От Jupyter Notebook ⟶ к MLOps pipeline

2023

в связке
с ClearML

для проекта видео
аналитики Антифрод

NAPOLEON IT

О себе

NAPOLEON IT

Team Lead CV (Napoleon IT)
Проекты видеоаналитики

Middle ML Engineer (EPAM SYSTEMS)
ETL/BIG Data проект для Coca-Cola

Трижды победитель всероссийских хакатонов hacks.ai
Самара, Екатеринбург, Санкт-Петербург

Технический эксперт всероссийских хакатонов hacks.ai

Наша повестка

01

Введение

- Как велась разработка до
- Проект Антифрод
- Почему мы пошли в MLOps

02

MLOps pipeline

- Что это
- Что мы пробовали
- К чему мы пришли
- ClearML overview
- Как работает наш пайп

03

Заключение

- Преимущества и неудобства ClearML
- Как теперь ведется разработка
- Заключение

01

Введение

- Как велась разработка до
- Проект Антифрод
- Почему мы пошли в MLOps

Как велась разработка до

NAPOLEON IT

~40%
артефактов лежат локально

~45%
кода лежит локально

~65%
ручной однотипной работы для релиза

~35%
проекта удовлетворяет концепциям mlops

Проект Антифрод

В 2022 году в магазинах увеличилось число краж: чаще всего злоумышленников привлекали кондитерские изделия и колбаса. Зачастую товары уносят без оплаты вместе с корзинкой.

Тренд: хищения на кассах самообслуживания

80

краж в среднем
на 1 маг/мес

1 500 ₽

средняя сумма
кражи

на 34%

рост краж на КСО
в 2022 году

Проект Антифрод

NAPOLEON IT

Касса самообслуживания—зона, которая нуждается в максимальном контроле. Система CV способна защитить потенциальную точку возникновения фрода, предусматривая множество ситуаций.

01

Перестикеровка товаров

Покупатель взвесил авокадо как чеснок, производит покупку в зоне СО

02

Ошибочные действия

Покупатель увидел акцию “3 товара по цене 2” и пробил всего 2 товара на КСО

03

Вывоз тележки с товаром

Группа покупателей вывезла полную телегу товара через зону СО без оплаты покупки

04

Отсутствие сканирования товара

Покупатель положил в корзину 5 SKU, при этом оплатил только 3 SKU, 2 оставшихся пронес мимо сканера КСО

05

Сговор с ассистентом КСО

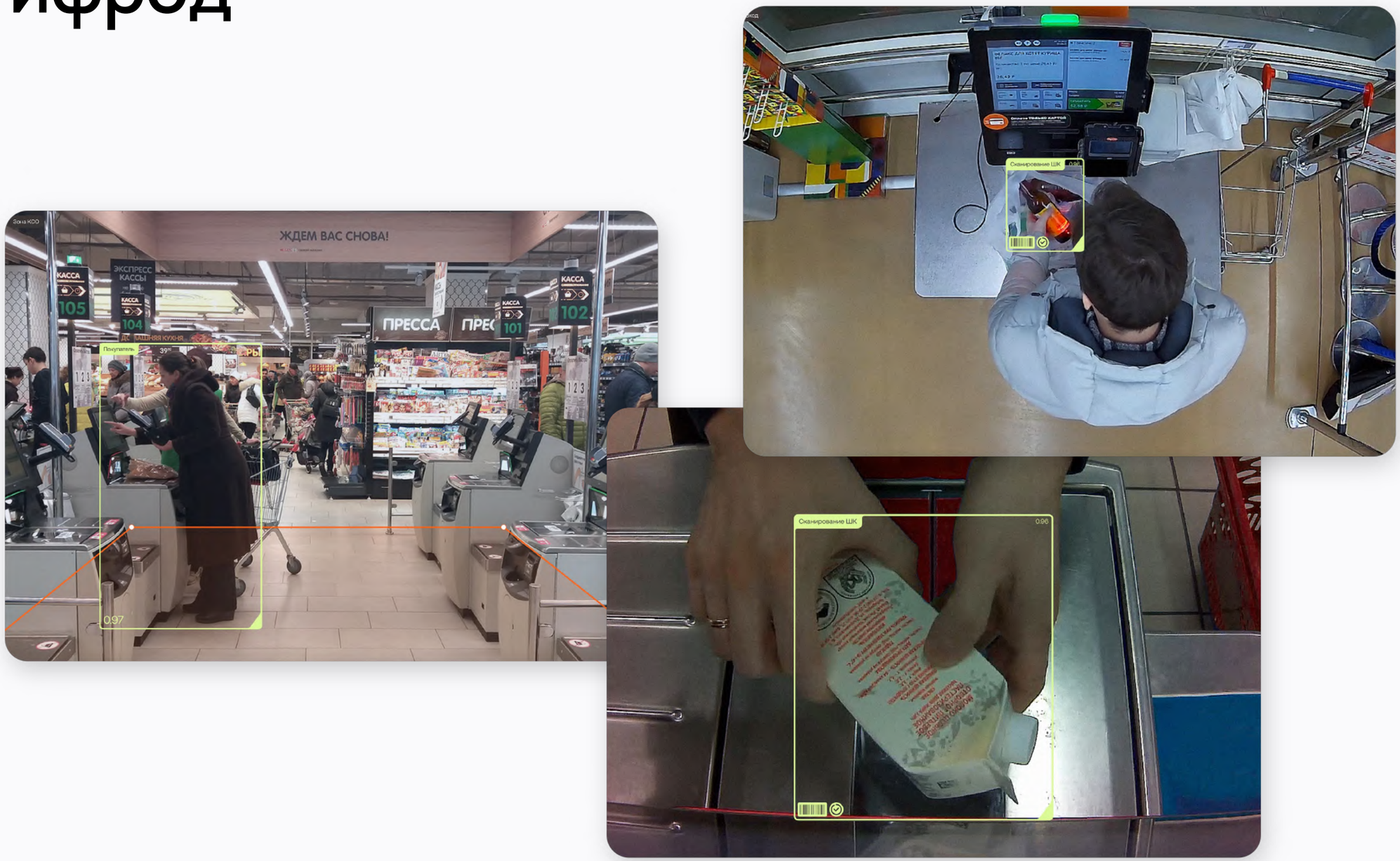
Покупатель пришел в зону КСО с товаром, ассистент зоны КСО под видом помощи аннулировал чек и отпустил покупателя

06

Техническая ошибка

Система селфскан выдала ошибку и не смогла добавить товар в корзину. Покупатель не заметил эту ошибку

Проект Антифрод



Почему мы пошли в MLOps

NAPOLEON IT

01

Поддержка как минимум 4-ех моделей

02

Сроки на разработку

03

Большое количество данных и экспериментов

02

MLOps pipeline

- Что это
- Что мы пробовали
- К чему мы пришли
- ClearML overview
- Как работает наш пайп

Что это

NAPOLEON IT

MLOps — это набор процедур, которые направлены на автоматизацию и ускорение процесса разработки новых моделей.

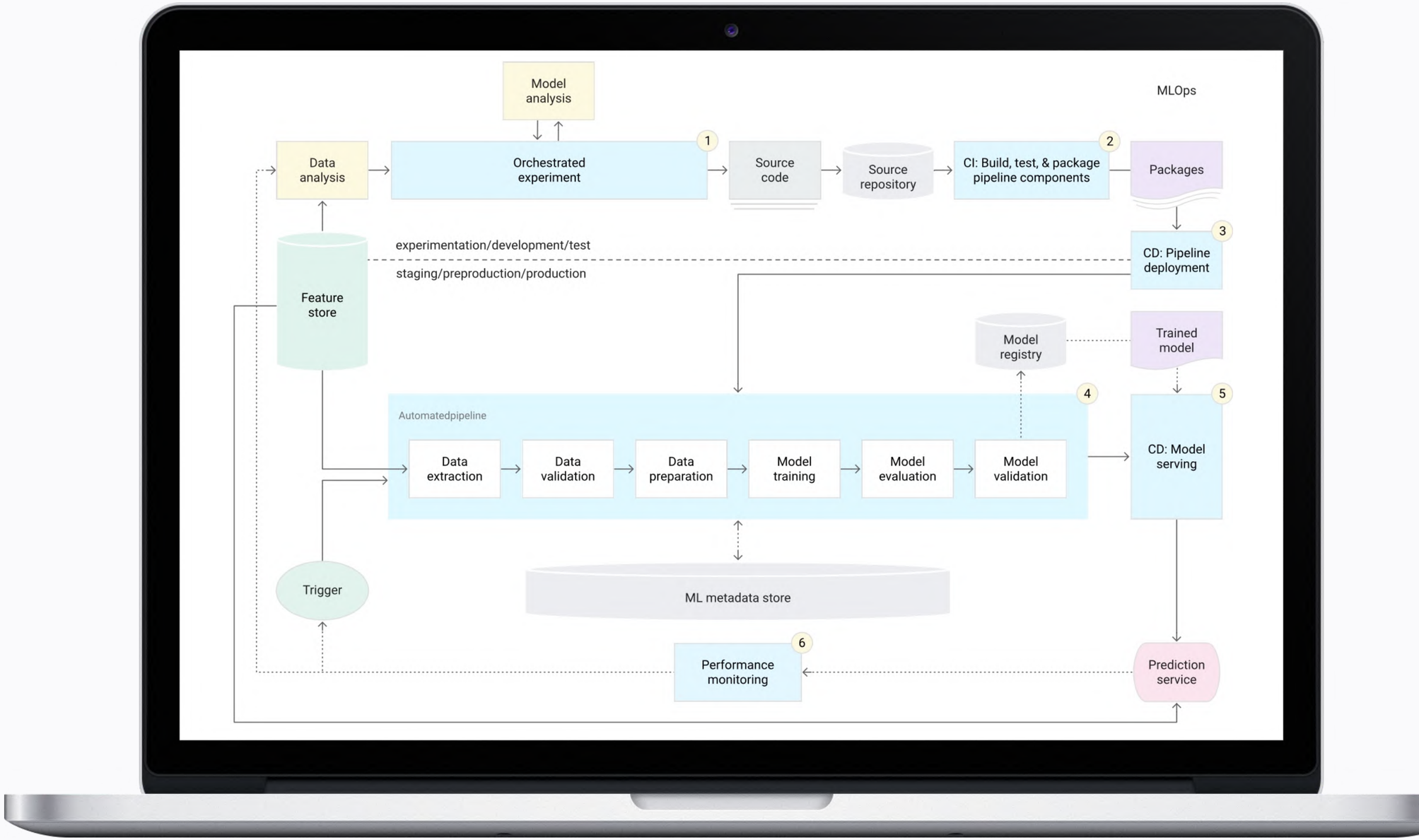
Включает в себя:

- процесс версионирования кода, данных, артефактов (модели, метрики, параметры)
- мониторинг
- возможность удаленного запуска задач

Основные принципы:

- версионирование (код, артефакты — данные, модели, метрики, эксперименты)
- автоматизация (уменьшение ручного вмешательства в релизы)
- взаимодействие с вычислительными мощностями

Что это


MLOps
Model analysis
Data analysis
Orchestrated experiment
1
Source code
Source repository
CI: Build, test, & package pipeline components
2
Packages
3
CD: Pipeline deployment
experimentation/development/test
staging/preproduction/production
Feature store
Model registry
Trained model
Automatedpipeline
4
5
Data extraction
Data validation
Data preparation
Model training
Model evaluation
Model validation
CD: Model serving
Trigger
ML metadata store
6
Performance monitoring
Prediction service

▸ Что мы пробовали

NAPOLEON IT

оркестратор

версионирование данных

хранение артефактов
и экспериментов

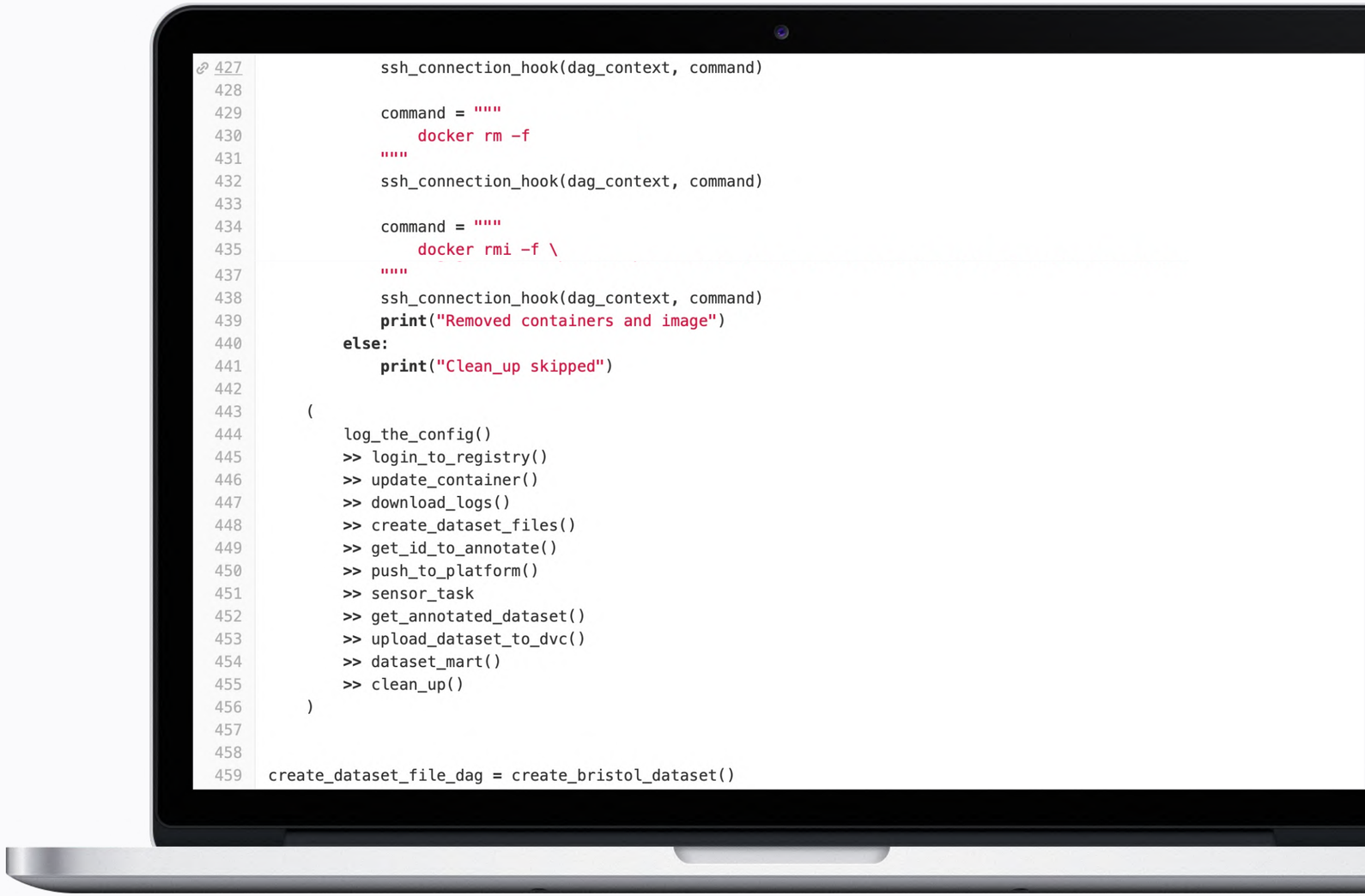
Что мы пробовали

NAPOLEON IT

Apache Airflow — это решение с открытым исходным кодом. Это оркестратор, который позволяет наладить разработку, планирование и мониторинг сложных рабочих процессов.

- Написан на Python
- Работает как планировщик ETL/ELT-процессов
- Есть хорошая админка
- Неограниченно расширяем

Что мы пробовали



```
427             ssh_connection_hook(dag_context, command)
428
429             command = """
430                 docker rm -f
431             """
432             ssh_connection_hook(dag_context, command)
433
434             command = """
435                 docker rmi -f \
437             """
438             ssh_connection_hook(dag_context, command)
439             print("Removed containers and image")
440         else:
441             print("Clean_up skipped")
442
443     (
444         log_the_config()
445         >> login_to_registry()
446         >> update_container()
447         >> download_logs()
448         >> create_dataset_files()
449         >> get_id_to_annotate()
450         >> push_to_platform()
451         >> sensor_task
452         >> get_annotated_dataset()
453         >> upload_dataset_to_dvc()
454         >> dataset_mart()
455         >> clean_up()
456     )
457
458
459 create_dataset_file_dag = create_bristol_dataset()
```

Но скрипт для подготовки датасета с помощью AirFlow занимает 459 строк...

▸ К чему мы пришли

NAPOLEON IT

оркестратор

версионирование данных

хранение артефактов
и экспериментов

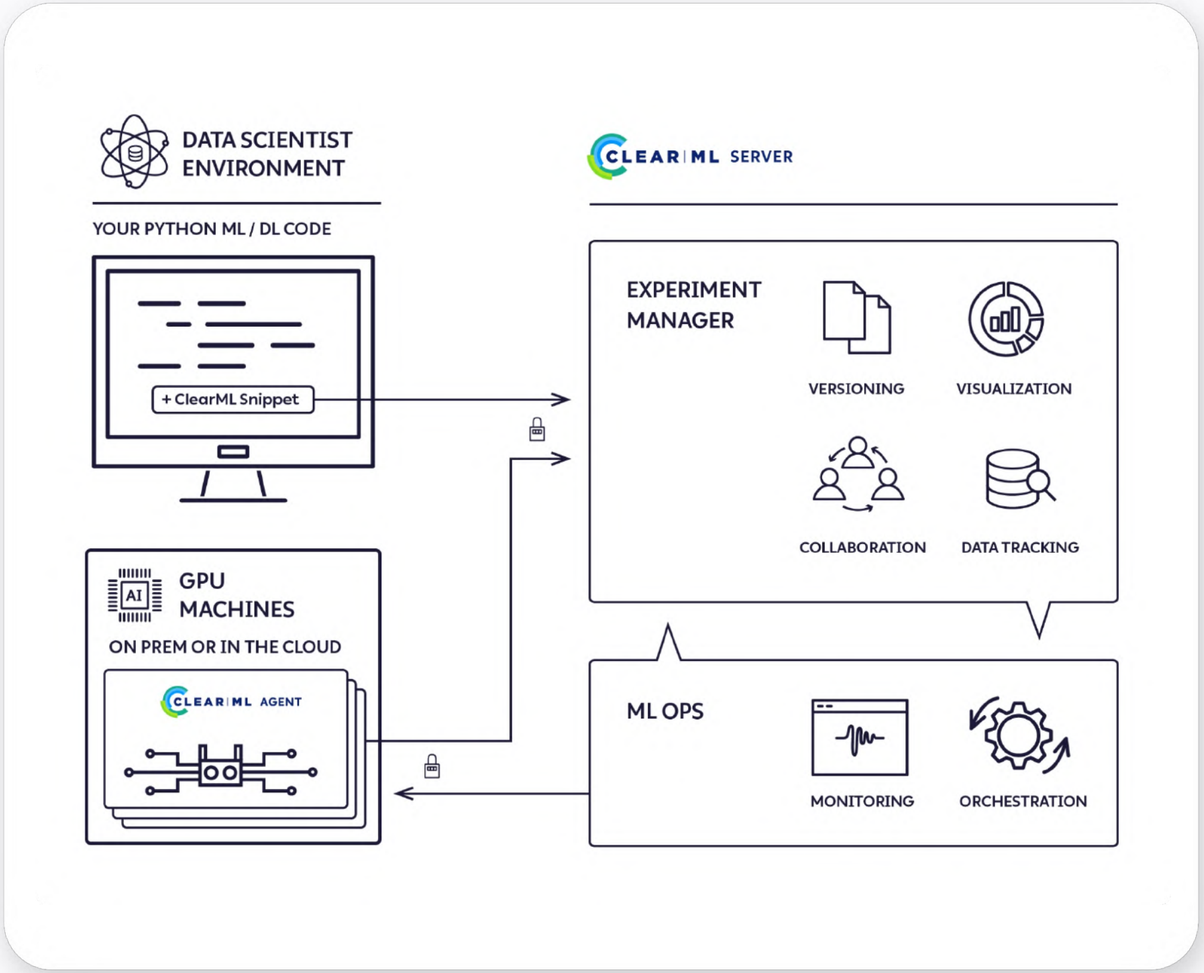
ClearML overview

NAPOLEON IT

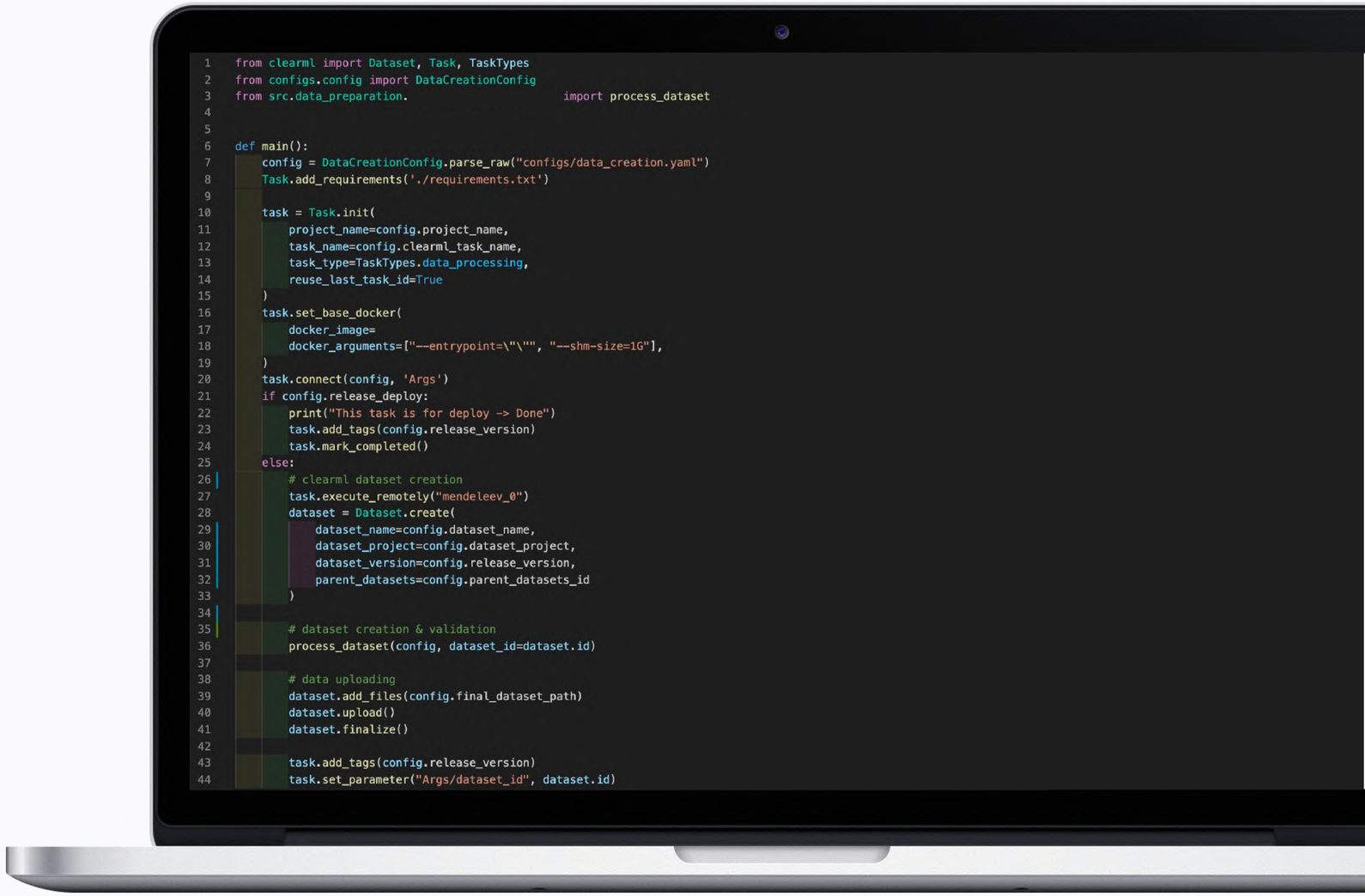
ClearML — инструмент с открытым исходным кодом для упрощения разработки и ведения экспериментов машинного обучения.

- Отслеживание и загрузка метрик и моделей с помощью нескольких строк кода
- Версионирование данных
- Возможность выстраивания пайпланов задач
- Удаленный запуск задач
- Воспроизведение экспериментов с помощью UI
- Мониторинг моделей

ClearML overview



ClearML. Snippet

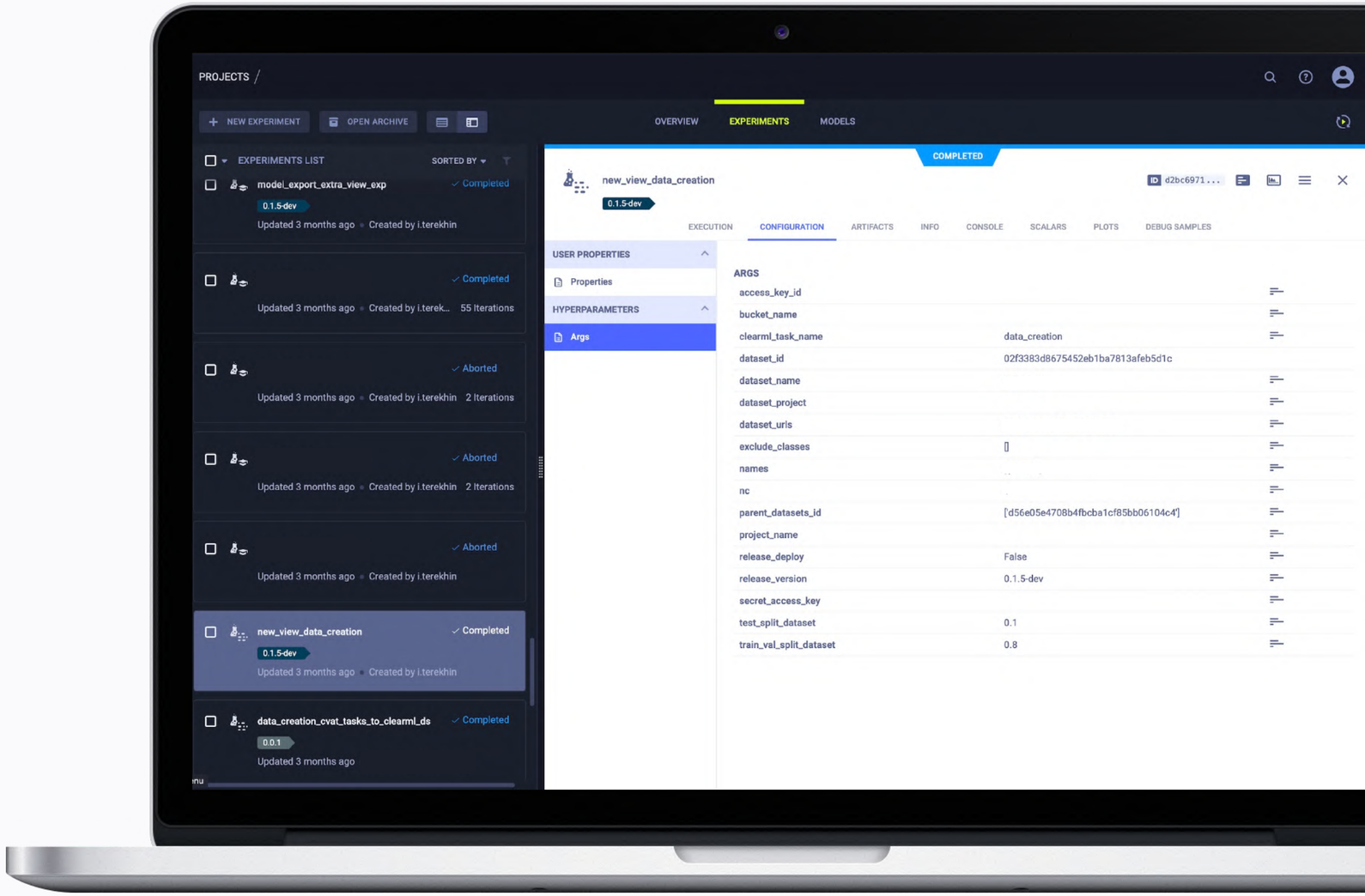


```python
from clearml import Dataset, Task, TaskTypes
from configs.config import DataCreationConfig
from src.data_preparation.                    import process_dataset


def main():
    config = DataCreationConfig.parse_raw("configs/data_creation.yaml")
    Task.add_requirements('./requirements.txt')

    task = Task.init(
        project_name=config.project_name,
        task_name=config.clearml_task_name,
        task_type=TaskTypes.data_processing,
        reuse_last_task_id=True
    )
    task.set_base_docker(
        docker_image=
        docker_arguments=["--entrypoint=\"\"", "--shm-size=1G"],
    )
    task.connect(config, 'Args')
    if config.release_deploy:
        print("This task is for deploy -> Done")
        task.add_tags(config.release_version)
        task.mark_completed()
    else:
        # clearml dataset creation
        task.execute_remotely("mendeleev_0")
        dataset = Dataset.create(
            dataset_name=config.dataset_name,
            dataset_project=config.dataset_project,
            dataset_version=config.release_version,
            parent_datasets=config.parent_datasets_id
        )

        # dataset creation & validation
        process_dataset(config, dataset_id=dataset.id)

        # data uploading
        dataset.add_files(config.final_dataset_path)
        dataset.upload()
        dataset.finalize()

        task.add_tags(config.release_version)
        task.set_parameter("Args/dataset_id", dataset.id)
```

Скрипт для создания датасета с помощью ClearML занимает 44 строки:

- Запуск на remote
- Создание датасета и его версии
- Повторный запуск через UI
- Автоматическое логирование

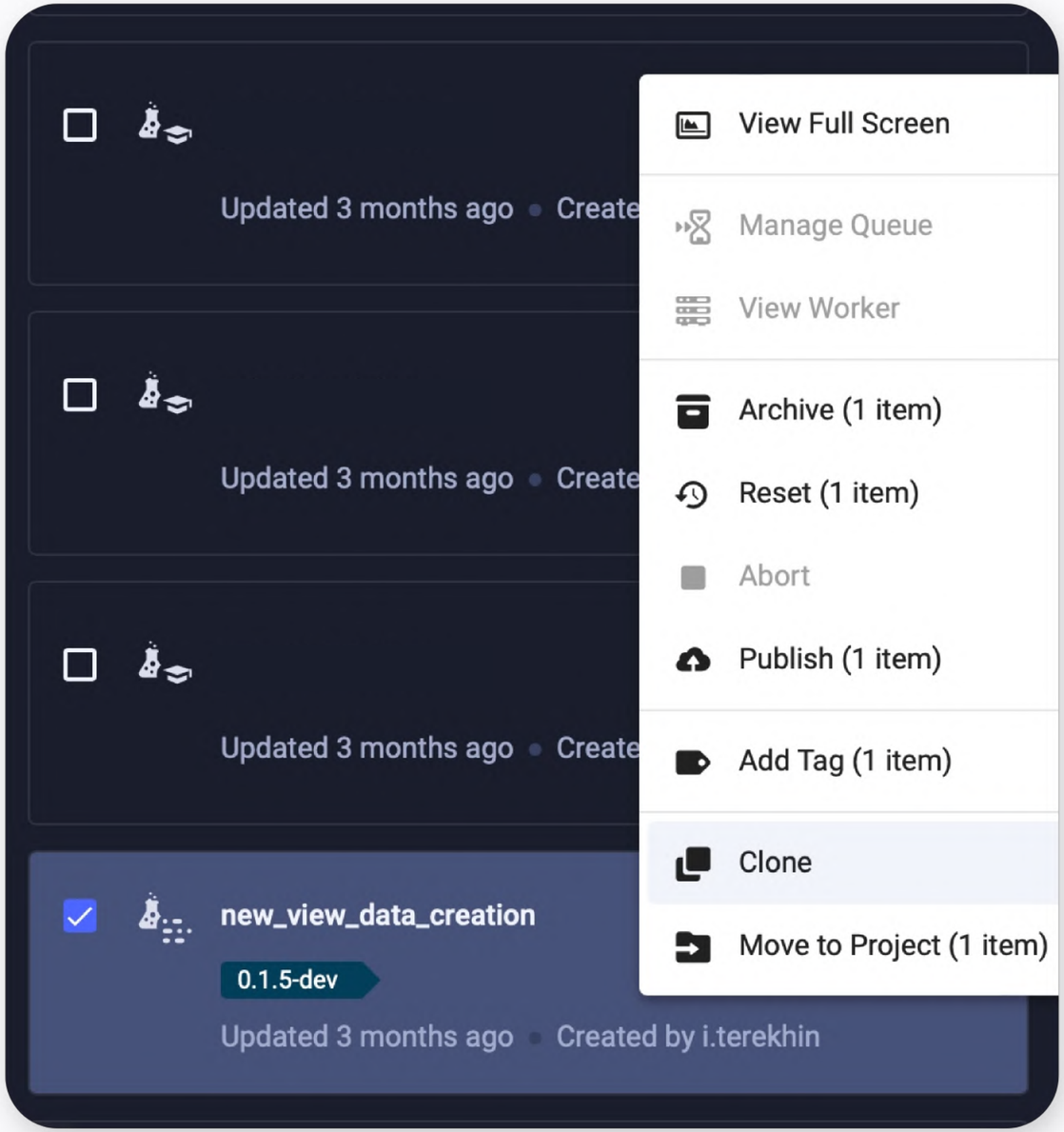
ClearML. Snippet



ClearML. Snippet

Возможные действия с тасками:

- Просмотр информации (графики, логи и тд)
- Поместить в очередь на remote
- Перезапустить с таким же конфигом
- Сделать read-only
- Добавить тэг
- Клонировать
- Переместить



ClearML. Experiment Manager

NAPOLEON IT

Внутри таски:

Execution

Конкретная ветка гитлаба и коммит, который запускается в таске, и изменения в скрипте по отношению к предыдущему

Configuration

Конфигурация таски

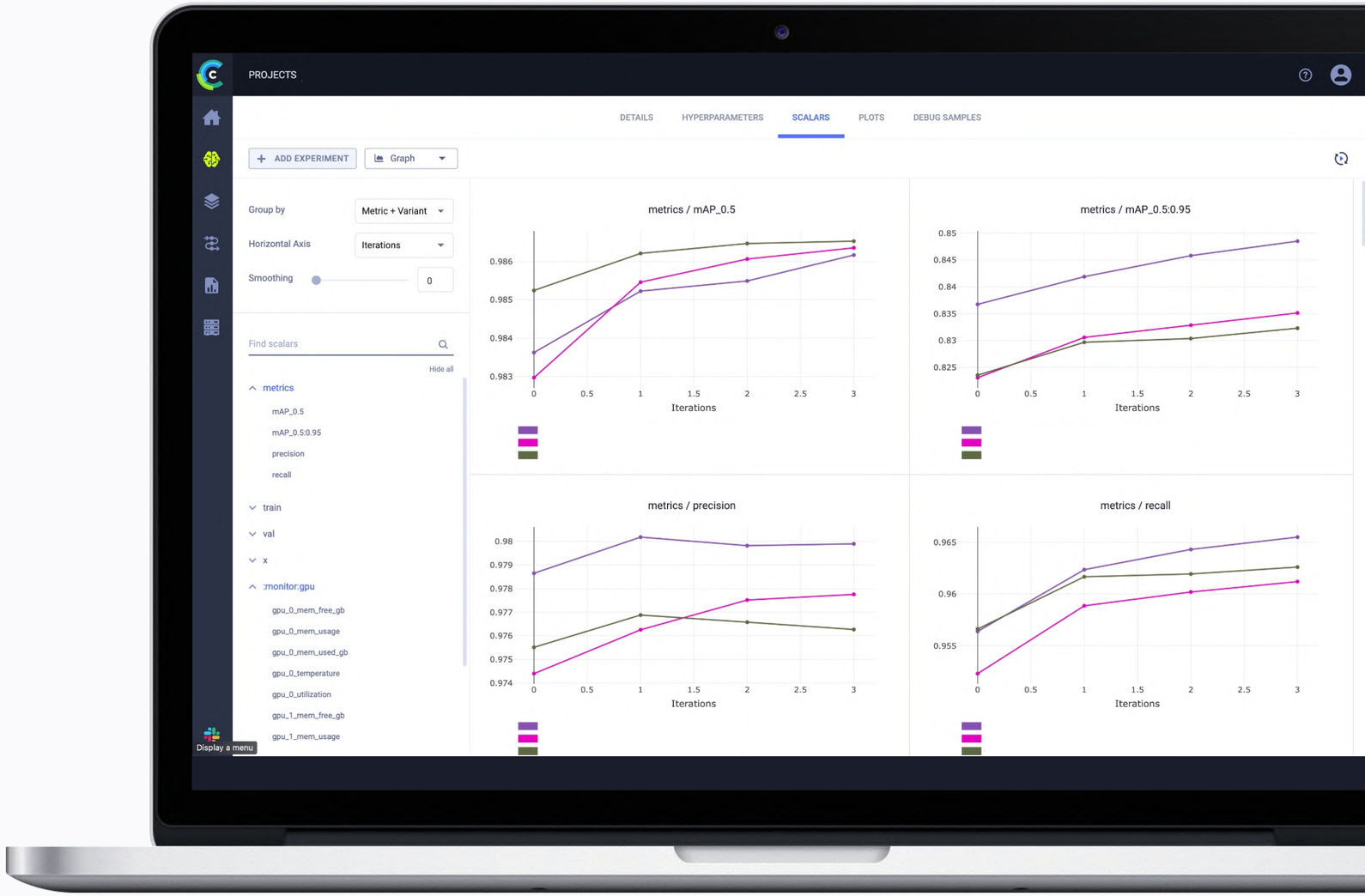
Artifacts

Все артефакты, которые появились в последствии выполнения таски

Info

На каком агенте запускалось, какие у него характеристики (гпу, память и тд), кто запускал, во сколько и тд

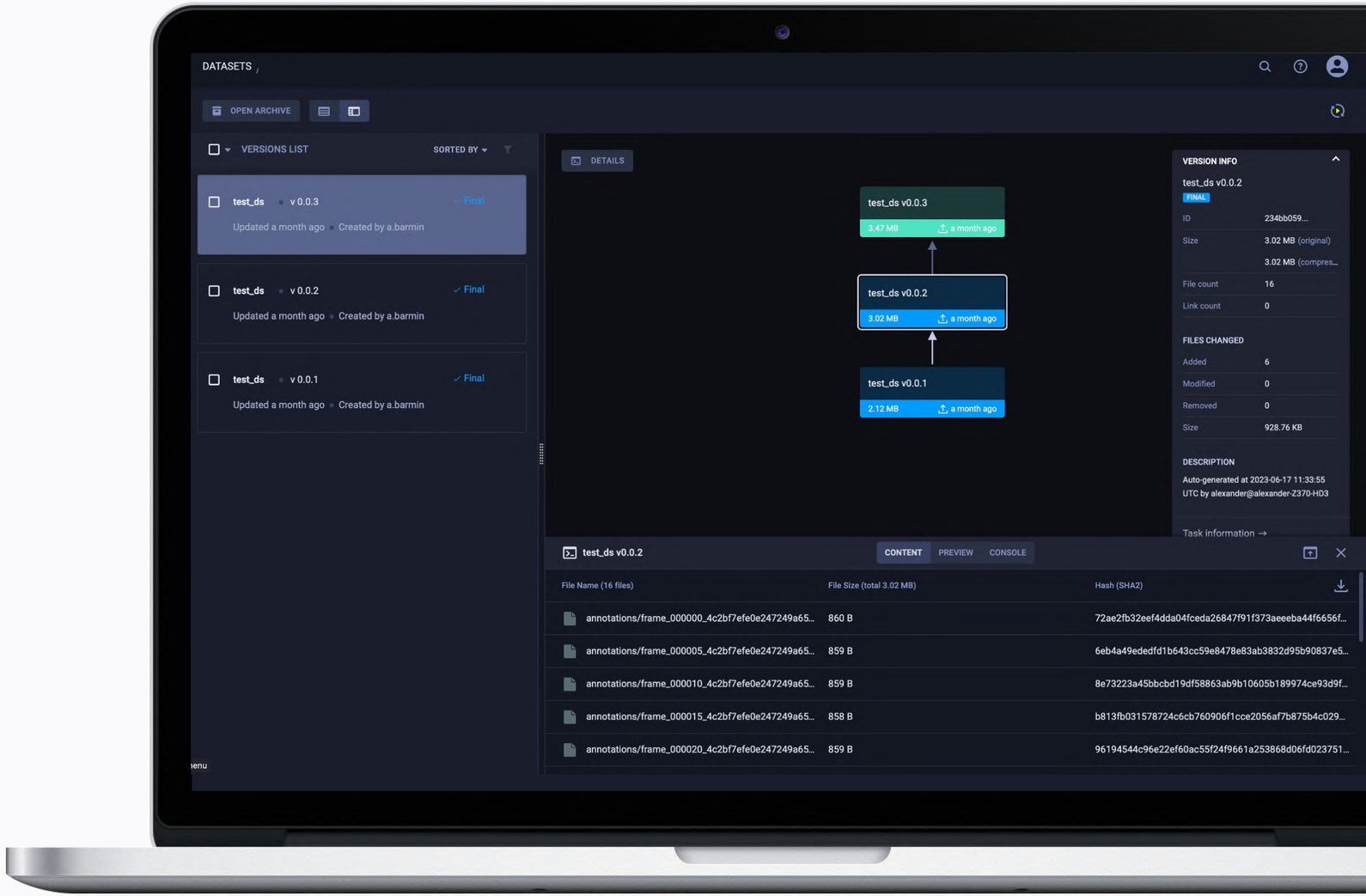
ClearML. Experiment Manager



- Возможность сравнения экспериментов

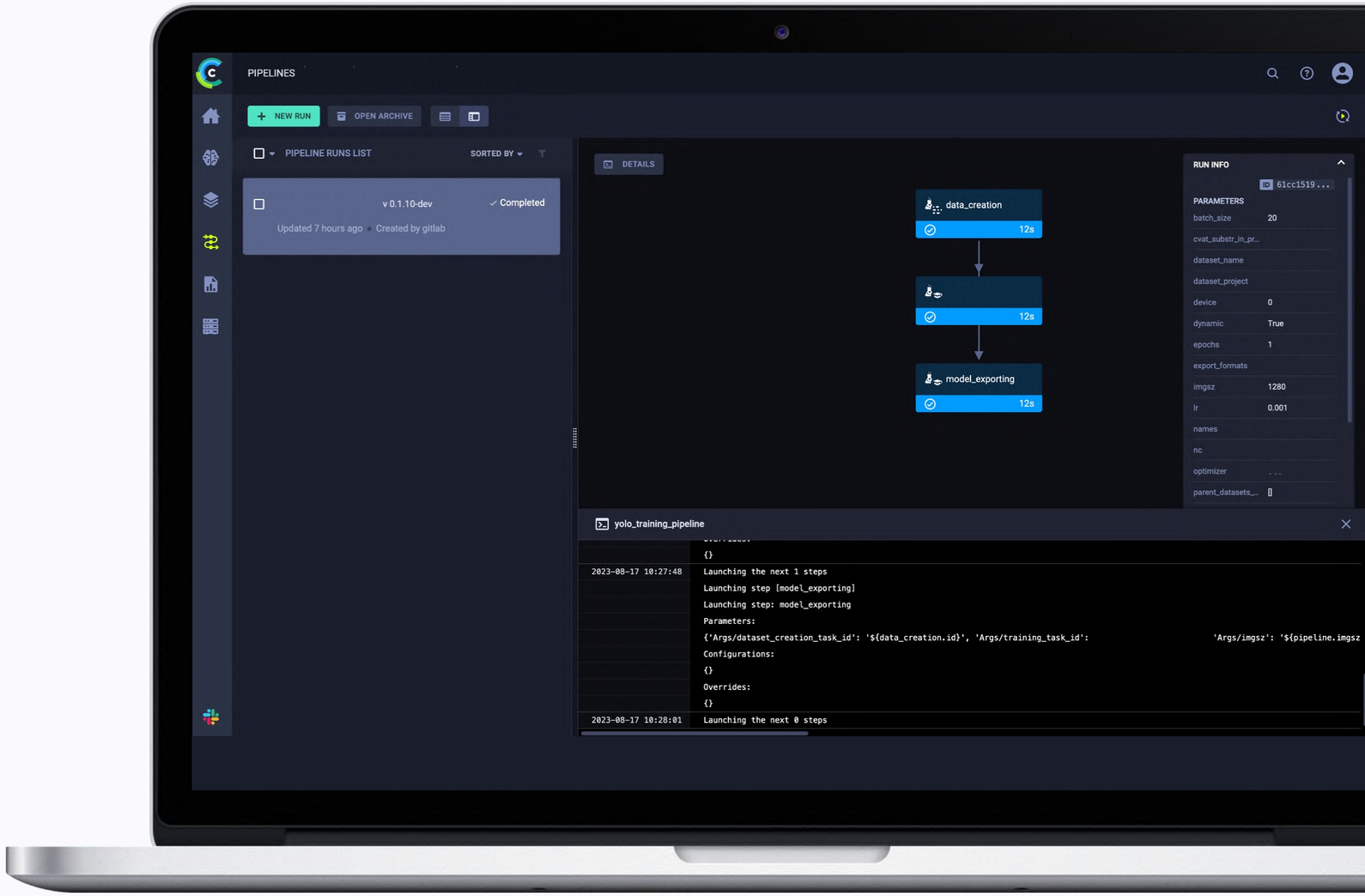
ClearML. Datasets

- Версионирование данных
- Видит отличия одной версии от другой
- В UI есть возможность предпросмотра датасета
- Простая логика выгрузки

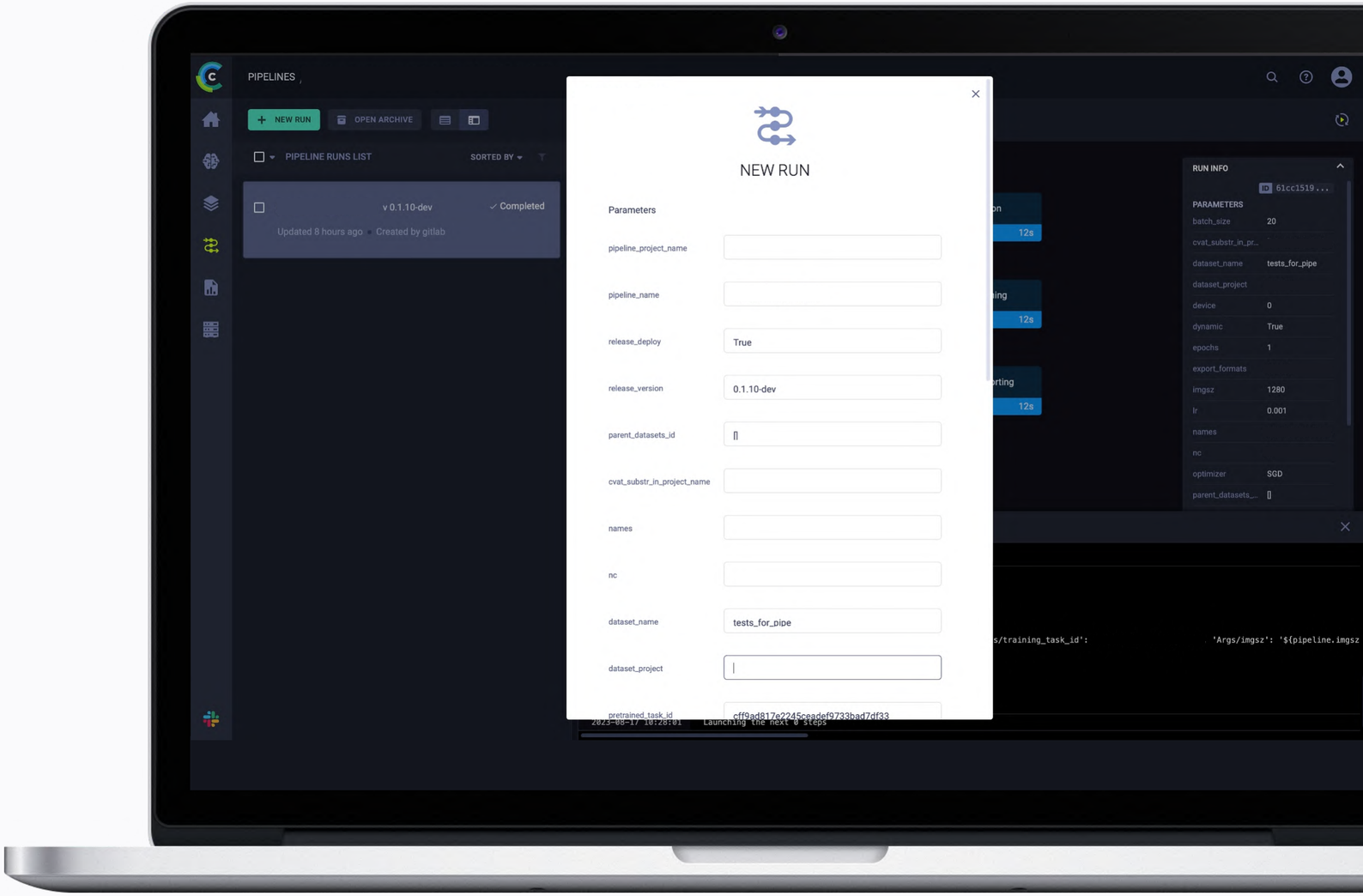


ClearML. Pipelines

- Отслеживание прогресса выполнения
- Возможность добавления параллельных этапов
- Возможность повторного запуска через UI

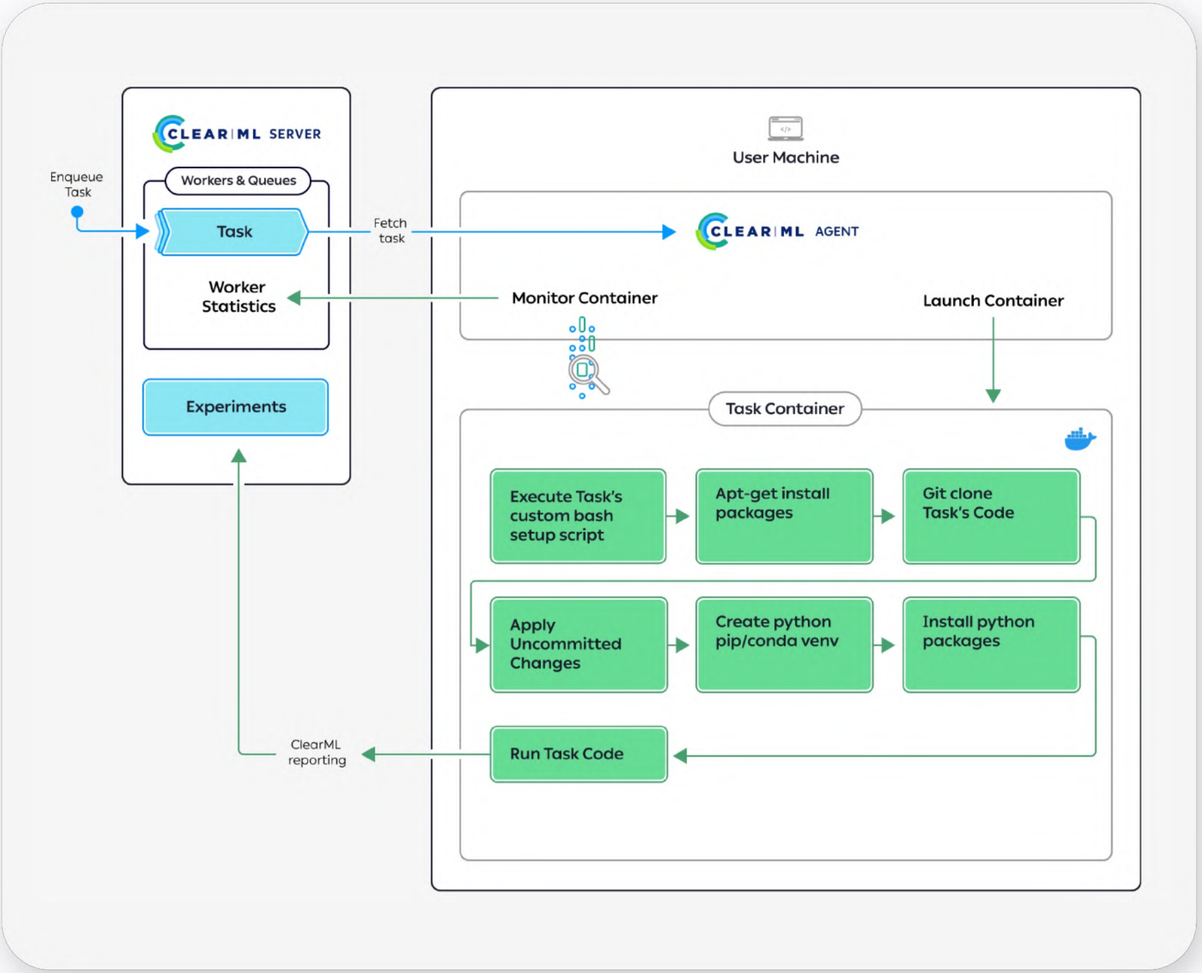


ClearML. Pipelines

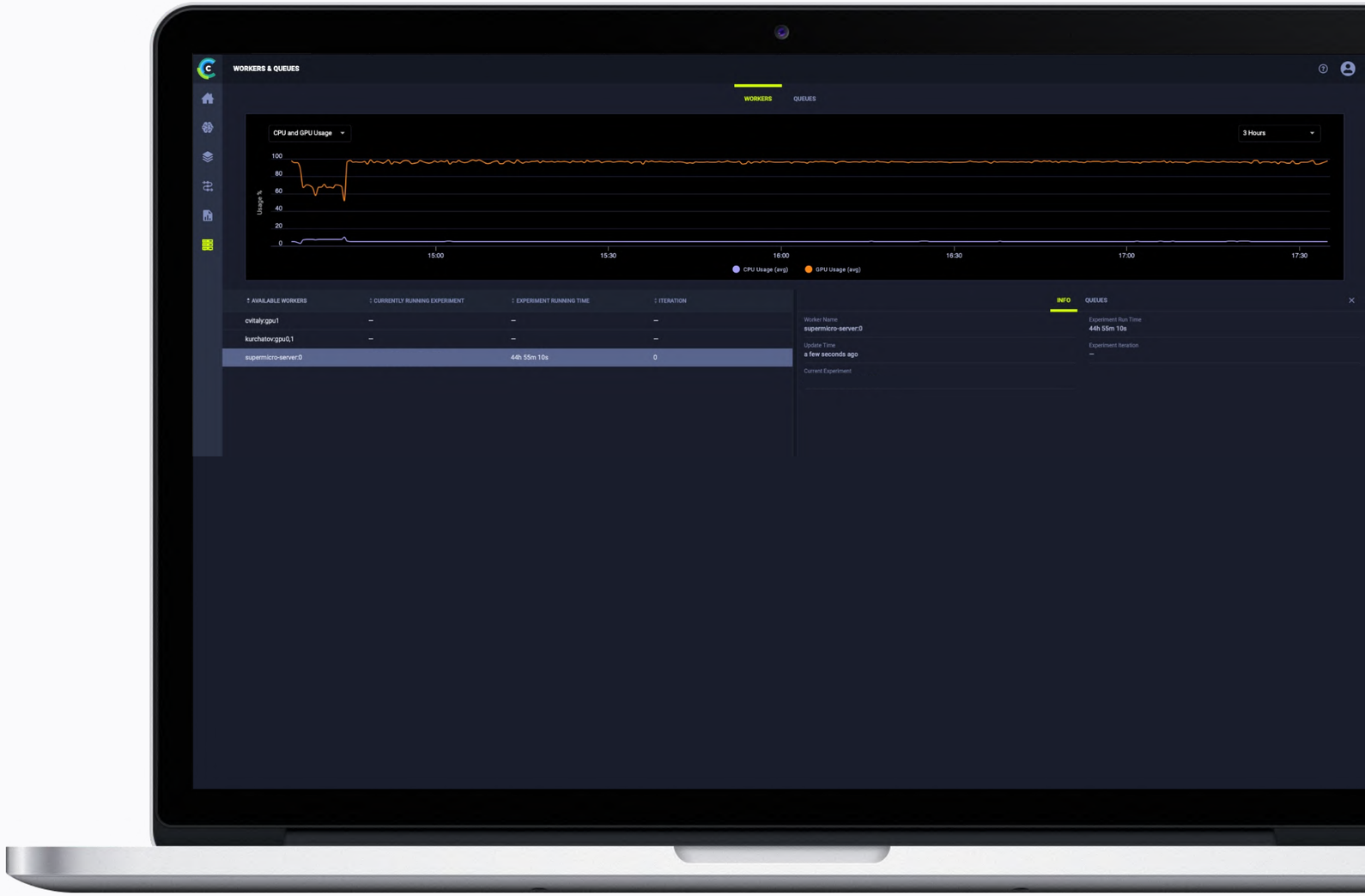


ClearML. Agent

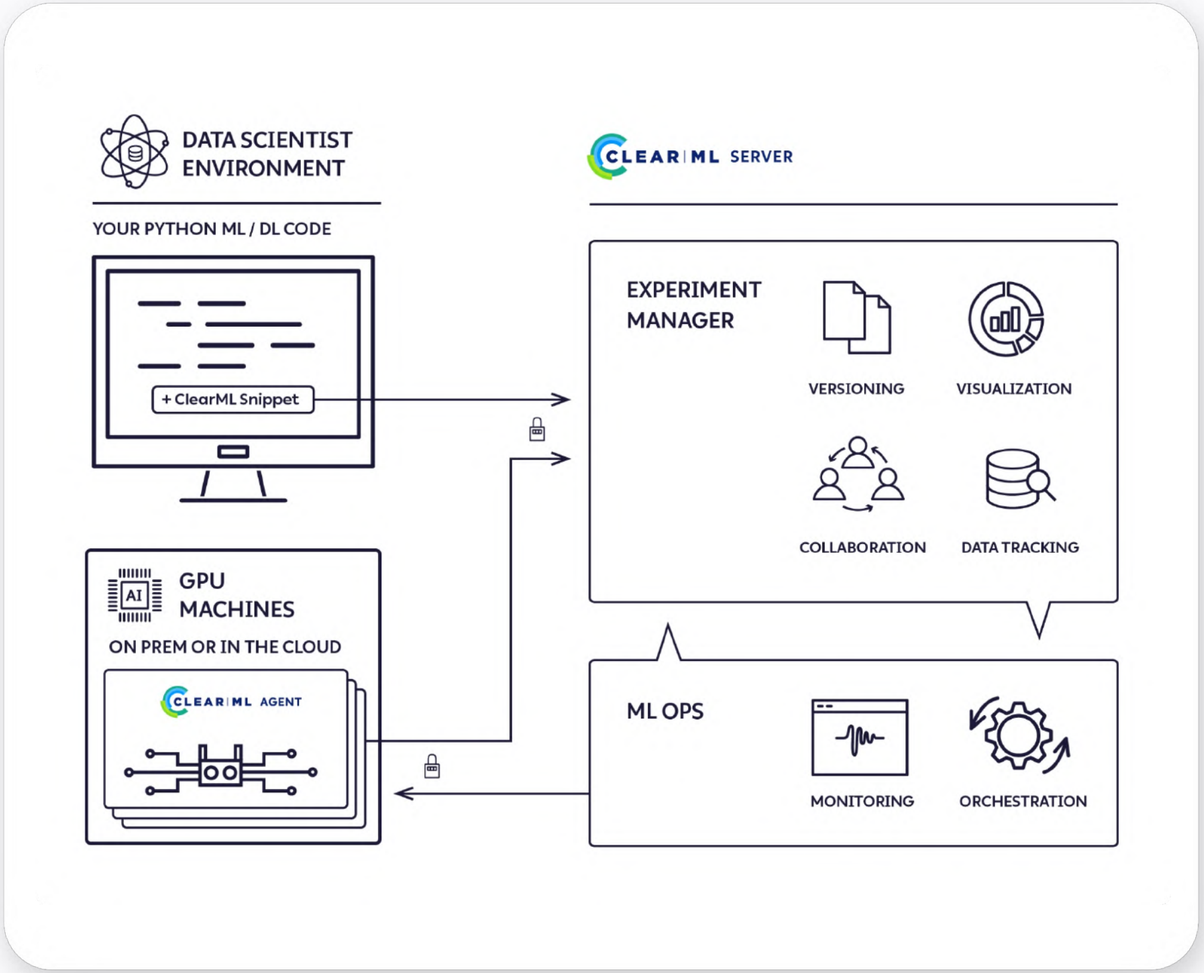
- Режимы pip/poetry/conda/docker
- Dynamic gpu-s
- Взаимодействие через UI
- Кэширование зависимостей
- Мониторинг очередей и агентов



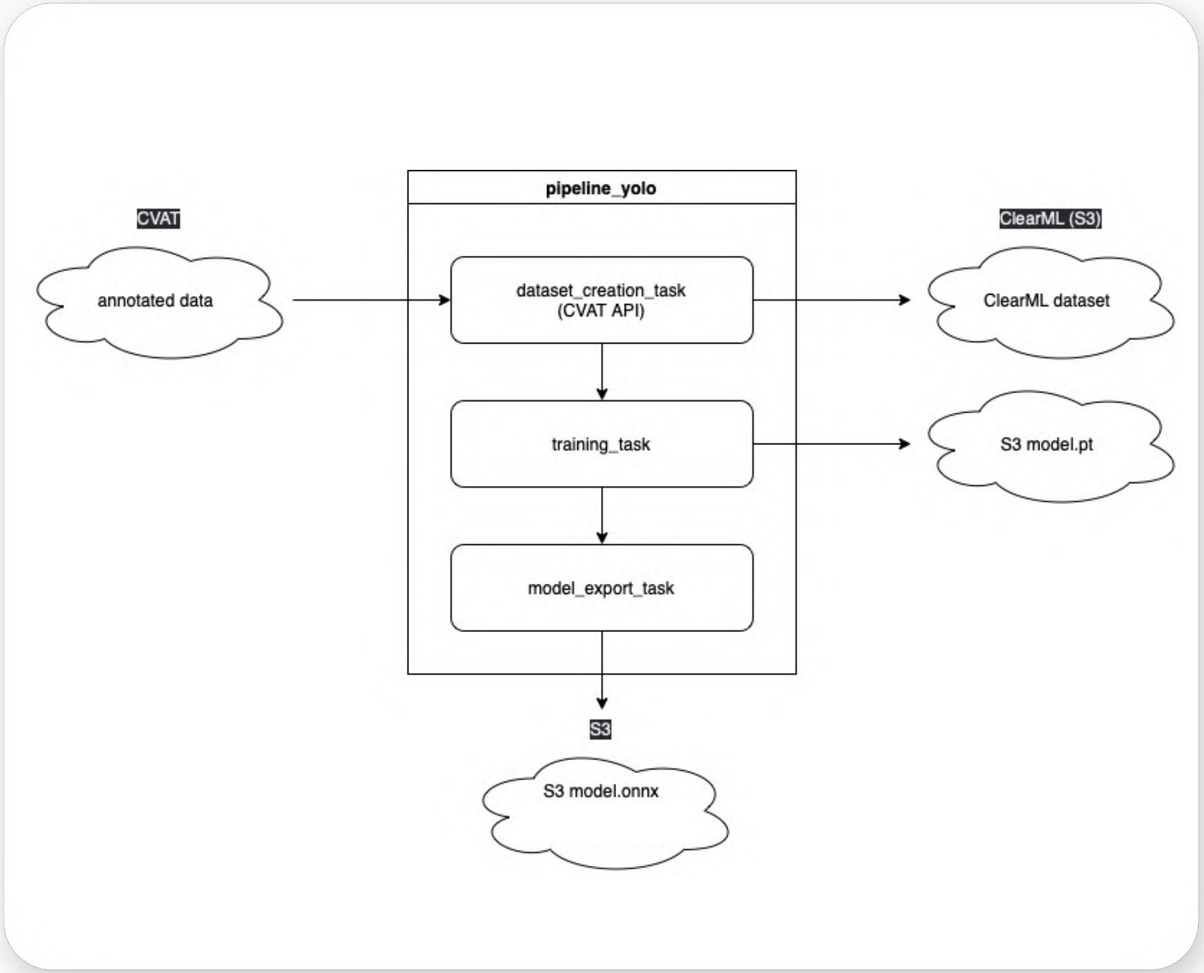
ClearML. Agent



ClearML.
ClearML overview

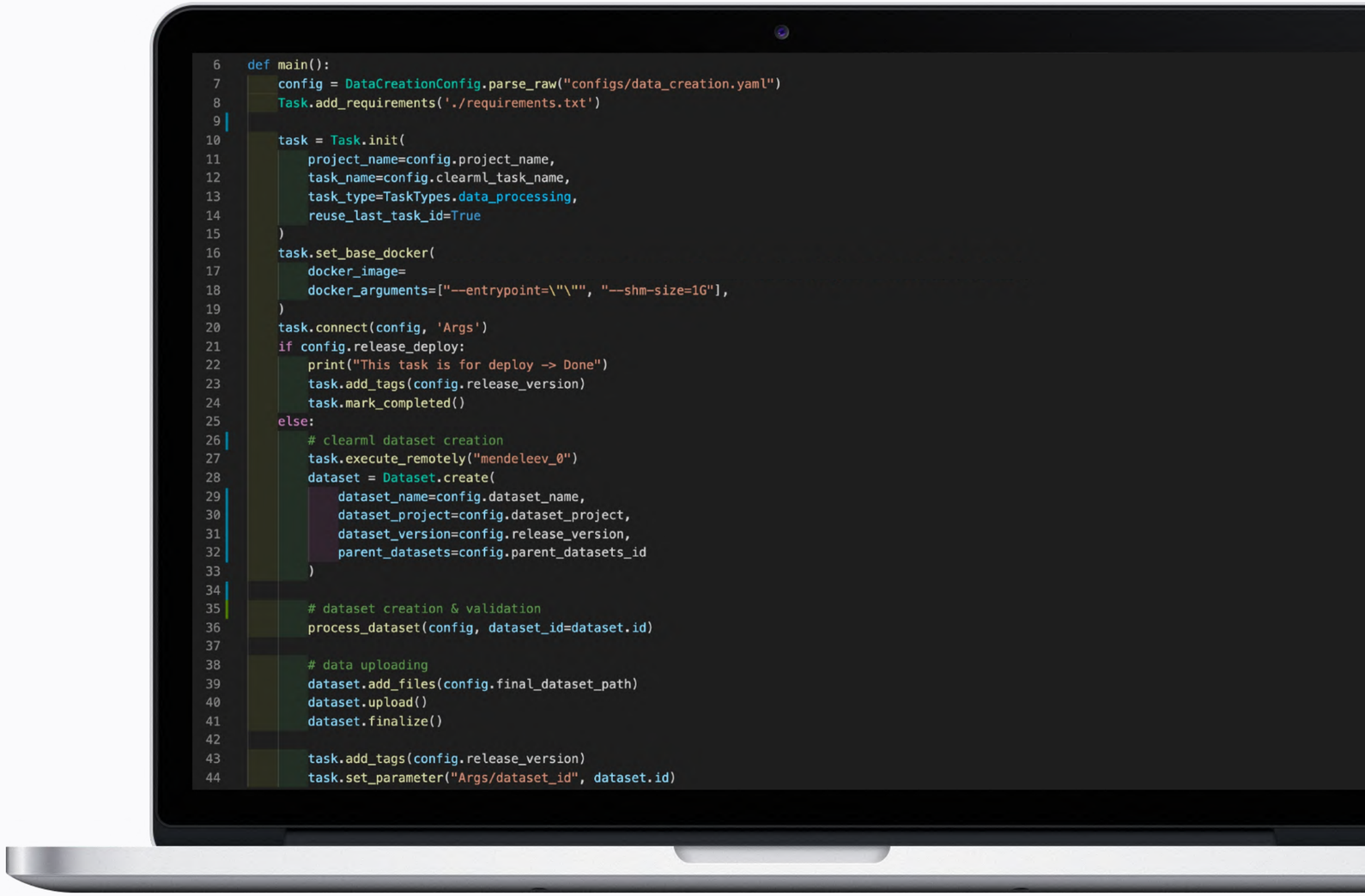


Принцип работы нашего пайпа



- Выгрузка с сервиса разметки CVAT
- Запуск обучения моделей на remote машинах с GPU
- Конвертация моделей

Принцип работы нашего пайпа

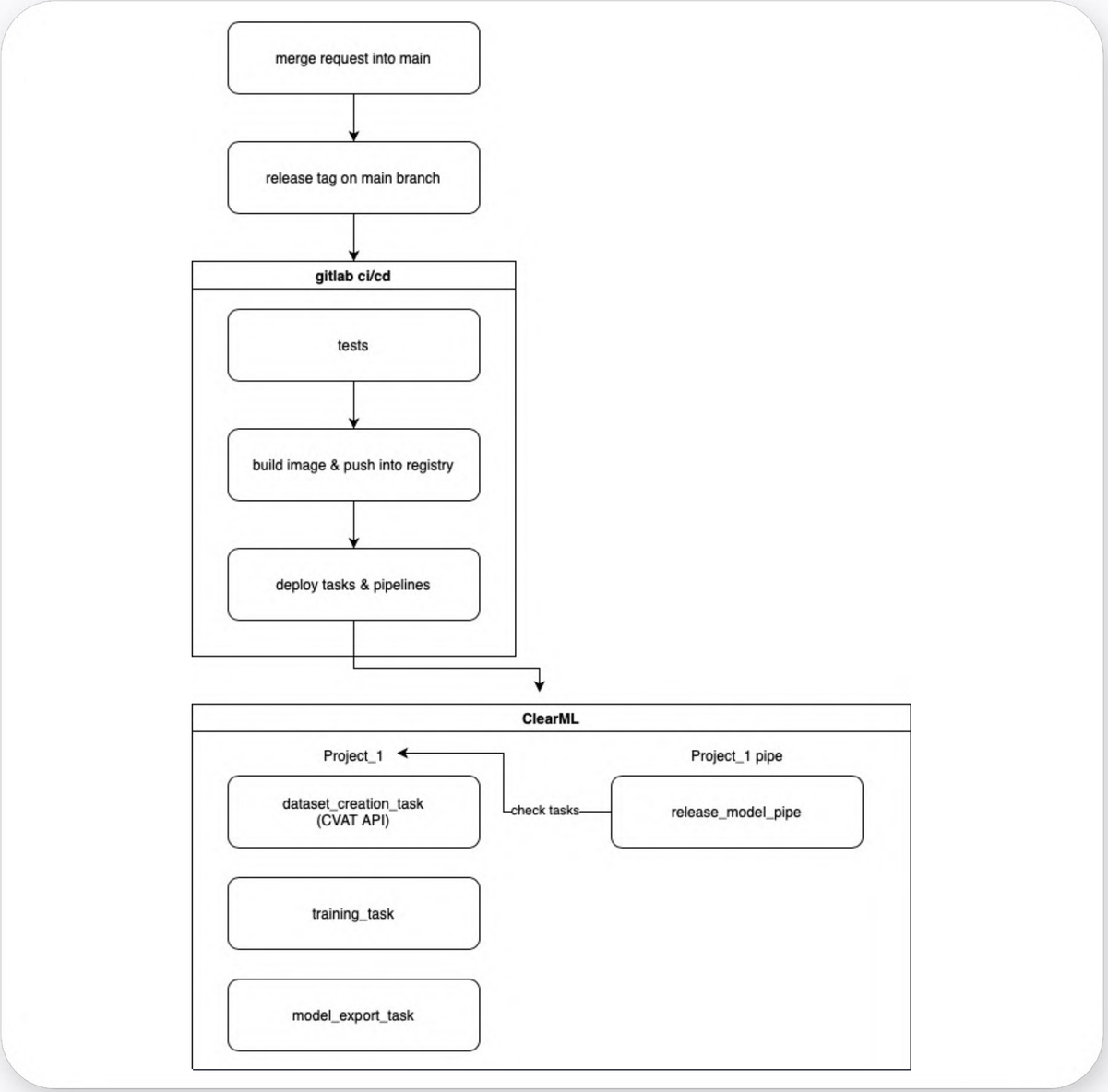


```python
def main():
    config = DataCreationConfig.parse_raw("configs/data_creation.yaml")
    Task.add_requirements('./requirements.txt')

    task = Task.init(
        project_name=config.project_name,
        task_name=config.clearml_task_name,
        task_type=TaskTypes.data_processing,
        reuse_last_task_id=True
    )
    task.set_base_docker(
        docker_image=
        docker_arguments=["--entrypoint=\"\"", "--shm-size=1G"],
    )
    task.connect(config, 'Args')
    if config.release_deploy:
        print("This task is for deploy -> Done")
        task.add_tags(config.release_version)
        task.mark_completed()
    else:
        # clearml dataset creation
        task.execute_remotely("mendeleev_0")
        dataset = Dataset.create(
            dataset_name=config.dataset_name,
            dataset_project=config.dataset_project,
            dataset_version=config.release_version,
            parent_datasets=config.parent_datasets_id
        )

        # dataset creation & validation
        process_dataset(config, dataset_id=dataset.id)

        # data uploading
        dataset.add_files(config.final_dataset_path)
        dataset.upload()
        dataset.finalize()

        task.add_tags(config.release_version)
        task.set_parameter("Args/dataset_id", dataset.id)
```

- 21-24 сделаны для деплоя тасок
- Подхватывает актуальный код
- Не выполняет на Gitlab Runner, а только деплоит

Принцип работы нашего пайпа

- Деплоим таски в проект ClearML
- Деплоим пайплайн в ClearML
- Используем механизм base_parent_project чтобы пайплайн подхватил последние задеплоенные таски с родительского проекта



03

Заключение

- Преимущества и неудобства ClearML
- Как теперь ведется разработка
- Выводы

Преимущества ClearML

NAPOLEON IT

Инструмент конкретно под ML

автоматическое логирование артефактов, датасеты, отображение метрик

Один инструмент содержит обширный функционал

мониторинг моделей, репорты, эксперименты, пайплайны и тд

Активная поддержка community

Неудобства ClearML

NAPOLEON IT

Недоступно разделение прав пользователей

по проектам, пространствам

Чтобы добавить нового пользователя нужно ребутать ClearML Server

Новый, развивающийся проект = несовпадение документации

Task в UI не подтягивает указанный commit

Достаточно проблематичный дебаг на remote тачках

Как велась разработка до

NAPOLEON IT

~40%
артефактов лежат локально

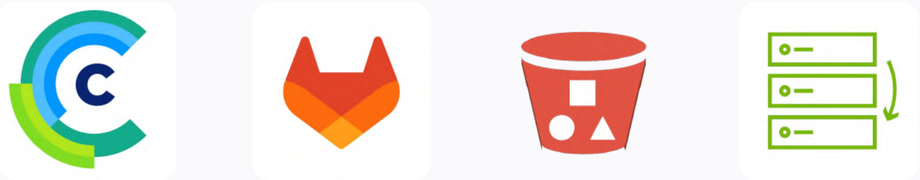
~45%
кода лежит локально

~65%
ручной однотипной работы для релиза

~35%
проекта удовлетворяет концепциям mlops

Как ведется разработка сейчас

NAPOLEON IT



0%
артефактов лежат локально

0%
кода лежит локально

~15%
ручной однотипной работы для релиза

~85%
проекта удовлетворяет концепциям mlops

Выводы

NAPOLEON IT

- MLOps – необходимая вещь в компаниях (ClearML – как вариант)
- Сократили необходимое время для релиза новой модели в 3 раза
- Меньше переживаний по поводу потери данных и артефактов из-за непредвиденных ситуаций
- Стало проще отслеживать эксперименты и прогресс моделей

NAPOLEON IT

Игорь Терехин

Team Lead CV

Telegram: spannenberger
Email: i.terekhin@napoleonit.com