Reference Architecture

Easily Configure and Deploy VMs and Containers with SUSE® Rancher Prime and SUSE Virtualization, using KIOXIA CD8 Series Data Center NVMe™ SSDs in Dell™ PowerEdge™ R760 Servers

Gerson Guevara
Jacob Durante
Adil Rahman
Tyler Nelson
Scott Harlin

Table of Contents

Introduction ... 3

Background ... 3

System Configuration ... 4

SUSE® Virtualization Cluster Initialization ... 4

SUSE Virtualization Cluster Configuration ... 5

Virtual Machine Network ... 5

Cloud Configuration Templates ... 5

OS Upload Image ... 6

SUSE Rancher Prime Configuration ... 6

Kubernetes Cluster Deployment ... 8

Test Methodology ... 10

Test Case Description ... 10

Test Results ... 11

Test Analysis ... 12

Summary ... 12

Introduction

This reference architecture (RA) showcases integration between SUSE® Virtualization and SUSE Rancher Prime software utilizing KIOXIA CD8 Series data center NVMe™ SSDs on Dell™ PowerEdge™ R760 rack servers. It focuses on the ease of deploying containerized applications, and their storage requirements, in one unified platform.

The RA presents testing, test methodology, test results and test analysis for a configuration that includes three Dell PowerEdge R760 servers with Intel® Xeon® CPUs and the ability to utilize up to sixteen PCIe® 4.0 NVMe SSDs per server.

The steps performed to initialize the cluster showcase easy set-up of a hyperconverged infrastructure (HCI), in addition to supporting a Kubernetes® management platform.

The test results showed that the underlying storage subsystem in the HCI environment served many storage requests simultaneously for the containers residing in the Kubernetes cluster through fast, scalable local storage driven by KIOXIA CD8 Series SSDs.

Reference Architecture Summary

The cluster consisted of three Dell PowerEdge R760 servers, each deployed with one KIOXIA CD8 Series PCIe 4.0 data center NVMe SSD, to demonstrate ease of set-up for SUSE Virtualization and SUSE Rancher Prime software.

SUSE Virtualization software provided and provisioned the underlying compute, memory and storage resources through the harvester-longhorn driver to the Kubernetes layer.

SUSE Rancher Prime software provided the ability to easily set-up and manage different Kubernetes distributions, such as RKE2 or K3s.

Multiple containers were run simultaneously to emulate the I/O blender effect and to showcase balanced throughput performance and low latency from the underlying KIOXIA CD8 Series SSDs.

Background

Containers are a form of virtualization and are used to package services with their dependencies. They are intended to be lightweight and able to run without depending on the environment that each container is deployed in. Kubernetes open-source platforms are used to automate the management, deployment and scaling of containerized applications enabling users and developers to manage large-scale data processing and deploy microservice architectures. As a result of this automation, Kubernetes platforms are commonly used for DevOps and ITOps.

The main challenge with Kubernetes platforms is how complex its structure and deployment are for administrators. Some of this complexity includes the initial set-up of a Kubernetes cluster, scaling out applications, managing the storage capacity needs of various containerized workloads and managing the network. Servicing many containers, with different storage needs for capacity and performance, is an additional challenge that needs to be considered.

SUSE Rancher Prime software is a Kubernetes management platform designed to simplify the deployment and management of Kubernetes clusters. SUSE Virtualization software is used in hyperconverged infrastructures to deploy virtual machines (VMs). Both software platforms serve to abstract compute, memory and storage resources from various hosts into a cluster of resources that can be provisioned for use by VMs or containers. When SUSE Rancher Prime and SUSE Virtualization software are combined with Dell PowerEdge servers and KIOXIA CD8 Series NVMe SSDs, the results demonstrate a complete solution that empowers the scaling out of large Kubernetes deployments.

KIOXIA CD8 Series NVMe SSDs provide balanced performance and low latency. They are built on the high-speed PCIe interface that uses lanes to directly connect to the CPU and delivers significantly improved performance and low latency versus SAS and SATA SSD options.

System Configuration

Testing was then conducted to demonstrate the performance of the configuration. The system configuration used for testing now follows:

Server Information	
Server Model	Dell™ PowerEdge™ R760
Number of Servers	3
BIOS Version	2.4.4
Processor Information	
Processor Model	Intel® Xeon® Gold 6430
Number of Processors	2
Core Count	32
Core Frequency	2,100 MT/s*
System Memory Information	
Total System Memory	256 gigabytes[1] (GB)
Number of Memory Modules	16
Size	16 GB
Type	DDR5
Speed	DDR5-4400
SSD Information	
SSD Model	KIOXIA CD8 Series
Number of SSDs	3
Capacity	7,680 GB
Form Factor	2.5-inch[2] (15 mm)
Interface	PCIe® 4.0 / NVMe™ 1.4
DWPD[3]	1
Operating System Information	
Operating System	SUSE® Virtualization
Version	v1.4.0
VM Operating System	SUSE Linux Enterprise Server 15 SP6
Version	sles15-sp6-minimal-vm.x86_64-cloud-qu1.qcow2
Kubernetes® Management Tool	SUSE Rancher Prime
Version	v2.10.1
RKE2 Version	v1.31.4+rke2r1
Load Generation Tool	yasker/kbench
Commit Version	8fd580c

*MT/s = megatransfers per second

SUSE Virtualization Cluster Initialization

The three Dell PowerEdge R760 servers were used to create a SUSE Virtualization cluster. The first host was installed with the SUSE Virtualization operating system (OS), and the option to create a new SUSE Virtualization cluster was selected. Host information was set through the on-screen prompt, where the installation drive for the OS and drive to store data was selected. The KIOXIA CD8 Series SSDs were deployed as the data drives for the hosts. The SUSE Virtualization network was configured to utilize a 100 gigabits[1] per second (Gb/s) dual ported network interface card (NIC) in an active-backup configuration. IPv4 addresses were assigned to the host, along with a domain name system (DNS), and a chosen virtual IP address, to access the SUSE Virtualization user interface (UI). The cluster token and the host password were then set.

The other two hosts also had the SUSE Virtualization OS installed, and the option to join an existing SUSE Virtualization cluster was selected. The role of these two hosts was set to default, and all other configurations in the first host were used. The two remaining hosts were joined to the cluster using the management virtual IP and cluster token provided by the first host.

SUSE® Virtualization Cluster Configuration

Once the SUSE Virtualization installation across the hosts was completed, the SUSE Virtualization UI was accessed via the virtual IP address.

Virtual Machine Network

The VM network was created and deployed using the default namespace. An UntaggedNetwork type was selected utilizing the default management network as the cluster network as depicted in Figure 1:

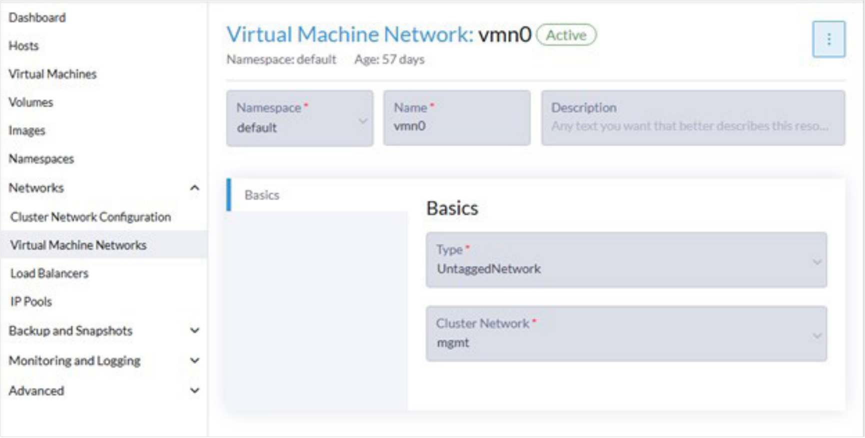

Figure 1: Virtual machine network settings
(image reproduced with permission from SUSE S.A.)

Cloud Configuration Templates

Cloud configuration templates utilize cloud-init, which is an industry standard multi-distribution method for providing startup scripts to VMs. These templates were created for the default namespace to enable a dynamic host configuration protocol (DHCP) IP assignment and set static logins to the VMs in SUSE Virtualization software. By utilizing these cloud configuration templates, users can script tasks that need to be performed when a VM is being created. The DHCP cloud configuration template is responsible for assigning IP addresses to future VMs that are created. The username-setup cloud configuration template allows all VMs that are created to use a specific username and password when installing the OS on them.

Figure 2 depicts the network data template, and Figure 3 shows the user data template created in SUSE Virtualization software:

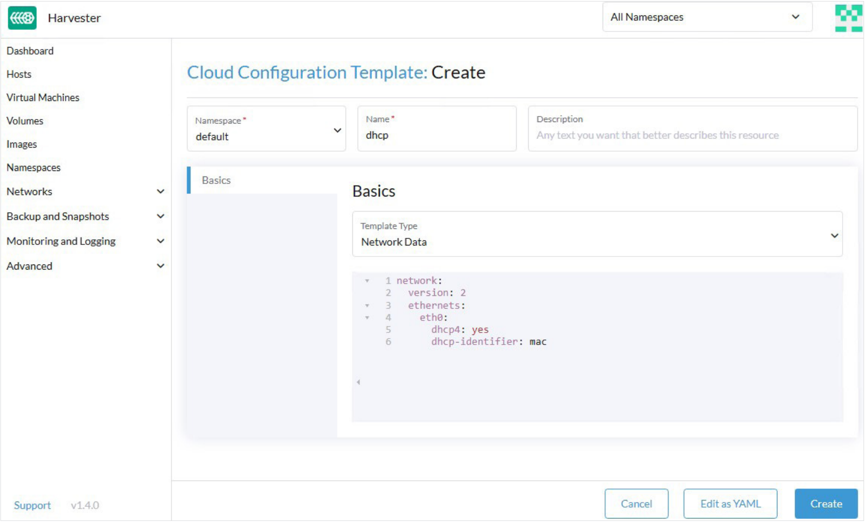

Figure 2: Network data template
(image reproduced with permission from SUSE S.A.)

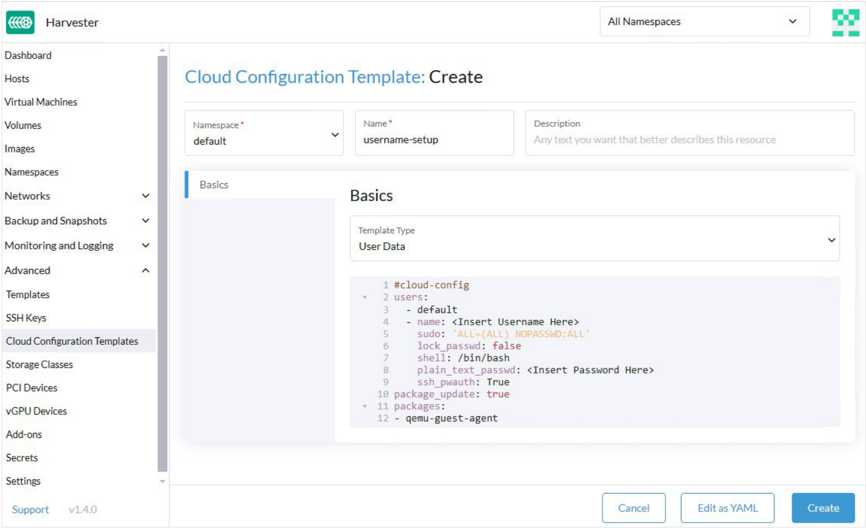

Figure 3: User data template

(image reproduced with permission from SUSE S.A.)

OS Upload Image

The SUSE® Linux Enterprise Server 15 SP6 OS image was then uploaded to the SUSE Virtualization cluster. This OS image was used for every VM that was created and included the initial SUSE Rancher Prime VM, the control plane / etcd nodes and the worker nodes. The OS image upload details are shown in Figure 4:

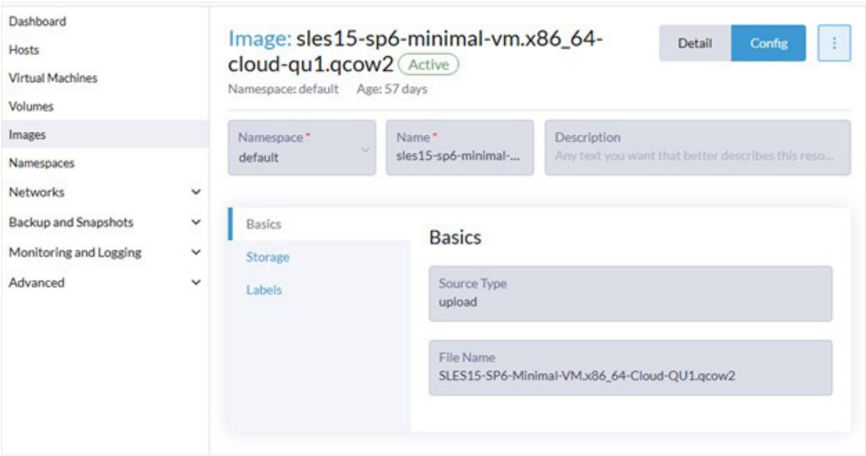

Figure 4: OS image upload

(image reproduced with permission from SUSE S.A.)

SUSE Rancher Prime Configuration

SUSE Rancher Prime software was deployed in a VM using the default namespace. This VM allows users to access the SUSE Rancher Prime UI, which grants the ability to manage multiple Kubernetes® clusters and their respective pods. The VM was created with eight cores and a total of 24 GB of memory. A volume was created for the VM of 100 GB for utilizing the SUSE Linux Enterprise Server 15 SP6 OS. The network was set to the VM network created in earlier steps of vmn0. The username-setup and DHCP cloud configuration templates were selected in the advanced options. Figure 5 shows the basic configuration. Figure 6 shows the volume configuration. Figure 7 shows the network configuration.

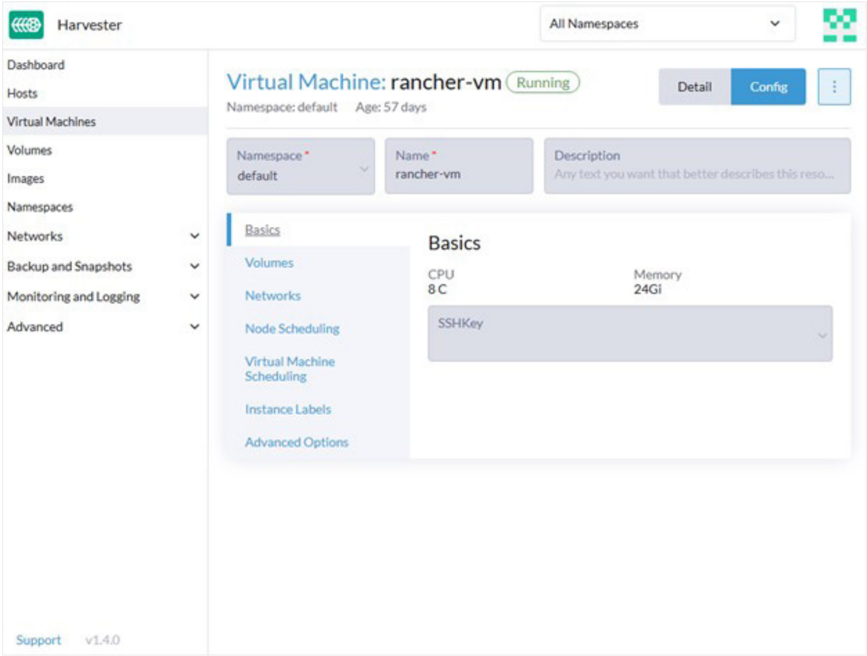

Figure 5: SUSE® Rancher Prime basic configuration
(image reproduced with permission from SUSE S.A.)

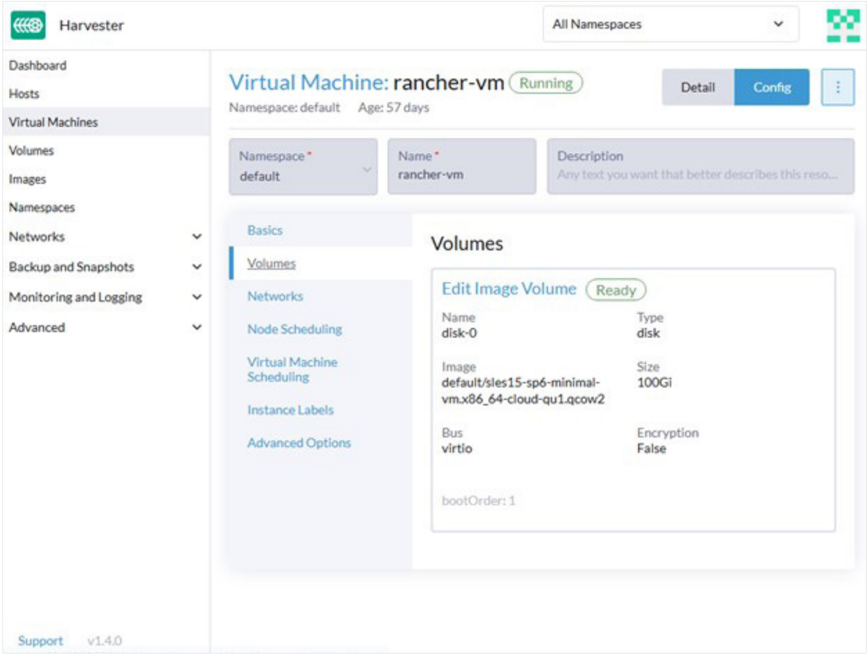

Figure 6: SUSE Rancher Prime volume configuration
(image reproduced with permission from SUSE S.A.)

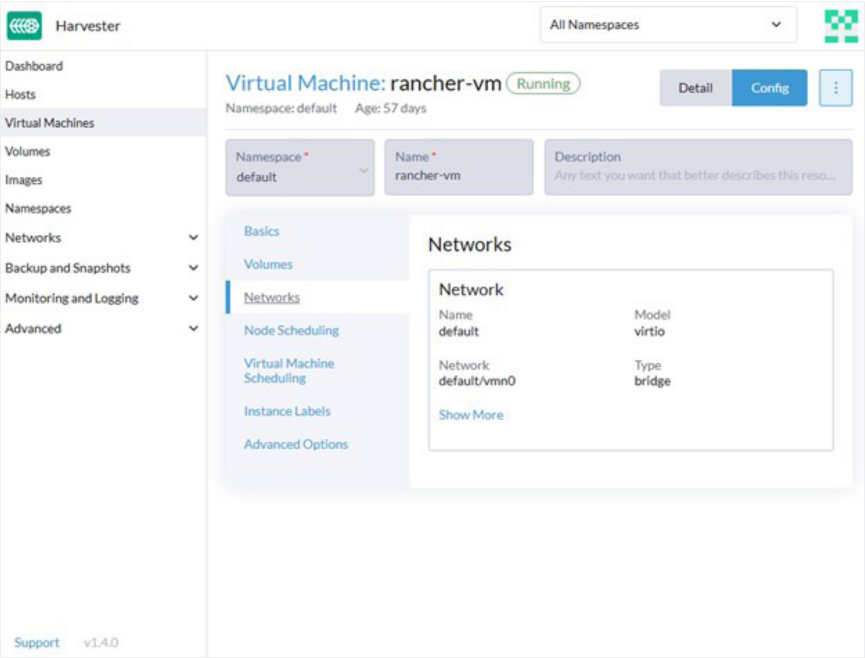

Figure 7: SUSE® Rancher Prime network configuration
(image reproduced with permission from SUSE S.A.)

The newly created VM had a SUSE Enterprise license key added. The swap capability was disabled and iptables were installed. The RKE2 distribution method was then installed onto the system, alongside kubectl. The git-core and helm libraries were installed, and the cattle-system and cert-manager Kubernetes® namespaces were created. The helm library was then used to install jetstack cert-manager, and then subsequently, rancher-stable. The SUSE Rancher Prime UI was then accessible via the fully qualified domain name.

Lastly, access to the SUSE Virtualization cluster was added to SUSE Rancher Prime via the virtualization management tab. A registration URL was listed from SUSE Rancher Prime, which was also provided to the SUSE Virtualization UI via the settings tab for the cluster registration URL parameter. These steps ultimately allow for the creation of VMs through the SUSE Rancher Prime UI, which is necessary in the following steps to assign VMs with Kubernetes cluster roles, such as control plane / etcd and worker nodes, in the creation of new Kubernetes clusters.

Kubernetes Cluster Deployment

With the SUSE Rancher Prime VM successfully installed to manage the Kubernetes deployments, a separate Kubernetes cluster needs to be created to host the containerized applications. SUSE Virtualization software was selected as the node provisioner for a new K3s cluster in the cluster management tab of the SUSE Rancher Prime UI. A new cloud credential must be created for the SUSE Virtualization-cluster.

In the cluster settings, three nodes were provisioned for control plane and etcd services, while six nodes were provisioned as workers. Each control plane / etcd node was assigned two cores, 4 gibibytes[4] (GiB) of memory, and 40 GiB of storage, while each worker node was assigned twenty cores, 40 GiB of memory, and 500 GiB of storage as depicted in the figures below. Various Kubernetes cluster actions and concepts, such as workload deployment, CronJobs, DaemonSets, Deployments, Jobs, StatefulSets, and others can now be easily accessed and managed in the SUSE Rancher Prime UI for the new Kubernetes cluster. Figure 8 shows the Kubernetes cluster deployment for control plane and etcd services. Figure 9 shows the Kubernetes cluster deployment for workers. Figure 10 shows the Kubernetes cluster deployment for workloads.

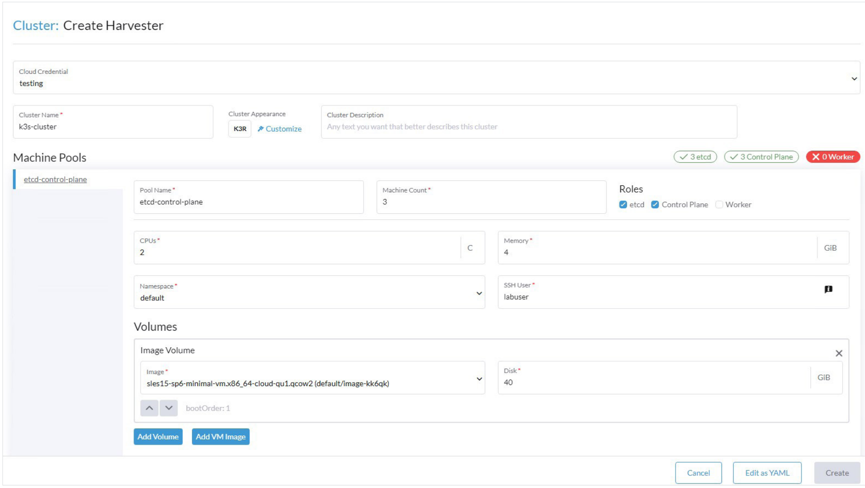

Figure 8: Kubernetes® cluster deployment for control plane and etcd services

(image reproduced with permission from SUSE S.A.)

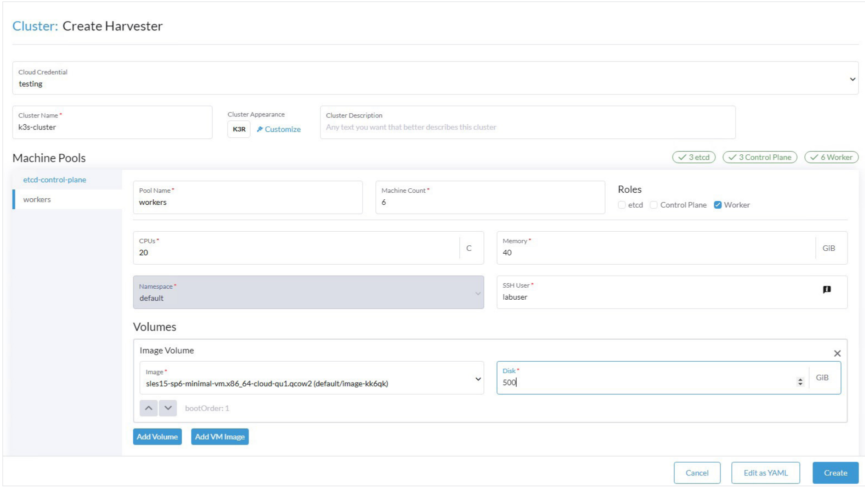

Figure 9: Kubernetes cluster deployment for workers

(image reproduced with permission from SUSE S.A.)

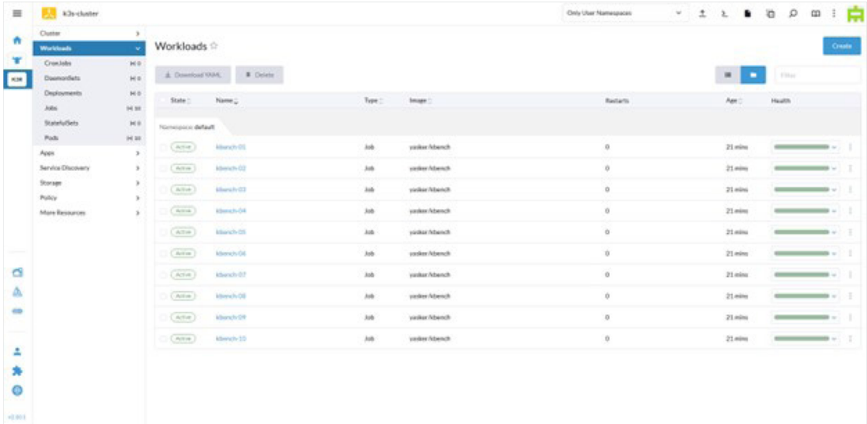

Figure 10: Kubernetes cluster deployment for workloads

(image reproduced with permission from SUSE S.A.)

Test Methodology

Kubernetes® clusters can run many unique types of containerized workloads, all of which will have their own unique storage needs. These workloads can include traditional relational databases, unstructured databases, large object stores, web servers, distributed file systems and other storage-centric workloads. The HCI cluster and the underlying storage subsystem must be able to handle the various I/O requests while still maintaining low read and write latencies to ensure that application performance is not lagging.

These containerized workloads create persistent volume claims (PVC), which are dynamically provisioned by the harvester-longhorn container storage interface (CSI). When PVCs are created, each workload that is deployed will use storage differently, and each PVC can submit different mixed I/O workloads to potentially the same drive causing an I/O blender effect that can impact both application and storage performance.

An I/O blender effect refers to a mix of different workloads originating from a single application or multiple applications on a physical server within containerized environments. VMs and containers are typically the originators of I/O blender workloads. When an abundance of applications run simultaneously in them, both sequential and random data I/O streams are sent to SSDs. Any sequential I/O that exists at that point will be typically mixed with all the other I/O streams, essentially becoming random read/write workloads. As multiple servers and applications process these workloads, and move data at the same time, the SSD activity changes from just sequential or random read/write workloads into a large mix of random read/write I/Os, hence, the I/O blender effect.

The system configuration was tested using KBench, a benchmarking tool for Kubernetes storage that utilizes the Flexible I/O[5] benchmarking tool underneath, to show how it handles the I/O blender effect. KBench created a PVC for requesting a given capacity of storage, which was then fulfilled by the default CSI.

To emulate the I/O blender effect, multiple simultaneous instances of KBench were run at different time intervals, to drive varying I/O workloads. SUSE® Virtualization software utilized the harvester-longhorn CSI driver to dynamically provision storage to requesting container workloads. The harvester-longhorn CSI driver utilized the Longhorn™ v1 data engine. KBench was run with the test parameters shown as follows:

KBench Test Parameters	Values
Number of Instances	10
PVC Size	33 GiB
Size of Test Volume	30 GiB

Test Case Description

Within KBench, the 'full' run was used on all containerized instances against the cluster. The full workload ran each of the following workload profiles sequentially. Since each KBench instance was initiated at a separate time, all the following I/O workloads were handled by the KIOXIA CD8 Series SSDs simultaneously. The full run consisted of the I/O workloads shown below:

Workload	blocksize	iodepth	numjobs
100% Sequential Read	128k	64	1
100% Sequential Write	128k	64	1
100% Random Read	4k	64	32
100% Random Write	4k	64	32

The testing metrics gathered during the I/O blender test included the following:

- ***Average Drive Read Throughput:***
  The average read throughput achieved from the underlying storage subsystem while the I/O blender workload was running. The result was measured in gigabytes per second (GB/s). The higher result is better. See Figures 11 and 13.
- ***Average Drive Write Throughput:***
  The average write throughput achieved from the underlying storage subsystem while the I/O blender workload was running. The result was measured in GB/s. The higher result is better. See Figures 11 and 13.

- ***Average Drive Read Latency:***
  The time it took to perform a drive read operation. It included the average time it took for the workload generator to not only issue the read operation, but also the time it took to complete the read operation and receive an acknowledgment for completion. The result was measured in milliseconds (ms). The lower result is better. See Figures 12 and 13.
- ***Average Drive Write Latency***
  The time it took to perform a drive write operation. It included the average time it took for the workload generator to not only issue the write operation, but also the time it took to complete the write operation and receive an acknowledgment for completion. The result was measured in ms. The lower result is better. See Figures 12 and 13.

Test Results

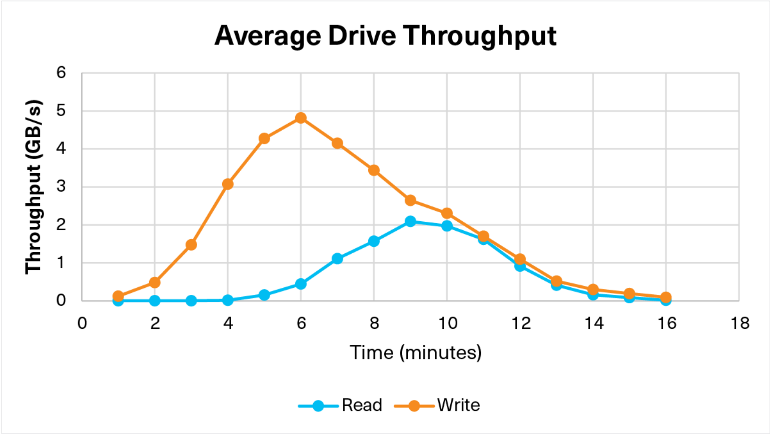

Figure 11: Average read/write drive throughput results

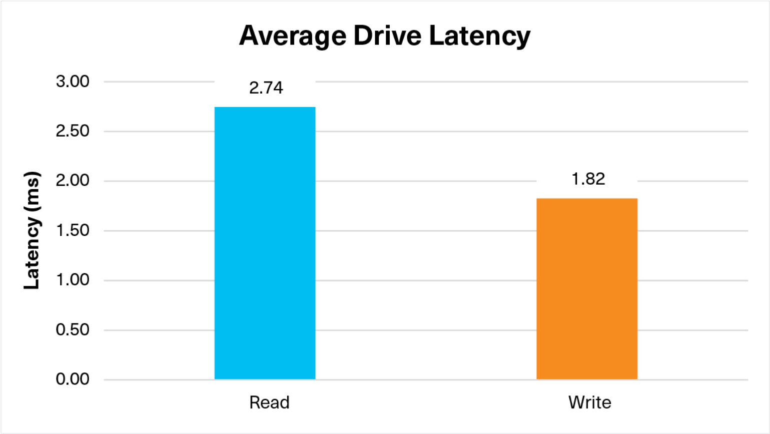

Figure 12: Average read/write drive latency results

Test	Average Read Throughput	Average Write Throughput	Average Read Latency	Average Write Latency
10 KBench Instances	2.09 GB/s	4.82 GB/s	2.74 ms	1.82 ms

Figure 13: Summation of the average read/write drive throughput and latency results

Test Analysis

The SUSE® Virtualization cluster nodes were configured with a 100 Gb/s network, which allowed for a theoretical maximum of 12.5 GB/s. However, due to protocol and software overhead, the interface speed was less than the theoretical maximum. During the KBench runs, the combined network bandwidth between transmitting and receiving was 8.3 GB/s. The data replica count was set to three, which showed that the KIOXIA CD8 Series SSDs were not only handling I/O from the KBench containers, but also was replicating I/O across the cluster for data redundancy. The handling and replicating of I/O introduced significant network utilization, which can quickly become the bottleneck in solutions that require a higher number of active containers.

From testing conducted in the KIOXIA Innovation Lab, it is recommended for production use cases to introduce a separate storage network from the management layer, which requires an additional NIC to place that storage traffic. This recommendation can help alleviate overall network congestion and enable more containers to run.

Despite the higher network utilization across the cluster, the test results showed that the maximum drive read for the cluster was 2.09 GB/s and the maximum drive write throughput was 4.82 GB/s. These results were shown at the peak of the I/O blender workload, where all KBench instances were running various I/O workloads at the same time. The storage subsystem was able to still maintain an average of 2.74 ms for read latency and 1.82 ms for write latency. The KIOXIA CD8 Series SSDs were able to handle the I/O blender effect and service all the varying I/O requests from the different containers. The drives are capable of much more, as a 7,680 GB capacity KIOXIA CD8 Series SSD can support up to 7.1 GB/s of sequential read performance and 6.0 GB/s of sequential write performance[6].

Summary

SUSE Rancher Prime and SUSE Virtualization software combined with KIOXIA CD8 Series data center NVMe™ SSDs deployed in Dell™ PowerEdge™ 760 rack servers offer a complete solution that provides the storage needs of both container and VM deployments. SUSE Rancher Prime software allows system administrators to easily manage multiple Kubernetes® clusters in one platform, while providing balanced performance and low-latency storage to their various containerized applications and their respective storage needs. This combination highlights that KIOXIA CD8 Series SSDs work well for large-scale deployments that focus on low latency.

Additional KIOXIA CD8 Series SSD information is available here.

Additional SUSE Rancher information is available here.

Additional SUSE Virtualization information is available here.

Additional SUSE Security information is available here.

Additional SUSE Observability information is available here.

Additional SUSE Storage information is available here.

Additional Dell PowerEdge information is available here.

FOOTNOTES:

[1] Definition of capacity – Kioxia Corporation defines a megabyte (MB) as 1,000,000 bytes, a gigabyte (GB) as 1,000,000,000 bytes and a terabyte (TB) as 1,000,000,000,000 bytes. A computer operating system, however, reports storage capacity using powers of 2 for the definition of 1Gbit = 2^{30} bits = 1,073,741,824 bits, 1GB = 2^{30} bytes = 1,073,741,824 bytes and 1TB = 2^{40} bytes = 1,099,511,627,776 bytes and therefore shows less storage capacity. Available storage capacity (including examples of various media files) will vary based on file size, formatting, settings, software and operating system, and/or pre-installed software applications, or media content. Actual formatted capacity may vary.

[2] 2.5-inch indicates the form factor of the SSD and not the drive's physical size.

[3] DWPD: Drive Write(s) Per Day: One full drive write per day means the drive can be written and re-written to full capacity once a day, every day, for the specified lifetime. Actual results may vary due to system configuration, usage, and other factors.

[4] A gibibyte (GiB) means 2^{30}, or 1,073,741,824 bytes.

[5] Flexible I/O (FIO) is a free and open-source disk I/O tool used both for benchmark and stress/hardware verification. The software displays a variety of I/O performance results, including complete I/O latencies and percentiles.

[6] The KIOXIA CD8 Series SSD performance specifications provided by Kioxia Corporation are accurate as of this publication. Specifications are subject to change. Read and write speed may vary depending on the host device, read and write conditions, and the file size.