neu.ro

Gartner
COOL VENDOR 2019
In AI Core Tech

“Зелёные” DGX, виртуальные GPU, оркестрация, интеграция и взаимозаменяемость, – или как мы строим MLOps платформу будущего

Мария Давыдова, Head of Product @ Neu.ro, COO @ AIIA

30 октября 2021 года

Обо мне

- Разработчик > 15 лет
- Инженер MLOps > 2 лет
- Участник конференций и митапов > 15 лет
- Community manager > 10 лет
- Блоггер > 10 лет

Разработчикам ПО и инженерам МашО необходимы качественные инструменты для того, чтобы делать качественные продукты

Мария Давыдова, aka **NeuroMasha**

neu.ro

О Neu.ro

Мы - признанная команда специалистов по ML и разработке ПО, нацеленная решать проблемы MLOps и поддерживать ответственный подход к ИИ

Компания

CVPR 2018 SALT LAKE CITY • JUNE 18-22

NeurIPS

ACL 2019 ANNUAL MEETING

Competition Winners

kaggle Masters

python Core Developers

170+ Papers in ML/DL

2019 Beta Launch

Партнерства

NVIDIA INCEPTION PROGRAM

AI Infrastructure Alliance

Google Cloud

Microsoft Partner Network

aws partner network

Опыт

pdc Panasonic

MIT Massachusetts Institute of Technology

REVENUEGRID

SAMSUNG

EMBL European Molecular Biology La

HOKUYAKU TAKEYAMA HOLDINGS

MEGAFON

О чём речь?

Машинное обучение != Разработка ПО

Некоторые задачи MLOps

- **Масштабирование API**
  - Классический DevOps
- **Автоматизация выкладки моделей**
  - DevOps + тестирование моделей
- **Отслеживание экспериментов**
  - Множество переменных и выходов данных
- **Мониторинг качества прогнозов моделей**
  - Как отслеживать? Как реагировать?
- **Дрейф данных (data drift)**
  - Данные пользователей со временем меняются
- **Управление обновлением данных**
  - Отслеживание изменения бинарных файлов

Image source: https://xkcd.com/1425/ (24.09.2014)

Компоненты системы MLOps

Hardware

- Высокопроизводительные GPU
- Хранилища
- CPU, память, сеть

Software

- Решение ряда разнородных задач, от сбора и версионирования данных до деплоя и мониторинга API
- Средства автоматизации процессов

Wetware

- Data architects, data engineers, data scientists, software developers, devops engineers, mlops engineers, ...

Почему ML != SD?

Критерий	Традиционная разработка ПО	Модели машинного обучения
Статус проекта	Версия кода	Версия кода + Версия данных
Время сборки	Несколько минут	Несколько часов/дней
Стоимость сборки	Несколько центов/долларов	Несколько сотен/тысяч долларов
Поведение	Детерминированное	Недетерминированное
Производительность со временем	Стабильная	Обычно падает
Мониторинг	Системные показатели	Системные показатели + качество модели
Ответственность	Разработчики	Распределена между командами

Требования к идеальной системе MLOps

01	**Воспроизводимость** **Reproducibility**	• Повторяемость экспериментов через версионирование ассетов и метаданных
02	**Масштабирование** **Scalability**	• Получение необходимых ресурсов как в облаке, так и в гибридной инфраструктуре
03	**Непрерывность** **Continuity**	• Автоматизация • CI/CD/CT
04	**Коллаборация Collaboration**	• Совместная работа разных команд в одном окружении
05	**Ответственность** **Accountability**	• Управление правами доступа • Логи аудита
06	**Этичность** **Ethics**	• Мониторинг и анализ работы моделей • Объяснение результатов (explainability)
07	**Перспектива** **Future-proof**	• Способность к развитию системы • Избежание vendor lock-in

И как этого добиться?

Ингредиент 1. Надежная инфраструктура

- **Облака**
  - AWS
  - Azure
  - GCP
  - Другие провайдеры с VMware либо managed K8s
- **On-premise**
- **DGX**
  - 8 x A100 - это масштабируемость от **56 x 1/7 A100** (MIG) для инференса до **8 x A100 с 320 GB RAM** для действительно больших моделей, в одном корпусе
  - DGX можно также объединять в суперкомпьютеры
  - У Neu.ro есть кластер DGX в Исландии на **зелёной энергии**

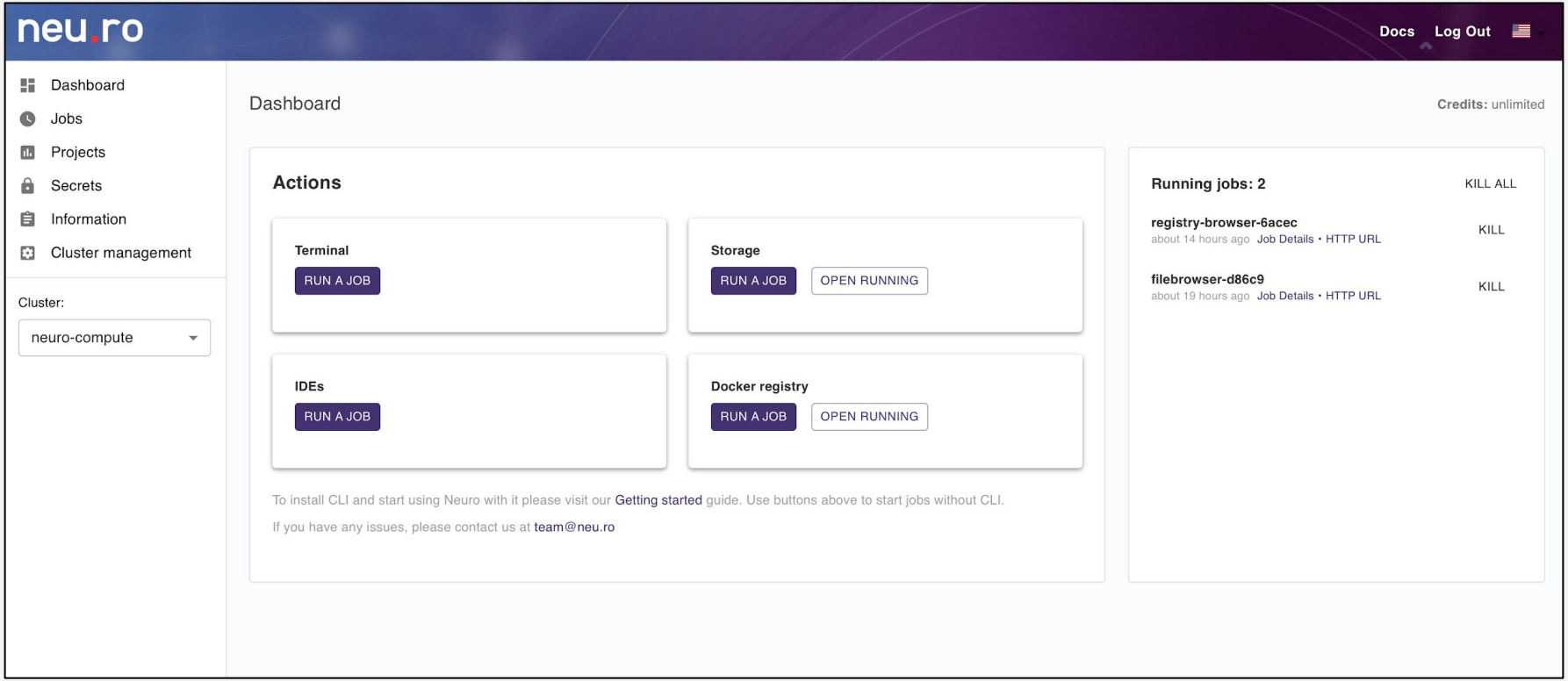
Ингредиент 2. K8s + Tools

- **Базовая оркестрация ресурсов и контейнеров**
  - Все выполняемые на платформе задания (jobs) контейнеризованы
- **Ingress out-of-the-box**
  - У любого контейнера предсказуемый URL
- **Prometheus + Grafana для репортов**
- **Docker Registry для хранения контейнеров**
- **Хранилища**
  - NFS для быстрого сетевого хранилища
  - PVC для монтируемых дисков
  - Object storage для длительного хранения ассетов
- **Базы данных**

Ингредиент 3. Программный слой Neu.ro

- **Продвинутая оркестрация ресурсов**
  - Round robin
  - Preemptible nodes
  - Различные ресурсные квоты
  - Отчёты
- **Оркестрация прав**
  - Все элементы системы (задания-контейнеры, хралище, образы Docker, роли, проекты) находятся под управлением системы прав доступа
  - Role-based access control & SSO
  - Секреты для доступа к third-party services
- **Оркестрация процессов**
  - Собственный движок пайплайнов (Neu.ro Flow) с YAML
  - Многоуровневые ациклические графы исполнения
  - Модульность

Ингредиент 4. Интерфейсы



Ингредиент 4. Интерфейсы

run

Run a job with predefined resources...

Usage

```
neuro job run [OPTIONS] IMAGE [CMD]...
```

Run a job with predefined resources configuration.

`IMAGE` container image name.

`CMD` list will be passed as commands to model container.

Examples

```
# Starts a container pytorch:latest on a machine with smaller GPU resources
# (see exact values in `neuro config show`) and with two volumes mounted:
#   storage:///<home-directory>   --> /var/storage/home (in read-write mode),
#   storage://neuromation/public --> /var/storage/neuromation (in read-only mode).
$ neuro run --preset=gpu-small --volume=HOME pytorch:latest

# Starts a container using the custom image my-ubuntu:latest stored in neuromation
# registry, run /script.sh and pass arg1 and arg2 as its arguments:
$ neuro run -s cpu-small image:my-ubuntu:latest --entrypoint=/script.sh arg1 arg2
```

Options

Name	Description
`-s` , `--preset PRESET`	Predefined resource configuration (to see available values, run `neuro config show`)

async-for **monitor**(*id: str*) → AsyncIterator[bytes] [source]

Get job logs as a sequence of data chunks, e.g.:

```
async for chunk in client.jobs.monitor(job_id):
    print(chunk.encode('utf8', errors='replace')
```

Parameters: **id** (*str*) – job **id** to retrieve logs.
Returns: **AsyncIterator** over **bytes** log chunks.

async-with **port_forward**(*id: str, local_port: int, job_port: int, *, no_key_check: bool = False*) → None [source]

Forward local port to job, e.g.:

```
async with client.jobs.port_forward(job_id, 8080, 80):
    # port forwarding is available inside with-block
```

Parameters:
- **id** (*str*) – job **id**.
- **local_port** (*int*) – local TCP port to forward.
- **jot_port** (*int*) – remote TCP port in a job to forward.

coroutine **resize**(*id: str, *, w: int, h: int*) → None [source]

Resize existing TTY job.

Parameters:
- **id** (*str*) – job **id**.
- **w** (*int*) – New screen width.
- **h** (*int*) – New screen height.

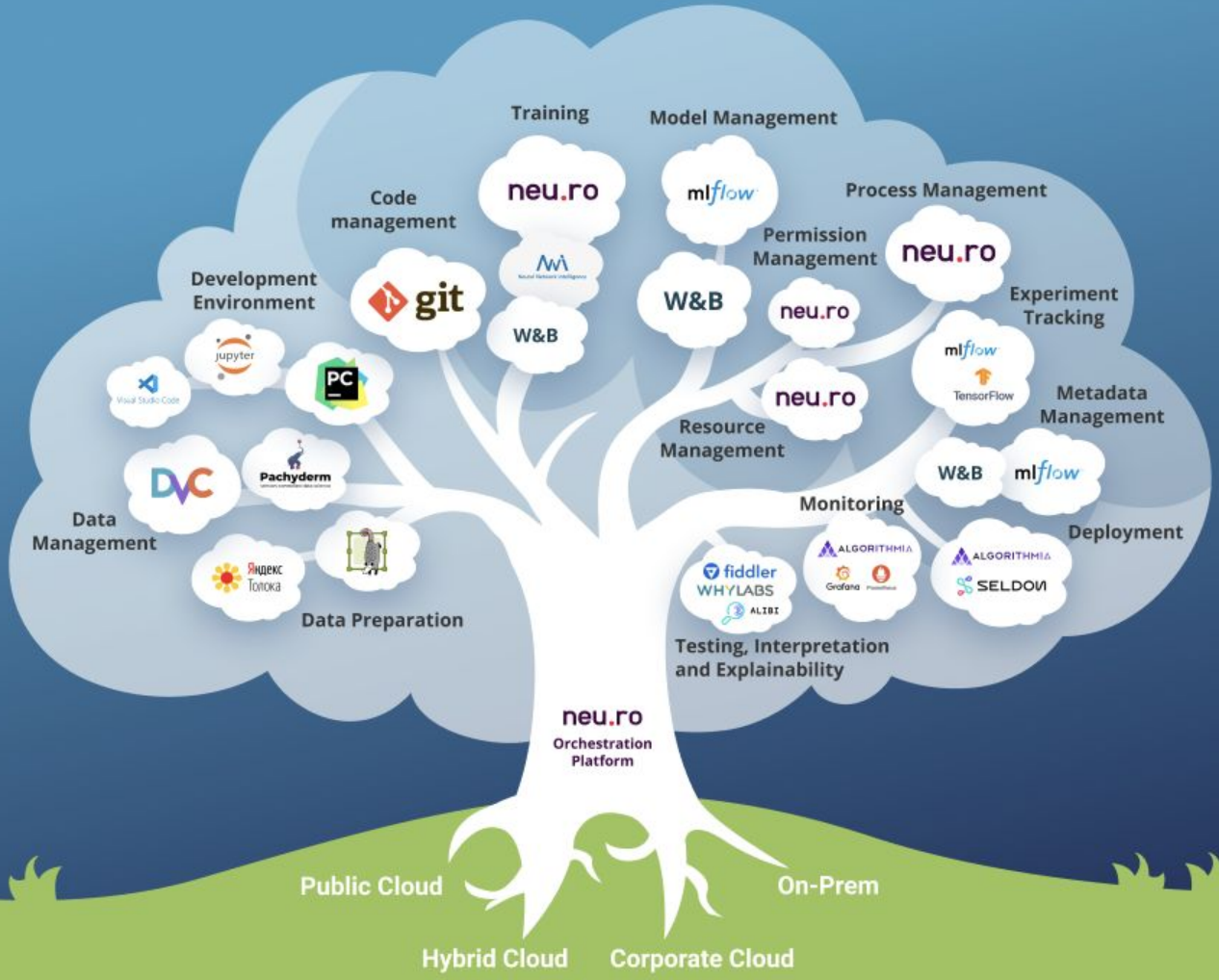
coroutine **run**(*container: Container, *, name: Optional[str] = None, tags: Sequence[str] = (), description: Optional[str] = None, is_preemptible: bool = False, schedule_timeout: Optional[float] = None, life_span: Optional[float] = None*) → JobDescription [source]

Start a new job.

Parameters:
- **container** (*Container*) – container description to start.
- **name** (*str*) – optional container name.
- **name** – optional job tags.
- **desciption** (*str*) – optional container description.
- **is_preemtible** (*bool*) – a flag that specifies is the job is *preemptible* or not, see Preemption for details.
- **schedule_timeout** (*float*) – minimal timeout to wait before reporting that job cannot be scheduled because the lack of computation cluster resources (memory, CPU/GPU etc).
- **life_span** (*float*) – job run-time limit in seconds. Pass *None* to disable.

Ингредиент 5. Инструменты

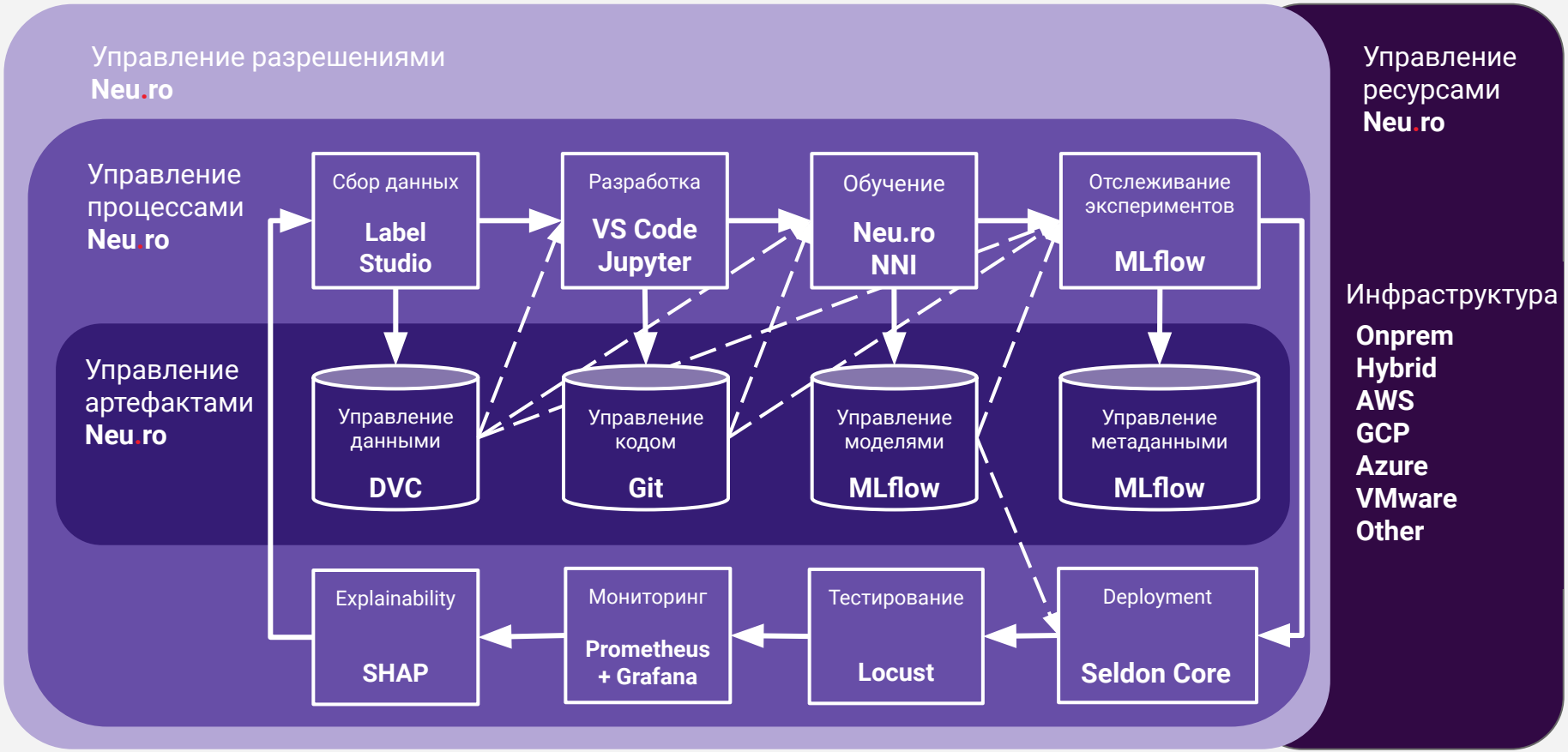
- **Библиотеки и фреймворки**
  - TensorFlow, Pytorch, etc
- **Пакеты (CLI)**
  - DVC, etc
- **Docker-native**
  - Jupyter, Label Studio, etc
- **K8s-native**
  - Pachyderm, Seldon, etc
- **SaaS**
  - Weights & Biases, etc
- **Desktop**
  - PyCharm, VS Code, etc


Training
Model Management
Code management
neu.ro
mlflow
Process Management
Permission Management
neu.ro
Development Environment
git
W&B
W&B
Experiment Tracking
neu.ro
mlflow
TensorFlow
Metadata Management
neu.ro
Resource Management
W&B
mlflow
Monitoring
Deployment
Pachyderm
Data Management
ALGORITHMIA
Grafana
ALGORITHMIA
SELDON
fiddler
WHYLABS
ALIBI
Data Preparation
Testing, Interpretation and Explainability
neu.ro
Orchestration Platform
Public Cloud
On-Prem
Hybrid Cloud
Corporate Cloud

Ингредиент 6. Интеграции

- **В веб-интерфейс**
  - Jupyter Notebooks, JupyterLab, FileBrowser, etc
- **В Neu.ro Flow actions**
  - MLflow, TensorBoard, etc
- **В документацию и рецепты**
  - См. https://docs.neu.ro и https://github.com/neuro-inc
- **Специализированные интеграции K8s-native инструментов**
  - mlflow-to-seldon - автоматизированный деплоймент моделей из хранилища в MLflow “на горячую” в Seldon

Пример стека: Open source



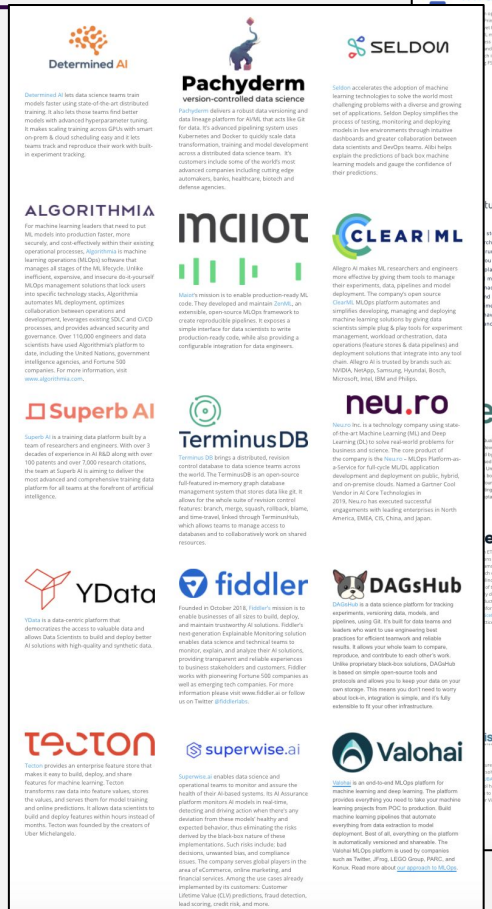
Трудности

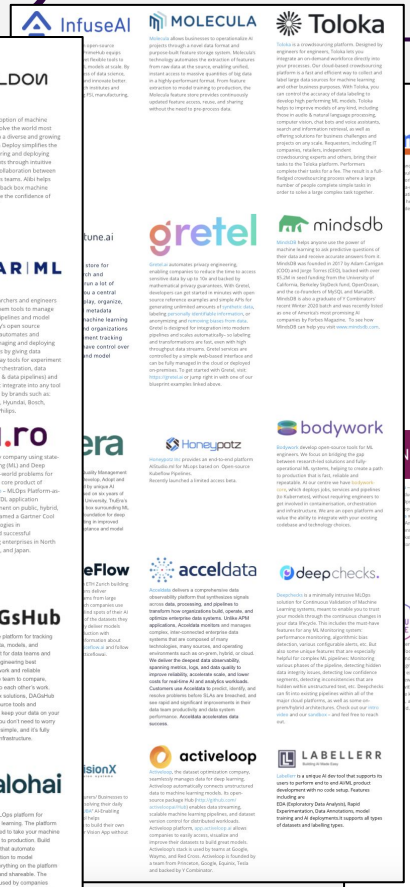
- Контроль ресурсов для K8s-native инструментов
- SSO и единое управление правами на всём стеке
- Мифические security & governance

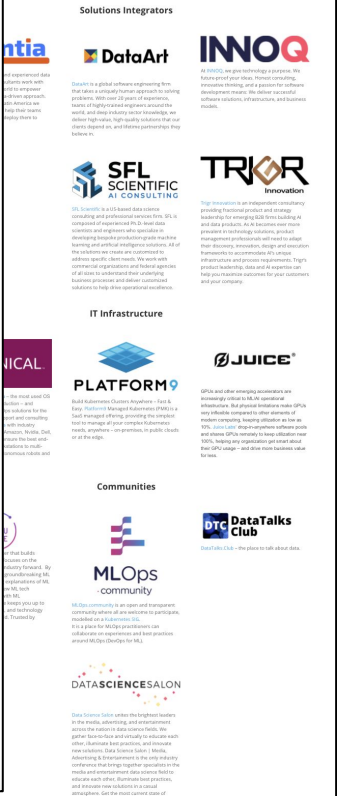
Об AIIA (AI Infrastructure Alliance)

Формирование стандартов
и канонического стека
в ML разработке

- Основан в начале 2020
- Активное развитие с конца 2020
- Более 60 компаний







v

Спасибо!

neu.ro