NVIDIA GTC 2024

10 Years of building GPU cloud

Lessons and observations from building a cloud infrastructure company at the emergence of the ML / AI revolution.

Contents

Background

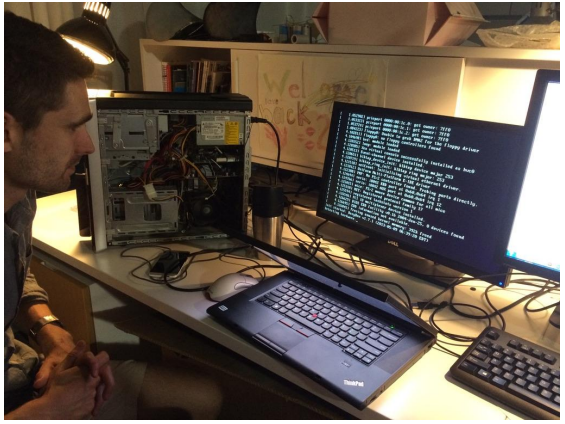
What the heck is a GPU?

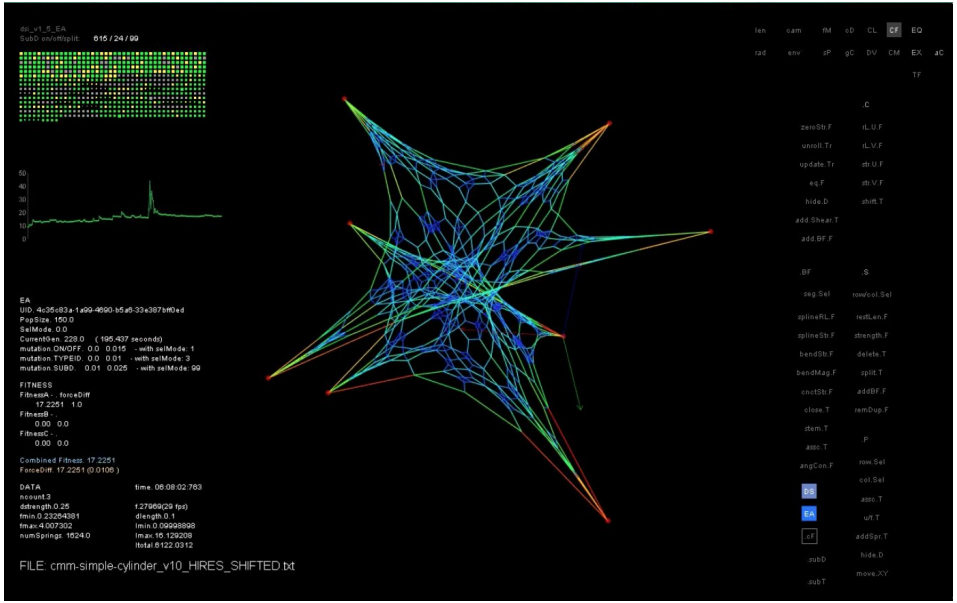
Paperspace is an architecture term that means *“2D viewport into your 3D workspace.”* The company started around an interest in the power of GPUs which we had encountered while working on software for structural simulation. AWS was still relatively new and no cloud provider specialized in GPUs exclusively.

We were admitted to YCombinator with a vision make GPUs accessible to anyone by building a new type of cloud.

Right: our first hypervisor tests in 2014.

Below: **SpringForm**, graduate research project on simulating high-performance tensile structures.





DigitalOcean

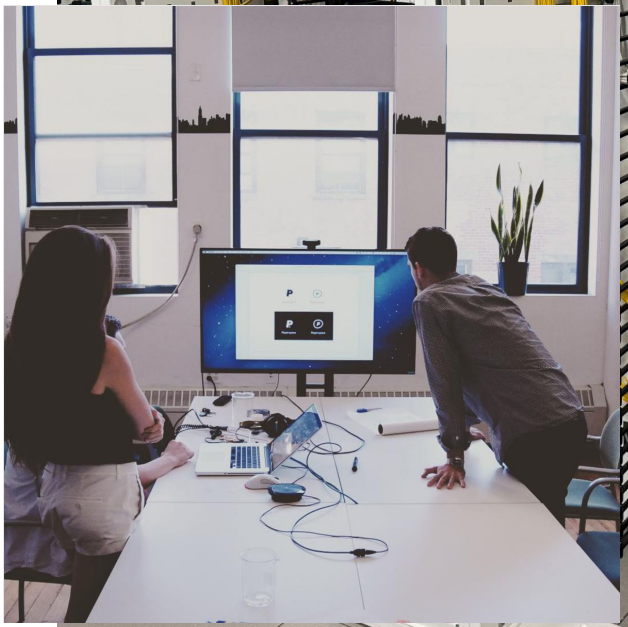
Background

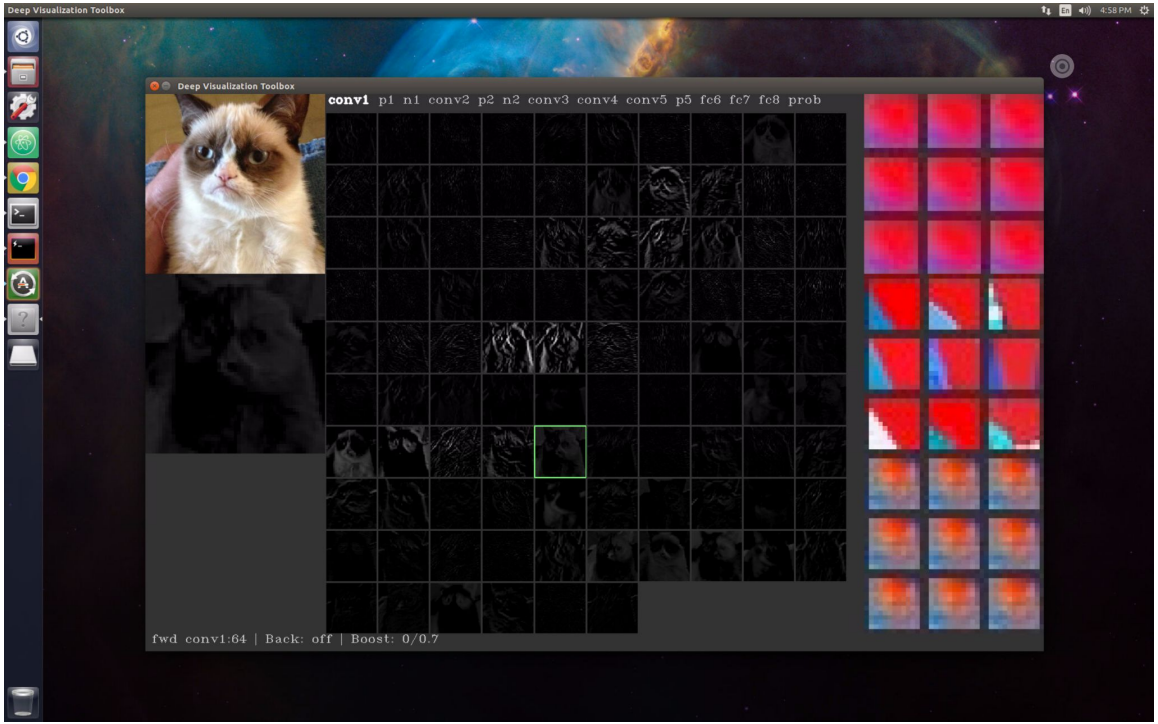
Building a GPU-native cloud

The first GPUs we brought online in 2015 were Kepler architecture NVIDIA GPUs with only 1536 cuda cores. By the time we sold the company in 2023 (9 years later!) we had built and launched systems with every generation of NVIDIA card – from Kepler to Maxwell to Ampere and now the Hopper H100!

The normal calculations around storage performance, network bandwidth, and power consumption had to be revisited with this new type of compute.

And as we brought these new systems online it was clear that the software that powered them was equally important.





Our bare-metal DaaS product, Machine Learning in–a-box (MLiaB 1.0), running a CNN visualization for the Fast.AI cours, *2017*

2014 - 2021: Developing cloud tools for ML

What made GPU cloud any different for building software?

We quickly started to sketch out the emerging developer stack. The NVIDIA GPU was such a powerful new tool but the ecosystem around it was still very much developing.

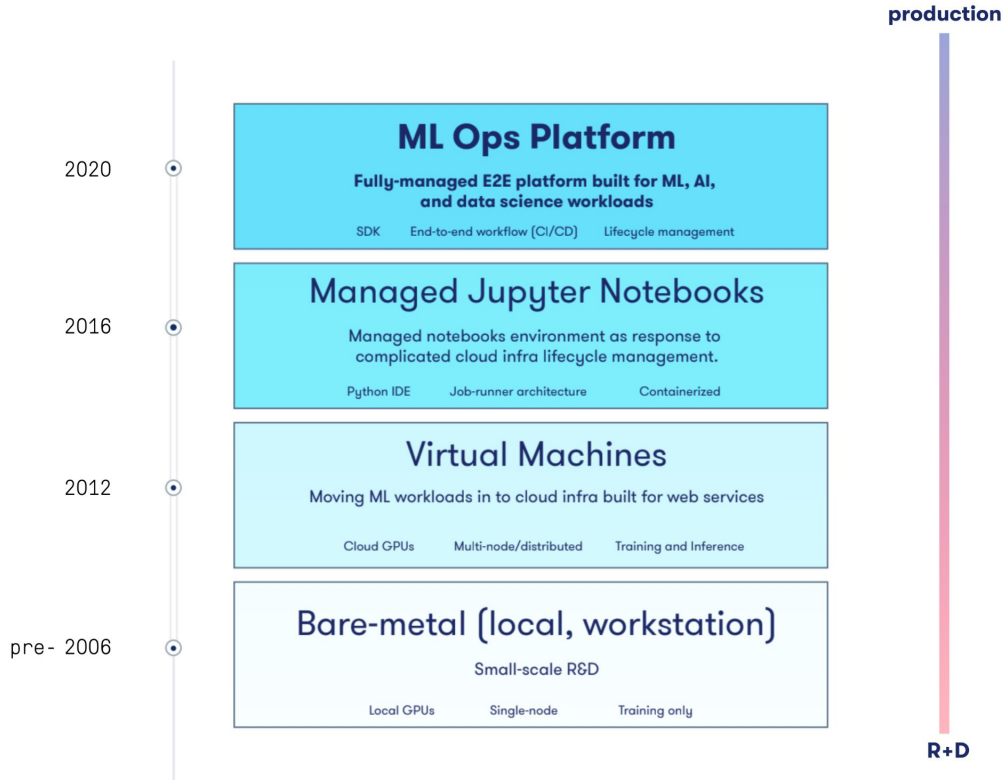
We observed early users of Torch and MXNet and the Fast.ai course to mine ideas on what these new tools would look like and what the new GPU-native “stack” would look like.

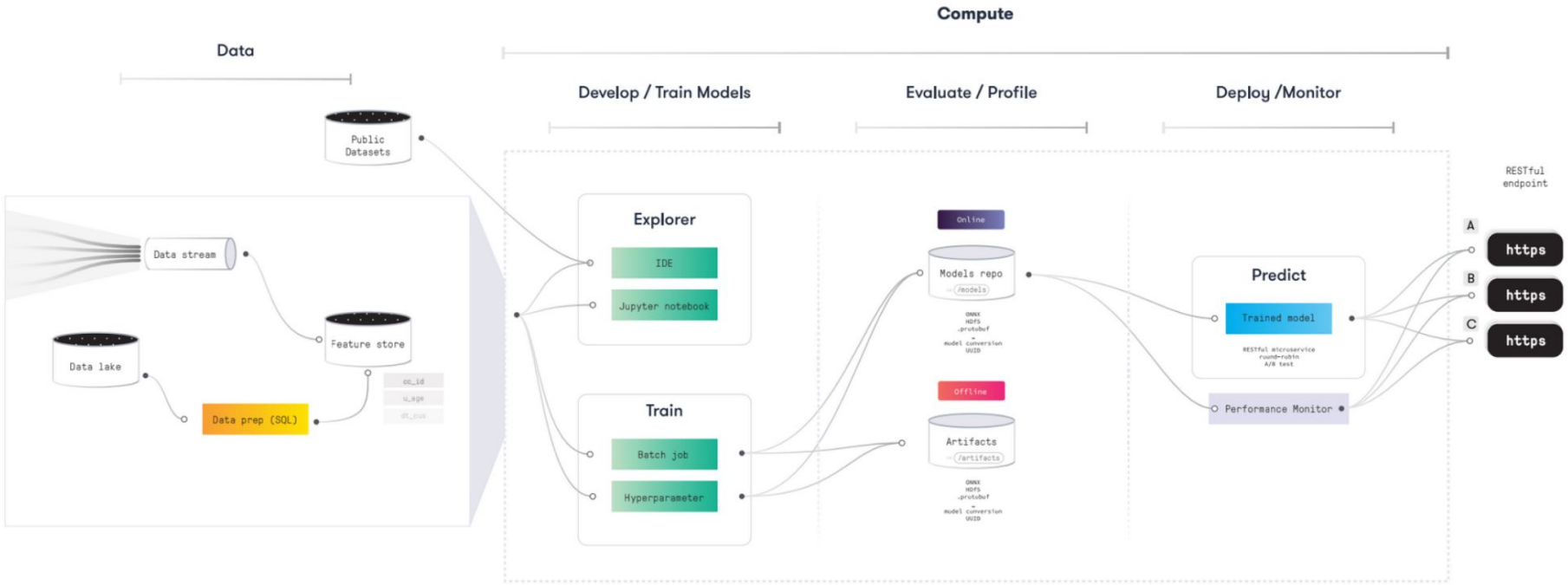
Many best practices had yet to emerge (NVIDIA-Docker was brand new) and it was clear that the rapidly emerging “deep learning” architecture would require a new software ecosystem and would establish new developer experiences.

MLOps: Emerging patterns of GPU cloud workflows

Many companies began working on what would become MLOps, the beginning of a formalized software development practice that took Machine Learning (Deep Learning) models and helped them integrate with the rest of the production software development ecosystem.

Ideas like CI/CD for ML and the need to create deterministic and reproducible artifacts became primary considerations.





A sketch of the early “MLOps” pipeline *2018*

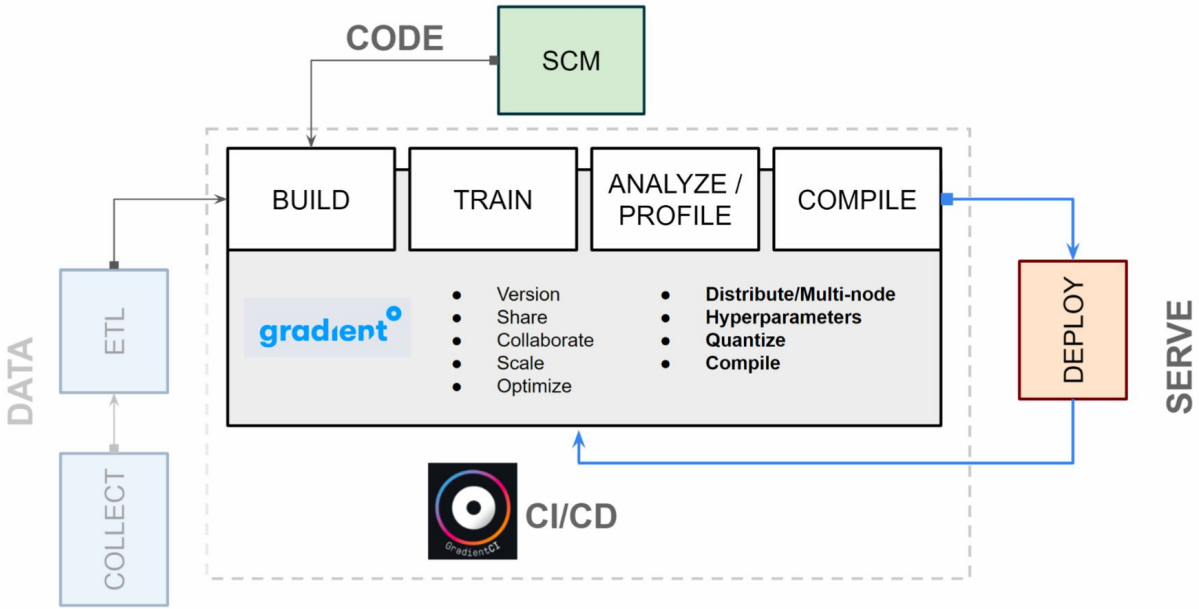
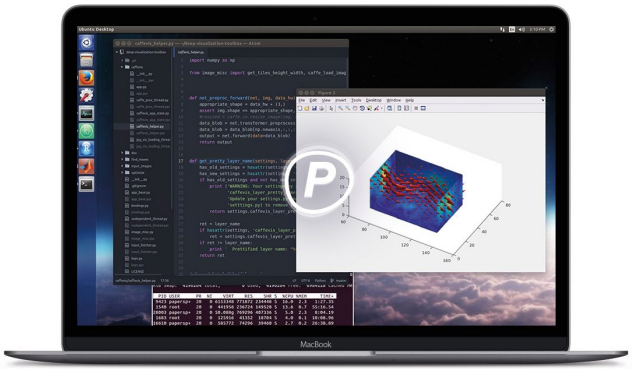


Diagram of the “inner-loop” of model development and deployment, *2020*

What are the fundamental building blocks of this new stack?

- **Fully-managed infrastructure**
- Unified dev experience
- Cluster management
- 1 click jupyter notebooks
- Language integrations / bindings
- ACL/team controls
- Data-integrity
- Versioning/reproducibility
- Team management / permissions
- Python CLI and SDK
- Native container support
- Hyperparameters sweeps

- Metrics collection
- Autoscaling
- Full lifecycle management
- gRPC & MPI
- Inference load-balancing
- Tag management
- TensorBoard integration
- Dataset tracking
- Persistent storage (POSIX & buckets)
- …



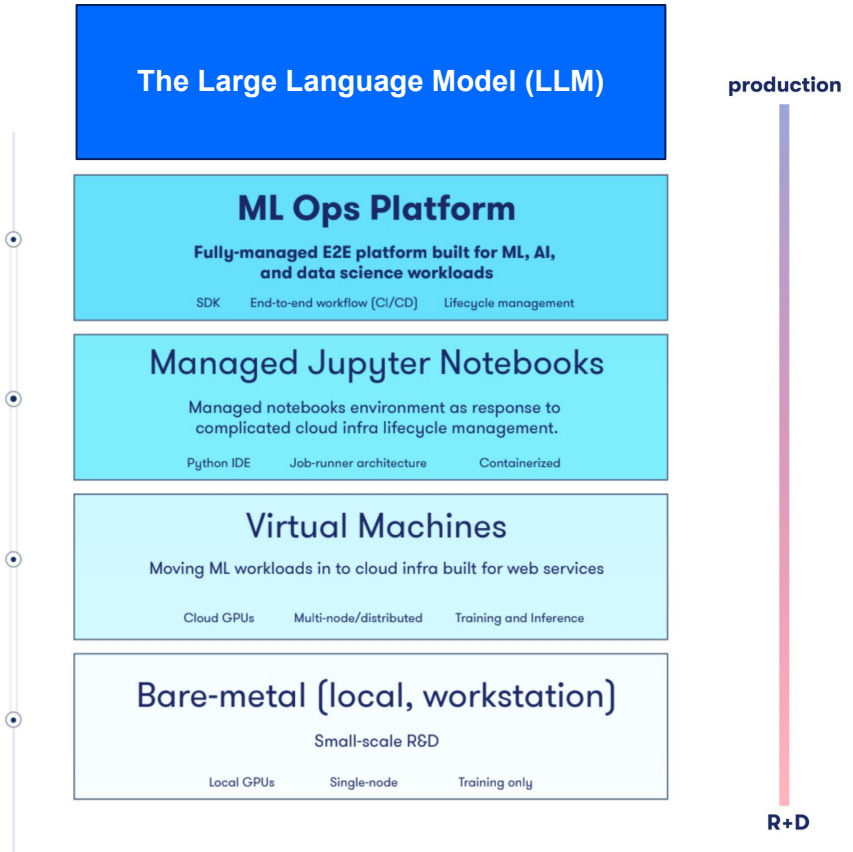
2022 - 2023

LLMs as phase shift

The emergence of Large Language Models like GPT signaled a major leap in AI capabilities. These models didn't just enhance existing applications; they opened doors to new possibilities, reshaping the landscape of AI development and its potential.

The scientists had built something that the engineers now could run with. AI was having an “API” moment and this time the entire development process was upended.

- How does one deal with non-deterministic APIs?
- What types of programming languages should we use to talk to this new intelligence?
- BDD, TDD, to AIDD, what doesn’t change in this new reality?

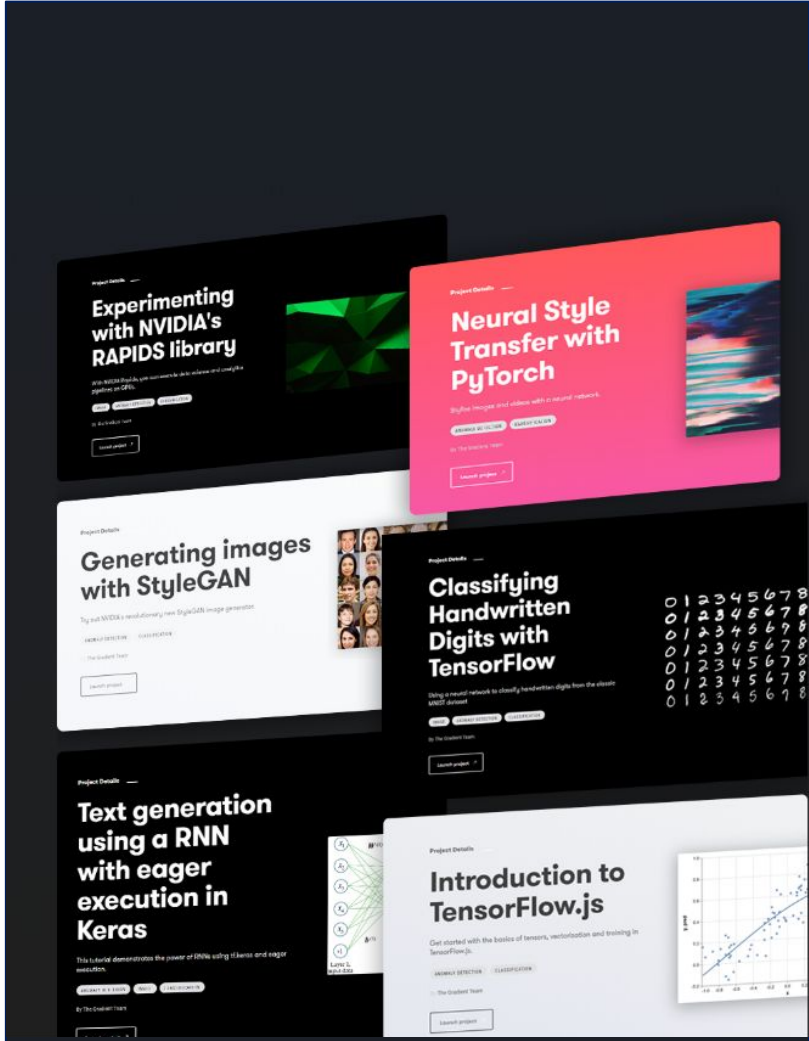


2024 & Beyond: Design patterns for the future

Intelligence as a new primitive

Looking forward, intelligence itself (through the core building blocks of LLMs) is becoming a core component in software development – as fundamental as databases and UI frameworks once were.

This integration will fundamentally reshape how software is built and what types of software we can imagine.



DigitalOcean

2024 & Beyond: Design patterns for the future

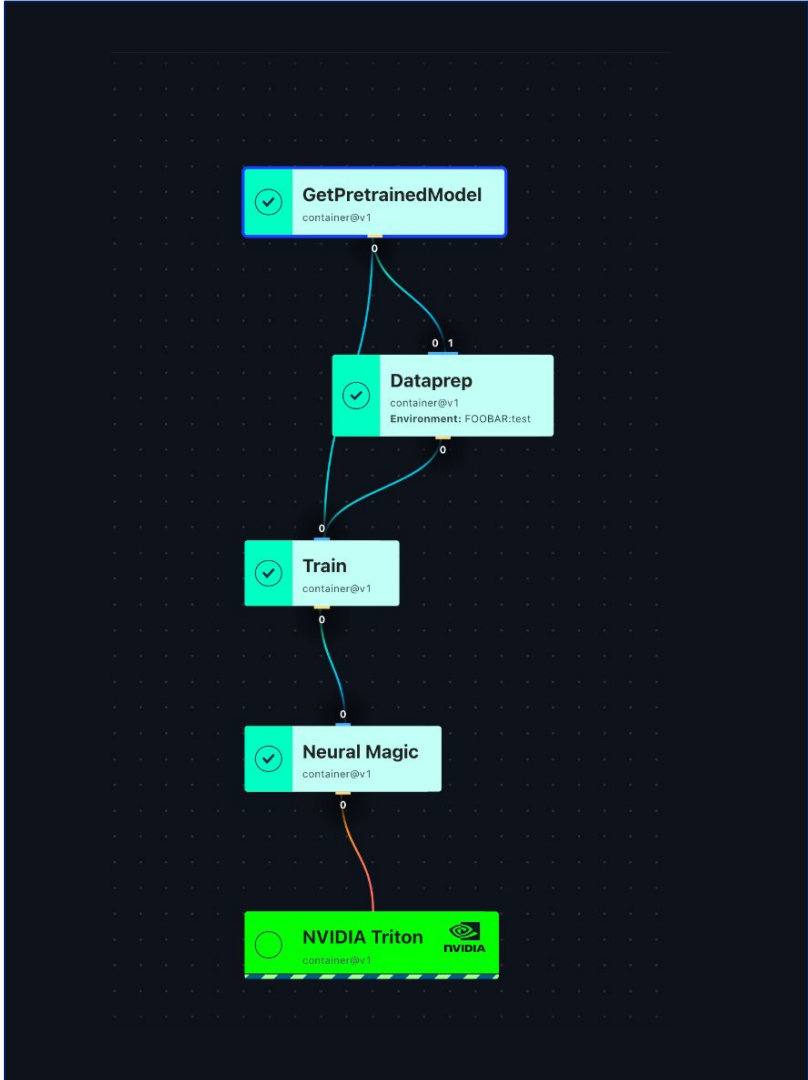
Data-First Development

In 10 years of building cloud GPU systems, the compute has always been the obvious differentiator. But data has emerged as possibly the most valuable component.

Data has become as crucial as code. In the future, we will see a 'data-first' approach to understanding how software projects are built, where the availability, quality, and utility of data will dictate development strategies, driving more personalized and effective AI-first projects.

As we build the next generation of our compute systems at DigitalOcean we are spending as much time studying data as we are compute.

DigitalOcean



2024 & Beyond: Design patterns for the future

AI-Driven Development

In the software world there is a persona known as the “10x engineer” – a person so talented and knowledgeable that they can product as much as 10 other engineers on any given team.

Seemingly overnight this capability has come within reach for anyone who writes and deploys code.

The role of the developer itself is evolving. With AI-assisted tools, we're entering an era where a developer's ability to collaborate with AI will be as important as their coding skills, leading to unprecedented levels of productivity and creativity.

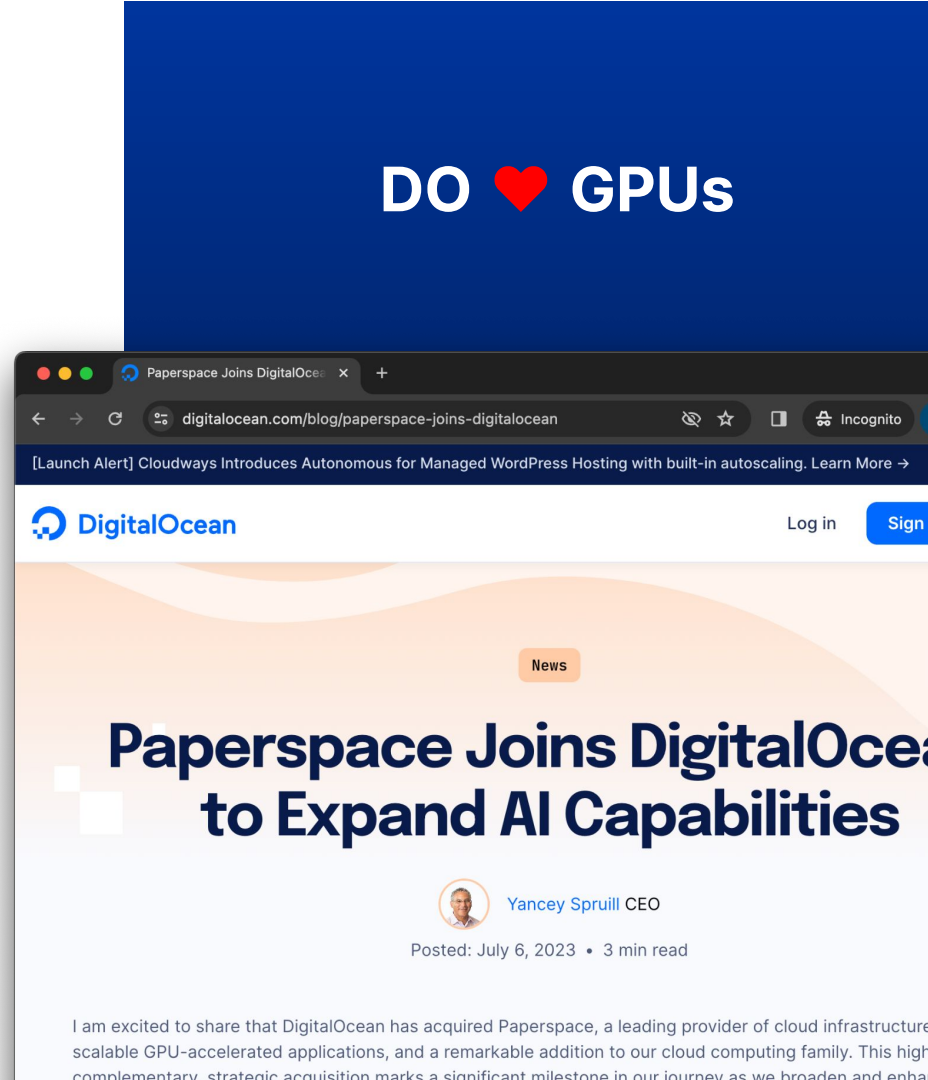
2024 & Beyond: Design patterns for the future

What's next

10 years have gone by in the blink of an eye. Being part of a rapidly changing industry is exhilarating and the LLM revolution is the next (and most exciting) shift yet.

We recently joined forces with DigitalOcean where we will be able to expand our cloud to a much larger scale than ever before and to help a new generation of software developers build out this future of intelligence.

DigitalOcean



Thank you