

Operational analysis of Nutanix database service (NDB) running on Nutanix hyperconverged infrastructure (HCI)

Simplifying database lifecycle management with Nutanix NDB

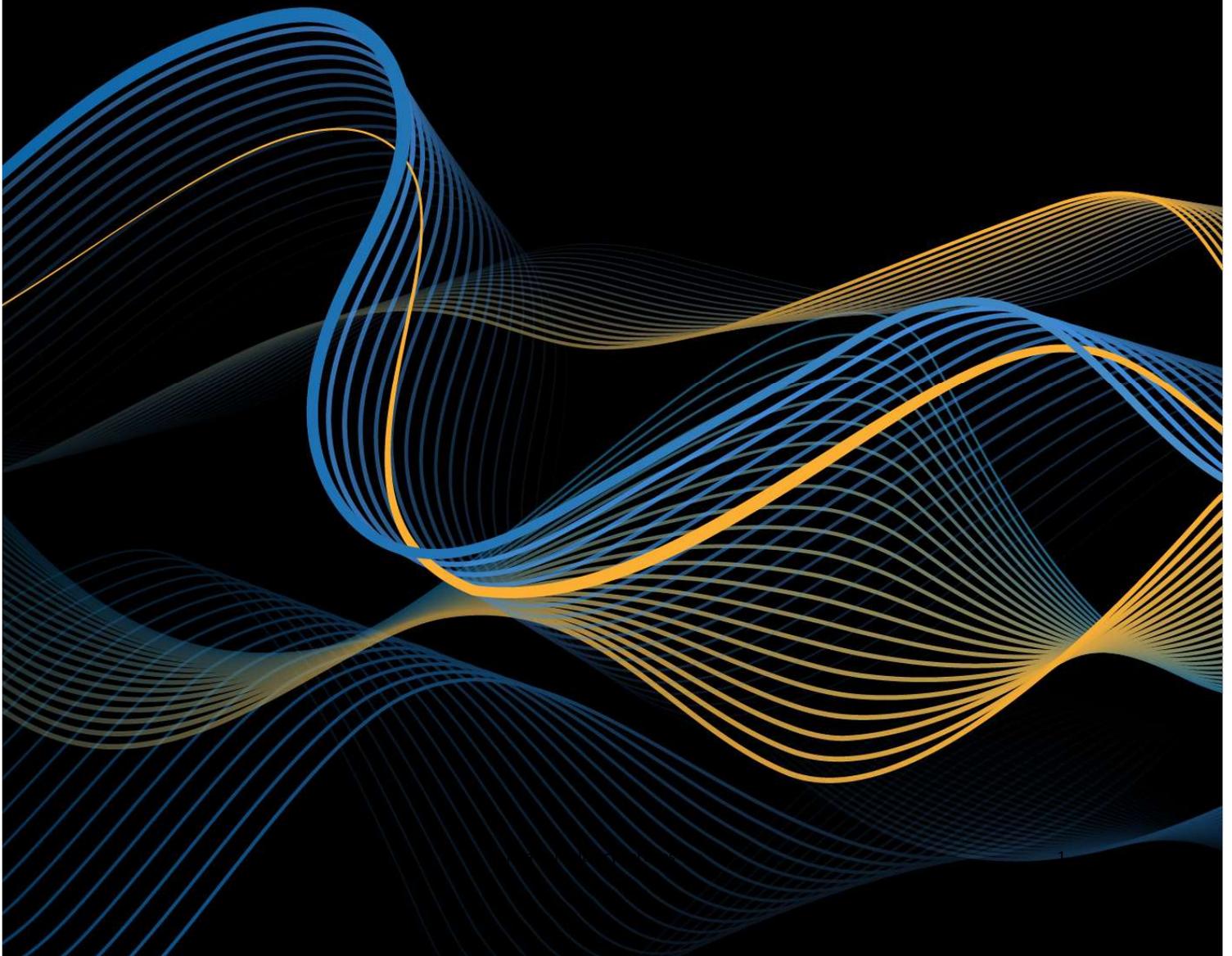


Table of Contents

Executive Summary:	3
1. Introduction to Nutanix Database Service	4
2. HA Cluster Provisioning	9
3. Patch Update	10
4. Bulk Patching	11
5. Snapshot Operations	12
6. Restore	13
7. Database cloning	14
8. Conclusion	15

Executive Summary:

Databases are the backbone of enterprise applications, analytics, and decision-making. As environments grow more complex, managing their lifecycle – provisioning, patching, cloning, backup & restore – becomes critical to ensure availability, security, and operational resilience.

This report focuses exclusively on controlled operational tests of Nutanix Database Service (NDB) *lifecycle operations* under conditions designed to reflect enterprise-scale deployments. The goal is to validate automation efficiency, consistency, speed, and scalability across diverse workflows. This report does not include transactional performance or query latency measurements.

Key Findings: Testing on a 40 TB, 3-node, high-availability PostgreSQL database cluster demonstrated:

- Provisioning high availability cluster in **29 minutes**
- Snapshot of the database in **5 minutes**
- Restore from snapshot in **22.5 minutes**
- Cloning from snapshot completed in **16 minutes** (including provisioning required VMs)
- Patching a 3 node, high availability cluster's PostgreSQL DB engines in under **10 minutes**

Additionally, a bulk-patching test was applied on a set of 20, 200 GB single-instance databases:

- Bulk patching the PostgreSQL DB engine for 20 databases was completed in **6.5 minutes**.

These results confirm that NDB can deliver predictable, repeatable lifecycle automation at scale, reducing time-to-value for critical database operations

1. Introduction to Nutanix Database Service

a. Database Lifecycle Challenges

Managing enterprise databases is inherently complex. Traditional approaches often rely on manual processes that are time-consuming, error-prone, and difficult to scale. As organizations grow and adopt hybrid environments, these inefficiencies become more pronounced – impacting availability, security, and operational agility. Common pain points include:

- Provisioning standardized database topologies (Single as well as high-availability instances) including VM, storage and engine configuration is time-consuming and error-prone
- Manual cloning and refreshes are slow and error prone
- Patch updates cause downtime and impact productivity
- Backup/Restore processes are often inconsistent or untested

b. Nutanix NDB Overview

Nutanix Database Service (NDB) is a unified Database-as-a-Service platform that runs on an underlying Nutanix cluster infrastructure, leveraging platform-native snapshot and storage capabilities while orchestrating database-aware workflows in the control plane.

NDB automates common lifecycle operations – provisioning, patching, cloning, backup, and restore – across multiple engines including Oracle, Microsoft SQL Server, PostgreSQL with pgvector, EDB, MongoDB, MySQL, and MariaDB - although this report focuses exclusively on PostgreSQL.

Unlike traditional tools that address isolated tasks, NDB provisions complete database environments (VMs, storage, and engine) using Profiles, which enforce standardized configurations and best practices for high availability and disaster recovery. This ensures consistency and repeatability while enabling administrators to retain operational control.

While this report focuses on operations within a single Nutanix cluster; NDB can also run on Nutanix Cloud Clusters (NC2), which extends Nutanix's infrastructure stack into public clouds. This enables the same operational capabilities across on-premises and cloud environments for hybrid deployments, although hybrid workflows were not evaluated in this report.

The following schematic shows how the Nutanix Database Service, from a high level architectural standpoint - provides a unified platform for database lifecycle automation, supporting multiple engines and deployment models.

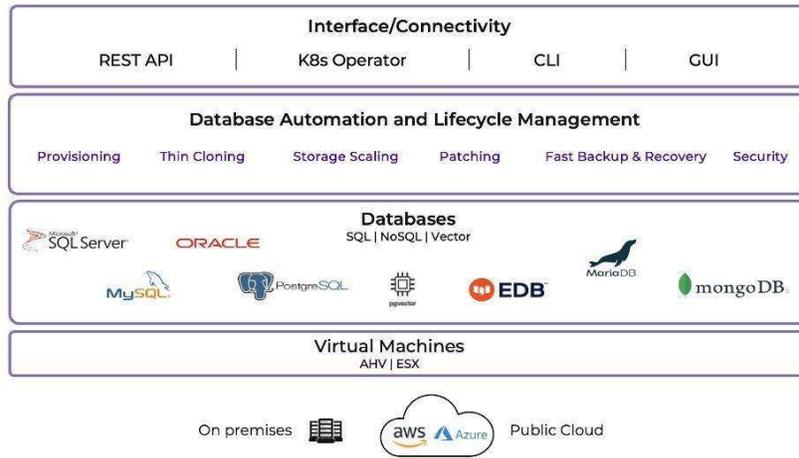


Figure 1: Layered view of the Nutanix Database Service (NDB) system architecture, illustrating connectivity options, database lifecycle functions, supported database engines, and the virtualized or cloud environments where they run.

c. Lifecycle Operational Testing Scope

This testing effort evaluates database lifecycle operations - the critical tasks that DBAs perform every day to maintain availability, performance and security. These include provisioning, patching, snapshot creation, restore, and cloning. By measuring the efficiency and repeatability of these workflows at enterprise scale, the report demonstrates the operational characteristics of lifecycle automation under realistic conditions (as detailed under the [Methodology & Approach](#) section of this report). This is distinct from and not inclusive of traditional database performance tests that focus on transactional throughput or query latency.

For these lifecycle tests, PostgreSQL is selected as the reference database engine. We selected PostgreSQL for its widespread enterprise adoption and robust HA capabilities. By using PostgreSQL, the tests reflect realistic scenarios encountered in production environments while leveraging an engine that is simultaneously open-source, widely popular and highly versatile. While these tests were conducted on PostgreSQL, NDB provides similar workflows and automations for other database engines.

Why this matters

Managing databases at scale is complex and costly. Automation of lifecycle operations reduces downtime and operational risk compared to manual processes.

Nutanix Database Service (NDB) provides automation and deployment capabilities across these lifecycle operations. By unifying VM provisioning, storage configuration, HA cluster orchestration, and database setup into a single workflow built on user-defined standardized profiles, NDB helps organizations reduce administrative effort and maintain consistency and adherence to best practices without compromising operational control.

d. Data Recovery Strategy with NDB

Modern data protection relies on two layers:

- *infrastructure-level resilience* and
- *application-level consistency*.

NDB's approach: Built on Nutanix Cloud Infrastructure (NCI) running on Nutanix Hyper Converged Infrastructure (HCI), NDB leverages space-efficient, redirect-on-write¹ snapshots at the storage layer and augments them with database-aware orchestration. This foundation enables efficient snapshot-based workflows while maintaining alignment with database engine state.

These snapshot-based workflows enable thin cloning, where the initial storage footprint is minimal because the clone references the original snapshot rather than duplicating the entire dataset. This approach speeds up provisioning and conserves capacity. For recovery, NDB creates a thin disk from the snapshot, mounts it to the target database server, and replays transaction logs to bring the database to a consistent state. This process enables recovery within minutes and supports point-in-time restoration without requiring a full data copy.

However, storage-level snapshots alone do not guarantee database consistency. For transactional systems (such as PostgreSQL), a crash-consistent snapshot may still require recovery steps. This is because crash-consistent snapshots capture the state of the storage at a point in time, but may leave in-flight transactions incomplete. This often requires replaying transaction logs during recovery, which can increase downtime and risk data integrity. NDB avoids this by ensuring application consistency before snapshot creation. NDB does this by:

- Integrating with the database engine to ensure snapshots are application-consistent. For PostgreSQL for example, NDB uses the Log Sequence Number (LSN) to confirm that all committed transactions are captured before snapshot creation, ensuring a recoverable and consistent state.
- Orchestrating pre-checks and quiescing steps before snapshot creation, ensuring that the database state is valid for restore or cloning operations.

These abstractions allow NDB to accelerate higher-level workflows – such as cloning and restore – by building on the underlying HCI snapshot primitives while adding database awareness (NDB also supports point-in-time recovery using transaction logs, but this was not evaluated in this report).

In this report, we run these operations – snapshot creation, restore, and cloning – on a 40 TB PostgreSQL HA cluster to validate speed, consistency, and resource efficiency under enterprise-scale conditions.

e. Technical Validation of NDB – Methodology & Approach

Methodology

Managing database lifecycle operations in an enterprise requires more than isolated feature checks; it demands validation under conditions that mirror real-world deployments. To achieve this, we designed a series of operational tests focused on efficiency, consistency, and repeatability across critical workflows such as provisioning, snapshot creation, restore, cloning, and patching. The following section details key test design considerations: the **choice of database engine**, the **parameters that define enterprise scale**, and the **approach used to measure and validate results across the different workflows**.

Database Engine Selection

PostgreSQL was selected as the reference engine for these tests due to its widespread enterprise adoption, robust high-availability (HA) capabilities, and flexibility as an open-source platform. While this report focuses on

¹ [The Nutanix Cloud Bible](#)

PostgreSQL, the operational benefits demonstrated here are applicable to other engines supported by NDB, including Oracle, Microsoft SQL Server, MySQL, and MongoDB.

PostgreSQL HA clusters were orchestrated using Patroni, an open-source tool that manages leader election and failover for PostgreSQL. NDB integrates with Patroni to automate cluster provisioning and lifecycle operations.

Scale Definition and Test Design

To ensure realistic validation, “production-scale” was defined as follows:

- Snapshot, Restore, and Cloning: Operations executed on a **single 40 TB PostgreSQL HA** cluster comprising three Patroni-managed nodes orchestrated by NDB.
- Bulk Patching: Concurrent patching across **20 single-instance PostgreSQL databases**, each sized at **200 GB**.

These configurations were chosen to reflect enterprise-scale deployments where large datasets and HA architectures are common. The tests were run while the database was idle to ensure consistent measurement of operation duration without workload interference.

Repetition and Measurement Strategy

For HA workflows (snapshot, restore, clone, patch):

- Each operation was repeated five times.
- The highest and lowest readings were discarded;
- The remaining three were averaged to validate consistency and eliminate outliers.

All HA workflow execution times presented in this report reflect these averaged values.

For bulk patching:

- The Test was executed once (as the workload inherently includes repetition across 20 instances).

Execution time was recorded through NDB’s control plane for all operations.

Workflows Covered

The following operations were validated:

- Provisioning HA clusters.
- Snapshot creation (on-demand).
- Restore from snapshot (HA to HA).
- Cloning from snapshot (HA to single-instance).
- Patching HA clusters.
- Bulk patching across 20 single-instance databases.

f. Test Environment Details

The environment was sized to reflect enterprise-scale workloads, designed for high availability and consistent performance under load.

Nutanix Cluster Configuration:

The cluster consisted of 3 nodes with the below specification:

- CPU Model - Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz
- Cluster CPU Capacity - 89.6 GHz

- No. of CPU Cores – 32
- Memory - 503.05 GiB
- Storage Capacity - 80.21 TiB; Disks - SSD: 12 Disks

PostgreSQL VM Setup

PostgreSQL configuration for 40 TB HA database tests

Used for Provisioning, Patch Update (rolling), Snapshot, Restore & Cloning:

- Test environment: PostgreSQL HA 3 node cluster
- Dataset size: 40 TB
- vCPUs: 16, (configured as 8 x 2 cores)
- Memory (GiB): 64
- VM Guest OS: Red Hat Enterprise Linux 9.4 (Plow)
- NDB version: 2.8.1
- PostgreSQL DB version: 16.6
- Upgrade Scenario: PostgreSQL version: 16.4 → 16.6

PostgreSQL single-instance fleet (20 × 200 GB)

Used only for Bulk Patching

- Test environment: 20 single-instance PostgreSQL databases
- Dataset size: 200 GB each
- vCPUs: 16, (configured as 8 x 2 cores)
- Memory (GiB): 64
- VM Guest OS: Red Hat Enterprise Linux 9.4 (Plow)
- NDB version: 2.8.1
- PostgreSQL DB version: 16.6
- Upgrade Scenario: PostgreSQL version: 16.4 → 16.6

NDB PostgreSQL tuning parameters (Database Parameter Profile)

- max_connections: 100
- max_prepared_transactions: 0
- min_wal_size: 80MB
- max_locks_per_transaction: 64
- max_wal_senders: 10
- max_wal_size: 1GB
- max_replication_slots: 10
- max_worker_processes: 8

All snapshot, clone, and restore operations were executed within

2. HA Cluster Provisioning

This test evaluated NDB's ability to provision a PostgreSQL HA cluster – a key requirement for enterprises managing large-scale environments or supporting rapid development cycles. The HA database is a three-node

Patroni cluster managed by NDB. Provisioning was executed via API using an automated script to create the HA cluster in parallel, without loading data.

Why This Matters:

Parallel provisioning reduces operational bottlenecks and accelerates time-to-value for database deployments. It validates NDB's scalability under concurrent operations, ensuring that control-plane orchestration and cluster resources can sustain high-demand scenarios without performance degradation. Strong results in parallel workflows also indicate efficiency for single-instance provisioning, where orchestration overhead and resource contention are minimized.

What We Tested:

Using NDB's API-driven automation, we provisioned a PostgreSQL HA cluster.

Each HA database consisted of a 3-node PostgreSQL Patroni cluster managed by NDB.

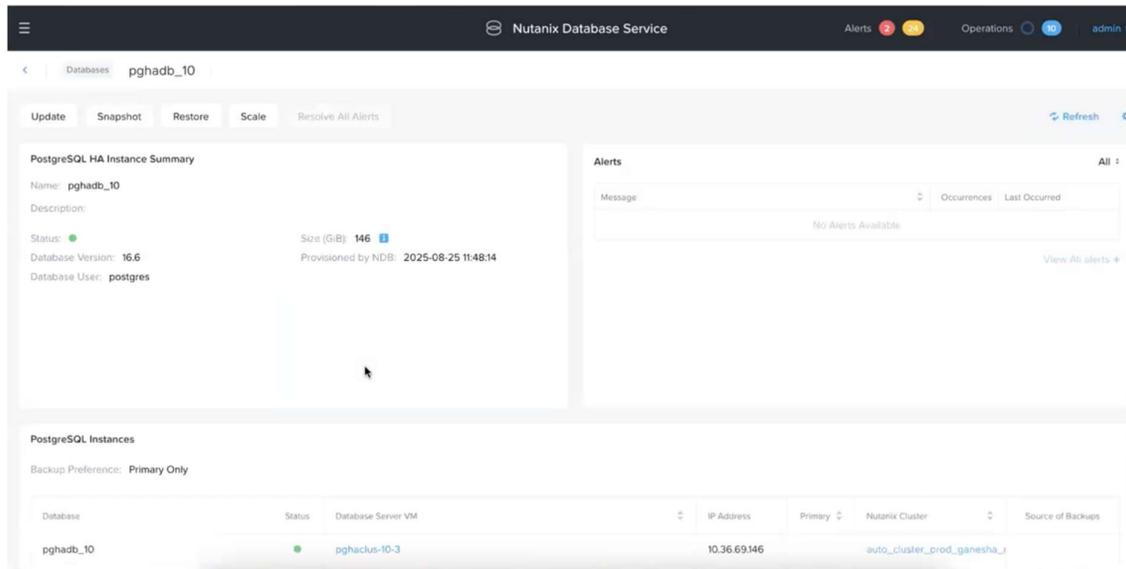


Figure 2: Properties view of a provisioned database from the NDB Dashboard

Results

The PostgreSQL HA cluster (comprising three Patroni-managed nodes) was provisioned in **29 minutes**.

3. Patch Update

This test evaluated NDB’s ability to apply rolling patches on a PostgreSQL high-availability (HA) cluster. The test was conducted on a 40 TB PostgreSQL HA cluster to validate speed and orchestration under idle conditions. NDB patched the HA cluster sequentially, updating follower nodes first, then promoting and updating the leader, ensuring continuous availability throughout the operation

Why This Matters:

Applying patches promptly helps address known vulnerabilities and maintain compliance. Rolling patching on an HA cluster ensures continuous availability during updates, reducing operational risk. Automation reduces human error and minimizes operational overhead compared to manual patching processes. NDB uses *software profiles* for patching, which define user-specific target database engine versions. These profiles enforce version control across all instances, ensuring consistent patch application across environments, preventing configuration drift and simplifying lifecycle management.

What We Tested:

Using NDB’s patching workflow, we upgraded PostgreSQL from version 16.4 to 16.6 on a three-node HA cluster. The operation was executed through NDB’s control plane without manual intervention. NDB ensures the database is in a consistent state by performing a quiesce-equivalent operation before applying the patch.

The figure below shows the NDB interface for patching PostgreSQL databases. It displays the current software profile version, recommended patch, and the “Update” button. Clicking “Update” initiates the automated patching process, which upgrades the database to the recommended version without manual intervention.

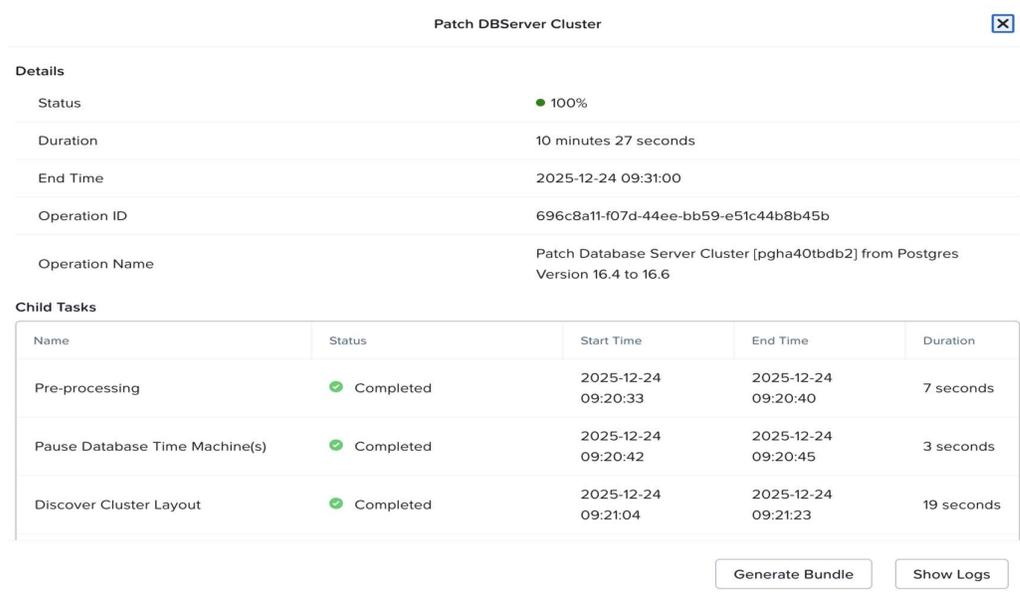


Figure 3: NDB patching interface showing version details for PostgreSQL - and the duration captured from one of the runs factored into the average results for this experiment.

Results

Patch operation completed in an average time of **9 minutes 15 seconds**.

4. Bulk Patching

This test evaluated NDB’s ability to patch multiple PostgreSQL database server VMs in parallel – a vital enabler for enterprises managing large fleets of databases. Unlike rolling patching on HA clusters, this scenario focuses on bulk patching across multiple single-instance deployments to validate orchestration efficiency at scale. While this test focused on single-instance PostgreSQL deployments, NDB also supports bulk patching of HA clusters using the same orchestration framework. NDB’s patching workflow also includes automated rollback if a failure is detected during the patch process, helping maintain consistency and reduce operational risk - though this workflow was not part of this test.

Why This Matters:

Bulk patching ensures that large environments remain secure and compliant without introducing operational delays. Manual patching of dozens of instances is time-consuming and error-prone. NDB’s automated workflow reduces human intervention, accelerates patch cycles, and maintains consistency across all instances, minimizing configuration drift and operational risk.

What We Tested:

Using NDB’s patching workflow, we upgraded PostgreSQL from version 16.4 to 16.6 across 20 database server VMs in parallel. The operation was initiated through NDB’s control plane using a single patch job, which applied the update concurrently to all target VMs. Each VM was validated for readiness using NDB’s pre-check workflow before patching began.

The figure below shows the NDB interface for configuring a bulk patching operation. It illustrates how administrators can select multiple database server VMs, define a maintenance window, and initiate the patching workflow through the control plane without manual intervention.

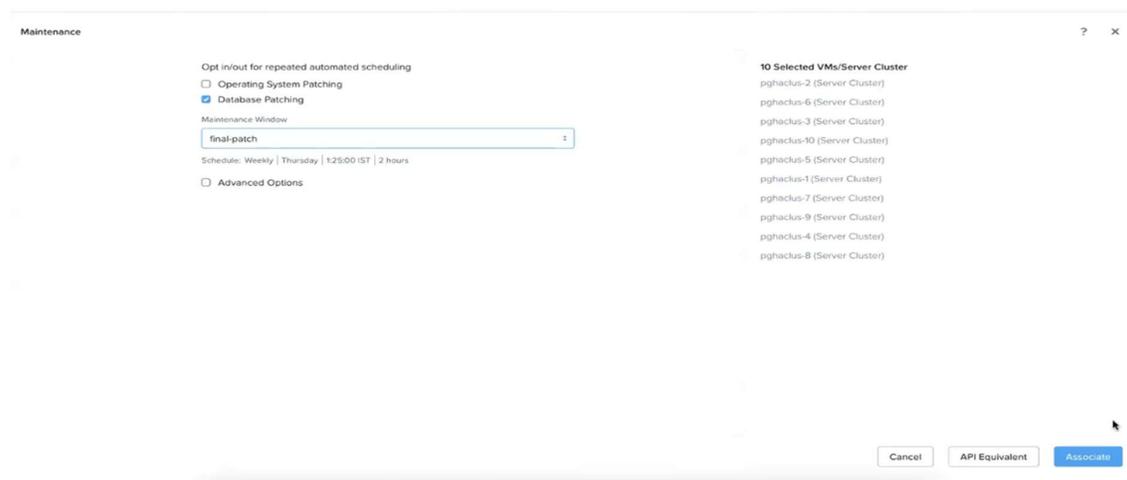


Figure 4: NDB bulk patching interface illustrating VM selection and maintenance window configuration

Results

The bulk patching operation completed in **6 minutes 27 seconds** for all 20 VMs.

Resource utilization remained within expected thresholds, confirming that NDB can handle concurrent patching without overloading cluster resources.

5. Snapshot Operations

This test evaluated NDB’s snapshot creation behavior on a large PostgreSQL HA database cluster with a 40 TB dataset. The objective was to measure snapshot duration and observe orchestration behavior under production-scale conditions. Snapshot operations were executed while the source database remained online, with NDB coordinating a brief quiesce-equivalent step to ensure application-consistent capture. NDB supports *scheduled* as well as *on-demand* snapshots.

Why This Matters:

Snapshot creation is a foundational lifecycle operation that supports recovery, cloning, and maintenance workflows. Evaluating snapshot behavior under production-scale conditions helps characterize system behavior in environments where availability and operational continuity are critical.

What We Tested:

The snapshot evaluation measured snapshot execution time for an on-demand snapshot operation within the defined PostgreSQL HA environment using NDB Time Machine (NDB’s data-protection & thin-cloning mechanism that provides point-in-time recovery (PITR) for databases. It uses application-consistent snapshots together with continuous log backups to restore a database to a specific point in time. It also supports creating space-efficient clones for development and testing purposes). Before initiating the snapshot, NDB performs a quiesce-equivalent operation to the database to ensure application consistency, validating transaction state prior to capture. Operation duration was recorded. This test did not evaluate snapshot replication to other clusters (for example, while NDB’s Data Access Management (DAM) policies can be used to establish a snapshot replica across clusters, this was not part of this test). Similarly, the restore workflow is also tested separately.

The screenshot below shows the NDB Operations view during execution of a snapshot operation for a PostgreSQL Time Machine entity. The operation was initiated on-demand and tracked through the NDB control plane, which records the operation state, start time, and completion status.

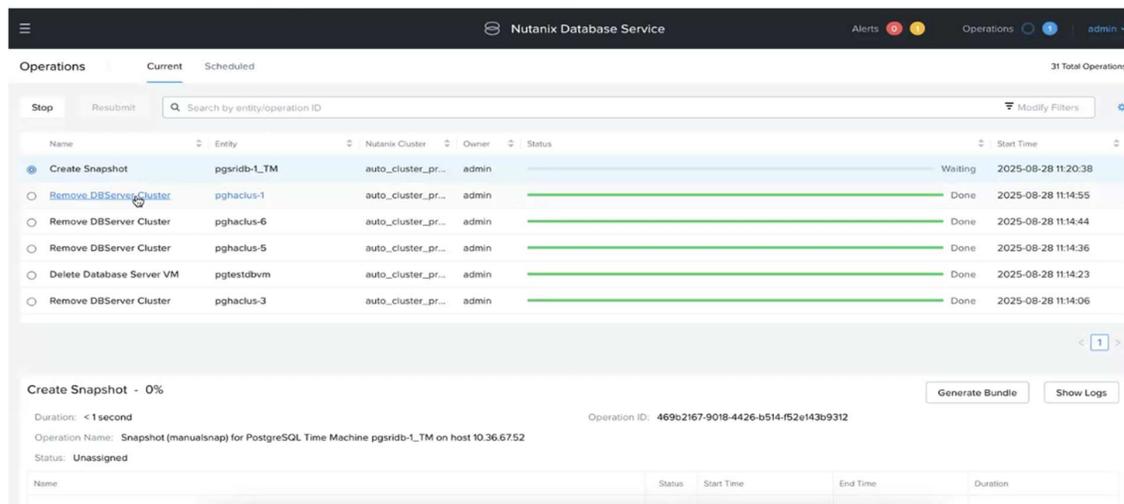


Figure 5: Monitoring the progress of an on-demand Snapshot Operation in NDB

Results

Snapshot creation completed in an average time of **5 minutes and 4 seconds** across all nodes in the HA cluster.

6. Restore

This test evaluated NDB’s restore operation using a snapshot previously captured from the 40 TB PostgreSQL high-availability (HA) database cluster described earlier in this report. The restore was performed on the same PostgreSQL HA environment to measure restore execution time under production-scale conditions.

Why This Matters:

Restore operations are critical for recovering from data corruption, human error, ransomware, or operational incidents. At scale, restore workflows must complete within predictable time bounds to minimize recovery time objectives (RTOs) and reduce operational risk. Evaluating restore behavior on a production-sized HA database environment helps characterize expected recovery timelines for real-world failure scenarios.

What We Tested:

The restore was initiated from a snapshot created on-demand using NDB Time Machine functionality. Operation duration was recorded and tracked through the NDB control plane, which logs operation state, start time, and completion status.

The screenshot below shows the NDB interface during execution of a restore operation for a PostgreSQL Time Machine entity. The operation was manually initiated and monitored through the control plane.

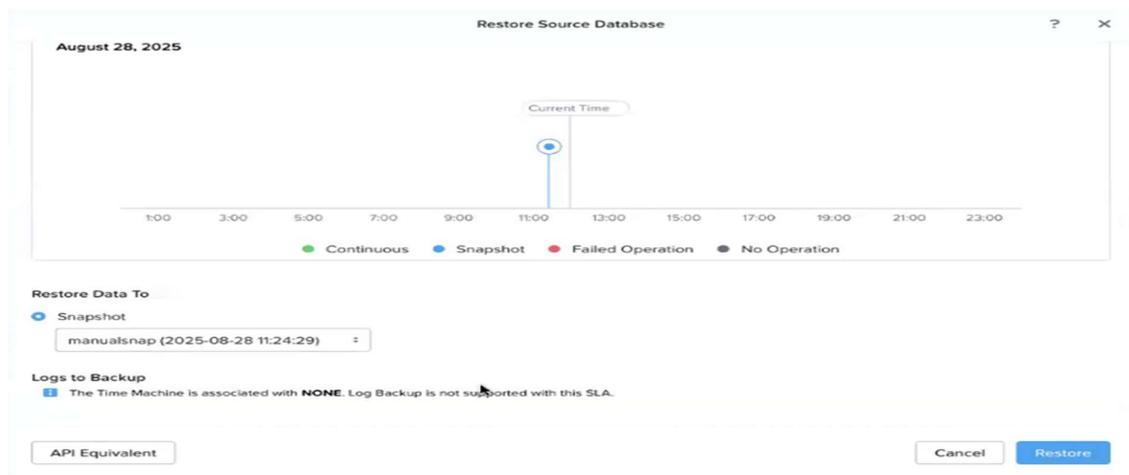


Figure 6: Preparing a restore operation, initiated from a snapshot

Result

Restore operation completed in an average time of **22 minutes 28 seconds** for the entire HA cluster.

7. Database cloning

This test evaluated NDB's ability to clone a 40 TB PostgreSQL high-availability (HA) database cluster from an existing snapshot. The objective was to measure cloning duration given that the workflow includes provisioning a new database server VM as part of the process. For both single-instances as well as high availability clusters, NDB currently produces single-instance clones.

Why This Matters:

Cloning enables rapid creation of database copies for development, testing, and recovery workflows. At scale, manual cloning of HA clusters is time-consuming and resource-intensive. NDB speeds up this process by using snapshot-based technology, reducing storage footprint and accelerating time-to-value for critical operations.

What We Tested:

Using NDB's cloning workflow, we initiated a clone of the 40 TB PostgreSQL HA cluster from a previously captured snapshot. The operation was executed through the NDB control plane and monitored for duration and orchestration behavior. This test did not include pre/post clone scripting.

The screenshot below shows the NDB interface during execution of the cloning operation. The interface displays the clone creation process, including VM provisioning and database initialization.

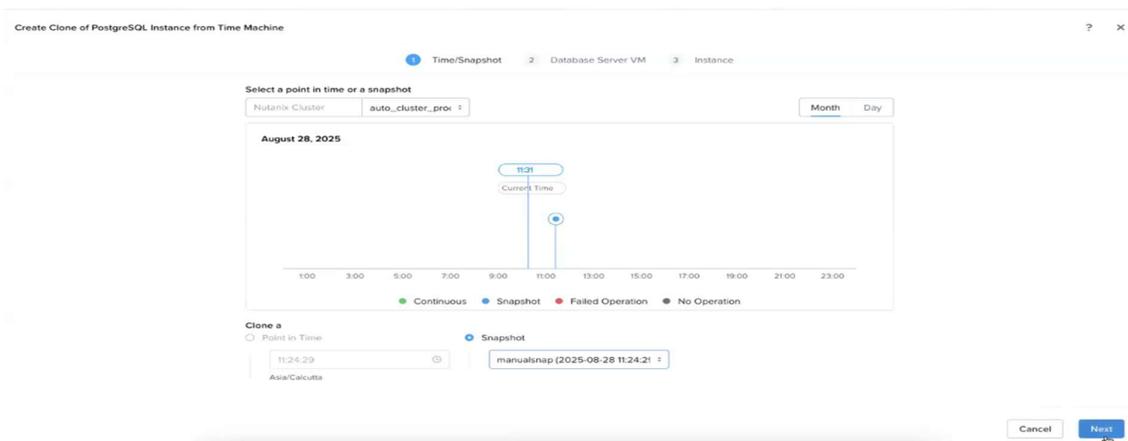


Figure 7: NDB Operations view showing clone creation from a snapshot

Results

Cloning performance was measured at two levels:

- Full end-to-end workflow, which includes provisioning a new database server VM and initializing the cloned database was completed in an average time of **16 minutes 5 seconds**.
- Within the overall workflow, the snapshot-based cloning step (excluding VM creation and database initialization) completed in an average time of **6 minutes 25 seconds**.

8. Conclusion

Modern enterprises need predictable, efficient database lifecycle operations to maintain high availability of critical databases, stay current with patch schedules, and accelerate development by simplifying essential data workflows such as provisioning, cloning, and snapshot management.

- **HA Cluster Provisioning:** A PostgreSQL HA cluster was provisioned in an average time of 29 minutes, validating NDB's ability to handle deployments without orchestration bottlenecks. *This matters because coordinating a VM, storage, and HA cluster setup manually (including Patroni, etcd and HAProxy) is complex and time-consuming; NDB automates these steps at scale, accelerating time-to-value.*
- **Patch Update:** Rolling patch applied to a 40 TB HA cluster in an average time of 9 minutes 15 seconds, maintaining availability throughout the process. *Manual patching of HA clusters is complex because it requires sequencing updates across multiple nodes without breaking quorum or introducing failover risk. For large clusters with significant data volumes, errors in this process can lead to downtime or data inconsistency. NDB automates this workflow, reducing operational risk and eliminating manual coordination.*
- **Bulk Patching:** 20 database server VMs patched in parallel in 6 minutes 27 seconds, confirming efficiency for fleet-wide maintenance. *This is important because large environments often require patching dozens of instances quickly. NDB's orchestration accelerates patch cycles, minimizes human error, and prevents configuration drift across the fleet.*
- **Snapshot Creation:** Application-consistent snapshot of a 40 TB HA cluster completed in an average time of 5 minutes 4 second, enabling rapid recovery points. *This is beneficial because snapshots are the foundation for recovery and cloning workflows. NDB's ability to capture a consistent snapshot quickly reduces operational risk and accelerates downstream processes like restore and clone.*
- **Restore:** Full restore from snapshot completed in an average time of 22 minutes 28 seconds, supporting aggressive recovery time objectives (RTOs). *The key takeaway is that NDB's rapid restore minimizes downtime during incidents. NDB's orchestration ensures predictable recovery timelines for large HA clusters, helping maintain business continuity.*
- **Cloning:** HA cluster clone from snapshot to a new single-instance VM completed in an average time of 16 minutes 5 seconds, including provisioning of a new database server VM. *This matters because cloning enables fast creation of database copies for development and testing. NDB's snapshot-based thin cloning accelerates time-to-value while conserving storage capacity.*

These results represent averaged execution times and demonstrate predictable performance, for critical lifecycle operations at enterprise scale, reducing complexity compared to traditional manual workflows, accelerating time-to-value for database lifecycle management. By leveraging Nutanix HCI's native snapshot capabilities and extending them with database-aware orchestration, NDB ensures consistency, speed, and resilience across provisioning, patching, and recovery workflows.

Disclaimer:

The goal of this document is to provide IT professionals with technical validation of Nutanix Database Service (NDB) capabilities under enterprise-scale conditions.

With the exception of the bulk patching test, which was run across 20 single-instance PostgreSQL databases, all other tests were conducted on a single Nutanix HCI cluster using PostgreSQL high-availability configurations with a 40 TB dataset.

The objective is to illustrate NDB's automation efficiency for lifecycle operations - provisioning, patch update, on-demand snapshot, restore, and cloning - under realistic conditions, not to provide comparative benchmarks against other solutions.

Results may vary based on workload characteristics, cluster sizing, network topology, and operational policies. Organizations should validate these workflows in their own environments before adopting them in production.

Scope of validation: Functional checks such as post-restore integrity verification, application query testing, and performance benchmarking were not part of this operational testing and should be performed in customer environments.

Contact

Visit <https://www.tcs.com>

About Tata Consultancy Services Ltd (TCS)

Tata Consultancy Services (TCS) (BSE: 532540, NSE: TCS) is a digital transformation and technology partner of choice for industry-leading organizations worldwide. Since its inception in 1968, TCS has upheld the highest standards of innovation, engineering excellence and customer service.

Rooted in the heritage of the Tata Group, TCS is focused on creating long term value for its clients, its investors, its employees, and the community at large. With a highly skilled workforce of over 607,000 consultants in 55 countries and 180 service delivery centres across the world, the company has been recognized as a top employer in six continents. With the ability to rapidly apply and scale new technologies, the company has built long term partnerships with its clients – helping them emerge as perpetually adaptive enterprises. Many of these relationships have endured into decades and navigated every technology cycle, from mainframes in the 1970s to Artificial Intelligence today.

TCS sponsors 14 of the world's most prestigious marathons and endurance events, including the TCS New York City Marathon, TCS London Marathon and TCS Sydney Marathon with a focus on promoting health, sustainability, and community empowerment. TCS generated consolidated revenues of US \$30 billion in the fiscal year ended March 31, 2025. For more information, [visit www.tcs.com](https://www.tcs.com)

Follow TCS on [LinkedIn](#) | [Instagram](#) | [YouTube](#).