RED HAT SUMMIT

Navigating Hybrid Cloud Integration

HACKATHON

Presenters: Christina Lin, Hugo Guerrero, Nicola Ferraro, Gary Gaughan
Date Thursday 8th 14:00

Agenda

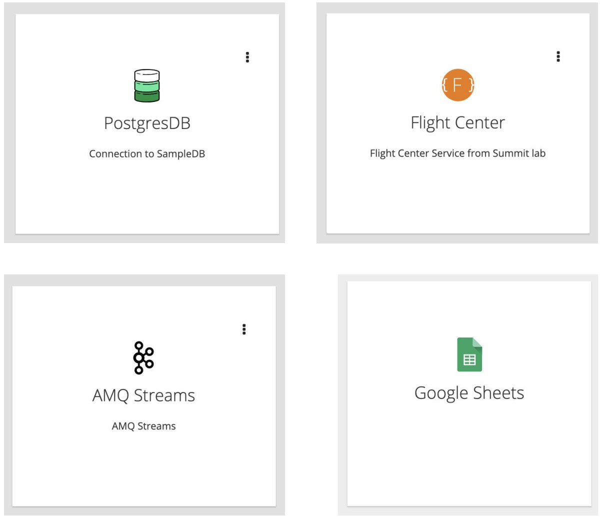
Three Hacks across two hours.

A set of integration challenges across green and brown field systems, SaaS applications, handling streaming of events, and usingAPIs.

Don’t worry if you are not familiar with the technology, there are guides, and we are here to guide you through the process.

At the end of this session ...walkaway with immediate integration knowledge and how to add value to your organisation on : Camel K, Kafka, APIs, Hybrid cloud environments and more...

Resources



A Redhat Fuse Online instance running on Openshift with:

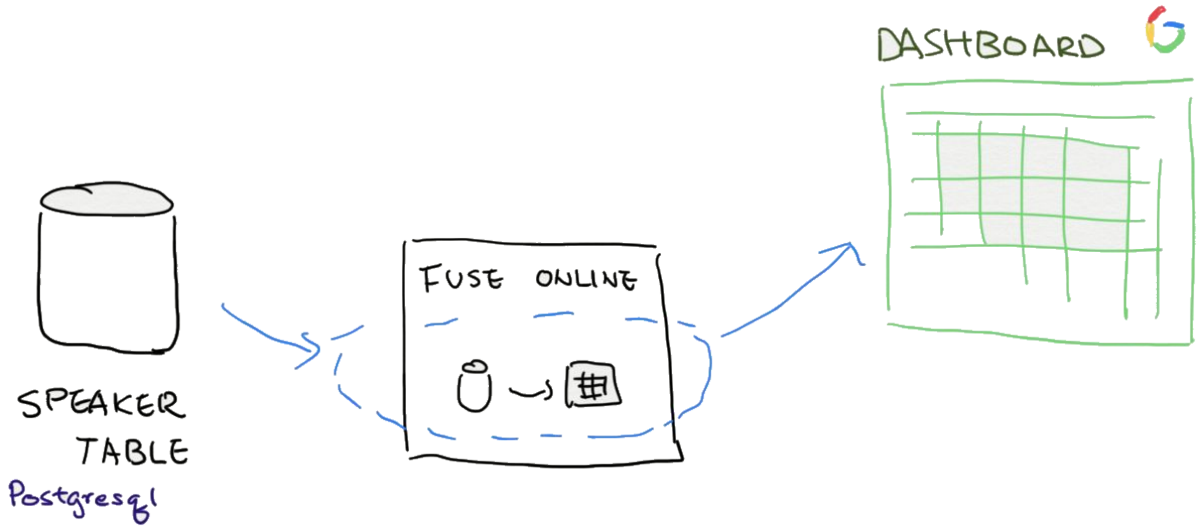
1. A legacy Database connection
2. A Google sheet connection
3. An AMQ Broker connection
4. An AMQ Streams Kafka instance
5. A Flight center booking API

Documentation for Fuse Online:

https://red.ht/2UNIWxy

HACK ONE

Hints:



Hints:

DB terminal access is available via the openshift console:
Applications->Pods->syndesis-db-####

Useful terminal commands
Psql
\l
\c sample db
\d

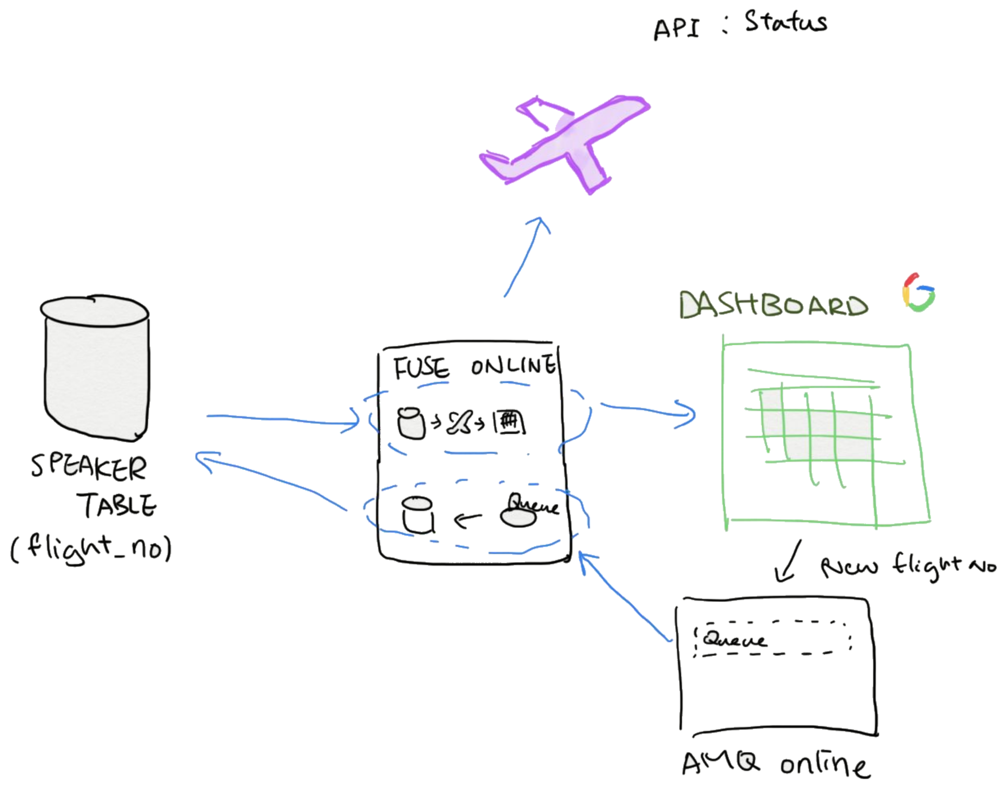
```
sampledb=# \d
List of relations
Schema | Name | Type | Owner
--------+-------------+----------+----------
public | contact | table | sampledb
public | speaker | table | postgres
public | todo | table | sampledb
public | todo_id_seq | sequence | sampledb
(4 rows)

sampledb=# select * from speaker;
speaker_name | departure_location | flight_no | traveler_cnt
---------------------+-------------------+----------+-------------
```

It's a good idea to sort them...

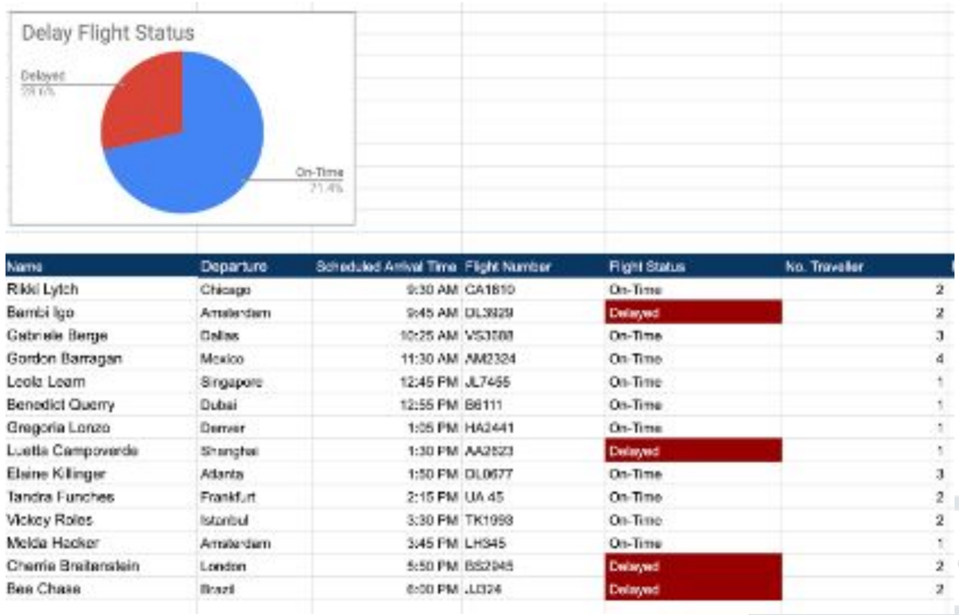
HACK TWO

Hints:


API : Status
DASHBOARD
FUSE ONLINE
Queue
SPEAKER TABLE (flight_no)
New flight No
Queue
AMQ online

Hints

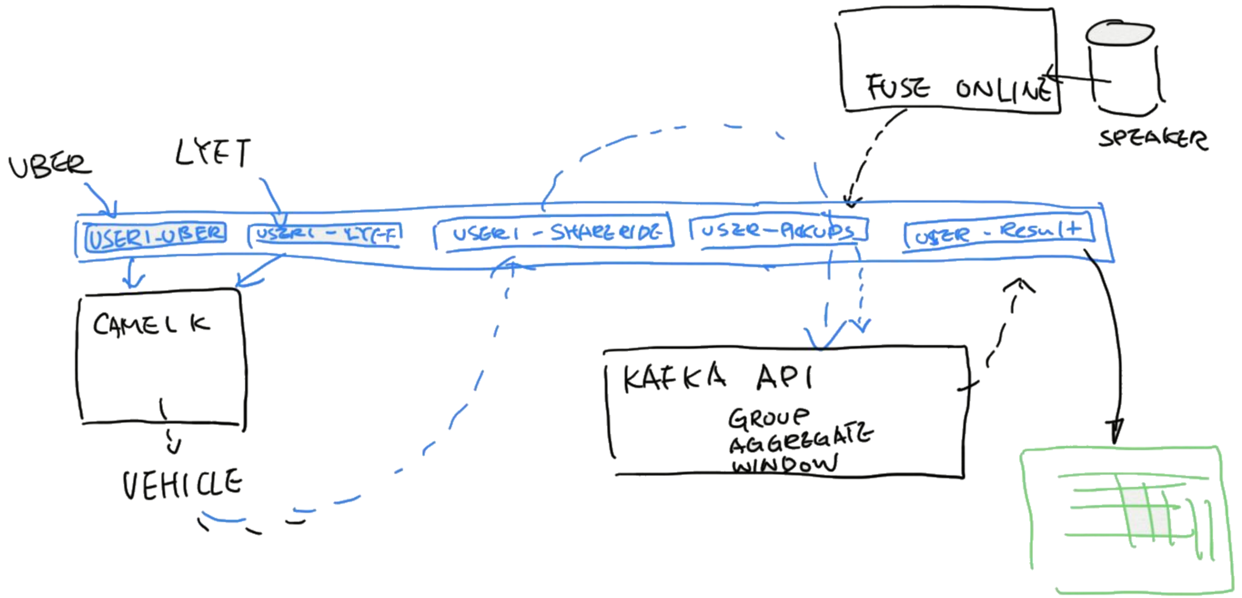
Inject the existing flight service into your integration
See screen result below



Name	Departure	Scheduled Arrival Time	Flight Number	Flight Status	No. Traveller
Rikki Lytch	Chicago	9:30 AM	CA1810	On-Time	2
Bambi Igo	Amsterdam	9:45 AM	DL3829	Delayed	2
Gabriele Berge	Dallas	10:25 AM	VS3788	On-Time	3
Gordon Barragan	Mexico	11:30 AM	AM2324	On-Time	4
Leola Leam	Singapore	12:45 PM	JL7465	On-Time	1
Benedict Querry	Dubai	12:55 PM	B6111	On-Time	1
Gregoria Lonzo	Denver	1:05 PM	HA2441	On-Time	1
Luetta Campoverde	Shanghai	1:30 PM	AA2523	Delayed	1
Elaine Killinger	Atlanta	1:50 PM	DL0677	On-Time	3
Tandra Funches	Frankfurt	2:15 PM	UA 45	On-Time	2
Vickey Roles	Istanbul	3:30 PM	TK1998	On-Time	2
Melda Hacker	Amsterdam	3:45 PM	LH345	On-Time	1
Cherrie Breitenstein	London	5:50 PM	BS2945	Delayed	2
Bee Chase	Brazil	6:00 PM	JL024	Delayed	2

HACK THREE

Hints:


UBER
LYFT
FUSE ONLINE
SPEAKER
USER1-UBER
USER1 - LYFT
USER1 - SHARERIDE
USER-PICKUPS
USER - Result
CAMEL K
VEHICLE
KAFKA API
GROUP
AGGREGATE
WINDOW

Hints (Camel K)

- Documentation: https://github.com/apache/camel-k

- The first thing a developer does is looking at what is flowing inside a route:

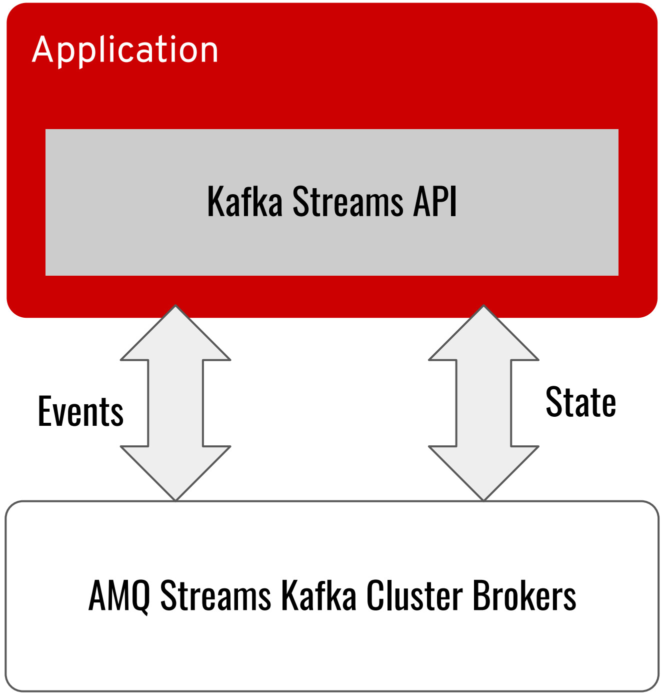
```
  from("...")
    .log("Received: ${body}")
```

- Dev mode is really useful: "kamel run It1.java It2.java **--dev**"

- Camel supports multiple dataformats for (un)marshalling data, like JSON and CSV

- When combining multiple heterogeneous sources, one may want to convert data to the same structure before "direct:process"-ing it

Kafka Streams



- Client library for stream processing (running outside brokers)
  - Embed stream processing features into regular Java applications
  - Create sophisticated topologies of independent applications
  - One-record-at-a-time processing (no microbatching)
- Kafka-to-Kafka semantics
  - Event/State management coordination
  - Stateful processing support
  - Transactions/exactly once

Kafka Streams
High Level Functional DSL

```
KStream words = builder.stream(“words”)


KTable countsTable = words.flatMapValues(value -> Arrays.asList(value.toLowerCase().split("\\W+")))
        .map((key, value) -> new KeyValue<>(value, value))
        .groupByKey(Serdes.String(), Serdes.String())
        .count(timeWindows, "WordCounts");

KStream counts = counts.toStream()


counts.to(“counts”)
```

Hints ()

Send speaker detail to a Kafka topic
See screen result below

https://tinyurl.com/rhsummithack

WINNERS OF SWAG

SWAG

RED HAT SUMMIT

THANK YOU

linkedin.com/company/Red-Hat

youtube.com/user/RedHatVideos

facebook.com/RedHatinc

twitter.com/RedHat

RED HAT SUMMIT

THANK YOU

linkedin.com/company/Red-Hat

facebook.com/RedHatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Kafka Streams
Abstractions

- KStream
  - Record stream abstraction
  - Read from/written to external topic as is
- KTable/GlobalKTable
  - Key/Value map abstraction
  - Read from/written to topic as a sequence of updates based on record key
  - Complex operations: joins, aggregations
- Stream/Table Duality
  - KStream -> KTable - read a stream as a changelog centered around the key
  - KTable -> KStream - table updates are produced as a stream
- Time windowing for aggregate operations