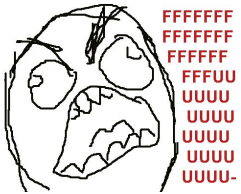
Taking the Reigns of the Cloud: A Human-Centric Approach to ML Experiments

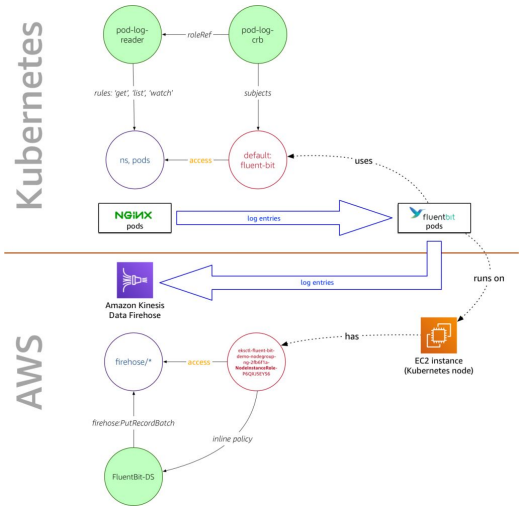
Artem Trofimov, JetBrains

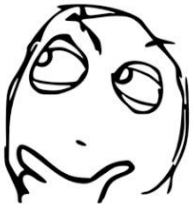
MLOps: It’s Not Rocket Science…

- Experiments may take days/weeks
- GPUs are expensive and it is hard to allocate them due to shortage
- Dev!=Prod (many computers)
- Complex unmanaged migration (data & env)
- A lot of data that is hard to transfer
- Reproducibility issues
- Inference is expensive or inefficient



ML Engineer





MLOps Engineer

Problem setup

- ML engineer often writes code using local laptop
- Transition from local experiment to remote runtime is non-trivial
  - A lot of data
  - GPU shortage
  - Huge variety of tools
  - …
  - VERY COMPLEX INTERFACES

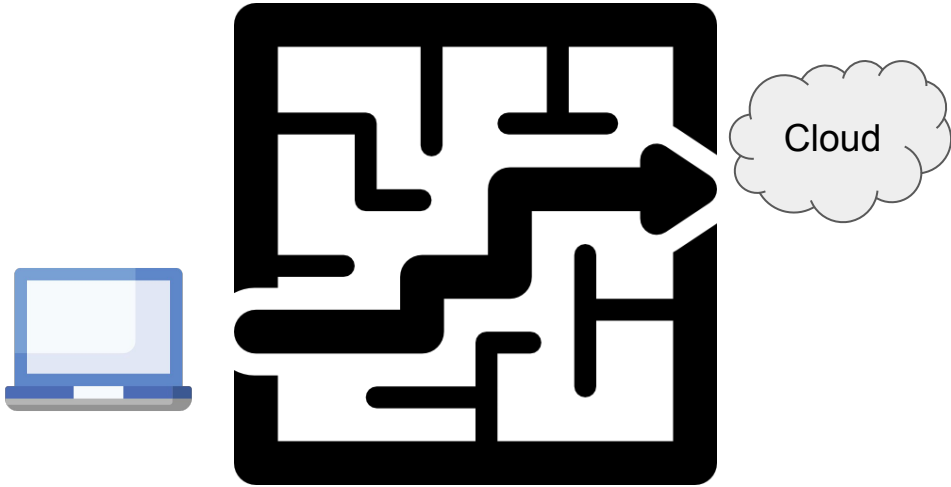
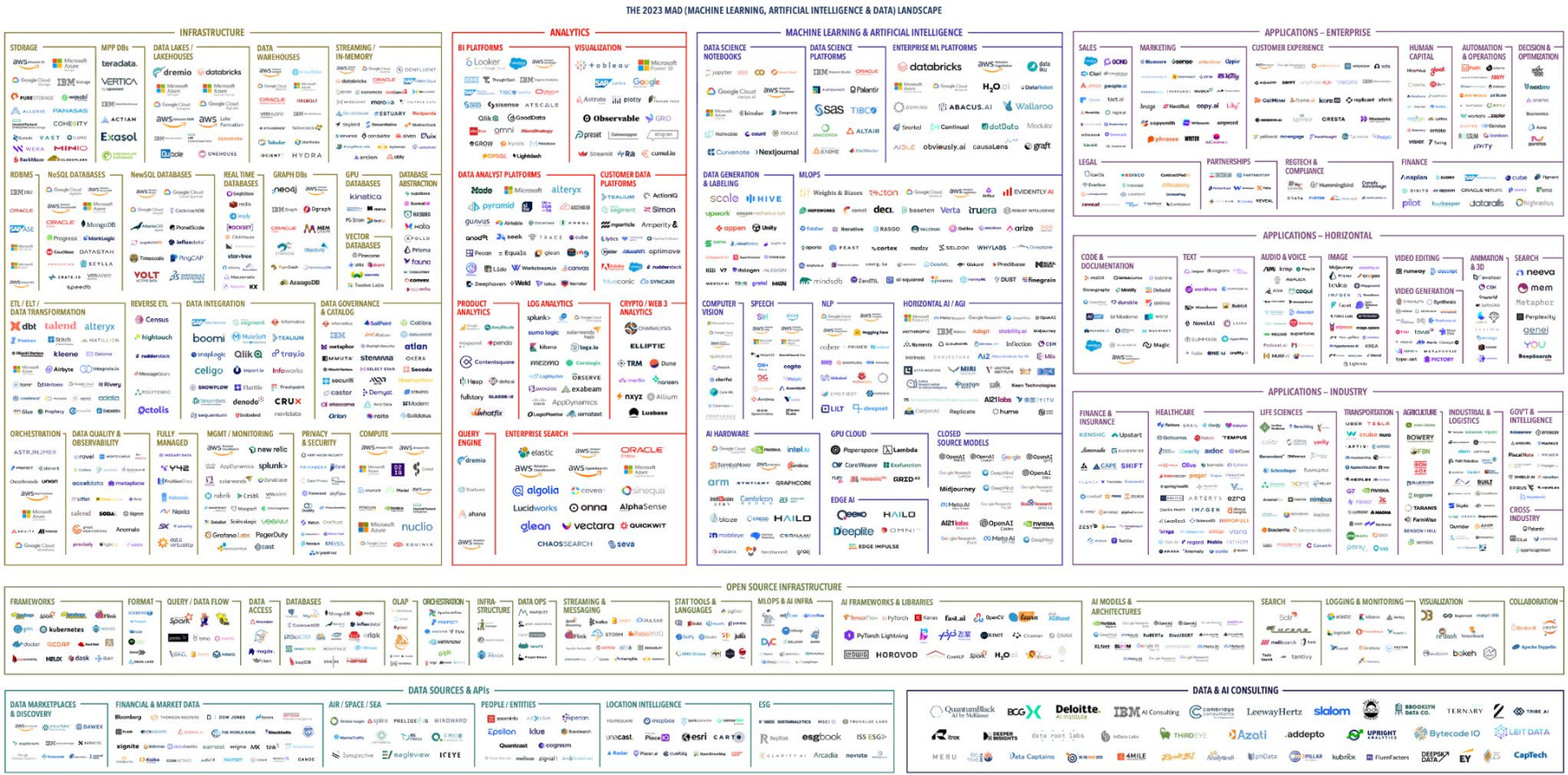


Image: Flaticon.com

Part 1
MLOps Toolkit: the Good the Bad, and the Complicated

https://i.redd.it/tjhnuyjx31181.jpg



Version 1.0 - Feb 2023 © Matt Turck (@mattturck), Kevin Zhang (@ykevinzhang) & FirstMark (@firstmarkcap) Blog post: mattturck.com/MAD2023 Interactive version: MAD.firstmarkcap.com Comments? Email MAD2023@firstmarkcap.com

FIRSTMARK EARLY STAGE VENTURE CAPITAL

https://mattturck.com/landscape/mad2023.pdf

VMs + SSH

- ✓ Full control
- ✓ Highly customizable

- − Boilerplate for env/data synchronization
- − Poor utilization due to manual lifecycle
- − Full control -> reproducibility issues
- − A lot of BASH

Image: Flaticon.com

Jupyter Notebooks

- ✓ Full control*
- ✓ Highly customizable*
- ✓ Integrated with IDE

*depends on a specific managed solution

- Poor utilization (remote dev on a GPU machine)
- Reproducibility issues due to black-box state

Image: Flaticon.com

Pipeline tools (Kubeflow, Ray, Flyte, Airflow, Prefect, etc.)

- ✓ Resources allocation on demand
- ✓ Structured stages
- ✓ Various resources for various stages*

*supported by some engines only

- Pretty high entry level
- Vendor lock on the code level
- No remote debug

Image: Flaticon.com

Task scheduling tools (MosaicML, SkyPilot, DStack, etc.)

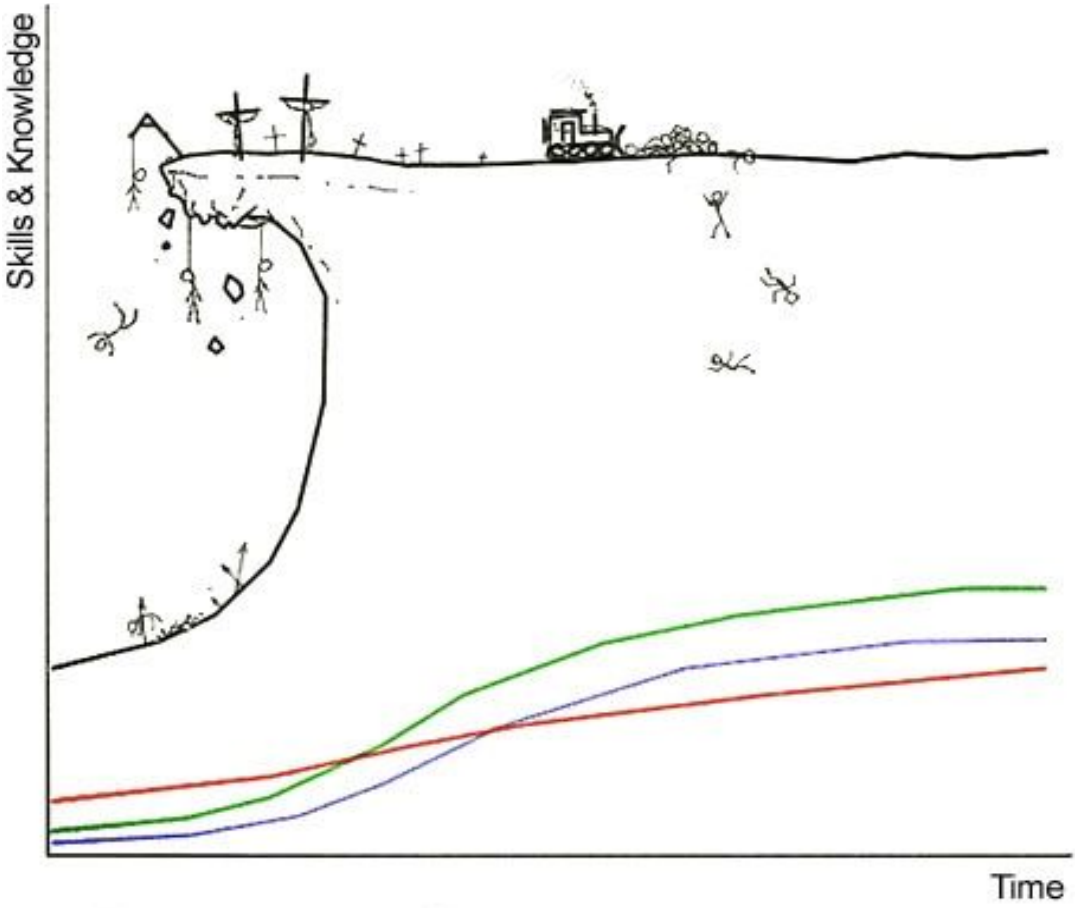
- ✓ Resources allocation on demand
- ✓ Simpler than pipeline tools

- − YAML configs
- − Cloud-specific settings
- − No remote debug

Image: Flaticon.com

Overview

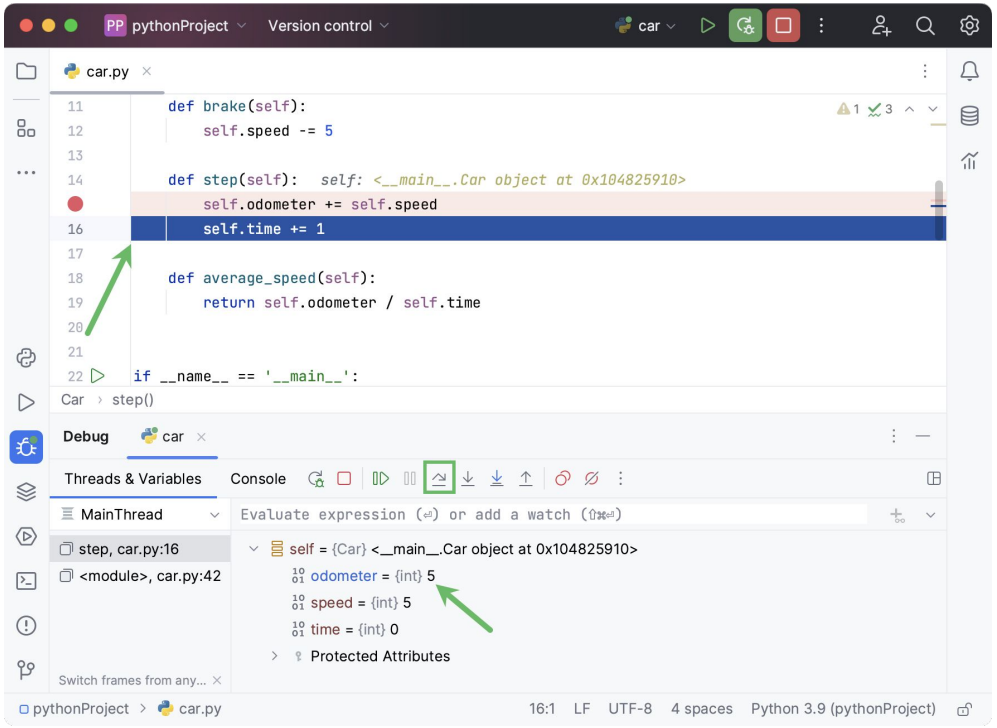
- Dev tools are simple and powerful, great for start but are not suitable for complex workflows
- Ops tools are ensuring reproducibility and scalability but have too complicated UX



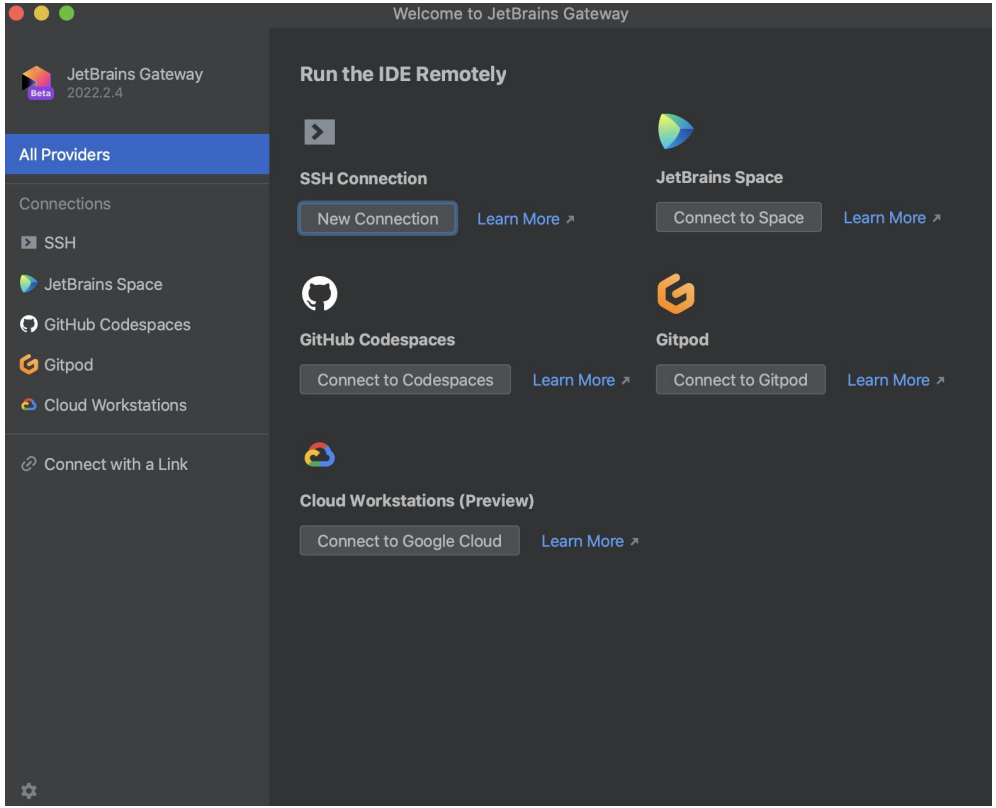
https://devpress.csdn.net/vue/62f0d74d7e668234661835e3.html

Part 2

Can ML folks stay within IDE? Or we have to use other tools?



IDE natively supports remote development!

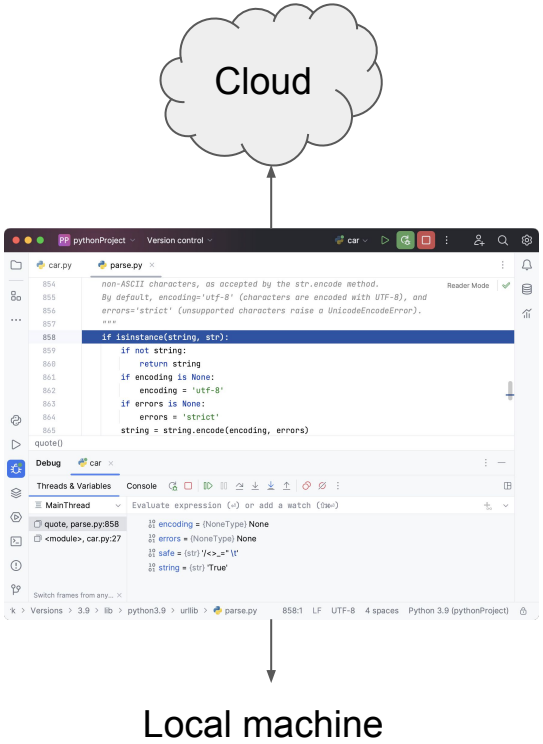


Does remote development solve the problem?

- Remote development is great for *development*
- It is very inefficient to write code using VM with 8xA100 GPUs
- We need a tool at the intersection of development and MLOps

Can we delegate *some* runs to a remote VM?

- User should indicate:
  - Hardware requirements
  - Overridden parameters
  - Data to mount
- System should start VM, transfer env/data to it and run the task

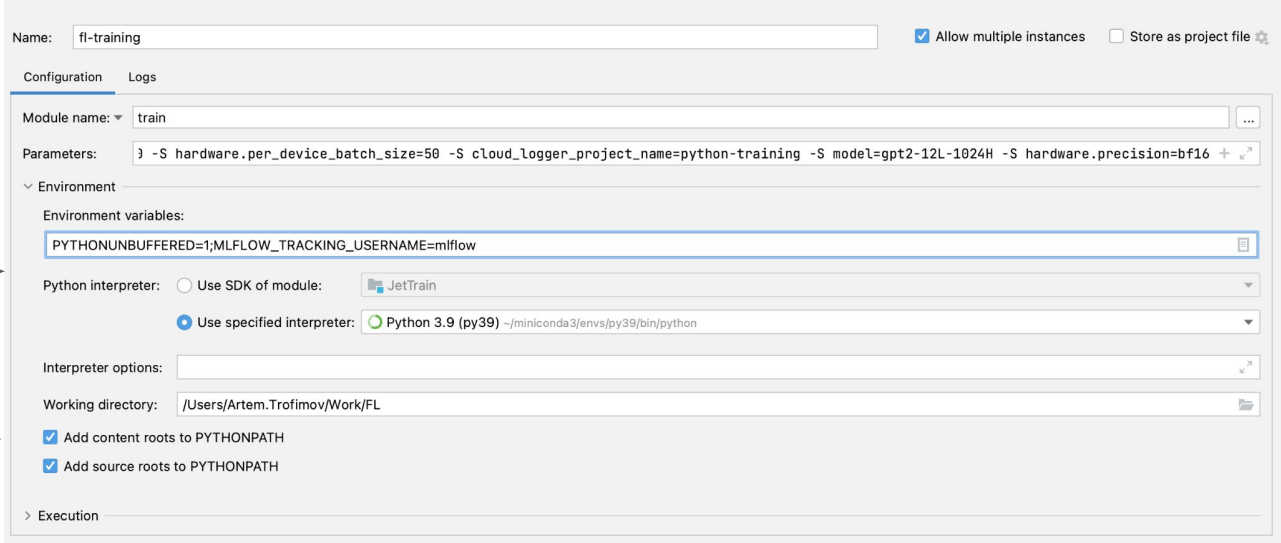


IDE has enough meta-information for remote run!

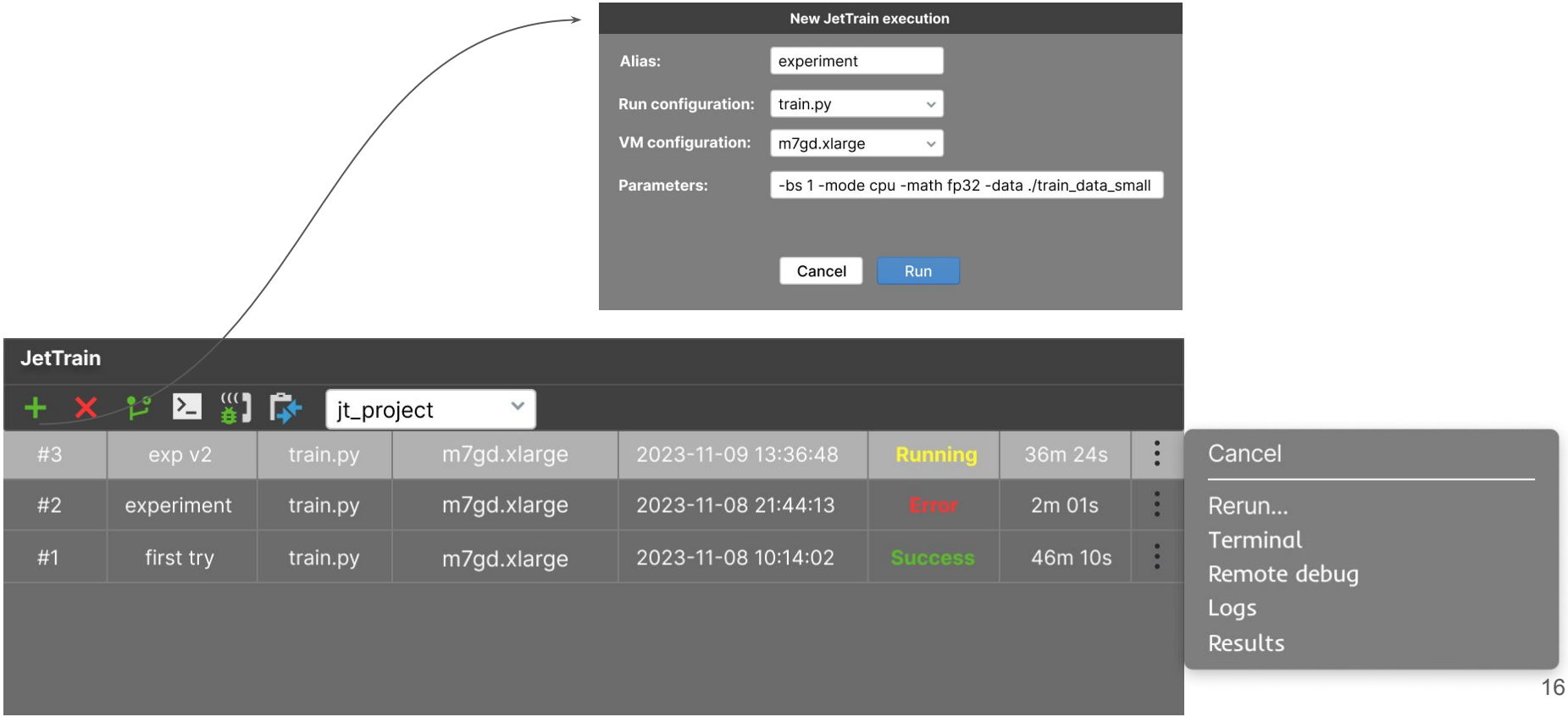
CMD to run

Env to setup/install

Data/code to sync



Running ML experiments from IDE: UX



IDE-native ML experiments

- ✓ UX similar to local runs
- ✓ No context switching
- ✓ On-demand resources

- − No pipelines
- − User should be familiar with IDE

Image: Flaticon.com

Open questions

- Tasks scheduling (under GPU shortage)
- Env synchronization
- Data synchronization

Part 3 Confronting The Obstacles: Which Technical Challenges We Have?

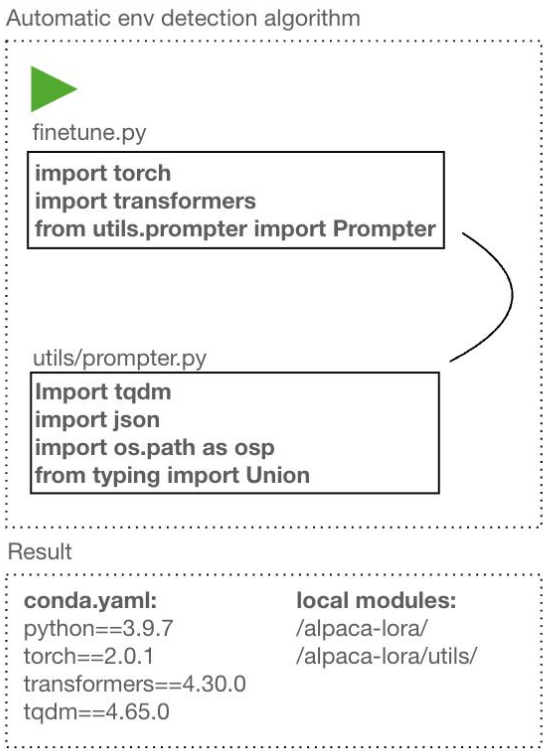
https://inspirationalperspective.com/2014/09/14/obstacles-vs-barriers/

Tasks scheduling, user management, billing, etc.

- These problems are really hard but fortunately we have TeamCity!
- We can delegate cloud integration, scheduling, logs delivery, etc., to TeamCity
- TeamCity brute forces available cloud providers and their zones to find VMs quickly!

Mirror, mirror on the wall, sync the best environment of all

- It is easy if user has requirements.txt/conda.yaml/etc.
- To do it automatically we can recursively analyze modules of global variables starting from main module
- This algorithm is precise but does not detect imports inside functions

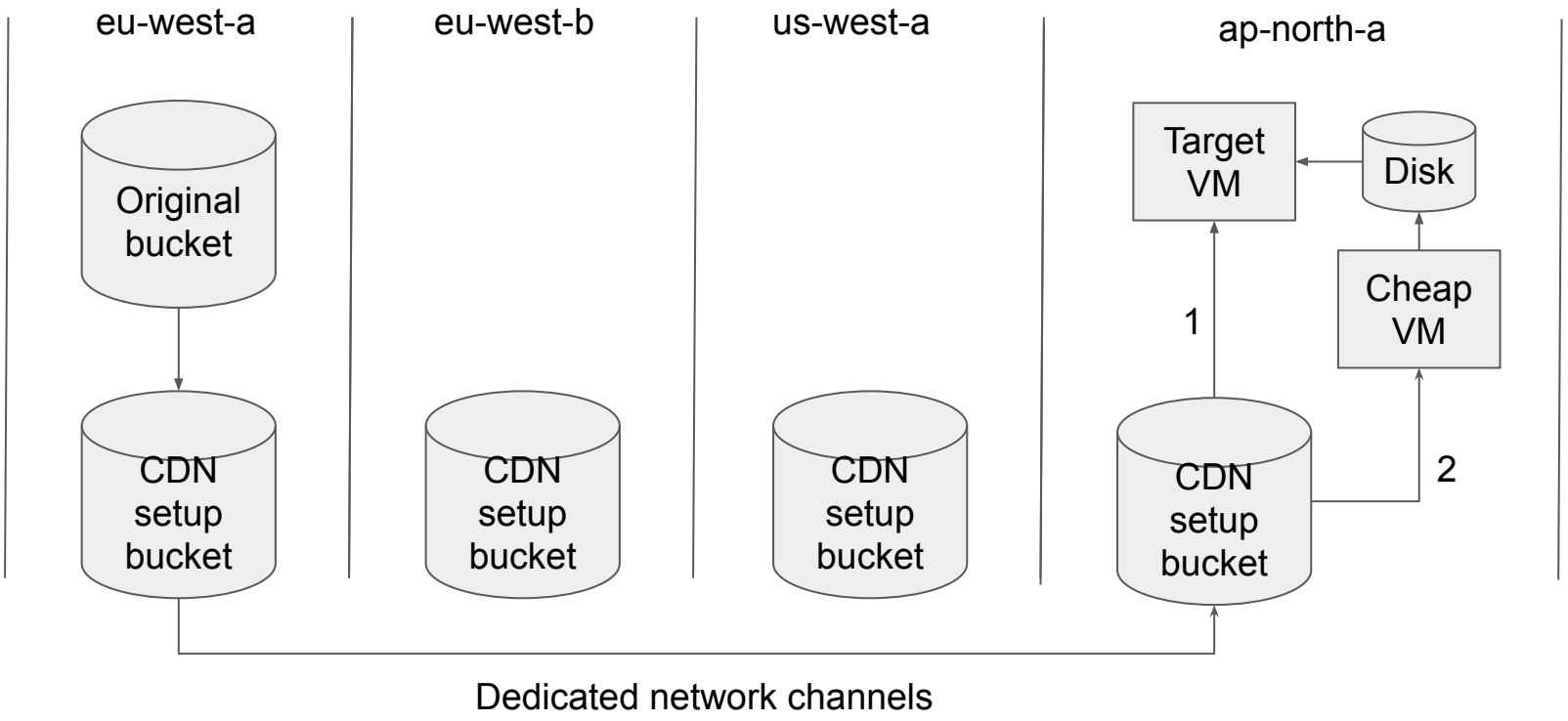


Data synchronization: a naive way

- aws s3 sync/cp
- If data and VM are in different data centers, data transfer can be slow and expensive

Image: Flaticon.com

Optimized way: data dances amid the great GPU drought



A Peek at Our Current MLOps Expedition

Generated by DALL-E

- First adopters: 5 internal ML teams
- 500+ hours/week VMs consumption
- Stable for long trainings (tested on 8+ days trainings)
- Private preview for external clients in Q1 2024

Our current status

- First adopters: 5 internal ML teams
- 500+ hours/week VMs consumption
- Stable for long trainings (tested on 8+ days trainings)
- Private preview for external clients in Q1 2024

IDE-native ML experiments? Let’s discuss!

feel free to contact me regarding any questions/ideas:
artem.trofimov@jetbrains.com
telegram: @tyooma