

Data Lakehouse

Dell Validated Design for Analytics Design Guide

Abstract

This design guide provides architecture and design information for the Dell Validated Design for Analytics — Data Lakehouse with Robin Cloud Native Platform and Delta Lake. Built on PowerSwitch networking and Intel-powered PowerEdge servers, it includes PowerScale and Elastic Cloud Storage (ECS) object storage for HDFS and Amazon S3.

Dell Technologies Solutions



Copyright

© 2022 Dell Inc. or its subsidiaries. All rights reserved. Dell Technologies, Dell, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

Revision history.....	5
Chapter 1: Introduction.....	6
Solution introduction.....	6
Overview.....	6
Document purpose.....	6
Audience.....	6
Design Guide introduction.....	7
Chapter 2: Solution concepts	8
Overview.....	8
Concepts.....	8
Lakehouse architecture.....	8
Container platform.....	8
Decoupled compute and storage.....	9
Chapter 3: Solution architecture.....	10
Architecture overview.....	10
Design overview.....	10
Platform features.....	11
Dell infrastructure.....	11
Infrastructure overview.....	11
Server infrastructure.....	12
Storage infrastructure.....	18
Network infrastructure.....	20
Container platform implementation.....	22
Container platform overview.....	22
Robin roles.....	22
Orchestration.....	23
Compute.....	24
Storage.....	24
Management and administration.....	25
Network.....	25
Workload design.....	26
Overview.....	26
Spark bundle.....	27
Kafka bundle.....	28
Chapter 4: Software infrastructure validation.....	30
Overview.....	30
Software components.....	30
Component validation.....	32
GPU validation.....	32
Platform validation.....	33
Spark validation.....	33

Kafka validation.....	35
High availability validation.....	36
Findings.....	37
Summary.....	37
Chapter 5: Sizing the solution.....	38
Sizing and scaling overview.....	38
Sizing guidelines.....	38
Cluster sizing.....	38
Lakehouse storage sizing.....	39
Control plane sizing.....	40
Worker node sizing.....	40
Workload sizing.....	41
Scaling guidelines.....	41
Scaling overview.....	41
Compute scaling.....	41
Storage scaling.....	42
Network scaling.....	42
Chapter 6: Summary.....	43
Overview.....	43
We value your feedback.....	43
Chapter 7: References.....	44
Dell Technologies documentation.....	44
Delta Lake documentation.....	45
Robin.io documentation.....	45
Dell Technologies Customer Solution Centers.....	45
Dell Technologies Info Hub.....	45
More information.....	45

Revision history

Table 1. Document revision history

Date	Document revision	Description of changes
October 2022	1.0	Initial release

Introduction

Topics:

- [Solution introduction](#)
- [Design Guide introduction](#)

Solution introduction

Overview

The concept of a data lakehouse is an open data management architecture that combines the best aspects of a data lake and a data warehouse in a single platform.

A traditional data lake tends to be flexible and cost-effective by storing data in its raw or natural form typically unstructured or semistructured. A data warehouse is a more advanced repository of data for reporting and analysis that tends to store data that is more structured. The data has typically been cleansed or operationalized for better data quality, often the result of extract-transform-load (ETL) or extract-load-transform (ELT) operations.

Data analytics usage is increasingly widespread, and changing in nature. Those trends, coupled with the need to access large amounts of disparate data by many different users, mean that organizations need a new, more integrated approach to data access. A data lakehouse combines the best of data lake and data warehouses, supporting business intelligence and machine learning technologies in one platform. The platform can store all types of data and provide it with a cloud-like, multiresource, and self-service interface for data scientists and other users.


The Dell Validated Design for Analytics — Data Lakehouse has been developed to address the needs of organizations deploying advanced analytics. It incorporates the concepts of a lakehouse architecture together with a container platform using decoupled compute and storage.

This document provides design guidance for data analytics infrastructure managers and architects by describing a predesigned, validated, and scalable architecture for advanced analytics on Dell hardware infrastructure.

Document purpose

This document describes a validated design and reference architecture for a data lakehouse platform that is integrated with a Kubernetes-based container platform. Together they address the needs of organizations deploying advanced analytics and AI workloads.

This design guide can be read alongside the associated white paper, [Dell Validated Design for Analytics – Data Lakehouse](#). The white paper discusses the benefits of a data lakehouse compared to legacy data lakes and warehouses. It also provides a more general overview of the data lakehouse concept and its component technologies.

 **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 enterprises with data lakes or a data lake strategy interested in empowering their organizations to act more quickly, effectively, and efficiently on their data. Audience roles include:

- Data and application administrators
- IT decision-makers

- Data scientists
- Data engineers

A data lakehouse can assist more traditional analytics customers looking to modernize their data collection. It can also help analytics systems to get more value from their data or standardize their data for modern analytics workloads.

Design Guide introduction

This document presents the key concepts of the solution, namely the lakehouse architecture, the container platform, and the use of decoupled compute and storage. It describes the solution architecture and components, including the Dell infrastructure and the node configuration that support the functions of the data lakehouse. The Dell storage configuration options for the lakehouse are described, including both PowerScale and ECS.

In addition to the lakehouse, the container platform architecture and components are described. The container platform in this architecture is Robin Cloud Native Platform (Robin CNP), which is based on Kubernetes. Several examples of application bundles that can be deployed are shown, including Spark and Kafka.

Going beyond the Dell hardware infrastructure, the software infrastructure components that were validated are described, including the Robin CNP.

Lastly, guidelines are presented for sizing and scaling the solution based on various workload requirements.

Solution concepts

Topics:

- [Overview](#)
- [Concepts](#)

Overview

The Dell Validated Design for Analytics — Data Lakehouse incorporates the concepts of a lakehouse architecture together with a container platform using decoupled compute and storage.

Concepts

Lakehouse architecture

Lakehouse architecture gained popularity in the late 2010s as an evolution of the well-established data warehouse and data lake architectures. The architecture provides the most significant capabilities of both data warehouses and data lakes in a single system, reducing cost and complexity without compromising functionality.

There is no formal definition of lakehouse architecture. It primarily describes a system that combines the open file formats and cost-effective scalable storage of data lakes with the ACID transactions and table-oriented schema definitions of data warehouses.

Lakehouse architecture is often based on a modern table format such as Delta Lake or Apache Iceberg. This format provides a table abstraction above the underlying storage layer in the lakehouse. The other key features of a lakehouse are:

- ACID transaction support for inserts, updates, and deletes
- Scalable metadata
- Schema enforcement and evolution
- Support for diverse datatypes ranging from unstructured to structured data
- Data versioning or time travel capability
- Support for SQL access
- Support for direct table access through APIs such as DataFrames
- Support for scalable storage using open file formats

Container platform

The Dell Validated Design for Analytics — Data Lakehouse is built on a container platform. This approach provides agility, flexibility, and scalability while supporting diverse analytics workloads.

The container platform abstracts the machine level details and host operating system dependencies, exposing them as a pool of compute, storage, and communications resources. The platform also provides application orchestration capabilities to streamline the deployment and management of analytics workloads.

Applications on the platform use the industry standard OCI image format. The images include all the required software components, isolating them from operating system dependencies. These images are run in containers, providing a high level of run-time isolation from other applications.

Analytics workloads are packaged into application bundles that combine application images and deployment specifications. The bundles simplify application launching and specification of resource requirements.

The container platform is Robin Cloud Native Platform, which is based on Kubernetes.

Decoupled compute and storage

This validated design for data analytics decouples compute and storage resources. This approach provides increased resource utilization, increased flexibility, and lower costs.

The system supports independent provisioning of storage and compute and enables the use of heterogeneous storage and compute resources. This design provides better balance between storage and compute for varying workloads.

Decoupling also simplifies the life cycle and management of the system by allowing independent management, scaling, and upgrades of storage and compute resources.

The design also separates runtime storage from data lake storage. Robin Cloud Native Storage provides runtime storage. PowerScale with the HDFS protocol, or ECS with the S3 protocol, provides data lake storage.

Solution architecture

Topics:

- [Architecture overview](#)
- [Dell infrastructure](#)
- [Container platform implementation](#)
- [Workload design](#)

Architecture overview

The Dell Validated Design for Analytics — Data Lakehouse has been developed to address the needs of organizations deploying advanced analytics. It incorporates the concepts of a lakehouse architecture along with a container platform using decoupled compute and storage.

Design overview

The architecture has been designed to support the requirements of modern data analytics. It provides a scalable, flexible platform for multiple analytics workloads, and supports the use of a lakehouse architecture. The figure below illustrates the overall lakehouse platform architecture.

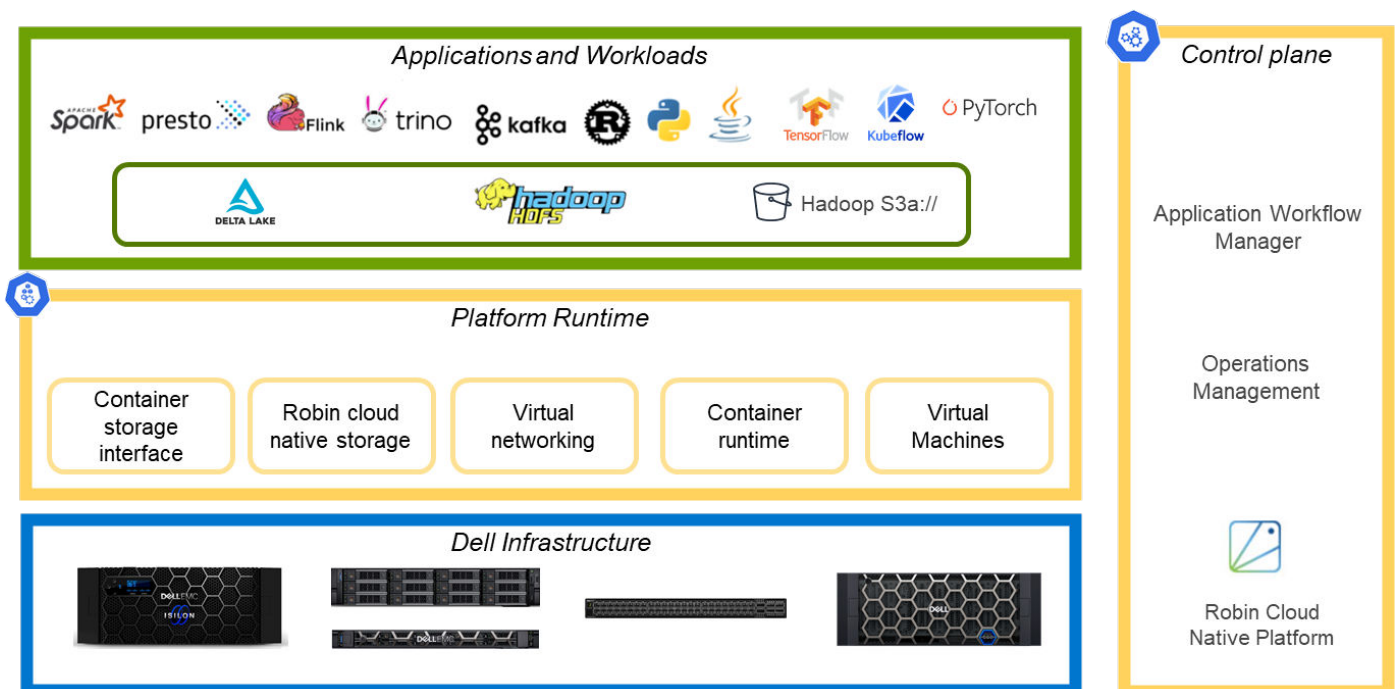


Figure 1. Lakehouse platform architecture

Dell infrastructure

The infrastructure layer provides the core compute, storage, and network resources for the platform. For details about the recommended infrastructure, see [Dell infrastructure](#). For additional sizing and scaling guidance, see [Sizing the solution](#).

Control plane	The control plane is the set of components that provide platform administration, operations management, and application orchestration capabilities. The Robin Cloud Native Platform provides the control plane.
Platform runtime	Managed by the control plane, the platform runtime components abstract the infrastructure details, exposing them as a pool of compute, storage, and communications resources. For details about both the control plane and platform runtime components, see Container platform implementation .
Applications and workloads	Applications and workloads are the components that provide the customer specified data analytics functions. These applications are primarily containerized and use the industry standard Open Container Initiative (OCI) image format. Applications can be packaged into application bundles that include runtime orchestration and deployment information. For details about core applications used for lakehouse architecture, see Workload design . Design guides for other solutions running on this platform can be found on the Dell Info Hub for Data Analytics .

Platform features

The architecture provides scalability of compute, storage, and networking at the infrastructure layer, scalability of the control plane and runtime layers, and scalability of application workloads. Storage and compute are decoupled to support independent scaling depending on application requirements and the network layer scales with infrastructure growth.

The architecture provides the flexibility to run multiple workloads on a single platform with resource management and operations management. Workloads can be containerized cloud native applications or can run in virtual machines. Users can be organized into tenants, and permissions and resource allocation can be controlled at the user and tenant levels.

The architecture also implements high-availability features at all levels. Redundant networking is used for the core platform, Elastic Cloud Storage (ECS), and PowerScale storage. The container platform implements High Availability (HA) through redundancy for the core components. It supports automatic replication for cloud native workloads, and provides application backup, snapshot, and restore capabilities.

Overall, the architecture enables data analytics on structured, semistructured, and unstructured data on a single platform. It gives developers, data scientists, and data engineers a single interface for deployment of advanced analytic pipelines. It provides administrators with a production grade platform that is based on the de facto standard Kubernetes ecosystem.

Dell infrastructure

Infrastructure overview

Dell infrastructure provides the compute, memory, storage, and network resources for the platform. [Lakehouse platform on Dell infrastructure](#) illustrates the required infrastructure components and their roles in the cluster.

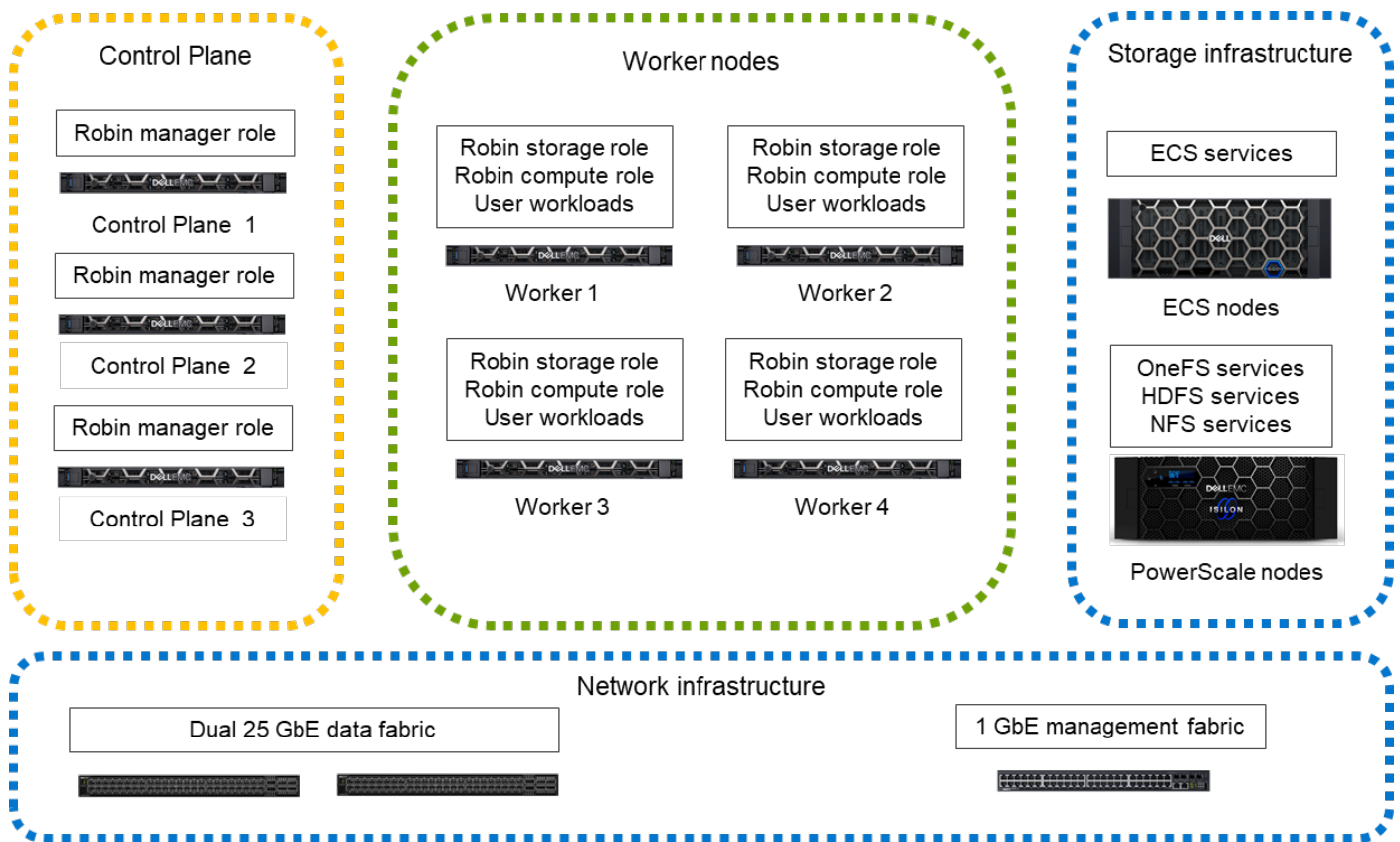


Figure 2. Lakehouse platform on Dell infrastructure

Control nodes	Control nodes support the core platform services that are defined in the Robin Cloud Native Platform Manager role.
Worker nodes	Worker nodes support the platform runtime services and customer workloads. The runtime services are defined in the Robin Cloud Native Platform Compute and Storage roles, while user deployments define the workloads. Supported nodes include standard and compute accelerated nodes.
Storage infrastructure	Storage infrastructure provides the lakehouse storage layer. PowerScale and ECS configurations are provided. You can use either or both.
Network infrastructure	Network infrastructure provides the required connectivity between the server and storage infrastructure and the on-premises network.

The architecture is flexible and supports a wide range of infrastructure configurations. Dell Technologies recommends the following configurations for a general-purpose lakehouse environment running analytics workloads. Individual node configurations include node-specific guidance. More guidance for cluster level sizing and scaling is provided in [Sizing the solution](#). Design guides for individual solutions on the platform may include more specific recommendations for those solutions.

Server infrastructure

The server infrastructure provides compute, memory, and some of the storage resources that are required to run customer workloads. A wide variety of PowerEdge server configurations are possible. The recommendations here support a wide variety of workloads typical in a lakehouse architecture implementation.

Lakehouse control plane node

Lakehouse control plane nodes support the core platform services that are defined in the Robin Cloud Native Platform (Robin CNP) Manager role. Dell recommends the configuration in [Lakehouse control plane node configuration](#) as a starting point for these nodes.

Table 2. Lakehouse control plane node configuration

Machine function	Component
Platform	PowerEdge R650xs server
Chassis	2.5 in chassis with up to 10 hard drives, including up to eight SAS or SATA drives, or nine NVMe drives
Chassis configuration	Riser configuration 4, one OCP 3.0 16-channel slot plus one 16-channel low profile slot
Power supply	Dual hot-plug, PSU (1+1), 800, mixed mode
Processor	Intel Xeon Gold 6326 2.9 G, 16 C/32 T, 11.2 GT/s, 24 M cache, turbo, HT (185 W) DDR4-3200
Memory capacity	64 GB (four 16 GB RDIMM, 3200 MT/s, dual rank)
Internal RAID storage controllers	PERC H755 SAS front
Disk—HDD	None
Disk—SSD	Two 800 GB SSD SAS ISE mixed-use 12 Gbps 512e 2.5 in hot-plug AG drives, 3 DWPD
Disk—NVMe	None
Open Compute Project (OCP) network interface controller	Broadcom 57414 dual port 10/25 GbE SFP28, OCP NIC 3.0
Extra network interface controllers	Broadcom 57414 dual port 10/25 GbE SFP28 adapter, PCIe low profile (optional)

Dell Technologies recommends the disk volume and partition layouts for this set of machines that are listed in [Lakehouse control plane node volumes](#) and [Lakehouse control plane node partitions](#).

Table 3. Lakehouse control plane node volumes

Usage	Volume type	Physical disks	Volume ID
Operating system and Robin CNP	RAID 1	Two 800 GB SAS SSDs	0

Table 4. Lakehouse control plane node partitions

Mount point	Size	File system type	Volume ID	Partition type	Description
/boot	1024 MB	XFS	0	Primary	This partition contains BIOS start-up files that must be within first 2 GB of disk.
/boot/efi	650 MB	VFAT	0	Extended	This partition contains EFI start-up files.
swap	6 GB	swap	0	Swap	This partition contains the operating system swap space partition.
/	200 GB	XFS	0	LVM	This partition contains the root file system.
/var	60 GB	XFS	0	LVM	This partition contains variable data like system logging files,

Table 4. Lakehouse control plane node partitions (continued)

Mount point	Size	File system type	Volume ID	Partition type	Description
					databases, mail and printer spool directories, transient, and temporary files.
<code>/var/lib/docker</code>	400 GB	XFS	0	LVM	This partition contains the container images.
<code>/home</code>	50 GB	XFS	0	LVM	This partition contains the user home directories.
<code>/home/robinds</code>	40 GB	XFS	0	LVM	This partition contains the Robin home directory.
<code>/home/robinds/ var/lib/pgsql</code>	20 GB	XFS	0	LVM	This partition contains the PostgreSQL database that Robin uses.
<code>/home/robinds/ var/log</code>	10 GB	XFS	0	LVM	This partition contains the Robin log files.
<code>/home/ robinds/var/ crash</code>	10 GB	XFS	0	LVM	This partition contains the Robin crash files.

Three control plane nodes are required for production clusters to provide high availability for the control plane. For pilot testing, the control plane services that are defined in the Robin CNP Manager role can be assigned to worker nodes instead of using dedicated control plane nodes.

Memory, storage, and processor have been sized to support all the required services in a production deployment.

The configuration includes two network ports to support high-availability (HA) networking. These ports can be from a single network card, or a pair of network cards for additional adapter level HA.

Two SDDs in a RAID 1 configuration are used for the operating system volume. The swap partition is small since swapping causes excessive latency for critical cluster infrastructure. The home directories are allocated in a separate small partition since user files should not be stored on infrastructure nodes. Most of the storage is allocated to the `/var` partition for runtime files.

You can use LVM to adjust the storage allocation between `/`, `/home`, and `/var` for specific needs.

General purpose lakehouse worker node

Lakehouse worker nodes support the platform runtime services and customer workloads. The runtime services are defined in the Robin Cloud Native Platform Compute and Storage roles. User deployments define the workloads. Dell Technologies recommends the configuration in [General purpose lakehouse worker node configuration](#) as a starting point for general-purpose workloads.

Table 5. General purpose lakehouse worker node configuration

Machine function	Component
Platform	PowerEdge R650 server
Chassis	2.5 in chassis with up to eight SAS or SATA hard drives, three PCIe slots, and two CPUs
Chassis configuration	Riser configuration 0, 2 CPUs, half length, low profile, three 16-channel slots, SW GPU capable

Table 5. General purpose lakehouse worker node configuration (continued)

Machine function	Component
Power supply	Dual hot-plug, redundant power supplies (1+1), 1400 W, mixed mode
Processor	Dual Intel Xeon Gold 6348 2.6 G, 28 C/56 T, 11.2 GT/s, 42 M Cache, turbo, HT (235 W) DDR4-3200
Memory capacity	256 GB (sixteen 16 GB RDIMM, 3200 MT/s, dual rank)
Internal RAID storage controllers	PERC H755 SAS front
Disk—HDD	None
Disk—SSD	Two 480 GB SSD SATA mixed-use 6 Gbps 512 2.5 in hot-plug AG drives, 3 DWPD
Disk—SSD	Three 3.84 TB SSD vSAS mixed-use 12 Gbps 512e 2.5 in hot-plug, AG drives, SED, 3 DWPD
Disk—NVMe	None
Boot optimized storage cards	None
Ethernet mezzanine adapters	Broadcom 57414 dual port 25 GbE OCP SFP28 LOM Mezzanine card
Extra network interface controllers	Broadcom 57414 dual port 10/25 GbE SFP28 adapter, PCIe low profile (optional)

Dell Technologies recommends the disk volume and partition layouts for this set of machines that are listed in [General purpose lakehouse worker node volumes](#) and [General purpose lakehouse worker node partitions](#).

Table 6. General purpose lakehouse worker node volumes

Usage	Volume type	Physical disks	Volume ID
Operating system	RAID 1	Two 480 GB SSDs	0
Robin Cloud Native Storage (Robin CNS)	No RAID	Three 3.84 TB SAS HDDs	1, 2, 3

Table 7. General purpose lakehouse worker node partitions

Mount point	Size	File system type	Volume ID	Partition type	Description
/boot	1024 MB	XFS	0	Primary	This partition contains BIOS start-up files that must be within first 2 GB of disk.
/boot/efi	650 MB	VFAT	0	Extended	The partition contains EFI start-up files.
/	Around 400 GB	XFS	0	LVM	This partition contains the root file system.
swap	6 GB	swap	0	Swap	This partition contains the operating system swap space partition.
/home	10 GB	XFS	0	LVM	This partition contains the user home directories.

Table 7. General purpose lakehouse worker node partitions (continued)

Mount point	Size	File system type	Volume ID	Partition type	Description
/var	60 GB	XFS	0	LVM	This partition contains variable data like system logging files, databases, mail and printer spool directories, transient, and temporary files.
None	3.84 TB	Robin CNS	1	Raw partition	Robin Cloud Native Storage manages this partition.
None	3.84 TB	Robin CNS	2	Raw partition	Robin Cloud Native Storage manages this partition.
None	3.84 TB	Robin CNS	3	Raw partition	Robin Cloud Native Storage manages this partition.

Dell Technologies recommends four worker nodes for a minimum deployment.

The configuration includes two network ports to support high-availability (HA) networking. These ports can be from a single network card, or a pair of network cards for additional adapter level HA.

Two SSDs in a RAID 1 configuration are used for the operating system volume. The swap partition is small since swapping causes excessive latency for critical cluster infrastructure. The home directories are allocated in a separate small partition since user files are not stored at the operating system level on production nodes. Most of the storage is allocated to the /var partition for runtime files. You can use LVM to adjust the storage allocation between /, /home, and /var for specific needs.

Three SSDs are allocated for use by Robin CNS. The services to support this storage are deployed with the Robin Storage role. This storage is exposed to workloads running on the cluster through the Kubernetes CSI interface. The recommended configuration provides approximately 11 TB of storage per node. This capacity is enough for typical runtime storage in a lakehouse environment where the bulk of the data is stored on external storage. The external storage can be either HDFS provided by PowerScale, or object storage provided by ECS. If more local storage is needed, up to three more SSDs can be added, and drive sizes can be increased.

Memory has been sized to support all the required services in a production deployment with enough headroom for user workloads. The most common change is to increase the memory size to support more containers or workloads requiring more memory.

The processors have been chosen to support compute intensive AI and ML workloads and include dual Intel Advanced Vector Extensions (AVX) units for maximum compute speed. Other processor choices are possible but should be made with memory requirements and overall power consumption in mind.

GPU-accelerated lakehouse worker node

GPU-accelerated lakehouse worker node support the platform runtime services and customer workloads that benefit from GPU acceleration. Robin Cloud Native Platform Compute and Storage roles define the runtime services, while user deployments define the workloads. Dell Technologies recommends the configuration in [GPU-accelerated lakehouse worker node configuration](#) as a starting point for GPU-accelerated workloads.

Table 8. GPU-accelerated lakehouse worker node configuration

Machine function	Component
Platform	PowerEdge R750 server
Chassis	2.5 in chassis with up to 16 SAS or SATA drives
Chassis configuration	Riser configuration 2, full-length, four 16-channel slots, two 8-channel slots, DW GPU-capable

Table 8. GPU-accelerated lakehouse worker node configuration (continued)

Machine function	Component
Power supply	Dual, hot-plug power supplies, 2400 W redundant, D mixed-mode
Processor	Intel Xeon Gold 6348 2.6 G, 28 C/56 T, 11.2 GT/s, 42 M cache, turbo, HT (235 W) DDR4-3200
Memory capacity	256 GB (sixteen 16 GB RDIMM, 3200 MT/s, dual rank)
Internal RAID storage controllers	PERC H755 SAS front
Disk—HDD	None
Disk—SSD	Two 480 GB SSD SATA mixed-use 6 Gbps 512 2.5 in hot-plug AG drives, 3 DWPD
Disk—SSD	Three 3.84 TB SSD vSAS mixed-use 12 Gbps 512e 2.5in hot-plug AG drives, SED, 3 DWPD
Disk—NVMe	None
Boot optimized storage cards	None
OCP 3.0 network interface controllers	Broadcom 57414 dual port 10/25 GbE SFP28, OCP NIC 3.0
Extra network interface controllers	None
GPU, FPGA, or acceleration cards	NVIDIA Ampere A30, PCIe, 165 W, 24 GB passive, double wide, full-height GPU with cable

Dell Technologies recommends the disk volume and partition layouts for this set of machines that are listed in [GPU-accelerated lakehouse worker node volume](#) and [GPU-accelerated lakehouse worker node partitions](#).

Table 9. GPU-accelerated lakehouse worker node volume

Usage	Volume type	Physical disks	Volume ID
Operating system	RAID 1	Two 480 GB SSDs	0
Robin Cloud Native Storage (Robin CNS)	No RAID	Three 3.84 TB SAS SSDs	1, 2, 3

Table 10. GPU-accelerated lakehouse worker node partitions

Mount point	Size	File system type	Volume ID	Partition type	Description
/boot	1024 MB	XFS	0	Primary	This partition contains BIOS start-up files that must be within first 2 GB of disk.
/boot/efi	650 MB	VFAT	0	Extended	The partition contains EFI start-up files.
/	Around 400 GB	XFS	0	LVM	This partition contains the root file system.
swap	6 GB	swap	0	Swap	This partition contains the operating system swap space partition.
/home	10 GB	XFS	0	LVM	This partition contains the user home directories.

Table 10. GPU-accelerated lakehouse worker node partitions (continued)

Mount point	Size	File system type	Volume ID	Partition type	Description
/var	60 GB	XFS	0	LVM	This partition contains variable data like system logging files, databases, mail and printer spool directories, transient, and temporary files.
None	3.84 TB	Robin CNS	1	Raw partition	Robin Cloud Native Storage manages this partition.
None	3.84 TB	Robin CNS	2	Raw partition	Robin Cloud Native Storage manages this partition.
None	3.84 TB	Robin CNS	3	Raw partition	Robin Cloud Native Storage manages this partition.

GPU accelerated worker nodes have similar capabilities to general purpose worker nodes while adding GPU acceleration. Dell Technologies recommends four worker nodes for a minimum deployment. You can use any mix of general purpose and GPU-accelerated nodes.

The configuration includes two network ports to support high-availability (HA) networking. These ports can be from a single network card, or a pair of network cards for additional adapter level HA.

Two SSDs in a RAID 1 configuration are used for the operating system volume. The swap partition is small since swapping causes excessive latency for critical cluster infrastructure. The home directories are allocated in a separate small partition since user files are not stored at the operating system level on production nodes. Most of the storage is allocated to the /var partition for runtime files. You can use LVM to adjust the storage allocation between /, /home, and /var for specific needs.

Three SSDs are allocated for use by Robin CNS. The services to support this storage are deployed with the Robin Storage role. This storage is exposed to workloads running on the cluster through the Kubernetes CSI interface. The recommended configuration provides approximately 11 TB of storage per node. This capacity is enough for typical runtime storage in a lakehouse environment where the bulk of the data is stored on external storage. The external storage can be either HDFS provided by PowerScale, or object storage provided by ECS. If more local storage is needed, up to three more SSDs can be added, and drive sizes can be increased.

Memory has been sized to support all the required services in a production deployment with enough headroom for user workloads. The most common change is to increase the memory size to support more containers or workloads requiring more memory.

The processors have been chosen to support compute intensive AI and ML workloads and include dual Intel Advanced Vector Extensions (AVX) units for maximum compute speed. Other processor choices are possible but should be made with memory requirements and overall power consumption in mind.

The GPU has been chosen to support Spark acceleration of SQL and dataframe operations using the [NVIDIA RAPIDS Accelerator for Apache Spark](#). These are typical workload operations in a lakehouse environment. One or two GPUs can be used in this configuration. AI and ML workloads may benefit from alternative GPU models.

Storage infrastructure

Lakehouse storage can use PowerScale with the HDFS protocol, or ECS for object storage. You can use either or both systems depending on workload requirements. Server internal storage provides runtime storage that applications require. The Robin Cloud Native Storage software manages runtime storage.

ECS and PowerScale are deployed as cluster-level systems. The node recommendations here can be used as guidance for new clusters, verification of compatibility with existing clusters, or expansion of existing clusters.

PowerScale infrastructure

Dell Technologies recommends the configuration in [PowerScale configuration](#) for storage in clusters using PowerScale for their primary lakehouse storage using HDFS.

Table 11. PowerScale configuration

Machine function	Component
Model	Isilon H5600 (hybrid)
Chassis	4U node
Nodes per chassis	4
Node storage	Twenty 10 TB 3.5 in 4 Kn SATA HDDs
Node cache	Two 3.2 TB SSDs
Usable capacity per chassis	500 TB
Front-end networking	Two 25 GbE (SFP28)
Infrastructure (back-end) networking	Two InfiniBand QDR or two 40 GbE (QSFP+)

The recommended configuration is sized for typical usage as lakehouse HDFS storage.

Two Ethernet network ports per node are included for connection to the cluster network or a PowerScale storage network. Two additional network ports are included for connection to the PowerScale back-end network. These additional ports can be either InfiniBand QDR or 40 GbE, depending upon on-site preferences.

One Isilon H5600 chassis supports four Isilon H5600. This configuration provides approximately 594 TB of usable storage. At 85% utilization, 500 TB of HDFS storage is a good guideline for available storage per chassis.

This configuration assumes that the PowerScale nodes are primarily used for HDFS storage. If the PowerScale nodes are used for other storage applications or clusters, you must account for it in the overall cluster sizing. You can also use other Isilon H5600 drive configurations.

ECS node

Dell Technologies recommends the configurations in [ECS EX500 node configuration](#) or [ECS EXF900 node configuration](#) for storage in clusters using ECS for their primary lakehouse storage using the `s3a://` protocol.

Table 12. ECS EX500 node configuration

Machine function	Component
Model	ECS EX500
Chassis	2U node
Nodes per rack	16
Node storage	960 GB SSD
Node cache	N/A
Usable capacity per chassis	Slightly less than 384 TB
Front-end networking	Two 25 GbE (SFP28)
Infrastructure (back-end) networking	Two 25 GbE (SFP28)

The ECS EX500 configuration provides a good balance of storage density and performance for lakehouse usage.

Two Ethernet network ports per node are included for connection to the cluster network or an ECS storage network. Two additional network ports are included for connection to the ECS back-end network.

Table 13. ECS EXF900 node configuration

Machine function	Component
Model	ECS EXF900
Chassis	2U node
Nodes per rack	16
Node storage	184 TB (twenty-four 7.68 TB NVMe drives)
Node cache	N/A
Usable capacity per chassis	Slightly less than 184 TB
Front-end networking	Two 25 GbE (SFP28)
Infrastructure (back-end) networking	Two 25 GbE (SFP28)

The ECS EXF900 configuration is an all-flash configuration and provides the highest performance for lakehouse usage.

Two Ethernet network ports per node are included for connection to the cluster network or an ECS storage network. Two additional network ports are included for connection to the ECS back-end network.

Server storage

The worker nodes include local SSD storage for use as runtime storage for workloads. This storage is exposed to pods running on the cluster through the Kubernetes CSI interface. Persistent and ephemeral volumes are supported. Robin Cloud Native Storage (Robin CNS) manages these storage devices through the Robin Storage role.

Robin CNS and the associated CSI driver support advanced features including:

- Replication for data availability
- Encryption for data security
- Compression and thin provisioning for storage efficiency
- Snapshots
- Backup and recovery

The Kubernetes storage class defines some of these features, while others are accessible through the Robin management interface. More details about the Robin storage capabilities can be found in the Robin.io document, [Robin Cloud Native Storage Overview](#).

The recommended configuration provides approximately 11 TB of storage per node. This capacity is enough for typical runtime storage in a lakehouse environment where the bulk of the data is on external storage. Alternative configurations, including dedicated storage nodes, are possible. Some analytics workloads, like databases, may require additional storage for persistent data outside the lakehouse storage.

Network infrastructure

The network is designed to meet the needs of a high performance and scalable cluster, while providing redundancy and access to management capabilities. The architecture is a leaf and spine model that is based on Ethernet networking technologies. It uses PowerSwitch S5248F-ON switches for the leaves and PowerSwitch Z9432F-ON switches for the spine.

Physical networking in this architecture is straightforward since most of the advanced capabilities of the system are implemented using software defined networking. The logical network is described in [Container platform implementation](#). This architecture has three physical networks, as shown in [Physical network infrastructure](#).

iDRAC (or BMC) network The iDRAC (or BMC) network is a secured and isolated network for switch and server hardware management, including access to the iDRAC9 module and Serial-over-LAN. This network optionally connects to corporate network management allowing more direct access to the hardware infrastructure. Each node on this network is assigned an individual IP address from the management address space.

Cluster data network The Cluster data network is the primary network for internode communication between all server and storage nodes. Each server node on this network is assigned a single IP address on this network.

Core data center network The Core data center network is the existing enterprise network. The cluster data network is interfaced with this network through switching and routing allowing cluster services to be exposed to system users.

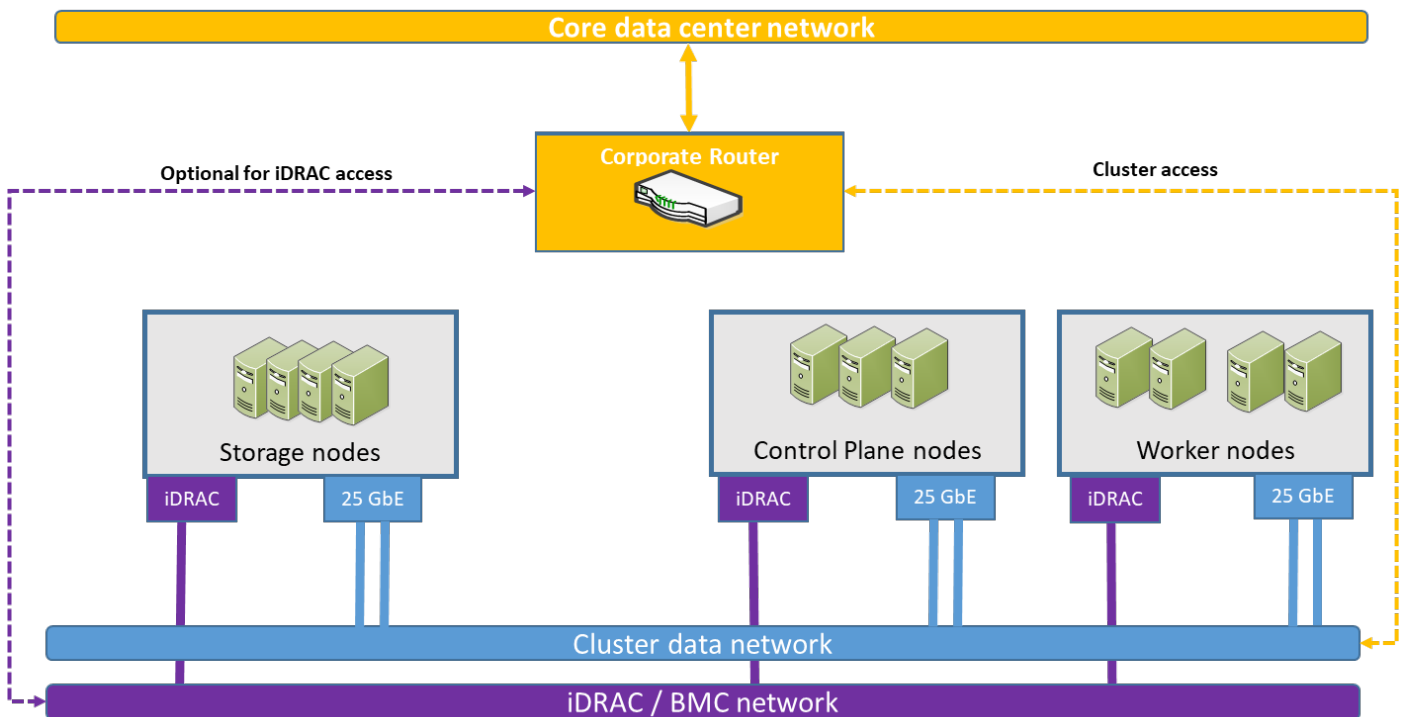


Figure 3. Physical network infrastructure

Management network fabric

The management network uses a PowerSwitch S3148 1 GbE switch for iDRAC connectivity and chassis management.

Cluster data network fabric

The cluster data network uses a scalable, resilient, nonblocking fabric with a leaf-spine design as shown in [Cluster data network connections](#). Each node on this network is connected to two S5248F-ON leaf switches with 25 GbE network interfaces. The switches run Dell SmartFabric OS10. SmartFabric OS10 enables multilayered disaggregation of network functions that are layered on an open-source Linux-based operating system.

On the server side, the two network connections are bonded and have a single IP address assigned.

On the switch side, the network design employs a Virtual Link Trunking (VLT) connection between the two leaf switches.

VLT technology enables a server to uplink multiple physical trunks into more than one S5248F-ON switch by treating the uplinks as one logical trunk. In a VLT environment, connected pair of switches acts as a single switch to a connecting server while all paths are active. It is possible to achieve high throughput while still providing resiliency against hardware failures. VLT replaces Spanning Tree Protocol (STP)-based networks, providing both redundancy and full bandwidth utilization using multiple active paths.

The VLT configuration in this design uses four 100 GbE ports between each Top of Rack (ToR) switch. The remaining 100 GbE ports can be used for high-speed connectivity to spine switches, or directly to the data center core network infrastructure.

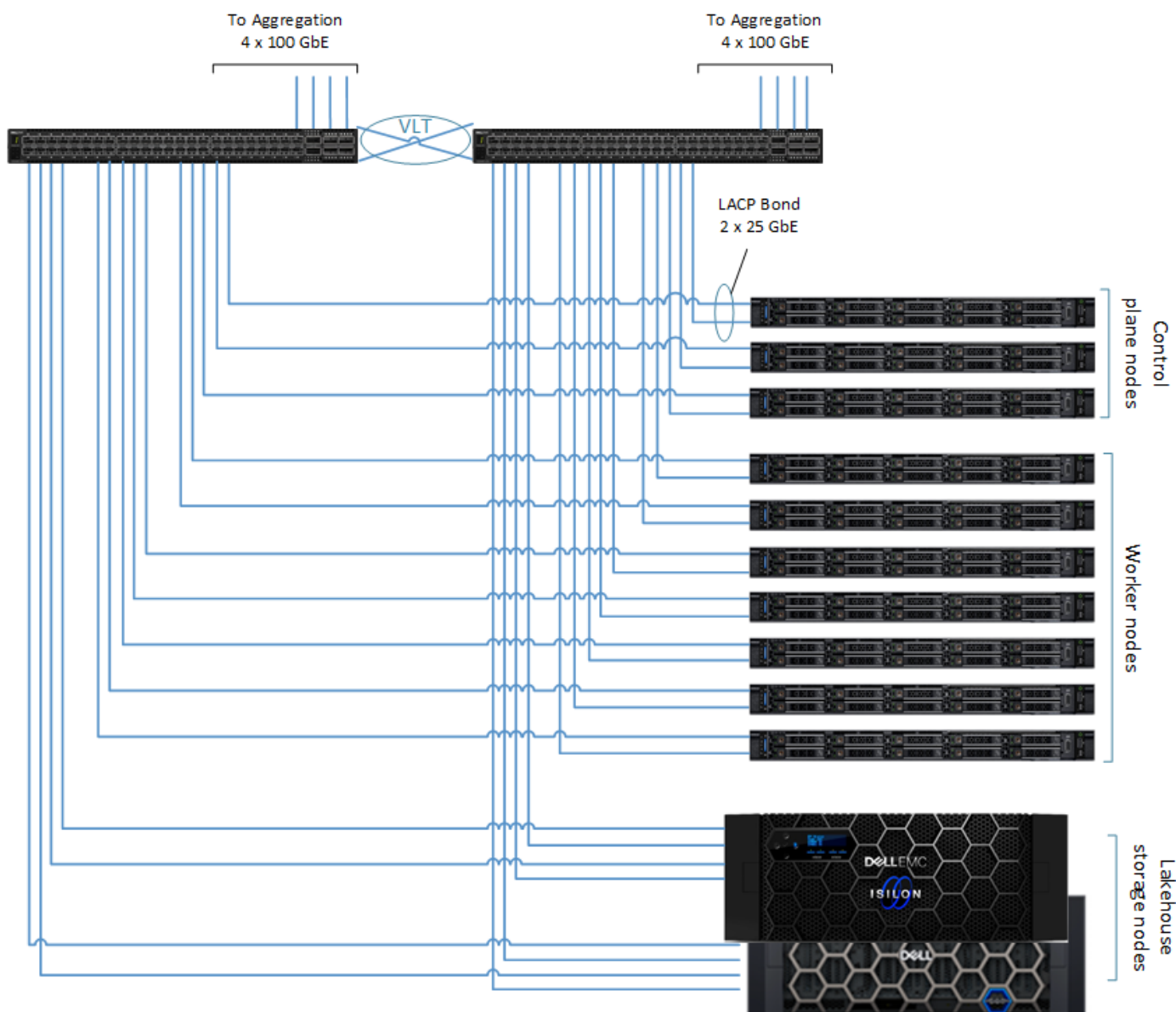


Figure 4. Cluster data network connections

Container platform implementation

Container platform overview

The container platform in this architecture is Robin Cloud Native Platform (Robin CNP), which is based on Kubernetes. The container platform abstracts the machine level details and host operating system dependencies, exposing them as a pool of compute, storage, and communications resources. It also provides capabilities to streamline the deployment, scaling, and life cycle management of analytics workloads. The container platform implementation supports the agility, flexibility, and scalability of the architecture while supporting diverse analytics workloads.

Robin roles

Robin CNP uses a role-based deployment model. Three roles can be assigned to nodes:

- Manager
- Compute
- Storage

These roles can be assigned to nodes depending on their configuration and intended usage. The roles are not mutually exclusive. Dell Technologies recommends the role assignments that are shown in [Recommended Robin role assignments](#).

Table 14. Recommended Robin role assignments

Robin CNP Role	Recommended node	Description
Manager	Control Plane 1	The Manager role provides the core control plane services that the cluster requires. The Manager role can be assigned to a maximum of three nodes. High availability deployments require three control plane nodes.
	Control plane 2	
	Control plane 3	
Compute	Worker 1	The Compute role indicates a node that is intended to host pods and their respective containers. Application deployments use memory and processor resources from these nodes.
	Worker 2	
	Worker 3	
	Worker 4	
	Worker <n>	
Storage	Worker 1	The Storage role indicates a node that is intended to provide storage services through Robin Cloud Native Storage. Volumes that are needed for deployed applications are created on devices from these nodes.
	Worker 2	
	Worker 3	
	Worker 4	
	Worker <n>	

Orchestration

The platform provides application orchestration capabilities to streamline the deployment and management of analytics workloads. The Robin Application Workflow Manager supports end-to-end automation and can deploy entire application pipelines seamlessly. Applications on the platform should be containerized and are run under Kubernetes as containers inside pods.

Applications are deployed using the Robin application bundle framework. Using this deployment framework, an application bundle contains all the resources necessary to deploy an application. Applications that are deployed with application bundles are known to the platform and integrated into the advanced application orchestration and management capabilities.

When an application is launched, the Robin scheduler provisions the required compute, storage, and network resources and then starts the application pods. Since the platform is aware of the entire application, the scheduler can use advanced placement techniques, including data locality, data affinity, and infrastructure awareness.

Once the application starts it can be managed using the platform life cycle management capabilities, including: resource scaling, back-ups, and snapshots.

Helm charts can also be used to deploy applications. After a Helm release is registered with the platform, the life cycle management capabilities are available to applications using Robin storage.

For more detail on application bundles, see [Workload design](#).

Tenants and resource management

The platform uses the concept of a tenant to organize users into groups. Tenants can be organized according to functional or business requirements. System resources are assigned to each tenant, and applications are installed and run within the limits of the tenant resources. Resource pools are used to define the available compute and storage resources for a tenant. IP pools are used to define the available network resources.

Virtualization

The platform supports Linux Kernel-based Virtual Machines (KVM) alongside containerized applications. This capability supports running workloads that have not been containerized.

Application bundles can be created that specify KVM parameters and the corresponding qcow2 image that the virtual machine uses. Once virtual machines launch, they are treated like any other application on the platform, with the full range of life cycle management operations available.

Compute

When an application is launched, the Robin scheduler provisions the application pods on nodes with the Compute role that have the appropriate processor and memory resources. The tenant resource pool and current utilization define the available resources. Compute resources are allocated at the container level. The application bundle specifies the amount of memory that is required and the processor core count that is wanted.

GPU accelerators

An application can also request GPU resources. In this case, the GPU type is in the application bundle. At application launch time, GPU resource requirements are in the scheduling decision. Entire GPUs and multiple-instance GPU resources can be requested. Multiple GPUs are supported, and nodes in the cluster can have varying GPU models and quantities. GPU resources are allocated and dedicated to the container for the life of the container.

The platform supports NVIDIA GPUs through the NVIDIA GPU driver that is installed at the operating system level, and the NVIDIA GPU operator for Kubernetes. To use GPU acceleration, application images must also include support libraries that implement GPU support at the user level. For example, NVIDIA CUDA and the NVIDIA RAPIDS Accelerator for Apache Spark.

Storage

The platform supports several different types of storage for ephemeral and persistent storage, and for data lakehouse storage.

Robin Cloud Native Storage

Robin Cloud Native Storage (Robin CNS) is a scalable, high-performance software defined storage system that is Kubernetes Container Storage Interface (CSI) compatible. It provides data resiliency through replication, and supports encryption, compression, and thin-provisioning. Robin CNS is application and infrastructure aware, allowing it to support data locality, snapshots, and backups for applications running on the platform. Robin CNS is hosted on nodes with the storage role. At installation time, it discovers available disks and pools them to provide storage to applications.

Robin CNS uses volumes as the unit of allocation. These volumes are analogous to Kubernetes **PersistentVolume** objects. The replication factor, encryption, and compression properties are specified when a volume is created. The storage class can also specify the preferred media type, either HDD, SSD, or NVMe.

Robin storage is exposed to applications through the Kubernetes CSI using a Kubernetes **StorageClass** object. Robin CNS ships with three predefined **StorageClasses**:

- `robin`—The default **StorageClass** that has no advanced features and can be used for standard ReadWriteOnly (RWO) and ReadWriteMany (RWX) volumes.
- `robin-repl-3`—A **StorageClass** that uses three replicas for data reliability.
- `robin-immediate`—A **StorageClass** that creates a volume when a Persistent Volume Claim (PVC) is created. It does not wait for the first consumer of that volume.

File collections

The platform uses file collections to store application bundles, images, and collected logs. File collections are exposed through the Robin file server, which runs on the control plane nodes. Storage for each file collection is allocated from Robin managed storage as a volume.

Kubernetes CSI storage

The platform can use any storage system that has a CSI compatible driver. This storage can be used for ephemeral and persistent storage exactly like Robin Cloud Native Storage. However, the backup, snapshot, and migration capabilities of the platform are only available for applications using Robin CNS.

Lakehouse storage

The platform has two types of lakehouse storage available, using PowerScale with the HDFS protocol, or ECS with the S3 protocol. These storage systems are managed and scaled independently from the core platform, providing a decoupled storage and compute architecture.

Applications can connect to either or both lakehouse storage options directly over the network from the application level code. The Apache Hadoop client libraries (`hadoop-hdfs-client`) provide the `hdfs://` protocol, while the Apache Hadoop AWS libraries (`hadoop-aws`) provide the `s3a://` protocol. Depending on the application and its implementation, the images and application bundle may require that the necessary libraries be included. The application must handle authentication to any external storage.

Management and administration

All system administration and management are handled through a single web interface. Operations can also be performed and automated through the command line.

The Robin Cloud Native Platform offers integrated multitenancy to enable a shared cloud experience, with physical and logical separation between tenants. It includes integrated support for role-based access control to manage end-user credentials across each tenant. It also includes support for chargeback features to enable multiple departments and use cases.

The platform also includes highly efficient monitoring and metrics collection of hardware components, pods, and applications. These features provide holistic observability of all platform activity in a single location.

Network

The platform uses software defined networking (SDN) to handle communications for the cluster. SDN includes container-to-container connections, pod-to-pod connections, ingress to pod connections, and pod to external services like PowerScale and ECS. These network services are provided through the Kubernetes Container Network Interface (CNI) and supported by CNI compatible network plugins.

Each physical node in the cluster is assigned an IP address for its connection to the cluster data network. Each connection uses a pair of physical network ports that are bonded with IEEE 802.3ad dynamic link aggregation. This configuration provides both load balancing across physical links, and fault tolerance if a link fails. All networking above this layer is software-defined and uses IP addresses that are private to the cluster.

When applications running on the cluster request IP addresses, the addresses are allocated from the tenants IP pool. The IP pool specifies both the range of addresses available and the CNI driver to use. The platform supports three CNI driver options – Calico, Open vSwitch (OVS), and SR-IOV. Dell Technologies recommends using the OVS driver for most use cases, since it provides the best support for inbound access to applications from outside the cluster.

Inbound network access to applications is handled through a Kubernetes **NodePort** service. The **NodePort** configuration is specified in the application bundle and includes the external and internal port number mapping. Outbound network access from applications is handled through IP routing that is based on the destination.

Workload design

Overview

The platform orchestration capabilities use the Robin Application Workflow Manager. Applications are deployed using the Robin application bundle framework. Using this deployment framework, an application bundle specifies all the resources necessary to deploy an application. When an application bundle is installed, it appears in the application catalog, and platform users can deploy it. The figure below shows the main Robin application catalog screen.

NOTE: The application bundles shown are examples of applications that the Robin Cloud Native Platform can deploy. They are not standard bundles available from Robin.io.

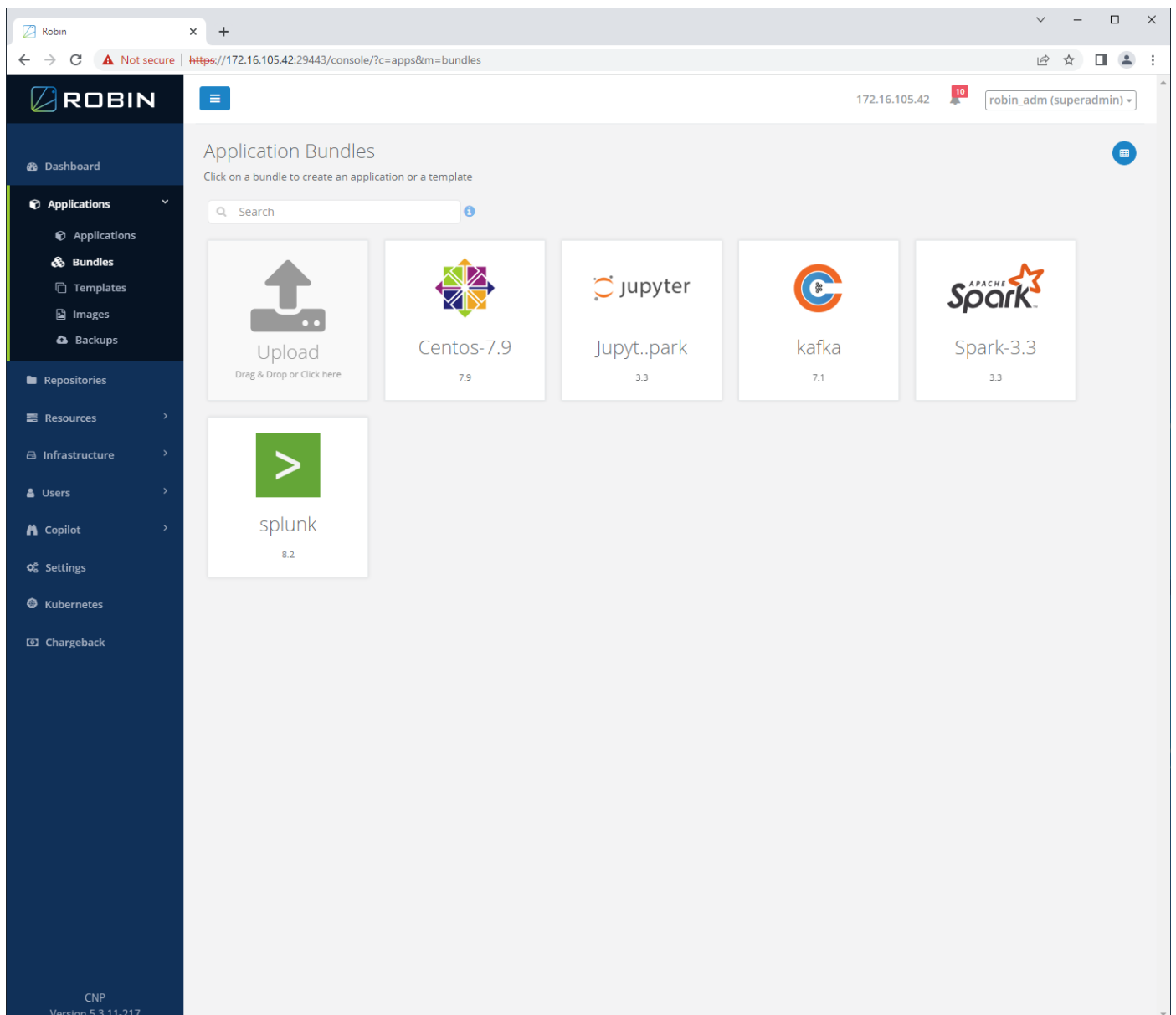


Figure 5. Application bundles catalog

When a bundle is deployed, the Robin scheduler provisions the required compute, storage, accelerator, and network resources and then starts the application pods. Since the platform is aware of the entire application, the scheduler can use advanced placement techniques, including data locality, data affinity, and infrastructure awareness. During deployment, users can specify deployment and runtime preferences in the launch screen.

The primary component of an application bundle is a manifest file which acts as a blueprint for the application. It describes the application components, any dependencies between services, resource requirements, affinity and anti-affinity rules, and custom actions required for application management. The bundle also contains icons, any supporting scripts, and specifies the necessary container images for the application. Bundles are packaged as compressed `tar` archives.

Detailed information about the creation and management of applications bundles is available in the [Robin Hyperconverged Kubernetes Bundle Building Guide](#).

Dell Technologies implemented and deployed workload bundles as part of the solution verification process. The following sections describe the implementation of two of these bundles.

Spark bundle

The Spark application bundle deploys an Apache Spark 3.3.0 stand-alone cluster. The figure below shows the application creation screen for the Spark bundle.

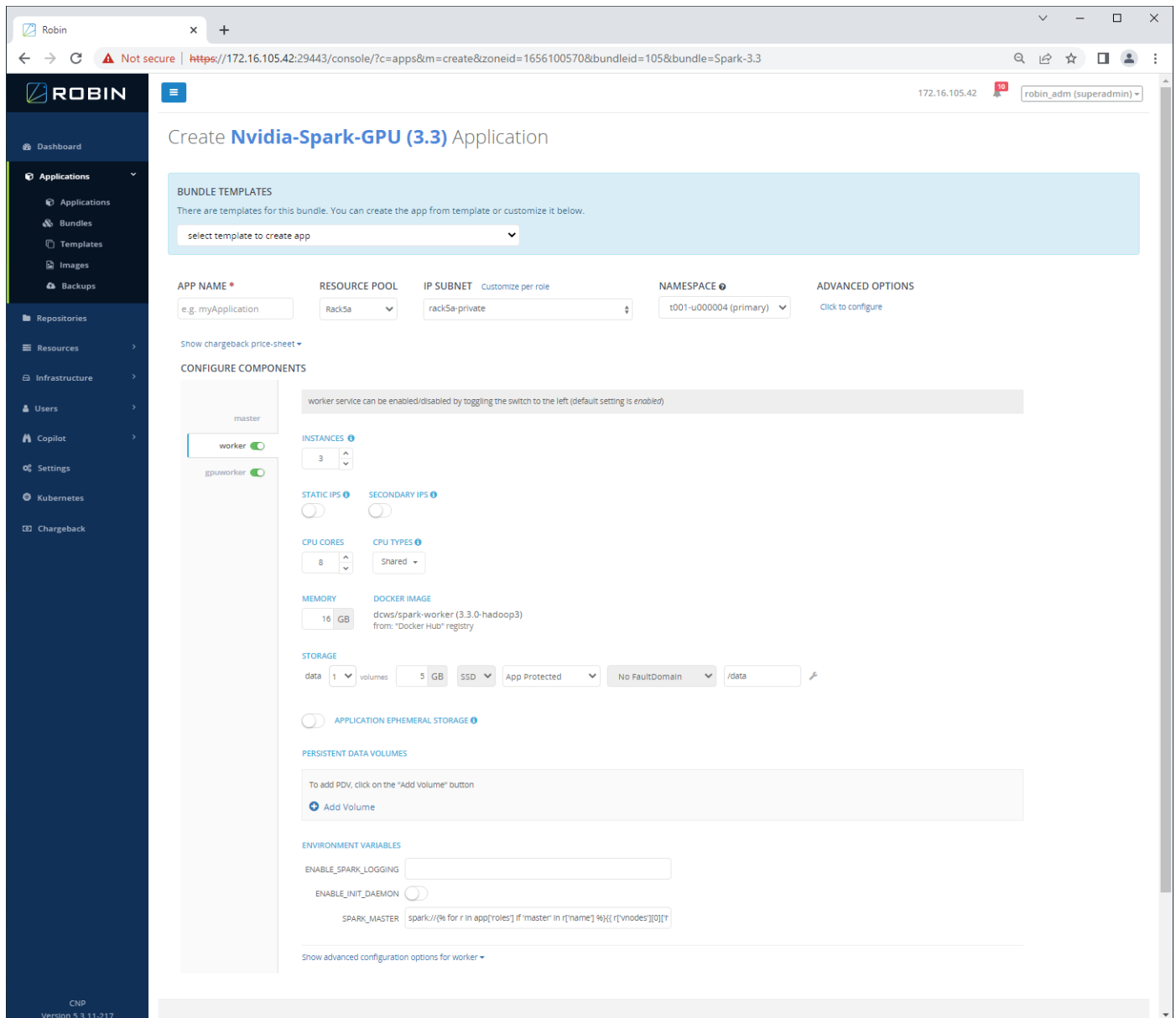


Figure 6. Spark 3.3 application creation screen

This bundle includes:

- Apache Spark 3.3.0

- Hadoop client libraries (for HDFS access)
- Hadoop AWS libraries (for S3 access)
- Delta Lake 1.2.1 libraries
- NVIDIA CUDA 2.1 (for GPU support)
- NVIDIA RAPIDS Accelerator for Apache Spark (for Spark GPU support)

At application creation time, the numbers of both Spark workers and Spark GPU workers can be specified. The bundle also allows resources to be specified, including number of cores and memory per worker.

Kafka bundle

The Kafka application bundle deploys a Confluent Kafka 7.1.0 cluster. The figure below shows the application creation screen for the Kafka bundle.

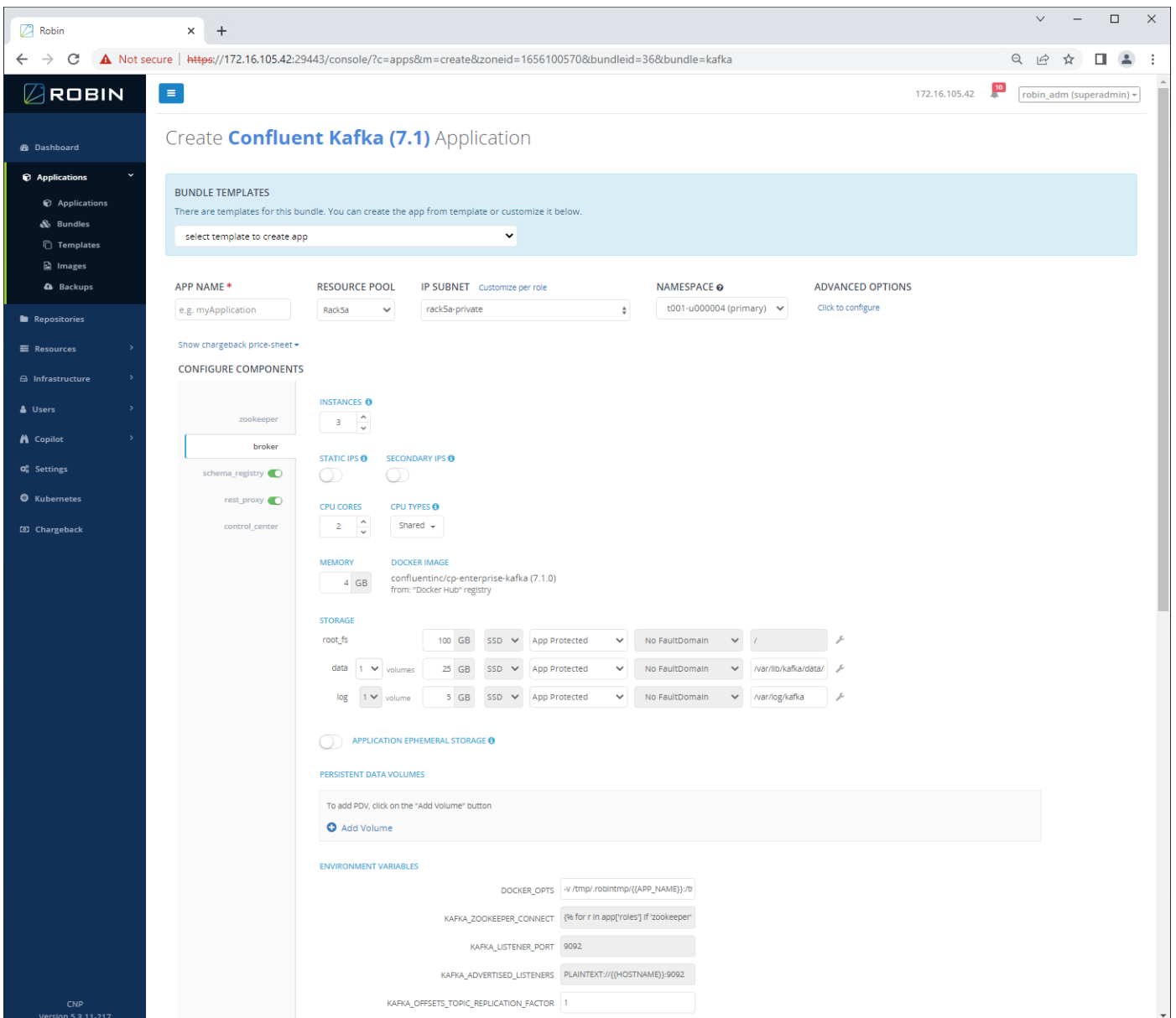


Figure 7. Kafka 7.1 application creation screen

This bundle includes the standard Kafka components, including:

- Kafka Broker
- Kafka Schema Registry

- Kafka REST proxy
- Confluent Control Center
- ZooKeeper (for runtime support)

At application creation time, the number of brokers can be specified. The bundle also allows resources to be specified, including number of cores and memory per broker. As part of the broker configuration, the location and amount of storage for the Kafka queues can also be specified.

Software infrastructure validation

Topics:

- [Overview](#)
- [Software components](#)
- [Component validation](#)
- [Findings](#)
- [Summary](#)

Overview

The following sections detail the efforts that Dell Technologies undertook to ensure that the Dell Validated Design for Analytics — Data Lakehouse integrated hardware and software created a valid working design. Exhaustive testing of each hardware and software component was not performed. Dell Validated Design for Analytics — Data Lakehouse verified that all the components work together as designed.

The following sections feature some of the tests that were performed and outline the goals and reasons for testing.

Software components

The software components and versions that are validated for the Dell Validated Design for Analytics — Data Lakehouse are listed in the tables below. The validated components may not precisely match the Dell Technologies-recommended configurations. For more information, email ai.assist@dell.com, or contact your Dell Technologies sales representative.

Table 15. Validated PowerEdge R650 firmware

Component	Version
PowerEdge BIOS	1.4.4
iDRAC	5.10.10.00
iDRAC service module embedded package	4.2.0.0, A00
Intel E810-XXV	20.5.13
PERC H755 SAS front	52.14.0-3708
PCIe SSD in slot 0, bay 1	1.2.3
BOSS-S2	2.5.13.4008
Backplane 1	3.57
Broadcom GbE BCM5720	22.00.6
Disk 0 on AHCI controller in slot 6	J004
Trusted Platform Module (TPM)	7.2.2.0
Dell operating system driver pack	21.10.02, A00
Dell 64-bit UEFI diagnostics	4301A66
System Complex Programmable Logic Device (CPLD)	1.0.5
Lifecycle controller	5.10.10.00

Table 16. Validated PowerEdge R750 firmware

Component	Version
PowerEdge BIOS	1.5.5
iDRAC	5.10.10.00
iDRAC service module embedded package	4.2.0.0, A00
PCIe SSD in slot 0, bay 1	VDV1DP25
BOSS-S2	2.5.13.4008
Backplane 1	3.57
NVIDIA Ampere A30 Tensor Core GPU	92.00.25.00.08
Broadcom GbE BCM5720	22.00.6
Mellanox NIC 1	26.30.10.04
Mellanox NIC 2	26.30.10.04
Intel Ethernet converged network adapter X710	20.5.13
Intel Ethernet 10 G 4P X710 OCP	20.5.13
Disk 0 on AHCI controller in slot 6	J004
Dell operating system driver pack	21.10.02, A00
Dell 64-bit UEFI diagnostics	4301A68
System CPLD	0.4.2
Lifecycle controller	5.10.10.00

Table 17. Validated software components

Category	Component	Version
Server operating system	Red Hat Enterprise Linux Server	8.4
	Linux kernel	4.18.0-305.el8.x86_64
Server file system	XFS	N/A
Switch operating system	Dell SmartFabric OS10	N/A
ECS software	ECS	3.7
PowerScale operating system	OneFS	9.2
Java Virtual Machine	Open JDK	8 (8u232-b09)
GPU computing platform	NVIDIA CUDA	2.1
GPU driver	NVIDIA Ampere A30 Tensor Core	470.82.01
Automation platform	Robin Cloud Native Platform	5.3.11-217
Container engine	Docker	20.10.8
Container platform	Enterprise Kubernetes	1.21.5
Data analytics engine	Apache Spark	3.3.0
Data lakehouse	Delta Lake	1.2.1
Data streaming distribution	Apache Kafka	7.1.0
Interactive development environment	Jupyter	See Validated Jupyter components
Programming language	Python	3.9

Table 18. Validated Jupyter components

Component	Version
IPython	8.0.1
ipykernel	6.8.0
ipywidgets	7.6.5
jupyter_client	7.1.2
jupyter_core	4.9.1
jupyter_server	1.13.5
jupyterlab	3.2.9
nbclient	0.5.10
nbconvert	6.4.1
nbformat	5.1.3
notebook	6.4.8
traitlets	5.1.1

Component validation

The testing objective was to run sample checks to validate the functionality of different components and services that Dell Technologies installed as parts of the Dell Validated Design for Analytics — Data Lakehouse.

GPU validation

Dell Technologies tested a pair of NVIDIA A100 GPUs as whole, nonpartitioned devices with driver version 470.82.01. The publicly available GPU operator was installed using the Helm chart after the Robin cluster had been provisioned. Doing so enabled discovery of the device by Kubernetes. In order to make the GPUs available within the Robin environment, Dell Technologies performed a `robin host probe --rediscover <nodename>` command. Doing so made GPUs available for deploying application bundles that require GPU resources.

The Dell Validated Design for Analytics — Data Lakehouse uses a Spark 3.3.0 bundle with an optional worker node that requires one or more GPUs. Enabling this worker when instantiating the application requests the GPU and causes the worker to register with the Spark master, indicating the extra resource.

Running jobs against the deployed Spark application stack spreads the work across all nodes. If the workload requests GPU resources for acceleration, the job is scheduled only on workers that have registered with a GPU. This job scheduling was verified by running tasks that use Spark SQL RAPIDS calls.

```
spark-shell --master spark://spark-master-01:7077 \  
  --conf spark.executor.extraClassPath=/opt/spark/jars/cudf-22.06.0.jar:/opt/spark/  
jars/rapids-4-spark_2.12-22.06.0.jar \  
  --conf spark.driver.extraClassPath=/opt/spark/jars/cudf-22.06.0.jar:/opt/spark/jars/  
rapids-4-spark_2.12-22.06.0.jar \  
  --conf spark.rapids.sql.concurrentGpuTasks=1 \  
  --driver-memory 2G \  
  --conf spark.executor.memory=4G \  
  --conf spark.executor.cores=4 \  
  --conf spark.task.cpus=1 \  
  --conf spark.executor.resource.gpu.amount=1 \  
  --conf spark.task.resource.gpu.amount=0.25 \  
  --conf spark.rapids.memory.pinnedPool.size=2G \  
  --conf spark.locality.wait=0s \  

```



```
--conf spark.sql.files.maxPartitionBytes=512m \
--conf spark.plugins=com.nvidia.spark.SQLPlugin
```

```
scala> val df = sc.makeRDD(1 to 10000000, 6).toDF
df: org.apache.spark.sql.DataFrame = [value: int]
```

```
scala> val df2 = sc.makeRDD(1 to 10000000, 6).toDF
df2: org.apache.spark.sql.DataFrame = [value: int]
```

```
scala> df.select($"value" as "a").join(df2.select($"value" as "b"), $"a" === $"b").count
res0: Long = 10000000
```

Checking the physical worker with GPU resources by running the `/usr/bin/nvidia-smi` tool shows the GPU that is used for processing.

```
[root@worker6 ~]# nvidia-smi
```

```
Mon Sep 19 14:46:55 2022
+-----+
| NVIDIA-SMI 470.82.01      Driver Version: 470.82.01      CUDA Version: 11.4      |
+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan   Temp   Perf   Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                               |                  |           Compute M. |
|                               |                  |           MIG M.     |
+-----+-----+-----+-----+-----+-----+
|    0   NVIDIA A100-PCI...   Off          | 00000000:B1:00.0 Off  |           0         |
| N/A    28C    P0      31W / 250W |      0MiB / 40536MiB |           0%      Default |
|                               |                  |           Disabled  |
+-----+-----+-----+-----+-----+-----+
|    1   NVIDIA A100-PCI...   Off          | 00000000:CA:00.0 Off  |           0         |
| N/A    30C    P0      36W / 250W |    39902MiB / 40536MiB |           0%      Default |
|                               |                  |           Disabled  |
+-----+-----+-----+-----+-----+-----+
| Processes:                                                       GPU Memory |
|  GPU   GI    CI          PID    Type   Process name                        Usage    |
|-----+-----+-----+-----+-----+-----+
|    1   N/A  N/A       1310307      C   ....0-openjdk-amd64/bin/java       39895MiB |
+-----+-----+-----+-----+-----+-----+

```

Platform validation

Dell Technologies tested the Delta Lake data lakehouse that was deployed on the Robin Cloud Native Platform (Robin CNP). Dell Technologies also tested integrating Dell PowerScale and Dell ECS storage arrays with Delta Lake. This platform enables the running of various data analytics applications such as Spark, Kafka, and Splunk on Robin CNP with different storage systems.

Dell Technologies used Delta Lake 1.2.1 with Spark 3.3.0 to validate all data lakehouse functionality. All the features of Delta Lake with Spark were validated, including:

- Atomicity, consistency, isolation, and durability (ACID) transactions
- Scalable metadata handling
- Updates, deletes, and merges
- Audit and time travel
- Schema enforcement and evolution
- Unified batch and streaming

Spark validation

These tests validate running a Spark bundle application on the Robin Cloud Native Platform using Delta Lake as its data lakehouse storage. Dell Technologies used the Apache Spark 3.3.0 prebuilt Hadoop version with Delta Lake 1.2.1.

Dell PowerScale and Dell ECS acted as the Delta Lake data lakehouse storage in order to access the data for read and write operations. Dell Technologies also validated Spark with an NVIDIA GPU on this platform.

Data can be written to or read from Spark using different API protocols, such as:

- Hadoop (hdfs://<IP>)
- S3 object storage protocol (s3a://<bucket Name>/)

i **NOTE:** Accessing the S3 storage was accomplished by using the `hadoop-aws:3.2.3` library for accessing the data from ECS through the S3 API protocol.

PowerScale

About this task

Spark with PowerScale testing consisted of the following steps:

Steps

1. Running Delta Lake with Spark required installing an additional package that is available from Delta.io:

```
/opt/spark/bin/pyspark --packages \
  io.delta:delta-core_2.12:1.2.1,org.apache.hadoop:hadoop-aws:3.2.3 --conf
"spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension" --conf
"spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog"
```

2. Data was read from Spark using HDFS:

```
>>> df1 = spark.read.format("delta").load('hdfs://172.16.105.201/data/hdfs-test-
table')
>>> df1.show(1)
```

The system returned a message showing only the top table row, similar to the following example:

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
|msisdn_1|msisdn_2|cell_1|cell_2|      operator_1|
  operator_2|duration|      timestamp|termination_status_1|termination_status_2|
  value_1|      value_2|type|transit_type|  tac_1|  tac_2|
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
|      1|      1|  259|  259|
TelecomMobile 1|TelecomMobile 1|      694|2015-01-02T03:31:58|      1|
  1|6.321016642869488|6.7185260289517865|CALL|      INT|1203900|35203107|
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
```

3. The data was overwritten and reread from Spark using HDFS:

```
>>> df2 = df1.write.format("delta").mode("overwrite").save("hdfs://172.16.105.201/
data/hdfs-test-table1")
>>> df3 = spark.read.format("delta").load('hdfs://172.16.105.201/data/hdfs-test-
table1')
>>> df3.show(1)
```

The system returned a message showing only the top table row, similar to the following example:

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
|msisdn_1|msisdn_2|cell_1|cell_2|      operator_1|
  operator_2|duration|      timestamp|termination_status_1|termination_status_2|
  value_1|      value_2|type|transit_type|  tac_1|  tac_2|
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
```

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      1|      1|   259|   259|           694|2015-01-02T03:31:58|           1|
TelecomMobile 1|TelecomMobile 1|           694|2015-01-02T03:31:58|           1|
|6.321016642869488|6.7185260289517865|CALL|           INT|1203900|35203107|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Results

Dell Technologies proved the validity of PowerScale reads and writes from Delta Lake using Spark.

ECS

About this task

Spark with ECS testing consisted of the following steps:

Steps

1. Running Delta Lake with Spark required installing an additional package that is available from Delta.io:

```

/opt/spark/bin/spark-shell --packages \
  io.delta:delta-core_2.12:1.2.1,org.apache.hadoop:hadoop-aws:3.2.3 --conf \
  "spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension" --conf \
  "spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog"

```

2. Data was read from Spark using S3:

```

sc.hadoopConfiguration.set("fs.s3a.access.key", "access key id")
sc.hadoopConfiguration.set("fs.s3a.secret.key", "secret key")
sc.hadoopConfiguration.set("fs.s3a.endpoint", "http://<s3api address: port>")
sc.hadoopConfiguration.set("fs.s3a.aws.credentials.provider", \
  "org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider") \
  spark.range(5).repartition(1).write.format("delta").save("s3a://<bucketname>/")

```

Results

Dell Technologies proved the validity of ECS reads and writes from Delta Lake using Spark.

Kafka validation

About this task

This test validates deploying a Kafka application bundle on Robin Cloud Native Platform with Delta Lake. Dell Technologies has integrated Kafka with Spark to stream the input data from Kafka, perform data transformation on Spark, and store the result on Delta Lake. Kafka testing consisted of the following steps:

Steps

1. Start a shell session on the Kafka broker:

```
kubectl exec -it kafka7-broker-01 /bin/bash
```

2. Stream data from a comma-separated values (CSV) file:

```
kubectl exec -it kafka7-broker-01 -- /bin/bash -c "/usr/bin/Kafka-console-producer --
broker-list kafka7-broker-01:9092 --topic test < /home/appuser/TestDataOne.csv"
```

3. Run the following command from the Spark container:

```
/spark/bin/pyspark --packages \
  io.delta:delta-core_2.12:1.2.1,org.apache.hadoop:hadoop-aws:3.2.3,\
  org.apache.spark:spark-sql-kafka-0-10_2.12:3.2.0 --conf
"spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension" --conf
"spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog"
```

The system returned a message similar to the following example:

```
spark container :

import sys
import os
import pyspark.sql.functions as F
from pyspark.sql import SparkSession
from pyspark.context import SparkContext
from pyspark.sql.functions import *
from pyspark.sql.functions import *
from delta.tables import *
hadoopConf = spark._jsc.hadoopConfiguration()
hadoopConf.set("fs.s3a.access.key", "<access_key>")
hadoopConf.set("fs.s3a.secret.key", "<secret_key>")
hadoopConf.set("fs.s3a.endpoint", "http://<s3_api_address>")
hadoopConf.set("fs.s3a.impl", "org.apache.hadoop.fs.s3a.S3AFileSystem")
hadoopConf.set("fs.s3a.aws.credentials.provider",
"org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider")

df = (spark\
  .readStream \
  .format("kafka") \
  .option("kafka.bootstrap.servers", "kafka7-broker-01:9092") \
  .option("subscribe", "kafkaspark") \
  .option("startingOffsets", "latest") \
  .load())

df.writeStream \
  .format("delta") \
  .outputMode("append") \
  .option("mergeSchema", "true") \
  .option("checkpointLocation", "/tmp/kafaktest") \
  .start("s3a://sparkdelta/kafkaspark2/")

df.selectExpr("CAST(value AS STRING)").show(20)
```

Results

Dell Technologies proved the basic functionality of Kafka for streaming the data, integrating Kafka with Spark for data processing, and loading the final data into Delta Lake.

High availability validation

This test validates deploying Robin Cloud Native Platform (Robin CNP) in high availability (HA) mode. For a highly available Kubernetes cluster, Dell Technologies recommends configuring a Kubernetes cluster with a minimum of three control plane nodes.

Robin uses the `keepalived` and `HAProxy` services to provide high availability to the Kubernetes API server. The `keepalived` service is responsible for managing a Virtual IP address (VIP) where all requests to the Kubernetes API server are sent. The `HAProxy` service is responsible for redirecting API server requests to instances of the API server running on each of the control plane nodes.

Dell Technologies used the following command to deploying Robin Cloud Native Platform in HA mode:

```
./robin-install-k8s-el8_5.3.11-217.sh master --vip=x.x.x.x \
  --vip-type=robin --vrid=1 --username=username --password=password \
  --robin-image-archive=robinbinimg-5.3.11-217.tar.gz \
  --k8s-image-archive=k8s-images-5.3.11-217.tar.gz
```

Dell Technologies has proved the basic Robin cluster HA functionality by manually simulating failover of a Kubernetes control plane node. When the primary manager node fails or becomes unhealthy, one of the secondary manager nodes takes over as primary. Data integrity is maintained for key metadata that is related to storage management for the Robin cluster and for deployed applications. Also, a mechanism is provided for recovering from hard failures.

Findings

All the tests ran successfully. They show that it is possible to integrate powerful software solutions with Dell hardware to create value greater than the individual parts.

Summary

The Dell Validated Design for Analytics — Data Lakehouse is a valuable solution for any group seeking to analyze large amounts of data. The ease and flexibility of integrating various vendor products into the solution bring great value to the enterprise.

Sizing the solution

Topics:

- [Sizing and scaling overview](#)
- [Sizing guidelines](#)
- [Scaling guidelines](#)

Sizing and scaling overview

Cluster sizing and scaling are two different but related considerations. Sizing is concerned with ensuring the cluster meets the workload requirements for storage and processing throughput. Scaling is concerned with growth of the cluster over time as capacity needs increase.

The architecture is a parallel scale-out system with decoupled compute and storage. Some sizing requirements can be addressed through scaling while others must be addressed through node level sizing.

Sizing and scaling of a cluster are complex topics that require knowledge of the workloads. This section highlights the main considerations that are involved but does not provide detailed recommendations for workload sizing. Design guides for specific workloads running on the platform include workload specific sizing guidance. Your Dell Technologies or authorized partner sales representative can help with detailed sizing calculations.

There are many parameters that are involved in cluster sizing. The primary parameters are:

- Storage capacity** Storage capacity is usually the first parameter that is used to size a cluster. Calculating storage capacity is important and straightforward. However, storage capacity should be calculated while taking the other sizing parameters into account to maintain a balance between storage and processing capability. The use of decoupled storage and compute simplifies this consideration since the balance can be adjusted at any time during the life of the cluster.
- Data volumes and growth rates** Data volume and growth rates each have multiple impacts on cluster sizing. Lakehouse storage capacity should account for growth due to data ingest and growth of ingest volumes over time. Data ingest also impacts network utilization. Since the lakehouse storage is external to the cluster nodes, network bandwidth is required to access it. The processing throughput requirements must be considered as well as the data size.
- Memory and processor capacity** Memory and processor requirements for jobs running on the cluster must be considered when sizing. Memory and processor capacity increases as nodes are added to the cluster. You can create heterogenous node configurations for workloads with specific requirements, which may be necessary.
- Service level agreements** Production cluster sizing must meet any performance requirements that SLAs specify. Critical path jobs that must meet a specific execution time or throughput may require adjusting the cluster sizing and balance between compute and storage accordingly. Overall cluster throughput is as important as storage capacity, and often influences the number of nodes independent of the required storage capacity.

Sizing guidelines

Cluster sizing

[Example cluster configurations](#) lists some cluster level starting points for possible deployments.

Table 19. Example cluster configurations

Configuration	Proof of concept (POC)	Pilot	Small and medium business (SMB)	Enterprise
Control plane nodes	0	3	3	3
Worker nodes	3	4	10	20
Available memory	597 GB	828 GB	2070 GB	4140 GB
Available physical cores	144	208	520	1040
Available server storage	34 TB	46 TB	115 TB	230 TB

NOTE: The available resources in [Example cluster configurations](#) are approximate. These resources are available to user workloads and exclude resources that the core platform services consume.

Proof of concept

The proof of concept (POC) configuration is a minimal configuration for basic evaluation. In this scenario three worker nodes are used to host the control plane, runtime services, and workloads. The worker nodes host the Robin Manager, Compute, and Storage roles in addition to any workloads. This configuration provides limited resources for workloads but is adequate for basic functionality evaluation. More worker nodes can be added to this configuration. For anything larger than five workers Dell Technologies recommends starting with, or upgrading to, the pilot configuration.

Converting a POC cluster to a production grade cluster requires redeploying the software, although the worker node hardware can be reused.

Pilot

The pilot configuration is a minimal production grade configuration. In this scenario three control plane nodes host the Robin Manager roles. Four worker nodes host the Compute and Storage roles along with any workloads. This configuration provides isolation between control plane and runtime functions. Dell Technologies recommends it for preproduction or development and test usage.

The pilot configuration can be scaled up by adding additional worker nodes without redeploying the existing nodes.

Small and medium business

The small and medium business (SMB) configuration is a small production grade configuration. Three control plane nodes host the Robin Manager roles, and ten worker nodes are available for workloads. This configuration provides enough resources to support one or two teams running analytics workloads.

The SMB configuration can be scaled up by adding additional worker nodes without redeploying the existing nodes.

Enterprise

The enterprise configuration is a large production grade configuration. Three control plane nodes host the Robin Manager roles, and 20 worker nodes are available for workloads. This configuration provides substantial resources for running analytics workloads supporting multiple teams.

Lakehouse storage sizing

The amount of expected lakehouse data primarily determines lakehouse storage sizing. This aspect of the sizing is independent of the compute cluster sizing.

The available network bandwidth between the compute and storage clusters must also be considered. Bandwidth on the storage and compute clusters scales in direct proportion to the number of nodes. However, the dense storage capacity possible with ECS and PowerScale can result in a large storage capacity without enough bandwidth to support the lakehouse data transfer

requirements. An analysis of workload data transfer requirements is necessary to correctly size the storage for both capacity and bandwidth.

The architecture is not limited to a single type of lakehouse storage. PowerScale can be used with the HDFS protocol, or ECS can be used with the S3 protocol. Any workload can reference either or both of these storage types. It is also possible to use multiple external PowerScale and ECS storage systems.

The network architecture allows both compute and storage clusters to use the same fabric. This configuration enables the network bandwidth to scale as either storage or compute nodes are added. The bandwidth available to the external storage systems should also be considered when referencing external storage that is not connected to the core cluster data network.

Control plane sizing

The control plane node sizing that Dell Technologies recommends in [Lakehouse control plane node](#) is adequate for all production clusters and should not be changed. The Robin management services must be deployed on three individual nodes.

The control plane services also consume a small quantity of worker node resources. Sizing in this design guide allocates 49 GB and four cores to these services. Approximately 320 GB of storage for the control plane is also required on worker nodes. This space is allocated from the boot drives and does not impact available server storage for user workloads.

Worker node sizing

All the available cluster resources across all worker nodes are pooled and allocated on demand. This configuration provides an abstraction where workloads can be mapped to available resources independent of the physical node used. Accelerators are considered a resource, and any workload pod that requires accelerator resources must run on a node with an available accelerator.

The recommended worker node sizes in this design are based on general-purpose usage. These worker nodes can support various analytics workloads without modification. However, there are scenarios where it is appropriate to change the configurations to match the intended workloads.

Heterogenous node configurations are possible. A cluster can include nodes with differing memory, compute, and storage sizes. The resources from all these nodes are added to the overall resource pool.

From a resource point of view, there is little difference between many small nodes and a few large nodes. If the nodes have enough resources to handle the largest expected pod resource request, the difference between nodes is transparent. However, three additional considerations are involved in this tradeoff; network bandwidth, fault zones, and operational overhead.

Available network bandwidth is proportional to the number of nodes. A few large nodes have less bandwidth than many small nodes, even if the aggregate memory and compute resources are the same. The bandwidth requirements for workloads should be factored into cluster and node sizing.

Fault zones are important for overall reliability of the infrastructure. Although the cluster can continue running when a node fails, resources from that node are lost on failure. Large node configurations in a small cluster can have a substantial impact on available resources when the node fails, even if it is a temporary failure. Sizing should ensure the loss of a node only impacts a small proportion of the overall cluster capacity.

Operational overhead is another consideration for sizing. Every node entails some operational overhead in terms of maintenance and monitoring, so larger nodes can be more efficient. One larger node can also be more energy-efficient than several smaller nodes. Operational capacity should be part of the overall sizing effort.

For parallel, scale-out workloads like Apache Spark the resources are allocated based on availability at the cluster level and multiple workload pods that are launched. As a result, workload pods can run on any physical node that can meet the resource requirements. Depending on the Spark job workload, many small pods or a few large pods may be appropriate. The container platform runtime is flexible in this aspect. It is possible to deploy Spark clusters dynamically based on the job itself, instead of requiring a fixed Spark cluster optimized for many types of jobs.

Some workloads may have large memory requirements that cannot be achieved by scaling out. You may have to increase the memory size in some or all nodes to account for the largest expected memory allocation for that workload.

The platform can support up to 100 pods or worker nodes. Cluster and node sizing should aim for substantially fewer pods than this limit.

Workload sizing

The resources requirements for the intended cluster workloads must be factored into node and cluster sizing. Detailed sizing of workload requirements is complex. However, once workload requirements are known, the mapping into cluster requirements is straightforward. The flexibility of the platform also allows for ongoing adjustment and fine-tuning, so the sizing does not have to be exact.

Apache Spark is used here as an example of how workload sizing should be mapped to cluster requirements. [Example Spark instance requirements](#) summarizes the resource requirements for three sample Spark clusters.

Table 20. Example Spark instance requirements

Spark cluster resources	Small instance		Medium instance		Large instance	
	Worker	Cluster	Worker	Cluster	Worker	Cluster
Number of pods (Spark workers)	4		8		12	
Memory (GB)	8	32	16	128	32	384
Cores	4	16	6	48	8	96
Lakehouse storage (GB)	2	8	8	64	2	24
Ephemeral storage (GB)	8	32	16	128	32	384

In the table above, three clusters with varying resource requirements and scale have been included. The resources for each Spark worker have been specified, and the expected number of worker pods is included. Based on these requirements, the total resource requirements for each cluster are calculated.

For lakehouse storage, the amount of "net new" lakehouse storage that is required is used in the calculation. If the jobs are expected to process existing data, no additional storage is required. If the jobs generate data, significant storage may be required. In the large instance only 24 GB of lakehouse storage was estimated, while the medium instance requires 64 GB. The medium instance is expected to generate more data than the large instance even though it uses less compute resources.

Based on these calculations, the cluster level resources can be determined. For this example, the pilot cluster configuration can support four medium Spark clusters before it runs out of cores, or it can support two large instances.

Scaling guidelines

Scaling overview

During the lifetime of the system, it usually becomes necessary to scale the platform to support larger workloads or increase compute and storage capacity. The architecture is designed to scale at the compute, storage, control plane, and workload levels. The design incorporates network scaling as part the infrastructure scaling. Compute and storage can be scaled independently.

Compute scaling

Compute scaling is accomplished by adding additional nodes to the cluster. After physical installation and provisioning of the operating system, the Robin compute and storage services can be added from the Robin management interface. The nodes become parts of the cluster. The control plane and network are automatically scaled as new nodes are added.

After nodes are added, the new nodes must be added to the appropriate resource pools. When adding heterogenous nodes with specific configurations, it may be advantageous to add those nodes to a specific resource pool. Doing so enables support for granular allocation of those nodes.

For more details of the node provisioning process, see the Robin.io document, [Managing Nodes](#).

Storage scaling

Storage scaling is accomplished by adding or upgrading nodes in the ECS or PowerScale storage cluster, using the storage cluster management tools.

Network scaling

This architecture scales network bandwidth as compute or storage nodes are added. When scaling either compute or storage, you must consider any balance-related network bandwidth effects since the bandwidth increases as nodes are added. Substantial changes in expected data transfer volumes should be considered to ensure that the available bandwidth on the compute and storage clusters is aligned.

Summary

Topics:

- [Overview](#)
- [We value your feedback](#)

Overview

The Dell Validated Design for Analytics — Data Lakehouse has been developed to address the needs of organizations deploying advanced analytics. It incorporates the concepts of a lakehouse architecture along with a container platform using decoupled compute and storage.


This document provides design guidance for data analytics infrastructure managers and architects by describing a predesigned, validated, and scalable architecture for advanced analytics on Dell hardware infrastructure. Topics that were discussed include:

- The cluster architecture that was designed for this application, including cluster server and storage infrastructure and its role in the system
- The cluster physical and logical network designs
- Details of the PowerEdge server, PowerScale storage, ECS storage, and PowerSwitch networking configurations
- The recommended software infrastructure components that are used in the architecture, including the Robin Cloud Native Platform
- Examples of workload packaging, deployment, and validation, including Apache Spark and Apache Kafka
- Cluster sizing and scaling guidance

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](#).

Authors: Dell Technologies Integrated Solutions Engineering, Technical Marketing, and Information Design & Development teams

 **NOTE:** For links to additional documentation for this solution, see the [Dell Technologies Info Hub for Data Analytics](#).

This document may contain language from third-party content that is not under Dell's control and is not consistent with Dell's current guidelines for Dell's own content. When such third-party content is updated by the relevant third parties, this document will be revised accordingly.

References

Topics:

- [Dell Technologies documentation](#)
- [Delta Lake documentation](#)
- [Robin.io documentation](#)
- [Dell Technologies Customer Solution Centers](#)
- [Dell Technologies Info Hub](#)
- [More information](#)

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 sales representative.

Additional information can be obtained at the [Dell Technologies Info Hub for Data Analytics](#). If you need additional services or implementation help, contact your Dell Technologies sales representative.

Table 21. Dell Technologies documentation

Document type	Location
Server specification sheets	PowerEdge R650 Spec Sheet
	PowerEdge R650xs Spec Sheet
	PowerEdge R750 Spec Sheet
Storage specification sheets	ECS EX500 Spec Sheet
	ECS EXF900 Spec Sheet
	Isilon H5600 Spec Sheet
Switch specification sheets	PowerSwitch S3100 Series Spec Sheet
	PowerSwitch S5200-ON Series Spec Sheet
	PowerSwitch Z9264F-ON Spec Sheet
Server manuals	PowerEdge R650 Manuals and Documents
	PowerEdge R650xs Manuals and Documents
	PowerEdge R750 Manuals and Documents
Storage manuals	ECS EX500 Manuals and Documents
	ECS EXF900 Manuals and Documents
	Isilon H5600 Manuals and Documents
Switch manuals	PowerSwitch S3100 Manuals and Documents
	PowerSwitch S5200-ON Series Manuals and Documents
	PowerSwitch Z9264F-ON Manuals and Documents

Delta Lake documentation

The following documentation on the [Delta Lake documentation website](#) provides additional and relevant information.

Table 22. Delta Lake documentation

Document type	Location
Lakehouse architecture introductory paper	Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics
Delta Lake project	Delta Lake Project website
Delta Lake documentation	Delta Lake documentation website

Robin.io documentation

The following documentation on the [Robin.io documentation website](#) provides additional and relevant information.

Table 23. Robin.io documentation

Document type	Location
Platform	Robin Cloud Native Platform
Storage	Robin Cloud Native Storage
Node management	Managing Nodes
Bundles	Robin Hyperconverged Kubernetes Bundle Building Guide

Dell Technologies Customer Solution Centers

Our global network of dedicated [Dell Technologies Customer Solution Centers](#) are trusted environments where world class IT experts collaborate with customers and prospects to share best practices; facilitate in-depth discussions of effective business strategies using briefings, workshops, or Proofs of Concept (PoCs); and help business become more successful and competitive.

Dell Technologies Customer Solution Centers reduce the risks that are associated with new technology investments, and can help improve speed of implementation.

All of the services of the Customer Solution Centers are available to all Dell Technologies customers at no charge. Contact your account team today to submit an engagement request.

Dell Technologies Info Hub

The [Dell Technologies Info Hub](#) is your one-stop destination for the latest information about Dell Solutions products. New material is frequently added, so browse often to keep up to date on the expanding Dell portfolio of cutting-edge products and solutions.

More information

For more information, including sizing guidance, technical questions, or sales assistance, email ai.assist@dell.com, or contact your Dell Technologies or authorized partner sales representative.