

# SQL Sentry Installation Recommendations

Last Modified on 11 November 2021

## Where to Install the SQL Sentry Components

The [SQL Sentry client](#), [monitoring service](#), and [database](#) are typically installed as follows:

- The [SQL Sentry database](#) is installed on a SQL Server [instance](#) on your network.
- The [SQL Sentry client](#) is installed on your workstation computer(s) with network connectivity to the SQL Sentry database.
- The [SQL Sentry monitoring service](#) is a Windows service and is installed on a dedicated server or VM with network connectivity to the SQL Sentry database server and monitored [targets](#).
  - For monitoring cloud targets, the SQL Sentry monitoring service may be installed either on-premises, or on a Windows virtual machine (IaaS) in the respective cloud platform.
- The [SQL Sentry portal service](#) is part of the optional [SQL Sentry Portal](#) feature that may be installed. It can be installed on a machine along with the SQL Sentry monitoring service and SQL Sentry client, or it can be installed on a machine by itself without any other SQL Sentry components (recommended).
  - If you are using the [EPI version](#) of SQL Sentry, then it must be installed on a machine that has the [SQL Sentry controller service](#).

**Note:** Installing the SQL Sentry database on Azure SQL Database Managed Instance is supported.

The SQL Sentry clients and monitoring services are each configured to connect to the same SQL Sentry database during setup.

**Important:** The SQL Sentry database must be installed on a SQL Server 2014 or higher instance. Azure SQL Database is not supported for the SQL Sentry database. For more information, see the [System Requirements](#) section.

## Use Fast Storage for the SQL Sentry Database

High disk latency (>5ms per read IO avg, >1-2ms per write IO avg) can limit scalability and cause sluggishness with the SQL Sentry client. The greater the number of monitored targets, the more data is collected and stored in the SQL Sentry database, and the more fast storage becomes important. Modern flash or hybrid storage systems should be used for maximum scalability and performance.

## Monitor the SQL Sentry Database (FREE)

You should always monitor the [SQL Sentry database](#) with [SQL Sentry](#) to ensure there are no CPU, memory, or disk bottlenecks. A free license is included in every SQL Sentry installation for this purpose. Simply add the SQL

Server as a monitored target.

## Install Components on the Same Network

Install the SQL Sentry client, monitoring service, and database on the same network for the best performance. For example, if the SQL Sentry client or monitoring service connect to a SQL Sentry database over a slow wide area network link or VPN over the internet, performance may suffer.

If a slow connection from your workstation to the SQL Sentry database is unavoidable, it is recommended to install the SQL Sentry client on a “jump box” on the same network as the database, and access it via RDP or another remote desktop technology.

## Installing the SQL Sentry Monitoring Service and Database on the Same Computer

For smaller environments, you may want to install a monitoring service on the same SQL Server machine where the SQL Sentry database is located. Doing so minimizes network overhead for communications between the monitoring service and the database. However, bear in mind that they will share the same compute resources which may cause CPU contention. For more information, see [System Requirements](#).

## Installing Multiple SQL Sentry Clients and Monitoring Services

Depending on the size of your SQL Server environment, you may need to install multiple SQL Sentry clients and monitoring services. Typically, each DBA has the SQL Sentry client installed on their workstation, and enough monitoring services are installed to handle the monitored target load (see examples below).

**Note:** There are two different options for installing SQL Sentry. The [Enhanced Platform Installer](#) option increases the speed and simplicity of installing and upgrading SQL Sentry implementations through a command line interface and is recommended for larger estates that will have more than one monitoring service or client.

## Installing SQL Sentry on Azure

In Azure, all the SQL Sentry components can be co-located on one IaaS VM or—to support larger numbers of monitored targets—must be set up on multiple VMs. As noted above, the SQL Sentry database may also be hosted on Azure SQL DB Managed Instance.

Consumption may increase if there are multiple sites (which we recommend mirroring to Azure Regions) and require redundancy in each site. In cases where redundancy is required, we recommend an N+1 approach with monitoring service VMs as described in the implementation examples ([below](#)). If you want to have a cluster for the SQL Sentry database for redundancy, that will also increase consumption.

In the SQL Sentry database implementation examples ([below](#)), the storage setup on the database is for two disks. One OS disk at 32 GB, and one data disk at an additional size dependent on workload. Unless workload demands moving TempDB to its own persistent disk, use the temp local disk.

#### Additional Information:

- For the SQL Sentry database, you can constrain the CPU count on the deployed system to reduce license cost for SQL Server. See the [Azure Constrained vCPU Capable VM Sizes](#) article from Microsoft for details.
- The storage estimates in the implementation examples ([below](#)) are simplified to request one managed disk of the size needed for data, but additional throughput and performance can be obtained by using multiple smaller disks as necessary. See the [Performance Guidelines for SQL Server in Azure Virtual Machines](#) article from Microsoft for more information.
- For more information about Azure consumption costs, please see the [Azure Pricing Calculator](#) from Microsoft.

## Apply the SQL Sentry Scalability Pack for more than 250 Targets

The [SQL Sentry Scalability Pack](#) implements partitioned clustered columnstore indexing, In-memory OLTP, and additional optimizations to achieve maximum scalability and performance, with reduced CPU, memory, and disk requirements. Contact us at [support.sentryone.com](mailto:support.sentryone.com) to schedule pack installation, which is free of charge. For more details, see the [SentryOne & Microsoft Achieve SQL Server Monitoring Performance Goals](#) blog post.

## Clustering SQL Sentry Monitoring Services

Multiple [monitoring services](#) can be installed to handle more than 100 monitored [targets](#), and/or to provide automatic redundancy and load balancing. There is no configuration required to implement a basic SQL Sentry cluster.

**Note:** Install more than one monitoring service and connect each to the same SQL Sentry database during setup. They automatically distribute the monitoring load, and if one monitoring service fails, the remaining monitoring service(s) picks up the load automatically. For more information, see the [Load Balancing and Fault Tolerance](#) article.

## Increased Fault Tolerance for the SQL Sentry Database

If increased fault tolerance is required for the [SQL Sentry database](#), install the database on a clustered SQL Server instance.

**Note:** Log shipping can also be used with the SQL Sentry database; however a separate SQL Sentry license is required for the standby server. Customers can obtain this standby license by visiting our [Customer Portal](#) and modifying the server name of their current license key to the name of the standby server and applying this license key to the SQL Sentry database on the standby server.

**Note:** You can also host the SQL Sentry database on an Availability group. For more information about Hosting the SQL Sentry database on an Availability group, see the [Hosting SQL Sentry Database on an Availability group](#) article.

**Important:** Due to performance considerations, we do not support **Synchronous-commit** mode for the Availability Group hosting the SQL Sentry database when monitoring over 200 targets or over high latency network links.

We recommend using **Asynchronous-commit** mode in this scenario.

See the following blogs from Microsoft for more information on this topic:

- [Analyze Synchronous Commit Impact on High Commit Rate Workloads](#)
- [AlwaysOn Asynchronous vs Synchronous Commit Performance for SharePoint](#)

## Implementation Examples - *Estimates Only*

### Monitoring Services and Targets per Site

Number of Monitoring Services	Maximum Number of Monitored Targets Range
1	40–100
2	80–200
3	120–300
5	200–500
10	400–1000

Each monitoring service requires a dedicated Windows server or VM with these minimum hardware specifications:

CPU	RAM	Azure VM	EC2 Class
4 cores 2.0+ GHz	12 GB	Standard_D4_v3 (4 cores (2.0+ GHz), 16 GB RAM) Standard SSD, E4: 32 GiB, single disk	m5.xlarge c5.2xlarge

**Note:** If monitoring and alerting redundancy is required, more than one monitoring service should be

installed in each site. When multiple monitoring services are installed in a site, use an n+1 approach to ensure that if a service fails, the remaining services will be able to handle the monitoring load.

**⚠ Important:** The actual number of services needed to optimally monitor an environment is heavily dependent on several factors such as:

- The number of databases on each target
- The number of disks on each target
- The workload on each target
- Geographical placement of datacenters
- The number of domains and the trust between those domains

SQL Sentry has settings to limit the maximum number of disks, databases, database files, and indexes to collect for each target, and these can be easily adjusted as needed. The SolarWinds support team ([support.sentryone.com](http://support.sentryone.com)) will work with you to ensure optimal configuration and performance.

## SQL Sentry Database Server

Number of Targets	Minimum Specs		Recommended Specs	
5	CPU	4 cores 1.6GHz	CPU	6 cores 2.0+GHz
	Memory	8 GB RAM	Memory	12 GB RAM
	Storage	10 GB	Storage	15 GB
	EC2 Class	db.m5.xlarge db.m4.xlarge	EC2 Class	db.m5.2xlarge db.m4.2xlarge
	Azure VM	Standard_D4s_v3 P4: 32 GiB (OS) P4: 32 GiB (Data)	Azure VM	Standard_D4s_v3 P4: 32 GiB (OS) P4: 32 GiB (Data)
	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 4 vCores Provisioned memory: 8 GB	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 8 vCores Provisioned memory: 12 GB

Number of Targets	Minimum Specs		Recommended Specs	
	10	CPU	6 cores 1.6GHz	CPU
Memory		12 GB RAM	Memory	16 GB RAM
Storage		20 GB	Storage	30 GB
EC2 Class		db.m5.2xlarge db.m4.2xlarge	EC2 Class	db.m5.2xlarge db.m4.2xlarge
Azure VM		Standard_D4s_v3 P4: 32 GiB (OS) P6: 64 GiB (Data)	Azure VM	Standard_D4s_v3 P4: 32 GiB (OS) P6: 64 GiB (Data)
Azure SQL Managed Instance		vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 8 vCores Provisioned memory: 12 GB	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 8 vCores Provisioned memory: 16 GB
25	CPU	8 cores 1.6GHz	CPU	8 cores 2.0+GHz
	Memory	20 GB RAM	Memory	32 GB RAM
	Storage	50 GB	Storage	75 GB
	EC2 Class	db.m5.2xlarge db.m4.2xlarge	EC2 Class	db.m5.2xlarge db.m4.2xlarge
	Azure VM	Standard_D8s_v3 P4: 32 GiB (OS) P10: 128 GiB (Data)	Azure VM	Standard_D8s_v3 P4: 32 GiB (OS) P10: 128 GiB (Data)
	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 8 vCores Provisioned memory: 20 GB	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 8 vCores Provisioned memory: 32 GB
	CPU	8 cores 1.6GHz	CPU	8 cores 2.0+GHz
	Memory	24 GB RAM	Memory	32 GB RAM

Number of Targets	Storage Minimum Specs		Storage Recommended Specs	
	50	100 GB	150 GB	
	EC2 Class	db.m3.2xlarge db.m4.2xlarge db.t2.2xlarge	EC2 Class	db.m5.2xlarge db.m2.4xlarge
	Azure VM	Standard_D8s_v3 P4: 32 GiB (OS) P15: 256 GiB (Data)	Azure VM	Standard_D8s_v3 P4: 32 GiB (OS) P15: 256 GiB (Data)
	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 8 vCores Provisioned memory: 24 GB	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 8 vCores Provisioned memory: 32 GB
100	CPU	8 cores 1.6GHz	CPU	12 cores 2.0+GHz
	Memory	48 GB RAM	Memory	64 GB RAM
	Storage	200 GB	Storage	300 GB
	EC2 Class	db.r3.2xlarge db.m2.4xlarge	EC2 Class	db.r4.4xlarge db.r3.4xlarge
	Azure VM	Standard_E16s_v3 P4: 32 GiB (OS) P15: 256 GiB (Data)	Azure VM	Standard_E16s_v3 P4: 32 GiB (OS) P15: 256 GiB (Data)
	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 8 vCores Provisioned memory: 48 GB	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 16 vCores Provisioned memory: 64 GB
	CPU	12 cores 2.0+GHz	CPU	16 cores 2.0+GHz
	Memory	96 GB RAM	Memory	128 GB RAM
	Storage	400 GB	Storage	600 GB

Number of Targets 200	<b>EC2 Class</b> <b>Minimum Specs</b> db.r4.4xlarge db.r3.4xlarge		<b>EC2 Class</b> <b>Recommended Specs</b> db.r5.4xlarge	
	Azure VM	Standard_E16s_v3 P4: 32 GiB (OS) P20: 512 GiB (Data)	Azure VM	Standard_E16s_v3 P4: 32 GiB (OS) P20: 512 GiB (Data)
	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 16 vCores Provisioned memory: 96 GB	Azure SQL Managed Instance	vCore model: General Purpose Compute hardware: Gen5 Provisioned compute: 16 vCores Provisioned memory: 128 GB
500*	CPU	16 cores 2.0+GHz	CPU	20 cores 2.0+GHz
	Memory	96 GB RAM	Memory	128 GB RAM
	Storage	400 GB	Storage	600 GB
	EC2 Class	db.r4.4xlarge db.r3.4xlarge	EC2 Class	db.r3.8xlarge
	Azure VM	Standard_E20s_v3 P4: 32 GiB (OS) P20: 512 GiB (Data)	Azure VM	Standard_E20s_v3 P4: 32 GiB (OS) P20: 512 GiB (Data)
	Azure SQL Managed Instance	vCore model: Business Class Compute hardware: Gen5 Provisioned compute: 16 vCores Provisioned memory: 96 GB	Azure SQL Managed Instance	vCore model: Business Class Compute hardware: Gen5 Provisioned compute: 24 vCores Provisioned memory: 128 GB
1000*	CPU	20 cores 2.0+GHz	CPU	24 cores 2.0+GHz
	Memory	192 GB RAM	Memory	256 GB RAM
	Storage	800 GB	Storage	1.2 TB
	EC2 Class	db.r4.8xlarge	EC2 Class	db.r4.16xlarge
	Azure VM	Standard_E32s_v3 P4: 32 GiB (OS) P20: 1 TiB (Data)	Azure VM	Standard_E32s_v3 P4: 32 GiB (OS) P20: 1 TiB (Data)



Number of Targets	Minimum Specs	Recommended Specs
	Azure SQL Managed Instance vCore model: Business Class Compute hardware: Gen5 Provisioned compute: 24 vCores Provisioned memory: 192 GB	Azure SQL Managed Instance vCore model: Business Class Compute hardware: Gen5 Provisioned compute: 24 vCores Provisioned memory: 256 GB

**⚠ Important:** \*Requires the [SQL Sentry Scalability Pack](#). When using Azure SQL Managed Instance, the Scalability Pack requires Business Class for the vCore model.

**📌 Note:** The EC2 DB Instance Class Types, Azure VMs, and Azure SQL Managed Instance recommendations are provided as examples. Please see:

- The [AWS DB Instance Class documentation](#) for the latest EC2 types and specifications available
- [vCore model overview - Azure SQL Database and Azure SQL Managed Instance](#) for the latest Azure VM and Azure SQL Managed Instance offerings available

**⚠ Important:** When running the SQL SentryDatabase server on VMware, adjustments may need to be made to the recommendations listed above to ensure that the VM is properly rightsized and vNUMA-aligned. Failure to do so can result in subpar performance.

**🔗 Additional Information:** Please see the following blog post, [Virtual Machine vCPU and vNUMA Rightsizing – Rules of Thumb](#) for more information.

The estimates provided are based on the default configuration of SQL Sentry. The hardware and resources needed by the [SQL Sentry database](#) to handle a monitored environment is heavily dependent on factors such as:

- The query workload
- Retention settings
- Collection interval settings

**🔗 Additional Information:** For more information see the [Performance Analysis Data Capacity Planning](#) article.