Building a real time data pipeline with Confluent for Kubernetes

Geetha Anne

Advisory Solutions Engineer

Confluent

June 15, 2022

Agenda

01 **Introduction**
The What, the Why, Vision

02 **What's in the product?**
CFK, Confluent Cloud, Confluent Platform

03 **Configuration API**

04 **Connectors**

05 **Planning a development**

06 **Architecture**

07 **Confluent Platform Support**

08 **Workflow**

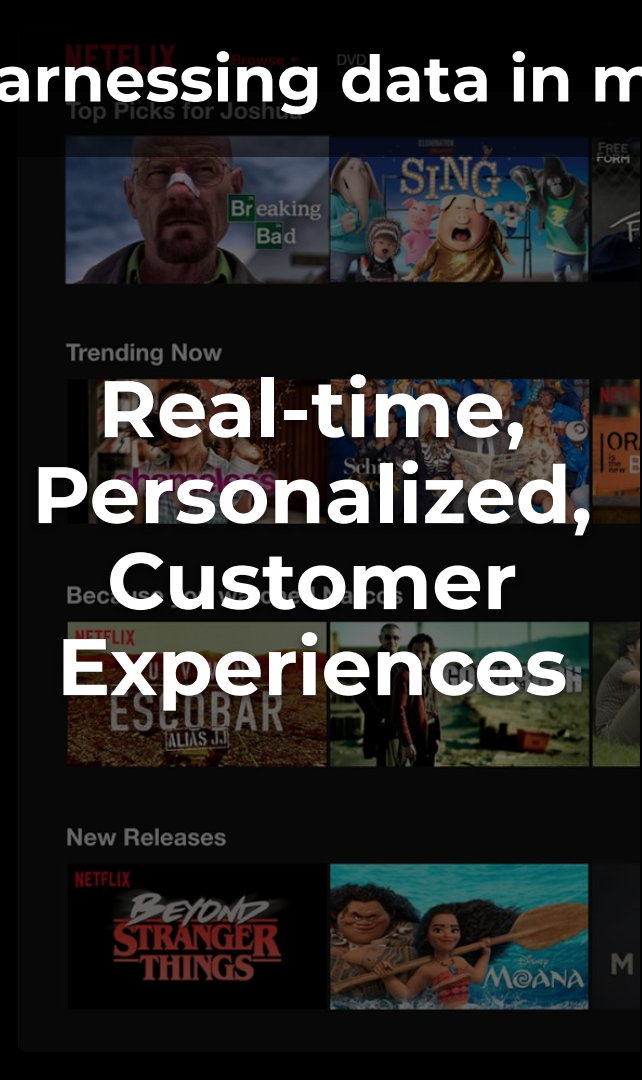
09 **Resources**

Today, Software Is the Interface

OLD WAY	NEW WAY
Slow	*Fast*
Batch processing	*Real-time stream processing*
Siloed	*Connected*

Businesses are harnessing data in motion

Data-Driven Operations



Real-time, Personalized, Customer Experiences

New Business Models

Setting data in motion requires a platform that spans across all of your environments

FULLY MANAGED SERVICE

Confluent Cloud

Apache Kafka Re-Engineered for the Cloud

Available on the leading public clouds

aws

Google Cloud

Microsoft Azure

SELF-MANAGED SOFTWARE

Confluent Platform

The Enterprise Distribution of Apache Kafka

Deploy on-premises or in your private cloud

Kubernetes

The de facto standard for building **cloud-native** platforms for **private infrastructures**

Confluent for K8s brings a truly cloud-native experience to our self-managed software product

Public Cloud Workloads

On-Premise Workloads

Confluent Cloud

A fully managed, cloud-native service for Apache Kafka®

Confluent Platform

A complete, enterprise-grade distribution of Apache Kafka®

On Kubernetes	On VMs / Bare Metal	
Confluent for Kubernetes	**Ansible Playbooks**	**Packages:** Docker, RPMs, Tarball

Generally Available now!

With **Confluent for Kubernetes**, we've **completely reimagined Confluent Platform** based on our expertise with Confluent Cloud to help customers **build their own private cloud Kafka service**

Agenda

01 **Introduction**
The What, the Why, Vision

02 **What's in the product?**
CFK, Confluent Cloud, Confluent Platform

03 **Configuration API**

04 **Connectors**

05 **Planning a development**

06 **Architecture**

07 **Confluent Platform Support**

08 **Workflow**

09 **Resources**

Confluent for Kubernetes

Introduces a declarative API-driven control plane to deploy and manage Confluent in private infrastructures

Declarative API for operating Confluent in production

Manage topics and RBAC policies through infrastructure-as-code

Integrates with cloud-native ecosystem for security, reliability, and DevOps automation

Runs on Kubernetes: the infrastructure runtime for cloud-native architectures

Confluent for K8s offers cloud-native benefits with additional control and customization

Cloud-native

- **Quickly scale to changing business demands** with single-command elastic scaling to meet any data in motion workload
- **Accelerate time-to-value with infrastructure -as-code** approach, combined with expert-backed configs to automatically deploy and manage all your Kafka infrastructure

Complete

- **Implement mission-critical use cases end-to-end** with infinite storage, disaster recovery, pre-built connectors, and SQL-based stream processing
- **Protect sensitive data** with automated security and cloud-native tooling
- **Minimize business disruption** with automated fault tolerance and rack awareness

Everywhere

- **Deploy with confidence** across market-leading Kubernetes distributions with a consistent operational experience
- **Build hybrid and multi-cloud architectures** that span across different regions and environments
- **Become cloud-ready** by easily migrating workloads to wherever your business needs them

Complete: Confluent for K8s completes Kafka with end-to-end capabilities

Connected

Democratize access to events for everyone with 120+ pre-built connectors, ksqlDB, and schema registry and validation

Secure

Automatically deploy security features with proper configurations with a single deployment specification

Reliable

Streamline recovery after a failure in your brokers or underlying infrastructure with automated fault tolerance

Connect your entire business with just a few clicks

Confluent Cloud connectors are rich with usability features to improve developer workflow

Data Output Preview

Test connector configurations by previewing its outputs, allowing you to add or correct configurations prior to launch

Single Message Transforms (SMTs)

Perform lightweight data transformations like masking and filtering in flight within the source/sink connector

Connect Log Events

View connector events in the console for contextual information and error debugging purposes

Everywhere: Deploy to any private cloud solution with confidence

Confluent for Kubernetes supports a broad ecosystem of market-leading Kubernetes distributions

VMware Tanzu

RED HAT® OPENSHIFT

RANCHER®

Anthos

amazon web services™

EKS on AWS Outposts

Microsoft Azure

AKS on Azure Stack

Build-your-own Kubernetes

Enterprise distributions

Private cloud services

Confluent Control Center

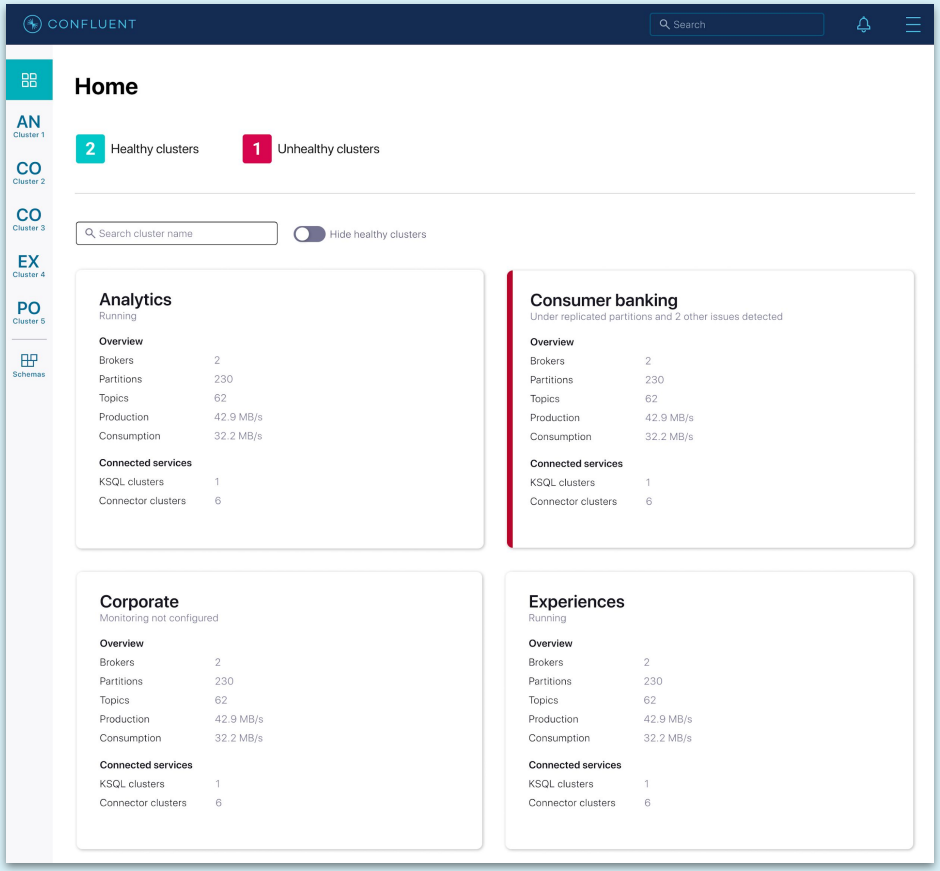
The simplest way to operate & build real-time applications

For Operators

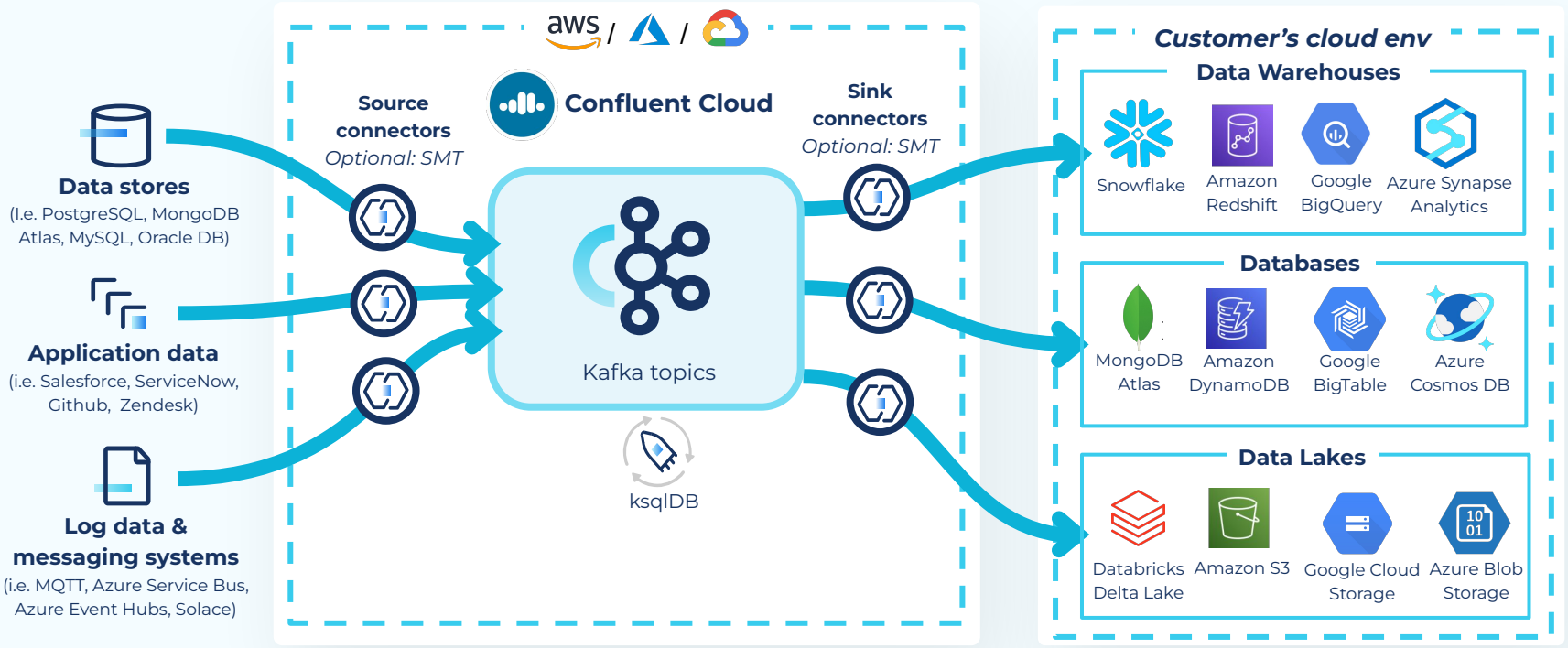
Centrally manage and monitor multi-cluster environments and security

For Developers

View messages, topics and schemas, manage connectors and build ksqlDB queries



Easily build real-time data pipelines to your data warehouse, database, and data lake



Agenda

01 **Introduction**
The What, the Why, Vision

02 **What's in the product?**
CFK, Confluent Cloud, Confluent Platform

03 **Configuration API**

04 **Connectors**

05 **Planning a development**

06 **Architecture**

07 **Confluent Platform Support**

08 **Workflow**

09 **Resources**

For successful deployment: Answer this upfront

Is my Kubernetes infrastructure ready for Confluent?

What is the right architecture for my deployment of Confluent on Kubernetes?

- Storage
- Networking
- Security

https://docs.confluent.io/operator/current/co-plan.html

Deployment workflow

At the high level, the workflow to configure, deploy, and manage Kafka Clusters using CFK is as follows:

1. Prepare your Kubernetes environment.

   For details, see Prepare Kubernetes Cluster for Confluent Platform.

2. Deploy Confluent for Kubernetes.

   For details, see Deploy Confluent for Kubernetes.

3. Configure Confluent Platform.

   For details, see Configure Confluent Platform.

4. Deploy Confluent Platform.

   For details, see Deploy Confluent Platform.

5. Manage Confluent Platform.

   For details, see Manage Confluent Platform with Confluent for Kubernetes.

Supported environments and prerequisites

- Confluent for Kubernetes 2.3.1 supports Kubernetes versions 1.18 - 1.23 (OpenShift 4.6 - 4.10) with any Cloud Native Computing Foundation (CNCF) conformant offering.
- Install kubectl.
- Configure the kubeconfig file for your cluster.
- Helm is required

What's in the product? The Base Constructs

Confluent Component Services CRD

Confluent Resources CRD

Secrets Abstraction

Troubleshooting plugin

First Class Automations

- Security
  - SASL/Plain, mTLS authentication
  - RBAC authorization
  - PEM -> Keystore, Truststore
- Fault Tolerance
  - Node failure, Rack (AZ/rack) awareness
- Networking
  - Load Balancer, Ingress configurations
- Upgrade/Updates
  - Safe Rolling update and upgrade

Kubernetes Native CRD for Confluent services

K8s Resources

Affinity

Annotations

Labels

Environment Variables

Tolerations

Enables: Integrate with Kubernetes Ecosystem

Configuration Overrides

Server properties

JVM

Log4j

Enables: Parity with all Confluent Platform features

Explore the API!

[API Reference documentation]

Kubernetes Native CRD for Confluent resources

topic.yaml

confluentRoleBinding.yaml

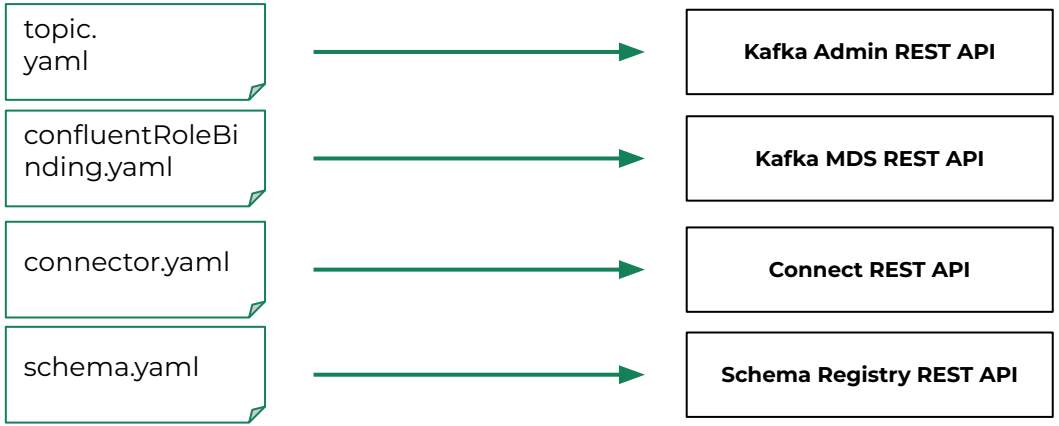
connector.yaml

schema.yaml

clusterlink.yaml

ksqldbApp.yaml

2H 2022 roadmap



Explore the API!
From CLI - $ kubectl explain confluentrolebinding.spec
On our docs: API Reference documentation

2 abstractions for Secrets Lifecycle Management

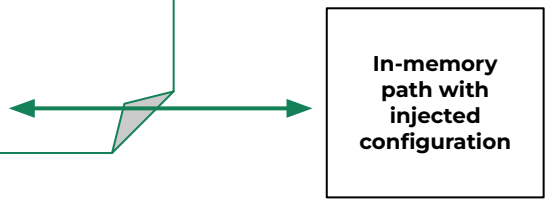
Kubernetes Secret

```
kafka.yaml
---
kafka:
  spec:
    listeners:
      internal:
        authentication:
          type: plain
            jaasConfig:
              secretRef: credential
```

Pod directory path

This is how Hashicorp Vault integrates for config securing

```
kafka.yaml
---
kafka:
  spec:
    listeners:
      internal:
        authentication:
          type: plain
            jaasConfigPassThrough:
              directoryPathInContainer: path
```



Troubleshooting Plugin

```
$ kubectl confluent
  cluster        Retrieve Confluent Platform cluster information.
  dashboard      Access to Confluent Platform UIs
  doc            Generate documentation for confluent-platform cli
  help           Help about any command
  http-endpoints Confluent Platform HTTP|s REST endpoints.
  migration      Convert Confluent Platform (CR) resources from v1 to v2.
  operator       Command related to Confluent Operator
  status         Confluent Platform status.
  support-bundle Support tool to capture and aggregate Confluent Platform(CP) deployment
  version        Confluent Platform build versions.
```

Confluent for Kubernetes 2.0 as Helm Chart

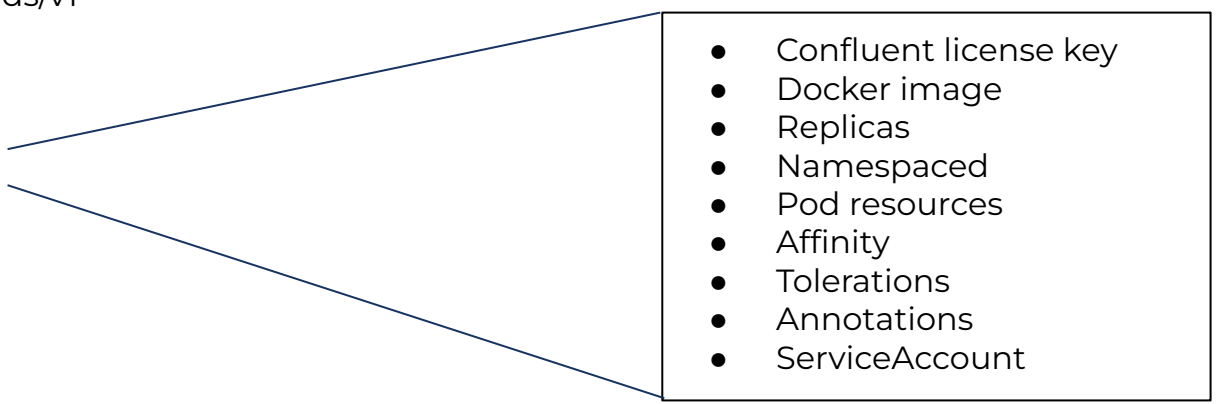
Available as a Helm Chart from our Helm Repository

$ helm repo add confluentinc https:/packages.confluent.io/helm

$ helm install operator confluentinc/confluent-for-kubernetes

/confluent-for-kubernetes

- /crds
- /resources/crds/v1
- /templates
- Chart.yaml
- values.yaml



Components and Resources as CRDs

Confluent component services and resources are defined as Custom Resources

$ kubectl apply -f my-kafka-cluster.yaml

The CRD API is designed with consistent configuration abstractions

Authentication	RBAC	LDAP
External Access	TLS	...

Use case

On-prem Oracle CDC to Marklogic

End-to-end streaming data pipeline that facilitates real-time data transfer from

1. On-premises relational datastore like Oracle PDB to a document-oriented NoSQL database, MarkLogic, with low latency, all deployed on the Kubernetes clusters provided by Google Cloud (GKE).
2. Apache Kafka® is leveraged using Confluent Cloud on AWS, depicting a true multi-cloud deployment.

Agenda

01 **Introduction**
The What, the Why, Vision

02 **What's in the product?**
CFK, Confluent Cloud, Confluent Platform

03 **Configuration API**

04 **Connectors**

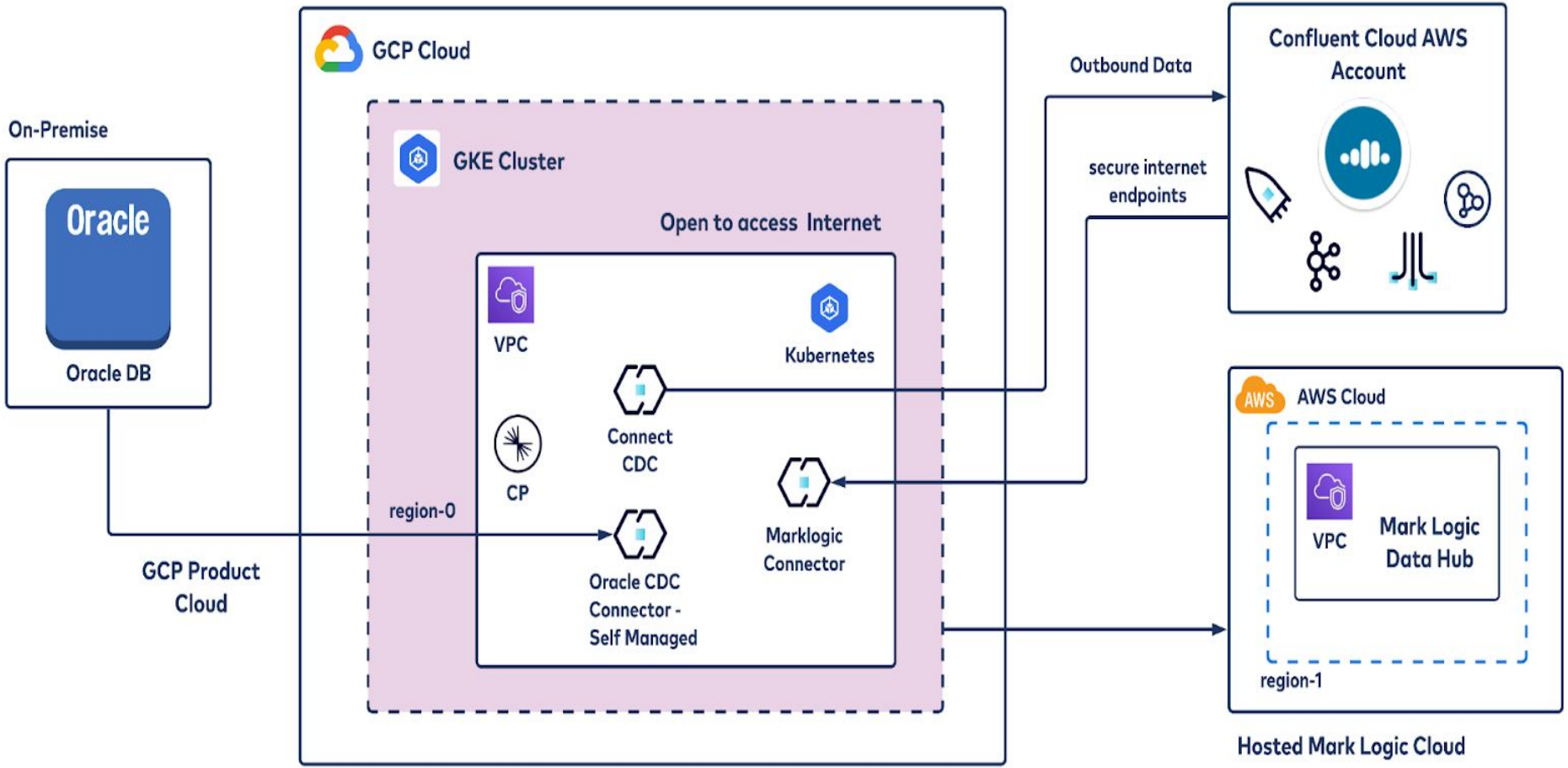
05 **Planning a development**

06 **Architecture**

07 **Confluent Platform Support**

08 **Workflow**

09 **Resources**


On-Premise
Oracle
Oracle DB
GCP Product Cloud
GCP Cloud
GKE Cluster
Open to access Internet
VPC
Kubernetes
Connect CDC
CP
region-0
Marklogic Connector
Oracle CDC Connector - Self Managed
Outbound Data
secure internet endpoints
Confluent Cloud AWS Account
AWS Cloud
VPC
Mark Logic Data Hub
region-1
Hosted Mark Logic Cloud

Prepare Docker images of connectors

We want to run the Oracle CDC and MarkLogic connectors in a Kubernetes cluster, you need to create custom Docker images of these connectors and make them available to CFK.

FROM confluentinc/cp-kafka-connect-base:latest

USER root

RUN confluent-hub install confluentinc/kafka-connect-oracle-cdc:2.0.1 --no-prompt

RUN confluent-hub install marklogic/kafka-marklogic-connector:1.6.0 --no-prompt

Deploy Confluent Platform on GKE

Confluent for Kubernetes to deploy and manage connectors and ksqlDB against Confluent Cloud and Schema Registry, it will rely on a Kubernetes operator to deploy components. You are going to refer to the Kubernetes namespace being used as ‘confluent.’

Confluent Platform is configured leveraging a configuration file from this example Git repository:

kubectl apply -f $CONFLUENT_HOME/confluent-platform.yaml

Desired Output:

connect.platform.confluent.io/connect created

ksqldb.platform.confluent.io/ksqldb-tls created

controlcenter.platform.confluent.io/controlcenter created

schemaregistry.platform.confluent.io/schemaregistry created

Connect section of yaml file is as follows

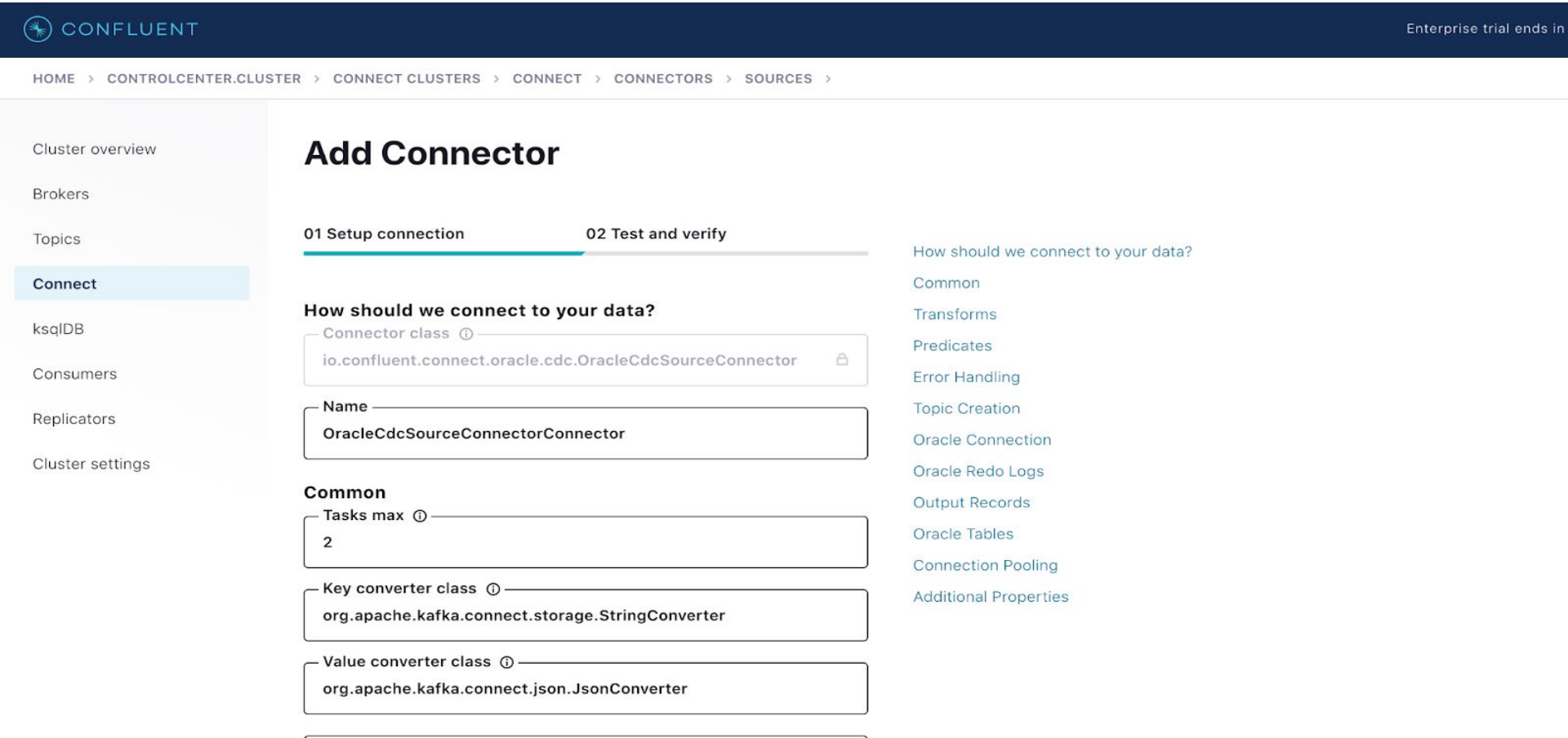
```
---
apiVersion: platform.confluent.io/v1beta1
kind: Connect
metadata:
  name: connect
  namespace: confluent
spec:
  replicas: 1
  image:
    application: confluentinc/cp-server-connect:7.0.1
    init: confluentinc/confluent-init-container:2.2.0-1
  dependencies:
    kafka:
      bootstrapEndpoint: kafka:9071
---
```

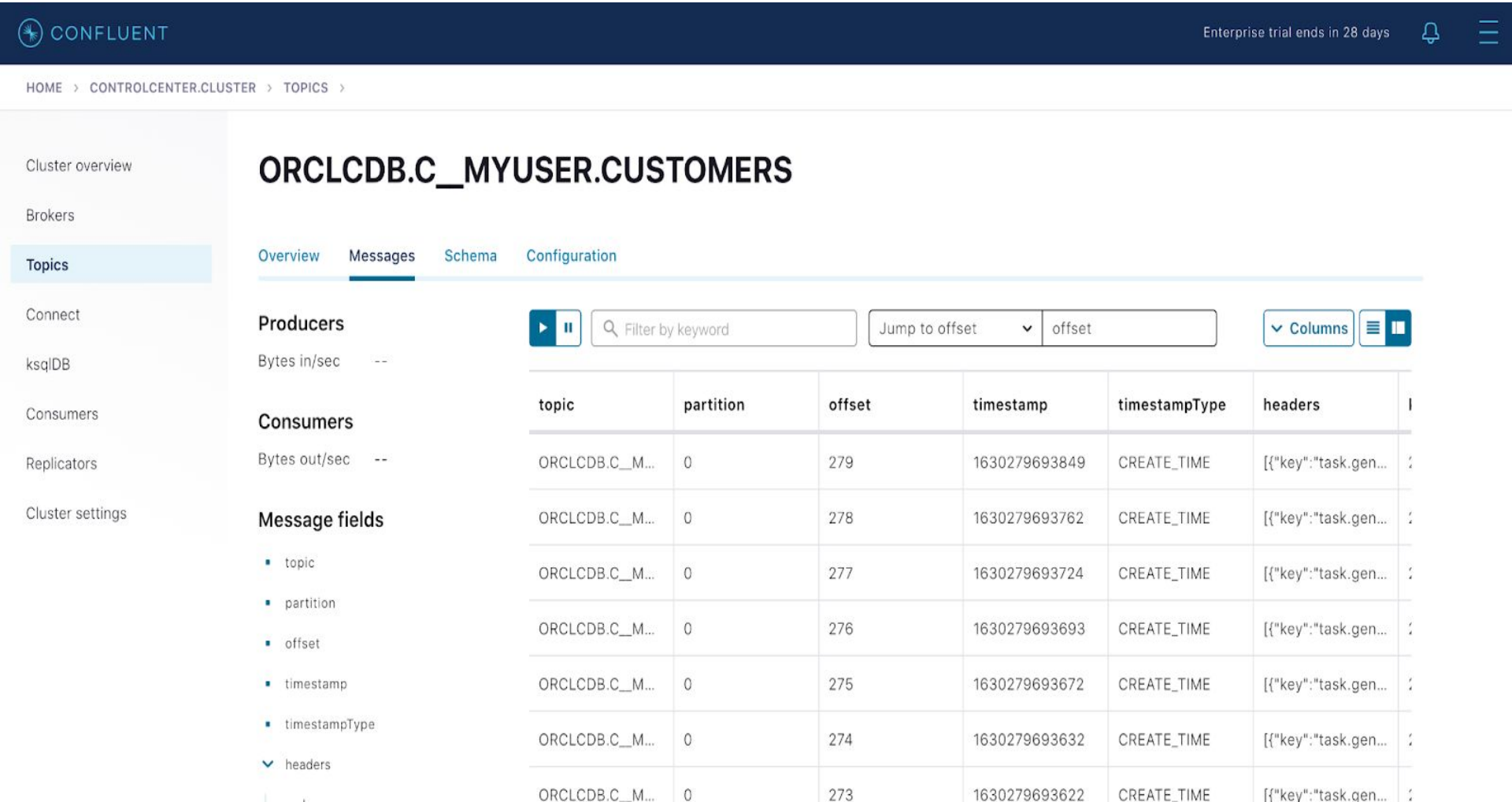
Deploy a self-managed Oracle CDC connector

The connector supports Oracle Database 11g, 12c, 18c, and 19c, and can either start with a snapshot of the tables or start reading the logs from a specific Oracle system change number (SCN) or timestamp. Identify the appropriate Oracle database (CDB, PDB, or RDS) and perform the following steps using Confluent's documentation:

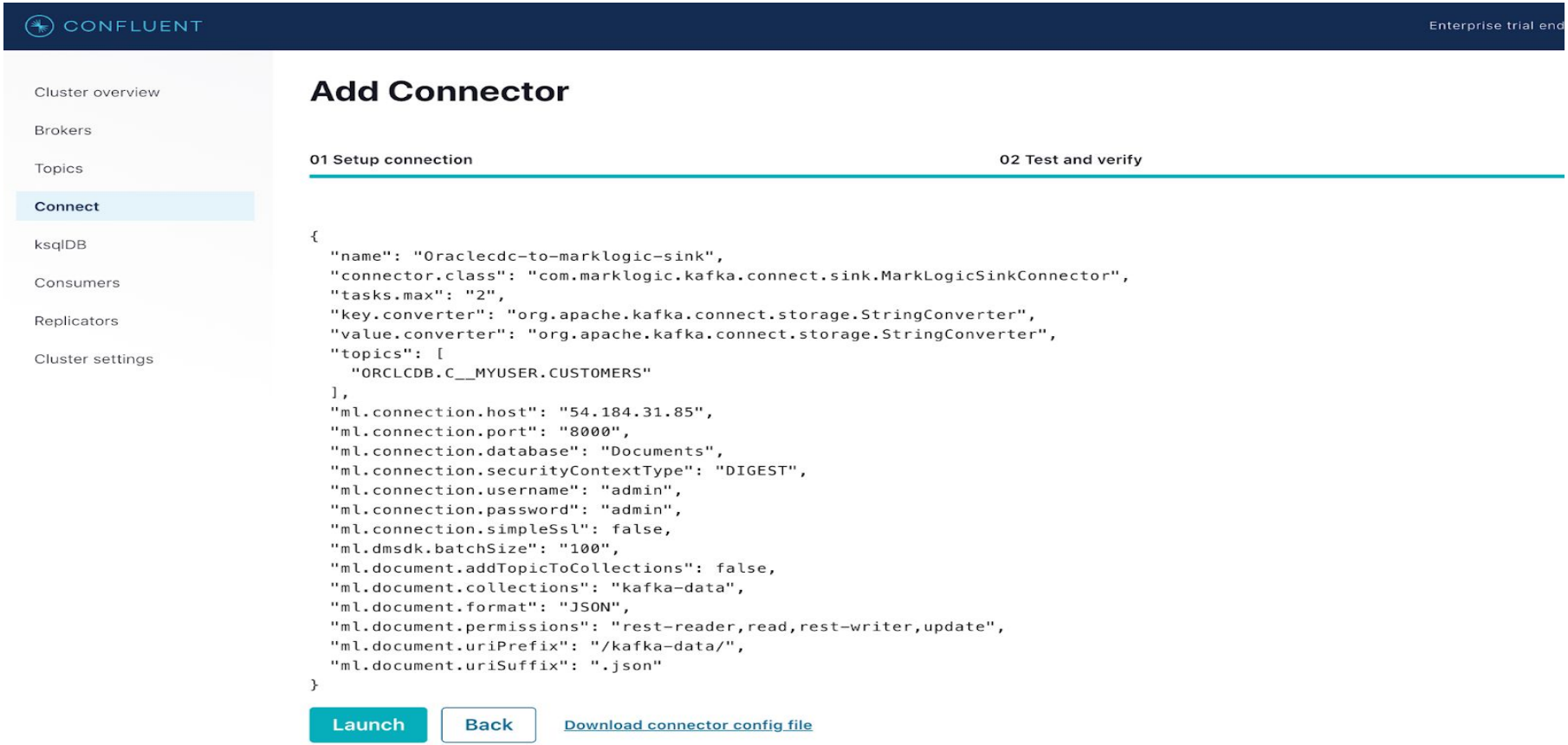
- Configure database user privileges
- Turn on ARCHIVELOG mode
- Enable supplemental logging for all columns
- Grant the user Flashback query privileges
- Validate startup configuration and prerequisite completion

Configure Oracle CDC connector



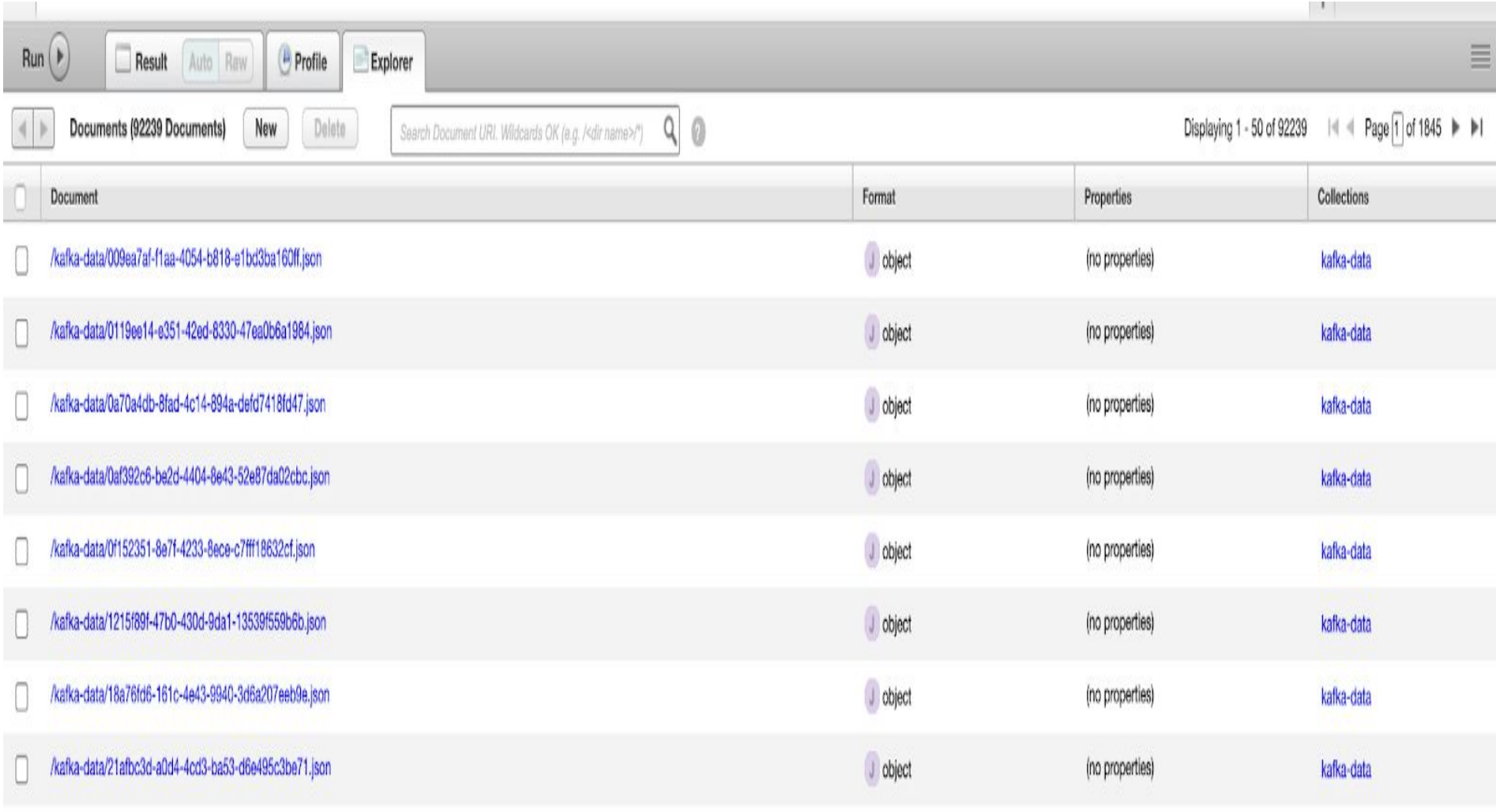

CONFLUENT
Enterprise trial ends in 28 days
HOME › CONTROLCENTER.CLUSTER › TOPICS ›
Cluster overview
Brokers
Topics
Connect
ksqlDB
Consumers
Replicators
Cluster settings
ORCLCDB.C__MYUSER.CUSTOMERS
Overview
Messages
Schema
Configuration
Producers
Bytes in/sec --
Consumers
Bytes out/sec --
Message fields
topic
partition
offset
timestamp
timestampType
headers
Filter by keyword
Jump to offset
offset
Columns
topic partition offset timestamp timestampType headers
ORCLCDB.C__M... 0 279 1630279693849 CREATE_TIME [{"key":"task.gen...
ORCLCDB.C__M... 0 278 1630279693762 CREATE_TIME [{"key":"task.gen...
ORCLCDB.C__M... 0 277 1630279693724 CREATE_TIME [{"key":"task.gen...
ORCLCDB.C__M... 0 276 1630279693693 CREATE_TIME [{"key":"task.gen...
ORCLCDB.C__M... 0 275 1630279693672 CREATE_TIME [{"key":"task.gen...
ORCLCDB.C__M... 0 274 1630279693632 CREATE_TIME [{"key":"task.gen...
ORCLCDB.C__M... 0 273 1630279693622 CREATE_TIME [{"key":"task.gen...

Deploy a self-managed MarkLogic Sink connector



REST API

```
curl -X PUT \
     -H "Content-Type: application/json" \
     --data '{
                {
  "name": "oraclecdc-to-marklogic-sink",
  "connector.class": "com.marklogic.kafka.connect.sink.MarkLogicSinkConnector",
  "tasks.max": "2",
  "key.converter": "org.apache.kafka.connect.storage.StringConverter",
  "value.converter": "org.apache.kafka.connect.storage.StringConverter",
  "topics": [
    "mltopic"
  ],
  "ml.connection.host": "34.221.56.67",
   "ml.connection.port": "8000",
```


Run
Result
Auto
Raw
Profile
Explorer
Documents (92239 Documents)
New
Delete
Search Document URI. Wildcards OK (e.g. /<dir name>/*)
Displaying 1 - 50 of 92239
Page 1 of 1845
Document
Format
Properties
Collections
/kafka-data/009ea7af-f1aa-4054-b818-e1bd3ba160ff.json
object
(no properties)
kafka-data
/kafka-data/0119ee14-e351-42ed-8330-47ea0b6a1984.json
object
(no properties)
kafka-data
/kafka-data/0a70a4db-8fad-4c14-894a-defd7418fd47.json
object
(no properties)
kafka-data
/kafka-data/0af392c6-be2d-4404-8e43-52e87da02cbc.json
object
(no properties)
kafka-data
/kafka-data/0f152351-8e7f-4233-8ece-c7fff18632cf.json
object
(no properties)
kafka-data
/kafka-data/1215f89f-47b0-430d-9da1-13539f559b6b.json
object
(no properties)
kafka-data
/kafka-data/18a76fd6-161c-4e43-9940-3d6a207eeb9e.json
object
(no properties)
kafka-data
/kafka-data/21afbc3d-a0d4-4cd3-ba53-d6e495c3be71.json
object
(no properties)
kafka-data

Agenda

01 **Introduction**
The What, the Why, Vision

02 **What's in the product?**
CFK, Confluent Cloud, Confluent Platform

03 **Configuration API**

04 **Connectors**

05 **Planning a development**

06 **Architecture**

07 **Security and Networking**

08 **Workflow**

09 **Resources**

Comprehensive Security on Kubernetes

Least Privilege Install

Namespaced deployments

Strict Rolebindings

Authentication

SASL Plain

mTLS

Authorization

Granular Role Based Access Control

Topic ACLs

Encryption

Client <> Kafka Broker TLS

Rest API TLS

Inter-broker TLS

Zookeeper TLS (new)

Security: New Functionality

Complete Security Automation out of the box

- Auto-generated TLS certificates
- Automated RBAC RoleBindings for CP component services

Flexible Security Configurations

- Configure multiple listeners, each with own TLS certificates and authentication schema
- Custom TLS certificate management

Dynamic TLS certificate management

Dynamic certs:

```
$ kubectl create secret tls
ca-pair-sslcerts --cert=ca.pem
--key=ca-key.pem

mykafka.yaml
---
kafka:
 spec:
  tls:
   autoGeneratedCerts: true
---
```

- **Provide** custom Root CA
- **Confluent Operator** generates, deploys and configures certificates for server components
- **Update** with new CA, Confluent Operator rolling updates certs

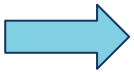
Custom TLS certificate management

Provide custom certs

```
Grouping Option 1:
      fullchain.pem
      privkey.pem
      cacerts.pem

Grouping Option 2:
      tls.crt
      tls.key
      ca.crt

Grouping Option 3:
      keystore.jks
      truststore.jks
      jksPassword.txt
```



Confluent Operator configures and updates server component certs

```
keystore.jks
truststore.jks
jksPassword.txt
```

External Access to Kafka

Load Balancer	Clients connect to Kafka using Kubernetes provider's load balancer
NodePort	Clients connect to Kafka at specified static ports
Ingress with port-based routing	Kubernetes Ingress controller manages clients' connection to Kafka using port-based routing
Ingress with SNI-based routing	Kubernetes Ingress controller manages clients' connection to Kafka using host-based routing

Agenda

01 **Introduction**
The What, the Why, Vision

02 **What's in the product?**
CFK, Confluent Cloud, Confluent Platform

03 **Configuration API**

04 **Connectors**

05 **Planning a development**

06 **Architecture**

07 **Security and Networking**

08 **Troubleshooting**

09 **Resources**

Troubleshooting

$ kubectl describe kafka

Status

- Conditions
- Listeners
- Services

Spec

$ kubectl get events

$ kubectl logs kafka-0

Technical Deep Dive Agenda

01 **Introduction**
The What, the Why, Vision

02 **What's in the product?**
The base constructs

03 **New Configuration API**

04 **Planning a deployment**

05 **Security Deep Dive**

06 **Networking Deep Dive**

07 **Confluent Platform Support**

08 **Troubleshooting**

09 **Resources**

Resources

Blog:
[Real-time data pipeline with Oracle CDC and MarkLogic using CFK and Confluent Cloud]

docs.confluent.io

github.com/confluentinc/confluent-kubernetes-examples

CRD API reference: https://docs.confluent.io/operator/current/co-api.html

Appendix