

# 소개

MSSQL 로그 수집기(LazyLog) 개요 및 필수 내용

## 로그 수집기 개요 [필독]

SQL 서버 운영 시 모니터링 대상 서버의 설정이나, 프로그램의 쿼리 성능, 서버의 이벤트 로그, 성능 카운터의 베이스라인을 수집해 두는 것은 매우 중요합니다. 이러한 베이스라인 데이터가 없다면 현재 서비스가 정상인지 (어제와 다른지) 판단에 오랜 시간이 걸릴 수 있습니다. 이에 네이버 클라우드에서는 SQL 서버 로그 수집 프로그램을 배포하고 있으며, 수집된 데이터는 장애 분석이나 성능 분석용 필수 데이터(윈도우 성능카운터, 윈도우 이벤트로그, SQL 에러로그 Agent 로그, 쿼리 구문, 쿼리 실행계획, 대기, 잠금 등 필수 DMV 스냅샷)입니다.

로그 수집 콘솔 응용 프로그램(LazyLog)은 SQL Server 2008, 2012, 2014, 2016 을 지원하며, 서버 리소스가 허락하는 한 무제한의 대상 서버, 다중 SQL 인스턴스, 다중 데이터베이스를 지원합니다. 로그 수집기 프로그램은 하나의 서버에서 동시에 여러 개 동작시킬 수 있으며, 이때는 폴더를 분리해 따로 실행하면 됩니다. 비슷한 서버 그룹을 묶어 모니터링 하면 서버의 설정이나 리소스 사용량 비교를 통해 정상 여부를 더욱 쉽게 판단할 수 있으므로 대상을 적절히 분리해 구성하는 것이 좋습니다. 여러 모니터링 대상 SQL 서버를 단 한 개의 그룹으로 모니터링 한다면, 윈도우 서비스프로그램으로도 사용 가능합니다. (설정 완료 후 Setup.exe 을 동작시키고 설정 파일 수정 후 서비스 시작).

LazyLog 는 사용자 정의 값(SqlLogManManager) 기반으로 대상 서버에 접근해 특정 데이터를 수집해 오고, 저장 기간을 초과한 데이터는 자동으로 삭제합니다. 수집 주기를 과도하게 짧게 설정하면 많은 로그를 저장해 데이터 저장용 데이터베이스가 매우 커지거나 수집 프로그램이 오동작할 수 있으므로, 수집 주기 조절과 데이터 보관 주기를 적절하게 설정해야 합니다. 또한 프로그램이 실행되기 위해 OS 여유 메모리를 최소 3GB 이상 남겨두길 권고합니다. (이 권고 여유 메모리는 SQL Server 운영시 OS 에 남겨두어야 할 메모리에 추가 확보할 메모리입니다.) 로그 수집 프로그램은 여러 모니터링 대상 서버로부터 받은 데이터를 메모리에 임시 보관하고 SQL Server 저장소 데이터베이스에 대량 입력(Bulk Insert)합니다.

관리 대상 SQL 서버가 매우 중요하다면, 해당 SQL 서버에 응용 프로그램이 설치되지 않기를 원할 것입니다. LazyLog 는 모든 데이터를 원격으로 수집 가능하게 개발되었으며, 로컬 SQL 서버를 대상으로도 모니터링 가능합니다. 이 프로그램은 수 TB 용량과 초당 수만 Batch Requests 를 수행하는 SQL 서버 수십 여대를 대상으로 로그수집 및 분석하고 있습니다. 원격 로그온 및 데이터 수집을 위해 윈도우 가장 계정(Windows Impersonation)과 SQL 인증이 지원됩니다. 성능 카운터와 이벤트 로그를 원격에서 읽기 위해 적절한 권한과 방화벽 개방이 필요합니다.

윈도우 성능카운터와 이벤트로그 수집을 위해 반드시 윈도우 인증(Active Directory or WorkGroup)이 사용되어야 하고, SQL Server 상태 모니터링을 위해서는 Windows 인증과 SQL 인증이 지원됩니다. (SQL 인증을 권고하며, 기본으로 SQL 인증이 설정되어 있습니다.) 계정의 암호설정은 반드시 LazyLog.exe 프로그램을 통해 암호화된 값 (안전한 암호 보관을 위해)을 생성해 사용해야 합니다. 설정은 두 곳에 저장 되는데, 첫 번째 LazyLog.exe.config 라는 XML 파일에는 글로벌 설정정보와 정보설정 데이터베이스 접근정보, 수집정보 저장용 데이터베이스 설정 정보가 저장되며, 두 번째 SqlLogManManger 라는 데이터베이스에 수집 대상 서버 접속 정보가 입력되어야 합니다.

로그 수집기 응용 프로그램(LazyLog.exe) 혹은 서비스 응용 프로그램이 설치되는 서버에서 (1)모니터링 대상 서버 정보가 들어있는 데이터베이스(SqlLogManManger)와 수집 데이터 저장용 데이터베이스(SqlLogManDmvRepository, SqlLogManPerfmonRepository) 서버에 접근 가능한지, (2)모니터링 대상 서버를 대상으로 성능카운터 모니터링 가능한지, (3)모니터링 대상 SQL Server 에 접근 가능한지 계정과 권한 테스트를 먼저 수행하고 설치한다면 빠른 시간에 설정을 마칠 수 있습니다.

수집 저장 데이터베이스가 느려지면 로그 수집 프로그램 메모리가 과도하게 사용될 수 있으므로, 로그 수집 프로그램은 데이터베이스 서버와 분리해 동작시키는 것이 좋습니다. 다수의 서버를 모니터링 할 경우, 저장소 스토리지로 SSD 를 추천합니다. 소수의 서버를 모니터링 해보고 천천히 대상을 늘려 수집 운영 서버 환경에서 로그 수집기 프로그램의 리소스 사용량을 판단해 모니터링 대상 서버를 늘려갑니다. 로그 수집 프로그램 오류 정보는 Log 폴더에 일별 아이템별로 생성되게 되어 있으므로, 수집에 문제가 있으면 참고합니다.

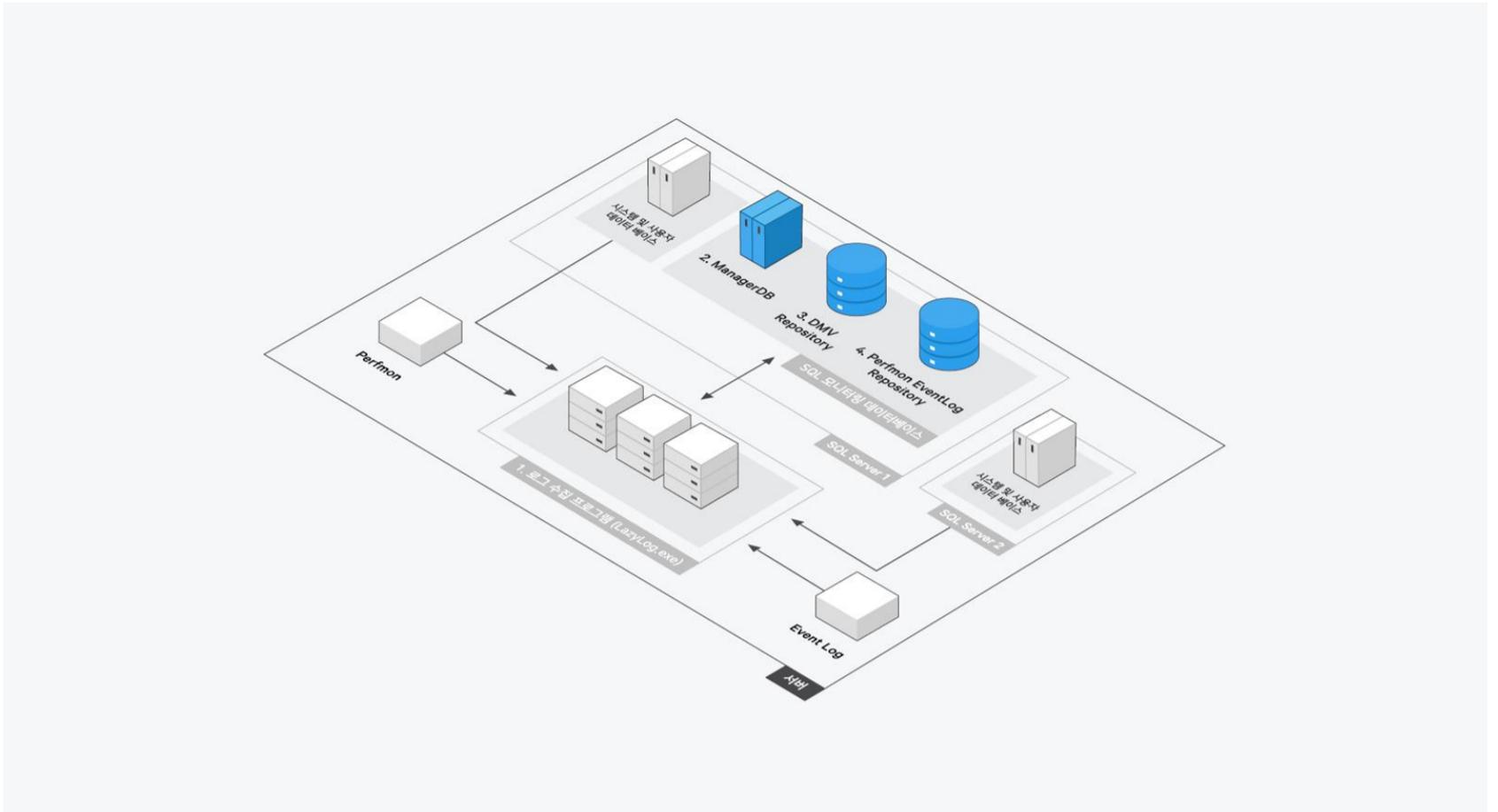
## 구성 가이드

MSSQL 로그 수집기(LazyLog) 설치 구조

### 로그 수집기 구성 방법

로그 수집기는 원격 수집이 가능하므로 수집 대상 서버와 별도로 설치하는 것을 권고하고 수집 대상 서버에 직접 설치도 가능합니다. 아래 몇몇 예시를 들어 설치 가능한 구성을 설명합니다. 본 설치 가이드에서는 방화벽과 계정 설정에 제한이 없는 로컬 설치(구성안 1)를 기본으로 설명합니다. 로컬 구성을 먼저 해보고 LazyLog 응용 프로그램 구성과 사용법을 익힌 후 원격 수집을 시도합니다.

## 구성안 1

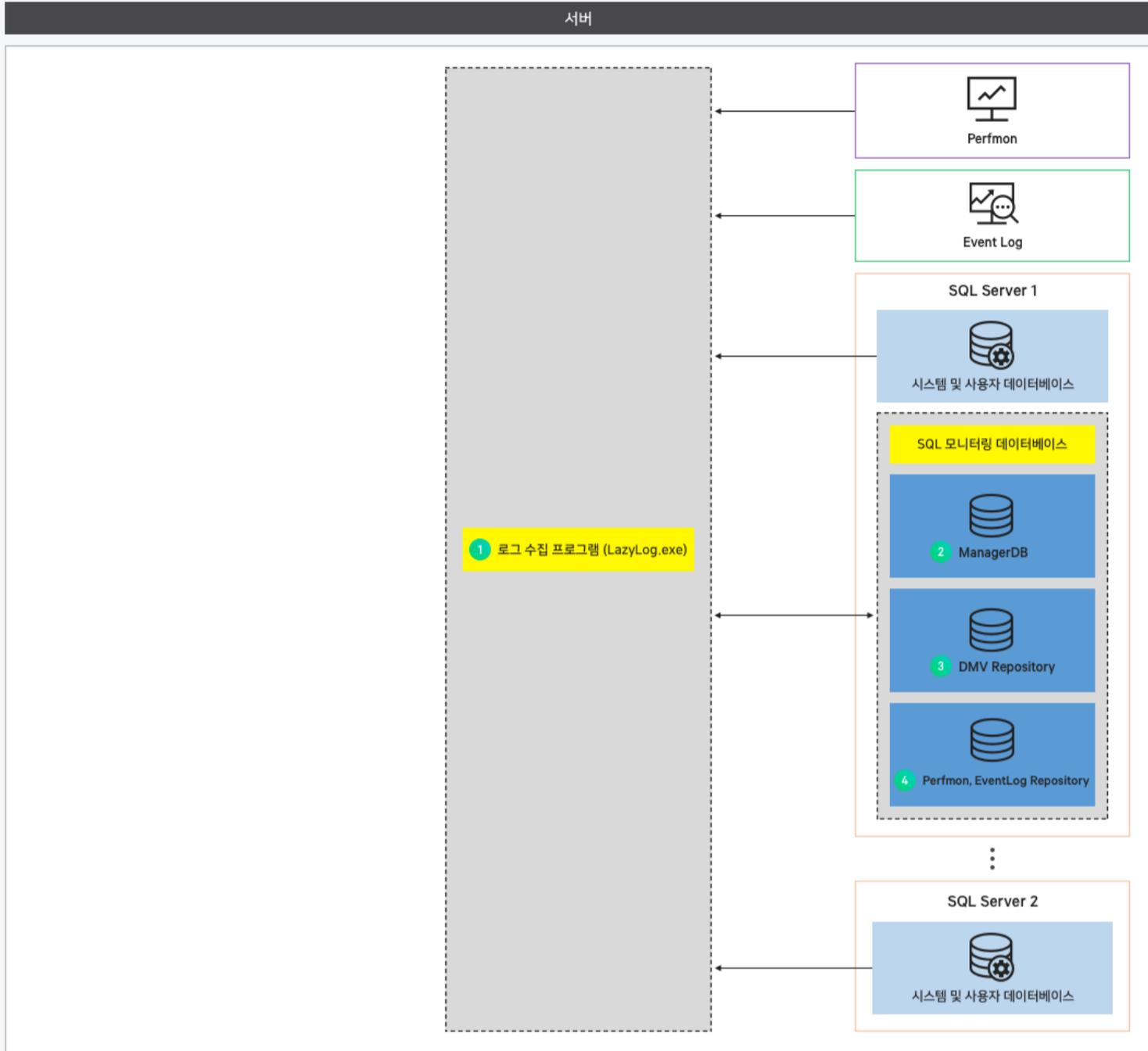


구성안 1은 가장 간단한 구성이며, LazyLog 사용법을 익히기 위한 구성입니다.

이 구성은 방화벽 설정이 필요하지 않습니다. 로그 수집 프로그램을 모니터링 대상 SQL Server에 직접 설치하고, 모니터링 데이터베이스도 같은 SQL Server에 생성해 사용하는 방법입니다. 이 경우 네트워크를 통하지 않아 성능 카운터나 이벤트 로그 수집 속도는 빠르나 로그 수집 프로그램이 이상 동작하거나 (일반적으로 Agent 프로그램이 운영서버에서 동작하는 경우의 문제들) SQL 모니터링 데이터베이스 용량이 늘어날 경우 운영 서버에 영향을 줄 수 있어 권고하지 않습니다.

1대의 서버만 모니터링하는 경우 DMV, 성능 카운터 정보만 데이터베이스에 저장하고 이벤트 로그는 데이터베이스에 저장하지 않아도 됩니다. (직접 이벤트 로그를 조회하는 것이 더 좋을 수 있기 때문입니다). 그러나, 어떤 특정 이벤트로 알람을 보내고자 할 때는 이벤트 로그를 SqlLogManPerfmonRepository(기본 저장소 이름이며 변경 가능합니다.)에 저장하면 좀 더 편리하게 관리 가능합니다.

## 구성안 2



구성안 2는 대부분의 모니터링 서버에 권장되는 모델입니다.

사용자 서버에는 어떠한 프로그램도 설치하지 않고 모니터링 서버를 따로 구성해 해당 서버에 SQL Server를 설치하고 ManagerDB(SqlLogManManager), DMV Repository(SqlLogManDmvRepository), Perfmon EventLog(SqlLogManPerfmonRepository)를 만들고 로그 수집 프로그램은 모니터링 대상 서버를 대상으로 원격으로 정보를 수집해 SQL 모니터링 데이터베이스에 데이터를 저장합니다.

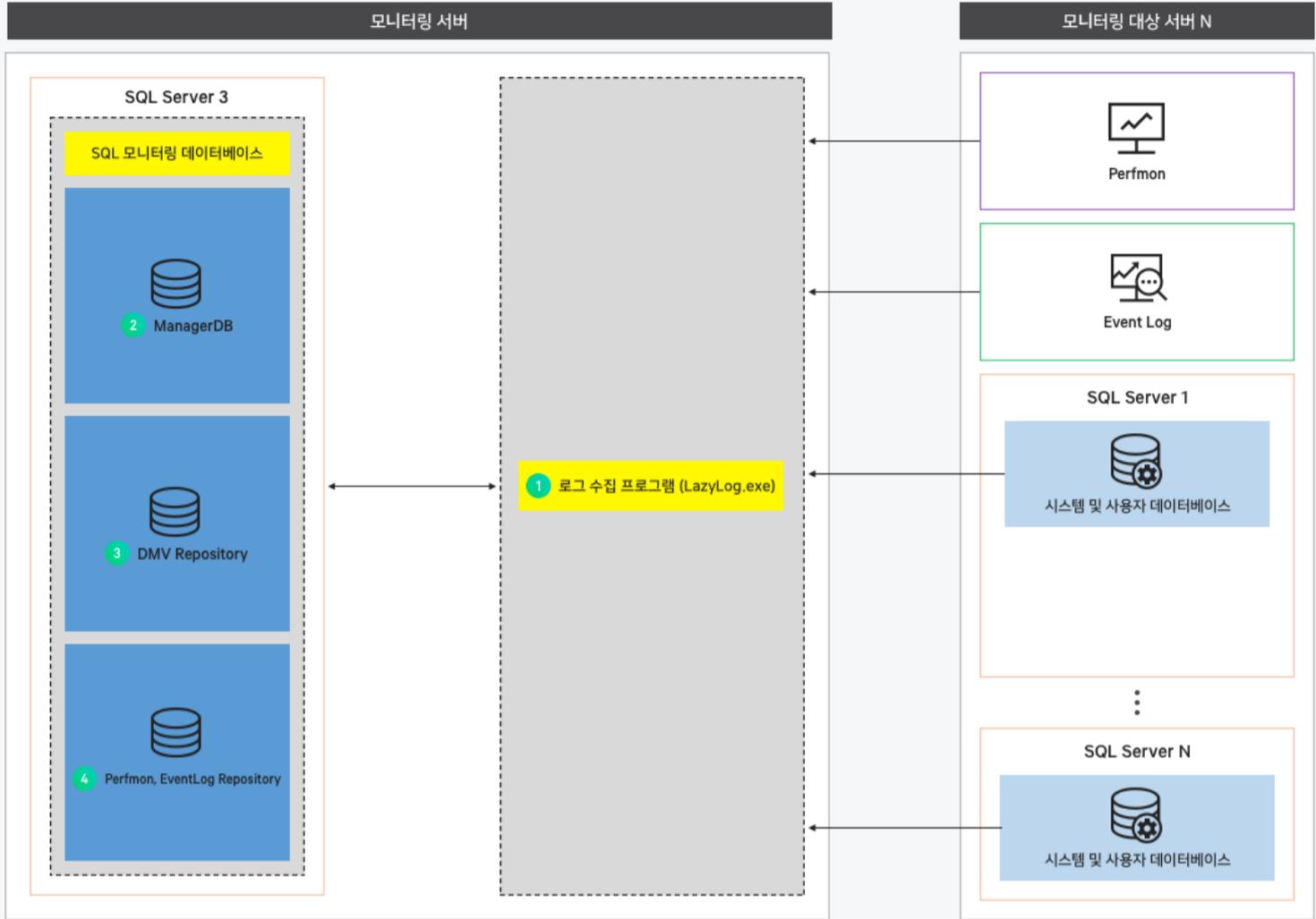
모니터링 서버를 따로 구성하고, 다수의 서버를 모니터링하면 성능 카운터 수집에 네트워크 대역폭 부담이 있을 수 있습니다. Manager 데이터베이스의 Config 테이블에 Perfmon 항목을 적절히 수정해 씁니다. (배포된 프로그램의 기본값은 동일 IDC 내 있는 원격 서버들을 모니터링하기 적절한 값으로 설정되어 있습니다.)

모니터링 서버에 문제가 생기거나, 모니터링 대상 서버의 문제가 모니터링으로 인해 발생하는 것을 검증하기 위해서 모니터링 대상 서버에서 모니터링 서버에 설정된 계정 정보를 차단하거나 수집 프로그램을 정지해보면 쉽게 모니터링이 해당 데이터베이스에 문제를 일으키는지 검증할 수 있는 장점이 있습니다.

이 모니터링 모델은 모니터링 대상 서버의 리소스가 거의 없어 (대상 서버 CPU 100% 등) 모니터링 쿼리도 동작하지 못할 경우 데이터를 수집해 오지 못할 수 있는 문제는 가지고 있습니다. 이 문제는 원격 모니터링 모델에서 피할 수 없는 문제입니다. 구성안 1을 적용하더라도 완전한 리소스 부족에서는 모니터링 데이터를 저장하지 못할 수 있습니다.

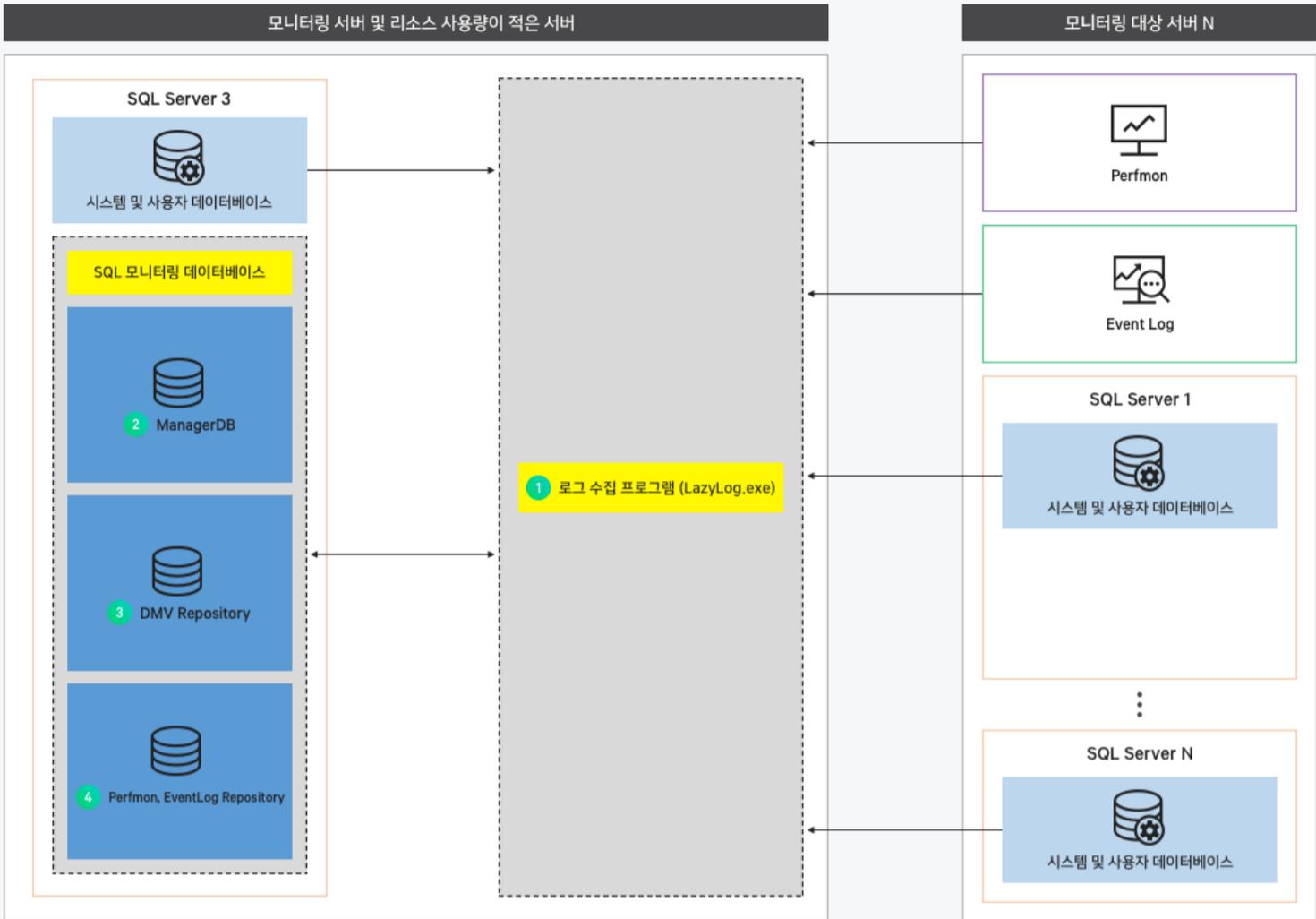
일부 ISP 업체에서는 445 포트 개방이 불가능 하므로, 모니터링 서버가 모니터링 대상 서버와 같은 IDC에 설치하기를 권장합니다. 이 모델은 독립 SQL Server가 모니터링에만 사용되므로 라이선스 비용이 부담이 될 수 있습니다. 적절한 CAL 라이선스 모델이 도움이 될 수 있습니다.

### 구성안 3



SQL Server 라이선스 문제가 있다면 서버 사용량이 낮은 서버에서 모니터링 프로그램과 모니터링 데이터베이스를 같이 설치해 운영할 수 있습니다.

## 구성안 4



로그 수집 프로그램의 리소스 사용량이 부담 된다면, 모니터링 프로그램만 다른 서버를 이용해 구성 가능합니다. 또 위 구성안 1, 2, 3, 4 구성을 적절히 혼합해 구성 가능합니다. 단 구성안 4와 같이 SQL 모니터링 데이터베이스가 다른 인스턴스나 서버로 구성된다면, 미리 구성된 동의어 2 개를 적절히 수정해야 정상적인 모니터링이 가능합니다. (동의어는 SqlLogManManager 데이터베이스에 SN\_CounterDetailsd와 SqlLogManPerformanceRepository 데이터베이스에 CounterDetailsAutoUpdated가 있습니다.)

동의어 : <https://msdn.microsoft.com/ko-kr/library/ms187552.aspx>

## 수집 데이터

MSSQL 로그 수집기(LazyLog)에서 수집 가능한 데이터

### 성능카운터

윈도우 성능카운터 수집이 가능하며, SqlLogManManager.dbo.CounterDetailsFilterinfo 테이블 정보를 기반으로 어떤 카운터가 서버에 있는지 읽어오고, 읽어온 카운터는 SqlLogManManager.dbo.CounterDetailsAutoUpdated에 저장됩니다.

SqlLogManManager.dbo.CounterDetailsFilterinfo의 FilterType [IN]은 수집 대상 오브젝트를 말하며, FilterType [NOTIN]은 수집 제외 대상 오브젝트입니다. 이 값은 설치 후 사용자가 직접 수정 가능합니다.

```
SELECT *
FROM SqlLogManManager.dbo.CounterDetailsFilterInfo
```

Idx	FilterType	ObjectName
1	IN	LogicalDisk
2	IN	:Buffer Manager
3	IN	:CLR
4	IN	:Databases
5	IN	:Latches
6	IN	Processor
7	IN	:SQL Statistics
8	IN	:Database Replica
9	IN	:Availability Replica
10	IN	SQLServer:General Statistics
11	IN	Network Adapter
12	NOTIN	Hyper-V
13	NOTIN	Per Processor Network
14	NOTIN	Processor Information
15	NOTIN	Physical Network Interface Card Activity

### SqlLogManManager.dbo.CounterDetailsFilterInfo

```
SELECT [MachineName]
, [ObjectName]
, [CounterName]
, [CounterType]
, [DefaultScale]
, [InstanceName]
, [InstanceIndex]
, [ParentName]
, [ParentObjectId]
, [TimeBaseA]
, [TimeBaseB]
, [IsEnabledYN]
FROM SqlLogManManager.dbo.CounterDetailsAutoUpdated
```

MachineName	ObjectName	CounterName	CounterType	DefaultScale	InstanceName	InstanceIndex	ParentName	ParentObjectId	TimeBaseA	TimeBaseB	IsEnabledYN
AD01047147	LogicalDisk	% Disk Read Time	542573824	0	._Total	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Read Time	542573824	0	C:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Read Time	542573824	0	D:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Read Time	542573824	0	E:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Read Time	542573824	0	F:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Read Time	542573824	0	HarddiskVolume1	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Time	542573824	0	._Total	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Time	542573824	0	C:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Time	542573824	0	D:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Time	542573824	0	E:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Time	542573824	0	F:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Time	542573824	0	HarddiskVolume1	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Write Time	542573824	0	._Total	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Write Time	542573824	0	C:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Write Time	542573824	0	D:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Write Time	542573824	0	E:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Write Time	542573824	0	F:	0		0	10000000	0	N
AD01047147	LogicalDisk	% Disk Write Time	542573824	0	HarddiskVolume1	0		0	10000000	0	N
AD01047147	LogicalDisk	% Free Space	537003008	0	._Total	0		0	3312640	0	Y
AD01047147	LogicalDisk	% Free Space	537003008	0	C:	0		0	3312640	0	Y
AD01047147	LogicalDisk	% Free Space	537003008	0	D:	0		0	3312640	0	Y
AD01047147	LogicalDisk	% Free Space	537003008	0	E:	0		0	3312640	0	Y
AD01047147	LogicalDisk	% Free Space	537003008	0	F:	0		0	3312640	0	Y
AD01047147	LogicalDisk	% Free Space	537003008	0	HarddiskVolume1	0		0	3312640	0	Y

### SqlLogManManager.dbo.CounterDetailsAutoUpdated

이 테이블의 구성은 윈도우 성능 카운터 스키마와 같고, 위 필터 정보에 해당하는 카운터 정보 모두를 저장하고 있습니다. IsEnabledYN 필드만 추가되었는데, 이 필드가 Y 인 것만 SqlLogManPerfmonRepository 데이터베이스에 15 초마다 저장됩니다. (저장 간격 제한 : 로컬 최소 5 초, 리모트 최소 15 초)

실제 저장되는 테이블은 SqlLogManPerfmonRepository 의 CounterData, CounterDetails, DisplayToID 입니다. 기본 수집 Y 항목을 수정하고자 한다면, 설치 폴더의 Resources 에서 UpdateCounterDetailsY.sql 를 수정해 업데이트하면 됩니다. 원래 카운터 항목에는 없는 디스크 볼륨별 Free, Total, Used Megabytes 를 계산해 입력되도록 디자인되어 있으므로 유용하게 쓰길 바랍니다

이 3 개 테이블의 스키마는 [https://msdn.microsoft.com/en-us/library/aa371915\(VS.85\).aspx](https://msdn.microsoft.com/en-us/library/aa371915(VS.85).aspx) 를 참고합니다.

- CounterData
- CounterDetails
- DisplayToID

# CounterData

The **CounterData** table contains a row for each counter that is collected at a particular time. There will be a large number of these rows.

The **CounterData** table defines the following fields:

- **GUID:** GUID for this data set. Use this key to join with the **DisplayToID** table.
- **CounterID:** Identifies the counter. Use this key to join with the **CounterDetails** table.
- **RecordIndex:** The sample index for a specific counter identifier and collection GUID. The value increases for each successive sample in this log file.
- **CounterDateTime:** The time the collection was started, in UTC time.
- **CounterValue:** The formatted value of the counter. This value may be zero for the first record if the counter requires two sample to compute a displayable value.
- **FirstValueA:** Combine this 32-bit value with the value of **FirstValueB** to create the **FirstValue** member of **PDH\_RAW\_COUNTER**. **FirstValueA** contains the low order bits.
- **FirstValueB:** Combine this 32-bit value with the value of **FirstValueA** to create the **FirstValue** member of **PDH\_RAW\_COUNTER**. **FirstValueB** contains the high order bits.
- **SecondValueA:** Combine this 32-bit value with the value of **SecondValueB** to create the **SecondValue** member of **PDH\_RAW\_COUNTER**. **SecondValueA** contains the low order bits.
- **SecondValueB:** Combine this 32-bit value with the value of **SecondValueA** to create the **SecondValue** member of **PDH\_RAW\_COUNTER**. **SecondValueB** contains the high order bits.

The **GUID**, **CounterID**, and **RecordIndex** fields make up the primary key for this table.

- CounterData
- CounterDetails
- DisplayToID

# CounterDetails

The **CounterDetails** table describes a specific counter on a particular computer.

The **CounterDetails** table defines the following fields:

- **CounterID:** A unique identifier in the database that maps to a specific counter name text string. This field is the primary key of this table.
- **MachineName:** The name of the computer that logged this data set.
- **ObjectName:** The name of the performance object.
- **CounterName:** The name of the counter.
- **CounterType:** The counter type. For a list of counter types and their formulas, see the Counter Types section of the [Windows Server 2003 Deployment Kit](#).
- **DefaultScale:** The default scaling to be applied to the raw performance counter data.
- **InstanceName:** The name of the counter instance.
- **InstanceIndex:** The index number of the counter instance.
- **ParentName:** Some counters are logically associated with others, and are referred to as parents. For example, the parent of a thread is a process and the parent of a logical disk driver is a physical drive. This field contains the name of the parent. Either the value in this field or the **ParentObjectID** field identifies a specific parent instance. If the value in this field is **NULL**, the value in the **ParentObjectID** field must be checked to identify the parent. If the values in both fields are **NULL**, the counter does not have a parent.
- **ParentObjectID:** The unique identifier of the parent. The value in either this field or the **ParentName** field identifies a specific parent instance. If the value in this field is **NULL**, the value in the **ParentName** field must be checked to identify the parent.

- CounterData
- CounterDetails
- DisplayToID

# DisplayToID

The **DisplayToID** table relates the user-friendly string displayed by the System Monitor to the GUID stored in the other tables.

The **DisplayToID** table defines the following fields:

- **GUID:** Unique identifier generated for a log. This field is the primary key of this table.
- **RunID:** Reserved for internal use.
- **DisplayString:** Name of the log file as displayed in the System Monitor.
- **LogStartTime:** Time the logging process started in yyyy-mm-dd hh:mm:ss:nnn format.
- **LogStopTime:** Time the logging process stopped in yyyy-mm-dd hh:mm:ss:nnn format. Multiple log files with the same **DisplayString** value can be differentiated by using the value in this and the **LogStartTime** fields. The values in the **LogStartTime** and **LogStopTime** fields also allows the total collection time to be accessed quickly.
- **NumberOfRecords:** Number of samples stored in the table for each log collection.
- **MinutesToUTC:** Value used to convert the row data stored in UTC time to local time.
- **TimeZoneName:** Name of the time zone where the data was collected. If you are collecting or have relogged data from a file collected on systems in your own time zone, this field will state the location.

**Note** Prior to Windows Vista, the data collector sets were stored in the registry at

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\SysmonLog\Log Queries**

. The fields listed above do not correspond to the values in registry. For Windows Vista, the data collector sets are not stored in the registry.

```

SQLQuery5.sql - (...47147#Naver (95))* x SQLQuery4.sql - (...47147#Naver (93))* SQLQuery3.sql - (...47147#Naver (60))* L04_TargetServerR...47147#Naver (92)) L01_InitScript
/***** SSMS의 SelectTopNRows 명령 스크립트 *****/
SELECT TOP (1000) [CounterID]
, [MachineName]
, [ObjectName]
, [CounterName]
, [CounterType]
, [DefaultScale]
, [InstanceName]
, [InstanceIndex]
, [ParentName]
, [ParentObjectID]
, [TimeBaseA]
, [TimeBaseB]
FROM [SqlLogManPerfmonRepository].[dbo].[CounterDetails]
where counterName like '%Megabytes%'

```

CounterID	MachineName	ObjectName	CounterName	CounterType	DefaultScale	InstanceName	InstanceIndex	ParentName	ParentObjectID	TimeBaseA	TimeBaseB
1	32	WWWAD01047147	LogicalDisk	Free Megabytes	65536	0	_Total	0	0	3312640	0
2	33	WWWAD01047147	LogicalDisk	Free Megabytes	65536	0	C:	0	0	3312640	0
3	34	WWWAD01047147	LogicalDisk	Free Megabytes	65536	0	D:	0	0	3312640	0
4	35	WWWAD01047147	LogicalDisk	Free Megabytes	65536	0	E:	0	0	3312640	0
5	36	WWWAD01047147	LogicalDisk	Free Megabytes	65536	0	F:	0	0	3312640	0
6	37	WWWAD01047147	LogicalDisk	Free Megabytes	65536	0	HarddiskVolume1	0	0	3312640	0
7	65	WWWAD01047147	LogicalDisk	Total Megabytes	65536	0	_Total	0	0	3312640	0
8	66	WWWAD01047147	LogicalDisk	Total Megabytes	65536	0	C:	0	0	3312640	0
9	67	WWWAD01047147	LogicalDisk	Total Megabytes	65536	0	D:	0	0	3312640	0
10	68	WWWAD01047147	LogicalDisk	Total Megabytes	65536	0	E:	0	0	3312640	0
11	69	WWWAD01047147	LogicalDisk	Total Megabytes	65536	0	F:	0	0	3312640	0
12	70	WWWAD01047147	LogicalDisk	Total Megabytes	65536	0	HarddiskVolume1	0	0	3312640	0
13	71	WWWAD01047147	LogicalDisk	Used Megabytes	65536	0	_Total	0	0	3312640	0
14	72	WWWAD01047147	LogicalDisk	Used Megabytes	65536	0	C:	0	0	3312640	0
15	73	WWWAD01047147	LogicalDisk	Used Megabytes	65536	0	D:	0	0	3312640	0
16	74	WWWAD01047147	LogicalDisk	Used Megabytes	65536	0	E:	0	0	3312640	0
17	75	WWWAD01047147	LogicalDisk	Used Megabytes	65536	0	F:	0	0	3312640	0
18	76	WWWAD01047147	LogicalDisk	Used Megabytes	65536	0	HarddiskVolume1	0	0	3312640	0

**디스크 볼륨별 Free, Total, Used Megabytes**

수집이 정상적으로 될 경우 아래와 같은 데이터가 조회 가능합니다.

SQLCMD 명령어 [서버]

```

select * from SqlLogManPerfmonRepository.[dbo].[DisplayToID]
select * from SqlLogManPerfmonRepository.[dbo].[CounterDetails]
select * from SqlLogManPerfmonRepository.[dbo].[CounterData]

```

```

select * from SqlLogManPerfmonRepository.[dbo].[DisplayToID]
select * from SqlLogManPerfmonRepository.[dbo].[CounterDetails]
select * from SqlLogManPerfmonRepository.[dbo].[CounterData]

```

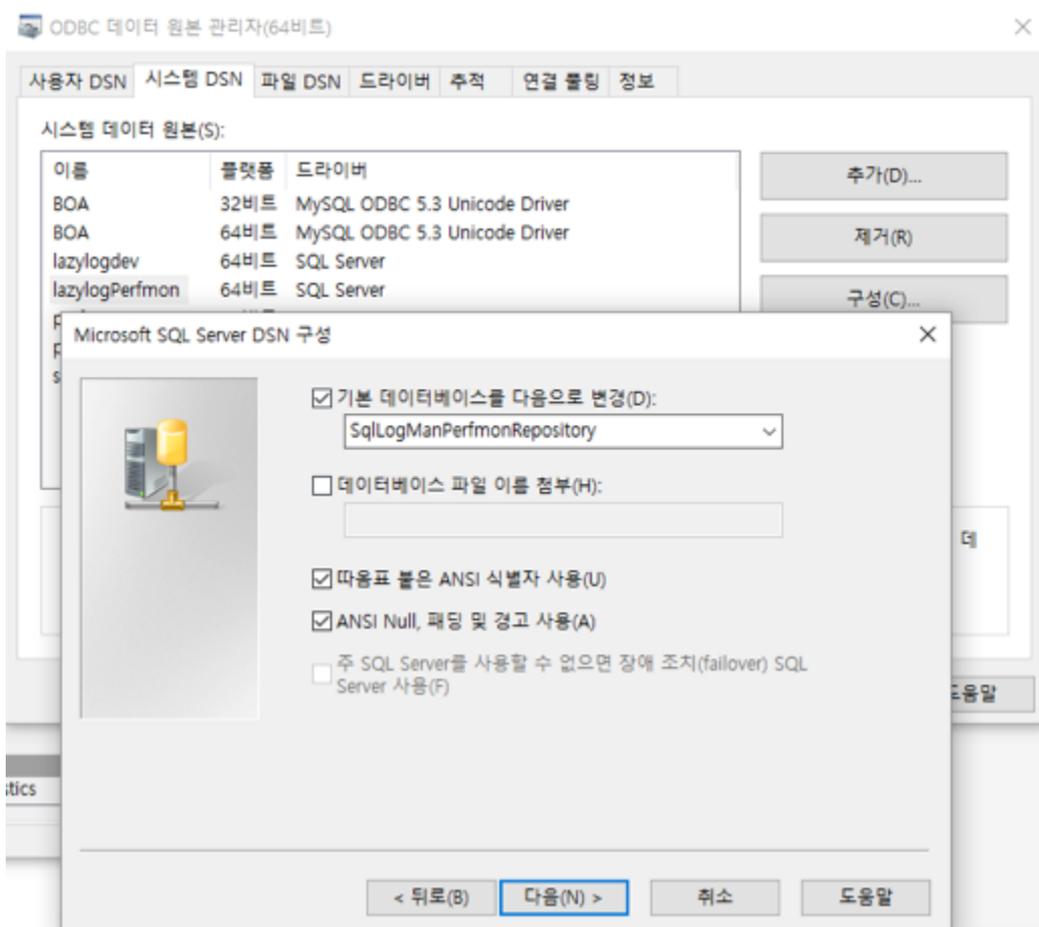
GUID	RunID	DisplayString	LogStartTime	LogStopTime	NumberOfRecords	MinutesToUTC	TimeZoneName
1	EC814398-89A5-4849-8CBB-E4203549FC42	0	BaselineCollect	2017-02-24 13:46:56	2017-02-24 14:55:55	151	-540 대한민국 표준시

CounterID	MachineName	ObjectName	CounterName	CounterType	DefaultScale	InstanceName	InstanceIndex	ParentName	ParentObjectID	TimeBaseA	TimeBaseB
1	1	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	_Total	0	0	3312637	0
2	2	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	C:	0	0	3312637	0
3	3	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	D:	0	0	3312637	0
4	4	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	E:	0	0	3312637	0
5	5	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	F:	0	0	3312637	0
6	6	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	HarddiskVolume1	0	0	3312637	0

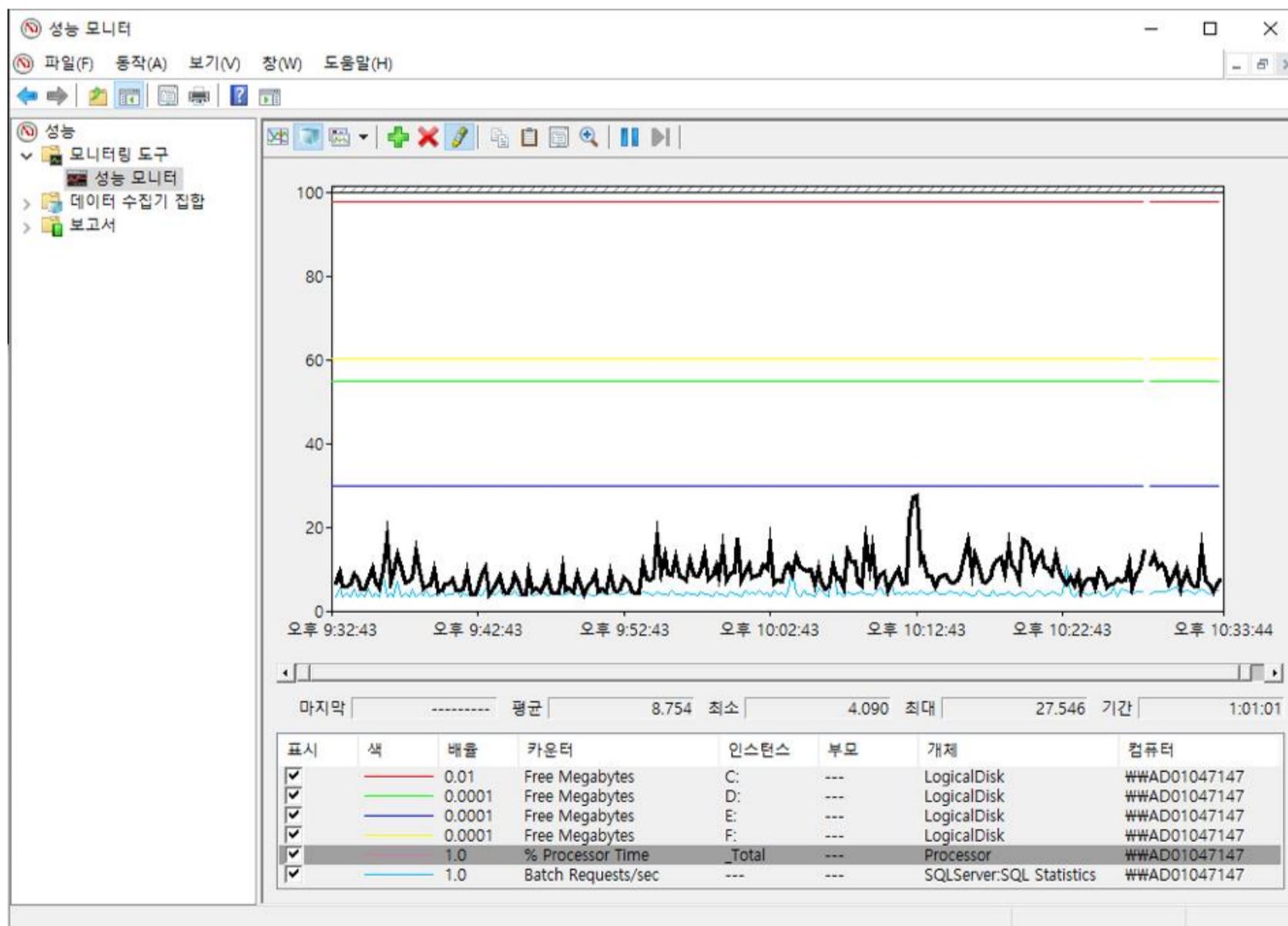
GUID	CounterID	RecordIndex	CounterDateTime	CounterValue	FirstValueA	FirstValueB	SecondValueA	SecondValueB	MultiCount
1	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:46:56.769	46,441,101,074,218.8	1463627	0	3151577	0	1
2	EC814398-89A5-4849-8CBB-E4203549FC42	2	2017-02-24 13:47:11.803	46,441,101,074,218.8	1463627	0	3151577	0	1
3	EC814398-89A5-4849-8CBB-E4203549FC42	3	2017-02-24 13:47:26.834	46,441,101,074,218.8	1463627	0	3151577	0	1
4	EC814398-89A5-4849-8CBB-E4203549FC42	4	2017-02-24 13:47:41.899	46,441,101,074,218.8	1463627	0	3151577	0	1
5	EC814398-89A5-4849-8CBB-E4203549FC42	5	2017-02-24 13:47:56.980	46,441,101,074,218.8	1463627	0	3151577	0	1
6	EC814398-89A5-4849-8CBB-E4203549FC42	6	2017-02-24 13:48:11.993	46,441,101,074,218.8	1463627	0	3151577	0	1
7	EC814398-89A5-4849-8CBB-E4203549FC42	7	2017-02-24 13:48:27.010	46,441,101,074,218.8	1463627	0	3151577	0	1
8	EC814398-89A5-4849-8CBB-E4203549FC42	8	2017-02-24 13:48:42.097	46,441,101,074,218.8	1463627	0	3151577	0	1
9	EC814398-89A5-4849-8CBB-E4203549FC42	9	2017-02-24 13:48:57.159	46,441,101,074,218.8	1463627	0	3151577	0	1

**정상 수행 중인 데이터를 SSMS 에서 조회한 결과**

윈도우 실행에서 odbcad32 를 실행(ODBC 데이터 원본 관리자(64 비트)) 및 설정 수행 후 윈도우 내장 성능 카운터 프로그램으로 연결하면 저장된 기록을 볼 수 있습니다. (윈도우 계정만 가능하므로, 해당 윈도우 계정이 모니터링 수집 서버에 로그인 및 data\_reader 권한이 있어야 합니다.)



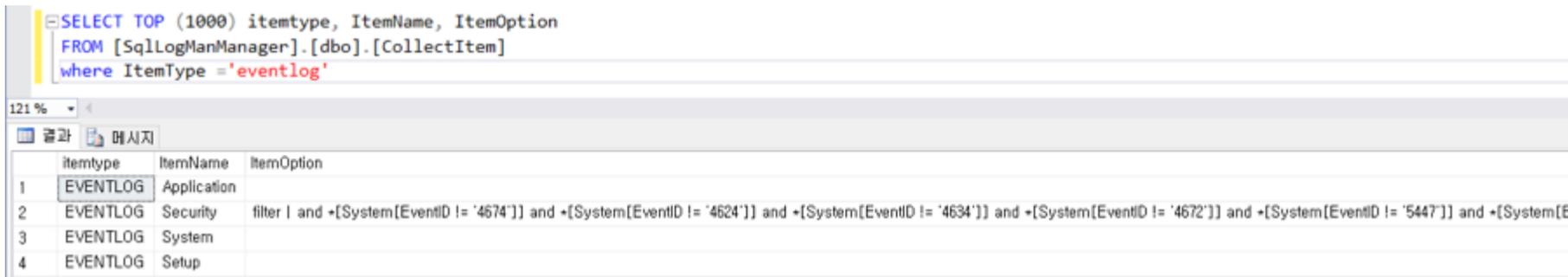
## ODBC 구성



LazyLog 에서 수집된 데이터를 Perfmon 응용 프로그램으로 조회

# 이벤트로그

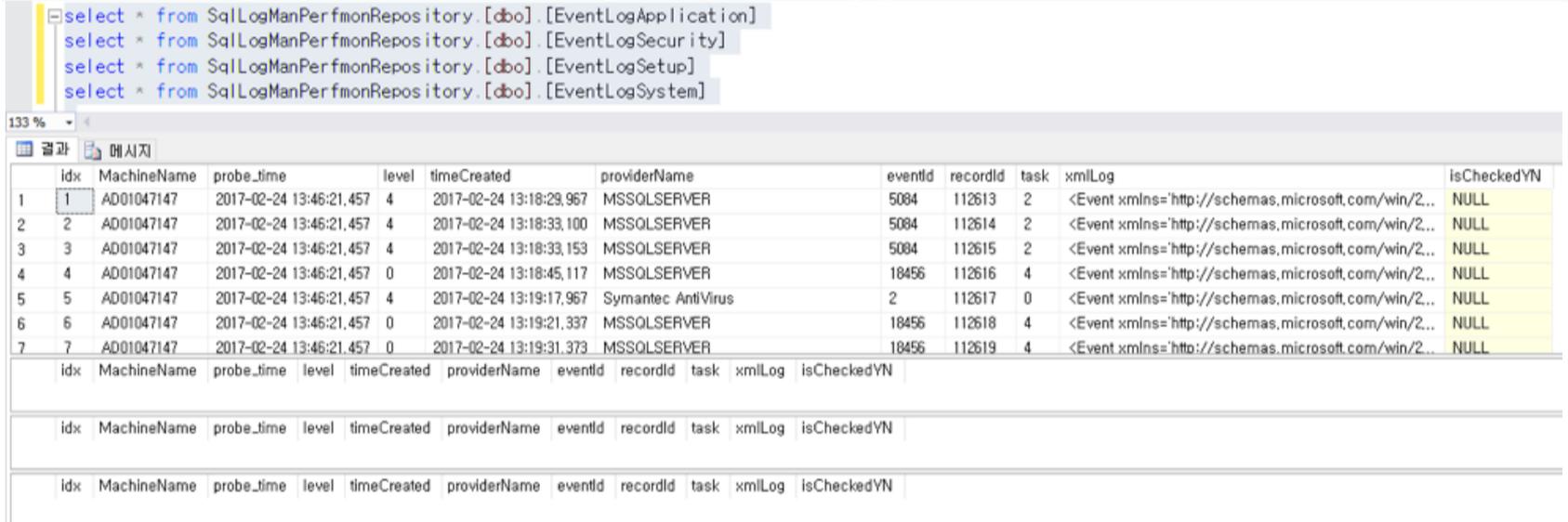
서버의 Application, Security, Setup, System 이벤트 로그를 읽어오는 것이 가능하며, 특정 정보를 필터해 저장하는 것도 가능합니다. 이 이벤트 정보의 서버별 필터는 SqlLogManManager.dbo.CollectItem 테이블의 ItemOption 에 정의되어 있습니다.



이 정보는 SqlLogManManager.dbo.CollectItemTemplate 테이블을 기준으로 SqlLogManManager.dbo.usp\_register\_CollectItem 프로시저를 통해 입력되었고, 개별 수정이나 전체 수정이 가능합니다.

## SQLCMD 명령어

```
select * from SqlLogManPerfmonRepository.[dbo].[EventLogApplication]
select * from SqlLogManPerfmonRepository.[dbo].[EventLogSecurity]
select * from SqlLogManPerfmonRepository.[dbo].[EventLogSetup]
select * from SqlLogManPerfmonRepository.[dbo].[EventLogSystem]
```



## LazyLog 를 통해 Application 로그를 저장한 경우 조회 결과

저장된 이벤트 로그는 직접 윈도우 기본 eventvwr 응용 프로그램에서 읽는 것과 같게 저장되어 있습니다. SMS 전송을 위해 IsCheckedYN 필드를 추가해 두었으니, 특정 이벤트에 메시지를 보낼 때 유용하게 쓰길 바랍니다 예제에서 EventLogSecurity, EventLogSetup, EventLogSystem 은 기본 모니터링에 바로 추가 가능한 항목입니다.

# DMV (Dynamic Management View)

Always On Availability 의 수행 현황, 현재 상태, 쿼리별 리소스 사용량, 쿼리, 쿼리 플랜, 잠금, 대기 분석, 파일별 I/O 양, 인덱스 구조, 통계 일자, 서버 메모리 현황, Agent 실행 기록 등을 수집해 저장하고 있습니다.

LazyLog 는 SQL Server 운영, 문제 해결에 도움이 되는 대부분의 정보를 수집하고 있습니다. 자세한 모니터링 항목은 아래 표를 참고합니다.

항목이름	중요도	항목유형	항목구분	설명
availability_replicas	2	AG (Availability Group)	DMV	가용성 복제본 정보
dm_hadr_auto_page_repair	2	AG (Availability Group)	DMV	가용성 복제본의 모든 가용성 데이터베이스에 대해 수행하는 자동 페이지 복구에 대한 시도 정보

항목이름	중요도	항목유형	항목구분	설명
dm_hadr_availability_group_states	2	AG (Availability Group)	DMV	로컬 인스턴스에 가용성 복제본이 있는 각 AlwaysOn 가용성 그룹에 대한 정보
dm_hadr_availability_replica_cluster_nodes	2	AG (Availability Group)	DMV	WSFC 클러스터에 있는 AlwaysOn 가용성 그룹의 모든 가용성 복제본에 대한 정보
dm_hadr_availability_replica_cluster_states	2	AG (Availability Group)	DMV	WSFC 클러스터에 있는 모든 AlwaysOn 가용성 그룹의 각 AlwaysOn 가용성 복제본에 대한 정보 (복제본 위치나 조인 상태에 상관없음)
dm_hadr_availability_replica_states	2	AG (Availability Group)	DMV	동일한 AlwaysOn 가용성 그룹의 로컬 복제본으로 각 로컬 가용성 복제본에 대한 행 및 각 원격 가용성 복제본에 대한 행을 반환
dm_hadr_cluster	2	AG (Availability Group)	DMV	클러스터 이름과 쿼럼 정보 반환
dm_hadr_cluster_members	2	AG (Availability Group)	DMV	WSFC 노드에 쿼럼이 있는 경우 쿼럼을 구성하는 각 멤버에 대한 행과 각 멤버의 상태 반환
dm_hadr_database_replica_cluster_states	2	AG (Availability Group)	DMV	AlwaysOn 가용성 그룹에서 가용성 데이터베이스의 상태에 대한 정보
dm_hadr_database_replica_states	1	AG (Availability Group)	DMV	현재 Primary 서버의 Queue에 쌓여있는 용량, Primary에서 Secondary로 보내는 속도, Secondary Queue Redo에 쌓여있는 용량, Secondary Redo반영 속도를 모니터링 할 수 있습니다. 이 값은 신규 인덱스를 생성, 대규모 데이터 수정 등을 수행할 때 필수 모니터링 항목입니다.
dm_hadr_name_id_map	2	AG (Availability Group)	DMV	AlwaysOn 가용성 그룹 매핑 정보
database_principals	3	Config	DMV	데이터베이스에 생성된 로그인 현황입니다. 계정의 생성일, SID, 기본 데이터베이스 등의 정보
dbcc_logininfo	1	Config	DMV	데이터베이스 로그 파일의 VLF(Virtual Log File) 파일 개수나, 현재 활성 로그 파일이 몇 개인지 판단할 수 있습니다. VLF 파일은 100개 미만으로 생성되어 있어야 하고, 이 개수가 과도하게 많으면 해당 SQL Server 시작이 매우 느리거나 백업 복구가 매우 느릴 수 있습니다.

항목이름	중요도	항목유형	항목구분	설명
dbcc_tracestatus	1	Config	DMV	해당 데이터베이스에 활성화된 TraceFlag 정보
dm_os_sys_info	2	Config	DMV	물리 메모리 용량, 커밋 메모리 용량, SQL Server 시작 시간, SQL Server Version 등
dm_server_services	2	Config	DMV	SQL Server, SQL Server Agent, Full-text Daemon 등 해당 서버의 SQL 서비스 시작유형, 현재 상태 Service Account 등
dm_xe_sessions	2	Config	DMV	서버에서 동작중인 모든 확장 이벤트 정보
fn_trace_getinfo	2	Config	DMV	모든 활성 프로파일러 정보
identity_columns	1	Config	DMV	모니터링 하는 데이터베이스의 모든 identity 속성의 최대값과 현재 값
server_principals	2	Config	DMV	서버의 모든 로그인 관련 정보
server_triggers	2	Config	DMV	서버 레벨 트리거의 모든 정보 (ex 로그인 트리거)
sp_configure	1	Config	DMV	서버의 설정 정보
sysaltfiles	2	Config	DMV	데이터베이스 크기, 자동증가 설정
sysdatabases	1	Config	DMV	데이터베이스 상태 값 (로그가 잘리지 않는다면 왜 잘리지 않는지, page verify option, recovery model 등의 정보 등을 알 수 있습니다.)
sysjobs	2	Config	DMV	모든 예약 작업 정보를 알 수 있으며, 활성화 비활성화 정보
backupset	1	Log	DMV	데이터베이스 백업 셋 정보
sp_readerrorlog	1	Log	DMV	SQL Server의 에러로그 정보(Agent 에러 정보도 포함, Express Edition의 경우 ItemOption에서 SqlAgentLog=False를 설정해야 합니다)
sysjobhistory	1	Log	DMV	예약작업 수행 결과
dbcc_memorystatus	3	Performance	DMV	서버의 가장 자세한 수준의 메모리 현황입니다. 서버의 메모리 관련 오류가 있을 때 자세한 정보를 수집 할 수 있습니다.

항목이름	중요도	항목유형	항목구분	설명
dbcc_sqlperf_logspace	1	Performance	DMV	로그파일(LDF 파일)의 사용량을 알 수 있습니다. AG 환경이나 미러링 환경의 경우 DBCC Loginfo의 Status 정보로 남은 공간을 추정 할 수 없습니다. (by design)
dm_db_index_usage_stats	1	Performance	DMV	인덱스 사용 상태 값 (클러스터 비 클러스터, 인덱스 사이즈, 복합 칼럼 순서, 통계 업데이트 일자, Unique 여부, Filtered 인덱스 여부, 활성 비활성 여부, index seek scan 회수 등을 저장합니다. 단 단편화 정보는 수행시간 문제로 수집하지 않았다.
dm_db_missing_index_group_stats	2	Performance	DMV	추가 인덱스 추천 기능으로 제안된 인덱스와 인덱스 생성 시 성능 향상 정도를 보여줍니다.
dm_db_task_space_usage	2	Performance	DMV	TEMPDB 공간 추적용으로 어떤 쿼리에 의해 어느 정도의 공간이 TEMPDB에 할당되어 있는지 보여줍니다.
dm_exec_query_stats2	1	Performance	DMV	쿼리별 1분 단위 통계를 수집합니다. CPU, Reads, Execution Count, Elapsed Time 등의 통계를 구합니다. 또한 연관된 Query String과 쿼리 실행계획을 보관합니다. 1분 통계 테이블 : dm_exec_query_stats2_b_summary 실제 쿼리 테이블 : dm_exec_query_stats2_b_statement 쿼리 플랜 테이블 : dm_exec_query_stats2_b_query_plan 실제 쿼리 통계 기본 데이터 (1시간) : dm_exec_query_stats2_b
dm_io_virtual_file_stats	2	Performance	DMV	데이터 파일 및 로그 파일에 대한 I/O 통계를 반환합니다. 하나의 물리 디스크에 여러 파일이 존재할 경우 어떤 파일에서 가장 많은 I/O가 발생하는지 판단할 주요 수치입니다.
dm_os_memory_clerks	2	Performance	DMV	SQL Server 인스턴스에서 현재 활성 상태의 모든 메모리 클럭을 반환합니다. SQL Server에서 사용하는 대부분의 메모리 할당은 해당 DMV에 등록 후 사용되므로 메모리 사용량을 추적하거나, 과도한 메모리 사용량을 보이는 개체를 확인할 때 주로 사용됩니다.
dm_os_wait_stats	1	Performance	DMV	쿼리 수행이 완료된 대기 시간을 보여줍니다. 누적값으로 기록되며, 과거 값과 현재 값의 차이를 일정 주기로 쿼리하면, 어떤 대기가 많이 발생하는지 알 수 있습니다.
dm_os_workers	1	Performance	DMV	현재 동작중인 쿼리와 상태값을 반환 합니다. SQL Server에서 현재 수행되는 사용자 쿼리와 시스템에서 발생하는 동작도 모두 담도록 디자인 되어 있으며, 현재 동작중인 서버 상태를 가장 자세히 알 수 있습니다. (dm_exec_query_stats2에

항목이름	중 요 도	항목유형	항목구분	설명
				는 동작이 완료된 통계가 있고, dm_os_workers는 현재 수행 중인 쿼리의 리소스 사용량 정보가 있습니다.)
sp_lock2	1	Performance	DMV	SQL Server 잠금 정보를 Tree 형태로 출력하도록 만들어졌으며, 원인 쿼리 분석이 가능합니다. (예: 65 > 70 > 81 라는 LockTree 결과는 65번 spid가 잠금 원인이며, 70번과 81번 spid가 수행 대기[잠금으로 수행이 불가능한 상황]하고 있다는 설명입니다.) 수집주기를 운영 환경에 맞게 적절히 조절할 필요가 있습니다. 기본값은 10초로 되어 있으나, 필요에 따라 그 수집 주기를 짧게 설정하는 것도 좋습니다.
sp_spaceused	1	Performance	DMV	테이블의 행수 및 공간 사용정보를 출력합니다.
suspect_pages	1	Performance	DMV	일관성이 깨진 페이지 정보를 출력합니다. 823, 824 오류를 모니터링 할 수 있습니다. 823, 824 오류는 물리 하드웨어 스토리지 일관성에 문제가 있을 수 있으므로 하드웨어 점검이 필요할 수 있습니다.
Application	1	Log	EVENTLOG	이벤트 로그의 Application 정보를 가져온다. 필터 조건을 추가해 원하는 조건의 데이터만 수집 가능합니다.
Security	3	Log	EVENTLOG	이벤트 로그의 Security 정보를 가져온다. 필터 조건을 추가해 원하는 조건의 데이터만 수집 가능합니다.
Setup	3	Log	EVENTLOG	이벤트 로그의 Setup 정보를 가져온다. 필터 조건을 추가해 원하는 조건의 데이터만 수집 가능합니다.
System	3	Log	EVENTLOG	이벤트 로그의 System 정보를 가져온다. 필터 조건을 추가해 원하는 조건의 데이터만 수집 가능합니다.
BaselineCollect	1	Performacne	PERFMON	Performance Counter 값을 가져온다. 미리 정의된 Performance Counter 값을 가져오며, 미리 정의된 카운터는 설치 폴더 하위의 Resources 폴더에 UpdateCounterDetailsY.sql 값을 기반으로 모니터링 합니다. 모니터링 기본값 변경을 위해 해당 쿼리를 바꾸거나 개별로 SqlLogManManager 데이터베이스의 CounterDetailsAutoUpdated 테이블의 값 IsEnabledYN를 조절합니다. 추가로 필요한 성능 카운터 정보가 있으면, CounterDetailsFilterInfo 테이블의 필터 정보를 수정합니다.

중요도는 LazyLog 개발자가 SQL Server 모니터링 시 주요하게 살펴봐야 할 내용을 1로 했으며, 특수한 문제 해결이나 그 중요도가 낮은 것은 3으로 설정했습니다.

특히 문제가 없다면, 모든 설정을 활성화해 모니터링 합니다. 저장 공간 문제가 있다면 아이템의 수집 주기를 변경하거나 삭제, 한계 시간을 조절하면 됩니다.

# 데이터베이스 스키마

MSSQL 로그 수집기(LazyLog) 데이터베이스 구조

## SqlLogManManager

수집 대상 서버 정보를 저장하는 테이블과, 수집 설정에 관계된 모든 설정이 들어있는 데이터베이스입니다. (이 데이터베이스와 lazylog.exe.config 파일만 백업하면 언제든지 수집 서버를 복구할 수 있습니다.)

테이블 이름	칼럼 이름	설명	CollectItem	CollectItem	자동 증가 값				
ItemType	DMV / EVENTLOG / PERFMON 값을 가질 수 있습니다.			CollectItem					
ItemName	dm_os_workers, dm_os_waitstats, dm_exec_query_stats2 등의 값이 들어 있습니다. 아이템이 DMV인 경우 SqlLogManDmvRepository에 ItemName_b로 저장됩니다. dm_exec_query_stats2 아이템의 경우는 테이블이 추가됩니다. dm_exec_query_stats2_b, dm_exec_query_stats2_b_query_plan, dm_exec_query_stats2_b_statement, dm_exec_query_stats2_b_summary 테이블로 결과가 저장됩니다. 모든 ItemName은 CollectItemTemplate 테이블에 기본값과 함께 저장되어 있습니다.				CollectItem				
ItemDesc	PERF, CONFIG, LOG로 나눈 분류 값이며, 프로그램에서 특별한 동작을 하지 않습니다.					CollectItem			
MachineId	Machine 테이블에 등록된 모니터링 대상 서버의 MachineId 값						CollectItem		
SqlInstanceid	MachineSqlInstance에 저장된							CollectItem	

테이블 이름	칼럼 이름	설명
	SqlInstanceId 값	
DatabaseName	<p>Connection String의 기본 데이터베이스 값입니다. 몇몇 아이템은 master 데이터베이스에 수집이 안 되고 직접 사용자 데이터베이스에 접속해야 합니다. CollectitemTemplate에 master와 UserDatabase로 구분되어 있습니다. UserDatabase로 분류된 아이템은 기본 아이템이 먼저 성공적으로 수집된 후 설정되어야 합니다.</p>	
CollectIntervalSec	수집 주기 초	
IsNtAuthYN		<p>Y / N 값을 가질 수 있으며, NT 인증인지 SQL 인증인지 구분하는 값입니다. ItemType 중 PERFMON과 EVENTLOG는 반드시 IsNtAuthYN이 Y로 설정되어야 합니다. DMV 관련 아이템은 Y/N 값을 가질 수 있습니다. DMV 관련 아이템은 IsNtAuthYN가 Y인 경우 모니터링 대상 서버를 찾을 때 Machine 테이블에 있는 MachineName, MachinePublicIp, NtUserName, NtPassword, NtDomain, MachineSqlInstance 테이블의 PortDefault 정보를 이용해 Windows Impersonation을 시도한 후 접근합니다. 대상 서버가 도메인 환경이면 NtDomain에는 도메인(예: minsouk.com)을 적고 없다면 hostname(예: LocalMachineHostnameA)을 적으면 됩니다. N인 경우 SQL 인증으로 동작하며, Machine의 MachinePublicIp, MachineSqlInstance 테이블의 SqlUsername, SqlPassword, PortDefault 값을 이용해 접근합니다. EVENTLOG와 PERFMON은 MachinePublicIp가 아니라 MachineName을 이용해 접근하</p>

테이블 이름	칼럼 이름	설명	
	로 AD 환경에서는 도메인, 아닌 경우에는 hostname을 windows/system32/drivers/etc/hosts에 등록하고, NtDomain에 기입해야 정상적으로 수집 가능합니다.		
IsEnabledYN	Y / N 값을 가질 수 있으며, N일 경우 해당 아이템은 수집하지 않습니다.		
PurgeIntervalSec	수집 저장된 데이터 삭제 주기 초, 수집 주기와 같게 설정하는 것이 좋습니다.		
PurgeDurationHH	남겨둘 데이터의 한계 시간입니다. 24를 입력하면 24시간이 지난 데이터는 IsPurgeEnabledYN 이 Y인 경우 자동 삭제됩니다.		
IsPurgeEnabledYN	Y / N 값을 가지며 Y인 경우 데이터 삭제 동작을 수행합니다.		
ItemOption	<p>db_exec_query_stats2_b에는 topCnt, SrcDelHH, IsNewDmv, isGatherQueryStatement, isGatherQueryPlan 옵션을 가질 수 있습니다. topCnt는 dm_exec_query_stats를 last_execution_time desc 기준으로 정렬해 top으로 가져올 때 최대값을 기록합니다. 초기값은 20000입니다. SrcDelHH dm_exec_query_stats2_b의 데이터 유지 시간입니다. 기본값은 1이며 1시간입니다. 특수한 경우가 아니면 1시간 이상 유지할 필요가 없습니다. 집계 값은 따로 dm_exec_query_stats2_b_summary에 저장되기 때문입니다.</p> <p>isNewDmv는 SQL Server 2012 SP3 부터 total_dop ~ max_used_threads 값을 수집할 수 있습니다. SP 버전마다 달라 기본값은 false로 설정되어 있습니다.</p>		

테이블 이름	칼럼 이름	설명		
	<p>만약 이 값이 나오는 dm_exec_query_stats를 수집하면 true를 입력하면 됩니다. isGatherQueryStatement는 쿼리문 수집 여부이며, 기본값은 true입니다. isGatherQueryPlan은 플랜 수집 여부이며, 기본값은 true입니다. ad-hoc 쿼리로 동작해 수많은 플랜이 생길 경우 false로 설정합니다. true로 설정되어 있더라도 최소 부하로 플랜을 수집해 온다. 쿼리문과 플랜은 1분에 최대 부하를 일으키는 쿼리 100개를 수집합니다.</p> <p>EVENTLOG Security의 경우 filter 조건을 명시할 수 있습니다. 기본값은 filter   and <code>\\*[System[EventID != '4674']] and \\*[System[EventID != '4624']] and \\*[System[EventID != '4634']] and \\*[System[EventID != '4672']] and \\*[System[EventID != '5447']] and \\*[System[EventID != '4673']] and \\*[System[EventID != '4776']] and \\*[System[EventID != '4670']]</code>입니다.</p> <p>PERFMON BaselineCollect의 경우 counterReload 옵션을 가질 수 있고 기본값은 Y입니다. 서버의 모든 성능 카운터 수집이 완료되면 N으로 자동으로 변경됩니다.</p>			
CollectItemTemplate		<p>수집 기본값을 저장하고 있는 테이블입니다. IsEnabledYN 이 모두 N으로 설정되어 있습니다. isNtAuthYN은 EVENTLOG, PERFMON은 Y DMV 관련 항목은 N으로 등록되어 있습니다. usp_register_CollectItem 저장 프로시저를 이용해 아이템 등록 시 이 값을 참조해 수행합니다.</p>		
Config	Type	PERFMON / EVENTLOG / DMV		

테이블 이름	칼럼 이름	설명
	Name	<p>값을 가질 수 있습니다.</p> <p>아이템 이름 CollectPerServerItemSleepMiliSec : 성능 카운터 아이템 수집 시 각 아이템 간 휴지 시간 (Thread.Sleep 값)입니다. 동일 IDC내 원격 서버의 경우 150을 권고하고, 예를 들어 성능 카운터 수집 간격이 10초이며, 수집 성능카운터 아이템이 10개 라면, CollectPerServerItemSleepMiliSec은 최대 1000 ms까지 줄 수 있을 것입니다. 이 값은 서버의 아이템 수집 항목과 수집주기에 의존됩니다. 수집 간격 사이에 최소 1000 ms까지 설정 가능하지만 900ms 정도까지 설정하는 것이 좋고, 최소값은 25ms 이상 확보해야 합니다. 로컬 서버의 경우 더 짧은 시간으로 설정 가능합니다.</p> <p>CollectIntervalSec : 성능 카운터 수집 주기입니다. 기본값은 15초입니다. 원격서버의 경우 15초 이상의 값을 설정해야 합니다. 서버별로 다른 값을 줄 수 없습니다. 대상서버의 성능 카운터를 추가할 경우 이 값은 변경되어야 합니다.</p> <p>CounterDetailsThreads : 최초 성능 카운터를 수집 할 경우, 각 카운터 정보를 읽어야 하며, 이 값은 CounterDetailsFilterInfo 테이블 설정을 통해 성능 카운터 카테고리를 필터 합니다. 한번에 너무 많은 서버의 성능 카운터를 읽어오면 네트워크 병목을 유발할 수 있으므로 적절하게 설정하면 됩니다. 1로 설정하면 한번에 하나의 서버에서 성능 카운터 정보를 수집합니다. 이 값은 수집 프로그램 내 SemaphoreSlim 클래스의 동시 수행 숫자를 제어함</p>

테이블 이름	칼럼 이름	설명
		<p>니다. 수집 대상 서버를 모두 동시에 수집하길 원한다면 수집 대상 서버 숫자만큼 적어주면 됩니다. 기본값 1을 적용해 순차적으로 수집 되는걸 권장 합니다.</p> <p>PerfmonReset : LazyLog.exe 프로그램 실행 중 항목의 Value 값이 Y로 업데이트 되면, 수 초 후 성능 카운터만 수집을 다시 시작합니다. 전체 수집 프로그램을 재시작하지 않기 위한 옵션이며, LazyLog.exe 프로그램을 종료하고 다시 시작해도 같은 동작을 수행합니다.</p> <p>DmvReset : LazyLog.exe 프로그램 실행 중 항목의 Value 값이 Y로 업데이트 되면, 수 초 후 DMV 만 수집을 다시 시작합니다. 전체 LazyLog 프로그램을 재시작하지 않기 위한 옵션이며, LazyLog.exe 프로그램을 종료하고 다시 시작해도 같은 동작을 수행합니다.</p> <p>EventLogReset LazyLog.exe : 프로그램 실행 중 항목의 Value 값이 Y로 업데이트 되면, 수 초 후 이벤트 로그만 수집을 다시 시작합니다. 전체 LazyLog 프로그램을 재시작하지 않기 위한 옵션이며, LazyLog.exe 프로그램을 종료하고 다시 시작해도 같은 동작을 수행합니다.</p>
	Value	모두 문자 값이나 프로그램에서 적절히 변환해 사용됩니다. (위 Name 칼럼 설명 참조)
CounterDetailsAutoUpdated		성능 카운터 수집을 위해 CounterDetailsFilterInfo 테이블 값 기준으로 서버에서 읽어옵니다. CounterDetailAutoUpdated는 서버에서 읽어온 성능 카운터 정보의 최초 저장 위치이며, 기본으로 IsEnabledYN 이 Y로 설정

테이블 이름	칼럼 이름	설명
		<p>된 카운터들은 lazylog.exe 하위 Resources 폴더에 UpdateCounterDetailsY.sql 값에 의해 설정됩니다. 성능 카운터를 추가하고자 한다면, SqlLogManManager 데이터베이스의 CounterDetailsAutoUpdated 테이블에서 해당 항목의 IsEnabledYN 값을 Y로 수정 후 CollectItem에서 해당 서버의 성능 카운터 항목 ItemOption에 counterReload=Y를 설정합니다. (Y로 설정된 성능 카운터를 N으로 바꾸는 절차도 같습니다.)</p>
<b>CounterDetailsFilterInfo</b>	idx	identity 값
	FilterType	IN / NOTIN 값을 설정할 수 있으며, ObjectName 기준으로 IN은 포함, NOTIN은 수집하지 않습니다.
	ObjectName	성능카운터 ObjectName입니다.
<b>Machine</b>	Machineld	identity 값
	IdcName	사용되지 않습니다
	MachineName	EventLog 수집시 EventLogSession 클래스에서 사용됩니다. PERFMON 수집시 PerformanceCounter 클래스에서 사용됩니다. ActiveDirectory 환경이 아니라면, windows/system32/etc/hosts에 hostname과 IP가 등록되어 있어야 합니다.
	MachinePublicIp	DMV 수집 시 대상서버 접근 IP
	MachinePrivateIp	사용되지 않습니다

테이블 이름	칼럼 이름	설명	
	MachineClusterIp	사용되지 않습니다	
	MachineAglp	사용되지 않습니다	
	NtUserName	Windows Impersonation 인증에서 사용할 계정 이름	
	NtPassword	Windows Impersonation 인증에서 사용할 암호 lazylog.exe “암호”를 수행해 얻은 암호화된 값을 넣어야 합니다. 암호화된 암호는 프로그램 내부에서 정상 암호값으로 DeCrypt 해 사용됩니다.	
	NtDomain	Windows Impersonation 인증에서 사용할 도메인 이름입니다. Active Directory 환경이 아닐 경우 반드시 hostname을 적습니다.	
	NtSessionAuthenticationType	KERBEROS, NEGOTIATE, NTLM, DEFAULT를 지원하며 EventLogSession 클래스에서 사용됩니다. 기본값은 DEFAULT입니다.	
	IsEnabledYN	Y / N을 설정할 수 있으며, N일 경우 해당 서버의 모든 아이템을 수집하지 않습니다. (lazylog.exe 재 시작 혹은 Config 값 조절로 해당 수집을 재 시작해야 합니다.)	
<b>MachineSqlDatabase</b>	Machineld	Machine 테이블의 Machineld 값	
	SqlInstanceld	MachineSqlInstance 테이블의 SqlInstanceld 값	
	DatabaseName	SqlLogManDmvRepository 데이터베이스의 sysdatabases_b에 수집된 데이터베이스 이름	

테이블 이름	칼럼 이름	설명
	DatabaseDesc	사용되지 않습니다
	IsEnabledYN	usp_register_CollectItem 저장프로시저를 이용해 생성할 때 스크립팅 여부를 결정합니다. N은 스크립트 되지 않습니다.
<b>CollectItem</b>	CollectId	사용되지 않습니다

## SqlLogManDmvRepository

DMV 수집 결과 데이터가 저장되는 데이터베이스입니다.

테이블 이름	칼럼 이름	설명
SqlLogManManager 데이터베이스의 CollectItem 테이블에 있는 ItemName에 #_b (베이스라인 의미)를 가진 모든 테이블	idx	identity 값
	MachineName	수집 대상 서버 호스트명
	FullinstanceName	수집 대상 SQL 서버 인스턴스 이름
	probe_time	수집 요청 시간
	etc	각 itemname 명령어 기반으로 유추
dm_exec_query_stats2_b		dm_exec_query_stats2 아이템의 데이터
dm_exec_query_stats2_b_summary		dm_exec_query_stats2_b의 delta 값 (현재 설정 기준으로 1분 기준 delta 값) 이 테이블은 dm_exec_query_stats_b가 3번이상 수집되어야 최초로 과거 값 기반으로 생성됩니다.
dm_exec_query_stats2_b_statement		dm_exec_query_stats2_b_summary에 있는 query_hash에 해당하는 쿼리 문장
dm_exec_query_stats2_b_query_plan		dm_exec_query_stats2_b_summary에 있는 plan_handle에 해당하는 xml plan

## SqlLogManPerfmonRepository

성능 카운터와 이벤트로그를 저장하는 데이터베이스입니다.

테이블 이름	칼럼 이름	설명
--------	-------	----

테이블 이름	칼럼 이름	설명
CounterData		성능 카운터 기본 데이터베이스 스키마를 따름 (수집된 성능 값)
CounterDetails		성능 카운터 기본 데이터베이스 스키마를 따름 (수집된 성능 카운터 종류)
DisplayToID		성능 카운터 기본 데이터베이스 스키마를 따름 (수집 성능카운터 Set 정의이며, lazylog.exe.config의 PerfmonItemName 값 BaselineCollect는 DisplayString으로 사용됩니다).
EventLogApplication		윈도우 이벤트 로그의 Application 정보 (최초 수집 시 30분 이전 로그부터 가져옵니다)
EventLogSecurity		윈도우 이벤트 로그의 Security (최초 수집 시 1분 이전 로그부터 가져옵니다)
EventLogSetup		윈도우 이벤트 로그의 Setup (최초 수집 시 1일 이전 로그부터 가져옵니다)
EventLogSystem		윈도우 이벤트 로그의 System (최초 수집 시 30분 이전 로그부터 가져옵니다)

## 설치

MSSQL 로그 수집기(LazyLog) 설치 가이드

### 프로그램 설치

#### 개요

1. 암호를 평문으로 저장하면, 해킹이나 보안에 문제가 있으므로 암호를 다시 암호화해 사용해야 합니다. [필수]
2. 성능카운터와 이벤트로그를 모니터링 대상 서버에서 읽어오기 위해 대상 서버에서 윈도우 계정설정, 권한설정, 필수 서비스시작, 방화벽 설정이 필요합니다.
3. 모니터링 대상 서버를 대상으로 모니터 서버에서 성능카운터, 이벤트로그, DMV 를 조회할 수 있는지 먼저 확인이 필요합니다
4. 저장소 데이터베이스 3 개를 생성합니다.
5. lazylog.exe.config 파일에는 항목의 글로벌 설정과 수집기 설정이 저장되는 데이터베이스, 수집 결과가 저장되는 데이터베이스 정보가 기록됩니다.
6. SqlLogManManager 데이터베이스에 수집 대상 서버의 접속 정보를 저장합니다.
7. SqlLogManManager 데이터베이스에 서버별 수집 아이템을 등록합니다. (성능카운터 수집여부, 이벤트로그 각 항목 수집여부, DMV 각 항목 수집여부 등)
  - 설정이 복잡해 등록 스크립트를 생성해주는 프로시저가 제공되므로 스크립트를 생성해 SQLCMD 나 SSMS 에서 실행합니다.
8. 기본 수집과 확장 수집이 있는데, 확장 수집의 경우
  - **SqlLogManDmvRepository.dbo.sysdatabases\_b** 수집이 완료되어야 합니다. (기본 아이템 중 sysdatabases 를 수집해야 함)
9. SqlLogManDmvRepository 데이터베이스에는 SQL Server 관련 모니터링 항목이 저장됩니다.
10. SqlLogManPerfmonRepository 데이터베이스에는 성능카운터와 이벤트로그 수집 결과가 저장됩니다.

### 콘솔 모드 설치

LazyLog.zip 파일의 압축을 해제 하면 아래와 같은 파일이 있습니다.

Resources	2017-03-13 오후...	파일 폴더	
SQLScript	2017-03-13 오후...	파일 폴더	
LazyLog.exe	2017-03-13 오후...	응용 프로그램	734KB
LazyLog.exe.config	2017-02-28 오후...	XML Configuratio...	2KB
LazyLogService.zip	2017-03-13 오후...	압축(ZIP) 폴더	919KB
MSSQL Log Collector (LazyLog) 사용자 가이드.docx	2017-03-14 오후...	Microsoft Word ...	869KB

항목	설명
Resources	UpdateCounterDetailsY.sql 파일이 있음
SQLScript	윈도우 명령어 모음과 설치 Step에서 사용하는 SQL 스크립트 파일이 있음
LazyLog.exe	로그수집 프로그램 (콘솔 응용 프로그램)
LazyLog.exe.config	로그수집 설정 파일 (XML 설정파일)
LazyLogService.zip	로그수집 프로그램 (Service 설치용 응용 프로그램 설치 파일)

탐색기 보기 옵션에서 파일 확장명에 체크를 하지 않으면, **LazyLog.exe** 는 **LazyLog**, **LazyLog.exe.config** 는 **LazyLog.exe** 로 보이므로 **LazyLog.exe.config** 와 **LazyLog.exe** 를 혼동하지 않도록 주의합니다.

## 암호 생성

LazyLog 에서 사용하는 암호는 LazyLog 를 통해 생성한 암호화된 문자열을 사용해야 합니다. 암호화된 문자열 생성은 cmd 에서 LazyLog “암호문자열”을 입력하여 생성합니다. 아래는 3 개의 암호를 생성한 예제입니다. 테스트 설정의 경우 모두 같은 암호를 설정하는 것도 좋은 방법입니다. 그러나, 실제 모니터링 환경에서는 서버별 암호가 모두 다를 수 있습니다. 테스트의 경우 암호도 가이드와 같게 생성해 실습하면 더욱 쉽게 설정 할 수 있습니다.

```

관리자: C:\WINDOWS\system32\cmd.exe

C:\Users\Naver\Documents\Visual Studio 2015\Projects\conQueue\conQueue\bin\x64\Release>lazylog "P@ssw0rd1"
암호 생성에 사용된 문자 : P@ssw0rd1
생성된 암호 문자 : BaQJi8Br/qBUv01q/F1Z1Q==

C:\Users\Naver\Documents\Visual Studio 2015\Projects\conQueue\conQueue\bin\x64\Release>lazylog "P@ssw0rd2"
암호 생성에 사용된 문자 : P@ssw0rd2
생성된 암호 문자 : Xck00Pvn7Lq0GjFVAjehhg==

C:\Users\Naver\Documents\Visual Studio 2015\Projects\conQueue\conQueue\bin\x64\Release>lazylog "P@ssw0rd3"
암호 생성에 사용된 문자 : P@ssw0rd3
생성된 암호 문자 : qcWw7DCqnN7vqMZ6b9ilzw==

```

## 모니터링 대상 서버 설정

### 1. 서버 설정

lazyNtUser 는 SqlLogManManager 데이터베이스에 설정될 윈도우 가장 인증 계정입니다. 아래는 모니터링 대상 서버에서 실행해야 할 명령어이며, 관리자 모드로 실행되어야 합니다. 모니터링 저장소 서버와 모니터링 대상 서버가 같을 경우 아래 붉은색 명령어는 수행하지 않아도 됩니다.

#### ○ 윈도우 CMD 명령어 [클라이언트]

- sc config remoteregistry start=auto
- net start remoteregistry
- netsh advfirewall firewall set rule name="Netlogon 서비스 (NP-In)" new enable=yes
- net user lazyNtUser P@ssw0rd2 /ADD
- net localgroup "Performance Log Users" lazyNtUser /add
- net localgroup "Performance Monitor Users" lazyNtUser /add
- net localgroup "Event Log Readers" lazyNtUser /add

## 2. SQL Server 설정

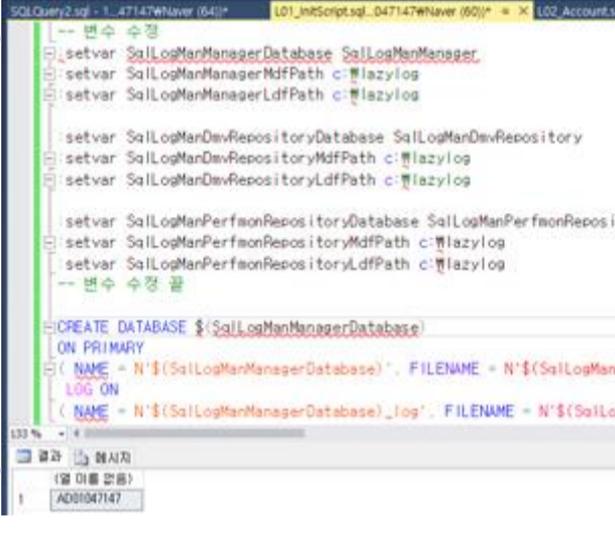
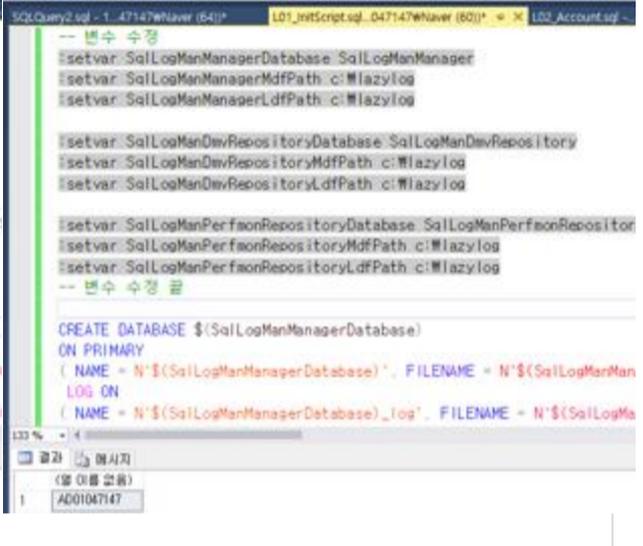
lazyClient 라는 계정은 SqlLogManManager 데이터베이스에 설정될 계정 정보입니다. SQL Server 관리자로 아래 명령을 SQLCMD 모드로 실행합니다. SQLScript 하위 폴더에 L03\_Account\_lazyClient.sql 파일을 엽니다

### ○ SQLCMD 명령어 [클라이언트]

- -- 변수 수정
- :setvar AccountName lazyClient
- :setvar AccountPassword P@ssw0rd1
- -- 변수 수정 끝

### ○ lazyClient SQL 인증 계정이 P@ssw0rd1 라는 암호로 생성됩니다.

본 가이드의 모든 SQL 쿼리문은 SQLCMD 모드에서 동작시킨다. SSMS 에서 SQLCMD 모드로 동작 시키기 위해 아래 설명을 참고합니다.

		
SQLCMD 모드 변환 방법	일반 모드	SQLCMD 모드 (변수 부분이 반전되어 있음)

## 모니터링 저장소 데이터베이스 생성

### 1. 저장소 데이터베이스가 생성될 폴더를 생성합니다.

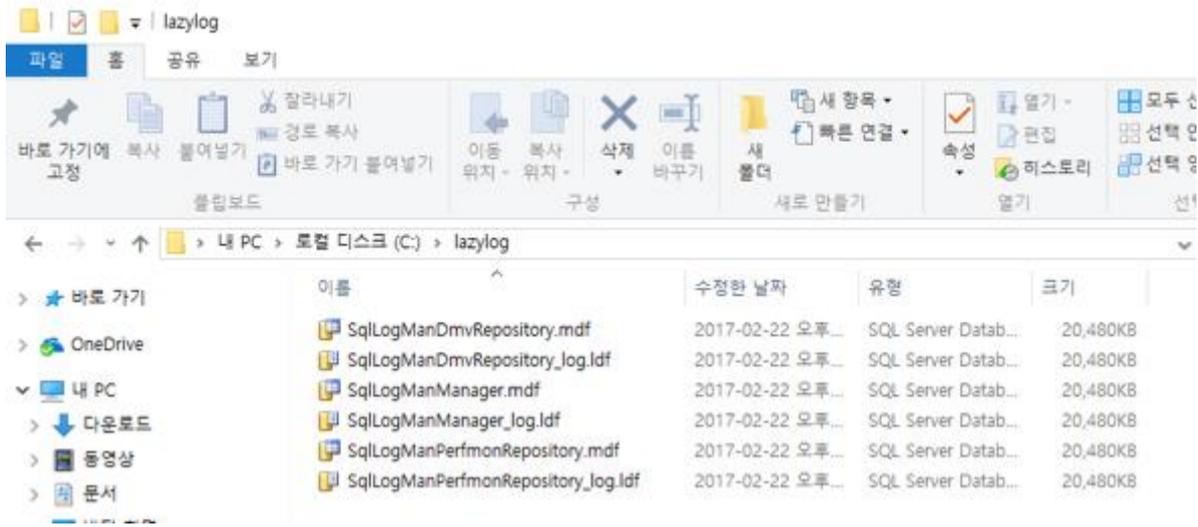
- 윈도우 CMD 명령어
- mkdir c:\lazylog  
폴더가 생성되면 됩니다. (데이터베이스 이름은 자유롭게 변경 가능합니다. 나중에 해당 폴더에 데이터베이스가 생성됩니다.) 처음 실습에서는 그대로 유지합니다.

### 2. SQLScript 하위 폴더에 L01\_InitScript.sql 파일을 엽니다.

### ○ SQLCMD 명령어

- -- 변수 수정
- :setvar SqlLogManManagerDatabase SqlLogManManager
- :setvar SqlLogManManagerMdfPath c:\lazylog
- :setvar SqlLogManManagerLdfPath c:\lazylog
- :setvar SqlLogManDmvRepositoryDatabase SqlLogManDmvRepository
- :setvar SqlLogManDmvRepositoryMdfPath c:\lazylog
- :setvar SqlLogManDmvRepositoryLdfPath c:\lazylog
- :setvar SqlLogManPerfmonRepositoryDatabase SqlLogManPerfmonRepository
- :setvar SqlLogManPerfmonRepositoryMdfPath c:\lazylog
- :setvar SqlLogManPerfmonRepositoryLdfPath c:\lazylog
- -- 변수 수정 끝

이 스크립트는 SqlLogManManager, SqlLogManDmvRepository, SqlLogManPerfmonRepository 3 개의 데이터베이스를 만드는 스크립트입니다. 변수 수정 부분을 확인하고 SQLCMD 모드에서 실행합니다. 아래와 같이 데이터베이스 3 개가 생성됩니다.



## 모니터링 저장소용 계정 생성

위 단계에서 만든 데이터베이스에 접근할 계정을 만들어 줍니다.

1. SQLScript 하위 폴더에 L02\_Account\_lazyServer.sql 파일을 엽니다.

- o SQLCMD 명령어

- o -- 변수 수정
- o :setvar AccountName lazyServer
- o :setvar AccountPassword P@ssw0rd3
- o -- 변수 수정 끝

2. 변수를 수정하고 SQLCMD 모드로 실행합니다.

lazylog 응용 프로그램이 구동하는 곳에서 ip, port, 계정으로 SSMS 로 모니터링 저장소 서버에 로그인 이 되는지 반드시 테스트 합니다. (SSMS 로 로그인할 경우는 원래의 암호 P@ssw0rd3 로 접근) GUI 로 설정할 경우, 옵션에 따라 암호를 변경해줘야 로그인되는 경우가 있습니다. lazyServer 의 암호 P@ssw0rd3 는 Lazylog.exe.config 에 넣을 때는 qcWa7DCqnN7vqMZ6b9ilzw==로 암호화된 값을 입력합니다.

3. 생성한 계정과 IP 패스워드로 LazyLog.exe 파일이 구동되는 서버에서 저장소 데이터베이스 서버에 IP 와 Port 로 접근 가능한지 테스트합니다.

## LazyLog.exe.config 설정

이 파일은 LazyLog 응용 프로그램이 위 단계에서 생성한 데이터베이스에 접근할 정보를 기입해 주는 단계입니다.

notepad 로 수정하면 됩니다. IP, Port, 계정 이름, 암호화된 계정 비밀번호를 적절히 수정합니다. **127.0.0.1,1433** 으로 접근할 경우 **SSMS** 로 접근이 되는지 테스트 후 설정 합니다.

```
<appSettings>
  <add key="FileLogYN" value="Y" />
  <add key="UsePerfmonYN" value="Y" />
  <add key="UseEventmonYN" value="Y" />
  <add key="UseSqlmonYN" value="Y" />
  <add key="UseLocalKey" value="Y" />
  <add key="NClavisKey" value="" />

  <add key="PerfmonItemName" value="BaselineCollect" />

  <add key="SqlLogManManagerInstanceName" value="AD01047147" />
  <add key="SqlLogManManagerIP" value="127.0.0.1" />
  <add key="SqlLogManManagerPort" value="1433" />
  <add key="SqlLogManManagerDatabase" value="SqlLogManManager" />
  <add key="SqlLogManManagerIsNtAuthYN" value="N" />
  <add key="SqlLogManManagerUserName" value="lazyServer" />
  <add key="SqlLogManManagerPassword" value="qcWa7DCqnN7vqMZ6b9ilzw==" />

  <add key="SqlLogManDmvRepositoryInstanceName" value="AD01047147" />
  <add key="SqlLogManDmvRepositoryIP" value="127.0.0.1" />
  <add key="SqlLogManDmvRepositoryPort" value="1433" />
  <add key="SqlLogManDmvRepositoryDatabase" value="SqlLogManDmvRepository" />
  <add key="SqlLogManDmvRepositoryIsNtAuthYN" value="N" />
  <add key="SqlLogManDmvRepositoryUserName" value="lazyServer" />
  <add key="SqlLogManDmvRepositoryPassword" value="qcWa7DCqnN7vqMZ6b9ilzw==" />

  <add key="SqlLogManPerfmonRepositoryInstanceName" value="AD01047147" />
  <add key="SqlLogManPerfmonRepositoryIP" value="127.0.0.1" />
  <add key="SqlLogManPerfmonRepositoryPort" value="1433" />
  <add key="SqlLogManPerfmonRepositoryDatabase" value="SqlLogManPerfmonRepository" />
  <add key="SqlLogManPerfmonRepositoryIsNtAuthYN" value="N" />
```

```

<add key="SqlLogManPerfmonRepositoryUserName" value="lazyServer" />
<add key="SqlLogManPerfmonRepositoryPassword" value="qcWa7DCqnN7vqMZ6b9ilzw==" />
<add key="ClientSettingsProvider.ServiceUri" value="" />
</appSettings>

```

항목	설명
FileLogYN	LazyLog 응용 프로그램의 에러 로그를 남길지 여부, Y일 경우 설치 폴더에 Log 라는 폴더가 생기고 일자별 아이템별 로그가 생성됩니다. 수집이 정상화되면 N으로 바꾸고 프로그램을 재 시작해 로그파일이 서버에 쌓이지 않도록 합니다.
UsePerfmonYN	LazyLog 응용 프로그램이 성능카운터 수집 여부를 결정합니다. Y일 경우 성능카운터가 수집되며, N일 경우 SqlLogManManager.dbo.CollectItem에 Y로 설정되더라도 어떤 아이템도 수집되지 않습니다.
UseEventmonYN	LazyLog 응용 프로그램이 이벤트로그 수집 여부를 결정합니다. Y일 경우 이벤트로그가 수집되며, N일 경우 SqlLogManManager.dbo.CollectItem에 Y로 설정되더라도 어떤 아이템도 수집되지 않습니다.
UseSqlmonYN	LazyLog 응용 프로그램이 DMV 수집 여부를 결정합니다. Y일 경우 DMV가 수집되며, N일 경우 SqlLogManManager.dbo.CollectItem에 Y로 설정되더라도 어떤 아이템도 수집되지 않습니다.
UseLocalKey	반드시 Y 이어야 합니다.
PerfmonItemName	성능 카운터 저장 데이터베이스(SqlLogManPerfmonRepository) DisplayToID 테이블의 DisplayString 이름입니다. 해당 이름으로 성능 카운터 저장소가 구성하고, 윈도우 성능 카운터로 다시 그림을 그려볼 때 조회할 이름으로 사용됩니다. 기본값은 BaselineCollect가 사용됩니다.
SqlLogManManagerInstanceName	수집 설정 저장소 인스턴스 이름. (프로그램에서 사용되지 않습니다.)
SqlLogManManagerIP	수집 설정 저장소 IP
SqlLogManManagerPort	수집 설정 저장소 SQL Service Port
SqlLogManManagerDatabase	수집 설정 데이터베이스 이름.
SqlLogManManagerIsNtAuthYN	윈도우 인증의 경우 Y, SQL 인증의 경우 N을 입력합니다. (N을 권고합니다.)
SqlLogManManagerUserName	수집 설정 데이터베이스의 계정 이름
SqlLogManManagerPassword	수집 설정 데이터베이스의 암호 (실제 데이터베이스에 접근하는 암호가 P@ssw0rd3 이라면 LazyLog.exe를 이용해 생성한 암호 값 qcWa7DCqnN7vqMZ6b9ilzw==를 입력해야 합니다. 암호화 관련 오류의 대부분은 계정별 암호 값을 잘못 적는 경우이거나 암호가 너무 긴 경우입니다.)

- 나머지 SqlLogManDmvRepositoryXXXX 와 SqlLogManPerfmonRepositoryXXXX 는 DMV 결과를 저장할 데이터베이스와 성능카운터와 이벤트 로그를 저장할 데이터베이스 정보입니다.

## Hosts 파일 추가

모니터링 서버에서 모니터 대상 서버로 이름으로 접근 불가능 할 경우 windows/system32/drivers/etc/hosts 파일에 추가해 이름으로 접근 가능하게 합니다. 워크그룹 환경에서 호스트네임 추가는 성능카운터와 이벤트로그 수집에 필수입니다.

## 모니터 서버에 대상 서버 정보 입력

모니터링 대상 서버를 등록하는 과정입니다. SQLScript 하위 폴더에서 L04\_TargerServerRegister.sql 을 열고 아래 변수를 수정하고 SQLCMD 모드로 실행합니다.

- SQLCMD 명령어

```
-- 변수 수정

:setvar ManagerDatabaseName SqlLogManManager
-- 설정이 저장될 데이터베이스 이름
:setvar IdcName NCloud
-- 서버위치 (프로그램에 영향 없음)
:setvar MachineName AD01047147
-- 모니터링 대상 서버 호스트 이름 (원격일 경우 hosts 에 등록해 접근 가능하게 해둘것)
:setvar MachinePublicIp 127.0.0.1
-- 모니터링 대상 아이피
:setvar lazyNtUserName lazyNtUser
-- 성능모니터와 이벤트로그를 읽을 대상 서버에 생성한 윈도우 인증
:setvar lazyNtUserPasswordEncrypted Xck00Pvn7Lq0GjFVAjehhg==
-- 위 계정의 암호 (lazylog.exe 로 암호화해야 동작합니다. lazylog.exe "P@ssw0rd2"와 같이 생성)
:setvar NtDomainNameIfNotAdEnvHostname AD01047147
-- AD 환경이면 호스트네임을 제외한 도메인 이름 hostname.adServer.com -> adServer.com 만 입력, workgroup 일 경우
hostname 입력
:setvar FullInstanceName AD01047147
-- 모니터링 대상 SQL 인스턴스 이름 (해당 서버에서 syssservers 의 0 번 이름)
:setvar SqlUsername lazyClient
-- 모니터링 대상 SQL 인스턴스에 설정된 SQL admin 계정
:setvar SqlPassword BaQJi8Br/qBUv0lq/F1Z1Q==
-- 위 계정의 암호 (lazylog.exe 로 암호화해야 동작합니다. lazylog.exe "P@ssw0rd1"와 같이 생성)
:setvar RoleName 테스트서버
-- 서버 메모 (프로그램에 영향 없음)
:setvar SqlServerMajorVersion 2014
-- SQL 서버의 주 버전 2008, 2012, 2014, 2016 중 한개의 값
:setvar SqlPort 1433
-- 모니터링 대상 서버의 포트

-- 변수 수정 끝
```

## 기본 모니터링 등록

SQLScript 하위 폴더에서 L05\_TargetItemRegisterBasic\_01.sql 파일을 열고 아래 변수를 수정하고 SQLCMD 모드로 실행합니다.

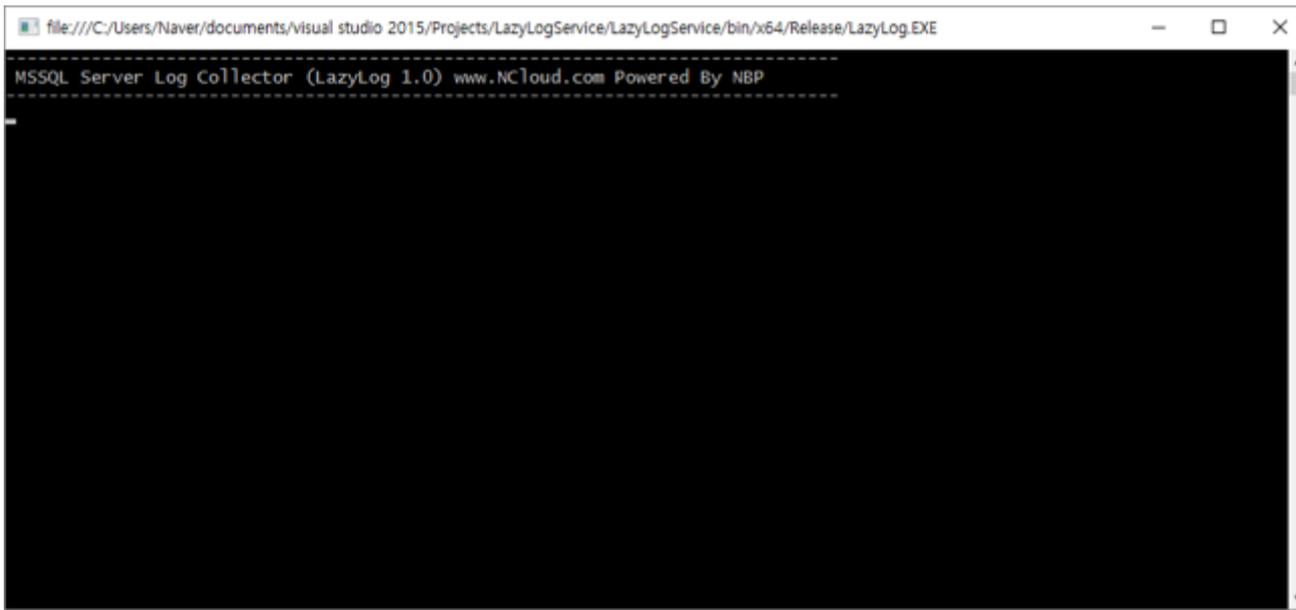
```
-- 변수 수정

:setvar machineName AD01047147
-- MachineName 은 hostname
:setvar FullInstancename AD01047147
--FullInstanceName 은 해당 서버의 select srvname from syssservers where srvid = 0 결과값입니다
:setvar SqlLogManManagerDatabase SqlLogManManager
-- lazylog 모든 설정이 저장된 매니저 데이터베이스 이름 (모니터링 저장소 데이터베이스)

-- 변수 수정 끝
```

## LazyLog.exe 실행

윈도우 탐색기에서 LazyLog.exe 를 더블클릭해 실행합니다. 아래는 실행중인 콘솔 모드 LazyLog.exe 가 정상 실행되는 모습입니다.



## 기본 모니터링 정상 수집 확인

### 1. 성능카운터

정상 수집이 되는지 확인하기 위해 LazyLog 가 구동되고 난 후 약 2 분 후 아래 테이블을 확인합니다.

#### ○ SQLCMD 명령어

- `select * from SqlLogManPerfmonRepository.[dbo].[DisplayToID]`
- `select * from SqlLogManPerfmonRepository.[dbo].[CounterDetails]`
- `select * from SqlLogManPerfmonRepository.[dbo].[CounterData]`

아래와 같이 출력되면 정상수집이 되고 있습니다.

```

select * from SqlLogManPerfmonRepository.[dbo].[DisplayToID]
select * from SqlLogManPerfmonRepository.[dbo].[CounterDetails]
select * from SqlLogManPerfmonRepository.[dbo].[CounterData]
  
```

GUID	RunID	DisplayString	LogStartTime	LogStopTime	NumberOfRecords	MinutesToUTC	TimeZoneName
1	0	BaselineCollect	2017-02-24 13:46:56	2017-02-24 14:55:55	151	-540	대한민국 표준시

CounterID	MachineName	ObjectName	CounterName	CounterType	DefaultScale	InstanceName	InstanceIndex	ParentName	ParentObjectID	TimeBaseA	TimeBaseB
1	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	_Total	0		0	3312637	0
2	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	C:	0		0	3312637	0
3	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	D:	0		0	3312637	0
4	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	E:	0		0	3312637	0
5	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	F:	0		0	3312637	0
6	WWWAD01047147	LogicalDisk	% Free Space	537003008	0	HarddiskVolume1	0		0	3312637	0

GUID	CounterID	RecordIndex	CounterDateTime	CounterValue	FirstValueA	FirstValueB	SecondValueA	SecondValueB	MultiCount
1	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:46:56.769	46,4411010742188	1463627	0	3151577	0	1
2	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:47:11.803	46,4411010742188	1463627	0	3151577	0	1
3	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:47:26.834	46,4411010742188	1463627	0	3151577	0	1
4	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:47:41.899	46,4411010742188	1463627	0	3151577	0	1
5	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:47:56.980	46,4411010742188	1463627	0	3151577	0	1
6	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:48:11.993	46,4411010742188	1463627	0	3151577	0	1
7	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:48:27.010	46,4411010742188	1463627	0	3151577	0	1
8	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:48:42.097	46,4411010742188	1463627	0	3151577	0	1
9	EC814398-89A5-4849-8CBB-E4203549FC42	1	2017-02-24 13:48:57.159	46,4411010742188	1463627	0	3151577	0	1

### 2. 이벤트로그

정상 수집이 되는지 확인하기 위해 LazyLog 가 구동되고 난 후 약 2 분 후 아래 테이블을 확인합니다.

3. `select * from SqlLogManPerfmonRepository.[dbo].[EventLogApplication]`
4. `select * from SqlLogManPerfmonRepository.[dbo].[EventLogSecurity]`
5. `select * from SqlLogManPerfmonRepository.[dbo].[EventLogSetup]`
6. `select * from SqlLogManPerfmonRepository.[dbo].[EventLogSystem]`

아래와 같이 출력되면 정상수집이 되고 있습니다.

```

select * from SqlLogManPerfmonRepository.dbo.[EventLogApplication]
select * from SqlLogManPerfmonRepository.dbo.[EventLogSecurity]
select * from SqlLogManPerfmonRepository.dbo.[EventLogSetup]
select * from SqlLogManPerfmonRepository.dbo.[EventLogSystem]

```

idx	MachineName	probe_time	level	timeCreated	providerName	eventId	recordId	task	xmlLog	isCheckedYN
1	AD01047147	2017-02-24 13:46:21.457	4	2017-02-24 13:18:29.967	MSSQLSERVER	5084	112613	2	<Event xmlns='http://schemas.microsoft.com/win/2...	NULL
2	AD01047147	2017-02-24 13:46:21.457	4	2017-02-24 13:18:33.100	MSSQLSERVER	5084	112614	2	<Event xmlns='http://schemas.microsoft.com/win/2...	NULL
3	AD01047147	2017-02-24 13:46:21.457	4	2017-02-24 13:18:33.153	MSSQLSERVER	5084	112615	2	<Event xmlns='http://schemas.microsoft.com/win/2...	NULL
4	AD01047147	2017-02-24 13:46:21.457	0	2017-02-24 13:18:45.117	MSSQLSERVER	18456	112616	4	<Event xmlns='http://schemas.microsoft.com/win/2...	NULL
5	AD01047147	2017-02-24 13:46:21.457	4	2017-02-24 13:19:17.967	Symantec AntiVirus	2	112617	0	<Event xmlns='http://schemas.microsoft.com/win/2...	NULL
6	AD01047147	2017-02-24 13:46:21.457	0	2017-02-24 13:19:21.337	MSSQLSERVER	18456	112618	4	<Event xmlns='http://schemas.microsoft.com/win/2...	NULL
7	AD01047147	2017-02-24 13:46:21.457	0	2017-02-24 13:19:31.373	MSSQLSERVER	18456	112619	4	<Event xmlns='http://schemas.microsoft.com/win/2...	NULL

## DMV 확인

기본 모니터링이 정상적으로 수행되는지 확인하기 위해 `SqlLogManDmvRepository.dbo.sysdatabases_b` 테이블 내용을 확인합니다. 이 내용이 정상적으로 수집되어야 확장 모니터링을 등록할 수 있습니다.

- SQLCMD 명령어

```
select * from SqlLogManDmvRepository.dbo.sysdatabases_b order by probe_time desc
```

아래와 같이 사용자 데이터베이스 리스트가 보이면 정상 수집이 완료된 것입니다.

```

select * from SqlLogManDmvRepository.dbo.sysdatabases_b order by probe_time desc

```

idx	MachineName	FullInstanceName	probe_time	name	database_id	source_database_id	owner_sid	create_date	compatibili
1	AD01047147	AD01047147	2017-03-14 15:14:10.333	SqlLogManPerfmonRepository	20	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2017-03-13 11:58:16.117	120
2	AD01047147	AD01047147	2017-03-14 15:14:10.333	SqlLogManDmvRepository	19	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2017-03-13 11:58:15.543	120
3	AD01047147	AD01047147	2017-03-14 15:14:10.333	SqlLogManManager	18	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2017-03-13 11:58:15.187	120
4	AD01047147	AD01047147	2017-03-14 15:14:10.333	NaverODay	17	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-10-12 16:34:16.620	100
5	AD01047147	AD01047147	2017-03-14 15:14:10.333	remoteBackupProjectFull	16	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-06-30 17:53:31.140	120
6	AD01047147	AD01047147	2017-03-14 15:14:10.333	backupDashboard	15	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-06-27 18:16:36.920	120
7	AD01047147	AD01047147	2017-03-14 15:14:10.333	indexScriptDb	14	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-05-31 10:10:38.550	120
8	AD01047147	AD01047147	2017-03-14 15:14:10.333	remoteBackupProject	13	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-04-28 10:38:16.783	120
9	AD01047147	AD01047147	2017-03-14 15:14:10.333	TEMPDB_TEST	12	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-04-12 14:13:15.873	120
10	AD01047147	AD01047147	2017-03-14 15:14:10.333	poc	11	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-04-01 09:54:39.417	120
11	AD01047147	AD01047147	2017-03-14 15:14:10.333	config	10	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-03-31 14:03:42.810	120
12	AD01047147	AD01047147	2017-03-14 15:14:10.333	clrtst	9	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-03-30 11:36:21.970	120
13	AD01047147	AD01047147	2017-03-14 15:14:10.333	XEventExcel	8	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-07-19 06:53:07.250	110
14	AD01047147	AD01047147	2017-03-14 15:14:10.333	ftsService	7	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-03-10 11:52:11.557	120
15	AD01047147	AD01047147	2017-03-14 15:14:10.333	adventureworks2012dw	6	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-02-02 20:26:16.587	110
16	AD01047147	AD01047147	2017-03-14 15:14:10.333	SmartData	5	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-02-02 18:56:16.270	120
17	AD01047147	AD01047147	2017-03-14 15:14:10.333	msdb	4	NULL	0x01	2014-02-20 20:49:38.857	120
18	AD01047147	AD01047147	2017-03-14 15:14:10.333	model	3	NULL	0x01	2003-04-08 09:13:36.390	120
19	AD01047147	AD01047147	2017-03-14 15:14:10.333	tempdb	2	NULL	0x01	2017-03-09 16:18:11.530	120
20	AD01047147	AD01047147	2017-03-14 15:14:10.333	master	1	NULL	0x01	2003-04-08 09:13:36.390	120
21	AD01047147	AD01047147	2017-03-14 09:14:09.903	SqlLogManPerfmonRepository	20	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2017-03-13 11:58:16.117	120
22	AD01047147	AD01047147	2017-03-14 09:14:09.903	SqlLogManDmvRepository	19	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2017-03-13 11:58:15.543	120
23	AD01047147	AD01047147	2017-03-14 09:14:09.903	SqlLogManManager	18	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2017-03-13 11:58:15.187	120
24	AD01047147	AD01047147	2017-03-14 09:14:09.903	NaverODay	17	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-10-12 16:34:16.620	100
25	AD01047147	AD01047147	2017-03-14 09:14:09.903	remoteBackupProjectFull	16	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-06-30 17:53:31.140	120
26	AD01047147	AD01047147	2017-03-14 09:14:09.903	backupDashboard	15	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-06-27 18:16:36.920	120
27	AD01047147	AD01047147	2017-03-14 09:14:09.903	indexScriptDb	14	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-05-31 10:10:38.550	120
28	AD01047147	AD01047147	2017-03-14 09:14:09.903	remoteBackupProject	13	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-04-28 10:38:16.783	120
29	AD01047147	AD01047147	2017-03-14 09:14:09.903	TEMPDB_TEST	12	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-04-12 14:13:15.873	120
30	AD01047147	AD01047147	2017-03-14 09:14:09.903	poc	11	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-04-01 09:54:39.417	120
31	AD01047147	AD01047147	2017-03-14 09:14:09.903	config	10	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-03-31 14:03:42.810	120
32	AD01047147	AD01047147	2017-03-14 09:14:09.903	clrtst	9	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-03-30 11:36:21.970	120
33	AD01047147	AD01047147	2017-03-14 09:14:09.903	XEventExcel	8	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-07-19 06:53:07.250	110
34	AD01047147	AD01047147	2017-03-14 09:14:09.903	ftsService	7	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-03-10 11:52:11.557	120
35	AD01047147	AD01047147	2017-03-14 09:14:09.903	adventureworks2012dw	6	NULL	0x0105000000000005150000060C9B9437B2C0ED8E0936F...	2016-02-02 20:26:16.587	110

## 확장 모니터링 등록

1. 사용자 데이터베이스 정보 확인 및 복사

확장 모니터링은 기본 모니터링 항목 중 바로 위에서 확인한 `sysdatabase` 가 정상 수집 되어야 수집 가능합니다.

SQLScript 하위 폴더에서 `L06_TargetItemRegisterAdvanced_01.sql` 파일을 열고 아래 변수를 수정 후 SQLCMD 모드로 실행하면, 사용자 데이터베이스 정보를 `SqlLogManManager.dbo.MachineSqlDatabase` 에 입력하게 됩니다.

- o SQLCMD 명령어

- o -- 변수 수정

- o :setvar machineName AD01047147

- o -- 모니터링 대상 hostname
- o :setvar SqlLogManManagerDatabase SqlLogManManager
- o -- lazylog 모든 설정이 저장된 매니저 데이터베이스 이름 (모니터링 저장소 데이터베이스)
- o :setvar SqlLogManDmvRepositoryDatabase SqlLogManDmvRepository
- o -- DMV 수집 정보가 저장될 데이터베이스 이름
- o -- 기본 아이템 수집에서 sysdatabases 가 수집되어야 정상 등록이 가능합니다.
- o -- 변수 수정 끝

정상적으로 수행되면 아래와 같은 출력이 나옵니다.

```

select @@servername

-----
-- 확장 아이템 등록을 위한 데이터베이스 정보 복사
-- 모니터링 저장소 데이터베이스에서 실행
-----

-- 변수 수정
:setvar machineName AD01047147
-- 모니터링 대상 hostname
:setvar SqlLogManManagerDatabase SqlLogManManager
-- lazylog 모든 설정이 저장된 매니저 데이터베이스 이름 (모니터링 저장소 데이터베이스)
:setvar SqlLogManDmvRepositoryDatabase SqlLogManDmvRepository
-- DMV 수집 정보가 저장될 데이터베이스 이름
-- 기본 아이템 수집에서 sysdatabases 가 수집되어야 정상 등록이 가능하다.

--select * from MachineSqlDatabase
--select * from machine
-- 변수 수정 끝
  
```

	MachineName	FullInstanceName	MachineId	SqlInstanceld	DatabaseName	DatabaseDesc	IsEnabledYN
1	AD01047147	AD01047147	1	1	master		Y
2	AD01047147	AD01047147	1	1	tempdb		N
3	AD01047147	AD01047147	1	1	model		N
4	AD01047147	AD01047147	1	1	msdb		Y
5	AD01047147	AD01047147	1	1	SmartData		Y
6	AD01047147	AD01047147	1	1	adventureworks2012dw		Y
7	AD01047147	AD01047147	1	1	ftsService		Y
8	AD01047147	AD01047147	1	1	XEventExcel		Y
9	AD01047147	AD01047147	1	1	clrttest		Y
10	AD01047147	AD01047147	1	1	config		Y
11	AD01047147	AD01047147	1	1	poc		Y
12	AD01047147	AD01047147	1	1	TEMPDB_TEST		Y
13	AD01047147	AD01047147	1	1	remoteBackupProject		Y
14	AD01047147	AD01047147	1	1	indexScriptDb		Y
15	AD01047147	AD01047147	1	1	backupDashboard		Y
16	AD01047147	AD01047147	1	1	remoteBackupProjec...		Y
17	AD01047147	AD01047147	1	1	NaverODay		Y
18	AD01047147	AD01047147	1	1	SqlLogManManager		Y
19	AD01047147	AD01047147	1	1	SqlLogManDmvRepo...		Y
20	AD01047147	AD01047147	1	1	SqlLogManPerfmonR...		Y

## 2. 확장 모니터링 등록 스크립트 생성

SQLScript 하위 폴더에서 L06\_TargetItemRegisterAdvanced\_02.sql 파일을 열고 아래 변수를 수정 후 SQLCMD 모드로 실행합니다.

### o SQLCMD 명령어

- o -- 변수 수정
- o :setvar machineName AD01047147
- o -- 모니터링 대상 hostname
- o :setvar SqlLogManManagerDatabase SqlLogManManager
- o -- lazylog 모든 설정이 저장된 매니저 데이터베이스 이름 (모니터링 저장소 데이터베이스)
- o -- 변수 수정 끝

정상 수행이 되면, 실행할 스크립트가 출력됩니다.

```

-- 확장 아이템 등록을 위한 스크립트 생성
-- 변수 수정
:setvar machineName AD01047147
-- 모니터링 대상 hostname
:setvar SqlLogManManagerDatabase SqlLogManManager
-- lazylog 모든 설정이 저장된 매니저 데이터베이스 이름 (모니터링 저장소 데이터베이스)
-- 변수 수정 끝
    
```

121 %

결과 메시지

실행할 스크립트

```

1 exec SqlLogManManager.dbo.usp_register_CollectIte...
2 exec SqlLogManManager.dbo.usp_register_CollectIte...
3 exec SqlLogManManager.dbo.usp_register_CollectIte...
4 exec SqlLogManManager.dbo.usp_register_CollectIte...
5 exec SqlLogManManager.dbo.usp_register_CollectIte...
6 exec SqlLogManManager.dbo.usp_register_CollectIte...
7 exec SqlLogManManager.dbo.usp_register_CollectIte...
8 exec SqlLogManManager.dbo.usp_register_CollectIte...
9 exec SqlLogManManager.dbo.usp_register_CollectIte...
10 exec SqlLogManManager.dbo.usp_register_CollectIte...
11 exec SqlLogManManager.dbo.usp_register_CollectIte...
12 exec SqlLogManManager.dbo.usp_register_CollectIte...
13 exec SqlLogManManager.dbo.usp_register_CollectIte...
14
    
```

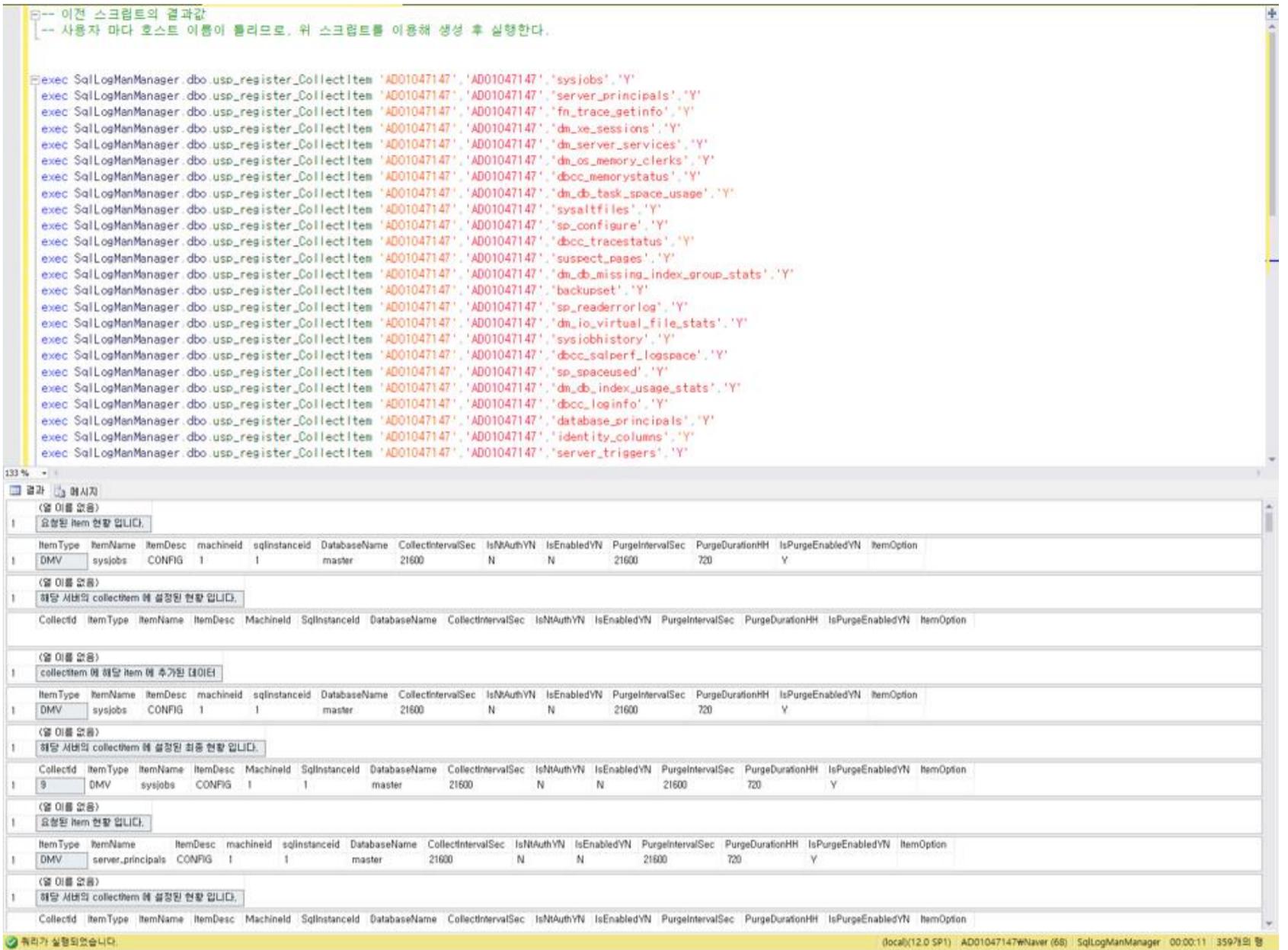
실행할 스크립트

```

1 exec SqlLogManManager.dbo.usp_register_CollectIte...
2 exec SqlLogManManager.dbo.usp_register_CollectIte...
3 exec SqlLogManManager.dbo.usp_register_CollectIte...
    
```

	MachineId	IdcName	MachineName	MachinePublicIp	MachinePrivateIp	MachineClusterIp	MachineAgIp	NtUserName	NtPassword	NtDomain	NtSessionAuth
1	1	NCloud	AD01047147	127.0.0.1				lazyNtUser	Xck00Pvn7Lq0GjFVAjehhg==	AD01047147	Default
2	1	NCloud	AD01047147	127.0.0.1				lazyNtUser	Xck00Pvn7Lq0GjFVAjehhg==	AD01047147	Default
3	1	NCloud	AD01047147	127.0.0.1				lazyNtUser	Xck00Pvn7Lq0GjFVAjehhg==	AD01047147	Default
4	1	NCloud	AD01047147	127.0.0.1				lazyNtUser	Xck00Pvn7Lq0GjFVAjehhg==	AD01047147	Default
5	1	NCloud	AD01047147	127.0.0.1				lazyNtUser	Xck00Pvn7Lq0GjFVAjehhg==	AD01047147	Default
6	1	NCloud	AD01047147	127.0.0.1				lazyNtUser	Xck00Pvn7Lq0GjFVAjehhg==	AD01047147	Default
7	1	NCloud	AD01047147	127.0.0.1				lazyNtUser	Xck00Pvn7Lq0GjFVAjehhg==	AD01047147	Default
8	1	NCloud	AD01047147	127.0.0.1				lazyNtUser	Xck00Pvn7Lq0GjFVAjehhg==	AD01047147	Default

생성된 스크립트 중 “실행할 스크립트”라고 된 스크립트 2 개 세트를 SSMS 에서 실행합니다.



### 3. 확장 모니터링 스크립트 활성화

SQLScript 하위 폴더에서 L06\_TargetItemRegisterAdvanced\_04.sql 파일을 열고 아래 변수를 수정 후 SQLCMD 모드로 실행합니다.

#### o SQLCMD 명령어

- o -- 변수 수정
- o :setvar machineName AD01047147
- o -- 모니터링 대상 hostname
- o :setvar FullInstanceName AD01047147
- o --FullInstanceName 은 해당 서버의 select srvname from syssservers where srvid = 0 결과값 임
- o :setvar SqlLogManManagerDatabase SqlLogManManager
- o -- lazylog 모든 설정이 저장된 매니저 데이터베이스 이름 (모니터링 저장소 데이터베이스)
- o -- 변수 수정 끝

## LazyLog.exe 재 시작

확장 모니터링 항목을 수집하기 위해 콘솔 LazyLog.exe 를 재시작합니다. (콘솔 애플리케이션 종료 및 시작)

## 확장 모니터링 정상 수집 확인

기본 모니터링이 정상임을 확인하고 확장 모니터링이 정상인지 확인할 경우 sp\_spaceused\_b 가 정상 수집되면 모든 수집이 정상입니다.

#### • SQLCMD 명령어

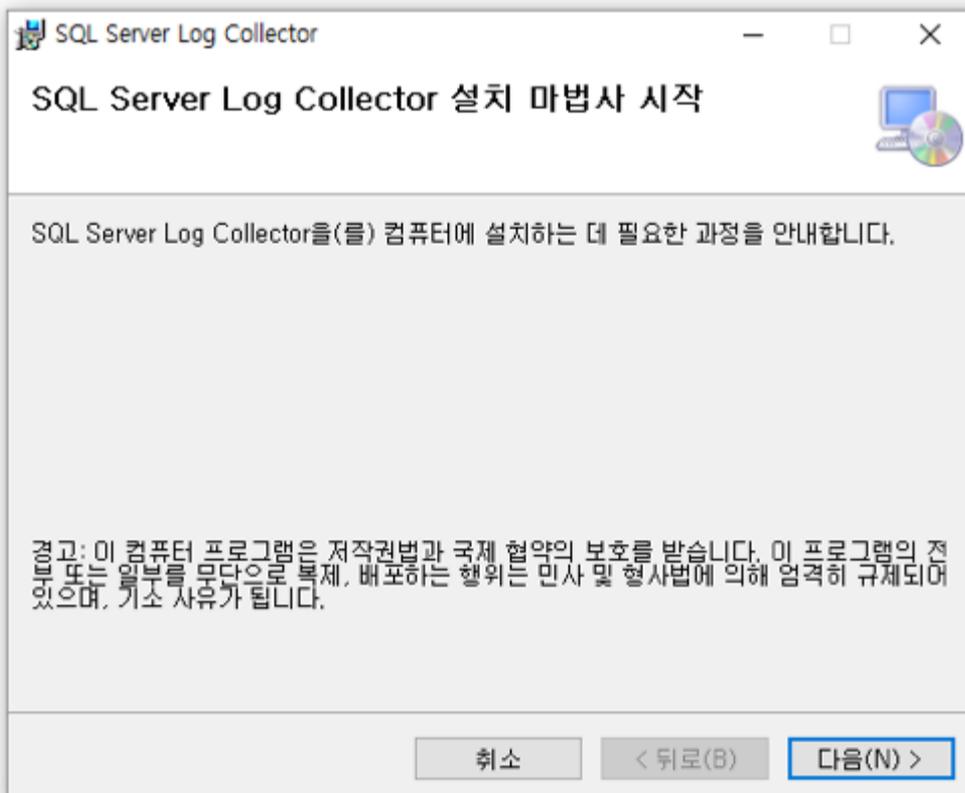
- select \* from SqlLogManDmvRepository.dbo.sp\_spaceused\_b order by probe\_time desc
- 아래와 같이 db\_name 별 테이블사이즈가 정상 출력이 되면 모든 수집이 정상적입니다.

idx	MachineName	FullInstanceName	probe_time	name	rows	reservedKb	dataKb	index_sizeKb	unusedKb	db_name	
1	2604	AD01047147	AD01047147	2017-03-14 15:14:10.940	EventLogSecurity	40473	163152	159864	3192	96	SqlLogManPerfmonRepository
2	2603	AD01047147	AD01047147	2017-03-14 15:14:10.940	CounterData	493246	69904	42672	27168	64	SqlLogManPerfmonRepository
3	2602	AD01047147	AD01047147	2017-03-14 15:14:10.940	EventLogApplication	228	552	448	48	56	SqlLogManPerfmonRepository
4	2601	AD01047147	AD01047147	2017-03-14 15:14:10.940	EventLogSystem	22	80	48	32	0	SqlLogManPerfmonRepository
5	2600	AD01047147	AD01047147	2017-03-14 15:14:10.940	EventLogSetup	0	0	0	0	0	SqlLogManPerfmonRepository
6	2599	AD01047147	AD01047147	2017-03-14 15:14:10.940	DisplayToID	1	48	8	40	0	SqlLogManPerfmonRepository
7	2598	AD01047147	AD01047147	2017-03-14 15:14:10.940	CounterDetails	76	32	8	24	0	SqlLogManPerfmonRepository
8	2597	AD01047147	AD01047147	2017-03-14 15:14:10.843	CounterDetailsAutoUpdated	1819	392	320	16	56	SqlLogManManager
9	2596	AD01047147	AD01047147	2017-03-14 15:14:10.843	MachineSqlInstanceAg	0	0	0	0	0	SqlLogManManager
10	2595	AD01047147	AD01047147	2017-03-14 15:14:10.843	MachineSqlInstance	1	32	8	24	0	SqlLogManManager
11	2594	AD01047147	AD01047147	2017-03-14 15:14:10.843	MachineSqlDatabase	20	16	8	8	0	SqlLogManManager
12	2593	AD01047147	AD01047147	2017-03-14 15:14:10.843	Machine	1	32	8	24	0	SqlLogManManager
13	2592	AD01047147	AD01047147	2017-03-14 15:14:10.843	CounterDetailsFilterInfo	15	16	8	8	0	SqlLogManManager
14	2591	AD01047147	AD01047147	2017-03-14 15:14:10.843	Config	6	16	8	8	0	SqlLogManManager
15	2590	AD01047147	AD01047147	2017-03-14 15:14:10.843	CollectItemTemplate	46	16	8	8	0	SqlLogManManager
16	2589	AD01047147	AD01047147	2017-03-14 15:14:10.843	CollectItem	120	32	24	8	0	SqlLogManManager
17	2588	AD01047147	AD01047147	2017-03-14 15:14:10.830	result	530	72	56	8	8	backupDashboard
18	2587	AD01047147	AD01047147	2017-03-14 15:14:10.830	tblx	2	16	8	8	0	backupDashboard
19	2586	AD01047147	AD01047147	2017-03-14 15:14:10.630	tAthens_NicDetail	320707	55176	54168	400	608	remoteBackupProjectFull
20	2585	AD01047147	AD01047147	2017-03-14 15:14:10.630	fusecheckdaily	933	584	136	8	440	remoteBackupProjectFull
21	2584	AD01047147	AD01047147	2017-03-14 15:14:10.630	tMdsApiResults.JoinComputerSystem	933	584	160	8	416	remoteBackupProjectFull
22	2583	AD01047147	AD01047147	2017-03-14 15:14:10.630	tDimCmdbBackup	80135	188264	187880	8	376	remoteBackupProjectFull
23	2582	AD01047147	AD01047147	2017-03-14 15:14:10.630	serverip	106827	13280	12760	144	376	remoteBackupProjectFull
24	2581	AD01047147	AD01047147	2017-03-14 15:14:10.630	allDataBackup	15460	2440	2064	8	368	remoteBackupProjectFull
25	2580	AD01047147	AD01047147	2017-03-14 15:14:10.630	tDimCmdbBackup2	80735	189544	189176	8	360	remoteBackupProjectFull
26	2579	AD01047147	AD01047147	2017-03-14 15:14:10.630	ipaddresssplited_backup	137141	9032	8736	8	288	remoteBackupProjectFull
27	2578	AD01047147	AD01047147	2017-03-14 15:14:10.630	ipaddresssplited	148750	16200	15936	8	256	remoteBackupProjectFull
28	2577	AD01047147	AD01047147	2017-03-14 15:14:10.630	tMdsApiResults.JoinComputerSystem...	11089	1672	1448	8	216	remoteBackupProjectFull
29	2576	AD01047147	AD01047147	2017-03-14 15:14:10.630	SmartAppErrorLog	434	216	112	8	96	remoteBackupProjectFull
30	2575	AD01047147	AD01047147	2017-03-14 15:14:10.630	allData	1819...	240904	240808	8	88	remoteBackupProjectFull
31	2574	AD01047147	AD01047147	2017-03-14 15:14:10.630	tDimCmdb2	80943	189672	189592	8	72	remoteBackupProjectFull
32	2573	AD01047147	AD01047147	2017-03-14 15:14:10.630	owfsRemote	51	72	8	8	56	remoteBackupProjectFull
33	2572	AD01047147	AD01047147	2017-03-14 15:14:10.630	owfsStauts	1505	328	264	8	56	remoteBackupProjectFull
34	2571	AD01047147	AD01047147	2017-03-14 15:14:10.630	tClubRosIndex	719	728	664	8	56	remoteBackupProjectFull
35	2570	AD01047147	AD01047147	2017-03-14 15:14:10.630	tDimCmdb	117210	293928	293864	8	56	remoteBackupProjectFull
36	2569	AD01047147	AD01047147	2017-03-14 15:14:10.630	tRCAFELOG_CLUBITEM	7	88	24	8	56	remoteBackupProjectFull

## 윈도우 서비스로 설치

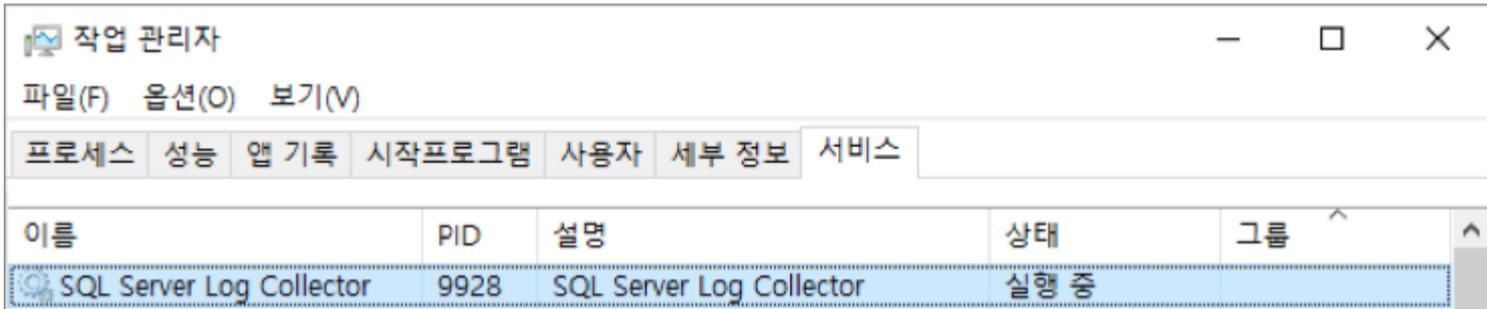
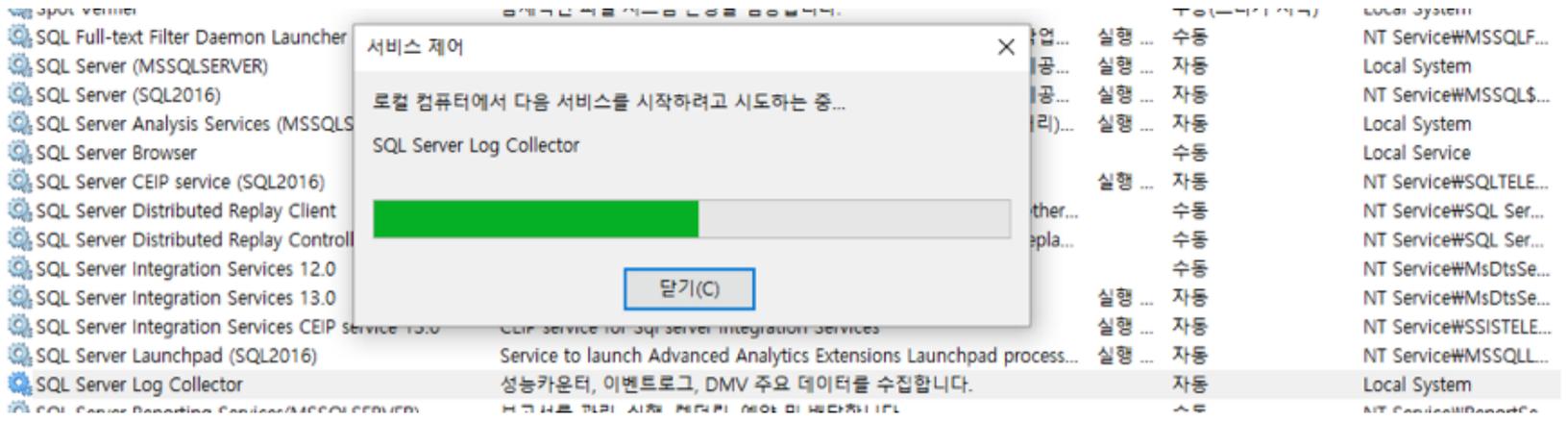
콘솔 모드에서 모든 설정이 완료되면, LazyLogService.zip 파일을 압축 해제하고 간단하게 Setup.exe 를 실행하고 Next 를 클릭해 설치를 완료합니다. (윈도우 서비스로 LazyLog 를 구동할 경우, 한 개의 서비스만 구동 가능하므로 다수의 LazyLog 를 구동해야 한다면, 콘솔 모드로 사용합니다.)

LazyLogInstall.msi	2017-03-13 오후...	Windows Installer...	1,061KB
setup.exe	2017-03-13 오후...	응용 프로그램	560KB



C:\Program Files\Naver Business Platform\SQL Server Log Collector 폴더의 LazyLog.exe.config 파일 중 appSettings 부분만 이전에 설정한 콘솔 애플리케이션의 appSettings 설정을 복사해 넣습니다.

cmd 프롬프트에서 net start "SQL Server Log Collector"을 입력해 서비스를 시작하거나, services.msc 에서 SQL Server Log Collector 를 시작합니다.



서비스로 구동할 경우 모니터링 서버가 다시 시작되어도 자동으로 수집을 시작합니다.

모든 수집이 정상이면, FileLogYN 설정을 N 으로 설정하길 권고합니다.

## 설치 문제 해결

MSSQL 로그 수집기(LazyLog) 설치 문제 해결을 위한 가이드

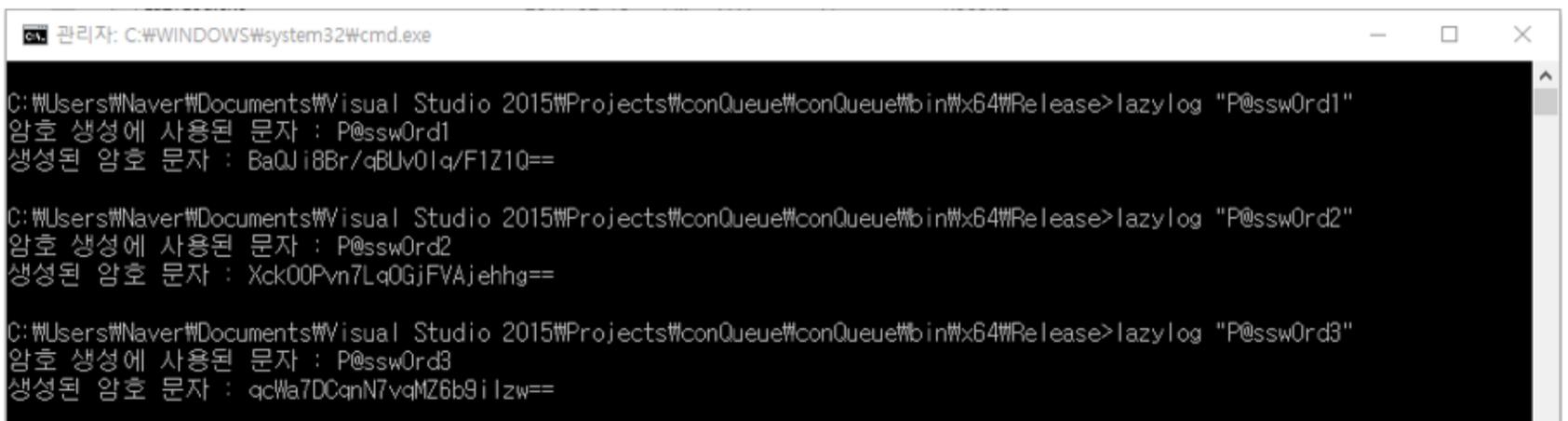
### 문제 해결

문제 해결에서는 예제와 같은 계정으로 설정했다고 가정하고 살펴봅니다. 다른 문제는 Log 파일을 참고해 해결하기 바라며, 대부분의 오류는 계정을 잘못 설정하거나 TCP 445 포트가 ISP 사업자에 의해 막힌 경우였으며, 이 경우 성능 카운터와 이벤트 로그 데이터를 수집할 수 없었습니다. 이 경우 차단된 포트를 쓰지 않도록 수집 프로그램 위치를 변경하면 됩니다.

### Machine 테이블

#### 1. WorkGroup 서버

- **AD01047147** 이라는 호스트명을 가지는 모니터링 서버가 있다면
- **windows/system32/drivers/etc/hosts** 에 host 를 alias 로 등록합니다.
- Ping 과 Telnet 등으로 1433 모니터링 SQL Server 에 접근이 되는지 테스트 합니다.
- MachineName 에 AD01047147 을 기입하고, NtDomain 에도 동일하게 AD01047147 를 입력합니다.
- NtUserName 에 lazyNtUser 를 입력합니다.
- NtPassword 에 암호화된 암호를 적습니다. (lazylog.exe 를 통해 암호를 암호화해 적습니다.)



- IsEnabledYN 이 Y 인지 확인

## 2. Active Directory 서버

- 모니터링 대상 호스트가 **mhv1.minsouk.com** 서버가 있다면
- NtDomain 에 minsouk.com 만 적습니다. (**hostname** 은 제외)
- MachineName 에 mhv1[호스트이름만] 기입하면 됩니다.  
(슬래시 @ 등의 다른 내용이 들어가면 안 됩니다.)
- NtUserName 에 lazyNtUser 나 모니터링 아이디를 적습니다. (**도메인 명 제외**)
- NtPassword 에 위 계정의 암호를 lazylog.exe 를 통해 암호화한 값을 적습니다.
- IsEnabledYN 이 Y 인지 확인

## MachineSqlInstance 테이블

SQL 서버가 기본 인스턴스라면, MHV1 만 입력하고, 명명된 인스턴스로 설치되었다면 FullInstacneName 을 모두 입력합니다. (MHV1\SQL2016 와 같이 입력되면 됩니다.)

DMV 항목 등을 CollectItem 에서 IsNtAuthYN 값이 Y 일 경우는 Machine 에 기록된 NtUserName 과 NtPassword 값을 이용해 접근하고, N 일 경우는 MachinePublicIp 와 PortDefault 를 이용해 접근합니다.

IsEnabledYN 이 Y 인지 확인합니다. 아래 그림을 참조해 수정합니다.

```

select * from machine
select * from machineSqlInstance
select * from collectitem
where itemName in (
    'Application'
    , 'BaselineCollect'
    , 'dm_exec_query_stats2'
    , 'dm_os_workers'
    , 'dm_os_wait_stats'
    , 'sp_lock2'
    , 'sysdatabases'
    , 'dm_os_sys_info'
)
order by machineid, itemname
    
```

MachineId	IdcName	MachineName	MachinePublicIp	MachinePrivateIp	MachineClusterIp	MachineAgIp	NtUserName	NtPassword	NtDomain	NtSessionAuthenticationType	IsEnabledYN
1	NCloud	AD01047147	127.0.0.1				lazyNtUser	[REDACTED]	AD01047147	Default	Y
2	NCloud	mhv1	192.168.137.4				administrator	[REDACTED]	minsouk.com	Default	Y

MachineId	SqlInstanceId	FullInstanceName	SqlUsername	SqlPassword	RoleName	SqlVersion	PortDefault	IsEnabledYN
1	1	AD01047147	lazyClient	[REDACTED]	테스트서버	2014	1433	Y
2	6	MHV1				2014	1433	Y

CollectId	ItemType	ItemName	ItemDesc	MachineId	SqlInstanceId	DatabaseName	CollectIntervalSec	IsNtAuthYN	IsEnabledYN	PurgeIntervalSec	PurgeDurationHH	IsPurgeEnabledYN	ItemOption
1	7	Application	LOG	1	NULL	NULL	10	Y	Y	10	48	Y	
2	8	PERFMON	BaselineCollect	1	NULL	NULL	NULL	Y	Y	20	720	Y	counterReload=N (Processed by lazylog
3	1	DMV	dm_exec_query_stats2	1	1	master	60	N	Y	60	48	Y	topCnt=20000, SrcDelHH=1, IsNewDmv=fal
4	6	DMV	dm_os_sys_info	1	1	master	21600	N	Y	21600	720	Y	
5	3	DMV	dm_os_wait_stats	1	1	master	60	N	Y	60	48	Y	
6	2	DMV	dm_os_workers	1	1	master	10	N	Y	10	48	Y	
7	4	DMV	sp_lock2	1	1	master	10	N	Y	10	48	Y	
8	5	DMV	sysdatabases	1	1	master	21600	N	Y	21600	720	Y	
9	121	EVENTLOG	Application	6	NULL	NULL	10	Y	Y	10	48	Y	
10	122	PERFMON	BaselineCollect	6	NULL	NULL	NULL	Y	Y	20	720	Y	counterReload=N (Processed by lazylog
11	123	DMV	dm_exec_query_stats2	6	7	master	60	Y	Y	60	48	Y	topCnt=20000, SrcDelHH=1, IsNewDmv=fal
12	128	DMV	dm_os_sys_info	6	7	master	21600	Y	Y	21600	720	Y	
13	125	DMV	dm_os_wait_stats	6	7	master	60	Y	Y	60	48	Y	
14	124	DMV	dm_os_workers	6	7	master	10	Y	Y	10	48	Y	
15	126	DMV	sp_lock2	6	7	master	10	Y	Y	10	48	Y	
16	127	DMV	sysdatabases	6	7	master	21600	Y	Y	21600	720	Y	

쿼리가 실행되었습니다. | (local)\(12.0 SP1) | AD01047147\Naver (103) | SqlLogManManager 00:00:00 20개의 행

위 설정 중 1 번 MachineId 는 WorkGroup 서버이고, 6 번 MachineId 는 ActiveDirectory 서버의 설정입니다. LazyLog 는 위 정보를 기반으로 가장해 윈도우 성능카운터, 이벤트로그, DMV 를 읽어옵니다.

## 내부ConnectionString 생성 로직

```
참조 83개
public string GetTargetServerConnString(Config.CollectItem item)
{
    if (item == null)
        log.WriteLine("Target Server Connection ITEM이 비어 있습니다.");

    string ConnectionString = string.Empty;
    try
    {
        if (item.PortDefault == "") item.PortDefault = "1433";
        if (item.IsNtAuthYN == "N")
        {
            ConnectionString =
                @"Data Source=" + item.MachinePublicIp
                + "," + item.PortDefault
                + ";Initial Catalog=" + item.DatabaseName
                + ";User ID="
                + item.SqlUserName
                + ";Password ="
                + item.SqlPassword;
        }
        else if (item.IsNtAuthYN == "Y")
        {
            ConnectionString =
                @"Data Source=" + item.MachinePublicIp
                + "," + item.PortDefault
                + ";Initial Catalog=" + item.DatabaseName
                + ";Integrated Security=True;";
        }
    }
    catch (Exception e)
    {
        log.WriteLine(string.Format("Exception {0} \r\n {1}", e.Message, e.StackTrace));
    }
    return ConnectionString;
}
```

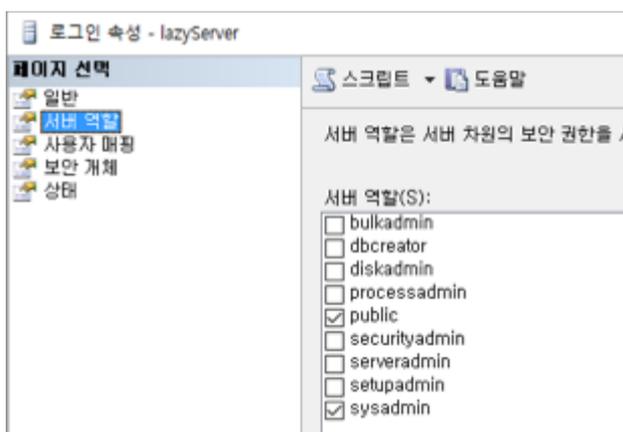
워크 그룹 환경에서 **DMV** 수집을 위해 대상서버에 호스트 이름으로 **Ping** 이나 **telnet** 으로 **1433** 포트가 정상 동작 하는지 살펴봐야 합니다(사용자 지정 포트일 경우 해당 포트 확인). 성능 카운터와 이벤트로그 수집을 위해 **Ping** 이나 **445** 포트가 정상 동작 하는지 확인되어야 합니다.

## 계정별 접근 테스트

### 1. lazyServer

**lazylog.exe.config** 에 설정되며, 수집된 데이터를 저장할 **SQL Server** 에 계정과 권한이 설정되어야 합니다.

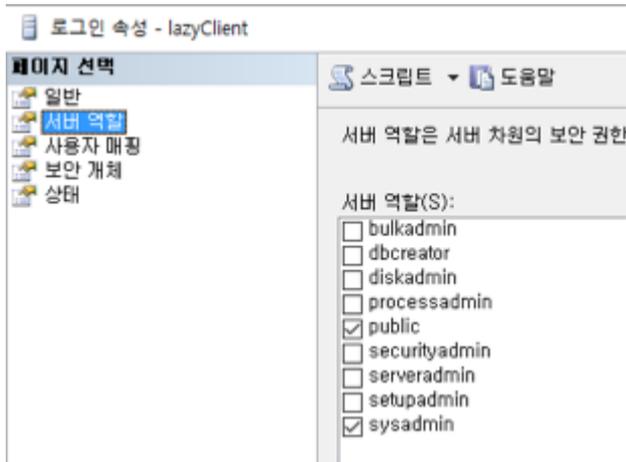
- SQL Server 계정으로 저장소 데이터베이스에 접근 가능해야 합니다. (lazyLog.config.XML 에 설정) SSMS 나 SQLCMD 로 테스트를 수행할 때는 lazyServer 계정이 암호화되기 전 암호값으로 lazyLog.exe 프로그램이 동작하는 컴퓨터에서 SqlLogManManager 데이터베이스가 있는 SQL Server 에 로그인이 되는지 확인합니다. (IP,Port 정확하게 테스트)
- lazyServer 가 sysadmin 이나 dbo 권한을 가지고 있는가?



### 2. lazyClient

**MachineSqlInstance** 에 설정되며, 수집 대상 **SQL Server** 에 계정과 권한이 설정되어야 합니다.

- SQL Server 계정으로 모니터링 대상 서버에 SSMS 나 SQLCMD 프로그램으로 해당 서버에 접근이 가능한가? (lazyClient 계정은 SqlLogManManager 데이터베이스의 Machine 과 MachineSqlInstance 정보로 대상 서버에 접근합니다. SSMS 에서 테스트할 때는 암호화하기 전 값으로 입력해야 합니다.)
- lazyClient 가 sysadmin 권한을 가지고 있는가?

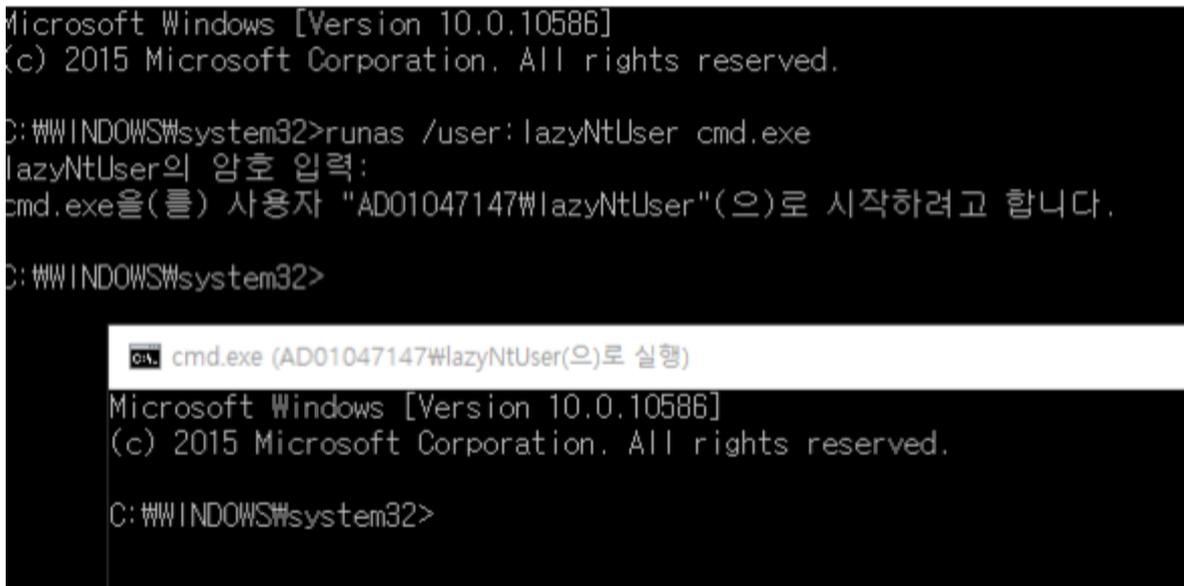


### 3. lazyNtUser

**Machine** 테이블에 설정되며, 대상 컴퓨터에 계정과 권한이 설정되어야 합니다.

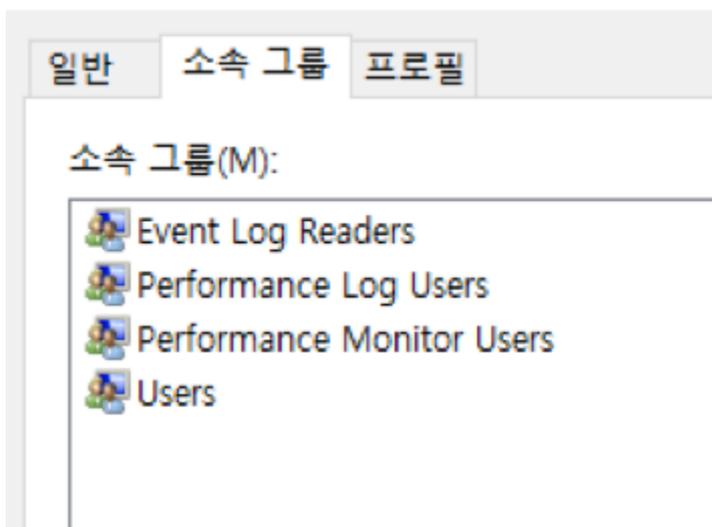
- 성능 카운터와 이벤트로그를 읽기 위해 사용됩니다. (SQL 인증은 허용되지 않습니다.)
- 설정한 암호가 정확한지 수집 대상 서버에 Administrator 로 로그인해 **Runas** 명령어로 다음과 같이 실행합니다.
- runas /user:lazyNtUser cmd.exe

관리자: 명령 프롬프트



- cmd 창이 정상적으로 lazyNtUser 로 시작되면 암호는 정확합니다.
- lazyNtUser 의 권한을 확인합니다. 대상 서버에 설정되어야 할 권한은 아래와 같습니다.

### lazyNtUser 속성



## 계정항목 확인

L07\_FullCollectItemStatus.sql 내용을 기준으로 아래 쿼리를 실행하고, IsEnabledYN 에 모두 Y 인가?

```
-- 전체 등록 아이템 확인
-- 변수 수정
:setvar SqlLogManManagerDatabase SqlLogManManager
-- 변수 수정 끝

USE $(SqlLogManManagerDatabase)
GO

select a.machinename, a.IsEnabledYN, b.IsEnabledYN, c.IsEnabledYN
, c.CollectIntervalSec, c.PurgeDurationHH
, b.FullInstanceName, c.itemtype, c.DatabaseName, c.itemname, c.ItemOption
from SqlLogManManager.dbo.machine a
join SqlLogManManager.dbo.MachineSqlInstance b
on a.machineid = b.machineid
join SqlLogManManager.dbo.collectitem c
on a.Machineid=c.machineid
and b.SqlInstanceId = c.SqlInstanceId
--where
```

	machinename	IsEnabledYN	IsEnabledYN	IsEnabledYN	CollectIntervalSec	PurgeDurationHH	FullInstanceName	Itemtype	DatabaseName	Itemname	ItemOption
1	AD01047147	Y	Y	Y	60	48	AD01047147	DMV	master	dm_exec_query_stats2	topCnt=20000, SrcDelHh=1, IsNewDmv=false, IsGath...
2	AD01047147	Y	Y	Y	10	48	AD01047147	DMV	master	dm_os_workers	
3	AD01047147	Y	Y	Y	60	48	AD01047147	DMV	master	dm_os_wait_stats	
4	AD01047147	Y	Y	Y	10	48	AD01047147	DMV	master	sp_Lock2	
5	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	sysdatabases	
6	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	dm_os_sys_info	
7	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	sysjobs	
8	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	server_principals	
9	AD01047147	Y	Y	Y	300	48	AD01047147	DMV	master	fn_trace_getinfo	
10	AD01047147	Y	Y	Y	300	48	AD01047147	DMV	master	dm_xe_sessions	
11	AD01047147	Y	Y	Y	300	720	AD01047147	DMV	master	dm_server_services	
12	AD01047147	Y	Y	Y	60	48	AD01047147	DMV	master	dm_os_memory_clerks	
13	AD01047147	Y	Y	Y	300	48	AD01047147	DMV	master	dbcc_memorystatus	
14	AD01047147	Y	Y	Y	20	48	AD01047147	DMV	master	dm_db_task_space_usage	
15	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	sysaltfiles	
16	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	sp_configure	
17	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	dbcc_tracestatus	
18	AD01047147	Y	Y	Y	20	48	AD01047147	DMV	master	suspect_pages	
19	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	dm_db_missing_index_group_stats	
20	AD01047147	Y	Y	Y	20	48	AD01047147	DMV	master	backupsset	
21	AD01047147	Y	Y	Y	10	48	AD01047147	DMV	master	sp_readererrorlog	
22	AD01047147	Y	Y	Y	20	48	AD01047147	DMV	master	dm_io_virtual_file_stats	
23	AD01047147	Y	Y	Y	20	48	AD01047147	DMV	master	sysjobhistory	
24	AD01047147	Y	Y	Y	60	48	AD01047147	DMV	master	dbcc_sqlperf_logspace	
25	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	master	sp_spaceused	
26	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	msdb	sp_spaceused	
27	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	SmartData	sp_spaceused	
28	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	adventurewor...	sp_spaceused	
29	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	tsService	sp_spaceused	
30	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	XEventExcel	sp_spaceused	
31	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	clntest	sp_spaceused	
32	AD01047147	Y	Y	Y	21600	720	AD01047147	DMV	config	sp_spaceused	

## 원격 수집을 위한 포트 및 서비스 확인

### 1. WorkGroup 서버

- 워크그룹일 경우 대상 서버에 lazyNtUser 의 암호와 동일하게 lazyNtUser 를 lazylog.exe 가 동작하는 서버에 계정과 암호를 같게 생성하고 해당 계정으로 윈도우 로그인을 합니다.
- 실행에서 Perfmon 을 수행합니다.
- 카운터 추가 메뉴에서 다음 컴퓨터에서 카운터 선택란에 수집대상 IP 를 입력합니다.
- 원격 컴퓨터의 성능 카운터 수집이 정상적으로 되는지 확인합니다.

### 2. Active Directory 서버

- 실행에서 Perfmon 을 수행합니다.
- 원격 컴퓨터의 성능 카운터 수집이 정상적으로 되는지 확인합니다.

### 3. 방화벽

- 동일 IDC 내에서 방화벽이 없는 환경에서 테스트 하거나, 동일 네트워크 환경에서 테스트 해보고 동작하는지 테스트 합니다.
- RPC 포트가 막혀 있을 경우, 수집 프로그램을 방화벽 안쪽에 위치시키고 수집 결과만, 외부 SQL Server 를 지정하는 것도 좋은 방법입니다.
- 네트워크 패킷 분석 프로그램을 대상 컴퓨터와 수집 컴퓨터 양쪽에 트레이스를 걸고 RPC 445 포트가 정상인지 모니터링 합니다.
- 네트워크 패킷 분석 프로그램을 대상 컴퓨터와 수집 컴퓨터 양쪽에 트레이스를 걸고 TCP 1433 포트가 정상인지 모니터링 합니다.

### 4. 모니터링 대상 서버의 필수 서비스

- Remote Registry 서비스가 활성화되어 있는가?

- Remote Procedure Call(RPC) 서비스가 활성화되어 있는가?

## 5. SQL Server

- sp\_readerrorlog 에 어떤 오류가 있는지 확인합니다.
- 프로파일러나 확장 이벤트로 쿼리와 에러를 확인합니다.

## 활용 예

MSSQL 로그 수집기(LazyLog) 활용 예시

### DMV 데이터 활용 예시

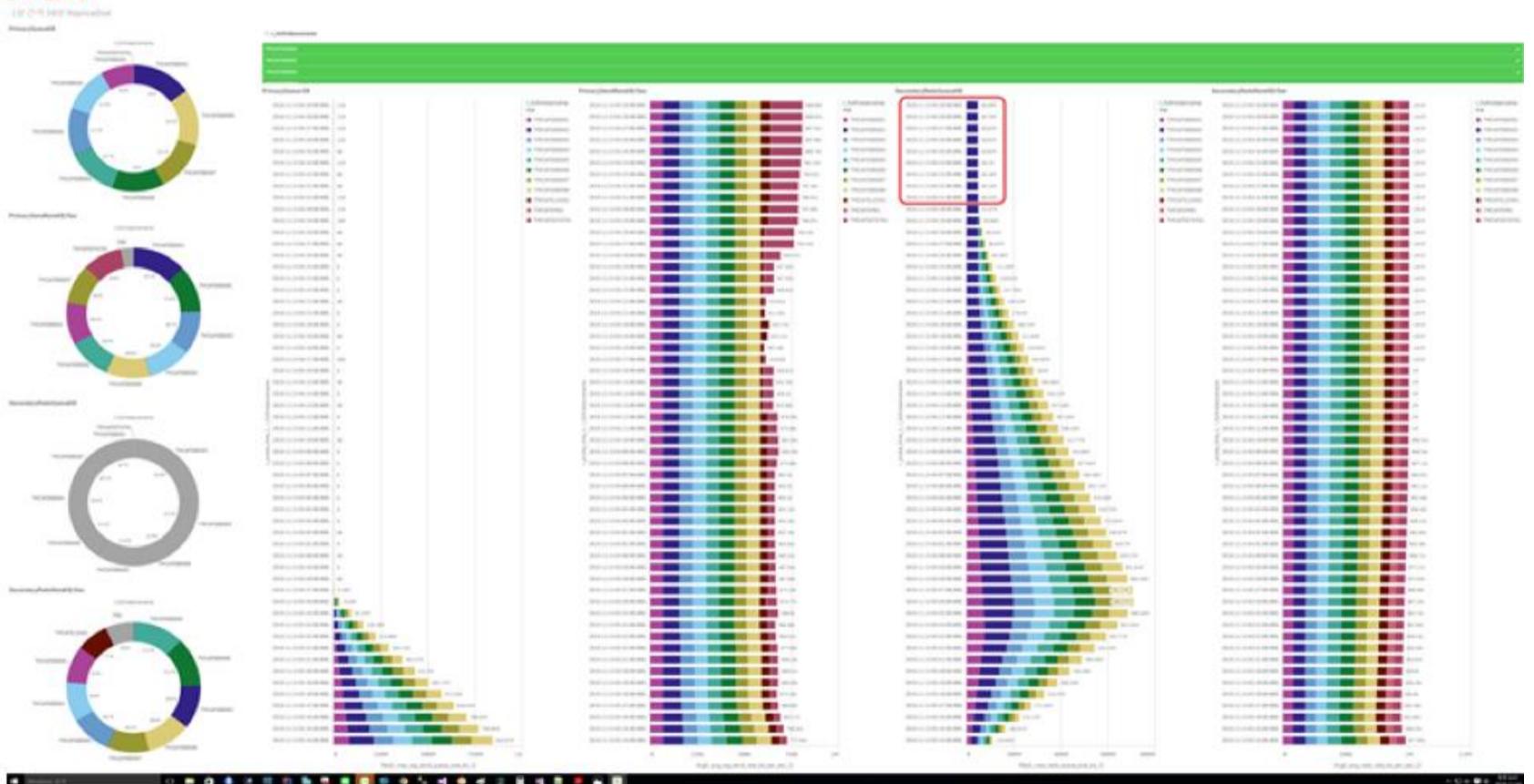
아래 대시보드는 위 LazyLog 수집 데이터를 이용해 단순히 그래프화한 것입니다. 대시보드나 그래프를 그리기 위해 Excel 이나 다양한 대시보드 툴이 이용될 수 있습니다.

#### AG 모니터링

아래 그림은 SQL Server 고가용 솔루션인 AlwaysOn 모니터링을 구성한 예입니다. 아래 그림 1의 스택 차트는 스택별 개별 서버를 의미하며, 현재 Primary 서버의 Queue 에 쌓여있는 용량, Primary 에서 Secondary 로 보내는 속도, Secondary Queue Redo 에 쌓여있는 용량, Secondary Redo 반영 속도를 각각 표현합니다.

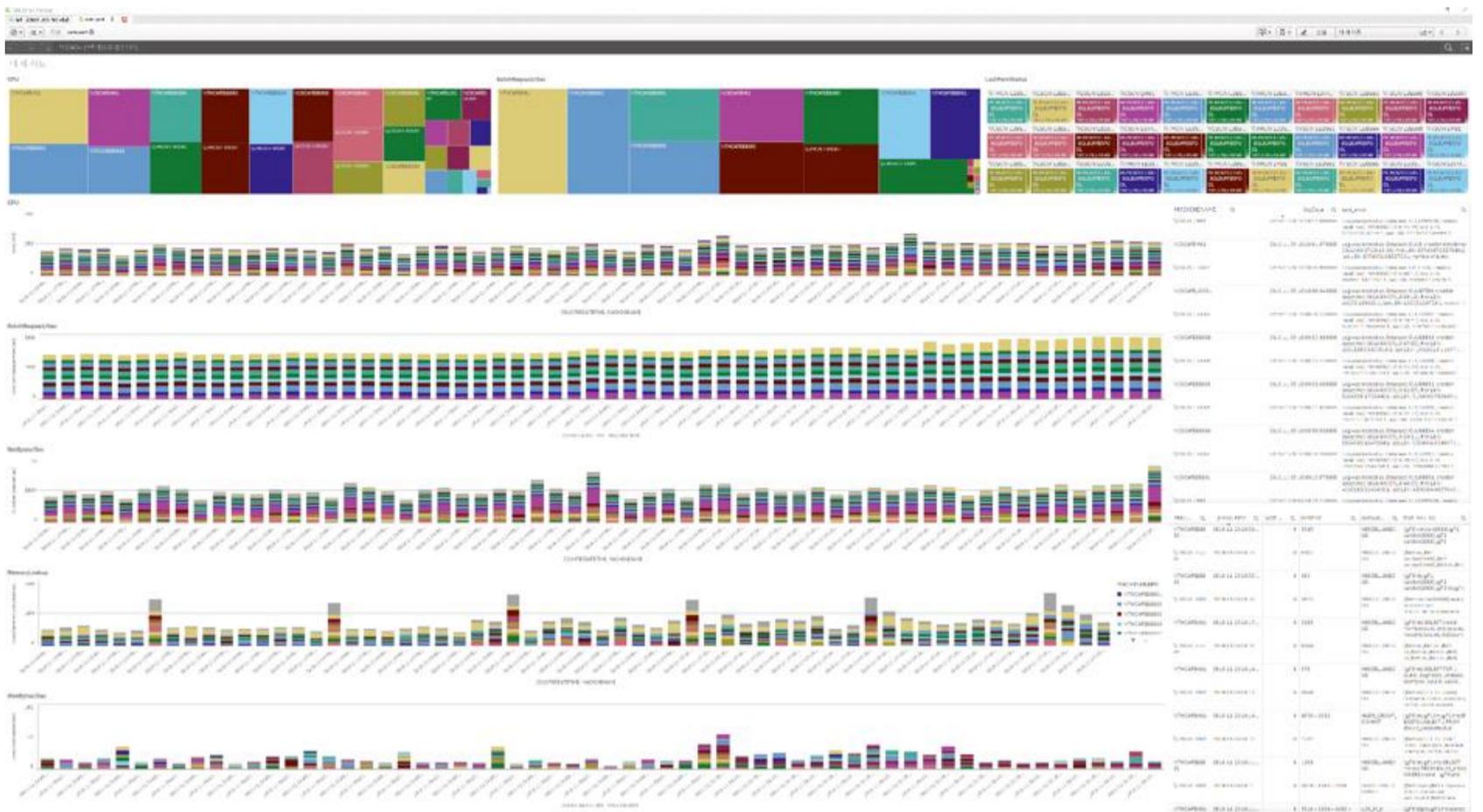
중앙 붉은색 박스를 살펴보면, 한 개 서버의 로그 적용이 정지된 상태를 모니터링 할 수 있습니다. 이 경우 보조서버로 모든 로그가 전송된 상태로 모니터링 되었고 적용만 정지되거나 매우 느린 것으로 보입니다. 이 경우 보조서버를 재시작해 문제를 해결한 경우입니다. 이 문제를 모니터링 하고 있지 않고 계속 서비스를 했다면, 최악의 경우 AG 를 재구성해야 하는 상황이 되었을 수 있습니다. (보조 서버에서 로그 적용은 동작중인 상태와 재시작시 조금 다른 로직으로 디자인 되어 있고, 재시작으로 로그 적용을 하면 더 빠르게 적용 할 수 있습니다.) LazyLog 에는 dm\_hadr\_database\_replica\_states\_b 테이블에 아래 정보가 20 초 단위로 모니터링 되고 있습니다.

[그림1]



#### 서버 리소스 사용률 모니터링

아래 그림은 같은 그룹의 서버를 모니터링 하며, CPU, BatchRequests/Sec, Memory 상황과 네트워크 전송율을 한번에 모니터링 하도록 구성한 페이지의 예입니다. 평소와 다르게 동작하는 서버가 있다면 즉시 발견 가능합니다. LazyLog 에는 성능카운터 정보가 15 초 단위로 기록되어 있습니다.



### 쿼리 리소스 사용량 검증

아래 그림은 같은 역할을 수행하는 2대의 서버를 대상으로 쿼리 모니터링을 한 예입니다. 같은 역할을 수행하는 데이터베이스이므로 쿼리도 비슷하게 유입되고 CPU WorkerTime 도 비슷해야 합니다. 그런데 특정 시간대에 한 서버에서 특정 쿼리만 리소스 사용량이 높아진 것을 바로 인지 가능한 경우입니다.

중앙 버블 차트는 X 축은 수행 회수, Y 축은 WokerTime 입니다. 쿼리를 보고 특정 시간대 어떤 쿼리가 가장 리소스를 많이 소모하는지 즉시 알 수 있습니다. LazyLog 의 dm\_exec\_query\_stats2\_b\_summary 에는 1 분 단위 쿼리별 리소스 사용량이 상세히 기록되어 있습니다.



# 잠금 모니터링

아래는 잠금을 모니터링 한 예입니다. 서버에서 잠금이 생기면 잠금의 원인 쿼리는 무엇이고 대기하는 쿼리가 어떤 쿼리인지 즉시 알 수 있습니다. LazyLog 의 sp\_lock2\_b 에는 10 초 단위 잠금 모니터링이 쿼리와 함께 상세히 기록되어 있습니다. OLTP 환경에서 잠금 모니터링이 중요하다면 수집 간격을 적절하게 조절합니다. (3 초 권장)

MACHINENAME	l_probe_time	wait_sec	locktree	lastwaittype	text_lock_sql
WTMCAFEM81	2017-03-14 PM 9:49:44	0	5896 + 2433	MISCELLANEOUS	[@P0 int] SELECT clubid, memberid, delegatedate, acceptdate, enddate, status FROM dbo.cit_delegatemember WHERE clubid = @P0
WTMCAFEM81	2017-03-14 PM 9:49:44	0	3826	MISCELLANEOUS	[@P0 int,@P1 int] UPDATE dbo.cit_clublog SET uparticlcount = ISNULL(uparticlcount,0) + @P0 WHERE clubid = @P1
WTMCAFEM81	2017-03-14 PM 9:49:44	0	5896	MISCELLANEOUS	[@P0 int,@P1 varchar(8000),@P2 varchar(8000),@P3 varchar(8000),@P4 varchar(8000),@P5 varchar(8000),@P6 varchar(8000),@P7 varchar(8000),@P8 varchar(8000),@P9 varchar(8000),@P10 varchar(8000),@P11 varchar(8000),@P12 varchar(8000),@P13 varchar(8000),@P14 var...
WTMCAFEM81	2017-03-14 PM 9:49:44	0	1948	MISCELLANEOUS	[@P0 varchar(8000)] SELECT memberid, invitepenaltydate, writepenaltydate, emailpenaltydate, penaltycount, penaltystartdate, writepenaltystartdate, invitepenaltystartdate, emailpenaltystartdate FROM dbo.cit_b...
VCSQLMON01-INFRA	2017-03-14 PM 9:49:28	0	144	PAGELATCH_EX	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11),@SrcDelHH bigint) set nocount on set transaction isolation level read uncommitted whi...
VCSQLMON01-INFRA	2017-03-14 PM 9:49:28	0	144 + 116 + 75	LCK_M_X	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11),@SrcDelHH bigint) set nocount on set transaction isolation level read uncommitted whi...
VCSQLMON01-INFRA	2017-03-14 PM 9:49:28	1	144 + 116 + 132	LCK_M_X	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11),@SrcDelHH bigint) set nocount on set transaction isolation level read uncommitted whi...
VCSQLMON01-INFRA	2017-03-14 PM 9:49:28	2	144 + 116	LCK_M_X	insert bulk dm_exec_query_stats_5 ([MachineName] NVarChar(100) COLLATE Korean_Wansung_CI_AS, [FullInstanceName] NVarChar(100) COLLATE Korean_Wansung_CI_AS, [probe_time] DateTime, [sp_handle] VarBinary(64), [statement_start_offset] int, [statement_end_of...
VCSQLMON01-INFRA	2017-03-14 PM 9:49:18	0	144	PAGELATCH_EX	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11),@SrcDelHH bigint) set nocount on set transaction isolation level read uncommitted whi...
VCSQLMON01-INFRA	2017-03-14 PM 9:49:18	2	144 + 132	LCK_M_U	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11),@SrcDelHH bigint) set nocount on set transaction isolation level read uncommitted whi...
VCSQLMON01-INFRA	2017-03-14 PM 9:49:18	1	144 + 132 + 82	LCK_M_X	insert bulk dm_exec_query_stats_5 ([MachineName] NVarChar(100) COLLATE Korean_Wansung_CI_AS, [FullInstanceName] NVarChar(100) COLLATE Korean_Wansung_CI_AS, [probe_time] DateTime, [sp_handle] VarBinary(64), [statement_start_offset] int, [statement_end_of...
WTMCAFEM81	2017-03-14 PM 9:48:42	0	2139	MISCELLANEOUS	[@P0 int] SELECT TOP 1 clubid, begindate, enddate, opentype, cat1id, cat2id, weeklymember, weeklyarticle, weeklycomment, searchreadct, answercount, selectedanswercount, membercount, bestcase ...
WTMCAFEM81	2017-03-14 PM 9:48:42	0	4627	MISCELLANEOUS	[@P0 int] SELECT clubid, memberid, delegatedate, acceptdate, enddate, status FROM dbo.cit_delegatemember WHERE clubid = @P0
WTMCAFEM81	2017-03-14 PM 9:48:42	0	5786	MISCELLANEOUS	[@P0 int,@P1 varchar(8000)] SELECT clubid, memberid, nickname, nickyscale, jointd, memberlevel, lastvaid...
WTMCAFEM81	2017-03-14 PM 9:48:42	0	757	MISCELLANEOUS	[@P0 int,@P1 varchar(8000)] select clubmember.clubid, clubmember.memberid, clubmember.jointd, clubmember.memberlevel, clubmember.lastvaid, clubmember.ceremonyyn, clubmember.realnameyn, clubmember.group...
WTMCAFEM81	2017-03-14 PM 9:48:42	0	2139 + 7927	MISCELLANEOUS	[@P0 int,@P1 varchar(8000),@P2 varchar(8000),@P3 varchar(8000),@P4 varchar(8000),@P5 varchar(8000),@P6 varchar(8000),@P7 varchar(8000),@P8 varchar(8000),@P9 varchar(8000),@P10 varchar(8000),@P11 varchar(8000),@P12 varchar(8000),@P13 varchar(8000),@P14 var...
WTMCAFEM81	2017-03-14 PM 9:48:42	0	3026	MISCELLANEOUS	[@P0 varchar(8000)] SELECT memberid, nickname, sex, sexopen, emailopen, birthdayopen, nameopen, solaryn, mindcode, area, areacopen, agreeyn, badmemberyn, showinterestedcale, alimauth, alimauth...
VCSQLMON01-INFRA	2017-03-14 PM 9:48:07	0	75	PAGELATCH_EX	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11) while (1=1) begin delete top (1000) dm_os_workers_b with(pagelock) ...
VCSQLMON01-INFRA	2017-03-14 PM 9:48:07	0	75 + 82	LCK_M_X	insert bulk dm_os_workers_b ([MachineName] NVarChar(100) COLLATE Korean_Wansung_CI_AS, [FullInstanceName] NVarChar(100) COLLATE Korean_Wansung_CI_AS, [probe_time] DateTime, [session_init] int, [current_session_cnt] int, [max_woker_thread] int, [current_...
VCSQLMON01-INFRA	2017-03-14 PM 9:47:57	0	75	MISCELLANEOUS	insert bulk dm_os_wait_stats_b ([MachineName] NVarChar(100) COLLATE Korean_Wansung_CI_AS, [FullInstanceName] NVarChar(100) COLLATE Korean_Wansung_CI_AS, [probe_time] DateTime, [wait_type] VarChar(100) COLLATE Korean_Wansung_CI_AS, [waiting_tasks_count] bi...
VCSQLMON01-INFRA	2017-03-14 PM 9:46:55	0	57 + 121	LCK_M_X	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11) while (1=1) begin delete top (1000) dm_os_workers_b with(pagelock) ...
VCSQLMON01-INFRA	2017-03-14 PM 9:46:55	0	75 + 82	LCK_M_X	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11),@SrcDelHH bigint) set nocount on set transaction isolation level read uncommitted whi...
VCSQLMON01-INFRA	2017-03-14 PM 9:46:55	0	75 + 149	LCK_M_X	[@PurgeDuration] bigint,@MachineName varchar(11),@FullInstanceName varchar(11),@SrcDelHH bigint) set nocount on set transaction isolation level read uncommitted whi...

## 연관 정보 바로가기

아래 가이드에서 연관 정보를 확인할 수 있습니다.

- [MSSQL 서버 이미지 사용자 가이드](#)