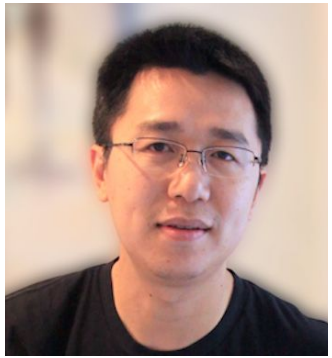
KubeCon | CloudNativeCon

North America 2024

Parasail

Tackling GPU Shortages and High Costs by Harnessing Hybrid Kubernetes Clusters

Xiaoman Dong, Alex Pucher



Xiaoman Dong
Founding Engineer

@xiaoman
@dongxiaoman

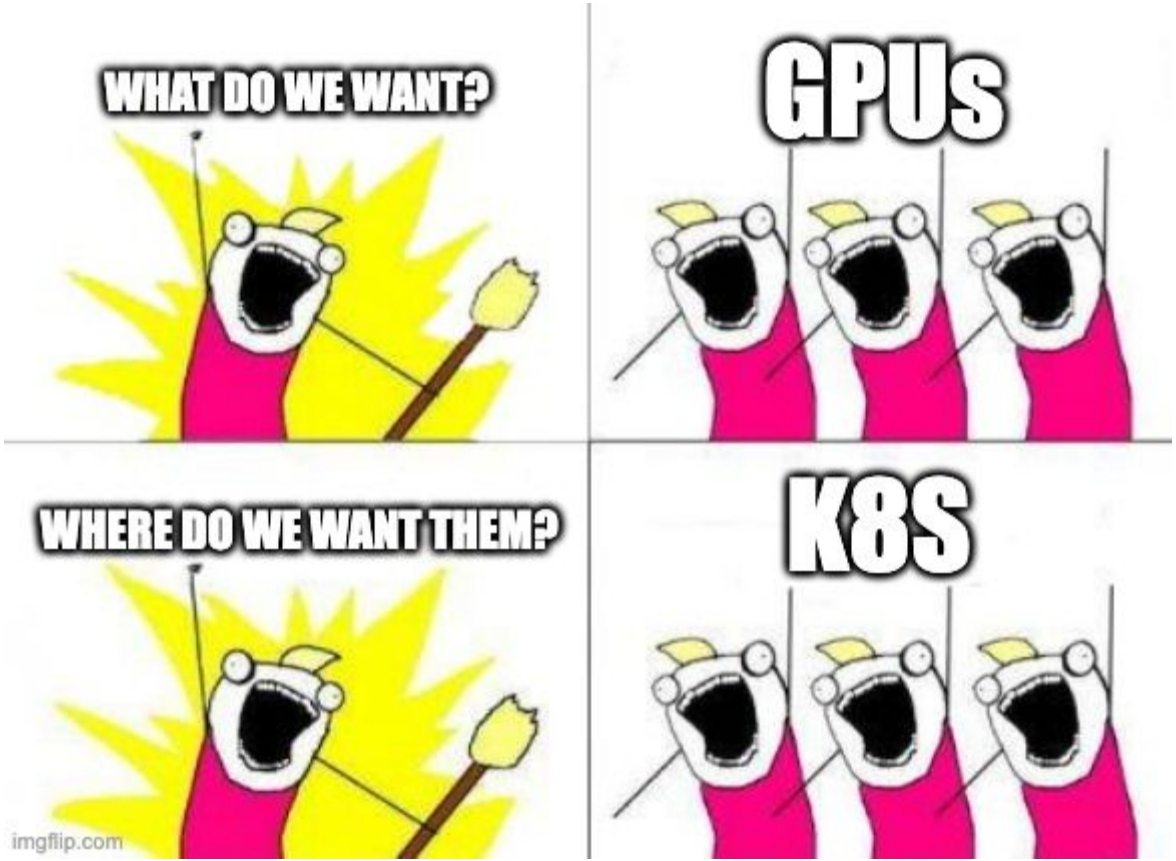
Alexander Pucher
Founding Engineer

@alexpucher
@apucher

^ QR for talk

Parasail
Simple, Scalable, AI Compute

Clouds in the era of Gen AI

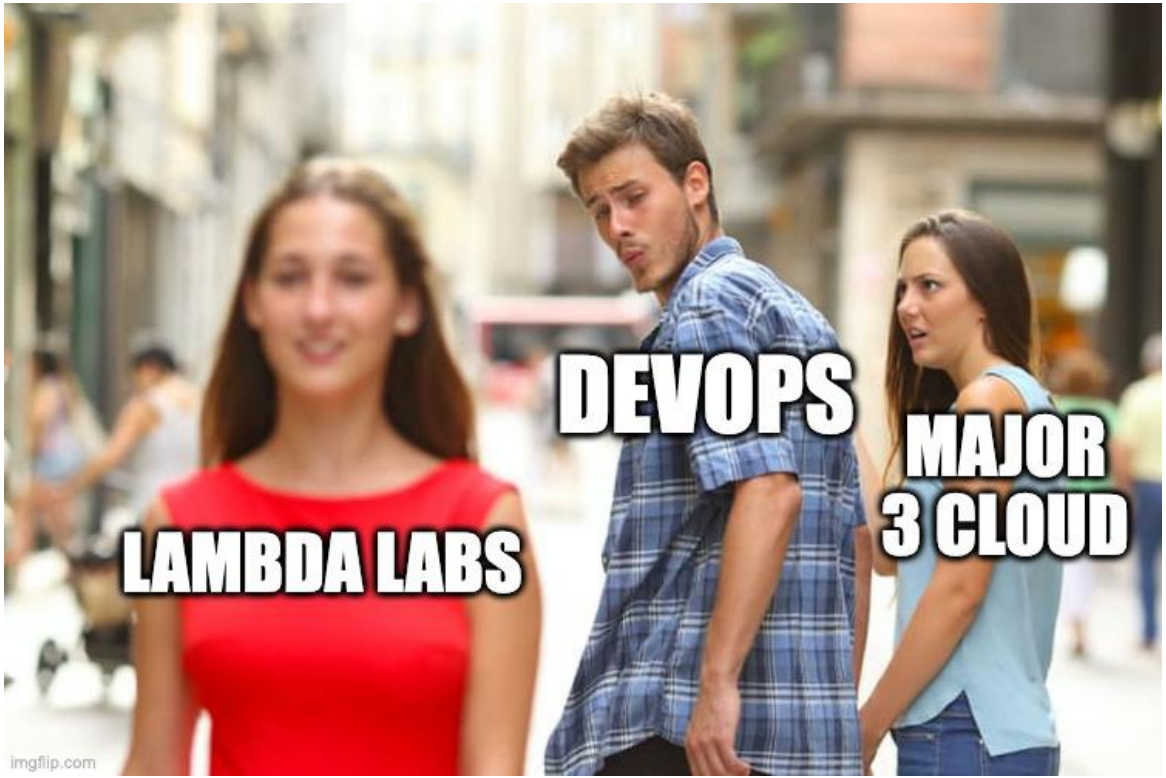


Relationship Advice

“Just use AWS.”

- anonymous, 2023

Relationship Reality



Using GPUs from Lambda & Co

"It has port forwarding, but ..."

Networking

TLS, NAT, egress and ingress, vpc hole punching

Operations

Bare metal "on an island" experience

Quotas still exists

Repeat work, multiple clusters, federation

One cluster to rule them all

We want:

- Single kubernetes cluster across multiple GPU clouds
- Freedom from quotas and lock-in with any provider

How to enable it:

- Secure node-to-node connectivity
- Resilience to GPU node failures
- Distribute inference workloads across providers

AI workloads tolerate some latency

Chat-Style: Synchronous

Availability is critical. Some tolerance of latency. Seconds.
Unit of work fits single node. Limited bandwidth requirements.

Batch: Asynchronous

Cost is critical. Latency doesn't matter. Minutes to hours.
Unit of work fits single node. Potentially high bandwidth requirements.

Training

Fine-tuning jobs, not base-model training
Out of scope. Single-node fine-tuning is doable.

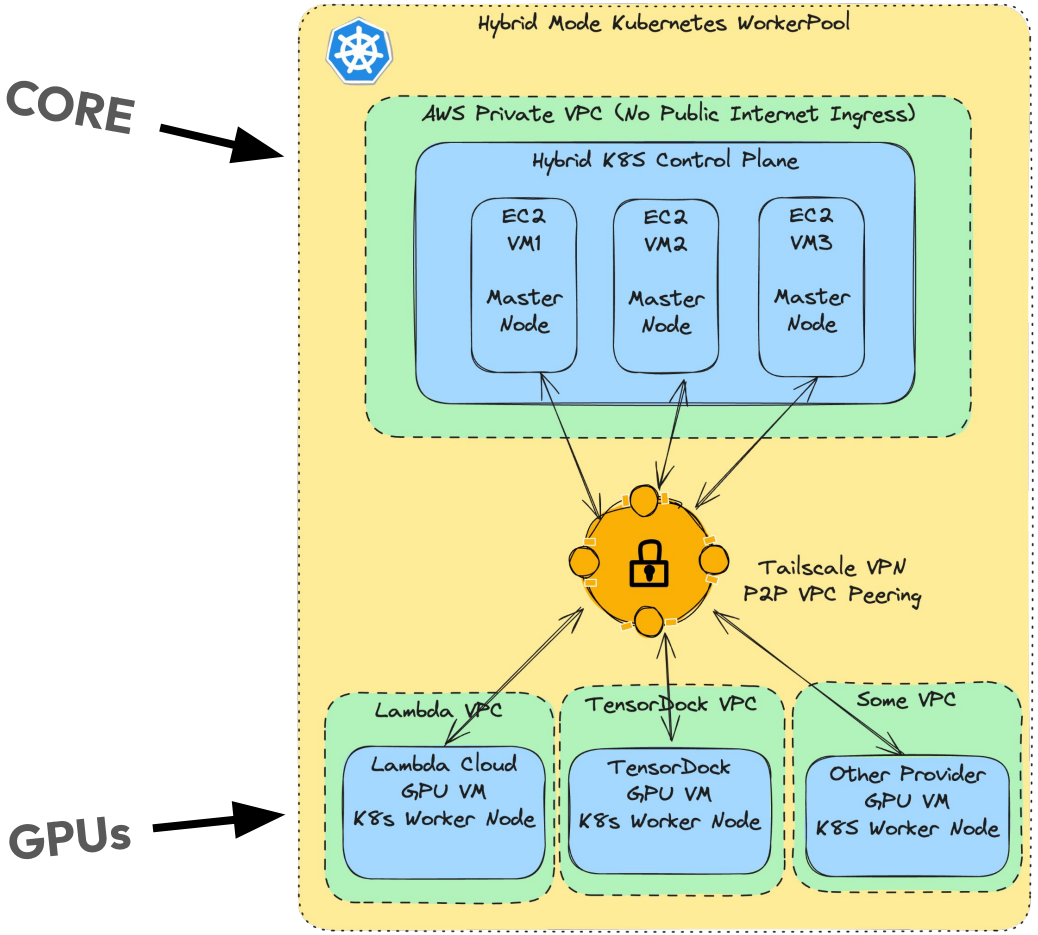
Run K3s + Tailscale (WireGuard)

Freedom from lock-in

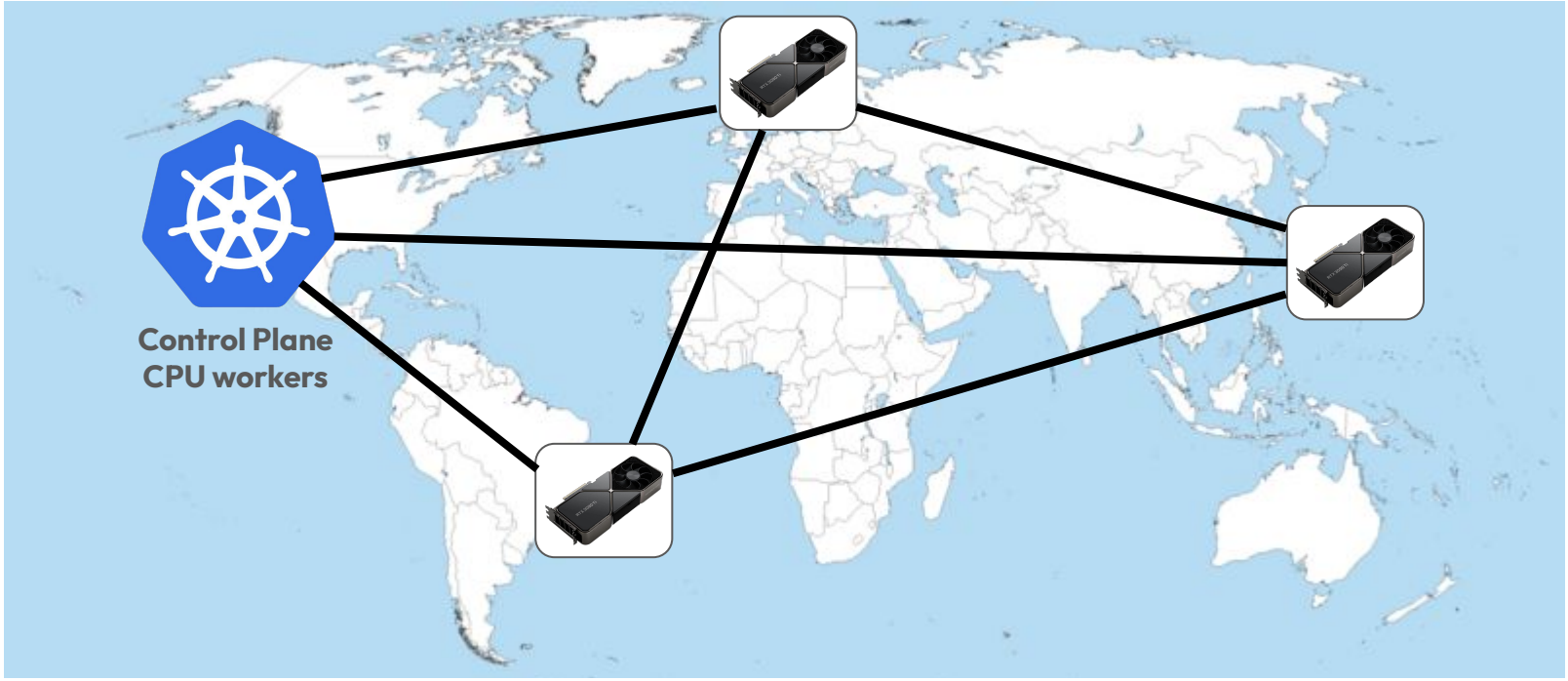
Better selection

Better pricing

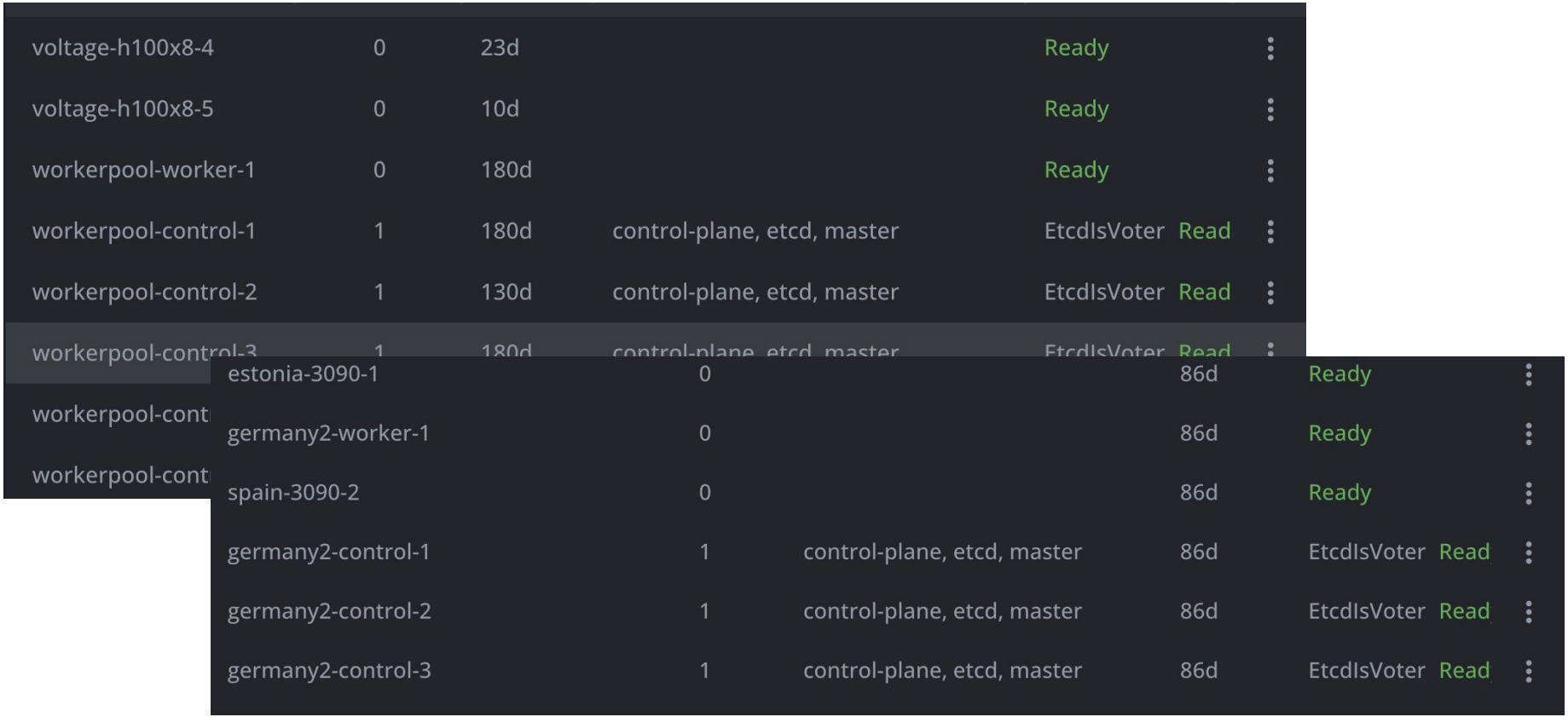
Kubernetes!



A real world example



(live demo of clusters)



voltage-h100x8-4	0	23d		Ready	
voltage-h100x8-5	0	10d		Ready	
workerpool-worker-1	0	180d		Ready	
workerpool-control-1	1	180d	control-plane, etcd, master	EtcdIsVoter Read	
workerpool-control-2	1	130d	control-plane, etcd, master	EtcdIsVoter Read	
workerpool-control-3	1	180d	control-plane, etcd, master	EtcdIsVoter Read	
workerpool-cont					
workerpool-cont					
estonia-3090-1	0			86d	Ready
germany2-worker-1	0			86d	Ready
spain-3090-2	0			86d	Ready
germany2-control-1	1	control-plane, etcd, master		86d	EtcdIsVoter Read
germany2-control-2	1	control-plane, etcd, master		86d	EtcdIsVoter Read
germany2-control-3	1	control-plane, etcd, master		86d	EtcdIsVoter Read

Our Journey

* We aren’t networking experts. Just DevOps struggling.

Integrating GPUs into k8s – Can we delegate?

“The best way to run Kubernetes is to have someone else run it.”

“How about adding external GPU nodes into EKS, AKS, GKE?”

Azure ARC and AWS EKS Anywhere? Lock-In and we’ll end up running k8s ourselves anyways

Integrating GPUs into k8s – Creating Our Cluster

Create Kubernetes the hard way

Etcd, Control Plane
VxLAN, Bridge, NAT

Steep learning curve

Connectivity and networking
Security and TLS everywhere

Rancher K3s got us started

Limits to network customization
Teams running K8s are heros

Integrating GPUs into k8s: over the Public Internet

IPsec/OpenVPN

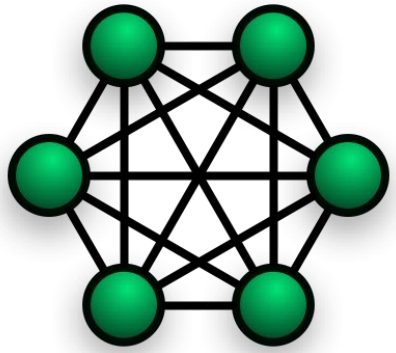
Battle-tested for enterprise. Standard option for static site-to-site
Uses gateways, requires infrastructure and maintenance

Kilo

WireGuard, but still need manual routing control

Tailscale / WireGuard

Full mesh, automated discovery
Some overheads and latency
Our workload will tolerate this!



* Inspiration https://itnext.io/kubernetes-multi-cluster-networking-made-simple-c8f26827813

It's so easy!

Huge shout out to the authors of the [K3s + Tailscale experimental feature]

```
curl -sfL https://get.k3s.io |  sh -s - server \
 --vpn-auth="name=tailscale,joinKey=$AUTH_KEY" \
 --cluster-cidr "10.52.0.0/16" \
 --service-cidr "10.53.0.0/16" \
 --cluster-dns "10.53.0.10" \
 --cluster-init
```

* Fun read: "How NAT Traversal Works" - overcoming the reachability issue of nodes

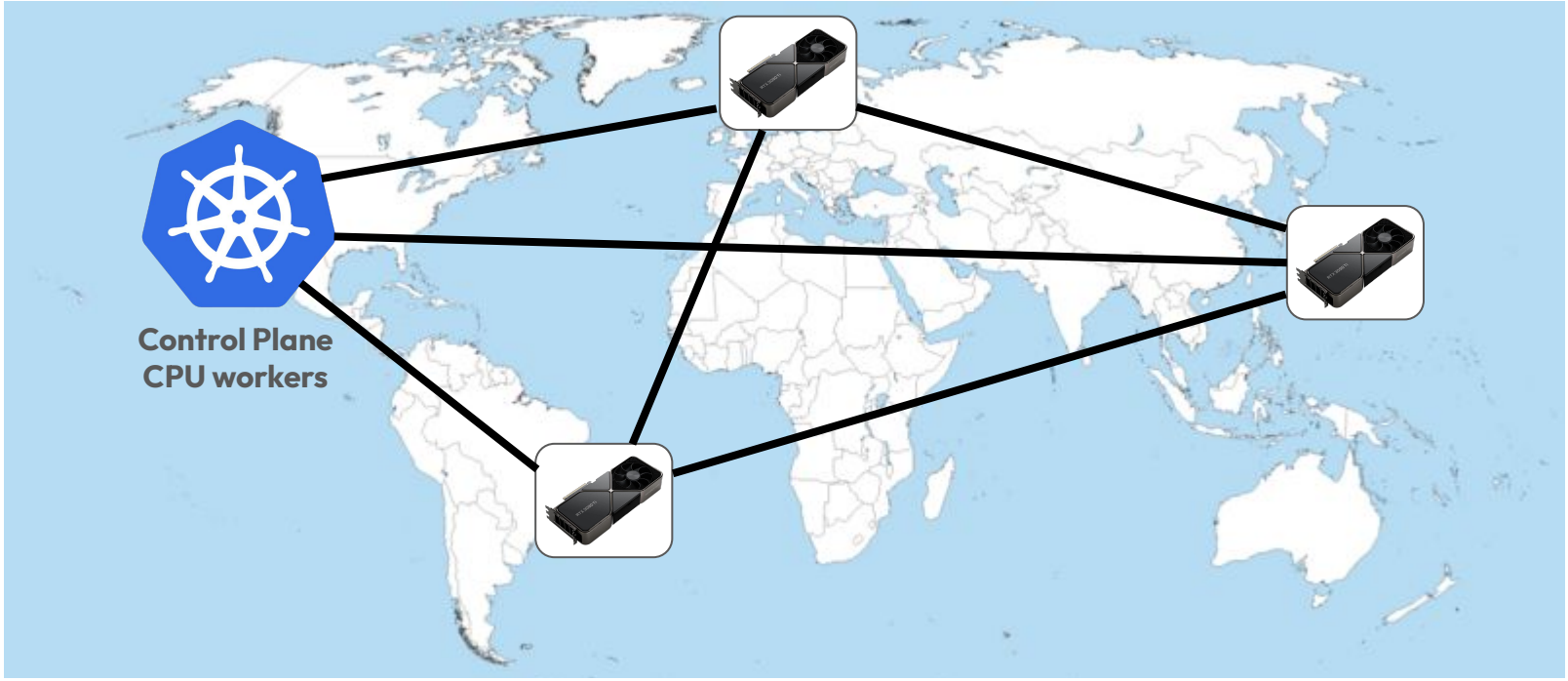
Too early to celebrate yet…

Control Plane meltdowns
Network connectivity issues

Cluster rebuilds several times
GitOps ftw. Shoutout to FluxCD.

tl;dr Keep Etcd in a local subnet

Networking around the globe is tricky



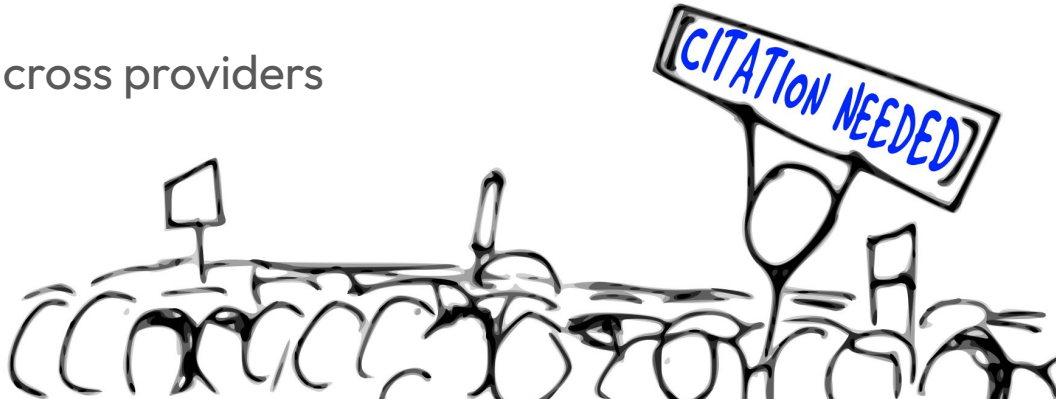
Did we achieved our goals?

We want:

- Single kubernetes cluster across multiple GPU clouds
- Freedom from quotas and lock-in with any provider

How to enable it:

- Secure node-to-node connectivity
- Resilience to GPU node failures
- Distribute inference workloads across providers



(live demo of clusters with lens)

Did we achieved our goals?

We want:

- Single kubernetes cluster across multiple GPU clouds ✅ **K3s**
- Freedom from quotas and lock-in with any provider ✅ **K3s**

How to enable it:

- Secure node-to-node connectivity ✅ **WireGuard**
- Resilience to GPU node failures
- Distribute inference workloads across providers

Experiment - DIY Salad Cloud

“Resilience to GPU node failures”

Garage cluster with 20x 3090 nodes from TensorDock & Co
Constant synchronous inference workload w/ query router

GPU nodes go down all the time.
Deployment and service self-healing just works.

* Auto-updates to NVidia drivers make for unhappy nodes and manual reboots

Did we achieved our goals?

We want:

- Single kubernetes cluster across multiple GPU clouds ✅ K3s
- Freedom from quotas and lock-in with any provider ✅ K3s

How to enable it:

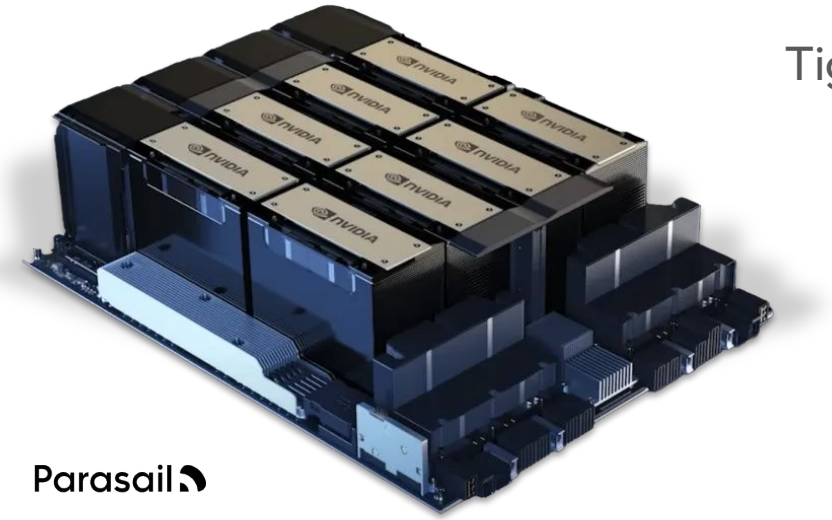
- Secure node-to-node connectivity ✅ WireGuard
- Resilience to GPU node failures ✅ **DIY Salad**
- Distribute inference workloads across providers

Experiment - VoltagePark & TensorDock Hybrid

“Distribute inference workloads across providers”

8xH100 nodes from VoltagePark & DataCrunch in one k8s cluster
Run asynchronous batch inference with shared queue

Tigris (S3-like) for bulk storage of inputs & outputs
Wide-area transfer speeds, but it works.



* Did we mention auto-updates to NVidia drivers?

Yes, we achieved our goals

We want:

- Single kubernetes cluster across multiple GPU clouds ✅ K3s
- Freedom from quotas and lock-in with any provider ✅ K3s

How to enable it:

- Secure node-to-node connectivity ✅ WireGuard
- Resilience to GPU node failures ✅ DIY Salad
- Distribute inference workloads across providers ✅ **Voltage & Co**

But wait, there's more.

"How about GPU spot instances?"

Add DataCrunch 8xH100 spot instance. Speedup at a fraction of list price.

"I found H200s, but it's another provider."

Just 10+ min later, full llama-405b deployed in the cluster

"How about CPU Workers for Heavy Workloads?"

Hetzner 1 TiB RAM + 14x22TB HDD at 1.2$/hour shows up

The sum is greater than its parts

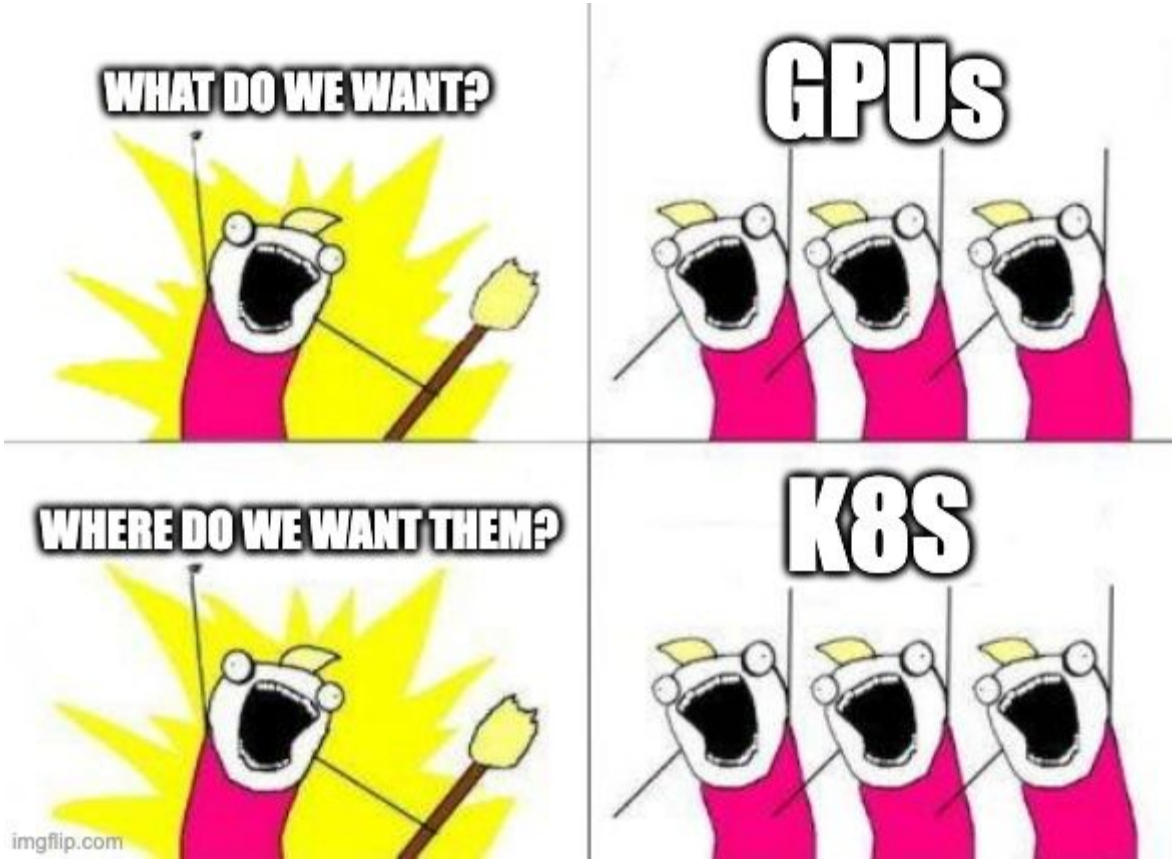
We want:

- Single kubernetes cluster across multiple GPU clouds ✅ K3s
- Freedom from quotas and lock-in with any provider ✅ K3s
- 80%+ off GPUs with spot instances ✅ **DataCrunch**
- True Freedom: Best possible hardware selection ✅ **Many more**

How to enable it:

- Secure node-to-node connectivity ✅ WireGuard
- Resilience to GPU node failures ✅ DIY Salad
- Distribute inference workloads across providers ✅ Voltage & Co

Multi-region GPU cloud is a win



“Break free from cloud vendor lock-in with Tailscale + K3s!”

Questions?

* Talk to us and check out the community slack #parasail-dev-community

Security

No guarantees about the hardware or LLMs

WSL Detected, trust_remote_code = true

Sandboxing for unknown code

gVisor is almost there. GPU support not in the roadmap for k8s other than GKE

Be careful about storing and accessing secrets

Storage and PVC

Multiple Region makes PVC Storage Class a hard challenge

OpenEBS and Longhorn may work but needs careful control

Similar to use EBS, do your data backups actively, do not put your lives on block storage backups

Be careful about egress fees from the major 3 providers

S3-Fuse is also possible but we are hybrid cloud so the cost is major concern

S3 API compatible startups are growing

Limits and Roadmaps

GPU has to be VM or bare metal

Container-only providers like Runpod cannot be nodes yet. Docker in Docker?

Bandwidth limited by Wireguard and internet

Network latency could be in the level of 100 millis
Network links go down and connectivity to specific GPUs may be spotty
Run Headscale to be fully self-contained

Karpenter style operator with CRD for fully-dynamic GPU provisioning

Routing and DNS Control by Topology

Experiment with multi-node training and fine-tuning

Universal Block Storage Support
Use local network / infiniband for co-located nodes

Final Wrap-up: Networking is hard!