Monitoring SAP HANA Database server

eG Enterprise v6

Restricted Rights Legend

The information contained in this document is confidential and subject to change without notice. No part of this document may be reproduced or disclosed to others without the prior permission of eG Innovations Inc. eG Innovations Inc. makes no warranty of any kind with regard to the software and documentation, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Trademarks

Microsoft Windows, Windows 2008, Windows 2012, Windows 7, Windows 8, and Windows 10 are either registered trademarks or trademarks of Microsoft Corporation in United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Copyright

Table of Contents

Table of Figures

1

Monitoring SAP HANA Database Server

SAP HANA is a flexible, data-source-agnostic appliance that allows customers to analyze large volumes of SAP ERP data in real-time, avoiding the need to materialize transformations. SAP HANA is a hardware and software combination that integrates a number of SAP components including the SAP In-memory database, Sybase Replication technology and SAP® LT (Landscape Transformation) Replicator. SAP HANA is delivered as an optimized appliance in conjunction with leading SAP hardware partners.

Figure 1 depicts the architecture of SAP HANA.

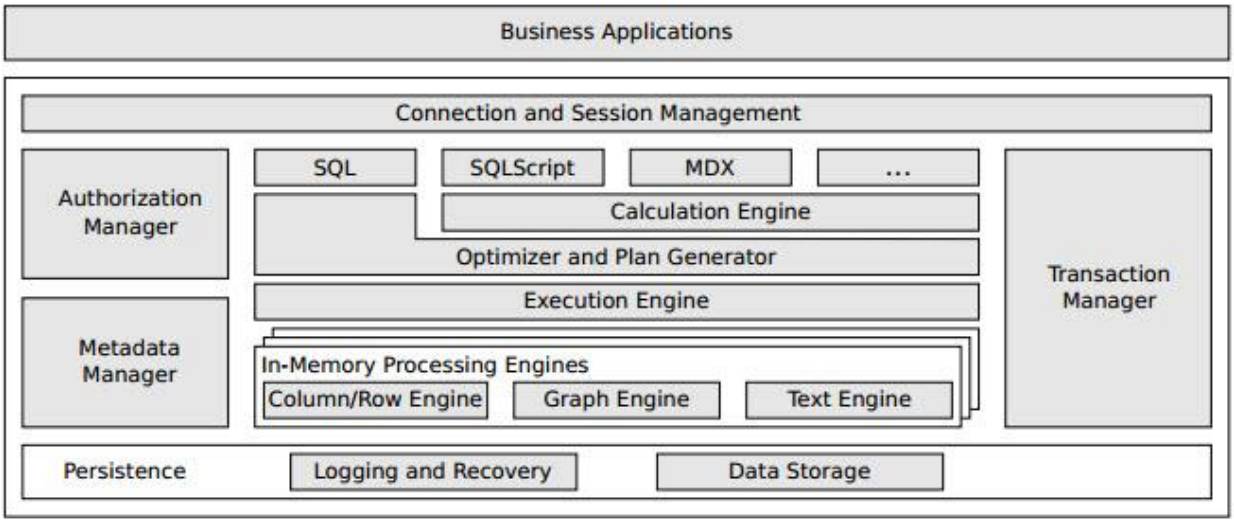

Figure 1.1: The architecture of SAP HANA

The heart of the SAP HANA DB consists of a set of in-memory processing engines. Relational data resides in tables in column or row layout in the combined column and row engine, and can be converted from one layout to the other to allow query expressions with tables in both layouts. Graph data and text data reside in the graph engine and the text engine respectively; more engines are possible due to the extensible architecture. All engines keep all data in main memory as long as there is enough space available. As one of the main distinctive features, all data structures are optimized for cache-efficiency instead of being optimized for organization in traditional disk blocks. Furthermore, the engines compress the data using a variety of compression schemes. When the limit of available main memory is reached, entire data objects, e.g., tables or partitions, are unloaded from main memory under control of application semantic and reloaded into main memory when it is required again. From an application perspective, the SAP HANA DB provides multiple interfaces, such as standard SQL for generic data management functionality or more specialized languages as SQLScript and MDX. SQL queries are translated into an execution plan by the plan generator, which is then optimized and executed by the execution engine. Queries from other interfaces are eventually transformed into the same type of execution plan and executed in the same engine, but are first described by a more expressive

abstract data flow model in the calculation engine. Irrespective of the external interface, the execution engine can use all processing engines and handles the distribution of the execution over several nodes. As in traditional database systems, the SAP HANA DB has components to manage the execution of queries. The session manager controls the individual connections between the database layer and the application layer, while the authorization manager governs the user's permissions. The transaction manager implements snapshot isolation or weaker isolation levels – even in a distributed environment. The metadata manager is a repository of data describing the tables and other data structures, and, like the transaction manager, consists of a local and a global part in case of distribution.

While virtually all data is kept in main memory by the processing engines for performance reasons, data has also to be stored by the persistence layer for backup and recovery in case of a system restart after an explicit shutdown or a failure. Updates are logged as required for recovery to the last committed state of the database and entire data objects are persisted into the data storage regularly.

Owing to its fail-proof architecture, the SAP HANA database server is widely used in many mission-critical IT infrastructures delivering essential services to end-users. In such environments, even a second's non-availability of the database server will not be tolerated, as it may significantly delay service delivery and affect user satisfaction with the service. To ensure high uptime of such services, the availability and overall performance of the SAP HANA database server should be continuously monitored and errors brought to the immediate attention of the administrators.

The eG Enterprise Suite provides a dedicated *SAP HANA* monitoring model, which is capable of monitoring the core components of the SAP HANA architecture – from its memory structures, to its transactions and services – and promptly reporting failures to administrators to enable a speedy resolution.

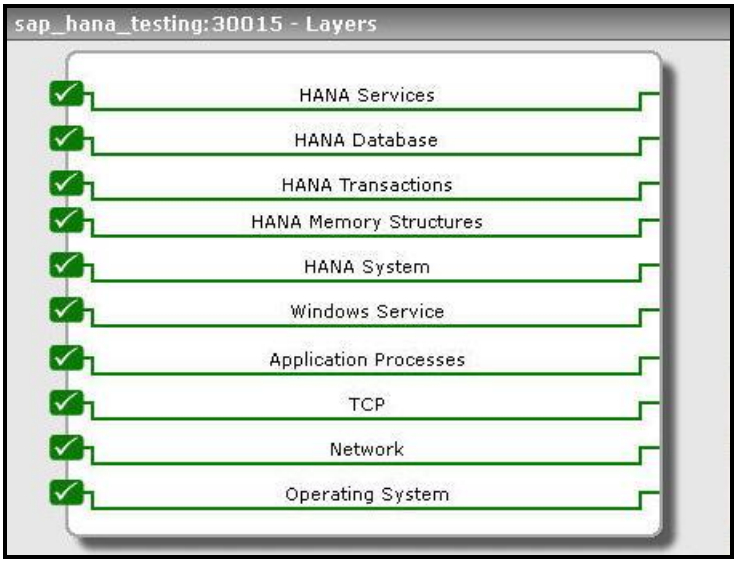

Figure 1.2: The layer model of the SAP HANA database server

Every layer of Figure 1.2 is mapped to a wide variety of tests that run periodic health checks on all performance-influencing parameters of the SAP HANA database server and extract a plethora of performance metrics.

For these tests to run, the pre-requisites detailed in Section 1.1 will have to be in place.

The metrics thus collected would be useful to figure out accurate answers to the following performance queries:

- Is the SAP HANA database server available? How long has the SAP HANA database server been up?
- How quickly does the SAP HANA database server respond to user requests?
- How many connections to the SAP HANA database server are currently active or running? Are any

connections idle?

- Are the critical services of the SAP HANA database server currently active? If so, for how long has each service been up?
- How quickly were requests serviced by each service of the SAP HANA database server? Does any service process requests slowly? How many requests are currently pending for processing with that service?
- How many threads are currently active for each service?
- How much CPU and memory does each service use? Is any service utilizing these resources excessively?
- Is enough physical memory available for use with each service?
- Is the server overloaded with query execution requests? If so, which service on the server is handling the maximum executions?
- Is any service performing too many rollbacks? If so, which one is it?
- Are any transactions blocked?
- Is any expensive SQL statement executing on the SAP HANA database server? If so, what is the maximum duration of its execution?
- Is the server utilizing its cache well? Or should the cache be resized to ensure its effective usage?
- Is any disk volume on the server running out of space? If so, which one is it – data, log, or trace?
- Are the disk volumes used by the services adequately sized? Or is any service using its disk volumes excessively? If so, which service is it and which disk volume is being over-utilized – data or log?
- Is the server hogging the CPU? If so, where is maximum CPU time being spent – in system-level processing? in handling user requests? In waiting for I/O? or in just being idle?
- Does the server have sufficient physical memory to discharge its duties?

The **Operating System**, **Network**, **TCP**, **Application Processes and Windows Service** layers of a *SAP HANA database* server model are similar to that of a *Windows Generic* server model. Since these tests have been dealt with in the *Monitoring Unix and Windows Servers* document, the below mentioned sections focus only on the layers pertaining to the SAP HANA database server alone.

1.1 Pre-requisites for Monitoring the SAP HANA Database Server

To monitor the SAP HANA database server:

- A special database user account vested with the **Monitoring** role has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script to be used for user creation is as follows:

  create user <username> password <password>

  grant monitoring to <username>

  A sample script for user creation is:

  create user sapegadmin password eGurkha@123;
  grant monitoring to sapegadmin;

The credentials of this user have to be passed to the **USER** and **PASSWORD** parameters of the eG tests for SAP HANA.

- To connect to SAP HANA via JDBC, the eG agent needs the driver file provided by SAP, called **ngdbc.jar**. This driver file is part of the SAP HANA Client installation, which in turn will be part of your SAP HANA installation.

  To make this driver file available to the eG agent, do the following:

  - Copy the **ngdbc.jar** file from the SAP HANA client installation path to the **<EG_INSTALL_DIR>\lib** directory (on Windows; on Unix, this will be the **/opt/egurkha/lib** directory) on the eG agent host.

    The table below provides the default location of the **ngdbc.jar** on different platforms.

Platform	Package version	Default installation path
Linux x86, 32 bit	32 bit	/usr/sap/hdbclient
Linux x86, 64 bit	64 bit	/usr/sap/hdbclient
Linux x86, 64 bit	32 bit	/usr/sap/hdbclient32
Other Unix/Linux platforms	-	/usr/sap/hdbclient
Windows x86, 32 bit	32 bit	C:\Program Files\sap\hdbclient
Windows x86, 64 bit	64 bit	C:\Program Files\sap\hdbclient
Windows x86, 64 bit	32 bit	C:\Program Files (x86)\sap\hdbclient

    If the HANA client is installed by a user without administrative rights, then that client will be available only to that user.

  - Then, restart the eG agent.

1.2 The HANA Services Layer

The **HANA Services** layer tracks the availability, responsiveness and the health of the services associated with a SAP HANA database server.

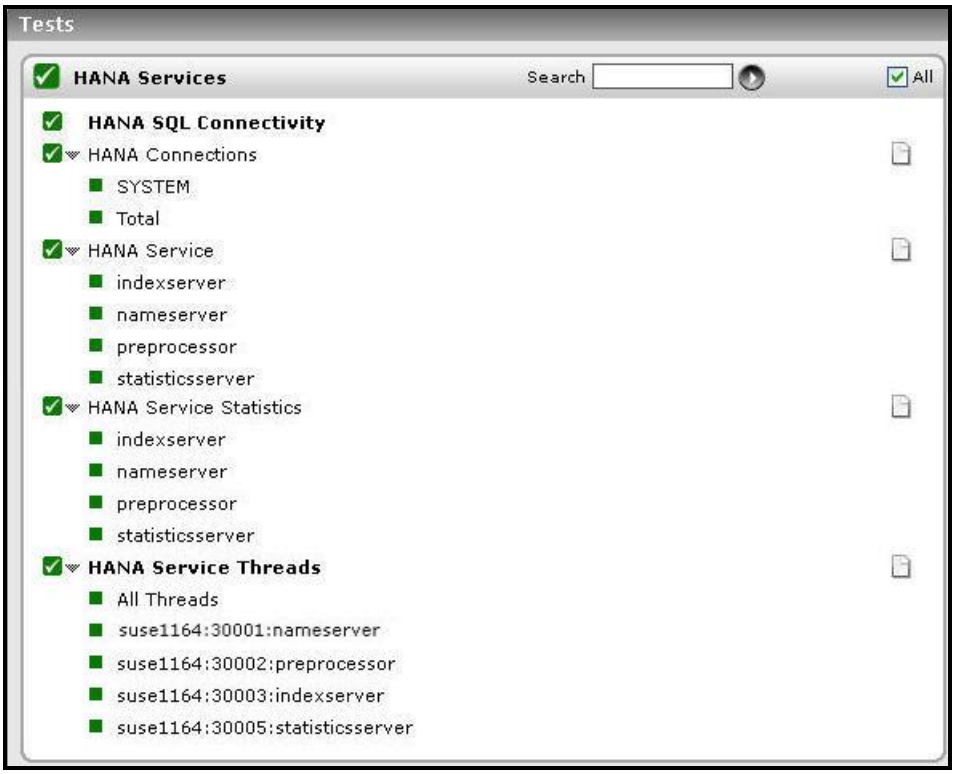

Figure 1.3: The tests mapped to the HANA Services layer

1.2.1 HANA SQL Connectivity Test

This test reports the availability and responsiveness of the SAP HANA database server.

Purpose	Reports the availability and responsiveness of the SAP HANA database server
Target of the test	A SAP HANA database server
Agent deploying the test	A remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as “Not applicable” by the agent if the server is not up.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every SAP HANA database server being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Availability:
Indicates the availability of this SAP HANA database server.</td><td>Number</td><td>A value of 100 indicates that the SAP HANA database server is available and a value 0 indicates that the database server is not available.</td></tr>
<tr><td>Response Time:
Indicates the time taken by this SAP HANA database server to respond to a user request.</td><td>Secs</td><td>Ideally, the value of this measure should be low. A sudden increase in response time is indicative of a bottleneck at the database server. This could even be owing to a connection delay and/or long running queries to the database.</td></tr>
</table>

1.2.2 HANA Connections Test

User connections to the SAP HANA database server are a good indicator of the load on the database server. If users complain about the sluggish performance of the database server, administrators may want to check the user connections to the server to figure out whether the database server is stressed due to an overload condition. If so, administrators may also want to identify the user who has imposed the maximum load on the server and the nature of that load – i.e., whether the user’s connections are active or are idle predominantly, and the maximum time for which a user connection to the server remained alive and idle. The **HANA Connections** test reveals all this! This test not only discovers who is currently logged into the database server, but for each login user, reports the number of

open connections, the count of running and idle connections, and the duration of the oldest idle connection. This way, the test measures the current connection load on the server, points to those users who are responsible for overloading the server with numerous connections, and also indicates to administrators whether these are genuine running connections or idle connections that can be terminated to ease the strain on the server.

Purpose	Not only discovers who is currently logged into the database server, but for each login user, reports the number of open connections, the count of running and idle connections, and the duration of the oldest idle connection. This way, the test helps administrators identify those users who are responsible for overloading the server with numerous connections, and also indicates to administrators whether these are genuine running connections or idle connections that can be terminated to ease the strain on the server.
Target of the test	A SAP HANA database server
Agent deploying the test	A remote agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST -** The host for which the test is to be configured. 3. **PORT** – The port number at which the specified **HOST** listens to. The default port is *30015*. 4. **USER** – In order to monitor a SAP HANA database server, a special database user account with *Monitoring* privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is: *create user <username> password <password>* *grant monitoring to <username>* A sample script for user creation is: *create user sapegadmin password eGurkha@123;* *grant monitoring to sapegadmin;* The name of this user has to be specified here. 5. **PASSWORD –** The password of the specified **USER**. 6. **CONFIRM PASSWORD –** Confirm the password by retyping it here. 7. **ISPASSIVE –** If the value chosen for this parameter is **Yes**, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up. 8. **SHOW USER -** Set this flag to **Yes** if you wish to view the connection details of each individual user. By doing so, each user pertaining to the SAP HANA database server will appear as a *descriptor* of this test. By default, this flag is set to **No** which implies that the *TOTAL* descriptor will alone appear for this test indicating the connection details of all the users collectively.
Outputs of the test	One set of results for every SAP HANA database server being monitored; if the **SHOW USER** flag is set to **Yes**, then this test will report one set of results for every user who is currently connected to the SAP HANA database server

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Open Connections: Indicates the number of currently open connections for this user.	Number	
	Idle Connections: Indicates the number of connections of this user that are currently idle on the database server.	Number	Ideally, the value of this measure should be low. A high value is indicative of a large number of idle connections, which in turn may cause unnecessary consumption of critical server resources. Idle connections also unnecessarily lock critical database objects, thereby denying other users access to those objects for performing important tasks. Use the detailed diagnosis of this measure to view the details of the idle connections.
	Running Connections: Indicates the number of connections established by this user that are currently running.	Number	
	Oldest Connection: Indicates the maximum time duration for which a connection established by this user was idle during the last measurement time.	Secs	A low value is desired for this measure. A high value indicates that the connections to the database server have been idle for a very long time.

1.2.3 HANA Service Test

SAP HANA database consists of the following services:

- **Index Server:** Contains the actual data and the engines for processing the data. It also coordinates and uses all the other servers.
- **Name Server:** Holds information about the SAP HANA database topology. This is used in a distributed system with instances of HANA database on different hosts. The name server knows where the components are running and which data is located on which server.
- **Statistics Server:** Collects information about Status, Performance and Resource Consumption from all the other server components. From the SAP HANA Studio we can access the Statistics Server to get status of various alert monitors.
- **Preprocessor Server:** Used for Analysing Text Data and extracting the information on which the text search capabilities are based.
- **XS Engine:** This is an optional component. Using XS Engine clients can connect to SAP HANA database to fetch data via HTTP.

If the SAP HANA database server is unable to perform any of the functions detailed above, it could be because the corresponding service is not running. Likewise, processing bottlenecks experienced by any of these critical services can consequently degrade the performance of the database server. It is hence imperative that administrators check

on the status and processing ability of each of the critical services listed above now and then, so that failures and slowdowns can be proactively detected and promptly resolved, and the uninterrupted functioning of the database server ensured. The **HANA Service** test helps administrators achieve the same! This test auto-discovers each service of the SAP HANA database server and reports the current status, uptime, responsiveness, and the request processing ability of each service, thus shedding light on services that are not running or are under-performing. By identifying problematic services, focusing on their issues, and solving them, administrators can make sure that the database server operates at peak capacity at all times!

Purpose	Auto-discovers each service of the SAP HANA database server and reports the current status, uptime, responsiveness, and the request processing ability of each service, thus shedding light on services that are not running or are under-performing		
Target of the test	A SAP HANA database server		
Agent deploying the test	A remote agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST -** The host for which the test is to be configured. 3. **PORT** – The port number at which the specified **HOST** listens to. The default port is *30015*. 4. **USER** – In order to monitor a SAP HANA database server, a special database user account with *Monitoring* privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is: *create user <username> password <password>* *grant monitoring to <username>* A sample script for user creation is: *create user sapegadmin password eGurkha@123;* *grant monitoring to sapegadmin;* The name of this user has to be specified here. 5. **PASSWORD –** The password of the specified **USER**. 6. **CONFIRM PASSWORD –** Confirm the password by retyping it here. 7. **ISPASSIVE –** If the value chosen for this parameter is **Yes**, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up.		
Outputs of the test	One set of results for every service of the SAP HANA database server being monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

<table>
<tr><td rowspan="6">test</td><td>Active Status:
Indicates whether/not this service is `currently active on the SAP HANA database server.</td><td></td><td>This measure reports the value Yes if this service is active and No, if otherwise.

The values that this measure reports and their corresponding numeric values have been listed 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 one of the Measure Values listed in the table above to indicate whether/not this service is active on the SAP HANA database server. In the graph of this measure however, the same will represented using the numeric equivalents only.</td></tr>
<tr><td>Uptime:
Indicates the time duration for which this service has been up since the last time the server was started.</td><td>Mins</td><td>A low value is indicative of intermittent breaks in the availability of a service.</td></tr>
<tr><td>Request Rate:
Indicates the rate at which the requests were issued to this service from the clients.</td><td>Requests/Sec</td><td></td></tr>
<tr><td>Average Response Time:
Indicates the average time taken by this service to respond to requests from the clients.</td><td>MilliSec</td><td>A low value is desired for this measure. A high value or a consistent increase in the value is indicative of a processing bottleneck with the service.</td></tr>
<tr><td>Active Request Count:
Indicates the number of active requests that are being processed by this service.</td><td>Number</td><td>This is a good indicator of the current load on a service.</td></tr>
<tr><td>Pending Request Count:
Indicates the number of requests that are yet to be processed (i.e., pending in the queue) by this service.</td><td>Number</td><td>A low value is desired for this measure. A high value or a consistent increase in the value is indicative of a processing bottleneck with the service.</td></tr>
</table>

	Active Thread Count: Indicates the number of threads that are actively servicing requests for this service.	Number	If the *Pending Request Count* measure is high and indicate an overload or a processing slowdown with a service, you may consider spawning additional threads for that service to enable it to handle the additional load and clear all pending requests.
	Thread Count: Indicates the total number of threads for this service.	Number	
	Open File Count: Indicates the number of files that are currently open through this service in the SAP HANA database server.	Number	This indicates the load imposed by a service on the database server.
	Has the HANA service been restarted? Indicates wheterher/not the HANA service was restarted during the last measurement period.		If the service was restarted, this measure will report the value *Yes*. If it was not restarted, then this measure will report the value *No*. The numeric values that correspond to these measure values are as follows: **Measure Value** / **Numeric Value** Yes / 1 No / 0 **Note:** By default, this measure reports the **Measure Values** in the table above to indicate whether/not the service rebooted. However, in the graph of this measure, the same is represented using the numeric equivalents only.

1.2.4 HANA Service Statistics Test

Each of the services executing on the SAP HANA database server consume critical CPU and memory resources of the server. Excessive resource usage by a single service can impact resource availability for other services, thereby adversely impacting the overall performance of the services and the server. Hence, in the event of a slowdown in the operations of the database server, it would be good practice for administrators to take a look at the resource usage of the individual services, so that they can determine whether the poor performance of the server is because one/more services are hogging the resources. To achieve this, administrators can use the **HANA Service Statistics** test. For each service running within SAP HANA, this test reports the CPU and memory usage of the service, thus enabling administrators to accurately identify resource-hungry services that can potentially cause server performance to degrade.

Purpose	For each service running within SAP HANA, this test reports the CPU and memory usage of the service, thus enabling administrators to accurately identify resource-hungry services that can potentially cause server performance to degrade.

<table>
<tr><td>Target of the test</td><td colspan="3">A SAP HANA database server</td></tr>
<tr><td>Agent deploying the test</td><td colspan="3">A remote agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as “Not applicable” by the agent if the server is not up.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every service of the SAP HANA database server being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Process CPU Usage:
Indicates the percentage of CPU that is used by the current process that is associated with this service.</td><td>Percent</td><td></td></tr>
<tr><td>Total CPU Usage:
Indicates the total percentage of CPU used by all processes that are associated with this service.</td><td>Percent</td><td>A high value or a consistent increase in the value of this measure is a cause for concern, as it indicates abnormal CPU usage by the service.</td></tr>
</table>

	Process Logical Memory: Indicates the amount of logical memory used by this service.	MB	Ideally, the value of these measures should be low. Compare the value of each of these measures across services to identify that service that is engaging in memory-intensive operations.
	Process Physical Memory: Indicates the amount of physical memory used by this service.	MB	
	Used Memory: Indicates the amount of memory used by this service in the SAP HANA database server.	MB	The value of this measure constitutes both physical memory and swap memory usage. Compare the value of this measure across all the services to identify the service that is consuming memory resources excessively.
	Free Memory: Indicates the amount of memory that is currently available for use by this service.	MB	This value is the sum of free physical memory and free swap memory. Ideally, this value should be high for a service. A consistent drop in the value of this measure is indicative of a memory-hungry service, which can, in the long run, cause a severe memory contention within SAP HANA.
	Total Memory: Indicates the total amount of memory used by this service.	MB	The total memory constitutes both physical memory and swap memory.
	Percentage of Free Memory: Indicates the percentage of memory (both physical memory and swap memory) that is currently available for use by this service.	Percent	A high value is desired for this measure. A consistent decrease in this value indicates that the service is steadily draining SAP HANA off its memory resources.
	Physical Memory Usage: Indicates the percentage of physical memory that is currently used by this service.	Percent	A low value is desired for this measure. A consistent increase in this value indicates that the service is steadily draining SAP HANA off its memory resources.

1.2.5 HANA Service Threads Test

By closely monitoring thread usage of each of the services of a SAP HANA database server, administrators can gauge the workload generated by every service and how well each service handles the load. The **HANA Service Threads** test does just that! This test reports the count of threads that services requests for a service and the maximum time that the threads took to process service requests. In the process, the test indicates how busy a service is, and points to those services that may run into a processing bottleneck.

Purpose	Reports the count of threads that services requests for a service and the maximum time that the threads took to process service requests. In the process, the test indicates how busy a service is, and points to those services that may run into a processing bottleneck.		
Target of the test	A SAP HANA database server		
Agent deploying the test	A remote agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST -** The host for which the test is to be configured. 3. **PORT** – The port number at which the specified **HOST** listens to. The default port is *30015*. 4. **USER** – In order to monitor a SAP HANA database server, a special database user account with *Monitoring* privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is: *create user <username> password <password>* *grant monitoring to <username>* A sample script for user creation is: *create user sapegadmin password eGurkha@123;* *grant monitoring to sapegadmin;* The name of this user has to be specified here. 5. **PASSWORD –** The password of the specified **USER**. 6. **CONFIRM PASSWORD –** Confirm the password by retyping it here. 7. **ISPASSIVE –** If the value chosen for this parameter is **Yes**, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up.		
Outputs of the test	One set of results for every *Thread: Service* of the SAP HANA database server being monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**Thread Count:** Indicates the number of threads that are currently handling the requests for this service.	Number	This is a good indicator of the workload generated by a service on SAP HANA.
	Longest Thread: Indicates the maximum time that a thread took to serve requests for this service.	MilliSec	A high value for this measure is a cause of concern, as it could indicate a potential service slowdown.

1.3 The HANA Database Layer

The tests associated with this layer (see Figure 3) monitor the uptime of the SAP HANA database server and reports how well each port of the SAP HANA database server handles executions, compilations, rollbacks etc.

Figure 1.4: The tests mapped to the HANA Database layer

1.3.1 HANA Uptime Test

Frequent database restarts or intermittent breaks in database availability are signs of poor database server health. Administrators hence need to track the uptime of each database in SAP HANA, so that they can quickly compare that with service level targets to identify trouble-prone or unstable databases. To achieve this, administrators can use the **HANA Uptime** test. This test auto-discovers the databases on SAP HANA and reports the uptime of each database, so that those databases that went down frequently can be identified.

Purpose	Auto-discovers the databases on SAP HANA and reports the uptime of each database, so that those databases that went down frequently can be identified
Target of the test	A SAP HANA database server

<table>
<tr><td>Agent deploying the test</td><td colspan="3">A remote agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as “Not applicable” by the agent if the server is not up.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every database on the SAP HANA database server being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Uptime:
Indicates the total time for which this database was up since its last reboot.</td><td>Mins</td><td>A low value for this measure may be either due to planned maintenace of the database or a severe problem with the availability of the database. to determine such conditions.</td></tr>
</table>

	Has the HANA server been restarted? Indicates whether/not the HANA server was restarted during the last measurement period.		If the server was restarted, this measure will report the value *Yes*. If it was not restarted, then this measure will report the value *No*. The numeric values that correspond to these measure values are as follows: **Measure Value** / **Numeric Value** Yes / 1 No / 0 **Note:** By default, this measure reports the **Measure Values** in the table above to indicate whether/not the server rebooted. However, in the graph of this measure, the same is represented using the numeric equivalents only.

1.3.2 HANA Workload Test

The workload of the SAP HANA database server is typically characterized by the preparation, compilation, and executon of query statements for data definition, system control, and data manipulation such as inserts, updates, commits, and rollbacks. One of the key indicators of the performance of the server is its ability to handle this workload! While sudden/consistent surges in the load on the server can cause queries to queue up and choke the server, an unexpected slowdown in transaction execution can adversely impact the user-perceived performance of the server. Besides, the type of statements being submitted to the server for execution also affect its processing efficiency – for instance, rollbacks are basically expensive operations, and need to be kept at a bare minimum; a large number of rollbacks can hence, greatly degrade server performance. A healthy SAP HANA database therefore, is one which is not only moderately loaded, but has little-to-no rollback statements to execute, and is capable of processing its load in minimal time! To ensure the good health of the SAP HANA database, administrators should be able to:

- Proactively detect a potential overload/processing slowdown;
- Isolate the exact service that is facing the brunt of this problem condition;
- Understand the nature of the load on that service and identify the query types that could be contributing to the high load/slowdown

The **HANA Workload** test enables administrators to achieve each of the goals outlined above. This test auto-discovers the port at which each service on the server listens, and for each port reports the total load on that port (i.e., service), the type of transactions handled by that port – whether updates, commits, or rollbacks, and the rate at which that port processes the load. This way, administrators can tell which port (i.e., service) is overloaded and/or is experiencing a processing bottleneck, and what type of transactions are responsible for this problem condition – updates? commits? or rollbacks?

Purpose	Auto-discovers the port at which each service on the server listens, and for each port reports the total load on that port (i.e., service), the type of transactions handled by that port – whether updates, commits, or rollbacks, and the rate at which that port processes the load
Target of the test	A SAP HANA database server

<table>
<tr><td>Agent deploying the test</td><td colspan="3">A remote agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each service port of the SAP HANA database server being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Execution Count:
Indicates the total workload of this port/service during the last measurement period.</td><td>Number</td><td>This is the total count of all executed statements for data manipulation, data definition, and system control, and is a good indicator of the workload of the port/service. By comparing the value of this measure across all ports, you can identify the overloaded port/service.</td></tr>
<tr><td>Compilation Count:
Indicates the number of compilations performed by this port/service during the last measurement period.</td><td>Number</td><td>Ideally, the value of this measure should be less than the value of the Execution Count measure.
When the Compilation Count measure is higher than the Execution Count measure, it indicates that the same query is compiled too many times which in turn is a problem that needs to be probed immediately.</td></tr>
</table>

	Update Transaction Count: Indicates the number of records that are updated in the SAP HANA database through this port/service during the last measurement period.	Number	Once you are able to precisely point to the overloaded service/port, you may want to compare the value of the *Update Transaction Count, Commit Count*, and *Rollback Count* measures for that service/port, to figure out which type of transaction is the maximum on that port. A high number of rollbacks is a cause for concern, as it increases processing overheads.
	Commit Count: Indicates the number of transaction commits that happened in the SAP HANA database through this service/ port during the last measurement period.	Number	Once you are able to precisely point to the overloaded service/port, you may want to compare the value of the *Update Transaction Count, Commit Count*, and *Rollback Count* measures for that service/port, to figure out which type of transaction is the maximum on that port. A high number of rollbacks is a cause for concern, as it increases processing overheads.
	Rollback Count: Indicates the number of transactions that are rolled back in the SAP HANA database through this service/port during the last measurement period.	Number	Once you are able to precisely point to the overloaded service/port, you may want to compare the value of the *Update Transaction Count, Commit Count*, and *Rollback Count* measures for that service/port, to figure out which type of transaction is the maximum on that port. A high number of rollbacks is a cause for concern, as it increases processing overheads.
	Rollback Percent: Indicates the percentage of transactions that are rolled back in the SAP HANA database through this service/port during the last measurement period.	Percent	Compare the value of this measure across services/ports to know which service/port executed the maximum number of rollbacks. Any value higher than 30 is an indicator of bad performance, since work performed up to the rollback point is wasted. The cause of the rollbacks has to be probed.
	Execution Rate: Indicates the rate at which this service/port processed the load.	Executions/Sec	Ideally, the value of this measure should be high. A low value is indicative of a processing bottleneck. You may want to compare the value of this measure acros services/ports to know which service/port is the slowest in load processing.
	Compilation Rate: Indicates the rate at which compilations were performed on the SAP HANA database through this service/port.	Compilations/Sec	Ideally, the value of this measure should be high. A low value is indicative of a compilation bottleneck. You may want to compare the value of this measure across services/ports to know which service/port is the slowest in compilation.

	Update Transaction Rate: Indicates the rate at which the records were updated on the SAP HANA database through this service/port.	Updates/Sec	Ideally, the value of this measure should be high. A low value is indicative of problems when updating records in the database. You may want to compare the value of this measure across services/ports to know which service/port is the slowest in updating records.
	Commit Rate: Indicates the rate at which transactions were committed to the database by this service/port.	Commits/Sec	Ideally, the value of this measure should be high. A decrease in this measure during the monitoring period may indicate that the applications are not doing frequent commits. This may lead to problems with logging and data concurrency. You may want to compare the value of this measure across services/ports to know which service/port seldom commits. The cause for the same has to be probed.
	Rollback Rate: Indicates the rate of unit of work rollbacks since the last measurement period.	Rollbacks/Sec	Ideally, the value of this measure should be low. A high rollback rate is an indicator of bad performance, since work performed up to the rollback point is wasted. You may want to compare the value of this measure across services/ports to know which service/port performs rollback operations on the database frequently. The cause for the same has to be probed.

1.4 The HANA Transactions Layer

The **HANA Transactions** layer monitors each type of transaction and reports how many transactions are currently open, blocked and the maximum time taken for execution of a transaction.

Figure 1.5: The tests mapped to the HANA Transactions layer

1.4.1 HANA Transactions Test

Long-running transactions to the SAP HANA database server not only keep critical resources and database objects locked for indefinite time periods, but they can also prevent execution of other transactions, thereby degrading the overall performance of the database server. It is hence imperative that such transactions are rapidly isolated and killed. The **HANA Transactions** test makes this possible! For each type of transaction, this test reports the number of transactions that are currently open and the maximum time taken for executing transactions. Using this test, you could figure out which transaction is taking too long to execute and is currently blocking other transactions.

Purpose	For each type of transaction, this test reports the number of transactions that are currently open and the maximum time taken for executing transactions. Using this test, you could figure out which transaction is taking too long to execute and is currently blocking other transactions.
Target of the test	A SAP HANA database server
Agent deploying the test	A remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up.
8. EXCLUDE TRANSACTION TYPE - Specify a comma-separated list of transaction types that need to be excluded from the scope of monitoring in this text box. By default, none is specified here.
9. 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 enabled/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.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every transaction type on the SAP HANA database server being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Open Transactions:
Indicates the total number of transactions of this type that are currently open in the SAP HANA database server.</td><td>Number</td><td>This measure is a good indicator of the current workload of the database server.</td></tr>
</table>

	Oldest Transaction: Indicates the maximum time that a transaction of this type took to execute.	Secs	A low value is desired for this measure. A gradual/sudden increase in the value of this measure is a clear indicator that some transaction is taking too long to execute. This is a cause for concern, as such transactions can block other transactions from executing on the server and can even cause server performance in general to deteriorate. Under such circumstances, you can take the help of the detailed diagnosis of this measure to identify the long-running transactions. The detailed diagnosis lists the the name of the transaction, the type of the transaction, the time duration of the transachetion, and the memory utilized by the transaction, thus shedding light on long-running and resource-intensive transactions.

1.4.2 HANA Blocked Transactions Test

When a transaction attempts to access a database object that is already being used by another transaction, a transaction block occurs, whereby the original transaction blocks/prevents the subsequent transaction from executing on the server. If the block is not identified and released in time, then the user experience with the server will suffer! Administrators should hence be able to quickly identify blocked transactions, analyze which transactions are blocking them and why, and endeavor to clear the block quickly, so that users can continue interacting wih the server without any interruptions. This is where the **HANA Blocked Transactions** test helps. For every transaction type, this test reports the number of transactions of that type that are currently blocked and provides detailed diagnostics related to such transactions, so that administrators can rapidly identify the blocked transactions, accurately analyze its root-cause, and promptly remove the block.

Purpose	Reports the number of transactions that are currently blocked and provides detailed diagnostics related to such transactions, so that administrators can rapidly identify the blocked transactions, accurately analyze its root-cause, and promptly remove the block
Target of the test	A SAP HANA database server
Agent deploying the test	A remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as “Not applicable” by the agent if the server is not up.
8. EXCLUDE TRANSACTION TYPE - Specify a comma-separated list of transaction types that need to be excluded from the scope of monitoring in this text box. By default, none is specified here.
9. DETAILED DIAGNOSISTo 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 enabled/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.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every transaction type in the SAP HANA database server being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

test	**Blocked Transactions:** Indicates the number of transactions of this type that are currently blocked.	Number	Ideally, the value of this measure should be zero. A gradual increase in the value of this measure is a cause of concern, as it indicates a consistent rise in blocked transactions. This calls for immediate action. The detailed diagnosis capability, if enabled lists all the blocked transactions and the users who have initiated those transactions. Using this detailed diagnosis, you can figure out the root cause of the blocking.

1.5 The HANA SQL Layer

The **HANA SQL** layer monitors the SQL statements executing on the SAP HANA database server and reports the numerical statistics of the SQL statements and the maximum time duration for executing such statement.

Figure 1.6: The tests mapped to the HANA SQL layer

1.5.1 HANA Expensive Statements Test

Expensive statements are individual SQL queries whose execution time exceeded a configured threshold. By default, this threshold is 1 second (1000000 microseconds). Such statements, besides being a resource-drain, may also block the execution of other SQL statements, degrading the performance of the database server in the process. To avoid such eventualities, these long-running statements need to be isolated and optimized rapidly. This is where the **HANA Expensive Statements** test helps. This test reports the count of expensive SQL statements and the maximum duration for which such statements ran. Using the detailed diagnosis of the test, you can also identify the expensive SQL statements and attempt to fine-tune them in order to improve their performance, and consequently, enhance server performance.

Purpose	Reports the count of expensive SQL statements and the maximum duration for which such

	statements ran
Target of the test	A SAP HANA database server
Agent deploying the test	A remote agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST -** The host for which the test is to be configured. 3. **PORT** – The port number at which the specified **HOST** listens to. The default port is *30015*. 4. **USER** – In order to monitor a SAP HANA database server, a special database user account with *Monitoring* privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is: *create user <username> password <password>* *grant monitoring to <username>* A sample script for user creation is: *create user sapegadmin password eGurkha@123;* *grant monitoring to sapegadmin;* The name of this user has to be specified here. 5. **PASSWORD –** The password of the specified **USER**. 6. **CONFIRM PASSWORD –** Confirm the password by retyping it here. 7. **DD RECORDS -** Specify the number of expensive SQL statements for which the detailed analysis is to be reported in the **DETAILED DIAGNOSIS** section. By default, the value specified in this text box is *10*. This indicates that detailed analysis of the *top 10* expensive SQL statements executing on the SAP HANA database server will alone be listed in the **DETAILED DIAGNOSIS** section. 8. **ISPASSIVE –** If the value chosen for this parameter is **Yes**, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up. 9. **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 enabled/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.
Outputs of the test	One set of results for every SAP HANA database server being monitored

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Statement Count: Indicates the total number of expensive SQL statements executed on the SAP HANA database server.	Number	A low value is desired for this measure. A gradual/sudden increase in the number of expensive SQL statements can hamper the performance of the SAP HANA database server. Some of the consequences of executing the expensive SQL statements are as follows: • While an expensive SQL statement is being executed, as it has to fetch more blocks, already existing blocks in the database buffer are displaced to accommodate the blocks of this statement. This impacts performance of the other SQL statements and could impact performance of subsequent requests. • As the database will be busy readying many blocks to satisfy expensive SQL statements, other critical database tasks are very likely to slow down, which eventually may impact the performance of the SAP HANA database server • While an expensive SQL statement is being executed, a work process is blocked till the completion of the request. So, it is not available for other requests which increases the wait time of other processes. You can use the detailed diagnosis of this test to identify the expensive statements and optimize them for better performance.
	Maximum Duration: Indicates the maximum time duration taken to execute an expensive SQL statement on the SAP HANA Database server.	MilliSec	A high value for this measure is a cause of concern. In order to fine tune the time duration, you may have to optimize the query by changing the query structure or adding index to the database table.

1.6 The HANA Memory Structures Layer

The **HANA MEMORY STRUCTURES** layer tracks the usage statistics of each cache available with the SAP HANA database server.

Figure 1.7: The tests mapped to the HANA Memory Structures layer

1.6.1 HANA Cache Test

A cache that is right-sized and well-used can significantly boost database server performance! A cache is said to be effectively utilized only if it is able to service the maximum number of requests to the database; this greatly reduces direct disk accesses and related overheads, and thus improves database performance. On the contrary, ineffective cache usage can be the key contributor to a slowdown or degradation in server performance, as it increases disk accesses. To understand how the database caches are utilized and to promptly capture abnormalities in cache usage, administrators have to continuously monitor the size and usage of each of the database caches. The **HANA Cache** test helps administrators in this regard. This test monitors each cache of the SAP HANA database server and reports its usage – both in terms of size and its request serving ability. In the process, the test proactively alerts administrators to the under-utilization and improper size of the cache, and helps them quickly initiate corrective measures.

Purpose	Monitors each cache of the SAP HANA database server and reports its usage – both in terms of size and its request serving ability. In the process, the test proactively alerts administrators to the under-utilization and improper size of the cache, and helps them quickly initiate corrective measures.
Target of the test	A SAP HANA database server
Agent deploying the test	A remote agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST -** The host for which the test is to be configured. 3. **PORT** – The port number at which the specified **HOST** listens to. The default port is *30015*. 4. **USER** – In order to monitor a SAP HANA database server, a special database user account with *Monitoring* privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is: *create user <username> password <password>* *grant monitoring to <username>* A sample script for user creation is: *create user sapegadmin password eGurkha@123;* *grant monitoring to sapegadmin;* The name of this user has to be specified here. 5. **PASSWORD –** The password of the specified **USER**. 6. **CONFIRM PASSWORD –** Confirm the password by retyping it here. 7. **ISPASSIVE –** If the value chosen for this parameter is **Yes**, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up.
Outputs of the test	One set of results for every cache of the SAP HANA database server being monitored

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Total Size: Indicates the total size of this cache.	MB	
	Used Size: Indicates the space in the cache that is currently in use.	MB	Ideally, the value of this measure should be low. A high value or a consistent rise in the value is a sign that the cache is about to run out of space. If a cache does not have adequate space, it cannot hold additional entries, and hence cannot service requests properly. You may hence have to allocate more space to the cache or free-up space in the cache to accommodate more entries.
	Entries: Indicates the number of entries i.e., requests or transactions that are available for retrieval from this cache.	Number	Ideally, the value of this measure should be high.
	Inserts: Indicates the number of INSERT entries that can be retrieved from this cache based on a received request.	Number	
	Invalidates: Indicates the number of INVALIDATE entries that can be retrieved from this cache based on a received request.	Number	
	Hits: Indicates the number of times this cache was hit for an entry without having to read from the disk during the last measurement period.	Number	A high value is desired for this measure. A low value or a consistent drop in this value is a cause for concern, as it indicates ineffective cache usage. This means high disk accesses, and poor server performance. One of the common reasons for bad cache usage is improper cache size. A cache that does not have sufficient space to accommodate entries will not be able to service requests effectively. You may hence have to allocate more space to the cache or free-up space in the cache to accommodate more entries.

	Misses: Indicates the number of times this cache was missed for an entry during the last measurement period.	Number	A low value is desired for this measure. A high value or a consistent increase in this value is a cause for concern, as it indicates ineffective cache usage. This means high disk accesses and poor server performance. One of the common reasons for bad cache usage is improper cache size. A cache that does not have sufficient space to accommodate entries will not be able to service requests effectively. You may hence have to allocate more space to the cache or free-up space in the cache to accommodate more entries.

1.7 The HANA System Layer

The **HANA System** layer provides you with a wealth of knowledge related to how well the files present in the disks of the SAP HANA database server is utilized, the CPU utilization od the SAP HANA database host, the memory utilization od the SAP HANA database host and the usage statistics of each type of volume file in the SAP HANA database server.

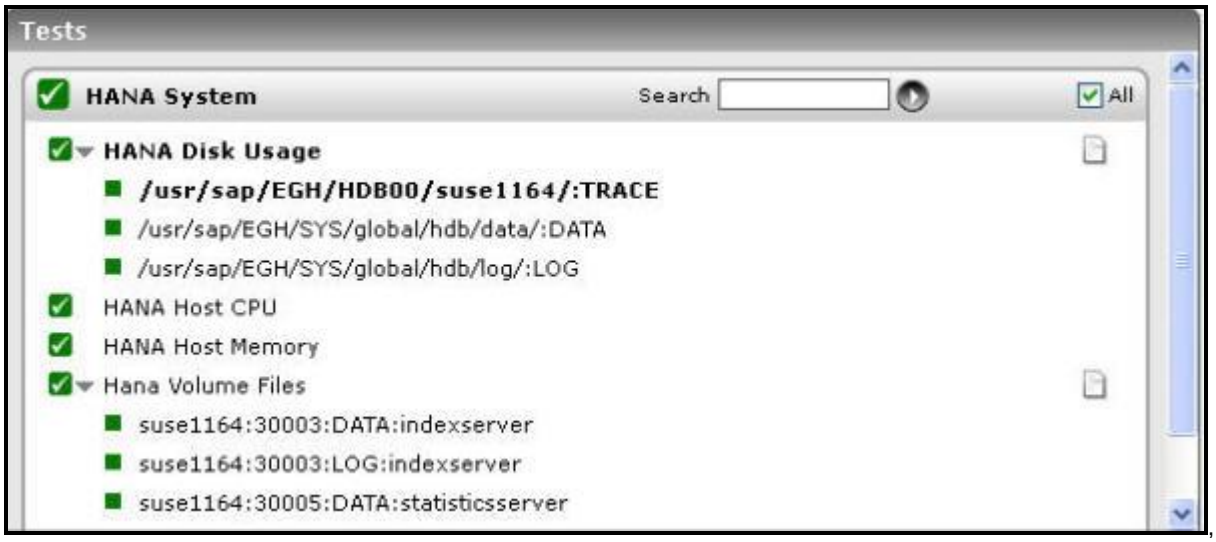

Figure 1.8: The tests mapped to the HANA System layer

1.7.1 HANA Disk Usage Test

Since the SAP HANA database resides in volatile main memory, a persistency layer in the SAP HANA database engine ensures the durability of the transactions. In the event of a failure the persistency layer ensures that the database can be restored to the most recent committed state after a restart. Transactions are either completely executed or completely un-done. To ensure durability of data changes completed in the memory of the SAP HANA server nodes, database data is written to disks. The persistency layer stores data in disk volumes, organized in pages. There are two types of volumes: **DATA** and **LOG**. The data volume contains data pages and undo log information. The log volume only records redo log information. In addition, if tracing is enabled, a **TRACE** volume is created, which stores trace files that aid in system troubleshooting.

To enable the SAP HANA database server to provide reliable and uninterrupted data management services, each of these disk volumes should have adequate free space. If a SAP HANA database server were to rapidly run out of disk space, and administrators had no knowledge of this anomaly until it was too late, it could result in the loss of critical data, much to the displeasure of database users. To avoid this, the space usage in each disk volume of the SAP HANA database server host should be closely monitored and administrators forewarned of space contentions. The **HANA Disk Usage** test makes this possible! This test monitors the space usage in each disk volume of the SAP HANA server, proactively detects a potential space crunch in any volume, and promptly alerts administrators to the same, so that the disk space crisis can be averted.

Purpose	Monitors the space usage in each disk volume of the SAP HANA server, proactively detects a potential space crunch in any volume, and promptly alerts administrators to the same, so that the disk space crisis can be averted		
Target of the test	A SAP HANA database server		
Agent deploying the test	A remote agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST -** The host for which the test is to be configured. 3. **PORT** – The port number at which the specified **HOST** listens to. The default port is *30015*. 4. **USER** – In order to monitor a SAP HANA database server, a special database user account with *Monitoring* privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is: *create user <username> password <password>* *grant monitoring to <username>* A sample script for user creation is: *create user sapegadmin password eGurkha@123;* *grant monitoring to sapegadmin;* The name of this user has to be specified here. 5. **PASSWORD –** The password of the specified **USER**. 6. **CONFIRM PASSWORD –** Confirm the password by retyping it here. 7. **ISPASSIVE –** If the value chosen for this parameter is **Yes**, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up.		
Outputs of the test	One set of results for every disk volume on the SAP HANA database server host being monitored; each disk volume is represented by *<Full_path_to_the_disk>:<VolumeType>*, where the *VolumeType* can be **DATA**, **LOG**, or **TRACE**.		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**Total Size:** Indicates the total size of this disk volume.	MB	
	Free Size: Indicates the amount of space that is unused in this disk volume.		A high value is desired for this measure.
	Free Percentage: Indicates the percent of space that is currently unused in this disk volume.	Percent	A consistent decrease in the value of this measure is indicative of gradual erosion of space in a disk volume. You may want to free up more space by either adding more disks or by removing unnecessary or obsolete files from the disk. This depends upon the *VolumeType*. For instance, in case of the **DATA** volume, you must not delete files from this volume manually, as it can corrupt the SAP HANA system. On the other hand, if the **TRACE** volume runs out of space, you can delete the trace files to make room for more trace data.
	Growth: Indicates the percentage growth in space usage in this disk volume since the last measurement period.	Percent	If a consistent growth is noticed in the value of this measure, it is indicative of gradual erosion of space in a disk volume. You may want to free up more space by either adding more disks or by removing unnecessary or obsolete files from the disk. This depends upon the *VolumeType*. For instance, in case of the **DATA** volume, you must not delete files from this volume manually, as it can corrupt the SAP HANA system. On the other hand, if the **TRACE** volume runs out of space, you can delete the trace files to make room for more trace data.

1.7.2 HANA Host CPU Test

Without adequate processing power, a SAP HANA database server can stop functioning. This is why, it is imperative that administrators keep an eye on the CPU usage of the database server, so that if CPU usage exceeds permissible limits, it can be quickly detected and treated to ensure optimal performance of the server. This is where the **HANA Host CPU** test helps. This test monitors the overall CPU utilization of the SAP HANA database server and reveals where the maximum CPU time is spent – in serving user requests? in system-level processing? in waiting for I/O? or being idle? This way, the test not only points administrators to the excessive CPU usage by the SAP HANA server, but also indicates the reason for the same, so that administrators can initiate the correct remedial measures.

Purpose	Monitors the overall CPU utilization of the SAP HANA database server and reveals where the maximum CPU time is spent – in serving user requests? in system-level processing? in waiting for I/O? or being idle?

<table>
<tr><td>Target of the test</td><td colspan="3">A SAP HANA database server</td></tr>
<tr><td>Agent deploying the test</td><td colspan="3">A remote agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as “Not applicable” by the agent if the server is not up.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every SAP HANA database server being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>CPU Usage:
Indicates the percentage of CPU that is currently utilized by the SAP HANA database server.</td><td>Percent</td><td>A value close to 100% hints a potential CPU contention at the host, which must be cleared pre-emptively, to ensure that the performance of the server is not compromised.</td></tr>
<tr><td>System CPU:
Indicates the percentage of CPU time the SAP HANA server spent in system-level processing.</td><td>Percent</td><td>Compare the value of this measure with the value of the User CPU, Wait I/O CPU, and Idle CPU measures to understand where SAP HANA spent the maximum CPU time - in serving user requests? in system-level processing? in waiting for I/O? or being idle? In the event of a slowdown, you can use the results of this comparative analysis to determine what caused the CPU contention.</td></tr>
</table>

	User CPU: Indicates the percentage of CPU time that the host spent in user-level processing.	Percent	Compare the value of this measure with the value of the *System CPU*, *Wait I/O CPU*, and *Idle CPU* measures to understand where SAP HANA spent the maximum CPU time - in serving user requests? in system-level processing? in waiting for I/O? or being idle? In the event of a slowdown, you can use the results of this comparative analysis to determine what caused the CPU contention.
	Wait I/O CPU: Indicates the percentage of time the CPU had to wait for I/O operations on this SAP HANA database host.	Percent	Compare the value of this measure with the value of the *System CPU*, *User CPU*, and *Idle CPU* measures to understand where SAP HANA spent the maximum CPU time - in serving user requests? in system-level processing? in waiting for I/O? or being idle? In the event of a slowdown, you can use the results of this comparative analysis to determine what caused the CPU contention.
	Idle CPU: Indicates the percentage of time the CPU was idle on this SAP HANA database host.	Percent	Compare the value of this measure with the value of the *System CPU*, *User CPU*, and *Wait I/O CPU* measures to understand where SAP HANA spent the maximum CPU time - in serving user requests? in system-level processing? in waiting for I/O? or being idle? In the event of a slowdown, you can use the results of this comparative analysis to determine what caused the CPU contention.

1.7.3 HANA Host Memory Test

By proactively detecting a potential memory contention on a SAP HANA server, administrators can easily avert slowdowns or outages of the SAP HANA database server. The **HANA Host Memory** test helps administrators in this exercise. This test reports how well the SAP HANA server uses the physical and swap memory allocated to it, and leads administrators to abnormal memory usage patterns (if any).

Purpose	Reports how well the SAP HANA server uses the physical and swap memory allocated to it, and leads administrators to abnormal memory usage patterns (if any)
Target of the test	A SAP HANA database server
Agent deploying the test	A remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable" by the agent if the server is not up.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every SAP HANA database server being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Total Physical Memory:
Indicates the total amount of physical memory allocated for the SAP HANA database server.</td><td>MB</td><td></td></tr>
<tr><td>Free Physical Memory:
Indicates the amount of physical memory that is currently available on the SAP HANA database server.</td><td>MB</td><td>A high value is desired for this measure.</td></tr>
<tr><td></td><td>Used Physical Memory:
Indicates the amount of physical memory that is currently used by the SAP HANA database server.</td><td>MB</td><td>Ideally, the value of this measure should be low. A consistent increase in this value could be a cause for some serious concern, as it indicates a gradual, but steady erosion of valuable memory resources. If this unhealthy trend is not repaired soon, it could severely hamper server performance, causing anything from a slowdown to a complete system meltdown.</td></tr>
</table>

	Total Swap Space: Indicates the total amount of swap memory allocated for the SAP HANA database server.	MB	Swap space is space on a hard disk used as the virtual memory extension of a computer's real memory (RAM). The least recently used files in RAM can be "swapped out" to the hard disk until they are needed later so that new files can be "swapped in" to RAM.
	Free Swap Space: Indicates the amount of swap memory that is currently available for the SAP HANA database server.	MB	Ideally, the value of this measure should be high, as having an appropriate amount of swap space is important for optimal system performance.
	Used Swap Space: Indicates the amount of swap memory that is currently used by the SAP HANA database server.	MB	A consistent rise in the value of this measure is a cause for concern.
	Percentage of Free Physical Memory: Indicates the percentage of physical memory that is currently available on the SAP HANA database server.	Percent	A high value is desired for this measure. A consistent drop in this value is a sign that physical memory is being over-utilized by the server.
	Percentage of Free Swap Space: Indicates the percentage of swap memory that is currently available on the SAP HANA database server.	Percent	A very low value indicates that the swap space configured may not be sufficient. A value close to 100% may imply that the swap space configured may be too large.

1.7.4 HANA Volume Files Test

Since the SAP HANA database resides in volatile main memory, a persistency layer in the SAP HANA database engine ensures the durability of the transactions. In the event of a failure the persistency layer ensures that the database can be restored to the most recent committed state after a restart. Transactions are either completely executed or completely un-done. To ensure durability of data changes completed in the memory of the SAP HANA server nodes, database data is written to disks. The persistency layer stores data in disk volumes, organized in pages. There are two types of volumes: data and log. Every process/service in the SAP HANA database which needs to persist data, such as the name, index and statistic servers, writes regularly into its own data and log volume. The data volume contains data pages and undo log information. The log volume only records redo log information. In the event of a crash or recovery, redo log information is used to roll forward changes that have not been persisted on the data pages while undo log information is used to roll back changes for uncommitted transactions. If the data or log volume mapped to a service does not have sufficient free space, then these volumes will not be able to store critical undo/redo log details and data pages, thus causing the disaster recovery engine of the database server to fail! This in turn may cause critical data loss and service failures! To prevent such irredeemable losses, it is of utmost importance to closely monitor the space usage in the data and log volumes of each service, detect a space crunch even before it occurs, and right-size the volumes accordingly. To help administrators with this, the eG agent runs the **HANA Volume Files** test on the SAP HANA database server. For each service of the SAP HANA database server, this test reports how that service uses each of its disk volumes (data or log), thereby pointing administrators to services

that are over-utilizing their volumes and may require additional space if any permanent damage to performance is to be averted.

<table>
<tr><td>Purpose</td><td colspan="3">For each service of the SAP HANA database server, this test reports how that service uses each of its disk volumes (data or log), thereby pointing administrators to services that are over-utilizing their volumes and may require additional space if any permanent damage to performance is to be averted</td></tr>
<tr><td>Target of the test</td><td colspan="3">A SAP HANA database server</td></tr>
<tr><td>Agent deploying the test</td><td colspan="3">A remote agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. PORT – The port number at which the specified HOST listens to. The default port is 30015.
4. USER – In order to monitor a SAP HANA database server, a special database user account with Monitoring privileges has to be created in every SAP HANA database instance that requires monitoring. The syntax of the script that is used for user creation is:
create user <username> password <password>
grant monitoring to <username>
A sample script for user creation is:
create user sapegadmin password eGurkha@123;
grant monitoring to sapegadmin;
The name of this user has to be specified here.
5. PASSWORD – The password of the specified USER.
6. CONFIRM PASSWORD – Confirm the password by retyping it here.
7. ISPASSIVE – If the value chosen for this parameter is Yes, then the SAP HANA database server under consideration is a passive server in a SAP HANA cluster. No alerts will be generated if the server is not running. Measures will be reported as “Not applicable” by the agent if the server is not up.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every volume type (data or log) of each service of the SAP HANA database server being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Used Size:
Indicates the amount of space that this service is currently using in this volume.</td><td>MB</td><td>A low value is desired for this measure.</td></tr>
</table>

	Used Percent: Indicates the percentage of space that this service is currently using in this volume.	Percent	A low value is desired for this measure. If a consistent growth is noticed in the value of this measure, it is indicative of gradual erosion of space in a disk volume. You may want to free up more space by either adding more disks or by removing unnecessary or obsolete files from the disk. This depends upon the volume type. For instance, in case of the **DATA** volume, you must not delete files from this volume manually, as it can corrupt the SAP HANA system. On the other hand, if the **TRACE** volume runs out of space, you can delete the trace files to make room for more trace data.
	Growth: Indicates the rate at which this volume file has grown since the last measurement period.	MB/Sec	If a consistent growth is noticed, it is indicative of gradual erosion of space in a disk volume. You may want to free up more space by either adding more disks or by removing unnecessary or obsolete files from the disk. This depends upon the volume type. For instance, in case of the **DATA** volume, you must not delete files from this volume manually, as it can corrupt the SAP HANA system. On the other hand, if the **TRACE** volume runs out of space, you can delete the trace files to make room for more trace data.

2

Conclusion

This document has described in detail the monitoring paradigm used and the measurement capabilities of the eG Enterprise suite of products with respect to the **Nginx Server**. For details of how to administer and use the eG Enterprise suite of products, refer to the user manuals.

We will be adding new measurement capabilities into the future versions of the eG Enterprise suite. If you can identify new capabilities that you would like us to incorporate in the eG Enterprise suite of products, please contact support@eginnovations.com. We look forward to your support and cooperation. Any feedback regarding this manual or any other aspects of the eG Enterprise suite can be forwarded to feedback@eginnovations.com.