
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

PRT215

SPONSORED BY CONFLUENT

Modernize your database with Confluent and Amazon RDS

Geetha Anne

Sr. Cloud Partner SA
Confluent

Srinivas Kesanapally

Senior Manager, Data and Analytics PSA
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

1. **The problem**
   Migrating to a modern managed database is a complex process

2. **Why Confluent and AWS**
   Database and data modernization with Confluent and AWS

3. **The solution**
   Proposed architecture and action plan

4. **Takeaways**
   Food for thought and next steps

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The problem

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Modern, cloud-native databases power business-critical applications with lower operational overhead

Self-managed databases

- Rigid architecture that can make it hard to integrate with other systems
- Expensive in both upfront and ongoing maintenance costs
- Slower to scale to meet evolving demands



Cloud databases

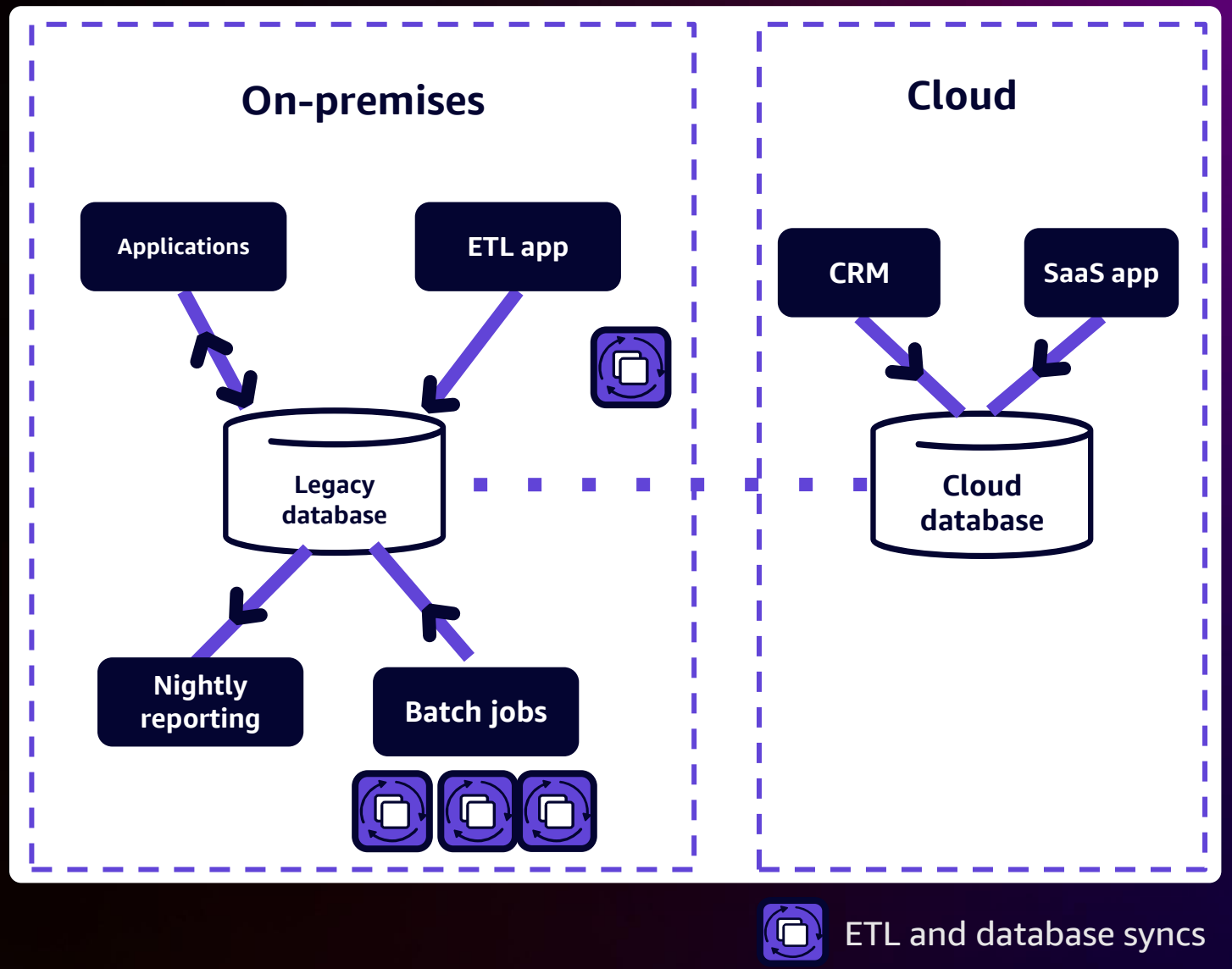
- Lower TCO by decoupling storage from compute and using consumption-based pricing
- Increased overall flexibility and business agility
- Worry-free operations with built-in auto scaling and maintenance cycles

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integrating multiple legacy systems to the cloud could be a complex multi-year process

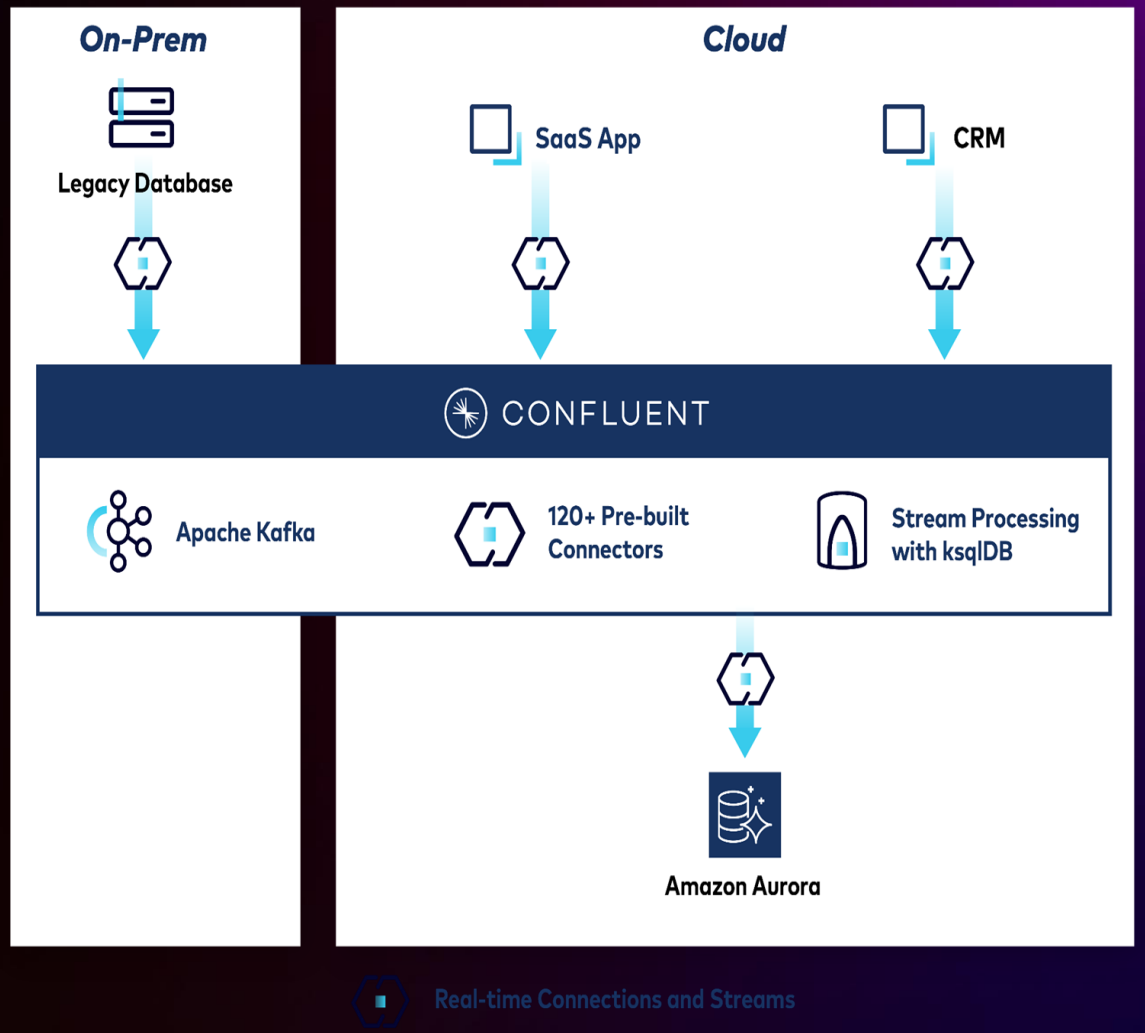
1. **Time- and resource-intensive**
   Replacing or refactoring legacy data systems across environments is not easy. During this process, data visibility can be limited.
2. **Insight blind spots**
   Getting actionable data from disparate data sources is cumbersome. Most data insight comes from nightly loads, merges, and batch updates to create a complete view.
3. **Data silos across environments**
   Difficulties with integrating multiple data silos and data formats.



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Easily modernize your database by integrating legacy with the cloud using Confluent

1. **Accelerate and simplify migration**
   Link on-premises and cloud for easier data movement across environments, with real-time event streaming to start migrating immediately
2. **Enhance data in real time**
   Process data in flight with ksqlDB and use our fully managed service to enrich or transform your data before inserting into the cloud database
3. **Reduce total cost of ownership**
   Leverage fully managed services and avoid prohibitive licensing costs from existing solutions offered by legacy vendors

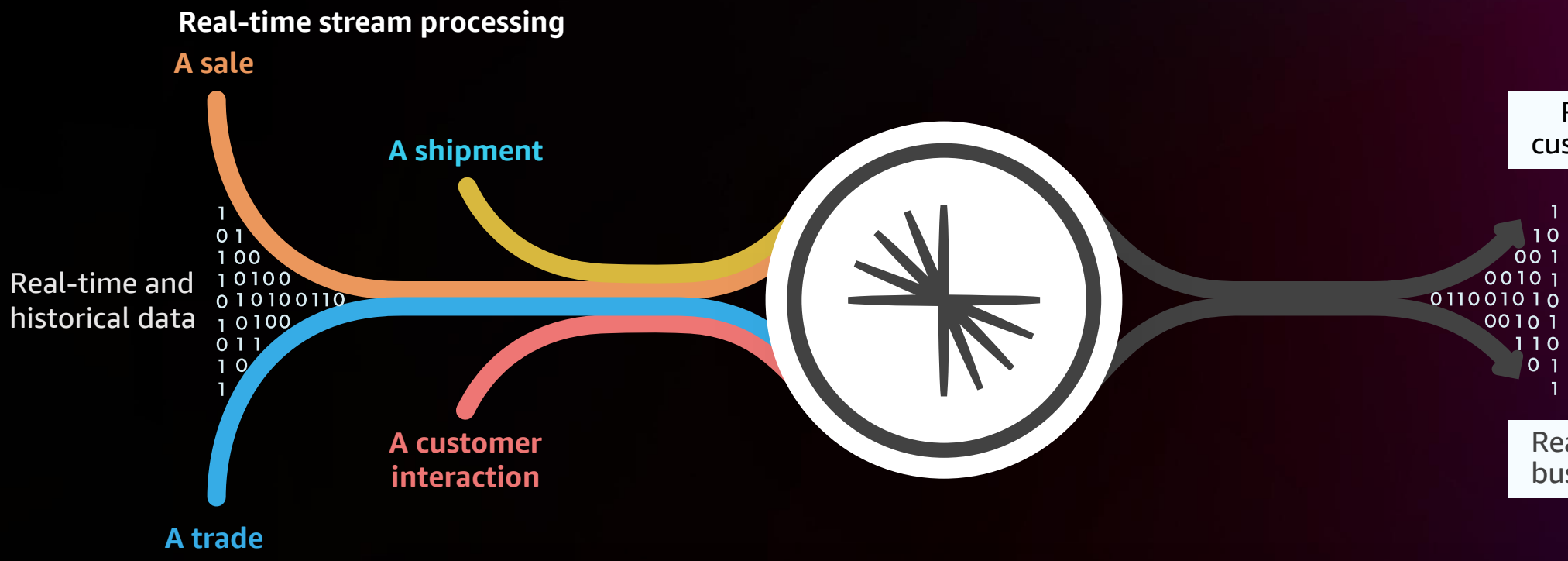


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why Confluent and AWS?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

A new paradigm is required for data in motion



“We need to shift our thinking from everything at rest, to everything in motion.”

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Operationalizing Kafka can be difficult

KAFKA CAN BE HARD IN EXPERIMENTATION. IT OFTEN GETS HARDER (AND RISKIER) AS YOU ADD MISSION-CRITICAL DATA AND USE CASES.

Key challenges:

Operational burden and resources

Manage and scale platform to support ever-growing demand

Security and governance

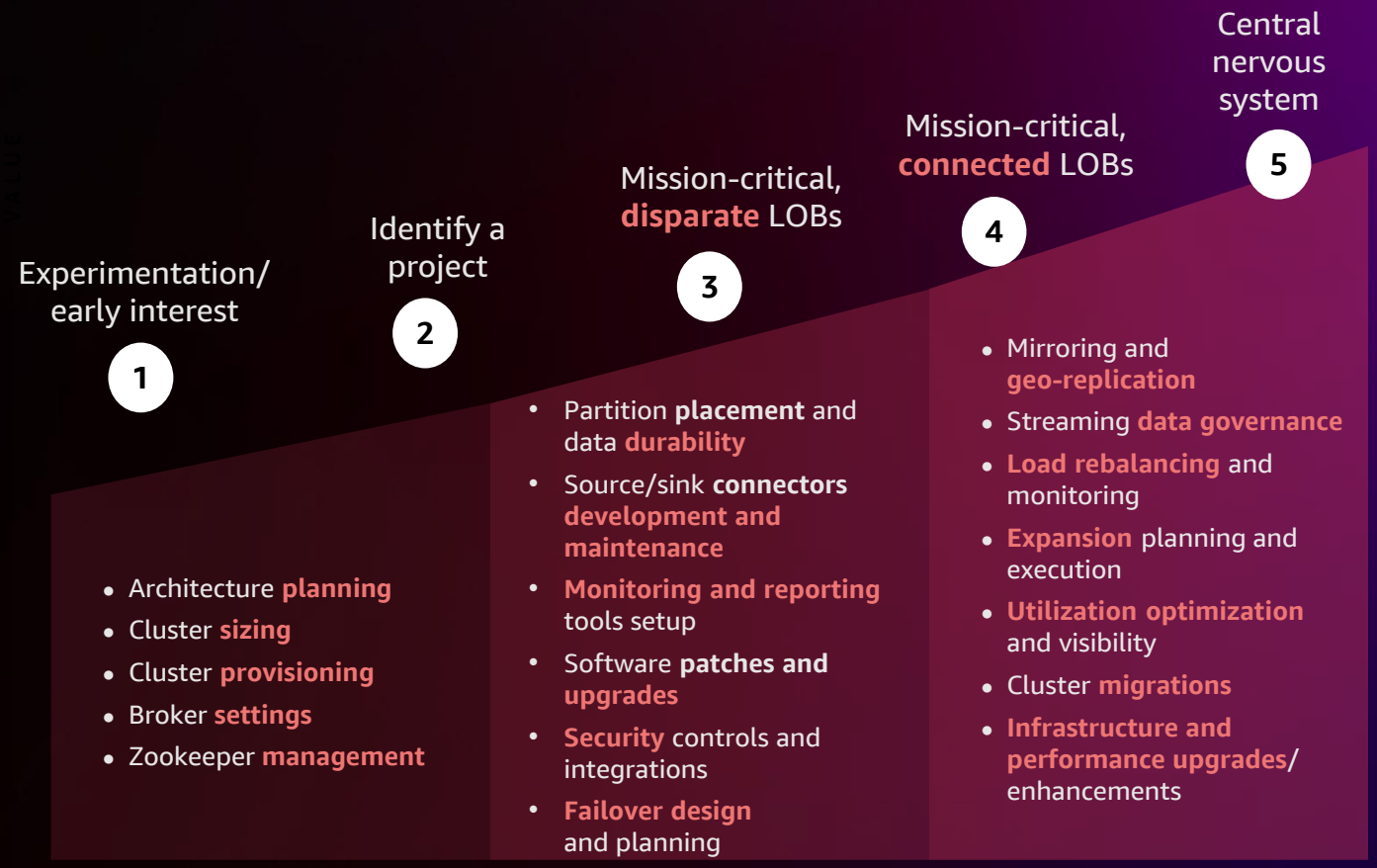
Ensure streaming data is as safe and secure as data-at-rest as Kafka usage scales

Real-time connectivity and processing

Use valuable legacy data to power modern, cloud-based applications and experiences

Global availability

Maintain high availability across environments with minimal downtime



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Self-managed solutions aren’t flexible enough

Managing databases at scale is challenging

1. **Time to migrate**
   It can take months to transfer all of your accrued data via appliances or one-off systems. Constraints like compute power or transfer bandwidth are hard to bypass.

2. **Dependencies**
   Need to identify and coordinate interdependencies between multiple applications and teams, while not losing consistency or integrity of the data.

You are migrating, not modernizing

Modernizing your database is not a “set it and forget it” act. Utilizing your new cloud database, you’ll want to rethink and reinvent the way you analyze and act on your data.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cloud-native

Confluent is a cloud-native Kafka solution



Elastic

Scale up instantly to meet any demand and scale back down to avoid over-provisioning infrastructure



Global

Create a consistent data fabric throughout your organization by linking clusters across your different environments



Infinite

Store unlimited data on Confluent to enhance your real-time applications and use cases with a broader set of data

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Everywhere

Confluent provides deployment flexibility to span all of your environments

FULLY MANAGED SERVICE

Confluent Cloud

Cloud-native service for Apache Kafka

aws

SELF-MANAGED SOFTWARE

Confluent Platform

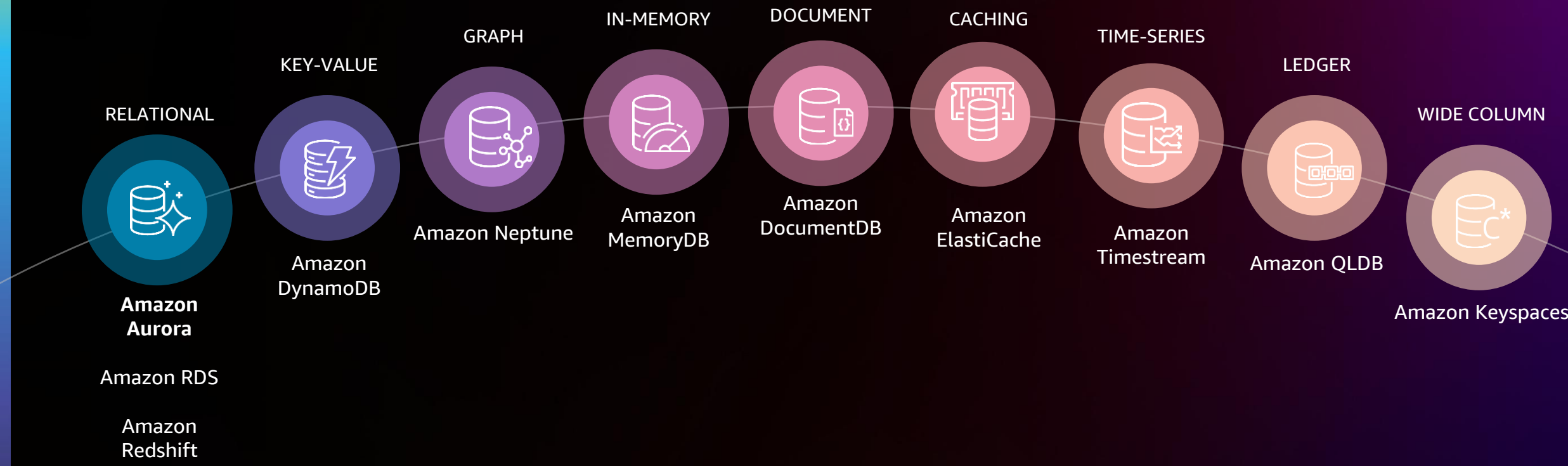
The enterprise distribution of Apache Kafka



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Broad portfolio of purpose-built databases



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

A cost-effective enterprise database . . .

Amazon Aurora – enterprise database

at open-source price, delivered as a managed service



Speed and **availability** of high-end commercial databases

Simplicity and **cost-effectiveness** of open-source databases

Drop-in **compatibility** with MySQL and PostgreSQL

Simple **pay-as-you-go** pricing

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

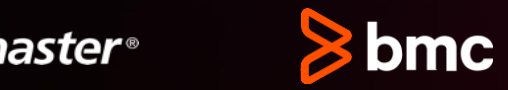
Aurora is the fastest-growing service in AWS history

AURORA IS USED BY 955 OF THE TOP 1,000 AWS CUSTOMERS







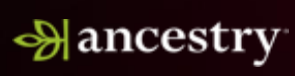








































































© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

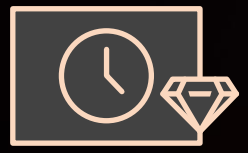
Amazon Aurora

Designed for unparalleled high performance and availability at global scale with full MySQL and PostgreSQL compatibility at one-tenth the cost of commercial databases



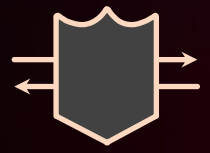
Performance and scalability

- Scale out up to 15 read replicas
- Decoupled storage and compute enabling cost optimization
- Fast database cloning
- Distributed, dynamically scaling storage subsystem



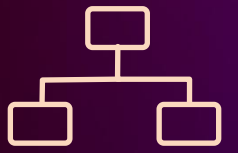
Availability and durability

- 6 copies of data across 3 AZs (customers pays for 1)
- Automatic, continuous, incremental backups with point-in-time recovery (PITR)
- Failovers in < 10 seconds
- Fault-tolerant, self-healing, auto-scaling storage
- Global Database for disaster recovery



Highly secure

- Network isolation
- Encryption at rest/in transit
- Supports multiple secure authentication mechanisms and audit controls



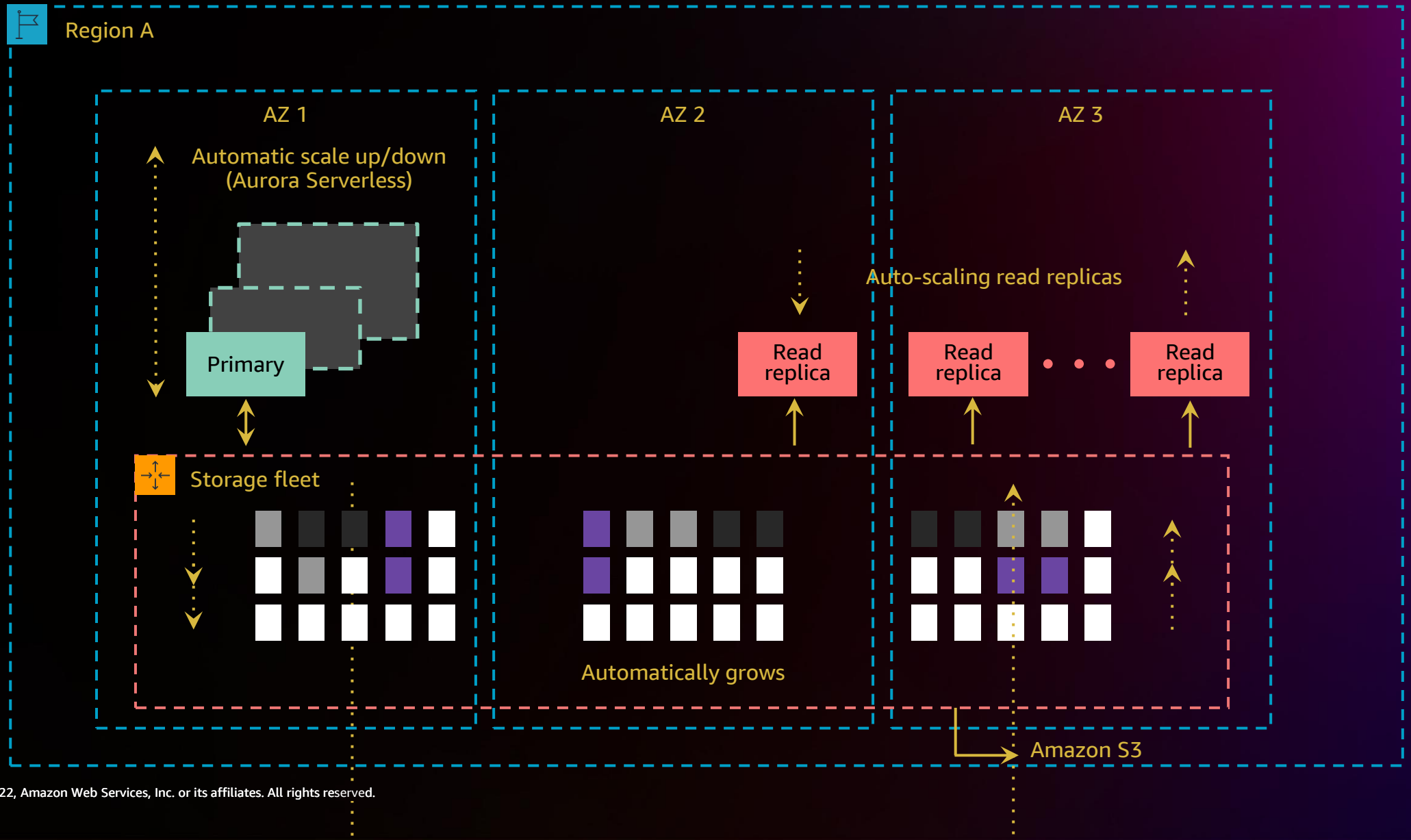
Fully managed

- Automates time-consuming management of administration tasks like hardware provisioning, database setup, patching, and backups
- Serverless configuration options

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora core features



Up to
99.99% SLA



Serverless with instant
fine-grained scaling



Global Database



Encryption at rest
and in transit



Automatic,
continuous backups



Point-in-time
recovery (PITR)



6 copies across
3 Multi-AZ



Full MySQL and
PostgreSQL compatibility



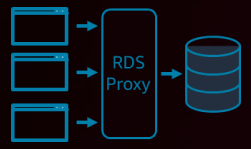
Fast database
cloning



Scale out up to
15 read replicas



Fault-tolerant, self
healing, auto-scaling
storage



RDS Proxy

Examples of Aurora integration with other AWS services



Amazon EKS



AWS Lambda



AWS CloudTrail



Amazon S3



IAM



AWS KMS



Amazon
SageMaker



AWS
DevOps Guru



Amazon
CloudWatch

aws

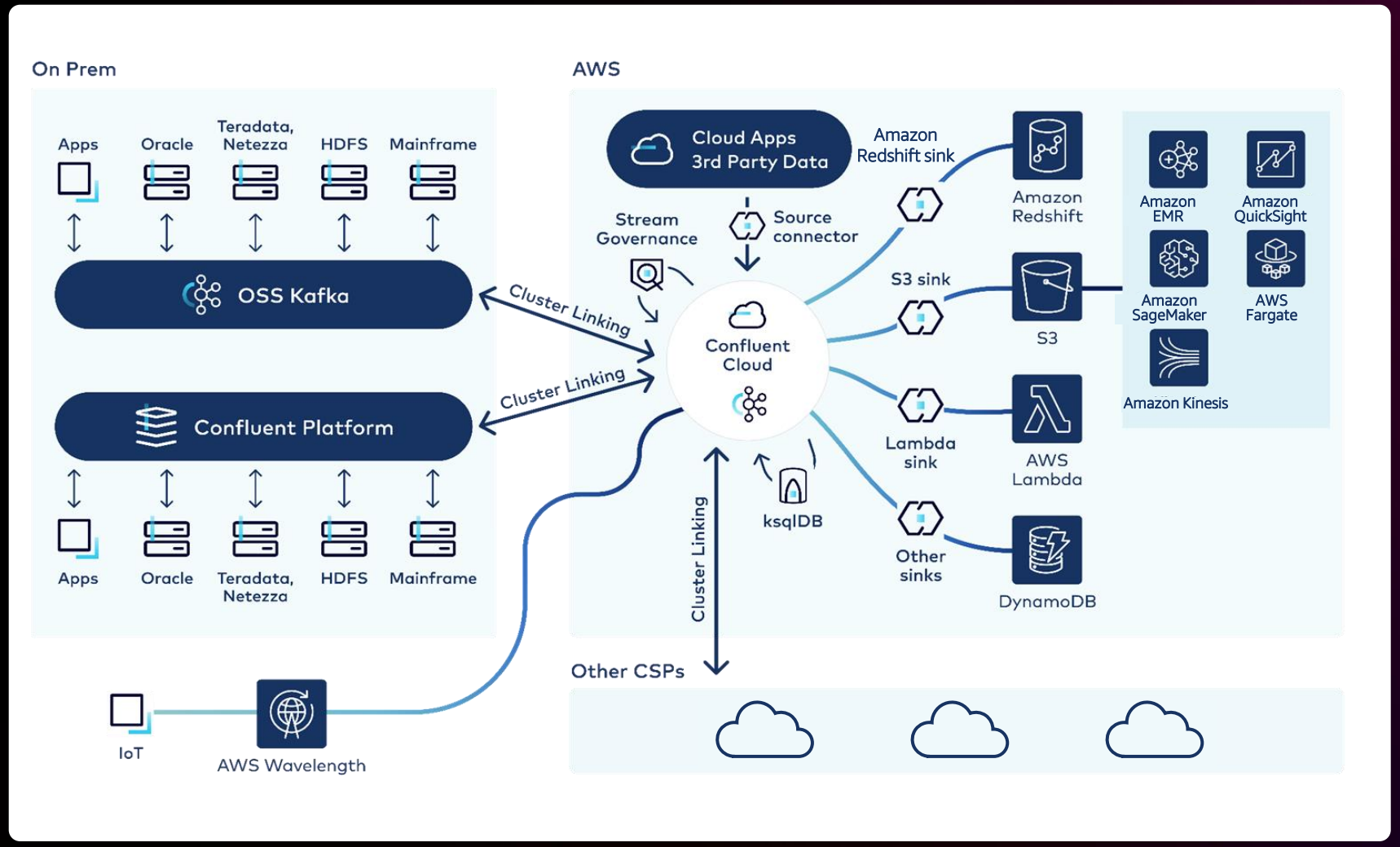
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The solution

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Understanding the architecture: AWS

HOW DOES THIS LOOK IN PRACTICE?



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Three-phase plan

MODERNIZE YOUR DATABASE WITH AWS AND CONFLUENT



1. *Migrate*

- Choose the workloads that you'd like to migrate to the cloud
- Seamlessly integrate your data source via **managed Confluent source connectors**



2. *Optimize*

- Perform real-time data transformations using **ksqlDB**
- Find the most useful queries for your cloud data
- Work with our ecosystem of partners to find the best use of your data



3. *Modernize*

- Use our **managed sink connectors** to send data into your cloud database of choice
- Continue migrating workloads into the cloud as chances arise

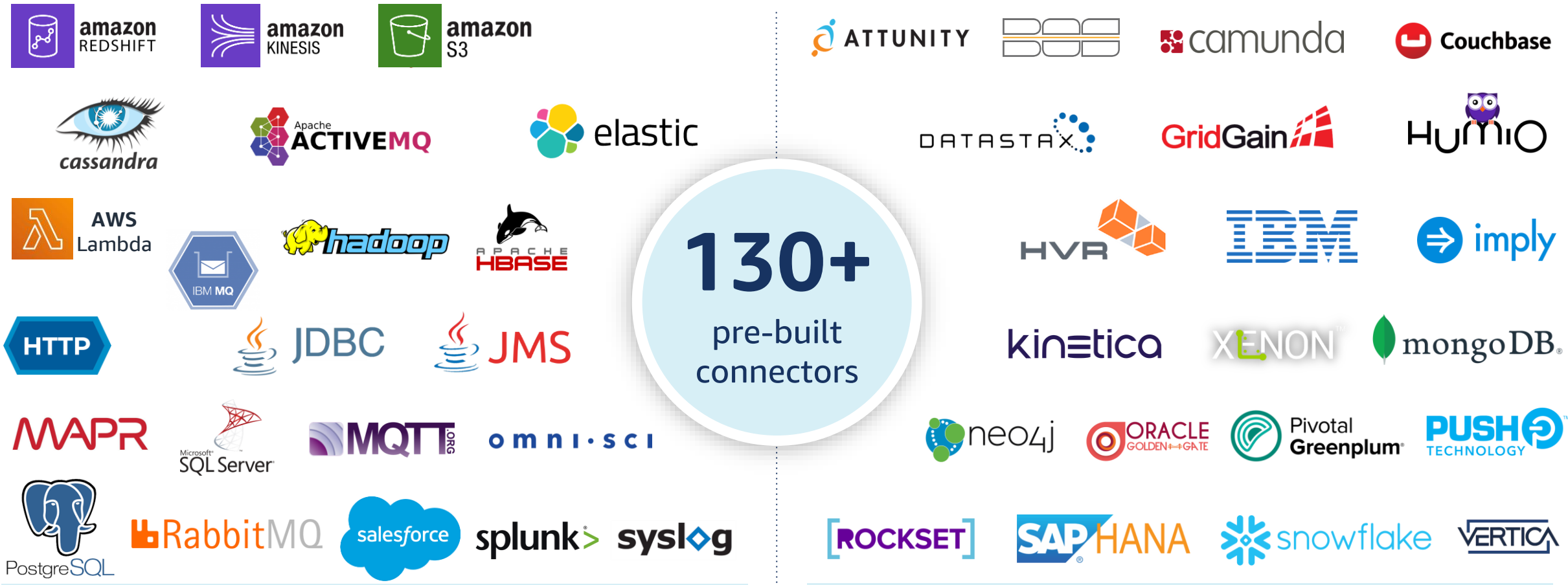
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Migrate:
Source connectors

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Instantly connect popular data sources and sinks



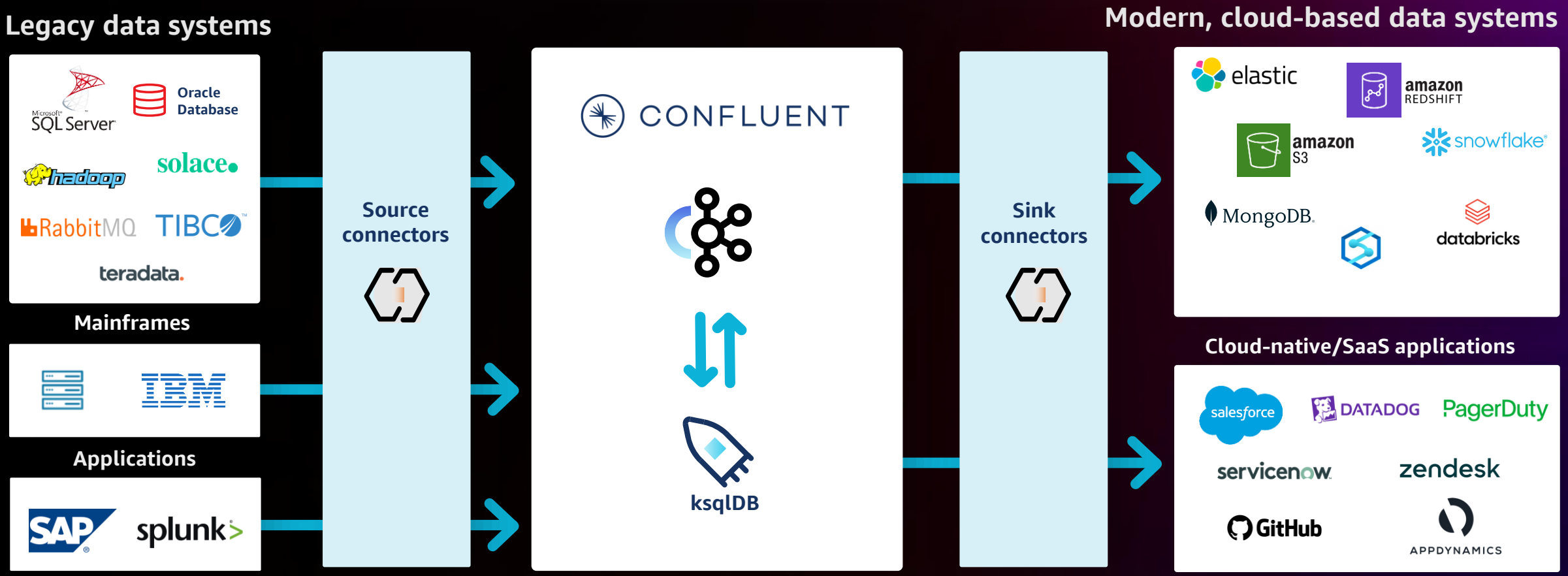
100+ Confluent supported

30+ partner supported, Confluent verified

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Modernize and bridge your entire data architecture with Confluent's robust connector portfolio



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Modernize:
Sink connectors

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3 modalities of stream processing with Confluent

Kafka clients

```
ConsumerRecords<String, String> records = consumer.poll(100);
Map<String, Integer> counts = new DefaultMap<String, Integer>();
for (ConsumerRecord<String, Integer> record : records) {
  String key = record.key();
  int c = counts.get(key)
  c += record.value()
  counts.put(key, c)
}
for (Map.Entry<String, Integer> entry : counts.entrySet()) {
  int stateCount;
  int attempts;
  while (attempts++ < MAX_RETRIES) {
   try {
     stateCount = stateStore.getValue(entry.getKey())
     stateStore.setValue(entry.getKey(), entry.getValue() + stateCount)
     break;
   } catch (StateStoreException e) {
     RetryUtils.backoff(attempts);
   }
 }
}
```

Kafka streams

```
builder
   .stream("input-stream",
         Consumed.with(Serdes.String(), Serdes.String()))
   .groupBy((key, value) -> value)
   .count()
   .toStream()
   .to("counts", Produced.with(Serdes.String(), Serdes.Long()));
```

ksqlDB

```
SELECT x, count(*) FROM stream GROUP BY x EMIT CHANGES
```

Flexibility ⟷ **Simplicity**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

ksqlDB at a glance

All the key features of a modern streaming solution.

What is it?
ksqlDB is an event-streaming database for working with streams and tables of data.

- **Aggregations**
- **Joins**
- **Windowing**
- **Dual query support**
- **Event-time**
- **Out-of-order handling**
- **Exactly-once semantics**
- **User-defined functions**




```
CREATE TABLE activePromotions AS
  SELECT rideId,
       qualifyPromotion(distanceToDst) AS promotion
  FROM locations
  GROUP BY rideId
  EMIT CHANGES
```

How does it work?

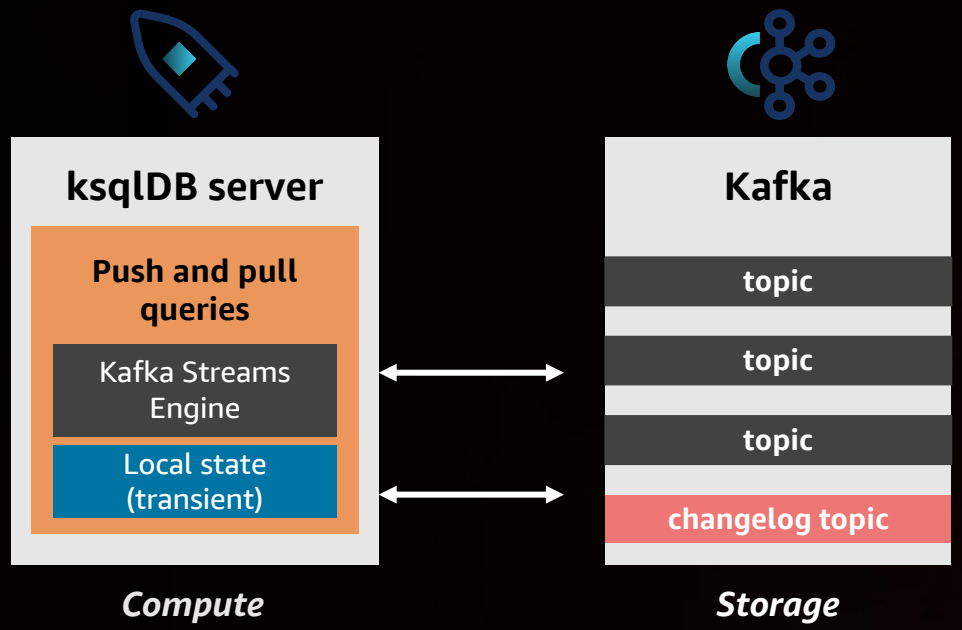
It separates compute from storage, and scales elastically in a fault-tolerant manner.

It remains highly available during disruption, even in the face of failure to a quorum of its servers.

Built on the best technology, available as a fully managed service

Kafka is the backbone of ksqlDB

ksqlDB is built on top of Kafka's proven streaming foundation. Its design reuses Kafka to achieve elasticity, fault tolerance, and scalability for stream processing and analytics.



Use a fully managed service

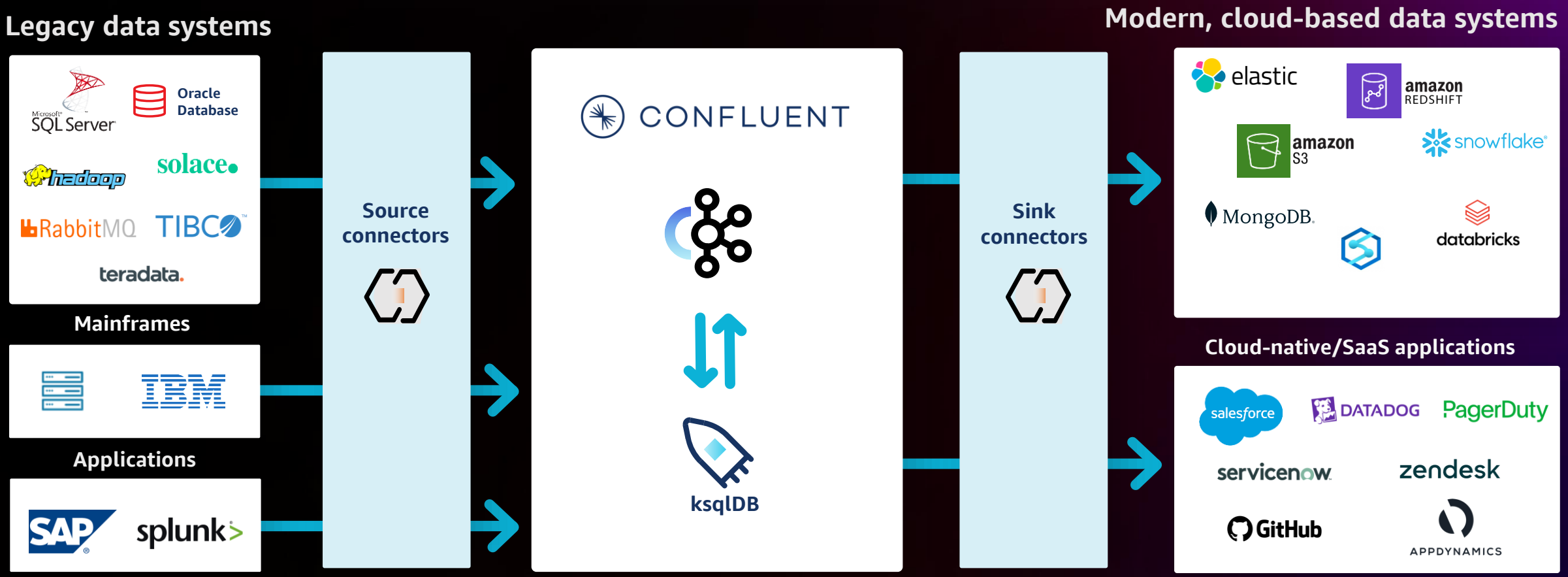
With Confluent Cloud ksqlDB, you need not worry about any of the details of running it. You can forget about

- Clusters
- Brokers
- Scaling
- Upgrading
- Monitoring

And you pay only for what you use.

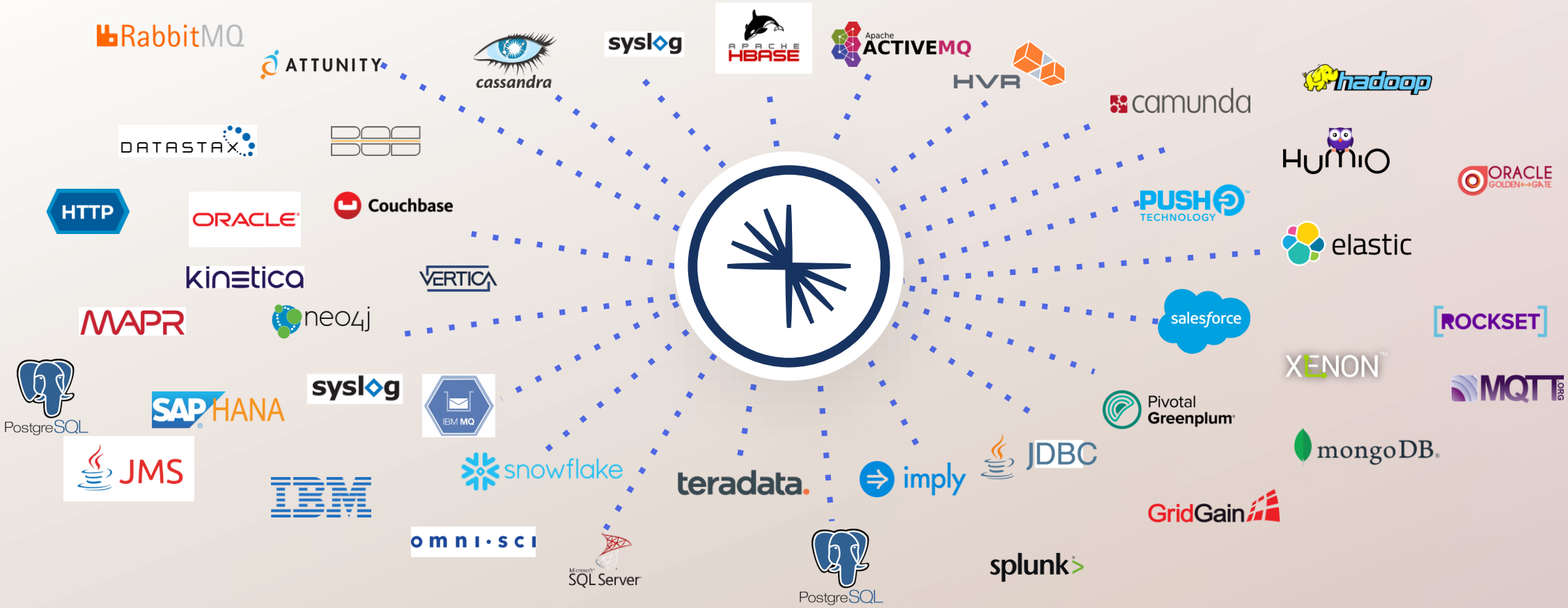
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Accelerate your migration from legacy on-prem systems to modern, cloud-based technologies



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Confluent, the central nervous system of data

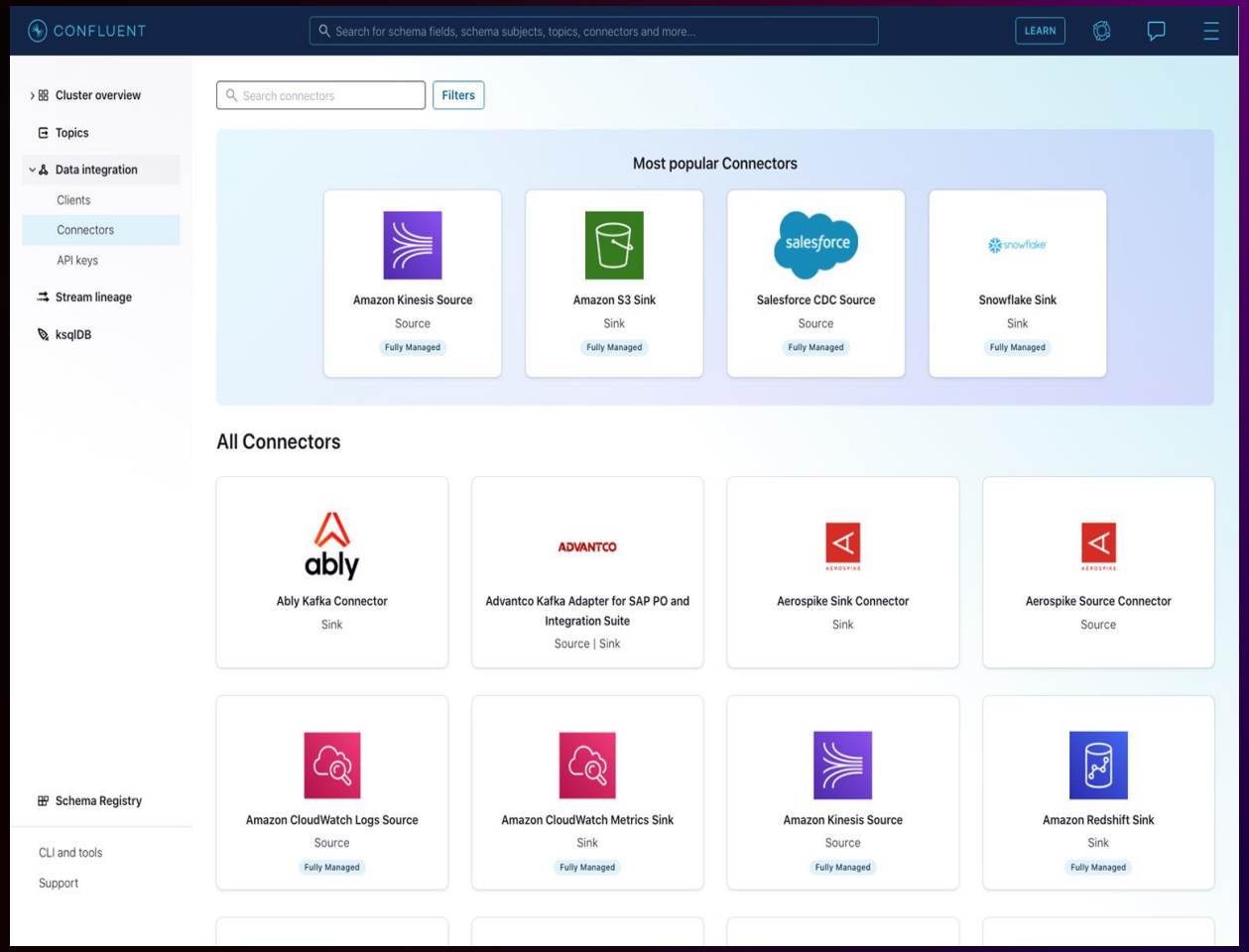


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Confluent Cloud

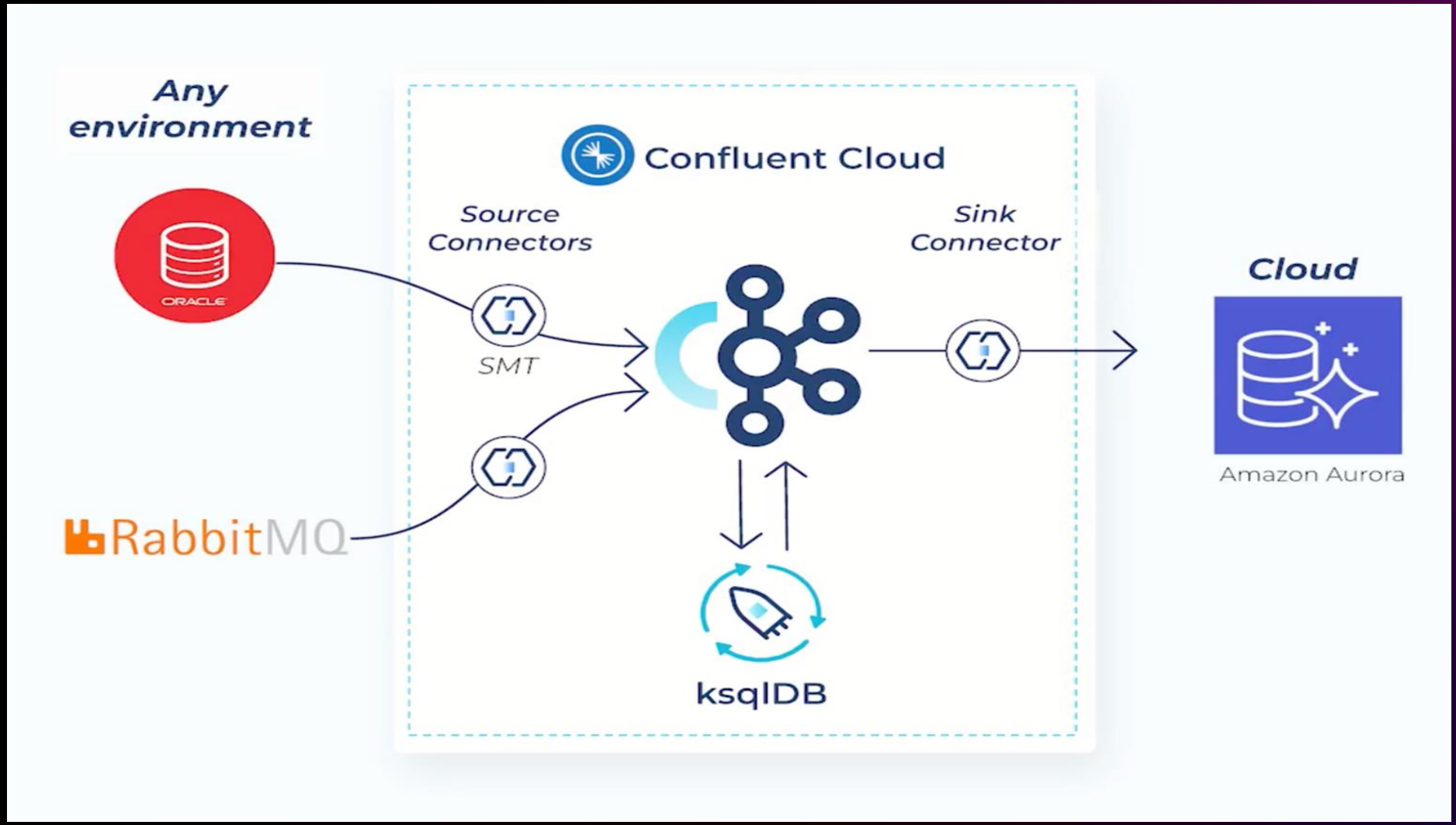
FULLY MANAGED CONNECTORS

- Limited set of the larger Connector Catalogue
- Elastic scaling with no infrastructure to manage
- Connector networking configuration dependent on your clusters networking
- Limited configuration options
- Stable source IPs are available for certain connectors



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Proposed architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Three-phase plan

MODERNIZE YOUR DATABASE WITH AWS AND CONFLUENT



1. Migrate

- Choose the workloads that you'd like to migrate to the cloud
- Seamlessly integrate your data source via **managed Confluent source connectors**



2. Optimize

- Perform real-time data transformations using **ksqlDB**
- Find the most useful queries for your cloud data
- Work with our ecosystem of partners to find the best use of your data



3. Modernize

- Use our **managed sink connectors** to send data into your cloud database of choice
- Continue migrating workloads into the cloud as chances arise

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Easily modernize your database by powering streaming ETL pipelines with Confluent

1. *Simplify and accelerate migration*

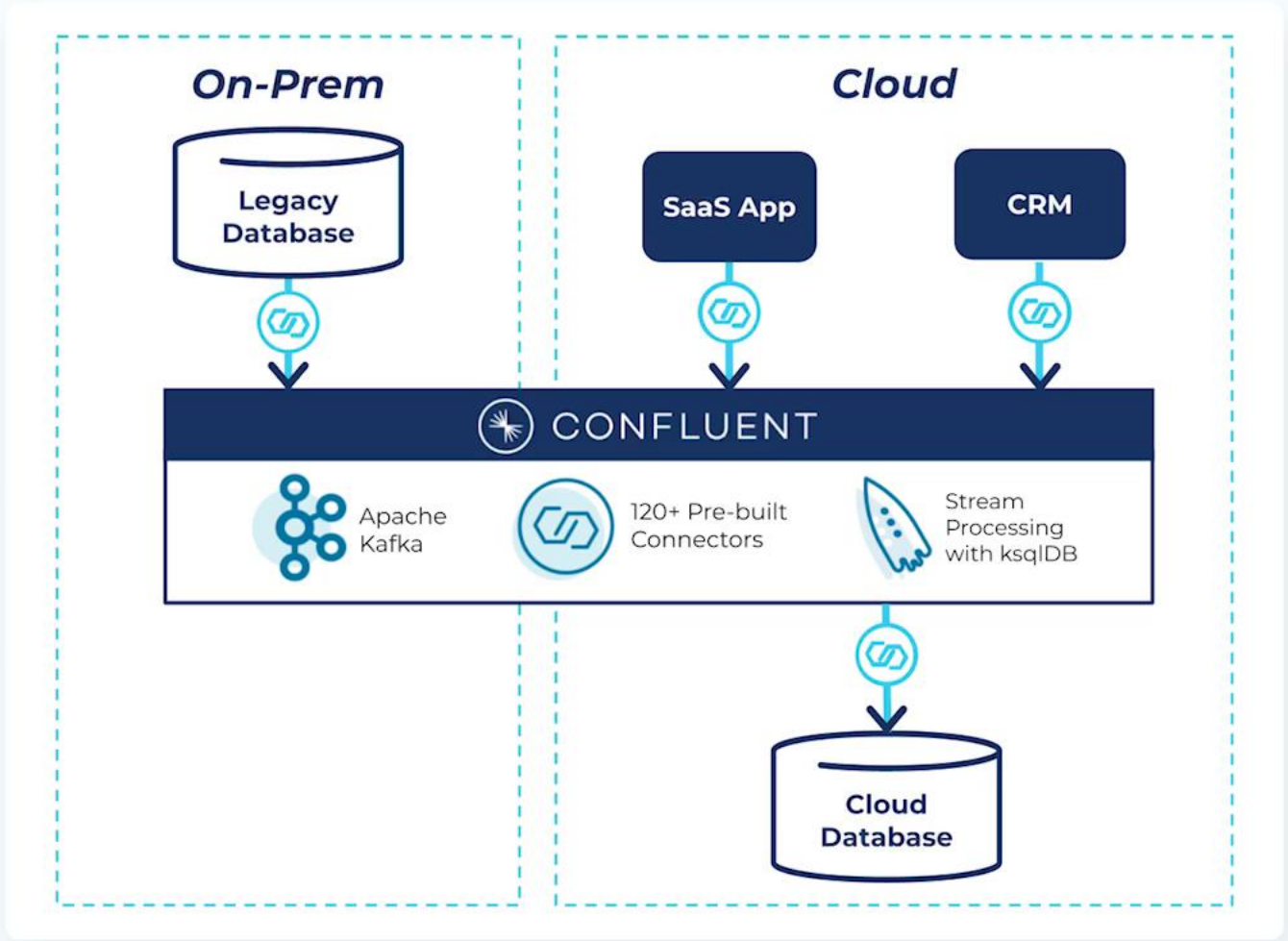
Link on-prem and cloud for **easy data movement across environments and process data in flight** with ksqlDB stream processing

2. *Stay synchronized in real-time*

Move from batch to real-time streaming and access change data capture technology using Confluent and our CDC connectors

3. *Reduce total cost of ownership*

Leverage fully managed services and avoid prohibitive licensing costs from existing solutions offered by legacy vendors



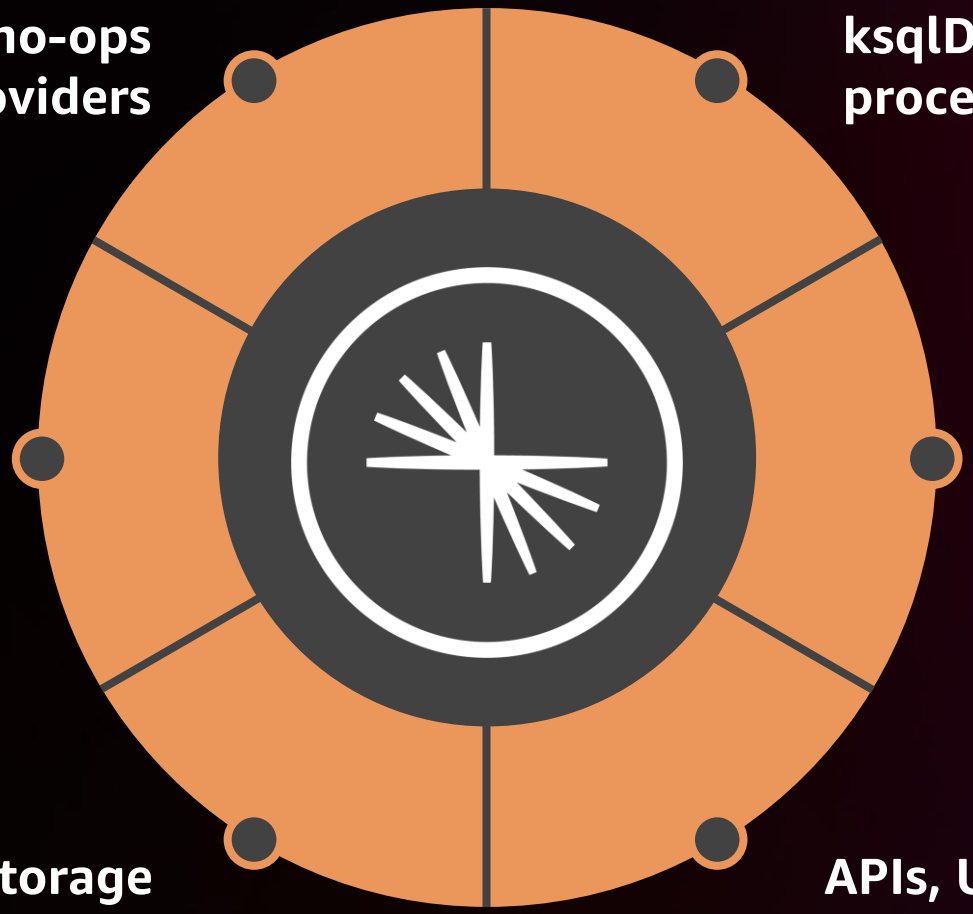
Real-time connections & streams

Cloud-native, complete, everywhere

WITH KAFKA AT ITS CORE

Fully managed no-ops for multiple cloud providers

ksqlDB and stream processing, analytics



Security and data governance

Connectors

Infinite storage

APIs, UIs, CLIs

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.