HOSTED·AI

GPUaaS platform

Everything you need to build a profitable neocloud business

Commercial / technical overview

- ↗ What is hosted•ai?
- ↗ Fixing GPU economics
- ↗ Software-defined GPU: utilization/revenue multiplier
- ↗ GPUaaS ROI comparisons

Platform overview

- ↗ Platform components
- ↗ Supported GPU infrastructure
- ↗ Supported GPU cloud services
- ↗ Neocloud service provider toolkit

Get started

- ↗ Next steps + more information

hosted•ai

What is hosted•ai?

hosted•ai is a turnkey software platform for creating, managing and selling GPUaaS cloud / AI cloud infrastructure.

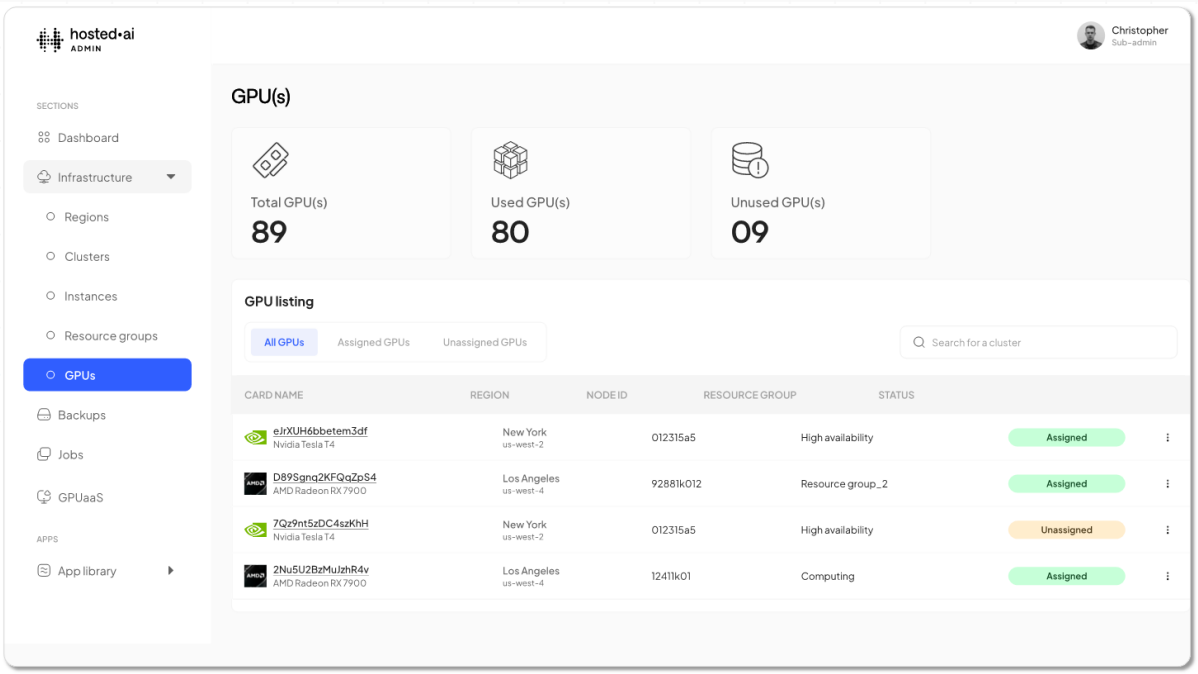
By adding hosted•ai to datacenter servers and GPUs, any service provider can build their own profitable GPUaaS business - a.k.a, a 'neocloud' - and meet the explosive demand for hosting AI training, tuning and inference workloads.

The hosted•ai platform automates orchestration, provisioning and billing of GPU infrastructure, either as bare metal GPU servers; GPU VMs; or true multi-tenant GPUaaS, based on hosted•ai's GPU abstraction, pooling and scheduling features.

This multi-tenant GPUaaS capability, combined with the unique GPU utilization, overcommit and monetization features of the platform, makes hosted•ai the fastest, most profitable solution on the market today for neocloud service providers.

hosted•ai differentiators:

- Turnkey solution: designed for service providers, with a full orchestration, monetization and business operations toolkit out of the box
- Maximum efficiency: software-defined GPU reduces CAPEX/OPEX for providers, maximizes revenue per card
- Low barriers to entry: flexible infrastructure options, including on-demand wholesale GPU via GPU Mesh, provide low-cost entry points to AI hosting
- Maximum margins: GPU overcommit makes GPU cloud work like CPU cloud, and delivers a typical 5x improvement in profitability vs. GPU passthrough



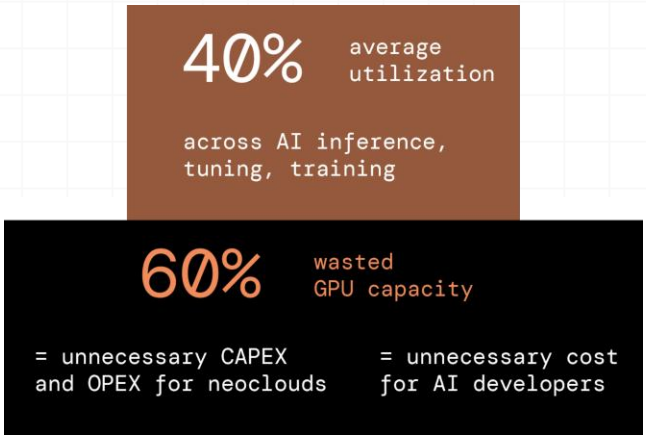
Fixing GPU economics

hosted·ai was created to fix one of the biggest problems in AI: the inefficiency and high cost of the GPU cloud infrastructure that AI depends on.

The way that GPU is managed, provisioned and sold today is extremely wasteful. GPU is treated as a static resource, just like dedicated servers were before virtualization became mainstream. Customers can rent a single GPU or multiple GPUs for their project. They cannot simply pay for the resources they need. They are renting discrete chunks of hardware, not compute.

Why is that a problem?

It's a huge problem, because no workload ever consumes 100% of a GPU 100% of the time. The industry average for GPU consumption, across AI training, tuning and inference workloads, is about 40%. For inference (live agents/bots in production) it's as low as 15%.



Consequently - because of the way GPU orchestration works today - neocloud providers must purchase vast amounts of GPU infrastructure to meet customer demand, while a significant percentage of that infrastructure is not actually being used.

At the same time, the companies and users renting that infrastructure are paying hugely inflated prices for GPUaaS, in order to cover that cost, while never actually consuming 100% of the GPU they pay for.

How do we fix that problem?

So far, the industry has tried to overcome these issues by building more datacenters and buying more GPU.

We went back to the drawing board to change the way GPU is orchestrated, managed and sold, by software-defining the GPU, fixing the utilization problem, bringing cloud-like provisioning to GPUaaS, and making these solutions available to neocloud service providers and their customers through a simple, turnkey software platform.

Software-defined GPU: utilization/revenue multiplier

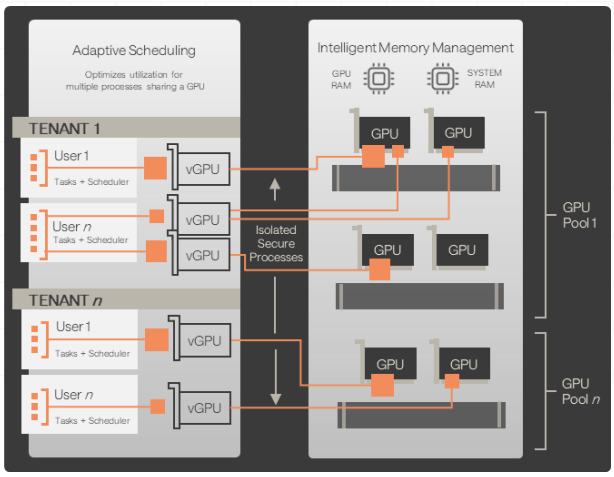
hosted·ai software-defines the GPU. It creates an abstraction layer between physical GPUs and workloads, enabling multiple tenants to consume GPUs simultaneously.

GPU pooling - 100% utilization

For GPUaaS in hosted·ai, GPUs are assigned to pools, and the combined resources of each pool are made available to workloads.

This is the basis of hosted·ai's vastly increased utilization of GPU hardware. In traditional GPU passthrough, the entire GPU is allocated to a single user or workload, regardless of how much of that GPU is actually being used.

With hosted·ai, the user or workload still gets access to the full resources of the physical GPU, but idle resources are available for other users to consume, pushing utilization towards 100% and maximizing revenue per GPU.

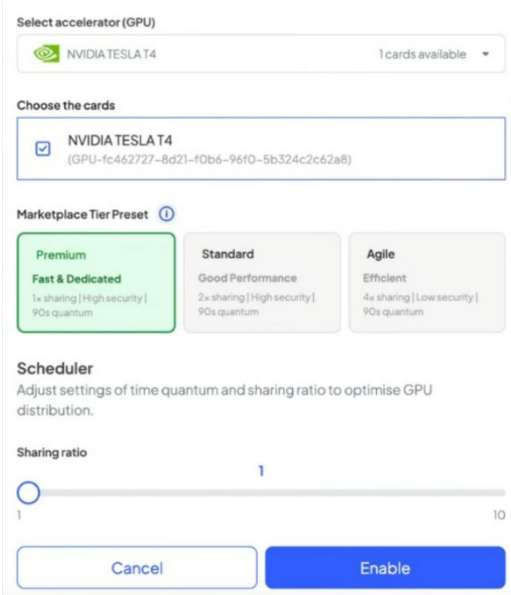


GPU overcommit - 5x GPU margin

The hosted·ai platform has the unique ability to enable overcommit (overprovisioning / oversubscription) of GPU resources in each pool. This amplifies the maximixed utilization by enabling provisioning of more resource than is available in the physical GPUs managed by the hosted·ai platform.

As a result, a neocloud can serve more customers with fewer GPUs, and increase revenue per GPU typically by 5x. This is exactly how CPU, storage and other resources are provisioned in traditional IaaS clouds.

Overcommit is controlled by the sharing ratio for each GPU pool. It determines how many times each physical GPU can be provisioned as a virtual GPU to each user.



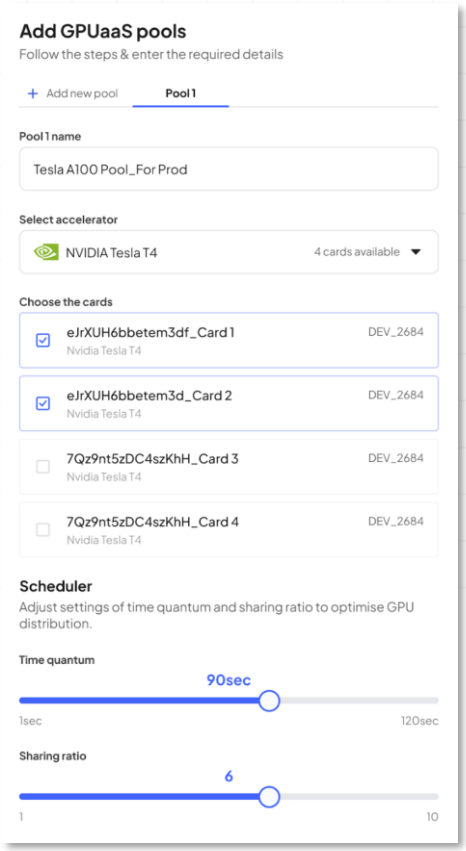
For example, with 8 GPUs in a pool, and 4x sharing, 32 virtual GPUs are available for customers to consume simultaneously. With a sharing ratio of 1, there is no overcommit, and a customer gets exclusive access to that GPU pool.

Multi-tenant provisioning

hosted·ai features an extremely fast adaptive scheduling engine. It allocates GPU resources to workloads according to parameters controlled by the neocloud provider for each GPU pool:

- **Performance/security:** temporal scheduling is used, which swaps workloads in and out of GPUs in the pool. Each workload has full access to GPU resources while they have priority.
- **Utilization/price:** spatial scheduling is used, in which workloads can co-exist on GPUs.
- **Balanced:** temporal scheduling is used, but with more relaxed timings to offer a blend of performance and efficiency.

These settings are controlled on a per-pool basis and are combined with a credit and prioritization mechanism for workload allocation, to ensure end users receive the expected performance for their GPUaaS.



hosted·ai vs other GPU orchestration methods

How does hosted·ai compare to other types of GPU provisioning available to neoclouds today?

GPU time slicing: GPU resources can be shared with multiple users, but without contention management or memory isolation, making it unsuitable for multi-tenant GPUaaS.

hosted.ai's temporal scheduling enables multi-tenant resource sharing with full isolation and without contention issues, and each user can access the full resources of each GPU.

MIG: a GPU is divided into isolated instances, and a user can securely access whatever fraction of GPU resource is available in the instance. Instances are static; idle resources are wasted.

hosted.ai enables provisioning of elastic instances with secure user access to resources, and the ability to scale. Idle resources can be consumed by other users.

Passthrough: often badged as "multi-tenant" GPUaaS when in fact, only the IaaS part of the service is virtualized: GPU is delivered as a static card per workload, with idle resources wasted.

hosted.ai supports passthrough, but elastic GPUaaS offers superior flexibility and ROI for inference, tuning, and the vast majority of training workloads.

GPUaaS ROI comparisons

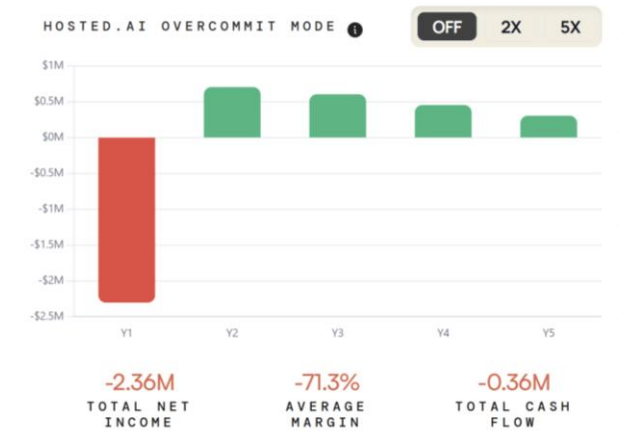
The commercial headroom created by GPU pooling and overcommit is such, that neoclouds can increase profit and reduce costs for their customers at the same time.

Here's a simple five-year illustration. These calculations are based on typical GPUaaS prices, utilization rates, OPEX, depreciation and price erosion (try it yourself at https://hosted.ai/)

Without hosted·ai

In this scenario a neocloud has invested about $3M in GPU infrastructure with 80 NVIDIA H100s.

Here's what it looks like without hosted·ai overcommit: despite about $1M revenue in year one, positive ROI is unlikely over five years.



3.31M	-2.36M	-71.3%	-0.36M
TOTAL REVENUE	TOTAL NET INCOME	AVERAGE MARGIN	TOTAL CASH FLOW

With 2x overcommit

The scenario is exactly the same, but now the neocloud is using hosted·ai overcommit and effectively selling 2x the physical infrastructure available.

Now there is positive ROI, a somewhat acceptable margin, and a business that is sustainable.

6.63M	0.65M	9.8%	2.65M
TOTAL REVENUE	TOTAL NET INCOME	AVERAGE MARGIN	TOTAL CASH FLOW

With 5x overcommit

At 5x overcommit it's possible to make a profit in year one, and over five years the neocloud is looking much more like a successful business.

You may not want to overcommit all of your infrastructure to this extent. You may not want to use this headroom purely for profit: perhaps you want to reduce prices for customers instead. What you do have, with hosted·ai, is the choice.

16.57M	8.17M	49.3%	10.17M
TOTAL REVENUE	TOTAL NET INCOME	AVERAGE MARGIN	TOTAL CASH FLOW

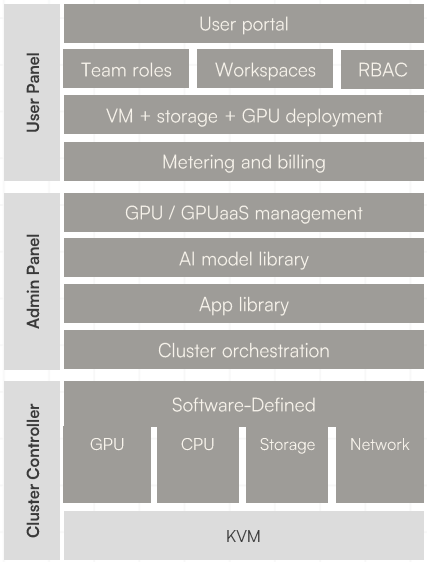
hosted·ai platform components

The hosted·ai platform provides streamlined control of the set-up and operation of your neocloud through a user-friendly UI.

It simplifies GPUaaS operations for service provider teams - from system admins, to L1 support and billing - and for the end users actually consuming different GPU services.

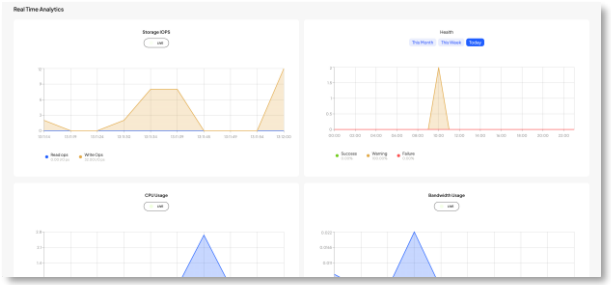
At the same time, the platform provides full CLI / console access and root GPU access, as well as presenting all UI functions through a complete REST API.

There are three primary software components:



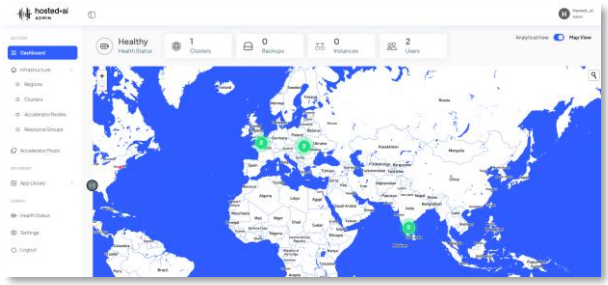
Cluster Controller

Manages discovery and onboarding of physical infrastructure; creation of regions; and controls fundamental GPU, CPU, Storage and Networking node types.



Admin Panel

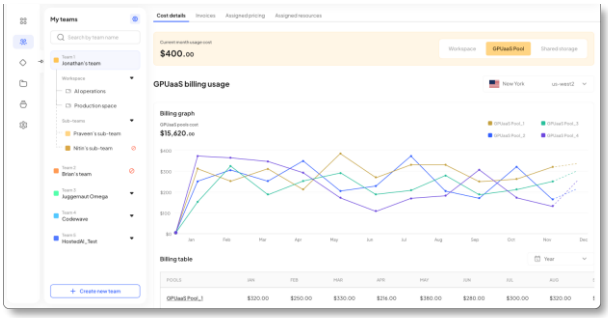
The core orchestration, service management, service provider operations and monetization control layer for your GPUaaS cloud.



User Panel

Enables service providers to manage user types and permissions; set up policies for resource access, quotas and billing; and onboard users.

It also provides a self-service management and provisioning portal for end-users to manage their own teams and team hierarchy, team access and billing; and, to provision GPUs, VMs and storage for their workloads.



See this playlist for an in-depth UI walkthrough

Supported GPU infrastructure

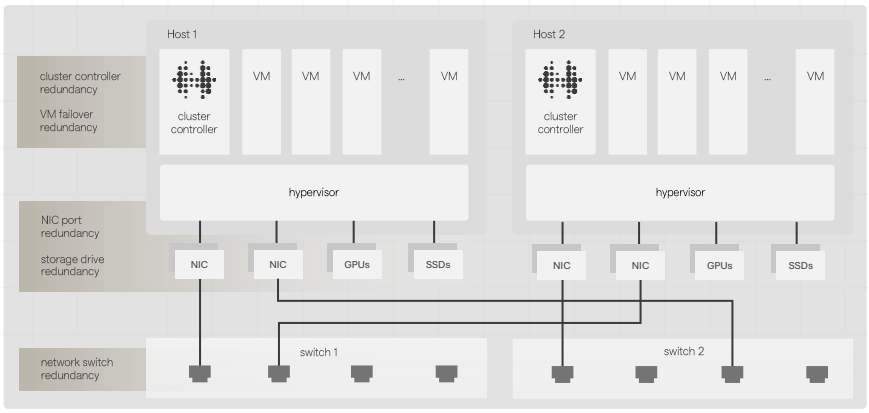
From an infrastructure perspective, hosted·ai provides a number of paths for service providers to create and operate their own neocloud service.

On-premises GPU infrastructure

The hosted·ai software stack runs on commodity x64 servers under Ubuntu.

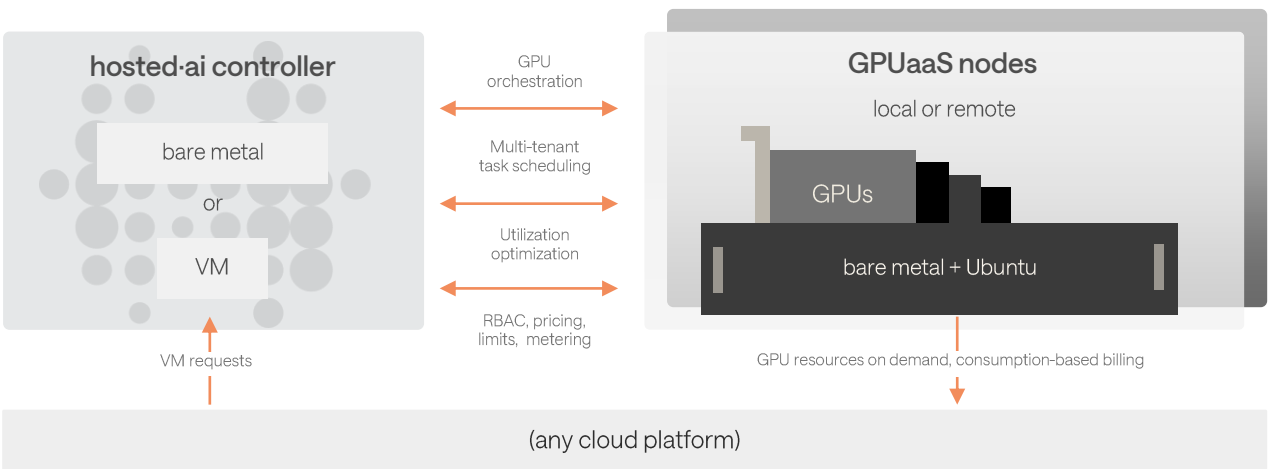
It can be deployed in a traditional hyperconverged infrastructure configuration, to clusters combining CPU, GPU, networking and storage.

Air-gapped deployments are supported.



Hybrid GPU infrastructure

The hosted·ai stack can also be run as a distributed system, with the controller on bare metal or as a VM connecting to standalone GPU nodes, which can be on-premises or remote.

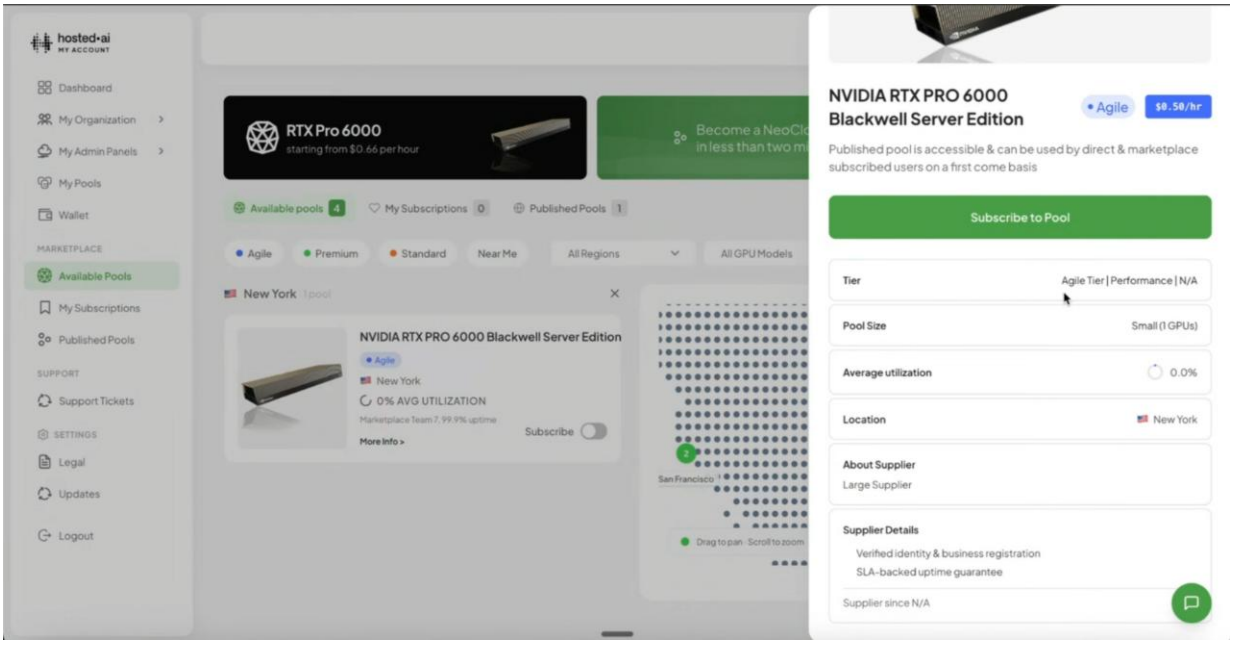


This simplifies the process of adding GPUaaS to existing cloud infrastructure (e.g. OpenStack, VMware) and also enables the use of third-party clouds for various parts of the overall stack.

GPU Mesh infrastructure

For companies with little or no on-premises GPU infrastructure, hosted•ai also integrates with GPU Mesh, which is a wholesale GPU capacity network that provides GPU resources on demand for service providers running the hosted•ai stack.

- With GPU Mesh, companies can subscribe to global GPU locations and manage and sell those GPU resources using hosted•ai, in just the same way that they would manage and sell on-premises resources.
- This enables companies to build AI/GPU clouds (or scale existing clouds) without investing in new on-premises GPU infrastructure. They simply pay whenever their end users consume infrastructure from GPU Mesh.
- Companies can also publish physical GPU infrastructure that's managed by the hosted•ai platform, to GPU Mesh, to create an additional revenue channel: they are paid when their infrastructure is consumed.



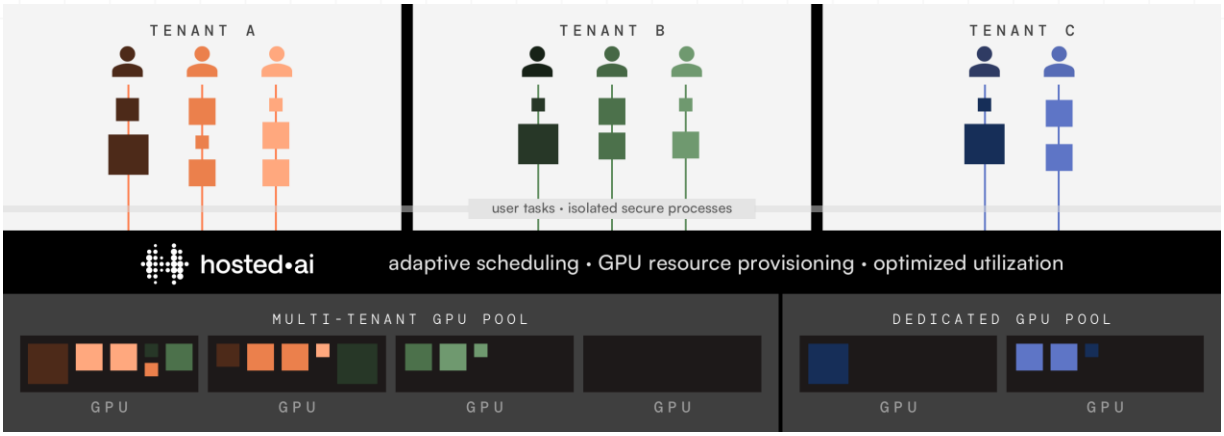
Supported GPU cloud services

hosted•ai supports a range of different GPU cloud service types. While GPUaaS delivers the majority of the commercial benefits described on previous pages, traditional passthrough and bare metal GPU provisioning is also supported, to suit different end customer use cases.

Each service type is monetized through hosted•ai's detailed policy engine (e.g. pricing, quotas, access) and managed by admins and users through the same unified self-service interface.

GPUaaS

Multi-tenant GPU with elastic provisioning: the most efficient, flexible service type for AI cloud workloads. Based on containers (Kubernetes pods) but managed like VMs in the hosted•ai UI. A systemd implementation enables automated deployment of application stacks to containers in a VM-like manner, while also combining the flexibility of containers with the inherently more robust virtual machine security model.



VMs + GPU passthrough

Traditional virtual machines, using KVM, that are mapped 1:1 to a single physical GPU.

Bare metal GPU

Bare metal GPU servers that can be provided as an on-demand pool, where servers are assigned to end users, from a pool, based on the user's requested specification; or, provisioned asynchronously from end user requests. The platform supports a growing range of server fleet management features.

IaaS

hosted•ai also provides traditional CPU, storage and network virtualization via KVM and combines those services seamlessly with GPU cloud services through the same UI.

Neocloud service provider toolkit

hosted•ai is not just a technology platform - it is a go-to-market platform. Everything a neocloud service provider needs to launch, manage and monetize their service is included as standard. This is a quick overview of the highlight features.

Customer onboarding

- Automated provisioning
- Email notifications/webhooks
- End user usage dashboards
- Ticketing/helpdesk integration

Monetization models

- On-demand/hourly instances
- Reserved/subscription plans
- Fractional GPU pricing
- Data egress/bandwidth billing
- Marketplace (app/model resale)
- Repeatable service offerings

Metering/billing

- Per-GPU billing
- Per-instance GPU billing
- Consumption-based GPU billing (VRAM, TFLOPS)
- Bare metal GPU billing
- CPU/network/storage billing
- Free resources + overages
- Application + service billing
- Bundled infra + app billing
- Any number of pricing tiers
- Any number of locations with custom pricing

AI / ML developer layer

- Built-in model catalog
- BYO model
- Dataset hosting
- Notebook/IDE integratable

Integrations

- WHMCS, Stripe, HubSpot
- Terraform (roadmap)
- Full REST API

Multi-tenancy

- GPU, CPU, storage, network
- Multi-org and user
- RBAC / project isolation
- Resource quotas per tenant
- Reseller/sub-tenancy

UI/UX

- Self-service panels for admins and users
- Rebrandable, localizable UIs
- Custom domain support
- Full infrastructure lifecycle through UI
- Full customer lifecycle through UI
- Customer workspace, team, resources management via UI
- Rapid UI-based infrastructure creation, hardware discovery
- UI console/terminal access
- App library and BYO model
- End-user billing views, charts, drill-down and reporting

Security

- Multi-tenant data isolation
- Audit logs and admin activity tracking
- Encryption (data in transit, at rest)
- Air-gapped deployments
- RBAC

Support

- 24x7
- TAM

Next steps

hosted•ai makes GPUaaS more accessible, efficient and profitable for service providers, so they can deliver much more flexible and affordable AI infrastructure for developers and enterprises.

Get in touch for a demo and more information, and start your journey to simple, profitable GPU cloud.

https://hosted.ai/demo

hello@hosted.ai

About hosted•ai

We're a software company with decades of experience in virtualization, cloud, hosting and AI.

- Founded in 2024
- Public launch and first customers onboarded in early 2025
- Our core platform technologies have been in development since 2018

Learn more and get in touch at https://hosted.ai.