WEKA

WEKA AI RAG Reference Platform (WARRP)

REFERENCE ARCHITECTURE

Contents

Executive
Summary

WEKA is redefining the enterprise data stack by setting a new standard for AI infrastructure. The WEKA® Data Platform is a cloud- and hardware-agnostic software solution designed to enhance the speed, scale, simplicity, and sustainability of performance-intensive workloads, enabling customers to achieve faster discoveries, insights, and business outcomes.

In today's AI and machine learning landscape, Retrieval-Augmented Generation (RAG) has become a powerful framework for improving the efficiency of inferencing workloads. RAG combines retrieval-based methods with generative models to deliver highly accurate, contextually relevant results. As organizations adopt RAG pipelines, optimizing data platform performance and I/O efficiency becomes critical for smooth, scalable operations.

This whitepaper explores the WEKA AI RAG Reference Platform (WARRP) and the importance of RAG in inferencing workloads, examining the different layers required to run RAG pipelines in production vs a RAG proof-of-concept setup. WARRP also describes the specific platforms WEKA utilized to achieve this efficient pipeline. Successful RAG inferencing depends on efficiently retrieving relevant data from large, often distributed, knowledge sources, such as company documents and databases, which need to be transformed and embedded to be accessed by LLMs to generate accurate responses. For production-scale operations, efficiently managing the movement of models and tokens between compute and data resources is essential. Any performance bottlenecks in the data platform or I/O can significantly impact latency, throughput, and accuracy, making it crucial to fine-tune these components for high-speed access and processing.

We also discuss key metrics for measuring the efficiency of RAG pipelines in production environments, including Time to First Token (TTFT), Time Per Output Token (TPOT), Cost Per Token, and Total Tokens - Time to Last Token. These metrics help gauge inferencing pipeline performance and identify areas for optimization. By analyzing these metrics, organizations can better understand system behavior under load, pinpoint inefficiencies, and ensure the RAG pipeline meets operational goals at scale.

Optimizing data platform and I/O performance is vital to the success of RAG in managing complex inferencing workloads. This whitepaper provides a detailed guide for developers and IT professionals looking to improve the performance of their RAG-based systems by utilizing WARRP as a blueprint that can be used as is or adapted specifically for customer needs.

Benefits of WARRP with the WEKA Data Platform

Running Retrieval-Augmented Generation (RAG) pipelines can be highly demanding, especially when dealing with large model repositories and complex AI workloads. Leveraging WARRP design with the AI-native WEKA® Data Platform can significantly boost the performance and efficiency of RAG across the entire AI pipeline.

Manage Massive Model Repositories of Hundreds of Terabytes

Modern AI models are growing exponentially in size, with some repositories containing hundreds of terabytes of data. The WEKA Data Platform enables seamless management of these large-scale repositories by optimizing data storage and retrieval through its scale-out high-performance architecture. This allows AI teams to store vast amounts of data while maintaining fast access and retrieval times, which is crucial for RAG pipelines instead of copying models and containers to local NVMEs of the GPU servers for quicker access.

Reduce Time to Load Models to Inference Servers

Loading large AI models to inference servers can be time-consuming, slowing down workflows and limiting scalability. With WEKA's advanced data processing capabilities, AI models can be loaded onto inference servers significantly faster. By reducing latency in data access and eliminating bottlenecks, the platform ensures that your AI pipelines can continue running smoothly and efficiently, even as model sizes grow.

Accelerate Time to Output GenAI Artifacts

Generating artifacts from Generative AI models, such as text, images, or other media, requires fast and efficient processing. WEKA accelerates this process by optimizing the data flow within your RAG pipelines, reducing the time it takes to store these artifacts. Whether you're working with language models, visual generators, or other GenAI systems, WEKA ensures quick and reliable artifact production by minimizing data movement and processing delays.

Boost VectorDB & Embedding Performance

RAG pipelines rely heavily on Vector Databases (VectorDB) and embedding generation to process and store high-dimensional data vectors. WEKA's platform accelerates VectorDB operations and embedding generation through its high-performance I/O and low-latency architecture. This allows for faster querying, storage, and retrieval of embeddings, which is critical in AI models that rely on real-time recommendations, retrieval, and context-aware responses.

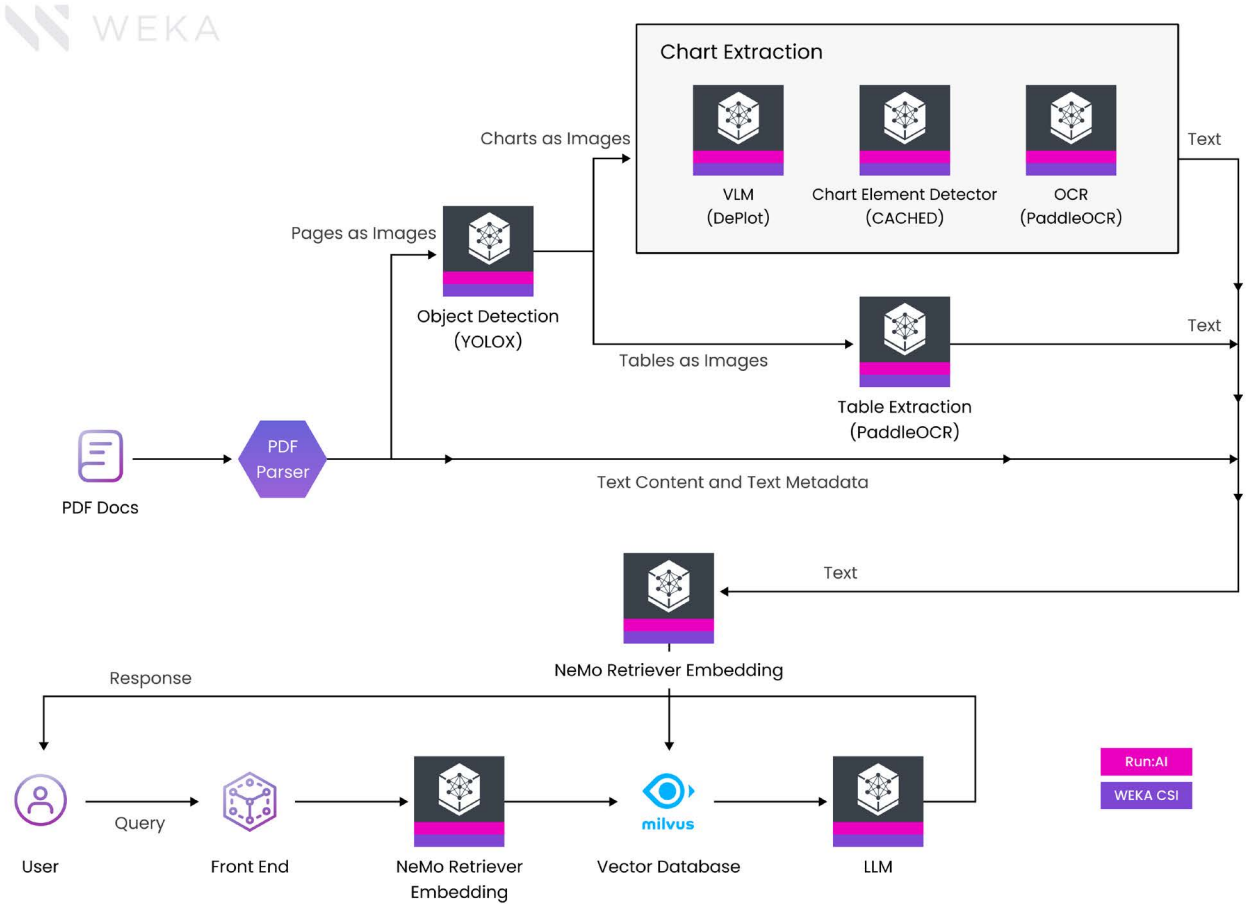
Enable Token Checkpointing to Switch Between GPUs

In large-scale AI training and inferencing environments, GPU resource utilization is key. WEKA supports token checkpointing, enabling you to switch between GPUs seamlessly during model training or inference. By allowing efficient state saving and resource swapping, token checkpointing on WEKA ensures that GPU resources are fully utilized without unnecessary delays or interruptions, increasing overall efficiency and throughput.

Integrate Training and Inferencing Farms

One of the standout features of the WEKA Data Platform is its ability to unify training and inferencing environments. Traditional pipelines often separate training and inference into distinct workflows, but with WEKA, you can run both on the same infrastructure, eliminating the need for separate farms. This integration simplifies the infrastructure and accelerates model deployment and iteration by streamlining data access and GPU usage across training and inferencing processes.

Solution Benefits



RAG is a widely used design pattern in modern AI applications that enhances generative AI functionalities. Its ability to ground large language models' (LLMs) outputs with relevant, factual, and up-to-date information has gained traction. The key strength of RAG lies in combining non-parametric knowledge with the parametric knowledge of LLMs, enabling more accurate and contextually relevant responses to user queries.

Trillions of PDF files are generated annually, each likely consisting of multiple pages filled with various content types, including text, images, charts, and tables. This goldmine of data can only be used as quickly as humans can read and understand it.

However, with generative AI and retrieval-augmented generation (RAG), untapped data can be used to uncover business insights that can help employees work more efficiently and reduce costs.

Imagine being able to accurately extract the knowledge contained in massive volumes of enterprise data—effectively talking to the data—to quickly make your digital human an expert on any topic. In turn, this enables your employees to make smarter decisions faster.

Building a cost-effective enterprise-grade RAG pipeline

Here are the benefits of using NVIDIA NIM™ microservices to create a multimodal PDF data extraction pipeline: cost and stability.

Cost has three considerations:

- **Time to market:** NVIDIA NIM microservices are designed to be easy-to-use and scalable model inference solutions, enabling enterprise application developers to focus on working on their application logic rather than spending cycles on building and scaling out the infrastructure. NVIDIA NIM microservices are containerized solutions with industry-standard APIs and Helm charts to scale.
- **Cost of deployment:** NVIDIA NIM uses the full suite of NVIDIA AI Enterprise software to accelerate model inference, maximizing the value enterprises can derive from their models and, in turn, reducing the cost of deploying the pipelines at scale. Figure 2 demonstrates the improvements in accuracy and throughput achieved in testing this ingestion and extraction pipeline.
- **Token Optimization:** Integrating NVIDIA NIM microservices with Run:ai and WEKA Data Platform provides optimal cloud, on-premises and hybrid infrastructure performance. Leveraging the WEKA Data Platform accelerates multiple stages of the AI data pipeline, reducing model load times, increasing GPU utilization, decreasing total token delivery time, and reducing the cost of RAG inferencing.

Solution Overview

In this whitepaper, we provide a modular reference architecture for generative AI applications that optimize inference metrics by leveraging RAG, NVIDIA NIM on cloud, on-premises, or hybrid infrastructure using Run:ai, LlamaIndex, LangChain, Milvus, and WEKA Data Platform.

Solution Technology

WARRP is a reference architecture designed to help customers understand and implement Inferencing RAG pipelines and take advantage of the WEKA Data Platform's capabilities for accelerating multiple components, such as the model loading time and vector databases to accelerate embeddings and retrievals. The WARRP architecture enables workload portability, allowing for multi-datacenter and multicloud utilization to facilitate global pipelines.

The reference architecture consists of a flexible and modular infrastructure that can be deployed on cloud, on-premises and hybrid infrastructures. The reference architecture describes the different layers of a RAG Inference environment from the infrastructure layer (which can be on-premises or on any cloud environment) up through the orchestration layer (Run.AI and Kubernetes) and data ingestion layer (Nv-Ingest and Milvus VectorDB) to the Application layer, using the model or most likely a chain of different models, to achieve a business outcome (such as a chatbot, a recommender engine, a cyber security analysis tool, etc..) The architecture is configured to blend and scale independently. Leveraging NVIDIA AI Enterprise with NVIDIA NIM, Run:ai and the WEKA Data Platform, the WARRP architecture is designed to optimize GenAI application performance and have portability across multiple cloud infrastructures.

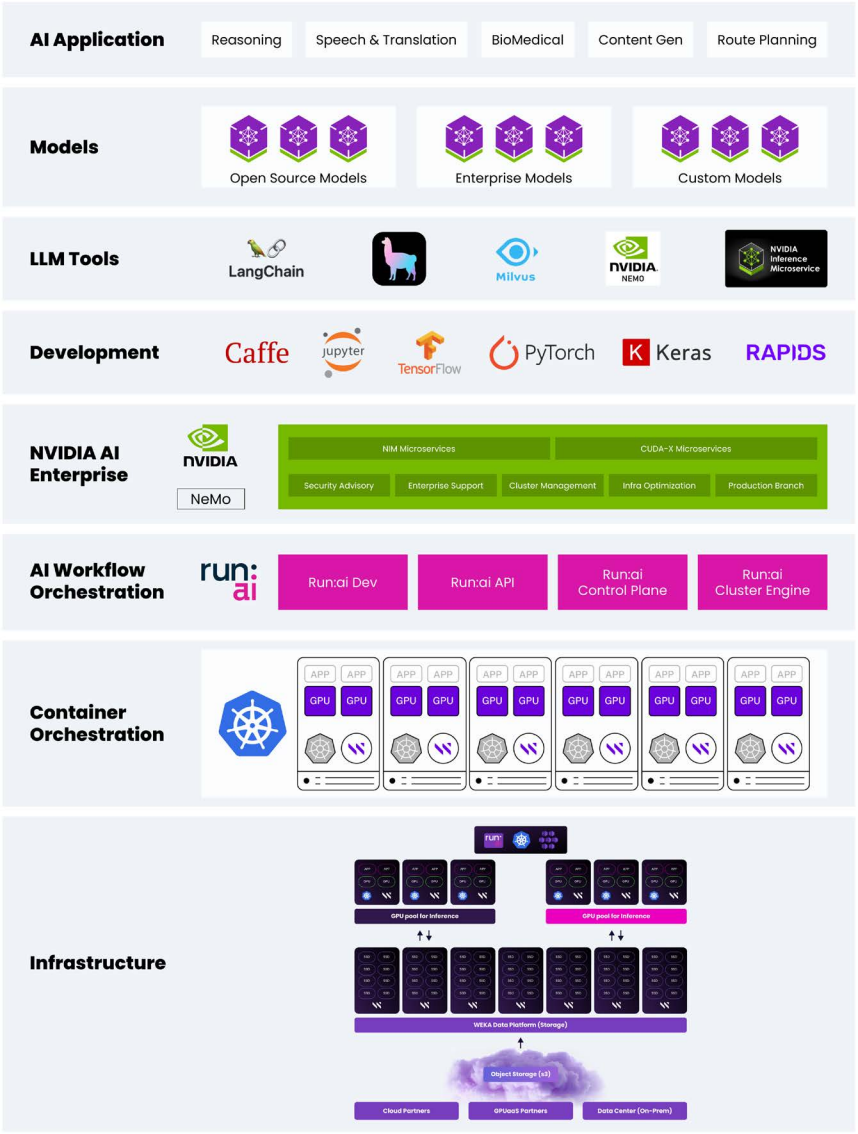
Each layer of the WARRP reference architecture can contain one or more components that provide services to the entire pipeline. We describe the currently WEKA-validated components that can be replaced with different WEKA-validated components that offer the same service. For example, we utilize Milvus as the VectorDB in the Utility Apps layer. However, due to the WEKA Data Platform's capability to support high-performance databases as the storage layer, this can be replaced with a different VectorDB that users require (such as MongoDB Atlas or Postgres with PGVector, etc.)

Run:ai's software stack runs on top of Kubernetes, providing streamlined operational capabilities, right-sized resources that can be autoscaled on demand, reporting service level policy, GPU utilization and model response latency post-deployment.

Key components of the WARRP architecture are NVIDIA AI Enterprise, NIM microservices, NVIDIA frameworks and NIM AI blueprints that streamline and accelerate the entire pipeline of testing and deployment. Currently, WARRP is utilizing the following NVIDIA software components.

- NVIDIA AI Enterprise is an end-to-end, cloud-native software platform that accelerates data science pipelines and streamlines the development and deployment of production-grade co-pilots and other generative AI applications.
- NVIDIA NIM microservices are used in various locations to run multiple services such as large language models (LLMs) and embedding models.

- NV-ingest is a scalable, performance-oriented document content and metadata extraction microservice used to extract data in preparation for embedding into the VectorDB. This is a pre-GA utility from NVIDIA.
- NVIDIA Triton™ Inference Server is open-source software that standardizes AI model deployment and execution across every workload and is used as the inference server component of the reference architecture.
- NVIDIA Generative AI Examples - a repo that provides tools for setting up a variety of different RAG use cases
- NVIDIA NeMo: a scalable and cloud-native generative AI framework built for researchers and developers working on Large Language Models, Multimodal, and Speech AI (Automatic Speech Recognition and Text-to-Speech)
- NVIDIA NeMo™ Retriever microservice synthetic data generation mines datasets to create contextual queries from pdf-sourced embeddings.
- GenAI-perf provides key performance metrics such as time to first token and token throughput.



Technology Overview

Model Application and Observability: LangChain

Context retrieval is a mainstay in LLM engineering. In this reference architecture, we leverage NVIDIA NIM to quickly deploy the integration of LangChain's observability for simple tracing, monitoring and evaluation of RAG applications. LangChain is a powerful tool when used to enhance Retrieval-Augmented Generation (RAG) pipelines in Generative AI Large Language Model (LLM) applications.

LangChain Overview

LangChain is an open-source framework that simplifies the development of applications using LLMs. It provides tools and abstractions to improve LLMs' customization, accuracy, and relevancy, enabling the creation of various applications like chatbots, question-answering, content generation, and summarizers.

LangChain consists of LangChain Libraries, LangChain Templates, LangServe, and LangSmith, offering interfaces, integrations, reference architectures, and a developer platform for building and deploying LLM-powered applications. The framework includes standard interfaces for Model I/O, Retrieval, and Agents, enabling you to integrate data sources, LLMs, and tools to build complex applications.

LangChain is part of a rich ecosystem of tools, supported by an active community, and simplifies AI development by abstracting the complexity of data source integrations.

Orchestration: Run:ai

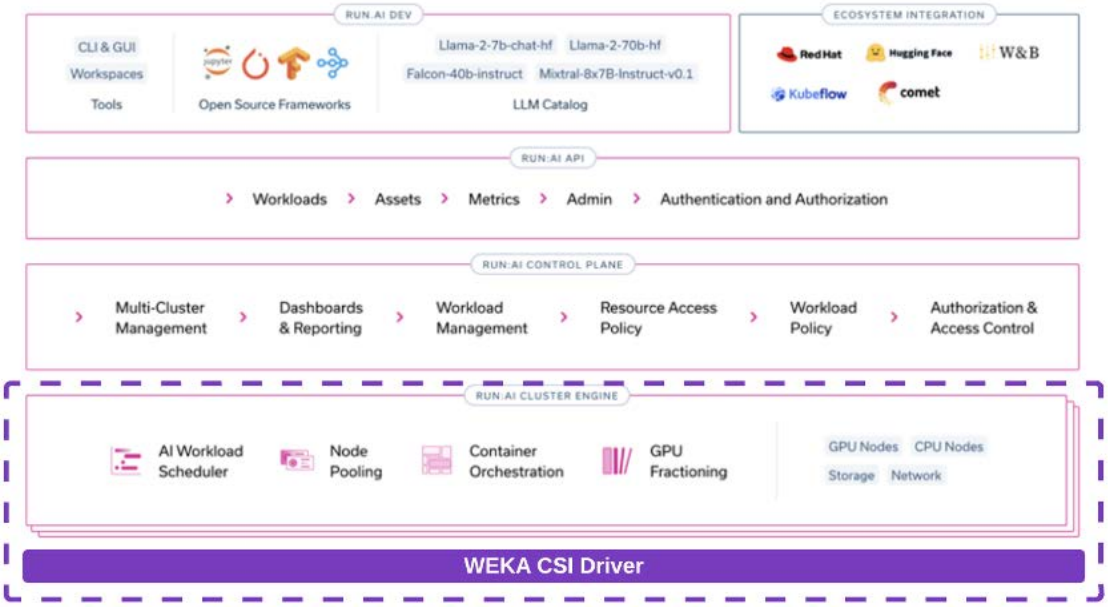
Overview of Run:ai

Run:ai's AI workload and GPU orchestration platform revolutionizes AI and machine learning operations by addressing key infrastructure challenges through dynamic resource allocation, comprehensive AI lifecycle support, and strategic resource management. By pooling resources across environments and utilizing advanced orchestration and accelerators, Run:ai significantly enhances GPU efficiency and workload capacity. Its policy engine and open architecture foster strategic alignment with business objectives, enabling seamless integration with external tools and systems. This results in significant increases in GPU availability, workloads, and GPU utilization, all with zero manual resource intervention, accelerating innovation and providing a scalable, agile, and cost-effective solution for enterprises.

Run:ai allows you to:

- Centralize Control and Visibility – Run:ai provides comprehensive dashboards and analytics, offering IT teams insight into all resources and workloads. Align resource allocation with business goals by setting policies and priorities across departments, projects, or users.
- Optimize GPU Utilization and ROI – Automated resource management and efficient GPU sharing enable higher utilization, maximizing the return on investment for each GPU.

- Build on a Truly Open and Extendable Platform – Leverage built-in workflows optimized for the complete AI development lifecycle, or extend the platform with seamless integrations for any AI/ML applications, including MLOps tools.
- Accelerate Hybrid Cloud – Run:ai provides centralized control and visibility across both on-premises and cloud resources, empowering organizations to effectively manage hybrid cloud AI infrastructures and easily scale AI initiatives.



Using Run:ai with the WEKA Data Platform

Run:ai and WEKA create a unique AI architecture that combines innovations to develop an efficient, containerized AI stack that runs both on-premises and in the cloud. WEKA's software-defined data platform dramatically increases data performance and scale, while Run:ai automates AI workload and GPU orchestration. Together, these capabilities significantly enhance the efficiency and productivity of your team.

Run:ai and WEKA enable your team to:

- Innovate faster with less risk. Run experiments and train models across multiple clouds at high speed—with the same proven stack everywhere.
- Achieve the best return on investment. Fully utilize all GPU and storage resources across multiple clouds to run workloads faster and more efficiently in less time.
- Start small and scale in a cloud-smart way. Invest in your AI/ML workflows and less on infrastructure using a hybrid multi-cloud approach to grow and scale.

Vector Database: Milvus

Milvus is a high-performance, highly scalable vector database running efficiently across various environments, from laptops to large-scale distributed systems. It is available as both open-source software and a cloud service.

Milvus is an open-source project under LF AI & Data Foundation distributed under the Apache 2.0 license. Most contributors are high-performance computing (HPC) experts specializing in building large-scale systems and optimizing hardware-aware code. Core contributors include Zilliz, ARM, NVIDIA, AMD, Intel, Meta, IBM, Salesforce, Alibaba, and Microsoft professionals.

Unstructured Data, Embeddings, and Milvus

Unstructured data, such as text, images, and audio, varies in format and carries rich underlying semantics, making it challenging to analyze. To manage this complexity, embeddings convert unstructured data into numerical vectors that capture its essential characteristics. These vectors are then stored in a vector database, enabling fast and scalable searches and analytics.

Milvus offers robust data modeling capabilities, enabling you to organize unstructured or multi-modal data into structured collections. It supports a wide range of data types for different attribute modeling, including common numerical and character types, various vector types, arrays, sets, and JSON, saving you from the effort of maintaining multiple database systems.

AI Integrations

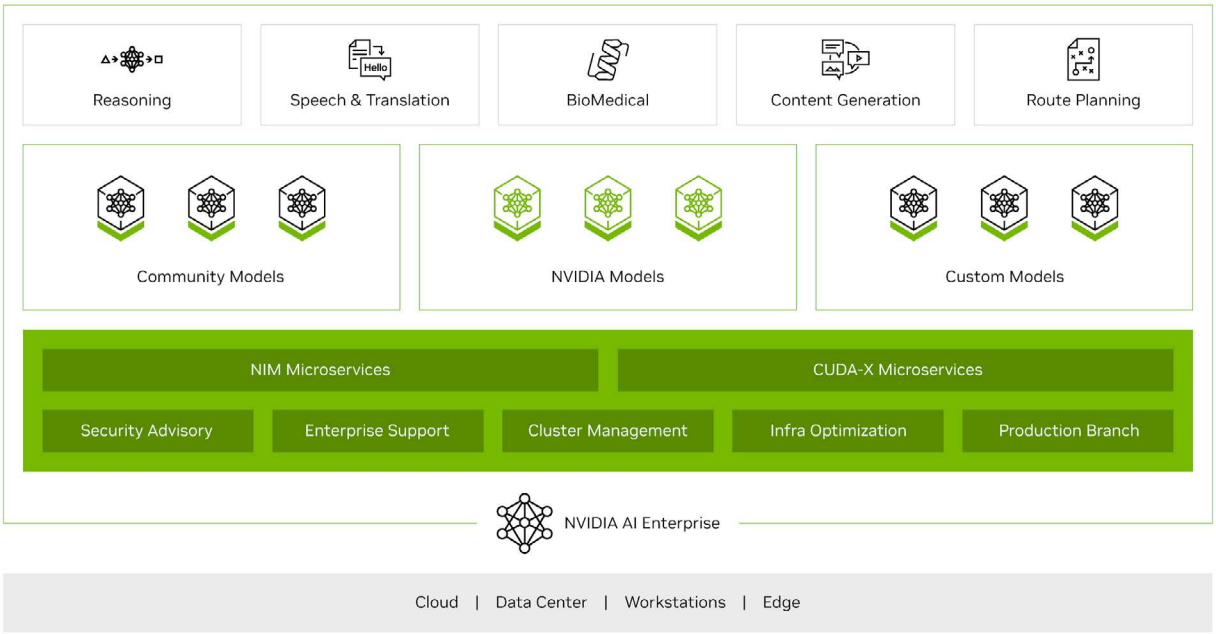
Embedding Model Integrations Embedding Models convert unstructured data to their numeric representation in high-dimensional data space so that you can store them in Milvus. Currently, PyMilvus, the Python SDK, integrates several embedding models so that you can quickly prepare your data into vector embeddings. For details, see Embedding Overview.

Reranking Model Integrations In the realm of information retrieval and generative AI, a reranker is an essential tool that optimizes the order of results from initial searches. PyMilvus also integrates several reranking models to optimize the order of results returned from initial searches. For details, refer to Rerankers Overview.

LangChain and other AI tool integrations. In the GenAI era, tools such as LangChain, are gaining attention from application developers. As a core component, Milvus usually serves as the vector store in such tools.

Enterprise-Grade AI Software Platform From NVIDIA

NVIDIA AI Enterprise is an end-to-end, cloud-native software platform that accelerates data science pipelines and streamlines the development and deployment of production-grade co-pilots and other generative AI applications. Easy-to-use microservices provide optimized model performance with enterprise-grade security, support, and stability to ensure a smooth transition from prototype to production for enterprises that run their businesses on AI.



NVIDIA AI Enterprise is tightly integrated with accelerated platforms to speed up AI workloads through software optimization. This improves efficiency and performance, reduces associated energy costs, data center footprint and investments, and time to production, and contributes to more sustainable AI computing.

Enterprise-grade security, stability, manageability, and support

As AI rapidly evolves and expands, the complexity of the software stack and its dependencies grows. NVIDIA AI Enterprise is designed for running mission-critical AI that businesses run on by offering regular releases of security patches for critical and common vulnerabilities and exposures (CVEs), production branch for API stability, end-to-end management software, and enterprise support with service-level agreements (SLAs).

Cloud-native and certified to run everywhere

NVIDIA AI Enterprise is optimized and certified to ensure reliable performance running AI in the public cloud, virtualized data centers, or on the NVIDIA DGX platform. This provides the flexibility to develop applications once and deploy them anywhere, reducing the risk of moving from pilot to production caused by infrastructure and architectural differences between environments.

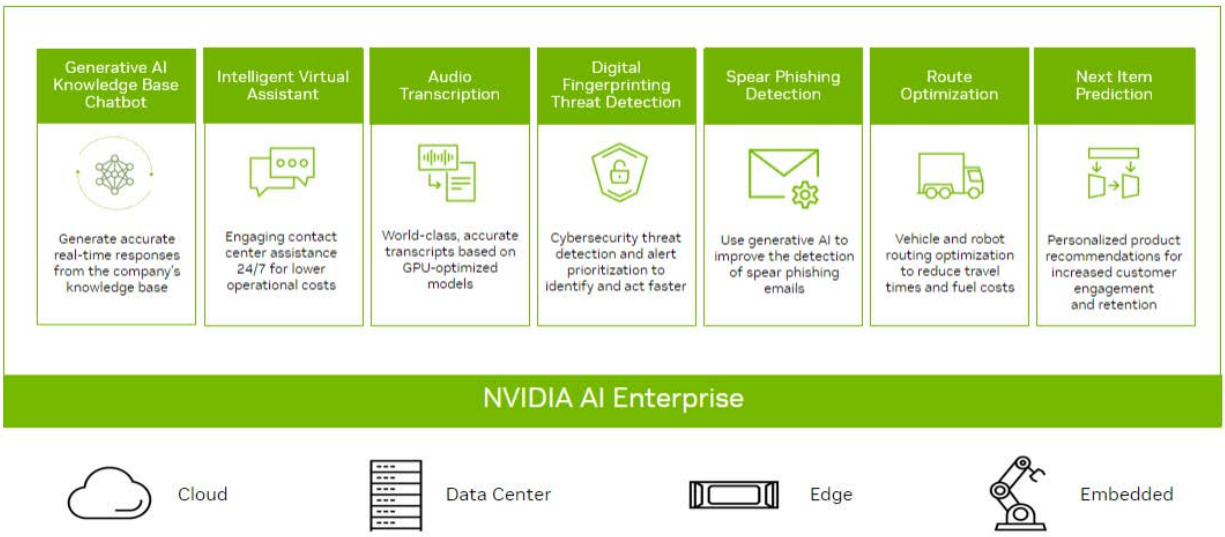
NVIDIA AI Software Supported

NVIDIA AI application frameworks, NVIDIA pre-trained models, and all other NVIDIA AI software available in the NVIDIA NGC™ catalog are supported by an NVIDIA AI Enterprise license. With 100+ AI frameworks and pre-trained models, including NVIDIA NeMo™, NVIDIA Maxine™, cuOpt, and more to be added, look for the “NVIDIA AI Enterprise Supported” label on NGC.

Organizations start their AI journey by experimenting and piloting with the open, freely available NVIDIA® NGC libraries and frameworks. Now, when they're ready to move from pilot to production, enterprises can easily transition to a fully managed and secure AI platform with an NVIDIA AI Enterprise subscription. This gives enterprises deploying business-critical AI, the assurance of business continuity with NVIDIA AI Enterprise Support and access to NVIDIA AI experts.

NVIDIA AI Workflows

NVIDIA AI Enterprise includes new AI solution workflows for building AI applications including contact center intelligent virtual assistants, audio transcription, and cybersecurity digital fingerprinting to detect anomalies. These packaged AI workflow examples include NVIDIA AI frameworks and pre-trained models, as well as resources such as Helm Charts, Jupyter Notebooks, and documentation to help customers get started in building AI-based solutions more easily. NVIDIA's cloud-native AI workflows run as microservices that can be deployed on Kubernetes alone or with other microservices to create production-ready applications.



Key Benefits:

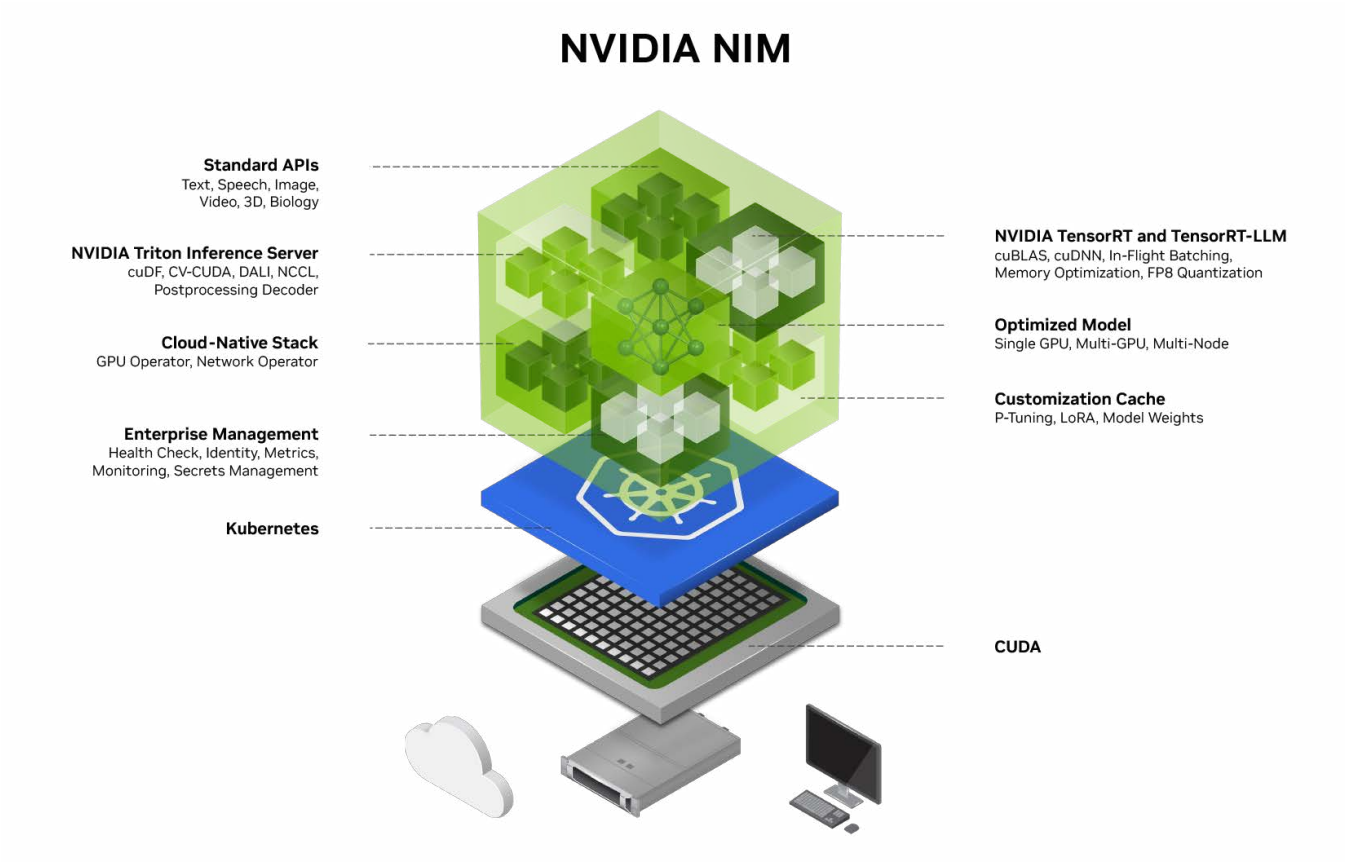
- Reduce development time, at a lower cost
- Improve accuracy and performance
- Confidence in the outcome by leveraging NVIDIA's expertise

Production-Ready Pre-trained Models

Pretrained AI models make high-performing AI development easy, quick, and accessible by eliminating the need to build models from scratch. NVIDIA AI Enterprise includes pre-trained models without encryption for healthcare and vision AI tasks such as people detection, vehicle detection, federated learning model, image registration, etc. By accessing these pre-trained models, developers can view the model's weights and biases, which can help explain and understand model bias. In addition, unencrypted models are easier to debug and integrate into custom AI apps.

NVIDIA NIM Microservices

NVIDIA NIM™, part of NVIDIA AI Enterprise, is a set of easy-to-use microservices designed for secure, reliable deployment of high performance AI model inferencing across the cloud, data center and workstations. These prebuilt containers support a broad spectrum of AI models—from open-source community models to NVIDIA AI Foundation models, as well as custom AI models. NIM microservices are deployed with a single command for easy integration into enterprise-grade AI applications using standard APIs and just a few lines of code. Built on robust foundations including inference engines like Triton Inference Server, TensorRT, TensorRT-LLM, and PyTorch, NIM is engineered to facilitate seamless AI inferencing at scale, ensuring that you can deploy AI applications anywhere with confidence. Whether on-premises or in the cloud, NIM is the fastest way to achieve accelerated generative AI inference at scale.



NVIDIA Accelerated Computing

WEKA and NVIDIA deliver paralleled performance, scalability, and efficiency for AI and data-intensive workloads. The WEKA Data Platform is optimized to provide ultrafast, low-latency access to massive datasets, ensuring that NVIDIA accelerated compute clusters never wait for data to process. With its advanced architecture, WEKA removes the typical bottlenecks found in traditional storage systems, enabling GPUs to operate at their full potential. This is crucial for workloads like AI model training, deep learning, 3D rendering, and real-time analytics, where the infrastructure must rapidly process large volumes of data to achieve timely results.

WEKA's NVIDIA-certified solutions significantly enhance accelerated compute utilization, allowing organizations to maximize the performance of their accelerated infrastructure. NVIDIA GPUs are designed for high-speed data processing, but their efficiency is often underutilized when waiting for data from storage. WEKA's platform solves this by delivering data at speeds that match the processing power of GPUs, ensuring continuous data flow and eliminating idle time. This leads to significantly improved compute resource utilization, driving better performance and faster time-to-insight for AI and ML workloads.

WEKA Data Platform leverages NVIDIA Magnum IO™ GPUDirect™ Storage (GDS) , a groundbreaking technology that allows NVIDIA GPUs to directly access data from storage, bypassing the CPU entirely. This innovation eliminates traditional bottlenecks in data transfer, significantly reducing latency and improving overall system performance. However, the true power of GDS is realized when paired with WEKA's high-performance data platform, specifically designed to handle the demands of AI, machine learning (ML), and other data-intensive applications.

The combination of NVIDIA and WEKA technologies is particularly impactful in AI and ML environments. AI model training involves processing vast amounts of data, and the speed at which this data is fed into NVIDIA GPUs directly affects the training time. With NVIDIA GDS eliminating the CPU bottleneck and WEKA Data Platform providing high-throughput data access, AI teams can train models faster, iterate more efficiently, and gain insights in less time. This accelerates the entire AI pipeline, from data ingestion to model training and inference, ultimately reducing the time it takes to bring AI-driven innovations to market.

Beyond AI and ML, WEKA's NVIDIA-certified solutions benefit other data-intensive applications such as scientific simulations, financial modeling, video processing, and energy exploration. These applications require powerful compute resources and fast, reliable data access. WEKA's ability to deliver parallel data streams to multiple GPUs simultaneously ensures that even the largest datasets are processed efficiently without storage bottlenecks slowing down operations. This is critical for industries where real-time decision-making and high-performance computing are essential.

WEKA and NVIDIA offer a cutting-edge solution for organizations looking to unlock the full potential of their accelerated computing infrastructure. Together, they provide a seamless and powerful platform that accelerates data access, maximizes GPU utilization, and improves performance across a wide range of data-intensive workloads. Whether you're working on AI model training, real-time data analytics, or scientific simulations, the combination of WEKA's high-performance data platform and NVIDIA's accelerated computing technologies enables faster processing, reduced latency, and more efficient operations. This partnership empowers organizations to stay ahead in the competitive, data-driven landscape by delivering faster time-to-insight and enabling the successful execution of the most demanding workloads.

The WEKA AI-Native Data Platform™

NVIDIA HGX™ servers are engineered to deliver unparalleled performance for AI, machine learning (ML), and high-performance computing (HPC) workloads. These servers maximize computational efficiency and throughput, making them ideal for the most demanding applications. Integrating the WEKA Data Platform with NVIDIA HGX H100 servers elevates performance, offering NVIDIA customers an unmatched data processing and storage experience for their AI, ML, and HPC workloads.

The WEKA Data Platform is designed to complement the computational power of NVIDIA HGX H100 servers, providing seamless, high-speed data access and storage. NVIDIA customers benefit from the WEKA Data Platform's rich enterprise feature set, including local snapshots, automated tiering, dynamic cluster rebalancing, private cloud multi-tenancy, backup, encryption, authentication, key management, user groups, quotas, and more.

The WEKA Data Platform is a software-based solution that is purpose-built to modernize enterprise data stacks. Its advanced AI-native, data pipeline-oriented architecture delivers exceptional performance at scale, providing an optimal environment for data-intensive applications that makes GPUs and AI, ML, and HPC workloads run faster and work more efficiently. Cloud and hardware-agnostic, WEKA's software enables seamless data portability across on-premises, cloud, and edge deployments and in hybrid and multicloud environments, offering native compatibility with the industry's broadest selection of server hardware, hyperscale clouds, and GPU clouds. WEKA can be purchased as subscription software, a fully managed service, or a turnkey ready-to-use appliance, WEKApod, giving customers a wide range of deployment options.

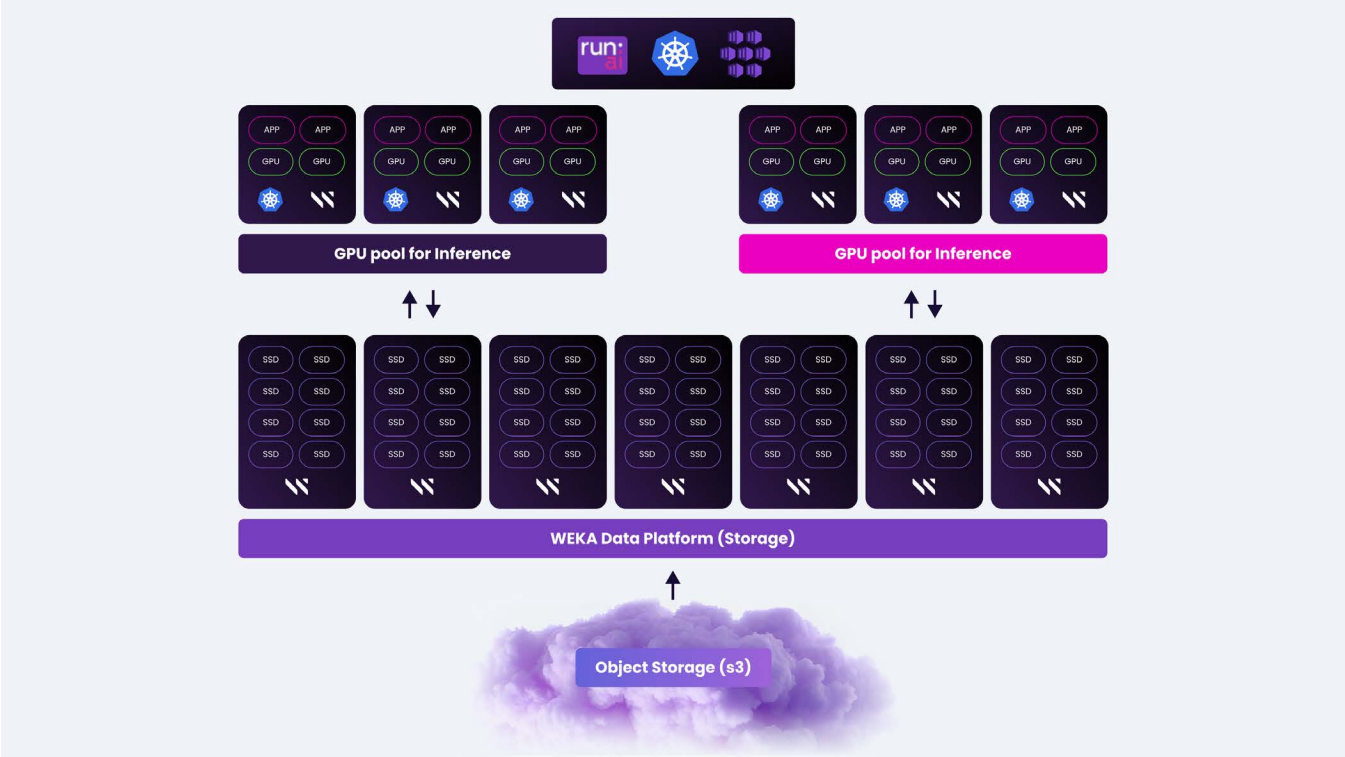
Technology Requirements

Cloud Architecture Overview

This architecture outlines a Kubernetes environment running on AWS cloud infrastructure that leverages shared storage via the WEKA Data Platform, a Milvus Vector DB instance as the vector database for embeddings, and GPU scheduling and orchestration provided by Run:ai. The architecture is designed to support scalable, high-performance AI and machine learning workloads, efficiently managing compute resources and data.

Components:

1. AWS Cloud Infrastructure: Hosts the Kubernetes cluster, and provides scalable compute, networking, and storage services.
2. Kubernetes Cluster: Manages containerized workloads and services across a distributed set of nodes.
3. WEKA Data Platform: Delivers high-performance shared storage, allowing fast data access across Kubernetes nodes.
4. Milvus Vector DB: A high-performance GPU-accelerated, highly scalable vector database that has AI integrations and stores embeddings. It is critical for Retrieval-Augmented Generation (RAG) and other inferencing workloads.
5. Run:ai: Orchestrates and schedules GPU resources, enabling efficient hardware utilization for deep learning and AI models.
6. CSI Plugin: Connects Kubernetes with the WEKA Data Platform's shared storage for seamless data access across the environment.



Deployment Steps:

1. Deploy WEKA Data Platform Cluster

   Start by deploying the WEKA Data Platform on AWS. This involves provisioning a set of EC2 instances and configuring a WEKA cluster. WEKA provides shared storage accessible to all Kubernetes nodes, which is essential for managing data-intensive AI workloads. Terraform is used to deploy a WEKA cluster on an organization's preferred cloud infrastructure, providing orchestrated deployment in minutes. WEKA added support for automated deployments using Terraform to provide a familiar experience for WEKA customers deploying to their cloud platform of choice and for deployment consistency across increasingly multiple cloud deployments. The WEKA Cloud Deployment Manager (CDM) gives customers a standard way to deploy WEKA across AWS, Azure, and Google Cloud. The WEKA CDM provides a GUI-driven step-by-step experience configuring WEKA Data Platform software and the cloud infrastructure supporting the WEKA environment.

2. Deploy Kubernetes Cluster

   Set up a Kubernetes cluster on AWS, which will act as the core orchestration layer for containerized applications. The Kubernetes compute resources will then have a series of ansible playbooks deploying Kubernetes using Kubespray, creating a control plane and configuring each worker pool with a WEKA CSI driver, linking the compute and storage domains through WEKA's DPDK mode CSI driver.

   - Deploy Master Nodes: Initialize the master nodes, which are responsible for managing the cluster, scheduling workloads, and maintaining cluster health.

- Add Worker Nodes: Add worker nodes to the cluster. These nodes will run the containerized applications and manage GPU resources.

3. Deploy Distributed Milvus Cluster

   Deploy a distributed Milvus cluster as a Kubernetes pod to serve as the vector database for embeddings. The Milvus database is hosted on the WEKA Data Platform and will store embeddings and other data structures used in Generative AI workloads, providing low-latency access for inferencing tasks in the RAG pipeline.

4. Install NVIDIA GPU Operator

   Install the NVIDIA GPU Operator on the Kubernetes cluster to enable GPU scheduling and management. The GPU Operator helps manage the lifecycle of GPUs, including driver installation and resource monitoring.

5. Install WEKA CSI Plugin

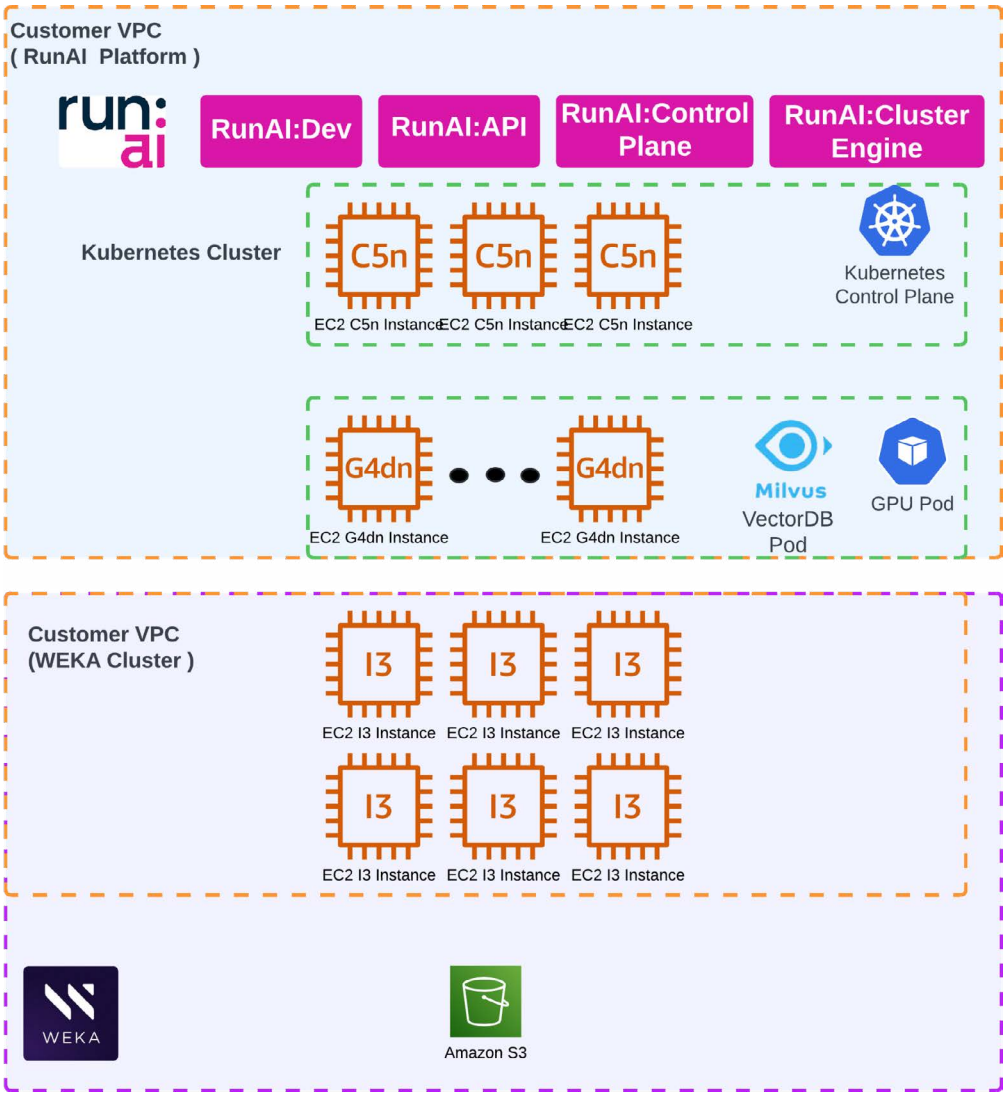
   Integrate the WEKA Data Platform with Kubernetes by installing the WEKA CSI plugin. This plugin enables Kubernetes to mount the WEKA Data Platform as persistent storage volumes, making it easy for pods to access shared data across the cluster.

6. Deploy Run:ai

   Finally, Run:ai will be deployed on the Kubernetes cluster to manage and orchestrate GPU workloads. RunAI enhances resource utilization by scheduling deep learning tasks efficiently, ensuring optimal use of the available GPU resources across the Kubernetes environment.

This architecture enables a flexible, high-performance, cloud-native environment that handles data-intensive Generative AI workloads. WEKA's high-speed data platform, MongoDB's robust vector database, and Run:ai's GPU orchestration work together to streamline large-scale AI operations, making this solution suitable for organizations running complex machine learning and inferencing tasks in the cloud.

Cloud Configuration Details



	QTY	INSTANCE TYPE	FUNCTION
WEKA Cluster	6	i3en.6xlarge	WEKA Backends
Kubernetes Cluster			
Control Plane	3	c5n.9xlarge	Kubernetes Control Plane
GPU Workers	N	g4dn.12xlarge	Kubernetes GPU Worker Nodes

Building RAG Pipeline on NVIDIA AI Enterprise using NVIDIA NIM

Creating a Kubernetes deployment for a Retrieval-Augmented Generation (RAG) inferencing pipeline with NVIDIA AI Enterprise involves orchestrating several components to ensure optimized processing and efficient data handling. Below is a high-level summary of the components and steps required:

Key Components

1. **NVIDIA NGC and Enterprise AI Suite:** Provides AI-optimized infrastructure and NVIDIA NGC containers for AI inferencing, including pre-built deep learning frameworks like PyTorch.
2. **NVIDIA NIM:** A set of microservices designed to deploy scalable AI pipelines, enabling model inferencing through NVIDIA's optimized infrastructure.
3. **NVIDIA NeMo Retriever:** NIM inference microservices that allow organizations to seamlessly connect custom models to diverse business data and deliver highly accurate responses for AI applications using RAG.
4. **WEKA CSI Driver:** A Kubernetes Container Storage Interface (CSI) driver that connects WEKA's high-performance file system to Kubernetes, offering scalable and high-speed storage for large datasets.
5. **Milvus Vector Database:** Serves as a vector store to handle high-dimensional embeddings produced by large language models, facilitating efficient similarity search for retrieval-augmented generation.
6. **LangChain:** Framework to manage large language models (LLMs) and build chains that integrate with vector stores, simplifying pipeline development and improving the flexibility of RAG systems.
7. **NVIDIA Triton Inference Server:** Manages inferencing requests across GPUs, optimizing the serving of models and maximizing resource utilization.
8. **NVIDIA TensorRT-LLM:** Enhances LLMs by accelerating inferencing on NVIDIA GPUs, essential for real-time or low-latency applications within the RAG pipeline.

Deployment Steps

1. **Environment Preparation:**
   - Deploy a Kubernetes cluster with NVIDIA GPU support, utilizing NVIDIA's Enterprise AI suite for optimized performance.
   - Configure GPU nodes with NVIDIA drivers, NVIDIA Container Toolkit, and the WEKA CSI Driver to handle data storage and retrieval.
2. **Storage and Data Management Setup:**
   - Deploy WEKA's CSI driver in Kubernetes for scalable data storage. Configure the WEKA Data Platform for high-performance data access, which is critical for rapid data retrieval and model training.
3. **Vector Database and Data Indexing:**
   - Set up Milvus as a vector store within the Kubernetes cluster to enable efficient similarity search. Integrate it with LangChain or LlamaIndex to simplify query handling, embedding storage, and retrieval for RAG processes.

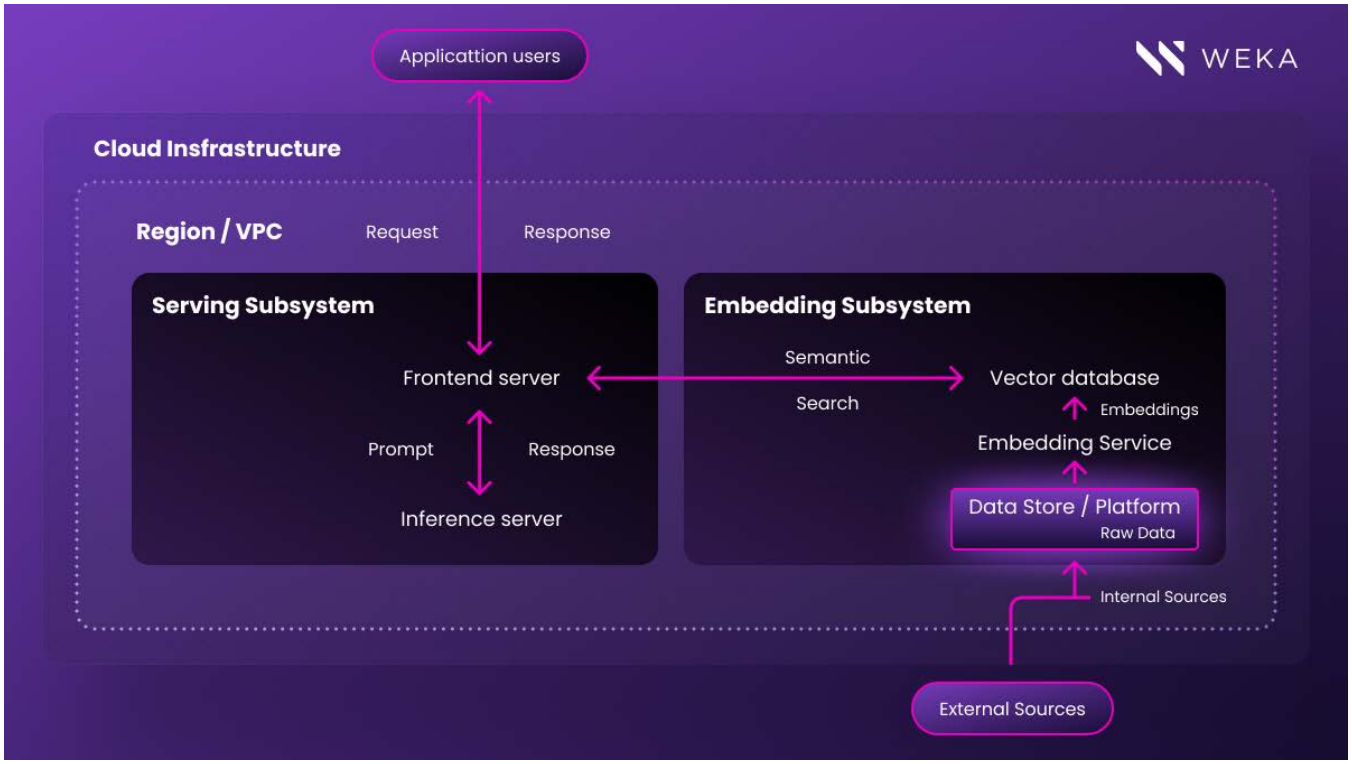
4. **Model Loading and Optimization:**
   - Deploy NVIDIA Triton Inference Server to manage model serving across GPUs, ensuring efficient resource allocation and support for concurrent requests.
   - Use NVIDIA TensorRT-LLM for model optimization, enhancing throughput by reducing latency and increasing inference speed for LLMs.
5. **Configure the Pipeline of NVIDIA NIM Microservices:**
   - Configure NVIDIA NIM microservices to orchestrate the deployment pipeline. Define microservices for each component (e.g., data preprocessing, embedding generation, vector storage, and inferencing) to ensure smooth communication and scalability.
6. **Pipeline Development with LangChain:**
   - Using LangChain, build a pipeline that connects various components, such as embedding generation (via PyTorch models), embedding storage in Milvus, and inferencing calls to Triton. This allows the RAG system to retrieve relevant information and generate accurate responses.
7. **Testing and Optimization:**
   - Test the pipeline for latency, accuracy, and performance. Fine-tune components (e.g., adjusting Milvus indexing, refining data storage on the WEKA Data Platform, optimizing Triton inference configurations) to achieve desired throughput and latency.
8. **Monitoring and Scaling:**
   - Implement monitoring for all components, leveraging NVIDIA Enterprise AI monitoring tools. Configure autoscaling policies within Kubernetes for load balancing and scalability, ensuring reliable performance across varying demand levels.

Summary

This deployment leverages NVIDIA AI Enterprise and NIM microservices in Kubernetes to streamline and scale a RAG inferencing pipeline. Through integration with the WEKA Data Platform for storage, Milvus for vector management, optimized model inferencing with NVIDIA Triton and TensorRT-LLM, and the simplified orchestration from LangChain and LlamaIndex, this setup achieves a high-performance, end-to-end inferencing pipeline that efficiently handles RAG workloads.

Solution Design

The following diagram shows a high-level view of an architecture for a RAG-capable generative AI workflow in the Run:ai Kubernetes Cluster:



The architecture contains a serving subsystem and an embedding subsystem.

The serving subsystem handles the request-response flow between the application and its users. The subsystem includes a front-end server and an inference server. The serving subsystem interacts with the embedding subsystem through a vector database, Milvus, hosted on the WEKA Data Platform.

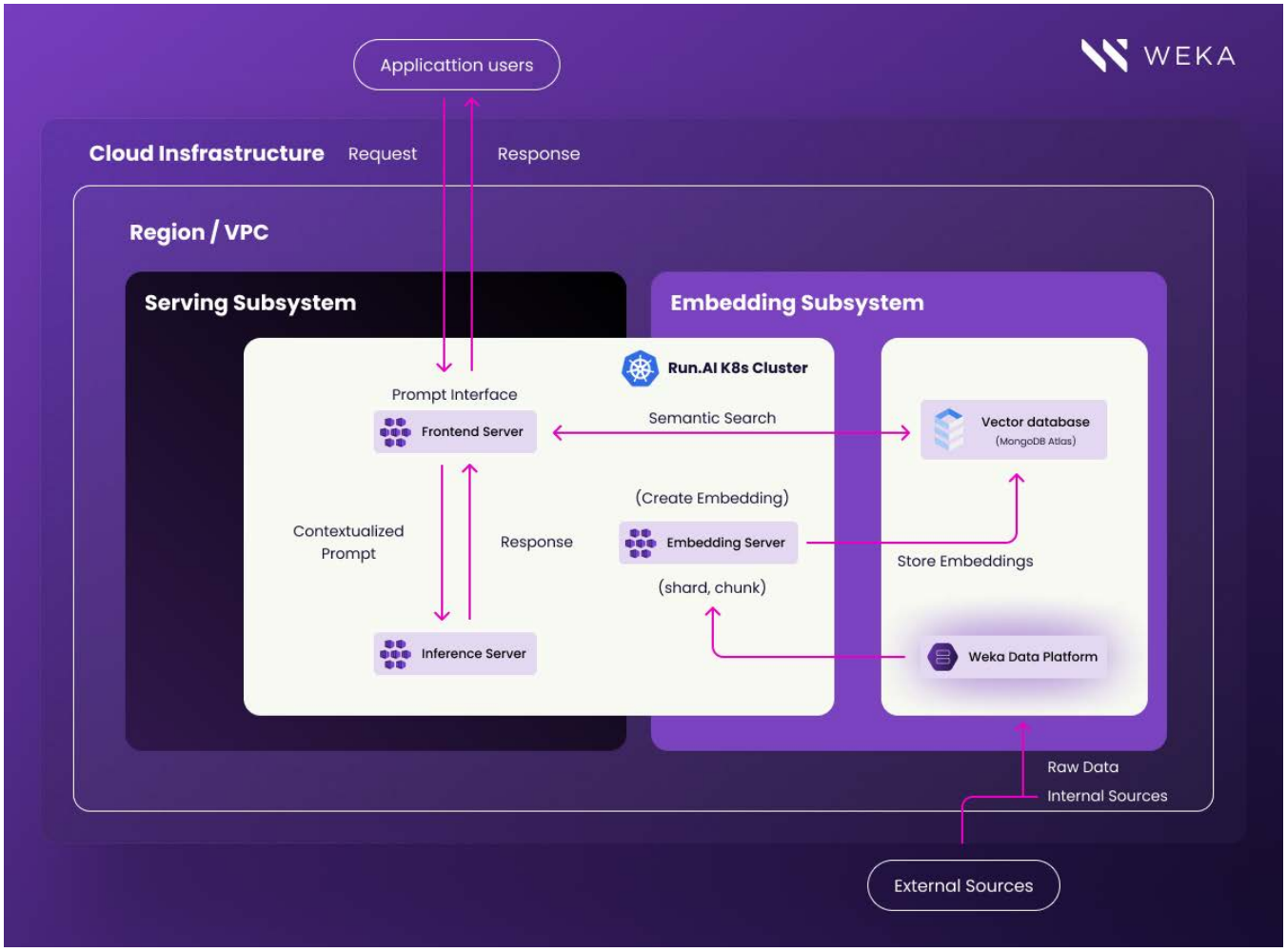
The embedding subsystem enables the RAG capability in the architecture. This subsystem does the following:

Ingests data from data sources in the WEKA Data Platform, on-premises, and other cloud platforms.

Converts the ingested data to vector embeddings.

Stores the embeddings in the vector database.

The following diagram shows a detailed view of the architecture:



The preceding diagram shows that the frontend server, inference server, and embedding service are deployed in a Run:ai Kubernetes cluster. RAG data is ingested into a cloud data platform. The architecture uses Milvus as the vector database to store embeddings and perform semantic searches. Vector databases are designed to efficiently store and retrieve high-dimensional vectors.

Technology Overview

Embedding System

The following is the flow of data in the embedding subsystem:

1. Data from external and internal sources is uploaded to the WEKA Data Platform by human users or programmatically. The uploaded data might be in files, databases, or streamed.
2. When the embedding service receives a notification of a data upload event, it does the following:
   a. Retrieves data directly from the WEKA Data Platform with CSI drivers through multiple protocols.

b. Reads the uploaded data and preprocesses it using MongoDB Atlas. The preprocessing can include chunking the data and transforming it into a suitable format for embedding generation.

c. Writes the vectorized embeddings to the Milvus vector database on the WEKA Data Platform.

As described in the following section, when the serving subsystem processes user requests, it uses the embeddings in the vector database to retrieve relevant domain-specific data.

Serving Subsystem

The following is the request-response flow in the serving subsystem:

1. A user submits a natural-language request to a front-end server through a web-based chat interface. The frontend server runs on the Run.AI Kubernetes cluster.
2. The frontend server runs a LangChain process that does the following:
   a. Converts the natural-language request to embeddings using the same model and parameters the embedding service uses.
   b. Retrieves relevant grounding data by performing a semantic search for the embeddings in the vector database. Semantic search helps find embeddings based on the intent of a prompt rather than its textual content.
   c. Constructs a contextualized prompt by combining the original request with the retrieved grounding data.
   d. Sends the contextualized prompt to the inference server, which runs on the Run.AI Kubernetes cluster.
3. The inference server uses the NVIDIA Triton Inference Server serving framework to serve an open-source LLM.
4. The LLM generates a response to the prompt, and the inference server sends the response to the frontend server. Prompt memory is implemented to increase the efficiency of the RAG Pipeline.
5. The frontend server sends the filtered response to the user.

Solution Verification

Measuring the efficiency of inferencing in Large Language Models (LLMs) for generative AI applications relies on several key metrics like Latency and Throughput. We focus on using these metrics to help evaluate the performance of LLMs in terms of speed, scalability, and resource utilization. In this study, we will assess the effect of the underlying data platform and storage protocols.

Key Metrics for LLM Inference Efficiency

GenAI-Perf is a command line tool for measuring the throughput and latency of generative AI models as served through an inference server. GenAI-Perf collects a diverse set of metrics that capture the performance of the inference server.

- **Time to First Token:** Time between when a request is sent and when its first response is received, one value per request in benchmark
- **Inter Token Latency:** Time between intermediate responses for a single request divided by the number of generated tokens of the latter response, one value per response per request in benchmark

- **Request Latency:** Time between when a request is sent and when its final response is received, one value per request in benchmark
- **Output Sequence Length:** Total number of output tokens of a request, one value per request in benchmark
- **Input Sequence Length:** Total number of input tokens of a request, one value per request in benchmark
- **Output Token Throughput:** Total number of output tokens from benchmark divided by benchmark duration
- **Request Throughput:** Number of final responses from benchmark divided by benchmark duration

Challenges in Measuring LLM Inference Performance

Several challenges are associated with measuring LLM, including effectively inferring performance metrics, testing consistency, token length variability, and availability of performance data.

- Testing Consistency: Variability in testing conditions, such as different hardware setups or varying prompt types, can lead to inconsistent results.
- Token Length Variability: Different LLMs may have different token lengths, making comparing throughput directly across models challenging.
- Data Availability: Sometimes, inference performance data might not be available for certain models, complicating direct comparisons.

This reference architecture addresses these challenges by instrumenting and implementing observability into the overall application framework so that we can run reproducible and predictable benchmarks and analyze the various factors influencing LLM Inference performance. This also becomes incredibly important in production environments and at scale. By focusing on these metrics and addressing the associated challenges, organizations can effectively measure and optimize the efficiency of LLM inference for generative AI applications.

Conclusion

The WEKA AI RAG Reference Platform (WARRP) presented in this whitepaper offers a robust, infrastructure-agnostic solution designed for ease of deployment and consistent performance across multiple environments. The reference architecture empowers developers and administrators of generative AI (GenAI) applications and cloud architects enabling them to design infrastructure to run a generative artificial intelligence application with retrieval-augmented generation (RAG). By ensuring the ability to run the same workload on different cloud platforms such as AWS, GCP, Azure, OCI and on-premises setups with minimal configuration changes, this architecture enables organizations to achieve workload portability without compromising performance. Whether deployed in public or private cloud settings, the artificial intelligence application pipeline running on WARRP demonstrates stable behavior and predictable results, simplifying hybrid- and multi-cloud operations.

Organizations can achieve significant performance improvements by integrating the WEKA Data Platform in the AI stack, particularly in multi-model inference scenarios. The architecture benefits from WEKA's advanced storage and I/O optimizations, enabling orders of magnitude faster full pipeline execution and acceleration on key metrics. The WEKA Data Platform's ability to load and unload models efficiently further accelerates and efficiently delivers tokens for user prompts, particularly in complex, chained inference workflows involving multiple models.

This architecture will lay the foundation for future expansion and customization. The modular design allows for the seamless integration of additional components such as NVIDIA NIM and application workloads and interchanging components to provide a flexible matrix of key technologies such as cloud platforms, vector databases and Kubernetes distributions for more specialized use cases. The long-term objective remains to create a flexible, truly infrastructure-agnostic framework that can support a wide variety of LLM deployments, ensuring scalability, adaptability, and cutting-edge performance in production environments.

WKA411-01 11/24