Dell Scalable Architecture for Retrieval-Augmented Generation (RAG) with NVIDIA Microservices

March 2024

H19975

Technical White Paper

Abstract

This Dell Reference Design describes a solution for deploying the NVIDIA cloud-native framework for generative AI models on Dell infrastructure. The resulting end-to-end solution is enterprise grade, enabling efficient utilization of resources to achieve better return reducing cost of development and deployment.

Dell Technologies AI Solutions

Dell
Reference Design

Contents

Contents

Introduction

In the era of rapid digital transformation, the demand for sophisticated AI solutions in data centers with robust hardware infrastructure is undeniable. As companies constantly venture into AI technologies capable of processing and interacting with vast volumes of domain-specific data in a manner akin to human cognition, the choice of hardware becomes pivotal. This recognition guides the exploration of NVIDIA's Retrieval-Augmented Generation (RAG) framework, an innovative solution designed to enhance the accuracy and reliability of AI interactions. Now, with NVIDIA AI Enterprise and Dell Technologies products, the complemented RAG framework signifies a leap forward in revolutionizing AI interactions with on-premises Large Language Models (LLM) deployment capabilities from desktop to cloud or data center. It paves the way toward more sophisticated and personalized AI applications that are more accurate, context-aware, and reliable. This partnership between NVIDIA and Dell Technologies highlights a fundamental industry truth: software can only reach its full potential when paired with powerful hardware—a domain where Dell Technologies excels.

Overview

The collaborative partnership between Dell Technologies and NVIDIA has resulted in a spotlighted solution for Generative AI Large Language Models (LLMs), focusing on the Llama 2 model. This model, developed by Meta and available for download from Meta or Hugging Face, is a crucial component of the solution. The deployment of the Llama 2 model within NVIDIA's cloud-native framework, NeMo, is explored, highlighting the combined strengths of both companies in advancing the field of Generative AI and LLMs.

Dell Technologies, recognized globally for leading digital transformation, offers a range of products that perfectly complement the NVIDIA RAG microservice. These products range from PowerEdge servers and networking to PowerScale storage and other infrastructure solutions designed for the data center. This integrated relationship enables the transformative NVIDIA RAG microservice framework to create best-in-class solutions on Dell Technologies' platforms, enhancing the accuracy and reliability of AI-driven workloads in data centers.

A path has been established for businesses to leverage this advanced AI technology from testing to production. The focus is on assisting teams in understanding and overcoming the challenges in deployment and scale. In partnership with NVIDIA, Meta, and Hugging Face, Dell enables customers to deploy LLMs on-premises, from desktop to cloud or desktop to data center, ensuring businesses can leverage the transformative power of AI wherever they are.

By leveraging NVIDIA AI Enterprise, businesses can access technical support for RAG on PowerEdge, which features optimized containers that enhance GPU accessibility. NVIDIA's microservices, including Kubernetes containers for RAG, training, and data curation, have been meticulously optimized at every step of the RAG process. This comprehensive optimization ensures businesses can deploy highly efficient, GPU-accelerated AI solutions, marking a significant advancement in AI-driven Technical Support and interactive applications.

About this document

This document is a guide to enabling cloud-like operational workflow to deploy LLMs with NVIDIA to deliver Generative AI Large Language Models by Meta and Hugging Face. Harnessing the partnership with NVIDIA's cloud-native framework called NeMo within Dell Technologies' ecosystem, businesses can reference this document to deploy highly accurate and reliable AI-driven applications. It aims to provide a Dell reference design with

NVIDIA and Dell's infrastructure to deploy advanced AI solutions. Covering technical insights into RAG architecture, deployment strategies, and practical use cases, this guide empowers organizations with the knowledge to implement this technology effectively. From initial testing to full-scale production, we aim to provide a clear path for leveraging advanced AI technologies, focusing on overcoming deployment and scalability challenges.

Note: The contents of this document are valid for the described software and hardware versions. For information about updated configurations for newer software and hardware versions, contact your Dell Technologies sales representative.

Audience

This document is intended for a technical audience consisting of solution architects, data scientists, data administrators, and IT professionals involved in deploying and managing AI applications. Readers are expected to have a foundational knowledge of AI and machine learning concepts, familiarity with Large Language Models (LLMs), and an understanding of data center operations. The guidance provided is geared towards effectively aiding stakeholders in architecting, deploying, and scaling advanced AI solutions using NVIDIA RAG to enhance the accuracy and reliability of AI-driven chatbots in their organizations.

Revision history (optional)

Table 1. Revision History

Date	Version	Change summary
March 2024	1	Initial release

Background

What are LLMs?

Large Language Models (LLMs) are machine learning models that have revolutionized the field of artificial intelligence by demonstrating an unprecedented ability to understand, develop, and interact with human language. These models, trained on vast datasets, can comprehend nuances, context, and complexities of language, enabling them to perform a wide range of linguistic tasks. They have been used to answer questions, draft essays, summarize texts, translate languages, and generate creative content.

LLMs form the backbone of modern AI applications, from simple automated responses to sophisticated chatbot interactions, embodying the cutting-edge of natural language processing technology. Their ability to understand and generate text in a human-like manner has opened new avenues for human-computer interaction, making them an integral part of many services that require natural language understanding and generation.

LLMs represent a groundbreaking leap in artificial intelligence, reshaping the landscape of human-computer interaction. Therefore, the future of LLMs promises even more significant advancements in our ability to communicate and interact with machines naturally and intuitively.

What is RAG?

Retrieval-augmented generation (RAG) represents a significant advancement in the capabilities of LLMs by incorporating an external data retrieval step into the generative process. This approach allows the model to dynamically fetch relevant information from a vast corpus of data, including user-specific datasets, enriching its responses with accurate, up-to-date, and context-specific content. RAG fundamentally enhances the utility and effectiveness of chatbots by enabling them to access and use external knowledge bases, making them more informative and versatile.

The fundamental concept of RAG lies in its ability to 'connect the dots' between the LLM's general knowledge and the user's specific data. For instance, a user can import a collection of PDFs containing domain-specific knowledge into a database that the LLM can understand. The LLM can then leverage this domain-specific knowledge to generate responses that are not only based on its inherent general understanding but also tailored to the user's specific dataset.

This approach underscores the importance of bringing AI to data rather than the other way around. It emphasizes privacy, security, and using existing data effectively within organizations. RAG enhances the model's responses by allowing a generic LLM to access and use domain-specific data, making it more informative and versatile.

Why RAG and LLM? Significanc of Generative AI Chatbots

Integrating RAG with generative AI chatbots marks a paradigm shift in chatbot development and deployment. Traditional chatbots, constrained by the scope of their training data, often need help with responding to queries requiring specialized or current knowledge accurately. RAG addresses this limitation by empowering chatbots to retrieve and incorporate external data into their responses, improving their accuracy, reliability, and relevance.

The combination of RAG and LLMs presents a powerful solution that bridges the gap between retrieval-based and generative chatbots, creating a private, secure LLM that can parse and index your data within your data center. This synergy addresses security concerns associated with storing private, sensitive information in a cloud with an AI model by reversing the traditional model - it brings AI to your data instead of bringing your data to AI.

Solution overview

Dell and NVIDIA: A New Era for Chatbots

NVIDIA and Dell have joined forces to enhance the capabilities of AI chatbots further. By leveraging NVIDIA's state-of-the-art RAG framework, developers can implement RAG and LLMs with greater efficiency and scalability. This collaboration underscores the benefits of combining industry-leading hardware and advanced AI software. Key features of using NVIDIA's RAG for AI applications include its ability to provide more informed and accurate responses, its scalability to handle large datasets, and its flexibility to adapt to various use cases.

Business Challenges and Use Cases

The integration of NVIDIA AI Enterprise with Dell Technologies' hardware addresses a wide range of business challenges and use cases:

Accelerating AI Initiatives with RAG

Situation: Achieving superior AI capabilities quickly and efficiently is crucial for businesses to stay competitive, as this enables faster and more informed decision-making, streamlines operations, and accelerates development cycles in response to market demands. However, traditional AI development, such as fine-tuning or training and the associated deployment processes, are time-consuming and resource-intensive.

Solution: Using proprietary company data, NVIDIA RAG enables rapid deployment by integrating off-the-shelf large language models, such as Llama 2 13b. This approach shortens the development cycle and yields better-performing chatbots and AI applications in less time than models fine-tuned on the same datasets. Dell Technologies' scalable hardware solutions ensure that these advanced models are supported by robust computing and storage capabilities, facilitating quick and effective AI rollouts.

Streamlined Access to Information: Enhancing Internal Interactions

Situation: Organizations often need help to efficiently disseminate internal policies and knowledge to employees and provide consistent, accurate information over time. This challenge can lead to reduced productivity and compromised operational efficiency.

Solution: By leveraging NVIDIA AI Enterprise, companies can ingest their internal policy documents, FAQs, knowledge bases, and technical information into a vector database, enabling the creation of chatbots that provide immediate, accurate answers. This technique significantly reduces employees' time searching for information, streamlining operational efficiency and boosting overall productivity. Deploying this solution on Dell Technologies' hardware ensures reliable performance across all internal communication channels, even under heavy query loads.

Customer Service Enhancement

Situation: Ensuring customers receive consistent, accurate, and policy-compliant information across service channels can be a daunting challenge for businesses. This difficulty is magnified at scale, where maintaining the quality of customer interactions becomes increasingly complex, potentially consisting of the customer experience.

Solution: Customer-facing chatbots, enhanced with RAG and company-specific data, can deliver high-quality communication, ensuring customers receive correct information about returns, compensation, or problem resolution in line with company policies. This enhances

the customer experience and aligns customer interactions with company standards, ensuring trust and satisfaction.

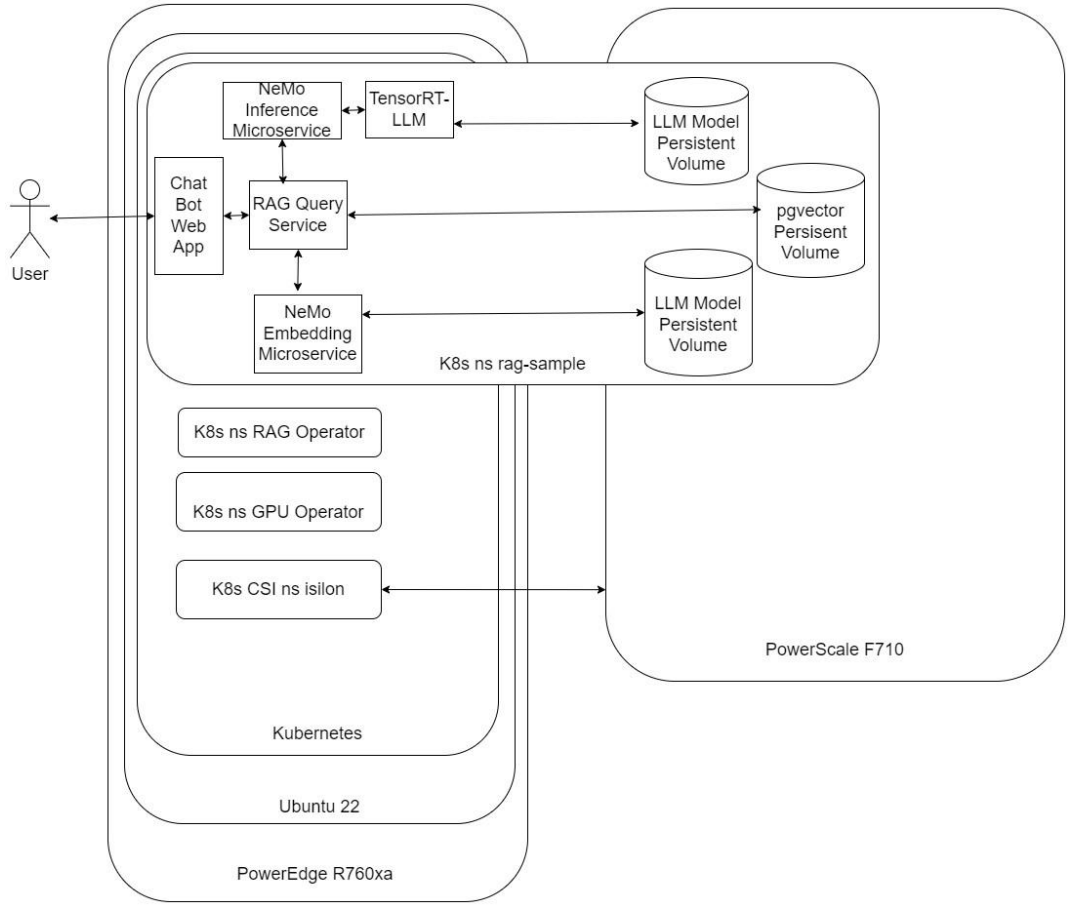
These use cases highlight the practical advantages of deploying NVIDIA RAG on Dell Technologies' infrastructure, showcasing the solutions' ability to address common challenges organizations face in developing, deploying, and scaling AI models. The synergy between NVIDIA's advanced AI technologies and Dell's robust hardware ecosystem mitigates operational challenges. It empowers businesses to harness the full potential of AI to innovate and improve their operations.

Solution approach

Paving the Way for Advanced AI Deployments with NVIDIA RAG and Dell Technologies

The successful deployment of NVIDIA RAG on Dell Technologies' infrastructure requires a strategic approach that carefully balances hardware and software considerations to achieve optimal performance, scalability, and efficiency. A critical aspect of this deployment involves the management of containerized workloads and services, for which we leverage Kubernetes—an open-source platform renowned for its portability, extensibility, and comprehensive support for declarative configurations and automation.

Figure 1. High-level diagram



Kubernetes: Orchestration and Management

Kubernetes is an open-source container orchestration platform that offers several compelling reasons for enterprise adoption. It has simplified deployment and scaling,

improved resource utilization, allowed fault tolerance, self-healing, service discovery, and load balancing, and provided application portability to avoid vendor lock-in. These benefits lay the groundwork for a robust deployment strategy, ensuring that the technical requirements of NVIDIA RAG applications are met with efficiency and scalability at the forefront.

Dell CSI PowerScale for Kubernetes: Optimized Storage for Containerized Environments

Central to our deployment strategy is integrating the pivotal feature from Kubernetes that our solution capitalizes with the Persistent Volume (PV) storage subsystem. This subsystem abstracts the details of storage provisioning and consumption, enabling applications like NVIDIA RAG to efficiently access the persistent storage they require for content, databases, and AI models. This is where Kubernetes' ability to allow users to request storage using Persistent Volume Claims (PVC) becomes invaluable, mainly when using Storage Classes for tailored specific needs.

We harness the capabilities of Dell Technologies Container Storage Modules (CSM) to enable a simple and consistent integration with Kubernetes Container Storage Integration (CSI) with PowerScale OneFS. This leverages the hardware's robustness and scalability to meet the demanding storage requirements of NVIDIA RAG applications. The Container Storage Interface (CSI) driver for Dell PowerScale is essential, facilitating seamless integration and optimized management, provisioning, and scaling of persistent storage within Kubernetes clusters. This ensures our solution benefits from high-performance and scalable storage, highlighting the operational simplicity and consistency Kubernetes' storage management capabilities provide.

NVIDIA Cloud Native Stack and NVIDIA AI Application Framework: Comprehensive AI Platform

Complementing this sophisticated storage and orchestration framework is NVIDIA AI Enterprise, an end-to-end, cloud-native software platform designed to accelerate data science pipelines and streamline the development and deployment of production-grade AI applications.

NVIDIA Cloud Native Stack (formerly Cloud Native Core) is a collection of software designed to run cloud-native workloads on NVIDIA GPUs. The foundation and components are based on Ubuntu, Kubernetes, Helm, NVIDIA GPU, and Network Operator. NVIDIA actively contributes to open-source projects and communities, including container runtimes, Kubernetes operators, and monitoring tools. Applications developed using NVIDIA Cloud Native technologies are cloud-native and enterprise-ready. This includes GPU operators to automate the life cycle management of software required to expose GPUs on Kubernetes. It enhances GPU performance, utilization, and telemetry, allowing organizations to focus on building applications.

NVIDIA AI Enterprise, the software layer of the NVIDIA AI platform, offers 100+ frameworks, pre-trained models, and development tools to accelerate data science and streamline the development and deployment of production AI, including generative AI, computer vision, and speech AI. Its integration into our deployment strategy speeds up time to value with enterprise-grade security, stability, manageability, and support while mitigating the risk of open-source software, ensuring business continuity and a reliable platform for running mission-critical AI applications.

Together, these components create a robust ecosystem to facilitate the rapid deployment and effective scaling of NVIDIA RAG on Dell Technologies' infrastructure. By strategically leveraging Kubernetes for orchestration, Dell Technologies for optimized storage, and NVIDIA for comprehensive AI platforms, organizations can fully leverage AI to drive their mission forward, ensuring a seamless, scalable, and efficient deployment strategy that meets the advanced requirements of modern AI applications.

Large Language Models

The decision to use LLMs was driven by their ability to understand context, generate relevant responses, and interact in a manner that is almost indistinguishable from a human. This level of sophistication allows us to provide users with a more engaging and efficient service.

Why Llama-2-13B-Chat Model?

Among the various LLMs available, deploying with the Llama-2-13B-Chat model was based on several key factors:

1. **Performance**: The Llama-2-13B-Chat model has superior performance in tests, outperforming open-source chat models on most benchmarks. Its performance is on par with popular closed-source models like ChatGPT and PaLM.
2. **Optimization for Dialogue**: Unlike many LLMs, the Llama-2-13B-Chat model is fine-tuned explicitly for dialogue use cases. This means it can provide accurate and contextually relevant responses in a conversational setting.
3. **Advanced Training Techniques**: The model uses supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF), aligning it closely with human preferences for helpfulness and safety.

Deploying Different Inference Models

While we have chosen the Llama-2-13B-Chat model, it is important to note that organizations can deploy different inference models based on their specific needs from your organization and team NGC Private Registry. The choice of model can be influenced by factors such as the task's nature, the required accuracy level, and the computational resources available. Additional models can be found in NVIDIA's NGC Private Registry at https://registry.ngc.NVIDIA.com/models.

Document Ingestion

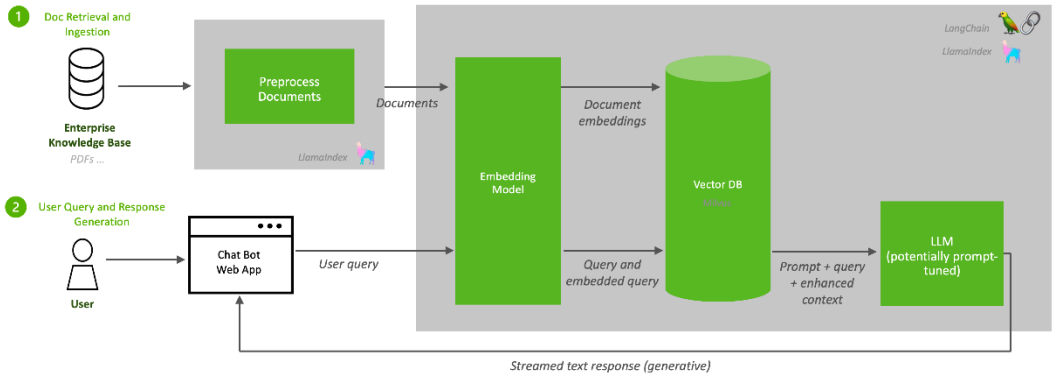
The first function of a RAG pipeline is to populate a repository with information from enterprise data sources. A large language model can only answer questions from the data used to train it. RAG supplements an LLM's foundational information with a database of additional knowledge that is more up to date, more relevant, or proprietary to the data owner.

Popular LLM programming frameworks such as LlamaIndex and LangChain provide connectors to many familiar enterprise data sources. This example pipeline uses an unstructured connector to input unstructured data such as text documents and PDFs.

A preprocessor prepares the data before it is added to the RAG database. Data preprocessing often determines the quality and relevancy of the data retrieved from the database to respond to a query. The preprocessor in this example splits the data into

chunks based on sentence length. Other preprocessing activities include anonymization, deduplication, or toxicity filtering.

The processed data is then sent to the embedding microservice. Embedding converts data chunks into vector representations that can be efficiently searched and retrieved. This example uses the NV-Embed-QA embedding model, developed by NVIDIA, for efficient, GPU-accelerated document embedding.



The embeddings are then stored in a vector database for efficient search and retrieval through indexing. Many commercial and open-source vector databases have various cost/performance/reliability tradeoffs. This example pipeline uses PGvector, a vector database implementation of the popular Postgres SQL database. PGvector supports several standard vector search and vector indexing approaches.

The PGvector database deployed in this reference design is backed by Dell PowerScale Network Attached Storage (NAS) for data persistence. The volume is exposed to the PGvector pod as a Kubernetes persistent volume claim from the Isilon storage class.

While PGvector is not GPU accelerated, the reference design supports using several GPU-accelerated vector databases such as Milvus, FAISS, and Redis.

Also note that, although it is not shown in this reference design, RAG administrators should implement a process for regular or continuous data ingestion to keep the database up to date.

Response Generation

Response generation is the RAG function that answers user queries once the pipeline is deployed into production. The Nemo Inference Microservice (NIM) generates the responses through LLM inference. The NIM uses the Llama-2-13b model in this example pipeline to create responses.

Next, the LLM initiates a search and retrieves relevant data from the vector database to enhance its response. In this design, the user query is vectorized using the same NV-Embed-QA model used during the document ingestion phase.

Vectorizing the query with the same embedding model facilitates efficient similarity search of the data embeddings. A critical distinction between RAG and traditional keyword search is that the vector database performs a semantic search to retrieve vectors that most closely resemble the intent of the user's query. The vectors are returned to the LLM as context to enhance the response generation. The LLM generates an answer streamed to the user and citations to the retrieved data chunks.

Although not implemented in this reference design, the pipeline supports prompt tuning to enhance retrievals' accuracy and relevance.

Solution design

The solution consists of a cloud-like model with Dell Technologies providing the infrastructure of PowerEdge with NVIDIA GPUs, running Ubuntu Server. Kubernetes simplifies management and connects to PowerScale using the Container Storage Interface driver to provide persistent LLM and RAG volumes. NVIDIA Cloud Stack and NVIDIA AI Enterprise simplify model management and deployment at scale.

Hardware design

Server

PowerEdge R760xa is a high-performance, scalable server for intensive GPU applications. The R760xa is a purpose-built server designed to boost acceleration performance for AI workloads like inferencing and RAG. We use two R760xa servers, each containing 2x NVIDIA H100 GPUs for 4 GPUs across 2 R760xa's. The PowerEdge R760xa can connect to the PowerScale using high-speed 100 GB Ethernet networking. A third server, such as the PowerEdge R660, should be used to create a 3-node Kubernetes deployment.

Storage

PowerScale F710 running OneFS 9.7 was selected due to its ability to handle demanding workloads. PowerScale with OneFS 9.7 provides a flexible, secure, and efficient scale-out NAS solution. PowerScale can deliver high-performance data access for applications like AI/ML. The OneFS operating system provides the intelligence behind the highly scalable, high–performance modular PowerScale storage solution that can grow with your business.

Networking

Dell PowerSwitches are used throughout the build for different types of traffic. The PowerSwitch N3248TE-ON is used for the Out of Band network to connect to iDRACs and management ports. The PowerSwitch S5248F-ON is used for in-band North-South server traffic to handle front-end client connections to Ubuntu and Kubernetes front-end service IPs. The PowerScale NAS has 100Gb Z9664F-ON switches to support NAS front-end traffic and different Z9664F-ON for backend traffic.

Figure 2. Network diagram

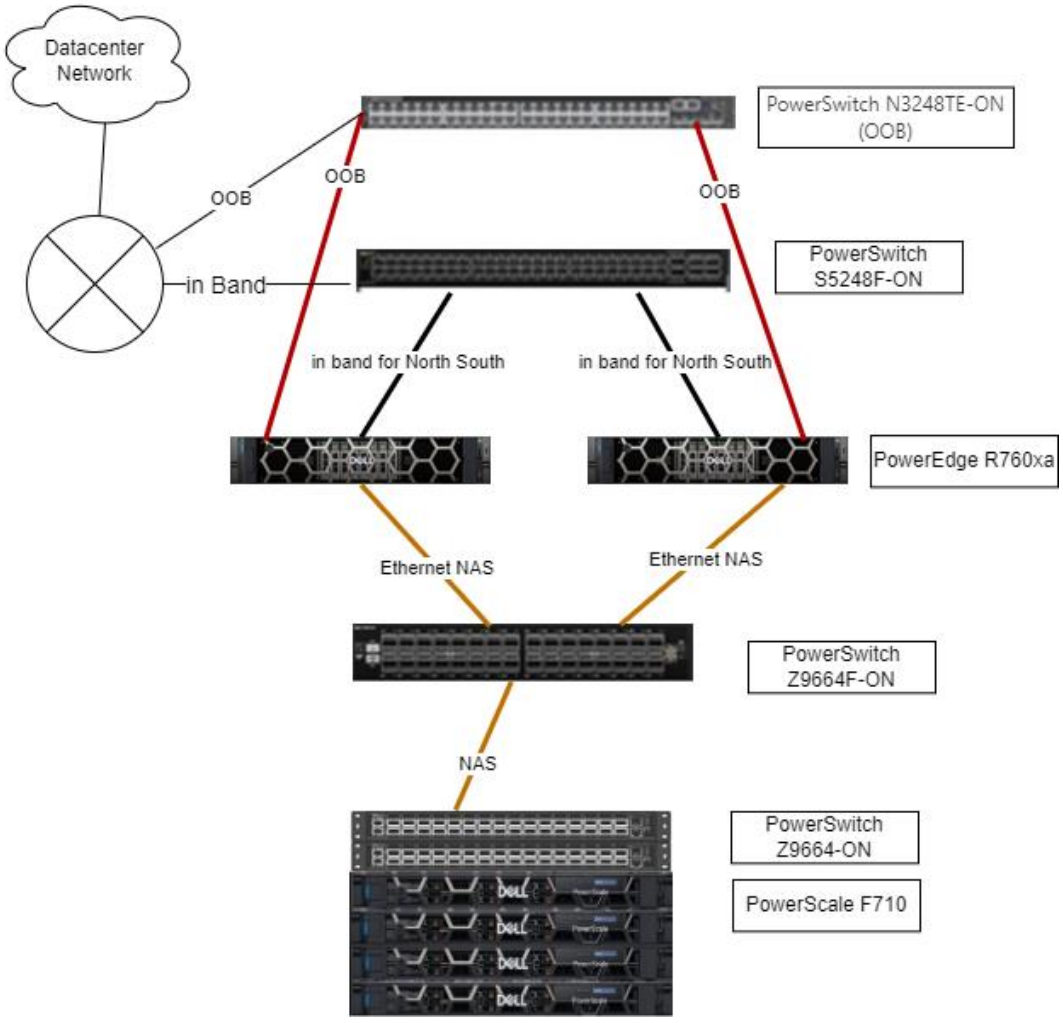
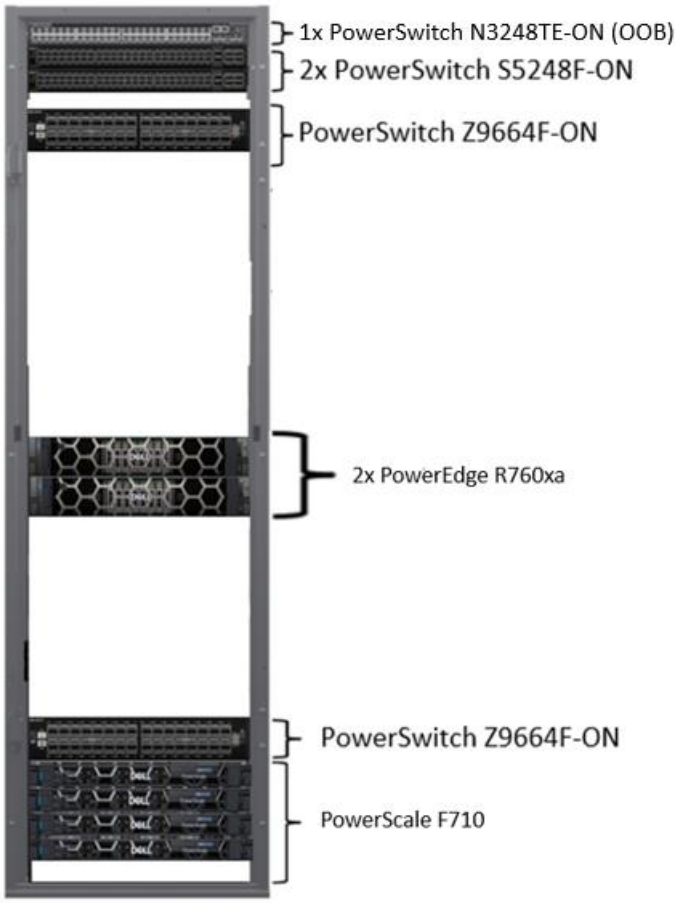


Figure 3. Rack Configuration



Software design

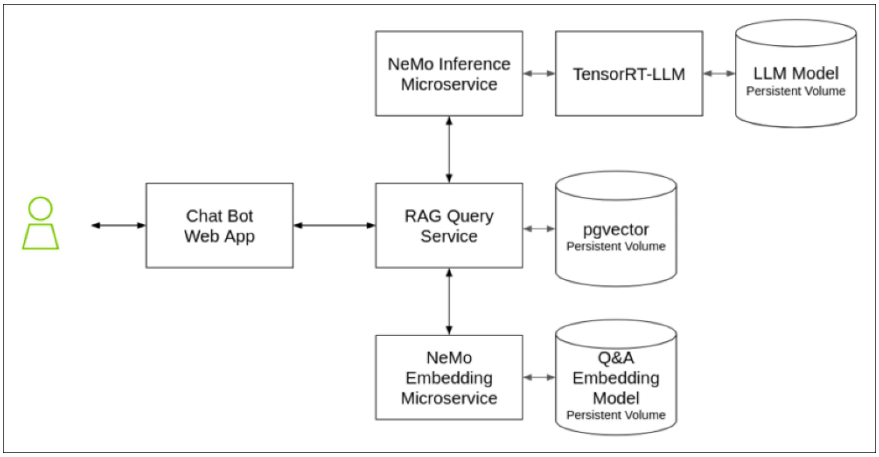
AI or Solution Software

NVIDIA Enterprise RAG has a modular architecture combining popular open-source LLM programming frameworks like LangChain, LlamaIndex, and Hugging Face with NVIDIA GPU acceleration. This architecture helps enterprises develop and deploy scalable production chatbots quickly and easily without sacrificing open-source innovation.

NVIDIA Cloud Native Stack

A collection of software formerly known as Cloud Native Core, Cloud Native Stack can assist with deploying NVIDIA GPU Operator and has cloud-native workload examples.

NVIDIA Enterprise RAG Text QA Pipeline Architecture



Software architecture for the Text QA chatbot RAG pipeline deployed. The NVIDIA RAG LLM Operator also includes sample pipelines for everyday RAG use cases, including Multi-modal QA, Event-driven Agents, Structured Data Processing, and Multi-turn conversation chatbots.

The RAG pipeline includes the following software components:

- **Frontend** – A sample chatbot web interface implemented in Gradio. It supports text query, document upload, document retrieval, and citation.
- **Query router** – A sample RAG application implemented as LangChain runnable with LlamaIndex retrieval functions. The query sends API requests to the inference and embedding microservices and is also used for document preprocessing, indexing, and retrieval.
- **NVIDIA NeMo LLM Inference microservice** (NIM) – built on NVIDIA using TRT-LLM and Triton inference server, NIM delivers best-in-industry performance and scalability for LLM inference. It supports OpenAI-compatible REST APIs for LLM chat and completion. It also supports LLM architectures like Llama, Hugging Face, Gemma, Mistral, and NVIDIA proprietary model formats.
- **NeMo Retriever Embedding Microservice (NREM)** -- The NeMo Retriever microservices simplify and accelerate the document embedding, retrieval, and querying functions at the heart of a RAG pipeline. This reference design uses the NeMo Embedding microservice to GPU-accelerate vector embedding computations. Future versions of this DRD will incorporate additional NeMo Retriever microservices as they become available.
- **PGvector database** – PGvector adds vector similarity search to the popular Postgres open-source SQL database. It supports exact and approximate nearest neighbor search, HNSQ, and IVF Flat index types for L2 distance, inner product, and cosine distance vector comparison.
- **NVIDIA RAG LLM Operator** enables quick and easy deployment of RAG application pipelines into Kubernetes clusters. NVIDIA customers can deploy their pipelines to any on-premises or cloud-based Kubernetes cluster by standardizing RAG pipeline deployment to the Kubernetes operator pattern without modification.

This reduces the complexity of life cycle management and enables seamless connection, scaling, and deployment of RAG pipelines without cloud or vendor lock-in.

- **NVIDIA API catalog** - RAG developers can experience the benefits of GPU acceleration on this NVIDIA-hosted LLM evaluation platform. The API Catalog includes APIs to power every stage of a RAG pipeline that benefits from GPU acceleration. Developers can start developing their RAG applications on the API Catalog and export the required models as NIM containers to deploy them on-premises without rewriting any code.
- **Llama-2-13b-chat** - The Llama 2 family of LLMs are pretrained and fine-tuned generative text models from Meta. This example uses a 13 billion parameter chat model optimized for dialogue. The model is compiled as a TensorRT engine for efficient inference on NVIDIA GPUs and easy integration with the NVIDIA Triton inference server to support multi-GPU tensor and pipeline parallelism.

Prerequisites

This reference design was tested on the following hardware and software:

Component	Version
NVIDIA GPUs	H100, L40S, A100-80
NVIDIA GPU driver	535.129.03
Operating System	Ubuntu Linux 22.04
Kubernetes	1.26-1.28
Container runtime	contained 1.6, 1.7
Ubuntu	22
Server	R760xa
Storage	PowerScale F710 OneFS 9.7
Network	PowerSwitch Z9664F-ON/S5248F-ON
Dell CSI PowerScale Driver	2.9.1

Implementation guidance

This section will provide step-by-step guidance on implementing the outlined hardware and software solution, from initial setup to full-scale deployment, ensuring organizations can effectively leverage NVIDIA RAG on Dell Technologies hardware for their AI initiatives.

Summary of deployment steps

- Base Infrastructure and operating system
- Kubernetes
- Dell CSI Driver for PowerScale
- RAG Sample pipeline

Initial Setup and Configuration

- Infrastructure deployment
    - Deploy Ubuntu and relevant drivers
    - Ensure PowerEdge R760xa can communicate to PowerScale
- Kubernetes can be deployed with NVIDIA's Cloud Native Stack GitHub repository or with your preferred method.
    - https://github.com/NVIDIA/cloud-native-stack/blob/master/install-guides/Ubuntu-22-04_Server_x86-arm64_v11.0.md#Installing-Kubernetes
    - Verify you have a GPU operator, and nodes show GPUs are available in the context of Kubernetes

Software Installation and Deployment

- Dell CSI Driver for PowerScale
    - Verify you have the CSI driver installed in Kubernetes, which is unique for this solution. A deployment link can be found below. PowerScale will be used to store the RAG model persistently.
    - https://dell.github.io/csm-docs/docs/csidriver/installation/helm/isilon/
    - You have a Kubernetes storage class called "Isilon."
    - Verify that you can deploy the workload using the Storage Class.
- RAG Sample pipeline
    - https://docs.NVIDIA.com/ai-enterprise/rag-llm-operator/0.4.1/pipelines.html

- In this sample RAG pipeline, change StorageClass from "local-path" to "Isilon" in the following three files. See Appendix.
  - pvc-embedding.yaml
  - pvc-inferencing.yaml
  - pvc-pgvector.yaml
- In the helmpipeline_app.yaml file, change the StorageClass from "local-path" to "Isilon." Change accessMode to ReadWriteMany. You must also add your NGC API Key for the password and the secret apiKey. See Appendix.

Scaling and Management

- **Monitoring:** Regularly check the status of your Kubernetes environment and the CSI driver.
- **Upgrading/Updating:** Keep your Kubernetes environment and the CSI driver up to date to ensure optimal performance and security.
- **Backup and Disaster Recovery Planning: Regularly back up** your data and have a disaster recovery plan to protect against data loss.

Redeployments and Expansions

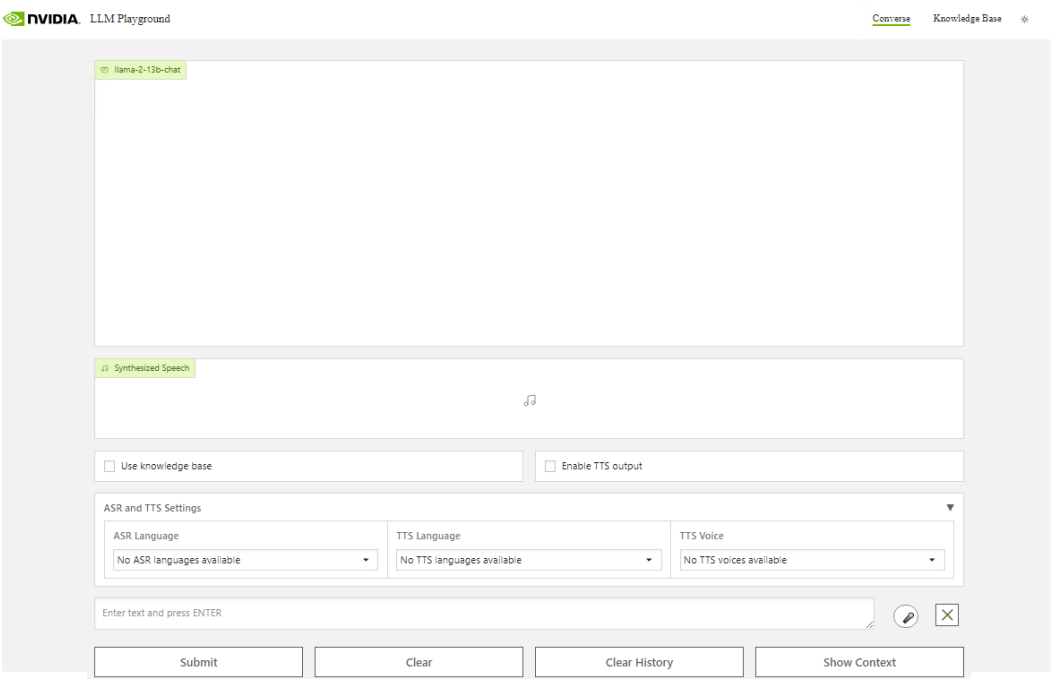
- As your needs evolve, you may need to redeploy or expand your cluster. This could involve adding more nodes to your Kubernetes environment, expanding your storage with PowerScale, or scaling up your use of the RAG model.
- All components of this solution are designed with ease of scaling in mind. Utilizing Kubernetes can add hardware resources with little or no disruption to the running deployment.
- GPU resources are primarily consumed as concurrent or active inferences increase. Adding additional Servers with GPUs allows a cluster to expand, and demand grows. For specific scaling guidance, consult your Dell Technologies representatives.

Results or findings

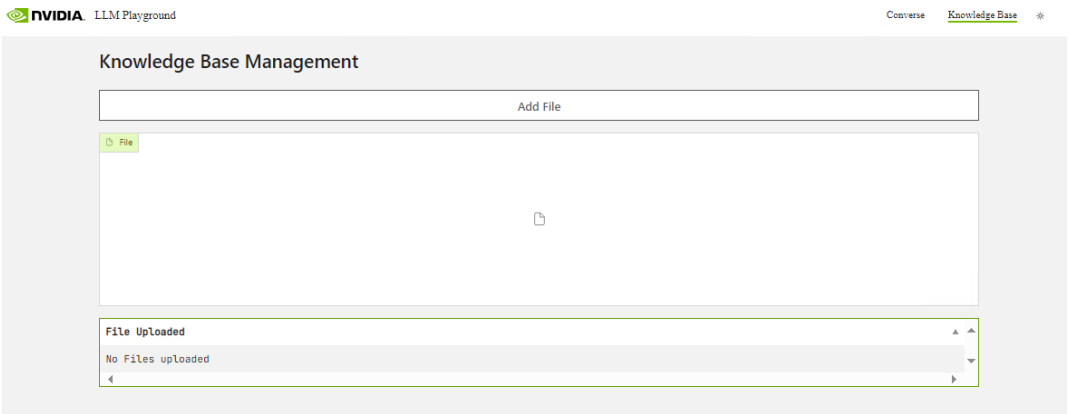
Implementing NVIDIA's RAG on Dell Technologies' hardware will drive significant improvements in AI-driven applications, impacting operational efficiency, customer satisfaction, and innovation capacity.

Test results

We can now view the front end of the RAG LLM Playground. We have Llama-2-13b-chat running on-premises. We can ingest documents for queries.

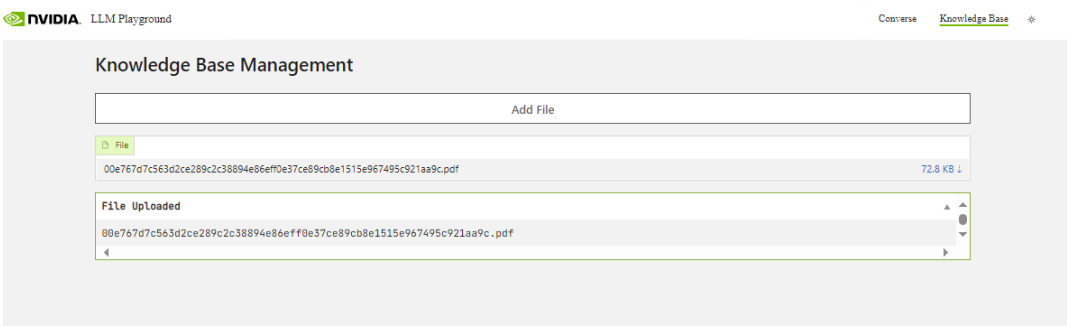


To add documents to be queried by the LLM, Select Knowledge Base.

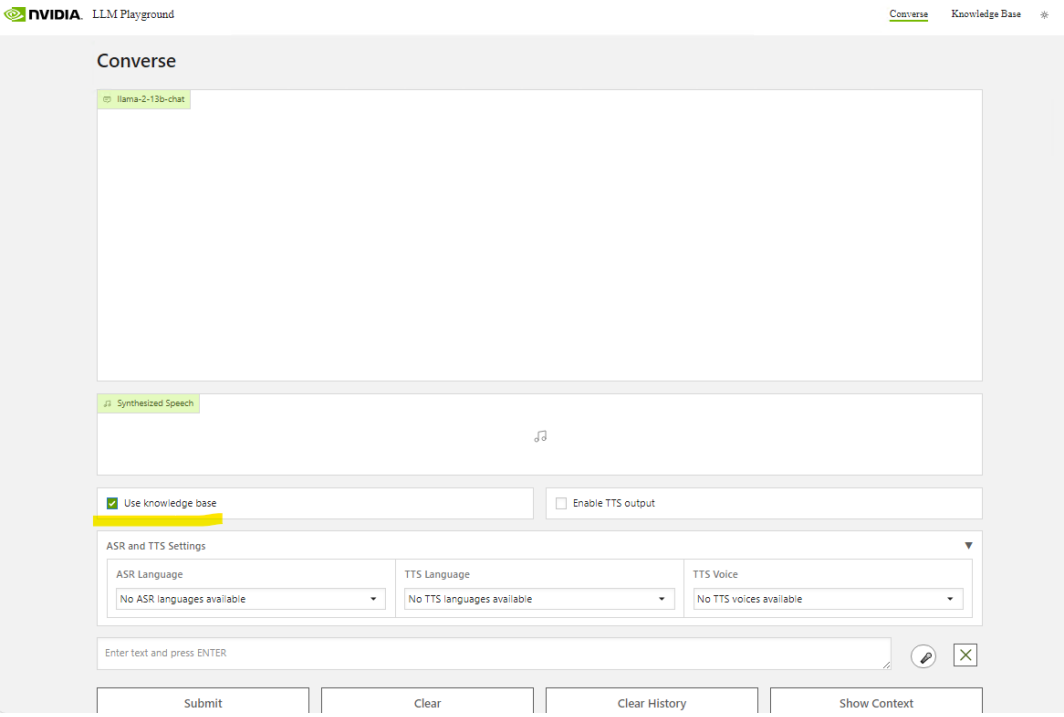


Select Add File to add PDFs to the RAG LLM.

Output:



Select Converse and begin to ask it questions about your data by selecting the checkbox for "Use Knowledge Base."



We can now ask the LLM about the information contained within our documents.

Note: You can also import documents by installing python notebook or through the FASTAPI.

Python Notebook:

Install Python Notebook (pip install notebook)

Jupyter Notebook Link:

https://github.com/NVIDIA/GenerativeAIExamples/blob/main/notebooks/05_dataloader.ipynb

Copy the contents of the dataloader notebook into your notebook file. Then, put your PDFs in a data folder or edit the path to the data in the notebook.

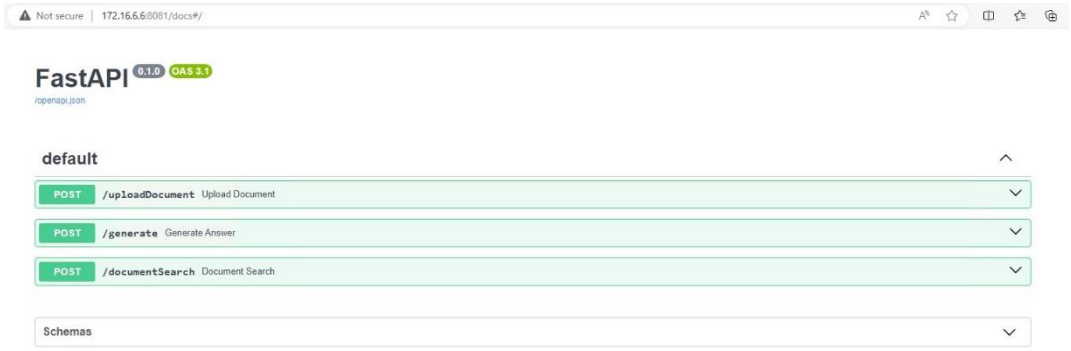

```
[2]: import time

start_time = time.time()
NUM_DOCS_TO_UPLOAD=3000
upload_pdf_files("data", "http://172.16.6.6:8081/uploadDocument", NUM_DOCS_TO_UPLOAD)
print(f"--- {time.time() - start_time} seconds ---")
```

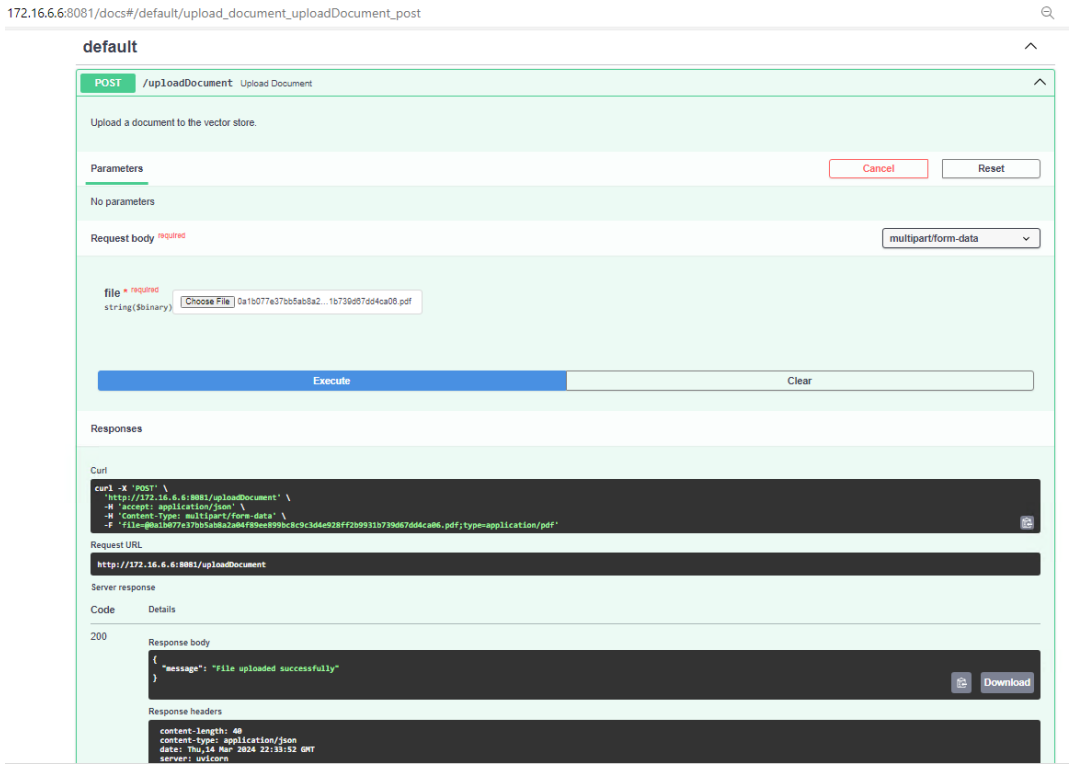
```
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
{"message":"File uploaded successfully"}
--- 14056.464277029037 seconds ---
```

To use the FASTAPI method to ingest documents, do a port forward on the query router.

```
demo@k8s-master:~$ kubectl port-forward --address 172.16.6.6 query-router-975d48c85-jbn9c 8081:8081 -n rag-sample
Forwarding from 172.16.6.6:8081 -> 8081
Handling connection for 8081
Handling connection for 8081
Handling connection for 8081
Handling connection for 8081
Handling connection for 8081
```

Results or findings





Conclusion

As AI continues to evolve, staying ahead of future directions and emerging trends is crucial for organizations looking to maintain a competitive edge. In this paper, we have demonstrated the deployment of Generative AI workloads on Dell infrastructure in collaboration with NVIDIA. From initial considerations and implementation strategies to anticipated outcomes and future directions, we successfully traversed the landscape of deploying RAG. This partnership has enabled us to leverage the power of Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) to enhance AI chatbot capabilities.

Organizations must evaluate their AI capabilities and consider how integrating NVIDIA RAG and Dell Technologies hardware can elevate their operations and strategic initiatives. For further exploration and assistance, readers are directed to contact Dell Technologies and NVIDIA for consultations, technical support, and access to a wealth of resources designed to facilitate the successful deployment of AI Solutions.

For more information, reach out to your Dell contact.

We value your feedback

Dell Technologies and the authors of this document welcome your feedback on this document and the information that it contains. Please contact the Dell Technologies Solutions team by email.

Authors: Benjamin Gordy, Justin Potuznik, Tiffany Fahmy

Contributors: Jacob Liberman (NVIDIA), Paul Montgomery, Bryan Mcfeeters, Dell AI Technical Marketing Team

References

Dell Technologies documentation

The following Dell Technologies documentation provides additional and relevant information related to this solution.

- *Dell CSM CSI PowerScale Driver for Kubernetes Installation*
- *Dell Technologies Info Hub for AI Solutions*
- *Dell Generative AI Solutions (Dell.com/ai)*
- *White Paper – Generative AI in the Enterprise*
- *Design Guide – Generative AI in the Enterprise - Inferencing*

Partner documentation

The following NVIDIA documentation provides additional and relevant information:

- *NVIDIA Cloud Native Stack*
- *NVIDIA GPU Operator*
- *NVIDIA Enterprise RAG LLM Operator*
- *NVIDIA Sample RAG Pipeline*
- *NVIDIA AI Enterprise*

Appendix A - File to review

Code updates

1. Update PVC files to "Isilon" storage class

```
cd /rag-sample-pipeline_v0.4.0/examples/

cat pvc-embedding.yaml pvc-inferencing.yaml pvc-pgvector.yaml |
grep storageClassName
```

Output:

```
  storageClassName: Isilon

  storageClassName: Isilon

  storageClassName: Isilon
```

2. Update Storage Class to Isilon in helmpipeline_app.yaml

```
grep -i storageclass /home/demo/rag-sample-
pipeline_v0.4.0/config/samples/helmpipeline_app.yaml | grep -v "#"
```

Output:

```
        storageClass: "Isilon"
        storageClass: "Isilon"
          storageClass: "Isilon"
```

3. Update the Storage AccessMode to: ReadWriteMany

```
grep -i accessMode /home/demo/rag-sample-
pipeline_v0.4.0/config/samples/helmpipeline_app.yaml
```

Output:

```
        accessMode: ReadWriteMany  # If using an NFS or similar
setup, you can use ReadWriteMany
        accessMode: ReadWriteMany  # If using an NFS or similar
setup, you can use ReadWriteMany
          accessMode: ReadWriteMany  # If using an NFS or similar
setup, you can use ReadWriteMany
```