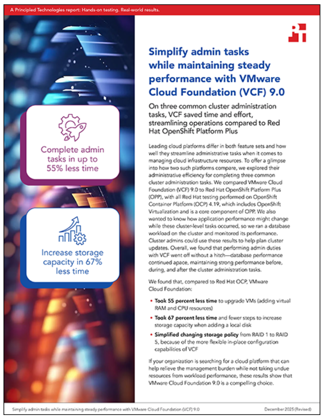
A Principled Technologies report: Hands-on testing. Real-world results.

Simplify admin tasks while maintaining steady performance with VMware Cloud Foundation (VCF) 9.0

On three common cluster administration tasks, VCF saved time and effort, streamlining operations compared to Red Hat OpenShift Platform Plus

Complete admin tasks in up to 55% less time

Increase storage capacity in 67% less time

Leading cloud platforms differ in both feature sets and how well they streamline administrative tasks when it comes to managing cloud infrastructure resources. To offer a glimpse into how two such platforms compare, we explored their administrative efficiency for completing three common cluster administration tasks. We compared VMware Cloud Foundation (VCF) 9.0 to Red Hat OpenShift Platform Plus (OPP), with all Red Hat testing performed on OpenShift Container Platform (OCP) 4.19, which includes OpenShift Virtualization and is a core component of OPP. We also wanted to know how application performance might change while these cluster-level tasks occurred, so we ran a database workload on the cluster and monitored its performance. Cluster admins could use these results to help plan cluster updates. Overall, we found that performing admin duties with VCF went off without a hitch—database performance continued apace, maintaining strong performance before, during, and after the cluster administration tasks.

We found that, compared to Red Hat OCP, VMware Cloud Foundation:

- **Took 55 percent less time** to upgrade VMs (adding virtual RAM and CPU resources)
- **Took 67 percent less time** and fewer steps to increase storage capacity when adding a local disk
- **Simplified changing storage policy** from RAID 1 to RAID 5, because of the more flexible in-place configuration capabilities of VCF

If your organization is searching for a cloud platform that can help relieve the management burden while not taking undue resources from workload performance, these results show that VMware Cloud Foundation 9.0 is a compelling choice.

Simplify admin tasks while maintaining steady performance with VMware Cloud Foundation (VCF) 9.0 December 2025 (Revised)

The science behind the report:

Simplify admin tasks while maintaining steady performance with VMware Cloud Foundation (VCF) 9.0

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report Simplify admin tasks while maintaining steady performance with VMware Cloud Foundation (VCF) 9.0.

We concluded our hands-on testing on October 1, 2025. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on September 23, 2025 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to http://facts.pt/calculating-and-highlighting-wins.
Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Results of our testing.

	VMware Cloud Foundation	Red Hat OpenShift Platform Plus
VM upgrades		
Time to complete VM upgrades (seconds)	36.6	82.8
Database performance before VM upgrade (TPM)	853,469	532,680
Database performance after VM upgrade (TPM)	1,095,378	511,020
Increasing storage capacity		
Time to increase storage capacity (seconds)	51.2	156.9
Database performance before storage expansion (TPM)	1,148,593	577,365
Database performance after storage expansion (TPM)	1,179,049	569,671

VM upgrade

Database performance (TPM)

VMware Cloud Foundation database performance during VM upgrade (TPM)

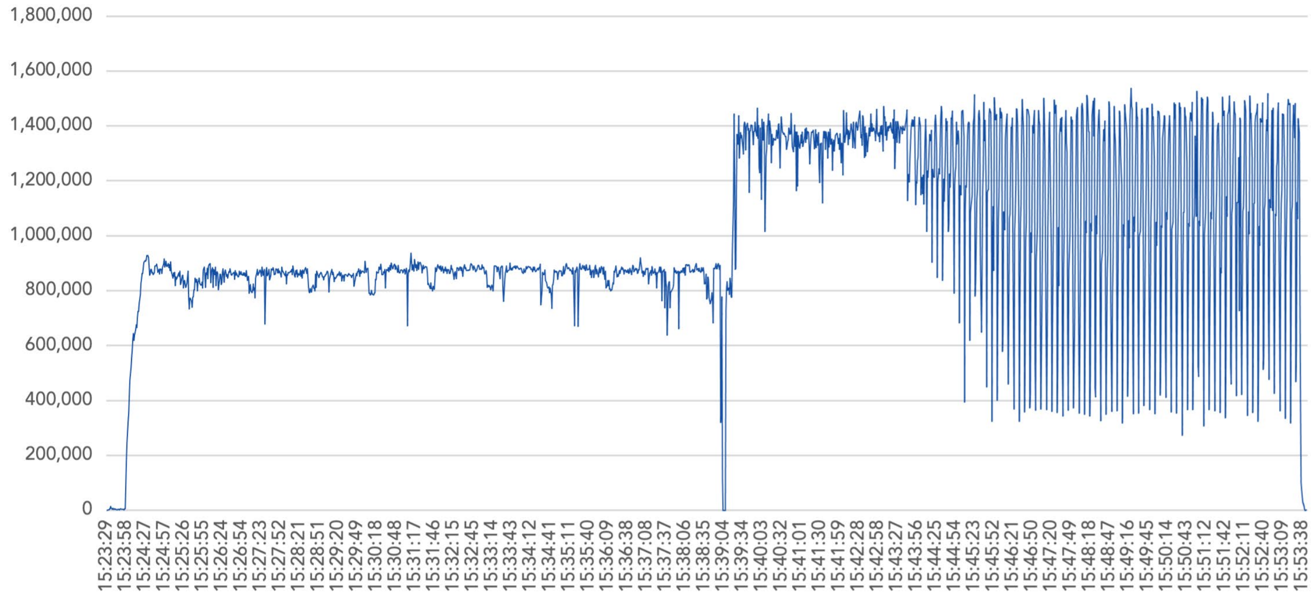

Figure 5: VCF 9 database performance during the VM upgrade. Higher is better. Source: PT.

Red Hat OpenShift Platform Plus database performance during VM upgrade (TPM)

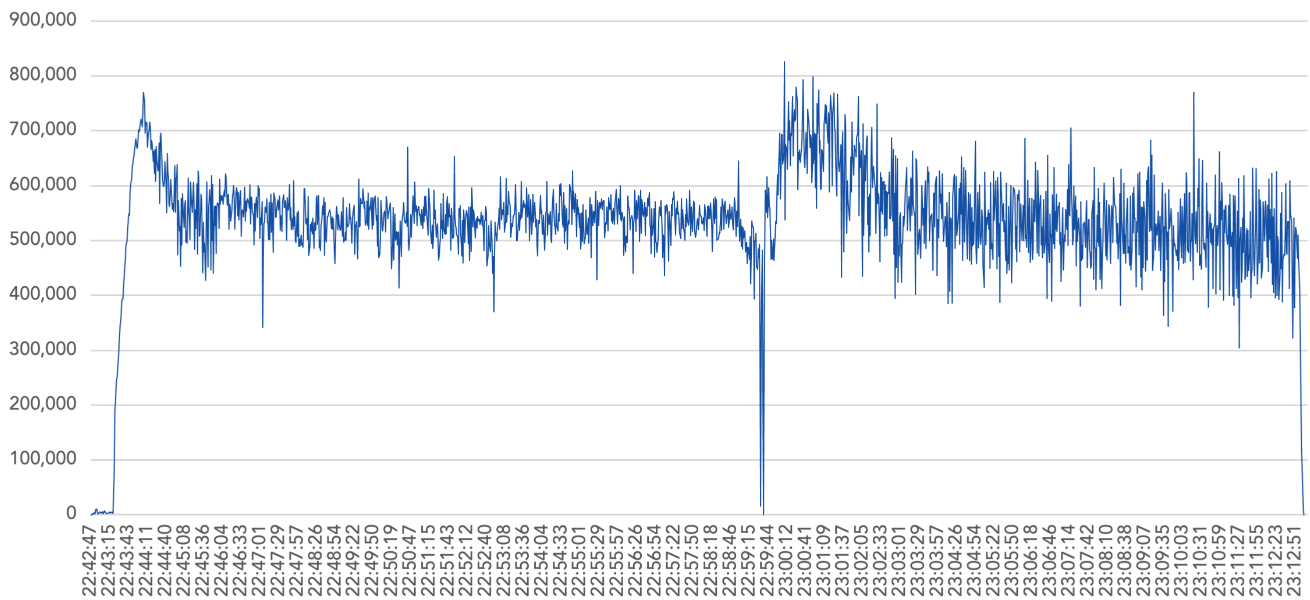

Figure 6: OCP database performance during the VM upgrade. Lower is better. Source: PT.

CPU utilization (%)

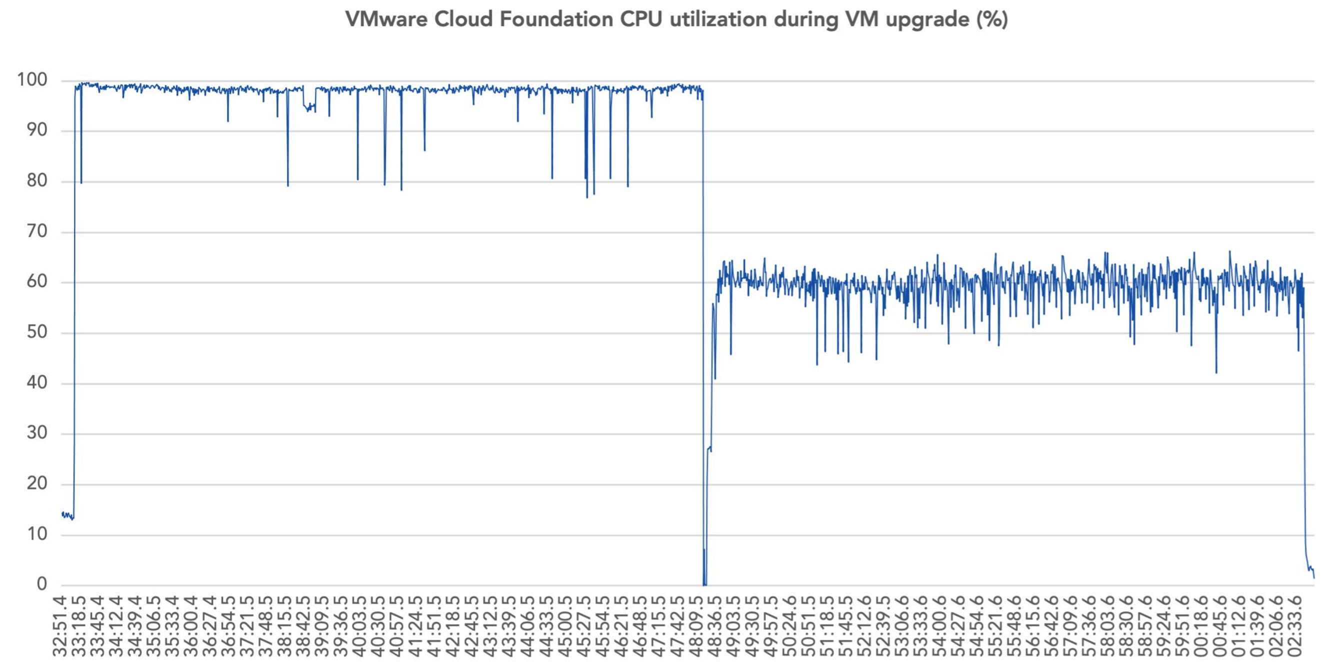

Figure 7: VCF 9 CPU utilization during the VM upgrade. Lower is better. Source: PT.

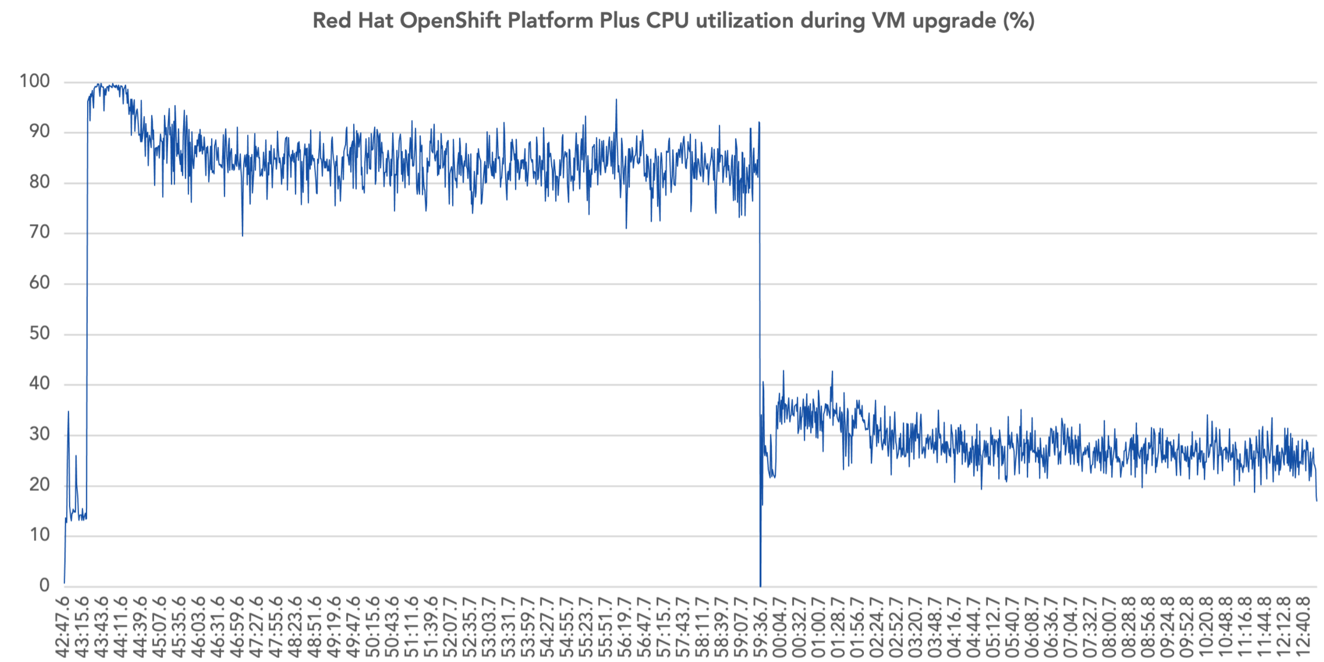

Figure 8: OCP CPU utilization during the VM upgrade. Lower is better. Source: PT.

Datastore expansion

Database performance (TPM)

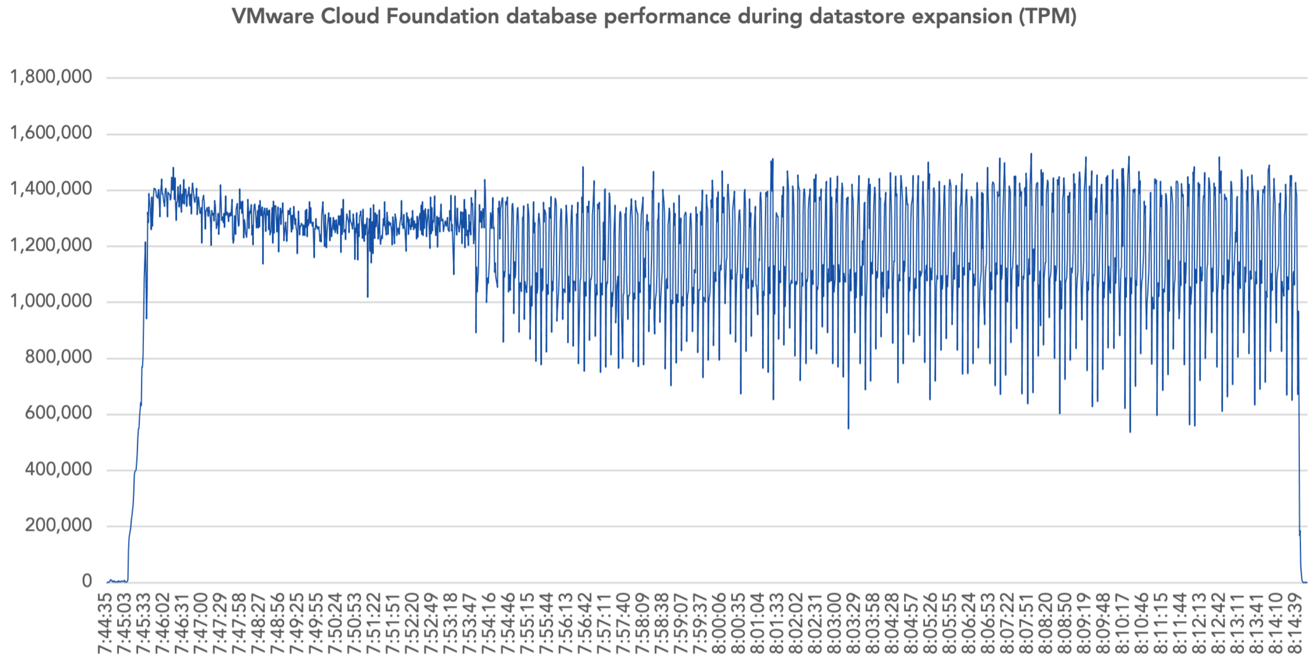

Figure 9: VCF 9 database performance during the datastore expansion. Higher is better. Source: PT.

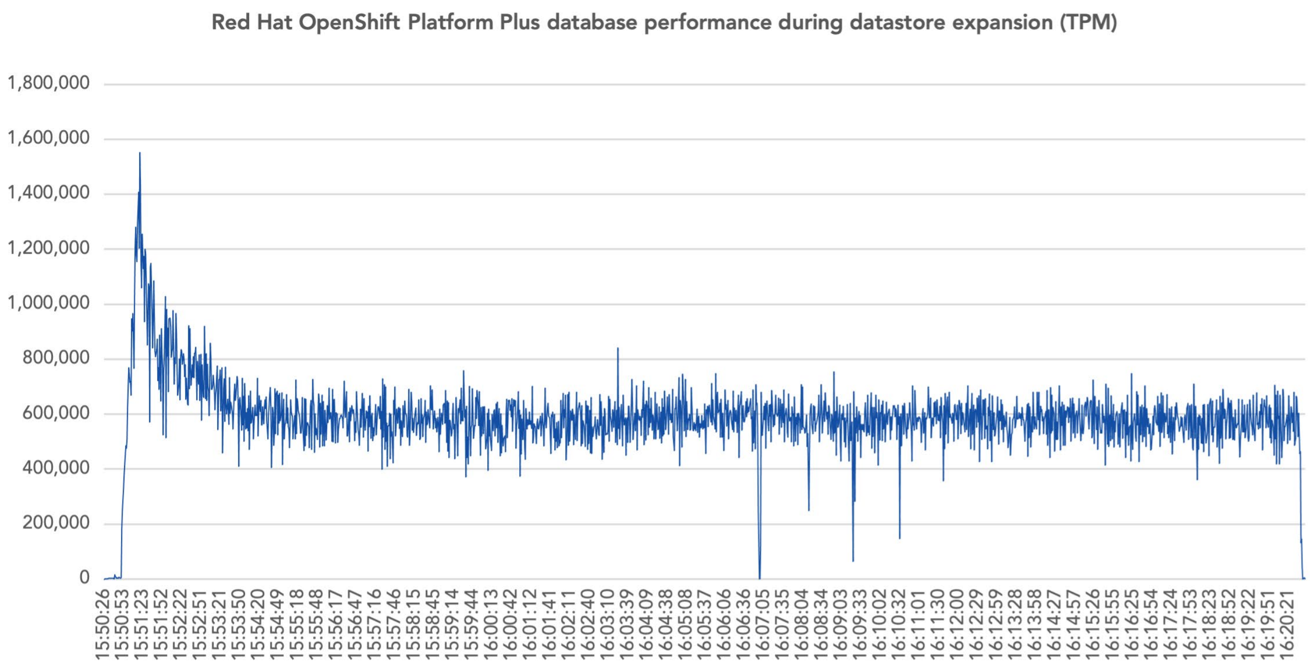

Figure 10: OCP database performance during the datastore expansion. Higher is better. Source: PT.

Storage policy change

Database performance (TPM)

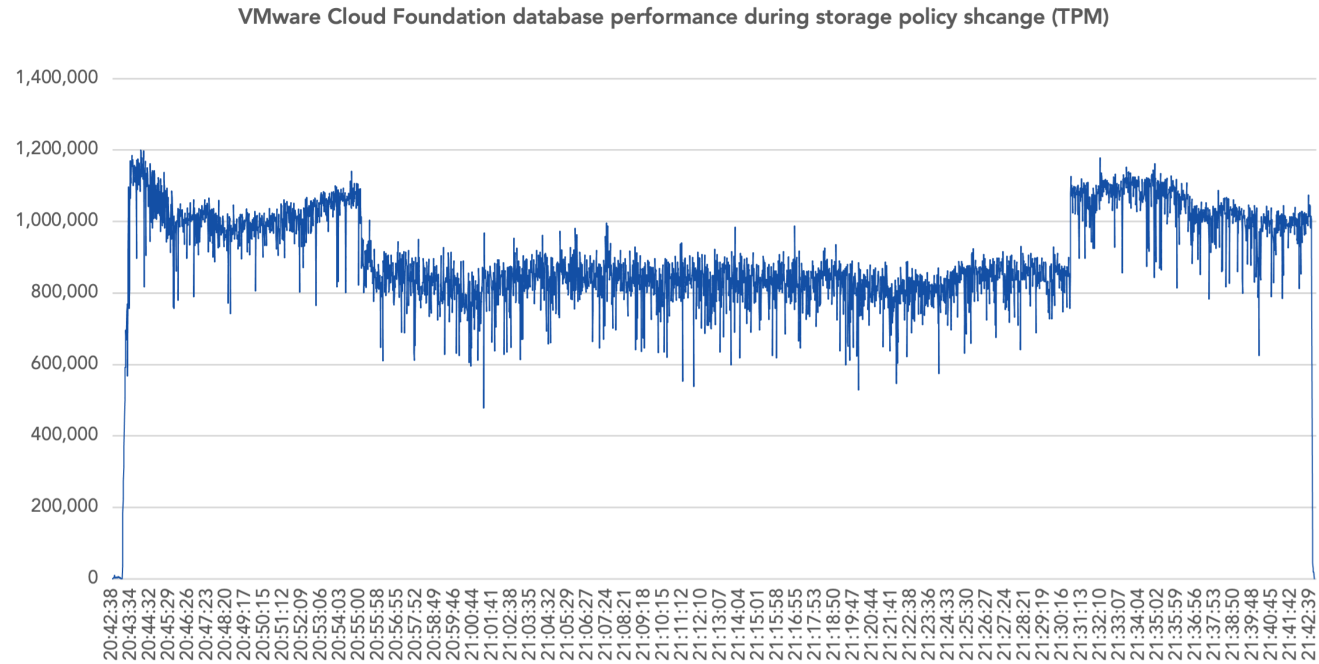

Figure 11: VCF 9 database performance during the storage policy change. Higher is better. Source: PT.

System configuration information

Table 2: Detailed information on the systems we tested.

System configuration information	Dell PowerEdge R640	
	VCF – 4 nodes	OCP – 4 nodes
BIOS name and version	Dell 2.22.2	Dell 2.22.2
Operating system name and version/build number	VMware ESXi 9.0.0, 24755229	OpenShift Container Platform 4.19
Date of last OS updates/patches applied	28 July 2025	28 July 2025
Power management policy	Performance	Performance
Processor		
Number of processors	2	2
Vendor and model	Intel Xeon Platinum 8260	Intel Xeon Platinum 8260
Core count (per processor)	24	24
Core frequency (GHz)	2.4	2.4
Stepping	7	7
Memory module(s)		
Total memory in system (GB)	768	768
Number of memory modules	24	24
Vendor and model	Hynix HMA84GR7CJR4N-WM	Hynix HMA84GR7CJR4N-WM
Size (GB)	32	32
Type	DDR4	DDR4
Speed (MHz)	2,933	2,933
Speed running in the server (MHz)	2,933	2,933
Storage controller 1		
Vendor and model	Dell BOSS-S1	Dell BOSS-S1
Cache size	0	0
Firmware version	2.5.13 3024	2.5.13 3024
Local storage for OS		
Number of drives	1	1
Drive vendor and model	Intel SSDSCKKB240G8R	Intel SSDSCKKB240G8R
Drive size (GB)	240	240
Drive information (speed, interface, type)	6Gbps, M2, SSD	6Gbps, M2, SSD
Storage controller 2		
Vendor and model	Dell HBA330 Mini (Embedded)	Dell HBA330 Mini (Embedded)
Cache size	0	0
Firmware version	16.17.01 00	16.17.01 00

System configuration information	Dell PowerEdge R640	
	VCF – 4 nodes	OCP – 4 nodes
Local storage for distributed storage		
Number of drives	4	4
Drive vendor and model	Dell Express Flash NVMe P4500	Dell Express Flash NVMe P4500
Drive size (TB)	4	4
Drive information (speed, interface, type)	8GT/s, PCIe 3.0, NVMe 1.2	8GT/s, PCIe 3.0, NVMe 1.2
Network adapter 1 (VMs, management)		
Vendor and model	Mellanox ConnectX-4 LX 25GbE SFP	Mellanox ConnectX-4 LX 25GbE SFP
Number and type of ports	2 x 25GbE	2 x 25GbE
FW version	14.32.20.04	14.32.20.04
Network adapter 2 (application and storage)		
Vendor and model	Mellanox ConnectX-5 Ex 100GbE QSFP	Mellanox ConnectX-5 Ex 100GbE QSFP
Number and type of ports	2 x 100GbE	2 x 100GbE
Driver version	16.35.30.06	16.35.30.06
Cooling fans		
Vendor and model	Dell	Dell
Number of cooling fans	8	8
Power supplies		
Vendor and model	Dell 0CMPGM	Dell 0CMPGM
Number of power supplies	2	2
Wattage of each (W)	1100	1100

How we tested

We received two four-node R640 clusters with identical hardware from Broadcom to perform these tests. The VCF cluster was fully set up when we received it, and included vSAN with four P4500 disks per node and cluster networking with one 10Gb connection and one 25Gb connection in a virtual distributed switch with port groups for vMotion, vSAN, VM, and Management traffic. We began our configuration for the VCF use-case testing from VM creation.

We received the OCP cluster after the cluster installation process, and needed to perform additional steps to configure comparable networking and storage. We created a shared storage pool across all four nodes in the OCP cluster using OpenShift Data Foundation and the local storage operator. We then installed OpenShift Virtualization to create and run VMs for testing the use cases. Finally, we changed the MTU on the default network to 8900, and set up a dedicated network for HammerDB traffic on the 10Gb connection (also MTU 8900), so that any live migration traffic would happen on the default openshift virtual network using the 25Gb connection without our HammerDB database workload disconnecting.

The goal of our study was to show that different management tasks can be made easier and done more quickly by using VCF as compared to performing those same tasks on OCP. We tested three different use cases, measuring the time and number of steps to complete each task, as well as the performance impact on a database workload while running the task. We set up a SQL Server VM and a client VM, with Windows Server 2022 and SQL Server 2022, and used HammerDB 5.0 and the TPROC-C benchmark to generate a 10,000 warehouse database to run a transactional workload while running through different management tasks. We ran each scenario three times and report the median time to complete each task.

Configuring the OCP environment

Installing the OpenShift Local Storage Operator

1. Log into the OpenShift Console.
2. Click Operators→OperatorHub.
3. In the search field, type Local Storage, and click Local Storage Operator.
4. Click Install.
5. Click Install.

Installing the OpenShift Data Foundation (ODF) Operator

1. Click Operators→OperatorHub.
2. In the search field, type Data Foundation, and click OpenShift Data Foundation Operator.
3. Click Install.
4. Click Install.

Configuring ODF to create a Local Volume Set

1. Click Storage→Data Foundation.
2. Click the Storage Systems tab.
3. Click Create StorageSystem.
4. Select Full Deployment for the deployment type, and Create a new StorageClass using local storage devices.
5. Click the check-box next to Set default StorageClass for virtualization.
6. Click Next.
7. Enter a name for the LocalVolumeSet, and click Next.
8. Click Yes.
9. Click Next.
10. Click Next.
11. Click Create StorageSystem.

Installing the OpenShift Virtualization Operator

1. Click Operators→OperatorHub.
2. In the search field, type Virtualization, and click OpenShift Virtualization Operator.
3. Click Install.
4. Click Install.

Creating an OpenShift Virtualization hyperconverged instance

1. Click Installed Operators→OpenShift Virtualization.
2. Under OpenShift Virtualization Deployment, click Create Instance.
3. Leave defaults, and click Create.

Creating a bootable volume

1. Click Virtualization→Bootable Volumes.
2. Click Add volume→With form.
3. Under source type, click Upload new volume.
4. Click Upload, navigate to the ISO to upload, and check the This is an iso file checkbox. Select the appropriate storage class, give the volume a name, and click Save.

Creating a virtual machine

1. Click Virtualization→VirtualMachines.
2. Click Create→From template.
3. Click the Windows Server 2022 VM template.
4. Click Customize Virtual Machine.
5. Under Name, click the pencil to give the VM a name.
6. Under CPU | Memory to assign, click the pencil and enter the following:CPU: 48
7. Memory: 192 GB
8. Under Workload profile, click the pencil, and select High performance.
9. Click the Disks tab.
10. Click Add disk→Volume.
11. Check Use this disk as a boot source checkbox.
12. Give the disk a name, and under PersistentVolumeClaim select the bootable volume created above.
13. Click Save.
14. Click Create VirtualMachine.

Installing the NMState Operator

During our testing, we had to take extra steps on the OCP platform to keep HammerDB connected. This is because when OCP creates a new VM to migrate workloads to, it gives the new VM a new IP address. This would cause HammerDB to disconnect, and the HammerDB benchmark is not designed to continue trying to reconnect automatically. In a real-world scenario with critical enterprise workloads, these applications would be designed to handle this scenario without intervention.

1. Click Operators→OperatorHub.
2. In the search field, type NMState, and click the Kubernetes NMState Operator.
3. Click Install.
4. Click Install.

Creating an NMState deployment

1. After the installation is complete, click View operator. Click the NMState tab.
2. Click Create NMState.
3. Click Create.

Creating a Node Network Configuration Policy (NNCP)

1. In the navigation pane, click NodeNetworkConfigurationPolicy.
2. Click Create NodeNetworkConfigurationPolicy.
3. Enter a Policy name, ensure the Policy interface is set to linux bridge br0, and click Next.
4. Enter the port, and click Create.

Creating a Network Attachment Definition

1. In the navigation pane, click NetworkAttachmentDefinitions.
2. Select the project that contains the VMs.
3. Click Create NetworkAttachmentDefinition.
4. Enter a name for the network attachment.
5. From the Network Type drop-down menu, select Linux bridge.
6. For the bridge name, enter br0.
7. Optionally, enter a VLAN tag, and click Create.

Assigning a second NIC to the VMs

1. Click on Virtualization→VirtualMachines.
2. Select the VM to edit.
3. Click the Configuration tab.
4. Click Network.
5. Click Add network interface.
6. Enter a name for the NIC.
7. From the Network drop-down menu, select the network attachment definition created in the previous step, and click Save.
8. Complete steps 1 through 7 for both the client and SQL VMs.

Configuring the VCF environment

Creating a virtual machine

1. Log into the vCenter.
2. Right-click on the cluster→New Virtual Machine…
3. Highlight Create a new virtual machine, and click Next.
4. Give the VM a name, and click Next.
5. Select the cluster for the compute resource, and click Next.
6. Select the vSAN datastore, and click Next.
7. Set the VM compatibility to ESXi 9.0 and later, and click Next.
8. Set the VM Guest OS version to Microsoft Windows Server 2022 (64-bit), and click Next.
9. Customize the following VM hardware:
10. CPU: 48
11. Memory: 192 GB
12. New Hard Disk: 100 GB
13. Click Next.
14. Click Finish.

Configuring the test environment

We created a SQL Server VM for our VM under test, and a client VM to run HammerDB and collect results with the steps below.

Installing Windows Server 2022 on a VM

1. Attach the Windows Server 2022 ISO (boot volume for OCP) to the VM.
2. Open the VM console, and start the VM.
3. When prompted to boot from DVD, press any key.
4. On the installation screen, leave language, time/currency format, and input method as default, and click Next.
5. Click Install now.
6. Enter the product key when prompted.
7. Select Windows Server 2022 Datacenter Edition (Desktop Experience), and click Next.
8. Check the box to accept the license terms, and click Next. Choose Custom: Install Windows only (advanced).
9. Select Drive 0 Unallocated Space, and click Next to begin installation.
10. After installation completes and the system restarts, set a password on the Settings page, and log in.
11. Install VMware Tools in the VM (if hosted on ESXi), or the QEMU guest agent and VirtIO driver (if hosted on OCP).
12. From Server Manager, disable Windows Firewall.
13. Run Windows Updates to ensure the system is up to date.

After installing Windows, added eight 150GB disks for database data, one 300GB disk for database logs, one 1TB disk for database backups, one 100GB disk for tempDB data, and one 50GB disk for templogs on the SQL VM. We initialized and formatted them with default settings in Disk Management before installing SQL Server.

Installing SQL Server

1. Attach the SQL Server 2022 installation media ISO to the VM.
2. Double-click the Setup application.
3. In the left pane, click Installation.
4. Click New SQL Server stand-alone installation or add features to an existing installation.
5. On the Edition window:
    a. Keep Specify a Free Edition checked.
    b. Select Evaluation, and click Next.
6. Accept the license terms by checking the box, and click Next.
7. Click Use Microsoft Update to check for updates, and click Next.
8. On the Install Rules page, click Next.
9. On the Azure Extension for SQL Server page, deselect the extension, and click Next.
10. On the Feature Selection screen, check:
    a. Database Engine Services
    b. Full-Text and Semantic Extractions for Search
11. Click Next.
12. Leave the default instance selected, and click Next.
13. Leave the default Service Accounts, and click Next.
14. On the Server Configuration tab:
    a. Choose Mixed Mode.
    b. Enter and confirm a password for the sa account.
    c. Click Add Current User to specify SQL Server administrators.
15. On the TempDB tab:
    a. Set TempDB location to the TempDB volume.
    b. Set TempLog location to the TempLog volume.
16. Click Next.
17. On the Error and usage reporting screen, click Next.
18. On the Installation Configuration Rules screen, verify no failures or warnings, and click Next.
19. On the Ready to Install screen, click Install.
20. Close the installation window when complete.
21. Download and install SQL Server Management Studio (SSMS).
22. Open SSMS and run the following SQL Queries to update SQL Server for HammerDB:

```
exec sp_configure 'show advanced options', '1'
reconfigure with override
exec sp_configure 'recovery interval','32767'
exec sp_configure 'max degree of parallelism','1'
exec sp_configure 'lightweight pooling','1'
exec sp_configure 'priority boost','1'
exec sp_configure 'max worker threads','3000'
exec sp_configure 'default trace enabled','0'
go
reconfigure with override
```

23. Create a database for HammerDB TPROC-C.

```
CREATE DATABASE [tpcc]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'tpcc01', FILENAME = N'E:\data\tpcc01.mdf' , SIZE = 8192KB , FILEGROWTH = 65536KB ),
( NAME = N'tpcc02', FILENAME = N'F:\data\tpcc02.ndf' , SIZE = 8192KB , FILEGROWTH = 65536KB ),
( NAME = N'tpcc03', FILENAME = N'G:\data\tpcc03.ndf' , SIZE = 8192KB , FILEGROWTH = 65536KB ),
( NAME = N'tpcc04', FILENAME = N'H:\data\tpcc04.ndf' , SIZE = 8192KB , FILEGROWTH = 65536KB ),
( NAME = N'tpcc05', FILENAME = N'I:\data\tpcc05.ndf' , SIZE = 8192KB , FILEGROWTH = 65536KB ),
( NAME = N'tpcc06', FILENAME = N'J:\data\tpcc06.ndf' , SIZE = 8192KB , FILEGROWTH = 65536KB ),
( NAME = N'tpcc07', FILENAME = N'K:\data\tpcc07.ndf' , SIZE = 8192KB , FILEGROWTH = 65536KB ),
( NAME = N'tpcc08', FILENAME = N'L:\data\tpcc08.ndf' , SIZE = 8192KB , FILEGROWTH = 65536KB )
```

```
 LOG ON
( NAME = N'tpcc_log', FILENAME = N'M:\logs\tpcc_log.ldf' , SIZE = 8192KB , FILEGROWTH = 65536KB )
 WITH LEDGER = OFF
GO
ALTER DATABASE [tpcc] SET COMPATIBILITY_LEVEL = 160
GO
ALTER DATABASE [tpcc] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [tpcc] SET ANSI_NULLS OFF
GO
ALTER DATABASE [tpcc] SET ANSI_PADDING OFF
GO
ALTER DATABASE [tpcc] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [tpcc] SET ARITHABORT OFF
GO
ALTER DATABASE [tpcc] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [tpcc] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [tpcc] SET AUTO_CREATE_STATISTICS ON(INCREMENTAL = OFF)
GO
ALTER DATABASE [tpcc] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [tpcc] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [tpcc] SET CURSOR_DEFAULT  GLOBAL
GO
ALTER DATABASE [tpcc] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [tpcc] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [tpcc] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [tpcc] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [tpcc] SET  DISABLE_BROKER
GO
ALTER DATABASE [tpcc] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [tpcc] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [tpcc] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [tpcc] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [tpcc] SET  READ_WRITE
GO
ALTER DATABASE [tpcc] SET RECOVERY FULL
GO
ALTER DATABASE [tpcc] SET  MULTI_USER
GO
ALTER DATABASE [tpcc] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [tpcc] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [tpcc] SET DELAYED_DURABILITY = DISABLED
GO
USE [tpcc]
GO
IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE is_default=1 AND name = N'PRIMARY') ALTER
DATABASE [tpcc] MODIFY FILEGROUP [PRIMARY] DEFAULT
GO
```

Installing HammerDB 5.0

1. Download the latest version of HammerDB from http://www.hammerdb.com/download.html.
2. Double-click the .exe file.
3. Choose English, and click OK.
4. Click Yes. Click Next.
5. Choose a destination location, and click Next.
6. Click Next.
7. Click Finish to complete the installation.

Generating a 10,000-warehouse database

1. Open powershell.
2. Navigate to the HammerDB directory.

```
cd 'C:\Program Files\HammerDB-5.0'
```

3. Run the hammerdb cli.

```
.\hammerdbcli.exe
```

4. Set the appropriate DB build options.

```
dbset db mssqls
diset connection mssqls_server [IP Address or FQDN]
diset connection mssqls_tcp true
diset connection mssqls_authentication sql
diset connection mssqls_pass [Password]
diset tpcc mssqls_count_ware 10000
diset tpcc mssqls_num_vu 96
diset tpcc mssqls_dbase tpcc
diset tpcc mssqls_raiseerror true
loadscript
```

5. Build the schema

```
buildschema
```

Once the schema build finished, we took a backup of the database with the `backup_db.sql` script in the Scripts we used for testing section to restore from in between runs.

Running the tests

For each use case, we dropped the existing database and restored a fresh database from a backup using the `drop_and_restore.sql` script in the Scripts we used for testing section.

Use case #1: Upgrading VMs

For this use case, we set the SQL VM's number of vCPUs to 8 and the RAM to 64 GB. We edited the `tpcc.tcl` file and changed the following options:

```
mssqls_rampup 10
mssqls_duration 20
```

We started performance monitor counters on the SQL VM and kicked off our test with run-test.ps1. At 15 minutes, we started a timer and ran the steps for VCF and OCP below to hot-add CPU and RAM to the VM. After adding RAM and CPU, we ran the following SQL query so SQL Server could see the newly added resources, and to verify that they were added:

```
RECONFIGURE WITH OVERRIDE;
GO
SELECT * FROM sys.dm_os_schedulers;
GO
```

Once we verified that SQL could see the hot-added CPUs, we stopped the timer. We took an average of the TPM for five minutes during steady states before and after the upgrade, and ran this test three times on each cluster, reporting the median of the three timings.

Adding resources in VCF

1. Right click the VM→Edit settings.
2. Add CPU & RAM, and click OK.
3. Adding resources in

OCP

1. Click Virtualization→VirtualMachines.
2. Click the Virtual Machine.
3. Click the Configuration tab.
4. Under CPU | Memory, click the pencil to edit the VM.
5. Add CPU & Memory, and click Save.

Use case #2: Expanding the datastore/storage pool

For this use case, we set the SQL VM's number of vCPUs to 48 and the RAM to 192 GB. We edited the `tpcc.tcl` file in the Scripts we used for testing section and changed the following options:

```
mssqls_rampup 10
mssqls_duration 20
```

We started performance monitor counters on the SQL VM and kicked off our test with run-test.ps1. At 15 minutes, we started a timer and ran the steps below to expand the storage for each cluster. We stopped the timer once we finished with the steps and verified the disks were added. We ran this test three times on each cluster, collecting the average TPM for five minutes during a steady-state before and after expanding the storage, and reporting the median of the three timings.

Expanding storage in VCF

1. Click the cluster.
2. Click the Configure tab.
3. Under vSAN, click Disk Management.
4. Click Claim Unused Disks.
5. Click the check box to claim all the disks, and click Create.

Expanding storage in OCP

1. Click Storage.
2. Click Data Foundation.
3. Click the storage system tab.
4. Click the action menu → next to the storage system.
5. Click Add capacity.
6. Select the storage class associated with the unused disks.
7. Click Add.

Use case #3: Changing storage policy on default storage pool

For this use case, we set the SQL VM's number of vCPUs to 48 and the RAM to 192 GB. We edited the `tpcc.tcl` file in the Scripts we used for testing section and changed the following options:

```
mssqls_rampup 15
mssqls_duration 45
```

We started performance monitor counters on the SQL VM and kicked off our test with run-test.ps1. At 15 minutes, we started a timer and ran the steps below to change the vSAN storage policy for the VMs disks from RAID 1 to RAID 5. We stopped the timer once we finished with the steps and vSAN finished resyncing all the objects. We ran this test three times, collecting the average TPM for five minutes during a steady-state before and after expanding the storage, and reporting the median of the three timings.

Changing storage policy with VCF

1. Right-click the VM→Edit Settings.
2. Next to Hard disks, click the drop-down menu. For each hard disk, click the drop-down next to Storage policy and change from Raid1 to Raid5 with compression.

Changing storage policy with OCP

Because we set up the storage using all four disks on each node in the default storage pool like we did with our vSAN setup on VCF, we weren't able to compare this use case on OCP because the default storage pool setup by ODF doesn't allow you to edit the pool after it has been created. In order to compare, we would need additional disks to create a second storage pool, or we would need to create a new storage class with the desired storage policy, and then copy data from our SQL VM's existing disks to the new disks using that storage class. Due to hardware limitations we were unable to add more disks, and the second option of a new storage class isn't comparable since you can't do it in-place like on VCF.

Scripts we used for testing

backup_db.sql

```
BACKUP DATABASE [tpcc] TO  DISK = N'Z:\backup\tpcc.bak' WITH NOFORMAT, NOINIT,  NAME = N'tpcc-Full
Database Backup', SKIP, NOREWIND, NOUNLOAD, COMPRESSION,  STATS = 10
GO
```

drop_and_restore.sql

```
EXEC msdb.dbo.sp_delete_database_backuphistory @database_name = N'tpcc'
GO
use [master];
GO
USE [master]
GO
ALTER DATABASE [tpcc] SET  SINGLE_USER WITH ROLLBACK IMMEDIATE
GO

GO
USE [master]
GO
DROP DATABASE [tpcc]
GO

USE [master]
RESTORE DATABASE [tpcc] FROM  DISK = N'Z:\backup\tpcc.bak' WITH  FILE = 1,  NOUNLOAD,  STATS = 5

GO
```

tpcc.tcl

```
#!/bin/tclsh

dbset db mssqls
diset connection mssqls_server [IP Address or FQDN]
diset connection mssqls_tcp true
diset connection mssqls_authentication sql
diset connection mssqls_pass [Password]
diset tpcc mssqls_count_ware 10000
diset tpcc mssqls_num_vu 96
diset tpcc mssqls_dbase tpcc
diset tpcc mssqls_rampup 15
diset tpcc mssqls_duration 45
diset tpcc mssqls_raiseerror true
loadscript

vuset vu 96
vuset logtotemp 1
vuset showoutput 1
vuset unique 0

tcset refreshrate 1
tcset logtotemp 1
tcset unique 0
tcset timestamps 1

tcstart
vucreate
vurun
vudestroy
after 2500
tcstop
```

copy-results.ps1

```
$run="C:\Users\Administrator\Documents\results\test"
$hdb="C:\Users\Administrator\AppData\Local\Temp\2"
$remotePerf="\\[SQL VM IP Address]\C$\PerfLogs\Admin\hammerdb\DataCollector01.blg"

mkdir $run

cp $hdb\hdbtcount.log $run\
cp $hdb\hammerdb.log $run\

Copy-Item -Path "$remotePerf" -Destination $run

rm $hdb\hdbtcount.log
rm $hdb\hammerdb.log
```

run-test.ps1

```
$hdbDir="C:\Program Files\HammerDB-5.0"
$resultsDir="C:\Users\Administrator\Documents\results"

cd "$hdbDir"
.\hammerdbcli.exe auto tpcc.tcl

Start-Sleep -s 360

cd $resultsDir
.\copy-results.ps1
```

Read the report ▶

This project was commissioned by Broadcom.

Principled Technologies®

Facts matter.®