**Sql Server Health Check List**

When we are checking the sql sever we need to consider the so many things

From the Basics

1.Check the all the services

Server,Agent,Browser

exec master.dbo.xp\_servicecontrol 'QUERYSTATE', 'MSSQLServer'

exec master.dbo.xp\_servicecontrol 'QUERYSTATE', 'SQLServerAgent'

exec master.dbo.xp\_servicecontrol 'QUERYSTATE', 'SQLBrowser'

### 2. Did all of your SQL Agent Jobs run successfully?

By using sql server agent and queries

use msdb

go

select 'FAILED' as Status, cast(sj.name as varchar(100)) as "Job Name",

cast(sjs.step\_id as varchar(5)) as "Step ID",

cast(sjs.step\_name as varchar(30)) as "Step Name",

cast(REPLACE(CONVERT(varchar,convert(datetime,convert(varchar,sjh.run\_date)),102),'.','-')+' '+SUBSTRING(RIGHT('000000'+CONVERT(varchar,sjh.run\_time),6),1,2)+':'+SUBSTRING(RIGHT('000000'+CONVERT(varchar,sjh.run\_time),6),3,2)+':'+SUBSTRING(RIGHT('000000'+CONVERT(varchar,sjh.run\_time),6),5,2) as varchar(30)) 'Start Date Time',

sjh.message as "Message"

from sysjobs sj

join sysjobsteps sjs

on sj.job\_id = sjs.job\_id

join sysjobhistory sjh

on sj.job\_id = sjh.job\_id and sjs.step\_id = sjh.step\_id

where sjh.run\_status <> 1

and cast(sjh.run\_date as float)\*1000000+sjh.run\_time >

cast(convert(varchar(8), getdate()-1, 112) as float)\*1000000+70000 --yesterday at 7am

union

select 'FAILED',cast(sj.name as varchar(100)) as "Job Name",

'MAIN' as "Step ID",

'MAIN' as "Step Name",

cast(REPLACE(CONVERT(varchar,convert(datetime,convert(varchar,sjh.run\_date)),102),'.','-')+' '+SUBSTRING(RIGHT('000000'+CONVERT(varchar,sjh.run\_time),6),1,2)+':'+SUBSTRING(RIGHT('000000'+CONVERT(varchar,sjh.run\_time),6),3,2)+':'+SUBSTRING(RIGHT('000000'+CONVERT(varchar,sjh.run\_time),6),5,2) as varchar(30)) 'Start Date Time',

sjh.message as "Message"

from sysjobs sj

join sysjobhistory sjh

on sj.job\_id = sjh.job\_id

where sjh.run\_status <> 1 and sjh.step\_id=0

and cast(sjh.run\_date as float)\*1000000+sjh.run\_time >

cast(convert(varchar(8), getdate()-1, 112) as float)\*1000000+70000 --yesterday at 7am

### 3 - Do you have a recent backup of all your SQL Server databases?

### The queries are used to retrive the failed the Backups on last day

Full backup

SELECT d.name AS "Database",

ISNULL(CONVERT(VARCHAR,b.backupdate,120),'NEVER') AS "Last Full Backup"

FROM sys.databases d

LEFT JOIN (SELECT database\_name,type,MAX(backup\_finish\_date) backupdate FROM backupset

WHERE type LIKE 'D'

GROUP BY database\_name,type) b on d.name=b.database\_name

WHERE (backupdate IS NULL OR backupdate < getdate()-1)

**Log backup**

SELECT d.name AS "Database",

ISNULL(CONVERT(VARCHAR,b.backupdate,120),'NEVER') AS "Last Log Backup"

FROM sys.databases d

LEFT JOIN (SELECT database\_name,type,MAX(backup\_finish\_date) backupdate FROM backupset

WHERE type LIKE 'L'

GROUP BY database\_name,type) b on d.name=b.database\_name

WHERE recovery\_model = 1

AND (backupdate IS NULL OR backupdate < getdate()-1)

Differential backups

SELECT d.name AS "Database",

ISNULL(CONVERT(VARCHAR,b.backupdate,120),'NEVER') AS "Last Differentail Backup"

FROM sys.databases d

LEFT JOIN (SELECT database\_name,type,MAX(backup\_finish\_date) backupdate FROM backupset

WHERE type LIKE 'I'

GROUP BY database\_name,type) b on d.name=b.database\_name

WHERE (backupdate IS NULL OR backupdate < getdate()-1)

### 4 - Are there any errors in your SQL Server Error Log?

Max 2 days

declare @Time\_Start datetime;

declare @Time\_End datetime;

set @Time\_Start=getdate()-2;

set @Time\_End=getdate();

-- Create the temporary table

CREATE TABLE #ErrorLog (logdate datetime

, processinfo varchar(255)

, Message varchar(500))

-- Populate the temporary table

INSERT #ErrorLog (logdate, processinfo, Message)

EXEC master.dbo.xp\_readerrorlog 0, 1, null, null , @Time\_Start, @Time\_End, N'desc';

-- Filter the temporary table

SELECT LogDate, Message FROM #ErrorLog

WHERE (Message LIKE '%error%' OR Message LIKE '%failed%') AND processinfo NOT LIKE 'logon'

ORDER BY logdate DESC

-- Drop the temporary table

DROP TABLE #ErrorLog

You need to check the xp\_readerrorlog 0,1,null,null,’date’,’date’,’desc’

Sp\_readerrorlog 1,1

### 5.Are you running out of space on any of your disks on your SQL Server?

### Xp\_fixeddrives

### Better to create alert on system

### 6 - Are you running low on server memory for SQL Server?

SELECT available\_physical\_memory\_kb/1024 as "Total Memory MB",

available\_physical\_memory\_kb/(total\_physical\_memory\_kb\*1.0)\*100 AS "% Memory Free"

FROM sys.dm\_os\_sys\_memory

### It is better to have than 300MB

### 7 - Are there any SQL Server statements in the cache that could use tuning?

### It is mainly check the sql queries work on long running queries and slow running queries and Expensive Queries by cpu time max work threads and logical read and physical reads

SELECT top 10 text as "SQL Statement",

last\_execution\_time as "Last Execution Time",

(total\_logical\_reads+total\_physical\_reads+total\_logical\_writes)/execution\_count as [Average IO],

(total\_worker\_time/execution\_count)/1000000.0 as [Average CPU Time (sec)],

(total\_elapsed\_time/execution\_count)/1000000.0 as [Average Elapsed Time (sec)],

execution\_count as "Execution Count",

qp.query\_plan as "Query Plan"

FROM sys.dm\_exec\_query\_stats qs

CROSS APPLY sys.dm\_exec\_sql\_text(qs.plan\_handle) st

CROSS APPLY sys.dm\_exec\_query\_plan(qs.plan\_handle) qp

order by total\_elapsed\_time/execution\_count desc

### 8 - How many connections do you have to your SQL Server instance?

Sp\_who2

Sp\_whoisactive

Sys.dm\_ecex\_requests

Number of connections

SELECT spid, kpid, blocked, d.name, open\_tran, status, hostname,

cmd, login\_time, loginame, net\_library

FROM sys.sysprocesses p

INNER JOIN sys.databases d

on p.dbid=d.database\_id

### 9 - How many requests is your SQL Server processing?

Batch request for sec if it is above or around 1000/sec than your sever very busy server

DECLARE @BRPS BIGINT

SELECT @BRPS=cntr\_value

FROM sys.dm\_os\_performance\_counters

WHERE counter\_name LIKE 'Batch Requests/sec%'

WAITFOR DELAY '000:00:10'

SELECT (cntr\_value-@BRPS)/10.0 AS "Batch Requests/sec"

FROM sys.dm\_os\_performance\_counters

WHERE counter\_name LIKE 'Batch Requests/sec%'

10.Check the blocking sessions

select \* from sysprocesses where blocked<>0

better to check with sp\_who2 ,you will get more information

SELECT spid ,sp.STATUS ,loginame = SUBSTRING(loginame, 1, 12) ,hostname = SUBSTRING(hostname, 1, 12)

,blk = CONVERT(CHAR(3), blocked) ,open\_tran ,dbname = SUBSTRING(DB\_NAME(sp.dbid),1,10)

,cmd ,waittype ,waittime ,last\_batch ,SQLStatement =

SUBSTRING

(qt.text,

er.statement\_start\_offset/2,

(CASE WHEN er.statement\_end\_offset = -1

THEN LEN(CONVERT(nvarchar(MAX), qt.text)) \* 2

ELSE er.statement\_end\_offset

END - er.statement\_start\_offset)/2 )

FROM master.dbo.sysprocesses sp

LEFT JOIN sys.dm\_exec\_requests er ON er.session\_id = sp.spid

OUTER APPLY sys.dm\_exec\_sql\_text(er.sql\_handle) AS qt

WHERE spid IN (SELECT blocked FROM master.dbo.sysprocesses)

AND blocked = 0

11.Open transaction

dbcc opentran()

SELECT

[s\_tst].[session\_id],

[s\_es].[login\_name] AS [Login Name],

DB\_NAME (s\_tdt.database\_id) AS [Database],

[s\_tdt].[database\_transaction\_begin\_time] AS [Begin Time],

[s\_tdt].[database\_transaction\_log\_bytes\_used] AS [Log Bytes],

[s\_tdt].[database\_transaction\_log\_bytes\_reserved] AS [Log Rsvd],

[s\_est].text AS [Last T-SQL Text],

[s\_eqp].[query\_plan] AS [Last Plan]

FROM

sys.dm\_tran\_database\_transactions [s\_tdt]

JOIN

sys.dm\_tran\_session\_transactions [s\_tst]

ON

[s\_tst].[transaction\_id] = [s\_tdt].[transaction\_id]

JOIN

sys.[dm\_exec\_sessions] [s\_es]

ON

[s\_es].[session\_id] = [s\_tst].[session\_id]

JOIN

sys.dm\_exec\_connections [s\_ec]

ON

[s\_ec].[session\_id] = [s\_tst].[session\_id]

LEFT OUTER JOIN

sys.dm\_exec\_requests [s\_er]

ON

[s\_er].[session\_id] = [s\_tst].[session\_id]

CROSS APPLY

sys.dm\_exec\_sql\_text ([s\_ec].[most\_recent\_sql\_handle]) AS [s\_est]

OUTER APPLY

sys.dm\_exec\_query\_plan ([s\_er].[plan\_handle]) AS [s\_eqp]

ORDER BY

[Begin Time] ASC;

GO

12.Chck the locks

Sp\_lock

select \* from sys.dm\_tran\_locks

by this querie you will get the Locked Object (Table)

SELECT

OBJECT\_NAME(p.[object\_id]) BlockedObject

FROM sys.dm\_exec\_connections AS blocking

INNER JOIN sys.dm\_exec\_requests blocked

ON blocking.session\_id = blocked.blocking\_session\_id

INNER JOIN sys.dm\_os\_waiting\_tasks waitstats

ON waitstats.session\_id = blocked.session\_id

INNER JOIN sys.partitions p ON SUBSTRING(resource\_description,

PATINDEX('%associatedObjectId%', resource\_description) + 19,

LEN(resource\_description)) = p.partition\_id

By this query you will get the Table name and which type of lock means page or rid or object level lock

SELECT

OBJECT\_NAME(p.OBJECT\_ID) AS TableName,

resource\_type, resource\_description

FROM

sys.dm\_tran\_locks l

JOIN sys.partitions p ON l.resource\_associated\_entity\_id = p.hobt\_id

We need to study on locks more like isolation level(ACID properties)

Locks type:Page locks..

Or shared locks and intent lock or Exclusive locks ...

13.Index Maintenance:

This index maintanace we need to do every 15 day s

By estimate the index usages liks index stats likse by using index and how may reads and wirte and updates happen .

Suppose by using index less read operations occures ,then you need to check the after 15 days same thing happen check with the developer and delete because indexes are prevent the write operation indexes are used to improve the select operations only .

SELECT \* FROM sys.dm\_db\_index\_physical\_stats (DB\_ID(N'<DatabaseName>'), OBJECT\_ID(N'<ObjectName>'), NULL, NULL , 'SAMPLED');

And reads and writes

SELECT OBJECT\_NAME(s.[object\_id]) AS [ObjectName], i.name AS [IndexName], i.index\_id,

user\_seeks + user\_scans + user\_lookups AS [Reads], s.user\_updates AS [Writes],

i.type\_desc AS [IndexType], i.fill\_factor AS [FillFactor], i.has\_filter, i.filter\_definition,

s.last\_user\_scan, s.last\_user\_lookup, s.last\_user\_seek

FROM sys.dm\_db\_index\_usage\_stats AS s WITH (NOLOCK)

INNER JOIN sys.indexes AS i WITH (NOLOCK)

ON s.[object\_id] = i.[object\_id]

WHERE OBJECTPROPERTY(s.[object\_id],'IsUserTable') = 1

AND i.index\_id = s.index\_id

AND s.database\_id = DB\_ID()

ORDER BY user\_seeks + user\_scans + user\_lookups DESC OPTION (RECOMPILE); -- Order by reads

--s.user\_updates DESC OPTION (RECOMPILE); -- Order by writes

Fragementation you will create maintanace plan for indexes like rebuild and reorganization but it is better to use script because if you create maintance plan for rebuil index all indexes are rebuild that is not nessary you need to rebuild index for frag>30 only ,it is better to go for >50

The index processes id Not ended process

14.Update stats

After index and fragementation Update stats are impartant

Update stats are date to date

SELECT

ss.name AS SchemaName

, st.name AS TableName

, s.name AS IndexName

, STATS\_DATE(s.id,s.indid) AS 'Statistics Last Updated'

, s.rowcnt AS 'Row Count'

, s.rowmodctr AS 'Number Of Changes'

, CAST((CAST(s.rowmodctr AS DECIMAL(28,8))/CAST(s.rowcnt AS DECIMAL(28,2)) \* 100.0)

AS DECIMAL(28,2)) AS '% Rows Changed'

FROM sys.sysindexes s

INNER JOIN sys.tables st ON st.[object\_id] = s.[id]

INNER JOIN sys.schemas ss ON ss.[schema\_id] = st.[schema\_id]

WHERE s.id > 100

AND s.indid > 0

AND s.rowcnt >= 500

ORDER BY SchemaName, TableName, IndexName

### Here you can consider the row changes .

### 15.Check DB

### It is helps to check the database Health

### 16.Hardware