



Administração de banco de dados

Aulas 04 e 05- Auditoria de processos e recursos

Gustavo Maia

Gustavo.Maia@FaculdadeImpacta.com.br

Sumário

- Definições
 - Auditoria
 - Recursos compartilhados
 - Processos
- Ferramentas de auditoria diversas

Auditoria

Auditoria é um processo de inspeção, que visa identificar infrações ou mesmo assegurar que tudo está em ordem.

Princípios gerais para desenvolver uma auditoria de segurança:

- **Propósito:** evitar auditorias desnecessárias
- **Precisão:** coletar o mínimo de dados para garantir o objetivo
- **Generalização X Especialização:** auditorias genéricas (ou gerais) devem ser usadas antes das técnicas específicas, porém são mais pesadas.
- **Proteger o rastro da auditoria:** evitar que seus processos interfiram no sistema, evita que informações sejam inseridas na auditoria sem especificação
- **Limpeza e armazenamento dos dados:** após jogar fora o que não é útil, deve-se armazenar corretamente os dados auditados.
- **Periodicidade:** defina a frequência com as informações serão coletadas novamente.

Recursos auditáveis

Os recursos mais comumente auditáveis, também chamados de recursos compartilhados, são aqueles que, por serem usados por todo mundo, nunca podem 'faltar' ou 'acabar'

Recursos compartilhados:

- **CPU** Quantidade computacional da máquina, ou poder computacional.
Termos associados: Socket, core, velocidade (GHz), flops (teraflops), cache
Normalmente aferida em percentual de utilização (%)
- **Memória** Quantidade de espaço para armazenamento volátil e trabalho temporário
Termos associados: física vs virtual, VMM, soft/hard pagefaults, buffer
Aferida pela quantidade (GBs), pelo % de utilização, movimentação/pagefaults
- **Disco** Quantidade de espaço para armazenamento definitivo
Termos associados: RAIDs, RPM, unidades, partições, discos, SAN, storage.
Aferida pela quantidade, % utilização, fila de acesso, IOPS
- **Rede** Quantidade de transferência cliente / servidor
Termos associados: throughput de rede, bytes read, bytes write
Aferida pela quantidade bytes enviados e recebidos.

Velocidade: CPU(cache) > Memória(física) > Memória(virtual) = Disco(local) > Disco(remoto) > Rede(transferência)

Recursos auditáveis

Outros recursos comumente auditáveis são aqueles que interferem no uso ou no dia-a-dia das operações do banco. Basicamente tudo que faz um processo no SGBD (mssql) esperar é chamado de TIPO DE ESPERA (wait type)

A lista de tipos de espera é muito extensa, mas eles podem ser agrupados em tipos para facilitar a identificação dos 'gargalos' (pontos específicos de lentidão).

Exemplos de tipos de espera :

PAGEIOLATCH_SH / PAGEIOLATCH_EX /
LCK_M_IS / LCK_M_IX
CXCONSUMER
CXPACKET
SOS_WORK_DISPATCHER
WRITELOG

Listas de tipos de espera:

[SQL Server wait types - SQL Shack](#)

[sys.dm_os_wait_stats \(Transact-SQL\) - SQL Server | Microsoft Docs](#)

Processos

Toda conexão no SQL Server recebe uma Sessão, numerada (PID > 50) e que pode executar um comando por vez (mesmo que sejam um conjunto de instruções).

Toda sessão identifica um login (no instância) / usuário(no database), sempre deve estar associada à um banco de dados (mesmo que seja o padrão = master).

Uma sessão possui status:

- Background** - Normalmente utilizado para processos do sistema (PID <= 50)
- Sleeping** - Conexão aberta, mas nada sendo executado (awaiting command)
- Runnable** - Rodando algo (mesmo que esteja esperando por recursos - waiting)

Toda sessão possui um nível de isolamento (maneira como ela lida com a concorrência de outros processos) e configurações específicas que regem seu comportamento (timeout, xact_abort, concat_null_yields_null, quoted_identifier, etc).

Qualquer código executado por uma sessão é um processo (ou comando), ele SEMPRE consome algum dos recursos compartilhados e eventualmente 'esbarra' nos tipos de espera.

Processos

Toda conexão no SQL Server recebe uma Sessão, numerada (PID > 50) e que pode executar um comando por vez (mesmo que sejam um conjunto de instruções).

Toda sessão identifica um login (no instância) / usuário(no database), sempre deve estar associada à um banco de dados (mesmo que seja o padrão = master).

Uma sessão possui status:

- Background** - Normalmente utilizado para processos do sistema (PID <= 50)
- Sleeping** - Conexão aberta, mas nada sendo executado (awaiting command)
- Runnable** - Rodando algo (mesmo que esteja esperando por recursos - waiting)

Toda sessão possui um nível de isolamento (maneira como ela lida com a concorrência de outros processos) e configurações específicas que regem seu comportamento (timeout, xact_abort, concat_null_yields_null, quoted_identifier, etc).

Qualquer código executado por uma sessão é um processo (ou comando), ele SEMPRE consome algum dos recursos compartilhados e eventualmente 'esbarra' nos tipos de espera.

Ferramentas de auditoria

São ferramentas de auditoria (sem estarem em uma ordem específica):

- Gerenciador de tarefas (windows task manager / resource manager)
- Performance Monitor (windows perfmon)
- AWS (monitoramento detalhado)
- Monitor de atividade (activity Monitor)
- Procedures internas
- DMVs (Dynamic Management Views)
- SSMS Internal reports
- Visualizador de logs
 - MSSQL
 - Windows (event viewer)
- SQL Server profiler
- Extended Events
- SQL Agent
 - Alerts
- Triggers (Database level triggers)

Entre tantos outros...

Teste de carga

Antes de auditar, temos que gerar carga.

Normalmente a auditoria é utilizada apenas em PRODUÇÃO, porém, às vezes desejamos saber o 'custo' ou 'peso' de um processo antes de seu DEPLOY (ato de publicar o código em um ambiente).

Testes comuns incluem:

- **Execution tests**
Testes controlados, em um ambiente isolado, sem interferência de outros processos, muitas vezes com cache resetado e vários de coleta ligadas
- **Load tests**
Testes de carga, normalmente tentando emular a operação do dia-a-dia da operação
- **Stress tests**
Usado para descobrir a capacidade [máxima] ou para identificar 'gargalos' (qual recurso vai 'acabar' ou 'falhar' primeiro).

Isolando um ambiente de testes

Para limpar o cache:

```
DBCC FREEPROCCACHE
DBCC DROPCLEANBUFFERS
```

***Aviso: NUNCA RODAR EM PRODUÇÃO**

Para coletar estatísticas de execução:

```
set statistics time on
set statistics io on
set statistics xml on
```

Para encerrar outros processos:

```
KILL PID [ WITH STATUSONLY | COMMIT | ROLLBACK ]
```

Pausar o servidor

Previne que novas conexões sejam estabelecidas
Botão direito no servidor > Pause

***Aviso: NUNCA RODAR EM PRODUÇÃO**

Tornar o banco SINGLE_USER

```
ALTER DATABASE [teste_auditoria] SET SINGLE_USER WITH NO_WAIT
```

***Aviso: NUNCA RODAR EM PRODUÇÃO**

Tarefa - Gerar carga

Crie um processo (ou um conjunto de processos) em múltiplas sessões para gerar carga em seu servidor na AWS.

Gere uma quantidade razoável de dados (ou pegue um dos scripts prontos fornecidos)

Escolha entre:

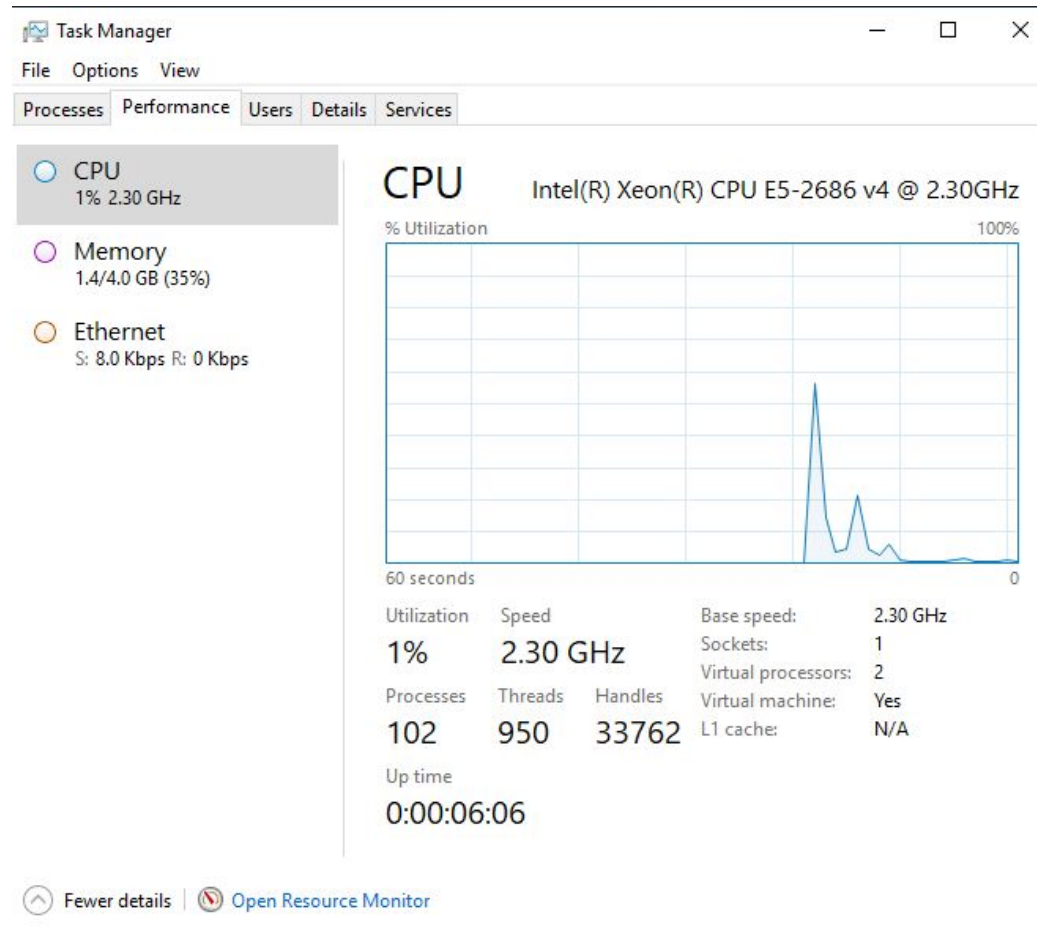
JOINS, ORDER BY, UNION, DISTINCT que são comandos pesados para CPU

Tabelas temporárias, variáveis do tipo tabela, consultas MUITO complexas, para gerar carga de memória.

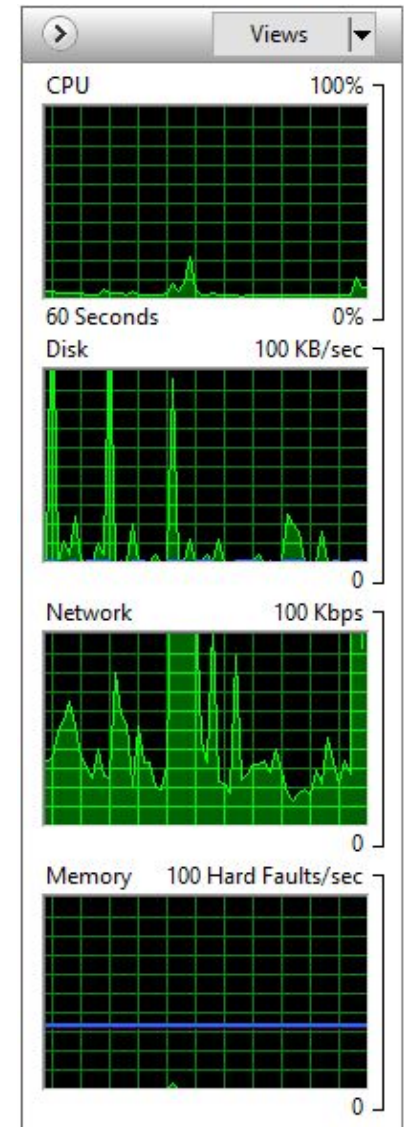
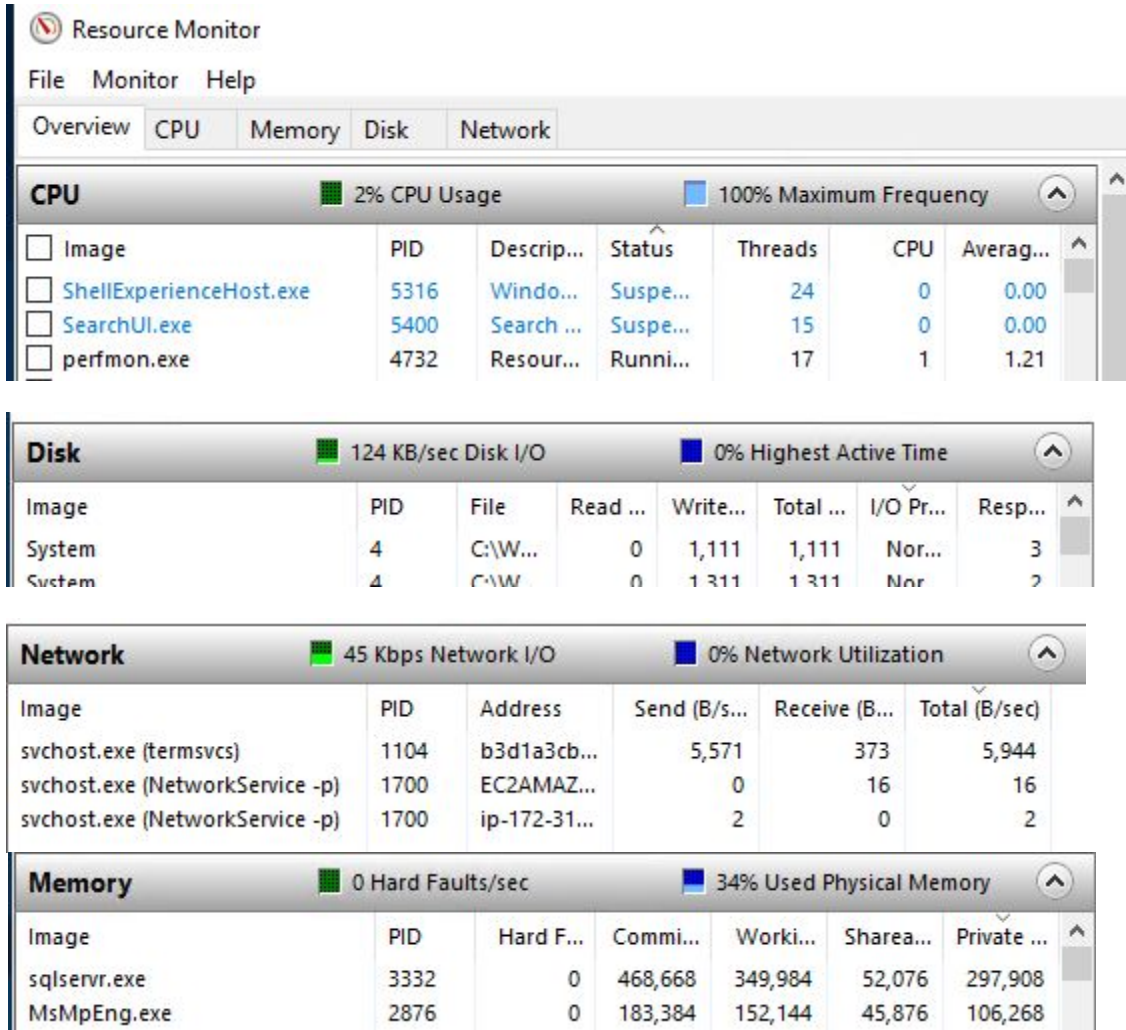
Grandes inserts / deletes / updates para gerar carga de disco.

O Ideal é conseguir gerar um pouco de carga (não precisa ser um stresstest) nos 3 recursos, porém, a CPU podemos tentar manter constante acima de 60%.

Gerenciador de tarefas - task manager



Gerenciador de tarefas - Resource manager



Performance Monitor (windows perfmon)

Coleta estatísticas baseado em contadores escolhidos.

Contadores para CPU:

Object: Processor – Counter: % Processor Time

Object: Processor – Counter: % Privileged Time

Contadores para DISCO:

Object: Physical Disk – Counter Disk Writes/Sec

Object: Physical Disk – Counter Disk Reads/Sec

Object: Physical Disk – Avg. Disk Queue Length

Contadores para Memória:

Object: Memory – Counter: Available Mbytes

Object: SQL Server: Buffer Manager: Buffer cache Hit Ratio

Object: SQL Server: Buffer Manager: Page Life Expectancy

Performance Monitor (windows perfmon)

Contadores para Processos:

Object: SQL Server: General Statistics: Processes Blocked

Object: SQL Server: General Statistics: User Connections

Object: SQL Server: Locks(_total): Average Wait Time (ms)

Contadores para Rede:

Object: Network Interface: Bytes received/sec

Object: Network Interface: Bytes sent/sec

Object: Network Interface: Current Bandwidth

Object: SQL Server: Databases: Log Bytes Flushed/sec (espelhamento)

Contadores para Conexões:

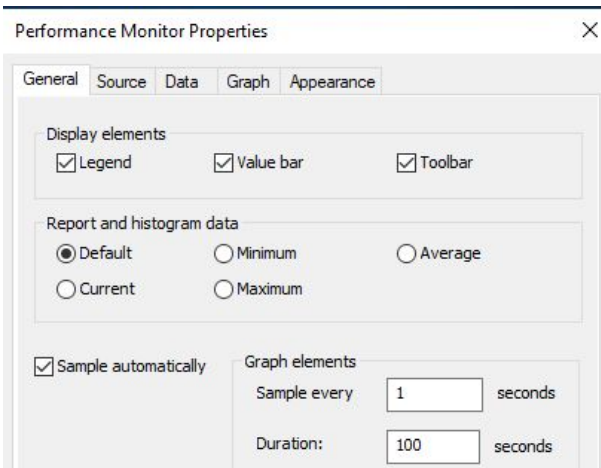
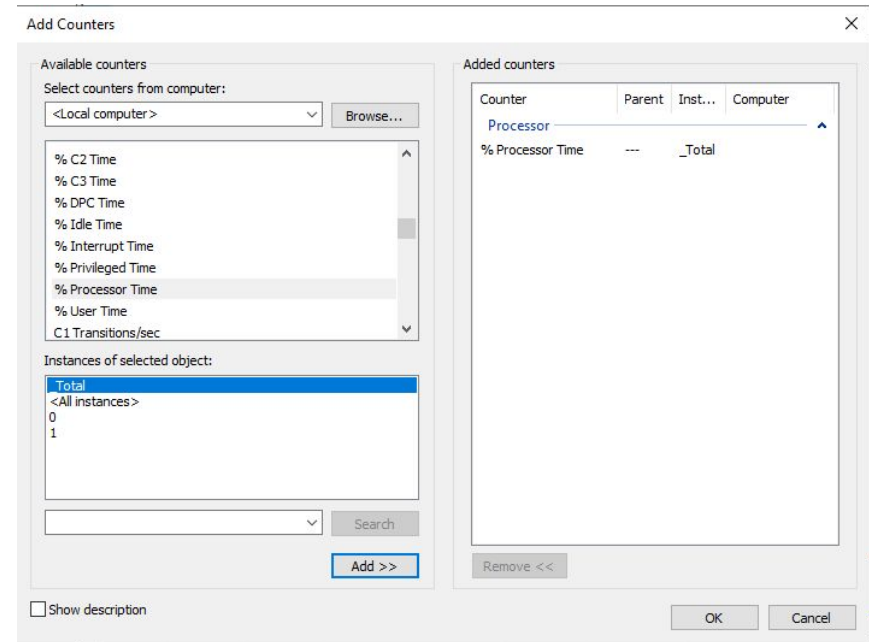
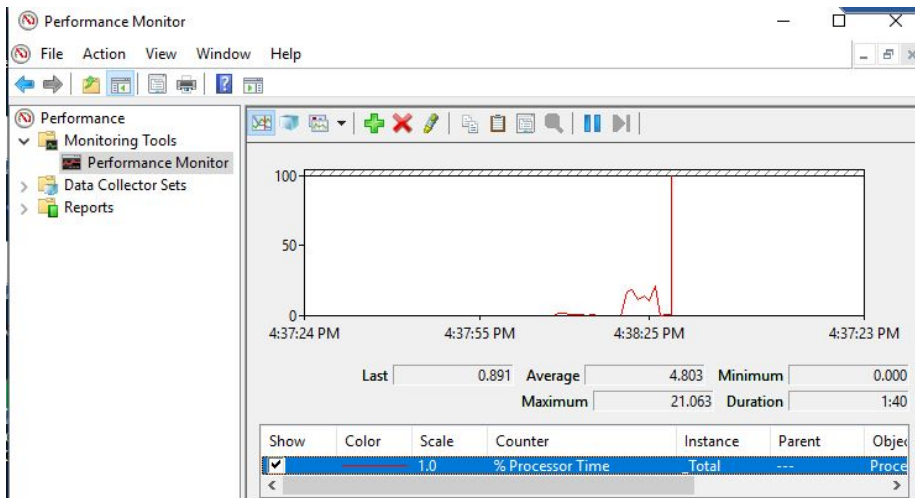
Object:SQL Server:General Statistics: Logical connections

Object:SQL Server:General Statistics: Logins/sec

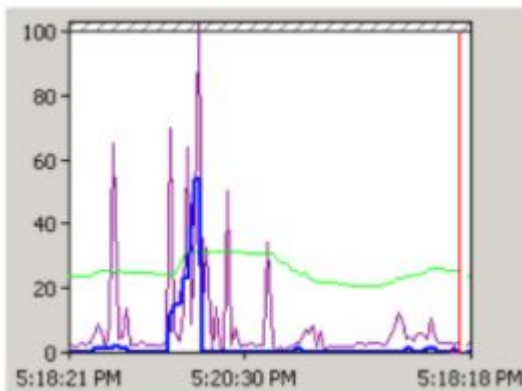
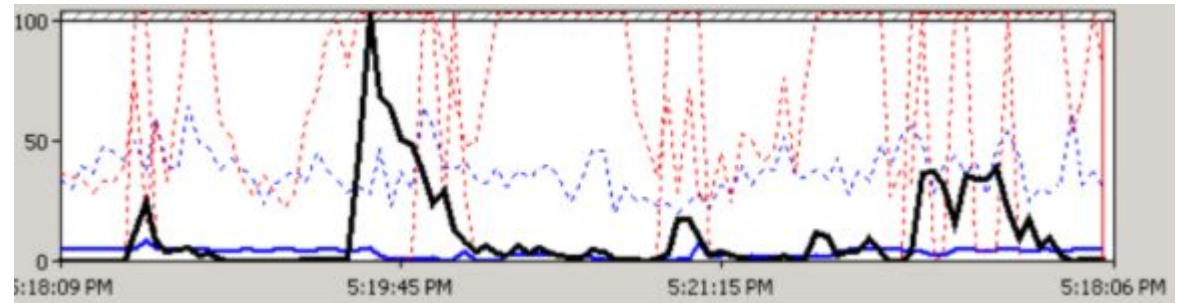
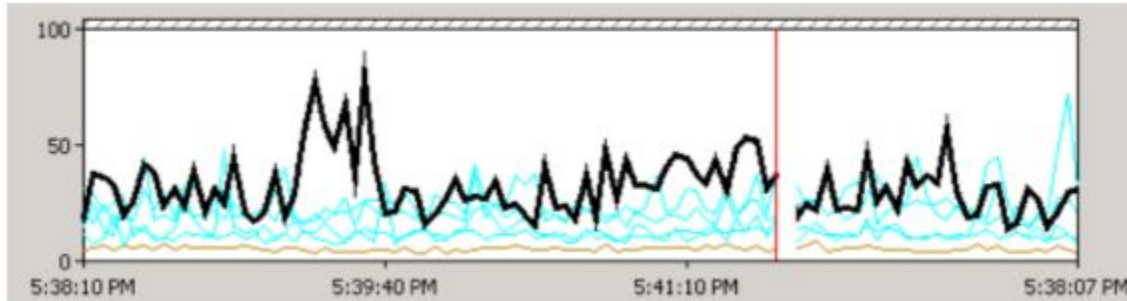
Object:SQL Server:General Statistics: Logouts/sec

Performance Monitor (windows perfmon)

Iniciar > executar > perfmon



Performance Monitor (windows perfmon)



AWS - monitoramento detalhado

Conectar
Estado da instância ▼
Ações ▲
Executar instâncias ▼

Estado da inst... ▼
Tipo de inst... ▼

⊖ Interrompido ⊕ 🔍 t2.medium

Obter log do sistema
Obter captura de tela da instância
Gerenciar monitoramento detalhado
Gerenciar alarmes do CloudWatch

Conectar
Visualizar detalhes
Gerenciar estado da instância
Configurações de instância ▶
Redes ▶
Segurança ▶
Imagem e modelos ▶
Monitorar e solucionar problemas ▶

Zona
US-E

EC2 > Instâncias > i-0869e168642983e0f > Gerenciar monitoramento detalhado

Monitoramento detalhado [Informações](#)

Sua instância está habilitada para monitoramento básico por padrão. Como opção, você pode habilitar

ID de instância

 i-0869e168642983e0f (Principal)

Monitoramento detalhado

☒ Habilitar

AWS - CloudWatch

Detalhes

Segurança

Redes

Armazenamento


Verificações de status

Monitoramento

Tags

Adicionar ao painel

1h 3h 12h 1d 3d 1sem personalizado ▾

 ▾

Utilização de CPU (...)

Falha ao verificar st...

Falha ao verificar st...

Falha ao verificar st...

CloudWatch ×

Painéis

*** Infraestrutura**

▼ Alarmes

Alarme 0

Insufficient data 0

OK 1

Faturamento

▼ Logs

Grupos de logs

Insights

▼ Métricas

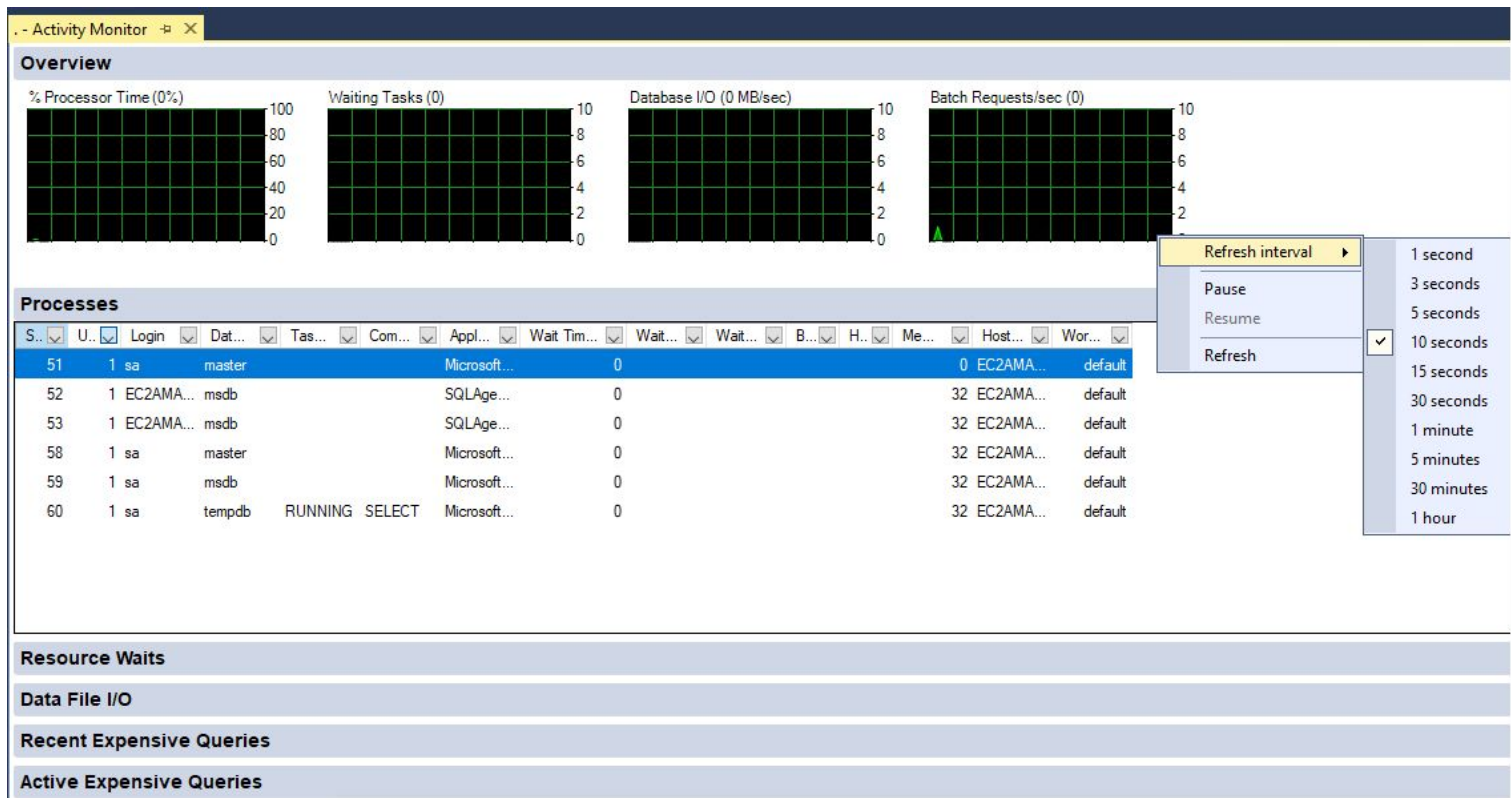
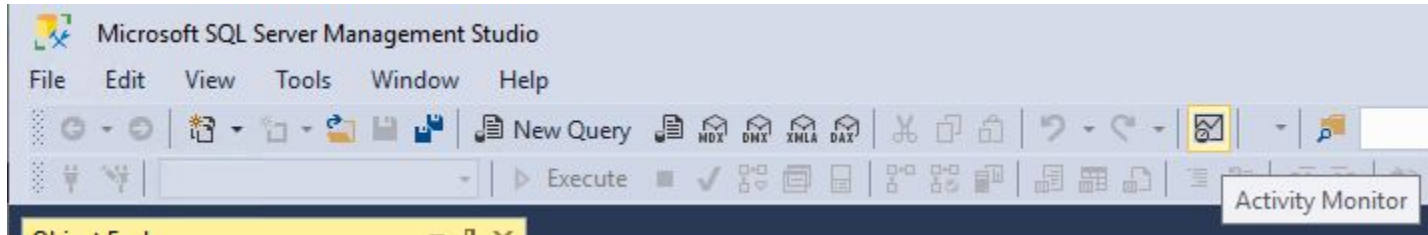
Explorer Nova

▼ Eventos

Regras

Barramentos de eventos

Monitor de Atividade - Activity Monitor



Procedures internas

SQLQuery1.sql - (local).master (sa (54))*

sp_who

236 %

Results Messages

	spid	ecid	status	loginame	hostname	blk	dbname	cmd	request_id
1	1	0	background	sa		0	NULL	XIO_LEASE_RENEWAL_WORKER	0
2	2	0	background	sa		0	NULL	XIO_RETRY_WORKER	0
3	3	0	background	sa		0	NULL	XTP_CKPT_AGENT	0

SQLQuery1.sql - (local).master (sa (54))*

sp_who2

236 %

Results Messages

	SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramName	SPID	REQUESTID
1	1	BACKGROUND	sa	.	.	NULL	XIO_LEASE_RENEWAL_WORKER	0	0	03/11 16:19:54		1	0
2	2	BACKGROUND	sa	.	.	NULL	XIO_RETRY_WORKER	15	0	03/11 16:19:54		2	0
3	3	BACKGROUND	sa	.	.	NULL	XTP_CKPT_AGENT	0	0	03/11 16:19:54		3	0

SQLQuery1.sql - (local).master (sa (54))*

sp_lock

236 %

Results Messages

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	52	4	0	0	DB		S	GRANT
2	53	4	0	0	DB		S	GRANT
3	54	32767	-571204656	0	TAB		Sch-S	GRANT

DMVs - Dynamic Management Views

Introduzidas no Microsoft SQL Server 2005, ajudam o administrador a terem uma visão completa do sistema.

DMVs para gerenciamento de CPU

-- working threads

```
SELECT scheduler_id,
       current_tasks_count,
       runnable_tasks_count
FROM   sys.dm_os_schedulers
WHERE  scheduler_id < 255
```

	scheduler_id	current_tasks_count	runnable_tasks_count
1	0	3	0
2	1	4	0
3	2	4	0
4	3	2	0

-- consumers

```
SELECT TOP 50 Sum(QS.total_worker_time) AS TOTAL_CPU_TIME,
              Sum(QS.execution_count)   AS TOTAL_EXECUTION_COUNT,
              Count(*)                   AS NUMBER_OF_STATEMENTS,
              SQL_TEXT.text,
              QS.plan_handle
FROM   sys.dm_exec_query_stats QS
CROSS apply sys.Dm_exec_sql_text(sql_handle) AS SQL_TEXT
GROUP BY SQL_TEXT.text,
         QS.plan_handle
ORDER BY Sum(QS.total_worker_time) DESC
```

	TOTAL_CPU_TIME	TOTAL_EXECUTION_COUNT	NUMBER_OF_STATEMENTS	TEXT	PLAN_HANDLE
1	20798052375	20234	1	(@CODIG...	0x06001300DB57C238E07E579EE20200000100000000000000...
2	9013884253	1598	1	(@0 int,@...	0x060013008082A104303E10A9230300000100000000000000...
3	2737370439	543	1	(@CODIG...	0x0600130094387334D0DD35EFF60200000100000000000000...
4	2410039614	25001	1	(@0 int,@...	0x060013003CAB7B2CB01A1D3ADA0200000100000000000000...

DMVs - Dynamic Management Views

DMVs para gerenciamento de Memória

-- allocation

```
SELECT type,
       Sum(pages_kb) AS pages_kb,
       ( 100.00 * Sum(pages_kb) )
       / Sum(Sum(pages_kb)) OVER() AS perc
FROM   sys.dm_os_memory_clerks
WHERE  pages_kb != 0
GROUP BY type
ORDER BY perc DESC
```

-- consumers

```
SELECT TOP 10 session_id,login_time,host_name,program_name,login_name,nt_domain,nt_user_name,status,
              cpu_time,memory_usage,total_scheduled_time,total_elapsed_time,last_request_start_time,
              last_request_end_time,reads,writes,logical_reads,transaction_isolation_level,
              lock_timeout,deadlock_priority,row_count,prev_error
FROM   sys.dm_exec_sessions
ORDER BY memory_usage DESC
```

	type	pages_kb	perc
1	MEMORYCLERK_SOSNODE	49608	22.203523345746204
2	MEMORYCLERK_SQLSTORENG	41072	18.382984818103695
3	MEMORYCLERK_SQLBUFFERPOOL	37296	16.692924663420223
4	MEMORYCLERK_SQLCLR	32600	14.591091377828702
5	MEMORYCLERK_SQLGENERAL	18272	8.178172443425952
6	USERSTORE_SCHEMAMGR	11616	5.199083357204239
7	USERSTORE_OBJPERM	3440	1.539673446004010
8	MEMORYCLERK_SQLQUERYPLAN	3296	1.475221999427098

session_id	login_time	host_name	program_name	login_name	nt_domain	nt_user_name	status	cpu_time	memory_usage	total_scheduled_time	total_elapsed_time
51	2021-03-11 14:09:51.000	DESKTOP-ISKKAB8	SQLServerCEIP	NT SERVICE\SQLTELEMETRY	NT SERVICE	SQLTELEMETRY	sleeping	0	4	1	1
59	2021-03-11 13:57:26.637	DESKTOP-ISKKAB8	Microsoft SQL Server Management Studio - Consulta	DESKTOP-ISKKAB8\gbmai	DESKTOP-ISKKAB8	gbmai	running	31	4	31	215
61	2021-03-11 13:57:23.027	DESKTOP-ISKKAB8	Microsoft SQL Server Management Studio	DESKTOP-ISKKAB8\gbmai	DESKTOP-ISKKAB8	gbmai	sleeping	0	4	19	31

DMVs - Dynamic Management Views

DMVs para gerenciamento de Disco

```
-- consumers
SELECT TOP 25 Db_name(D.database_id) AS
    DATABASE_NAME,
    Quotename(Object_schema_name(D.object_id, D.database_id))
    + N'.' AS
    OBJECT_NAME,
    D.database_id,
    D.object_id,
    D.page_io_latch_wait_count,
    D.page_io_latch_wait_in_ms,
    D.range_scans,
    D.index_lookups
FROM (SELECT database_id,
    object_id,
    Row_number()
    OVER (
        partition BY database_id
        ORDER BY Sum(page_io_latch_wait_in_ms) DESC) AS ROW_NUMBER,
    Sum(page_io_latch_wait_count) AS
    PAGE_IO_LATCH_WAIT_COUNT,
    Sum(page_io_latch_wait_in_ms) AS
    PAGE_IO_LATCH_WAIT_IN_MS,
    Sum(range_scan_count) AS RANGE_SCANS,
    Sum singleton_lookup_count) AS INDEX_LOOKUPS
FROM sys.dm_db_index_operational_stats(NULL, NULL, NULL, NULL)
WHERE page_io_latch_wait_count > 0
GROUP BY database_id,
    object_id) AS D
LEFT JOIN (SELECT DISTINCT database_id,
    object_id
    FROM sys.dm_db_missing_index_details) AS MID
ON MID.database_id = D.database_id
AND MID.object_id = D.object_id
WHERE D.row_number > 20
ORDER BY page_io_latch_wait_count DESC
```

	DATABASE_NAME	OBJECT_NAME	database_id	object_id
1	master	[sys].[sysscalartypes]	1	50
2	master	[sys].[sysmultiobjrefs]	1	75
3	master	[sys].[sysnsobjs]	1	44
4	master	[sys].[syssingleobjrefs]	1	74
5	master	[sys].[sysxsrvs]	1	43
6	master	[sys].[sysprfiles]	1	24
7	master	[sys].[sysrowsets]	1	5
8	master	[sys].[sysasymkeys]	1	95

range_scans	index_lookups
24	316
3101	0
20	0
121577	246536
1	1
7	2
233	525
10	122

DMVs - Dynamic Management Views

DMVs para gerenciamento de Disco

-- Database IO analysis.

```
WITH iofordatabase
AS (SELECT Db_name(VFS.database_id) AS DatabaseName,
CASE
WHEN smf.type = 1 THEN 'LOG_FILE'
ELSE 'DATA_FILE'
END AS DatabaseFile_Type,
Sum(VFS.num_of_bytes_written) AS IO_Write,
Sum(VFS.num_of_bytes_read) AS IO_Read,
Sum(VFS.num_of_bytes_read
+ VFS.num_of_bytes_written) AS Total_IO,
Sum(VFS.io_stall) AS IO_STALL
FROM sys.Dm_io_virtual_file_stats(NULL, NULL) AS VFS
JOIN sys.master_files AS smf
ON VFS.database_id = smf.database_id
AND VFS.file_id = smf.file_id
GROUP BY Db_name(VFS.database_id),
smf.type)
SELECT Row_number() OVER(ORDER BY io_stall DESC) AS RowNumber,
database_name,
databasefile_type,
Cast(1.0 * io_read / ( 1024 * 1024 ) AS DECIMAL(12, 2)) AS IO_Read_MB,
Cast(1.0 * io_write / ( 1024 * 1024 ) AS DECIMAL(12, 2)) AS IO_Write_MB,
Cast(1. * total_io / ( 1024 * 1024 ) AS DECIMAL(12, 2)) AS IO_TOTAL_MB,
Cast(io_stall / 1000. AS DECIMAL(12, 2)) AS
IO_STALL_Seconds,
Cast(100. * io_stall / Sum(io_stall) OVER() AS DECIMAL(10, 2)) AS IO_STALL_Pct
FROM iofordatabase
ORDER BY io_stall_seconds DESC;
```

	RowNumber	database_name	databasefile_type
1	1	tempdb	DATA_FILE
2	2	master	DATA_FILE
3	3	msdb	DATA_FILE

	IO_Read_MB	IO_Write_MB	IO_TOTAL_MB
1	23.83	3.59	27.41
2	21.89	0.91	22.80
3	34.50	0.01	34.51

DMVs - Dynamic Management Views

DMVs para tipos de espera

```
-- WAIT TYPES
SELECT TOP 10 [Wait type] = wait_type,
               [Wait time (s)] = wait_time_ms / 1000,
               [% waiting] = CONVERT(DECIMAL(12, 2), wait_time_ms * 100.0 / Sum(
                                   wait_time_ms)
                                   OVER()))
FROM sys.dm_os_wait_stats
WHERE wait_type NOT LIKE '%SLEEP%'
ORDER BY wait_time_ms DESC;
```

	Wait type	Wait time (s)	% waiting
1	SOS_WORK_DISPATCHER	3225592	74.01
2	LOGMGR_QUEUE	201842	4.63
3	CLR_AUTO_EVENT	201721	4.63
4	HADR_FILESTREAM_IOMGR_IOCOMPLETION	100922	2.32
5	DIRTY_PAGE_POLL	100921	2.32
6	REQUEST_FOR_DEADLOCK_SEARCH	100920	2.32
7	XE_TIMER_EVENT	100918	2.32
8	XE_DISPATCHER_WAIT	100837	2.31
9	ONDEMAND_TASK_QUEUE	100739	2.31
10	CHECKPOINT_QUEUE	87876	2.02

DMVs - Dynamic Management Views

DMVs para tipos de Locks

```

SELECT DISTINCT l.resource_type,
                l.resource_associated_entity_id,
                CASE
                    WHEN l.resource_type = 'OBJECT' THEN
                        Object_name(l.resource_associated_entity_id)
                    ELSE Object_name(sp.object_id)
                END AS
                resource_associated_entity_name,
                l.request_status,
                l.request_mode,
                request_session_id,
                l.resource_description,
                Substring(qt.text, ( r.statement_start_offset / 2 ) + 1,
                ( CASE
                    WHEN r.statement_end_offset = -1 THEN Len(
                        CONVERT(NVARCHAR(max), qt.text)) *
                        2
                    ELSE r.statement_end_offset
                END - r.statement_start_offset ) / 2) AS query_text
FROM sys.dm_tran_locks l
LEFT JOIN sys.partitions sp
    ON sp.hobt_id = l.resource_associated_entity_id
LEFT JOIN sys.dm_exec_requests r
    ON l.request_session_id = r.session_id
CROSS apply sys.Dm_exec_sql_text(r.sql_handle) AS qt
WHERE l.resource_database_id = Db_id()
ORDER BY 3,
        4

```

DMVs - Dynamic Management Views

DMVs para tipos de Blocks

```
SELECT DISTINCT t1.resource_type,
               t1.resource_database_id,
               t1.resource_associated_entity_id,
               CASE
                 WHEN t1.resource_type = 'OBJECT' THEN
                   Object_name(t1.resource_associated_entity_id)
                 ELSE Object_name(sp.object_id)
               END AS
               resource_associated_entity_name
               --,OBJECT_NAME(sp.OBJECT_ID) AS ObjectName
               ,
               t1.request_mode,
               t1.request_session_id,
               t2.blocking_session_id,
               Substring(qt.text, ( r.statement_start_offset / 2 ) + 1,
               ( CASE
                 WHEN r.statement_end_offset = -1 THEN Len(
                   CONVERT(NVARCHAR(max), qt.text)) *
                                   2
                 ELSE r.statement_end_offset
                 END - r.statement_start_offset ) / 2) AS query_text
FROM sys.dm_tran_locks AS t1
JOIN sys.dm_os_waiting_tasks AS t2
  ON t1.lock_owner_address = t2.resource_address
LEFT JOIN sys.partitions sp
  ON sp.hobt_id = t1.resource_associated_entity_id
LEFT JOIN sys.dm_exec_requests r
  ON t1.request_session_id = r.session_id
CROSS apply sys.Dm_exec_sql_text(r.sql_handle) AS qt
```

DMVs - Dynamic Management Views

DMVs para consultas em execução (minha popular sp_who3)

```
CREATE OR ALTER PROCEDURE Sp_who3 AS
SELECT      Sessions.session_id AS SessionID,
            Sessions.login_name AS LoginName,
            Sessions.host_name AS HostName,
            Sessions.program_name AS ProgramName,
            Sessions.client_interface_name AS ClientInterfaceName,
            Requests.wait_time AS WaitTime,
            Requests.cpu_time AS CPUTime,
            Requests.total_elapsed_time AS ElapsedTime,
            Requests.reads AS Reads,
            Requests.writes AS Writes,
            Requests.logical_reads AS LogicalReads,
            Requests.row_count AS [RowCount],
            Requests.granted_query_memory*8 AS GrantedQueryMemoryKB,
            CONVERT(BIGINT, (Requests.cpu_time+1))*CONVERT(BIGINT, (Requests.reads*10+Requests.writes*10+Requests.logical_reads*1)) AS Score,
            CONVERT(BIGINT, (Requests.cpu_time+1))*CONVERT(BIGINT, (Requests.reads*10+Requests.writes*10+Requests.logical_reads*1)) * 100.0 / Sum(CONVERT(BIGINT,
(Requests.cpu_time+1))*CONVERT(BIGINT, (Requests.reads*10+Requests.writes*10+Requests.logical_reads*1))) OVER() AS Score_perc,
            Statements.text AS BatchText,
            Len(Statements.text) AS BatchTextLength,
            Requests.statement_start_offset2 AS StatementStartPos
            CASE
                WHEN Requests.statement_end_offset = -1 THEN Len(CONVERT(NVARCHAR(max), Statements.text))*2
                ELSE Requests.statement_end_offset
            END/2 AS StatementEndPos (
            CASE
                WHEN Requests.statement_end_offset = -1 THEN Len(CONVERT(NVARCHAR(max), Statements.text))*2
                ELSE Requests.statement_end_offset
            END - Requests.statement_start_offset)/2 AS StatementTextLength
            CASE
                WHEN Requests.sql_handle IS NULL THEN ' '
                ELSE Substring( Statements.text, (Requests.statement_start_offset2)/2, (
                    CASE
                        WHEN Requests.statement_end_offset = -1 THEN Len(CONVERT(NVARCHAR(max), Statements.text))*2
                        ELSE Requests.statement_end_offset
                    END - Requests.statement_start_offset)/2 )
            END
            AS StatementText
FROM sys.dm_exec_sessions AS Sessions
JOIN sys.dm_exec_requests AS Requests
ON Sessions.session_id = Requests.session_id
CROSS apply sys.dm_exec_sql_text(sql_handle) AS Statements
CROSS apply sys.dm_exec_query_plan(plan_handle) AS QueryPlans
ORDER BY score DESC
go
```

Relatórios internos - SSMS

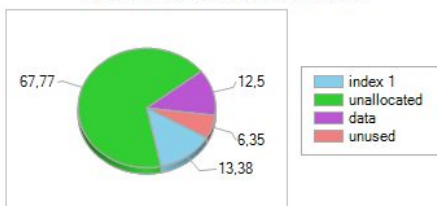
Novo Banco de Dados...
Nova Consulta
Script de Banco de Dados como ▶
Tarefas ▶
Políticas ▶
Facetas
Iniciar PowerShell
Azure Data Studio ▶
Relatórios ▶
Renomear
Excluir
Atualizar
Propriedades

Relatórios Padrão ▶
Relatórios Personalizados...
Uso do Disco

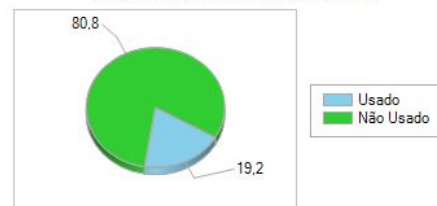
Uso do Disco
Classificação de dados
Uso do Disco por Tabelas Principais
Uso do Disco por Tabela
Uso do Disco por Partição
Eventos de Backup e Restauração
Todas as Transações
Todas as Transações de Bloqueio
Principais Transações por Idade
Principais Transações por Número de Transações Bloqueadas
Principais Transações por Número de Bloqueios
Estatísticas de Bloqueio de Recursos por Objetos
Estatísticas de Execução de Objeto
Histórico de Consistência do Banco de Dados
Uso de Memória por Objetos com Otimização de Memória
Visão Geral da Análise de Desempenho da Transação
Estatísticas de Uso do Índice
Estatísticas Físicas do Índice
Histórico de Alteração do Esquema
Estatísticas de Usuário

Espaço Total Reservado	16,00 MB
Espaço de Arquivos de Dados Reservado	8,00 MB
Espaço de Log de Transações Reservado	8,00 MB

Uso de Espaço dos Arquivos de Dados (%)

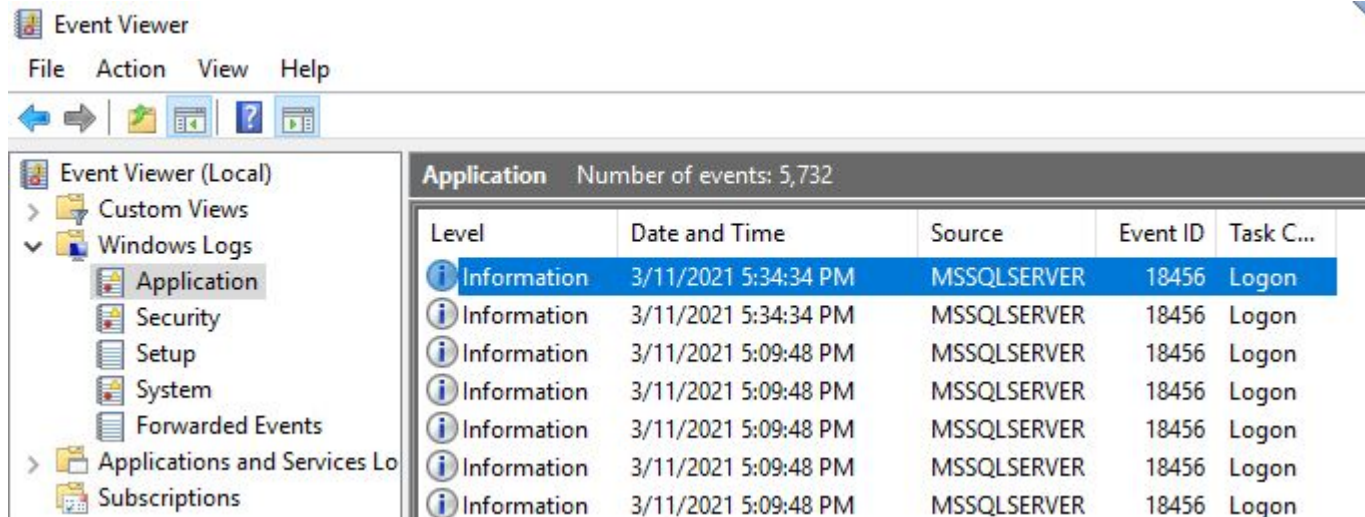
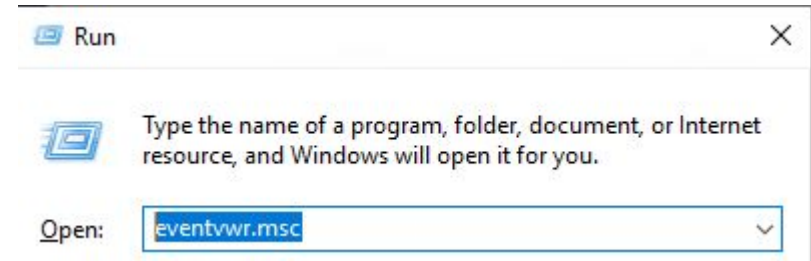


Uso de Espaço do Log de Transações (%)

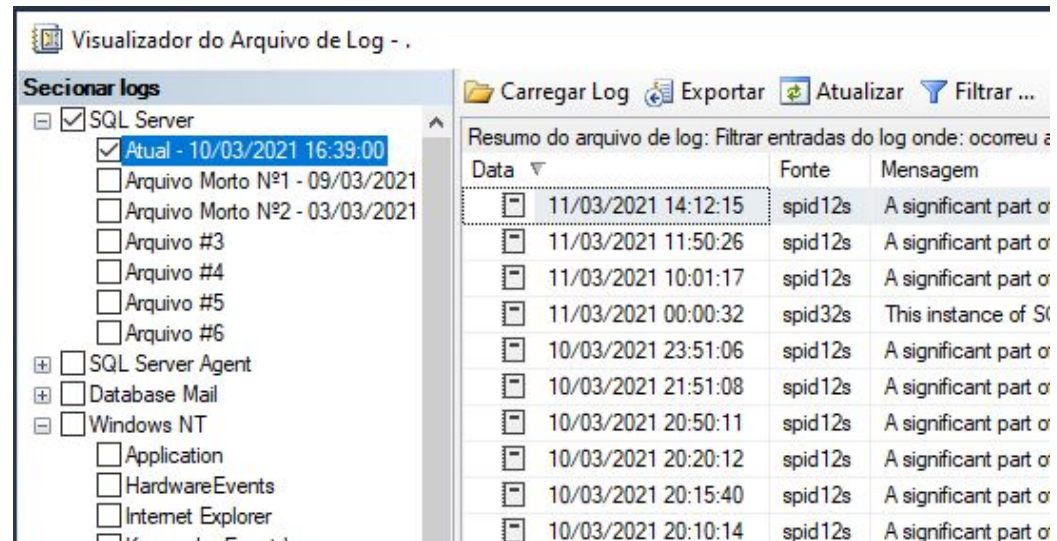
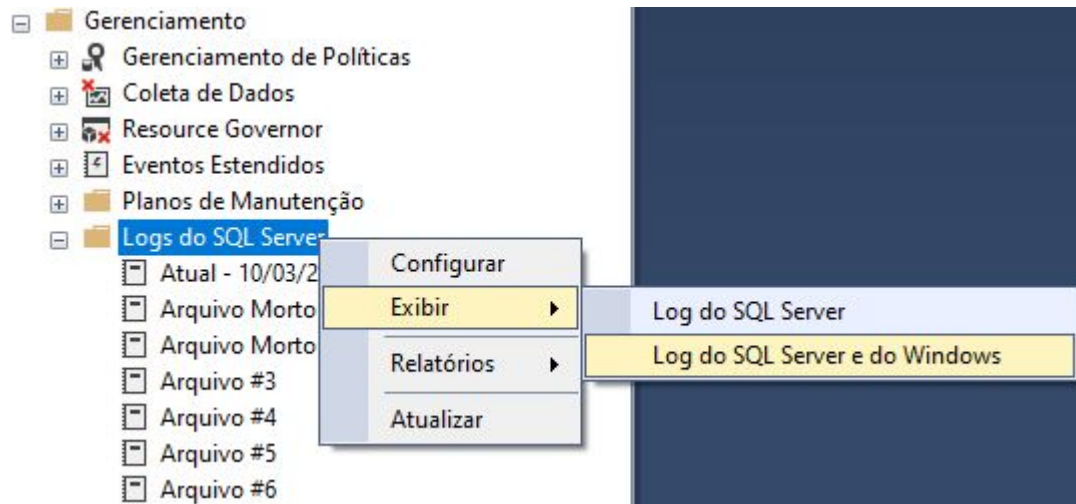


Visualizador de logs - Event viewer

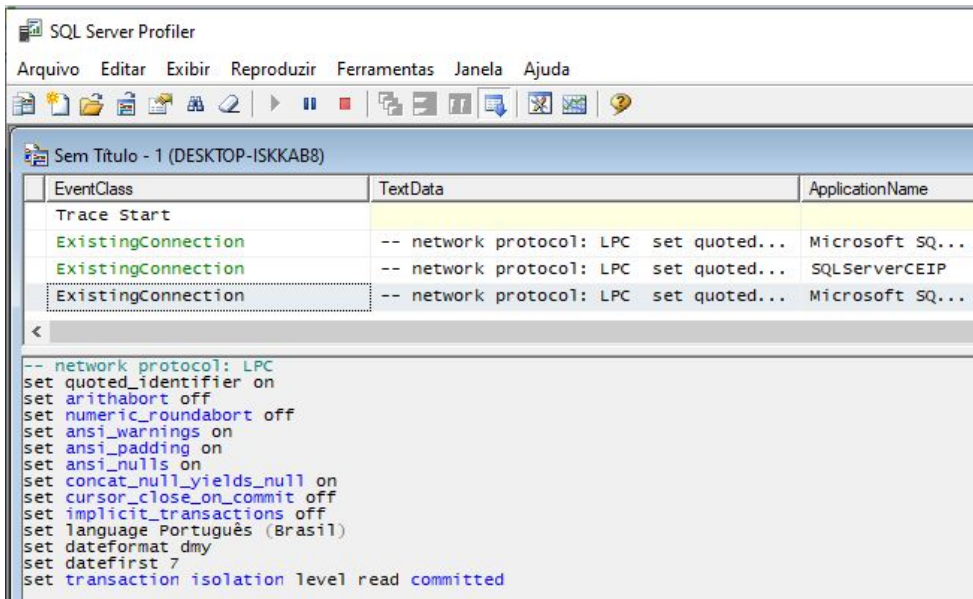
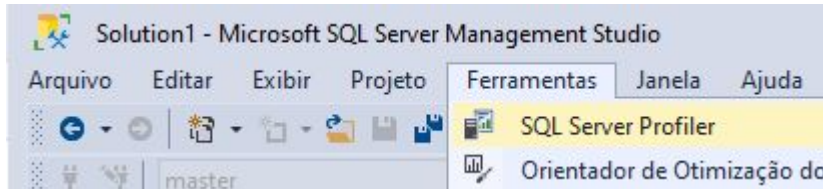
iniciar > executar > eventvwr.msc



Visualizador de logs - SSMS



SQL Server profiler



Propriedades do Rastreamento

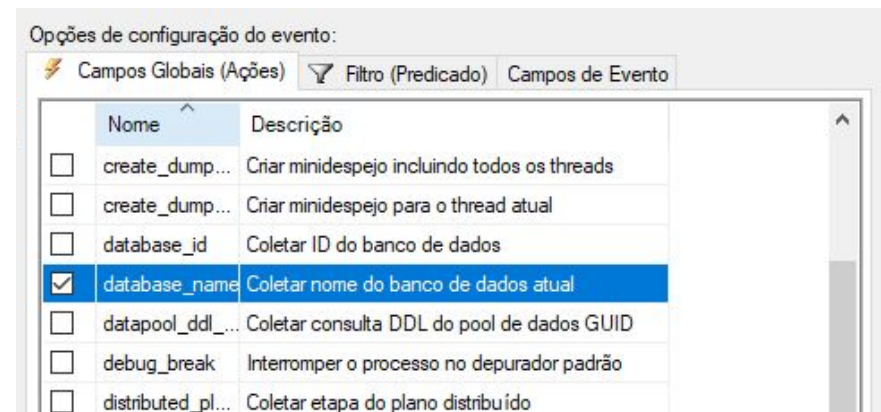
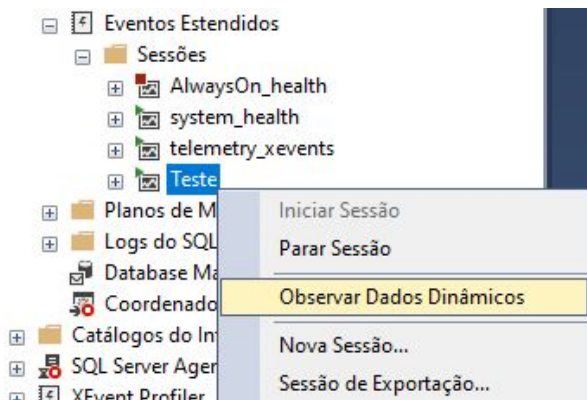
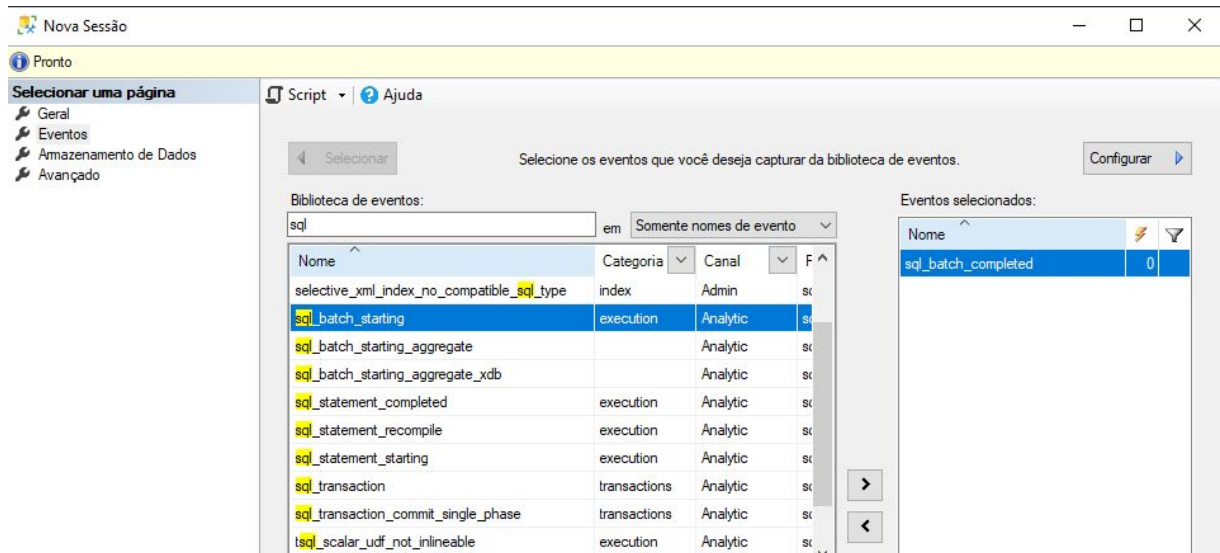
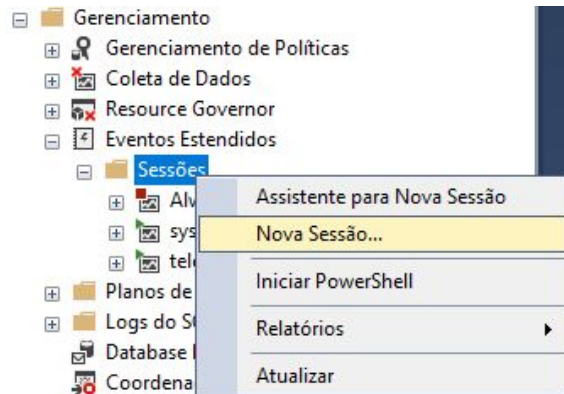
Geral Seleção de Eventos

Revise os eventos selecionados e as colunas de eventos a serem rastreados. Para v todas as colunas".

Eventos	TextData	ApplicationName	N
Security Audit			
<input checked="" type="checkbox"/> Audit Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> Audit Logout		<input checked="" type="checkbox"/>	
Sessions			
<input checked="" type="checkbox"/> ExistingConnection	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Stored Procedures			
<input checked="" type="checkbox"/> RPC:Completed	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
TSQL			
<input checked="" type="checkbox"/> SQL:BatchCompleted	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> SQL:BatchStarting	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

SQL Server Extended Events

Veio para substituir o SQL Profiler, é mais leve e poderoso.



SQL Server Agent

Utilizado para agendar tarefas, permite acompanhar sua execução ou gerar alertas específicos

SQL Server Agent

Trabalhos

Monitor de Atividade do Trabalho

Alertas

Operadores

Proxies

Logs de Erros

XEvent Profiler

Monitor de Atividade do Trabalho - .

Selecionar uma página

Todos os Trabalhos

Atualizar

Filtrar ...

Ajuda

Atividade do Trabalho do Agente:

Nome	Habilitado	Status	Resulta...	Última ...	Próxima...	Categoria	Execut
Backup Database	sim	Ocioso	Falha	23/09/...	não ag...	[Não c...	sim
Backup LOG	não	Ocioso	Êxito	23/09/...	11/03/...	[Não c...	sim
MaintenancePlan.S...	sim	Ocioso	Êxito	23/04/...	não ag...	Manut...	sim
Restore database	sim	Ocioso	Desco...	nunca	não ag...	[Não c...	sim
syspolicy_purge_his...	sim	Ocioso	Êxito	10/10/...	12/03/...	[Não c...	sim

Visualizador do Arquivo de Log - .

Carregar Log

Exportar

Atualizar

Filtrar ...

Pesquisar ...

Parar

Resumo do arquivo de log: Nenhum filtro aplicado

Data	ID da Etapa	Servidor	Nome do Trabalho	Nome
23/09/2020 22:15:59		DESKTOP-ISKKAB8	Backup Database	
23/09/2020 22:15:59	1	DESKTOP-ISKKAB8	Backup Database	Back
23/09/2020 21:44:28		DESKTOP-ISKKAB8	Backup Database	
23/09/2020 21:41:03		DESKTOP-ISKKAB8	Backup Database	
23/09/2020 21:39:36		DESKTOP-ISKKAB8	Backup Database	

Detalhes da linha selecionada:

Data

23/09/2020 22:15:59

Log

Histórico do Trabalho (Backup Database)

ID da Etapa

1

Servidor

DESKTOP-ISKKAB8

Nome do Trabalho

Backup Database

Nome da Etapa

Backup Database

Duração

00:00:01

SQL Server Agent - alertas

Utilizado para agendar tarefas, permite acompanhar sua execução ou gerar alertas específicos

Nome: ☒ Habilitar

Tipo: Alerta de evento do SQL Server

Definição de alerta de evento

Nome do banco de dados: <todos os bancos de dados>

Os alertas serão gerados com base em:

☐ Número do erro:

☒ Severidade: 001 - Diversas Informações do Sistema

☐ Gerar alerta quando a mensagem contiver:

Texto da mensagem:

Nome: ☒ Habilitar

Tipo: Alerta de condição de desempenho do SQL Server

Definição de alerta de condição de desempenho

Objeto: General Statistics

Contador: Processes blocked

Instância:

Alertar se o contador

cai abaixo
toma-se igual a
acima

Valor:

Triggers de sistema

Existem inúmeros eventos que podem ser capturados. Lista de eventos:

[Provedor WMI para classes e propriedades de eventos de servidor - SQL Server | Microsoft Docs](#)

```
...
SELECT Getdate(),
       CONVERT(NVARCHAR(100), CURRENT_USER),
       CONVERT(NVARCHAR(100), SYSTEM_USER),
       CONVERT(NVARCHAR(100), Host_name()),
       CONVERT(NVARCHAR(100), @ip),
       Db_name(),
       @data.value('(/EVENT_INSTANCE/ObjectName)[1]', 'NVARCHAR(100)'),
       @data.value('(/EVENT_INSTANCE/PostTime)[1]', 'DATETIME'),
       @data.value('(/EVENT_INSTANCE/EventType)[1]', 'NVARCHAR(100)'),
       @data.value('(/EVENT_INSTANCE/SPID)[1]', 'INT'),
       @data.value('(/EVENT_INSTANCE/ServerName)[1]', 'NVARCHAR(100)'),
       @data.value('(/EVENT_INSTANCE/LoginName)[1]', 'NVARCHAR(100)'),
       @data.value('(/EVENT_INSTANCE/UserName)[1]', 'NVARCHAR(100)'),
       @data.value('(/EVENT_INSTANCE/ObjectType)[1]', 'NVARCHAR(100)'),
       @data.value('(/EVENT_INSTANCE/TSQLCommand)[1]', 'NVARCHAR(100)')
...

```

	Server Scope	Database Scope
DDL_SERVER_LEVEL_EVENTS (CREATE DATABASE, ALTER DATABASE, DROP DATABASE)	X	
DDL_ENDPOINT_EVENTS (CREATE ENDPOINT, ALTER ENDPOINT, DROP ENDPOINT)	X	
DDL_SERVER_SECURITY_EVENTS	X	
DDL_LOGIN_EVENTS (CREATE LOGIN, ALTER LOGIN, DROP LOGIN)	X	
DDL_GDR_SERVER_EVENTS (GRANT SERVER, DENY SERVER, REVOKE SERVER)	X	
DDL_AUTHORIZATION_SERVER_EVENTS (ALTER AUTHORIZATION SERVER)	X	
DDL_DATABASE_LEVEL_EVENTS		X
DDL_TABLE_VIEW_EVENTS		X
DDL_TABLE_EVENTS (CREATE TABLE, ALTER TABLE, DROP TABLE)		X
DDL_VIEW_EVENTS (CREATE VIEW, ALTER VIEW, DROP VIEW)		X
DDL_INDEX_EVENTS (CREATE INDEX, ALTER INDEX, DROP INDEX, CREATE XML INDEX)		X
DDL_STATISTICS_EVENTS (CREATE STATISTICS, UPDATE STATISTICS, DROP STATISTICS)		X
DDL_SYNONYM_EVENTS (CREATE SYNONYM, DROP SYNONYM)		X
DDL_FUNCTION_EVENTS (CREATE FUNCTION, ALTER FUNCTION, DROP FUNCTION)		X
DDL_PROCEDURE_EVENTS (CREATE PROCEDURE, ALTER PROCEDURE, DROP PROCEDURE)		X



Obrigado !

Gustavo Maia
Gustavo.Maia@FaculdadeImpacta.com.br