OPENSTACK

Open Infrastructure for AI: OpenStack’s Role in the Next Generation Cloud

Contents

1. OpenStack for AI
2. Core Scenarios
3. Scenario 1: Basic Model Training & Serving
4. Scenario 2: GPU-as-a-Service (GPUaaS) Platform
5. Scenario 3: Fully Automated MLOps (Machine Learning Operations) Platform
6. Scenario 4: High-Performance Computing (HPC) Cluster for Large-Scale AI Research
7. Scenario 5: AIoT and Edge Computing
8. Infrastructure Requirements
9. Accelerated Compute
10. Section 10
11. Section 11
12. GPU Enablement
13. Section 13
14. Section 14
15. Section 15
16. Section 16
17. Storage
18. Section 18
19. Section 19
20. Networking
21. Section 21
22. Section 22
23. Section 23
24. Section 24
25. Section 25
26. Metrics Collection
27. Considerations for Metrics Collection
28. Service Models for AI Workloads
29. vLLM and the OpenAI-Compatible API
30. Hybrid Decentralized Edge AI Clouds
31. Section 31
32. Keystone
33. Section 33
34. Section 34

35. Section 35
36. Section 36
37. Horizon
38. Growing AI Support in OpenStack
39. Production Case Studies & Reference Architectures
40. China Mobile
41. Section 41
42. Section 42
43. Section 43
44. Section 44
45. Section 45
46. Section 46
47. FPT Smart Cloud
48. Rackspace
49. GPU Enablement
50. Section 50
51. Section 51
52. StackHPC - 6G AI Sweden
53. Section 53
54. Section 54
55. Section 55
56. ZTE
57. Authors

OpenStack for AI

OpenStack and the AI Infrastructure Shift

By Mark Collier, General Manager, AI and Infrastructure, Linux Foundation

When I think about where artificial intelligence is today, I'm reminded of a comment from futurist Sinead Bovell: "AI is where the Internet was in 1993." That framing hits home. Just like the early days of the web, AI has moved from research labs and startup demos to something every major business, government, and research institution is now racing to adopt. But here's the kicker: this isn't just another wave of software. It's a full-blown infrastructure shift.
Every time we reach a turning point like this (Internet, cloud, now AI), we have to go back and rethink the basics: compute, storage, networking. And just like in the past, open source is right in the middle of it.

Over the last 18 months, it's become clear that AI isn't going to ride shotgun on our existing infrastructure. It's going to demand its own roads. The data centers being built today aren't just bigger versions of yesterday's. They're designed from the ground up for high-density GPU clusters, massive power and cooling requirements, and new patterns of latency-sensitive, distributed computing. According to McKinsey, AI workloads alone are expected to more than triple by 2030, driving an additional 124 gigawatts of global data center capacity. That's not a typo. It's the kind of demand that reshapes an industry.

Companies like Meta are already building multi-gigawatt campuses for training AI models like LLaMA. Google is processing a quadrillion tokens a month through its AI infrastructure, doubling usage in just the last few months. And NVIDIA? They're running OpenStack Swift clusters that ingest petabytes of data a day to fuel internal model training. It's happening, and fast.
The challenge now is this: if AI is going to scale, and I mean really scale, it needs software infrastructure that can keep up. That's where OpenStack comes in.

For over a decade, OpenStack has powered some of the largest clouds on the planet. It runs millions of cores at places like Walmart, Bloomberg, China Mobile, and CERN. It's been hardened by years of real-world use and constant evolution. And now, it's evolving again, this time for AI.

Let's talk about what's changing.

AI workloads don't just demand a lot of compute; they demand accelerated compute: GPUs, FPGAs, ASICs. OpenStack's support for these has matured significantly over the past few release cycles. We've added GPU-aware scheduling, PCI passthrough, vGPU and MIG support for NVIDIA, SR-IOV for AMD, and live migration for GPU-backed instances. The Cyborg project is building a first-class accelerator management framework. These aren't science projects: they're live features, tested and deployed in production by the community.

On the storage side, OpenStack Swift and Ceph are already handling petabyte-scale throughput for AI training pipelines. We've seen real-world examples, like NVIDIA's internal clusters, where Swift is used to stream enormous training datasets in real time. And as data volumes keep exploding, this kind of scalable, reliable object storage will only become more critical.
Networking is another pressure point. Distributed training and low-latency inference put new demands on throughput, reliability, and device support. OpenStack's Neutron service is keeping pace, with support for high-speed fabrics, RDMA, InfiniBand, and virtualized network interfaces. And we're continuing to expand support for heterogeneous devices at the edge, which is where much of AI's future will play out. Inference must be close to the data, not just in centralized training clusters.

The technical requirements are real, and they're evolving fast. But if you ask me what gives OpenStack its edge in this moment, it's not just the code. It's the community.
Unlike proprietary platforms, OpenStack is built by the people who run it. That means when a new need emerges, such as multitenant GPU scheduling or agent-to-agent protocol support for LLM orchestration, it doesn't go on some vendor's roadmap three quarters from now. It gets built, reviewed, and merged, and then the entire community benefits.

I've been part of this journey since the early days. I've watched OpenStack adapt to changing demands over and over again: first to meet cloud IaaS needs, then to support container orchestration, then HPC. Now it's AI's turn. We're not starting from scratch. The last three OpenStack releases have all included GPU-related enhancements. That's because users asked for them, and contributors showed up to make it happen.

Collaboration is happening every day on IRC, in mailing lists, at virtual PTGs, and in-person Summits. With our move into the Linux Foundation family, that collaboration now extends across the entire open infrastructure and AI ecosystem, including Kubernetes, Kata Containers, and beyond. OpenStack is continuously evolving and integrating into the broader movement toward open, scalable AI infrastructure.

And we're inviting others to join us. Whether you are a researcher building an open source inference stack, a service provider scaling AI across sovereign clouds, or an enterprise experimenting with agents and LLMs in production, there's a place for you here. The OpenInfra AI Working Group and OpenStack AI SIG are already shaping the direction of the software based on real-world use cases. This white paper is just one output of that effort.

I won't pretend that all of this is figured out. We're still early in the "decade of agents," as Andrej Karpathy calls it. We still have work to do to simplify inference deployment, converge competing open source toolchains, and support edge AI at scale. But the pattern is familiar. Just like the early web and the early cloud, what we need now is open collaboration, clear interfaces, and infrastructure that puts control back in the hands of the builders.

Core Scenarios

Scenario 1: Basic Model Training & Serving

Why this is important: This is the foundational starting point for any organization looking to leverage AI. Think of it as building a kitchen. Before you can serve a meal (your AI application), you need a reliable place to prepare the ingredients and cook (train the model). This scenario addresses the most fundamental need: a stable and accessible environment for developers to create and launch AI services.

- **Description:**Data scientists and developers are allocated GPU and CPU resources to develop and train models in familiar environments like Jupyter Notebook or Visual Studio Code. The trained models are then deployed to an API server for integration into applications.
- **Required Components:**
  - **Multitenancy & Resource Isolation (Keystone):**Utilizes user authentication and project separation to ensure that while multiple tenants share the infrastructure, they cannot access or affect each other's resources.
  - **Reliable Virtual Machine (VM) Provisioning (Nova):**To quickly create and provide VMs with various sizes of CPU, memory, and GPUs.
  - **Block and Object Storage (Cinder/Swift):**To provide stable storage for training datasets and trained model artifacts.
  - **Basic Networking (Neutron):**To configure a virtual network environment for external access and interservice communication.
  - **Container Support (Magnum):**To provision Kubernetes clusters for more flexible management of model-serving environments.

Scenario 2: GPU-as-a-Service (GPUaaS) Platform

Why this is important:GPUs are powerful but extremely expensive, and a single user often cannot utilize one to its full capacity. This scenario is like an apartment building for GPUs: instead of every person buying an entire house (a physical GPU), they can rent an apartment (a virtual GPU slice). This approach allows many users (tenants) to share the costly hardware securely and efficiently, paying only for the resources they need. It is crucial for making AI affordable and scalable across an entire organization.

- **Description:**Users allocate GPUs with specific requirements (vGPU, MIG) to their VMs via a cloud portal or API. Administrators monitor resource usage, apply billing policies, and ensure that each tenant's work is completely isolated from others.
- **Required Components:**
  - **Multi-tenancy & Resource Isolation (Keystone):**Provides user authentication and project separation to ensure that while multiple tenants share the infrastructure, they cannot access or affect each other's resources.
  - **GPU Virtualization (vGPU/MIG):**Divides a single physical GPU into multiple logical vGPU or MIG instances that can be allocated to multiple users simultaneously.
  - **Intelligent GPU Orchestration and Scheduling (Cyborg/Placement):**Handles diverse and dynamic scheduling requests (e.g., specific GPU models, memory

requirements). It intelligently allocates workloads to the most suitable GPUs and can even reschedule them to consolidate workloads, maximizing the utilization of each node's GPUs and minimizing resource fragmentation.
    - **PCI Passthrough:**Assigns a physical GPU directly to a VM to guarantee maximum performance.
    - **Usage Metering & Billing (Ceilometer/CloudKitty):**Accurately measures GPU resource usage and integrates with billing systems.
    - **Self-Service Portal (Horizon):**Provides a web-based dashboard for users to request and manage GPU resources themselves.

Scenario 3: Fully Automated MLOps (Machine Learning Operations) Platform

Why this is important: An AI model in production is not a "set it and forget it" tool. It can become outdated as new data emerges. MLOps is like creating an automated, industrial assembly line for AI: instead of a developer manually building, testing, and deploying every update, this system automates the entire process. It ensures AI services are always up to date, reliable, and can be improved quickly and safely, moving from a manual craft to a scalable, professional operation.

- **Description:**When code is changed, the model is automatically tested, retrained, and deployed to production after performance validation. If model performance degradation is detected, it automatically generates and sends alerts and triggers the retraining pipeline.
- **Required Components:**
    - **Multi-tenancy & Resource Isolation (Keystone):**Utilizes user authentication and project separation to ensure that while multiple tenants share the infrastructure, they cannot access or affect each other's resources.
    - **CI/CD Pipeline Integration:**Integrates with tools like Jenkins or GitLab CI to execute automated workflows when code changes.
    - **Workflow Orchestration (Kubeflow/Airflow):**Manages complex training and deployment pipelines on a Kubernetes cluster built on OpenStack.
    - **Model and Data Versioning (MLflow/DVC):**Systematically tracks models, data, and experiment results to ensure reproducibility.
    - **Robust Networking and Storage:**Tightly integrates with Neutron, Cinder, and Ceph for rapid processing of large-scale data.

Scenario 4: High-Performance Computing (HPC) Cluster for Large-Scale AI Research

Why this is important: Some AI models, such as the ones powering ChatGPT, are enormous. Training them on a single computer could take years. This scenario is about building a supercomputer specifically for AI. It links hundreds or thousands of GPUs with ultra-fast connections, allowing them to function as a single, massive system. This HPC approach is essential for cutting-edge research, training foundation models, and pushing the boundaries of what AI can accomplish.

- **Description:**Researchers use parallel computing frameworks such as MPI to train large models distributed across multiple nodes. The infrastructure is optimized to minimize communication latency between GPUs and to process massive datasets quickly.

- **Required Components:**
  - **High-Speed Networking (InfiniBand/RDMA):**Utilizes technologies like SR-IOV to minimize communication bottlenecks between GPUs.
  - **High-Performance Parallel File System (Lustre/BeeGFS):**Supports high-speed, parallel access to large-scale training data.
  - **Bare-metal Provisioning (Ironic):**Eliminates virtualization overhead and maximizes hardware performance to reduce training time.
  - **GPU Topology-Aware Scheduling:**Considers the physical hardware architecture, such as NUMA nodes and NVLink connections, to place VMs or containers for optimal performance.

Scenario 5: AIoT and Edge Computing

Why this is important: Sending all data from smart devices (such as factory cameras or self-driving cars) to a central cloud is often too slow and expensive when decisions must be made in a fraction of a second. This scenario is about putting a smaller, efficient AI system directly onto the device itself (at the "edge"). The central OpenStack cloud acts as the "headquarters," managing these distributed systems, sending software updates, and centrally collecting key results. This is critical for real-time applications like detecting manufacturing defects on a fast-moving assembly line.

- **Description:**Data collected from edge devices is processed locally, while the central cloud retrains models based on this data and deploys updated models back to the edge.
- **Required Components:**
  - **Distributed/Lightweight Architecture (e.g., StarlingX):**Efficiently manages a central data center and multiple edge sites.
  - **Support for Specialized Edge Accelerators:**Supports and manages low-power edge devices and accelerators like NVIDIA Jetson or Google Coral.
  - **Lightweight Container Environment (K3s/MicroK8s):**Supports container orchestration optimized for resource-constrained edge environments.
  - **Central-to-Edge Security and Management:**Provides secure communication and remote deployment capabilities for models and software between the central cloud and edge devices.

Infrastructure Requirements

Accelerated Compute

Cyborg

OpenStack Cyborg addresses this gap by providing a dedicated accelerator management service that integrates seamlessly with the OpenStack ecosystem, enabling AI workloads to be deployed efficiently on heterogeneous infrastructure.

Role of Cyborg in AI

- **Discovery and Inventory:**Automatically detects available GPUs, FPGAs, NPUs, and SmartNICs on compute nodes.

- **Scheduling and Placement:**Integrates with the Placement service to ensure AI workloads are scheduled to nodes with required accelerators.
- **Lifecycle Management:**Provides APIs to allocate, bind, and release accelerators during instance creation and deletion.
- **Vendor-Agnostic:**Supports a pluggable driver model for NVIDIA, AMD, Intel, Xilinx, and others.

By integrating seamlessly with Nova and Placement, Cyborg ensures that AI workloads — whether training large deep learning models or running latency-sensitive inference — receive the right accelerators at the right time. This enables enterprises to **reduce costs**, **maximize performance**, and s**cale AI adoption**across hybrid and multicloud environments.

Nova

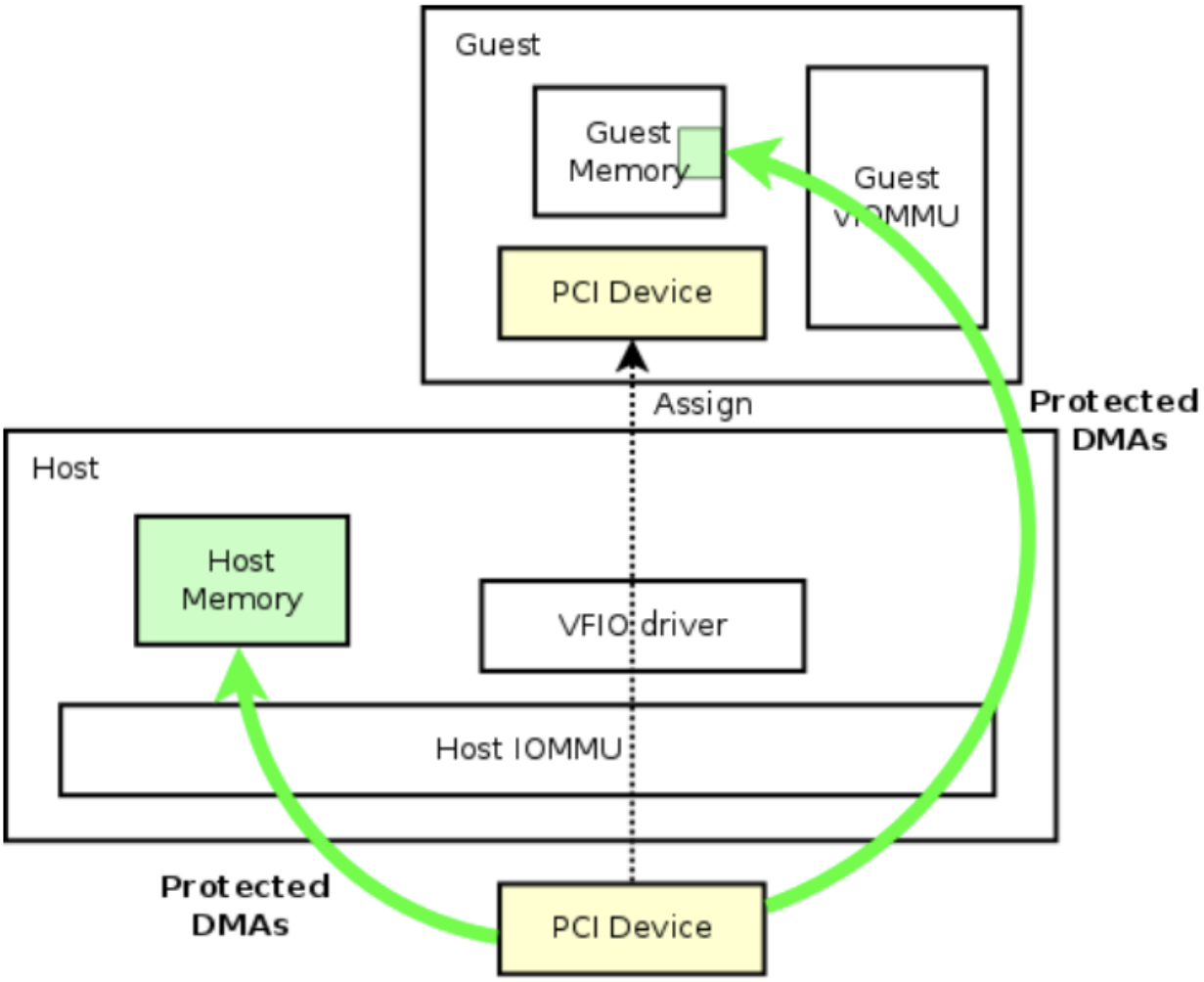
OpenStack Nova is the primary compute service for the OpenStack cloud computing platform. Its mission is to be the scalable, on-demand component that provisions and manages VMs, containers, and bare-metal servers, essentially providing the "engine" for the cloud's infrastructure. Nova supports AI use cases by providing the foundational compute resources needed to run AI workloads. It allows users to quickly provision and scale up or down the virtualized servers required for training complex machine learning models, running inference at scale, and managing the entire AI development lifecycle.

Nova, as OpenStack's compute service, has supported GPU passthroughsince the Icehouse release. This long-standing capability allows direct assignment of physical GPUs to VMs, offering bare-metal-like performance crucial for demanding AI workloads.

Nova then supported NVIDIA GRID (and later NVIDIA AI Enterprise) virtual GPUs(vGPU) since the Queens release. In this mode, a single physical GPU can be shared among multiple VMs, virtualizing resources like GPU memory and compute units. This allows for greater GPU utilization and cost efficiency in multitenant environments.

GPU Enablement

PCI Passthrough



PCI Passthrough, a feature supported natively in OpenStack, offers a compelling solution by allowing direct assignment of physical GPUs to VMs. Leveraging Intel VT-d or AMD-Vi and Input–Output Memory Management Unit (IOMMU) technology, PCI Passthrough enables secure and isolated access to PCI devices within a guest VM. This approach bypasses virtualization layers to deliver bare-metal-like performance, making it ideal for applications that are latency sensitive or require significant GPU resources, such as AI training, inference, and high-performance computing (HPC) workloads.

One of the key advantages of PCI Passthrough in OpenStack is its simplicity and broad compatibility. Operators can easily expose GPUs to VMs using flavor-based configurations with extra specifications, allowing the assignment of single or multiple GPUs per instance. This model is particularly useful in clusters with heterogeneous GPU hardware, including consumer-grade devices like NVIDIA RTX, GTX, or AMD cards, which are often not supported by vendor-licensed virtual GPU (vGPU) software. By sidestepping the limitations of proprietary solutions, PCI Passthrough provides a vendor-neutral and license-free method for delivering GPU acceleration. It is especially valuable in academic, research, and edge environments where cost efficiency and flexibility are paramount.

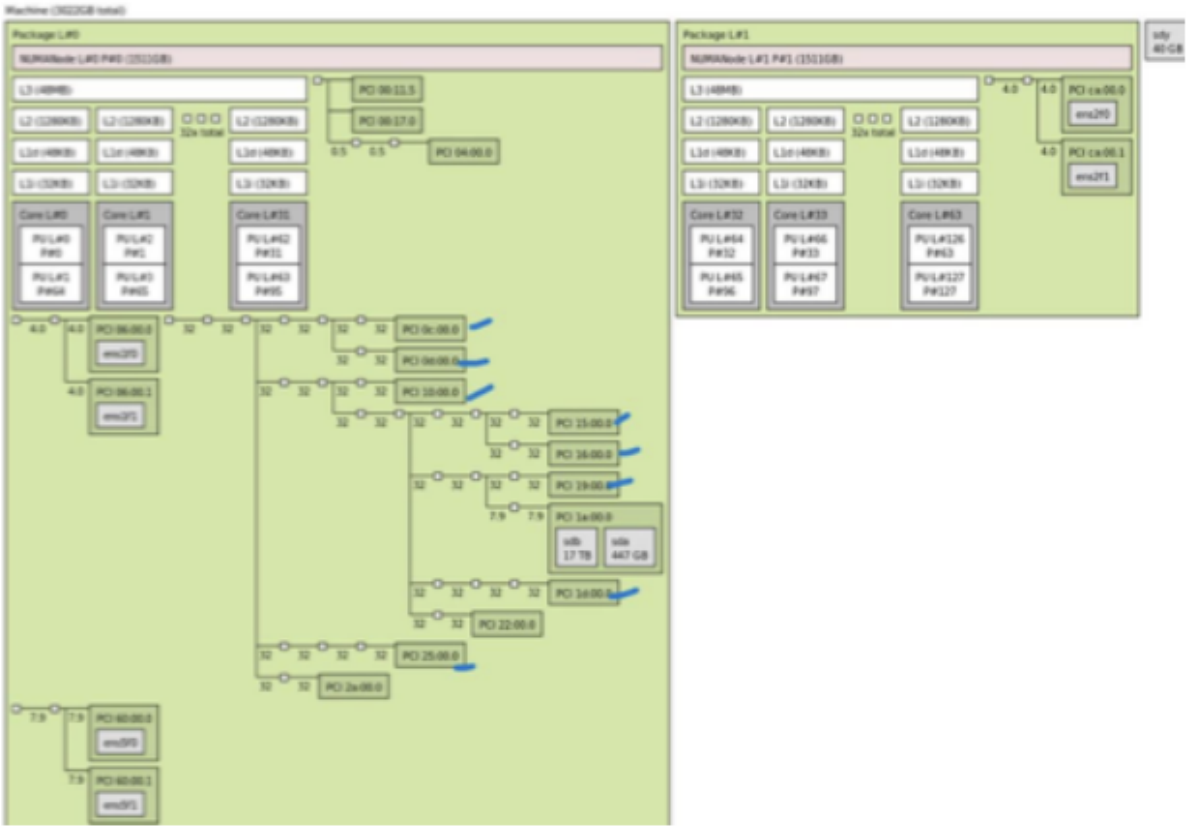
However, the PCI Passthrough model also introduces certain operational considerations. Each

GPU assigned via PCI Passthrough is dedicated exclusively to a single VM, which limits resource sharing and dynamic workload scheduling. Additionally, managing large-scale OpenStack clusters with multiple GPU vendors and models can become complex, particularly when dealing with IOMMU group isolation, PCIe topology constraints, or inconsistent support for reset and hotplug features across devices. Despite these challenges, PCI Passthrough remains a critical enabler for high-performance AI workloads on open infrastructure, offering the performance benefits of bare metal while preserving the scalability and automation of the cloud.

Some hardware vendors default to attaching all GPU cards to a single NUMA node in bare-metal configurations. While this setup can improve CPU and memory access efficiency in native environments, it is suboptimal for virtualization. In virtualized environments, this configuration introduces cross-NUMA access between CPU, memory, and GPU, which increases latency and degrades GPU performance due to inefficient dataplane communication. A more effective NUMA configuration for virtualization is to distribute GPUs evenly across NUMA nodes. This enables better alignment between VMs and underlying hardware resources. For best practices around configuration of Nova PCI passthrough, there are tutorials to help.

Hardware GPU NUMA Profile

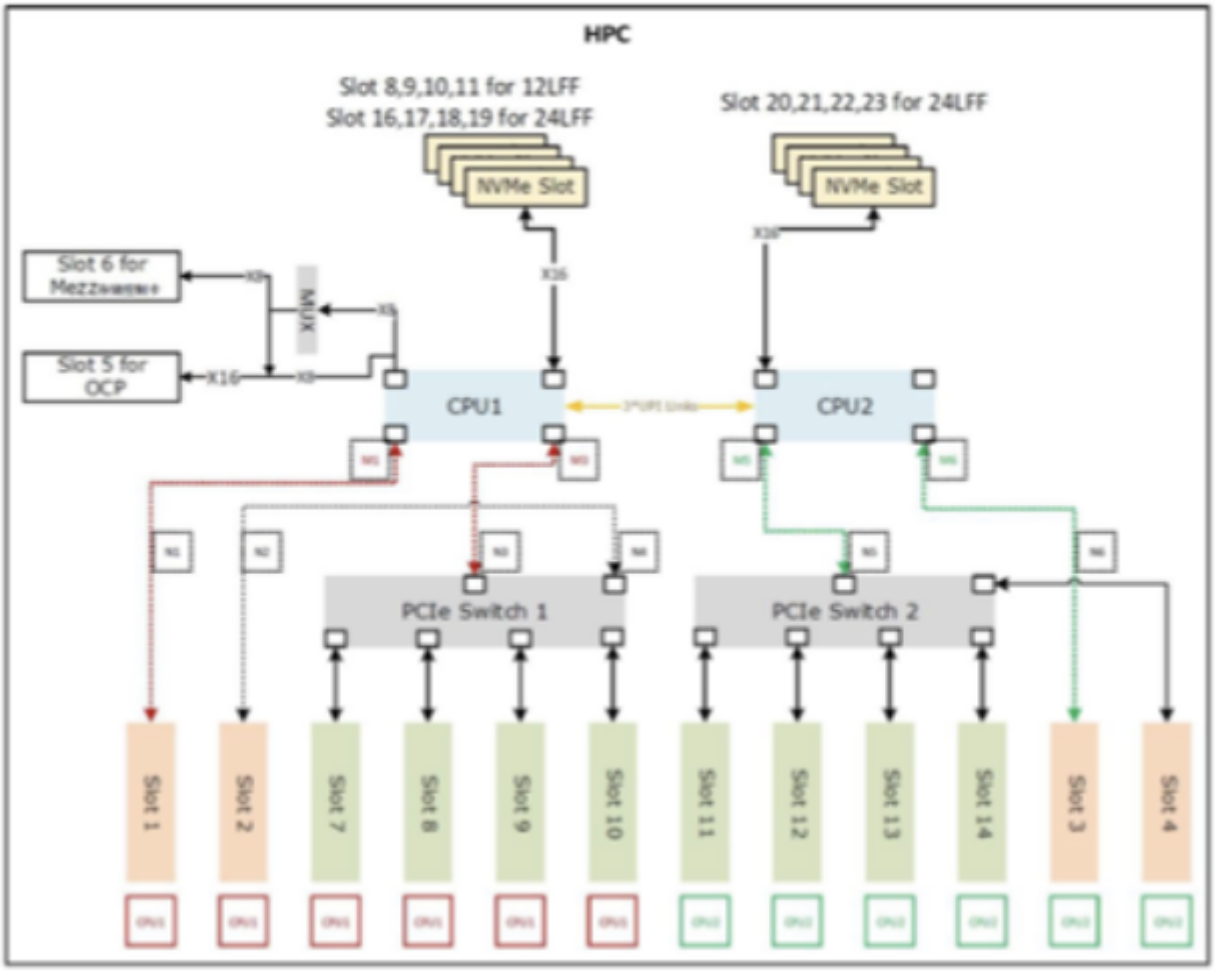
Two common GPU PCIe/NUMA topologies are found on standard servers, and you can verify which one you have with lstopo. Each topology favors different workloads; choosing the wrong one can degrade performance and increase end-to-end latency.



Profile A — GPUs concentrated on one NUMA node:

- Best for bare-metal jobs that keep CPU memory access local to that socket (e.g., single-tenant training).
- Risky for virtualized or multisocket CPU usage because cross-NUMA memory traffic to the GPUs adds latency.

Profile B — GPUs distributed across NUMA nodes:



- Better for virtualization and multi-VM scenarios where you can align each VM's vCPUs and memory with the GPU's NUMA node (NUMA affinity).
- Reduces cross-socket hops and improves latency consistency.

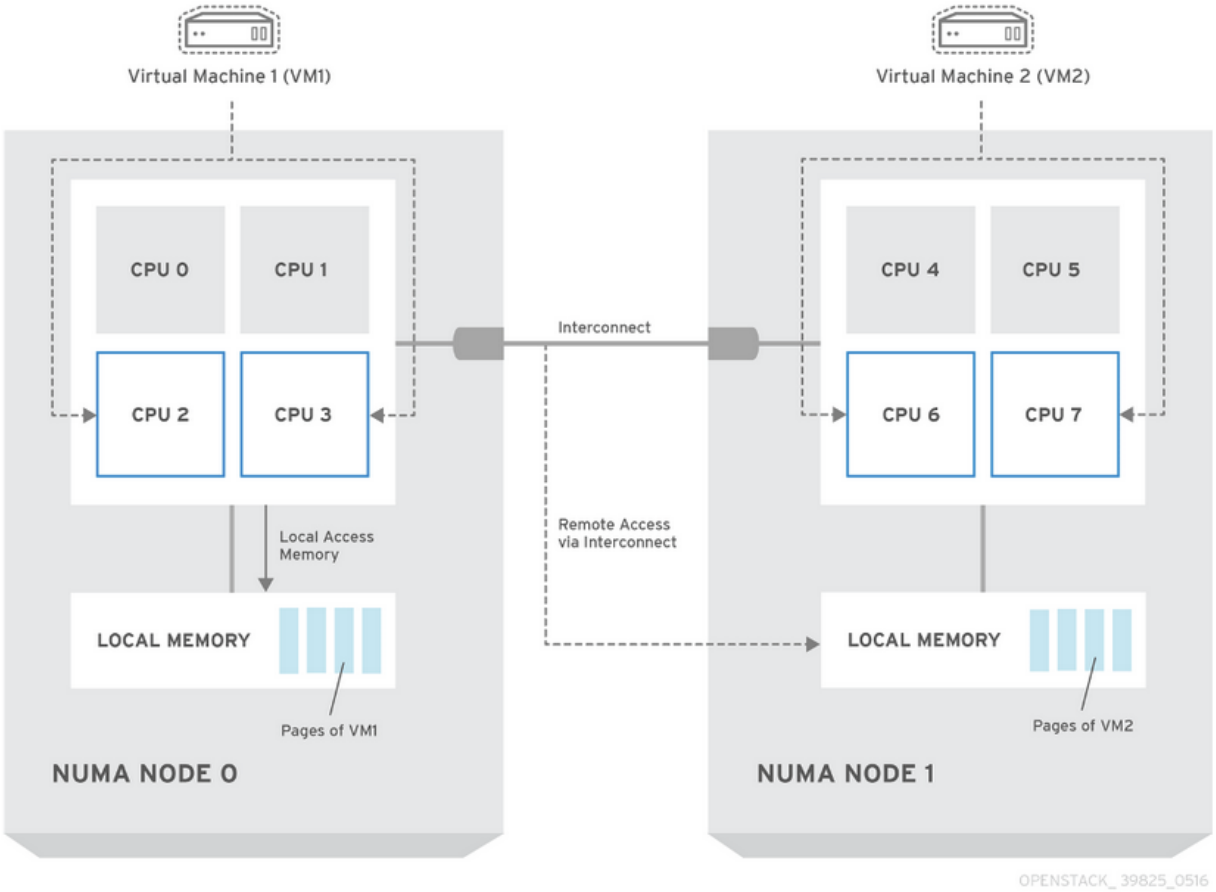
GPU-CPU affinity configuration. **GPU–CPU affinity configuration:**Aligning vCPUs with GPUs on the same NUMA node reduces latency when GPUs transfer data between memory, CPU, and GPU, resulting in improved VM performance.

hw:pci_numa_affinity_policy

Type: str

The NUMA affinity policy of any PCI passthrough devices or SR-IOV network interfaces attached to the instance. If **required**, only PCI devices from one of the host NUMA nodes the instance VCPUs are allocated from can be used by said instance. If **preferred**, any PCI device can be used, though preference will be given to those from the same NUMA node as the instance VCPUs. If **legacy** (default), behavior is as with **required** unless the PCI device does not support provide NUMA affinity information, in which case affinity is ignored. Only supported by the libvirt virt driver.

CPU Pinning technology:Improves the performance of GPU VMs by assigning dedicated CPU cores to each instance.



NVIDIA vGPU and MIG

Both NVIDIA's vGPU and Multi-Instance GPU (MIG) serve the same purpose of sharing GPU resources between multiple consumers, yet they are fundamentally different approaches. NVIDIA introduced vGPU more than a decade ago, whereas MIG was first introduced with the NVIDIA Ampere architecture in 2020. Let's see how those technologies compare.

The vGPU is a software-driven virtualization that allows fractional allocation of memory and

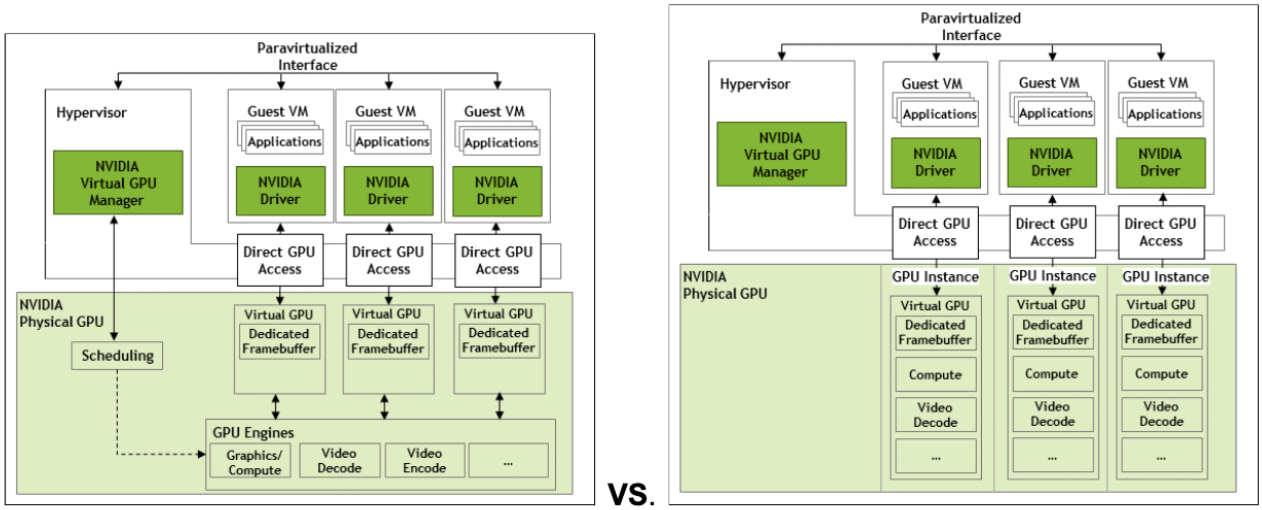
GPU features to the guest operating system. The vGPU allows time-sharing of GPU very similar to how hypervisors allow sharing of a CPU between different guest VMs with vCPU cores.

MIG, on the other hand, is a hardware-enforced partitioning mechanism that splits a GPU into several isolated, independent GPU instances with dedicated compute cores, memory and cache. This fundamental difference between the two approaches has practical implications.

MIG provides deterministic, no-noisy-neighbor isolation and predictable throughput because each MIG instance maps to the fixed hardware resources. It is ideal for latency-sensitive inference, throughput isolation, or dense workloads that one would encounter in today's multitenant cloud environments designed for AI and ML.

On the contrary, the vGPU is prone to noisy-neighbor problems due to time-sharing of the same hardware. Therefore, it is well suited for environments where workloads have variable resource demands, such as virtual desktop infrastructure (VDI) and general-purpose workloads like 3D or video rendering.

Schematically, the difference between the vGPU and MIG is illustrated in the diagram below:



It is worth noting that MIG is supported by high-end data center and professional GPU models such as the NVIDIA A30/A100/H100/H200 and RTX Pro 6000, whereas vGPU support can be found across a broader set of NVIDIA GPU models. The number of usable MIG instances also varies by GPU model and is limited to a maximum of seven per card. That means no more than seven users can share one MIG GPU. With vGPU, the number of consumers is determined by the profiles, which divide available GPU memory into fixed chunks that can be as small as 1GB or as large as the whole VRAM available, meaning vGPU potentially allows more users to share one GPU. More on the impact of the GPU sharing with vGPU can be found in the NVIDIA documentation. The table below acts as a summary of vGPU vs MIG vs GPU passthrough approaches and their use-cases.

	Virtual GPU (vGPU)	Multi-Instance GPU (MIG)	GPU Passthrough
GPU Access	One GPU shared among multiple VMs/containers	One GPU shared among multiple VMs/containers	One GPU dedicated to one VM. Multiple GPUs can be attached to one VM.
Performance	**Moderate to high**; depends on allocated profile and GPU load	**Moderate to high**; depends on MIG instance size	**Near-native**; full GPU performance
Scalability	**High**; supports multi-user, multi-VM environments	**Moderate to high**; supports multi-user, multi-VM environments; max 7 instances per one GPU	**Limited**; one GPU per VM
Isolation	**Lower**; VMs share GPU resources	**High**; hardware resources are dedicated to a single consumer	**High**; full GPU isolation per VM
Use Cases	VDI, rendering and non-latency sensitive generic workloads	AI/ML, latency-sensitive workloads	AI/ML, compute-heavy workloads
Licensing	**Required** (e.g., NVIDIA GRID license)	**Required** (e.g., NVIDIA AI Enterprise license) in case of MIG instances attached to VMs. **Not required** on bare metal or with containers.	**Not required**

AMD SR-IOV

AMD's MxGPU technology represents a hardware-based approach to GPU virtualization, enabling a single physical GPU to be partitioned and shared among multiple VMs. At the heart of this technology is the Single Root I/O Virtualization (SR-IOV), a standard developed by the PCI Special Interest Group (PCI-SIG), and the pivotal role of the AMD PF Host driver, also called the GIM (GPU-IOV Module) driver, which was recently open sourced on GitHub. MxGPU uses SR-IOV to create a "Virtual Function" (VF) from a single "Physical Function" (PF), which can then be safely virtualized in a VM. In the case of a multi-GPU system, it allows the usage of multiple interconnected GPUs with an xGMI bridge in a single VM.

The GIM driver is the host-side kernel driver that orchestrates the entire MxGPU virtualization process. It acts as the intermediary between the hypervisor and the physical GPU, managing

the creation and configuration of the VF.
On the guest side, there is a VF Driver, which is the component of the ROCm software stack installed within the VM's operating system. This driver enables the guest to recognize, communicate with, and efficiently use its assigned virtual GPU resources.

High Speed GPU Interconnections

- NVLink: NVIDIA NVLink is a high-bandwidth, low-latency interconnect that allows GPUs to communicate directly with one another and with CPUs. By bypassing the PCIe bus, NVLink provides significantly higher bandwidth and lower latency, enabling faster data transfer for AI training and inference.
- Infinity Fabric: AMD’s Infinity Fabric is a scalable interconnect architecture that links CPUs, GPUs, and other components. It provides high throughput and low latency, supporting efficient communication in heterogeneous compute environments. For AI workloads, Infinity Fabric helps optimize multi-GPU configurations, reducing bottlenecks during large-scale training.

Storage

AI workloads present complex storage challenges: massive datasets, high-throughput demands, and fault-tolerant architectures. Ceph, an open source, software-defined storage system, offers a unified, scalable, and fault-tolerant storage solution that integrates seamlessly with OpenStack, making it an optimal choice for AI-centric cloud deployments.

AI workloads typically span three phases. The data preparation phase requires high-throughput sequential reads and support for diverse data types, including structured datasets, multimedia files, and sensor data. The model training phase demands sustained high-bandwidth storage to feed GPU accelerators continuously and handle random data access patterns, while checkpointing functionality is critical for saving intermediate model states. The inference serving phase relies on low-latency access to trained models and real-time data processing, requiring the storage system to handle concurrent requests without performance degradation.

Scalability, performance, and efficient data management are critical for AI workloads. Storage systems must handle large datasets while maintaining linear performance as infrastructure scales.

Manila

It is common practice within the research industry to utilize shared filesystems for distributing tools, libraries, and datasets to enable further processing on compute entities; for storing and sharing results; or, more generally, for managing auxiliary files across different projects (namely OpenStack tenants).

For use cases that require shared, file-based access, traditional block or object storage options may not be ideal. In contrast, the Manila project has a strong track record of offering Network File System as a Service (NFSaaS), particularly—but not exclusively—when used in conjunction with Ceph as the backend storage for NFS shares.

The Manila project allows users to create, export, and mount file shares within tenant instances, or even share them across multiple tenants, in a manner similar to cloud services such as AWS Elastic File System (EFS).

Capabilities include:

- Role-based access control
- Access lists
- Layering
- Quota management
- Snapshot creation
- Support for a variety of backends, from traditional storage vendors (e.g., NetApp, Pure Storage) to open source solutions such as CephFS

To fully unleash the capabilities of Ceph at the storage layer, CephFS can be enabled to provide file-based access (in addition to the block and object types mentioned earlier), allowing Manila to use it as a backend. CephFS relies on Ceph Metadata Server (MDS) daemons to manage file and directory metadata, in conjunction with monitors (MONs) and object storage daemons (OSDs) as part of the Ceph cluster.

There are typically two integration options for connecting the Manila project with CephFS, and the choice between them depends on specific use cases, security requirements, and administrative preferences.

- CephFS native driver: instances connect directly to the public Ceph network using the CephFS client and protocol.
- CephFS-NFS driver: Manila uses the NFS-Ganesha service as a broker between the tenant network and the Ceph storage network. This enhances isolation but adds complexity. Clients access shares via the NFSv4 protocol.
- In summary, Manila can be an interesting alternative for many of the use cases described in this white paper.

Ceph

Ceph provides a unified storage model that consolidates block, object, and file storage under a single system, reducing complexity while supporting diverse AI data types. Built on the RADOS foundation, it ensures fault tolerance and scalability, making it ideal for modern AI infrastructure. Its high-performance interfaces include:

- RBD for persistent volumes with snapshots and thin provisioning
- RGW for S3-compatible object storage suited for training datasets and artifacts,
- CephFS for collaborative workflows and distributed training.

Ceph integrates deeply with OpenStack to simplify AI deployments. Cinder and Nova enable efficient VM storage and persistent volumes for AI pipelines. Glance and Swift facilitate the rapid deployment of pre-configured AI environments and scalable object storage. Keystone integration ensures secure, multitenant environments, allowing isolated AI projects to coexist efficiently.

For optimal AI performance, Ceph’s BlueStore engine delivers higher IOPS and lower latency,

essential for training and inference workloads. Erasure coding offers a balance between storage efficiency and performance, while NVMe SSDs and high-speed networks are critical for meeting the high-throughput demands of GPU-based workloads.

Ceph's flexibility, open source foundation, and active development community make it well suited for evolving AI technologies like GPU Direct Storage and containerized AI workflows. Organizations adopting Ceph gain vendor independence, scalability, and the performance needed for next-generation AI applications.

Networking

AI is often described as a data-centric discipline, and the efficient movement of large volumes of data also presents a networking challenge. Networking performance is critical to ensuring that AI workloads run efficiently.

Every aspect of open infrastructure presents the system architect with choices to consider, and networking is an area where the right decision can have the greatest impact.

A wide range of options available for networking form a continuum from convenience to performance:

- Standard paravirtualised networking in OpenStack can be optimised for aggregate throughput by enabling options such as multiqueueing in the hypervisor virtualisation engine.
- OpenStack supports high-performance Ethernet networking through technologies such as SR-IOV and Open vSwitch hardware offloads, creating a thinner virtualisation layer by implementing network functions in hardware.
- Beyond Ethernet networking, OpenStack supports the integration of InfiniBand networking, including multitenant network isolation through InfiniBand partition keys.
- For the highest performance demands, OpenStack can provision bare-metal compute infrastructure, removing compute virtualisation overhead entirely.

Optimising Paravirtualised Networking

Networking performance can often be improved without relying on hardware solutions.

- Increase Maximum Transmission Unit (MTU) sizes, such as 9,000-byte "jumbo frames." Paravirtualised software networking is often limited by packet processing rates. Larger Ethernet frames reduce the packet processing rate for a given quantity of data.
- Enable VIF multiqueue in the hypervisor networking stack. With highly concurrent network connectivity, the aggregate bandwidth can be increased by dedicating multiple hypervisor CPU cores to packet processing for the network data path. VIF multiqueue is enabled via the image property hw_vif_multiqueue_enabled (read more).
- Streamline data paths by using provider VLANs for high-speed access to external storage. In some circumstances, internal networks can also be configured with stripped-down functionality, such as disabling port security, to reduce overhead.

Hardware-Offloaded Ethernet

OpenStack networking can harness the power of hardware-offloaded Ethernet.

For many years, single root IO virtualisation (SR-IOV) has provided VMs with access to high-performance networking. Conventional SR-IOV supports basic functionality, such as multitenant VLAN network isolation, but typically does not support important advanced functionality such as port security, fault tolerance or live migration.

Recent innovations in high-end network adapters now allow:

- Single SR-IOV ports from a bonded pair of physical interfaces, transparently providing VMs with a redundant network connection capable of using both links in an active-active fashion.
- Live migration support for VMs with high-performance network connectivity, allowing workloads to move to another hypervisor (for example, when hypervisor maintenance is necessary).
- RDMA over Converged Ethernet (RoCE) is a networking protocol derived from high-performance computing (HPC). RoCE underpins various protocols used for data transfer and communication, including for distributed AI use cases and high-performance storage. RoCE implements native InfiniBand protocols in an Ethernet network, but introduces a small degree of performance overhead compared with InfiniBand itself.
- Hardware offload of network flow rules. In conventional OpenStack networking, flow rules are the low-level mechanism for configuring core networking components such as Open vSwitch. Flow rules define a wide range of virtual network functionality, including port security, floating IPs and load-balancers.

These functionalities are only available when using high-end network interfaces equipped with this capability.

InfiniBand in OpenStack

InfiniBand networks are frequently included in high-end AI infrastructure for maximum performance. InfiniBand differs from Ethernet in several key ways.

- InfiniBand can support IPv4 addressing and routing, but its underlying architecture is substantially different from Ethernet. Configuration is managed centrally by an overseeing control process known as the subnet manager. (Note: a subnet in InfiniBand terminology is quite different from the concept in Ethernet-based networks.)
- InfiniBand uses partitions to enable logical network separation, similar to Ethernet-based VLANs. Multitenant network isolation is implemented using InfiniBand partition keys.
- The subnet manager is a software service that can be deployed alongside OpenStack control plane services. OpenSM is a free and open source subnet manager that provides basic capabilities. NVIDIA maintains a fork of OpenSM that supports virtualisation and SR-IOV functionality. NVIDIA Unified Fabric Manager (UFM) is an alternative commercial solution with enterprise capabilities. Dynamic support for multitenant network isolation requires features implemented in UFM.

- OpenStack supports InfiniBand networking through the networking-mlnx Neutron driver. In OpenStack terms, it extends both Nova and Neutron to fully support InfiniBand virtual interfaces (VIFs)

In more detailed, practical terms, InfiniBand in OpenStack

- Allows for the definition of InfiniBand-based physical networks within OpenStack.
- Handles GUID assignment to VFs.
- Equates GUID to MAC addresses from Neutron.
- Offers a DHCP server for IPoIB.
- Maps the VLAN provider-segment to an InfiniBand-compliant partition definition.
- Assigns the appropriate GUID to the correct partition.
- Allows attaching IB-based VIF to VMs.

In cloud computing, there are frequent assumptions that overlook InfiniBand functionality, which must be taken into account when implementing an InfiniBand-enabled cloud infrastructure:

- Default builds of cloud software images lack even basic drivers for InfiniBand.
- In some Linux distributions, cloud-init does not support InfiniBand network devices.
- In many cases, instances are expected to have a primary Ethernet interface. A purely InfiniBand system is likely to test many assumptions made about Linux networking.
- The use of ConfigDrive to provide cloud-init metadata is recommended.

Once InfiniBand is deployed and configured, it will deliver maximum performance for AI workloads in a virtualised environment.

Networking for Bare-Metal Infrastructure

To meet the ultimate requirements for performance, OpenStack supports bare-metal compute and networking. OpenStack Ironic presents physical hardware as VMs and physical network ports as virtual interfaces. Many of the abstractions of software-defined infrastructure have equivalents in bare-metal environments.

- Multitenant network isolation for Ethernet VLAN networks can be implemented using the networking-generic-switch Neutron driver, which supports a range of popular network switch vendors.
- The driver can also support bonded network ports and VLAN-tagged trunk interfaces.
- InfiniBand networking is supported for bare-metal compute alongside virtualised compute, using the same networking-mlnx Neutron driver.

Bare-metal networking is less flexible than its virtual counterpart and cannot always provide the same functionality. For example, port security cannot easily be implemented in a bare-metal context. As a result, bare-metal compute resources are frequently isolated from direct external access and paired with virtualised compute infrastructure as a gateway for controlling access to bare-metal compute.

Neutron

Artificial Intelligence (AI) workloads demand substantial compute resources, typically distributed across large clusters. Consequently, the networks connecting these servers face stringent requirements. To meet them, networking must provide high bandwidth, low latency, and lossless data transmission.

The OpenStack networking service, Neutron, addresses these challenges through several key features:

- **Quality of Service (QoS):**Configures policies that guarantee bandwidth and/or packets per second (PPS) for network ports. Neutron works together with the Placement API service to enforce these resource guarantees, ensuring that VMs receive the predictable network performance they require.
- **SR-IOV Integration:**Allows direct attachment of physical network device functions to VMs. This approach bypasses the virtual switch, providing VMs with native, hardware-level networking performance and significantly reduced latency and CPU overhead.
- **DPDK-Accelerated Virtual Switching:**Utilizes plugins such as networking-dpdk to leverage the Data Plane Development Kit (DPDK) framework. This delivers high-performance, user-space packet processing, dramatically accelerating data throughput for VMs engaged in network-intensive tasks.
- **Stateless Security Groups:**Offer a performance-enhanced alternative for packet filtering. In scenarios where network security is essential but the overhead of stateful connection tracking is undesirable, stateless security groups can significantly improve network throughput by simplifying firewall rule processing.

Metrics Collection

Considerations for Metrics Collection

GPU workloads introduce new monitoring requirements in addition to standard platform metrics. Vendor-supported agents (such as NVIDIA’s dcgm-exporter) report detailed information on GPU performance, power use, and utilization.. Administrators must decide what boundaries to monitor and whether the chosen tools support the intended model.

With full-device PCI passthrough, vendor agents cannot run at the platform level because the GPU is fully assigned to the guest VM. In this case, the agent must run inside the VM. By contrast, when GPUs are virtualized using vGPU, MIG, or SR-IOV, administrators can run monitoring agents either on the compute node or inside the VM, depending on requirements.

Service Models for AI Workloads

For OpenStack-based private clouds, one of the emerging challenges is delivering large language model (LLM) inference as a first-class service within the platform. While OpenStack has long provided virtualization, storage, and networking primitives for traditional workloads, AI inference introduces a new set of requirements: fine-grained GPU memory management, low-latency request serving, and efficient utilization of high-cost accelerator resources.

This is where vLLM becomes highly relevant. By acting as a high-performance inference runtime that can be deployed on top of OpenStack's compute and accelerator infrastructure, vLLM transforms raw GPU capacity into an optimized service layer, enabling operators to run inference workloads with cloud-native efficiency.

At the core of vLLM is PagedAttention, an innovation that directly addresses GPU memory inefficiencies inherent in transformer-based models. Traditional inference servers in OpenStack deployments often rely on static KV cache allocations that lead to memory fragmentation and poor GPU utilization, especially in multitenant environments. PagedAttention introduces a virtual memory abstraction for the KV cache, organizing it into fixed-size pages that can be allocated, evicted, and remapped on demand. This approach allows multiple tenants, each running requests of different sequence lengths, to share GPU resources without wasting memory. For OpenStack operators, this means higher consolidation ratios (more concurrent requests per GPU), better tenant fairness, and lower TCO for AI services offered within the cloud.

Another core strength of vLLM is continuous batching, which aligns directly with the multitenant scheduling challenges faced in OpenStack environments. Traditional static batching forces workloads to wait until enough requests arrive to form a batch, leading to underutilized GPUs and unpredictable latencies. vLLM's continuous batching engine dynamically merges new requests into already running execution graphs, achieving near-optimal GPU saturation under highly variable tenant traffic patterns. This ensures that OpenStack deployments can handle diverse inference workloads — from bursty real-time chatbot requests to large batch analytics jobs — while maintaining predictable latency.

Operationally, vLLM provides an OpenAI-compatible API that can be exposed as an OpenStack service endpoint, making it straightforward for tenants to consume LLM inference through familiar interfaces. It supports Hugging Face models and LoRA fine-tuned variants out-of-the-box, and scales horizontally across OpenStack clusters using distributed inference and tensor parallelism. Combined with OpenStack's orchestration tools, operators can deliver LLM-as-a-Service offerings that are elastic, multitenant aware, and backed by enterprise-grade SLAs. vLLM enables OpenStack to extend beyond infrastructure-as-a-service into AI inference-as-a-service, bridging the gap between GPU hardware availability and production-grade model serving. Its innovations — PagedAttention for memory efficiency, continuous batching for scheduling efficiency, and an API surface aligned with developer expectations — make it a natural fit for OpenStack operators seeking to modernize their cloud platforms for the AI era.

vLLM and the OpenAI-Compatible API

One of the most practical advantages of vLLM for model service platforms is its support for an OpenAI-compatible API layer. This feature allows organizations to expose LLM inference through the same API contract widely adopted in the industry, without requiring developers or applications to change their existing integration logic. By mirroring the OpenAI API schema for endpoints such as /completions, /chat/completions, and /embeddings, vLLM enables seamless adoption for applications already built against commercial LLM providers.

From a service architecture perspective, this compatibility provides two significant benefits:

1. Lower migration barriers. Enterprises can move workloads from proprietary SaaS environments to private or hybrid clouds. Applications previously locked into OpenAI's hosted APIs can redirect traffic to a vLLM-backed endpoint within an enterprise-controlled environment, such as an OpenStack cluster, with minimal configuration changes.
2. Multicloud flexibility. The same application can run against either external APIs or local vLLM deployments, depending on cost, latency, or compliance requirements.

Technically, vLLM's API layer is tightly integrated with its runtime optimizations, such as PagedAttention and continuous batching. This means that requests arriving at the /chat/completions endpoint are automatically scheduled into the continuous batching engine, ensuring high GPU utilization even under spiky or heterogeneous workloads. Token streaming is supported out of the box, enabling real-time applications such as chatbots, copilots, or agent frameworks to consume outputs incrementally without custom plumbing. Furthermore, vLLM supports fine-tuned models and adapters (e.g., LoRA-based variants) under the same API surface, giving operators the flexibility to expose both foundation models and domain-specialized versions to end users without API fragmentation.

Operationally, the OpenAI-compatible API enables service standardization. Monitoring, logging, and quota enforcement can be applied consistently across tenants, regardless of whether they are consuming commercial APIs or private vLLM endpoints. For OpenStack or Kubernetes-based environments, this API can be fronted with standard service components such as Ingress controllers, allowing it to scale horizontally and support enterprise-grade SLAs.

The OpenAI-compatible API in vLLM bridges the gap between inference optimization and developer adoption, ensuring that organizations can deploy LLM services in their own infrastructure without sacrificing compatibility or performance. It turns vLLM from a research-optimized runtime into a production-ready, developer-friendly platform for LLM-as-a-Service.

Hybrid Decentralized Edge AI Clouds

AI's future demands more than centralized clouds. Real-time applications, such as in autonomous systems or smart factories, require decisions to be made in microseconds, near the data source. This leads to the Hybrid Decentralized Edge AI Cloud model, which merges a central OpenStack cloud with a constellation of smaller, high-speed edge servers.

This architecture leverages a network of smaller servers, deployed closer to the data, to act on reactive working data (RWD) instantly. The central OpenStack cloud is the "brain," training large foundational models and pushing updates. The edge servers are the "limbs," performing low-latency inference.

Core Components & OpenStack Integration

- Micro-Servers at the Edge: Small, resource-efficient servers handle inference locally. OpenStack's Nova and Cyborg can provision lightweight containers and manage specialized edge accelerators (like NVIDIA Jetson) on these nodes. In addition, micro-servers at the edge can also reduce hallucination rates in smaller language models. With

that decrease, the edge is the right place to have small, purpose-driven agents to make reliable, near real-time decisions without relying on central servers. An example is a manufacturing use case where lower latency leads to more reliable, near real-time decision-making on the floor of the facility. Micro servers at the edge enable lighter-weight agents on the manufacturing floor to handle tasks like monitoring and optimization without needing to contact a central server for decisions.
- Microsecond Connectivity: Ultra-low-latency networking is essential. Neutron can be optimized for secure, direct communication between the central cloud and edge nodes, bypassing traditional network hops.
- Decentralized Storage: Instead of sending all data to the cloud, Ceph can provide local, resilient storage at the edge. Only small, summarized data is sent back to the central cloud for model refinement. But when needed, data can be transported in S3 format for ease of integration and manipulation. Additionally, LF Edge projects such as EdgeLake are building on the foundation layers of OpenInfra to leverage all three of these points to significantly enhance AI Inference modeling updates.

This hybrid approach unlocks real-time intelligence for business, academic, and operational purposes, enabling everything from instant quality control on a factory floor to immediate analysis of scientific data, all while leveraging OpenStack's flexibility and open source foundation.

Keystone

In AI workloads, Keystone—the identity and access management (IAM) service—acts as the security backbone, ensuring that only authorized entities (people, AI agents, or services) can access specific resources in a secure and auditable way. It's the foundation for securing the entire AI pipeline from data to deployment.

Securing Data and Models

AI workloads rely on two invaluable assets: data for training and the trained models themselves.

- **Data Access Control:**Training data can be highly sensitive (e.g., personal user data, financial records). Keystone defines precisely who or what can read, write, or delete this data. This prevents unauthorized access and data breaches. For example, an AI training service might have read-only access to a specific data storage bucket but no permission to delete it.
- **Model Protection:**A trained AI model is a significant piece of intellectual property. Keystone may be used to control who can access the model files, deploy them, or call the model's API for predictions (inference). This reduces the risk of model theft or tampering

Controlling Access to Infrastructure

AI training and inference often require powerful and expensive compute resources like GPUs and TPUs.

- **Cost and Resource Management:**Keystone ensures that only authorized data scientists or automated services can spin up or shut down these costly resources. This

prevents accidental resource deletion and helps control cloud spending by restricting who can provision infrastructure.
- **Environment Segregation:**Keystone is used to create strict boundaries between development, testing, and production environments. This ensures that an experiment in a development environment can't accidentally disrupt the live production model.

Enabling Secure Automation

Modern AI workflows are highly automated. AI agents and CI/CD pipelines perform tasks like data processing, model training, and deployment without human intervention.

- Service Accounts and Roles: Keystone supports multiple concepts of authentication and authorization. Authorization is based on the RBAC concept with a set of roles defining the allowed actions. For automation processes, Keystone provides a concept of application credentials. These are special non-human identities with very specific, granular roles that may vary from the general roles granted to the account and may be time limited. Work is in progress to support ephemeral service accounts.
- Example: An automated training script running via a service account might be granted permission to:

    1. Read data from data-bucket-A.
    2. Start a training job on a GPU cluster.
    3. Write the final model to model-registry-B.

    It would have no other permissions, meaning it couldn't access other data or services, providing a very secure and limited scope of operation.

Ensuring Compliance and Auditing

For many industries, tracking who accessed what and when is a legal or regulatory requirement (e.g., GDPR, HIPAA).

- **Audit Trails:**Keystone provides a comprehensive log of every action taken, creating a clear audit trail. If a data breach occurs or a model behaves unexpectedly, these logs are crucial for investigating the cause.
- **Policy Enforcement:**Keystone allows organizations to enforce company-wide security policies, ensuring the entire AI workload adheres to established compliance standards.

Horizon

OpenStack Horizon supports AI workloads by providing a unified web dashboard to manage the fundamental cloud infrastructure that powers them. It simplifies the process of provisioning and managing computing, storage, and networking resources without requiring complex command-line tools.

Key Areas of Support

- **Compute Power:**Through Nova, Horizon lets users launch and manage VMs with the necessary hardware. This includes selecting VM flavors that are configured with GPUs or other hardware accelerators essential for training large AI models.

- **Scalable Storage:**AI models require vast amounts of data. Horizon provides an interface to provision different types of storage for this data, including:
- **Block Storage (Cinder):**Ideal for an operating system and for datasets that need fast, direct access for a single instance.
- **Object Storage (Swift):**Used for large, unstructured datasets that can be accessed by multiple instances, such as a large collection of training images.
- **Network Management:**Horizon simplifies network configuration for AI environments. Users can easily create private networks to isolate their workloads, manage floating IPs for external access, and configure security groups to control traffic, ensuring a secure and well-organized environment.
- **Reproducibility:**To ensure AI experiments are consistent, developers can create custom images of pre-configured VMs with the necessary libraries. Horizon makes it easy to upload and use these images to quickly spin up identical environments for testing or team collaboration.
- R**esource and User Management:**For teams sharing a cloud, Horizon's Keystone service allows administrators to manage users, assign roles, and set quotas. This is crucial for controlling resource allocation and managing costs effectively across multiple projects and teams.

In short, OpenStack Horizon makes the underlying cloud infrastructure easy to use, allowing data scientists and developers to focus on their AI projects rather than on infrastructure management.

Growing AI Support in OpenStack

As AI usage expands, OpenStack will continue to evolve to meet new requirements. This progress depends on community collaboration, particularly feedback from AI users and cloud operators.

One key way that collaboration happens is through the OpenInfra AI Working Group. The OpenInfra AI WG’s goal is to surface use cases and enhance ways in which OpenInfra projects support AI workloads. The group is open to the OpenInfra community to ensure all perspectives are represented. Meetings are held periodically and typically focus on case study presentations and collaborative projects such as white papers. To get involved, subscribe to the mailing list.

Come Shape the Future of AI on OpenStack!

Production Case Studies & Reference Architectures

China Mobile

China Mobile Cloud (ECloud): Empowering Enterprise AI Innovation and Applications with Robust Computing Power

As a gold-level member of the OpenInfra Foundation and the cloud computing brand under CMCC(China Mobile Communications Group Co.,Ltd), ECloud is committed to providing secure, reliable, and high-performance cloud and AI solutions for global enterprises. Leveraging CMCC's robust network resources and proprietary technologies, ECloud actively supports the global digital economy and empowers businesses to achieve intelligent transformation. Amidst the surging wave of artificial intelligence (AI) development, ECloud delivers robust and reliable support for diverse AI workloads through its global infrastructure, multi-computing architecture, and dual technological underpinnings of OpenStack and COCA computing-native platforms.

Driving the intelligent transformation of education and industry with world-leading computing power

ECloud has deployed computing resources both domestically and internationally to ensure low-latency, highly reliable network connectivity required for AI applications. ECloud has established a “4+N+31+X” distributed computing architecture within China, with a total computing capacity of 20 EFlops. ECloud achieved unified management and intelligent scheduling of CPU, DPU, and GPU computing architectures, providing robust heterogeneous computing power support for AI training and inference. ECloud has launched Public Cloud, Private Cloud, and Edge Cloud platforms in international markets such as Germany and Pakistan, and completed the global rollout of its full series of “Public Cloud + Private Cloud + Edge Cloud” products. ECloud has independently developed an AI-powered multilingual management platform with unified architecture, agile delivery, lightweight deployment, and operational capabilities.

Case 1: Clustered Computing Power helps a province in China build a large model for the AI industry

ECloud has signed a strategic agreement with partners to leverage local clean energy and climate advantages in jointly developing large-model AI applications. The project employs high-performance bare-metal server clusters to deliver ultimate performance support for training and inference of AI large models, successfully advancing the implementation of artificial intelligence technologies in smart industry, smart healthcare, cultural tourism, and other sectors. ECloud has emerged as a new engine driving the local digital economy.

Case 2: Dedicated Computing Power Empowers Chinese Energy Manufacturing Firm to Launch Digital Intelligence Strategy Plan

ECloud helps clients launch digital and intelligent strategic initiatives by building large-model platforms with applications and platform capabilities. To support these intelligent computing scenarios and satisfy rapid computing power delivery demands, ECloud delivers resources through a hybrid approach combining public cloud resources with dedicated computing power

via a “lease instead of build” model. This serves as a benchmark case within China's energy and chemical sectors, empowering clients to implement their “Digital and Intelligent China” strategy.

Case 3: Contributed to the establishment of a certain overseas cloud

ECloud has facilitated the implementation of a certain overseas cloud, providing crucial infrastructure support for the digital transformation and AI technology development of that region. The public cloud brand of this location is built with a dual-region local pool, achieving a dual-site layout in one place, and has been selected as a typical case of digital economy cooperation between China and the SCO countries.

ECloud's AI Technology Advantages

As enterprises actively embrace artificial intelligence technologies, commonly encountered challenges include limited computing resources, data security risks, and complex technical implementation. ECloud builds upon its self-developed COCA computing power native platform to establish comprehensive capabilities spanning three dimensions: computing power, security, and applications. This helps enterprises achieve seamless AI transformation in one go.

At the level of computing power, ECloud leverages its Pan Shi servers to achieve compatibility with major CPU platforms and mainstream heterogeneous chips such as NVIDIA . This enables true “one-cloud-fits-all” capabilities across diverse scenarios from AI inference to 4K rendering. Its proprietary DPU chip further accelerates performance through hardware-level virtualization, boosting sixth-generation cloud host performance by up to 80%. General-purpose computing products deliver exceptional cost-effectiveness with a 55% reduction in per-core costs, significantly lowering the barrier for SMEs to deploy AI applications. This enables traditional IT budgets to unlock extraordinary computing power.

Leveraging the COCA computing power native architecture, ECloud supporting sub-second loading and efficient operation of trillion-parameter models including DeepSeek-R1, enabling zero-cost migration across GPU ecosystems. Combined with ECloud's AI data lake and ultra-fast file storage, enterprises can efficiently process multimodal data and train models, rapidly seizing the high ground in AI innovation. This truly achieves “deep resonance between computing power and AI,” reshaping the new infrastructure for the era of large models.

At the scheduling layer, ECloud leverages its nationwide computing network infrastructure and open-source OpenStack foundation to innovate the Grand Cloud Hybrid Elastic Computing System and Computing Network Brain. Utilizing AI algorithms to real-time sense business demands and resource statuses, it dynamically recommends optimal resource combinations. This enables minute-level rapid delivery of optimal computing power at a scale of thousands of machines, making intelligent computing power as readily available and on-demand as water or electricity.In terms of security and reliability, ECloud host service delivers an SLA of up to 99.995%. It builds a comprehensive health view of computing power across the entire chain, enabling fault prediction and seamless recovery, while supporting full-machine backup and cross-domain disaster recovery, it provides all-around protection for highly sensitive scenarios

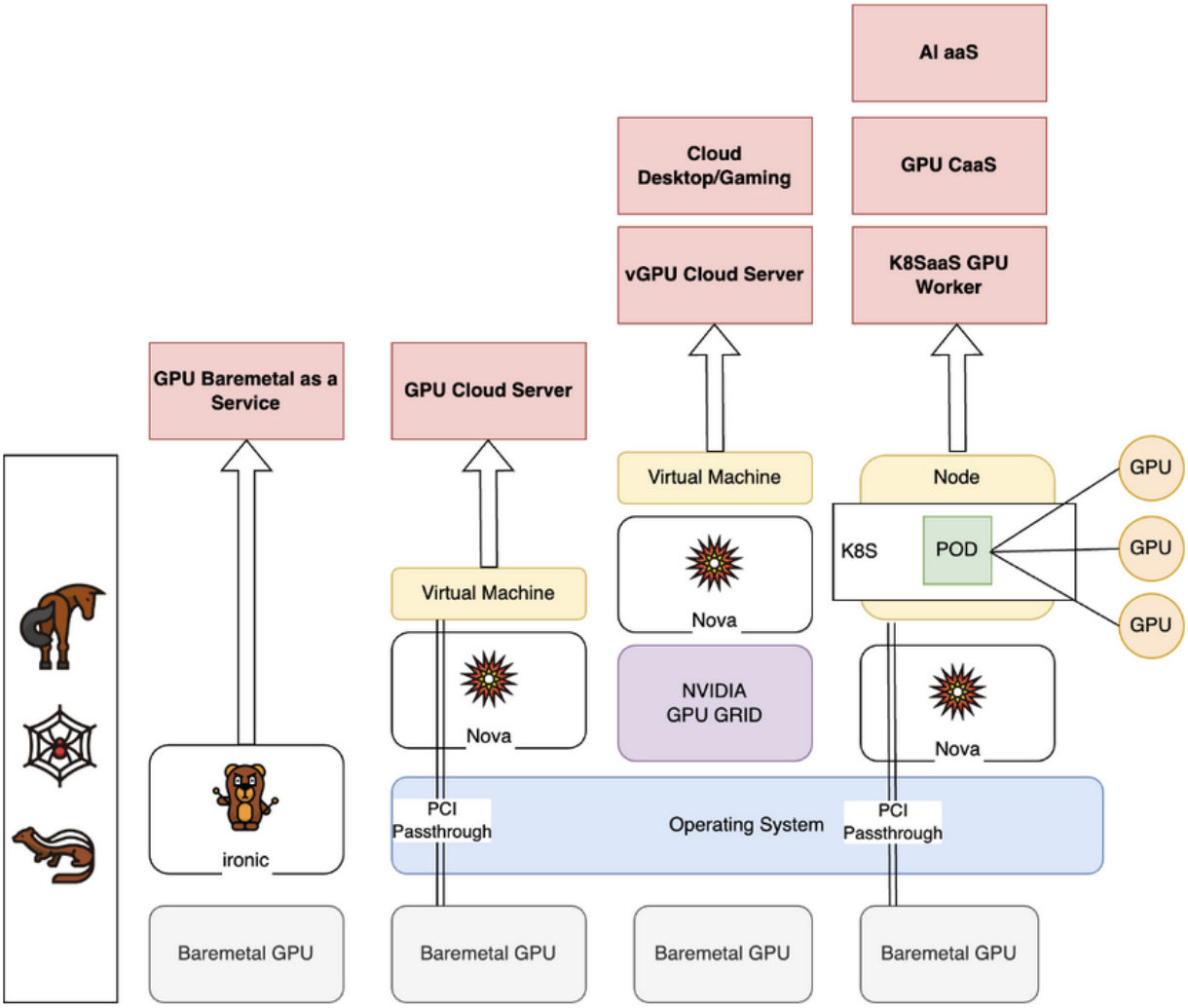
like government affairs and finance, ensuring business continuity and data security in the AI era.

Currently, ECloud ranks among the top contributors in major open-source communities such as CNOCS, Linux, and Kube-OVN, demonstrating its deep technical expertise and commitment to open collaboration and innovation. Committed to advancing the intelligent era, ECloud leverages globally leading computing infrastructure, open and compatible multi-architecture systems to provide enterprises with full-stack AI support—from training to inference, and from chips to frameworks. Whether building distributed computing networks domestically or advancing multi-cloud strategies overseas, ECloud remains committed to its vision of "Computing power is everywhere, intelligence is on demand," helping industries across all sectors embrace artificial intelligence efficiently, securely, and cost-effectively.

ECloud aspires to become a world-class cloud service provider, continuously enhancing its technological capabilities like COCA platform and Pan Shi servers, while expanding its global computing power network and ecosystem collaboration advantages. ECloud will collaborate with more enterprises and partners to jointly advance the large-scale implementation of AI technology and drive industrial intelligent upgrades.

Choosing ECloud is choosing a comprehensive AI transformation strategy, and selecting a high-performance cloud service partner. Let us join hands with ECloud to unleash innovation potential through powerful computing capabilities and jointly embrace a smart future.

FPT Smart Cloud



FPT Smart Cloud offers a host of customizable OpenStack services to its users through the AI Factory:

- GPU H100/H200 Bare metal as-a-service, powered by OpenStack Ironic
- GPU/vGPU cloud instances via PCI-Passthrough & SRIOV technology
- Cloud workstation/desktop using OpenStack Nova
- GPU Kubernetes engine provisioning using OpenStack Magnum
- GPU container as-a-service on top of GPU K8S Engine with NVIDIA MIG technology
- Self-service capabilities with different card lines or multiple cards dedicated to a VM in one cluster
- OpenStack add-on, ready-to-use services including load balancing (Octavia), GPU VM Auto-scaling (Senlin), and storage backup (Cinder)
- FPT Cloud Desktop accelerates with GPU by integrating OpenStack with OpenUDS

- When it comes to the infrastructure technology backing these use cases, FPT Smart Cloud adopted open source infrastructure software, specifically OpenStack, for several reasons:
- Flexibility for customization
- A broad set of tools available to support hardware optimization/acceleration & offload (NUMA, SRIOV, CPU PIN, Multi Queue VIF)
- Mature cloud ecosystem for AI applications (VMs, Storage, Autoscaling, Automation, Loadbalancer, Kubernetes provisioning form)
- Variety of models of supporting GPU workloads (VM PCI passthrough, vGPU, MIG)

OpenStack provides unmatched flexibility and customization for AI workloads, particularly at large scale. It offers a wide range of options to accelerate GPU performance and optimize infrastructure for demanding AI tasks. Unlike traditional, closed-source platforms such as VMware, OpenStack empowers users with open innovation, community-driven development, and deep integration capabilities tailored to AI needs.

As AI workloads and technologies evolve rapidly, open source platforms like OpenStack enable faster adoption and shorter time-to-market compared to proprietary enterprise cloud solutions. The openness of the ecosystem, combined with strong hardware compatibility and scalable architecture, makes OpenStack a more powerful and future-ready foundation for AI infrastructure.

	OpenStack	VMware Cloud Director
Provision VM	PCI Passthrough vGPU	vGPU
Life cycle management	Resize Cold-migration Evacuate	Fix on initial host
Multi GPU per VM	Supported with flavor	Complex
vGPU live migration	Supported (from Caracal)	Supported
Multi model/vendor	Supported with alias	Complex
GPU opt.	Numa, Hugepages, CPU Pin	Lack of support
Network Opt.	SRIOV, DPDK, HW NIC	Lack of support
Storage Opt.	Ephemeral local NVME/SSD	Lack of support
Add-on service	Magum, Zun, Senlin	N/A
Instance Live-Resize	Unsupported	Supported

Rackspace

OpenStack plays a foundational role in Rackspace Technology's AI strategy, serving as the flexible, open infrastructure layer that makes AI innovation possible. Through FAIR (the Foundry for AI at Rackspace), we've developed a methodology for responsibly and effectively harnessing AI innovation. By combining FAIR's framework with the scalability and openness of OpenStack, Rackspace empowers customers to build, deploy, and scale AI workloads securely and efficiently.

GPU Enablement

Rackspace is utilizing OpenStack to deliver GPU-enabled solutions in a variety of ways:

GPU-passthrough instances are available in both our OpenStack Flexpublic cloud as well as our OpenStack Businesshybrid-cloud options. Building on top of this functionality, Rackspace Spotprovides managed GPU-enabled Kubernetes clusters that are ideal for containerized AI workloads.

The combination of Rackspace Spot and OpenStack Flex provides access to several GPU types:

- A30
- H100
- P40

Rackspace's fully-private OpenStack Enterpriseclouds can also be deployed with GPUs for on-premise AI workloads that can be co-located with private data for in-house development that do not require any internet round-trip data transfers.

Soon, Rackspace Spot Enterprise will enable the same managed Kubernetes cluster capabilities for on-premise deployments on top of OpenStack to enable containerized AI GPU workloads.

AI-Specific Products

Rackspace Private Cloud AIhelps harness the power of AI in a secure environment, leveraging a full AI software stack and the latest AI-optimized hardware.
OpenStack is a keys part of how AI value is delivered to our customers.

OpenStack is used for the control plane of our on-premise Rackspace AI Anywheresolution.

AI-Enabled OpenStack

Currently in development, Rackspace is investigating ways to leverage Agentic AI for operational benefit and control of cloud deployments. Through the combination of the large body of cloud data we have accumulated, our many years of operational experience, and the 100% API-driven nature of OpenStack, Rackspace aims to harness the benefits that AI provides for insights and agency to simplify and automate OpenStack deployments in the future.

StackHPC - 6G AI Sweden

Infrastructure competitive for AI workloads is some of the most performance-intensive available in the market today. Harnessing the full potential of such infrastructure is critical to maximising returns on very significant hardware and software investments.

6G AI Swedenhas the ambition to provide Swedish companies with world-class AI capabilities, whilst maintaining absolute data sovereignty. To realise this ambition, 6G AI Sweden selected a technology stack centred around OpenStack, Kubernetes and open infrastructure solutions, designed and deployed in partnership with StackHPC.

Hardware Infrastructure

Compute

6G AI Sweden's AI compute nodes are based on the NVIDIA "HGX" reference architecture:

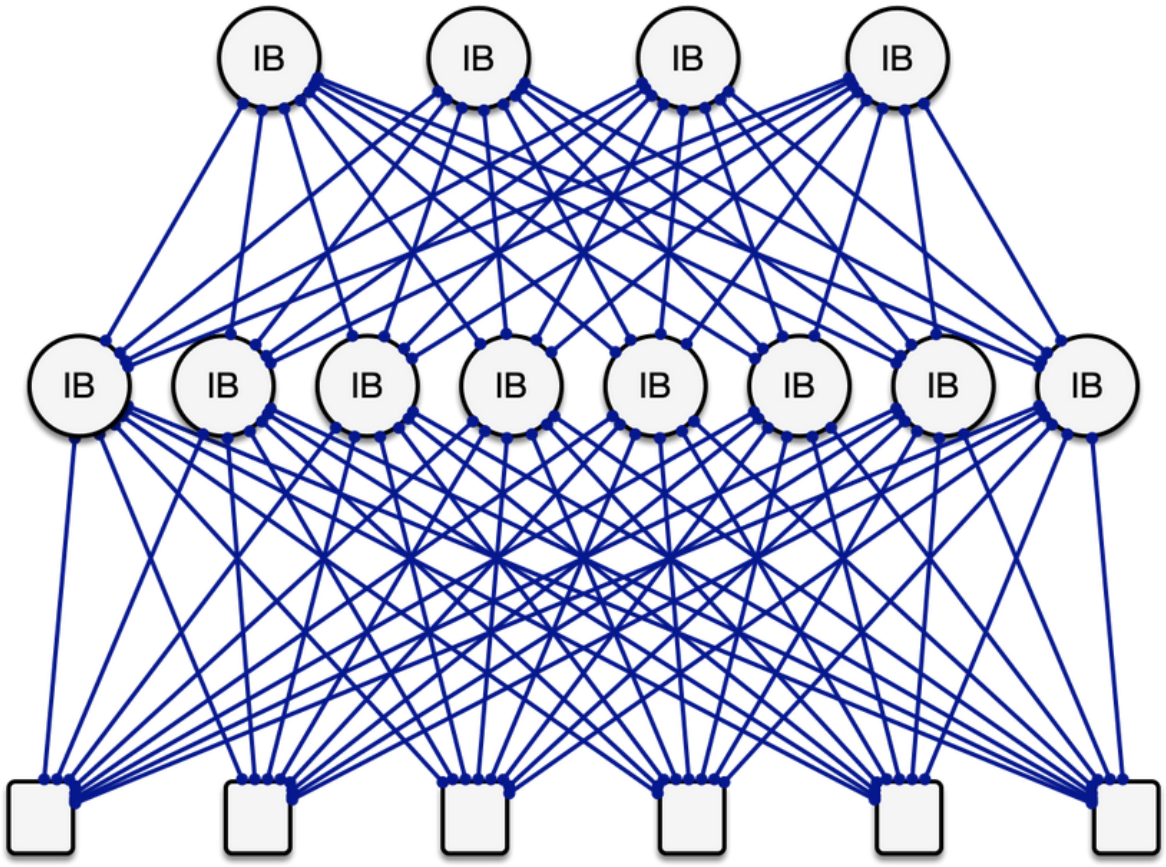
- 8x NVIDIA H200 GPUs
- 8x 400G NDR InfiniBand high-speed networking
- 8x local NVME storage, local to each GPU
- 2x200G Bluefield-3 Ethernet smart NIC
- 1G Ethernet for provisioning
- Dedicated BMC connection

All server hardware is managed using the industry-standard Redfish protocol.

A dedicated hypervisor resource provides virtualised compute for control, monitoring and supporting functions.

Networking

6G AI Sweden's high-speed InfiniBand fabric is implemented with 400G NDR InfiniBand from NVIDIA, implemented as a multi-rail network interconnecting the H200 GPU devices.



An example InfiniBand network fabric, in a fat tree topology, in which each compute node (bottom row) has 8 InfiniBand links. Multiple links may exist between switches to produce a non-blocking network, in which all nodes could communicate simultaneously on all links at full network bandwidth.

The infrastructure's Ethernet network is based on 800G NVIDIA switches and network fabric, broken out into 2x200G bonded links connecting the compute nodes and storage.

In addition, management Ethernet networks exist for server provisioning, control and power/ management.

Storage

6G AI Sweden selected high-performance storage from VAST data, capable of providing object, block and file storage services with multitenant isolation.

OpenStack Cloud Infrastructure

For AI use cases, a significant advantage of OpenStack is its native support for bare metal, virtualization and containerization, all couched within the same reconfigurable infrastructure. OpenStack's multi-tenancy model and fine-grained policy are also well-suited to cloud-native service providers for AI infrastructure.

OpenStack Kayobe provisions clouds using infrastructure-as-code principles. Hardware configuration, OS provisioning and network configuration are all defined using a single version-controlled source repo. OpenStack services are deployed and configured via Kolla-Ansible. Kayobe's precise configuration and management of infrastructure was selected for its flexibility, ease of operation and built-in support for high-performance computing.

For the AI compute nodes, a bare-metal cloud infrastructure was created using OpenStack Ironic. The servers are deployed using Ironic's virtual media driver, streamlining the bootstrap process and avoiding the need for DHCP and iPXE steps during the deployment.

Multitenant isolation is a critical requirement for 6G AI's business model. Bare-metal compute infrastructure is provisioned into client tenancies. Ironic implements functionality for effective management of bare metal in a multitenant environment, such as configurable steps for deployment and cleaning.

Neutron networking is implemented using OVN. Multitenant network isolation is implemented using virtual tenant networks, connecting a tenant's VMs and bare-metal compute nodes.

- Tenant networks in Ethernet are VLANs, enabling easy intercommunication between bare metal and virtualized compute resources. Multitenant Ethernet network isolation is implemented using the networking-generic-switch Neutron driver. Separate networks for cleaning and provisioning keep other phases of the infrastructure lifecycle safely behind the scenes.
- Tenant networks in InfiniBand are partitions, allocated as partition keys (pkeys) and managed with the networking-mellanox Neutron driver, integrated with NVIDIA's Unified Fabric Manager (UFM).

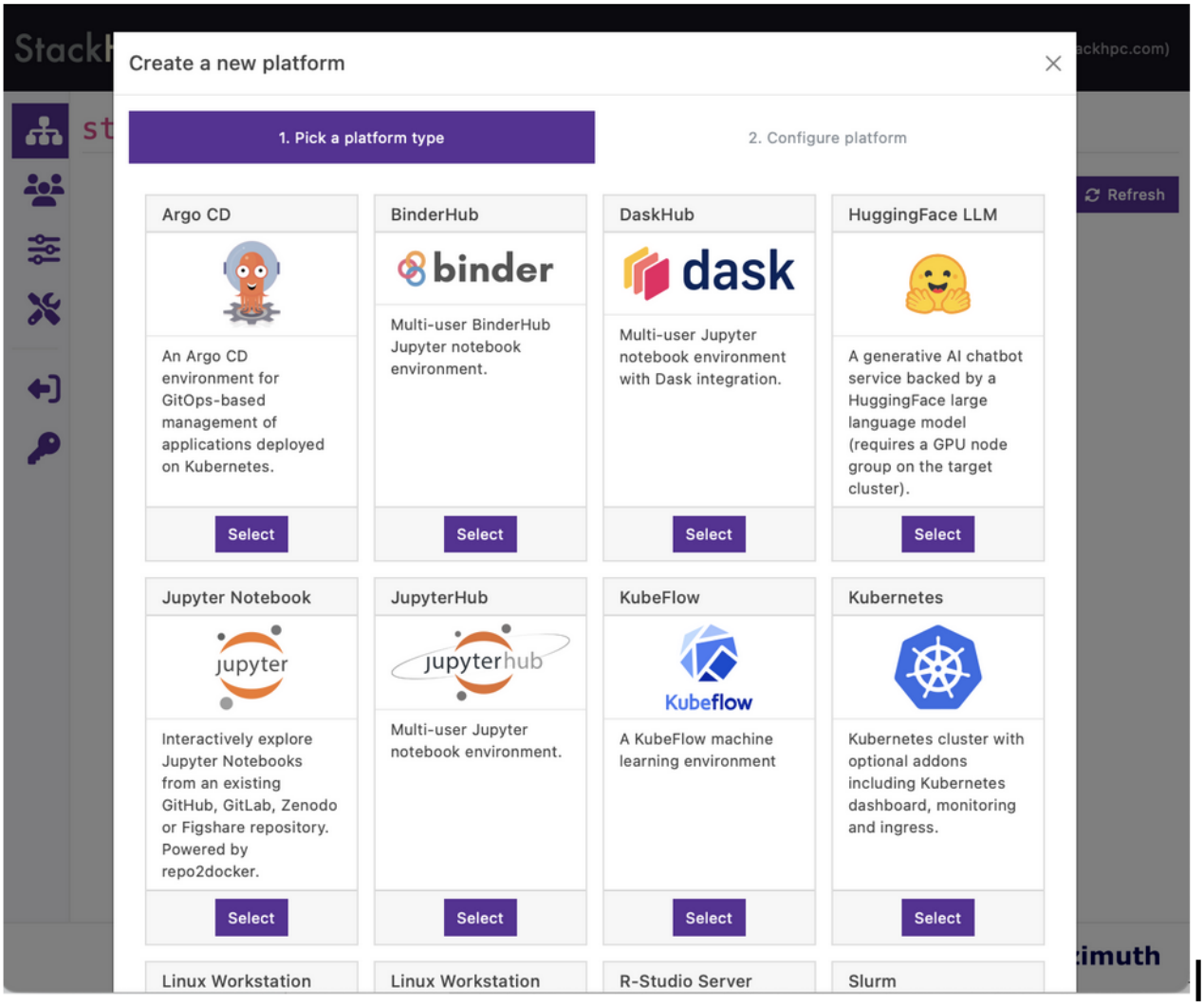
The VAST Data storage provides backing services for Glance, Cinder and Manila, plus S3 object storage for tenant workloads. For virtualised infrastructure, block storage is implemented using VAST Data's newly developed NVME-over-TCP Cinder driver.

File storage is implemented using VAST Data's Manila driver, providing multitenant orchestration of high-performance file storage, all using the latest optimisations of industry-standard NFS.

Azimuth and Kubernetes

World-class AI capabilities require world-class compute platforms, building on high-performance infrastructure to enable users to get straight to AI productivity. StackHPC is the lead developer and custodian of the Azimuth Cloud Portal, an intuitive portal for providing

compute platforms on a self-service basis. Azimuth is free and open source software, and StackHPC offers services for deployment, configuration, extension, maintenance, and support of the project.

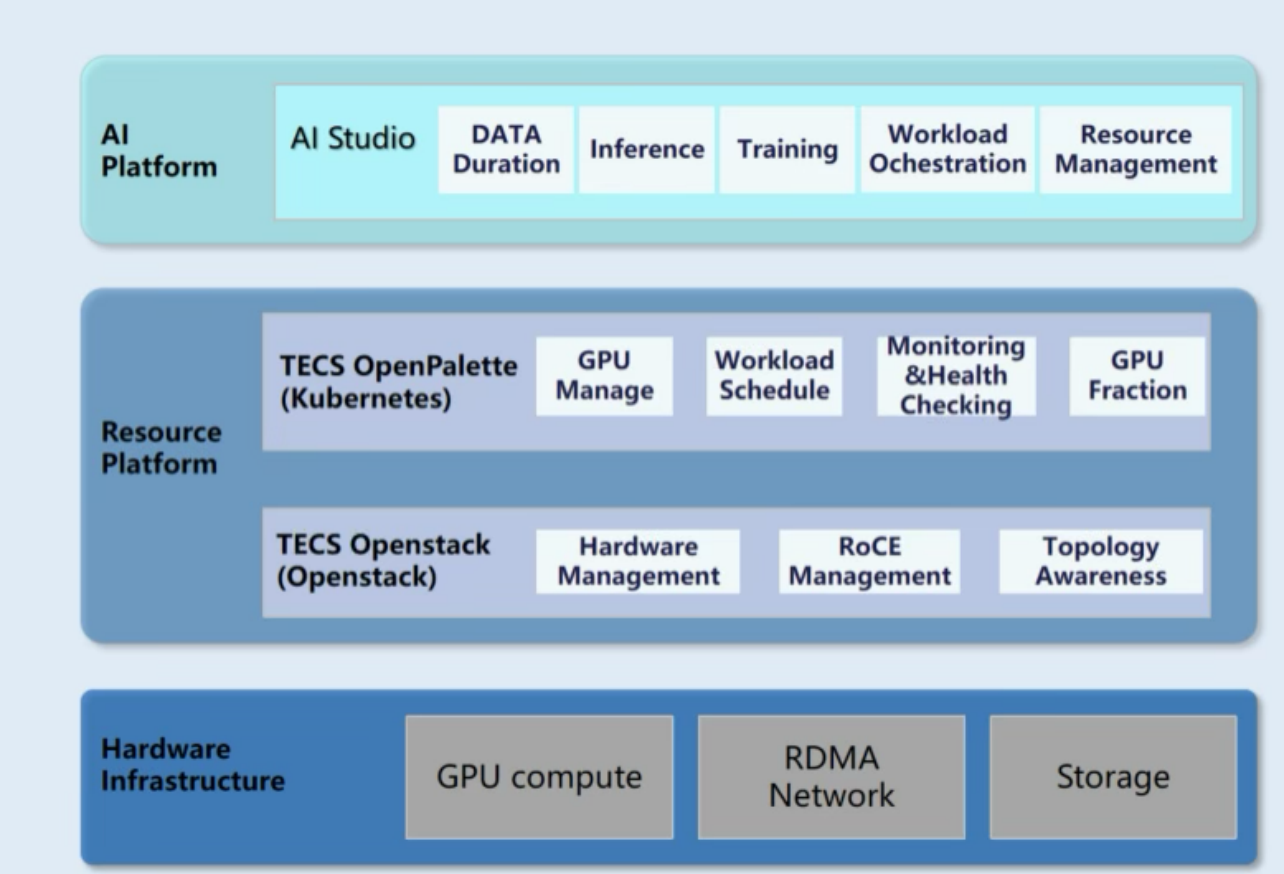


Platform creation in Azimuth involves selecting from a curated set of predefined infrastructure-as-code recipes, ready configured with deep integrations into the underlying infrastructure.

Most AI workloads are deployed as containerised applications using Kubernetes. In the 6G AI cloud, Kubernetes clusters span VM resources and bare-metal AI compute nodes. Kubernetes can be deployed as a platform in Azimuth or as a GitOps-driven workflow via FluxCD. In both cases, Kubernetes is deployed using industry-standard Cluster API and configured using Helm. 6G AI Sweden provides the NVIDIA NGC Catalog of containerised software applications and Helm charts.

ZTE

1. ZTE's AI infrastructure management architecture and solutions



Architecture: ZTE has launched a three-tier software architecture to solve the problem of AI:

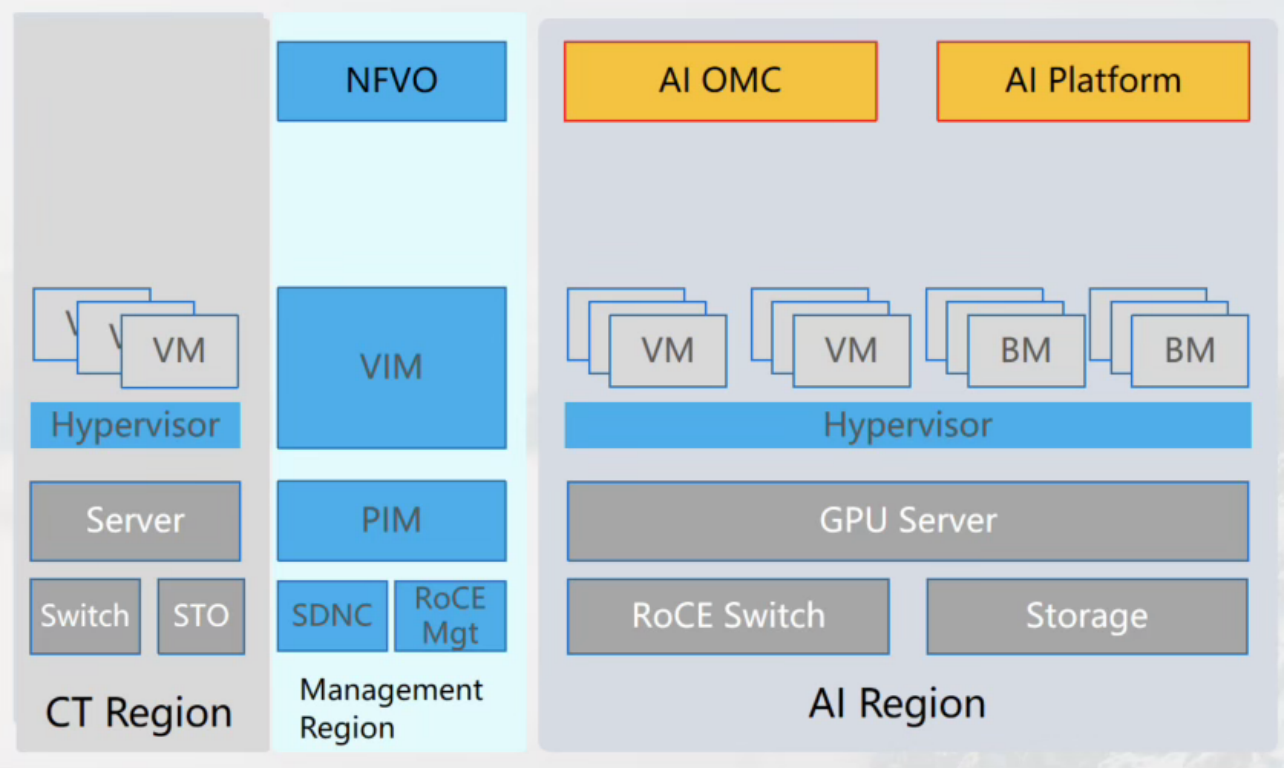
- Hardware infrastructure
- Resource platform
- AI platform.

ZTE provides a full set of AI hardware equipment. Includes GPU servers, RDMA switches, and high-performance storage devices. At the software level, the solution consists of OpenStack, Kubernetes, and ML layers☐
.

OpenStack is responsible for managing infrastructure hardware, including servers, network devices, and storage docking.☐
Kubernetes is the upper layer. Combined with the K8S platform as a computing power scheduling base. Add scheduling-related components to enhance AI scheduling capabilities.☐
AI Studio is ZTE's self-developed AI platform as a workload management layer. Provides the tool chain required for machine learning. Complete a variety of large model tasks such as model development, pre-training, inference deployment, and application development.

2. The case of ZTE AI infrastructure management

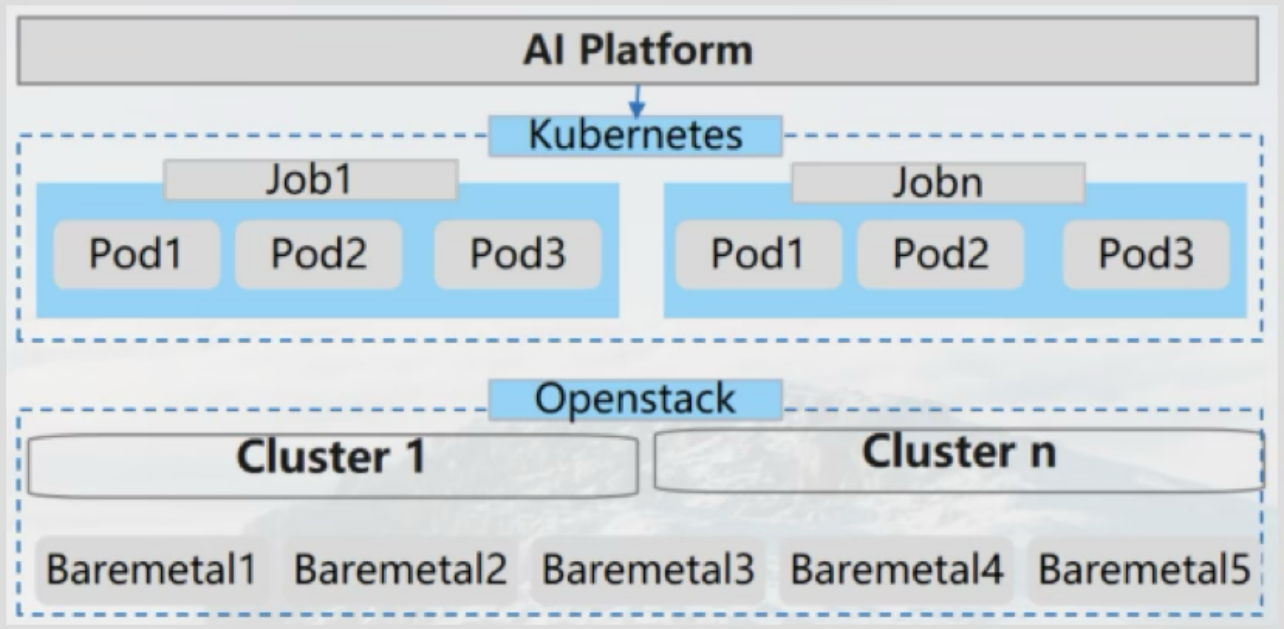
Telecom network cloud



Customer questions: Satisfaction of inference requirements, satisfaction of training requirements, and operation and maintenance strategies☐
Solution: Center training, edge inference. Enhance operations and maintenance.

IT resource pool

OpenStack distributes virtual machines + ironic baremetal.☐



*Training scenario:*Baremetal is scheduled to implement model training.☐
*Inference scenario:*The application runs on the virtual machine, and the model runs on the baremetal.

Authors

Amine Badaoui, Keshav Bareja, , Sylvain Bauza, Mark Collier, Dmitry Galkin, Artem Goncharov, Thel Gunther, Petr Kubica, Li Liu, Jimmy McArthur, Mike McDonough, Kendall Nelson, Mauro Oddi, Kyunghwan Oh, Tatiana Ovchinnikova, Allison Price, Sang Tran Quoc, Adrian Reza, Chris Sibbitt, Stig Telfer & Steve Westmoreland