Monitoring Microsoft SQL Azure Database Server

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: HOW DOES EG ENTERPRISE MONITOR MICROSOFT SQL AZURE DATABASE SERVER? 2

CHAPTER 3: HOW TO MONITOR MICROSOFT SQL AZURE DATABASE SERVER USING EG ENTERPRISE? 3

3.1 Managing the Microsoft SQL Azure Database Server 3

3.2 Configuring Tests 5

CHAPTER 4: MONITORING THE MICROSOFT SQL AZURE DATABASE SERVER USING EG ENTERPRISE 7

4.1 The SQL Azure Server Layer 9

4.1.1 SQL Azure Copy Status Test 10

4.1.2 SQL Azure Resource Statistics Test 15

4.1.3 SQL Azure Wait Statistics Test 17

4.2 The SQL Azure Memory Layer 20

4.2.1 SQL Azure Memory Grants Test 20

4.2.2 SQL Azure Query Plans Test 22

4.2.3 SQL Azure Root Blockers Test 24

4.3 The SQL Azure Database Layer 27

4.3.1 SQL Azure Database Size Test 28

4.3.2 SQL Azure DB Status Test 31

4.3.3 SQL Azure Missing Indexes Test 39

4.3.4 SQL Azure Top Tables Test 42

4.4 The SQL Azure Service Layer 44

4.4.1 SQL Azure Network Test 45

4.4.2 SQL Azure Sessions Test 48

4.4.3 SQL Azure Top Procedures Test 51

4.4.4 SQL Azure Top Queries Test 54

ABOUT EG INNOVATIONS 58

Table of Figures

Figure 3.1: Managing a Microsoft SQL Azure database server 4
Figure 3.2: The list of unconfigured tests for Microsoft SQL Azure database server 5
Figure 3.3: Configuring the SQL Azure Copy Status test 5
Figure 4.1: The layer model of the target Microsoft SQL Azure database server 7
Figure 4.2: The tests associated with the SQL Azure Server layer 10
Figure 4.3: The tests associated with the SQL Azure Memory layer 20
Figure 4.4: The tests associated with the SQL Azure Database layer 27
Figure 4.5: The tests associated with the SQL Azure Service layer 44

Chapter 1: Introduction

SQL Azure database is a managed cloud database (SaaS) provided as part of Microsoft Azure. Azure SQL database is the intelligent, fully managed relational cloud database service that provides the broadest SQL Server engine compatibility, so that you can migrate your SQL Server databases without changing your apps.

Any performance degradation or unavailability of the database servers can severely impact the performance of the entire service, often causing customer dissatisfaction and lost business revenue. Continuous monitoring of the SQL Azure database servers are hence imperative. This is where the eG Enterprise helps database administrators!

Chapter 2: How Does eG Enterprise Monitor Microsoft SQL Azure Database Server?

eG Enterprise can monitor Microsoft SQL Azure database server in an agentless manner only. The remote agent used to monitor the Microsoft SQL Azure database server should be deployed on a remote Windows host in the environment.

Chapter 3: How to Monitor Microsoft SQL Azure Database Server Using eG Enterprise?

The broad steps for monitoring the Microsoft SQL Azure database server using eG Enterprise are as follows:

- Managing the Microsoft SQL Azure database server
- Configuring the tests

These steps have been discussed in the forthcoming sections.

3.1 Managing the Microsoft SQL Azure Database Server

The Microsoft SQL Azure database server cannot be automatically discovered by eG Enterprise. This implies that you will have to manually add the server into the eG Enterprise system to manage it. Follow the steps below to achieve the same:

1. Follow the Components -> Add/Modify menu sequence in the **Infrastructure** tile of the **Admin** menu.
2. Next, select *Microsoft SQL Azure* from the **Component type** drop-down and then click the **Add New Component** button.
3. When Figure 3.1 appears, provide the **Host IP/Name** of the Microsoft SQL Azure database server that you want to manage.

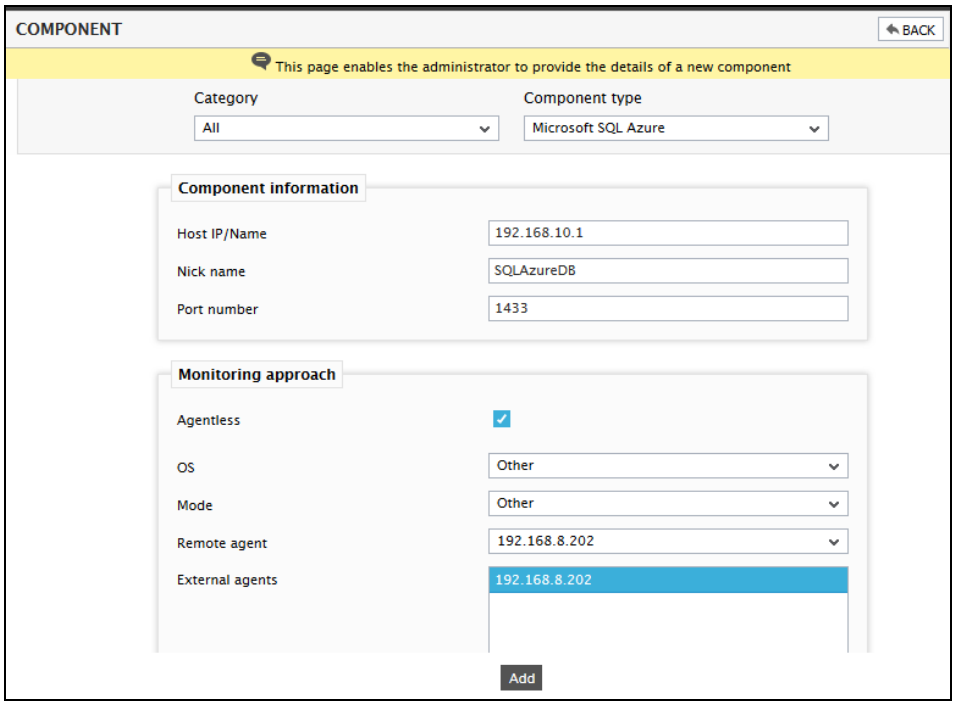

Figure 3.1: Managing a Microsoft SQL Azure database server

4. Then, provide a **Nick name** for the server.
5. The **Port number** will be set as 27017 by default. If the Microsoft SQL Azure database server is listening on a different port in your environment, then override this default setting.
6. On the other hand, if you are monitoring a Microsoft SQL Azure database server in an agentless manner, then do the following:
   - Select the **Agentless** check box.
   - Pick the **OS** on which the Microsoft SQL Azure database server is running.
   - Set the **Mode** to **Other**.
   - Select the **Remote agent** that will be monitoring the Microsoft SQL Azure database server. **Note that the Remote agent you choose should run on a Windows host.**
   - Choose an external agent for the server by picking an option from the **External agents** list box.
   - Finally, click the **Add** button to add the Microsoft SQL Azure database server for monitoring.
7. Finally, click the **Signout** button at the right, top corner of the eG admin interface to sign out.

3.2 Configuring Tests

When you try to sign out of the eG admin interface, a **LIST OF UNCONFIGURED TESTS** page will appear, revealing the list of tests mapped to the Microsoft SQL Azure database server that require manual configuration.

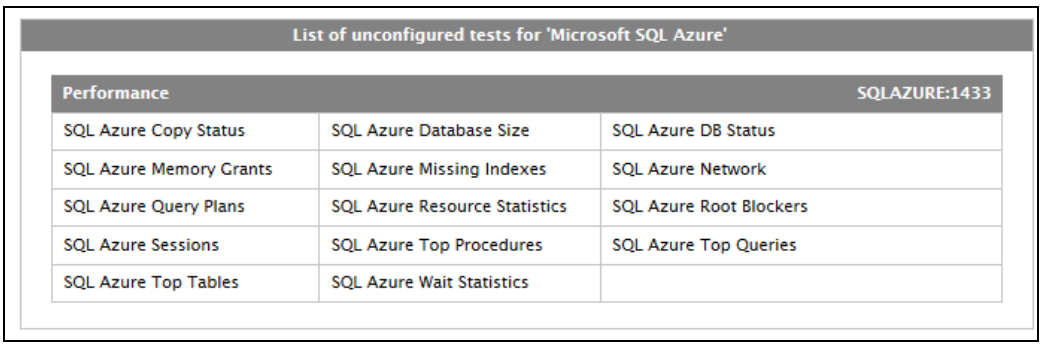

Figure 3.2: The list of unconfigured tests for Microsoft SQL Azure database server

Click the **SQL Azure Copy Status** test to configure it. Figure 3.3 then appears.

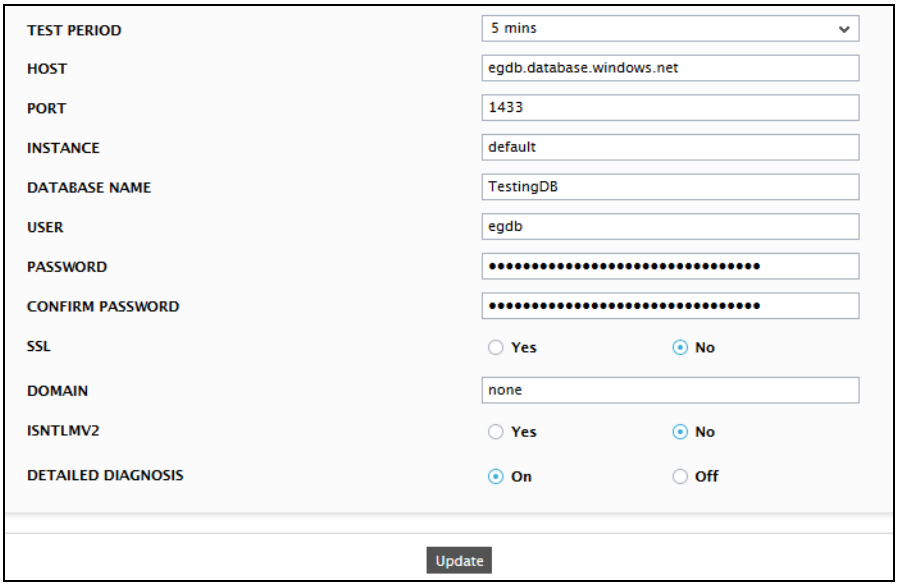

Figure 3.3: Configuring the SQL Azure Copy Status test

To know how to configure the test, refer to **Monitoring the Microsoft SQL Azure Database Server Using eG Enterprise**. Once this test is configured, all other tests will be configured automatically.

Finally, sign out of the eG administrative interface. Then, login to the eG monitoring console to view the state of and metrics reported by the specialized monitoring model that eG Enterprise offers for the Microsoft SQL Azure database server.

Chapter 4: Monitoring the Microsoft SQL Azure Database Server Using eG Enterprise

The pre-built Microsoft SQL Azure database server monitoring model that eG Enterprise offers (see Figure 4.1), provides in-depth monitoring for the Microsoft SQL Azure database servers.

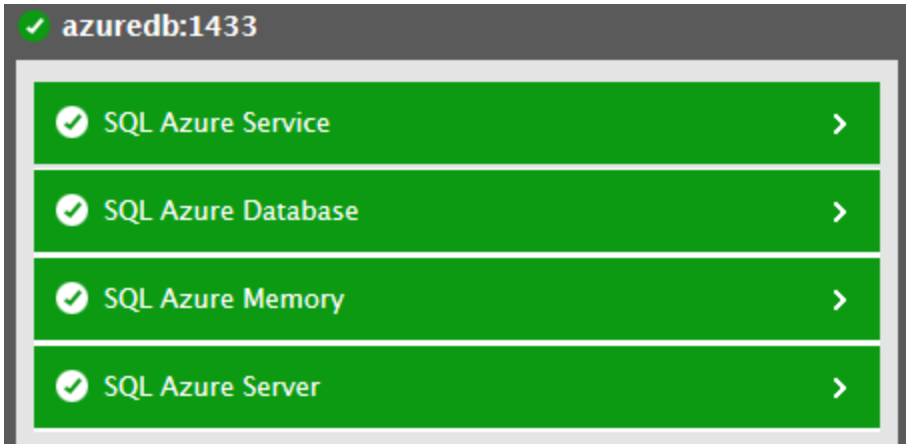

Figure 4.1: The layer model of the target Microsoft SQL Azure database server

Each of the layers of this hierarchical model reports a wide variety of metrics ranging from the basic operating system-level statistics to individual database related measurements to those indicating the database engine health. Using these measurements , administrators can easily answer the following questions:

- What is the availability and responsiveness of the target Microsoft SQL Azure database server?
- How long does the target database server take for query execution?
- How many sessions of each user are currently open on the database instance?
- How many sessions of each user are currently running on the database instance?
- How many sessions of each user are blocked on the database instance?
- How many sessions of each user were rolled back?
- What is the maximum CPU time taken per execution of the SQL procedure on the target database instance?
- What is the maximum time elapses per execution of the SQL procedure on the target database instance?

- What is the maximum amount of data spill that occurred per execution of the SQL procedure on the target database instance?
- What is the maximum CPU time taken per execution of the SQL query on the target database instance?
- What is the maximum time elapses per execution of the SQL query on the target database instance?
- What is the maximum amount of data spill that occurred per execution of the SQL query on the target database instance?
- What is the space allocated for each file type on the target database instance?
- What is the amount of space allocated for each file type on the target database instance?
- How much of space is available for the files of each file type on the target database instance?
- What is the growth rate of the files of each file type?
- How well the space was utilized by the files of each file type?
- What is the current state of each database instance?
- Is the database instance in standby mode/READ_ONLY mode/user access mode?
- How many indexes are found to be missing in the queries that are currently executing on the database instance?
- How many times the resources were impacted due to user queries when the indexes were missing?
- How many tables are currently oversized i.e., the tables were larger than the size configured?
- What is the size of the table among the tables that are oversized?
- What is the maximum amount of memory consumed by a query executing on the database instance?
- How many queries were waiting for memory on the database instance?
- What is the maximum time for which the query had to wait for memory?
- What is the maximum query cost based on how much memory the query will use on the database instance?
- What is the maximum amount of memory consumed by each query plan?
- What is the total number of root blocker processes on the database instance?

- How many processes are blocked by the root blockers?
- Is geo-replication feature enabled on the database instance?
- Is the database instance a primary database?
- What is the current status of replication on the database instance?
- What is the mode of the secondary database?
- What is the percentage of DTU utilized?
- What is the current maximum database DTU setting in percentage for the database instance?
- What is the maximum concurrent workers (requests) in percentage of the limit of the service tier of the target database instance?
- What is the maximum concurrent sessions in percentage of the limit of the service tier of the target database instance?
- What is the maximum time taken for a wait of each task type?
- What is the total signal wait time (across wait types) during which wait events of each type waited for a signal?
- How many tasks of each type were waiting on the target database instance?

Each layer of Figure 4.1 has been discussed in detail in the forthcoming sections.

4.1 The SQL Azure Server Layer

Using the tests mapped to this layer, you can determine:

- whether/not geo-replication is enabled;
- whether/not the target database instance is a primary database instance;
- the mode of the secondary database;
- the replication status of the database;
- how effectively the Microsoft SQL Azure database server utilizes the sessions and process resources it is configured with;
- the number, nature, and duration of waits etc.

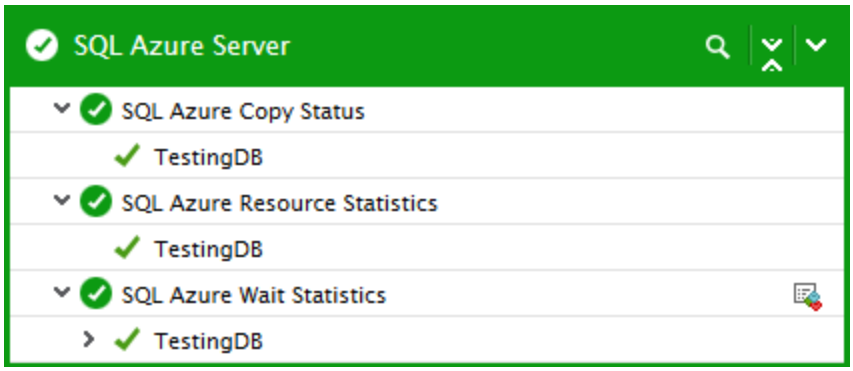

Figure 4.2: The tests associated with the SQL Azure Server layer

4.1.1 SQL Azure Copy Status Test

Replication provides redundancy and increases data availability. With multiple copies of data on different database servers, replication provides a level of fault tolerance against the loss of a single database server.

Active geo-replication enables you to configure up to four readable secondary databases in the same or different data center locations (regions). Secondary databases are available for querying and for failover if there is a data center outage or the inability to connect to the primary database. The failover must be initiated manually by the application of the user. After failover, the new primary has a different connection end-point. Azure SQL Database auto-failover groups is a SQL Database feature designed to automatically manage geo-replication relationship, connectivity, and failover at scale. With it, the customers gain the ability to automatically recover multiple related databases in the secondary region after catastrophic regional failures or other unplanned events that result in full or partial loss of the SQL Database service's availability in the primary region. Additionally, they can use the readable secondary databases to offload read-only workloads. If you are using active geo-replication and for any reason your primary database fails, or simply needs to be taken offline, you can initiate failover to any of your secondary databases. When failover is activated to one of the secondary databases, all other secondaries are automatically linked to the new primary. If swift failover does not occur, then, there may be a lag in data transfer to the secondary databases. This may lead to the unavailability of data to the secondary databases which would eventually hamper the replication process. Therefore, it is necessary to check whether the geo-replication feature is enabled, and if enabled, it becomes mandatory to check the status of the primary and secondary databases and the status of replication to the secondary databases. The **SQL Azure Copy Status** test helps administrators in this regard!

This test auto-discovers the database instances on the target Microsoft SQL Azure database server and for each database instance, reports whether/not geo-replication is enabled. If geo-replication is

enabled, this test reports whether the database instance is a primary or secondary database and also reports the current status of replication.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for every Microsoft SQL Azure database server instance being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be none. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the

Parameter	Description
	managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 ("NTLMv2") was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to No, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to Yes if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to Yes. If not, then set the SSL flag to No.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Is Geo-replication enabled?	Indicates whether/not the geo-replication feature is enabled on this database instance.		The values reported by this measure and its numeric equivalents are mentioned in the table below: **Measure Value** / **Numeric Value** Yes – 1 No – 0 **Note:** By default, this measure reports the **Measure Value**s listed in the table above to indicate whether/not the geo-replication feature is enabled on this database instance. The graph of this measure however, represents the status of a server using the numeric equivalents only.
Is Primary?	Indicates whether/not this database instance is the primary database.		The values reported by this measure and its numeric equivalents are mentioned in the table below:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Yes</td><td>1</td></tr>
<tr><td>No</td><td>0</td></tr>
</table>

Note: By default, this measure reports the Measure Values listed in the table above to indicate whether/not this database instance is the primary database. However, the graph of this measure is represented using its corresponding numeric equivalents only - 0 or 1.
</td></tr>
<tr><td>Is Secondary?</td><td>Indicates whether/not this database instance is the secondary database.</td><td></td><td>The values reported by this measure and its numeric equivalents are mentioned in the table below:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Yes</td><td>1</td></tr>
<tr><td>No</td><td>0</td></tr>
</table>

Note:
By default, this measure reports the Measure Values listed in the table above to indicate whether/not this database instance is the secondary database. However, the graph of this measure is represented using its corresponding numeric equivalents only - 0 or 1.
</td></tr>
<tr><td>Replication Status</td><td>Indicates the current status of replication on this database instance.</td><td></td><td>The values reported by this measure and its numeric equivalents are mentioned in the table below:</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Seeding</td><td>1</td></tr>
<tr><td>Catch up</td><td>2</td></tr>
<tr><td>Pending</td><td>3</td></tr>
</table>
Note:
By default, this measure reports the Measure Values listed in the table above to indicate the current status of replication on this database instance. However, the graph of this measure is represented using its corresponding numeric equivalents - 1 to 3.
</td></tr>
<tr><td>Replication lagging</td><td>Indicates the time duration in seconds, that replication to this database instance lags.</td><td>Seconds</td><td>A high value for this measure is a cause of concern. If the value of this measure is high, then, it may indicate that the databases are out of sync and data cannot be accessed from the databases when a failover occurs.</td></tr>
<tr><td>Secondary database mode</td><td>Indicates the mode of the secondary database.</td><td></td><td>The values reported by this measure and its numeric equivalents are mentioned in the table below:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>No</td><td>0</td></tr>
<tr><td>All</td><td>1</td></tr>
</table>
Note:
By default, this measure reports the Measure Values listed in the table above to indicate the mode of the secondary database. However, the graph of this measure is represented using its corresponding numeric equivalents only - 0 or 1.
</td></tr>
</table>

4.1.2 SQL Azure Resource Statistics Test

This test monitors how effectively the Microsoft SQL Azure database server utilizes the session and process resources it is configured with. If the maximum limit to which the resource allocation can grow is violated, it is bound to deteriorate the performance of the server, as the server might not have the bandwidth to handle the additional sessions/processes. Similarly, a potential DTU (Database Transaction Unit) limit breach by the database server can be proactively captured and averted.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for the target Microsoft SQL Azure database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.

Parameter	Description
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be *none*. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 (“NTLMv2”) was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Average CPU usage	Indicates the average compute utilization in percentage of the limit of the service tier of the database instance.	Percent	
Average data IO usage	Indicates the average I/O utilization in percentage based on the limit of the service tier of the database instance.	Percent	
Average memory usage	Indicates the average memory utilization in percentage of the limit of the service tier of the database instance.	Percent	

Measurement	Description	Measurement Unit	Interpretation
Average of log write	Indicates the average resource utilized in percentage for writes of the limit of the service tier of the database instance.	Percent	
Current DTU limit	Indicates the current maximum database DTU setting in percentage for the database instance during this interval.	Percent	
DTU usage	Indicates the percentage of DTU utilized by the database instance.	Percent	
Workers usage	Indicates the maximum concurrent workers (requests) in percentage of the limit of the service tier of the database instance.	Percent	
Sessions usage	Indicates the maximum concurrent sessions in percentage of the limit of the service tier of the database instance.	Percent	

4.1.3 SQL Azure Wait Statistics Test

In Microsoft SQL Azure database server, wait types represent the discrete steps in query processing, where a query waits for resources as the instance completes the request. By analyzing wait types and their wait times, administrators can receive quick and objective evidence of performance bottlenecks and their probable causes. The **SQL Azure Wait Statistics** test enables this analysis. For every type of wait that is currently experienced by the server, this test reports the number, nature, and duration of waits, thereby leading you to the specific wait types that may have contributed to a general slowdown / deterioration in server performance.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for every type of wait on the target Microsoft SQL Azure Database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, *none* is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be *none*. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 (“NTLMv2”) was concocted to address the security issues

Parameter	Description
	present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Tasks maximum wait time	Indicates the maximum time taken for a wait of this task type during the last measurement period.	Seconds	Use the detailed diagnosis of this measure to figure out the tables that are consuming the maximum disk space.
Tasks signal wait time	Indicates the total signal wait time (across wait types) during which wait events of this type waited for a signal.	Seconds	The signal wait is the time between when a worker has been granted access to the resource and the time it gets scheduled on the CPU. A high value for this measure may imply a high CPU contention. To know which wait type registered the highest signal wait time and probably caused the CPU contention, compare the value of this measure across wait types.
Tasks wait time	Indicates the total wait time (across wait types) during which wait events of this type occurred during the last measurement period.	Seconds	When a user complaints that query execution takes too long, you can compare the value of this measure across wait types to know which type of wait is the key contributor to delays in query processing.
Waiting tasks	Indicates the number of waits of this task type during the last measurement period.	Number	This counter is incremented at the start of each wait.

4.2 The SQL Azure Memory Layer

This layer tracks the memory consumed by the SQL queries, the maximum time the SQL query had to wait for memory, the count of root blockers, the maximum amount of memory consumed by each query plan etc.

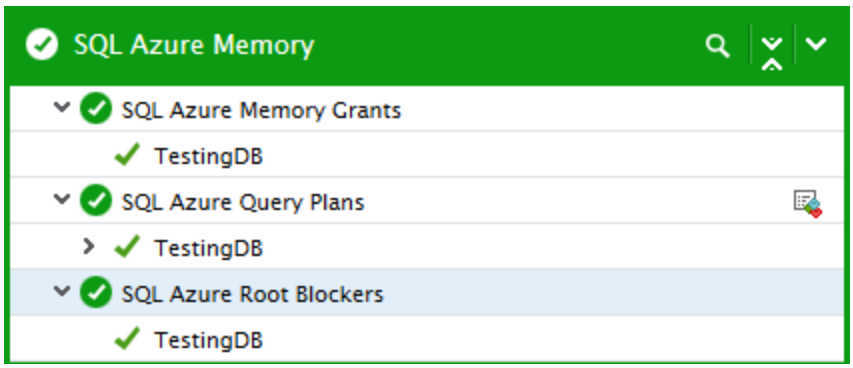

Figure 4.3: The tests associated with the SQL Azure Memory layer

4.2.1 SQL Azure Memory Grants Test

If the memory resource allocation is not done properly on the target Microsoft SQL Azure database server, query execution may get delayed due to the insufficient memory resources. The users may also experience delayed response to their queries and also the performance of the Microsoft SQL Azure database server may degrade drastically. To avoid such unpleasant situation, administrators should continuously monitor the memory utilization to accurately find out whether the memory issues are responsible for slow down in the query execution. This can be achieved using the **SQL Azure Memory Grants** test.

By closely monitoring the memory resources of the target Microsoft SQL Azure database server, this test reports the maximum amount of memory consumed by the query executing on the server and the number of queries that were waiting for memory. This test also reveals the query cost based on how much memory the query is about to utilize and the maximum time for which the query has to wait for memory. Using this test, administrators can easily pinpoint which query is waiting too long for execution.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for the target Microsoft SQL Azure database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be none. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 (“NTLMv2”) was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to No, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to Yes if NTLMv2 is enabled on the target host.

Parameter	Description
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to Yes. If not, then set the SSL flag to No.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Maximum memory consumed by query	Indicates the maximum amount of memory consumed by a query executing on this database.	MB	
Queries waiting for memory	Indicates the number of queries waiting for memory in this database.	Number	
Maximum query cost by memory	Indicates the maximum query cost based on how much memory the query will use on this database.	Seconds	The cost of an execution plan is just an estimate based on how many CPU and I/O resources that the query optimizer estimates that query will use. A high value for this measure is a cause of concern.
Maximum wait duration for memory	Indicates the maximum time for which the query had to wait for memory in this database.	Seconds	A high value for this measure is a cause of concern as this may sometimes indicate that the database is inaccessible.

4.2.2 SQL Azure Query Plans Test

When SQL Azure server executes a query it uses a query plan to determine how to access data and complete the query. SQL server offers administrators and developers the ability to see these query plans to assist in query optimization. Without query plans it would be difficult to figure out how to tune and optimize the queries. The Query Execution Plans describe the steps and the order used to access or modify data in the Microsoft SQL Azure database server. The Query Plan defines how SQL statements are physically executed by the server. The Query Plan describes the data retrieval and storage methods that are used by the Query Optimizer to execute a specific query. The executed Query Plans are also stored in the Procedure Cache, so they can be retrieved and reused

if a similar query is executed. If the Query Plan in the stored Procedure Cache consumes too much of memory, more query plans cannot be stored in the cache. This would delay the execution of the query plans on the SQL Azure databases. Therefore, it is necessary to monitor the memory consumed by each query plan in the cache. The **SQL Azure Query Plans** test helps administrators in this regard!

This test auto-discovers the query plans executed on the target Microsoft SQL Azure database server, and reports the maximum amount of memory consumed by each query plan.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for every Microsoft SQL Azure database server instance being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.

Parameter	Description
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be none. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 (“NTLMv2”) was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to No, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to Yes if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to Yes. If not, then set the SSL flag to No.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Maximum memory consumed query plan	Indicates the maximum amount of memory consumed by this query plan.	MB	Use the detailed diagnosis of this measure to figure out the query plans that consumed the maximum amount of memory.

4.2.3 SQL Azure Root Blockers Test

One common problem encountered with databases is blocking. Suppose that process A is modifying data that process B wants to use. Process B will be blocked until process A has completed what it is doing. This is only one type of blocking situation; others exist and are common. What matters to a database administrator is identifying when blocking is a problem and how to deal with it effectively. When blocking is bad enough, users will notice slowdowns and complain about it. With a large number of users, it is common for tens or hundreds of processes to be blocked when slowdowns are noticed. Killing these processes may or may not solve the problem because 10 processes may be blocked by process B, while process B itself is blocked by process A. Issuing 10 kill statements for the processes blocked by B probably will not help, as new processes will simply become blocked by

B. Killing process B may or may not help, because then the next process that was blocked by B, which is given execution time, may get blocked by process A and become the process that is blocking the other 9 remaining processes. When you have lots of blocking that is not resolving in a reasonable amount of time you need to identify the root blocker, or the process at the top of the tree of blocked processes. Imagine again that you have 10 processes blocked by process B, and process B is blocked by process A. If A is not blocked by anything, but is itself responsible for lots of blocking (B and the 10 processes waiting on B), then A would be the root blocker. (Think of it as a traffic jam.) Killing A (via kill) is likely to unblock B, and once B completes, the 10 processes waiting on B are also likely to complete successfully.

This test monitors the number of root blocker processes in the target Microsoft SQL Azure database server. In addition, this test also helps administrators to the number of processes that are blocked by the root blockers and the maximum time for which the processes were blocked.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for every Microsoft SQL Azure database server instance being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:**

Parameter	Description
	Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be *none*. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 ("NTLMv2") was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Root blockers	Indicates the number of root blocker processes.	Number	Usually, the number of root blocker processes should be low. If this value increases suddenly, this is a cause for concern. Likewise, if a root-blocker process has been blocking other processes for a long time, it is a reason for further investigation. The detailed diagnosis for this test, if enabled, provides details of the root blocker processes - their SPIDs, programs running these processes, and the queries being issued by these

Measurement	Description	Measurement Unit	Interpretation
			processes. It is usually the case that killing any root-blocker process that has been running for a long while will get the database running well again.
Blocked processes	Indicates the number of processes that are blocked by the root blockers.	Number	Use the detailed diagnosis of this measure to know which processes are blocked.
Maximum waiting time	Indicates the waiting time - i.e., blocked time - of that process(es) that was blocked for the maximum duration.	Secs	If the value of this measure matches or exceeds the MAX BLOCKING TIME configuration of this test, it indicates that one/more processes have been blocked for a very long time. You can then use the detailed diagnosis of this measure to identify these blocked processes and figure out who initiated such processes and their resource usage. Processes that are resource hogs can thus be identified.

4.3 The SQL Azure Database Layer

The test associated with this layer (see 4.3) monitors the status, space utilization, missing indexes and the number of top tables on the target Microsoft SQL Azure database server.

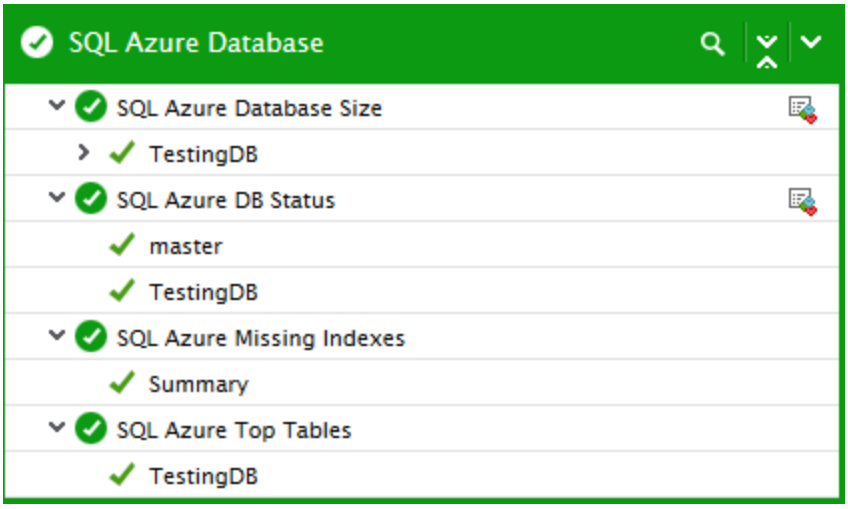

Figure 4.4: The tests associated with the SQL Azure Database layer

4.3.1 SQL Azure Database Size Test

Periodic monitoring of the usage of the files of each file type in the database is essential to ensure that the files of the database are always adequately sized to handle current and future loads. The **SQL Azure Database Size** test monitors the usage of the files of the target Microsoft SQL Azure database server, and indicates if it requires resizing.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for each file type on the target Microsoft SQL Azure database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.

Parameter	Description
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be none. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 (“NTLMv2”) was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to No, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to Yes if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Allocated size	Indicates the space allocated for this file type.	GB	
Free space on maximum	Indicates the amount of disk space that is currently available for use by the files of this file type.	GB	A high value implies that the database files have adequate space for growth. If the value of this measure is low, then you may have to fine-tune your auto-growth settings accordingly.
Percentage free space	Indicates the percentage of Max file size that is currently available for use by the files of this file type.	Percent	If many files of this file type are set to auto-grow till disk capacity is reached, then, a high value of this measure indicates that there is enough space for the files to grow. A low value indicates that there is very little room for file growth.
Growth rate	Indicates the growth rate of the files of this file type.	MB/sec	

Measurement	Description	Measurement Unit	Interpretation
Maximum size	Indicates the maximum size upto which the database files of this file type can grow.	GB	Each database file that is associated with your database has an auto-growth setting. There are three different settings you can use to identify how your database files will grow. They can grow by a specific size, a percentage of the current size, or not grow at all. Additionally you can set your files to unrestricted growth, which means they will keep growing as they need more space or you run out of disk space. Or you can restrict the growth of a database file to grow no larger than a specified size. Each one of these different auto-grow settings have defaults, or you can set them for each database file. If the auto-growth setting is not enabled at all for a file in a file group, then the amount of space that was originally allocated to that file will be regarded as the Max file size of that file. On the other hand, if the Auto-growth setting is enabled for a file in the file group, then the Max file size of that file will be one of the following: • If a specific size limit is explicitly set for the file, then this will be considered as the Max file size of that file. • If no size limit is set for the file, then the total capacity of the disk drive in which that file resides will be considered as the Max file size of

Measurement	Description	Measurement Unit	Interpretation
			that file. So, if a file group consists of a few data files for which auto-growth is enabled and a few others for which it is disabled, then the Max file size of that file group will be a sum total of the following: • The sum of the space allocated to each of the files for which auto-growth is not enabled; • The sum of the maximum size limits, if defined, for each file for which auto-growth is disabled; • The sum of the total capacity of the disks containing the auto-growth-enabled files for which no size limit is defined.
Used size	Indicates the amount of space that was already utilized by the files of this file type.	GB	A high value for this measure indicates that the space in the files are depleting at a faster pace.

4.3.2 SQL Azure DB Status Test

If a user complaints of problems while accessing a database, the knowledge of the current state of that database and the mode using which users are allowed to access the database will enable administrators to promptly diagnose the reason for such an occurrence. This test auto-discovers all the database instances on the target Microsoft SQL Azure database server and reports the current state of each database instance and the user access mode thereby enabling administrators to easily troubleshoot issues related to database access.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for each database instance of the target Microsoft SQL Azure database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be none. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 (“NTLMv2”) was concocted to address the

Parameter	Description
	security issues present in NTLM. By default, the ISNTLMNV2 flag is set to No, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to Yes if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Status</td><td>Indicates the current state of this database instance.</td><td></td><td>The states that can be reported by this measure, the numeric value that corresponds to each state, and a brief description of the state are provided below:
<table>
<tr><th>Numeric Value</th><th>State</th><th>Description</th></tr>
<tr><td>0</td><td>ONLINE</td><td>Database is available for access. The primary filegroup is online, although the undo phase of recovery may not have been completed.</td></tr>
<tr><td>1</td><td>RESTORING</td><td>One or more files of the primary filegroup are being restored, or one or more secondary files are being</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Numeric Value</th><th>State</th><th>Description</th></tr>
<tr><td></td><td></td><td>restored off-line. The database is unavailable in this case.</td></tr>
<tr><td>2</td><td>RECOVERING</td><td>Database is being recovered. The recovering process is a transient state; the database will automatically become online if the recovery succeeds. If the recovery fails, the database will become suspect. The database is unavailable in this case.</td></tr>
<tr><td>3</td><td>RECOVERY_PENDING</td><td>SQL Server has encountered a resource-related error during recovery. The database is not damaged, but files may be missing or system</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Numeric Value</th><th>State</th><th>Description</th></tr>
<tr><td></td><td></td><td>resource limitations may be preventing it from starting. The database is unavailable. Additional action by the user is required to resolve the error and let the recovery process be completed.</td></tr>
<tr><td>4</td><td>SUSPECT</td><td>At least the primary filegroup is suspect and may be damaged. The database cannot be recovered during startup of SQL Server. The database is unavailable. Additional action by the user is required to resolve the problem.</td></tr>
<tr><td>5</td><td>EMERGENCY</td><td>User has</td></tr>
</table>
</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			<table><tr><th>Numeric Value</th><th>State</th><th>Description</th></tr><tr><td></td><td></td><td>changed the database and set the status to EMERGENCY. The database is in single-user mode and may be repaired or restored. The database is marked READ_ONLY, logging is disabled, and access is limited to members of the sysadmin fixed server role. EMERGENCY is primarily used for troubleshooting purposes. For example, a database marked as suspect can be set to the EMERGENCY state. This could permit the system administrator</td></tr></table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Numeric Value</th><th>State</th><th>Description</th></tr>
<tr><td></td><td></td><td>access to the database. Only members of the sysadmin fixed server role can set a database to the EMERGENCY state.</td></tr>
<tr><td>6</td><td>OFFLINE</td><td>Database is unavailable. A database becomes offline by explicit user action and remains offline until additional user action is taken. For example, the database may be taken offline in order to move a file to a new disk. The database is then brought back online after the move has been completed.</td></tr>
</table>
The detailed diagnosis of this measure reports the user access mode of the
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>database, the database recovery model, and the log re-use wait state of the database.
Note:
By default, this measure reports the States listed in the table above to indicate the current status of the database instance. The graph of this measure however, represents the status of the database using the numeric equivalents only i.e., 0 to 6 only.</td></tr>
<tr><td>Is in standby?</td><td>Indicates whether/not this database instance is in standby mode.</td><td></td><td>The values reported by this measure and its numeric equivalents are mentioned in the table below:
<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>No</td><td>0</td></tr><tr><td>Yes</td><td>1</td></tr></table>Note:
By default, this measure reports the Measure Values listed in the table above to indicate whether/not the database instance is in standby mode. The graph of this measure however, is represented using the numeric equivalents only i.e., 0 or 1.</td></tr>
<tr><td>Is read only?</td><td>Indicates whether/not this database instance is in READ_ONLY mode.</td><td></td><td>The values reported by this measure and its numeric equivalents are mentioned in the table below:
<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>No</td><td>0</td></tr><tr><td>Yes</td><td>1</td></tr></table>Note:
By default, this measure reports the Measure Values listed in the table above to</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>indicate whether/not the database instance is in READ_ONLY mode. The graph of this measure however, is represented using the numeric equivalents only i.e., 0 or 1.</td></tr>
<tr><td>User access mode</td><td>Indicates the user access mode to this database instance.</td><td></td><td>The states that can be reported by this measure, the numeric value that corresponds to each state, and a brief description of the state are provided below:
<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Multi user</td><td>1</td></tr><tr><td>Single user</td><td>2</td></tr><tr><td>Restricted user</td><td>3</td></tr></table>
Note:
By default, this measure reports the Measure Values listed in the table above to indicate the user access mode to this database instance. The graph of this measure however, represents the status of the database using the numeric equivalents only i.e., 1 to 3 only.</td></tr>
</table>

4.3.3 SQL Azure Missing Indexes Test

Microsoft SQL Azure database server allows you to put indexes on table columns, to speed up WHERE and JOIN statements on those columns. If a SQL query takes longer (much longer) to complete, it could be because one/more of these indexes are 'missing'. When the query optimizer optimizes a query, it identifies those indexes it would have liked to have used but were not available - these are called 'missing indexes'. With the help of the **SQL Azure Missing Indexes** test, you can be promptly alerted when the query optimizer finds one/more 'missing indexes'. Besides reporting the count of the missing indexes, the test also reveals which queries require these indexes, thus enabling you to quickly initiate index creation and query optimization.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for each database instance of the target Microsoft SQL Azure database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be none. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 (“NTLMv2”) was concocted to address the

Parameter	Description
	security issues present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Missing indexes	Indicates the total number of missing indexes found in the queries that are currently executing on this database instance.	Number	The detailed diagnosis of this measure reveals the tables and the specific columns in those tables that are missing indexes. If the missing indexes are adversely impacting query performance, then database administrators can use this information to decide on the ideal approach to improving query performance - should new indexes be created? Or should queries be optimized to use existing indexes?
Missing indexes based on user impact	Indicates the number of times the resources of this database instance were impacted due to user queries when the indexes were missing.	Number	
Missing indexes based on user seeks	Indicates the number of times the missing indexes of this database instance appeared in the result set of the queries based on user requests.	Number	

4.3.4 SQL Azure Top Tables Test

When faced with a disk space crunch on their critical Azure SQL database servers, administrators may want to know which database instances are hogging the disk space, and which tables on each database instance have grown beyond permissible limits. The **SQL Azure Top Tables** test provides administrators with this information.

The test auto-discovers the database instances with tables that exceed a configured size limit, and reports the count of such 'large sized tables' for each database instance. You can use the detailed diagnosis of the test to know which tables in a database are of a large size.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for each database instance on the target Microsoft SQL Azure database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can

Parameter	Description
	specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, *none* is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be *none*. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 ("NTLMv2") was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Maximum table size	Indicates the maximum	Number	Use the detailed diagnosis of this

Measurement	Description	Measurement Unit	Interpretation
	size of the table among the tables that are exceeding the configured table size.		measure to figure out the tables that are consuming the maximum disk space.
Table count	Indicates the number of tables that are currently of a size greater than the value configured against the Table Size.	Number	

4.4 The SQL Azure Service Layer

Besides revealing the availability and responsiveness of the database server, the tests mapped to this layer also sheds light on the idle and blocked user sessions on the server, the procedures that consumed the maximum physical reads and logical reads on the server, and the maximum physical reads and logical reads consumed per SQL query execution.

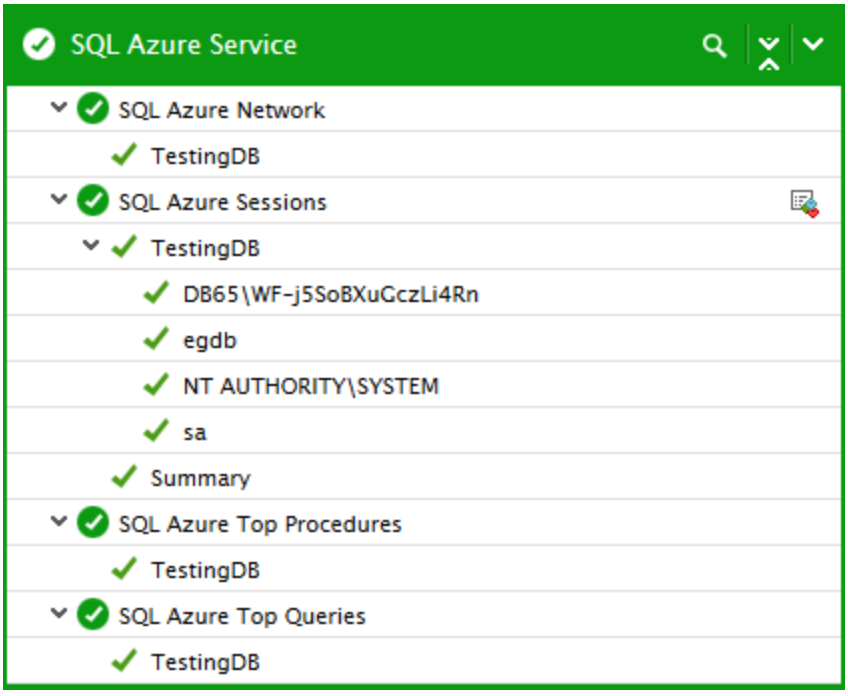

Figure 4.5: The tests associated with the SQL Azure Service layer

4.4.1 SQL Azure Network Test

This test monitors the availability and response time from clients by a Microsoft SQL Azure database server from an external perspective.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for the target Microsoft SQL Azure database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can

Parameter	Description
	continue to be *none*. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 ("NTLMv2") was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability. • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Server availability	Indicates the availability of the server.	Percent	The availability is 100% when the server is responding to a request and 0% when it is not. Availability problems may be caused by a misconfiguration/malfunctioning of the database server, or because the server has not been started. The availability is

Measurement	Description	Measurement Unit	Interpretation
			100% when the instance is responding to a request and 0% when it is not. Availability problems may be caused by a misconfiguration/malfunctioning of the database instance, or because the instance is using an invalid user account. Besides the above, this measure will report that the server is unavailable even if a connection to the database instance is unavailable, or if a query to the database fails. In this case, you can check the values of the *Connection availability* and *Query availability* measures to know what is exactly causing the database instance to not respond to requests - is it owing to a connection unavailability? or is it due to a query failure? Using the detailed diagnosis of this measure, you can easily find out unavailability of the server.
Response time	Indicates the time taken by the database to respond to a user query. This is the sum total of the connection time and query execution time.	Seconds	A sudden increase in response time is indicative of a bottleneck at the database server.
Connection availability	Indicates whether the database connection is available or not.	Percent	If this measure reports the value 100, it indicates that the database connection is available. The value 0 on the other hand indicates that the database connection is unavailable. A connection to the database may be unavailable if the database is down or if the database is listening on a port other than the one configured for it in the eG manager or owing to a poor network link. If the *Server availability* measure reports the value 0, then, you

Measurement	Description	Measurement Unit	Interpretation
			can check the value of this measure to determine whether/not it is due to the unavailability of a connection to the server.
Query availability	Indicates whether the database query is executed successfully or not.	Percent	If this measure reports the value 100, it indicates that the query executed successfully. The value 0 on the other hand indicates that the query failed. In the event that the SQL availability measure reports the value 0, check the value of this measure to figure out whether the failed query is the reason why that measure reported a server unavailability.
Connection time to database server	Indicates the time taken by the database connection.	Seconds	A high value could indicate a connection bottleneck. Whenever the SQL response time of the measure soars, you may want to check the value of this measure to determine whether a connection latency is causing the poor responsiveness of the server.
Query processing time	Indicates the time taken for query execution.	Seconds	A high value could indicate that one/more queries to the database are taking too long to execute. Inefficient/badly designed queries to the database often run for long periods. If the value of this measure is higher than that of the Connection time measure, you can be rest assured that long running queries are the ones causing the responsiveness of the server to suffer.

4.4.2 SQL Azure Sessions Test

In the database context, the connection between the user process and the server process is called a session. The server process communicates with the connected user process and performs tasks on behalf of the users. The **SQL Azure Sessions** test is used by an eG agent to track user activity related to a database server instance.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for each *database instance:user* combination on the target Microsoft SQL Azure Database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be *none*. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.

Parameter	Description
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 ("NTLMv2") was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.
Detailed Diagnosis	

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total sessions	Indicates the total number of sessions that are currently open on the database instance for this user.	Number	A high value may indicate that there is a high load on the database instance.
Running sessions	Indicates the number of sessions that are currently running on the database instance for this user.	Number	
Sleeping sessions	Indicates the number of sessions that were idle on the database instance for this user.	Number	
Suspended sessions	Indicates the number of sessions initiated by this user on the database instance were suspended.	Number	
Rollback sessions	Indicates the number of sessions that were rolled back for this user on the database instance.	Number	
Dormant sessions	Indicates the number of sessions that were	Number	

Measurement	Description	Measurement Unit	Interpretation
	dormant for this user on the database instance.		
Pending sessions	Indicates the number of sessions that were pending for this user on the database instance.	Number	
Blocked sessions	Indicates the number of sessions that were blocked for this user on the database instance.	Number	Blocking occurs when one session holds a lock on a resource that another session is requesting. As a result, the requesting session will be blocked - it will hang until the holding session gives up the locked resource. In almost every case, blocking is avoidable. In fact, if you find that your session is blocked in an interactive application, then you have probably been suffering from the lost update bug as well, perhaps without realizing it. That is, your application logic is flawed and that is the cause of blocking. Ideally, the value of this measure should be zero. The detailed diagnosis capability, if enabled, lists all the blocked sessions of this database.

4.4.3 SQL Azure Top Procedures Test

This test monitors each database instance in the target Microsoft SQL Azure database server and reports the maximum physical reads and logical reads consumed per SQL procedure execution. The maximum CPU time and time elapsed per procedure execution are also identified. The detailed diagnosis of this test lists the top procedures that consumed the maximum physical and logical reads. Administrators can use the detailed diagnosis and figure out the procedures that are executing with delays.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for the target Microsoft SQL Azure database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be *none*. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 (“NTLMv2”) was concocted to address the

Parameter	Description
	security issues present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Maximum physical reads	Indicates the maximum physical reads consumed per execution of an SQL procedure on the SQL Azure database instance.	Reads/execution	The detailed diagnosis of this measure lists the Database ID, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, Last physical reads count, Physical reads per exec, Logical writes count and Logical writes per exec.
Maximum logical reads	Indicates the maximum logical reads consumed per execution of an SQL procedure on the SQL Azure database instance.	Reads/execution	The detailed diagnosis of this measure lists the Database ID, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, Last physical reads count, Physical reads per exec, Logical writes count and Logical writes per exec.

Measurement	Description	Measurement Unit	Interpretation
Maximum CPU time	Indicates the maximum CPU time taken per execution of the SQL procedure on the SQL Azure database instance.	Seconds/execution	The detailed diagnosis of this measure lists the Database id, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, CPU time (Seconds), CPU time per exec (seconds), Logical writes count, and Logical writes per exec.
Maximum elapsed time	Indicates the maximum time elapsed per execution of an SQL procedure on the SQL Azure database instance.	Seconds/execution	The detailed diagnosis of this measure lists the Database id, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, Elapsed time (Seconds), Elapsed time per exec (Seconds), Logical writes count, and Logical writes per exec.
Maximum spills	Indicates the maximum amount of data spill that occurred per execution of an SQL procedure on the SQL Azure database instance.	Spills/execution	The detailed diagnosis of this measure lists the Database id, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, Last spills count, Last spills per exec, Logical writes count, and Logical writes per exec.

4.4.4 SQL Azure Top Queries Test

The real test of the performance of a database server lies in how quickly the database responds to queries. Whenever users complaint of slow execution of their queries, administrators need to know the reason for the delay - is it because the queries themselves are badly designed? or is it due to a longer CPU wait time? or is it because of some queries consuming too much of physical reads per execution? The **SQL Azure Top Queries** test helps with this root-cause analysis.

This test monitors each database instance in the target Microsoft SQL Azure database server and reports the maximum physical reads and logical reads consumed per SQL query execution. The maximum CPU time and time elapsed per query execution are also identified. The detailed diagnosis of this test lists the top queries that consumed the maximum physical and logical reads.

Administrators can use the detailed diagnosis and figure out the queries that are executing with delays.

Target of the test : A Microsoft SQL Azure database server

Agent deploying the test : An external/remote agent

Outputs of the test : One set of results for the target Microsoft SQL Azure Database server being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number through which the target SQL Azure database server communicates. The default port is 1433.
Instance	Enter the name of a specific SQL Azure database server instance that is to be monitored. The default value of this parameter is "default". To monitor an SQL Azure database server instance named "CFS", enter this as the value of the Instance parameter. **Note:** If you specify a particular instance name in this text box, ensure that you do not modify the Port parameter.
Database Name	Specify a comma-separated list of database instances that need to be monitored.
User	Provide the name of the Microsoft SQL Azure database server user. **Note:** Ensure that the user you have specified is capable of logging into the database server instances mentioned in the Database Name text box. Please be noted that you can specify only one User against multiple Database Names.
Password	Here, specify the password corresponding to the user.
Confirm Password	Confirm the password by retyping it in this text box.
Domain	By default, none is displayed in the Domain text box. If the 'SQL server and Windows' authentication has been enabled for the server being monitored, then the Domain can continue to be *none*. On the other hand, if 'Windows only' authentication has been enabled, then, in the Domain text box, specify the Windows domain in which the

Parameter	Description
	managed SQL Azure database exists. Also, in such a case, the User and Password that you provide should be that of a user authorized to access the monitored SQL Azure database.
ISNTLMNV2	In some Windows networks, NTLM (NT LAN Manager) may be enabled. NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. NTLM version 2 ("NTLMv2") was concocted to address the security issues present in NTLM. By default, the ISNTLMNV2 flag is set to **No**, indicating that NTLMv2 is not enabled by default on the target Microsoft SQL Azure database server host. Set this flag to **Yes** if NTLMv2 is enabled on the target host.
SSL	If the target Microsoft SQL Azure database server being monitored is an SSL-enabled server, then set the SSL flag to **Yes**. If not, then set the SSL flag to **No**.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Maximum physical reads	Indicates the maximum physical reads consumed per execution of an SQL query on the SQL Azure database instance.	Reads/execution	The detailed diagnosis of this measure lists the Database id, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, Physical reads count, Physical reads per exec, Logical writes count, and Logical writes per exec.
Maximum logical	Indicates the maximum	Reads/execution	The detailed diagnosis of this

Measurement	Description	Measurement Unit	Interpretation
reads	logical reads consumed per execution of an SQL query on the SQL Azure database instance.		measure lists the Database id, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, Logical reads count, Logical reads per exec, Logical writes count and Logical writes per exec.
Maximum CPU time	Indicates the maximum CPU time taken per execution of the SQL query on the SQL Azure database instance.	Seconds/execution	The detailed diagnosis of this measure lists the Database id, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, CPU time (Seconds), CPU time per exec (seconds), Logical writes count and Logical writes per exec.
Maximum elapsed time	Indicates the maximum time elapsed per execution of an SQL query on the SQL Azure database instance.	Seconds/execution	The detailed diagnosis of this measure lists the Database id, Object name, Type desc, Sql handle, Cached time, Last execution time, Execution count, Elapsed time (Seconds), Elapsed time per exec (Seconds), Logical writes count and Logical writes per exec.

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.