



The science behind the report:

Deliver better return on investment and faster data analysis while performing image classification tasks with Dell Technologies APEX Private Cloud compared to a set of comparable Amazon EC2 instances

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 [Deliver better return on investment and faster data analysis while performing image classification tasks with Dell Technologies APEX Private Cloud compared to a set of comparable Amazon EC2 instances](#).

We concluded our hands-on testing on 11/11/2021. 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 09/22/2021 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: Time (hours) required to process a 1TB dataset during the TPROC-H data warehouse analysis workload from HammerDB. Both solutions had 8 vCPUs, 32 GB of memory, and 2.5 TB of storage (VMware® vSAN™ storage for Dell Technologies APEX Private Cloud and Elastic Block Storage [EBS] gp2 storage for the AWS solution).

Solution	Time (hours)
Single APEX Private Cloud VM	10.5
Single AWS EC2 m5.2xlarge instance	40

Table 2: Query processing rate (queries per second) during the ResNet50 image classification workload. Each solution had 1 NVIDIA® T4 GPU, 4 vCPU cores, 16 GB of memory, and 1 TB of storage (vSAN storage for the APEX Private Cloud solution and EBS gp2 storage for the AWS solution).

Solution	Processing rate (queries per second)
Single APEX Private Cloud VM	4,701
Single AWS EC2 g4dn.xlarge instance	4,850

On the APEX Private Cloud solution, we also performed the image classification tests concurrently with the database HammerDB workload. We performed the hour-long image classification tests three different times at 5-hour intervals (5, 10 and 15 hours) during the data warehousing test. Tables 3 and 4 show the performance results for these concurrent workloads.

Table 3: Time required to process a 1TB dataset on twelve APEX Private Cloud VMs during a TPROC-H data warehouse analysis workload from HammerDB. Each VM had 8 vCPUs, 32 GB of memory, and 2.5 TB of vSAN storage.

APEX Private Cloud - Database VM	Time
Database VM1	17h 8m
Database VM2	15h 33m
Database VM3	16h 25m
Database VM4	16h 53m
Database VM5	15h 45m
Database VM6	16h 27m
Database VM7	14h 53m
Database VM8	14h 46m
Database VM9	14h 11m
Database VM10	16h 5m
Database VM11	15h 35m
Database VM12	15h 25m

Table 4: Query rate (queries per second) achieved by the nine APEX Private Cloud VMs during a ResNet50 image classification workload. Each VM had one NVIDIA T4 GPU, four vCPU cores, 16 GB of memory, and 1 TB of vSAN storage.

APEX Private Cloud - Machine learning VM	Query rate (queries per second)
ML VM1	4,701
ML VM2	4,701
ML VM3	4,701
ML VM4	4,701
ML VM5	4,701
ML VM6	4,701
ML VM7	4,701
ML VM8	4,701
ML VM9	4,701
Total rate	42,309

Total cost of ownership estimate analysis

Table 5: Results of our total cost of ownership (TCO) analysis. We based our calculations on three-year, no-money-upfront plans, and assumed 100% utilization (730 hours per month). Note: We tested Amazon EC2 g4dn.xlarge instances for machine learning workloads and m5.2xlarge instances for database workloads.

	Dell Technologies APEX Private Cloud 3 accelerator-optimized nodes ¹	Amazon EC2 service US East - N. Virginia ²
Machine learning VMs (x9)³		
Compute (EC2 Instance Savings Plan)	–	\$1,491.39
Elastic Block Storage - gp2	–	\$921.60
Database VMs (x12)³		
Compute (EC2 Instance Savings Plan)	–	\$4,677.84
Elastic Block Storage - gp2	–	\$3,072.00
Additional on-premises costs		
Monthly support ⁴	Included	\$1,011.40
Solution cost per month	\$5,090.00	\$11,174.23
Total solution cost (3 years)	\$183,240.00	\$402,272.28
Software cost for Windows Server OS (3 years) ⁵	\$96,941.25	Included
Facilities costs - Data center space, power, and cooling (3 years) ^{6,7}	\$24,075.09	N/A
Monthly maintenance cost (3 years) ⁸	\$7,607.52	N/A
Totals/comparison		
Total cost (3 years)	\$311,863.86	\$402,272.28
3-Year savings with APEX Private Cloud	\$90,408.42	
Price advantage for APEX Private Cloud	22%	

- On June 9, 2021, Dell provided us a pricing quote that included a Dell VxRail™ cluster, power distribution, switches, racks, solution deployment, ongoing support from Dell and solution recovery at the end of the subscription.
- Cost on AWS EC2 does not account for specific workload use patterns, data transfer costs, or additional storage options. “AWS Pricing Calculator,” accessed January 26, 2022, <https://calculator.aws/#/estimate?id=634afcd05edc4ab8022d5156621323d163cd140>
- The machine learning VMs had 1 GPU, 4 vCPUs, 16 GB of RAM, and 1,024 GB of storage. The twelve database VMs had 8 vCPUs, 32 GB of RAM, and 2,560 GB of storage.
- Support on APEX Private Cloud includes ProSupport Plus for Enterprise with Mission Critical support. Support on AWS includes the Business support plan.
- Windows Server licensing cost based on three nodes, each with two processors and 24 cores per processor. This requires nine 16-core license bundles. Total license costs: \$55,395. We estimate yearly support costs at 25 percent of the total three-year license cost: \$13,848 per year, or \$41,546.25 for three years.
- We calculated the \$10,189.38 3-year power and cooling cost for three nodes using input power estimates from the Dell Enterprise Infrastructure Planning Tool at <https://dell-ui-eipt.azurewebsites.net/#/>. We calculated 5,877.96 kWh per node per year for the tested server configuration running a transactional workload at 100% utilization for 730 hours per month. We based cooling costs on an estimate of 1.7 PUE, and based the a cost of \$0.1133 per kWh on the U.S. commercial average cost from Electric Power Monthly at https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a, accessed January 26, 2022.
- Our data center facilities cost estimate accounts for three 2U nodes, two 1U production switches, and a single 1U management switch. We estimate this would cost \$385.71 per month, or \$13,885.71 for three years assuming facilities cost \$200 per square foot per month. At that rate, a standard 42U server rack that takes up nine square feet of space would cost \$1,800 per month, or \$42.86 per U.
- Maintenance cost over three years covers updates and maintenance up to four hours per month. We based this cost on a Systems Administrator I salary and benefits at \$99,326 per year. Hourly cost of \$52.83 based on 47 work-weeks per year and 40 hours of work per week. “Systems Administrator I,” accessed January 26, 2022, <https://www.salary.com/tools/salary-calculator/systems-administrator-i-benefits>.

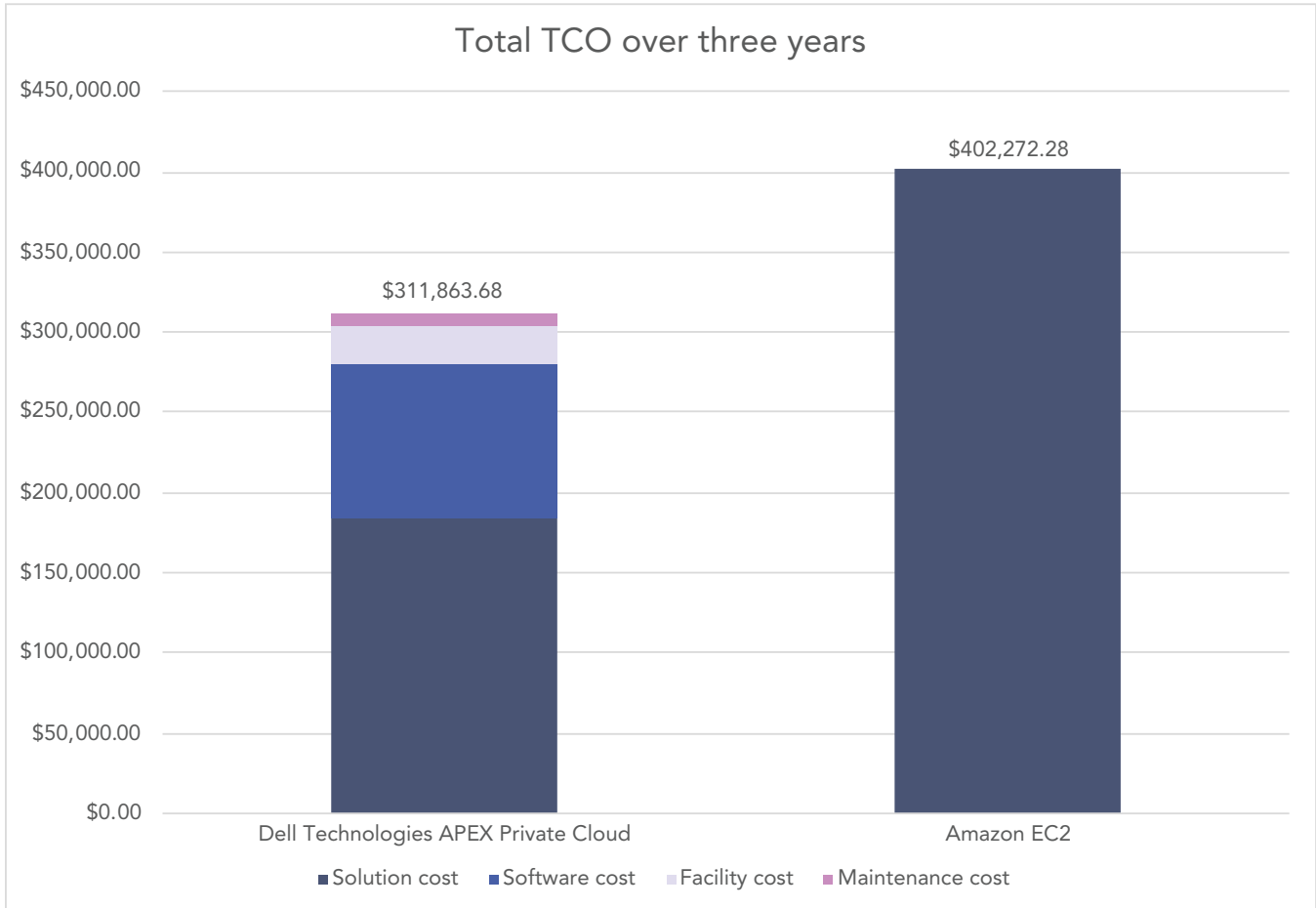


Figure 1: Comparison of estimated three-year total cost of ownership for the APEX Private Cloud and Amazon AWS EC2 solutions we tested.

System configuration information

Table 6: Detailed configuration information for the computational cluster under test.

Server configuration information	3 x Dell EMC VxRail V570F
BIOS name and version	Dell 2.10.2
BIOS settings	Default
Operating system name and version/build number	VMware ESXi™, 7.0.2, 17867351
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 6248R
Core count (per processor)	24
Core frequency (GHz)	3.00
Stepping	Model 85 Stepping 7
Memory module(s)	
Total memory in system (GB)	768
Number of memory modules	12
Vendor and model	N/A
Size (GB)	64
Type	DDR-4
Speed (MHz)	2,933
Speed running in the server (MHz)	2,933
Storage controller	
Vendor and model	Dell HBA330
Cache size (GB)	0
Firmware version	16.17.01.00
Local storage (type A)	
Number of drives	2
Drive vendor and model	Intel SSDSCKKB240G8R
Drive size (GB)	240
Drive information (speed, interface, type)	6Gbps, SATA, SSD
Local storage (type B)	
Number of drives	8
Drive vendor and model	Micron MTFDDAK1T9TDD
Drive size (GB)	1,900
Drive information (speed, interface, type)	6Gbps, SATA, SSD

Server configuration information		3 x Dell EMC VxRail V570F
Distributed storage		
Storage type	vSAN	
Number of nodes	3	
Number of disk groups	9	
Number of disks per disk group	6 (in six disk groups), 7 (in three disk groups)	
Network adapter		
Vendor and model	Mellanox ConnectX-4 LX	
Number and type of ports	2 x 25GbE	
Firmware version	14.28.45.12	
Cooling fans		
Number of cooling fans	6	
Power supplies		
Number of power supplies	2	
Wattage of each (W)	2,000	

Table 7: Configuration details for the Amazon EC2 instances we used to perform database test runs on AWS.

		Amazon EC2 m5.2xlarge instance
General information		
Date testing ended	10/1/21	
Cloud service provider	Amazon Web Services (AWS)	
Region	us-east-1	
Cloud VM or instance details		
Number of VMs	1	
VM or instance size	m5.2xlarge	
BIOS name and version	Amazon EC2 1.0	
Number of vCPUs	8	
Number of cores/threads	24/48	
Memory (GB)	32	
Processor vendor and model	Intel Xeon Platinum 8259CL	
Network interface(s)	1 x ENA, default settings	

Amazon EC2 m5.2xlarge instance	
Operating system information	
Image name and UUID	Windows_Server-2019-English-Full-Base-2021.07.14, ami-0f25b344af3f73199)
OS	Microsoft Windows Server 2019
OS build number	17763.2114
Kernel version	1809
Date of patches last applied	07/30/21
Changes made from CSP image	No
Instance storage (volume type 1)	
Number of volumes	1
Volume use in this test	OS
CSP volume type	EBS gp2
Volume size (GB)	180
Encryption type	Not encrypted
Instance storage (volume type 2)	
Number of volumes	1
Volume use in this test	Data
CSP volume type	EBS gp2
Volume size (GiB)	2,380
Encryption type	Not encrypted

Table 8: Configuration details for the Amazon EC2 instances we used for test MLPerf runs on AWS.

Amazon EC2 g4dn.xlarge instance	
General information	
Date testing ended	9/9/21
Cloud service provider	AWS
Region	us-east-1
Cloud VM or instance details	
Number of VMs	1
VM or instance size	g4dn.xlarge
BIOS name and version	Amazon EC2 1.0
Number of vCPUs	4
Number of cores/threads	24/48
Memory (GB)	16
Processor vendor and model	Intel Xeon Platinum 8259CL
Network interface(s)	1 x ENA, 5.11.0-1017-aws

Amazon EC2 g4dn.xlarge instance	
Operating system information	
Image name and UUID	ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20210430
OS	Ubuntu 20.04.3 LTS
Kernel version	5.11.0-1017-aws
Date of patches last applied	9/2/21
Changes made from CSP image	No
Instance storage (volume type 1)	
Number of volumes	1
Volume use in this test	OS
CSP volume type	EBS gp2
Volume size (GB)	8
Encryption type	Not encrypted
Instance storage (volume type 2)	
Number of volumes	1
Volume use in this test	Data
CSP volume type	EBS gp2
Volume size (GiB)	1,024
Encryption type	Not encrypted

How we tested

Preparing the APEX servers and VMs for ML testing

1. Download the NVIDIA vGPU package for vSphere 7 and Linux (NVIDIA-GRID-vSphere-7.0-460.91.03-462.96.zip) from the following URL: <https://www.nvidia.com/download/index.aspx>.
2. Obtain the IP Address for the local NVIDIA license server. See the documentation in the ZIP file for details (460.91.03-462.96-grid-licensing-user-guide.pdf).
3. Extract the vSphere and Linux installation files:

```
unzip NVIDIA-GRID-vSphere-7.0-460.91.03-462.96.zip \  
NVD-VGPU_460.91.03-1OEM.700.0.0.15525992_18308486.zip \  
NVIDIA-Linux-x86_64-460.91.03-grid.run
```

4. For each of the three ESXi servers, perform the following steps:
 - a. Use vCenter to place the ESXi server into maintenance mode.
 - b. Copy the vSphere vGPU package to the ESXi server:

```
ssh NVD-VGPU_460.91.03-1OEM.700.0.0.15525992_18308486.zip root@A.B.C.D:/tmp/
```

- c. Log onto the ESXi server, and install the vGPU package:

```
ssh root@A.B.C.D \  
esxcli software vib install -v \  
/tmp/ NVD-VGPU_460.91.03-1OEM.700.0.0.15525992_18308486.zip
```

- d. Use vCenter to exit maintenance mode.
- e. Reboot the server.
- f. To set the default VM graphics to Shared Direct, perform the following steps:
 - i. From vCenter, Select the Configure tab for the ESXi server
 - ii. Select Host Graphics under the Graphics entry
 - iii. Select Edit
 - iv. Select Shared Direct and click OK.
 - v. Reboot the ESXi server.
- g. Disable ECC on the vCPU:

```
ssh root@A.B.C.D nvidia-smi -e 0
```

5. Create and prepare the first ML VM.
 - a. Use vCenter to create the VM with four vCPUs, 16 GB of RAM, a 60GB OS volume, and a 1TB data volume. The volumes are created on VXRail shared storage.
 - b. Migrate the VM to the appropriate server (see step 6).
 - c. To configure the VM to use one vCPU, perform the following steps:
 - i. On the VM, Select Edit Settings.
 - ii. In the New device menu, select Shared PCI Device, and click Add.
 - iii. In the GPU Profile menu, select grid_t4-4q.
 - d. Install Ubuntu 20.04 to the OS volume, using the default options.
6. Repeat step 5 to create a total of nine VMs, being sure to place only 3 VMs per ESXi server.

Installing NVIDIA vGPU software on each APEX VM

1. Transfer the Linux vGPU software to the VM:

```
scp NVIDIA-Linux-x86_64-460.91.03-grid.run ubuntu@J.K.L.M
```

2. Log onto the VM as the ubuntu user.
3. Install the vGPU software:

```
sudo apt-get install -y gcc make linux-headers-$(uname -r)
chmod a+rx NVIDIA-Linux-x86_64-460.91.03-grid.run
sudo ./NVIDIA-Linux-x86_64-460.91.03-grid.run
```

4. For the APEX VM only, configure the NVIDIA license file using the following steps:
 - a. Edit the file `/etc/nvidia/gridd.conf`
 - b. Add the IP address of your license server to the `ServerAddress` line.
 - c. Set the `FeatureType` to 1.
5. Restart the vGPU daemon:

```
systemctl restart nvidia-gridd
```

6. Format and mount the data disc:

```
sudo parted /dev/sdb
sudo mkfs.xfs /dev/sdb1
sudo mkdir -p /mnt/mlperf
sudo mount /dev/sdb1 /mnt/mlperf
sudo chown ubuntu /mnt/mlperf
```

7. Add the prerequisite software:

```
sudo apt update
sudo apt upgrade
sudo install git
wget https://get.docker.com
mv index.html dock_inst.sh
sudo sh ./dock_inst.sh
sudo systemctl --now enable docker
```

8. Add the user to docker group:

```
sudo usermod -aG docker $USER
```

9. Move the docker data directory to the data disk:

```
sudo systemctl stop docker
sudo mv /var/lib/docker /mnt/mlperf/;sudo ln -s /mnt/mlperf/docker /var/lib
sudo systemctl start docker
```

10. Install the NVIDIA container toolkit:

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \  
  && curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add - \  
  && curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list \  
  | sudo tee /etc/apt/sources.list.d/nvidia-docker.list \  
sudo apt update \  
sudo apt install nvidia-docker2 \  
sudo systemctl restart docker
```

Preparing the AWS instance for ML testing and installing the NVIDIA vGPU software

1. Deploy one g4dn.xlarge instance with Ubuntu 20.04.
2. Attach a 1TB volume to the instance.
3. Log onto the instance.
4. Format and mount the data disc:

```
sudo parted /dev/sdb \  
sudo mkfs.xfs /dev/sdb1 \  
sudo mkdir -p /mnt/mlperf \  
sudo mount /dev/sdb1 /mnt/mlperf \  
sudo chown $USER /mnt/mlperf
```

5. Install the prerequisite packages:

```
sudo apt update \  
sudo apt upgrade linux-aws \  
sudo apt install aws-cli gcc make linux-headers-$(uname -r) git \  
sudo shutdown -r reboot
```

6. Log onto the instance, and use the AWS cli to authenticate.
7. Download the vGPU software:

```
aws s3 cp s3://ec2-linux-nvidia-drivers/grid-12.3/NVIDIA-Linux-x86_64-460.91.03-grid-aws.run . \  
chmod a+rx NVIDIA-Linux-x86_64-460.91.03-grid-aws.run \  
sudo /bin/sh NVIDIA-Linux-x86_64-460.91.03-grid-aws.run \  
sudo reboot
```

8. On the vCPU, disable ECC:

```
nvidia-smi -e 0
```

Installing MLPerf 0.7 inference benchmark on each VM or instance

1. Log onto the VM as the default user.
2. Download MLPerf 0.7 inference benchmark:

```
cd ~
git clone https://github.com/mlperf/inference_results_v0.7.git
mkdir /mnt/mlperf/scratch
export MLPERF_SCRATCH=/mnt/mlperf/scratch
```

3. Download or copy a previously downloaded archive of the ResNet50 image data. The data is available from <https://www.kaggle.com/c/imagenet-object-localization-challenge/overview/description>.
4. Extract the image data:

```
mkdir -p /scratch/mlperf/data/imagenet
cd /scratch/mlperf/imagenet
unzip ~/imagenet-object-localization-challenge.zip
```

5. Create the NVIDIA container:

```
cd ~/inference_results_v0.7/closed/NVIDIA
make pre_build
```

6. Configure the benchmark for one T4 vGPU:

- a. Edit file `configs/resnet50/Server/config.json`. In the T4x1 stanza, change the value assigned to `server_target_qps` to the target value; for example, 4701 or 4900.
- b. Edit the file `vi code/common/system_list.py`, and add the following line to the head of the `system_list` array:

```
("T4x1", "Tesla T4-4Q", 1),
```

7. Download the ML model, and preprocess the image data:

```
make download_model BENCHMARKS=resnet50
make preprocess_data BENCHMARKS=resnet50
```

Running the MLPerf 0.7 inference test

1. Log onto the VM/instance as the default user.
2. Prepare the benchmark for the run:

```
cd ~/inference_results_v0.7/closed/NVIDIA
make clean
make build BENCHMARKS=resnet50
make generate_engines RUN_ARGS="--benchmarks=resnet50 --scenarios=Server --config_ver=default"
```

3. Run the benchmark:

```
make run_harness RUN_ARGS="--benchmarks=resnet50 --scenarios=Server --config_ver=default --test_mode=AccuracyOnly"
```

Creating the Microsoft SQL Server 2019 VM in VMware vCenter

1. In VMware vCenter, navigate to Virtual Machines.
2. To create a new VM, click the icon.
3. Leave Create a new virtual machine selected, and click Next.
4. Enter a name for the VM, and click Next.
5. Place the VM on the desired host with available CPUs, and click Next.
6. Select the appropriate datastore to host the VM, and click Next.
7. Select the appropriate guest OS, and click Next.
8. In the Customize Hardware section, use the following settings:
9. Set the vCPU count to 8
10. Set the Memory to 32 GB
11. Add one 180GB VMDK for the OS and one 2380GB VMDK for database files, logs, and backup
12. Attach the OS ISO to the CD/DVD drive.
13. Click Next.
14. Click Finish.
15. Repeat steps 1-10 to create 11 more matching VMs.

Creating the SQL Server VM in AWS

1. Log into AWS, and navigate to the AWS Management Console.
2. Click EC2.
3. Click Launch instance, and to open the Launch Instance in the dropdown wizard, click Launch instance.
4. In the search window, type `Windows Server`, and press enter.
5. Next to Microsoft Windows Server 2019 Base on Quick Start, click Select .
6. On Choose Instance Type, select m5.2xlarge, and click Next: Configure Instance Details.
7. On Configure Instance, set the following:
 - a. Number of instances: 1
 - b. Purchasing option: Leave unchecked
 - c. Network: Default
 - d. VPCSubnet: Choose the region you are working in
 - e. Auto-assign Public IP: EnablePlacement
 - f. Group: Leave unchecked
 - g. Capacity Reservation: Open
 - h. Domain join directory: No Directory
 - i. IAM role: None
 - j. Shutdown behavior: Stop
 - k. Click Next: Add Storage and on Add Storage, set the following:
 - l. Size: 180GB, 2380GB
 - m. Volume Type: gp2
 - n. Delete on Termination: Checked
 - o. Encryption: Not Encrypted
 - p. Click Next: Add Tags
8. On Add Tags, add any tags you wish to use. Click Next: Configure Security Group.
9. On Configure Security Group, leave defaults, and click Review and Launch.
10. On Review, click Launch.
11. Choose the appropriate option for the key pair, and click Launch Instances.

Installing Microsoft Windows Server 2019

We performed the following setup steps on each SQL Server VM created in vCenter and AWS.

1. Attach the Windows Server 2019 ISO to the VM.
2. Open the VM console, and start the VM.
3. When the system prompts you to boot from DVD, press any key.
4. When the installation screen appears, leave language, time/currency format, and input method as default, and click Next.
5. Click Install now.
6. When the installation prompts you, enter the product key.
7. Select Windows Server 2019 Datacenter Edition (Server with a GUI), and click Next.
8. Check I accept the license terms, and click Next.
9. Click Custom: Install Windows only (advanced).
10. Select Drive 0 Unallocated Space, and click Next. This starts Windows automatically, and Windows will restart automatically after completing.
11. When the Settings page appears, fill in the Password and Reenter Password fields with the password you set up previously.
12. Install VMware Tools in the VMs hosted on the ESXi servers.
13. From Server Manager, disable Windows Firewall.
14. Run Windows Updates.

Installing SQL Server 2019

1. Attach the installation media ISO for SQL Server 2019 to the VM.
2. Click Run SETUP.EXE. If Autoplay does not begin the installation, navigate to the SQL Server 2019 DVD, and double-click it.
3. In the left pane, click Installation.
4. Click New SQL Server stand-alone installation or add features to an existing installation.
5. Specify Evaluation as the edition you are installing, and click Next.
6. To accept the license terms, click the checkbox, and click Next.
7. Click Use Microsoft Update to check for updates, and click Next.
8. At the Feature Selection screen, select Database Engine Services, Full-Text and Semantic Extractions for Search, Client Tools Connectivity, and Client Tools Backwards Compatibility.
9. Click Next.
10. At the Instance configuration screen, leave the default selection of default instance, and click Next.
11. At the Server Configuration screen, accept defaults, and click Next.
12. At the Database Engine Configuration screen, select the authentication method you prefer. For our testing purposes, we selected Mixed Mode.
13. Enter and confirm a password for the system administrator account.
14. Click Add Current user. This may take several seconds.
15. Click Next.
16. At the Ready to Install screen, click Install.
17. Close the installation window.
18. In the SQL Server Installation Center, click Install SQL Server Management Tools.
19. Click Download SQL Server Management Studio.
20. Click Run.
21. When the Microsoft SQL Server Management Studio screen appears, click Install.
22. When the Installation completes, click Close.

Locking pages in memory

1. Click Start, and type `Local Security Policy`. Open the program when it pops up in the search.
2. Expand Local Policies, and click User Rights Assignment.
3. In the right-hand pane, scroll down, and double-click Lock pages in memory.
4. Click Add User or Group, type `NT Service\MSSQLSERVER`, and click OK.
5. To close the Properties window, click OK, and close the Local Security Policy window.

Installing HammerDB 4.2

1. Download HammerDB from <https://hammerdb.com/download.html>.
2. Double-click the .exe file, choose English, and click OK.
3. Click Yes.
4. Click Next.
5. Choose a destination location, and click Next.
6. Click Next.
7. Click Finish.

Setting the SQL Server memory reserve and max degree of parallelism (MAXDOP)

1. Open the SQL Server Management Studio.
2. Right-click the SQL Instance, and click Properties.
3. Click Advanced node, scroll down to the Max Degree of Parallelism, and change the value. Click OK.
4. Right-click the SQL Instance again, and go to Memory.
5. Set the Max Memory to 90 percent of the total memory in the system. Click OK, and close the Properties window.
6. Right click the SQL Instance, and restart the service. Click Yes when prompted.

Configuring the tempdb database

1. Open the SQL Server Management Studio.
2. Expand Databases and System databases, and right click tempdb.
3. Add files, and change the starting size as necessary.
4. Right-click the SQL Instance, and restart the service. When prompted, click Yes.
5. To move the tempdb to the database drive, open a new query, and run the following for the number of tempdb files your system has:

```
USE [master]
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = tempdev, FILENAME = 'E:\TempDB\tempdb.mdf')
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = temp2, FILENAME = 'E:\TempDB\tempdb_mssql_2.ndf')
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = temp3, FILENAME = 'E:\TempDB\tempdb_mssql_3.ndf')
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = temp4, FILENAME = 'E:\TempDB\tempdb_mssql_4.ndf')
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = temp5, FILENAME = 'E:\TempDB\tempdb_mssql_5.ndf')
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = temp6, FILENAME = 'E:\TempDB\tempdb_mssql_6.ndf')
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = temp7, FILENAME = 'E:\TempDB\tempdb_mssql_7.ndf')
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = temp8, FILENAME = 'E:\TempDB\tempdb_mssql_8.ndf')
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = templog, FILENAME = 'E:\TempDB\templog.ldf')
GO
```

6. Right click the SQL Instance, and restart the service. When prompted, click Yes.

Running the tests

We performed the following steps on the AWS SQL Server VM. We also ran this HammerDB test simultaneously across all vCenter SQL Server VMs.

1. On the VM you're testing, restore the database under test.
2. Make sure you've configured your SQL settings and tempdb according to the instructions in the previous section and to the instance on which you're running on the test.
3. Open HammerDB.
4. Select Options→Benchmark.
5. Choose MSSQL Server and TPROC-H.
6. Expand SQL Server → TPROC-H → Schema Build.
7. Double-click Options, change the driver to ODBC Driver 17 for SQL Server, set the scale to match your database, set MAXDOP to match SQL's, and check the box for Clustered Columnstore. Click OK.
8. Expand Driver Script, double-click Options, and click OK to load.
9. Expand Virtual User, and double-click Options.
10. Choose one user.
11. Select Show Output, Log Output to Temp, and Use Unique Log Name.
12. Click OK.
13. To load the driver script, double-click Load.
14. Double-click Create users.
15. To capture performance metrics on the system, start Performance monitor set to record CPU, Memory, and drive usage information.
16. To begin the run, click Start.
17. When the run finishes, stop Perfmon, and save the HammerDB results file and Perfmon output.
18. Stop the HammerDB user.
19. Double-click User options, and set the number of users to the appropriate count for the multi-stream test.
20. Double-click Create users.
21. To capture performance metrics on the system, start Performance monitor set to record CPU, Memory, and drive usage information.
22. To begin the run, click Start on HammerDB.

Run the MLPerf 0.7 inference test on nine VMs

We performed the following steps at the 5, 10, and 15 hour marks of the HammerDB runs on the APEX setup. The ESXi servers, VMware VMs are configured as described in the ML testing preparation sections at the beginning of this methodology.

1. Log onto the auxiliary system.
2. Start remote sessions to the VMs using screen for automation. The list of IP addresses for the VMs is in the second and third lines.

```
screen
hosts="192.168.100.11 192.168.100.12 192.168.100.13 \
      192.168.100.14 192.168.100.15 192.168.100.16 \
      192.168.100.17 192.168.100.18 192.168.100.19
for host in $hosts; do
  screen ssh $host
done
```

3. Prepare the benchmarks for the run

```
for i in {1..9}; do
  screen -p $i -X stuff '^Mcd ~/inference_results_v0.7/closed/NVIDIA^M'
  screen -p $i -X stuff '^Mmake attach_docker
done
sleep 30
for i in {1..9}; do
  screen -p $i -X stuff
  '^Mmake clean; make build BENCHMARKS=resnet50; make generate_engines RUN_ARGS="--
benchmarks=resnet50 --scenarios=Server --config_ver=default"^M'
done
sleep 75
```


4. Check each screen session has completed.
5. Run the benchmark

```
for i in {1..9}; do
  screen -p $i -X stuff \
    '^Mmake run_harness RUN_ARGS="--min_duration=600000 --max_duration=2400000
--benchmarks=resnet50 --scenarios=Server --config_ver=default --test_mode=PerformanceOnly"^M'
done
```

Read the report at <http://facts.pt/IOpaHrs> ►

This project was commissioned by Dell Technologies.



Facts matter.®

Principled Technologies is a registered trademark of Principled Technologies, Inc. All other product names are the trademarks of their respective owners.

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:

Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.