

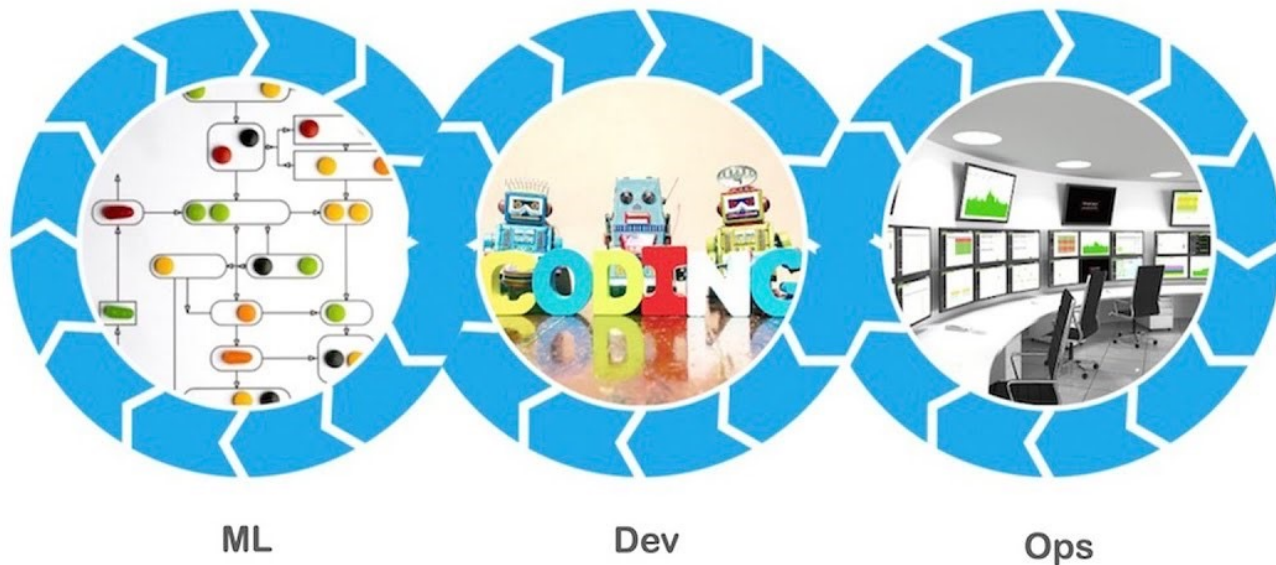
Автоматизация разработки и
эксплуатации программного
обеспечения

MLOps

Определение

- **MLOps** — набор практик, нацеленных на надежное и эффективное развертывание и поддержание моделей машинного обучения на производстве.

$$\text{MLOPS} = \text{ML} + \text{Dev} + \text{Ops}$$



- Слово является смесью слов "машинное обучение" (ML) и практик непрерывной разработки — [DevOps](#) в области программного обеспечения. Модели машинного обучения тестируются и разрабатываются в изолированных экспериментальных системах. Когда алгоритм готов к запуску, MLOps используется учеными в области данных, DevOps, и инженерами машинного обучения для его доставки в производственные системы

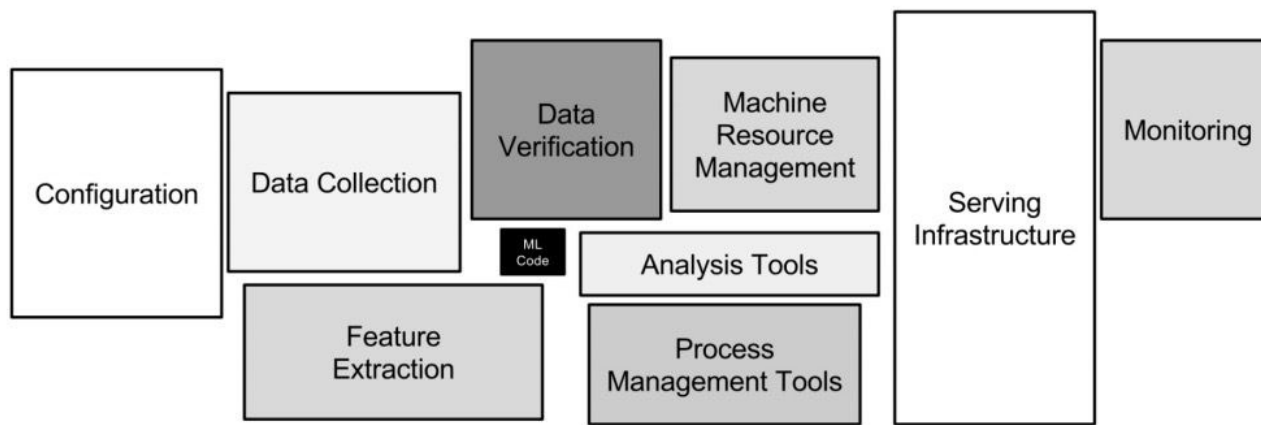
История появления

- Легко поверить, что подобная инженерная дисциплина зародилась в недрах большой IT-компании. Так что можно довериться версии, что MLOps, как осмысленный подход, зародился в Google, где Д. Скалли (D. Sculley) пытался спасти свои нервы и время от рутинных задач вокруг вывода ML-моделей в прод. Теперь Скалли широко известен как «The Godfather of MLOps» — одноименный подкаст легко найти в интернете.
- Д. Скалли начал рассматривать работу с моделями с точки зрения технического долга команды. Да, можно быстро выпускать новые версии моделей, но стоимость поддержки получившейся системы будет сильно сказываться на бюджете компании.
- Результатом его работы стала статья «Hidden Technical Debt in Machine Learning Systems» ("Скрытый технический долг в системах машинного обучения"), которая вышла в 2015 году на конференции NeurIPS. Именно дату публикации этой статьи можно считать точкой отсчета существования MLOps.

- D. Sculley (Д. Скалли) «Hidden Technical Debt in Machine Learning Systems»

Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley, gholt, dgg, edavydov, toddphillips}@google.com
Google, Inc.



Ключевые этапы

- 1. Автоматизация процессов
- 2. Подготовка и обучение моделей
- 3. Использование контейнеров
- 4. Управление версиями
- 5. Непрерывная интеграция и развертывание
- 6. Мониторинг и обратная связь
- 7. Совместная работа и командная разработка
- 8. Возможность использовать собранные контейнеры во внешних приложениях

1. Автоматизация процессов: MLOps предполагает автоматизацию всех этапов жизненного цикла моделей машинного обучения. Автоматизация процессов сбора данных из различных источников (баз данных, API, файловые хранилища и т.д.), автоматизация предобработки данных, включающей очистку, нормализацию, трансформацию и другие необходимые преобразования, использование инструментов для эффективного управления данными, включая версионирование, отслеживание изменений и репликацию.

2. Подготовка и обучение моделей:
автоматизация процессов подготовки,
очистки и разбиения данных на
обучающую и тестовую выборки,
автоматизация процесса обучения
моделей, включая выбор и настройку
гиперпараметров, использование
инструментов для отслеживания
экспериментов, сравнения результатов и
управления версиями обученных моделей.

3. Использование контейнеров: контейнеры позволяют упаковывать модели машинного обучения вместе со всеми необходимыми зависимостями (библиотеки, конфигурационные файлы) в единый образ. Это обеспечивает изоляцию модели от окружающей среды и гарантирует, что она будет работать одинаково во всех средах (разработка, тестирование, эксплуатация).

4. Управление версиями: MLOps требует систем управления версиями для кода, данных и моделей, чтобы обеспечить надежность и воспроизводимость процессов разработки и эксплуатации моделей. Использование систем контроля версий, таких как Git, позволяет отслеживать и управлять изменениями в исходном коде моделей, скриптов, конфигурационных файлов и другом вспомогательном коде.

5. Непрерывная интеграция и развертывание: MLOps предполагает использование CI/CD/CT практик для автоматизации процессов интеграции, тестирования и развертывания моделей.

6. Мониторинг и обратная связь: Важной концепцией MLOps является мониторинг производительности моделей в реальном времени, а также обратная связь для улучшения качества моделей, на основе полученных данных. Для моделей, развернутых в производственной среде, необходимо осуществлять мониторинг ключевых метрик производительности, таких как точность, ошибки, время отклика, потребление ресурсов и т.д.

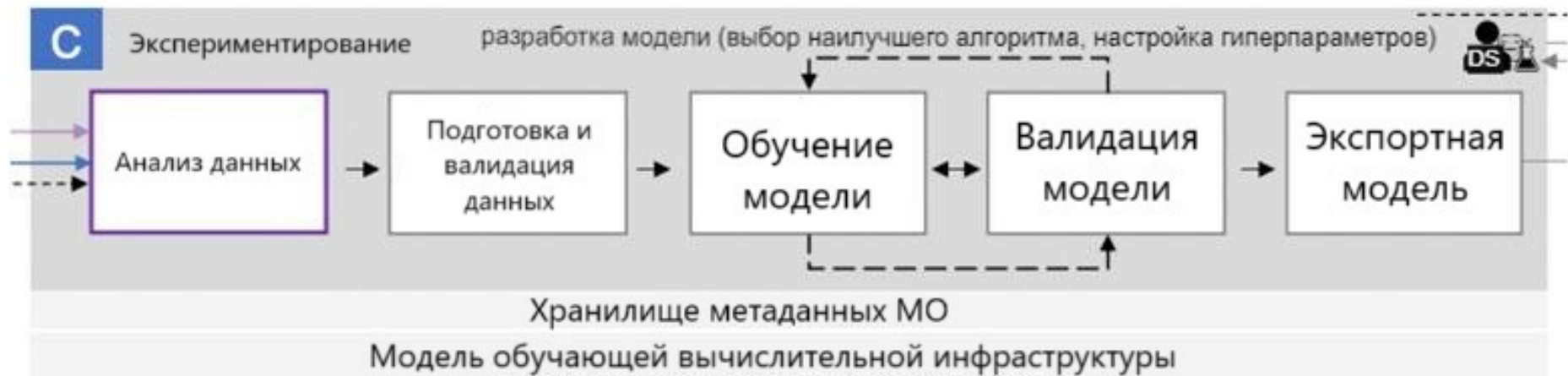
7. Совместная работа и командная разработка: MLOps подчеркивает важность совместной работы разработчиков, инженеров данных, аналитиков и специалистов по машинному обучению для эффективной разработки и управления моделями.

8. Возможность использовать собранные контейнеры во внешних приложениях: используя принципы MLOps, можно обеспечить согласованность, надежность и масштабируемость Streamlit-приложения, которое позволяет быстро создавать интерактивные веб-приложения для демонстрации и визуализации моделей машинного обучения.

Отличие от DevOps

- У DevOps и MLOps есть фундаментальное сходство, потому что принципы MLOps были выведены из принципов DevOps. Но в реализации они сильно отличаются:
- Гибридный состав команды
- Тестирование
- Автоматизированное развёртывание
- Снижение производительности системы в процессе производства из-за изменения профилей данных или просто из-за перекоса в обучении и обслуживании

Типичная модель машинного обучения



Представим себе реальное производство с соответствующими этапами обучения модели

Подключение к источникам данных и их извлечение

Очистка и преобразование данных

Вычисление новых фичей

Добавление результата

Полученные фичи важно использовать максимально эффективно: сохранять и переиспользовать в задачах других ML-разработчиков компании.

Весьма логично, что мало кто хочет все это переделывать каждый раз при каком-то обновлении.

И поэтому в MLOps появляется несколько заметных отличий:

- Непрерывная интеграция (CI) — это уже не только тестирование и проверка кода и компонентов, но и тестирование и проверка данных, схем данных и моделей.
- Непрерывное развёртывание (CD) — это уже не отдельный программный пакет или сервис, а система (конвейер машинного обучения), которая должна автоматически развёртывать другой сервис (сервис прогнозирования модели) или отменять изменения в модели.
- Непрерывное тестирование (CT) — это новое свойство, уникальное для систем машинного обучения, которое позволяет автоматически переобучать модели и использовать их.

Лучшие практики MLOps

- **1. Команда**
- **2. Данные**
- **3. Цель** (показатели и ключевые показатели эффективности)
- **4.1 Модель**
- **4.2 Обучение**
- **5. Код**
- **6. Развертывание**

1. Команда

- -Используйте платформу совместной разработки (GitHub)
- -Работа с общим бэклогом
- -Общайтесь, согласовывайте свои действия и сотрудничайте с другими членами команды (Scrum-методология)

2. Данные

- -Используйте проверки работоспособности для всех внешних источников данных
- -Отслеживайте, идентифицируйте и учитывайте изменения в источниках данных.
- -Напишите повторно используемые скрипты для очистки и объединения данных
- -Комбинируйте и изменяйте существующие функции, чтобы создавать новые функции понятным человеку способом
- -Убедитесь, что маркировка данных выполняется в рамках строго контролируемого процесса
- -Сделайте наборы данных доступными в общей инфраструктуре (частной или общедоступной)

3. Цель

- -Не думайте слишком много о том, какую цель вы выберете для непосредственной оптимизации, сначала отслеживайте показатели (метрики).
- -Выберите простой, наблюдаемый и соотносимый показатель для вашей первой цели
- -Обеспечьте конфиденциальность

4.1 Модель

- -Упростите первую модель и создайте правильную инфраструктуру
- -Запуск интерпретируемой модели (т.е. модели, которая позволяет понять и объяснить, как и почему она принимает определенные решения или предсказания) упрощает отладку.

4.2 Обучение

- - -Зафиксируйте цель обучения в метрике, которую легко измерить и понять
- - -Активно удаляйте или архивируйте неиспользуемые функции
- - - Изучайте экспертную оценку обучающих сценариев
- - -Включайте параллельные обучающие эксперименты
- - -Автоматизируйте оптимизацию гиперпараметров
- - -Постоянно измеряйте качества и производительности модели
- - -Используйте управление версиями для данных, моделей, конфигураций и обучающих сценариев

5. Код

- -Запускайте автоматические регрессионные тесты
- -Используйте статический анализ для проверки качества кода
- -Используйте непрерывную интеграцию

6. Развертывание

- -Планируйте запуск и итерации.
- -Автоматизируйте развертывание модели
- -Проводите постоянный мониторинг поведения развернутых моделей
- -Включите автоматический откат для производственных моделей
- -Включите теневое развертывание (тестовое развертывание параллельно рабочем)
- -Упрощайте ансамбли
- -Записывайте производственные прогнозы с указанием версии модели, версии кода и входных данных
- -Человеческий анализ системы и перекося в обучении
- -Измерьте разницу между моделями
- -При выборе моделей утилитарная производительность превосходит возможности прогнозирования.
- -Выполняйте проверки эволюционирующих профилей данных

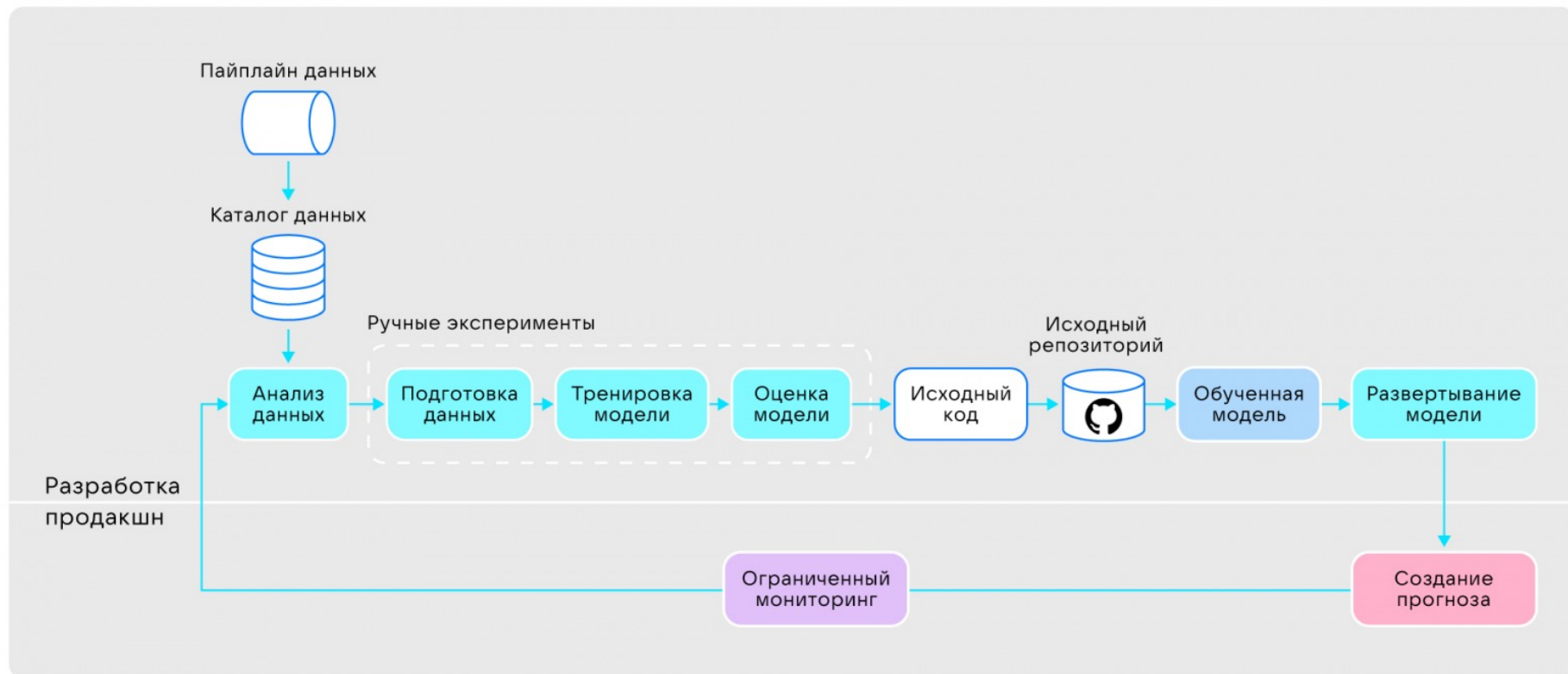
Как реализовать MLOps?

- По данным Google, существует три способа внедрения MLOps:
- Уровень MLOps 0 (ручной процесс)
- Уровень MLOps 1 (автоматизация конвейера ML)
- Уровень MLOps 2 (автоматизация конвейера CI / CD)

Уровень MLOps 0

- Типичен для компаний, которые только начинают использовать машинное обучение. Если ваши модели редко меняются или обучаются, то полностью ручного процесса машинного обучения и работы специалистов по обработке данных может быть достаточно.
- **Характеристики**
- Ручной, управляемый скриптом и интерактивный процесс: каждый шаг выполняется вручную, включая анализ данных, подготовку данных, обучение модели и проверку. Он требует ручного выполнения каждого шага и перехода от одного шага к другому.
- Разделение между машинным обучением и операциями: процесс отделяет специалистов по обработке данных, которые создают модель, от инженеров, которые используют модель в качестве сервиса прогнозирования. Специалисты по обработке данных передают обученную модель в качестве артефакта для развертывания в инфраструктуре API.
- Редкие итерации выпуска: предполагается, что ваша команда специалистов по обработке данных управляет несколькими моделями, которые нечасто меняются — либо меняется реализация модели, либо модель переобучается на новых данных. Новая версия модели выпускается всего пару раз в год.
- Отсутствие непрерывной интеграции (CI), непрерывного развёртывания (CD) и уж тем более непрерывного тестирования.
- Развертывание относится к сервису прогнозирования (т. е. микросервису с REST API)
- Отсутствие активного мониторинга производительности: процесс не отслеживает и не регистрирует прогнозы и действия модели.
- У команды разработчиков может быть собственная сложная система для настройки, тестирования и развёртывания API, включая безопасность, регрессионное и нагрузочное тестирование.
- **Проблемы**
- На практике модели часто ломаются при внедрении в реальный мир. Модели не могут адаптироваться к изменениям в динамике окружающей среды или к изменениям в данных, описывающих окружающую среду.

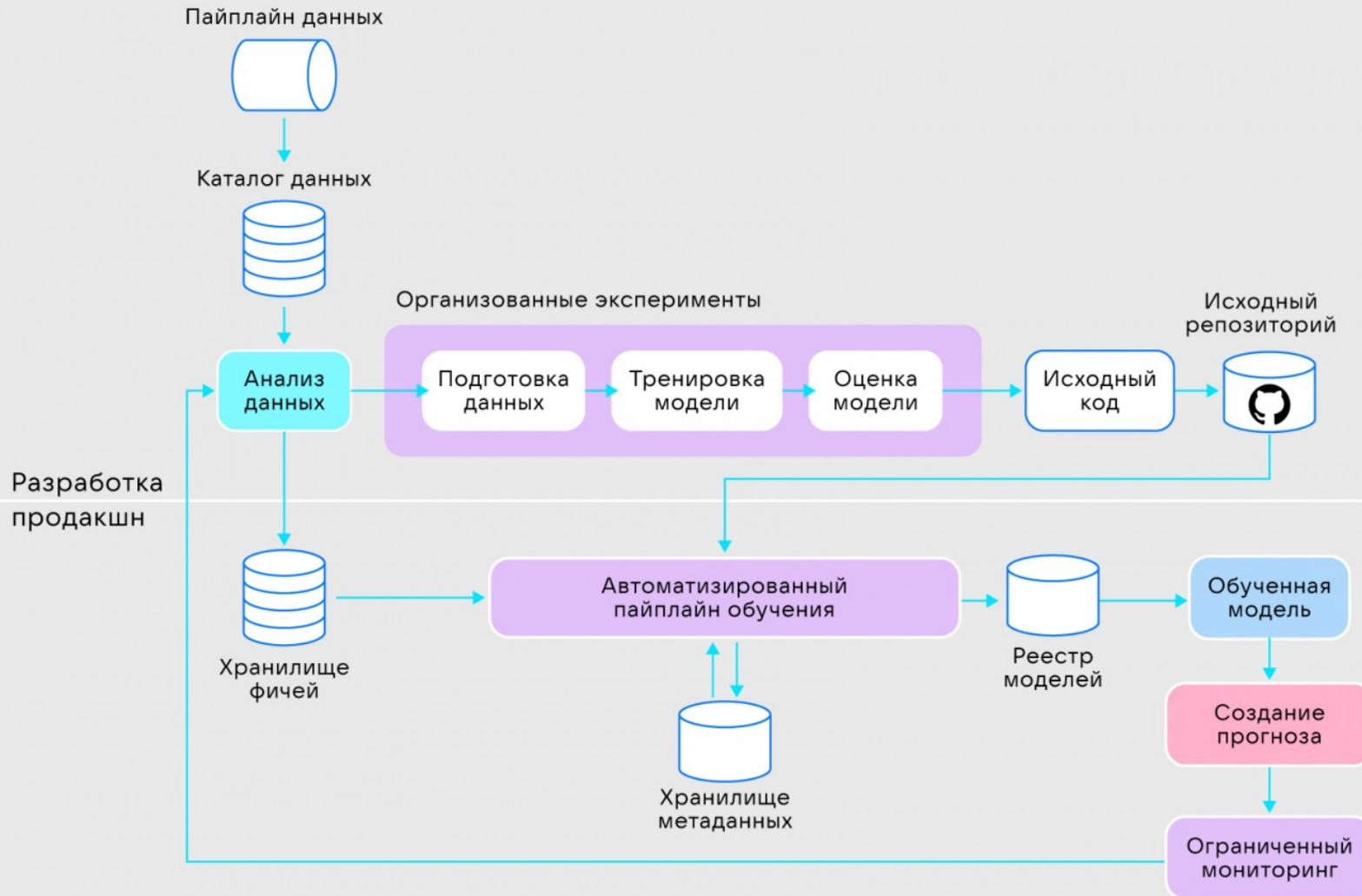
Уровень MLOps 0



Уровень MLOps 1

- **Цель:** Основная задача MLOps 1-го уровня — автоматизация конвейера машинного обучения для непрерывного обучения (СТ) модели. Это обеспечивает постоянную поставку услуг по прогнозированию.
- **Характеристики:**
 - - Быстрый эксперимент: Автоматизированная организация и выполнение этапов эксперимента с машинным обучением.
 - - Автоматическое обучение: Модель обучается в рабочей среде на актуальных данных с использованием триггеров в реальном времени.
 - - Экспериментально-операционная симметрия: Одинаковый конвейер используется в разработке и производственной среде, объединяя подходы DevOps.
 - - Модульный код: Компоненты конвейеров многократно используемы и могут быть доступны для различных пайплайнов.
 - - Непрерывная поставка: Автоматизированный процесс развёртывания модели как сервиса онлайн-прогнозирования.
-
- **Дополнительные компоненты:**
 - - Проверка данных и модели: Автоматизированные этапы проверки данных и модели в производственном конвейере.
 - - Хранилище функций: Централизованное хранилище для стандартизации функций.
 - - Управление метаданными: Запись информации о выполнении конвейера для обеспечения воспроизводимости и устранения ошибок.
 - - Триггеры конвейера: Автоматизация переобучения моделей на новых данных по различным триггерам (по запросу, по графику и др.).
- **Проблемы:** Этот подход подходит для развёртывания новых моделей на основе данных, но требует системы CI/CD для быстрого внедрения новых идей и компонентов машинного обучения.

Уровень MLOps 1



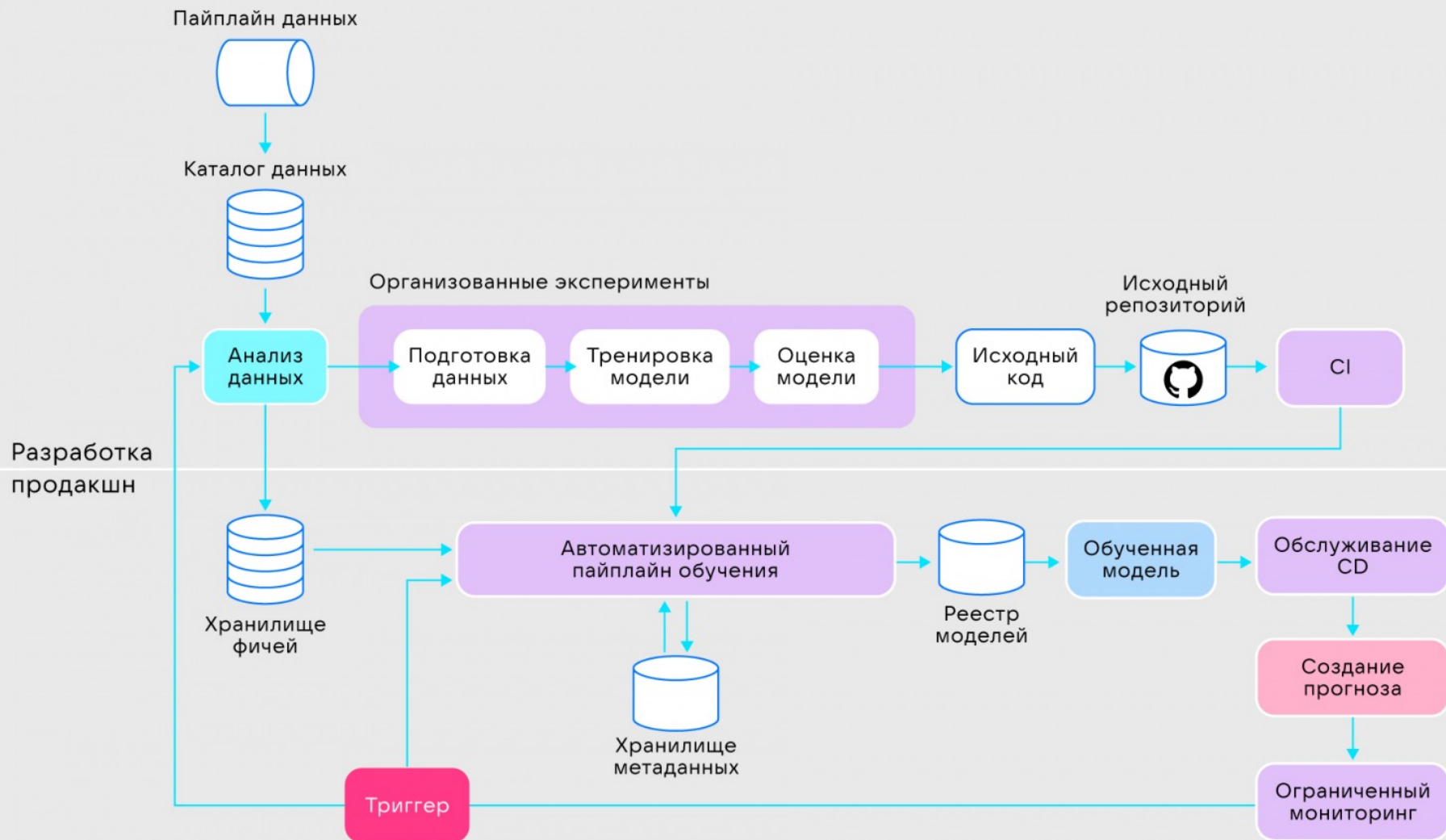
Уровень MLOps 2

Этот уровень подходит технологичным компаниям, которым приходится ежедневно, если не ежечасно, переобучать свои модели, обновлять их за считанные минуты и одновременно развертывать на тысячах серверов. Без сквозного цикла MLOps такие организации просто не выживут.

Эта настройка MLOps включает в себя все перечисленные ранее компоненты.

Проблема: затраты

Уровень MLOps 2



Инструменты MLFlow

continuous_deployment_pipeline >

continuous_deployment_pipeline-2024_05_26-22_59_59_285673

Register model

Overview Model metrics System metrics **Artifacts**

▼ model
 ► metadata
 MLmodel
 conda.yaml
 model.pkl
 python_env.yaml
 requirements.txt
 estimator.html

model/MLmodel 1.45KB

Path: file:///Users/iana/Library/Application Support/zenml/local_stores/0c496041-e535-44a0-8d4c-b340cde8590e/mlruns/614075710536826426/34311e8ea838434d

artifact_path: model

flavors:

python_function:

env:

conda: conda.yaml

virtualenv: python_env.yaml

loader_module: mlflow.sklearn

model_path: model.pkl

predict_fn: predict

python_version: 3.9.6

sklearn:

code: null

pickled_model: model.pkl

serialization_format: cloudpickle

sklearn_version: 1.5.0



mlflow_version: 2.12.1

model_size_bytes: 926

model_uuid: b151482102604f7d86451875cfd39173

url: 34311e8ea838434d-00000000000000000000000000000000

mlflow

DOCS COMMUNITY CODE  



An open source platform for the
machine learning lifecycle

Latest News

MLFlow 1.27.0 released,
including new MLflow
Pipelines component! (28
Jun 2022)

MLFlow 1.26.1 released!
(27 May 2022)

MLFlow 1.26.0 released!
(16 May 2022)

MLFlow 1.24.0 released!
(28 Feb 2022)

[News Archive](#)



WORKS WITH ANY ML
LIBRARY, LANGUAGE &
EXISTING CODE



RUNS THE SAME WAY IN ANY
CLOUD



DESIGNED TO SCALE FROM 1
USER TO LARGE ORGS



SCALES TO BIG DATA WITH
APACHE SPARK™

Инструменты

MLFlow

- Жизненный цикл машинного обучения контролируется платформой с открытым исходным кодом MLflow и включает центральную регистрацию модели, ее развертывание и экспериментирование.
- MLflow может использоваться командой любого размера как индивидуально, так и коллективно. Библиотеки не имеют отношения к инструменту.
- Его может использовать любой язык программирования и библиотека машинного обучения.
- Чтобы упростить обучение, развертывание и управление приложениями машинного обучения, MLflow взаимодействует с рядом платформ машинного обучения, включая TensorFlow и PyTorch.
- Кроме того, MLflow предоставляет простые в использовании API-интерфейсы, которые можно включить в любые существующие программы или библиотеки машинного обучения.
- MLflow имеет четыре ключевые функции, облегчающие отслеживание и планирование экспериментов:
- MLflow Tracking — API и пользовательский интерфейс для регистрации параметров, версий, метрик и артефактов кода машинного обучения, а также для последующего отображения и сопоставления результатов.
- Проекты MLflow — упаковка кода машинного обучения в повторно используемый воспроизводимый формат для передачи в производство или обмена с другими специалистами по данным.
- Модели MLflow — поддержка и развертывание моделей в ряде систем обслуживания моделей и систем логического вывода из различных библиотек машинного обучения.
- Реестр моделей MLflow — центральное хранилище моделей, которое обеспечивает совместное управление всем жизненным циклом модели MLflow, включая управление версиями модели, переходы между стадиями и аннотации.

Инструменты. Kubeflow

Kubeflow

The Machine Learning Toolkit for Kubernetes

[Get Started](#) [Contribute](#)

Kubeflow

- Home
- Notebooks
- Tensorboards
- Models
- Snapshots
- Volumes
- Experiments (AutoML)
- Experiments (KFP)
- Pipelines
- Runs
- Recurring Runs
- Artifacts
- Executions

kubeflow-user (Owner)

Model server details

flowers

OVERVIEW DETAILS METRICS LOGS YAML

Predictor: flowers-predictor-default-hb8r5

Total CPU Usage

Total Memory Usage

Request Volume by Revision

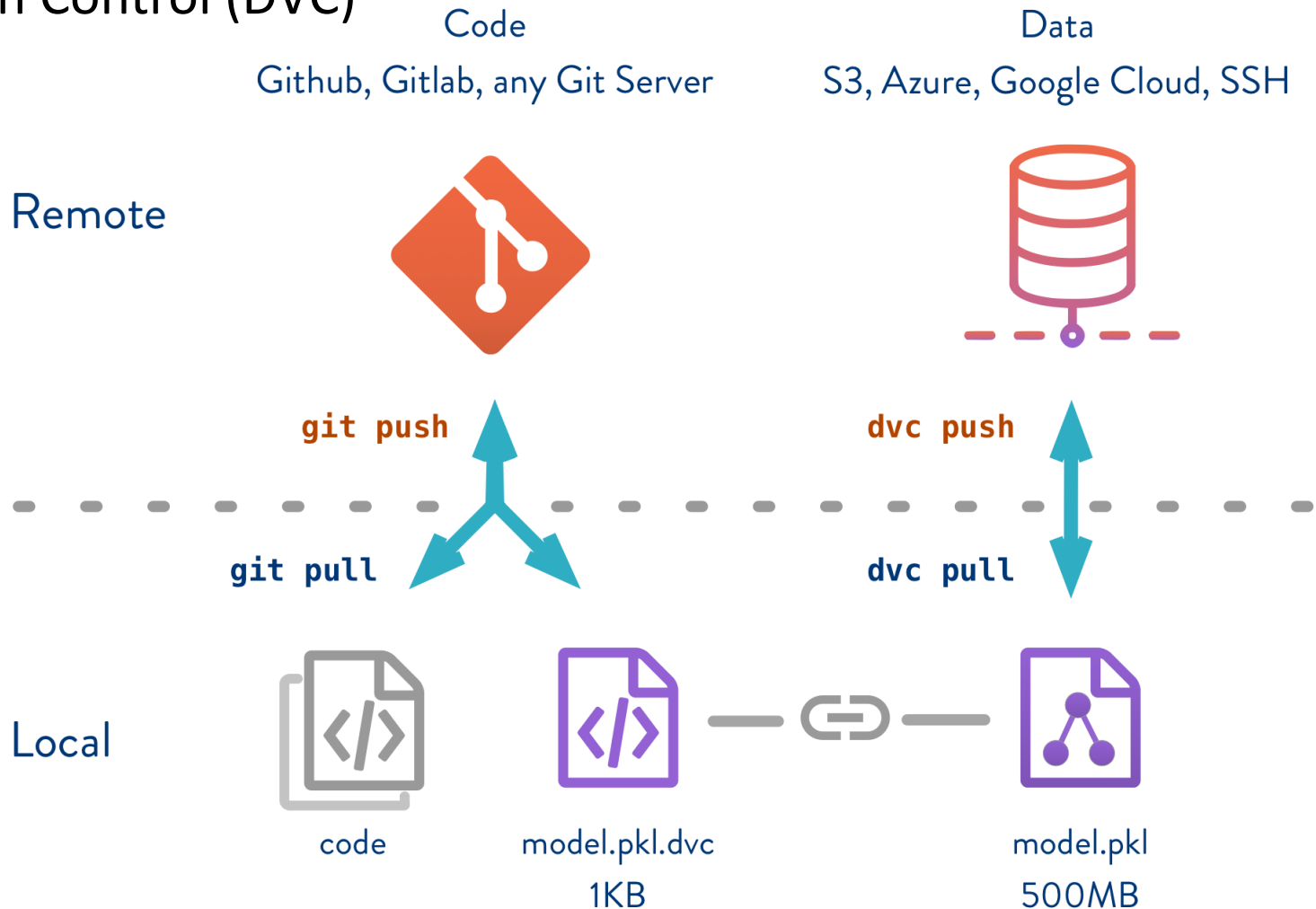
Response Time by Revision

Инструменты. Kubeflow

- Набор инструментов машинного обучения для Kubernetes называется Kubeflow. Упаковка контейнеров Docker и управление ими, помощь в обслуживании системы машинного обучения.
- Упрощая оркестровку запуска и развертывание рабочих процессов машинного обучения, он способствует масштабируемости моделей машинного обучения.
- Это проект с открытым исходным кодом, который включает в себя тщательно подобранную группу дополнительных инструментов и сред, адаптированных к различным потребностям ML.
- С помощью Kubeflow Pipelines можно справиться с длительными задачами обучения машинному обучению, ручным экспериментированием, воспроизводимостью и задачами DevOps.
-
- Для нескольких этапов машинного обучения, включая обучение, разработку конвейера и обслуживание Ноутбуки Jupyter, Kubeflow предлагает специализированные услуги и интеграцию.
-
- Это упрощает управление и отслеживание жизненного цикла ваших рабочих нагрузок ИИ, а также развертывание моделей машинного обучения (ML) и конвейеров данных в кластерах Kubernetes.
- Он предлагает:
- Ноутбуки для использования SDK для взаимодействия с системой
- пользовательский интерфейс (UI) для контроля и мониторинга запусков, заданий и экспериментов.
- Чтобы быстро разрабатывать комплексные решения без необходимости каждый раз перестраивать и повторно использовать компоненты и пайплайны.
- Kubeflow Pipelines предлагается как ключевой компонент Kubeflow или как отдельная установка.

Инструменты. Контроль Версий Данных. Git

Data Version Control (DVC)



Инструменты. Контроль Версий Данных. Git



















































































































- Решение для управления версиями с открытым исходным кодом для проектов машинного обучения называется DVC или Data Version Control.
- Какой бы язык вы ни выбрали, это экспериментальный инструмент, помогающий в определении конвейера.
- DVC использует код, управление версиями данных и воспроизводимость, чтобы помочь вам сэкономить время, когда вы обнаружите проблему с более ранней версией вашей модели машинного обучения.
- Кроме того, вы можете использовать конвейеры DVC для обучения вашей модели и распространения ее среди членов вашей команды. DVC может управлять организацией больших данных и управлением версиями, а данные можно хранить в легкодоступном виде.
- Хотя он включает в себя некоторые (ограниченные) функции отслеживания экспериментов, в основном он фокусируется на данных и управлении версиями конвейера.
- Он предлагает:
- Не зависит от хранилища, поэтому можно использовать различные типы хранилищ.
- Он также обеспечивает отслеживание статистики.
- готовые средства объединения этапов машинного обучения в DAG и запуска всего конвейера от начала до конца
- Всю разработку каждой модели машинного обучения можно проследить, используя весь ее код и происхождение данных.
- Воспроизводимость за счет точного сохранения исходной конфигурации, входных данных и программного кода для эксперимента.

Инструменты. Apache Airflow



DAGs

Search:

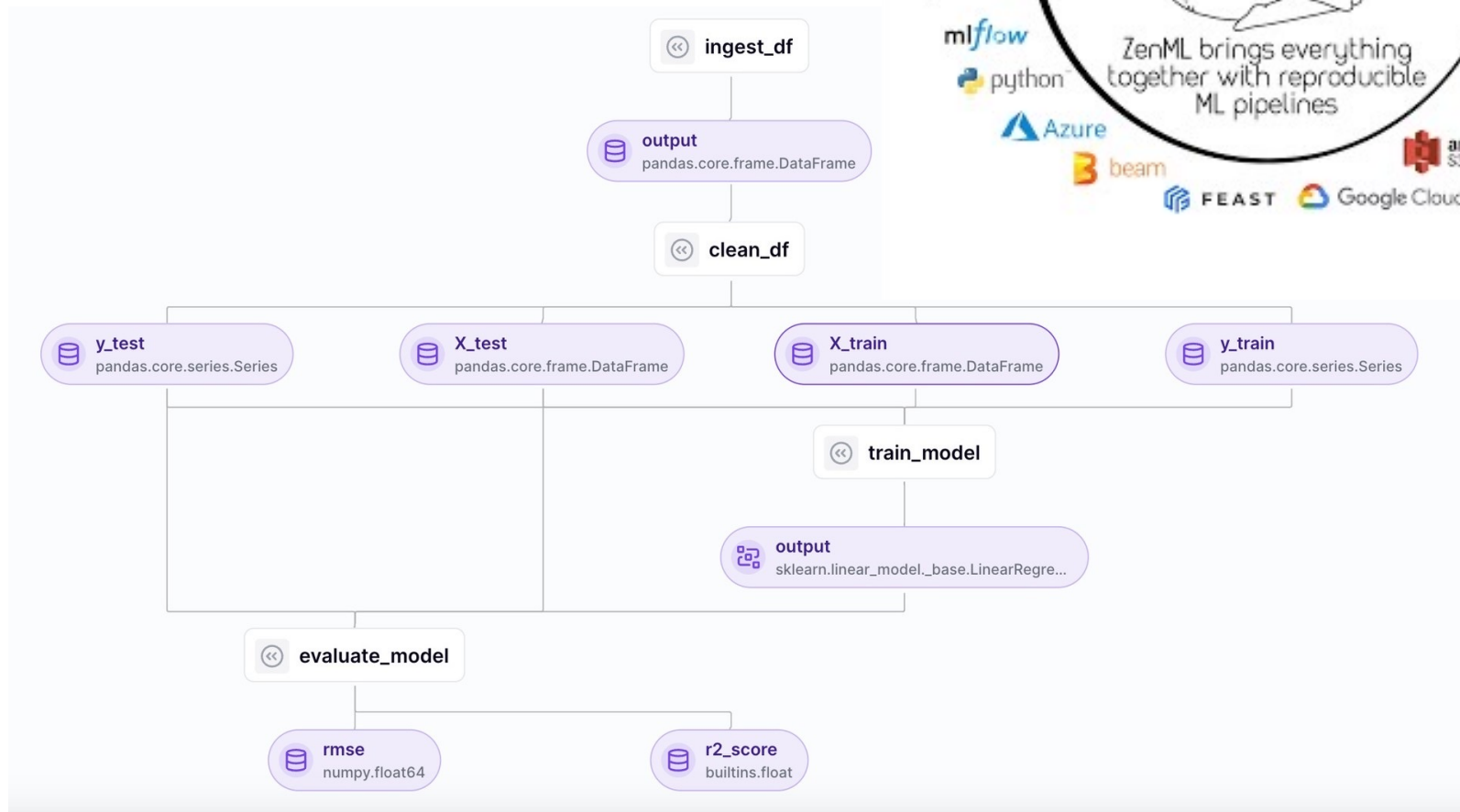
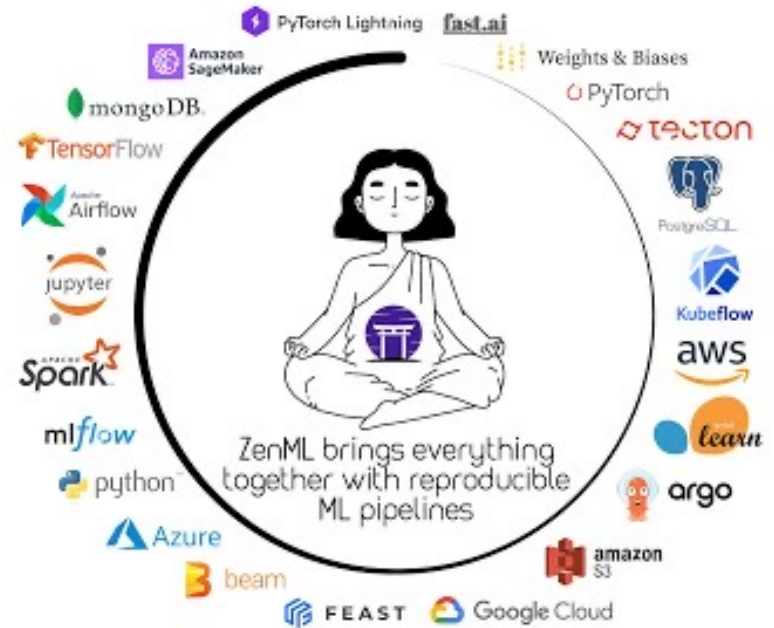
		DAG	Schedule	Owner	Recent Tasks 	Last Run 	DAG Runs 	Links
		example_bash_operator	0 0 ***	airflow	 6       	2018-09-06 00:00 	 5  	       
		example_branch_dop_operator_v3	* / 1 * * * *	airflow	 3  1     1  5 	2018-09-05 00:56 	 54  3 	       
		example_branch_operator	@daily	airflow	 5       	2018-09-06 00:00 	 2  	       
		example_xcom	@once	airflow	 3       	2018-09-05 00:00 	 1  	       
		latest_only	4:00:00	Airflow	 2       	2018-09-07 16:00 	 35  	       

Showing 1 to 5 of 5 entries

Инструменты. Apache Airflow

- Платформа с открытым исходным кодом для программирования, планирования и мониторинга рабочих процессов.
- Характеристики
- Динамичность: позволяет создавать динамические конвейеры.
- Масштабируемость: масштабируется для поддержки сложных рабочих процессов.
- Расширяемый: легко интегрируется с другими системами.

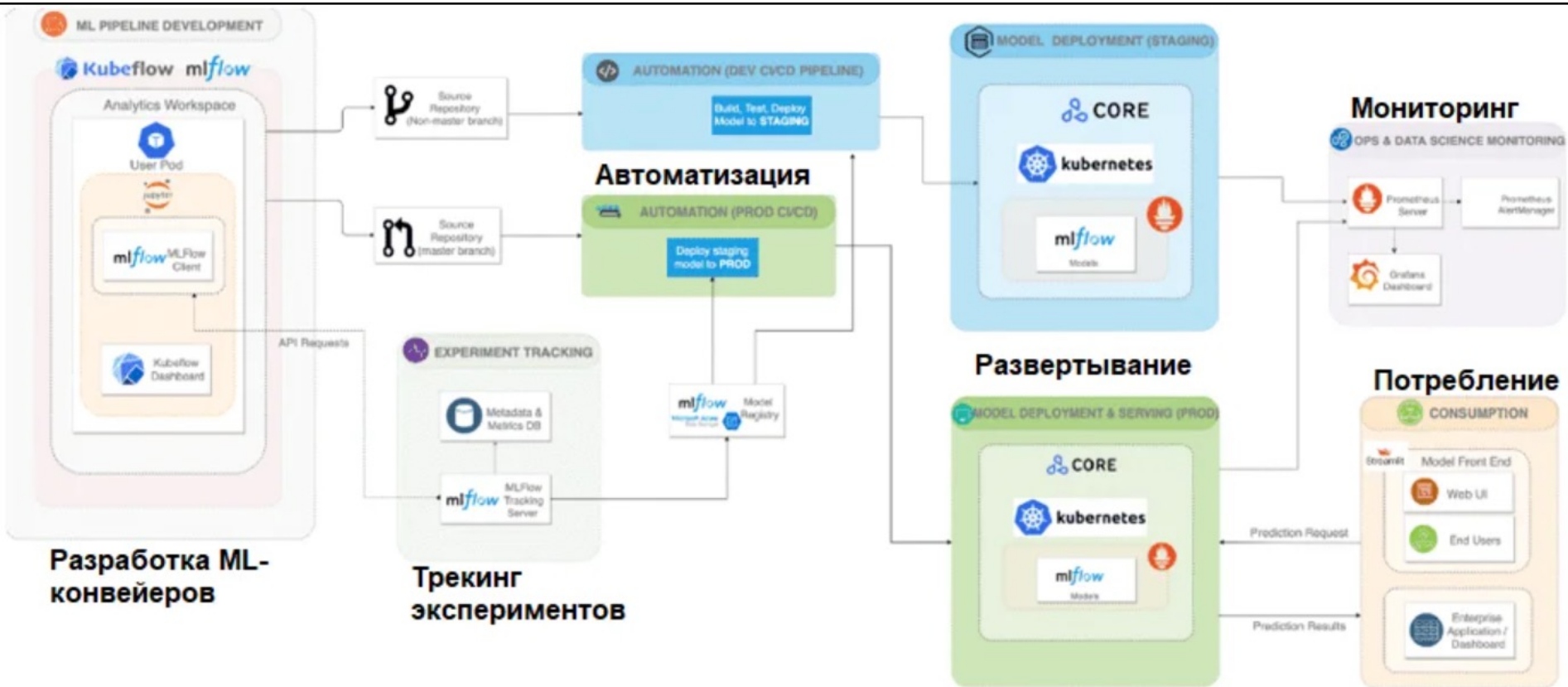
ZenML



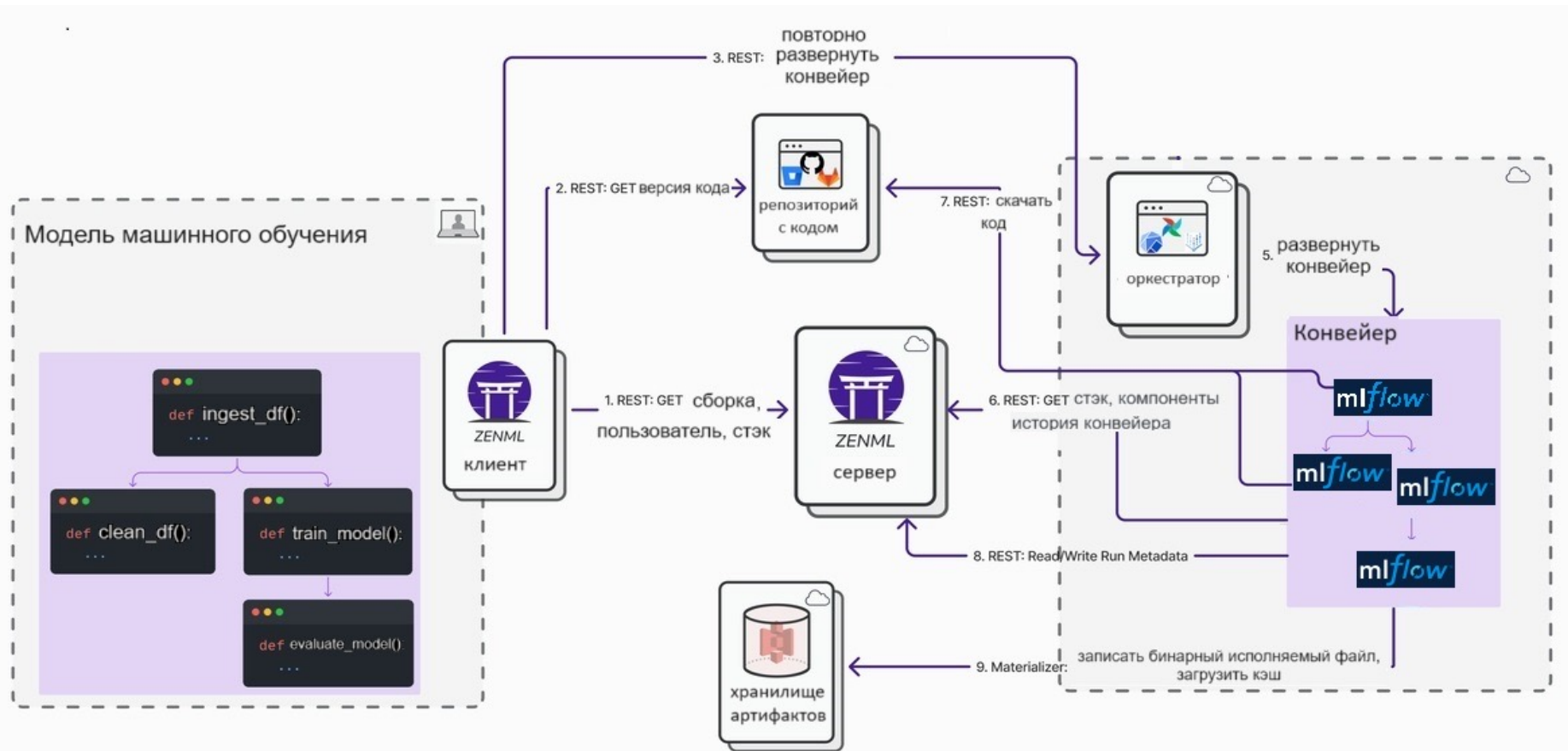
ZenML

- ZenML предоставляет платформу для управления полным жизненным циклом модели — от разработки до развертывания и мониторинга, легко интегрируется с популярными инструментами для машинного обучения, такими как MLflow, Kubernetes, Docker. ZenML обеспечивает воспроизводимость экспериментов и развертываний за счет использования конвейеров.
- ZenML предоставляет более унифицированную платформу, охватывающую весь жизненный цикл машинного обучения, в то время как MLflow больше фокусируется на отслеживании экспериментов и развертывании моделей
- В то же время он позволяет интегрировать некоторые функции MLFlow.
-
- Получается, есть целый стек ML-компонентов, которые нужно уметь настраивать, поддерживать и эксплуатировать. Часть про «настраивать и поддерживать» требует использования вспомогательного ПО.
- Например:
- Kubernetes — в качестве среды оркестрации контейнеров.
- KeyCloak — в качестве единой точки авторизации пользователей.
- Grafana, Prometheus, Loki — в качестве средств мониторинга и сбора логов и пр.

Архитектура MLOps-платформы на открытых инструментах



Архитектура из лабораторной работы



Что же насчет
специалистов?



MLOps-инженеры пока еще могут считаться единорогами.

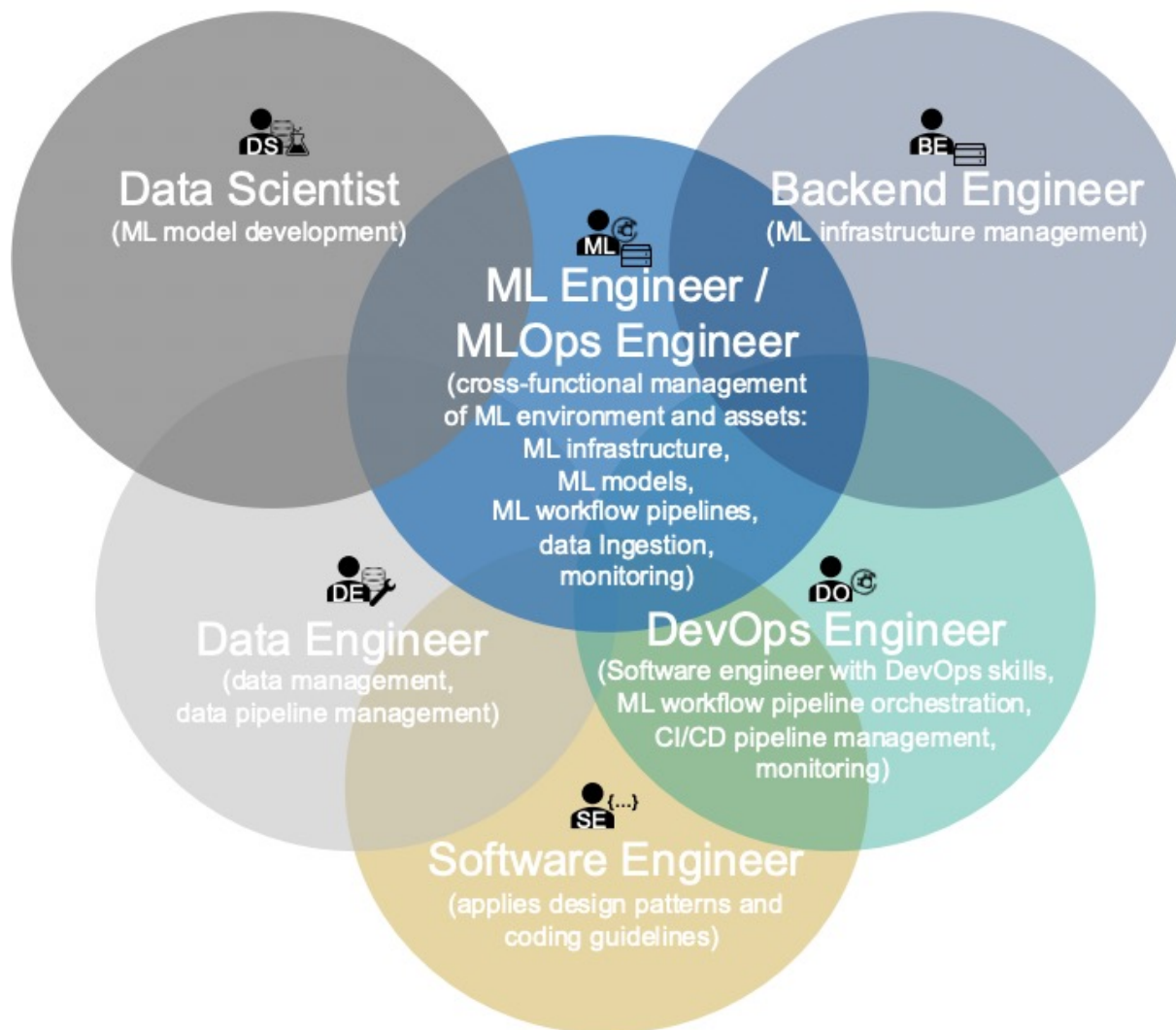
На сегодняшний день чаще всего задачи внедрения ML-систем отдают DevOps-специалистам. Если они отвечают за CI/CD, пусть еще и ML-пайплайны себе заберут. Естественно, у подхода есть минусы:

- деплой ML-моделей отличается от деплоя кода,
- нужно разбираться в Kubernetes на высоком уровне, а такие специалисты и без ML на вес золота,
- нужно использовать Iaas и дружить с Terraform,
- вишенкой на торте является процесс Continuous Training, который подразумевает написание Python-скриптов для автоматизации взаимодействия разных компонентов в оркестраторах.

Как минимум, нашему DevOps-специалисту потребуется много времени на изучение всех процессов и технологий. В одной версии идеального мира MLOps-специалист — следующая стадия развития DevOps-инженера. Когда уже все Terraform-файлы написал, сконфигурировал с помощью Ansible и в Kubernetes через Argo CD запустил.

Полезно, когда MLOps-специалист понимает мир ML-разработки, знает сложности и может аргументированно корректировать пайплайны. Таким образом, в другой версии идеального мира MLOps-инженер — ML-разработчик, который не только ML-модели готов обучать, но и с инфраструктурой разбираться.

На данный момент идеальный MLOps-инженер представляется как «воин дракона»: ученик, учитель, data scientist, backend developer, ML-инженер, data-инженер, devops, software developer и остальное в одном человеке.



Спасибо за ваше внимание!

