LIQID

Inference-as-a-Service with Composable Kubernetes

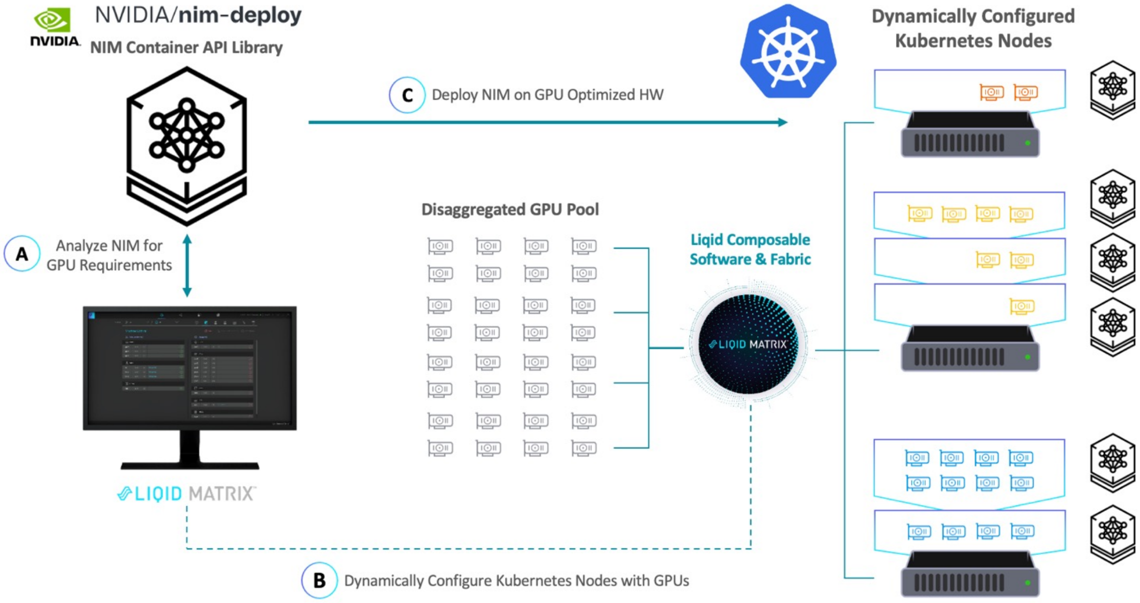
Unlock the full potential of AI with NIM™ inference microservices on dynamically configurable Kubernetes clusters - deploy and manage containers with composable infrastructure for scalable and efficient GPU deployments

Introduction

In the rapidly evolving world of artificial intelligence (AI), the need for scalable, flexible, and efficient infrastructure is paramount. Traditional infrastructure often struggles to keep pace with the dynamic demands of AI workloads, particularly when deploying large and varied AI models such as NIM inference microservices. This white paper presents a technology demonstration showcasing the capabilities and benefits of deploying NIMs on a dynamic Kubernetes infrastructure, leveraging Liqid's advanced GPU orchestration capabilities and NVIDIA accelerated computing.

NIM microservicess and Kubernetes have emerged as the preferred method for deploying AI workloads. NIM microservicess provide pre-optimized, containerized inference models that can be easily deployed across various environments, ensuring high performance and consistency. When combined with Kubernetes, a powerful orchestration platform, organizations can dynamically manage, and scale AI workloads based on real-time demands. This synergy allows for the efficient use of computational resources, enabling the deployment of diverse AI models from small to large, all while maintaining the agility needed to adapt to changing workloads. The combination of NIM microservicess and Kubernetes simplifies the deployment process, reduces operational overhead, and accelerates the delivery of AI-powered solutions, making it the go-to approach for modern enterprises seeking to harness the full potential of AI.

Composability further enhances the NIM microservices and Kubernetes AI deployment model by allowing resources such as GPUs to be dynamically pooled and optimally allocated based on the specific needs of each specific NIM container. By integrating composability with NIM microservices and Kubernetes, organizations can achieve the highest levels of GPU efficiency and flexibility, tailoring their infrastructure in real-time to optimize performance for each AI task. This leads to the best GPU resource utilization, reduced costs, and the ability to quickly scale and adapt to varying demands, making AI deployments more powerful and responsive to business needs.


NVIDIA/nim-deploy
NIM Container API Library
C Deploy NIM on GPU Optimized HW
Dynamically Configured Kubernetes Nodes
Disaggregated GPU Pool
Liqid Composable Software & Fabric
LIQID MATRIX
A Analyze NIM for GPU Requirements
LIQID MATRIX
B Dynamically Configure Kubernetes Nodes with GPUs

Overview: NIM-as-a-Service Deployed with Dynamically Composable Kubernetes

This demonstration illustrates a novel approach to deploying and managing AI workloads by dynamically allocating NVIDIA GPU resources within a Kubernetes cluster. By utilizing NVIDIA NIM microservices alongside a dynamic composable Kubernetes environment, we demonstrate how to optimize resource utilization, improve scalability, and support diverse application deployments. AI models come in various sizes, with the number of parameters being a key indicator of their complexity and GPU requirements. Composable GPUs can help optimize the bare metal infrastructure for different model sizes being deployed:

Technology Demonstration

1. Initial Setup

» Kubernetes Cluster: The demo begins with a Kubernetes cluster consisting of servers with no GPUs attached to the nodes.

» GPU Pool: A pool of GPUs is connected to the fabric, ready to be allocated as needed.

2. Deploying the NIM Container Pod

» NIM Deployment: We deploy a standard NIM container pod to the Kubernetes cluster using Helm, a popular Kubernetes package manager.

» Pending State: Upon deployment, the container pod enters a pending state because no nodes in the cluster have the required GPUs as specified in the pod specification.

3. Dynamic GPU Allocation

- **Liqid Orchestration:** Liqid's platform analyzes the NIM container pod in its pending state and determines the optimal number of GPUs required.
- **GPU Hot-Plugging:** The required number of GPUs are attached to the appropriate node in real-time through GPU hot-plug.

4. Kubernetes Node Updates

- **nvidia-gpu-operator:** Once the GPUs are attached, we update the nvidia-gpu-operator to reflect the new hardware configuration in the node labels.
- **Pod Deployment:** With the node labels updated, Kubernetes automatically proceeds to deploy the NIM container pod.

5. Continuous Optimization

- **Subsequent NIM Deployments:** When the next NIM container is deployed, it may enter a pending state due to insufficient GPU resources. Liqid's platform once again analyzes the requirements, attaches the necessary GPUs to a node, and Kubernetes handles the deployment automatically.
- **GPU Resource Management:** If a container is terminated by the user, Liqid's platform can optionally move the GPUs back to the free pool, making them available for future workloads.

6. Automation and API Integration

- **Automation in Kubernetes:** The entire process of GPU allocation and management is planned to be automated within the Kubernetes environment, significantly reducing the need for manual intervention.
- **NVIDIA Cloud Function (NCF) Integration:** The NIM requests are mapped to an NVIDIA Cloud Function (NCF), enabling the conversion of the request into an API-based deployment. This provides enhanced features related to scaling, performance, and Service Level Agreements (SLA), further optimizing the AI deployment process.

Goals and Benefits

The main objective of this demonstration is to highlight how our platform allows customers to deploy various NIM sizes (e.g., 8B vs. 70B) on precisely sized GPU infrastructure. This approach supports diverse applications while maximizing resource utilization by dynamically assigning the exact quantity and type of GPU required by each application to the Kubernetes worker node. Different model sizes demand different GPU resources, and this method ensures real-time allocation based on specific requirements.

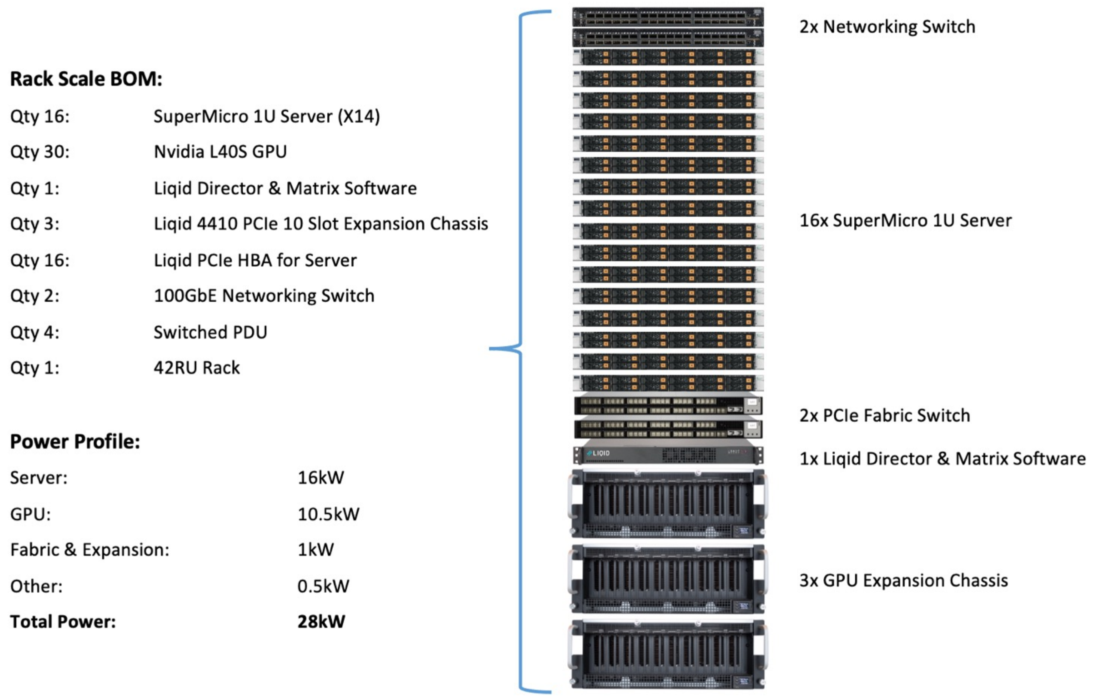
Maximized Resource Utilization: By dynamically allocating GPUs in real-time based on specific container requirements, we ensure that resources are used efficiently, reducing waste and lowering operational costs.

Scalability and Flexibility: The ability to dynamically attach and detach specific types and quantities of GPUs on-demand enables seamless scalability, supporting models and workloads of various sizes and complexities.

Increased NIM Density Per Node: Deploying more NIMs per node allows multiple AI models to run simultaneously, maximizing resource use and supporting diverse workloads on the same infrastructure. Supports up to 30x physical GPUs per node for increased container density.

Automation and API-Driven Management: Automating the process within Kubernetes and integrating with NVIDIA Cloud Functions streamlines operations, enabling faster deployments and more consistent performance.

Test Setup



Rack Scale BOM:

Qty 16:	SuperMicro 1U Server (X14)
Qty 30:	Nvidia L40S GPU
Qty 1:	Liqid Director & Matrix Software
Qty 3:	Liqid 4410 PCIe 10 Slot Expansion Chassis
Qty 16:	Liqid PCIe HBA for Server
Qty 2:	100GbE Networking Switch
Qty 4:	Switched PDU
Qty 1:	42RU Rack

Power Profile:

Server:	16kW
GPU:	10.5kW
Fabric & Expansion:	1kW
Other:	0.5kW
Total Power:	**28kW**

Item	Manufacturer	Model	Description
Server	SuperMicro	X14 CloudDC	X14 DP CloudDC with Intel® Xeon® 6 and BlueField-3
GPU	Nvidia	L40S 48GB	GPU for AI, Deep Learning, and Graphics
Networking	Nvidia	Spectrum-X	NVIDIA Spectrum SN5000
Storage	Liqid	LQD3000	NVMe 16TB per X14 Node
Fabric	Liqid	SmartStack-30	30x GPU Expansion, Fabric, and Matrix SW

NIM	GPU Qty	GPU Model	GPU Cost	Description
Llama 3.1 8B Instruct	2	H100	$60,000	Latency focused
	1	H100	$30,000	Throughput focused
	2	A100	$30,000	Latency
	1	A100	$15,000	Throughput focused
	2	L40S	$15,000	Latency
	2	L40S	$15,000	Throughput focused
Llama 3.1 70B Instruct	8	H100	$240,000	Latency focused
	4	H100	$120,000	Throughput focused
	8	A100	$120,000	Latency
	4	A100	$60,000	Throughput focused
	8	L40S	$60,000	Latency
	8	L40S	$60,000	Throughput focused
Llama 3.1 405B Instruct	16	H100	$480,000	Latency focused
	8	H100	$240,000	Throughput focused
	16	A100	$240,000	Latency focused
	8	A100	$120,000	Throughput focused
	N/A	L40S		Nvidia does not provide guidance
	N/A	L40S		Nvidia does not provide guidance

Telco and Edge Use Case

Offering “Inference as a Service” at the edge could greatly benefit telco customers by improving the efficiency, speed, and scalability of their operations. By running AI models closer to the data source, edge computing reduces latency, enabling real-time decision-making and enhancing customer experiences. This also lowers bandwidth usage, as less data needs to be transmitted to the cloud, reducing costs. Additionally, inference at the edge can provide enhanced security and privacy by processing sensitive data locally. Overall, telcos could use this to differentiate their services, improve performance, and drive new business opportunities in sectors like healthcare, retail, and manufacturing.

Leveraging a dynamic GPU environment to deliver inference-as-a-service enables the most efficient infrastructure at the Edge has several key advantages for Telco provides including:

» **Resource Efficiency:** Allocate GPUs dynamically to match workload needs.
» **Scalability:** Easily scale GPU resources for varying edge workloads.
» **Flexibility:** Reassign GPUs across tasks like AI or data processing.
» **Reduced Footprint:** Use fewer GPUs for multiple applications.
» **Agility:** Quickly adjust GPU resources for real-time needs.
» **Cost Savings:** Minimize hardware costs by optimizing GPU use.
» **Reliability:** Reallocate GPUs during failures for uptime.
» **Simplified Management:** Centrally manage all GPU resources.
» **Power Efficiency:** Reduce energy use by optimizing GPU allocation.
» **Future-Proofing:** Upgrade GPUs easily without overhauls.

Telco providers are uniquely positioned to lead in offering Inference-as-a-Service at the edge due to their vast network infrastructure, proximity to end users, and expertise in managing distributed systems. With low-latency access to data and the ability to deploy AI workloads closer to the edge, telcos can deliver real-time AI services, optimize network performance, and enable new use cases like smart cities and autonomous vehicles. Their infrastructure allows for seamless scaling, making them ideal providers of AI-powered edge services.

Conclusion

This demonstration highlights the significant advantages of deploying NVIDIA NIM™ inference microservices in a dynamic composable Kubernetes environment. By leveraging Liqid's dynamic GPU orchestration and NVIDIA accelerated computing, we provide a solution that not only meets the demands of modern AI workloads but also drives efficiency, scalability, and flexibility.

As AI continues to evolve and expand, the need for adaptive and resource-efficient infrastructure will only grow. This NIM-as-a-Service approach, deployed with dynamic Kubernetes, represents a forward-thinking solution that is ready to meet the challenges of today's AI landscape and beyond.

Reference

Liqid SmartStack

SuperMicro Servers

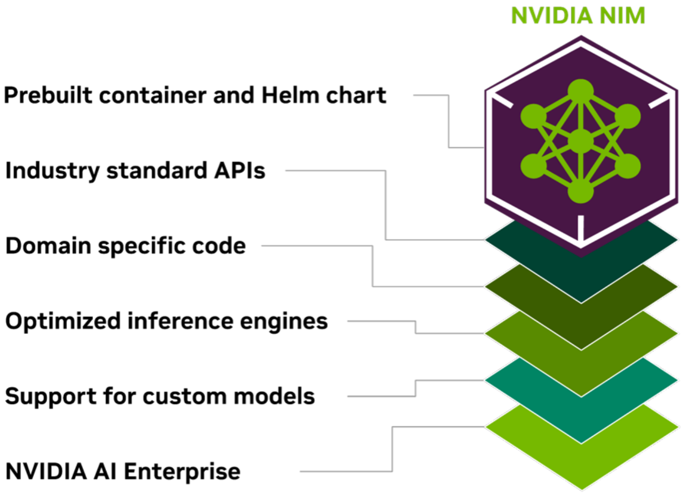
NVIDIA NIM

APPENDIX - A

Composable Infrastructure: Composable infrastructure is an advanced IT architecture that allows bare metal resources—such as compute, storage, and networking—to be dynamically pooled and allocated on-demand. This approach transforms traditionally fixed resources into fluid, software-defined assets that can be quickly assembled and reconfigured based on workload requirements. By leveraging composability, organizations achieve higher resource efficiency, scalability, and agility, enabling them to rapidly adapt to changing business needs while optimizing costs.

NVIDIA NIM inference microservices: NIM microservices are pre-optimized, containerized AI models designed for efficient deployment across diverse environments. This encapsulates powerful AI inference capabilities within microservices, making it easy to deploy, scale, and manage AI workloads in cloud, on-premises, or hybrid infrastructures. NIM microservices streamline the AI deployment process by providing ready-to-use models that can be integrated into applications with minimal effort, ensuring high performance and flexibility in meeting varying computational demands.

Kubernetes: Kubernetes is a robust open-source platform that automates the deployment, scaling, and management of containerized applications. It orchestrates containers across a cluster of machines, ensuring that applications run efficiently, reliably, and consistently, regardless of the underlying infrastructure. Kubernetes provides key features such as automated rollouts, service discovery, load balancing, and self-healing, making it indispensable for managing complex, distributed systems at scale. By leveraging containers, Kubernetes enables agile and resilient application development, simplifying operations in cloud-native environments.


NVIDIA NIM
Prebuilt container and Helm chart
Industry standard APIs
Domain specific code
Optimized inference engines
Support for custom models
NVIDIA AI Enterprise

HEADQUARTERS

11400 Westmoor Circle
Ste 225
Westminster, CO 80021

ONLINE

www.liqid.com
info@liqid.com

TELEPHONE

+ 1 303.500.1551