HITACHI

Reference Architecture Guide

Hybrid Cloud DevTest Solution for VMware Production Site

MK-SL-396-00

Hitachi Vantara

Table of Contents

Notices and Disclaimer

About This Document

Introduction

Companies that develop business applications and systems are constantly working to improve product quality and respond to customer requests. As the pace of change in society and customer needs accelerates, companies must quickly develop and improve their business applications to keep up. Efficient development and continuous improvement are key to a company's business success.

In general, using copies of production data currently in use for application and system development and testing allows for rapid and efficient development and testing. The following are four reasons for this:

- It reduces the time and effort required to create test data and allows for comprehensive testing that includes a wide range of cases compared with artificial test data.
- It replicates the actual execution environment of the application, leading to early bug detection and an improved user experience.
- It enables faster application improvements based on end-user feedback.
- It minimizes the effort and errors involved in creating dummy data. By using snapshots of production data copies, you can quickly deploy more accurate and realistic development and testing environments. Snapshots can be obtained quickly from the base volume.

This solution allows you to enjoy these benefits by using copies of production data for development and testing.

With this solution, instead of preparing development and testing servers and storage on-premises as in the past, the servers and storage are prepared in the cloud.

This offers the following benefits:

- Preparing hardware equipment such as servers and storage requires numerous tasks, including estimating hardware requirements, planning and approving budgets, ordering equipment, and managing transportation, installation, and configuration. By using the cloud, the lead time for infrastructure preparation is significantly reduced.
- When introducing servers or storage, hardware maintenance such as firmware updates are required. By using the cloud, hardware maintenance is handled by the cloud vendor, reducing operational overhead.
- When deploying expensive hardware such as servers and storage, the initial investment (CAPEX) is high. By using the cloud, these upfront costs can be reduced, and the costs for virtual machines and storage can be managed as operational expenses (OPEX).
- When conducting development and testing using cloud-native block storage (such as Amazon EBS, Azure Disk Storage, or Google Cloud Persistent Disk), it is necessary to create and use the actual cloud block storage from snapshots for each virtual machine used in development and testing, resulting in high usage costs. This issue becomes more significant as the number of development and testing virtual machines increases, reducing capacity efficiency. By using VSP One SDS Cloud, which supports compression and snapshots, you can improve capacity efficiency and reduce costs.

Hitachi Vantara's hybrid cloud development and testing solution helps enterprises achieve business success.

Intended Audience

This document is intended for CIOs, CFOs, infrastructure managers, developers, and testers at companies that develop business applications and systems.

Accessing product downloads

Product software, drivers, and firmware downloads are available on Hitachi Vantara Support Connect: https://support.hitachivantara.com/.

Log in and select Product Downloads to access the most current downloads, including updates that may have been made after the release of the product.

Getting Help

Hitachi Vantara Support Connect is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to Hitachi Vantara Support Connect for contact information: https://support.hitachivantara.com/en_us/contact-us.html.

Hitachi Vantara Community is a global online community for customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to community.hitachivantara.com, register, and complete your profile.

Document Revisions

Revision Number	Date	Details
v1.0	October 2025	Initial release

Comments

Send any comments on this document to Docs-Feedback@hitachivantara.com. Include the document title, including the revision level, and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Thank you.

Solution Overview

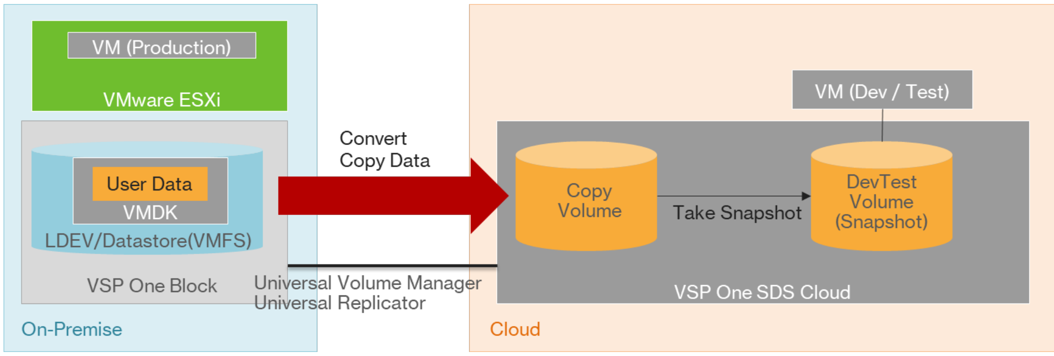

Figure 1 - Solution Overview

With this solution, you can develop and test production workloads hosted on on-premises VMware vSphere in the cloud.

As shown in Figure 1, the on-premises Hitachi VSP One Block serves as storage for the Datastore that contains production workloads. The Hitachi VSP One SDS Cloud, running on Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure, serves as the storage for copies of production data used for development and testing in the cloud. Production data from this copied production data in the VSP One SDS Cloud, and these snapshots are attached to development and testing virtual machines for use in development and testing operations. You can quickly deploy development and testing environments by using these snapshots of production data copies.

The VMware VMDK files that you use are converted to the Logical Unit (LU) format provided by Hitachi VSP One Block, using the method described in the following section. As a result, you do not need to modify the existing VMware production environment.

Hitachi Vantara offers two key storage solutions, Hitachi VSP One Block, an on-premises enterprise block storage platform, and Hitachi VSP One SDS Cloud, a cloud-based software-defined storage (SDS) solution, to build a hybrid cloud development and testing infrastructure that delivers the benefits described above. These storage solutions provide Universal Volume Manager and Universal Replicator, which enable seamless data integration and mobility between on-premises and cloud environments. Additionally, as mentioned earlier, VSP One SDS Cloud supports compression and snapshots, offering a cost-efficient and scalable solution.

This solution enables rapid, flexible, and efficient development and testing workflows. Furthermore, by using VSP One SDS Cloud, a cloud-based SDS, this solution supports deployment tools such as Terraform and Ansible, thereby reducing infrastructure preparation lead times, lowering initial costs, minimizing operational overhead, and improving capacity efficiency compared with traditional cloud-native development and testing solutions.

Key Solution Components

Hitachi VSP One Block

The Hitachi Virtual Storage Platform One Block series simplifies system setup and management through the new VSP 360 management offering. Dynamic Drive Protection reduces RAID complexity, and always-on compression and deduplication enhance simplicity.

Dynamic Carbon Reduction optimizes energy usage by switching CPUs to ECO mode during low activity. Adaptive Data Reduction (ADR) is always on, enhancing efficiency and reducing the overall CO2 footprint.

Thin Image Advanced (TIA) integrates with major snapshot ecosystems, prioritizing security by defending against threats and ensuring data confidentiality. CyberArk Privileged Access Manager plugins enhance block storage system security by prioritizing data confidentiality, ensuring compliance, and actively defending against security threats.

- Hitachi Virtual Storage Platform One Block includes the following 3 dedicated models:
- VSP One Block 24 - 256 GB Cache + SW Advanced Data Reduction (ADR) + 24 cores

- VSP One Block 26 - 768 GB Cache + 2x Compression Accelerator Module (CAM) + 24 cores
- VSP One Block 28 - 1 TB Cache + 4x CAM + 64 cores

All have the same drive count (72 NVMe flash drives, the appliance, and 2 × media trays) and they support Fibre Channel, iSCSI, and NVMe TCP connectivity. The new capabilities remove complexity such as data reduction always being on, Dynamic Drive Protection removes complicated RAID setup, and Dynamic Carbon Reduction delivers real-world reduction in power consumption. In addition, the models are FIPS compliant.

In short, the Hitachi Virtual Storage Platform One Block series combines simplicity, sustainability, and robust security features to optimize system management, energy efficiency, and data protection.

See https://www.hitachivantara.com/en-us/products/storage-platforms/block-storage/midrange/vsp-one-block for more information.

Hitachi VSP One SDS

Hitachi Virtual Storage Platform One SDS (VSP One SDS) is designed as a flexible, software-defined storage solution that can be deployed across different environments.

It offers the following deployment options and data protection methods:

- Bare Metal configuration uses general-purpose x86 servers to build a virtual storage system. Ideal for on-premises environments needing high performance and capacity with low initial cost.
- Cloud-Based configuration that is deployable in public cloud environments such as Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP), offering scalability and agility for hybrid cloud strategies.
- Hitachi Polyphase Erasure Coding (HPEC), a proprietary method that provides high capacity efficiency and fault tolerance.
- Mirroring duplication stores a full copy of user data on another node for redundancy and quick recovery.
- VSP One SDS offers unified management with centralized control of multiple storage nodes as a single system using REST APIs. You can easily add or reduce storage capacity, and dynamic scaling without service disruption is supported. It can be deployed on-premise or in the cloud to support traditional and modern databases, applications, virtual machines, and containers everywhere. It is available as cloud, block, and software-defined storage.

Rather than choosing between fixed hardware models, decision-makers can tailor VSP One SDS deployments based on performance, capacity, and resiliency needs – whether on bare metal, in the cloud, or as part of a hybrid strategy.

Hitachi Thin Image Advanced (Hitachi VSP One Block)

Thin Image Advanced (TIA) enables you to perform cost-effective replication by storing only the differential data between primary volumes (P-VOLs) and secondary volumes (S-VOLs).

TIA stores snapshots of Hitachi Virtual Storage Platform (VSP) storage systems. If a logical data failure occurs in the storage system due to erroneous data update or virus infection, you can restore it using the stored snapshot of the data.

For more information, see https://www.hitachivantara.com/content/dam/hvac/pdfs/datasheet/thin-image-snapshotsoftware-datasheet.pdf.

Hitachi Universal Volume Manager (UVM)

Universal Volume Manager (UVM) enables you to connect volumes in external storage systems to Hitachi Virtual Storage Platform (VSP) storage systems, and manage those volumes as if they were one system.

Typically, if a system consists of multiple storage systems, the host must be connected to each of the storage systems. When configuring the connections from the host to the volumes, the system administrator must follow specific instructions for each of the storage systems.

With UVM, the administrator configures the connection from the host to your storage system, and then uses mapped volumes in an external storage system in the same way as volumes in the local storage system.

External volumes can be used in situations such as the following:

- Backing up target VSP storage volumes to an external storage system.
- Using the capacity of external storage system through the target VSP.
- Migrating data from legacy external storage system to the new target VSP.

Hitachi Universal Replicator (UR)

Hitachi Universal Replicator is an asynchronous remote copy function based on Hitachi's proprietary technology.

By storing journal (change history) data on a disk and then transferring it to a remote site, UR creates replica volumes on a separate disk array at the remote site without involving the server (host). This process maintains consistency with the main site's primary volume in a non-synchronous manner. The created replicas are fully physical volumes.

By storing journal (change history) data on a disk and then transferring it to a remote site, update processing at the remote site is performed asynchronously with the main site's update operations while preserving the update order. This ensures logical data consistency by maintaining the sequence of update reflections at the remote site.

By recording journal (change history) data on disk before transmission, stable long-distance data replication is achieved even across IP networks with fluctuating bandwidth or during business traffic fluctuations that exceed line capacity.

VMware `vmkfstools`

VMware `vmkfstools` is an ESXi Shell command used for managing VMFS volumes, storage devices, and virtual disks. You can perform a wide range of storage operations using the `vmkfstools` command. For example, you can create and manage VMFS datastores on a physical partition or manipulate virtual disk files stored on VMFS or NFS datastores.

You can also convert a VMDK file to a SAN-based disk using Raw Device Mapping (RDM).

Solution Design

Figure 2 shows a high-level infrastructure diagram of this solution.

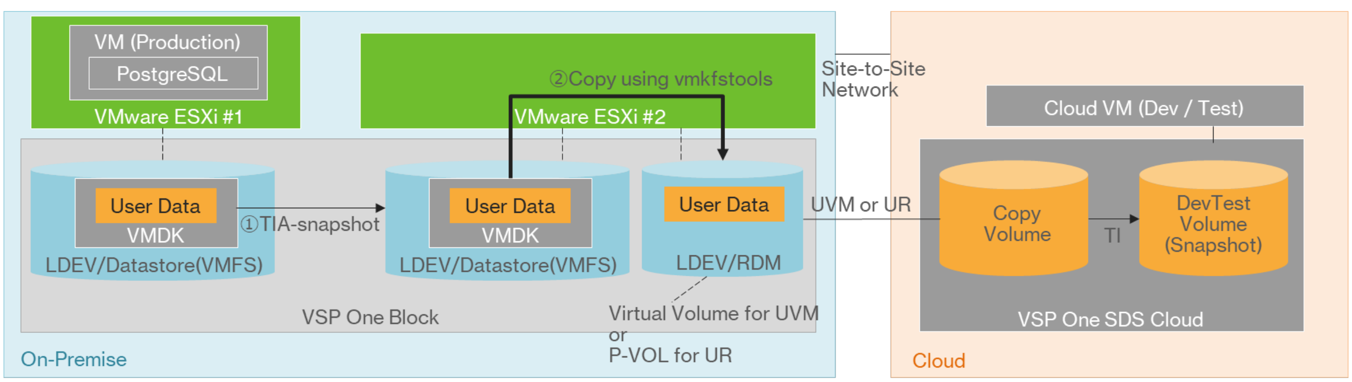

Figure 2 High-Level Infrastructure

This solution supports two configurations.

One configuration uses Hitachi Universal Volume Manager (UVM) and the other using Hitachi Universal Replicator (UR). In both configurations, VMware ESXi #1/#2, the Datastore, and VSP One SDS Cloud perform the same roles.

The on-premises Hitachi VSP One Block stores production data for production operations.

VMware ESXi #1 on HA800 #1 hosts production virtual machines. VMware ESXi #2 on HA800 #2 is used to copy production data (using `vmkfstools`, as described later). VMware vCenter Server manages the VMware ESXi servers.

VSP One SDS Cloud stores copies of production data. Snapshots provided by VSP One SDS Cloud are accessed by cloud-based virtual machines.

Cloud-based virtual machines are used for development and testing. These virtual machines are Amazon EC2 instances in the case of AWS, and Virtual Machines in the case of Azure or Google Cloud.

A site-to-site network is used to connect on-premises and cloud environments and transfer production data from on-premises to the cloud.

- For AWS, you can use AWS Direct Connect or AWS Site-to-Site VPN.
- For Azure, you can use Azure ExpressRoute or Azure VPN Gateway.
- For Google Cloud, you can use Cloud Interconnect or Cloud VPN.

In this solution, the standard VMware ESXi tool `vmkfstools` is used to copy (convert) a VMDK file containing production data to a SAN-based disk (LDEV) using Raw Device Mapping (RDM). By using `vmkfstools`, data can be copied from on-premises to the cloud solely with VMware ESXi and Hitachi VSP One Block functionality, without requiring third-party tools, thereby simplifying the architecture and reducing operational complexity.

The VMware VMDK files used by users are converted into the LU format provided by Hitachi VSP One Block, so users do not need to modify their existing VMware production environment. In Scenario 1 of this solution, the UVM feature is used to map the virtual volume within VSP One Block (Virtual Volume for UVM in the figure) to the Copy Volume within VSP One SDS Cloud.

These virtual volumes can be used as volumes within VSP One Block, even though they are actually located within VSP One SDS Cloud. Since these virtual volumes do not consume drive capacity within VSP One Block, drive capacity can be saved. However, it is important to note that the data transfer rate for a single UVM multiplex decreases as the distance increases.

In Scenario 2 of this solution, the UR feature is used to copy data from a volume within VSP One Block (P-VOL for UR in the figure) to a Copy Volume within VSP One SDS Cloud (UR S-VOL). UR supports long-distance asynchronous data transfer and does not have the distance constraints of the architecture in Scenario 1. However, note that the UR P-VOL consumes disk capacity within Hitachi VSP One Block equal to the total capacity of user data within VMDK files to be copied to UR S-VOL. In this scenario, there is also the advantage that database data masking can be performed on the on-premises side, because the volume entities are located there.

Data masking is the process of replacing original characters or numbers with different ones to conceal the original database data from developers and testers in DevTest environments.

A comparison of Scenario 1 and Scenario 2 is shown in Table 1. Users can select either configuration depending on their requirements.

Item	Scenario 1 (UVM)	Scenario 2 (UR)
On-premises storage capacity	✔No additional capacity required	✖Additional capacity required (same capacity as the production Datastore capacity)
Transfer distance restrictions	✖UVM has transfer distance limitations	✔UR supports long-distance synchronous transfers
Data masking on the on-premises side	✖Data masking is not possible on the on-premises side	✔Data masking is possible on the on-premises side

Table 1 - Comparison between Scenario 1 and Scenario 2

Engineering Validation

Hardware Components

The hardware components used in this solution are listed in Table 2.

Component	Configuration
Storage	Hitachi VSP One Block B20 Series

Table 2 - Hardware Components

Software Components

The software components used in this solution are listed in Table 3.

Component	Configuration
Hitachi VSP One SDS Cloud Software Version	AWS: 01.17.00.30 Azure: 01.18.00.50 Google Cloud: 01.18.00.60
VMware vSphere	8.0.1
Operating system of on-premises and cloud VMs	Ubuntu 24.04
File system of on-premises and cloud VMs	ext4
PostgreSQL version	v16.9

Table 3 - Software Components

Validation Scenarios

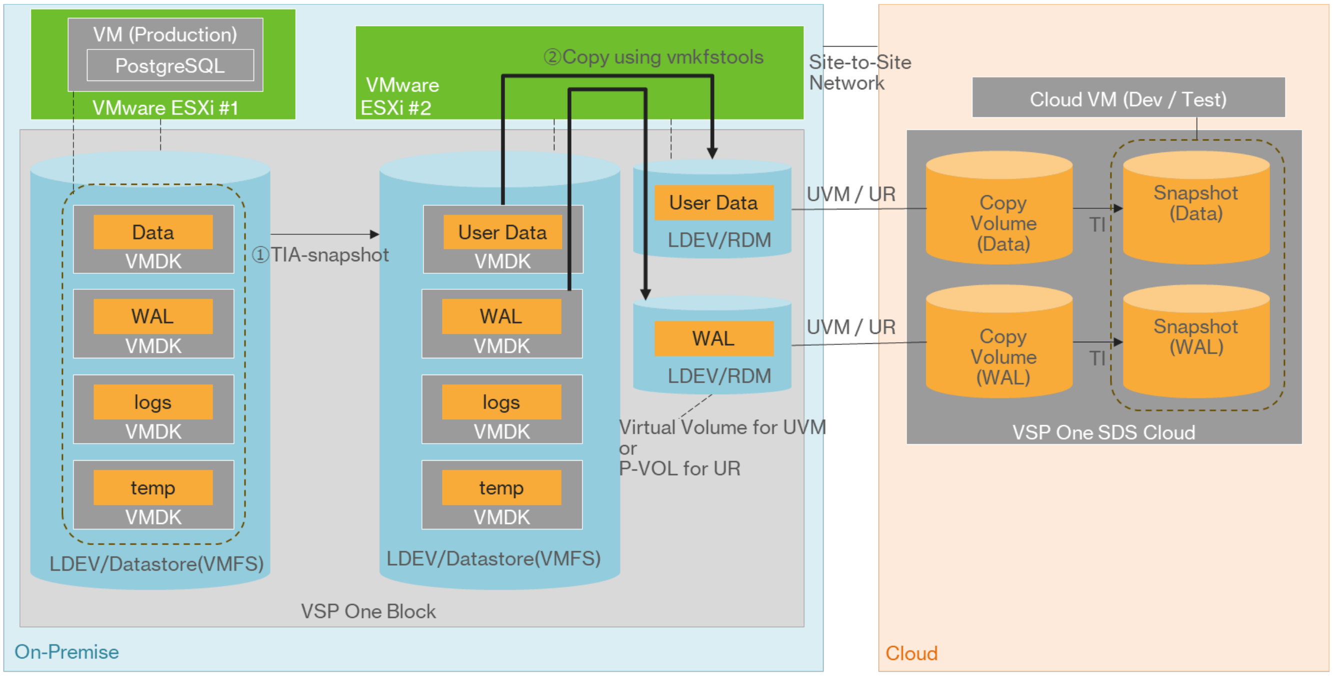

Figure 3 Configuration of Test Scenarios

This white paper examines two scenarios. Scenario 1, which uses a configuration with UVM, whereas Scenario 2, which uses a configuration with UR (Figure 3).

Both scenarios use Ubuntu as the operating system and PostgreSQL as the middleware. PostgreSQL uses four directories: Data, WAL, logs and temp, which store database table data, database transaction logs, PostgreSQL logs, and PostgreSQL temporary data, respectively. In this verification, these directories were stored in ext4 file systems created on separate VMDK files, and these VMDK files were attached to the production virtual machine.

The verification flow for Scenarios 1 and 2 is listed in Table 4.

When backing up PostgreSQL, the backup mode based on the `pg_backup_start` command was used. Workload generation was performed using `pgbench`, a simple program for running PostgreSQL benchmark tests: [https://www.postgresql.org/docs/current/pgbench.html]. After generating workload for a period, a TIA-Snapshot was acquired while continuing the workload. In the UVM-based scenario, the VMDK files within the Snapshot were copied to VSP One SDS Cloud using `vmkfstools`, followed by acquisition of Snapshots of those volumes. In the UR-based scenario, the VMDK files within the Snapshot were copied to volumes (P-VOLs) within VSP One Block. And then, the volumes were copied to volumes (S-VOLs) within the cloud environment, followed by acquisition of Snapshots of those volumes. In both these scenarios, only PostgreSQL Data and WAL directories were copied; logs and temp directories were excluded.

For PostgreSQL recovery on the cloud side, the Snapshot volume containing the Data and WAL directories was first connected using iSCSI, and the file system was repaired using `fsck.ext4`.

Because the snapshot is taken while the VM is still running, the ext4 file system on that copy is “crash-consistent” rather than cleanly unmounted. Some journal and metadata updates may still reside in memory. Before mounting the volume elsewhere, `fsck.ext4` must replay the journal and run an integrity check so that the file system is returned to a clean, consistent state.

And then file system was mounted, and the PostgreSQL service was started to verify the database.

The success criteria for database verification were as follows:

- Tables can be read using SELECT statements.
- The entire database can be retrieved using a database dump.
- No database corruption was detected using `pg_amcheck`.

As a result of the database verification, it was confirmed that database backup, copy, and recovery operations functioned normally in both Scenarios 1 and 2.

Step	Scenario 1 (UVM)	Scenario 2 (UR)
1	Suspend production operations. Or, if using a DBMS, start backup mode.	Same as Scenario 1
2	Use the snapshot function of Hitachi Thin Image Advanced (TIA) to acquire a snapshot of the LDEV corresponding to VMware ESXi #1.	Same as Scenario 1
3	If backup mode was started in the DBMS, exit backup mode.	Same as Scenario 1
4	Scan the acquired snapshot (LU) from VMware ESXi #2.	Same as Scenario 1
5	On VMware ESXi #2, resign and mount the Datastore found by the scan. This allows VMware ESXi #2 to recognize the VMDK file (replica of production data).	Same as Scenario 1
6	Log in to VMware ESXi #2 using SSH.	Same as Scenario 1
7	On VMware ESXi #2, run `vmkfstools` to copy (convert) the VMDK file to a UVM virtual volume based on RDM.	On VMware ESXi #2, run `vmkfstools` to copy (convert) the VMDK file to a volume (UR P-VOL) based on RDM.
8	N/A	Copy the UR P-VOL data to UR S-VOL using Universal Replicator.
9	Unmount the aforementioned Datastore on VMware ESXi #2.	Same as Scenario 1
10	Acquire a snapshot (Hitachi Thin Image) of the Copy Volume in VSP One SDS Cloud.	Same as Scenario 1
11	Make the Cloud VM recognize the snapshot.	Same as Scenario 1
12	Mount the file system.	Same as Scenario 1
13	Start the PostgreSQL service.	Same as Scenario 1
14	Verify the database on the Cloud VM.	Same as Scenario 1

Table 4 - Table 4 Validation Flow

Evidence for Step #1

```
sudo -u postgres psql
psql (16.9 (Ubuntu 16.9-0ubuntu0.24.04.1))
Type "help" for help.

postgres=# SELECT pg_backup_start('hitachi', true);
 pg_backup_start
-----------------
 F/22000060
(1 row)

postgres=# \! pgbench -U prod -i -s 100 mydb
Password:
dropping old tables...
creating tables...
generating data (client-side)...
10000000 of 10000000 tuples (100%) done (elapsed 15.03 s, remaining 0.00 s)
vacuuming...
creating primary keys...
done in 19.67 s (drop tables 0.05 s, create tables 0.02 s, client-side generate 15.13 s, vacuum 0.20 s, primary keys 4.27 s).
postgres=# \! pgbench -U prod -c 20 -j 4 -T 300 mydb
Password:
pgbench (16.9 (Ubuntu 16.9-0ubuntu0.24.04.1))
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 100
query mode: simple
number of clients: 20
number of threads: 4
maximum number of tries: 1
duration: 300 s
number of transactions actually processed: 1722012
number of failed transactions: 0 (0.000%)
latency average = 3.483 ms
initial connection time = 110.462 ms
```



```
tps = 5741.877860 (without initial connection time)
```

Evidence for Step #2

```
Before create snapshot
raidcom get snapshot -snapshotgroup db1
SnapShot_name                  P/S   STAT  Serial# LDEV#  MU# P-LDEV#  PID    % MODE SPLT-TIME
db1                            P-VOL PAIR   800104     2    3      -    1    - G--A -
create snapshot
raidcom modify snapshot -snapshotgroup db1 -snapshot_data create
sleep 60
raidcom map snapshot -ldev_id 0x0002 0x0004 -mirror_id 3
sleep 5
After create snapshot
raidcom get snapshot -snapshotgroup db1
SnapShot_name                  P/S   STAT  Serial# LDEV#  MU# P-LDEV#  PID    % MODE SPLT-TIME
db1                            P-VOL PSUS   800104     2    3      4    1    - G--A 68b5a714
```


Used script for Step #2

```
Write-Host "Before create snapshot"
Write-Host "raidcom get snapshot -snapshotgroup db1"
raidcom get snapshot -snapshotgroup db1
Write-Host "create snapshot"
Write-Host "raidcom modify snapshot -snapshotgroup db1 -snapshot_data create"
raidcom modify snapshot -snapshotgroup db1 -snapshot_data create
Write-Host "sleep 60"
sleep 60
Write-Host "raidcom map snapshot -ldev_id 0x0002 0x0004 -mirror_id 3"
raidcom map snapshot -ldev_id 0x0002 0x0004 -mirror_id 3
Write-Host "sleep 5"
sleep 5
Write-Host "After create snapshot"
Write-Host "raidcom get snapshot -snapshotgroup db1"
raidcom get snapshot -snapshotgroup db1
```


Evidence for Step #3

```
postgres=# SELECT pg_backup_stop();
NOTICE:  WAL archiving is not enabled; you must ensure that all required WAL segments are copied through other means to complete the backup
                          pg_backup_stop
---------------------------------------------------------------------------
 (12/66D01768,"START WAL LOCATION: F/22000060 (file 000000010000000F00000022)+
 CHECKPOINT LOCATION: F/22000098                                    +
 BACKUP METHOD: streamed                                            +
 BACKUP FROM: primary                                               +
 START TIME: 2025-09-01 04:52:18 UTC                                +
 LABEL: hitachi                                                     +
 START TIMELINE: 1                                                  +
 ","16389 /mnt/temp                                                 +
 ")
(1 row)
```

Evidence for Step #14

```
azureuser@hc-devtest-sdsc02-io-host:~$ sudo -u postgres psql -d mydb -c "SELECT * FROM pgbench_accounts LIMIT 10;"
 aid | bid | abalance |                                        filler
-----+-----+----------+--------------------------------------------------------------------------------------
   1 |   1 |        0 |
   3 |   1 |        0 |
   4 |   1 |        0 |
   5 |   1 |        0 |
   6 |   1 |        0 |
   7 |   1 |        0 |
   8 |   1 |        0 |
   9 |   1 |        0 |
  10 |   1 |        0 |
  11 |   1 |        0 |
(10 rows)


azureuser@hc-devtest-sdsc02-io-host:~$ sudo -u postgres psql -d mydb -c "SELECT COUNT(*) FROM pgbench_accounts;"
  count
----------
```

```
 10000000
(1 row)

azureuser@hc-devtest-sdsc02-io-host:~$ sudo -u postgres psql -d mydb -c "SELECT SUM(abalance) FROM pgbench_accounts;"
   sum
----------
 -1335740
(1 row)

azureuser@hc-devtest-sdsc02-io-host:~$ sudo -u postgres /usr/lib/postgresql/16/bin/pg_amcheck -d pgbench-test --all --heapallindexed
pg_amcheck: warning: skipping database "pgbench-test": amcheck is not installed
pg_amcheck: warning: skipping database "postgres": amcheck is not installed
pg_amcheck: warning: skipping database "template1": amcheck is not installed
azureuser@hc-devtest-sdsc02-io-host:~$ echo "exit code = $?"
exit code = 0
azureuser@hc-devtest-sdsc02-io-host:~$ sudo -u postgres /usr/lib/postgresql/16/bin/pg_dump -d pgbench-test > /dev/null
azureuser@hc-devtest-sdsc02-io-host:~$ echo "exit code = $?"
exit code = 0
```

Command	Action	Purpose
`sudo -u postgres psql -d mydb -c "SELECT * FROM pgbench_accounts LIMIT 10;"`	Runs a simple SELECT to fetch the first ten rows of the `pgbench_accounts` table.	Confirms that the server is running, the table exists, and basic reads work without error.
`sudo -u postgres psql -d mydb -c "SELECT COUNT(*) FROM pgbench_accounts;"`	Runs COUNT() to obtain the total number of rows in the pgbench_accounts table.	Verifies that the expected 10 million rows are present, ensuring no data is missing.
`sudo -u postgres psql -d mydb -c "SELECT SUM(abalance) FROM pgbench_accounts;"`	Runs SUM() to calculate the SUM of the `abalance` column across the entire table.	Demonstrates that aggregate functions and index access operate correctly, indicating data consistency.
`sudo -u postgres /usr/lib/postgresql/16/bin/pg_amcheck -d pgbench-test --all –heapallindexed`	Runs `pg_amcheck`, PostgreSQL's low-level integrity checker, against all relations.	Confirms that no structural corruption is detected; exit code 0

		indicates the check passed.
`sudo -u postgres /usr/lib/postgresql/16/bin/pg_dump -d pgbench-test > /dev/null`	Performs a full logical dump of the `pgbench-test` database.	Ensures that every table can be read and exported without error, confirming recoverability.

Conclusion

This white paper validates a solution that enables you to perform development and testing of production workloads hosted on on-premises VMware vSphere in the cloud.

This solution offers the following benefits.

- You can efficiently develop and improve business applications and systems, supporting overall company success.
- By using copies of production data for development and testing, developers and testers can perform their tasks quickly and efficiently.
- Infrastructure managers can reduce deployment tasks and shorten lead times.
- Infrastructure administrators can reduce system operational overhead.
- Initial implementation costs can be reduced; CAPEX is minimized, and usage fees can be covered through OPEX.
- Capacity efficiency can be improved, and OPEX for cloud-native solutions can be reduced.

Best Practices

When building the system described in this white paper, follow the best practices.

Best Practices for UVM

- **Connection Configuration**: Register the initiator's IQN for VSP One Block with the SDSC to enable access to SDSC volumes.

Best Practices for UR

- **Connection Configuration**: Use the port of the active node of the controller being used as the SDSC port for connecting to the UR.
- **RAID Manager IP Address and UDP Port**: Use the IP address and UDP port of the controller being used as the RAID Manager IP address and UDP port.
- **Volume Assignment to Controllers**: Assign the SDSC volumes used by UR to the controllers in use.
- **Pair Operations:** After creating or resynchronizing UR pairs and confirming that the pair status is PAIR, use the `pairsplit` command to ensure that data from the primary site is reliably flushed to the secondary site.

