

# Digital Assistant on Dell APEX Cloud Platform for Red Hat OpenShift with Red Hat OpenShift AI

Leveraging Retrieval Augmented Generation (RAG), Large Language Model (LLM) and Dell ObjectScale

July 2024

H20066

## Design Guide

### Abstract

This design guide describes the architecture and design of the Dell Technologies Validated Design for deploying a digital assistant on Dell APEX Cloud Platform for Red Hat OpenShift using Red Hat OpenShift AI and Dell ObjectScale object storage. This solution leverages a LLM and the RAG technique in combination with a set of vectorized documents.

Dell Technologies Solutions

## Copyright

The information in this publication is provided as is. Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2024 Dell Inc. or its subsidiaries. All Rights Reserved. Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Intel, the Intel logo, the Intel Inside logo and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Other trademarks may be trademarks of their respective owners. Published in the USA 07/24 Design Guide H20066.

Dell Inc. believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

# Contents

|  |           |
|--|-----------|
| <b>1. Introduction</b>   | <b>5</b>  |
| Introduction .....   | 6         |
| Business challenge .....   | 6         |
| Solution introduction.....   | 7         |
| Solution benefits .....  | 8         |
| Design guide introduction.....                                     | 9         |
| Terminology .....  | 9         |
| <b>2. Solution Concepts</b>  | <b>10</b> |
| Overview .....   | 11        |
| <b>3. Solution Architecture and Requirements</b>                   | <b>18</b> |
| Physical architecture .....  | 19        |
| Logical architecture .....   | 21        |
| Digital assistant design .....                                     | 22        |
| Solution requirements .....  | 23        |
| <b>4. Solution Deployment</b>                                      | <b>25</b> |
| Introduction .....   | 26        |
| Dell APEX Cloud Platform for Red Hat OpenShift configuration ..... | 26        |
| Pre-requirements .....   | 26        |
| Dell ObjectScale .....   | 27        |
| Digital Assistant Solution components deployment .....             | 27        |
| Deployment summary .....   | 35        |
| <b>5. Solution Validation</b>                                      | <b>36</b> |
| Digital Assistant demonstration .....                              | 37        |
| LLM load testing.....  | 38        |
| <b>6. Summary and Conclusion</b>                                   | <b>41</b> |
| Summary .....  | 42        |
| We value your feedback.....  | 42        |
| <b>7. References</b>   | <b>43</b> |
| Dell Technologies documentation .....                              | 44        |
| Red Hat documentation .....  | 44        |
| Llama 2 documentation .....  | 44        |
| LangChain documentation .....                                      | 44        |

Vector store documentation .....44

Gradio documentation .....44

vLLM documentation .....45

**Appendix A Open-Source License 46**

Open-Source Licensing Information .....47

# 1. Introduction

This chapter presents the following topics:

|                                       |          |
|---------------------------------------|----------|
| <b>Introduction .....</b>             | <b>6</b> |
| <b>Business challenge.....</b>        | <b>6</b> |
| <b>Solution introduction.....</b>     | <b>7</b> |
| <b>Solution benefits.....</b>         | <b>8</b> |
| <b>Design guide introduction.....</b> | <b>9</b> |
| <b>Terminology .....</b>              | <b>9</b> |

## Introduction

LLMs are highly sophisticated AI models that are designed to understand and generate human-like text, enabling a wide range of natural language processing applications. One limitation of LLMs is that once they are trained and generated, they do not have access to information beyond the date that they were trained. Retrieval Augmented Generation (RAG) can extend the functionality of the LLMs by retrieving facts from an external knowledge base hosted using a vector database such as Redis or PGVector.

In this solution, we have deployed an LLM-based digital assistant that provides answers to user questions. These answers remain up-to-date and contain information unique to the organization by augmenting the model with relevant documentation. It leverages the data science pipelines in Dell APEX Cloud Platform for Red Hat OpenShift AI to ingest the data periodically. It also offers advanced features such as choice of different LLMs, different vector databases, along with options to change the hyperparameters within the user interface.

## Business challenge

Artificial Intelligence (AI) is evolving rapidly and is becoming critical for businesses to remain competitive. Because of compliance challenges and to preserve confidential data, enterprise customers deploy different AI and GenAI workloads in controlled on-premises environments, such as data center, edge, and colocation. However, choosing the right platform, technology, and architecture can be challenging.

Selecting the most suitable platform for AI workloads can be a complex task, considering the multitude of available options. An ideal platform should be cloud-native, software-defined, and offer seamless integration with hardware to manage, upgrade, and monitor. The platform should also provide scalability and facilitate ease of use for both developing and deploying AI workloads.

AI applications must scale to handle large datasets and accommodate an increase in user loads. Achieving scalability becomes challenging when the architecture lacks a clear separation between control, compute, block, and object storage nodes. This design limitation hinders the system's ability to adapt to varying workload and storage requirements.

Organizations often have trouble searching through their internal knowledge base, as they are limited to text-based search, which does not offer efficient querying. RAG-based search capability offers an intelligent, context-aware search method that generates comprehensive answers to users.

# Solution introduction

## Overview

This solution is designed to create a cloud-native AI application, with a focus on ease of deployment and manageability.

A text-based search is a technique that looks for matches of a query within the entire text of a set of documents. It searches for all occurrences of the words or phrases that are specified in the query, regardless of the context within the document. Text-based search could provide a less accurate outcome as it mainly considers words or phrases for retrieving the relevant information, while a RAG-based digital assistant can retrieve more accurate and relevant results using semantic search. Semantic search is a more advanced search technique which takes the context, synonyms, related terms, and overall meaning of the query into consideration to retrieve more relevant information.

This solution is designed using separate nodes for compute plane, block, and object storage, which enables nodes to scale independently based on compute and storage requirements.

Dell APEX Cloud Platform for Red Hat OpenShift is designed collaboratively with Dell Technologies and Red Hat to optimize and extend OpenShift deployments on-premises with an integrated operational experience. By combining Dell's expertise in delivering robust infrastructure solutions, with Red Hat's industry-leading [OpenShift Container Platform](#), this collaboration empowers organizations to start a transformative journey towards modernization and innovation.

Red Hat OpenShift Container Platform is a trusted and proven application platform that helps alleviate the complexity in cloud-native infrastructure. It provides a turnkey solution which offers a robust containerization platform and Kubernetes-based orchestration framework. These benefits enable organizations to build, deploy, and manage applications and databases across on-premises and multicloud environments.

Red Hat OpenShift AI is a flexible, scalable AI and ML platform that enables enterprises to create and deliver AI-enabled applications at scale across hybrid cloud environments. Built using open-source technologies, OpenShift AI provides trusted, operationally consistent capabilities for teams to experiment, serve models, and build and deliver innovative apps. The ML models developed using Red Hat OpenShift AI are portable to deploy in production, on containers, on-premises, at the edge, or in the public cloud.

Dell PowerFlex is a comprehensive and adaptable software-defined infrastructure that offers flexible deployment options such as Hyperconverged, two-layer deployment, compute only, storage only. We have leveraged the PowerFlex for the block storage requirement of our workloads, such as Vector store, OpenShift AI workbench and data science pipelines. PowerFlex delivers consistent, predictable outcomes at large scale for the most demanding mission-critical environments. The self-healing architecture of PowerFlex guarantees extreme application performance with minimal downtime.

Utilizing object storage for LLMs and large datasets is crucial for modern data-driven applications, due to their reliability and scalability needs. Dell ObjectScale object storage is leveraged in this solution to store Llama 2 model, relevant datasets, and artifacts. Dell ObjectScale is a software-defined enterprise-grade object storage. Built with a scale-out

architecture, ObjectScale clusters can expand from a few terabytes to petabytes and beyond without limits on the number of object stores, buckets, or objects.

## Solution benefits

A RAG-based digital assistant offers numerous benefits to the organization. Some key benefits include:

- **Enhanced Accuracy and Relevance:** By relying on organizations internal data, the digital assistant can deliver responses tailored to the specific context and operations of the organization. These benefits improve the accuracy and relevance of the information provided, reducing hallucination or misinformation.
- **Confidentiality and Security:** Organizations are looking to preserve their business critical and confidential data. By deploying the digital assistant solution in their on-premises environment, they can avoid sending their data to cloud or third-party APIs.
- **Efficiency and Productivity:** By quickly retrieving information from internal sources, employees can save time and effort when seeking answers. This benefit can boost productivity and reduce the need for lengthy manual and custom searches.
- **Consistent Knowledge Base:** RAG-based Digital assistant ensures up-to-date information and consistent response by referencing an internal knowledge base. This benefit reduces the risk of contradicting information and helps drive uniformity in user responses throughout the organization.
- **Flexibility:** This digital assistant offers advanced parameters that can tailor the response based on user needs and provides choices of multiple vector stores, catalogs, and LMs.

### Document purpose

The purpose of this document is to provide design guidance for deploying a digital assistant on Dell APEX Cloud Platform for Red Hat OpenShift with Red Hat OpenShift AI.

---

**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 design guide is intended for AI solution architects, data scientists, data engineers, IT infrastructure managers, and IT personnel who are interested in, or considering, implementing AI and ML deployments.

### Disclaimer

This document provides design guidance on deploying a digital assistant using open-source tools. It is the responsibility of the user to review and comply with the licensing terms and conditions for each tool mentioned in this document. Licensing information for each tool can be found on the respective tool's official website or in its documentation. See [Appendix A](#) to access the licensing page for some of the open-source tools used during validation of this solution.



## Design guide introduction

Dell validated solutions offer customers faster time-to-market and reduced risks compared to creating their own solutions. For this solution, Dell systems engineers validated a digital assistant on Dell APEX Cloud Platform for Red Hat OpenShift with Red Hat OpenShift AI.

This design guide first outlines the concepts involved in developing the solution and briefly explains the key elements of the solution such as architecture and design. It further elaborates on the deployment and configuration of Red Hat OpenShift AI. The guide also delves into the deployment, configuration, and validation of the digital assistant. Additionally, it elaborates on the results from the LLM load test conducted on the Llama 2 13B model, which is part of this solution.

## Terminology

The following table provides definitions for some of the terms that are used in this document.

**Table 1. Terminology**

| Term        | Definition                               |
|-------------|--|
| API         | Application Programming Interface        |
| RAG         | Retrieval Augmented Generation           |
| NLP         | Natural Language Processing              |
| LLM         | Large Language Model                     |
| vLLM        | Virtual Large Language Model             |
| LLaMA       | Large Language Model Meta AI             |
| Red Hat OCP | Red Hat OpenShift Container Platform     |
| RHCOS       | Red Hat Enterprise Linux CoreOS          |
| RHOAI       | Red Hat OpenShift AI                     |
| S3          | Simple Storage Service                   |
| DC          | Domain controller                        |
| DNS         | Domain Name System                       |
| FIPS        | Federal Information Processing Standards |
| ML          | Machine Learning                         |
| NFD         | Node Feature Discovery                   |
| PVC         | Persistent Volume Claim                  |
| SDS         | software-defined storage                 |
| CSM         | Container Storage Module                 |

## 2. Solution Concepts

This chapter presents the following topics:

**Overview** .....11

## Overview

This section discusses the concepts involved in building the solution. The following list includes the hardware and software layers in the solution stack.

- Dell APEX Cloud Platform for Red Hat OpenShift
- Dell PowerFlex
- Dell ObjectScale
- Dell PowerSwitch
- Red Hat OpenShift Container Platform
- Red Hat OpenShift AI
- Digital assistant related components (Llama 2, LangChain, embedding model, Vector store, Gradio, and vLLM)

### Dell APEX Cloud Platform

Dell APEX Cloud Platforms deliver innovation, automation, and integration across your choice of cloud ecosystems, empowering innovation in multicloud environments.

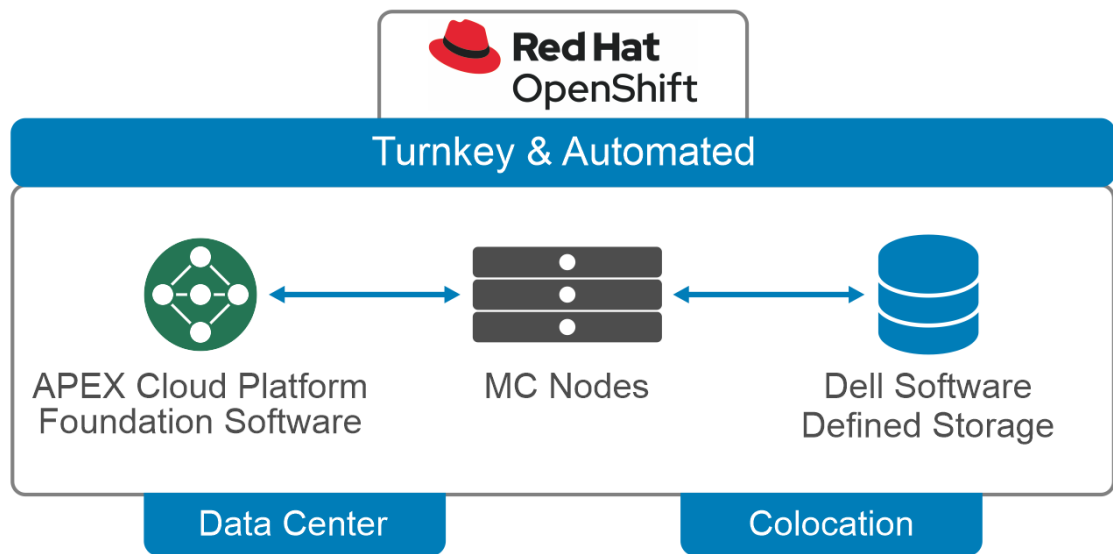
Dell APEX Cloud Platforms are a portfolio of fully integrated, turnkey systems integrating Dell infrastructure, software, and cloud-operating stacks that deliver consistent multicloud operations. This portfolio extends cloud operating models to on-premises and edge environments.

Dell APEX Cloud Platform provides an on-premises, private cloud environment with consistent full-stack integration for the most widely deployed cloud ecosystem software including Microsoft Azure, Red Hat OpenShift, and VMware vSphere.

For more information, see the [Dell APEX Cloud Platforms webpage](#).

### Dell APEX Cloud Platform for Red Hat OpenShift

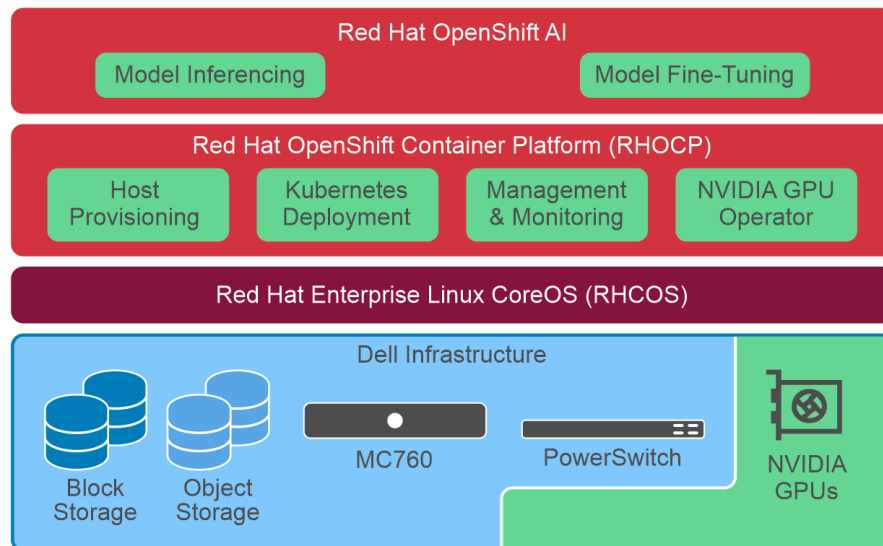
Dell APEX Cloud Platform for Red Hat OpenShift is designed collaboratively with Red Hat to optimize and extend OpenShift deployments on-premises with a seamless operational experience.



**Figure 1. Dell APEX Cloud Platform for Red Hat OpenShift high-level physical architecture**

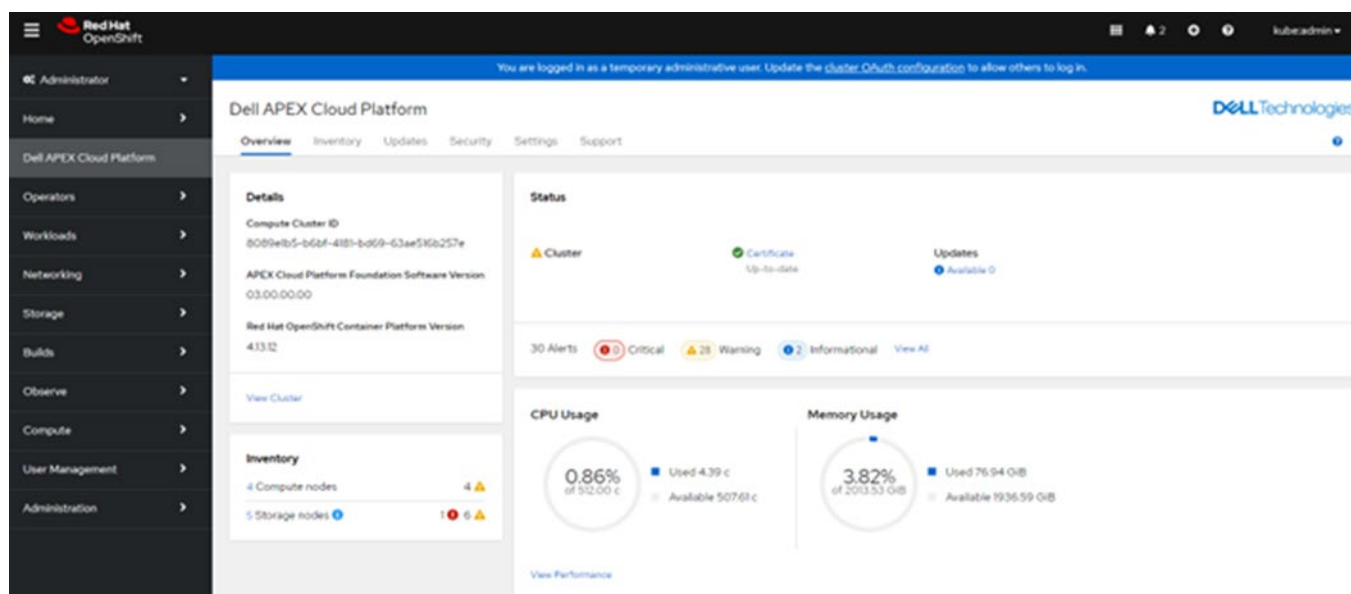
This turnkey platform provides:

- Deep integrations and intelligent automation between layers of Dell and OpenShift technology stacks, accelerating time-to-value and eliminating the complexity of management using different tools in disparate portals.
- Simplified, integrated management using the OpenShift Web Console.
- A bare metal architecture delivers the performance, predictability, and linear scalability needed to meet even the most stringent SLAs.



**Figure 2. High-level overview of OpenShift AI on Dell APEX Cloud Platform for Red Hat OpenShift logical architecture**

The Dell APEX Cloud Platform for Red Hat OpenShift introduces a new level of integration for running OpenShift on bare metal servers. The Dell APEX Cloud Platform Foundation Software mitigates this complexity by integrating the infrastructure management into the OpenShift Web Console. This integration enables administrators to update the hardware using the same workflow that updates the OpenShift software. It also enables OpenShift administrators to manage the infrastructure using the same management tools they use to control the cluster and the applications that run on it.



**Figure 3. Dell APEX Cloud Platform Foundation Software integration in OpenShift Web Console**

**Note:** The Federal Information Processing Standards (FIPS) is a ruleset that outlines methods for how data is handled and processed by encryption algorithms on endpoints and across various communication channels. FIPS is enabled by default in Dell APEX Cloud Platform for Red Hat OpenShift to provide a secure environment. This default setting is important to consider while developing your digital assistant applications, as some of the Python library may not be FIPS-compliant.

## Dell PowerFlex

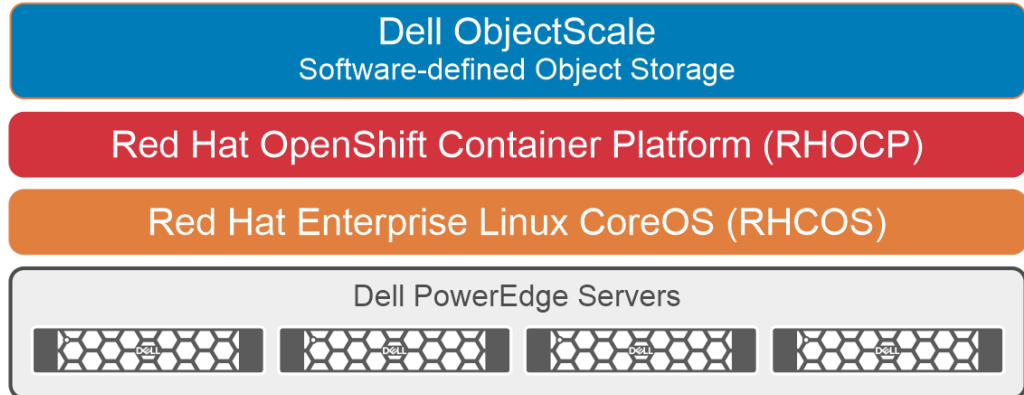
Dell PowerFlex is a dynamic and adaptable software-defined infrastructure meticulously crafted to modernize IT landscapes, increase business agility, and expertly handle modern workloads' intricacies. Its unparalleled performance and extensive scalability make PowerFlex a prime choice for consolidating diverse workloads, perfectly suited for demanding operational scenarios.

A noteworthy aspect of PowerFlex is its extreme performance. With its robust scale-out architecture, resource optimization capabilities, and low latency offerings, enterprises can meet stringent operational demands and ensure consistent performance benefits.

For more information, see the [Dell PowerFlex webpage](#).

## Dell ObjectScale

Dell ObjectScale provides high-performance containerized object storage built for demanding applications and workloads including Generative AI, analytics, and more. It is a software-defined, containerized object storage that delivers enterprise-class and high-performance in a Kubernetes-native environment. It empowers organizations to put data closer to the applications they support, reduce latency, and improve the user experience.



**Figure 4. Dell ObjectScale Storage Cluster**

Dell ObjectScale uses various Erasure Coding (EC) schemes for data protection. EC is a method of data protection in which data is broken into fragments, expanded, and then encoded with redundant data pieces. The data pieces are stored across different locations or storage media. The objective of EC is to enable data that becomes corrupted at some point in the disk storage process to be reconstructed. The data reconstruction process uses information about the data that is stored elsewhere in ObjectScale instance.

### Dell PowerSwitch

Dell Technologies stands at the forefront of GenAI networking innovation, offering solutions that meet the requirements of GenAI environments today and tomorrow, from the edge, core, and cloud. By focusing on open and extensible solutions.

Dell PowerSwitch is a dense, high-capacity, reliable spine and leaf and top-of-rack switches to meet the demands of modern AI fabrics and data center networks. The Dell PowerSwitch delivers performance from 10 GbE to 800 GbE of nonblocking network performance critical for GenAI applications. This allows customers to deploy AI clusters with low latency and high throughput using high bandwidth switching and new features like Advanced Routing, RoCEv2, Enhanced Hashing, and Priority Flow Control, for enhanced fabric performance and better congestion monitoring.

### Red Hat OpenShift

Red Hat OpenShift Container Platform is a consistent hybrid cloud foundation for containerized applications, powered by Kubernetes. Developers and DevOps engineers using Red Hat OpenShift Container Platform can quickly build, modernize, deploy, run, and manage applications anywhere, securely, and at scale.

Open-source technologies power Red Hat OpenShift Container Platform and offer flexible deployment options ranging from physical, virtual, private cloud, public cloud, and Edge. Red Hat OpenShift Container Platform cluster consists of one or more control-plane nodes and a set of worker nodes.

For more information about Red Hat OpenShift Container Platform, see the [Red Hat OpenShift web page](#).

## Red Hat OpenShift AI

Deploying AI applications can be complex due to a lack of integration among rapidly evolving tools. Popular cloud platforms offer attractive tools and scalability but often lock users in, limiting architectural and deployment options.

Red Hat OpenShift AI, formerly referenced as Red Hat OpenShift Data Science (RHODS), is a platform that unlocks the power of AI for developers, data engineers, and data scientists in Red Hat OpenShift. It is easily installed through a Kubernetes operator and provides a fully focused development environment called workbench that automatically manages the storage and integrates different tools. This provides users the ability to rapidly develop, train, test, and deploy machine learning models on-premises or in the public cloud environment.

Red Hat OpenShift AI allows data scientists and developers to focus on their data modeling and application development without waiting for infrastructure provisioning. The ML models developed using Red Hat OpenShift AI are portable to deploy in Production, on containers, on-premises, at the edge, or in the public cloud.

Red Hat OpenShift Pipelines is a cloud-native, continuous integration and continuous delivery (CI/CD) solution based on Kubernetes resources. It uses Tekton building blocks to automate deployments across multiple platforms by abstracting away the underlying implementation details. The data science pipeline is part of OpenShift AI that helps enhance data science projects by portable ML workflows. It enables standardization and automation of machine learning workflows to enable teams to develop and deploy data science models.

For more information about Red Hat OpenShift AI, see the [Red Hat OpenShift AI web page](#).

## Llama 2

Llama 2 is an open-access pre-trained LLM, from Meta that is freely available for research and commercial use. Llama 2 was trained on 40 percent more data than its predecessor, Llama 1, and has twice the context length (4096 compared to 2048 tokens), hence Llama 2 can understand and interpret the larger context better and provide users with more relevant and accurate information. Llama 2 can be used in use cases to build digital assistant for consumers and enterprise usage, language translation, research, code generation, and various AI-powered tools.

Llama-2-13b is used in this solution. It is more powerful and provides better responses to user queries than Llama-2-7b because of the increased number of weights, which are the building blocks of a language model's intelligence.

For more information about Llama 2, see the [Meta web page](#).

## LangChain

LangChain is an open-source framework for developing LLM-powered applications. It simplifies the process of building LLM powered applications by providing an abstracted standard interface that makes it easier to interact with different language models, including Llama 2.

LangChain's plug-and-play features allows users to use different data sources, LLMs, and UI tools without having to rewrite code and build powerful NLP applications with minimal effort.

LangChain provides various tools and APIs to connect language models to other data sources, interact with their environment, and build complex applications. Developers are required to use language models such as Llama 2 to build applications using LangChain. LangChain can be used to build digital assistants to generate a question-answering system over domain-specific information.

For more information about LangChain, see the [LangChain web page](#).

### Embedding model

Embedding models are the implementation of neural networks that are trained to represent words, phrases, or sentences as dense vectors in a high-dimensional space. all-mpnet-base-v2 is one such sentence-transforming model used in our solution for embedding the knowledge base into Vector store. Given an input text, in our case knowledge base document chunks, it outputs vectors which capture the semantic information, that will be later used to perform similarity search against user query.

For more information about sentence transformers, see the [Hugging Face sentence transformers web page](#).

### Vector store

Vector stores are databases designed to store and retrieve vector embeddings efficiently and to perform semantic searches.

**PGVector:** PGVector is a PostgreSQL extension that provides powerful functionalities for working with vectors in high-dimensional space. It introduces a dedicated datatype, operators, and functions that enable efficient storage, manipulation, and analysis of vector data directly within the PostgreSQL database.

For more information about the PGVector vector database, see the [PGVector webpage](#).

**Redis:** Redis is a popular in-memory data structure store. The Redis database features the ability to store embeddings with metadata for later use by LLMs. Redis vector database is an excellent choice for applications that have to store and search vector data quickly and efficiently. It offers an effective solution to efficiently query and retrieve relevant information from massive amounts of data.

For more information about the Redis vector database, see the [Redis webpage](#).

### Gradio

Gradio is an open-source Python library that enables incredibly fast development/prototyping of the ML web applications with user interfaces. It provides a simple and intuitive API which is compatible with all Python programs and libraries. Gradio provides various options to customize various elements of the user interface (UI).

Gradio is a fastest way to prototype any ML model with a friendly web interface. It has a unique capability to visualize the intermediate steps or thought processes during a language model's decision-making process. This unique feature makes Gradio a useful tool for analyzing and debugging the decision processing abilities for language models.

For more information about Gradio, see the [Gradio webpage](#).



**vLLM**

LLM models can be surprisingly slow, even on expensive hardware. vLLM serving runtime is a fast and easy-to-use LLM-inference engine that can help overcome this challenge. It uses high-throughput LLM-serving architecture with efficient memory management, enabled by PagedAttention, and it natively supports Llama 2 model, which also does not require weights to be converted to a specific format. PagedAttention is a new attention algorithm that allows attention keys and values to be stored in non-contiguous paged memory.

For more information about vLLM, see the [vLLM webpage](#).

### 3. Solution Architecture and Requirements

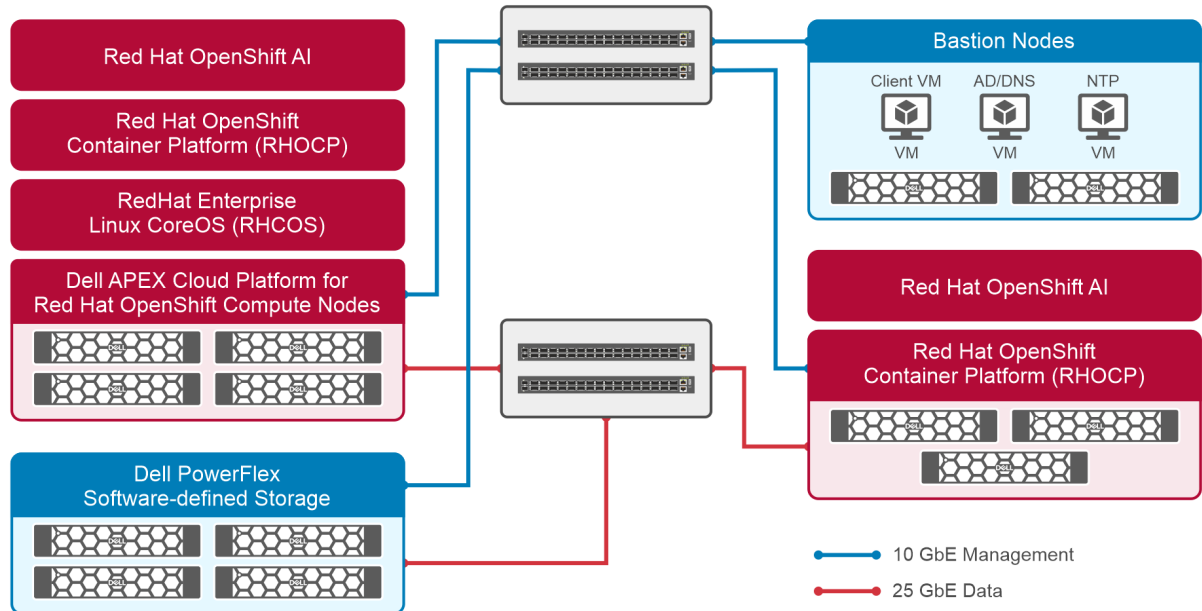
This chapter presents the following topics:

- Physical architecture .....19
- Logical architecture .....21
- Digital assistant design .....22
- Solution requirements .....23
- Digital Assistant design.....20
- Solution requirements .....21

## Physical architecture

### Overview

The following figure demonstrates the physical architecture design of the solution used to run the LLM-based application on Dell APEX Cloud Platform for Red Hat OpenShift.



**Figure 5. Dell APEX Cloud Platform for Red Hat OpenShift physical architecture**

The following sections describe the various hardware stacks involved in designing this solution.

### Dell servers

Dell APEX Cloud Platform for Red Hat OpenShift compute layer setup is configured with four MC 760 nodes for running AI workloads. [MC 760 server](#) is a 2U, two-socket fully featured enterprise rack server, designed to optimize even the most demanding workloads like artificial intelligence and machine learning. It provides a broad choice of core densities using Intel's 4th gen Xeon scalable processors.

MC 760 Servers offer:

- Up to two fourth Generation Intel Xeon Scalable processors with up to 56 cores for faster and more accurate processing performance.
- Accelerate in-memory workloads with up to 32 DDR5 RDIMMS up to 4400 MT/sec (2DPC) or 4800 MT/sec for 1DPC (16 DDR5 RDIMMs max).
- Support for GPUs including 2 x double-wide or 6 x single-wide for workloads requiring acceleration.

For more information about Dell MC 760 servers, see the [Dell APEX Cloud Platform MC 760 Hardware Requirements and Specifications page](#). This solution includes MC 760 servers with Intel(R) Xeon(R) Gold 6430 processors which offer 32 cores per socket, operating at a speed of 2.10 GHz. Additionally, servers are equipped with 2 X NVIDIA L40S GPU. The NVIDIA L40S GPU is a powerful data center GPU, which comes with 48 GB GDDR6 memory and supports PCIe Gen4 x16: 64 GB bi-directional interconnect

interface, delivering end-to-end acceleration for the next generation of AI workloads such as generative AI, LLM inferencing and training.

#### Storage

**Block Storage:** Dell PowerFlex is a software-defined infrastructure that delivers consistent predictable outcomes at large scale for the most demanding mission-critical environments. In our setup, four dedicated PowerFlex nodes are configured as storage nodes to provide persistent block storage for workloads, such as Vector store, OpenShift AI workbench and data science pipelines. The [Dell Container Storage Module \(CSM\)](#) enables simple and consistent integration and automation experiences, extending enterprise storage capabilities to Kubernetes for cloud-native state applications.

**Object Storage:** Dell ObjectScale is a high-performance containerized object storage that is designed for scalable and cost-effective data management and provides an ideal solution for housing massive LLMs, large datasets, and pipeline artifacts. As shown in the solution architecture, Dell ObjectScale is deployed separately which stores the Llama 2 model, other datasets, and Red Hat OpenShift AI pipeline artifacts.

#### Networking

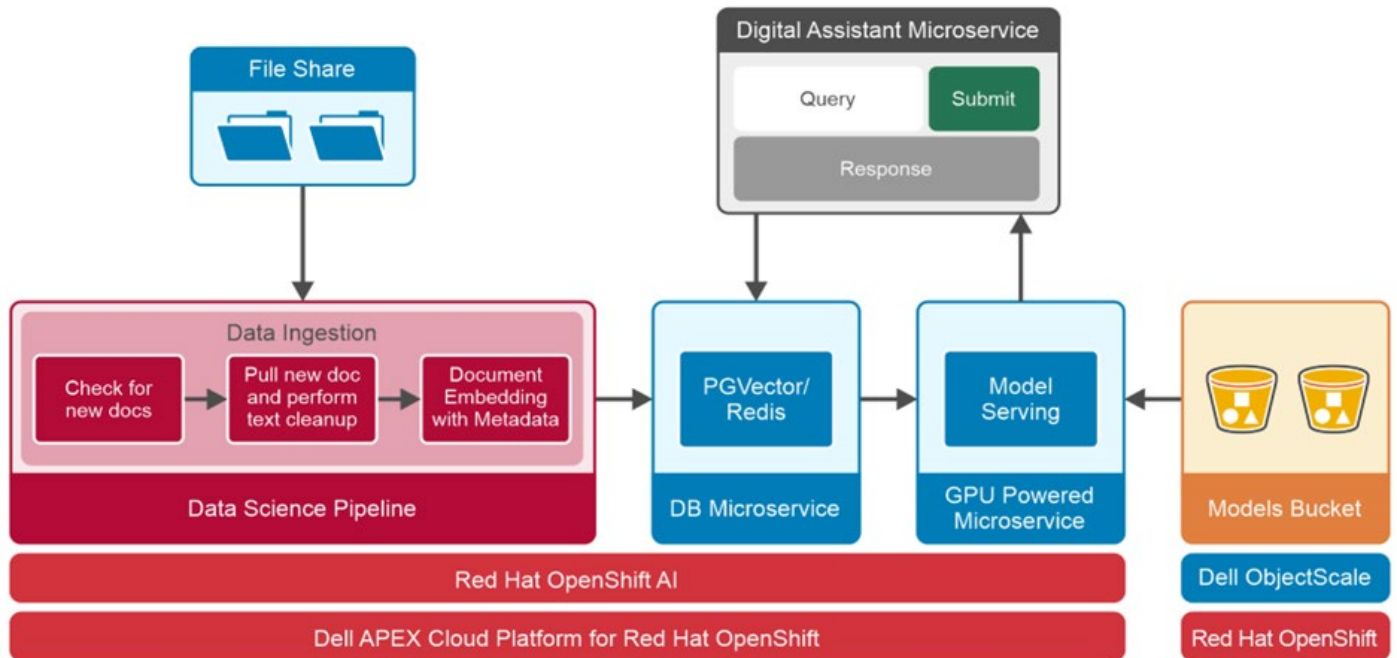
Two [25 GbE Dell PowerSwitch](#) network switches are being used for data plane connectivity, and two 10 GbE Dell PowerSwitch network switches are being used for management plane connectivity. Dell PowerSwitch enhances performance in data center fabrics of all sizes with efficient and flexible switching solutions.

#### Bastion node

These nodes are used to run VMs, and containers related to client application and tools required to manage the Red Hat OpenShift cluster, Object Storage, and LLM-based application. Also, infra VMs such as AD and DNS are running in these nodes.

## Logical architecture

The following figure demonstrates the logical architecture design of the digital assistant deployed on Dell APEX Cloud Platform for Red Hat OpenShift:



**Figure 6. Digital assistant logical architecture**

In our validated design, we use Llama 2 model for language processing, LangChain to integrate different tools of the LLM-based application together and to process the PDF files and web pages, Redis or PGVector to store vectors, KServe and vLLM to serve the Llama 2 model, Gradio for user interface and Dell ObjectScale object storage to store language model and other datasets. Solution components are deployed as microservices in the Red Hat OpenShift cluster.

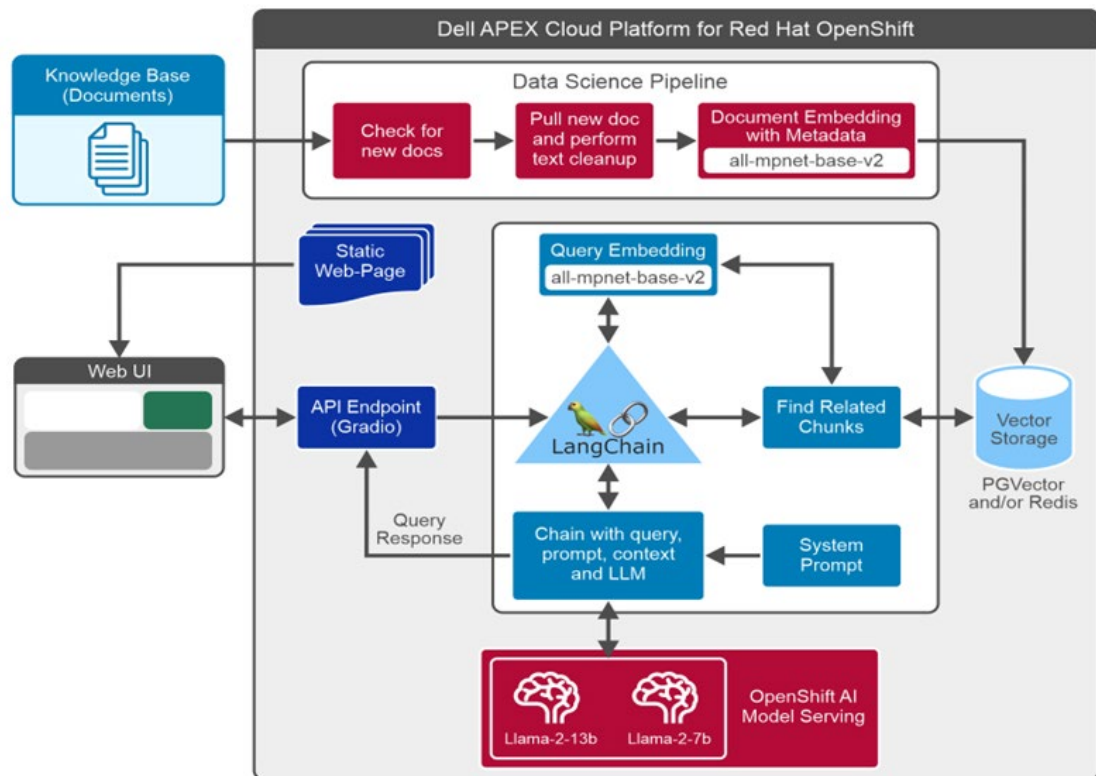
The following list details the roles and responsibilities of each microservice.

- **Digital assistant:** This microservice runs LangChain, which integrates different components of the LLM-based application together. It also provides the user interface, based on the Gradio UI framework, to interact with the digital assistant.
- **Data science pipeline:** A data science pipeline is scheduled to run periodically and leveraged to build a data ingestion workflow. Each task within the workflow runs as a separate microservice. This workflow is used to detect recently added or modified documents from file share, extract texts, perform cleanup of data, followed by creating and loading embeddings into PGVector database.
- **Vector database:** The digital assistant is compatible with both PGVector and Redis as a vector store. In this solution, both PGVector and Redis are deployed as a microservice. Vector store is used to store knowledge base embeddings, and to perform similarity search during context retrieval.

- **Model serving:** Red Hat OpenShift AI includes a single model serving platform that is based on KServe to serve LLMs. KServe provides a Kubernetes Custom Resource Definition for serving predictive and generative machine learning (ML) models. vLLM GPU powered microservice is used as the serving runtime in our solution to serve the Llama 2 model. The Llama 2 model weights and configuration files are copied from the Hugging Face repository and stored in Dell ObjectScale. Storing the models in object storage provides the capability of model versioning and eliminates the need of maintaining multiple copies locally.
- **File share:** Apache HTTP is deployed as a microservice to host the internal documents, which will be ingested into the vector store through data ingestion pipeline.

## Digital assistant design

RAG is an AI framework that allows LLMs to access additional domain-specific data and generate better and more accurate answers, without having to be retrained. The following figure describes the RAG based digital assistant architecture design.



**Figure 7. Digital assistant design**

The following list outlines digital assistant components and workflow.

- **Data Ingestion:** The data science pipeline is leveraged to ingest data, which is created and scheduled to run at specific intervals. This pipeline discovers any new files, extracts text, performs data cleanup, stores metadata, create manageable text

chunks, create embeddings for the text chunks and store text, metadata, and their embeddings into the PGVector or Redis vector database collection.

- **User Interface:** Gradio provides a simple and intuitive user interface to interact with the digital assistant. LangChain integrates the Gradio user interface with other components such as LLM and vector store to work together as a digital assistant.

This digital assistant solution is provided with advanced UI features which include choice of different LLMs, vector stores, and catalogs. Digital Assistant UI also provides additional options to users such as maximum sequence length, temperature control, retrieval score threshold, and maximum number of documents retrieved. These additional options allow users to have additional controls on the response from the digital assistant.

- **Query processing:** When users submit a query using the UI, the query will be first converted to vector embedding. Embedding is a process of converting text chunks to a fixed-sized vector. Semantic search is then performed for the query embedding against the knowledge base vector embeddings stored in the PGVector database. Results from PGVector are ranked and sent to the Llama 2 model along with predefined system prompts, the Llama model generates the response based on retrieved context from PGVector and its pretrained capabilities.

## Solution requirements

### Server details

| Components         | Compute Nodes                              | PowerFlex Storage Nodes               |
|--------------------|--|---------------------------------------|
| Nodes              | 4 x MC 760 servers                         | 4 x PowerFlex Rack Server             |
| CPU                | 2 X Intel(R) Xeon(R) Gold 6430             | 2 X Intel(R) Xeon(R) Gold 6430        |
| GPU                | 2 X NVIDIA L40S                            | N/A                                   |
| Memory             | 32 X 64 GB DDR-5 @ 4400 MT/s               | 16 X 64 GB DDR-5 @ 4400 MT/s          |
| Storage Controller | BOSS-N1                                    | BOSS-N1                               |
|                    |  | Dell HBA355i                          |
| Drives             | 2 X 894 GB NVMe SSDs                       | 2 X 894 GB NVMe SSDs                  |
|                    |  | 24 X 745 GB SSDs                      |
| NICs               | 2 X NVIDIA ConnectX-6 Lx 2x 25G, dual port | 2 X NVIDIA ConnectX-6 Lx 2x 25G SFP28 |
|                    | 2 X ConnectX-6 Lx 2x 25G SFP28             | 2 X NVIDIA ConnectX-6 Lx 2x 25G       |
| PSU                | 2400 W Redundant Power Supply              | 2400 W Redundant Power Supply         |

**Server firmware details**

| Firmware/Software   | Version        |
|---------------------|----------------|
| Server BIOS version | 2.0.0          |
| iDRAC               | 7.10.05.00     |
| GPU                 | 95.02.66.00.02 |
| System CPLD         | 1.0.7          |
| Dell OS Driver Pack | 23.08.08       |
| Identity Module     | 1.00           |
| Power Supply        | 00.23.54       |
| TPM                 | 7.2.2.0        |
| HBA355i             | 24.15.14.00    |
| BOSS-N1             | 2.1.13.2025    |
| NVIDIA ConnectX-6   | 26.38.10.02    |

**Software details**

| Software                                       | Version                                  |
|--|--|
| Server OS                                      | RHCOS 4.13                               |
| Red Hat OpenShift Container Platform           | 4.13.26                                  |
| Kubernetes                                     | 1.26.11                                  |
| APEX Cloud Platform Foundation Software        | 03.00.02.00                              |
| Dell ObjectScale                               | 1.3.0                                    |
| Red Hat OpenShift AI                           | 2.9.1                                    |
| Red Hat OpenShift serverless                   | 1.32.1                                   |
| Red Hat OpenShift service mesh                 | 2.5.1-0                                  |
| Red Hat OpenShift distributed tracing platform | 1.53.0-4                                 |
| Red Hat OpenShift Pipelines                    | 1.14.4                                   |
| NVIDIA GPU operator                            | 23.9.2                                   |
| Node Feature Discovery operator                | 4.13.0                                   |
| Redis enterprise operator                      | 7.4.2-12.0                               |
| Kiali operator                                 | 1.73.7                                   |
| Python   | 3.11                                     |
| LangChain                                      | 0.1.14                                   |
| LLM  | Llama-2-7b-Chat-hf & Llama-2-13b-Chat-hf |
| Gradio   | 4.24.0                                   |



## 4. Solution Deployment

This chapter presents the following topics:

|  |           |
|--|-----------|
| <b>Introduction .....</b>  | <b>26</b> |
| <b>Dell APEX Cloud Platform for Red Hat OpenShift configuration.....</b> | <b>26</b> |
| <b>Pre-requirements .....</b>  | <b>26</b> |
| <b>Dell ObjectScale.....</b>   | <b>27</b> |
| <b>Digital Assistant Solution components deployment .....</b>            | <b>27</b> |
| <b>Deployment summary.....</b>   | <b>35</b> |

## Introduction

Deploying an LLM-based application is a complex multi-step process that requires the right combination of hardware and software for a robust AI platform. Dell APEX Cloud Platform for Red Hat OpenShift reduces this complexity by providing an AI solution for data scientists and data engineers to deploy an LLM model.

We deployed a RAG based digital assistant using Dell APEX Cloud Platform for Red Hat OpenShift with Red Hat OpenShift AI. The goal of this solution is to enable users to ask queries related to domain-specific data and get accurate answers to the questions within the scope of the digital assistant. This chapter describes the necessary steps to deploy and configure the overall solution.

## Dell APEX Cloud Platform for Red Hat OpenShift configuration

### Overview

Dell APEX Cloud Platform for Red Hat OpenShift is designed collaboratively with Red Hat to optimize and extend OpenShift deployments on-premises with an integrated operational experience. Dell APEX Cloud Platform for Red Hat OpenShift is based on MC 760 servers, Red Hat Enterprise Linux CoreOS (RHCOS) and Red Hat OpenShift Container Platform.

### Preparation

The following list describes the high-level tasks required to set up Dell APEX Cloud Platform for Red Hat OpenShift:

- Rack and stack the servers as per the guidelines provided with server shipment.
- Bootstrap OS is pre-installed on MC 760 nodes before shipping. Coordinate with the Pro-Deploy team to deploy and configure Dell APEX Cloud Platform for Red Hat OpenShift.
- Log in and verify accessibility to the Red Hat OpenShift Container Platform dashboard.
- Set up the user accounts that are required to perform the solution deployment.

## Pre-requirements

**PowerFlex setup** Dell PowerFlex is a comprehensive and adaptable software-defined infrastructure, which is used in our solution for block storage requirements of Dell APEX Cloud Platform for Red Hat OpenShift workloads.

Reach out to Dell support team for Dell APEX Cloud Platform for Red Hat OpenShift deployment tool, which will also be used for deploying Dell PowerFlex storage cluster.

## Storage Class

Storage Class for Dell PowerFlex storage will be configured during the deployment and configuration of Dell APEX Cloud Platform for Red Hat OpenShift. After deployment, we need to set the PowerFlex storage as a default storage class. Perform the below command using OpenShift CLI to set the default storage class:

```
oc patch sc <PowerFlex Storage Class name> -p '{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-class": "true"}}}'
```

## Container Registry

Red Hat OpenShift Container Platform provides an internal, integrated container image registry that can be deployed to locally manage container images in the Red Hat OpenShift Container Platform cluster environment.

Follow the steps described in the [OpenShift Container Platform documentation](#) to configure the container registry.

## Dell ObjectScale

### Overview

Dell ObjectScale object storage is used in this solution to store and access Llama 2 models, datasets, and artifacts. ObjectScale empowers organizations to move faster and respond more effectively to rapidly changing business needs.

### Deployment

A detailed step-by-step installation procedure for Dell ObjectScale deployment on the Red Hat OpenShift environment is available on [Dell ObjectScale 1.3.x Installation Guide for Red Hat OpenShift](#).

## Digital Assistant Solution components deployment

The following are high-level steps involved in deploying the digital assistant solution components:

1. Install the pre-requisites.
2. Deploy a vector database to store document vectors.
3. Deploy a file server to host knowledge base documents.
4. Create and schedule a data science pipeline for data ingestion.
5. Prepare a model serving environment and deploy the model.
6. Deploy the digital assistant application.

### Pre-requisites

Below are the pre-requisites for deploying solution components:

---

**Note:** Some of the pre-requisite operators for OpenShift AI are pre-installed on Dell APEX Cloud Platform for Red Hat OpenShift as part of cluster deployment, which includes OpenShift distributed tracing platform operator, Kiali operator and Red Hat OpenShift Service mesh operator.

---

- [Install Node Feature Discovery operator](#) from Operator Hub
- [Install NVIDIA GPU operator](#) from Operator Hub
- [Install Red Hat OpenShift Serverless operator](#) from Operator Hub
- [Install Red Hat OpenShift Pipelines operator](#) from Operator Hub
- [Install Red Hat OpenShift AI operator](#) from Operator Hub
- [Install OpenShift AI components and validate.](#)
- Clone [digital assistant GitHub repository](#) using OpenShift CLI

```
git clone https://github.com/S-Ranjan/dell-digital-assistant.git
```

### Vector Store

#### PGVector:

The [Container file](#) builds a PostgreSQL 15 + PGVector image (PGVector is built from source). You can then deploy this container as any other PostgreSQL image.

A prebuilt image is available at <https://quay.io/repository/dellbizapps/ai/postgresql-15-PGVector-c9s>.

Go to “dell-digital-assistant/02-vectorstore/PGVector/” from the local clone.

```
cd dell-digital-assistant/02-vectorstore/PGVector/
```

---

**Note:** Change database name, user and password details in 01-db-secret.yaml.

---

Apply the files using `OC apply` to deploy PostgreSQL+PGVector server. Once deployed PGVector will be accessible at `postgresql.<your-project>.svc.cluster.local:5432`.

The PGVector extension must be manually enabled in the server. This can only be done as a Superuser. Follow the below steps to enable PGVector extension.

- Connect to the running server Pod, either through the terminal view in the OpenShift Console, or through the CLI with:

```
oc rsh services/postgresql
```

- Once connected, enter the following command. Replace vectordb with your database name:

```
psql -d vectordb -c "CREATE EXTENSION vector;"
```

If the command succeeds, it will print `CREATE EXTENSION`.

Your PGVector database is now ready to use.

#### Redis:

From the Operator Hub, [install the Redis Enterprise Operator](#).

You can install the operator with the default value in the namespace you want to create your Redis cluster. To create a Project in the OpenShift cluster, run the following:

```
oc new-project redisdb
```

Go to “dell-digital-assistant/02-vectorstore/redis/” from the local clone.

```
cd dell-digital-assistant/02-vectorstore/redis/
```

Create Security Context Constraints using 01-redis-scc.yaml:

```
oc apply -f 01-redis-scc.yaml
```

Provide the operator permissions for Redis Enterprise Operator and Cluster pods:

```
oc adm policy add-scc-to-user redis-enterprise-scc-v2
system:serviceaccount:redisdb:redis-enterprise-operator
```

```
oc adm policy add-scc-to-user redis-enterprise-scc-v2 system:serviceaccount:redisdb:redis-enterprise-operator
```

Create a Redis cluster using 02-redis-cluster.yaml:

```
oc apply -f 02-redis-cluster.yaml
```

Once you can deploy a database to host the vector store. The important parts in our scenario are to enable the search module and set enough memory to hold the initial index capacity. Here is an example:

Create the secret for the Redis database using following 03-db-secret.yaml definition as an example. Update the username and password before applying this manifest:

```
oc apply -f 03-db-secret.yaml
```

Create the redis database using following 04-redis-db.yaml definition as an example. Change the search module version as per your deployment.

```
oc apply -f 04-redis-db.yaml
```

Once the database is deployed, you will have:

- A secret named redb-my-doc (or the name you mentioned in the YAML). It holds the password to the default user account for this database.
- A service named my-doc-headless (or the name you put in the YAML). From this service, you will get: 1. the full URL to the service from within the cluster, such as my-doc-headless.<namespace>.svc.cluster.local, 2. the port that Redis is listening to, such as 14155.

With the above information, when asked for your Redis URL in the different notebooks or applications on this repo, the full URI you can construct will be in the form:  
redis://default:password@server:port.

## File Server

Below are the steps involved in deploying the file server:

- Run the below command to create a new project for the file server:

```
oc new-project filesaver
```

## 4: Solution Deployment

- Go to "dell-digital-assistant/03-file-server" from local clone and run the following command to deploy the file server:

```
oc apply -f deployment/
```

- Validate if the pod is up and running fine

```
oc -n fileserver get pods
```

- Use the following command construct to copy/sync files and folder structure to the file server container.

```
oc rsync <source> <destination> [-c <container>]
```

- Use the below command to copy index.html to the file server container. While running the below command, change the pod name and container name as per your deployment:

```
oc rsync index.html httpd-frontend-86bfd7d9b-8sbmv:/var/www/html/ -c httpd-rhel7
```

- Sync PDF files to a file server:

```
oc rsync /home/user/pdf/demo httpd-frontend-86bfd7d9b-8sbmv:/var/www/html/pdf  
-c httpd-rhel7
```

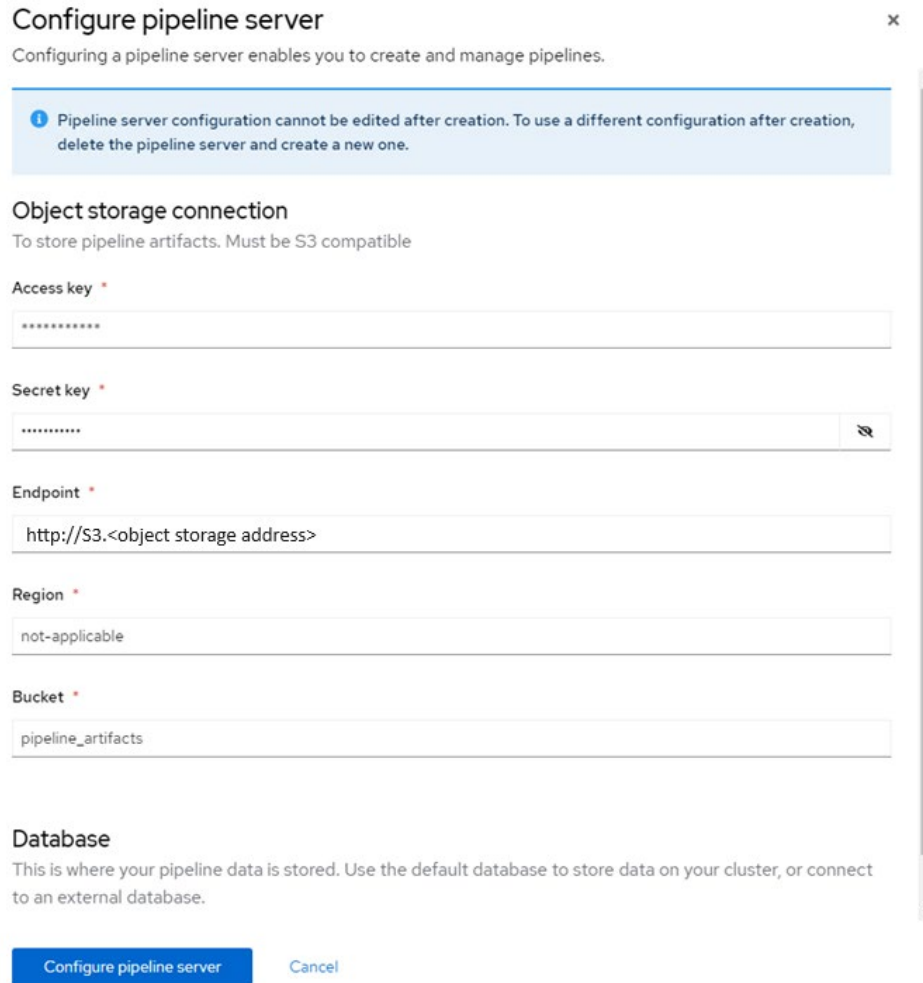
- Go to the route section under the file server namespace you can see the link to access the file server.

## Data science pipeline

A data science pipeline is leveraged to build a data ingestion workflow, which can be manually triggered or scheduled to run at specific time intervals.

Pipeline server is a server that is connected to your data science project which hosts your data science pipeline artifacts. The artifacts are stored in S3 compatible object storage, in our case it is Dell ObjectScale.

To configure a Pipeline server, go to OpenShift AI -> Pipelines -> Configure pipeline server -> configure pipeline server details.



**Configure pipeline server**

Configuring a pipeline server enables you to create and manage pipelines.

**Object storage connection**  
To store pipeline artifacts. Must be S3 compatible

Access key \*

Secret key \*

Endpoint \*

Region \*

Bucket \*

**Database**  
This is where your pipeline data is stored. Use the default database to store data on your cluster, or connect to an external database.

Configure pipeline server Cancel

**Figure 8. Pipeline server configuration.**

Below are the steps to create and trigger a Data science pipeline:

- Log in to your OpenShift AI workbench.
- Clone the Dell Digital assistant GitHub repository within the workbench using terminal.  
`git clone https://github.gtie.dell.com/S-Ranjan/dell-digital-assistant.git`
- Go to "dell-digital-assistant/04-datascience-pipeline/" using file browser.  
`cd dell-digital-assistant/04-datascience-pipeline/`
- You will find all the necessary notebook files to create a data ingest pipeline.

---

**Note:** Change database connection details according to your database configuration.

---

- Create a new Elyra file using new launcher -> Elyra pipeline editor -> drag and drop the notebook files and create a pipeline.

## 4: Solution Deployment

- Click each node, open properties, and choose the runtime image to use.

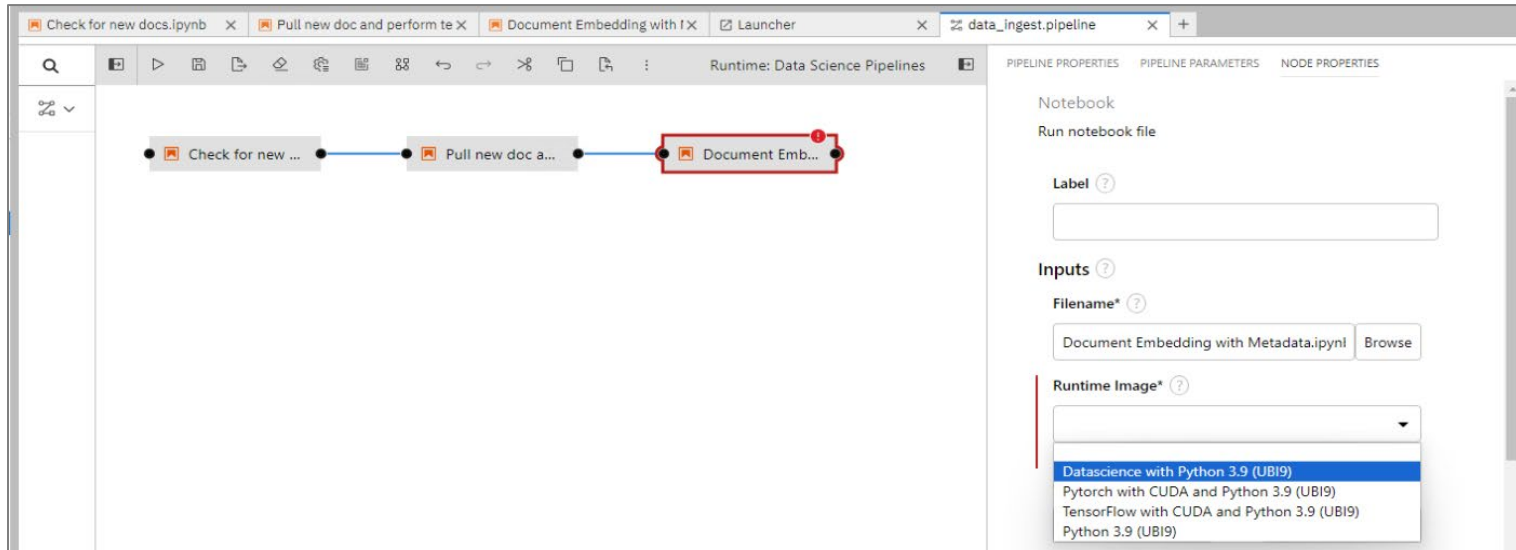


Figure 9. Elyra pipeline runtime image configuration.

- Each node in the pipeline saves information such as new files, cleaned up text files, and so on, in a persistent volume, which will be shared among the nodes in the pipeline.

Create a PVC within the project that the data science pipeline would be configured.

Go to Red Hat OpenShift Dashboard -> Storage -> PVC -> Create a new PVC.

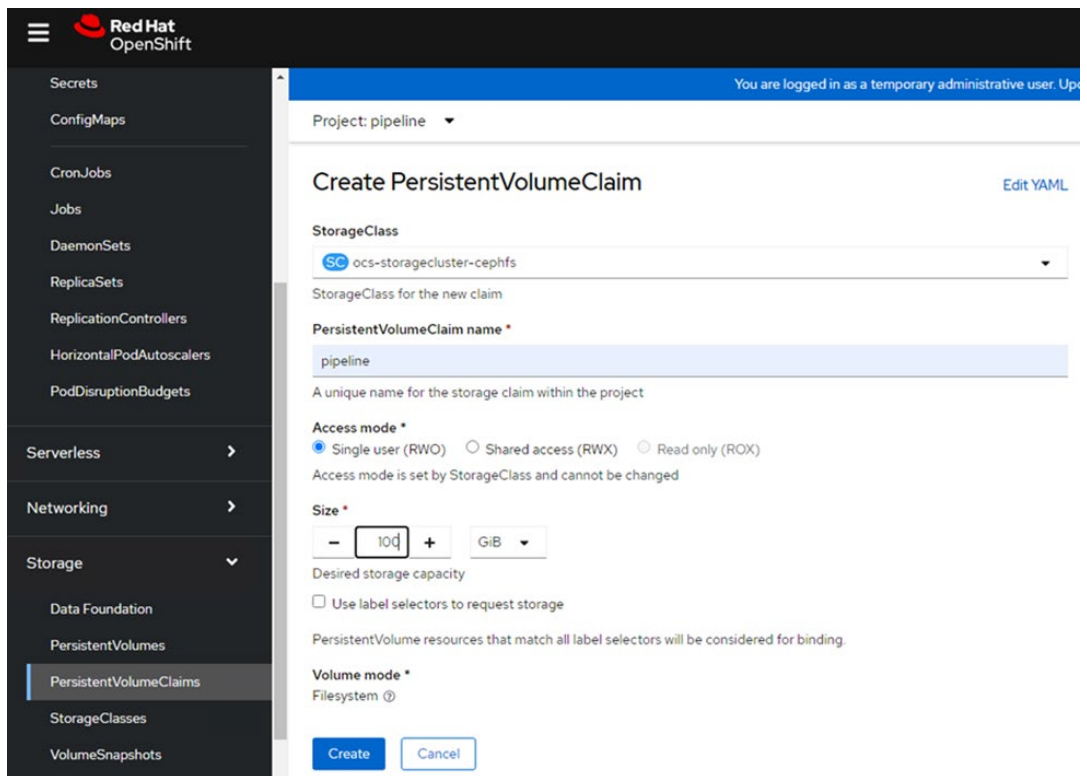


Figure 10. PVC creation for Data Science Pipeline



Once PVC has been created successfully, Mount the PVC to pipeline nodes by going to pipeline properties -> node defaults -> Volumes.

PIPELINE PROPERTIES PIPELINE PARAMETERS NODE PROPERTIES

project/subproject

### Node Defaults ?

#### Data Volumes ?

**Mount Path\***

/data

**Persistent Volume Claim Name\***

pipeline

**Sub Path**

relative/path/within/volume

☐ Mount volume read-only

Remove

Figure 11. Data Science Pipeline PVC volume mount.

- Save and run the pipeline, give this pipeline a name, and choose the runtime configuration as “Data Science Pipeline”.
- Go to OpenShift AI dashboard -> Data Science Pipelines -> Pipelines, verify if the pipeline we triggered in the previous step is listed here.
- Schedule the data ingest pipeline to run in the periodic interval of your choice.

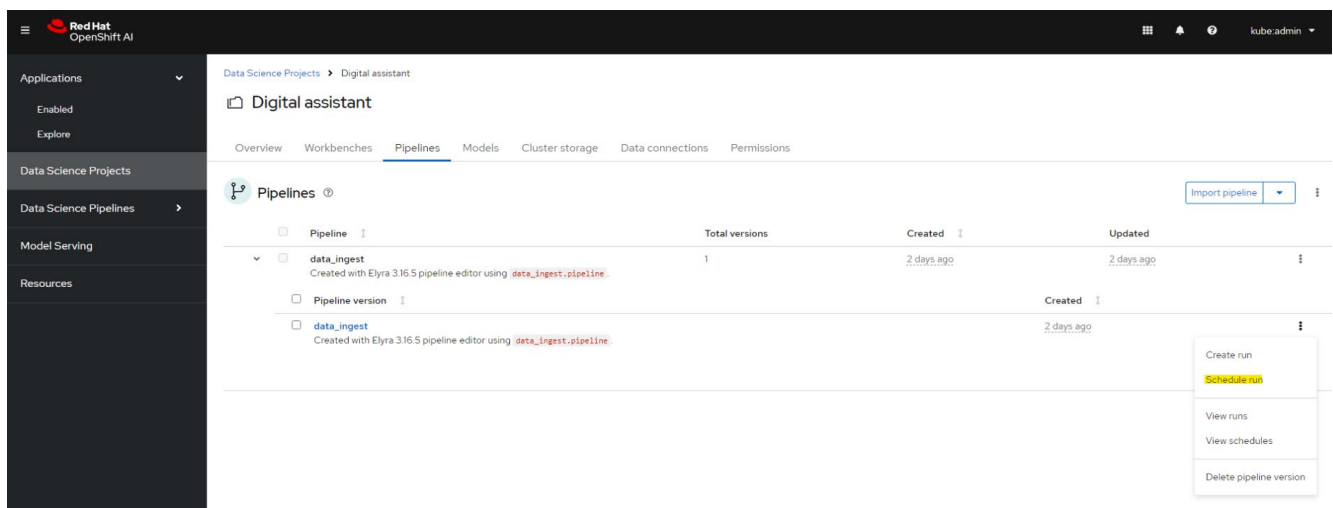


Figure 12. Data Science Pipeline scheduling

### vLLM Model Serving

For deploying large language models such as Llama 2, Red Hat OpenShift AI includes a single model serving platform that is based on the KServe component. KServe provides a Kubernetes Custom Resource Definition for serving predictive and generative machine learning (ML) models. Follow the below steps to deploy Llama 2 model on vLLM serving runtime leveraging the single model serving feature of Red Hat OpenShift AI.

**Installation:** First ensure that you have properly installed the necessary component of the Single-Model Serving stack, as documented [here](#).

Once the stack is installed, adding the runtime is straightforward:

- As an admin, in the OpenShift AI Dashboard, open the menu Settings -> Serving runtimes.
- Click Add serving runtime.
- For the type of model serving platforms this runtime supports, select Single model serving platform.
- Upload the file [vllm-runtime.yaml](#) from the current folder, or click Start from scratch and copy/paste its content.

---

**Note:** vllm-runtime.yaml can also be found in dell-digital-assistant/05-model-serving/ local clone folder.

---

The runtime will now be available when deploying a model.

**Model Deployment:** This runtime can be used in the exact same way as the pre-installed serving runtime in Red Hat OpenShift AI:

- Copy your model files in an object store bucket.
- Deploy the model from the dashboard.
- Make sure you have added a GPU to your GPU configuration, that you have enough VRAM (GPU memory) to load the model, and that you have enough standard memory (RAM). Although the model loads into the GPU, RAM is still used for the pre-loading operations.
- Once the model is loaded, you can access the inference endpoint provided through the dashboard.

**Usage:** This implementation of the runtime provides an OpenAI compatible API. So, any tool or library that can connect to OpenAI services can consume the endpoint.

Python and curl examples are provided [here](#).

Also, vLLM provides a full Swagger UI where you can get the full documentation of the API (methods, parameters), and try it directly without any coding. It is accessible at the address <https://your-endpoint-address/docs>.

---

**Note:** With Red Hat OpenShift AI version 2.10.0, vLLM serving runtime is available as pre-installed and fully supported by Red Hat.

---

**Digital Assistant** Follow the below procedure to deploy the digital assistant:

A pre-built container image of the application is available at:  
quay.io/dellbizapps/ai/dav2:v0.7

The [deployment folder](#) includes the necessary files to deploy the application.

Create a new project in the OpenShift cluster using the following command:

```
oc new-project dell-digital-assistant
```

Go to "dell-digital-assistant/06-digital-assistant/" from the local clone and run the following command:

```
oc apply -f deployment/
```

The "oc apply -f deployment/" command deploys the entire application configuration, including all necessary resources and settings, from the YAML files in the deployment/ directory.

## Deployment summary

### Overview

We have deployed a Vector database which hosts the knowledge base embeddings. Knowledge base pdf files are stored in a file server, and the Data science pipeline discovers new files based on the schedule and ingests the embeddings into the Vector database. LangChain integrates components such as Gradio UI framework, Llama 2 model which is served by vLLM serving runtime and Vector database.

All necessary components required for digital assistant to function are deployed and Digital assistant UI can be accessed by going to the route section under digital assistant namespace, where you will find the link.

# 5. Solution Validation

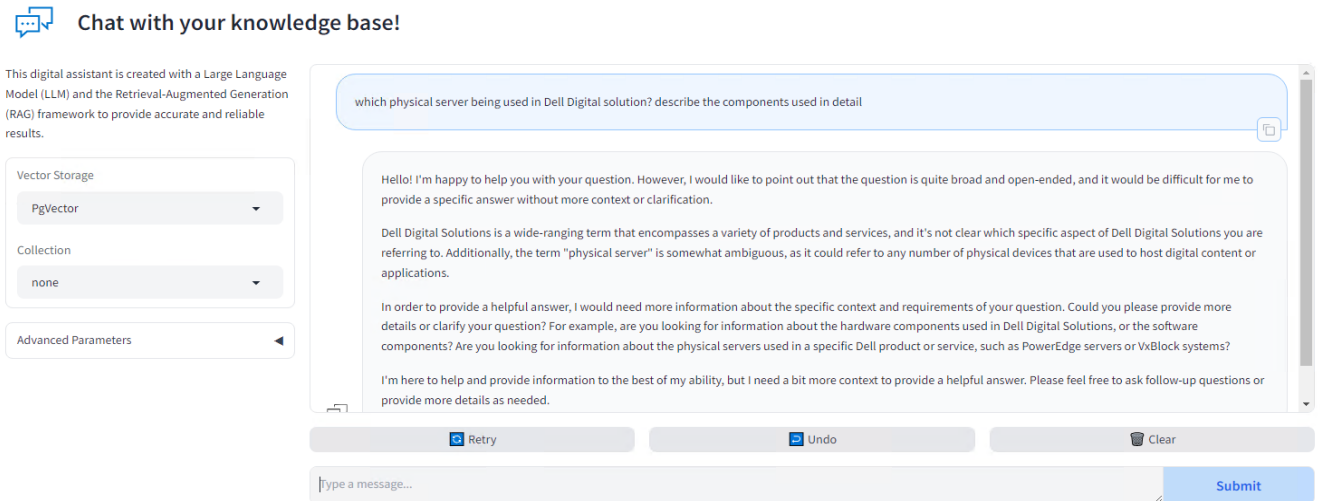
This chapter presents the following topics:

**Digital Assistant demonstration .....37**

**LLM load testing.....38**

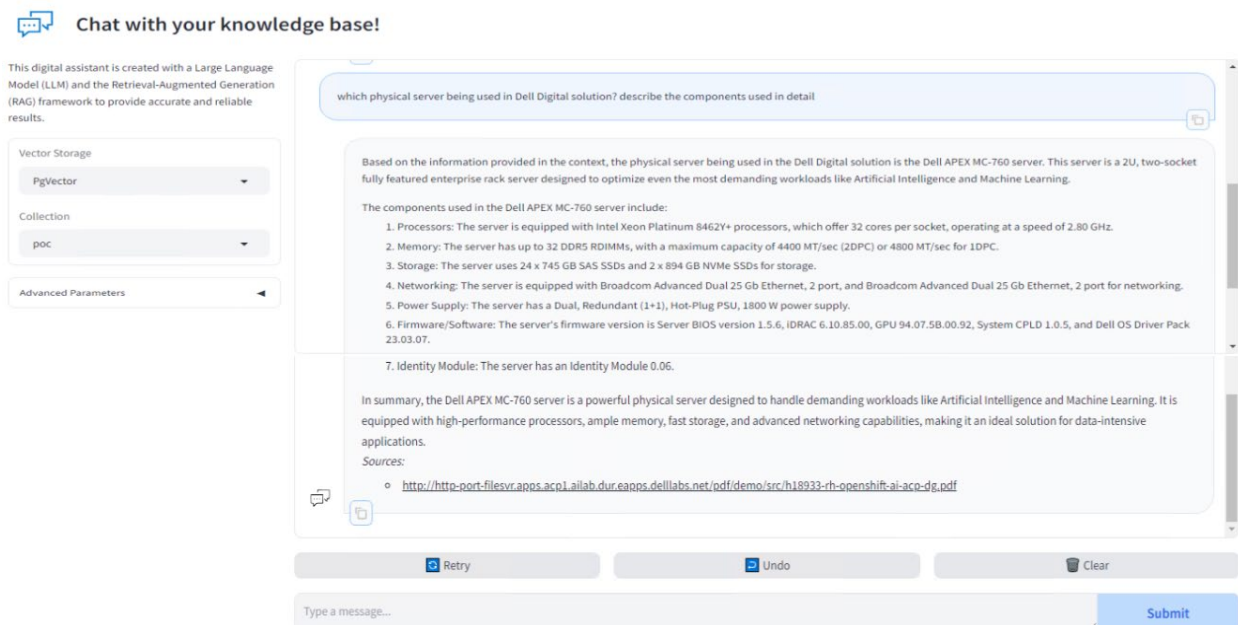
## Digital Assistant demonstration

The following figure shows an example of the user query without domain-specific context.



**Figure 13. Digital assistant response without context**

The following figure provides an example of the user query and its response from a domain-specific knowledge base, along with the source information.



**Figure 14. Digital assistant response with relevant context**

## Guardrail Demonstration

We have implemented a basic guardrail feature in the digital assistant to detect profanity in user queries. The following figure provides an example of the user query with a profane word and the digital assistant response.

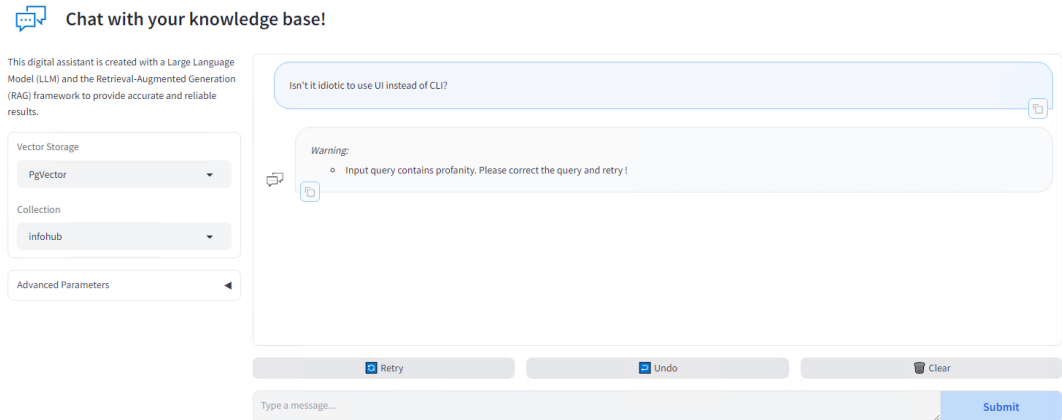


Figure 15. Profanity check demo

## LLM load testing

### Overview

To determine the optimal user load for a single instance of the Llama 2 model served with vLLM runtime, we developed a methodology to conduct user load tests and provide relevant performance metrics.

LLM load test is one such open-source tool from Red Hat, which is designed to perform load testing on LLMs running in different runtimes or behind different APIs. See [LLM load testing GitHub repository](#) for more information.

### Load test configuration

The following is the configuration of our instance used for validation.

Number of replicas: 1  
 CPU: 6 vCPU  
 Mem: 12 GB  
 GPU: 1 x NVIDIA L40s

Below are the parameters and its values from config.yaml (LLM load test) used during our validation with variation in concurrency to simulate different sets of user loads:

```
output:
  format: "json"
  dir: "./output/"
  file: "output.json"
warmup: False
warmup_options:
  requests: 10
  timeout_sec: 60
storage: # TODO
type: local
```

```

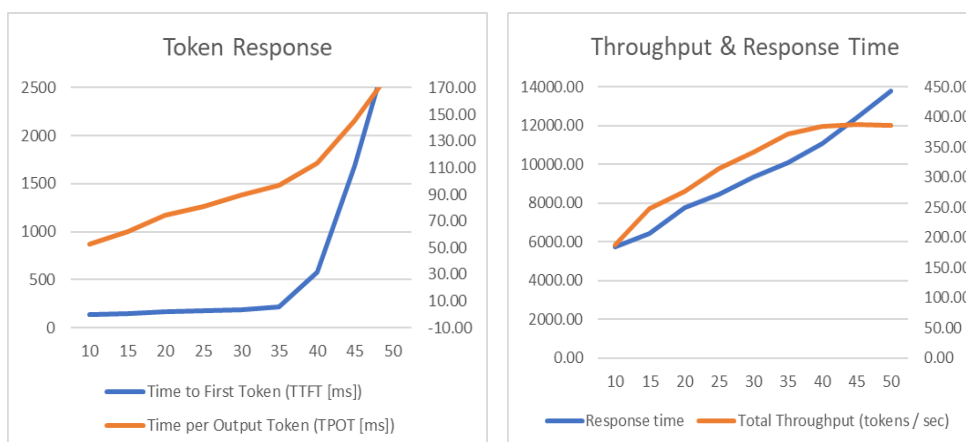
dataset:
  file: "datasets/openorca_large_subset_011.jsonl"
  max_queries: 1000
  min_input_tokens: 0
  max_input_tokens: 1024
  max_output_tokens: 256
  max_sequence_tokens: 1024
load_options:
  type: constant #Future options: loadgen, stair-step
  concurrency: 15
  duration: 600# In seconds. Maybe in future support "100s" "10m", etc...
plugin: "openai_plugin"
plugin_options:
  streaming: True
  model_name: "/mnt/models/"
  host: "https://newllama13b.apps.<your namespace domain>"
  endpoint: "/v1/chat/completions"
extra_metadata:
  replicas: 1

```

## Findings

Based on our test result, we have observed that in every run workload gets offloaded to the GPU, so there is minimal CPU and memory utilization on the instance. GPU overall utilization remains between 92 to 95 percent, and GPU memory utilization remains between 95 to 100 percent.

The graphs below highlight the matrices values for different user load.



### Matrices:

- **Time to First Token (TTFT):** The time taken for the first token to be generated and streamed to each user.
- **Time per Output Token (TPOT):** Average time to generate an output token for each user.
- **Response Time:** An LLM's response time provides a measure of how much time taken to generate a complete response from the time the query is submitted.

- **Total Throughput:** An LLM's throughput provides a measure of how many requests it can process or how much output it can produce in a given time span.

As per the graph, 35 concurrent users are the optimal load on a single replica of Llama 2 model served with vLLM serving runtime. Once the user load increases beyond 35, the token response time increases exponentially, and throughput remains constant.

In our validation we have tested a single replica of a model serving with a single GPU. Our cluster is equipped with two GPUs per node; based on the requirement, we could deploy more than single replica to handle an even larger number of users.



# 6. Summary and Conclusion

This chapter presents the following topics:

**Summary .....42**

**We value your feedback .....42**

## Summary

### Overview

Implementing a digital assistant with Dell APEX Cloud Platform for Red Hat OpenShift with Red Hat OpenShift AI has been developed to address the needs of organizations that need to develop and run custom RAG-based application using domain-specific information that is relevant to their own organization. RAG reduces the need to continuously train the model using an organization's internal knowledge base, feeding the data faster and requiring less resources compared to training a model from scratch or fine tuning.

The technology behind AI is evolving rapidly, and companies might lack AI expertise or have the time to design, deploy, and manage solution stacks at the pace required. Dell Technologies and Red Hat have joined to deliver customers an LLM-based digital assistant built on the reliable, efficient, and scalable Dell APEX Cloud Platform for Red Hat OpenShift, Dell PowerFlex and Dell ObjectScale object storage with Red Hat OpenShift AI.

We have successfully validated the RAG capabilities to provide domain-specific context to the LLM, enabling customers for building an AI solution grounded to their own organizations data, with speed and confidence. Validated workload-optimized architectures can reduce customer proof-of-concept time and eliminate significant design and testing time required to plan correct configurations before the deployment.

## We value your feedback

Dell Technologies and the authors of this document welcome your feedback on the solution and the solution documentation. Contact the Dell Technologies Solutions team by [email](#).

**Author:** Sanjeev Ranjan, Balakrishnan R, Abhishek Sharma

---

**Note:** For links to additional documentation for this solution, see the [Dell Technologies Solutions Info Hub for Artificial Intelligence](#).

---

## 7. References

This chapter presents the following topics:

|  |           |
|--|-----------|
| <b>Dell Technologies documentation .....</b> | <b>44</b> |
| <b>Red Hat documentation .....</b>           | <b>44</b> |
| <b>Llama 2 documentation .....</b>           | <b>44</b> |
| <b>LangChain documentation .....</b>         | <b>44</b> |
| <b>Vector store documentation.....</b>       | <b>44</b> |
| <b>Gradio documentation.....</b>             | <b>44</b> |
| <b>vLLM documentation .....</b>              | <b>45</b> |

## Dell Technologies documentation

The following Dell Technologies documentation provides additional and relevant information. Access to these documents depends on your login credentials. If you do not have access to a document, contact your Dell Technologies representative.

- [\*Dell ObjectScale 1.2.x Administration Guide\*](#)

## Red Hat documentation

The following Red Hat documentation provides additional and relevant information:

- [\*Product Documentation for OpenShift Container Platform 4.13\*](#)
- [\*Product Documentation for Red Hat OpenShift AI\*](#)

## Llama 2 documentation

The following Llama 2 webpage provides additional and relevant information:

- [\*Meta Llama 2\*](#)

## LangChain documentation

The following LangChain documentation provides additional and relevant information:

- [\*LangChain Python Docs\*](#)

## Vector store documentation

The following documentation provides additional and relevant information related to Vector store:

- [\*PGVector docs\*](#)
- [\*Redis docs\*](#)

## Gradio documentation

The following Gradio documentation provides additional and relevant information:

- [\*Gradio docs\*](#)

## vLLM documentation

The following vLLM documentation provides additional and relevant information:

- [vLLM docs](#)

# Appendix A Open-Source License

This appendix presents the following topics:

**Open-Source Licensing Information.....47**

## Open-Source Licensing Information

This document provides design guidance on configuring a digital assistant using open-source tools. Open-source tools are distributed under various licenses, each with its own terms and conditions. Review and understand the license terms of each open-source tool you use before using it in your own solution. This will help you to ensure that you are complying with the license requirements and that you are not infringing on the intellectual property rights of the copyright holders.

It is the responsibility of the user to review and comply with the licensing terms and conditions of each tool mentioned in this document. Licensing information for each tool can be found on the respective tool's official website or in its documentation. We have also listed links to some of the open-source tools license page below.

All example codes are present in the GitHub with [MIT License](#).

### Llama 2

Llama 2 is an open-source LLM. The licensing information for Llama 2 can be found on the [Meta web page](#).

### LangChain

LangChain is an open-source tool that can be used to work with LLMs. The licensing information for LangChain can be found on the [LangChain GitHub web page](#).

### Vector Store

Redis is an open-source, in-memory data structure store used as a database, cache, message Broker agent, and streaming. The licensing information for Redis can be found on the [Redis web page](#).

PGVector is an open-source PostgreSQL extension that provides adds Vector store capability to PostgreSQL. The licensing information for PGVector can be found on the [PGVector GitHub web page](#).

### Gradio

Gradio is an open-source tool that helps to generate an easy-to-use UI machine learning web apps with few lines of code. The licensing information for Gradio is available on the [Gradio GitHub web page](#).

### vLLM

vLLM is a fast and simple library for LLM inference and serving. The licensing information for vLLM is available on the [vLLM GitHub web page](#).