

Программа курса «AIRF: Data Pipeline на Apache Airflow»

О курсе: Наш углубленный 6 дневный курс Apache Airflow поможет освоить самый популярный инструмент для оркестрации, который используют в ведущих IT-компаниях. Вы научитесь превращать хаос в данных в управляемые, автоматизированные data pipelines, став настоящим архитектором данных.

Забудьте о медленных процессах. С Apache Airflow вы научитесь создавать и отслеживать сложные рабочие процессы как код. Этот практический тренинг проведет вас от создания первого DAG до продвинутых техник, включая настройку отказоустойчивых конвейеров и интеграцию с Big Data. Наш Apache Airflow курс — это ваша прямая инвестиция в карьерный рост.

Аудитория:

- Инженер данных
- Аналитик данных
- Разработчик DWH

Уровень подготовки:

- Знание базовых команд Linux.
- Базовый опыт программирования на Python/bash.
- Опыт программирования на SQL и понимание принципов работы реляционных баз данных.

Продолжительность курса: 24 академических часа, 5 дней

Содержание программы

1. Введение в Apache Airflow: ключевые концепции и подготовка к работе.

- **Что такое Airflow?** Знакомство с платформой и её ролью в современной инженерии данных.
- **Почему Airflow?** Сравнение с аналогами и конкурентами Luigi и Prefect, и разбор ключевых преимуществ.
- **История и архитектура:** Краткий экскурс в историю создания и обзор основных компонентов («Китов») Airflow.

Комплексная практика:

- Практическое занятие по настройке рабочего образа Apache AirFlow для дальнейшей работы.

2. Основы работы с Apache Airflow: от теории к первому конвейеру данных.

Архитектура и компоненты: Детальное рассмотрение верхнеуровневой архитектуры, основных компонентов и исполнителей (Executors), включая LocalExecutor.

Ключевые концепции:

- **DAG:** Что такое направленный ациклический граф, его основные параметры и жизненный цикл.
- **Операторы (Operators):** Обзор различных видов операторов, их параметров и принципов композиции (Composition).
- **Интерфейс:** Знакомство с **Web UI** для мониторинга и управления рабочими процессами.

Создание первого DAG:

- Изучение **TaskFlow API** для упрощенного написания кода.
- Работа с базовыми операторами: **EmptyOperator**, **BashOperator**, **PythonOperator**.

Комплексная практика:

- Написание первого DAG с использованием **PythonOperator**.
- Применение переменных (**Variables**) и подключений (**Connections**) для создания динамических и переиспользуемых задач.
- Работа с сенсорами (**Sensors**), включая **ExternalTaskSensor**, для создания зависимостей от внешних событий.

3. Расширенные возможности Apache Airflow: продвинутые техники и интеграции.

Управление потоком выполнения:

- **Trigger Rules:** Настройка правил запуска задач в зависимости от статуса предыдущих.
- **Backfill & Catchup:** Механизмы для запуска прошедших и пропущенных выполнений DAG.

Динамические пайплайны:

- **Templates & Macros:** Использование шаблонов Jinja и макросов для создания гибких задач.
- **Hooks, XComs & Dynamic Tasks:** Обмен данными между задачами и создание динамически генерируемых пайплайнов.
- **Интеграция с базами данных:** Применение **PostgresOperator** для работы с PostgreSQL.

Комплексная практика:

- Применение **Trigger Rules** для управления логикой ветвления в DAG.
- Использование **PostgresOperator** для взаимодействия с базой данных.
- Организация сложных DAG с помощью **TaskGroup** для улучшения читаемости и структурирования.
- Реализация обмена данными между задачами с помощью **XComs**.