

TECH NOTE

Migrating SQL Server to Nutanix

Copyright

Copyright 2022 Nutanix, Inc.

Nutanix, Inc.
1740 Technology Drive, Suite 150
San Jose, CA 95110

All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. Nutanix and the Nutanix logo are registered trademarks of Nutanix, Inc. in the United States and/or other jurisdictions. All other brand and product names mentioned herein are for identification purposes only and may be trademarks of their respective holders.

Contents

1. Executive Summary.....	4
2. Introduction.....	5
Audience.....	5
Purpose.....	5
Document Version History.....	6
3. Prerequisites.....	7
4. Migrate.....	9
5. Postmigration.....	12
6. SQL Server Always On Availability Groups.....	13
Set Up a SQL Server AG VM.....	13
Migrate a Non-AG Environment to a Nutanix AG Environment.....	14
Migrate an Existing AG Environment to a Nutanix AG Environment.....	17
7. Conclusion.....	20
8. Appendix.....	21
Designs and Configurations for SQL Server Environments.....	21
References.....	22
About Nutanix.....	24
List of Figures.....	25

1. Executive Summary

Many SQL Server customers with environments hosted on bare metal want to migrate to a cloud platform such as Nutanix Cloud Platform (NCP), but there are many issues they need to address, both before and after migration. Nutanix handles these migration challenges and migrates the SQL Server environment, giving the customer access to Nutanix Database Service (NDB; previously Nutanix Era) for their day-2 activities.

Nutanix offers significant capex savings in storage costs with zero-byte cloning for all UAT (user acceptance testing), test, and dev environments, and opex savings start as soon as the customer successfully migrates and can access all the automation that NDB brings to the datacenter.

2. Introduction

Audience

This tech note is part of the Nutanix Solutions Library. We wrote it for Nutanix Engineers in the field who are responsible for analyzing customer requirements, documenting customer pain points, and deploying a robust SQL Server environment on Nutanix. This document also explains the challenges involved with migrations and helps engineers build a statement of work (SoW) prior to migration. Readers of this document should already be familiar with SQL Server, migration and upgrade, backup and recovery, high availability and disaster recovery, the [Database Migration tech note](#), and the [Microsoft SQL Server on Nutanix best practice guide](#).

Purpose

In this document, we cover the following topics:

- Migration prerequisites
- Migration process
- Postmigration challenges
- Testing a migrated environment
- Migrating a single instance to a Nutanix SQL Server Always On availability groups (AG) environment
- Migrating an existing SQL Server AG environment to Nutanix

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

Document Version History

Version Number	Published	Notes
1.0	March 2022	Original publication.

3. Prerequisites

If you want to have a smooth migration, you must document prerequisites as part of the migration workflow. If the database has gone through many upgrades over the years, now is the time to document those changes: properly identifying and documenting the database and application characteristics that have evolved ensures that future migrations and upgrades are that much easier. The following list specifies a few things you should document:

- If you're moving the database (and associated application) from a non-Nutanix platform to a Nutanix platform, ensure that you have a proper cutoff backup of the database and the application associated with that database (sometimes called a backup set).
 - › We recommend using multiple backup (*.bak) files for your database backup because they make it easier to transmit and monitor the backup files. They can also be transferred in parallel where bandwidth is available. For example, we split a 500 GB database backup into 10 *.bak files of 50 GB each to make it easy for us to push or pull the files from multiple locations (such as cloud, hybrid, or shared backup locations). We also recommend using multiple vDisks in a RAID 0 as your backup disk.
- Analyze the database size and check how the database has grown over the years to help with capacity planning. Use the data growth trend information to create a new disk layout with proper configurations for the newly migrated SQL Server environment.
- List the instance configurations you need to apply after you migrate to Nutanix.
- List all the external scripts (including maintenance scripts, if any) and identify any hardcoded information (like passwords, directories, and Open Database Connectivity connections). You may also want to change this information now.
- Document any packages or services pointing to other databases.

- List all the agents and jobs associated with database functioning. There should be a matching list in the new Nutanix environment.
- Databases are usually connected to an application, so document the application connection information. If the database is connected to multiple applications or to other databases, document them, as you may need to reconnect them after you migrate.
- Document any specific database properties, like auto stats, compatibility levels, and database owners.
- Read the [Database Migration tech note](#).
- Read the [Microsoft SQL Server on Nutanix best practice guide](#), pull out all the information on how to set up the VM and the database, and include that information in your premigration checklist.

You need to get buy-in for the migration process from the business owner of the database (typically the applications team). Identify the key players, get the top 10 programs you need to test before migration, and measure and document their performance characteristics. Run these 10 programs after the migration to get a Go-No Go decision from the business team. We also recommend doing a test run of the whole migration process so that you can correct any migration flow issues, identify the timings, and discuss the downtime with the applications team.

4. Migrate

After fulfilling the prerequisites, you're ready to start your migration:

- Identify the CPU specification, how many cores (to make sure the license matches), and what speed.
- Identify how much memory you have to build a VM that's within the vNUMA boundary.
- Check that there's enough disk space available to host the migrated database and allow for future growth based on the capacity planning information you documented in the prerequisites.

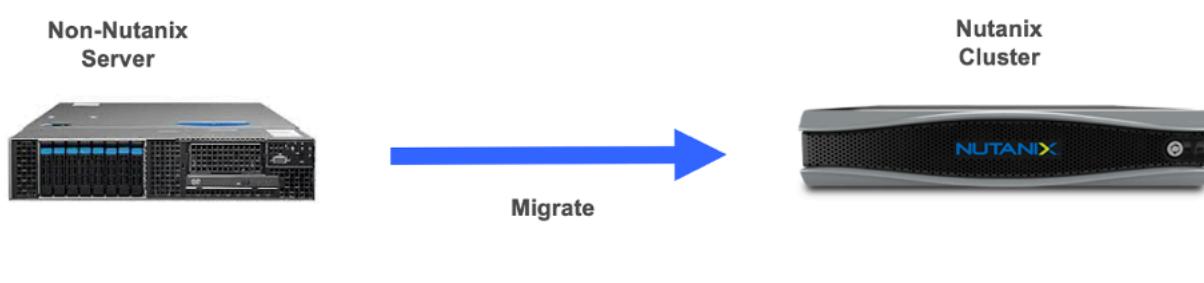


Figure 1: Bare Metal to Nutanix Migration

When you start building the VM that is going to host the migrated database, ensure that you follow these best practices:

- CPU: For the VM, allocate only the number of cores you have licensed for SQL Server.
- Memory: Configure the SQL Server memory to be smaller than the total VM memory to accommodate all the other system processes that also need memory. The SQL server memory requirement shouldn't be larger than the vNUMA boundary.

- I/O: Use separate vDisks for SQL Database files (minimum of two vDisks), TempDB, and UserLog. Add these vDisks when you build the VM.
- Network: Identify the vLAN allocated for the user VMs. Check with the network team for any LACP options.
- Backup: Use a single vDisk large enough to host the old database backup. If you want to retain the old backup, you need to plan for additional space to accommodate the first backup copy of the new database.
- Software: Attach the correct Windows OS to the first CD-ROM. Add a second CD-ROM and identify the correct VirtIO driver version to mount for it.

We've provided an overview of the next steps. For more details, see the [Database Migration tech note](#) and the [Microsoft SQL Server on Nutanix best practice guide](#).

- Start the VM.
- Install the OS.
- Install the drivers.
- Configure the host name and domain name.
- Format the vDisks and allocate appropriate names and drive letters for the Windows disks.
- Mount the SQL Server software.
- Install the SQL Server software.
- Install all the support tools used in the old environment.
- Configure memory.
- Set up other instance-level parameters.
- Set up the backup parameter pointing to the backup directory.
- Use a tool like WinSCP or mount the backup vDisk to copy the database backup to the new backup directory.

- Restore the database. We recommend using multiple *.ndf data files across two vDisks.

5. Postmigration

Check the newly migrated database in the same order you used when you collected information and prepared the checklist. The following list specifies some items you should look for:

- Check database properties like user access, compatibility, trustworthiness, and any automatic parameters that should match the old database.
- Ensure that the version and recovery mode are correct and match the old database.
- Check parameters like update statistics and text- and cursor-related settings, as they affect query performance. If any of these parameters have been tuned, carry them forward to the new database (remember that you're migrating the database, not upgrading it).
- Set up the linked server, dependent jobs, and everything else listed in the premigration checklist.
- Rebuild the indexes if required or just collect statistics for the indexes.
- Recompile all the packages and procedures, as they might have become stale.

Once you've set up the instance and database parameters, it's time to test the programs you agreed on with the business applications team. Once the business applications team compares the response times of the queries (programs or reports) and agrees that the new database is performing at a similar level or better than the old one, get a Go-No Go decision from them.

Finally, back up the database. This backup is your first (gold) backup of the migrated database. Remember to take a VM snapshot and set up protection domains.

6. SQL Server Always On Availability Groups

Once the team makes the Go-No Go decision and you prepare the first database backup and VM snapshot, it's time to decide which options you want to use for high availability (HA) and disaster recovery.

You can use the same HA setup or, because Nutanix is a virtualized environment, you can take advantage of its storage-level replication, protection domains, and vMotion options.

For disaster recovery, we recommend SQL Server AG because it protects customer data from database failures and data corruption and provides a quick turnover for business continuity.

Note: Far-site and near-site disaster recovery decisions are made based on the strength of the available bandwidth and the service-level agreements (SLAs) from and for the business applications team.

Set Up a SQL Server AG VM

You design, plan, and build SQL Server AG VMs the same way you do SQL Server standalone instance VMs, with a few additional considerations:

- Number of replicas
- Replica type
- Replica placement (on the same cluster, near site, or far site)
- Network bandwidth and latency between the clusters

Base your decisions for the replica type (synchronous, asynchronous, or a mix of both), the number of replicas, and the replica placement on the HA and disaster recovery requirements of the specific environment.

Migrate a Non-AG Environment to a Nutanix AG Environment

If the existing environment is a standalone instance or if the network connectivity between the environments is a challenge, use the following procedure to migrate to a SQL Server AG environment on Nutanix. This process involves slight downtime when you cut over to the new environment after taking the last log backup of the old environment. The full and database log backups should be made available (transported) for setting up the new environment.

To build and configure the AG database server VMs when migrating from a single-instance VM:

- Start the VM.
- Install the OS.
- Install the drivers.
- Configure the host name and domain name.
- Format the vDisks and allocate appropriate names and drive letters for the Windows disks.
- Mount the SQL Server software.
- Install the SQL Server software.
- Install all the support tools used in the old environment.
- Configure memory.
- Add the VM to the domain.
- Set up security credentials and other instance-level parameters.
- Set up the backup parameter pointing to the backup directory.
- Use a tool like WinSCP or mount the backup vDisk to copy the database backup to the new backup directory.

- Restore the full database backup on the replicas:

- › On the primary replica:

```
RESTORE DATABASE <DB name> FROM DISK = <Backup File path>
```

- › On the secondary replicas:

```
RESTORE DATABASE <DB name> FROM DISK = <Backup File path> WITH NORECOVERY
```

- Ensure that the Active Directory service account you're going to use for SQL Server is created and active.
- Change the SQL Server service account to the desired Active Directory account and restart the SQL Server service.
- Enable the Windows Server Failover Clustering (WSFC) feature and create a failover cluster. Include all the replica nodes as member nodes.
- Enable the SQL Server Always On feature on the SQL Server availability groups installed on the database server VMs.
- Grant NT AUTHORITY\SYSTEM the required privileges:

```
GRANT ALTER ANY AVAILABILITY GROUP TO [NT AUTHORITY\SYSTEM];
GRANT CONNECT SQL TO [NT AUTHORITY\SYSTEM];
GRANT VIEW SERVER STATE TO [NT AUTHORITY\SYSTEM]
```

- Create database mirroring endpoints on the desired port on the database server VMs and grant CONNECT ON ENDPOINT privilege to the domain user:

```
CREATE ENDPOINT [HADR_endpoint] STATE=STARTED AS TCP (LISTENER_PORT = <Port>
FOR DATA_MIRRORING (ROLE = ALL, ENCRYPTION = REQUIRED ALGORITHM AES)
GRANT CONNECT ON ENDPOINT::[HADR_endpoint] TO [<Domain user>]
```

- Restore database log backups on the replicas.

- › On the primary replica:

```
RESTORE LOG <DB name> FROM DISK = <Log Backup File path>
```

- › On the secondary replicas:

```
RESTORE LOG <DB name> FROM DISK = <Log Backup File path> WITH NORECOVERY
```

- › If any additional log backups occur before the database joins the secondary replica, restore all those log backups, in sequence, to the server instance that hosts the secondary replica using RESTORE WITH NORECOVERY.

- Create an availability group using the SQL Server Always On wizard from SQL Server Management Studio or Transact-SQL (T-SQL). Because you fully prepared the secondary database, choose the Join only option when you join the secondary replica database to the availability group. All the desired databases need to be added to the availability group.
- An example configuration for an asynchronous replica follows:
 - › On the primary replica:

```
CREATE AVAILABILITY GROUP [<Availability group name>] WITH
    (AUTOMATED_BACKUP_PREFERENCE = SECONDARY, DB_FAILOVER = OFF,
     DTC_SUPPORT = NONE, REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT = 0)
    FOR DATABASE [<Database name>] REPLICA ON <Replica1> WITH (ENDPOINT_URL
    = <Database mirroring endpoint URL for Replica1>, FAILOVER_MODE =
    Manual, AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT, BACKUP_PRIORITY
    = 50, SEEDING_MODE = MANUAL, SECONDARY_ROLE(ALLOW_CONNECTIONS =
    NO)), <Replica2> WITH (ENDPOINT_URL = <Database mirroring endpoint
    URL for Replica2>, FAILOVER_MODE = Manual, AVAILABILITY_MODE =
    ASYNCHRONOUS_COMMIT, BACKUP_PRIORITY = 50, SEEDING_MODE = MANUAL,
    SECONDARY_ROLE(ALLOW_CONNECTIONS = NO));
```

- › On all the secondary replicas:

```
ALTER AVAILABILITY GROUP [<Availability group name>] JOIN
ALTER DATABASE [<Database name>] SET HADR AVAILABILITY GROUP =
    [<Availability group name>]
```

- Ensure that you joined the replicas to the availability group.
- Ensure that the availability group is in a healthy state without any errors.
- Ensure that the replica database has the Synchronized (for synchronous replicas) or Synchronizing (for asynchronous replicas) status.
- Ensure that the application-related settings are correctly updated in the new AG setup.
- Cut over from the old environment.
- Take the first backup of the migrated database.
- Take the first snapshot of the VMs.

Migrate an Existing AG Environment to a Nutanix AG Environment

In this procedure, you configure and add one secondary replica on Nutanix infrastructure, then perform a failover from the earlier infrastructure to the newly constructed Nutanix environment. We don't expect any downtime during this migration procedure.

During this procedure, set up all the database VMs that need to be part of the availability group as secondary replicas of the existing availability group. You need a stable and reliable network connection between the old and new environments to successfully complete the migration. Perform the following steps to migrate an existing SQL Server AG environment to Nutanix:

- Start the VM.
- Install the OS.
- Install the drivers.
- Configure the host name and domain name.
- Format the vDisks and allocate appropriate names and drive letters for the Windows disks.
- Mount the SQL Server software.
- Install the SQL Server software.
- Install all the support tools used in the old environment.
- Configure memory.
- Add the VM to the domain.
- Set up security credentials and other instance-level parameters.
- Set up the backup parameter pointing to the backup directory.
- Use a tool like WinSCP or mount the backup vDisk to copy the database backup to the new backup directory.
- Ensure that the Active Directory service account you're going to use for SQL Server is created and active.

- Change the SQL Server service account to the desired Active Directory account and restart the SQL Server service.
- Enable the WSFC feature on any new VMs or nodes you created.
- Add the new VMs or nodes to the existing Windows failover cluster.
- Enable the SQL Server Always On feature on the SQL Server instance installed on the new replica VMs or nodes.
- Grant `NT AUTHORITY\SYSTEM` the required privileges on the new replica VMs or nodes:

```
GRANT ALTER ANY AVAILABILITY GROUP TO [NT AUTHORITY\SYSTEM];  
GRANT CONNECT SQL TO [NT AUTHORITY\SYSTEM];  
GRANT VIEW SERVER STATE TO [NT AUTHORITY\SYSTEM]
```

- Create database mirroring endpoints on the desired port on the new replica VMs or nodes and grant privilege to the domain user:

```
CREATE ENDPOINT [HADR_endpoint] STATE=STARTED AS TCP (LISTENER_PORT = <Port>)  
FOR DATA_MIRRORING (ROLE = ALL, ENCRYPTION = REQUIRED ALGORITHM AES)  
GRANT CONNECT ON ENDPOINT::[HADR_endpoint] TO [<domain user>]
```

- Restore the full database backup on the new replica VMs or nodes. Make sure that the file layout and location are the same as in the existing environment. Use the `WITH NORECOVERY` option:

```
RESTORE DATABASE <Database name> FROM DISK = <Backup file path> WITH NORECOVERY
```

- Restore database log backups on the replicas using the `WITH NORECOVERY` option:

```
RESTORE LOG <DB name> FROM DISK = <Log backup file path> WITH NORECOVERY
```

- › If any additional log backups occur before the database joins the secondary replica, restore all those log backups, in sequence, to the server instance that hosts the secondary replica using `RESTORE WITH NORECOVERY`.

- Add the new replica to the existing availability group using the SQL Server Always On wizard from SQL Server Management Studio or T-SQL from the primary replica. Because you prepared the secondary database, choose the Join only option when you join the secondary replica database to the availability group. Add all the AG member databases to the availability group.

- Update the primary replica and the newly added replica node to use synchronous commit availability mode. This mode ensures that you don't lose data when you fail over between the replicas.
- Ensure that you joined the replicas to the availability group.
- Ensure that the availability group is in a healthy state without any errors.
- Ensure that the replica database has the Synchronized (for synchronous replicas) or Synchronizing (for asynchronous replicas) status.
- Fail over to the new AG replica.
- Remove the old replicas from the new availability group.
- Evict the old cluster nodes (old replicas) from the Windows cluster.
- Make the desired configuration changes on the AG replica and ensure that all the components of the availability group are healthy or green.
- Take the first backup of the migrated database.
- Take the first snapshot of the VMs.

7. Conclusion

This document is a resource for engineers in the field who want to understand the challenges involved in migrating a non-Nutanix SQL Server environment to Nutanix. This document also helps engineers involved in SQL Server migrations build a robust SoW and understand the partners they need to onboard from the customer side for a successful migration.

For feedback or questions, contact us using the [Nutanix NEXT Community forums](#).

8. Appendix

Designs and Configurations for SQL Server Environments

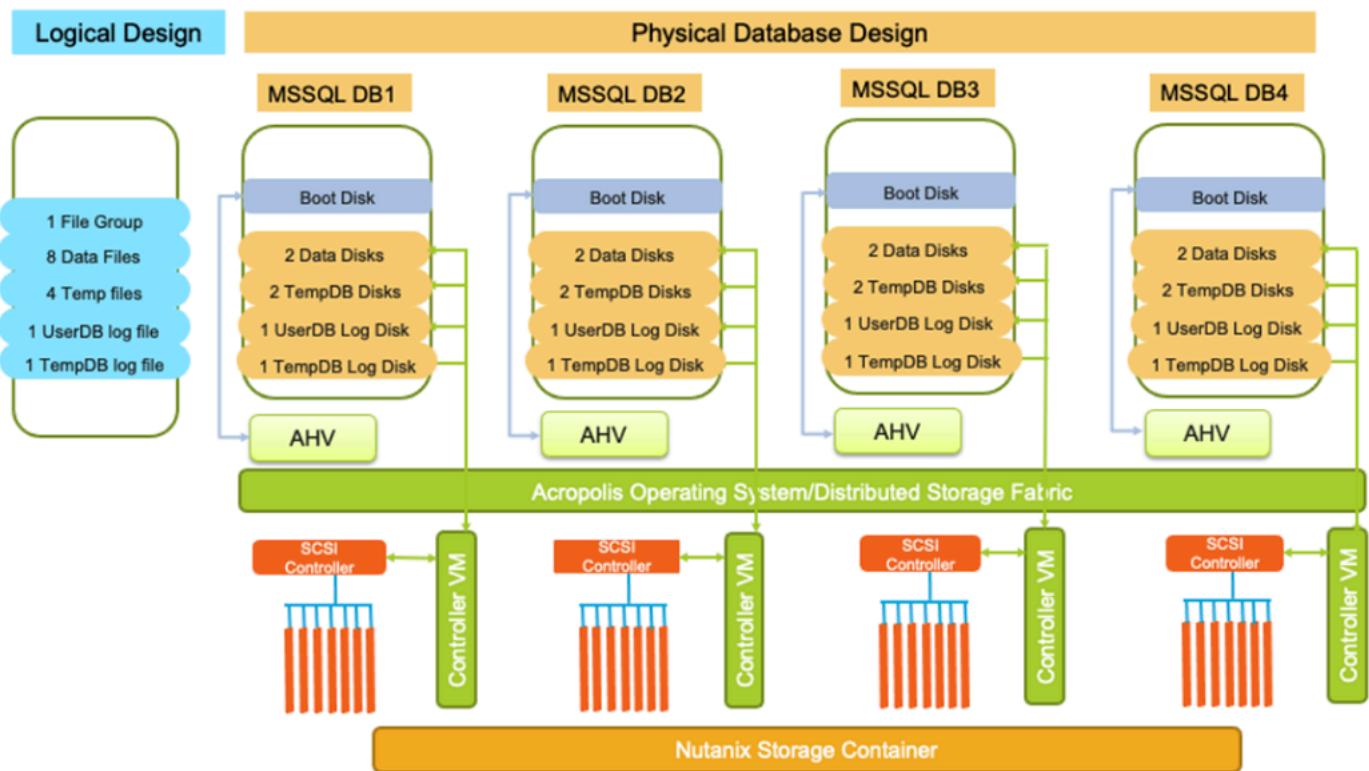


Figure 2: Logical and Physical Design for SQL Server Environments

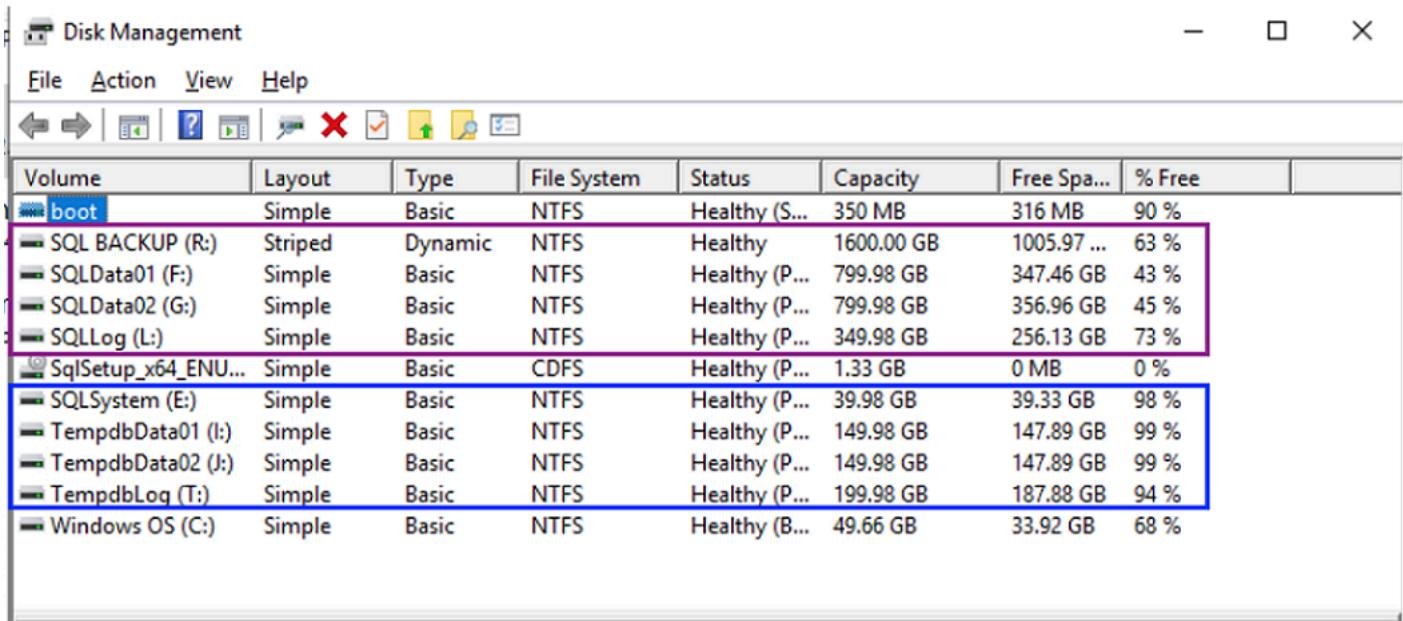


Figure 3: SQL Server Disk Layout Example

We used the [Microsoft SQL Server on Nutanix best practice guide](#) for our disk layout:

- 2 vDisks for SQL data
- 1 vDisk exclusively for UserLogs
- 2 vDisks for TempData
- 1 vDisk for the SQL system
- 1 vDisk exclusively for TempdbLog
- 1 vDisk for the OS
- 4 vDisks Striped for SQL backup

If you use this disk layout for the primary replica, all the replicas have a similar disk layout.

References

1. [Microsoft SQL Server on Nutanix best practice guide](#)

2. Database Workloads on Nutanix reference architecture
3. Prepare a secondary database for an Always On availability group
4. Back Up and Restore of SQL Server Databases
5. Always On availability groups: a high-availability and disaster-recovery solution
6. What is SQL Server Management Studio (SSMS)?
7. Monitor and Tune for Performance

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](https://twitter.com/nutanix).

List of Figures

Figure 1: Bare Metal to Nutanix Migration.....	9
Figure 2: Logical and Physical Design for SQL Server Environments.....	21
Figure 3: SQL Server Disk Layout Example.....	22