No one disputes that “high availability” is important for mission-critical SQL Server databases. But there is ongoing debate over what constitutes high availability. Indeed, ask 10 people “What is high availability?” and you might get 15 different answers.

Here is how TechTarget defines high availability: “In information technology, high availability refers to a system or component that is continuously operational for a desirably long length of time. Availability can be measured relative to ‘100% operational’ or ‘never failing.’ A widely-held but difficult-to-achieve standard of availability for a system or product is known as ‘five 9s’ (99.999%) availability.”

For the vast majority of applications, four-nines (99.99%) is generally accepted by both database and system administrators as constituting mission-critical high availability. For this white paper, high availability (HA) will be defined as ensuring the SQL Server databases are operating when and as needed. “When” takes into account the percentage of time the application is up and running, while “as” takes into account proper operation, with no data loss.

It is important to note there are substantially different availability requirements for different applications depending on how critical they are, how often they run, how quickly the failover must occur, how much data loss might be tolerable, etc. These and other considerations are all addressed in this white paper.

The Difference Between High Availability and Disaster Recovery

It is also important to note there are differences between high availability and disaster recovery (DR). For example, applications protected by HA provisions may also need separate DR provisions, and even applications that require no HA protection will likely need to be included in the DR strategy. This white paper focuses exclusively on HA and only mentions DR when such protection is somehow associated with the HA provisions.

The content, intended for database and system administrators, is organized into three sections followed by a brief conclusion. The first section forms a foundation by providing a detailed answer to “What is high availability?” The second section outlines the many HA options available with and for SQL Server. The third section introduces SANless failover clustering solutions from SIOS Technology for both Windows Server and Linux.

What is High Availability?

High availability has many moving parts, and that partly (no pun intended) explains why it means different things under different circumstances to different people. Outlined here are the elements required in an HA solution capable of keeping SQL Server databases operating when and as needed 24x7.

Redundancy is essential for ensuring high availability, and serves as the very foundation of all HA configurations. The goal is to eliminate as many single points of failure as possible at all levels, including in the servers, storage, network and software (applications and OS), as well as in the datacenter’s power, cooling and security infrastructure, etc.
Consider for example a load-balancer front-ending multiple Web servers, which share the load with active-active redundancy. While a failure in any of the servers may affect performance, the application(s) will keep running. But if the load-balancer itself is not configured with full redundancy, it creates a single point of failure. Similarly, storage area networks (SANs) and other forms of shared storage must also have fully redundant configurations to prevent creating any single points of failure.

The redundant resources must be configured in a way that makes it possible to detect and recover from all possible failures automatically with minimal downtime and no data loss. In other words, having a “spare” is not the same as having a “hot” standby ready to take over immediately should there be a failure.

For SQL Server, ensuring that the standby is fully up-to-date with the current data is what makes it hot, and this could be achieved using a shared drive (a single point of failure), a SAN or continuous data replication from the active to the standby, with each instance having its own locally-attached storage. For SQL Server instances covered by active Software Assurance, there is no additional license required in active-standby configurations because only one instance is running the software at any time.

For HA for SQL Server, the data replication should be synchronous; that is, both the active and standby instances are being updated simultaneously, and this requires a high-performance, low-latency network. For DR purposes, where the active and standby instances should be separated geographically to protect against widespread disasters, asynchronous replication is used to avoid adversely impacting on write performance.

When measuring the uptime needed for HA, it is important to take into account not only unplanned downtime (i.e. hard or soft failures), but also all of the planned downtime regularly needed for routine hardware and software maintenance. It is also important to measure uptime at the application level, which is something cloud service provider service level agreements (SLAs) normally do not do, creating an issue that is addressed in the next section.

### Requirements for High Availability

At a minimum, HA provisions must meet these four requirements:

1. Redundancy for the hardware, software and data to eliminate most if not all single points of failure
2. Automatic application-level detection of all failures regardless of the cause(s), including in hardware and/or software, from a network or power outage, human error, natural disaster, etc.
3. No data loss during all failures regardless of the cause(s)
4. Automatic rapid failover to minimize downtime experienced by the users
5. Manual failover and failback to prevent downtime during planned maintenance

<table>
<thead>
<tr>
<th>Availability %</th>
<th>Downtime per year</th>
<th>Downtime per month</th>
<th>Downtime per week</th>
<th>Downtime per day</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.9%</td>
<td>8.77 hours</td>
<td>43.83 minutes</td>
<td>10.08 minutes</td>
<td>1.44 minutes</td>
</tr>
<tr>
<td>99.95%</td>
<td>4.38 hours</td>
<td>21.92 minutes</td>
<td>5.04 minutes</td>
<td>43.20 seconds</td>
</tr>
<tr>
<td>99.99%</td>
<td>52.60 minutes</td>
<td>4.38 minutes</td>
<td>1.01 minutes</td>
<td>8.64 seconds</td>
</tr>
<tr>
<td>99.995%</td>
<td>26.30 minutes</td>
<td>2.19 minutes</td>
<td>30.24 seconds</td>
<td>4.32 seconds</td>
</tr>
<tr>
<td>99.999%</td>
<td>5.26 minutes</td>
<td>26.30 seconds</td>
<td>6.05 seconds</td>
<td>864 milliseconds</td>
</tr>
</tbody>
</table>

The “Number of Nines” provides a way to quantify high availability, which most IT professionals concur should be four-nines (99.99%) for mission-critical applications.
Recovery Point and Recovery Time Objectives

The two metrics normally used to assess both HA and DR provisions are the Recovery Time Objective and the Recovery Point Objective. RTO is the maximum tolerable duration of an outage. Online transaction processing applications generally have the lowest RTOs, and those that are mission-critical often have an RTO of only a few seconds.

RPO is the maximum amount of data loss that can be tolerated, and there are usually different RPOs for HA and DR. For HA needs, RPO is often zero to specify there should be zero data loss under all failure scenarios. For DR, some loss of data may be tolerable for some applications under some circumstances. In DR scenarios for failures affecting an entire datacenter or region, the desired RPO is typically determined by the value of each application’s data, with the most critical ones requiring RPOs specifying minimal to no data loss.

The ability to satisfy different recovery time and recovery point objectives raises an important difference between HA and DR involving data replication and the need for a potential tradeoff between RTO and RPO. In a high-throughput, low-latency LAN environment, where data replication can be synchronous, the active and standby databases can be updated concurrently. This enables full recoveries to occur automatically and in real-time, making it possible to satisfy the most demanding RTOs and RPOs (for example, a few seconds and zero, respectively) with no tradeoff necessary.

Across the WAN, by contrast, forcing the active instance to wait for the standby to confirm completion of updates for every transaction would adversely impact on performance. For this reason, data replication in the WAN is usually asynchronous. This can create a tradeoff between accommodating RTO and RPO that normally results in an increase in recovery times. Here’s why: Asynchronous replication means that the DR server usually does not have all the most recent data. To avoid potential data loss, automatic failover is normally disabled on the DR server, which increases the recovery time. Failover to the DR server is only initiated once it has been determined that the primary site is offline and not likely to come online anytime soon, and that every reasonable attempt has been made to avoid data loss.

To achieve an optimal balance between RTO and RPO, most DR strategies acknowledge the potential for some data loss under catastrophic failure scenarios. But these strategies also normally contain provisions for taking various steps to minimize the data loss, normally through making recovery a manual process.

High Availability Options with and for SQL Server

This section provides a summary of all the HA options available both with and for SQL Server. As will be shown, some of these options are not actually suitable for HA purposes; they are included here only to show how they fit into a broader backup, archiving and DR strategies. Failover clustering solutions from SIOS Technologies are covered in the next section.

Options Available with SQL Server

Always On Failover Cluster Instances (FCIs) has been a standard feature since SQL Server 7 running on what was then called Microsoft Cluster Server available on Windows NT 4.0. FCIs
afford two major advantages: inclusion in the Standard Editions of SQL Server; and protection for the entire SQL Server instance, including system databases. A notable disadvantage is the need for cluster-aware shared storage, such as a storage area network (SAN), which is not available in the public cloud. On-premises, by contrast, where shared storage can and often does exist, FCIs leverage Windows Server Failover Clustering (also a standard feature).

Always On Availability Groups replaced database mirroring in SQL Server 2012 Enterprise Edition, and this feature is also included in SQL Server 2017 for Linux. This is SQL Server’s more robust HA/DR offering, capable of delivering rapid, automatic failovers with no data loss for HA, and/or protecting against widespread disasters by leveraging asynchronous replication with minimal data loss. But it requires licensing the more expensive Enterprise Edition, making it cost-prohibitive for many applications, and it lacks protection for the entire SQL instance. For Linux, which lacks integral failover clustering, there is a need for additional commercial and/or open-source software to provide high availability.

A notable disadvantage with application-specific options like Always On Availability Groups is the need for administrators to use other HA and/or DR solutions for all non-SQL Server applications. Having multiple HA/DR solutions inevitably increases complexity and costs (for licensing, training, implementation and ongoing operations), which is why many organizations prefer using separate general-purpose or application-agnostic solutions.

Separate HA/DR Solutions

The options discussed here are intended to work with a wide range of applications. The single exception is database mirroring, which is a special form of data replication used only with database applications.

Backup and recovery procedures are commonly used with all applications for archiving purposes, and may be suitable for the disaster recovery needs of some applications. But full and incremental backups do not provide the continuous, real-time data replication needed for HA.

Data replication is necessary for highly available configurations, but is not sufficient. On its own, data replication is effectively a continuous data backup solution that could be part of a DR strategy. But without any ability to detect failures at the application level and automatically failover to a standby, it is only part (albeit an essential one) of a complete HA solution.

Database mirroring is a form of data replication, and this feature was replaced by the much more capable Always On Availability Groups in SQL Server 2012. Mirroring is currently in maintenance mode, making it likely to be removed in a future version. Having been deprecated in 2012, the feature was not included in SQL Server 2017 for Linux—and never will be. If being used successfully for an existing application, there is no need to make a change at this time. But mirroring is a risky choice for any new HA configurations.

Log shipping is a way to reconstruct the current edition of a database by applying the transaction logs asynchronously from the active to an earlier copy on a “warm” (vs. hot and ready-to-go) standby instance. Because replaying logs can take a considerable amount of time
to complete and verify, log shipping might be suitable for some applications for DR, but not HA.

Windows Server Failover Clustering (mentioned above) might appear to be the perfect HA solution for applications running on Windows Server. But like FCIs, it requires the use of shared storage that is not always available, and it lacks the ability to detect failures automatically at the application level. WSFC does indeed have a potential role to play in many HA configurations, including for SQL Server FCIs, but its use requires separate data replication provisions in a SANless environment, whether in an enterprise datacenter or in the cloud.

Linux requires the use of separate open-source and/or commercial software to configure fully functional HA failover clusters. Using open-source software like Pacemaker and Corosync requires creating (and testing) custom scripts for each application, and these scripts often need to be updated and retested after even minor changes are made to any of the software or hardware being used. But because getting the full HA stack to work dependably for every application can be extraordinarily difficult, only very large organizations have the wherewithal needed to even consider taking on the effort.

Hypervisors provide their own “high availability” features to facilitate a reasonably quick recovery from failures at the host level. But they do nothing to protect against failures of the operating system of the guest VM or the application running in it. In effect, these features assure “dial tone” to a VM. That's it. Nothing less, but also nothing more—and much more is needed for HA at the application level. It also does nothing to minimize the planned downtime associated with applying Windows Server and SQL Server updates.

Public clouds all offer features to maximize the availability of services, and these are usually accompanied by a money-back guarantee in a service level agreement. But as with the hypervisors, the SLAs only guarantee “dial tone” at the server level. SQL Server needs additional provisions to ensure the data is being replicated from the active to standby instances, preferably across availability zones, as well as to ensure that clients are automatically reconnected to the new active instance after a failover. So while a cloud service provider's infrastructure can be leveraged in HA configurations, additional provisions are needed to ensure SQL Server remains available.

Storage Spaces Direct (S2D) is software-defined storage that makes it possible for direct-attached drives to be shared among multiple servers. S2D also makes it possible to configure a hyper-converged SQL Server FCI where the storage is all local to the cluster nodes. S2D first became available with Windows Server 2016 Datacenter Edition and is supported in SQL Server 2016 and later. Configurations are limited to a single data center, which means it cannot be used in clusters that span availability zones or regions.

The functional limitations and/or cost-prohibitive nature of all these options have created a need for third-party solutions purpose-built for providing HA and/or DR protections, and SIOS Technologies has been serving this market since 1999.

**SANless Failover Clustering with SIOS Technology**

SIOS Technology offers two separate SANless failover clustering solutions—one for Windows
Server and one for Linux—that are both designed to provide complete and cost-effective HA and DR protection.


SIOS Protection Suite for Linux provides the equivalent of the SIOS DataKeeper Cluster Edition in a complete DR/HA solution that combines real-time data replication with application-level failover clustering. The suite eliminates the need for organizations to struggle with do-it-yourself open source software projects. SIOS Protection Suite supports the only version of SQL Server currently available for Linux, SQL Server 2017.

SIOS DataKeeper for Windows Server Cluster Edition and SIOS Protection Suite for Linux share the following advanced and cost-effective HA/DR capabilities:

- A complete HA solution that integrates data replication with continuous application-level monitoring and automatic failover
- The data replication can be either synchronous or asynchronous for HA and DR, and is always real-time, block-level, multi-target and WAN-optimized
- The automatic and manual failover/failback recovery policies are easy to configure and enable planned maintenance to be performed with minimal downtime
- Implementation entirely in software affords virtually unlimited scalability across private, public and hybrid cloud infrastructures
- Ability to leverage the value-added capabilities while overcoming the limitations in popular public cloud offerings, including Microsoft Azure, Amazon Web Services and the Google Cloud Platform
- Support for physical, virtual or cloud instances
- A storage-agnostic design that supports both local and shared storage
- Support for a shared-nothing “SANless” configurations that eliminate all potential single points of failure
- Ability to deploy robust HA configurations with SQL Server Standard Edition, eliminating the need to upgrade to SQL Server Enterprise Edition just for Always On Availability Groups
- Simple wizard-driven implementation and an intuitive graphical user interface for “single pane of glass” monitoring and management
As mentioned previously, an important difference between the Windows Server and Linux operating systems is that Linux lacks the equivalent of WSFC to form a solid foundation for carrier-class HA. For Windows Server, SIOS DataKeeper is able to use WSFC to provide system administrators with a familiar and application-agnostic HA solution, thereby dramatically simplifying implementation and operation. For Linux, the SIOS Protection Suite for Linux provides a complete HA/DR solution that eliminates the need to customize open-source software.

**Getting Started**

The myriad of high availability and disaster recovery options confronting database and system administrators can be intimidating. And seemingly minor mistakes in the choices and resulting configurations can have major adverse consequences.

The critical importance of business continuity, combined with the complexity involved in configuring HA/DR provisions, have led a growing number of organizations to use solutions purpose-built for HA and DR from vendors like SIOS Technology.

To make it easy to get started, SIOS offers free trial versions for use in a development environment, and these are available on the Web at us.sios.com. SIOS also offers comprehensive documentation, an assortment of templates that automate all or part of application-specific and/or cloud-specific configurations, responsive support, and a variety of other useful resources to help get you started.

To learn more about how your organization can benefit from the carrier-class HA and DR protection afforded by SANless failover clustering capabilities offered by SIOS DataKeeper for Windows Server and the SIOS Protection Suite for Linux, please contact SIOS by phone at (650)645-7000 or by email at info@us.sios.com.