NUTANIX
YOUR ENTERPRISE CLOUD

Oracle on AHV

Nutanix Best Practices

Version 3.0 • January 2019 • BP-2073

Copyright

Contents

1. Executive Summary

The Nutanix Enterprise Cloud provides a complete datacenter infrastructure solution for Oracle databases, eliminating the complexities and inefficiencies of traditional multitier datacenter environments. Whether you are virtualizing critical or tier-1 Oracle databases or running them on bare metal, Nutanix solutions bring the predictable performance, scalability, and cost benefits of web-scale architecture to your transactional and analytical Oracle database environments.

The hypervisor-agnostic Nutanix solution (VMware vSphere, Microsoft Hyper-V, or Nutanix AHV) delivers powerful self-healing, data protection, and disaster recovery capabilities to keep your databases and applications running and your critical data well protected. The Nutanix Enterprise Cloud provides near-instantaneous local and remote backups using snapshots for offloading RMAN (Recovery Manager) backups to tape and disk or WORM (write once, read many) for offsite backup. You can also use a Nutanix clone snapshot to easily refresh production-based test and development Oracle instances.

This best practices guide recommends Nutanix AHV cluster settings for running Oracle databases, as well as ESXi settings for iSCSI, networking, LUNs, Linux OS, and Oracle ASM.

2. Introduction

2.1. Audience

This best practices guide is part of the Nutanix Solutions Library for Oracle and is intended for individuals designing and maintaining AHV-based Nutanix solutions for a single instance of Oracle Database or Real Application Cluster (RAC). Readers of this guide should be familiar with Oracle administration and the Linux operating system.

2.2. Purpose

This document provides design, configuration, and optimization guidelines for a single instance of Oracle Database or RAC running on Nutanix AHV. In this document, we cover the following topics:

- Nutanix Enterprise Cloud overview.
- AHV best practices.
- Linux Operating System best practices.
- Volume (vDisks) configuration.
- Linux disk device settings.
- Logical Volume Manager (LVM) configuration.
- Oracle Database best practices.
- Windows operating system best practices.

For information specific to Oracle on VMware vSphere, please refer to the Oracle on vSphere best practices guide.

Table 1: Document Version History

Version Number	Published	Notes
1.0	April 2017	Original publication.
1.1	January 2018	Feature updates, including Erasure Coding.

Version Number	Published	Notes
1.2	March 2018	Added iSCSI script details.
1.3	June 2018	Updated kernel settings.
1.4	July 2018	Updated the Nutanix AHV Cluster Networking section.
1.5	August 2018	Updated the Linux Operating System Best Practices section.
2.0	November 2018	Updated product information and the Nutanix AHV Best Practices section.
3.0	January 2019	Updated the Linux Disk Device Settings section and added sections on Oracle Clusterware CSS Timeout Settings and Windows Operating System Best Practices.

3. Nutanix Enterprise Cloud Overview

Nutanix delivers a web-scale, hyperconverged infrastructure solution purpose-built for virtualization and cloud environments. This solution brings the scale, resilience, and economic benefits of web-scale architecture to the enterprise through the Nutanix Enterprise Cloud Platform, which combines three product families—Nutanix Acropolis, Nutanix Prism, and Nutanix Calm.

Attributes of this Enterprise Cloud OS include:

- Optimized for storage and compute resources.
- Machine learning to plan for and adapt to changing conditions automatically.
- Self-healing to tolerate and adjust to component failures.
- API-based automation and rich analytics.
- Simplified one-click upgrade.
- Native file services for user and application data.
- Native backup and disaster recovery solutions.
- Powerful and feature-rich virtualization.
- Flexible software-defined networking for visualization, automation, and security.
- Cloud automation and life cycle management.

Nutanix Acropolis provides data services and can be broken down into three foundational components: the Distributed Storage Fabric (DSF), the App Mobility Fabric (AMF), and AHV. Prism furnishes one-click infrastructure management for virtual environments running on Acropolis. Acropolis is hypervisor agnostic, supporting three third-party hypervisors—ESXi, Hyper-V, and XenServer—in addition to the native Nutanix hypervisor, AHV.

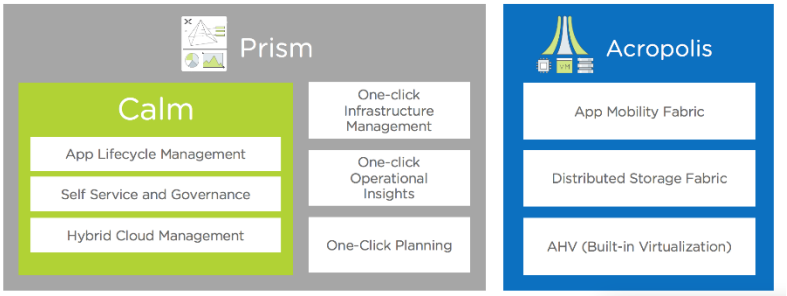

Figure 1: Nutanix Enterprise Cloud

3.1. Nutanix Acropolis Architecture

Acropolis does not rely on traditional SAN or NAS storage or expensive storage network interconnects. It combines highly dense storage and server compute (CPU and RAM) into a single platform building block. Each building block delivers a unified, scale-out, shared-nothing architecture with no single points of failure.

The Nutanix solution requires no SAN constructs, such as LUNs, RAID groups, or expensive storage switches. All storage management is VM-centric, and I/O is optimized at the VM virtual disk level. The software solution runs on nodes from a variety of manufacturers that are either all-flash for optimal performance, or a hybrid combination of SSD and HDD that provides a combination of performance and additional capacity. The DSF automatically tiers data across the cluster to different classes of storage devices using intelligent data placement algorithms. For best performance, algorithms make sure the most frequently used data is available in memory or in flash on the node local to the VM.

To learn more about the Nutanix Enterprise Cloud, please visit the Nutanix Bible and Nutanix.com.

4. Why Run Oracle on Nutanix

Digital transformation demands that IT evolve beyond offering basic services and become more agile, shortening the time to production and increasing scalability. Furthermore, business-critical applications such as Oracle databases require us to think beyond vanilla provisioning of the operating system.

Virtualization addresses these needs well, but sometimes environments must still use physical Oracle Database instances because of licensing or organizational requirements. The Nutanix Enterprise Cloud supports both contexts, bringing web-scale benefits to virtualized and bare metal deployments of Oracle databases.

Designed to run any application, the Nutanix Enterprise Cloud converges storage and compute within the same appliance, eliminating the complexity of separate standalone storage solutions. Nutanix supports VMware vSphere, Microsoft Windows Server 2012 R2 with Hyper-V, and Nutanix AHV, allowing customers to choose the right hypervisor for their needs.

Administrators can deploy Oracle Database on bare metal through Nutanix Volumes, which exports iSCSI volumes to a single Oracle instance or to Oracle RAC instances running on physical servers. Nutanix Volumes can also export volumes directly to VMs running on the same Nutanix cluster via Nutanix volume groups. With Volumes, not only do your nonvirtualized workloads get all the benefits of hyperconverged environments, such as backup and recovery, disaster recovery, snapshots, and clones, they actually perform better within the Nutanix cluster.

Running Oracle on the Nutanix Enterprise Cloud offers several advantages:

- Low latency and predictable performance, so DBAs can start small and then linearly and nondisruptively scale both performance and capacity as needs grow.
- Ability to handle a wide range of database workload types, including transactional, analytical, or a mix of the two on the same platform.
- Datacenter infrastructure deployment within hours, resulting in faster time to production.
- High availability and data redundancy, allowing the Oracle database and supporting applications to stay up and running even if the underlying hardware has issues.
- Integrated snapshot, remote replication, and metro-level availability functionality to protect the database, in addition to Oracle's native data protection capabilities.
- Simplified management, relieved of multipathing, zoning, and masking.
- Support of Oracle Automatic Storage Management (ASM) without any caveats or restrictions.

5. Nutanix AHV Best Practices

This section discusses selecting Nutanix nodes for better performance, in terms of CPU clock speed, while lowering Oracle licensing costs. We also discuss networking and storage best practices for a Nutanix cluster running on AHV, as a proper setup eliminates infrastructure performance bottlenecks. For more detailed information on AHV, please see our AHV best practices guide.

For vSphere-specific guidance, refer to the Oracle on vSphere best practices guide.

5.1. Nutanix Model Configurations

For any Oracle workload, select Nutanix cluster nodes with a higher clock speed and fewer cores. Higher clock speed means more I/O and lower latency, while having fewer cores reduces licensing costs. We recommend selecting nodes with at least 2.6 GHz CPU for Oracle databases.

5.2. Nutanix AHV Cluster Networking

When setting up a Nutanix AHV cluster, use only the 10 GbE or 40 GbE network interfaces for the br0-up. Set the balance mode for br0-up to balance-slb and next rebalance to 30 seconds.

Log on to a CVM and issue the following commands:

- Add 10 GbE NICs to bond0:

```
nutanix@CVM:~$ allssh manage_ovs --bridge_name br0 --bond_name br0-up --interfaces 10g
 update_uplinks
```

- Set bond mode balance to balance-slb:

```
nutanix@CVM:~$ allssh ssh root@192.168.5.1 "ovs-vsctl set port br0-up bond_mode=balance-slb"
```

- Set next rebalance to 30 seconds:

```
nutanix@CVM:~$ allssh ssh root@192.168.5.1 "ovs-vsctl set port br0-up other_config:bond-
rebalance-interval=30000"
```

Note: If you are upgrading AOS, be sure to check the release notes for your target AOS version. Consider engaging Nutanix Support if you encounter network connectivity issues when hosting or migrating Oracle RAC VMs.

5.3. AHV Cluster Administration

VM-host affinity and antiaffinity rules are supported with Oracle RAC on Nutanix. However, we do not recommend using antiaffinity rules with Oracle RAC VMs, as such a policy may cause issues during maintenance or other HA events. An antiaffinity policy is not necessary if a node does not have enough resources to run two RAC VMs at the same time, as those VMs are unable to reside together on that node regardless of the policy. For more information on AHV VM-host affinity and antiaffinity rules, refer to the Prism Central Guide for the version of AOS that you are using.

5.4. Nutanix Acropolis Distributed Storage Fabric

Create one storage pool when setting up storage for Oracle Database. Within that pool, create a storage container with a replication factor of at least 2 for data redundancy. This setting guarantees that the cluster loses no data in the event of a single-node failure. For more efficient space usage, enable inline compression on the container.

Note: Do not enable deduplication. If Oracle RAC is configured on the cluster, disable Nutanix Shadow Clones.

The following table lists the recommended Nutanix storage configuration.

Table 2: Nutanix DSF Configuration

Nutanix	Values	Comments
Number of Nutanix nodes	Minimum 3	Required for replication factor 2.
Nutanix storage pool	1	Standard AHV practice.
Container	1	Standard practice for both VM and Oracle data.
Compression	Enable	Standard practice for space savings.
Erasure coding	Disable	Not recommended for Oracle but can be enabled if usage capacity is a constraint.
Deduplication	Disable	Not recommended for Oracle.
Shadow Clones	Disable	Not recommended with Oracle RAC.

6. Linux Operating System Best Practices

When running Oracle Database on Linux, there are many areas to consider, such as performance, data protection, and efficiency. This section discusses the best practices for Oracle Linux, Red Hat Enterprise Linux, and CentOS.

> **Note:** If you are deploying the Oracle Linux operating system on AHV and your VM does not boot due to an unexpected page fault, refer to KB 4650.

6.1. Kernel Settings for Oracle

Following are the parameter settings for Oracle Database. These settings derive from the Oracle preinstall package. After entering them into the /etc/sysctl.conf file, make sure to run the command **sysctl -p** to make them effective. For more information about these settings, please visit the Oracle page Configuring Kernel Parameters and Resource Limits.

sysctl Settings

```
## oracle-rdbms-server-12cR1-preinstall
fs.file-max = 6815744
kernel.sem = 250 32000 100 128
kernel.shmmni = 4096
kernel.shmall = 1073741824
kernel.shmmax = 4398046511104
kernel.panic_on_oops = 1
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
net.ipv4.conf.all.rp_filter = 2
net.ipv4.conf.default.rp_filter = 2
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 9000 65500
```

> **Note:** The **## vm.nr_hugepages** setting is reserved for enabling HugePages. Please refer to the Configure HugePages section for more information.

limits.conf Settings

Following are the shell settings for the “grid” and “oracle” user. You can use the settings as shown or adjust them to fit your environment. For more information about these settings, please visit the Oracle kernel parameters page cited above.

```
grid  soft    nofile  131072
grid  hard    nofile  131072
grid  soft    nproc   131072
grid  hard    nproc   131072
oracle  soft    nofile  131072
oracle  hard    nofile  131072
oracle  soft    nproc   131072
oracle  hard    nproc   131072
oracle  soft    core    unlimited
oracle  hard    core    unlimited
oracle   soft   stack    10240
oracle   hard   stack    32768
```

The **## memlock** setting is reserved for enabling HugePages. Set it to a number slightly lower than the actual memory. Following is an example for a VM with 32 GB of physical memory. Please refer to the Configure HugePages section below for information on enabling HugePages.

```
oracle  soft    memlock 31457280
oracle  hard    memlock 31457280
```

Default Profile Settings

Append the following to /etc/profile:

```
if [ $USER = "oracle" ] || [ $USER = "grid" ] ; then
   if [ $SHELL = "/bin/ksh" ]; then
      ulimit -p 16384
      ulimit -n 65536
   else
      ulimit -u 16384 -n 65536
   fi
fi
```

6.2. Configure HugePages

Nutanix recommends using HugePages to improve performance, especially in a data warehouse environment. For large SGA (system global area) sizes, HugePages substantially improves virtual memory management. Without HugePages, the SGA memory is divided into 4 K pages that the Linux kernel must manage. With HugePages, the page size increases anywhere from 2 MB to 256 MB, depending on the kernel version and the hardware architecture, which reduces the total number of pages the kernel manages. In addition, the HugePages stay resident; they cannot be swapped out. Please refer to Oracle documentation for more information.

Use the following steps to configure HugePages.

- Edit the **memlock** setting in the /etc/security/limits.conf file. Set **memlock** in KB, with a value slightly lesser than the available RAM. For example, if you have 32 GB RAM installed, add the following:

```
oracle    soft    memlock 31457280
oracle    hard    memlock 31457280
```

- Display the value of **Hugepagesize**.

```
[root@oracle1 ~]# grep Hugepagesize /proc/meminfo
Hugepagesize:           2048 kB
```

- Create a shell script with the code displayed in the figure below to calculate the number of HugePages. Alternatively, you can download a script from Oracle.

```
#!/bin/bash
#
# hugepages_settings.sh
#
# Linux bash script to compute values for the
# recommended HugePages/HugeTLB configuration
#
# Note: This script does calculation for all shared memory
# segments available when the script is run, no matter it
# is an Oracle RDBMS shared memory segment or not.
# Check for the kernel version
KERN=`uname -r | awk -F. '{ printf("%d.%d\n",$1,$2); }'`
# Find out the HugePage size
HPG_SZ=`grep Hugepagesize /proc/meminfo | awk {'print $2'}`
# Start from 1 pages to be on the safe side and guarantee 1 free HugePage
NUM_PG=1
# Cumulative number of pages required to handle the running shared memory segments
for SEG_BYTES in `ipcs -m | awk {'print $5'} | grep "[0-9][0-9]*"`
do
   MIN_PG=`echo "$SEG_BYTES/($HPG_SZ*1024)" | bc -q`
   if [ $MIN_PG -gt 0 ]; then
      NUM_PG=`echo "$NUM_PG+$MIN_PG+1" | bc -q`
   fi
done
# Finish with results
case $KERN in
   '2.4') HUGETLB_POOL=`echo "$NUM_PG*$HPG_SZ/1024" | bc -q`;
          echo "Recommended setting: vm.hugetlb_pool = $HUGETLB_POOL" ;;
   '2.6' | '3.8' | '3.10' | '4.1' ) echo "Recommended setting: vm.nr_hugepages = $NUM_PG" ;;
    *) echo "Unrecognized kernel version $KERN. Exiting." ;;
esac
# End
```

Figure 2: Shell Script for Calculating HugePages

- Run the script to get the number of HugePages. Use that value for the parameter **vm.nr_hugepages** in the /etc/sysctl.conf file.

> **Note:** Make sure Oracle processes are running before executing this script.

```
[root@oracle1 ~1]# hugepages_setting.sh

Recommended setting: vm.nr_hugepages = 10503
```

- Disable Automatic Memory Management (AMM) on the database instance if it is enabled. Although AMM is the simplest way to manage instance memory because it automatically manages and tunes the database, it cannot be used with HugePages.
- Set the **SGA_TARGET** and **PGA_AGGREGATE_TARGET** parameters. For example, if you decide to reserve 20 percent of the total physical memory for the operating system and other non-Oracle applications, you can allocate the remaining 80 percent of memory as described in the table and bullet list below.

Table 3: Guidelines for Oracle SGA and PGA

OLTP	DSS
SGA_TARGET = (total_mem * 0.8) * 0.8, where total_mem is the total amount of physical memory available on the system	SGA_TARGET = (total_mem * 0.8) * 0.5, where total_mem is the total amount of physical memory available on the system
PGA_AGGREGATE_TARGET = (total_mem * 0.8) * 0.2, where total_mem is the total amount of physical memory available on the system	PGA_AGGREGATE_TARGET = (total_mem * 0.8) * 0.5, where total_mem is the total amount of physical memory available on the system

Note: Use these recommendations as guidelines only. You must monitor the memory usage and adjust accordingly based on your workload.

- If you are running Oracle 11.2.0.2 or later, you can set **USE_LARGE_PAGES** to **ONLY**, which prevents the database from starting if it is not backed up by HugePages.

```
PRODDB> alter system set USE_LARGE_PAGES=ONLY scope=spfile;
```

- Disable the Transparent HugePages option if it is not already disabled. Transparent HugePages differs from HugePages in that it dynamically allocates memory via a **khugepaged** kernel thread, rather than at boot time, as HugePages does. Transparent HugePages does not work well with Oracle databases, however; it can cause performance problems and node reboots in RAC installations.
- Edit the /etc/default/grub file and append transparent_hugepage=never to the GRUB_CMDLINE_LINUX line.

```
GRUB_CMDLINE_LINUX="crashkernel=auto rhgb quiet transparent_hugepage=never"
```

- Run the grub2-mkconfig -o /boot/grub2/grub.cfg command.

```
[root@oracle1 ~1]# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Reboot.

6.3. Network Configuration

The table below shows the required networking for setting up Oracle servers.

Table 4: Network Settings

Oracle	Speed	Purpose
Oracle Public Network	10 GbE	Public or Management Network
Oracle Private Network	10, 25, or 40 GbE	RAC Interconnect
Oracle Data Network	10, 25, or 40 GbE	iSCSI Network with Nutanix Volumes

On the Oracle VM, add **NOZEROCONF=yes** to the /etc/sysconfig/network file.

6.4. Configure Volumes (vDisks)

In a traditional three-tier architecture, DBAs often work with storage administrators to create a custom storage design to suit their Oracle environment, including designating the RAID level and block size for the Oracle datafiles or online redo logs. This process can become cumbersome to manage if there are multiple databases.

The Nutanix Enterprise Cloud eliminates all such problems associated with RAID and block size. Once you create the required number of volumes for your Oracle database, you're ready to deploy.

The table below shows the recommended number of vDisks for Oracle.

Table 5: Nutanix Volume Configuration for Oracle

Number of vDisks	Purpose	Comment
1	Boot disk	Can be used with LVM or Standard partition
8	Database datafiles / control files / redo log files	Can be used with Oracle ASM or file system with LVM
4	Database archive log files	Can be used with Oracle ASM or file system with LVM
4	Database RMAN backup files	Can be used with Oracle ASM or file system with LVM

Note: Nutanix does not recommend using a single vDisk for Oracle databases, as it causes poor performance.

The following table provides Oracle ASM disk group layouts.

Table 6: ASM Disk Group Layout

Number of vDisks	Purpose	Comment
2 volumes	OCR ASM DG (Normal Redundancy)	Use with vDisk or volume group
8 volumes	DATA ASM DG (External Redundancy)	Use with vDisk or volume group
4 volumes	FRA ASM DG (External Redundancy)	Use with vDisk or volume group

There are two ways to create volumes on Nutanix for your Oracle databases:

1. Nutanix native vDisks.
2. Nutanix volume groups.

6.5. Nutanix Native vDisks

The Nutanix native vDisk option simplifies VM administration, as it does not require Nutanix volume groups. When creating a VM using Prism, simply add more vDisks to the VM for the database just as you would add a vDisk to the VM for the boot disk. This is the best option if you have a database that does not have intensive I/O requirements.

6.6. Nutanix Volume Groups

Alternatively, you can attach Nutanix volume groups, which are a collection of vDisks, to the VMs. Volume groups enable you to separate the data vDisks from the VM's boot vDisk. This separation allows you to put the volume group that contains only the database vDisks into a Nutanix protection domain that you can then snapshot and clone. In addition, volume groups let you configure clustering for shared-disk access across multiple VMs. Such clustering includes Oracle RAC. To attach the volume group to multiple RAC VMs, you must create it with an attribute of **shared=true**, using the command line interface (aCLI); if using Prism, answer **yes** when attaching to a second RAC VM.

There are two methods of using Nutanix volume groups:

1. Default volume group.
2. Volume group with load balancing.

Default Volume Group

This type of volume group does not provide vDisk load balancing in a Nutanix cluster. All vDisks in this type of volume group have a single Nutanix CVM providing an I/O path at any one time. As

an example, you could have a four-node Nutanix cluster with a volume group that includes eight vDisks attached to a VM. A single node CVM owns the vDisks in this volume group. All I/O to the eight vDisks goes through this CVM.

Volume Group with Load Balancing (VGLB)

This new feature is available with AHV only, starting with AOS 5.6. It is not available with ESX or Hyper-V hypervisors (volume groups with Nutanix Volumes and the iSCSI in-guest initiator are load balanced across Stargates by default). VGLB allows AOS to load balance the vDisks across all CVMs in the cluster. In a four-node Nutanix cluster with eight vDisks, VGLB spreads the vDisks across all four CVMs (two vDisks per CVM), improving the volume group's performance. Use this type of volume group if your Oracle database requires more performance than the default volume group can deliver.

To enable VGLB, you must use the Nutanix aCLI. The figure below includes the command for enabling VGLB when creating a volume group.

```
<acropolis> vg.create vgtest load_balance_vm_attachments=true
vgtest: complete
```

Figure 3: Command for Enabling VGLB While Creating a Volume Group

> **Note:** If you want to use Nutanix Volumes for your Oracle Database, please refer to the Nutanix Volumes best practices guide. As an additional resource, the Linux on Nutanix AHV best practices guide provides detailed instructions and covers setting up iSCSI ifaces and dm-multipath.

Linux Disk Device Settings

To optimize your Oracle Database performance, change the following disk parameters from their default settings.

- Change the max_sector_kb to **1024** (the default is 512). With a newer Linux distribution, the default may already be set to 1024. Change the disk timeout to **60** (default is 30).

```
[root@oracle1 ~]# lsscsi | grep NUTANIX | awk '{print $NF}' | awk -F"/" '{print $NF}' | grep
 -v "-" | while read LUN

do

   echo 1024 > /sys/block/${LUN}/queue/max_sectors_kb

done

[root@oracle1 ~]# lsscsi | grep NUTANIX | awk '{print $NF}' | awk -F"/" '{print $NF}' | grep
 -v "-" | while read LUN

do

   echo 60 > /sys/block/${LUN}/device/timeout

done
```

Either put this command in the /etc/rc.local file, so that it runs the next time the server reboots, or use UDEV rules. For UDEV rules, you can create a file with the following content under the /etc/udev/rules.d directory.

```
ACTION=="add", SUBSYSTEMS=="scsi", ATTRS{vendor}=="NUTANIX ", ATTRS{model}=="VDISK", RUN+="/
bin/sh -c 'echo 1024 >/sys$DEVPATH/queue/max_sectors_kb'"

ACTION=="add", SUBSYSTEMS=="scsi", ATTRS{vendor}=="NUTANIX ", ATTRS{model}=="VDISK", RUN+="/
bin/sh -c 'echo 60 >/sys$DEVPATH/device/timeout'"
```

- Change the disk io_scheduler to **noop**. The default is **deadline**.

```
[root@oracle1 ~]# lsscsi | grep NUTANIX | awk '{print $NF}' | awk -F"/" '{print $NF}' | grep
 -v "-" | while read LUN

do

   echo noop > /sys/block/${LUN}/queue/scheduler

done
```

If you put this command in the /etc/rc.local file, it runs the next time the server reboots. Alternatively, you can put it in the **grub configuration** file. Use the following steps for GRUB2 configuration.

- Edit the /etc/default/grub file, add “elevator=noop” to the GRUB_CMDLINE_LINUX line, and save the file.

```
GRUB_CMDLINE_LINUX="crashkernel=auto rhgb quiet elevator=noop
```

Figure 4: Add elevator=noop to /etc/default/grub File

- Disable **transparent_hugepage**.
- In the same edit window, add “transparent_hugepage=never” to the end of the elevator=noop line and save the file.

- If the Linux kernel supports the blk_mq (block multiqueue) option, add the parameter “scsi_mod.use_blk_mq=1” to enable blk_mq and remove the elevator=noop option.

Figure 5: Enabling Block Multiqueue

- Run **grub2-mkconfig** to generate a new grub file and reboot.

```
[root@oracle1 ~]# grub2-mkconfig –o /boot/grub2/grub.cfg
```

6.7. Configure Logical Volume Manager (LVM)

This section discusses Linux Logical Volume Manager (LVM) best practices. When creating a file system for databases, be sure to use LVM striping across all vDisks for a specific mount point.

The following example shows how to create an LVM volume group with eight vDisks for a database datafiles mount point.

- Create the physical volumes.

```
[root@localhost ~]# pvcreate /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg /dev/sdh /dev/sdi /
dev/sdj
  Physical volume "/dev/sdc" successfully created
  Physical volume "/dev/sdd" successfully created
  Physical volume "/dev/sde" successfully created
  Physical volume "/dev/sdf" successfully created
  Physical volume "/dev/sdg" successfully created
  Physical volume "/dev/sdh" successfully created
  Physical volume "/dev/sdi" successfully created
  Physical volume "/dev/sdj" successfully created
```

- Create a volume group.

```
[root@localhost ~]# vgcreate vgdata /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg /dev/sdh /
dev/sdi /dev/sdj
  Volume group "vgdata" successfully created
```

- Create a logical volume. Make sure to use the -i (lowercase i) option to specify the number of vDisks to stripe across and the -I (uppercase i) option to specify the stripe size. The recommended stripe size is **512** KB.

```
[root@localhost ~]# lvcreate -l 383994 -i 6 -I 512 -n vol1 vgdata
  Logical volume "vol1" created.
```

- Create the file system. In this example, we are creating an ext4 file system.

```
[root@localhost ~]# mkfs.ext4 /dev/vgdata/vol1
```

- Mount the ext4 file system. If you are creating an XFS file system, please refer to the Mount Options table below.

```
[root@localhost ~]# mount /dev/vgdata/vol1 /u01/oradata -o
 noatime,nodiratime,discard,barrier=0
```

> **Note:** You can use **xfs_admin** for xfs or **e2label** for ext4 to label your file systems with a friendly name. Use the **LABEL=** option in the /etc/fstab file for ease of management.

The table below lists the recommended mount options for ext4 and XFS file systems.

Table 7: Mount Options for ext4 and XFS File Systems

File System Type	Mount Options
ext4	noatime,nodiratime,discard,barrier=0
XFS	noatime,nodiratime,discard,nobarrier,logbufs=8

6.8. Kernel Settings for the Proc File System

In addition to configuring LVM, you must also set the following kernel parameters for optimal performance. These settings can be found at the /proc/sys/vm documentation page.

Table 8: Proc File System Kernel Parameters

Settings	Value	Purpose
vm.overcommit_memory	1	Disable memory overcommit handling.
vm.dirty_background_ratio	5	As a percentage. Contains the number of pages at which the background kernel flusher threads will start writing out dirty data.
vm.dirty_ratio	15	As a percentage. Contains the number of pages at which a process that is generating disk writes will start writing out dirty data itself.

Settings	Value	Purpose
vm.dirty_expire_centisecs	500	This tunable is used to define when dirty data is old enough to be eligible for writeout by the kernel flusher threads.
vm.dirty_writeback_centisecs	100	The kernel flusher threads periodically wake up and write old data out to disk. This tunable expresses the interval between those wakeups.
vm.swappiness	0	This tunable defines how aggressively the kernel swaps memory pages. 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.

7. Oracle Database Best Practices

In this section we discuss Oracle best practices regarding storage (disk I/O), specifically Oracle ASM settings and Oracle Database. For information on Oracle parameters not related to storage, please visit the Oracle documentation page.

7.1. ASM Settings

The following table shows the recommended settings for all ASM disk groups.

Table 9: Oracle ASM Settings

Settings	Values
ASM Allocation Unit (AU) size	1 MB
ASM OCR disk group redundancy	Normal/High
ASM DATA disk group redundancy	External
ASM FRA disk group redundancy	External

7.2. Oracle Clusterware CSS Timeout Settings

The CSS misscount parameter represents the maximum time, in seconds, that the system can tolerate a missed network heartbeat before beginning to reconfigure the cluster to evict the node. In the Linux operating system, this parameter is set to 30 seconds by default. You can change the CSS misscount timeout to 60 seconds to avoid node eviction in case of network or I/O contention. For more information, please refer to Oracle MOS ID 294430.1.

7.3. Database Settings for Oracle Single Instance and RAC

The following table shows the recommended settings for Oracle single instances and RAC databases.

Table 10: Oracle Database Settings

Settings	Value	Purpose
filesystemio_options	setall	This parameter allows directIO and AsyncIO when using a file system such as ext. It is not required when using Oracle ASM.
SGA_TARGET	Refer to HugePages section	
PGA_AGGREGATE_TARGET	Refer to HugePages section	
RAC Interconnect	10, 25, or 40 GbE	Private interconnect between RAC nodes.
DB_FILE_MULTIBLOCK_READ_COUNT	128	This parameter minimizes I/O during table scans. Refer to Oracle Doc ID 1298860.1 for more information.
PARALLEL_THREADS_PER_CPU	1	Degree of parallelism. On Nutanix, it is 1 per vCPU.

8. Windows Operating System Best Practices

When running Oracle Database on Windows, be sure to use the latest VirtIO driver from Nutanix.

8.1. NTFS Recommendations

- Create striped volumes for an Oracle database. At minimum, you need to have two striped volumes: one for Oracle datafiles and online redo logs, and one for archived logs and any RMAN backup sets.
- Format the volume with a 64 K block size.
- Mount the volumes on a drive letter.

8.2. ASM Recommendations

- Configure the vDisks as **New Simple Volume** and do not assign a drive letter or drive path.
- Use Oracle asmtool.exe to label the disks.
- Create a DATA disk group and FRA disk group.

8.3. Additional Recommendations

- Use VGLB if the Nutanix AOS version supports it (AOS 5.6 and later).
- To determine the number of vDisks, refer to the Configure Volumes section above for guidelines.
- For networking guidelines, refer to the Network Configuration section above.
- For help with database settings, refer to the Database Settings section above.

9. Conclusion

This best practices guide outlines our recommended settings for Oracle databases on Nutanix AHV, including running Oracle RAC with vDisks. Nutanix web-scale engineering offers an array of benefits to both physical and virtualized instances of Oracle databases, such as low-latency storage performance, linear scalability, high availability, and simplified management.

Moreover, by removing the complexity of constantly managing and optimizing the underlying compute and storage architecture, DBA and infrastructure administrators can focus on higher value tasks, such as application enhancements and accelerating time to production.

Engage with Nutanix experts on the Nutanix NEXT Community, on Twitter @nutanix, or at your local Nutanix Connect User Groups to learn more about the benefits of the Nutanix Enterprise Cloud for Oracle databases and other business-critical applications. Send us an email at info@nutanix.com to set up your own customized briefing that includes sizing, TCO analysis, data protection, and disaster recovery design.

Appendix

References

1. Linux on Nutanix AHV best practices guide
2. Nutanix Volumes best practices guide

About Nutanix

Nutanix makes infrastructure invisible, elevating IT to focus on the applications and services that power their business. The Nutanix Enterprise Cloud OS 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

List of Tables