v1.6 | August 2022 | BP-2105

BEST PRACTICES

Linux on Nutanix AHV

NUTANIX
YOUR ENTERPRISE CLOUD

Copyright

Contents

1. Executive Summary

Nutanix provides a complete software-defined datacenter infrastructure solution for applications running on Linux, eliminating the complexities and inefficiencies of traditional multitier datacenter environments. Whether you are virtualizing critical tier-1 applications or running them on bare metal, Nutanix solutions bring the predictable performance, availability, scalability, and cost benefits of web-scale architecture to your Linux applications.

Nutanix provides the freedom to choose the hypervisor you want for your applications, including Nutanix AHV, VMware vSphere, and Microsoft Hyper-V. Solutions built on the Nutanix enterprise cloud deliver the performance required for business-critical applications, while powerful self-healing, data protection, and disaster recovery capabilities keep your applications running and your vital data well protected. Nutanix provides near-instantaneous local and remote data protection using snapshots. You can use these snapshots for offloading database backups to tape and disk or WORM (write once, read many) for offsite retention. Nutanix also enables one-click cloning, so administrators can easily and quickly apply these snapshots to refresh a test or development database from production.

This best practice guide collects recommended Nutanix AHV cluster settings and Linux OS settings, which include those for Oracle Linux, Red Hat Enterprise Linux, and CentOS. It covers vDisk (LUN) configuration and settings, Logical Volume Manager (LVM) configuration, file system settings for ext4 and XFS, and kernel parameters. Most of the recommendations in this guide also apply generally to other hypervisor environments on Nutanix.

2. Introduction

Audience

This best practice guide is part of the Nutanix Solutions Library. We wrote it for architects and system administrators responsible for designing and maintaining a robust Linux environment and its infrastructure. Readers should already be familiar with AHV-based Nutanix infrastructure and the Linux OS.

Purpose

In this document, we cover the following topics:

- An overview of Nutanix.
- Nutanix AHV best practices.
- Linux OS kernel settings.
- Linux networking.
- Nutanix Volumes.
- Configuration for volumes and vDisks.
- LVM configuration.
- File system mount options for ext4 and XFS.
- Linux disk device settings.

Document Version History

Version Number	Published	Notes
1.0	June 2018	Original publication.

Version Number	Published	Notes
1.1	July 2018	Updated the Nutanix AHV Cluster Networking and Nutanix Volumes sections.
1.2	January 2019	Updated product information and the Linux Disk Device Settings section.
1.3	December 2019	Updated LVM creation instructions.
1.4	June 2020	Updated product information and jumbo frames guidance.
1.5	March 2021	Updated load balancing recommendations.
1.6	August 2022	Updated the Logical Volume Manager Configuration and File System Mount Options sections.

3. Nutanix AHV Best Practices

This section discusses how to select Nutanix nodes that have the CPU clock speed you need for optimal performance. We also discuss networking and storage best practices for a Nutanix cluster running on AHV, as proper configuration eliminates infrastructure performance bottlenecks. For more detailed information on AHV, review the AHV best practice guide.

Nutanix Model Configurations

For a database 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 of CPU.

Nutanix AHV Cluster Networking

When you set up a Nutanix AHV cluster, use only the 10 GbE, 25 GbE, or 40 GbE network interfaces for the bond0. Set the balance mode for br0-up to balance-slb, with the next rebalance at 30 seconds.

Log on to a CVM and issue the following commands:

- Add the 10 GbE NICs to bond0:

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

- Set the bond balance mode 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"
```


Nutanix Distributed Storage

Create a storage container with a replication factor of at least 2 for data redundancy. This setting guarantees that the cluster doesn't lose any data in the event of a single-node failure. For more efficient space usage, enable inline compression, deduplication, or erasure coding on the container.

> Note: Don't enable deduplication on the container if you run any database workload.

The following table details the recommended Nutanix storage configuration.

Table: Recommended Nutanix Storage Configuration

Nutanix	Value	Rationale
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 database data.
Compression	Enable	Standard practice for space savings.
Erasure coding	Disable	Not recommended for databases but can be enabled if usage capacity is a constraint.
Deduplication	Disable	Not recommended for databases.

4. Linux Operating System Kernel Settings

The following table outlines the recommended OS kernel settings for a Linux VM. Specific applications or databases may require additional settings for optimal performance, so consult their documentation. You can find the following settings in the documentation for /proc/sys/vm/*.

Table: Kernel Settings

Settings	Value	Purpose
vm.overcommit_memory	1	Disables memory overcommit handling.
vm.dirty_background_ratio	5	Expressed as a percentage. Contains the number of pages at which the background kernel flusher threads start writing out dirty data.
vm.dirty_ratio	15	Expressed as a percentage. Contains the number of pages at which a process that is generating disk writes starts writing out dirty data itself.
vm.dirty_expire_centisecs	500	This tunable defines 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.

Settings	Value	Purpose
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.

5. Linux Networking

When you create VMs for applications or databases, Nutanix strongly recommends that you use at least 10 GbE on the Nutanix cluster. If you require high bandwidth and low latency, Nutanix recommends using multiple interfaces (10 GbE, 25 GbE, or 40 GbE) in a bonded fashion if your network switches support doing so.

6. Nutanix Volumes

Nutanix Volumes allows bare-metal servers or VMs to access vDisks in a Nutanix volume group (VG) natively on AHV or through iSCSI. For client iSCSI access, the Nutanix cluster provides a single data services IP address. If you require high bandwidth and low latency, you can use volume groups with load balancing (VGLB) on AHV or Nutanix Volumes iSCSI access to achieve the performance you need. For more information about Volumes, see the Nutanix Volumes section of the Prism Web Console Guide.

Jumbo Frames

The Nutanix CVM uses the standard Ethernet MTU (maximum transmission unit) of 1,500 bytes for all the network interfaces by default. The standard 1,500 byte MTU delivers excellent performance and stability. Nutanix does not support configuring the MTU on a CVM's network interfaces to higher values.

You can enable jumbo frames (MTU of 9,000 bytes) on the physical network interfaces of AHV, ESXi, or Hyper-V hosts and user VMs if the applications on your user VMs require them. If you choose to use jumbo frames on hypervisor hosts, be sure to enable them end to end in the desired network and consider both the physical and virtual network infrastructure impacted by the change.

iSCSI Settings

If you use Nutanix Volumes, configure the following iSCSI settings on the guest OS in the `/etc/iscsi/iscsid.conf` file and restart the `iscsid` process.

```
node.session.timeo.replacement_timeout = 120
node.conn[0].timeo.noop_out_interval = 5
node.conn[0].timeo.noop_out_timeout = 10
node.session.cmds_max = 2048
node.session.queue_depth = 1024
node.session.iscsi.ImmediateData = Yes
node.session.iscsi.FirstBurstLength = 1048576
node.session.iscsi.MaxBurstLength = 16776192
node.conn[0].iscsi.MaxRecvDataSegmentLength = 1048576
discovery.sendtargets.iscsi.MaxRecvDataSegmentLength = 1048576
```


When you use Nutanix Volumes, you can configure single path (single iSCSI NIC) or multipath (two iSCSI NICs). For bare-metal configurations, we recommend using multipath to avoid a single point of failure in the client NIC. If you use Volumes with a VM, you can only configure with a single NIC.

Set Up a Single iSCSI NIC

If you have multipath configured on a VM with a single NIC (which can occur if you use an organization-standard VM image or migrate a VM from a system that requires multipath) you should either disable multipath or increase the `fast_io_fail_tmo` value. If you leave multipath enabled, increase `fast_io_fail_tmo` to 120 seconds to allow clean cluster failover on the client VM:

```
# cat /etc/multipath.conf
defaults {
 fast_io_fail_tmo 120
      }
<…snip…>
```

What follows is an example of how to set up a single NIC for iSCSI on a VM. After you create the VM with a NIC dedicated for iSCSI traffic with an IP address, follow this procedure. Assume the iSCSI NIC is ETH1.

- Create an iface:

```
[root@localhost ~]# iscsiadm -m iface -I iface1 --op=new
[root@localhost ~]# iscsiadm -m iface -I iface1 --op=update -n
 iface.net_ifacename -v eth1
```

- Create a VG.

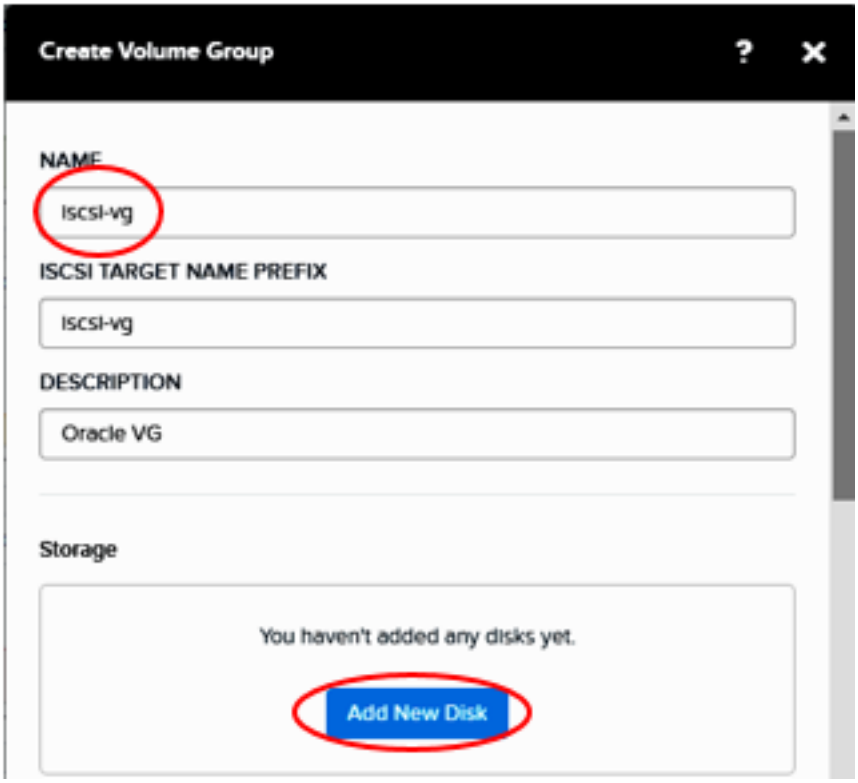

Figure 1: Create a VG

- Add disks to the VG.

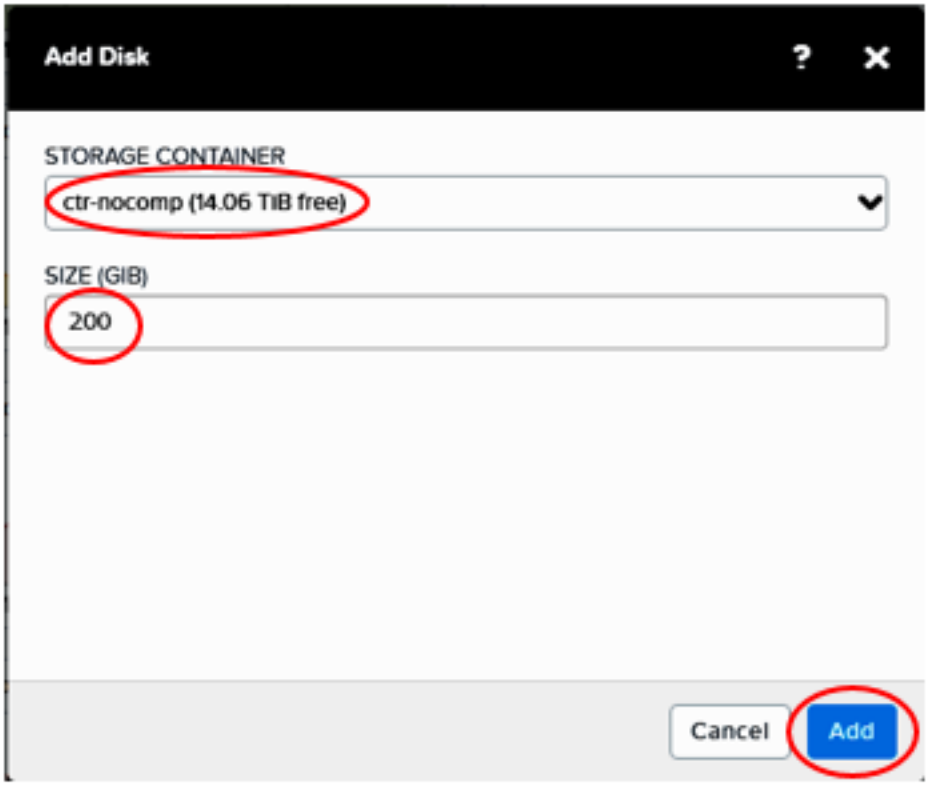

Figure 2: Choose a Storage Container and Size

- Save the VG.

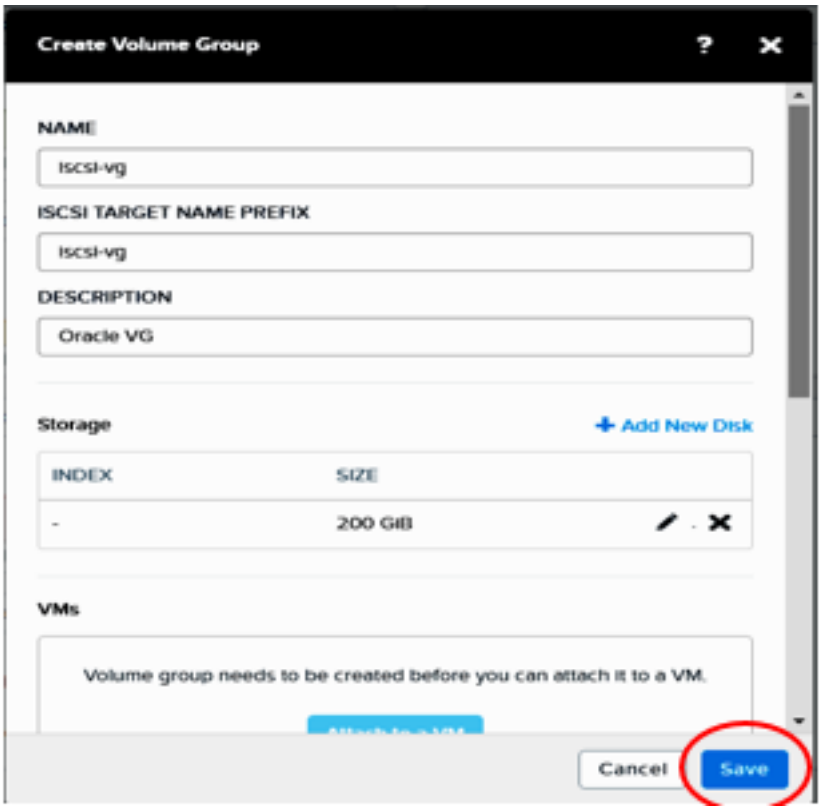

Figure 3: Save the VG

- Obtain the initiator IQN on the client.

```
[root@racnode3 ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1994-05.com.redhat:b46d2cffb62f
```

Figure 4: Initiator IQN

- Update the VG. This step allows you to specify the initiator IQN. Click Update.

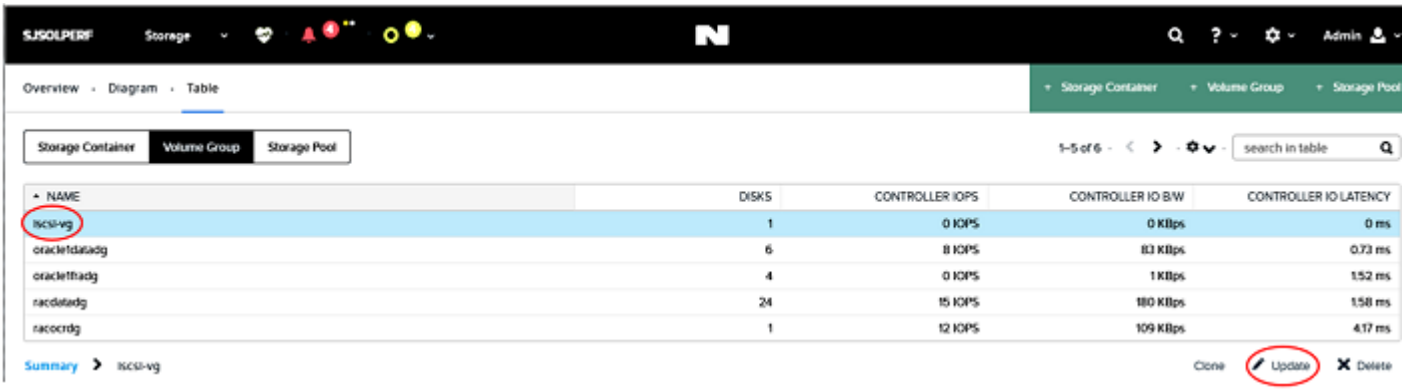

Figure 5: Update the VG

- Click Add New Client.

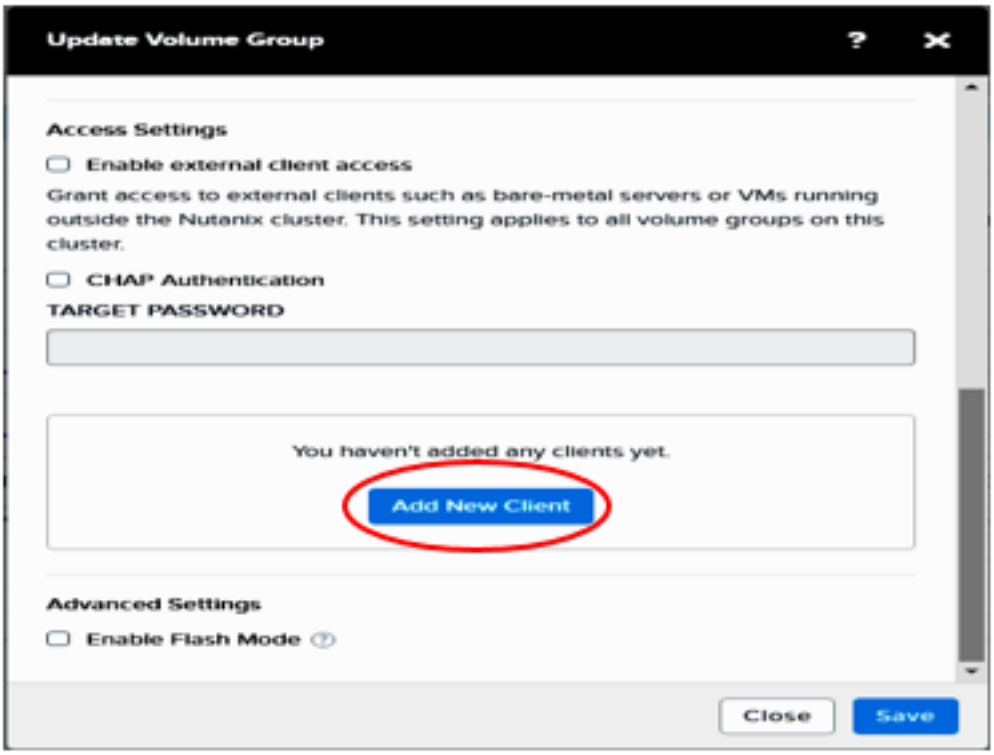

Figure 6: Add a New Client

- Enter the client IQN and click Add.

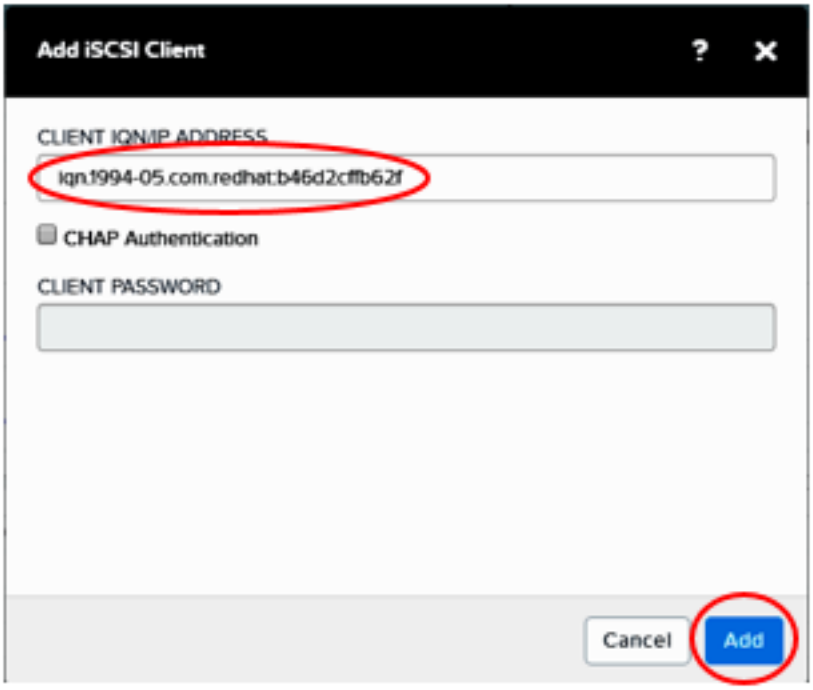

Figure 7: Client IQN

- Click Save.

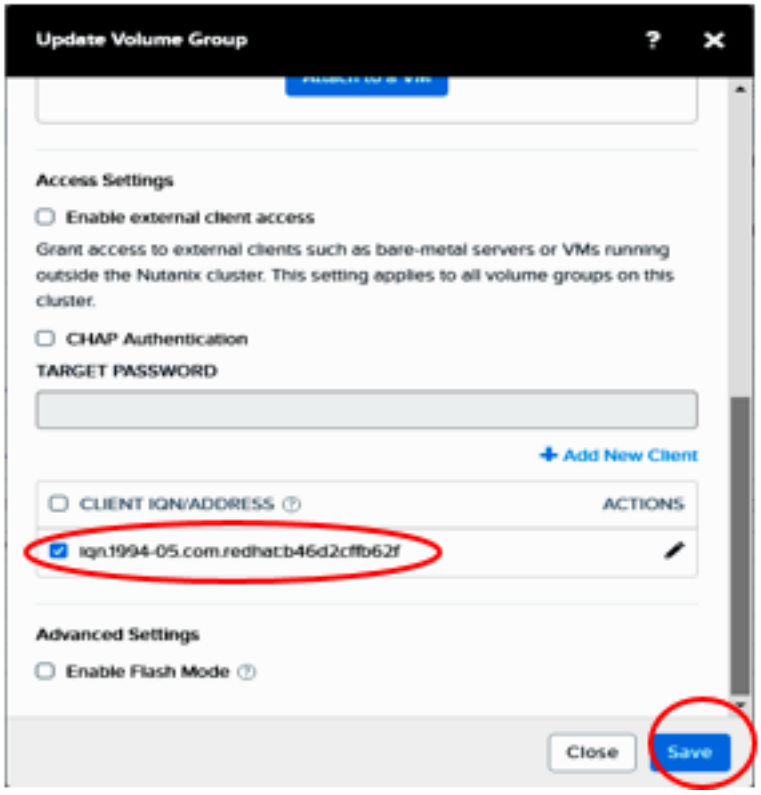

Figure 8: Save Changes to VG

- Run the iSCSI discover and login commands using the Nutanix external data service IP address:

```
[root@localhost ~]# iscsiadm -m discovery -t st -p <DSIP>
[root@localhost ~]# iscsiadm -m node - -login
```

Set Up Multiple iSCSI NICs

If you use multiple NICs for iSCSI traffic on your bare-metal servers, you need to configure dm-multipath. After you set up your iSCSI NICs using the above steps, follow this procedure to set up multipath on your bare-metal servers.

- Edit the `/etc/multipath.conf` file. If this file doesn't exist, create a new one. Enter the following text in this file:

```
devices {
        device {
                vendor                  "NUTANIX"
                product                 "Server"
                path_grouping_policy    multibus
```



```
                path_selector           "round-robin 0"
                features                "1 queue_if_no_path"
                path_checker            tur
                rr_min_io_rq            20
                rr_weight               priorities
                failback                immediate
        }
}

multipaths {
        multipath {
         wwid   1NUTANIX_NFS_1_0_4424_ed6ae886_4ecc_473d_aadc_765f706bca13

                alias   asm1
                }
}
```

- Enter the vDisk Universally Unique Identifier (UUID) and an alias name for it. Add all vDisks to this file. The preceding code block shows a sample of a single vDisk only.
- Start your `multipathd` service:

```
[root@localhost ~]# systemctl stop multipathd.service
[root@localhost ~]# systemctl start multipathd.service
[root@localhost ~]# multipath -r -p multibus
[root@localhost ~]# for a in $(multipath -ll | grep "NUTANIX" | awk '{print
 $1}') ; do dmsetup message $a 1 "queue_if_no_path" ; done
```

The command `for loop` ensures that each vDisk has a feature of 1 `queue_if_no_path`. You must verify that each vDisk has this feature by running the `multipath -ll` command.

The `for loop` command and the selection `multipath -r -p multibus` don't persist through a server reboot. To make sure that these elements are set on reboot, add the commands to a shell script and have it run automatically. You can put this shell script in the `/etc/rc.local file`. New Linux distributions may not have `/etc/rc.local` configured to run, so you might have to manually configure it.

7. Configuration for Volumes and vDisks

In a traditional three-tier architecture, application and database administrators often work with storage administrators to create a custom storage design to suit their environment, which includes designating the RAID level and block size for different types of files, such as database tablespaces or log files. This process can become cumbersome to manage, especially if there are multiple databases. Nutanix eliminates the problems associated with choosing the optimal RAID and block size. Once you create a VG with the required number of vDisks for your database or application, you're ready to deploy. The following table outlines the recommended minimum disk layout for a database such as Oracle. For other workloads refer to the specific application's best practice guide.

Table: Nutanix Volume Configuration

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 Filesystem with LVM
4	Database archive log files	Can be used with Oracle ASM or Filesystem with LVM
4	Database RMAN backup files	Can be used with Oracle ASM or Filesystem with LVM

Note: Nutanix recommends that you use multiple vDisks with OS-level striping for any applications that require high performance I/O.

There are two ways to create vDisks on Nutanix for your VMs:

1. Nutanix native vDisks.

2. Nutanix volume groups.

Nutanix Native vDisks

The Nutanix native vDisk option simplifies VM administration because it doesn't require Nutanix VGs. When you create a VM using Prism, add more vDisks to the VM for the application or database the same way you add a vDisk to the VM for the boot disk. If you have an application or database that doesn't have intensive I/O requirements, native vDisks are the best option.

Nutanix Volume Groups

Alternatively, you can attach Nutanix VGs, which are collections of vDisks, to the VMs. VGs enable you to separate the data vDisks from the VM's boot vDisk. This separation allows you to create a protection domain that consists only of the data vDisks for snapshots and cloning. In addition, VGs let you configure a cluster with shared-disk access across multiple VMs. Supported clustering applications include Oracle Real Application Cluster (RAC), IBM Spectrum Scale (formerly known as GPFS), Veritas InfoScale, and Red Hat Clustering. To attach the VG to multiple VMs when you use the aCLI, create it with a `shared=true` attribute. If you use Prism, answer `yes` when you attach the VG to a second VM. There are two ways to use Nutanix VGs: default VG and VGLB.

Default Volume Group

This type of VG provides the best data locality because it doesn't load-balance vDisks in a Nutanix cluster, which means all vDisks in default VGs have a single CVM providing their I/O. For example, in a four-node Nutanix cluster that includes a VG with eight vDisks attached to a VM, a single CVM owns all the vDisks and all I/O to the eight vDisks goes through this CVM.

Volume Group with Load Balancing (VGLB)

> Note: This feature is available with AHV (AOS 5.6 and later). If you use ESXi or Hyper-V, you can use Nutanix Volumes with iSCSI to achieve this functionality.

VGLBs distribute ownership of the vDisks across all the CVMs in the cluster, which can provide better performance. Use the VGLB feature if your

applications or databases require better performance than the default VG provides. Run the following command in the Nutanix aCLI to create a VGLB:

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


8. Logical Volume Manager Configuration

This section discusses Linux LVM best practices. When you create a file system for any applications or databases, use LVM striping across all vDisks for a specific mount point. The following example shows how to create an LVM VG 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 VG.

```
[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, which specifies the number of vDisks to stripe across, and the `-I` (uppercase i) option, which specifies the stripe size. The recommended stripe size is 512 KB.

```
[root@localhost ~]# lvcreate -l 383994 -i 8 -I 512 -n vol1 vgdata
 Logical volume "vol1" created.
```

- Create the file system. In this example, we're 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, refer to the Mount Options for ext4 and XFS File Systems table.

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

To give your file systems a friendly name, try `xfs_admin` for xfs or `e2label` for ext4. Use the `LABEL=` option in the `/etc/fstab` file for easy management.

File System Mount Options

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

Table: 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

If you're using a Linux distribution with Linux kernel 4.14 or higher, note that the `nobarrier` option on XFS has been deprecated, so you shouldn't use it in your environment. Refer to your Linux distribution vendor for information on XFS mount options.

For other file systems, refer to your Linux distribution vendor's documentation.

9. Linux Disk Device Settings

To optimize performance for your Linux VMs, change the following disk parameters from their default settings:

- Change the `max_sectors_kb` to `1024` (the default is 512). For newer Linux distributions, the default may already be set to 1024. Change the disk timeout to `60` (default is 30).

```
[root@localhost ~]# 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@localhost ~]# 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@localhost ~]# 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. Follow these 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 9: Add elevator=noop

- 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.

```
GRUB_CMDLINE_LINUX="crashkernel=auto rhgb quiet scsi_mod.use_blk_mq=1 transparent_hugepage=never"
```

Figure 10: Add scsi_mod.use_blk_mq=1

- Run `grub2-mkconfig` to generate a new grub file, then reboot.

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


10. Conclusion

This best practice guide outlines our recommended settings for a Linux VM running on Nutanix with AHV. With the proper settings for Nutanix cluster networking, as well as for volume, LVM, kernel configuration, and proper file system mount options, you can maximize Linux performance for any application. Nutanix removes the complexity of constantly managing and optimizing the underlying compute, network, and storage architecture, so you can focus on higher-value tasks for your business.

For feedback or questions, contact us using the Nutanix NEXT Community forums.

11. Appendix

References

1. Nutanix Volumes best practice guide
2. AHV best practice guide
3. Documentation for /proc/sys/vm/*
4. Prism Web Console Guide: Nutanix Volumes
5. Nutanix Volumes Guide

About Nutanix

Nutanix is a global leader in cloud software and a pioneer in hyperconverged infrastructure solutions, making clouds invisible and freeing customers to focus on their business outcomes. Organizations around the world use Nutanix software to leverage a single platform to manage any app at any location for their hybrid multicloud environments. Learn more at www.nutanix.com or follow us on Twitter @nutanix.

List of Figures