HPE Reference Configuration for deploying Microsoft SQL Server on Red Hat OpenShift Container Platform on HPE ProLiant DL360 and DL380 Gen10 with HPE Primera A650 Storage

CONTENTS

EXECUTIVE SUMMARY

Enterprise organizations everywhere and across all industries are embarking on a hybrid cloud journey. Their key goals are speed, agility, simplicity, consistency, and cost-effectiveness to support digital transformation, business innovation, and accelerated growth. To meet demanding business goals, many organizations are searching for solutions on how to support transactional data for cloud-native applications in the hybrid cloud environment.

This Reference Configuration is ideal for customers:

- To encourage customers upgrading or migrating their current SQL Server infrastructure to consider deploying databases on a container platform, to support a DevOps ecosystem for cloud-native applications.
- Looking to provision SQL Server instances quickly and to scale up, scale-out, or scale down the number of instances on a needed basis.
- Using Oracle OLTP databases who wish to stay on Linux® and migrate to a lower-cost solution.
- Customers looking to consolidate SQL Server production instances and use Kubernetes native features to provide pod-level high availability and manageability options.
- Customers looking to provide a database as a service to their end customers, which tracks the usage of resources at the instance level and removes the instance when the project is completed.

This Reference configuration is a primer to provide guidance for deploying and configuring Microsoft® SQL Server 2019 on Red Hat® OpenShift® using HPE ProLiant DL380 Gen10 servers, HPE ProLiant DL360 Gen10 servers, and with HPE Primera Storage 650. In this solution, validated and tested automation scripts are used to rapidly deploy the Red Hat OpenShift Container Platform on the HPE ProLiant DL360 and HPE ProLiant DL380 servers. This solution highlights the deployment of StatefulSet SQL Server instances using Helm chart and provisioning of SQL Server OLTP databases on containers. The solution also covers best practices for deploying SQL Server 2019 on a container platform.

Target audience: This Reference Configuration is for CTO, enterprise architects, IT managers, database engineers, and administrators. Working knowledge of server architecture, networking architecture, storage design, and Red Hat OpenShift or Kubernetes is recommended.

Document purpose: The purpose of this document is to describe a Reference Configuration, highlighting key implementation details and benefits to technical audiences.

This Reference Configuration describes solution configuration testing performed in September 2021.

INTRODUCTION

Today in a hybrid cloud environment supporting transactional databases for cloud-native applications, businesses face a constant challenge to keep pace with data processing and storage requirements generated by all aspects of their business. To meet the growing and evolving demand, you need a platform that is scalable, quickly deployed, easily configurable, and able to apply best practices on the provisioned transactional databases.

This Reference Configuration provides guidance to deploy StatefulSet of SQL Server instances using Helm chart on Red Hat OpenShift Container Platform to support cloud-native applications. This solution uses HPE validated automation scripts for rapidly deploying the Red Hat OpenShift Container Platform on HPE ProLiant DL360 and HPE ProLiant DL380 servers. The HPE Primera A650 Storage is used as persistent storage in the Red Hat OpenShift Container Platform. The SQL Server instances running on containers create persistent volumes on HPE Primera Storage to provision SQL Server OLTP databases.

SOLUTION OVERVIEW

The key components of this solution are HPE ProLiant DL360 Gen10 servers, HPE ProLiant DL380 Gen10 servers, HPE Primera A650 Storage, Aruba 8325 switches, Red Hat OpenShift, and Microsoft SQL Server 2019.

Figure 1 depicts the hardware components used for this solution. The solution configuration consists of three HPE ProLiant DL360 Gen10 servers, three HPE ProLiant DL380 Gen10 servers, and HPE Primera Storage A650. Two networks were configured to manage the management traffic and data traffic.

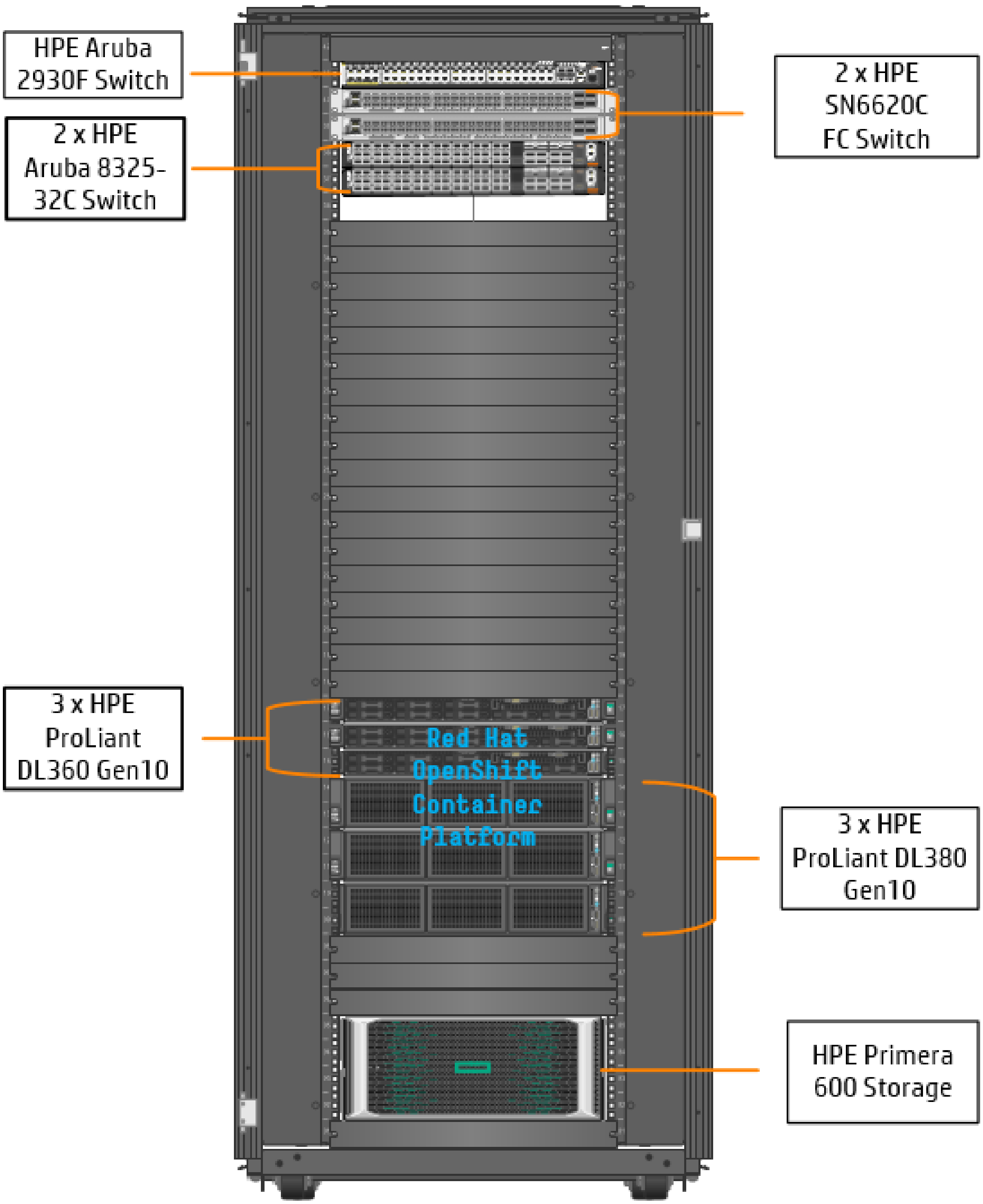


FIGURE 1. Front view of the HPE Rack and solution components featuring HPE ProLiant DL360 and HPE ProLiant DL380 servers, HPE Primera Storage, and Network Switches

HPE ProLiant DL360 Gen10 server

The HPE ProLiant DL360 Gen10 server is a secure, performance driven dense server that can be confidently deployed for virtualization, database, or high-performance computing. The HPE ProLiant DL360 Gen10 server delivers security, agility, and flexibility without compromise.

The HPE ProLiant DL360 Gen10 server supports the Intel® Xeon® Scalable Processor Family with up to 28 cores, plus 2933 MT/s HPE DDR4 SmartMemory supporting up to 3.0 TB max. With the added performance that HPE Persistent Memory and 10 NVMe bring, the HPE ProLiant DL360 Gen10 means business. Deploy this dense platform for diverse workloads in space-constrained environments and maintain it with ease by automating the most essential server lifecycle management tasks with HPE OneView and HPE iLO 5.

FIGURE 2. HPE ProLiant DL360 Gen10 server

HPE ProLiant DL380 Gen10 server

The HPE ProLiant DL380 Gen10 server delivers the latest in security, performance, and expandability. It supports the second-generation Intel® Xeon® Processor Scalable Family supporting HPE 2933 MT/s DDR4 SmartMemory. The HPE ProLiant DL380 Gen10 server has an adaptable chassis, including new HPE modular drive bay configuration options with up to 30 SFF, up to 19 LFF, or up to 20 NVMe drive options along with support for up to 3 double wide GPU options. Along with an embedded 4x1GbE, there is a choice of HPE FlexibleLOM or PCIe standup adapters which offer a choice of networking bandwidth (1GbE to 40GbE) and fabric allowing customers to adapt and grow to changing business needs. The HPE ProLiant DL380 Gen10 server comes with a complete set of HPE Technology Services, delivering confidence, reducing risk, and helping customers realize agility and stability.

FIGURE 3. HPE ProLiant DL380 Gen10 server

HPE Primera

HPE Primera 600 Storage is a Tier-0 enterprise storage solution designed for simplicity, resiliency, and performance when running mission-critical applications and workloads. Built upon proven resiliency and powered by the intelligence of HPE InfoSight, HPE Primera 600 Storage delivers instant access to data with storage that sets up in minutes, upgrades transparently, and is delivered as a service. It ensures always fast and always on storage for all mission-critical applications. HPE Primera 600 Storage is a parallel, multi-node, all-active platform that achieves predictably high performance at extremely low latency.

HPE Primera 600 Storage systems include the HPE Primera 630, HPE Primera 650, and HPE Primera 670. Each model is available in an all-flash-drive configuration or a converged-drive configuration. The converged drive configuration supports a mix of flash and no flash physical drives. All models come with factory-installed HPE Primera OS and HPE Primera UI.

The following summarizes the basic configurations. It does not include information about available add-on drive enclosures and other options. For more information on the number of host ports and maximum storage capacities, see the HPE Primera 600 Storage QuickSpecs.

- **HPE Primera A630 and C630:** The HPE Primera A630 is an all-flash-drive configuration and the HPE Primera C630 is a converged-drive configuration. The 630 includes a 2U base enclosure that contains 2 storage controllers. The 2U base enclosure can hold up to 24 small form factor physical drives.
- **HPE Primera A650 and C650:** The HPE Primera A650 is an all-flash-drive configuration and the HPE Primera C650 is a converged-drive configuration. The 650 is available with a 2U or 4U base enclosure that contains either 2 or 4 storage controllers. The 2U and 4U base enclosure can hold up to 24 or 48 small form factor physical drives.
- **HPE Primera A670 and C670:** The HPE Primera A670 is an all-flash-drive configuration and the HPE Primera C670 is a converged-drive configuration. The 670 is available with a 2U or 4U base enclosure that contains either 2 or 4 storage controllers. The 2U and 4U base enclosures can hold up to 24 or 48 small form factor physical drives.

FIGURE 4. HPE Primera 600 Storage series

Management interfaces for HPE Primera:

- **HPE Primera UI:** The HPE Primera UI is a graphical user interface for managing a single HPE Primera 600 Storage system. HPE Primera UI software is included in each HPE Primera 600 Storage system and does not require installation on a server.
- **HPE Primera CLI:** HPE Primera CLI is a text-based command-line interface for managing one HPE Primera 600 Storage system at a time. The functionality is included in the HPE Primera OS on HPE Primera 600 Storage systems. HPE Primera CLI client software can be installed on hosts (servers) running various computer operating systems.
- **HPE 3PAR StoreServ Management Console (SSMC):** HPE 3PAR SSMC is a graphical user interface for managing multiple HPE Primera 600 Storage systems or HPE 3PAR StoreServ Storage systems or both at a time. The software is available as a virtual appliance and can be downloaded from the HPE Software Depot. The software can be deployed in several supported virtual machine environments.

Aruba 8325-32C BF switch

The Aruba CX 8325 switch is an enterprise-class, game-changing solution, offering a flexible approach to dealing with the new application, security, and scalability demands of the mobile, cloud, and IoT era. With the following benefits:

- Simplify your IT operations with AOS-CX
- Accelerate IT provisioning
- Unparalleled visibility and analytics
- No downtime, even during upgrades

Figure 5 shows the Aruba 8325-32C BF switch.

FIGURE 5. Aruba 8325-32C BF switch

HPE Aruba 2930F switch

The HPE Aruba 2930F switch series is designed for customers creating smart digital workplaces that are optimized for mobile users with an integrated wired and wireless approach. These convenient layer 3 network switches include built-in uplinks and PoE power and are simple to deploy and manage with advanced security and network management tools such as Aruba ClearPass Policy Manager, Aruba AirWave, and cloud-based Aruba Central. A powerful Aruba ProVision ASIC delivers performance, robust feature support, and value with programmability for the latest applications. Stacking with Virtual Switching Framework (VSF) provides simplicity and scalability. The HPE Aruba 2930F switch supports built-in 1GbE or 10GbE uplinks, PoE+, Access OSPF routing, Dynamic Segmentation, robust QoS, RIP routing, and IPv6 with no software licensing required.

Figure 6 shows the HPE Aruba 2930F switch.

FIGURE 6. HPE Aruba 2930F switch

Microsoft SQL Server 2019

SQL Server 2019 challenges the status quo of relational databases by making it possible to ingest, manage, store, and analyze any type of data from anywhere. It expands the working definition of the data environment, enabling a consistent experience of security, identity, and data management across deployments out to the edge. With SQL Server 2019 you will get industry-leading performance with an intelligent database, enterprise-level security, and Artificial Intelligence (AI) built-in. SQL Server 2019 builds on the industry-leading capabilities of SQL Server 2017 to give you even more of the speed, security, and flexibility you need to run your business in the modern world.

Earlier releases of SQL Server introduced several in-memory features that provided greater performance gains such as memory-optimized tables, natively compiled stored procedures, and in-memory clustered columnstore indexes. Using these features together allows you to query operational data without affecting its transactional speed. This scenario, also known as Hybrid Transactional and Analytical Processing (HTAP), can be implemented with memory-optimized tables and clustered columnstore indexes to improve transactional speed and enable real-time visibility and insights into the transactions and operational data.

Memory-optimized TempDB SQL Server 2019 introduces a new feature called Memory-Optimized TempDB metadata, which effectively removes contention bottlenecks and unlocks a new level of scalability for TempDB-heavy workloads. In SQL Server 2019, the system tables involved in managing temporary table metadata can be moved into latch-free non-durable memory-optimized tables.

SQL Server In-Memory Database technologies leverage modern hardware innovation to deliver unparalleled performance and scale. SQL Server 2019 (15.x) builds on earlier innovations in this area, such as in-memory online transaction processing (OLTP), to unlock a new level of scalability across all your database workloads.

SQL Server 2017 on Linux marked the first time SQL Server was available on a platform other than Windows®, it is the same SQL Server database engine, with many similar features and services regardless of your operating system. SQL Server on Linux® is very efficient in terms of deployment time and patching. SQL Server on Linux and containers caters to customers and developers running Linux and containers as part of DevOps, providing them with the ability to leverage the power of SQL Server on any platform. SQL Server 2019 brings the Linux version closer to feature parity with SQL Server on Windows with the addition of key capabilities, like support for replication, Microsoft Distributed Transaction Coordinator (MSDTC), PolyBase, and Machine Learning Services as well as OpenLDAP support for third-party Active Directory providers.

See the Resources and additional links section for links to more details about the SQL Server 2019 release.

Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform unites developers and IT operations on a single platform to build, deploy, and manage applications consistently across hybrid cloud and multi-cloud infrastructures. Red Hat OpenShift helps businesses achieve greater value by delivering modern and traditional applications with shorter development cycles and lower operating costs. Red Hat OpenShift is built on open source innovation and industry standards, including Kubernetes and Red Hat Enterprise Linux.

Red Hat Enterprise Linux CoreOS

Red Hat OpenShift Container Platform uses Red Hat Enterprise Linux® CoreOS (RHCOS), a new container-oriented operating system that combines some of the best features and functions of the CoreOS and Red Hat Atomic Host operating systems. RHCOS is specifically designed for running containerized applications from the Red Hat OpenShift Container Platform and works with new tools to provide fast installation, operator-based management, and simplified upgrades. For Red Hat OpenShift Container Platform 4.6 deployment on bare metal infrastructure, you must use RHCOS for all control plane machines, Bootstrap nodes, and worker nodes.

HPE Container Storage Interface

The HPE Container Storage Interface (CSI) Driver is a multi-vendor and multi-backend driver where each implementation has a Container Storage Provider (CSP). The HPE CSI Driver allows any vendor or project to develop its own Container Storage Provider by using the CSP specification. This makes it very easy for third parties to integrate their storage solution into Kubernetes as all the intricacies are taken care of by the HPE CSI Driver. The CSI specification includes constructs to manage snapshots as native Kubernetes objects, and create new Persistent Volume Claim (PVC) by referencing those objects. Other capabilities include PVC expansion, inline ephemeral volumes, and the ability to present raw block storage to pods. For details about the HPE-CSI driver, Refer to https://scod.hpedev.io/csi_driver/.

Figure 7 shows the HPE-CSI driver architecture.

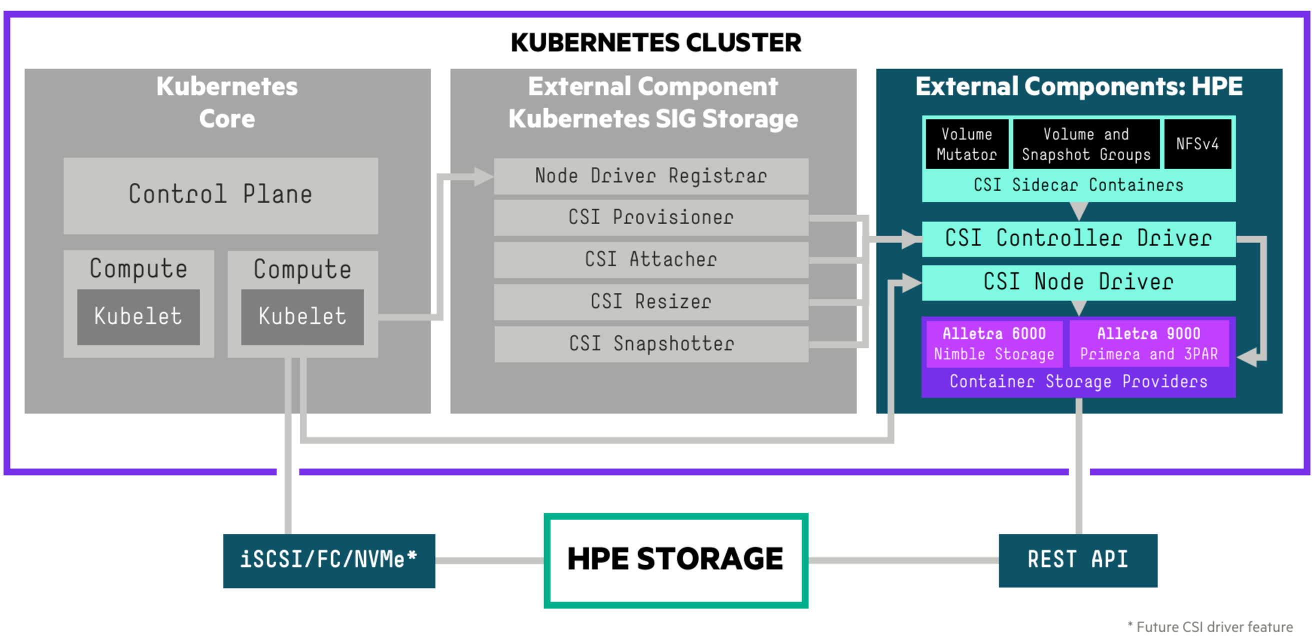


FIGURE 7. HPE CSI Driver for Kubernetes architecture

Figure 8 depicts the logical solution diagram of this Reference Configuration. The Red Hat OpenShift Container Platform is deployed using HPE validated automated scripts. This helps to reduce manual intervention needed during deployment and reduces deployment duration.

For detailed installation and configuration information, see the deployment guide at https://hewlettpackard.github.io/hpe-solutions-openshift/46-dl/. For the Red Hat OpenShift document, see https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html/installing_on_bare_metal/.

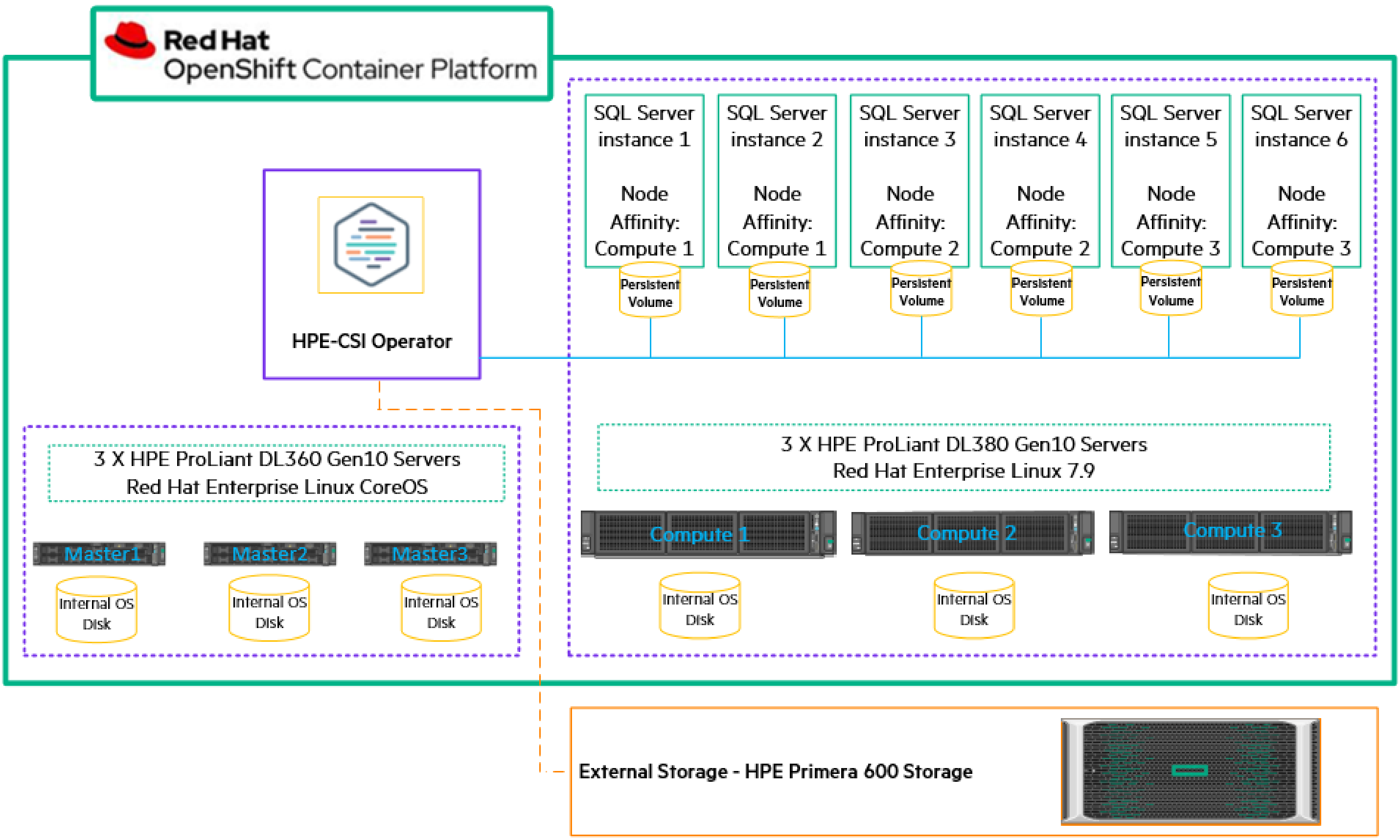


FIGURE 8. Logical solution diagram

The deployed OpenShift Container Platform has three HPE ProLiant DL360 Gen10 servers designated as master nodes with each master node running Red Hat CoreOS. The HPE ProLiant DL380 Gen10 servers are designated as Compute nodes and running Red Hat Enterprise Linux 7.9. The compute nodes can be deployed with Red Hat CoreOS as well. The compute nodes are configured to access HPE Primera A650 Storage through a Fibre Channel switch. The HPE-CSI 2.0 operator is installed on Red Hat OpenShift to create persistent volumes (PV) and assign the persistent volume claim (PVC) to the SQL Server containers for provisioning SQL Server OLTP databases.

The right-sizing of compute and memory is important and hence it is advisable to use up to ~80% of the available compute node resources such as compute and memory. This is to ensure resources are available for other management activities of the Red Hat OpenShift Container Platform. To avoid resource contention we can add resource limits in the deployment files for SQL Server, which then ensures that the resources each pod gets are from a guaranteed quality of service (QOS) which comes from a reserved pool of resources. The affinization ensures optimal performance by using ideal CPUs.

Each compute node has two pods, each pod running one SQL Server container. Thus, each node has two SQL Server container instances, one per pod. The SQL Server containers are affinitized to the compute node. When the need to scale and deploy SQL Server containers arises, additional compute nodes can be added. The details of the deployment of StatefulSet SQL Server instances on containers for this solution are provided in Appendix B.

SOLUTION COMPONENTS

The solution configuration consists of three HPE ProLiant DL360 Gen10 servers, three HPE ProLiant DL380 Gen10 servers, and HPE Primera A650 Storage.

Hardware

Table 1 summarizes hardware components that were utilized in the design and construction of this Reference Configuration.

TABLE 1. Hardware components utilized in this solution

Component	Qty	Description
HPE ProLiant DL360 Gen10 Server 2 x Intel Xeon-Gold 5218 (2.3 GHz/16-core/125 W) 12 x HPE 16GB DDR4-2933 RDIMM 2 x HPE 960 GB SATA 6G MU SFF SSD 1 x HPE InfiniBand FDR/Ethernet 40/50Gb 2-port 547FLR-QSFP Adapter	3	OpenShift master and bootstrap nodes
HPE ProLiant DL380 Gen10 Server 2 x Intel Xeon-Gold 6242 (2.80 GHz/16-core/150 W) 12 X HPE 32GB DDR4-2933 RDIMM 3 X HPE 960GB SATA 6G MU SFF SSD 1 X HPE InfiniBand FDR/Ethernet 40/50Gb 2-port 547FLR-QSFP Adapter	3	OpenShift compute nodes
HPE Primera A650 Storage 2 Controller nodes 12 x 3.84 TB SAS SSD	1	Storage for SQL Server database, etc. This storage is shared between different applications.
HPE B-series SN6620C Fibre Channel Switch	2	Fibre channel switch for providing connectivity between HPE Primera and Servers
HPE Aruba 2930F switch	1	A network switch for the iLO Management network
Aruba 8325-32C 32-port 100G	2	Layer 2 switch for 10 Gb network connectivity

Software

Table 2 shows the software components used in this solution configuration.

TABLE 2. Software used in this solution

Component	Version	Description
Red Hat OpenShift Container Platform	4.6.23	Operating System
Red Hat CoreOS	4.6	Operating System used for OpenShift master nodes in the cluster
Red Hat Enterprise Linux	7.9	Operating System used for computer nodes in the cluster

Application software

Table 3 shows the application software used in this solution configuration.

TABLE 3. Application Software used in this solution

Component	Version	Description
Microsoft SQL Server 2019	CU11	Microsoft SQL Server 2019 CU11 for database
HPE CSI	2.0	To provide HPE Primera volumes as storage for the SQL Server Databases inside the container

For this Reference Configuration, the Red Hat OpenShift 4.6 cluster was deployed by the user provisioned method (UPI) method.

BEST PRACTICES AND CONFIGURATION GUIDANCE FOR THE SOLUTION

This section provides the configuration of SQL Server instances on the Red Hat OpenShift environment, tuning of Red Hat OpenShift environment, the configuration of HPE-CSI, and provisioning volumes for SQL Server containers.

HPE ProLiant DL380 Gen10 UEFI settings

Workload Profiles, one of the HPE Intelligent System Tuning (IST) features, allows tuning of the resources in an HPE ProLiant Gen10 server by choosing a preconfigured workload profile. The server will automatically configure the BIOS settings to match the selected workload. The High-Performance Computing (HPC) workload profile designed for sustained maximum utilization rates for extended periods was selected to achieve the best performance for the transactional workload.

For all the HPE ProLiant DL380 Gen10 servers which are designated as compute nodes in Red Hat OpenShift, the workload profile is set to High-Performance Computing.

RHEL 7.9 Operating System settings

On the compute nodes, the following settings were applied:

- A custom-tuned profile has been used for RHEL tuning parameters. Custom tuned profile is shown in Appendix C
- The `/var/ directory` has adequate space available
- Multipath enabled
- NTP synchronization
- Best practices to be applied to tune OS, Refer to https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-performance-best-practices?view=sql-server-ver15.

Network connections

The master nodes and compute nodes of Red Hat OpenShift are in the same network and the internet is made available.

During boot, all the master and worker nodes in the OCP cluster shall have the same network as that of the Machine Config Server, to fetch ignition files. All the nodes in the cluster need to be assigned an IP address by the DHCP server.

The OCP 4.6 cluster also needs to have internet access to perform the following tasks:

- Access the Red Hat OpenShift Cluster Manager page to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- Access Quay.io to obtain the packages that are required to install your cluster.
- Obtain the packages that are required to perform cluster updates.

TABLE 4. Network Connection used in this solution

Network Name	Type	Description
Management connection	Ethernet	The network is connected to management switch and iLO' s of the servers and management network of storage.
Data_center network	Ethernet	This network is to provide for data traffic, authentication, and access to the application
Storage connection	HBA/FC	Connection for data movement between storage and FC switches.
IRF Link connection	Between switches	The connection between two Ethernet switches

Figure 9 depicts the network wiring in the rack between servers, storage, and network devices.

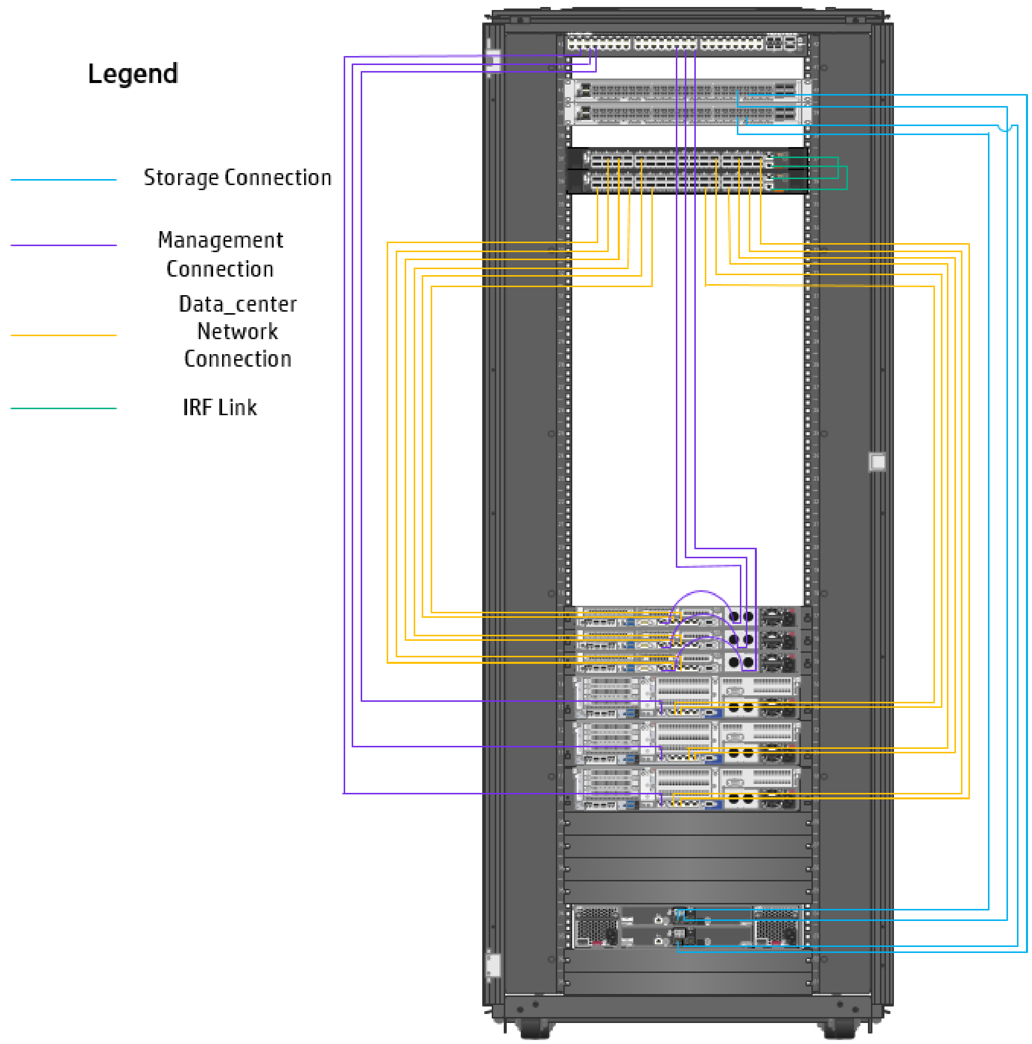


FIGURE 9. Rack Network wiring connections of servers, storage, and network devices

SQL Server settings

During the deployment of SQL Server as containers, the values to set the memory and vCPUs to be consumed are mentioned in Appendix C.

After deploying SQL Server as containers, the SQL Server instances were accessed by SQL Server Studio Management (SSMS) and the Max degree of parallelism is set to 1.

Affinitize SQL Server containers to specific compute nodes to avoid resource contention. The resource contention is avoided by deploying SQL Server containers with Quality of Service (QoS) set to guaranteed, by setting the resource limits and CPU manager to static. See Appendix B for more details.

Node port service created on Red Hat OpenShift cluster to connect to specific SQL Server instance running on container.

CAPACITY AND SIZING

Workload description

The OLTP workload tests were conducted using HammerDB, an open-source tool. For this Reference Configuration, HammerDB 3.3 was used to implement an OLTP workload.

Table 5 shows the provisioning of the databases inside the SQL Server containers running on three compute nodes.

TABLE 5. Provisioning of the databases

Compute Node	Database Name	Database Size	Datafile location	Logfile location
Worker1	tpcc_1	200 GB	/var/opt/mssql/userdata	/var/opt/mssql/userlog
Worker1	tpcc_2	200 GB	/var/opt/mssql/userdata	/var/opt/mssql/userlog
Worker2	tpcc_3	200 GB	/var/opt/mssql/userdata	/var/opt/mssql/userlog
Worker2	tpcc_4	200 GB	/var/opt/mssql/userdata	/var/opt/mssql/userlog
Worker3	tpcc_5	200 GB	/var/opt/mssql/userdata	/var/opt/mssql/userlog
Worker3	tpcc_6	200 GB	/var/opt/mssql/userdata	/var/opt/mssql/userlog

The performance characterization test was done to find out an optimal number of active users driving OLTP transactions on the SQL Server databases. The OLTP transactions were driven from the HammerDB testing tool by a varying number of active users and observed the overall performance of the entire cluster. The solution showed better performance among all the SQL Server containers deployed when active users were scaled to 20 for each database and the transaction per minute recorded almost equally among the databases. This shows the solution can be used to drive the workload uniformly across all the SQL Server containers hosting OLTP databases with sustained TPM. In Figure 10, the result indicates the driven workload on the databases showed with sustained transactions per minute across all the databases running on SQL Server containers. This also indicates when the equal workload is distributed among databases the system achieved desirable results and can be further scaled when needed.

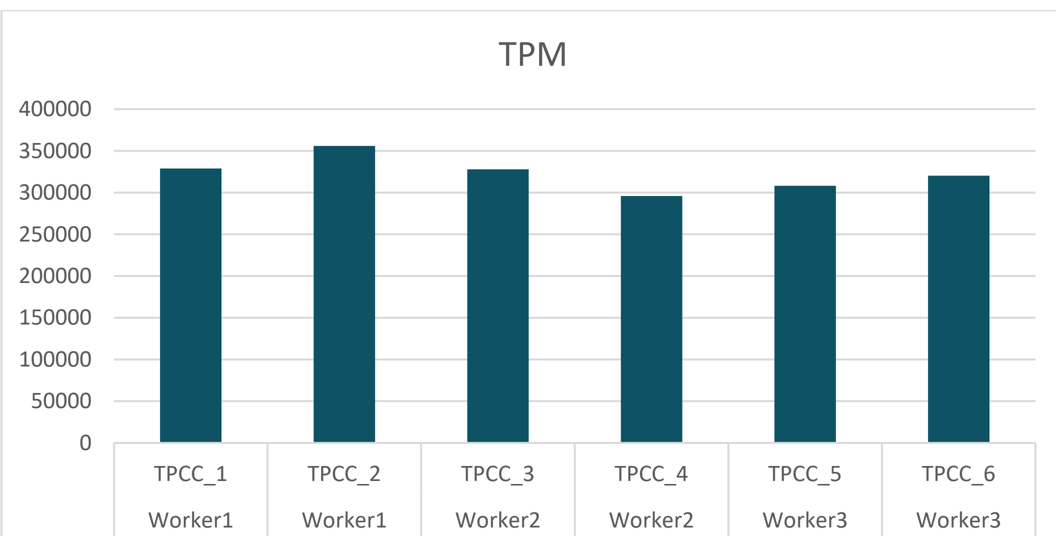


FIGURE 10. TPM results of database testing

SUMMARY

The solution demonstrates:

- To deploy SQL Server on Red Hat OpenShift Container Platform

- Highlight best practices
- Performance characterization

The solution shows HPE ProLiant DL360 Gen10 and HPE ProLiant DL380 Gen10 servers are validated to build container platforms for hybrid cloud environments. HPE Primera A650 Storage provides the required storage for the SQL Server databases by using the HPE-CSI driver. This solution showcases the easy deployment of SQL Server 2019 in containers and achieves expected performance to support DevOps applications in a hybrid cloud environment. The solution can be dynamically scaled by adding additional HPE ProLiant DL380 Gen10 servers as compute nodes.

IMPLEMENTING A PROOF-OF-CONCEPT

As a matter of best practice for all deployments, Hewlett Packard Enterprise recommends implementing a proof-of-concept using a test environment that matches as closely as possible the planned production environment. In this way, appropriate performance and scalability characterizations can be obtained and validated. For help with a proof-of-concept, contact a Hewlett Packard Enterprise Services representative (hpe.com/us/en/services/consulting.html) or your Hewlett Packard Enterprise partner.

APPENDIX A: BILL OF MATERIALS

NOTE
Part numbers are at the time of publication/testing and subject to change. The bill of materials does not include complete support options or other rack and power requirements. If you have questions regarding ordering, please consult with your Hewlett Packard Enterprise Reseller or Hewlett Packard Enterprise Sales Representative for more details. hpe.com/us/en/services/consulting.html.

TABLE A1. Bill of materials (sample for a solution designed on top of a CSS core platform. Note: remove this comment before publication)

Part number	Quantity	Description
		Rack
P9K10A	1	HPE 42U 600mmx1200mm G2 Kitted Advanced Shock Rack with Side Panels and Baying
P9K10A 001	1	HPE Factory Express Base Racking Service
		Master Nodes
867959-B21	3	HPE ProLiant DL360 Gen10 8SFF Configure-to-order server
867959-B21 0D1	3	Factory Integrated
867959-B21 ABA	3	HPE DL360 Gen10 8SFF CTO server
P02592-L21	3	Intel Xeon-Gold 5218 (2.3GHz/16-core/125W) FIO Processor Kit for HPE ProLiant DL360 Gen10
P02592-B21	3	Intel Xeon-Gold 5218 (2.3GHz/16-core/125W) Processor Kit for HPE ProLiant DL360 Gen10
P02592-B21 0D1	3	Factory Integrated
P00922-B21	36	HPE 16GB (1x16GB) Dual Rank x8 DDR4-2933 CAS-21-21-21 Registered Smart Memory Kit
P00922-B21 0D1	36	Factory Integrated
P18434-B21	6	HPE 960GB SATA 6G Mixed Use SFF (2.5in) SC 3yr Wty Multi-Vendor SSD
P18434-B21 0D1	6	Factory Integrated
P01366-B21	3	HPE 96W Smart Storage Lithium-ion Battery with 145mm Cable Kit
P01366-B21 0D1	3	Factory Integrated
804326-B21	3	HPE Smart Array E208i-a SR Gen10 (8 Internal Lanes/No Cache) 12G SAS Modular Controller
804326-B21 0D1	3	Factory Integrated

Part number	Quantity	Description
879482-B21	3	HPE InfiniBand FDR/Ethernet 40/50Gb 2-port 547FLR-QSFP Adapter
879482-B21 0D1	3	Factory Integrated
865414-B21	6	HPE 800W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit
865414-B21 0D1	6	Factory Integrated
BD505A	3	HPE iLO Advanced 1-server License with 3yr Support on iLO Licensed Features
BD505A 0D1	3	Factory Integrated
734811-B21	3	HPE 1U Cable Management Arm for Rail Kit
734811-B21 0D1	3	Factory Integrated
867998-B21	3	HPE 1U Gen10 Bezel Kit
867998-B21 0D1	3	Factory Integrated
874543-B21	3	HPE 1U Gen10 SFF Easy Install Rail Kit
874543-B21 0D1	3	Factory Integrated
		Worker Nodes
868703-B21	3	HPE ProLiant DL380 Gen10 8SFF Configure-to-order server
868703-B21 0D1	3	Factory Integrated
868703-B21 ABA	3	HPE DL380 Gen10 8SFF CTO server
P02510-L21	3	Intel Xeon-Gold 6242 (2.8GHz/16-core/150W) FIO Processor Kit for HPE ProLiant DL380 Gen10
P02510-B21	3	Intel Xeon-Gold 6242 (2.8GHz/16-core/150W) Processor Kit for HPE ProLiant DL380 Gen10
P02510-B21 0D1	3	Factory Integrated
P00924-B21	36	HPE 32GB (1x32GB) Dual Rank x4 DDR4-2933 CAS-21-21-21 Registered Smart Memory Kit
P00924-B21 0D1	36	Factory Integrated
P18434-B21	9	HPE 960GB SATA 6G Mixed Use SFF (2.5in) SC 3yr Wty Multi-Vendor SSD
P18434-B21 0D1	9	Factory Integrated
826694-B21	3	HPE DL38X Gen10 x16/x16 Riser Kit
826694-B21 0D1	3	Factory Integrated
871674-B21	3	HPE DL38X Gen10 Slot 1/2 x16/x16 FIO Riser Kit
804326-B21	3	HPE Smart Array E208i-a SR Gen10 (8 Internal Lanes/No Cache) 12G SAS Modular Controller
804326-B21 0D1	3	Factory Integrated
879482-B21	3	HPE InfiniBand FDR/Ethernet 40/50Gb 2-port 547FLR-QSFP Adapter
879482-B21 0D1	3	Factory Integrated
867810-B21	3	HPE DL38X Gen10 High-Performance Temperature Fan Kit
867810-B21 0D1	3	Factory Integrated
830272-B21	3	HPE 1600W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit
830272-B21 0D1	6	Factory Integrated
BD505A	3	HPE iLO Advanced 1-server License with 3yr Support on iLO Licensed Features
BD505A 0D1	3	Factory Integrated
733664-B21	3	HPE 2U Cable Management Arm for Easy Install Rail Kit

Part number	Quantity	Description
733664-B21 0D1	3	Factory Integrated
867809-B21	3	HPE Gen10 2U Bezel Kit
867809-B21 0D1	3	Factory Integrated
733660-B21	3	HPE 2U Small Form Factor Easy Install Rail Kit
733660-B21 0D1	3	Factory Integrated
		Storage
N9Z47A	1	HPE Primera 600 4-way Storage Base
581817-B21	1	HPE Configurator Defined Build Instruction Option
N9Z60A	1	HPE Primera A650 2-node Controller
N9Z60A 0D1	1	Factory Integrated
R1P29A	1	HPE Data Encryption LTU
R1P29A 0D1	1	Factory Integrated
R0P99A	6	HPE Primera 600 3.84TB SAS SFF (2.5in) FIPS Encrypted SSD
R0P99A 0D1	6	Factory Integrated
N9Z38A	2	HPE Primera 600 16Gb 4-port Fibre Channel Host Bus Adapter
N9Z38A 0D1	2	Factory Integrated
716195-B21	4	HPE External 1.0m (3ft) Mini-SAS HD 4x to Mini-SAS HD 4x Cable
716195-B21 0D1	4	Factory Integrated
		Networking
		HPE Aruba 8325-32C 32-port 100G Switching
R9F67A	2	Aruba 8325-32C 32-port 100G
R9F78A	18	HPE Aruba 100G QSFP28 to QSFP28 5M DAC Cab
R9F58A	2	HPE Aruba X2C2 RJ45 to DB9 Console Cable
JL260A B2E	1	HPE Aruba 2930F 48G 4SFP Switch United States 220 volt
BW932A	1	HPE 600mm Rack Stabilizer Kit
BW932A B01	1	HPE 600mm Rack include with Complete System Stabilizer Kit
R1B12AAE	3	Red Hat OpenShift Container Platform 1-32 Cores 1yr Subscription 9x5 RH Support E-LTU
Q9Y41AAE	1	HPE Network Orchestrator E-LTU
JL271A	6	HPE X240 100G QSFP28 to QSFP28 1m Direct Attach Copper Cable
H1K92A3	1	HPE 3Y Proactive Care 24x7 SVC
H1K92A3 R2M	6	HPE iLO Advanced Non Blade Support
H1K92A3 WAG	3	HPE DL360 Gen10 Support
H1K92A3 WAH	3	HPE DL38x Gen10 Support
J8J36AAE	6	Red Hat Enterprise Linux Server 2 Sockets 1 Guest 1 Year Subscription 24x7 Support E-LTU

APPENDIX B: DEPLOYING SQL SERVER AND CONFIGURATION ON RED HAT OPENSHIFT CONTAINER PLATFORM

Before performing the below operations, a dedicated jump box machine preferably of Linux has to be configured to connect to the Red Hat OpenShift Container Platform. Install OC CLI client tool and HELM.

An SQL Server namespace was created on the Red Hat OpenShift Container Platform from the CLI:

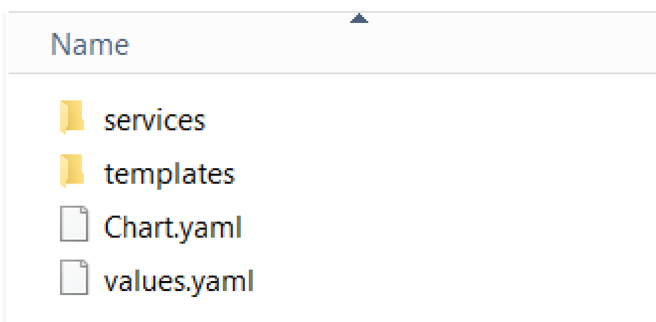
```
oc new-project sql-server
```

All the objects were created under the sql-server project.

The HPE-CSI operator was deployed on the Red Hat OpenShift Container Platform. Refer to details for installing the HPE-CSI Operator https://scod.hpedev.io/partners/redhat_openshift/index.html.

Microsoft has provided a sample reference to deploy StatefulSet SQL Server instances on a container platform. For more details about the sample refer to https://github.com/microsoft/mssql-docker/tree/master/linux/sample-helm-chart-statefulset-deployment. For our solution, we cloned the repo and customized the YAML files to deploy on the Red Hat OpenShift Container Platform.

The repo contains the following directory structure:



The values.YAML file was modified as below:

```
# Default values for mssql-latest.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicas: 2
image:
  repository: mcr.microsoft.com/mssql/rhel/server
  pullPolicy: IfNotPresent
  # Overrides the image tag whose default is the chart appVersion.
  tag: "2019-CU11-rhel-7.9"
ACCEPT_EULA:
    value: "y"
MSSQL_PID:
    value: "Enterprisecore"
MSSQL_AGENT_ENABLED:
    value: "false"
containers:
  ports:
      containerPort: 1433
podAnnotations: {}
nodeSelector: {}
podSecurityContext:
   fsGroup: 10001
service:
  port: 1433
```

In the templates folder, `sc. yaml` (storage class) file was modified to connect to HPE Primera A650 Storage via HPE-CSI operator.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: primera-storageclass
provisioner: csi.hpe.com
allowVolumeExpansion: true
parameters:
  csi.storage.k8s.io/fstype: xfs
  csi.storage.k8s.io/provisioner-secret-name: primera-secret
  csi.storage.k8s.io/provisioner-secret-namespace: hpe-csi
  csi.storage.k8s.io/controller-publish-secret-name: primera-secret
  csi.storage.k8s.io/controller-publish-secret-namespace: hpe-csi
  csi.storage.k8s.io/node-stage-secret-name: primera-secret
  csi.storage.k8s.io/node-stage-secret-namespace: hpe-csi
  csi.storage.k8s.io/node-publish-secret-name: primera-secret
  csi.storage.k8s.io/node-publish-secret-namespace: hpe-csi
  # Uncomment for k8s 1.15 for resize support
  csi.storage.k8s.io/controller-expand-secret-name: primera-secret
  csi.storage.k8s.io/controller-expand-secret-namespace: hpe-csi
  cpg: "SSD_r6"
  provisioning_type: "tpvv"
  accessProtocol: "fc"
volumeBindingMode: WaitForFirstConsumer
```

To deploy SQL Server containers on each worker node, the following `deploy.yaml` was executed.

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: "{{ include "sql-statefull-deploy.fullname" . }}-1"
  labels:
    {{- include "sql-statefull-deploy.labels" . | nindent 4 }}
spec:
  serviceName: {{ include "sql-statefull-deploy.fullname" . }}
  replicas: {{ .Values.replicas }}
  selector:
    matchLabels:
      {{- include "sql-statefull-deploy.selectorLabels" . | nindent 6 }}
  template:
    metadata:
      {{- with .Values.podAnnotations }}
      annotations:
        {{- toYaml . | nindent 8 }}
      {{- end }}
      labels:
        {{- include "sql-statefull-deploy.selectorLabels" . | nindent 8 }}
    spec:
      securityContext:
        {{- toYaml .Values.podSecurityContext | nindent 8 }}
      nodeSelector:
        kubernetes.io/hostname: worker1.local
      containers:
        - name: {{ .Chart.Name }}
          command:
            - /bin/bash
            - -c
            - cp /var/opt/config/mssql.conf /var/opt/mssql/mssql.conf && /opt/mssql/bin/sqlservr
```

```
        image: "{{ .Values.image.repository }}:{{ .Values.image.tag | default .Chart.AppVersion }}"



        resources:
         requests:
           memory: 64Gi
           cpu: 24
         limits:
           memory: 64Gi
           cpu: 24
        ports:
         - containerPort: {{ .Values.containers.ports.containerPort}}
        env:
         - name: MSSQL_PID
           value: "{{ .Values.MSSQL_PID.value}}"
         - name: ACCEPT_EULA
           value: "{{ .Values.ACCEPT_EULA.value | upper}}"
         - name: MSSQL_AGENT_ENABLED
           value: "{{ .Values.MSSQL_AGENT_ENABLED.value}}"
         - name: SA_PASSWORD
           valueFrom:
            secretKeyRef:
             name: mssql
             key: SA_PASSWORD
        volumeMounts:
         - name: mssql
           mountPath: "/var/opt/mssql"
         - name: mssql-config-volume
           mountPath: /var/opt/config
         - name: userdata
           mountPath: "/var/opt/mssql/userdata"
         - name: userlog
           mountPath: "/var/opt/mssql/userlog"
         - name: tempdb
           mountPath: "/var/opt/mssql/tempdb"
    volumes:
    - name: mssql-config-volume
      configMap:
       name: mssql
volumeClaimTemplates:
  - metadata:
     name: mssql
    spec:
     accessModes:
     - ReadWriteOnce
     resources:
      requests:
       storage: 1024Gi
     storageClassName: primera-storageclass
  - metadata:
     name: userdata
    spec:
     accessModes:
     - ReadWriteOnce
     resources:
      requests:
       storage: 2048Gi
     storageClassName: primera-storageclass
```

```
  - metadata:
      name: userlog
    spec:
      accessModes:
      - ReadWriteOnce
      resources:
       requests:
         storage: 512Gi
      storageClassName: primera-storageclass
  - metadata:
      name: tempdb
    spec:
      accessModes:
      - ReadWriteOnce
      resources:
       requests:
         storage: 512Gi
      storageClassName: primera-storageclass
```

In the above YAML file, the node selector helps to mention on which node the container has to be deployed and resources help specify how much vCPUs and memory the pod should reserve from the Red Hat OpenShift Container Platform. The above YAML file can be modified to suit the storage, vCPUs, Memory and on which compute node the containers are to be affinitized. After configuring all the YAML files as needed from the executing below command to install the SQL Server using Helm.

```
helm install .mssql
```

Successful installation of SQL Server containers will show running and the node it is running on as mentioned in Figure B1.

```
[root@visetup backup]# oc get pods -n sqlserver -o wide
NAME                           READY   STATUS    RESTARTS   AGE   IP            NODE          NOMINATED NODE   READINESS GATES
mssql-sql-statefull-deploy-1-0 1/1     Running   0          12h   10.130.4.10   vinworker1    <none>           <none>
mssql-sql-statefull-deploy-1-1 1/1     Running   0          12h   10.130.4.11   vinworker1    <none>           <none>
mssql-sql-statefull-deploy-2-0 1/1     Running   0          12h   10.128.6.44   vinworker2    <none>           <none>
mssql-sql-statefull-deploy-2-1 1/1     Running   0          12h   10.128.6.46   vinworker2    <none>           <none>
mssql-sql-statefull-deploy-3-0 1/1     Running   0          12h   10.128.2.53   vinworker3    <none>           <none>
mssql-sql-statefull-deploy-3-1 1/1     Running   0          12h   10.128.2.54   vinworker3    <none>           <none>
[root@visetup backup]#
```

FIGURE B1. Deployed StatefulSet SQL Server PODS running

To connect to SQL Server instances, a Kubernetes service needs to be created with the port number SQL Server is listening to, in this case, we use nodeport service.

```
apiVersion: v1
kind: Service
metadata:
  name: mssql-service-1-0
  namespace: sqlserver
spec:
 selector:
  statefulset.kubernetes.io/pod-name: mssql-sql-statefull-deploy-1-0
 ports:
  - protocol: TCP
    port: 1433
    targetPort: 1433
    nodePort: 30050
 type: NodePort
```

As mentioned in Figure B2 from SQL Server Studio, a management connection is established to the SQL Server instance and the database can be created to drive the user workload.

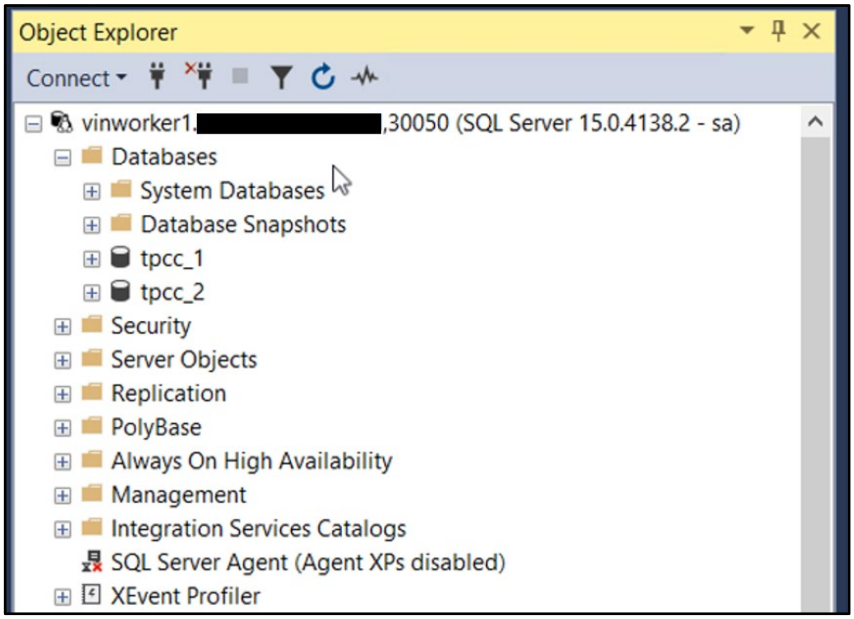


FIGURE B2. Deployed StatefulSet SQL Server PODS running

APPENDIX C: RHEL KERNEL SETTINGS USING A TUNED PROFILE AND NODE AFFINITY FOR THE CONTAINER

The RHEL tuning parameters that were set in `/usr/lib/tuned/mssql/tuned.conf` for each compute nodes are listed below.

```
# A tuned configuration for SQL Server on Linux
#
[main]
summary=Optimized for Microsoft SQL Server
include=throughput-performance
[cpu]
force_latency=5
[sysctl]
vm.swappiness = 1
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
vm.transparent_hugepages=always
vm.max_map_count=16000000
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.numa_balancing=0
kernel.sched_latency_ns=60000000
kernel.sched_migration_cost_ns = 500000
kernel.sched_min_granularity_ns=15000000
kernel.sched_wakeup_granularity_ns=2000000
```

To avoid resource contention within the host when more than one SQL Server container is running following settings were applied on the compute nodes.

```
oc label node worker1.local cpumanager=true
node/vinworker1.local labeled
oc label node worker2.local cpumanager=true
node/vinworker2.local labeled
oc label node worker3.local cpumanager=true
node/vinworker3.local labelled
```

Edit the `machineconfigpool` by executing following command and `custom-kubelet: cpumanager-enabled`.

```
oc edit machineconfigpool worker
[added custom-kubelet: cpumanager-enabled]
machineconfigpool.machineconfiguration.openshift.io/worker edited
cat cpumanager-kubeletconfig.yaml
apiVersion: machineconfiguration.openshift.io/v1
kind: KubeletConfig
metadata:
  name: cpumanager-enabled
spec:
  machineConfigPoolSelector:
    matchLabels:
      custom-kubelet: cpumanager-enabled
  kubeletConfig:
     cpuManagerPolicy: static
     cpuManagerReconcilePeriod: 5s
```

The following content is saved as `cpumanager-kubeletconfig.yaml` and created using `oc create -f cpumanager-kubeletconfig.yaml.`

```
apiVersion: machineconfiguration.openshift.io/v1
kind: KubeletConfig
metadata:
  name: cpumanager-enabled
spec:
  machineConfigPoolSelector:
    matchLabels:
      custom-kubelet: cpumanager-enabled
  kubeletConfig:
     cpuManagerPolicy: static
     cpuManagerReconcilePeriod: 5s
     topologyManagerPolicy: single-numa-node
```

RESOURCES AND ADDITIONAL LINKS

HPE Reference Architectures, hpe.com/info/ra

HPE Servers, hpe.com/servers

HPE Storage, hpe.com/storage

HPE Networking, hpe.com/networking

HPE Technology Consulting Services, https://www.hpe.com/us/en/services/consulting.html

HPE Primera Storage, https://www.hpe.com/psnow/doc/a00067738enw

Red Hat OpenShift Container Platform, https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/

Red Hat OpenShift Container Storage, https://access.redhat.com/documentation/en-us/red_hat_openshift_container_storage/4.6/

Red Hat OpenShift Container Platform 4.6 on HPE DL380 and DL360 Gen10 Servers, https://hewlettpackard.github.io/hpe-solutions-openshift/46-dl/

Microsoft SQL Server 2019 SQL Server 2019, https://www.microsoft.com/en-us/sql-server/sql-server-2019

HPE ProLiant DL380 Gen10 server, https://www.hpe.com/psnow/doc/a00008180enw

HPE ProLiant DL360 Gen10 server, https://www.hpe.com/psnow/doc/a00008159enw

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

Hewlett Packard Enterprise

a50005165enw, September 2024