **ALLES LIVE UND IN FARBE**

The screenshot shows two parts of a web application. The top part is a table titled "List of Bookmarks" with columns "Name", "URL", and "Added". It contains four rows of data:

Name	URL	Added
Kafka	<a href="http://kafka.com">http://kafka.com</a>	Fri Jun 05 06:09:40 GMT 2020
Confluent	<a href="http://confluent.io">http://confluent.io</a>	Fri Jun 12 15:04:05 GMT 2020
Jan	<a href="http://jan.com">http://jan.com</a>	Fri Jun 05 06:09:50 GMT 2020
Google	<a href="http://google.com">http://google.com</a>	Mon Jun 08 14:19:52 GMT 2020

The bottom part is a form titled "Submit new bookmark:" with fields for "Name" and "URL" and a "Submit" button. Below the form, a message says "Your are logged in as jan". At the bottom left is a "Reload" button.

# Motivation: Legacy Application vs. Microservices

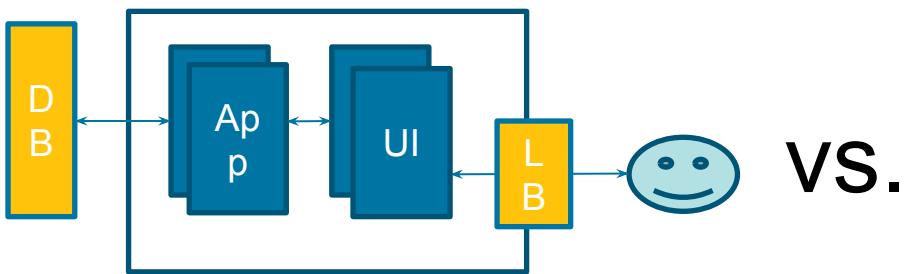


VS.

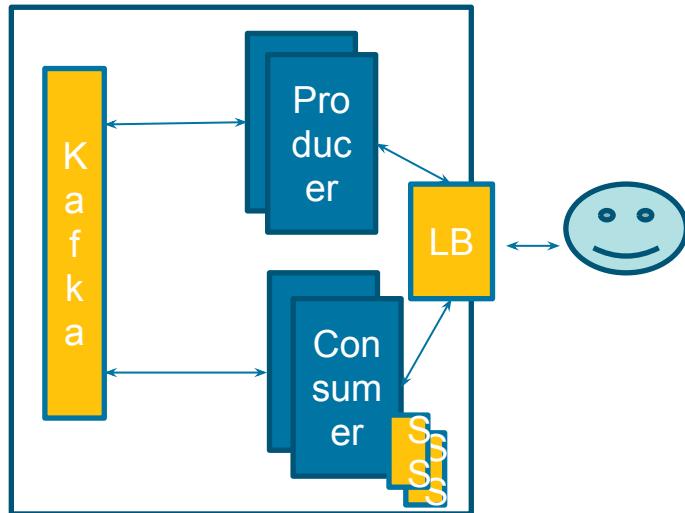


- Which system will give you better music experience? ...think quality
- Which system will be faster to build and operate? ....think speed
- Which system will be cheaper? ...think costs

# From Tradition Three Layer to Event Driven

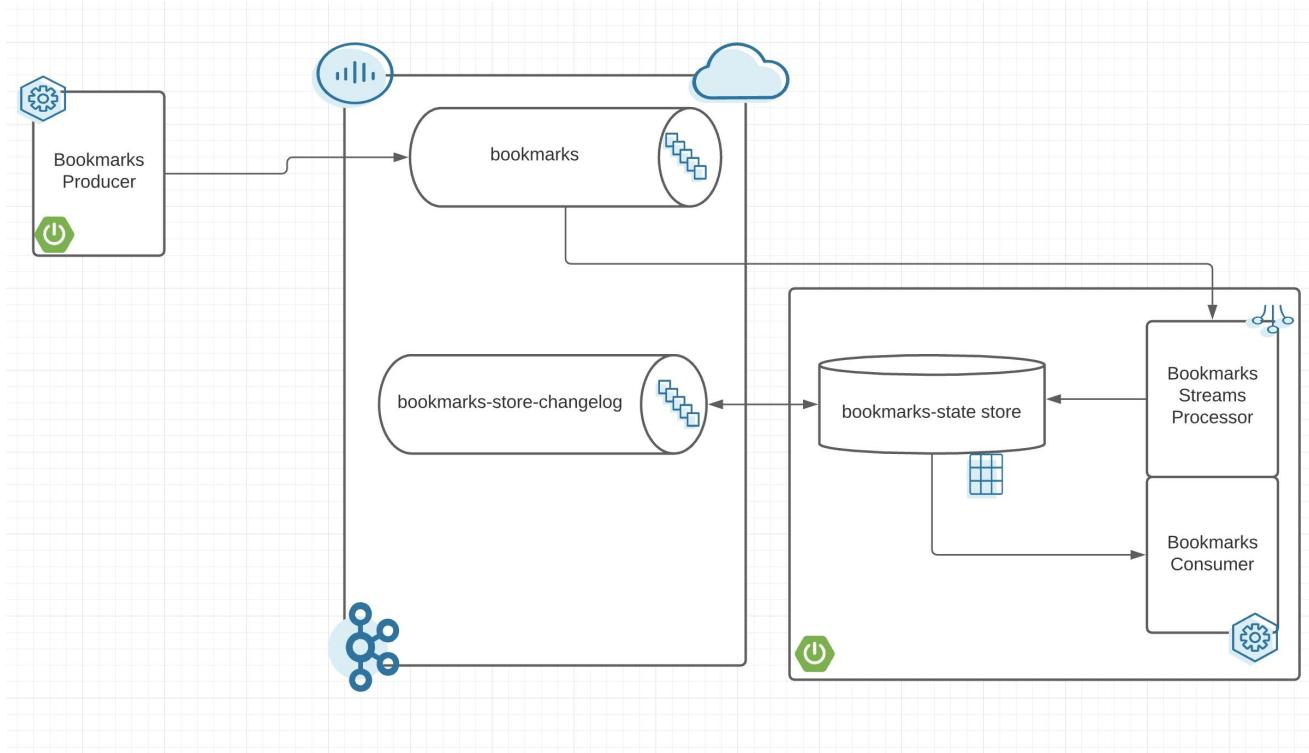


VS.



- Microservice Dependencies Decoupling ...think CQRS
- Data source dependency removal ...think Flexibility
- State Store for Microservices scaling ...think Scalability

# Demo Application - Architecture Detail





Level 6: A white document icon with a blue outline, featuring a large dollar sign (\$) symbol in the center.

## Endboss: Kostenoptimierung & TCO

Carsten Mützlitz, Solution Engineering Consultant

Jan Svoboda, Solution Engineering Consultant

Suvad Sahovic, Solution Engineering Consultant

# Confluent Cloud Calculator

<https://www.confluent.io/confluent-cloud/pricing>



## Estimate your monthly usage

Simple, transparent, usage-based pricing for use cases large and small. New signups can save up to \$200 off each of their first three Confluent Cloud monthly bills.



# Different Cloud Provider - different Prices

<https://www.confluent.io/confluent-cloud/pricing/#clusterDetails>

## DATA IN

\$ / GB write

### Basic

AWS	<b>\$0.13</b>
AZURE	<b>\$0.12</b>
GCP	<b>\$0.11</b>

### Standard

	SZ	MZ
AWS	<b>\$0.06</b>	<b>\$0.13</b>
AZURE	<b>\$0.05</b>	<b>\$0.12</b>
GCP	<b>\$0.04</b>	<b>\$0.11</b>



## DATA OUT

\$ / GB read

### Basic

AWS	<b>\$0.13</b>
AZURE	<b>\$0.12</b>
GCP	<b>\$0.11</b>

### Standard

AWS	<b>\$0.06</b>
AZURE	<b>\$0.05</b>
GCP	<b>\$0.04</b>



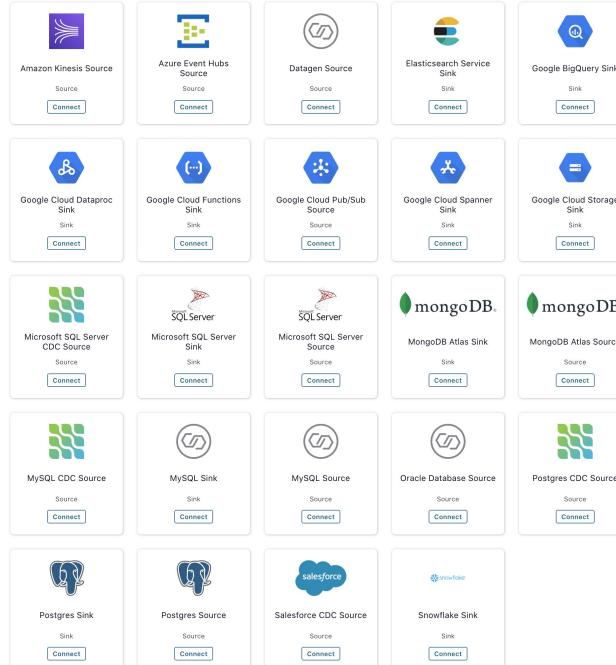


# Connectors

## self-developed vs self-managed vs fully-managed

- What is the best way for me to integrate other source/sink systems?
  - fully-managed by Confluent
  - self-managed from Confluent Hub
  - Self-developed

GCP: Connectors Dec. 2020



# Clients

## new development vs use existing apps



- JMS Clients
- REST API Clients
- MQTT Clients
- .NET Clients
- ...

### Tools and client configuration

CCloud CLI   **Clients**   CLI Tools   Confluent Platform Components   Kafka Connect

Expand the appropriate configuration to copy into your client code. Schema Registry is recommended to ensure consistent, reliable data in your topics.

For more information about the best practices for developing Apache Kafka applications with Confluent Cloud, check out the [whitepaper](#).

If necessary, create a key/secret pair for your Kafka Cluster.

[% Create New Kafka Cluster API key & secret](#)

▼ C#	CONFLUENT SUPPORTED
▼ C/C++	CONFLUENT SUPPORTED
▼ Clojure	
▼ Go	CONFLUENT SUPPORTED
▼ Groovy	
▼ Java	CONFLUENT SUPPORTED
▼ Kotlin	
▼ Node.js	
▼ Python	CONFLUENT SUPPORTED
▼ Ruby	
▼ Rust	
▼ Scala	
▼ Spring Boot	

ⓘ Need additional support?  
[See support plans](#)



# Network Connectivity

## Internet/Public Endpoint vs Private Endpoint (VPC/VNET, Private Link,...)

- Internet Endpoint -> Flexibility
- Private Endpoint -> Lower costs for data transfer

### Create cluster

1. Select cluster type —— 2. Region/zones —— 3. Networking —— 4. Security —— 5. Review and launch

**Internet**  
Cluster will only be accessible over the public internet

**PrivateLink**  
Cluster is accessible via the internet with ability to add PrivateLink connections.

**VPC Peering**  
Cluster will only be accessible via VPC Peering

# Infinite Retention for Standard or Dedicated



- Cost Reduction
- Simplify Architecture
- New Use cases  
(Simulation, Event Sourcing,...)
- Kafka as Data Lake
- Easier ML Training and Deployment
- Risk Mitigation for unplanned high storage
- ...

Standard Cluster Type limits:

Configuration & cost	Usage limits	Uptime SLA	
<strong>Cluster limits</strong>			
Ingress	up to 100 MBps	Egress	up to 100 MBps
Storage	Unlimited	Partitions	up to 2,048
Client connections	up to 1,000	Requests/sec	up to 15,000
Connection attempts/sec	up to 80	Request size	up to 100 MB
API keys	up to 20		

## Cluster settings



# Demo - Monitoring Options for Cost Control



## Easily monitor cluster health & usage

Monitor consumer lag and cluster metrics in Cloud UI and through the Metrics API



## Control costs with topic level metrics

Monitor topic level metrics using cloud Data flow and Metrics API



## Granularly spot rogue clients

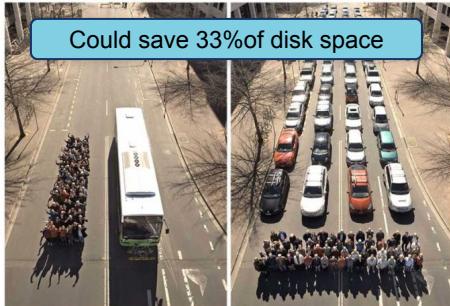
Monitor connections at topic level using cloud Data flow

# Why do I need to reduce message size?



- Customers are storing all the data kafka, e.g. New York Times
- reduce the message size = use less bandwidth = send more messages
- Cloud usage is less
  - read less data
  - write less data
  - store less data
  - => pay less money

**Linger** ([linger.ms](#)) on producer



**Compression** ([compression.type](#))

- compress before sending the data (only real data, no header)
- with lingering and gzip compression
- compression need more CPU, that is trade off

please see article on medium: <https://medium.com/swlh/exploit-apache-kafkas-message-format-to-save-storage-and-bandwidth-7e0c533edf26>

**Schema** (AVRO format)

- with lingering and AVRO Format (Schema) you reduce each record by not sending JSON data types

save ~70% of disk space for this sample

- with lingering, compression gzip and AVRO

save ~80% of disk space for this sample



# Why do I need to reduce message size?

Lingering	Compression	Compression Type	Avro	Topic size (in bytes)	Saved space (higher is better)
-	-	-	-	18,034	0
Yes	-	-	-	12,031	0.33
Yes	Yes	gzip	-	3,602	0.80
Yes	Yes	Snappy	-	5,661	0.68
Yes	-	-	Yes	5,559	0.69
Yes	Yes	gzip	Yes	3,305	0.81
Yes	Yes	Snappy	Yes	4,784	0.73

# Confluent Cloud delivers more value while reducing costs



## Increase speed to market

Deploy Kafka at scale within 1 week of starting with Confluent



## Reduce total cost of ownership

Operate more efficiently with lower infra cost, maintenance, and downtime risk



## Maximize return on investment

Deliver higher returns with your project by launching faster and reducing operational burden

# Confluent Cloud delivers more value while reducing costs



Increase speed  
to market

Reduce total cost  
of ownership

Maximize return  
on investment



**Reduce ongoing  
operating expenses**

Operate more efficiently with lower infra,  
maintenance, and development costs

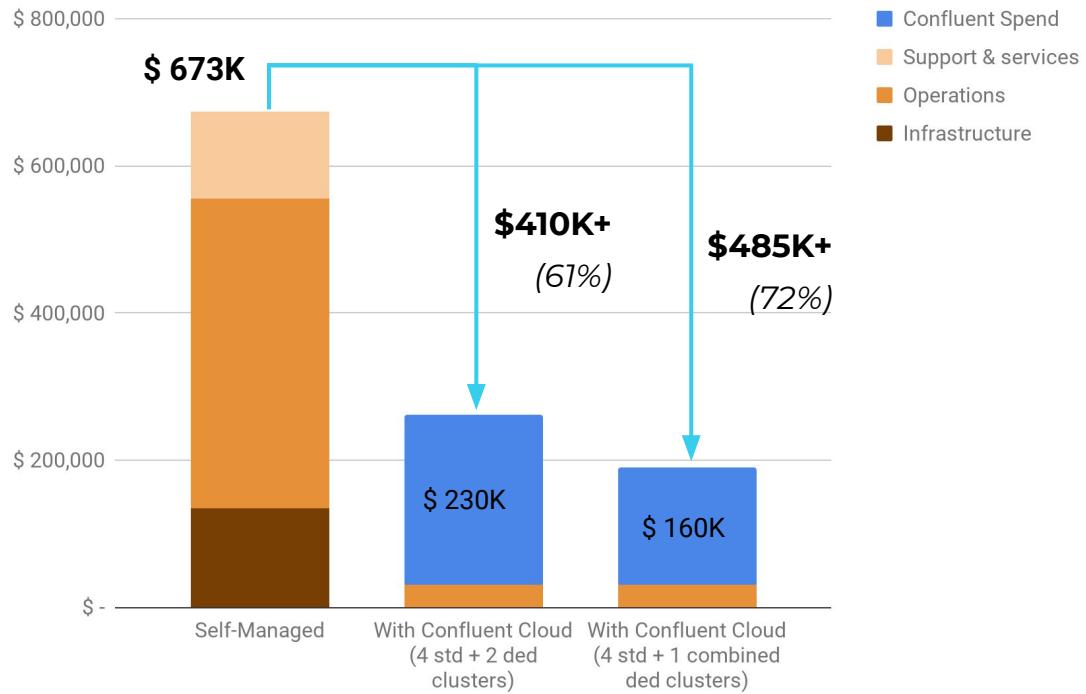


**Minimize risk and  
indirect costs**

Maintain peak productivity with lower risk and  
higher service availability and quality

# Cost of self-managed vs Confluent Cloud

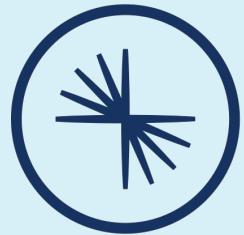
real customer sample



~60-70% cost savings annually with Confluent Cloud



CONFLUENT



CONFLUENT