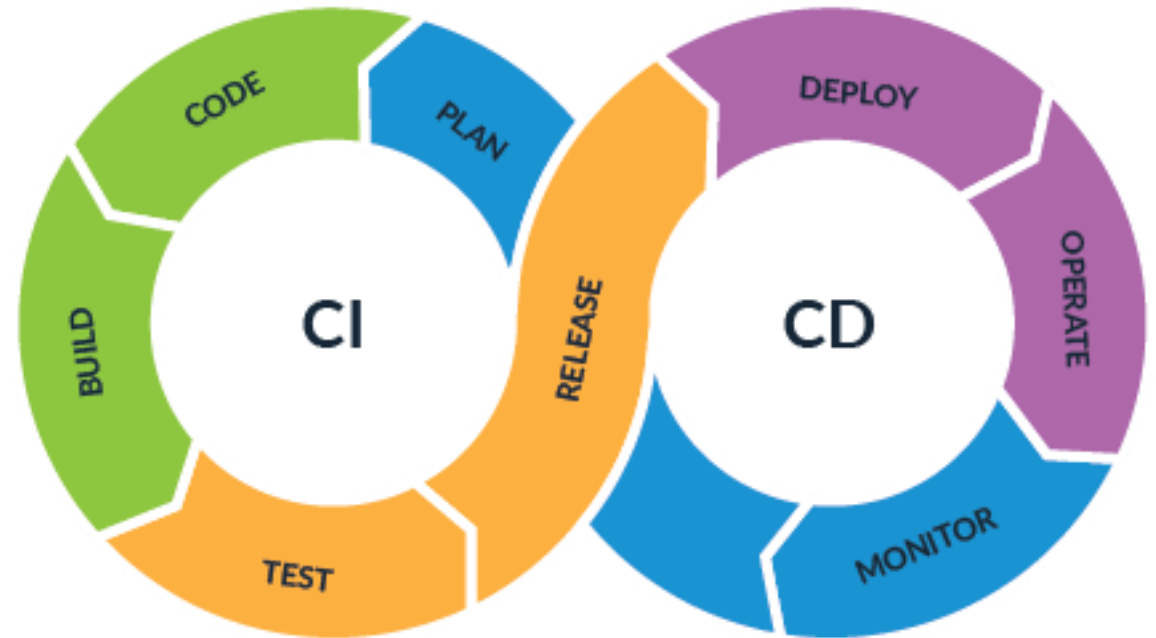


ML CI-CD

Семинар 8

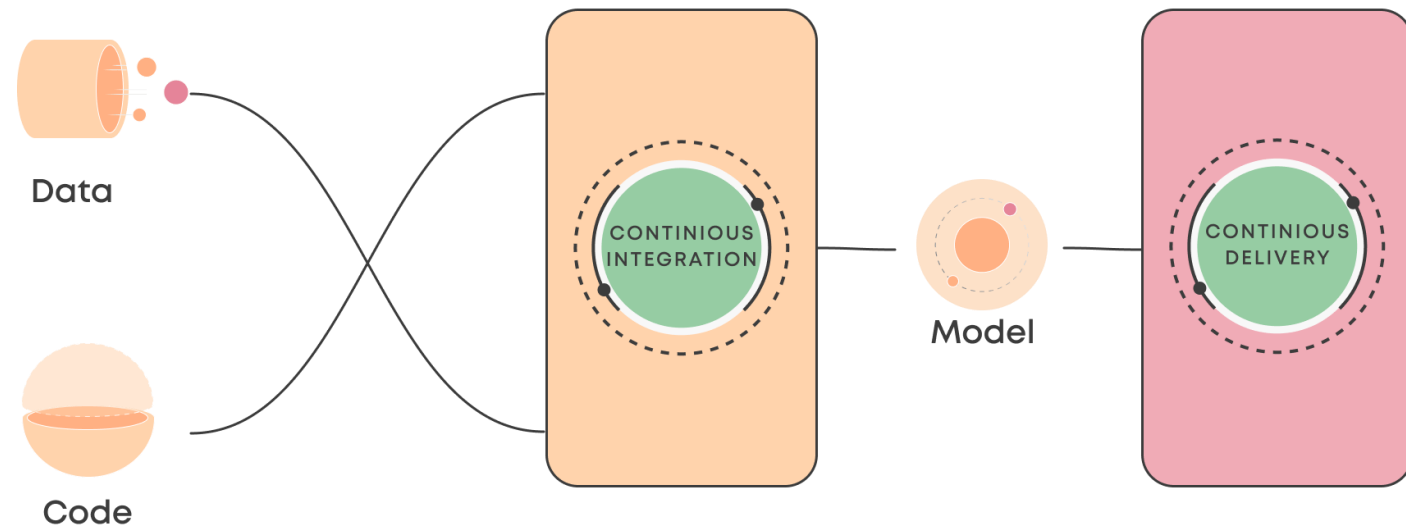
CI-CD

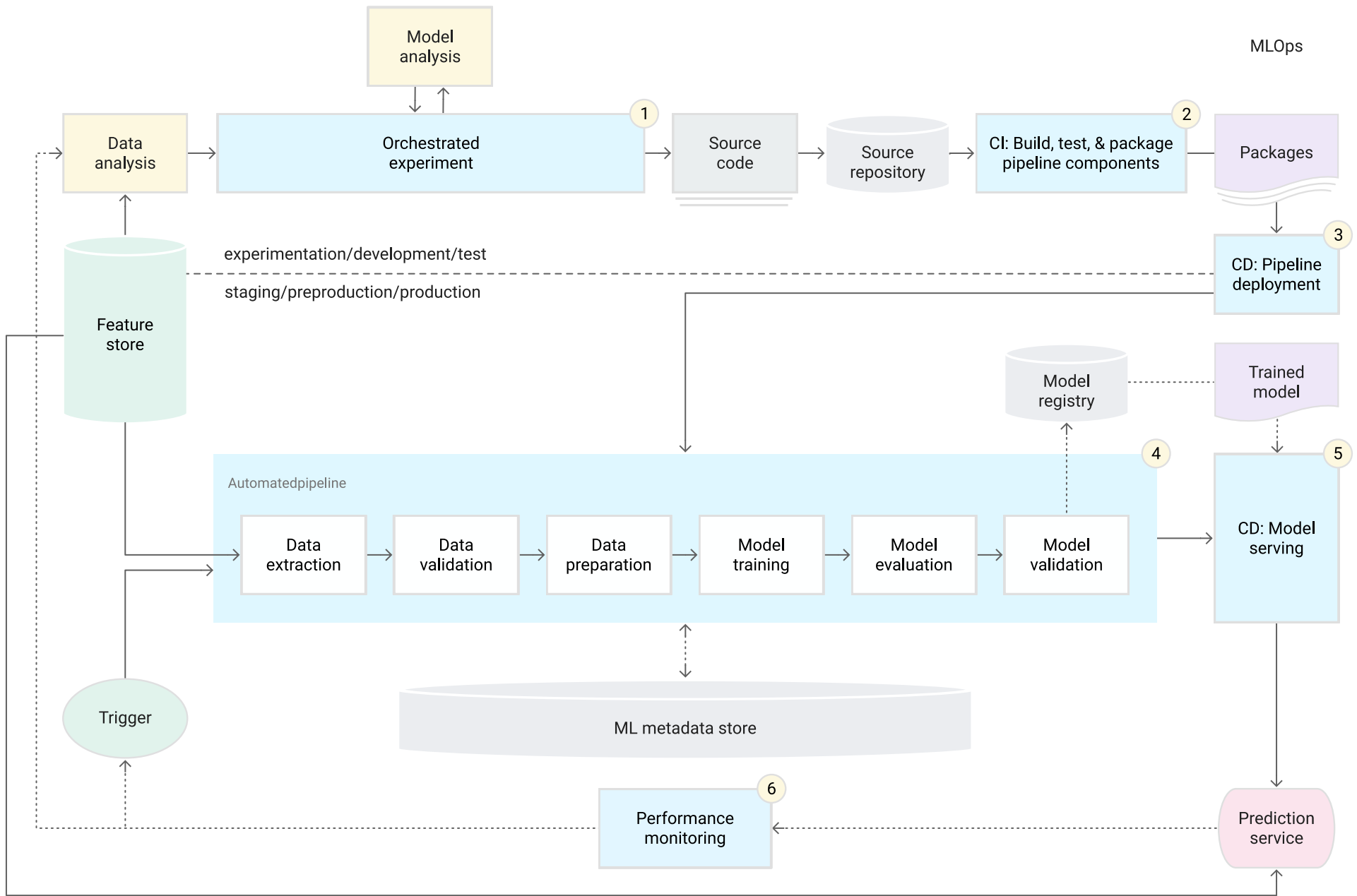
- CI (Continuous Integration):
Постоянная интеграция изменений кода.
- CD (Continuous Deployment/Delivery):
Автоматизация развертывания приложений.
- Преимущества CI/CD: Быстрая доставка, снижение ошибок, упрощение процессов.



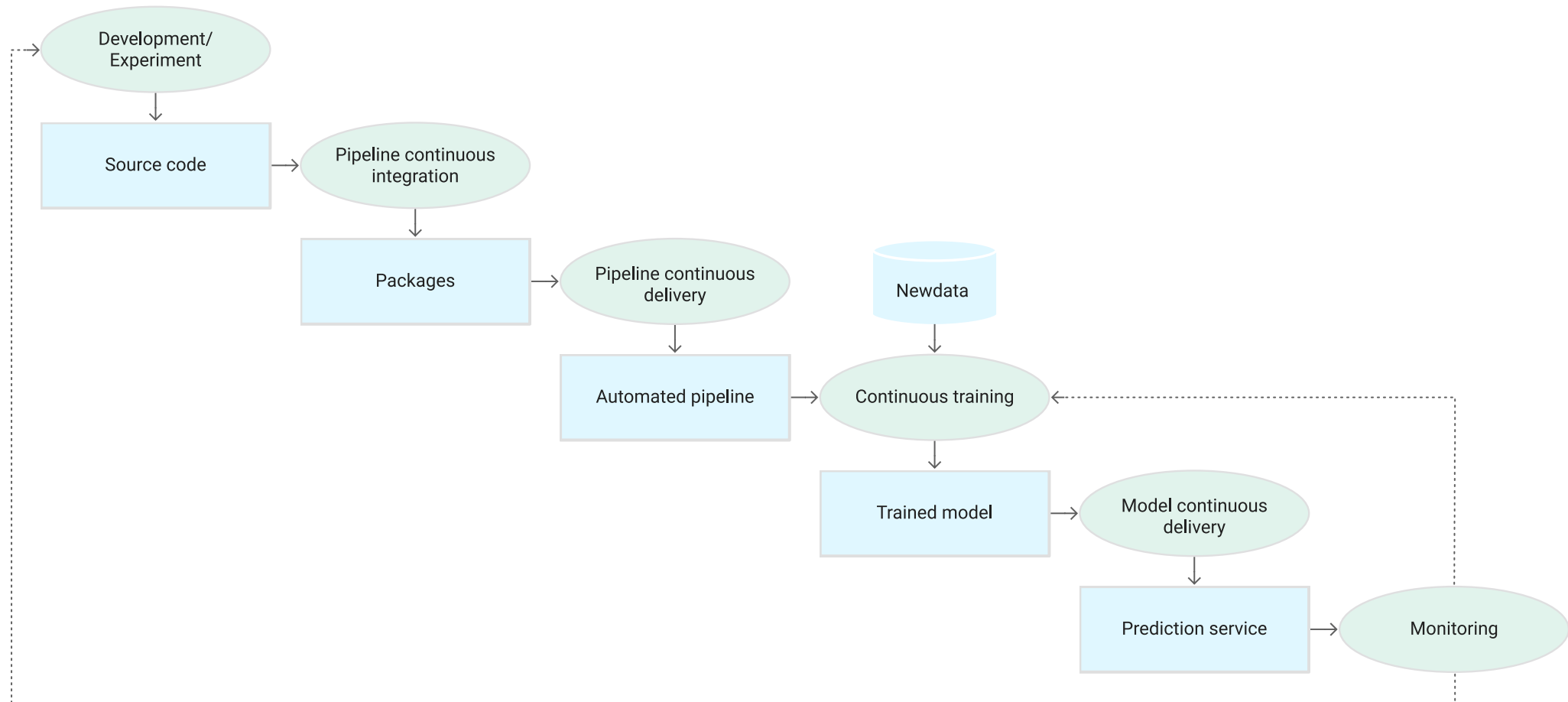
Применение CI-CD в МЛ

- Линтеринг и форматтинг
- Тестирование
- Continuous Training
- Генерация отчетов
- Поставка отчетов
- Поставка моделей
- Публикация библиотек





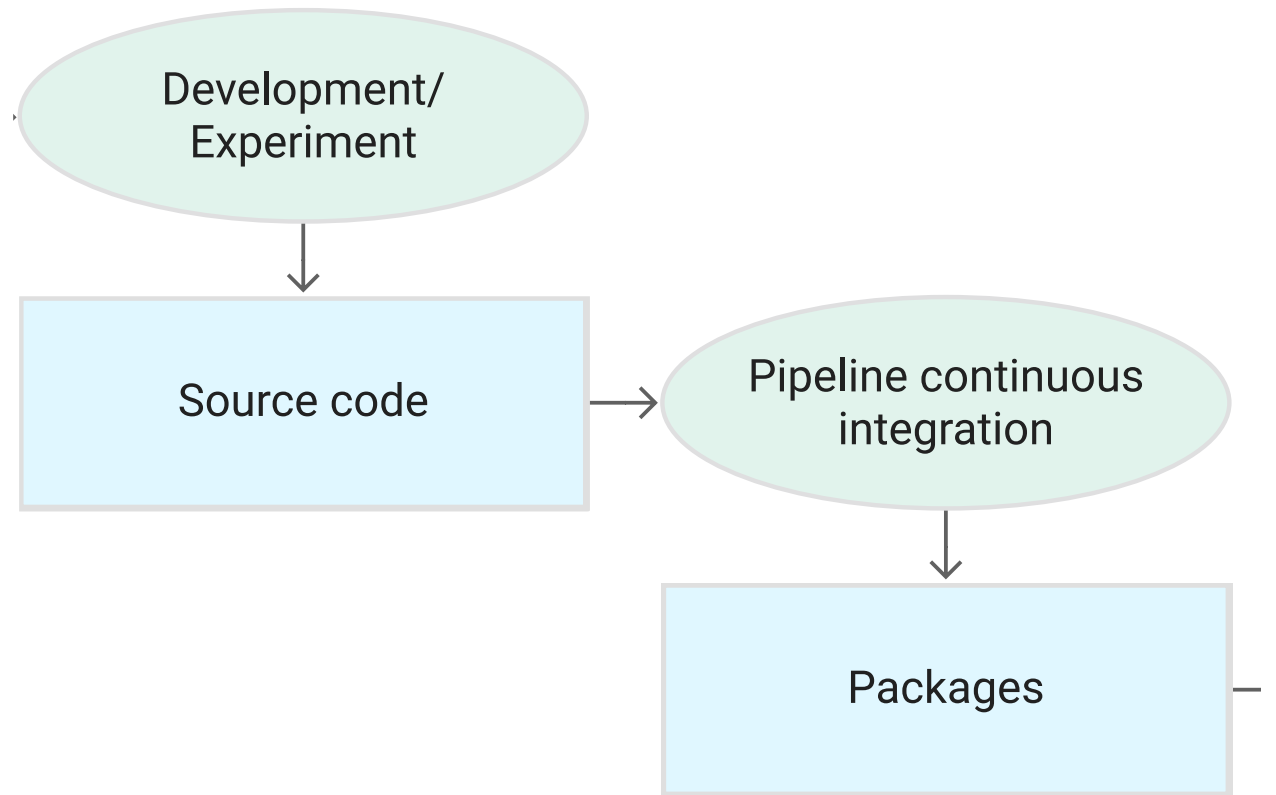
ML CI-CD



Задачи CI

Ответить на вопросы:

1. Собирается ли код?
2. Проходят ли тесты?
3. Достаточно ли качественный код?



Добавляем линтеры и тесты

Через GitHub Actions

Альтернативы:

- Jenkins
- OpenShift Pipelines
- GitLab CI/CD

Терминология

Workflow – набор инструкций

Job – набор из шагов (step) в рамках workflow, запускаемых на одном runner

Step – шаги, то, из чего состоит Job

Event – действие (создание PR, запушили коммит и пр.)

Action – переиспользуемый unit

Action

Переиспользуемый компонент

Можно писать свои action

```
steps:  
- uses: actions/checkout@v4  
- name: Set up Python ${{ matrix.python-version }}  
  uses: actions/setup-python@v3
```

Публикация пакета в PyPi

Для этого необходимо:

1. Зарегистрироваться в pypi
2. Получить API ключ
3. Добавить его в секреты

Публикация Docker Image в Docker Hub

```
name: ci

on:
  push:

jobs:
  docker:
    runs-on: ubuntu-latest
    steps:
      -
        name: Set up QEMU
        uses: docker/setup-qemu-action@v3
      -
        name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v3
      -
        name: Login to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${ secrets.DOCKERHUB_USERNAME }
          password: ${ secrets.DOCKERHUB_TOKEN }
      -
        name: Build and push
        uses: docker/build-push-action@v6
        with:
          push: true
          tags: user/app:latest
```

<https://github.com/marketplace/actions/build-and-push-docker-images>

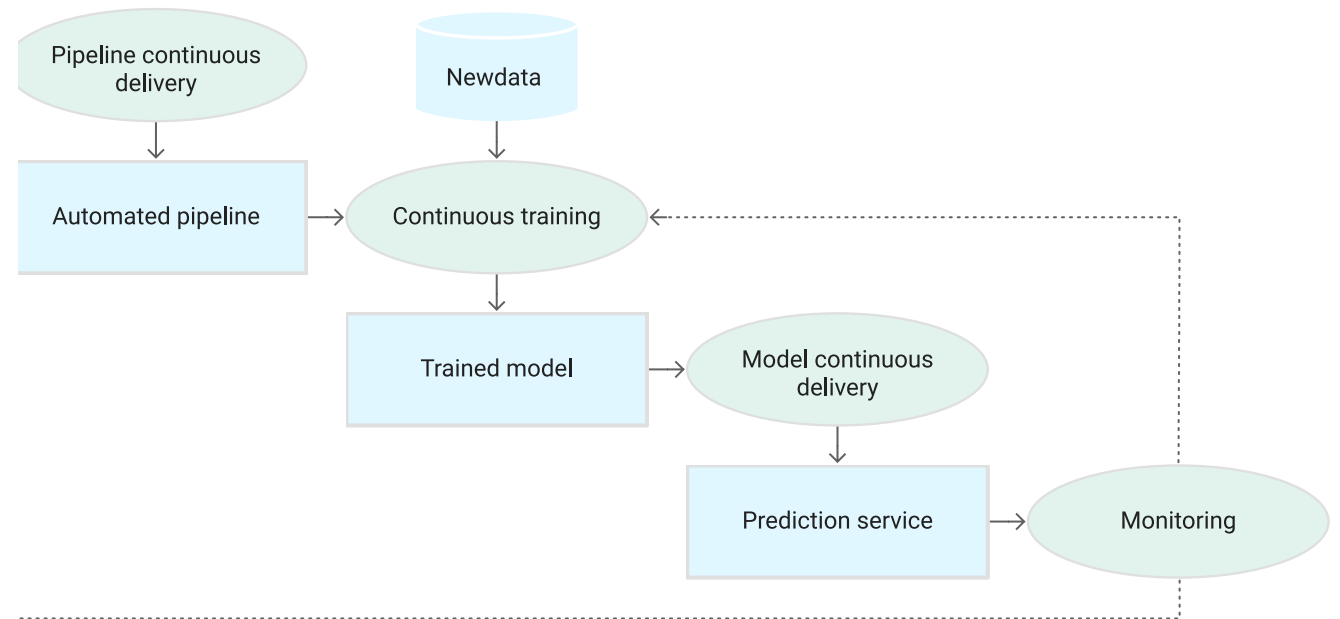
Задачи CD

Pipeline CD: вы развертываете артефакты, созданные на этапе CI, в целевой среде. Выход этого этапа — развернутый пайплайн с новой реализацией модели.





















Continuous training: пайплайн автоматически выполняется в проде на основе расписания или в ответ на триггер. Выход этого этапа — обученная модель, которая помещается в реестр моделей.

Model CD: обученная модель встает в сервис. Выход этого этапа — развернутый сервис прогнозирования модели.

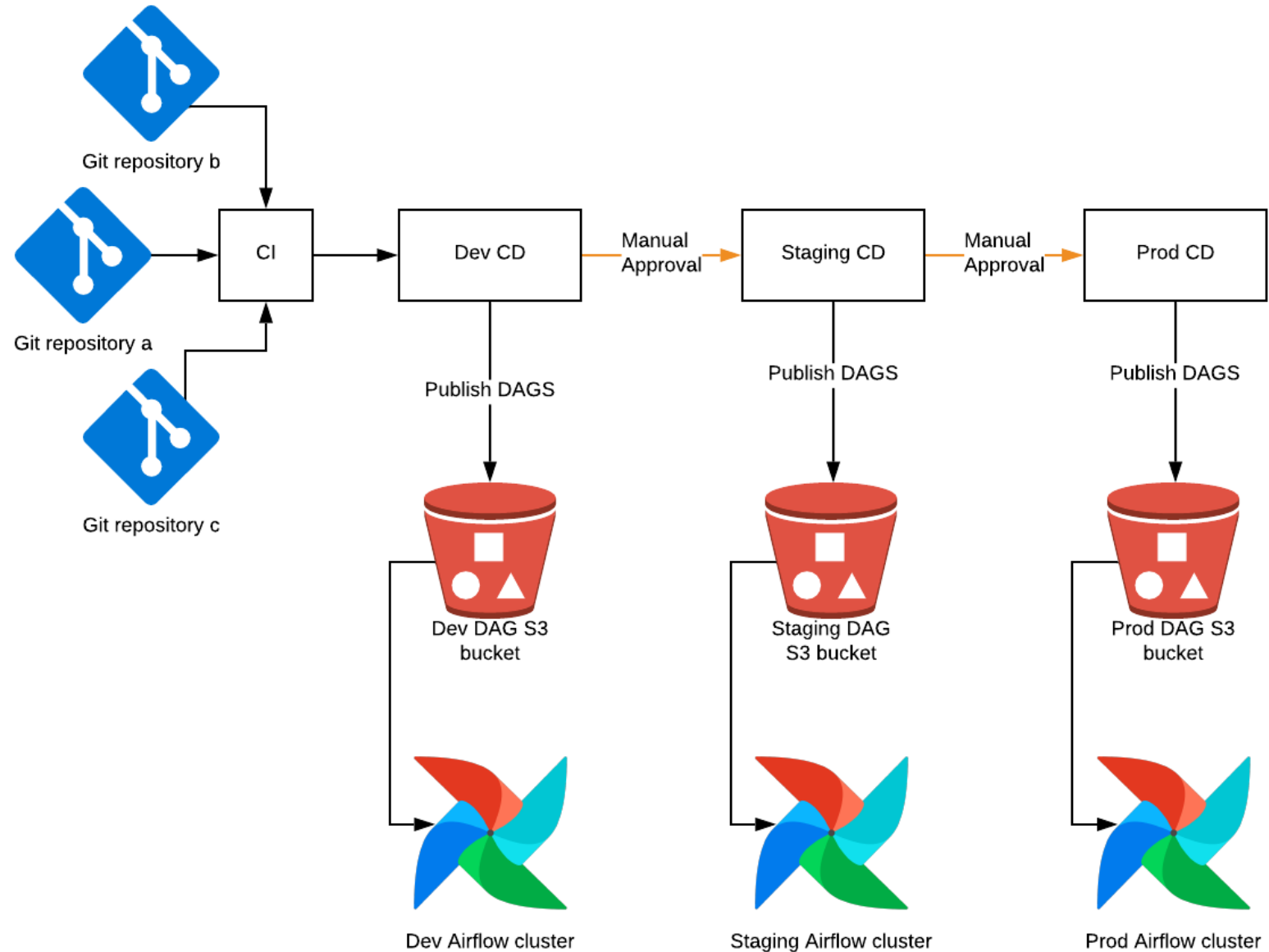
Мониторинг: сбор статистики по производительности модели на основе данных в реальном времени. Выход этого этапа — триггер для выполнения конвейера или выполнения нового цикла эксперимента.



Среды

Data	Used by	Deploys	Lifespan	Purpose		
 Seed data	 For Engineers	 Reloads constantly	 Recreated frequently	 Local testing during developing	Development	Pre-prod
 Seed data	 For machines	 Builds triggered by commits	 Lives as long as a test run	 Runs automated tests & QA	Test	
 Anonymized user data	 For entire dev team	 Reloads per pull request	 Long lived, imitate prod	 Sandbox for a release	Staging	
 Private user data	 For end-users	 Deploys per launch/release	 Lives forever	 For customers to enjoy	Prod	Prod

Среды



It is Wednesday,



my dudes

<https://tech.scribd.com/blog/2020/breaking-up-the-dag-repo.html>

Как обновлять код в Airflow

1. Пуш по кнопке в GitHub Actions / Jenkins / GitLab CI / etc.
2. Автоматическое подтягивание свежих файлов из S3
(например через objinsync)

```
$ /bin/objinsync pull  
s3://<S3_DAG_BUCKET>/airflow_home/dags  
<YOUR_AIRFLOW_HOME>/dags
```


Как обновлять инференс модели

При батч инференесе:

- Заливаем новую модель в Model Registry (MLFlow, ClearML)
- При следующем запуске новая модель автоматически подтянется (например, через тэг @latest)

При онлайн инференсе:

- Заливаем новую модель в Model Registry
- Останавливаем и перезапускаем сервис с новой моделью
- Заранее определяем endpoint в приложении, который позволит обновить действующую модель в проде без пересоздания сервиса

Микро ДЗ CI/CD

1. Настроить CI процесс линтинга и тестирования вашего кода
2. Настроить CD процесс создания и публикации библиотеки при PR в мастер
3. Настроить CD по сборке докер образа и его пуш в Docker Hub при обновлении тэга

Дополнительно:

<http://89.169.171.107:8000> – MLFlow

AWS_ACCESS_KEY_ID: YCAJE7EasWFd2LIH_j9tbt1Ar

AWS_SECRET_ACCESS_KEY: YCP5frOh73GPSCHB8_1OhKw7Nk259ak4wILSFhoF

MLFLOW_S3_ENDPOINT_URL: <https://storage.yandexcloud.net>

Bucket: mlops-storage-misis

Создавайте папку внутри бакета по пути mlops-storage-misis/<surname>_<name>