220-380
12 фев в 11:57

Аналитика рынка профессии с историей (Yandex Data Lake + Power BI)

Простой 7 мин 2.9K

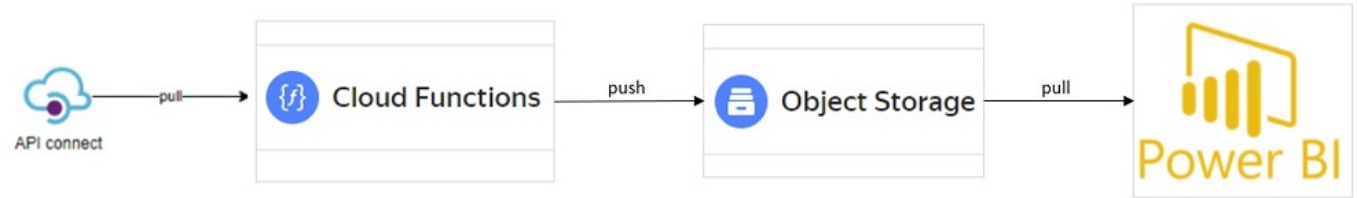
Python*, Визуализация данных*, Карьера в IT-индустрии, Облачные сервисы*, Data Engineering*

Туториал

Как-то я написал статью Аналитика рынка профессии в три клика. В ней мы подключались к сайту HH.ru из Power BI, забирали вакансии и строили дашборд для анализа.

Но подключение, описанное в статье, лишает анализ вакансий историчности. В дашборде хранится только один «снимок» данных на момент загрузки в него. Другими словами, при нажатии кнопки *Обновить* все данные в дашборде перезатрутся. Было бы интересно видеть общую картину во времени? Мне да. Тем более, сделать это не особо сложно, и стоит пару сотен рублей в год. При выборе инструментов был принцип - максимально просто и дешево.

Для того чтобы добавить историчности в дашборд обратимся к облачным технологиям. Нам нужно место, куда мы будем складывать и хранить наши «снимки» данных. Причем сырые ответы сайта в формате JSON. К этому хранилищу мы и будем подключаться из Power BI. Небольшая схема архитектуры на рисунке ниже. Алгоритм следующий, по расписанию запускается CloudFunction на Python, которая делает запрос к API сайта и складывает ответы в Stroge. Далее сырые ответы трансформируются и сохраняются в формате parquet, к которому и будет подключаться Power BI. Причем, при таком способе подключения в личном кабинете Power BI можно запланировать обновление (запланировать обновление можно не со всеми источниками).



Архитектура

Я решил использовать инфраструктуру Яндекс Облака. На мой взгляд, облака западных вендоров, например Microsoft Azure, более зрелые и удобные. Но проблема в том, что при регистрации нужно указать банковскую карту, вероятно это отрежет большую часть читателей.

Далее по шагам.
Заходим в Яндекс Облако, регистрируемся, привязываем банковскую карту и создаем два сервиса:

- Яндекс функции
- Хранилище S3, оно же Data Lake, оно же Yandex Object Stroge

Yandex Object Stroge

В Yandex Object Storage нужно создать каталог, далее перейти в Storage и создать bucket. В bucket настройка доступа Публичный, я не особо заморачивался с доступом. Так же необходимо создать статический ключ доступа к бакету. Действуем по инструкции от Яндекса.

Yandex Cloud Functions

Это самое сложное, но вы держитесь :-). В Cloud Functions нужно создать три функции со средой выполнения на Python. Код функций приложил ниже. Таймаут функций не менее 300 секунд. И триггеры к ним, разделенные по интервалам не менее 30 минут. У меня получилось вот так:

Триггеры

Фильтр по имени | Все триггеры | Все ресурсы

Имя	Описание	Статус	Тип триггера	Источник	Тип ресурса	Имя ресурса
-triger-every-day-step-3	—	Active	Таймер	0 12 ? * * *	Функция	-to-datamart-step-3
-triger-every-day-step-1	—	Active	Таймер	0 10 ? * * *	Функция	-list-vacancy-id-step-1
-triger-every-day-step-2	—	Active	Таймер	0 11 ? * * *	Функция	-list-vacancy-id-step-2

Триггеры функций

В редакторе кода для каждой функции нужно создать два файла, код прилагаю. Первый requirements.txt с необходимыми пакетами Python. И второй с кодом Python. В коде меняем значения переменных на ваши, они помечены, найдете.

▸ Код функций

Файл *requests.csv* нужен для хранения списка запросов, по которым вам требуется произвести парсинг. Этот файл нужно положить в ВАШ БАКЕТ - Lakehouse - Config - requests.csv.

Функция 1 считывает из бакета файл requests.csv с вакансиями, по которым мы хотим скачать ответы hh.ru, и записывает id вакансий в файл All_id_for_parse.csv. Питонисты, слишком не критикуйте код :). После этого функция 2 запрашивает информацию из hh.ru отдельно по каждому id вакансии. Два шага нужны потому что обращение по id каждой вакансии позволит получить более детальный json, чем при общей поисковой выдаче в функции 1. Функция 3 создает витрину - файл *datamart.parquet* с извлеченными данными из файлов json. Этот файл и будет нужен для Power BI.

> Вместо трех функций можно было бы оставить две, если использовать новый сервис Yandex query. Но на данный момент сервис не умеет читать вложенный json, коими являются наши json.

Если все получилось, и функции отработали, можно запускать Power BI.

Power BI

Для скачивания витрины в Power BI нужно использовать источник *Файл - Parquet*. В источнике прописать ссылку на файл, как "http://ВАШБАКЕТ.storage.yandexcloud.net/Lakehouse/Data/proccessed/datamart".
В файле *datamart.parquet* некоторые поля сохранены как словари и списки в виде текста, поэтому нужно будет применить преобразование в Power BI. В Power qwery делается это в два клика. В принципе все. Можно строить дашборд. Мой дашборд открыт и доступен по ссылке. Он двухстраничный.

Итог

Мы получили целый проект - Data Lake + BI. Его можно использовать в целях быстрого развертывания для демонстраций. Стоит дешево. Разумеется, решение нацелено на небольшие и возможно средние объемы данных. На данный момент цифры такие:

- За 15 дней пришло 700 MB сырых данных, формат json
- 700 MB сырых данных = 0,85 MB данных в витрине, формат parquet

- В год витрина прирастает на 21 MB, то есть загрузка может жить лет 10
- Каждый день тратится по 84 копейки, в год 295 р.
- Загрузка полностью автоматизирована; просто открывая отчет, можно просматривать новые данные

Теги: yandex.cloud, power bi, аналитика вакансий, аналитика данных, hr-процесс, поиск вакансий, data engineering

Хабы: Python, Визуализация данных, Карьера в IT-индустрии, Облачные сервисы, Data Engineering

6 Карма 0 Рейтинг

Игорь Горин @220-380

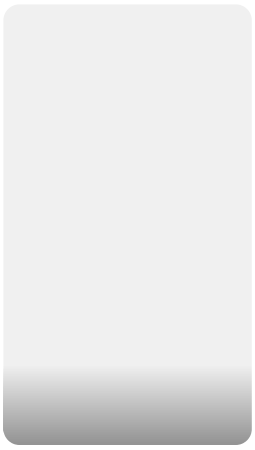
BW/BI developer

Комментарии 11

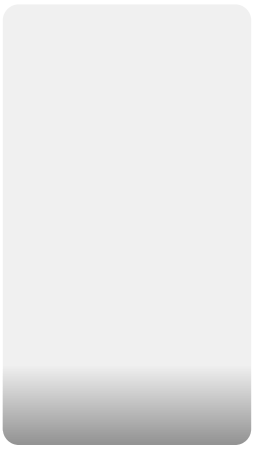
Публикации

ЛУЧШИЕ ЗА СУТКИ ПОХОЖИЕ

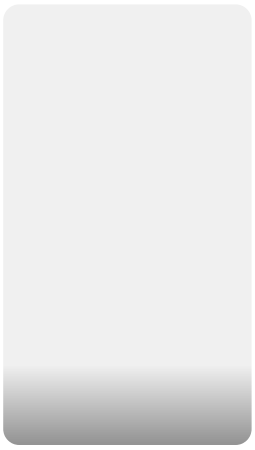
ИСТОРИИ



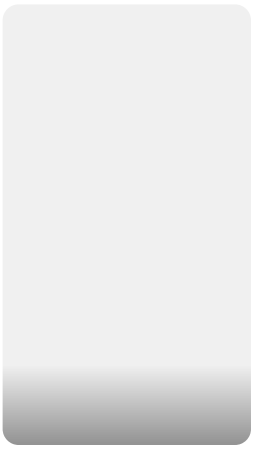
Топ-7 годноты из блогов компаний



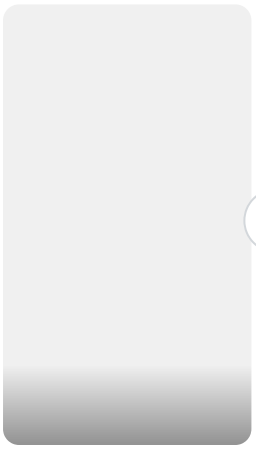
Что делать после того, как вас взяли



Где в России полетать на параплане? Чегем



Как почувствовать себя хакером



Один день джуна глазами мидла

Чем похо

РАБОТА

Python разработчик
162 вакансии

Django разработчик
60 вакансий

Data Scientist
124 вакансии

Все вакансии

Хабр

Настройка языка

Техническая поддержка

Вернуться на старую версию