v1.5 | January 2022 | BP-2061

BEST PRACTICES

PostgreSQL on Nutanix

NUTANIX
YOUR ENTERPRISE CLOUD

Copyright

Contents

1. Executive Summary 5

2. Introduction 6
- Audience 6
- Purpose 6

3. Solution Overview 8
- Invisible Infrastructure 8
- Compression 9

4. PostgreSQL on Nutanix Best Practices 11
- CPU 11
- Memory 14
- Storage 15
- Networking 19
- Sizing and Architecture 19
- Hypervisor 21
- Guest Operating System (Linux, Windows) 22
- Filesystem 25
- PostgreSQL Database 28
- High Availability 29

5. Monitoring 31
- Prism 31
- pgAdmin 32
- Glances 33

6. Conclusion 35

Appendix 36
- References 36
- About Nutanix 36

List of Figures..37
List of Tables..38

1. Executive Summary

The Nutanix enterprise cloud OS is a highly resilient converged compute and storage platform that brings the benefits of web-scale infrastructure to the enterprise. This document highlights why Nutanix is the ideal platform for virtualized instances of PostgreSQL database systems. For business-critical transactional and analytical workloads, Nutanix delivers the performance, scalability, and availability that your IT staff (including Basis and database administrators) requires. Nutanix systems offer:

- Localized I/O and flash for index and key database files, enabling low-latency operations.
- Infrastructure consolidation that lets you eliminate underused application silos and consolidate multiple workloads onto a single, dense platform, using up to 80 percent less space with 50 percent lower capex.
- Nondisruptive upgrades and scalability, including one-click node addition without system downtime.
- Data protection and disaster recovery to automate backups.
- Uncompromising simplicity that eliminates the risks from complicated configurations, manual provisioning, and mapping with disks, RAID, and LUNs.

In this best practice guide, we explain how the Nutanix platform can lower the TCO of your PostgreSQL environment while still delivering top-notch performance. We primarily focus on implementing PostgreSQL Database on Nutanix and provide settings for PostgreSQL and EnterpriseDB.

2. Introduction

Unless otherwise stated, the solution described in this document is valid on all supported AOS releases.

Audience

This best practice guide is part of the Nutanix Solutions Library. We wrote it for database solution architects, database administrators, storage architects, and system engineers responsible for designing, managing, and supporting Nutanix infrastructures running PostgreSQL. Readers should be familiar with PostgreSQL database administration, OS commands, and basic Nutanix design principles.

Purpose

We cover the following subject areas:

- Overview of the Nutanix solution for delivering PostgreSQL on a virtualized platform.
- The benefits of PostgreSQL on Nutanix.
- Choosing the right hardware for your PostgreSQL deployment.
- Design and configuration considerations when architecting PostgreSQL on AOS distributed storage.

Table 1: Document Version History

Version Number	Published	Notes
1.0	April 2017	Original publication.

Version Number	Published	Notes
1.1	September 2017	Updated platform overview and logical volume size.
1.2	September 2018	Updated Nutanix overview.
1.3	April 2019	Updated Nutanix overview and product naming.
1.4	March 2020	Content refresh.
1.5	January 2022	Refreshed content.

3. Solution Overview

Invisible Infrastructure

Invisible infrastructure enables enterprises to focus on business problems instead of repetitive and tedious management and maintenance tasks that add no value. Prism, the Nutanix consumer-grade management interface, offers uncompromising simplicity, with one-click infrastructure management, remediation, and operational insights. Eliminating complexity frees your IT staff to innovate and create.

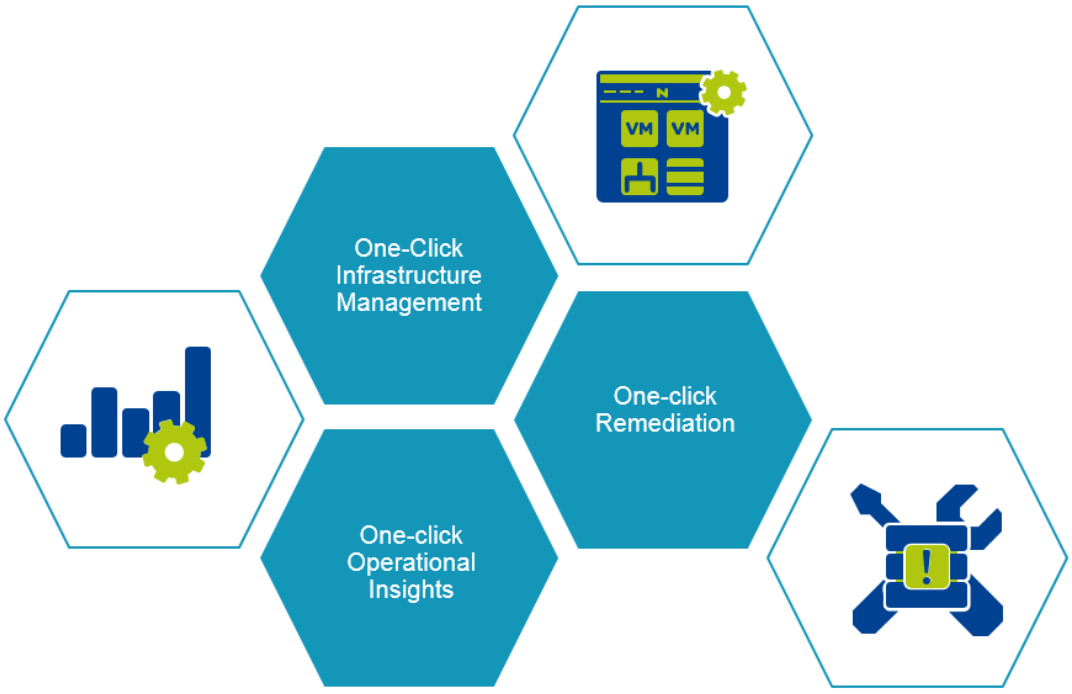

Figure 1: Nutanix Prism Overview

Deploy any application mix at any scale, all on a single platform. You can run PostgreSQL database VM workloads simultaneously while you isolate databases on dedicated hosts for licensing purposes. Nutanix offers high

IOPS and low latency; this combination means that database computing and storage requirements drive deployment density, rather than concerns about I/O or resource bottlenecks. Our testing shows that it's better to increase the number of database VMs on the Nutanix platform than to scale large numbers of database instances or schemas in a single VM. From an I/O standpoint, the Nutanix platform handles the throughput and transaction requirements of demanding transactional and analytical databases with AOS.

Era (the Nutanix software suite that automates and simplifies database administration) also helps hide the complexity of database operations and provides a set of common APIs, a CLI, and a GUI that simply work across multiple database engines. Era allows DBAs to define standards for their database provisioning needs with end-state driven functionality that includes HA database deployments for mission-critical clusters. Nutanix has two blog posts that deal with Era, one of which specifically deals with PostgreSQL on Nutanix and one that discusses Era's data durability in more detail.

Compression

The Nutanix Capacity Optimization Engine (COE) transforms data to increase data efficiency on disk, using compression as one of its key techniques. AOS distributed storage provides both inline and post-process compression to suit the customer's needs and the types of data involved.

Inline compression condenses sequential streams of data or large I/O sizes in memory before writing them to disk, while post-process compression initially writes the data as usual (in an uncompressed state), then uses the Nutanix MapReduce framework to compress the data cluster-wide. When using inline compression with random I/O, the system writes data to the oplog uncompressed, coalesces it, then compresses it in memory before writing it to the extent store. Nutanix uses LZ4 and LZ4HC for data compression. This method provides good compression ratios with minimal computational overhead and extremely fast compression and decompression rates.

The following figure shows an example of how inline compression interacts with the distributed storage write I/O path.

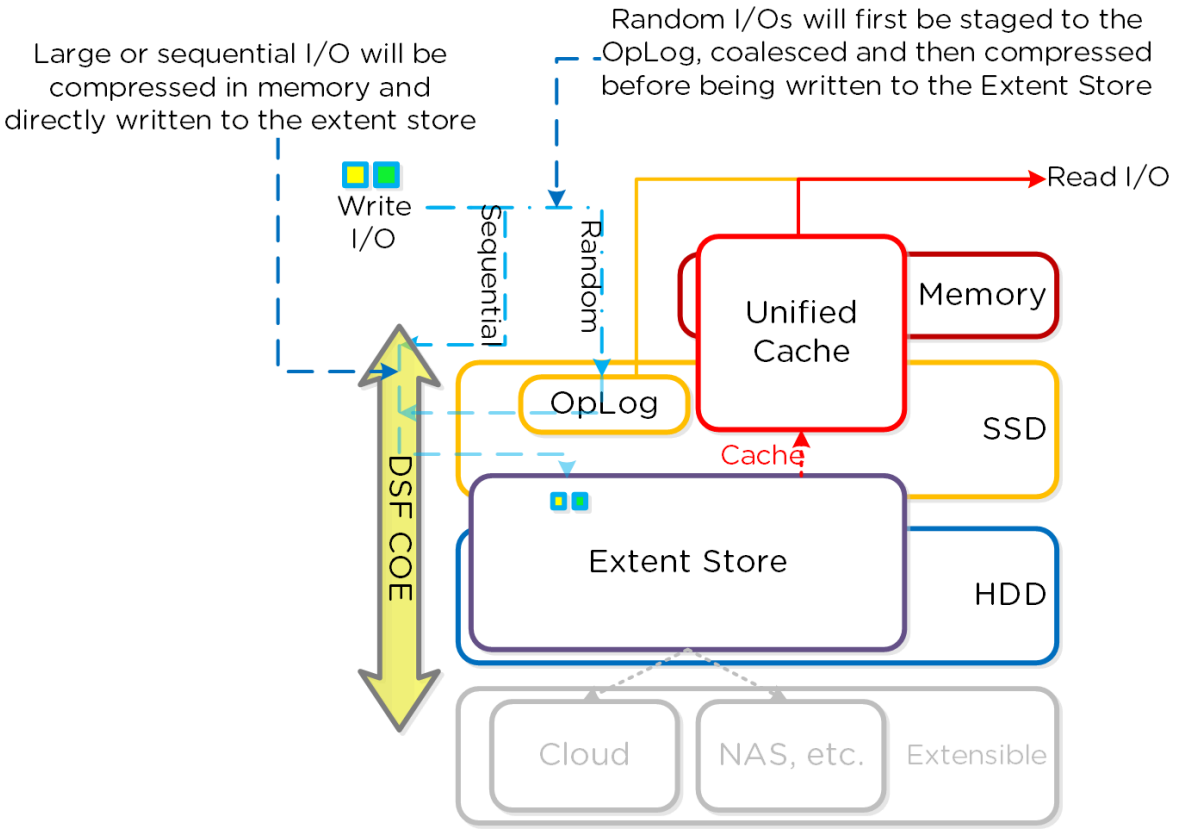

Figure 2: ILM and Compression

4. PostgreSQL on Nutanix Best Practices

> Note: Always refer to the Nutanix website, the support portal, and our collection of solutions-specific documents for the most up-to-date information.

PostgreSQL, often simply Postgres, is an object-relational database management system (ORDBMS) that emphasizes extensibility and standards compliance. As a database server, its primary functions are to store data securely and allow retrieval at the request of other software applications. It can handle workloads ranging from small, single-machine applications, to large, Internet-facing applications with many concurrent users, to mission-critical applications with high-volume transactions or reporting.

The PostgreSQL Global Development Group, a diverse collection of companies and individual contributors, developed PostgreSQL. It is free, open-source software released under the terms of the permissive PostgreSQL License.

In the PostgreSQL on Nutanix best practices that follow, note that most configurations and optimizations occur at the database and OS levels.

Many database administrators still believe that virtualizing databases isn't a good idea, and the countless tips, tricks, and white papers available on the subject only seem to add to the confusion. This guide is meant to make deploying PostgreSQL workloads on Nutanix as simple as possible, while giving you deep insight into every aspect of the solution to ensure reliable, consistent, and fast performance for your database workloads. This guide also acts as a bridge between the various silos in IT organizations, where database, virtualization, and infrastructure administrators need visibility and harmony to design a high-performance application solution.

CPU

> Tip: Work with your account team to choose the best CPU for the job.

Nutanix and its OEM vendors offer various CPU models designed for different purposes. Databases generally perform better with CPU types that have higher clock speeds (GHz) than core counts. With these CPU types, you get strong performance with fewer cores, saving on licensing. CPUs with higher clock speed also achieve lower latency. If licenses aren't a constraint, there are CPUs that offer both higher clock frequencies and a greater number of cores per socket. If you use a socket-based license, you can choose sockets with many cores or select from the single-socket options available from our OEM vendors.

Hyperthreading

Hyperthreading is available on Nutanix as a default, so it requires no administrative intervention.

Nonuniform Memory Access (NUMA)

> Tip: Size VMs within the boundary of a NUMA node whenever possible.

NUMA (nonuniform memory access) is a method of configuring a cluster of microprocessors in a multiprocessing system to share memory locally, improving performance and the system's expandability.

Local memory access times are very fast on NUMA-based systems because the memory controller connects directly to one processor. Remote memory access is many times slower than local.

The number of configured and available vSockets directly impacts how likely a given transaction is to require direct versus remote access. Modern hypervisors can provide vNUMA-aware scheduling, so VM configuration affects OS behavior.

Hot-Adding vCPUs

> Tip: Consider NUMA boundaries before hot-adding vCPUs. Hot-adding vCPUs can disable vNUMA in VMware vSphere. For more information, see VMware KB 2040375.

The effect of hot-adding vCPUs is particularly relevant to database VMs, which can be NUMA-wide. Because we generally size databases using vCPUs capable of handling peak workloads with an additional buffer, hot-add might not be an urgent use case. However, if you need to hot-add vCPUs beyond a NUMA

boundary, what you lose in vNUMA benefits depends on the workload and on NUMA optimization algorithms specific to the database vendor and the version of VMware vSphere you're using. To determine whether the performance tradeoff is warranted in your specific circumstances, VMware recommends determining the NUMA optimization benefits based on your own workload before setting the hot-add vCPU function.

vCPU Count

If workload monitoring shows that the database isn't using all the vCPUs, the extra vCPUs could cause scheduling constraints, especially under high workload.

> Tip: Start with a lower vCPU count for your database and scale up in smaller increments if you witness performance issues. More isn't always better.

CPU Reservations

Generally, setting CPU reservations isn't critical. Tests on vCPU overcommitment show graceful degradation in performance, which you can overcome without any downtime by rebalancing the workloads using VM migration tools across a Nutanix cluster. This solution assumes spare CPU capacity in the cluster.

> Tip: Avoid vCPU core oversubscription initially for tier-1 workloads.

CVM Utilization

> Tip: Take CVM utilization into account.

The CVM is a key component of the IP storage framework in the Nutanix stack. It acts as a scale-out storage controller and manages performance for Nutanix systems. By default, the CVM uses eight vCPUs. This default number is generally acceptable for most database workloads and works well without adding CPU overhead. However, for a very large database, you may need a larger CVM to account for the additional IOPS it has to handle while maintaining low latency. Keep this factor in mind when you decide on the CPU and account for it in your sizing and design.

Memory

Memory per vCPU

> Tip: Assign a minimum of 8 GB per provisioned vCPU.

This guideline applies to both database and application servers, although some applications require less memory. The database server is likely to need more RAM, so determine its allocation based on requirements and testing. Databases love RAM because they usually cache data in memory before the I/O move to storage, and RAM is faster than storage. The faster the database can access data from memory, the sooner it frees up CPU cycles for the next tasks, giving you better performance, lower latencies, and optimal use of your compute resources.

Memory Reservations

> Tip: For production systems under strict performance SLAs, set memory reservations equal to the VM size.

Production databases should have memory 100 percent reserved with no overcommitment allowed. Intel-based memory has become affordable, and we recommend using it.

Large Memory Pages

> Tip: Use large memory pages for databases.

Large page support is enabled by default on VMware vSphere and supported in Linux and Windows. Using large pages can increase translation lookaside buffer access efficiency and improve program performance.

> Note: Large pages can speed up memory allocation to a VM, as described in VMware KB 1021896.

OS Swap Space

> Tip: Configure the size of the OS swap space inside the VM to equal the vRAM assigned to it.

This setting ensures that the swap space can easily handle memory dumps during a leak error, if necessary, which is especially important for Windows. For 64-bit environments, configure the OS swap space to be at least 20 GB.

Virtualization Memory Overhead

> Tip: Take virtualization memory overhead into account.

Because we don't recommend oversubscribing memory when you design and implement database workloads, a correct sizing must provide enough physical memory to support all VM memory requirements. Hypervisors also need physical memory to operate and thus create a small amount of memory overhead per VM, depending on the assigned resources. Take that overhead into account.

Storage

Storage Options

Nutanix has some of the best storage options available in the industry today. You can choose either our hybrid nodes or an all-flash array (AFA). Our hybrid nodes have both SSDs and HDDs. SSDs handle both reads (for active data) and writes. HDDs store cold data, which the system doesn't need for active operations, on more economical disks.

> Tip: Select all-flash nodes whenever possible.

The cost difference between SSDs and HDDs has narrowed over the years, and AFAs provide far better performance and capacity than hybrid nodes at almost the same cost. These developments affect your database workloads positively. With AFAs, you get better IOPS overall, and thus lower CPU utilization.

> Tip: Choose the hardware model based on compute, storage, and licensing requirements.

Some specific guidelines for selecting a hardware model include:

- Keep the working set in SSD and the total database size within node capacity when possible.

- Select a model that can fit all the database storage on a single node. For databases too large to fit on a single node, ensure that there is ample bandwidth between nodes.
- Use node models with more memory for I/O-heavy workloads.
- Use a node with a memory size that is twice that of the largest single VM.
- Use a node that fits your organization's licensing constraints.

Storage Protocols

Nutanix supports common storage protocols, such as NFS (using distributed storage) and iSCSI (using Nutanix Volumes). Each of these storage protocols can achieve excellent performance.

> Tip: Use iSCSI in-guest storage where Windows Server Failover Clustering (WSFC) is required.

If required, our iSCSI technology, Nutanix Volumes, can use the Nutanix cluster as external storage for your existing hardware as an interim or even an ultimate solution.

Replication Factor

Nutanix relies on a replication factor for data protection and availability. This method provides the highest degree of availability because it doesn't require reading from more than one storage location or data recomputation on failure. However, this advantage comes at the cost of storage resources, as it requires full copies.

> Tip: We recommend a replication factor of 2 for most database workloads.

Environments with higher protection requirements can use replication factor 3 but doing so requires more nodes and storage capacity.

vDisks

- Use standard VMDK vDisks on a standard container or datastore.
- Create multiple vDisks for your database.
- Separate disk groups for logs and data.

We recommend that you assign the database log vDisk to a separate PVSCSI adapter, especially when you use VMware vSphere. Spread the database files across virtual SCSI controllers. This distribution maximizes parallel I/O processing in the guest OS. In AHV, the design is much simpler, with a single PVSCSI adapter.

> Tip: Use paravirtualized SCSI adapters for database data and log virtual disks wherever applicable (for example, with VMware vSphere).

Containers

> Tip: Create only the containers you need.

There's no technical advantage to using multiple containers in a Nutanix environment, as doing so doesn't impact performance, but customers can choose to have more than one for organizational and operational purposes like replication.

When you use vSphere, you might need to create two containers to gain maximum performance, especially in an all-flash node, because of the performance of a single NFS network thread. Nutanix AHV doesn't require more than one container for performance benefits.

Thin Disks

> Tip: Use thin disks.

You only need to use eager-zeroed thick disks when you run Oracle RAC, where you need them to enable clusterware features. There's no performance advantage in a Nutanix environment to using lazy-zeroed or eager-zeroed thick disks over thin disks.

Data Efficiency

- For containers created on Nutanix, enable inline compression (delay=0) for improved performance and space savings. With inline compression, customers gain much more usable disk capacity with no performance impact—in certain cases, performance even improves.
- Disable deduplication for database environments.

- Use erasure coding for archival workloads. Don't use erasure coding for database workloads unless Nutanix Support advises you to.
- Increase the queue depth to increase performance for I/O-intensive database workloads. See VMware KB article 2053145 on large-scale workloads with intensive I/O patterns for more information.
- In VMware vSphere, disable Storage I/O Control and Storage DRS, as they offer no benefit in a Nutanix environment. Nutanix has built-in noisy neighbor mitigation from the platform's web-scale design and the storage controller on each Nutanix node.
- Use Nutanix snapshots to create crash-consistent, point-in-time copies of systems as needed to avoid VM stun or pauses in the hypervisor. Snapshots require Nutanix Guest Tools (NGT).

> Note: A snapshot isn't a backup.

Adding Storage Capacity

> Tip: Add storage capacity with storage nodes.

Storage nodes provide additional storage capacity but don't run VMs, so they don't need additional hypervisor licenses. The system can use this added capacity for supplementary snapshots or other storage requirements.

Snapshot Behavior

When you delete a VMware snapshot (for example, when you back up a VM running a database in a three-tier setup), there might be a period when the VM is stunned. See VMware KB article 1002836 for more information. The stun can cause application servers to disconnect unless you configured them to automatically reconnect.

> Note: Don't delete a VMware snapshot while a batch job is running. This operation could cause the batch job to cancel. This limitation only applies to snapshots with VMware.

> Tip: We recommend using Nutanix snapshots wherever possible.

Networking

- Use hypervisor network control mechanisms (for example, VMware NIOC).
- Use the VMXNET family of paravirtualized network adapters. These adapters implement an optimized network interface that passes network traffic between the VM and the physical NICs with minimal overhead.
- Enable receive-side scaling (RSS) if available. RSS is usually enabled by default in Linux. For Windows, enable RSS in your network adapter settings.
- You can use e1000e for legacy applications on older operating systems. We don't recommend e1000e for prolonged usage, however—only as a way to get your environment working on Nutanix until you upgrade your OS to the latest generation.
- Use low-latency switches with at least 10 GbE connectivity.
- Use redundant 10 GbE uplinks from each Nutanix node.
- Don't use network fabric extenders for your core storage network. Network fabric extenders adversely impact the storage subsystem's performance, latency, and reliability.

Sizing and Architecture

It's better to have multiple smaller application servers than one large application server for your databases. This best practice reduces CPU context-switching between processes and generates less overhead. We also recommend having one VM per application server to manage workload distribution and resiliency.

Working Set Size (WSS)

As noted previously, size for your hot tier (SSD). The simplest way to estimate your active WSS is to take the maximum delta of your monthly backup and multiply that number by three.

For example, the following table shows six months of backup deltas for a database that initially went live with 100 GB. The delta is the difference in size between the previous database size and the next available size in the table.

Table 2: Sample Database Information

	Database Backup Size	Delta
Month 1	110 GB	10 GB
Month 2	125 GB	15 GB
Month 3	150 GB	25 GB
Month 4	180 GB	30 GB
Month 5	200 GB	20 GB
Month 6	230 GB	30 GB

In this case, the largest delta available is 30 GB, so the WSS for this particular database is 30 GB x 3 = 90 GB. Combine the WSS together for all your databases to determine the minimum SSD size as a starting point.

Note: The extent store (not the advertised size of the SSD) is directly proportional to the WSS.

This method for calculating WSS doesn't apply to every scenario, because certain database systems have far more read-heavy or write-heavy requirements based on the business that the system supports. Only use the delta calculation described above when no other performance data is available.

In certain scenarios, a quick estimate makes sense when sizing data isn't available. Remember that with Nutanix, you can start small and scale quickly. If no information is available, use this capability to your advantage: start small, monitor your databases, and adjust as needed.

Hardware (Node) Selection

Although Nutanix has certified and supports all available node types for running database workloads, certain node types are better suited for the different server types (VMs) that constitute a database implementation than others. Later, we provide an example hardware selection based on NX nodes offered

directly through Nutanix. The nodes you buy should be configured similarly to the ones in this example section. For information on node types available from our OEM vendors, please contact them directly.

As a guideline, we recommend the following approach: for application servers, choose nodes from the NX-3000 and NX-8000 series based on your performance requirements and the sizing done by your database team. For small and medium database servers, use nodes from the same series (whichever series you choose). For large database servers, we strongly recommend nodes from the NX-8000 series because of their greatly enhanced cold tier performance and their overall optimization for database workloads.

We also strongly recommend using all-flash nodes (SSDs) whenever possible. AFAs are available for all Nutanix node models for databases that require very high performance.

Sizing Support

To ensure alignment with your requirements, work with the Nutanix Sales Team to create a definitive design for your Nutanix cluster and choose the nodes for that design. The database specialists in the vBCA group of Nutanix Engineering's Solutions and Performance group (vbca@nutanix.com) can support these sizing exercises.

Migration Support

Migrating databases from older hardware or different operating systems can become critical as you move to Nutanix. We recommend engaging Nutanix Consulting Services to help you migrate your existing workloads from physical or virtual systems to Nutanix, while observing tested and proven best practices for your selected configuration. With their guidance, you can move your database workloads onto Nutanix without compromising the performance, reliability, or stability of your applications.

Hypervisor

Nutanix supports three different hypervisors:

- Nutanix AHV

- VMware ESXi
- Microsoft Hyper-V

All three of these hypervisors can run your database environments. It's up to the respective OEM (DELL, Lenovo, and so on) to support the different hypervisors for database virtualization. Check with your specific OEM about their current hypervisor support situation.

Hypervisor-related best practices include:

- If you're using AHV, review the AHV best practice guide.
- If you're using vSphere, review performance best practices for your specific version.
- Size Nutanix clusters for a minimum of n + 1 redundancy.
- Don't use resource pools unless you absolutely must. If you do use resource pools, ensure that the shares are sized correctly.
- Always use the latest Nutanix AOS version. You can update AOS without any downtime or impact to your production workloads.
- We recommend using the latest hypervisor upgrades, subject to testing and validation by your IT team. Using Nutanix Prism to upgrade your hypervisor generally doesn't require downtime.
- Use hypervisor HA (high availability) wherever possible. HA is a very effective method of protecting your database VMs against hardware component failures. In most SLA scenarios, hypervisor HA is more than sufficient to meet availability requirements, as long as the design accommodates reasonable failure scenarios.

Guest Operating System (Linux, Windows)

Although almost all Linux distributions support PostgreSQL, most customers deploying PostgreSQL tend to use Oracle Linux (OEL), Red Hat Linux (RHEL), CentOS, SUSE Linux (SLES), or Ubuntu. For MS Windows, we recommend Windows 2008 R2 and later.

Specific tips for guest operating systems include:

- If required for your hypervisor, install the latest version of guest tools in the guest OS. For AHV, you must install NGT. For VMware ESXi, install VMware Tools. Hyper-V doesn't require guest tools.
- Turn off all OS services that your organization doesn't need. Refer to individual OS hardening guides for more information on securing and optimizing your OS.
- Refer to VMware KB article 1006427 for guidance on how to minimize VM time drift.

Linux 2.6

From Linux 2.6 onward, set the Linux kernel I/O scheduler to NOOP. We recommend NOOP for VMs and SSDs, as it tries not to interfere with I/O processes and uses simple FIFO. ESX uses an asynchronous intelligent I/O scheduler; virtual guests should perform better when ESX handles I/O scheduling.

- To set NOOP at boot time, add the elevator option at the kernel line in the /etc/grub.conf file:

```
elevator=noop
```

- While it's possible to set the elevator option for a disk manually, we don't recommend doing so, because you must set it manually every time you add a new disk or device to the OS.
- Add the following line to the boot loader grub.conf:

```
iommu=soft elevator=noop apm=off transparent_hugepage=never powersaved=off
```

For these settings in Linux versions prior to 2.6, refer to your specific Linux guide. Some Linux distributions, such as SLES 12, don't have grub.conf anymore, so this setting may not be applicable.

VM Swap and Page Parameters

Tip: Set VM swappiness to 0 in /etc/sysctl.conf.

The swappiness control defines how aggressively the kernel swaps memory pages. Higher values increase aggressiveness; lower values decrease the amount of swap. A value of 0 instructs the kernel not to initiate swap until the

amount of free and file-backed pages is less than the high-water mark in a zone.

```
# sysctl -w vm.swappiness=0
```

Administrators can tune the following advanced parameters depending on requirements:

- Allow the Linux kernel to better handle memory overcommit:

  ```
  vm.overcommit_memory = 1
  ```

- The hugepages mechanism allows the Linux kernel to use the multiple page size capabilities of modern hardware architectures:

  ```
  vm.nr_hugepages=Same as your DB buffer memory, maximum to VM memory size
  ```

- Percentage of total available memory that contains free pages and reclaimable pages; the number of pages at which the background kernel flusher threads start writing out dirty data:

  ```
  vm.dirty_background_ratio = 5
  ```

- Contains, as a percentage of total available memory containing free pages and reclaimable pages, the number of pages at which a process that generates disk writes starts writing out dirty data:

  ```
  vm.dirty_ratio = 15
  ```

- This tunable defines when dirty data is old enough to be eligible for the kernel flusher threads to write it out:

  ```
  vm.dirty_expire_centisecs = 500
  ```

- How often the kernel flusher threads periodically wake up and write old data out to disk:

  ```
  vm.dirty_writeback_centisecs=100
  ```

Linux Maximum I/O Size

Note: The Linux maximum I/O size mostly applies to kernel 4.x and later but check in your kernel.

The default maximum I/O size set by Max_Sectors_KB restricts the largest I/O size that the OS issues to a block device. Whether you issue I/O at this size depends on the elevator (scheduler) you use, the driver, and the type of I/O your applications are issuing. However, large reads and writes are often at the maximum I/O size.

UDEV (user space /dev) can ensure that all block devices connected to your VM, even if they are hot-plugged, have the same consistent maximum I/O size applied. Simply create a file 71-block-max-sectors.rules under /etc/udev/rules.d/ with the following line:

```
ACTION==”add|change”, SUBSYSTEM==”block”, RUN+=”/bin/sh -c ‘/bin/echo 1024 > /sys
%p/queue/max_sectors_kb’”
```

If you don’t have UDEV in your distribution, you can use rc.local as an alternative and essentially apply the same command. As an example (reboot required):

```
echo 1024 > /sys/block/sd?/queue/max_sectors_kb
```

Note: Without UDEV, you must change max_sectors_kb on every single disk device in your OS individually.

Additional Guest OS Configuration Tips

On high-performance networks, increase network receive and transmit queues. Add these to rc.local:

```
/sbin/ethtool -G ethX rx 4096 tx 4096
```

For very high-performance database systems, add the following options to the boot loader (grub) configuration in addition to those listed above (PVSCSI required):

```
vmw_pvscsi.cmd_per_lun=254 vmw_pvscsi.ring_pages=32
```

On Windows, use the following command to adjust the OS queue:

```
REG ADD HKLM\SYSTEM\CurrentControlSet\services\pvscsi\Parameters\Device
/v DriverParameter /t REG_SZ /d "RequestRingPages=32,MaxQueueDepth=254"
```

Add options = -x to /etc/sysconfig/ntpd in time server option:

```
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
```

Ensure that your firewall policy and SELinux settings allow database ports to be communicated with.

Filesystem

To maximize your database’s performance and scalability in the Linux OS, we recommend using LVM (Logical Volume Manager). LVM creates a logical

volume that aggregates multiple disks to stripe reads and writes and serves as a scalable volume for maintenance.

> Tip: When you use LVM, we recommend striping volumes (don't concatenate) over multiple disks. Use the lvs --segments command to check whether striping or concatenation is in use.

Keep the logical volumes (LVs) and physical volumes (PVs) for data files separate from LVs and PVs used for redo logs and archive logs.

For example, assume you created four vDisks and presented them to the OS. The Linux OS recognizes them as sdb, sdc, sdd, and sde. Issue this command to identify the device names:

```
fdisk -l
```

Now create the physical volume for all these devices:

```
pvcreate /dev/sdb /dev/sdc /dev/sdd /dev/sde
```

Create a volume group (VG) out of these physical volumes. In this case, we've provided the name pgDataVG.

```
vgcreate pgDataVG /dev/sdb /dev/sdd /dev/sdc /dev/sde
```

Finally, create the LV for this VG. In this case, we're using the entire space in the VG for the LV and naming the LV pgDataLV.

```
lvcreate -l 100%FREE -i4 -I1M -n pgDataLV pgDataVG
```

> Note: In the command line above, you must change -i4 to the number of disks in your VG. If you have eight disks, this argument is -i8. This setting stripes the LV across all the disks in the VG.

The process above creates an LV for the PostgreSQL data volume. Some additional LV tips include:

- We recommend having at least two LVs: one for data files and one for log files.
- We recommend having four vDisks in each LV. The size of the vDisks depends on your database size and design. As a default, start with 50 GB vDisks for data disks and 10 GB vDisks for logs.
- Turn off read-ahead for each LV (following the example at the beginning of this section):

  ```
  lvchange -r 0 /dev/pgDataVG/pgDataLV
  ```

- Create filesystem:


```
mkfs.ext4 -b /dev/pgDataVG/pgDataLV
```

XFS can provide significant performance improvement, especially for OLAP databases. Most customers still prefer to use EXT4 because the filesystem has proven to be stable and reliable over the years and it's backward-compatible with a lot of older Linux versions as well. If you decide to use XFS as the filesystem, use the following mount option for it.

- Mount options for the LV: Assuming you created a directory in the path /pgsql/data in your OS, use the following options to mount the LV to the directory (ensure that they are in /etc/fstab for permanent effect):
- For EXT4:

```
mount -o noatime,barrier=0 /dev/pgDataVG/pgDataLV /pgsql/data
```

- For XFS:

```
mount -o inode64,nobarrier,noatime,logbufs=8 /dev/pgDataVG/pgDataLV /pgsql/data
```

Table 3: Parameter Key

barrier=0 / nobarrier	Don't use barriers to pause and receive assurance when writing. (In other words, trust the hardware.)
noatime	Don't update access times (atime) metadata on files after reading or writing them.
inode64	Use 64-bit inode numbering. (This setting has become the default in the most recent kernel trees.)
logbufs=8	Number of in-memory log buffers (between 2 and 8, inclusive).

- In Windows, when you format a disk, use 4,096 as the allocation unit size for PostgreSQL data and log disks. Using mount points is preferable to using drive letters in Windows.
- The Windows equivalent of LVs is storage spaces. For larger databases, you can explore using storage spaces in Windows.
- Ensure that all disks associated with multiple LVs are spread over multiple PVSCSI adapters. (Note: This action is not required for AHV.) The following figure depicts an example with two LV groups. This design can easily cater to a medium to large database's performance requirements.

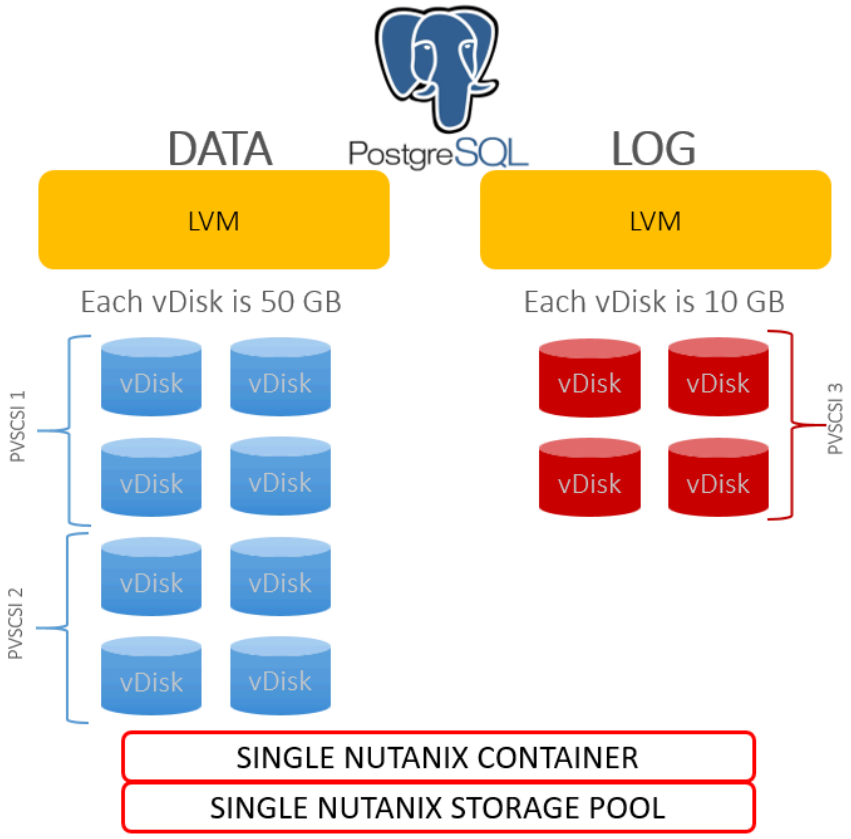

Figure 3: LVM Disks Spread Over Multiple PVSCSI Adapters

PostgreSQL Database

PostgreSQL datafiles (pgdata) and logfiles (pg_xlog/WAL) are usually pointed during installation. Refer to the PostgreSQL installation guide for more details compiling or using these options.

Most of the PostgreSQL parameters are stored in a file called /opt/postgresql.conf. The following table contains the output of an optimized postgresql.conf file. Use this postgresql.conf as guidance and read through each configuration to assess whether it applies to your database scenario and what the user-defined values can do.

Table 4: Optimized postgresql.conf File Output

# Optimized postgresql.conf Parameters for PostgreSQL on Nutanix	
max_connections = 1000	# User defined for concurrent connections.
unix socket directory = /tmp	# User defined.
shared_buffers = xx GB	# 25% of vRAM.
effective_cache_size = xx GB	# 50% of vRAM minimum.
work_mem =	# The maximum amount of memory to be used for intermediate results such as sorting. If you do more sorting, increase this number.
max_fsm_pages = free	
fsync = on	# On if you want to avoid file corruption, otherwise off.
hugepages = try	# Only supported on Linux.
random_page_cost :	# An estimate for the relative cost of disk seeks.
auto_vaccuming or statistics/ analyze = on	# To recover disk space.

- Install PostgreSQL binaries from the EnterpriseDB website. For other variants, refer to that variant's specific manual for setup details and any additional parameters.
- Advanced users may wish to use the tool pgtune for their postgresql.conf. pgtune is not a silver bullet, so use it carefully.

High Availability

1. Use on-hypervisor HA whenever possible.
2. Use PostgreSQL clustering if required.
3. Leave VMware DRS automation at the default level (3).
4. Let the hypervisor and DRS manage noisy neighbors and workload spikes.
5. Use at least n + 1 for HA configuration.
6. Use hypervisor antiaffinity rules to separate clustered nodes, application VMs, and database VMs.

7. Use a percentage-based admission control policy for VMware environments.
8. Review PostgreSQL VM restart priorities.

For a template for using PostgreSQL high availability with ZooKeeper, etcd, or Consul, see the Zalando Patroni GitHub. This blog post tells you more about how to use Ansible to set up a Patroni cluster, and this post explains how to scale a Patroni cluster. For additional information about PostgreSQL high availability and resilience on Nutanix, read this blog post.

5. Monitoring

Prism

Nutanix Prism is an end-to-end consumer-grade management solution for virtualized datacenter environments that combines several aspects of administration and reporting to offer unprecedented simplicity. Powered by advanced machine learning technology, Prism can mine large volumes of system data to automate common tasks and generate actionable insights for optimizing virtualization, infrastructure management, and everyday operations.

We designed Prism from the ground up to deliver a rich yet uncluttered experience. It provides an intuitive user interface to simplify and streamline routine datacenter workflows, eliminating the need to have disparate management solutions for different tasks.

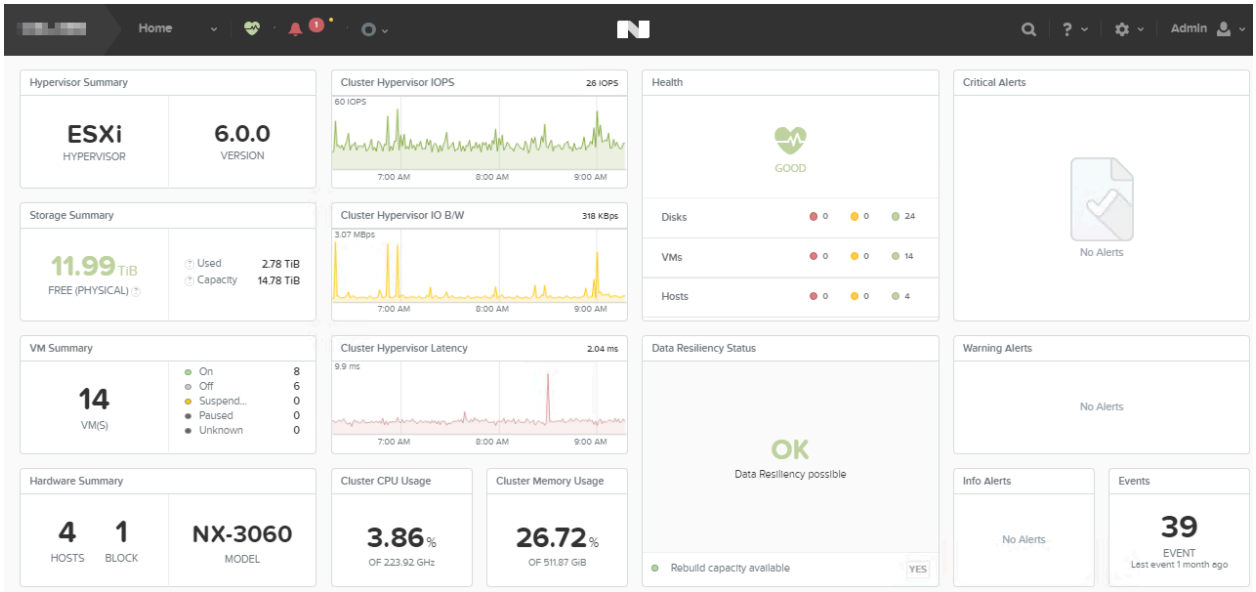

Figure 4: Prism Dashboard

Prism is included with all Nutanix licenses; for operations like managing multiple clusters or AI-powered capacity planning, you may want to investigate

our expanded product offerings such as Prism Central and Prism Pro. The Prism console offers a wealth of information, including storage performance, cluster performance, alerts management for any critical events or warnings, compression and deduplication ratios, and overall resource health. For more information on Prism, Prism Central, and Prism Pro, please refer to the Prism product page.

pgAdmin

pgAdmin is the most popular and feature-rich open-source administration and development platform for PostgreSQL. You can use this application on Linux, FreeBSD, Solaris, macOS, and Windows to manage PostgreSQL 9.2 and later on any platform, as well as commercial and derived versions of PostgreSQL such as EDB Postgres Advanced Server.

pgAdmin is designed to meet various user needs, from writing simple SQL queries to developing complex databases. The graphical interface can run on the desktop or on a web server and supports all common PostgreSQL features. The application includes a syntax-highlighting SQL editor.

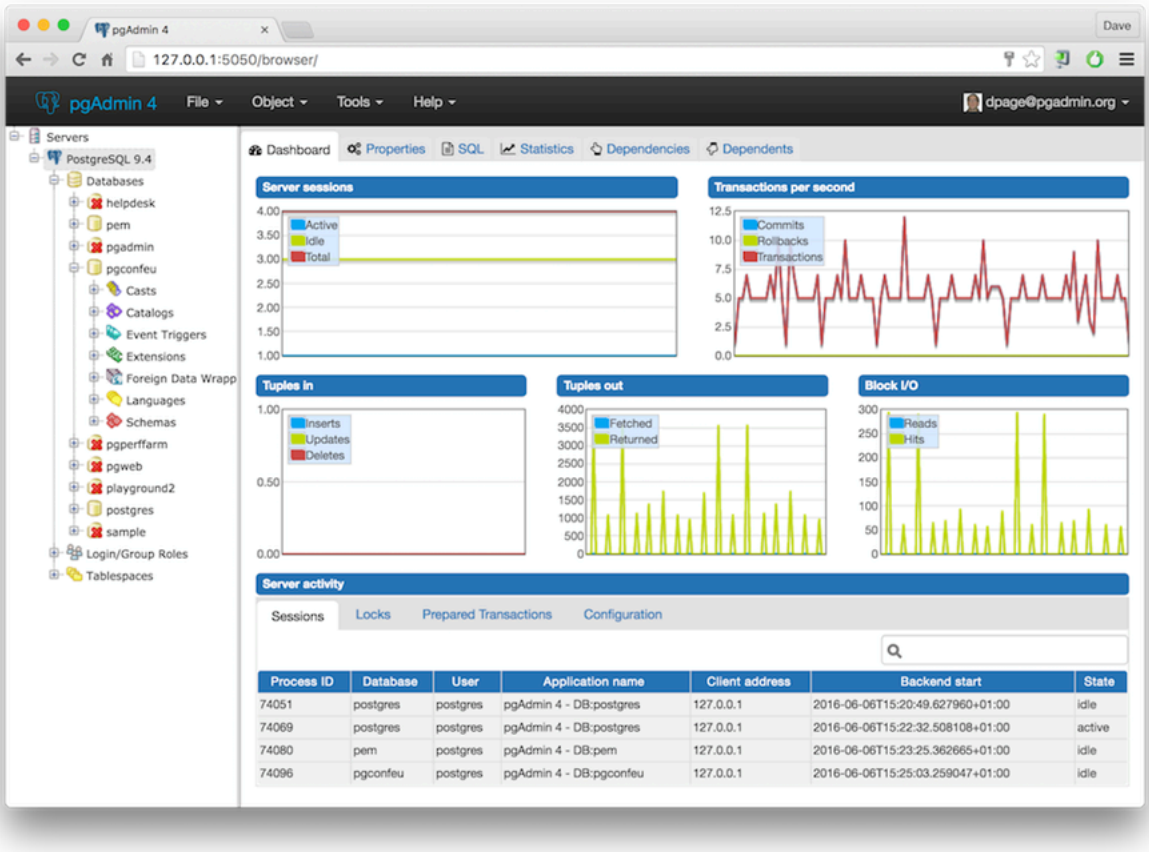

Figure 5: pgAdmin

Glances

Glances is a Python-based, cross-platform system monitoring tool that works with practically any platform, including Linux, OS X, and Windows. It offers insight into your database OS and allows you to correlate your data easily, even on your mobile device. It's also compatible with Docker.

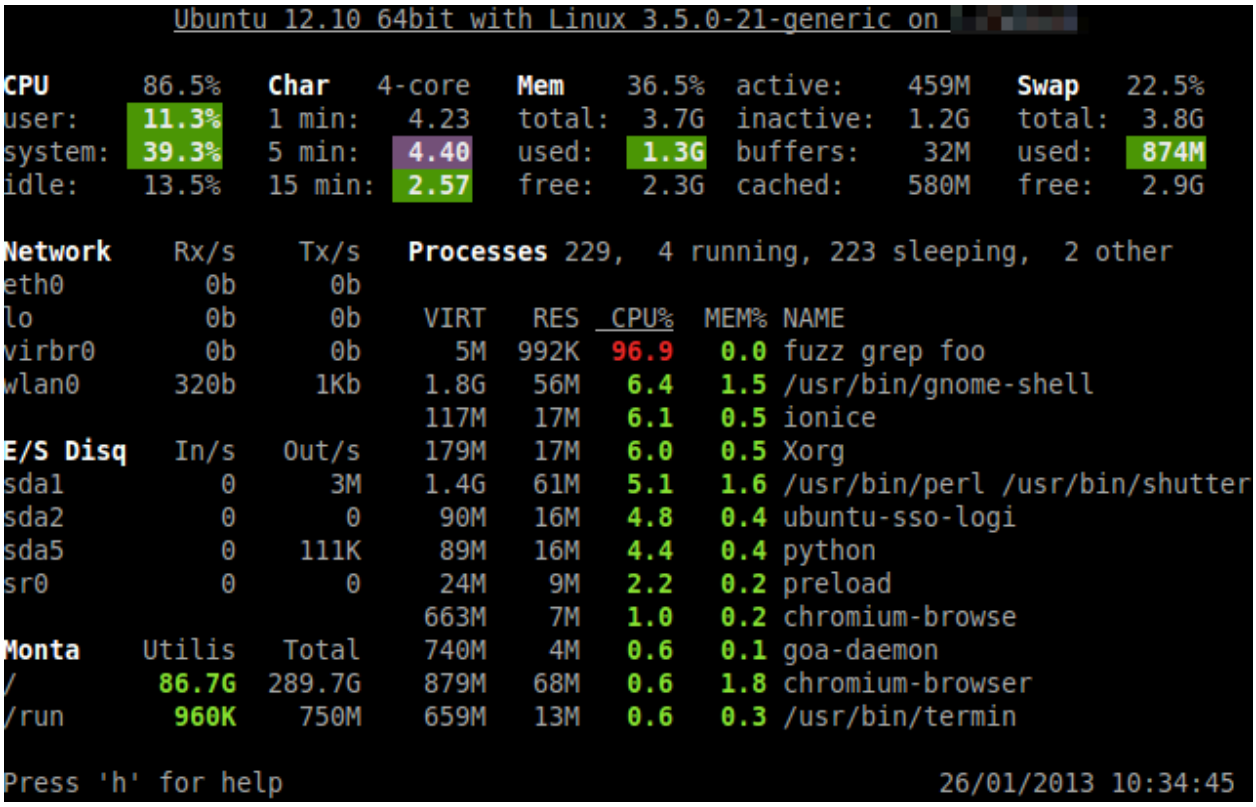

Figure 6: Glances

6. Conclusion

Most of the time, you can run workloads in Nutanix almost immediately—especially DevOps and critical workloads. However, when deploying high-performance, critical workloads on Nutanix, sometimes customers can benefit from some additional guidance—which is where this document comes in.

In this guide, we've covered general strategies and tips for designing your PostgreSQL database to run on Nutanix with uncompromised performance. Larger implementations also need a proper methodology and migration plan—and we can help. Please contact your Nutanix representative early when planning large-scale database projects, so you can move to Nutanix without impacting your IT or your business.

Appendix

References

1. PostgreSQL
2. EnterpriseDB
3. AHV Best Practices
4. VMware vSphere Support Center
5. Best Practices for Performance Tuning of Latency-Sensitive Workloads in vSphere VMs

About Nutanix

Nutanix makes infrastructure invisible, elevating IT to focus on the applications and services that power their business. The Nutanix enterprise cloud software leverages web-scale engineering and consumer-grade design to natively converge compute, virtualization, and storage into a resilient, software-defined solution with rich machine intelligence. The result is predictable performance, cloud-like infrastructure consumption, robust security, and seamless application mobility for a broad range of enterprise applications. Learn more at www.nutanix.com or follow us on Twitter @nutanix.

List of Figures

Figure 1: Nutanix Prism Overview....8
Figure 2: ILM and Compression....10
Figure 3: LVM Disks Spread Over Multiple PVSCSI Adapters....28
Figure 4: Prism Dashboard....31
Figure 5: pgAdmin....33
Figure 6: Glances....34

List of Tables

Table 1: Document Version History....6
Table 2: Sample Database Information....20
Table 3: Parameter Key....27
Table 4: Optimized postgresql.conf File Output....29