Проект по переводу отчетов международной страховой компании с Oracle на современный Big Data стек

Денис Шелемех
Иннотех
dshelem.com

PiterPy 2022

Описание доклада

- Доклад про бесшовную стыковку Data warehouses с Big Data stack
- Распределенная обработка данных и облака
- Будет полезен текущим и будущим Python Дата инженерам и разработчикам БД

О спикере

- Lead Data Engineer в компании Иннотех (InnoTech)
- Опытный, разносторонне подготовленный data engineer с опытом работы в IT более 10 лет и бэкграундами в веб-разработке и разработке корпоративных БД
- Был CTO и Founder собственного интернет-стартапа
- Люблю фантастику
- Воспитываю дочь и пытаюсь воспитывать серую голубую кошку

Содержание

С чего стартовали (ситуация клиента) - 1

- Клиентские планы страхования
- Большие кластера Oracle
- В кластере по 10-15 серверов
  - Тестовый кластер
  - Промышленный кластер
- Конфигурация сервера: 128 Гб оперативной памяти, 64 логических ядра процессора

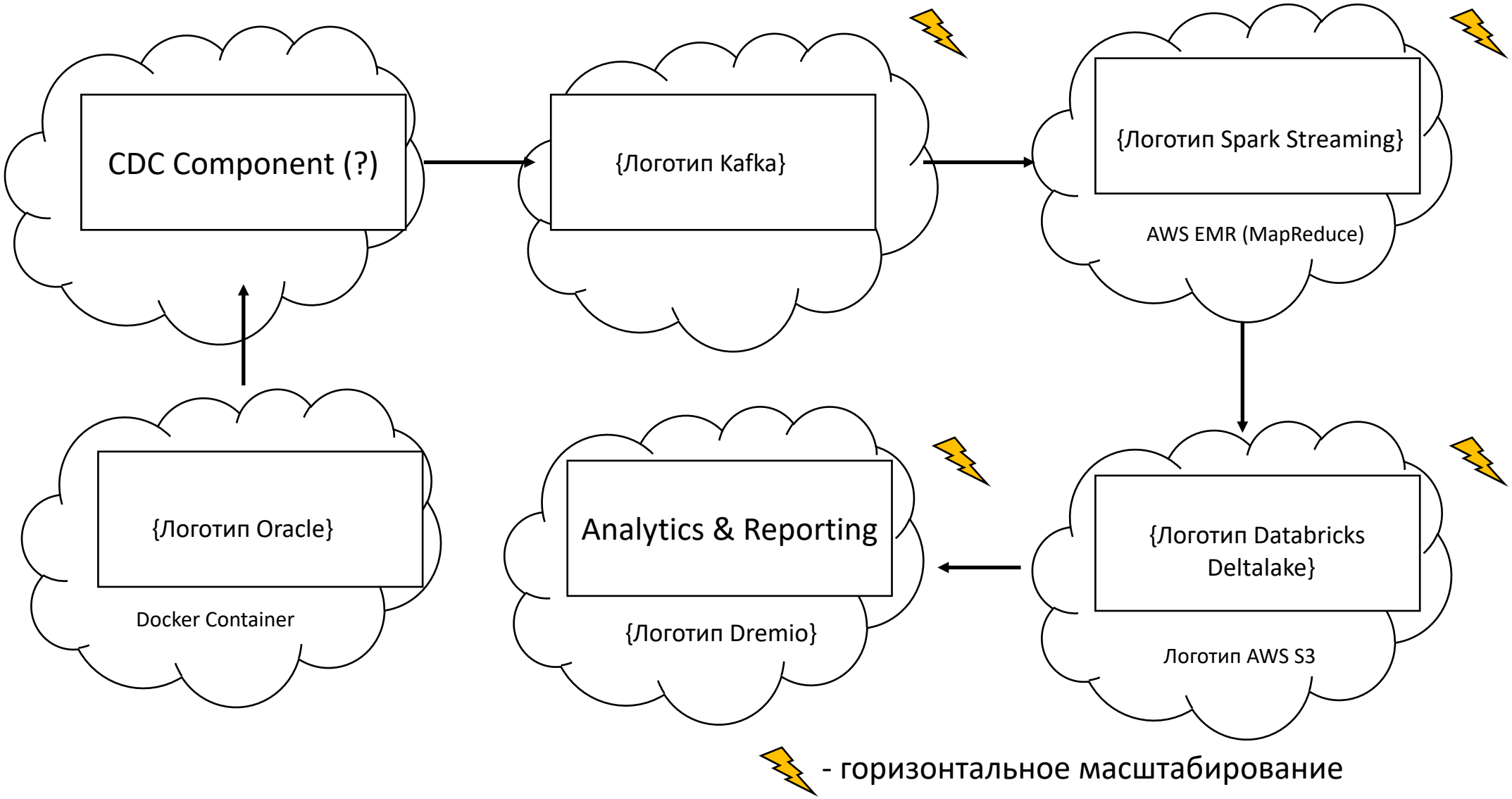
{Картинка – большая, неповоротливая корпоративная БД}

С чего стартовали (ситуация клиента) - 2

- ~150 отчетов в Oracle
- Простейшие отчеты строятся по несколько часов (на пике)
  - Вертикальное масштабирование и клиентские оптимизации – нет эффекта
- Выход - горизонтальное масштабирование и наши оптимизации
- Задача нашей команды – сделать Proof of Concept (PoC) решения на современном стеке Big Data

{Картинка – современный дата-центр}

Архитектура PoC



Настройка своей версии сервера Oracle

- Зачем?
  - На боевой кластер нас не пустят
- Сервер - маломощная EC2 машинка t3.small
- Использовали Docker контейнер
  - См. https://github.com/steveswinsburg/oracle19c-docker

- В целом, рекомендую только для тестовых целей

{Логотип Oracle} {Логотип Docker}

Поднятие БД на сервере Oracle (1) – начало

- Тестовый дамп БД объемом около 130 Гб в сжатом виде
  - ~0,1% от реального объема данных
- 3-4 дня на то, чтобы стать специалистом по восстановлению БД Oracle из дампа
- Oracle - 25 млн. строк кода на C [1)] из 1980-х
- Проблемы на этапе восстановления БД из дампа

{Логотип Oracle} {Динозавр (иконка)}

1) https://news.ycombinator.com/item?id=18442637

Поднятие БД на сервере Oracle (2) – проблемы

- Отсутствуют tablespace, роли и пользователи в дампе
- Анализ скриптом на Питоне огромного лога
  - Агрегация ошибок и генерация SQL-файла с инструкциями
- Следующая проблема - самые большие таблицы – партиционированные
  - Тот же разбор ошибок и предсоздание таких таблиц

{Картинка или иконка – механик возле автомобиля в ремонтной яме}

Поднятие БД на сервере Oracle (3)

- Скрипт SQL bd_pre_import.sql со 115 тыс. строк кода
- Скрипт bd_post_import.sql на 3,6 тыс. строк кода
- У клиента БД из дампа способен восстановить только сотрудник уровня Software Architect
  - Зауважал себя ☺

Переделка сервера Oracle на более производительную версию (1)

- EC2 t3.small + Oracle на Докере не тянет нагрузку
  - ~100% использование памяти и процессорной мощности
  - ∞ ожидание ответа на простые запросы к БД

1) переходим на более мощную машину
2) запуск полноценной версии сервера БД Oracle

Переделка сервера Oracle на более производительную версию (2)

- Нашли образ Linux и сервера Oracle на Amazon Marketplace
  - $300/мес. за его аренду
  - Поднимаем вручную
- Выяснилось, что сервер Oracle встает только на Oracle Linux
- Многочасовое гугление в поисках вменяемого мануала по поднятию сервера Oracle…
  - Найден здесь: https://oracle-base.com/articles/19c/oracle-db-19c-installation-on-oracle-linux-7

Переделка сервера Oracle на более производительную версию (3)

- Штатный Oracle мастер установки не заработал
  - Устанавливаем в silent mode через параметры командной строки
  - (… и помним про кодовую базу из 25 млн. строк кода родом из 1980-х…)

- …Не прошло и 5 рабочих дней, как мы имеем работающий сервер Oracle 🤩

{Картинка – люди в восторге}

Современный стек Big Data

- На контрасте с сервером Oracle и восстановлением БД из дампа, работа с компонентами современного стека Big Data была приятной

{Картинка – улыбающийся Fallout boy}

CDC компонент

- Change data capture – набор паттернов для отслеживания изменений данных (и передачи этих изменений для дальнейшей обработки)
- Таким образом, мы отслеживаем информацию о:
  - Вставке новых записей
  - Обновлении записей
  - Удалении записей

Выбор компонента для CDC

JDBC-based CDC

- 👍 Хорошо известная технология
- 👍 Надежность
- {Иконка - СТОП} Не поддерживает передачу информации об удалении записей

Выбор компонента для CDC

Debezium Kafka Connector

- 👍 Open-source project
- 👍 Полный спектр событий insert/update/delete
- 👍 Разработчики из команды Red Hat Linux
- 👍 Хорошая скорость разработки, поддержка
- 👍 Хорошее качество кода (github)

Последующее подтверждение книгой «Kafka. Definitive Guide»

Настройка брокеров Kafka (1)

- Два брокера Kafka (kafka + zookeeper на Docker образах от Bitnami) на машинах EC2 t3.medium
- На третьей - schema registry + avro serialization + debezium connector + сервис для мониторинга Prometheus
- После поднятия контейнеров все компоненты похожи на обычные Linux сервисы

Настройка брокеров Kafka (2)

- Avro Serialization (+Schema Registry) позволила значительно снизить дублирование информации
- Копались в исходном коде Debezium, чтобы понять до конца, как он работает

Pyspark – Streaming (1)

- Заполнение топиков данными из таблиц
- Сервисные компоненты на Питоне
  - SchemaRegistryClient
  - DeltaTable
  - Стриминг событий из Kafka

Pyspark – Streaming (2)

- Стриминг был требованием клиента
- Рекомендуемые параметры

```
# automatic schema evolution
# https://docs.delta.io/latest/delta-update.html#automatic-schema-evolution
spark.conf.set('spark.databricks.delta.schema.autoMerge.enabled', True)
```

```
# prevent producing a large number of small files
# https://docs.delta.io/latest/delta-update.html#performance-tuning
spark.conf.set('spark.databricks.delta.merge.repartitionBeforeWrite.enabled', True)
```

EMR Cluster (AWS)

- Запуск по требованию
  - Для стриминга
  - Для формирования отчетов
- Программные файлы и jar -> scp -> EMR Cluster master node

Pyspark – перенос отчетов

- Использовали Pyspark Dataframe API
- Рекомендуемые параметры

```
.config('spark.sql.streaming.schemaInference', 'true')
.config('spark.sql.sources.partitionOverwriteMode', 'dynamic')
```

- Основа – оптимизированный SQL код исходных отчетов

Pyspark – перенос отчетов - лайфхаки

- Вместо подзапросов (Oracle) – joinы
  - Enabler + классная оптимизация
- UDF функции на Питоне не писали (joinы)
  - Если бы делали -> Scala / Java

… Шеф, все пропало, или промежуточный фейл (1)

{Картинка в тему из фильма Бриллиантовая рука}

- Debezium не поддерживает скрытые столбцы в таблицах Oracle (а такие есть в большом количестве)
  - ‘select * from table’ -> данные только не из скрытых столбцов
  - Сделано в MySQL и Oracle, чтобы упростить клиентам развитие существующих приложений

… Шеф, все пропало, или промежуточный фейл (2)

- Что же делать?
  - Документация
  - Тикет на разработчиков

- … и тут нам повезло
  - Следующая версия, вышедшая через несколько дней, поддерживает скрытые столбцы

… промежуточный фейл (3)

- Чтобы делали если бы так не подфартило?
  - Дописывали бы под себя исходный код
    - Крупная продуктовая компания в РФ -> переделанный под себя Debezium первых версий

Databricks Deltalake

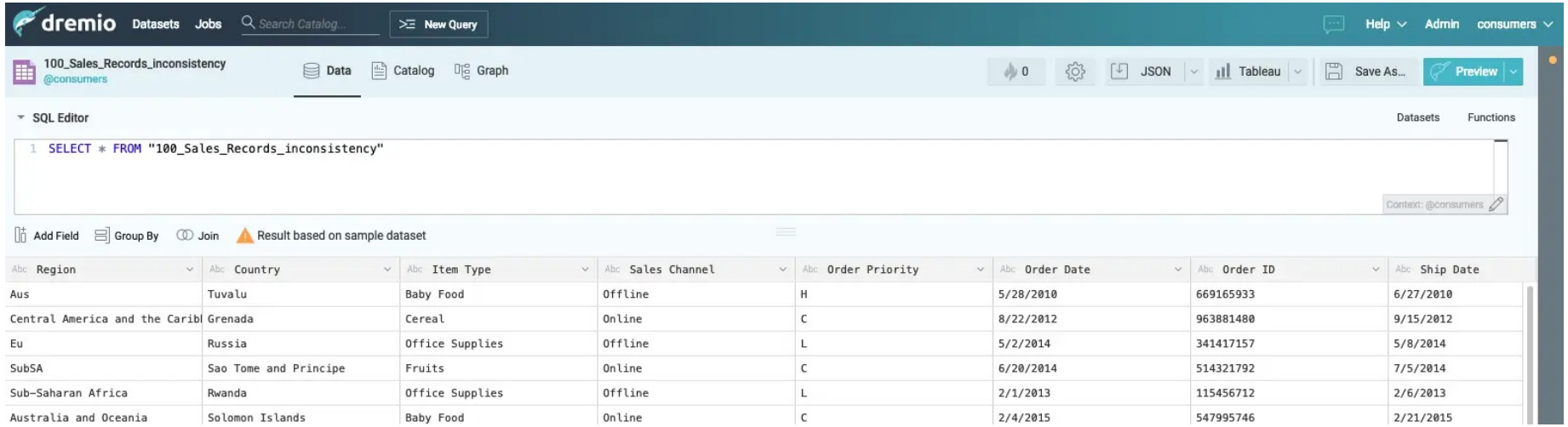
- Deltalake – интерфейс datalake поверх облака или HDFS
- Версионирование
- insert / update / merge по ID
- Поддержка Spark Streaming
- Поддерживается Dremio

- Рекомендую!

{ Картинка – улыбающийся Fallout boy }

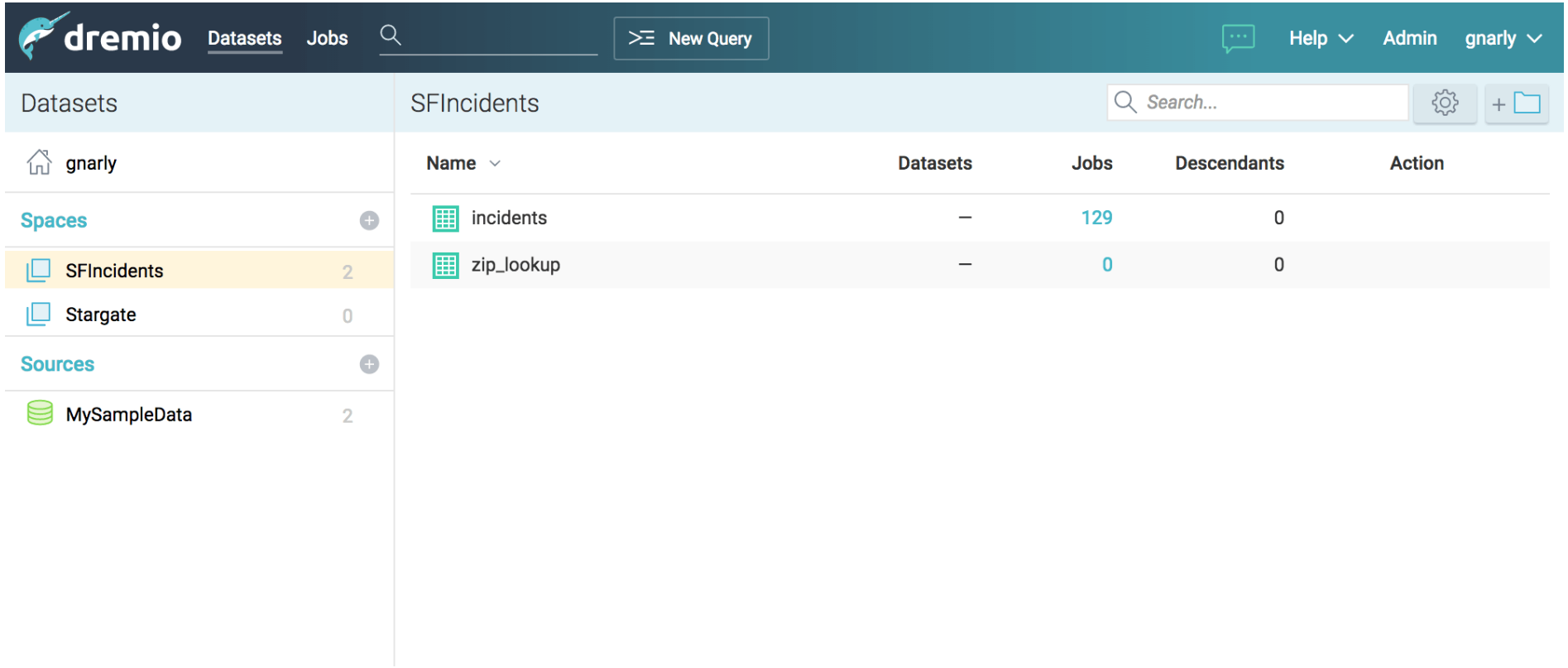
Dremio (1)

- Платформа для репортинга/аналитики поверх облака



Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date
Aus	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010
Central America and the Caribl	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012
Eu	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014
SubSA	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014
Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013
Australia and Oceania	Solomon Islands	Baby Food	Online	C	2/4/2015	547995746	2/21/2015

Dremio (2)


dremio
Datasets
Jobs
New Query
Help
Admin
gnarly
Datasets
SFIncidents
Search...
gnarly
Spaces
SFIncidents 2
Stargate 0
Sources
MySampleData 2
Name
Datasets
Jobs
Descendants
Action
incidents – 129 0
zip_lookup – 0 0

Dremio (3)

- 👍 Open-source project
- 👍 Поддерживает Deltalake
- 👍 Работает поверх облака
- 👍 Передача данных в BI systems
- 👍 User-friendly interface (SQL)
- 👍 Обновление данных для частых отчетов по расписанию

Dremio (4)

- 👎 Весьма требовательный к ресурсам
- 👎 Слабый и странный парсер SQL
- 👎 Нестабильная работа

- Сырой продукт


Первое Демо
{Шрифт - в формате 80-х}

Первое демо

- 25 человек (против 3-5 человек ранее)
- Цепочка от вставки, обновления записей в Oracle до записи в S3 и появления в Dremio
  - 3-4 репетиции
  - Кнопки жал один человек, а озвучивал другой
- Wow-момент 🏆

- …после успешного демо клиент утвердил Proof of Concept

Выводы и рекомендации

- Беритесь за непонятные для себя и сложные задачи. Проявите себя, прокачаете знания и навыки. И добавите пару ачивок в резюме ☺
- Стоит доверять себе и своей интуиции при выборе технологий для разработки проекта
- Все внутренние и внешние клиенты – визуалы
- Коммуникация крайне важна для успешной работы с клиентами

Аналитика (1)

- С Oracle работать сложно при любых сценариях
  - Требователен к ресурсам
  - Отвратительно работает в Докере
- Debezium крайне хорош при необходимости выборки CDC из спектра БД

Аналитика (2)

- Kafka отлично масштабируется и держит нагрузку
  - В рамках PoC достойно работал в Докере
- Python и Pyspark 👍. Максимально используем joiны.
  - UDF пишем на Scala/Java
- Deltalake – отличный интерфейсный движок поверх облака
- Dremio – сыроват

О компании Иннотех

- Основной подрядчик технологического развития Российского банка из Топ-10
- У нас удаленка, интересные задачи, масштабный проект, классная зарплата и плюшки
- За подробностями заходите на https://career.inno.tech/career ☺

КОМАНДА ИННОТЕХ

Вопросы? Комментарии?