v3.0 | May 2023 | BP-2023

BEST PRACTICES

MongoDB on Nutanix

NUTANIX
YOUR ENTERPRISE CLOUD

Copyright

Contents

1. Executive Summary

The Nutanix platform provides a full-stack database solution for production and development MongoDB implementations, with predictable performance, linear scalability, and web-scale cost efficiency. With powerful self-healing, data protection, and disaster recovery capabilities, Nutanix keeps your applications running and your critical data protected on VMware vSphere or Nutanix AHV.

Nutanix modernizes and simplifies database operations and management with Nutanix Database Service (NDB; formerly Era), helping you drive efficiency, agility, cost-effectiveness, and scalability. With one-click provisioning, patching, and cloning, NDB provides simplicity, security, and standardization on-premises and across clouds.

2. Introduction

Audience

This best practice guide is part of the Nutanix Solutions Library. We wrote it for those architecting, designing, managing, and supporting Nutanix infrastructures running MongoDB. Readers should be familiar with:

- MongoDB
- Nutanix cluster administration (hypervisor, networking, storage)
- Linux OS (guest OS that MongoDB runs on)

Purpose

This document covers the following topics:

- Overview of MongoDB and its use cases
- Benefits of running MongoDB on Nutanix
- Design and configuration considerations when architecting a MongoDB solution on Nutanix
- Linux OS best practices for MongoDB on Nutanix
- Hardware best practices for MongoDB deployment on Nutanix
- Best practices for MongoDB with Nutanix Database Service (NDB)

Document Version History

Version Number	Published	Notes
1.0	July 2015	Original publication.
1.1	July 2016	Updated platform overview.

Version Number	Published	Notes
1.2	July 2017	Minor edits throughout.
1.3	October 2018	Updated Nutanix overview.
1.4	November 2019	Updated Nutanix overview.
1.5	November 2020	Updated Nutanix overview.
2.0	February 2022	Updated Network, Site Planning and Solution Application, and Best Practice sections.
2.1	August 2022	Updated Site Planning and Solution Application and Best Practice sections.
3.0	May 2023	Major updates throughout.

3. MongoDB Overview

MongoDB is a document database that provides high performance, high availability, and easy scalability. MongoDB is a NoSQL(not only SQL) database, which means that it doesn't use traditional relational database management system (RDBMS) principles. MongoDB doesn't have the concept of tables, rows, or columns, nor is it compliant with ACID (atomicity, consistency, isolation, and durability) expectations.

MongoDB stores data in binary JSON (BSON) documents, where all the related data is placed together, offering a schema-less model that provides flexibility in terms of database design. JSON and BSON documents don't require a predefined schema, and the data structure can change over time.

Data Model

A MongoDB deployment can have one or many databases, and each database has a set of collections. Collections are conceptually like tables in an RDBMS, but they're schema-less. Each collection can have one or multiple documents, which are like rows in an RDBMS.

A MongoDB document is structured as follows:

```
{
 field1:value1;
 field2:value2;
 .
 .
 fieldN:valueN;
}
```

```
The following lines show an example MongoDB document:
{
  \_id: ObjectId(\"63cf42daadc8e61c49b23a5f\"),
  string: 'Hello World',
```



```
Boolean: true,
number: 10,
NumberInt: 100,
NumberLong: Long(\"4934937489\"),
date: ISODate(\"2023-01-24T02:30:50.830Z\"),
object: { a: 10, b: true },
array: \[ 1, 2, 3 \]
}
```

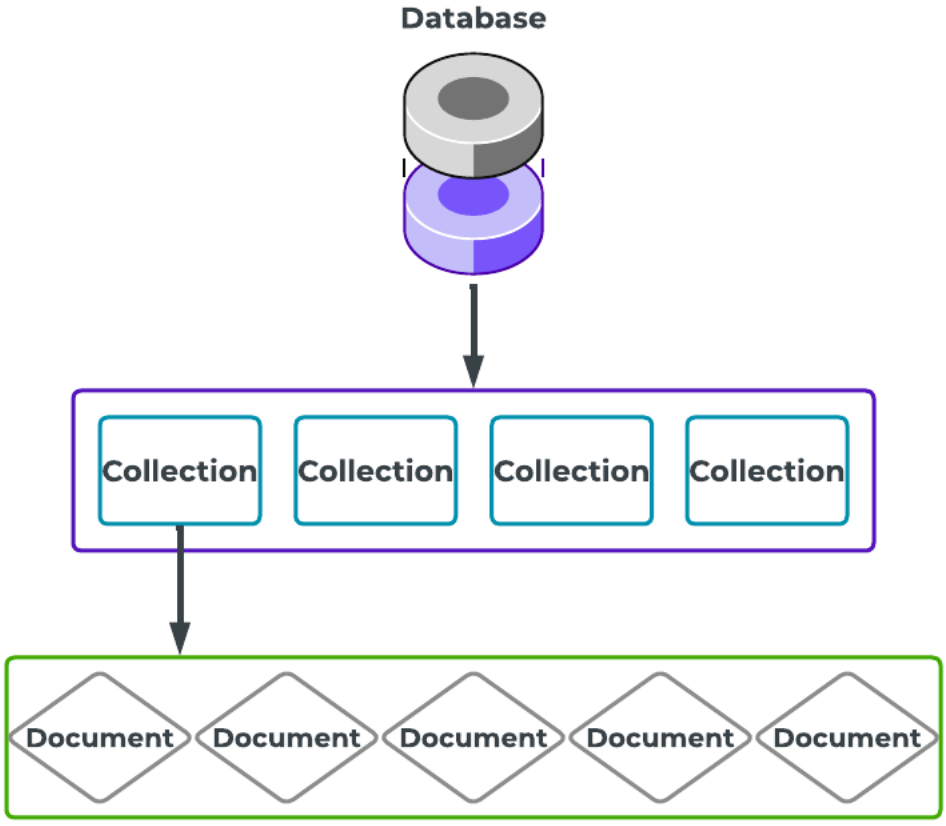

Figure 1: MongoDB Data Model

Core Processes

There are three core components in the MongoDB package:

1. mongod: The primary daemon process in a MongoDB system. This daemon handles all the data requests, manages data formats and access, and performs background management operations.
2. mongosh: An interactive JavaScript and Node.js interface (CLI) used for interacting with MongoDB deployments. You can use mongosh to test queries and operations directly on the database and as an interface to manage databases.

   Note: The legacy mongo shell was deprecated in MongoDB 5.0 and removed in MongoDB 6.0. The new MongoDB Shell mongosh offers numerous advantages over the legacy shell. The new shell has improved compatibility with the MongoDB Node.js driver, syntax highlighting, command history, and logging.

3. mongos: Used in MongoDB sharding; acts as a routing service that processes queries from the application layer and determines the location of the requested data in the sharded cluster (the server holding the data).

MongoDB Deployment Types

Standalone Deployment

A single-node or standalone deployment consists of a single mongod process connected to a client. The standalone deployment doesn't offer any high availability, nor does it ensure any data redundancy or recovery in case of failures, which is why we don't recommend standalone deployments for production environments

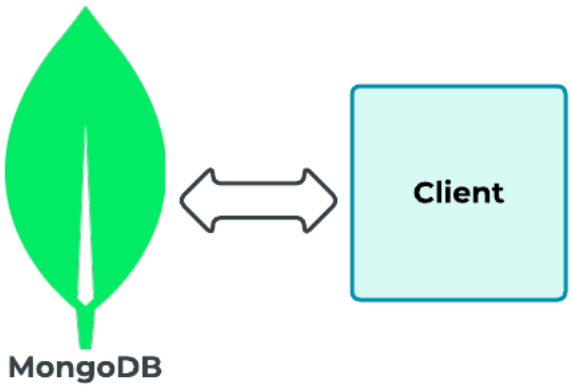

Figure 2: Standalone MongoDB Architecture

Replication Deployment

Replication deployments provide high availability and data redundancy by replicating data on multiple nodes. Replication deployments also simplify routine tasks such as backups, which the deployment can offload to the replica sets to free the primary node to handle application requests. In some cases, replication deployments can help scale reads by enabling clients to read from secondary nodes.

There are two types of replication supported by MongoDB: primary and replica (or primary-replica) and replica set. Primary-replica replication was used before MongoDB 3.2 and we don't recommend it. Replica set replication is similar to primary-replica replication but provides automatic failover, so we recommend it for MongoDB deployments.

A replica set consists of a primary node and multiple secondary nodes, and unlike primary-replica replication, there's no single node that's set as the primary. If the primary node goes down, an election mechanism automatically promotes one of the secondary nodes to primary, and client connections route to the new primary so that both the application and the data remain available.

You can configure secondary members for specific purposes (act as a cold standby, run applications that require separation from normal traffic, act as a historical snapshot, and so on). For more information, see Replica Set Secondary Members.

Note: A replica set can have up to 50 members but only 7 voting members.

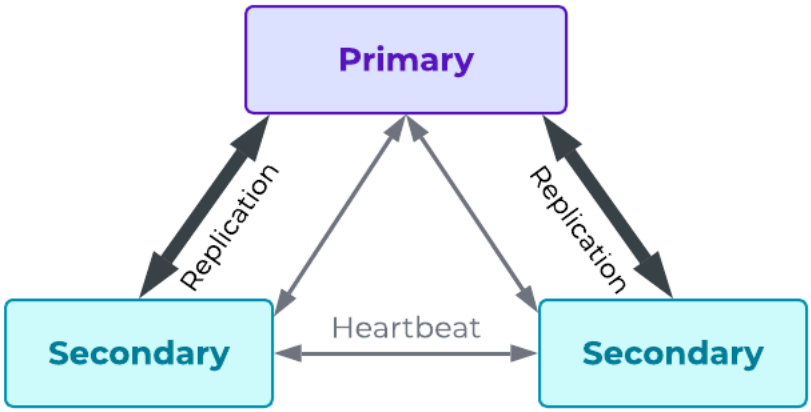

Figure 3: Replica Set Replication Diagram (based on MongoDB's Replica Set Secondary Members)

Sharded Deployment

Note: Sharded deployments distribute data across multiple servers, allowing parallel queries and faster data retrieval. This method can significantly improve the performance of MongoDB deployments with large datasets and high throughput, so we recommend using it in those scenarios.

MongoDB uses sharding to address the challenges of scaling to support large data sets and high throughput by horizontally dividing the datasets across multiple servers, where each server is responsible for handling its part of a dataset and no one server is overburdened.

Sharding is also called partitioning. In MongoDB, you can partition a collection, distributing its documents across multiple servers. The distributed documents are called shards. The key components of a sharded deployment are as follows:

1. mongos: Acts as a query router and routes read and write requests from the application to the shards.

2. Shards: Where the actual data resides. The data from all the shards combined forms the complete dataset for the sharded cluster.
3. Config servers: Store the sharded cluster's metadata and configuration settings. For production deployments, you should have at least three config servers to avoid having a single point of failure.

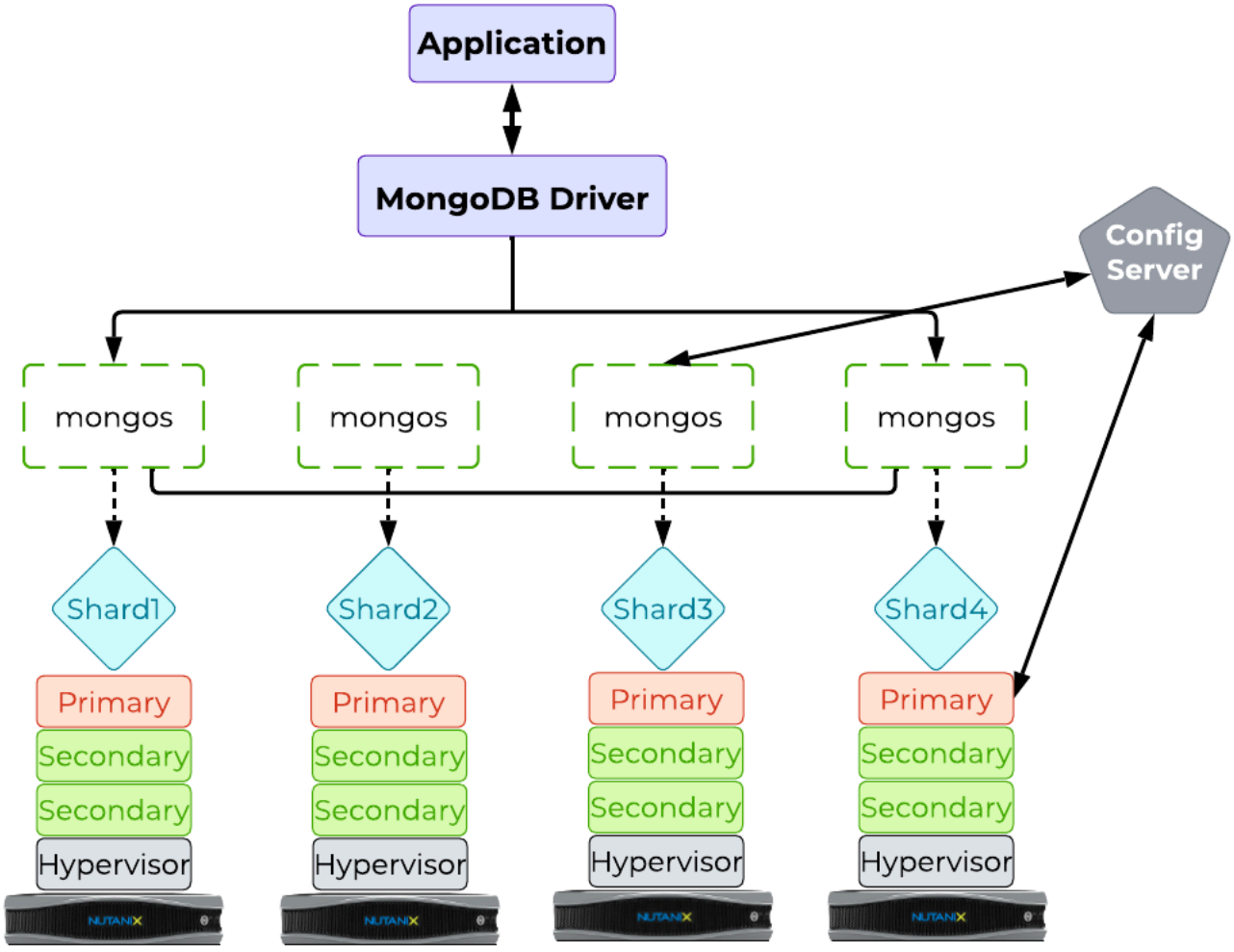

Figure 4: Sharded Deployment Diagram

4. Nutanix Overview

Controller VM

Nutanix Foundation provisions Controller VMs (CVMs) during the Nutanix cluster installation process. The number of vCPUs allocated to a CVM is based on the number of cores per socket in the server hardware.

Table: Recommended CVM vCPU and Memory Settings

Parameter	Default	Large MongoDB Workload
vCPU	12	16
Memory	32 GB	64 GB

Note: All vCPUs for a CVM must fit in a single socket. If you have more than 12 cores per socket, you can increase the number of vCPUs assigned to the CVM. For heavy I/O workloads, increase the CVM memory to 64 GB to provide metadata caching space and reduce latency.

Cluster Resilience

Nutanix AOS supports two levels of fault tolerance (FT): Clusters with three or four nodes are FT1 and clusters with five or more nodes are FT1 or FT2. To achieve application fault tolerance, ensure that the cluster has enough free memory and vCPU capacity available for critical workloads to fail over to another node. With AHV you achieve application fault tolerance by enabling high availability reservations.

Replication factor (2 or 3) is defined per storage container.

Table: Resilience Level Definitions

Resilience Level	Minimum Nodes	Node Failure Redundancy
FT1	3	n + 1
FT2	5	n + 2

Note: n = total number of nodes in cluster

Storage Container

Create a storage container with replication factor 2 (FT1) or 3 (FT2 clusters only) with inline compression enabled. The following table details the recommended Nutanix storage configuration.

Table: Recommended Nutanix Storage Configuration

Nutanix Feature	Property	Comments
Storage container	Create one for MongoDB	Standard practice for databases
Container compression	Enable with delay 0	Saves space and can improve the performance of database workloads running on Nutanix
Container erasure coding	Disable	Not recommended for databases because it consumes significant CPU time to generate parity and checksum
Container deduplication	Disable	Not recommended because it adds processing overhead and doesn't reduce space usage for databases

Virtual Disk Configuration

- Place data, log, and journal files on separate storage devices.
- Nutanix recommends that you create multiple virtual disks (vDisks) for each file system that requires low latency or high storage throughput. The optimal number and size of the vDisks depend on the size and activity of the database and the best practice guidance from the underlying database engine vendor.

- For the WiredTiger storage engine, we strongly recommend using an XFS file system for data-bearing nodes to avoid performance issues that might occur when using EXT4 with WiredTiger. You can also store indexes on a different storage device.

Table: vDisk Configuration

Number of vDisks	Purpose
2	OS, MongoDB software binaries
At least 4	Data
At least 2	Logs
At least 2	Journal

5. Why MongoDB on Nutanix

Digital transformation demands that IT services become more agile, shorten application deployment time, and increase scalability. Designed to run any application, Nutanix software converges storage and compute, eliminating the complexity of separate standalone storage solutions. Nutanix supports VMware vSphere, Microsoft Hyper-V, and Nutanix AHV, allowing you to choose the right hypervisor for your needs. Using the advanced storage functionality of Nutanix Volumes, your workloads get all the foundational Nutanix benefits, such as backup and recovery, disaster recovery, snapshots, clones, and high performance

Running MongoDB on Nutanix offers several advantages:

Nutanix Database Service (NDB)

NDB is a software suite that automates and simplifies database administration, bringing simplicity and invisible operations to database provisioning and life-cycle management. For more information see Nutanix Database Service.

Fault-tolerant platform

Nutanix uses high availability and data redundancy to withstand software and hardware failures, allowing your MongoDB instance to stay up and running even if the underlying hardware has issues.

Enterprise-grade performance

Nutanix provides low I/O latency (less than a millisecond) and predictable performance, while supporting a wide range of transactional and analytical database workloads.

Scalability

With the Nutanix solution, you can start small and scale performance and capacity both vertically and horizontally.

Data protection and disaster recovery

MongoDB and its associated infrastructure VMs are mission-critical and need enterprise-grade data management features, including backup and disaster recovery. Nutanix also offers integrated snapshots, remote replication, and metro-level availability to protect your data.

6. Solution Design

Running MongoDB on Nutanix gives you the flexibility to start small—with a minimum of three nodes—and then either scale up or scale out a node, a block, or multiple blocks at a time. With Nutanix, you can grow to massive scale without any impact on performance. Make sure, however, to apply the appropriate scaling. You might need to consider vertical scale before horizontal.

Table: General Platform Design Decisions

Item	Detail	Rationale
Minimum size	3 Nutanix nodes (3 AHV hosts)	Minimum size requirement for high availability replica set deployment
Scale approach	Incremental modular scale	Allow growth from proof-of-concept to massive scale
Scale unit	Nodes, blocks	Granular scale to meet the precise capacity demands; scale in node increments
Infrastructure services	Small deployments: shared cluster. Large deployments: dedicated cluster (node A from 3 blocks or an NX-3000 or 8000 Series)	Dedicated infrastructure cluster for larger deployments (best practice)
Cluster size	Typically 24–48 AHV hosts	Isolated fault domains
Storage pools	1 storage pool per cluster	Standard practice; ILM handles tiering
Containers	1 container for VMs; 1 container for ISO image store	Standard practice
Features and enhancements	Increase CVM memory to 64 GB	Best practice

Table: MongoDB Server Design Decisions

Item	Detail	Rationale
mongod server VMs; primary or secondary	Min: 3 (n + 1) per replica set. Scale: 1 replica set per block or shard	High availability for mongod servers
Server sizing	vCPU: 2–8. Memory: 16–64 GB (dependent on application working set size). Disk: 80 GB (OS) + 6 × n GB (data) + 20 GB (journal) + 100 GB (logs)	Standard sizing practice; multiple disks or RAID sets for data, journal files, and logs
Data protection	Replica sets; geographically dispersed secondaries	Ensures availability of mongod servers

Table: MongoDB Query Routers Design Decisions

Item	Detail	Rationale
mongos VMs	Min: 3 per cluster. Scale: application dependent	High availability for mongos servers
Server sizing	vCPU: 1–4. Memory: 4–16 GB	Memory based on size of shard metadata cache; allow approximately 1 MB per database connection
Data protection	Require additional VMs based on application growth or requirement	Ensures availability of mongos VMs

Table: MongoDB Configuration Server Design Decisions

Item	Detail	Rationale
mongod (config server) VMs	3 per cluster	High availability for mongod and config servers
Server sizing	vCPU: 1–4. Memory: 4–16 GB (application dependent)	Memory based on amount of metadata to be stored
Data protection	Two-phase commit between 3 VM quorum	3 VM quorum per single MongoDB cluster only

Table: Network Design Decisions

Item	Detail	Rationale
brNutanix virtual switch (vSwitch)	Use: AHV host to CVM communication. Uplinks: N/A	Nutanix default
br0 vSwitch	Use: All external AHV management traffic. Uplinks: bond0	Standard practice
bond0 vSwitch	NICs: 2 × 25 Gbps	Use both 25 Gbps adapters in an active-active configuration
Production VLAN	Network Site: Production. ID: varies. Capability: routable. Components: AHV hosts, Nutanix CVMs, MongoDB servers, query routers, configuration servers	Dedicated infrastructure VLAN; best practice

7. Hardware Best Practices

MongoDB is designed specifically with commodity hardware in mind and its performance depends largely on how you use the underlying compute, storage, and networking resources. Select hardware that meets your requirements.

Nutanix Node Type

The Nutanix Cloud Platform runs on commodity servers from a wide variety of hardware vendors, and you can use two different node types: fully hyperconverged (HCI) or storage-only. HCI nodes host storage services and guest VMs running MongoDB servers. Storage-only nodes aren't schedulable, so they only host storage services (no guest VMs). We designed storage-only nodes to expand the storage capacity of an existing Nutanix cluster.

Table: Nutanix Node Types

Node Type	Guest VMs	Storage	Purpose
HCI	Yes	Yes	General purpose nodes
Storage only	No	Yes	Add storage to a cluster

Processor

Select a processor with a high clock speed and 12 or more cores per socket on a dual-socket server. We recommend at least a 2.7 GHz CPU or faster for production MongoDB servers. At a minimum, this setup ensures that each mongod or mongos instance has access to two physical cores or one multicore physical CPU. Additionally, the WiredTiger storage engine is CPU bound so using a server with multiple cores significantly improves performance.

Memory

Memory size is one of the most important factors in sizing a MongoDB instance. MongoDB performs best when an application's working set (most frequently accessed data and indexes) fits in memory.

Nutanix recommends using a balanced memory configuration, which delivers the highest bandwidth between the processor and memory. The number of DIMMs required for balanced memory varies by model.

> Note: Starting in MongoDB 3.4, the default WiredTiger internal cache size is either 50 percent of (Total memory – 1 GB) or 256 MB, whichever is larger. For example, on a system with a total of 4 GB of memory, the WiredTiger cache uses 1.5 GB of memory: 0.5 × (4 GB – 1 GB) = 1.5 GB. Alternatively, a system with a total of 1.25 GB of memory allocates 256 MB to the WiredTiger cache: 0.5 × (1.25 GB – 1 GB) = 128 MB, which is less than 256 MB.

Storage

Because MongoDB performs all reads and writes through the in-memory data structure, accessing a working set not already in memory triggers a read from a disk, which is an expensive operation. For this reason, storage subsystem performance is critical in system design.

The Nutanix Cloud Platform is a scale-out architecture that distributes disks and I/O across a shared-nothing cluster that scales linearly. Nutanix supports various storage devices—performance (Optane, NVMe, SSD) or capacity (HDD)—and we use all-flash configurations for databases, including the following:

- All flash based on SATA SSD
- All flash based on NVMe SSD
- All-flash hybrid based on a combination of NVMe SSD and SATA SSD
- Premium all-flash hybrid based on a combination of Optane and 3D NAND NVMe SSDs

> Note: You can reduce long-term storage costs for backups and archived data by using Nutanix Objects or Files on a separate hybrid storage drive (SSD and HDD).

Keep the following information in mind when sizing MongoDB storage:

- MongoDB's disk access patterns don't have sequential properties, so using SSDs or NVMes can substantially improve performance.
- Using an XFS file system for a WiredTiger storage engine can help prevent performance issues that might occur when using EXT4.

Networking

MongoDB, as a distributed database, relies on efficient network transport for query routing and internode replication. Fast, low-latency, and highly available networking is critical for the resilience and performance of database workloads. We recommend that you use 25 GbE or faster uplinks, though you can use 10 GbE with Nutanix.

We've developed the following networking best practices:

- Use redundant 25 GbE or faster uplinks from each Nutanix node.
- Enable LACP in an active-active configuration.
- Connect all the nodes in a Nutanix cluster to the same pair of switches.
- The HTTP interface setting is disabled by default. Don't enable it in production deployments.
- Always use trusted environment settings. Apply network rules to prevent access from all unknown servers, systems, and networks. Enable authorization mode if you need it.
- Avoid overloading the mongod or mongos connection resources. Refer to MongoDB documentation for more information.

For more information on networking, refer to the following documents:

- Nutanix Physical Networking best practice guide
- AHV Networking best practice guide
- VMware vSphere Networking best practice guide

8. Linux OS Best Practices

Transparent Huge Pages

Transparent Huge Pages (THP) can add memory pressure and CPU usage. When you run MongoDB on Linux, disable THP for the best performance.

To turn off THP for Oracle Linux 7 and later and Red Hat Enterprise Linux 7 and later, add or modify the `transparent_hugepage=never` parameter in the `/etc/default/grub` file:

```
transparent_hugepage=never
```

Run the `grub2-mkconfig` command to regenerate the `grub.cfg` file:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

> Note: The file name may vary for your operating systems. Check your operating system documentation for the exact file name and the steps to disable Transparent HugePages.

Reboot the system to make this change permanent.

OS User Limits for <mongo-user> User

This value is defined in the configuration files shown in the following table, depending on RHEL or CentOS version.

Table: Configuration Files for OS User Limits

Version	Value	File
RHEL or CentOS 7	4096	`/etc/security/limits.d/20-nproc.conf`
RHEL or CentOS 6	1024	`/etc/security/limits.d/90-nproc.conf`

```
<mongo-user> hard cpu unlimited
<mongo-user> soft cpu unlimited
<mongo-user> hard memlock unlimited
<mongo-user> soft memlock unlimited
<mongo-user> hard nofile 64000
```



```
<mongo-user> soft nofile 64000
<mongo-user> hard nproc 192276
<mongo-user> soft nproc 192276
<mongo-user> hard fsize unlimited
<mongo-user> soft fsize unlimited
<mongo-user> hard as unlimited
<mongo-user> soft as unlimited
```

NUMA

Running MongoDB on a system with Non-Uniform Memory Access (NUMA) might cause a number of operational problems, including slow performance for periods of time and high system process usage.

Disable NUMA on VMs running mongod and mongos processes:

1. Add `numa=off` as follows to your kernel command line in the file `/etc/default/grub.conf`:

   ```
   GRUB_CMDLINE_LINUX=\"crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/
   swap rhgb quiet transparent_hugepage=never elevator=noop numa=off\"
   ```

2. Rebuild the grub file for the changes to take effect:

   ```
   grub2-mkconfig -o /etc/grub2.cfg
   ```

You can also invoke mongod with NUMA disabled or start the mongod process with the following numactl command:

```
numactl –interleave=all /usr/bin/mongod –f /etc/mongod.conf
```

zone_reclaim_mode on NUMA Systems

The Linux kernel can be inconsistent in enabling and disabling zone_reclaim_mode, which can lead to performance problems such as random huge CPU spikes, which cause large increases in latency and throughput.

Disable zone_reclaim_mode with one of the following commands:

```
echo 0 | sudo tee /proc/sys/vm/zone_reclaim_mode
```

```
sudo sysctl -w vm.zone_reclaim_mode=0
```

To ensure that zone_reclaim_mode is disabled, run the following command:

```
echo 0 > /proc/sys/vm/zone_reclaim_mode
```


Swappiness

Swappiness is a Linux kernel setting that influences the behavior of the virtual memory manager. Its value ranges from 0 to 100: the higher the value, the more strongly the system prefers swapping memory pages to disk over dropping pages from memory.

MongoDB performs best when swapping can be avoided or kept to a minimum, so set vm.swappiness to either 1 or 0 depending on your application needs and cluster configuration.

To check the current swappiness setting on your system, run the following command:

```
cat /proc/sys/vm/swappiness
```

To change swappiness, edit the `/etc/sysctl.conf` file:

```
vm.swappiness = 1
```

Run the command `sudo sysctl -p` to apply changes.

NTP Daemon

Use the Network Time Protocol (NTP) to synchronize time among your hosts. MongoDB, as a cluster, is dependent on time consistency across nodes, so you must run the NTP permanently on MongoDB hosts.

TCP Settings

Depending on your MongoDB settings you might need to adjust the Linux parameters. Make these changes in `/etc/sysctl.d/sysctl.conf`:

```
net.core.somaxconn = 4096
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 120
net.ipv4.tcp_max_syn_backlog = 4096
```


I/O Scheduler

For local block devices attached to a VM instance through the hypervisor, the guest operating system should use a noop scheduler for best performance. The noop scheduler allows the operating system to defer I/O scheduling to the underlying hypervisor.

To change this setting, edit the `/etc/default/grub` file and append `GRUB_CMDLINE_LINUX="elevator=noop"`.

Rebuild the grub file for the changes to take effect:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

For more information on Linux on AHV, see the Linux on AHV best practice guide.

9. MongoDB Installation on Linux

There are multiple ways to install MongoDB on Linux based on preference and requirements:

1. Install MongoDB using yum package manager.
2. Install MongoDB using tarball.
3. Install MongoDB using NDB.

Use Yum Package Manager

To use this method, you need to configure the MongoDB yum repository and create a /etc/yum.repos.d/mongodb-org.repo file to install MongoDB using yum commands. The following commands show an example:

```
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/\$releasever/mongodb-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
```

Alternatively, you can also download the .rpm files directly from the MongoDB repository. To install the files, run the following commands:

- For the latest version: `sudo yum install -y mongodb-org`
- For a specific version: `sudo yum install -y mongodb-org-5.0.13 mongodb-org-database-5.0.13 mongodb-org-server-5.0.13 mongodb-org-shell-5.0.13 mongodb-org-mongos-5.0.13 mongodb-org-tools-5.0.13`

You can specify any available MongoDB version; however, yum upgrades the packages when a new version is available. In order to prevent these automatic upgrades, pin the package using exclude in your `/etc/yum.conf` file:

```
exclude=mongodb-org,mongodb-org-database,mongodb-org-server,mongodb-org-shell,mongodb-org-mongos,mongodb-org-tools
```


Follow MongoDB documentation for installation steps based on your Linux distribution.

Use Tarball

While you can install MongoDB manually using a .tgz tarball, we recommend using the yum package manager. Using a package manager automatically installs all required dependencies.

To use the tarball method, download a specific version of MongoDB from the MongoDB Download Center, install all required dependencies, and run the installer.

Follow MongoDB documentation for installation steps based on your Linux distribution.

Use NDB

NDB enables you to easily register, provision, clone, and administer all your MongoDB databases on one or more Nutanix clusters with a single click.

NDB supports following MongoDB deployments on Nutanix:

- MongoDB single-node instance
- MongoDB replica set

NDB offers the following key services for MongoDB:

- One-click provisioning: With NDB you can easily provision database environments (production or otherwise) on your Nutanix clusters in a matter of minutes.
- Copy data management: NDB enables you to clone your MongoDB databases and refresh the database clones using snapshots or transaction logs. NDB uses storage-efficient Nutanix snapshots, decreasing the cost of storing multiple copies of your databases. NDB allows you to clone and refresh your clones to a specific point in time, making the clone and refresh operations highly granular.

- Database protection: NDB protects your database with full database-consistent backups in minutes, in addition to the following features:
  - One-click snapshots for simplifying database rollback
  - A robust time machine to explore your backup data
  - Instantaneous database backups with no overhead
- One-click patching: NDB offers out-of-band database patching to eliminate database configuration sprawl. NDB enables you to patch database server VMs and database server clusters with ease. You can apply the updates from an available software profile to the database server VM or cluster and perform the patching immediately or in a scheduled manner.

For more information about MongoDB with NDB, see the Nutanix Database Service MongoDB Database Management Guide.

10. Data Protection

Database backups are integral to ensuring business continuity and safeguarding data for any database management system. There are several tools available to back up MongoDB, such as mongodump or mongorestore, file system–level backups, and MongoDB Cloud Manager. You need to consider many factors —the size of your database, the data change rate, the RPO and RTO, cost, frequency of hot backups versus cold backups, and more—when you make decisions about backup method and strategy.

With Nutanix you can use NDB, which is an excellent method for managing MongoDB backups. NDB backs up your mission-critical databases using a combination of application-consistent snapshots and transaction logs.

NDB uses time machines to capture and maintain database snapshots and transaction logs of your source database as defined in the data access management policies and service-level agreement schedules (backup frequency and data retention). With time machines, you can use a snapshot ID, point-in-time recovery timestamp, or the latest snapshot to restore a single MongoDB instance or a replica set.

For more information, see NDB Time Machine Management.

11. MongoDB Sizing Best Practices

- Understand key performance metrics:
  - Read and write IOPS
  - Throughput: average reads and writes, peak throughput
  - Average versus peak latency
- Understand the workload type: whether the system is read-heavy, update-intensive, or both, you need to understand usage patterns.
- Calculate working set size as a percentage of the total data set. The working set is made of indexes and frequently accessed documents, and you need to know how big it is to calculate memory (RAM) size. For optimal performance, keep the working set in the WiredTiger cache.
- MongoDB data size: Estimate the number and size of documents per collection, calculate index size, and factor in WiredTiger compression (which provides storage savings). For an existing MongoDB instance, you can find this information under db.collection.stats.
- Consider data retention and future data growth.
- Deployment type (standalone, replica set, sharded): Size hardware based on the type and number of servers. Don't forget to factor in future growth.
- MongoDB relies heavily on caching, so memory is the most important hardware resource when it comes to MongoDB performance. In terms of hardware investments, prioritize memory.
- Always use all-flash NVMe or SSD. Don't use spinning drives (HDD).

12. Conclusion

Nutanix lets you virtualize MongoDB with the power and reliability of a high-performing underlying web-scale infrastructure. With the ability to incrementally add Nutanix compute and storage with one click, you can easily scale as your database grows—from a three-VM replica set in a single block up to multinode replica sets supporting global horizontal scale-out MongoDB systems. You can also scale vertically by upgrading virtual or physical hardware (memory or CPU) with zero downtime.

Nutanix eliminates the need for complex NAS and SAN environments by delivering locally attached storage for MongoDB servers. A cluster-wide SSD-backed hot data tier maintains low I/O latency and response times for running MongoDB, even in a demanding mixed-workload environment. For maximum cold-tier efficiency, Nutanix couples post-process compression with erasure coding.

Nutanix simplifies MongoDB management with Prism, streamlining database backups, test and QA environment rollouts, and seamless migration to the public cloud. Prism also provides cluster health overviews, full stack performance analytics, hardware and software alerting, storage utilization, and automated remote support.

Nutanix provides proven invisible infrastructure, allowing you to get the most out of critical applications like MongoDB while spending less time in the datacenter.

About Nutanix

Nutanix offers a single platform to run all your apps and data across multiple clouds while simplifying operations and reducing complexity. Trusted by companies worldwide, Nutanix powers hybrid multicloud environments efficiently and cost effectively. This enables companies to focus on successful business outcomes and new innovations. Learn more at Nutanix.com.

List of Figures