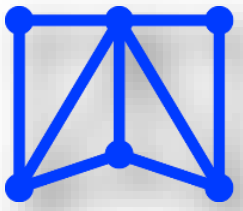


teqwave

PostgreSQL MP User Guide for Microsoft System Center Operations Manager

Version 4.1
User Guide,
Revision A

April, 2021

CONTENTS

CONTACTING TEQWAVE

At Teqwave we value the feedback from our customers. It is important not only to help you quickly with your technical issues, but it is our mission to listen to your input and build products that incorporate your suggestions.

CUSTOMER SUPPORT

Should you have a product issue, suggestion or question, please send an E-mail to the Teqwave support team at support@teqwave.com

ONLINE SUPPORT

If you have any questions about the Teqwave Management Pack for PostgreSQL, you may use the following resources:

- Support
  - http://teqwave.com/support/
- Online documentation
  - http://teqwave.com/resources/

ABOUT THIS DOCUMENT

This document describes the features included in the Teqwave Management Pack for PostgreSQL. It gives instructions for installing the Management Pack and monitoring your PostgreSQL database(s) in Microsoft System Center Operations Manager.

WHAT'S NEW IN THIS RELEASE

Version	Release Date	
1.0	July 2017	The initial version of the Teqwave Management Pack for PostgreSQL.
1.1	November 2017	Support for PostgreSQL 10.x Summary dashboards added
1.2	January 2018	Added some reports Various Fixes
1.3	April 2018	Support for SCOM 1801 Various Fixes
2.0	September 2018	Added replication monitoring (physical and logical) Support for SCOM 1807 Validated up to PostgreSQL 10.5 Various Fixes
3.0	November 2018	Added monitoring of PostgreSQL Servers running on Windows platform
3.1	March 2019	Database forecast reports added Improved knowledge base for several monitors Various Fixes
3.2	April 2019	Support for SCOM 2019 Support for PostgreSQL 11.x Various Fixes
4.0	December 2019	Custom query feature added (possibility to add custom queries and monitor/collect additional metrics) Support for PostgreSQL 12.x Various Fixes
4.1	April 2021	Support for PostgreSQL in containers Support for PostgreSQL 13.x Various Fixes

WELCOME TO TEQWAVE POSTGRESQL MP

The **Teqwave Management Pack for PostgreSQL** (**Teqwave PostgreSQL MP**) — integrates PostgreSQL database into the enterprise-wide automated management, monitoring and alerting workflow implemented using Microsoft System Center Operations Manager (Ops Mgr).

KEY FEATURES

The Teqwave PostgreSQL MP helps you monitor installations of PostgreSQL Server running on Linux or Windows computers that are managed by System Center Operations Manager.

The Teqwave PostgreSQL MP alerts you to problems with performance and availability so you can continuously monitor the PostgreSQL Servers on which your business depends.

The monitoring provided by this management pack includes availability monitoring, performance data collection, and default thresholds.

In addition to health monitoring capabilities, this management pack includes reports, diagnostics, and views that enable near real-time diagnosis and resolution of detected issues.

The Teqwave PostgreSQL MP provides the following advanced features:

- Discovers PostgreSQL components, including servers, instances, databases, and tablespaces.
- Proactively monitors the availability and capacity
    - Server and database connections,
    - Database / tablespace / instance size,
    - Available disk space, etc.
- Proactively notifies when the health is critical
    - Table or Index bloat,
    - Buffer cache hit ratio,
    - Number of temporary files,
    - Amount of data written to temporary files
    - Long running transactions,
    - Background writer stops, etc.
    - Streaming replication is not working
- Enables you to configure monitors and rules with custom SQL queries and monitor/collect additional metrics that are not included in MP by default
- Provides customizable dashboard views for monitoring the configuration, resource utilization and health state of the PostgreSQL components.
- Includes a set of performance and status monitors that diagnose the state of PostgreSQL resources.
- Includes an extensive knowledge base to speed up root-cause diagnosis and reduce resolution time for detected issues.

LICENSING

The Teqwave PostgreSQL MP is licensed on a per PostgreSQL server instance basis. All databases running on the server instance will be automatically monitored.

The license file is obtained from Teqwave and is a requirement during MP configuration. A free 30-day trial license is available.

SYSTEM REQUIREMENTS

Before you start installing the product, make sure your environment meets the following hardware and software requirements.

POSTGRESQL INFRASTRUCTURE

The Teqwave PostgreSQL MP supports the following PostgreSQL database versions:

Specification	Requirement

PostgreSQL	• 9.2 or newer on
	  ○ Linux platform (64-bit)
	  ○ Windows Server 2012 R2 (64-bit)
	  ○ Windows Server 2016 (64-bit)
	  ○ Windows Server 2019 (64-bit)
	• 10.x on
	  ○ Linux platform (64-bit)
	  ○ Windows Server 2012 R2 (64-bit)
	  ○ Windows Server 2016 (64-bit)
	  ○ Windows Server 2019 (64-bit)
	• 11.x on
	  ○ Linux platform (64-bit)
	  ○ Windows Server 2012 R2 (64-bit)
	  ○ Windows Server 2016 (64-bit)
	  ○ Windows Server 2019 (64-bit)
	• 12.x on
	  ○ Linux platform (64-bit)
	  ○ Windows Server 2012 R2 (64-bit)
	  ○ Windows Server 2016 (64-bit)
	  ○ Windows Server 2019 (64-bit)
	• 13.x on
	  ○ Linux platform (64-bit)
	  ○ Windows Server 2016 (64-bit)
	  ○ Windows Server 2019 (64-bit)

MICROSOFT SYSTEM CENTER OPERATIONS MANAGER

The Teqwave PostgreSQL MP supports the following versions and components of Operation Manager:

Specification	Requirement
Operations Manager	• Microsoft System Center 2012 R2 Operations Manager • Microsoft System Center 2016 Operations Manager • Microsoft System Center 1801 Operations Manager • Microsoft System Center 1807 Operations Manager • Microsoft System Center 2019 Operations Manager **Note:** Make sure that the latest available updates for System Center Operations Manager are installed.

Custom management packs usually have dependencies on some of the default management packs. The Teqwave PostgreSQL MPs have the following dependencies:

Management Pack	Dependencies
Teqwave Management Pack for PostgreSQL (Library)	• Health Library • Instance Group Library • Microsoft System Center Visualization Library • System Center Core Library • System Library • Windows Core Library
Teqwave Management Pack for PostgreSQL on Linux (Core)	• Linux Operating System Library • System Center Core Library • System Library • Teqwave Management Pack for PostgreSQL (Library) • Unix/Linux Core Library
Teqwave Management Pack for PostgreSQL (Presentation)	• Microsoft System Center Visualization Configuration Library • Microsoft System Center Visualization Library • System Library • Teqwave Management Pack for PostgreSQL (Library)
Teqwave Management Pack for PostgreSQL (Advanced Dashboards)	• Microsoft SQLServer Visualization Library 6.6.0 or newer • Microsoft System Center Visualization Library • System Library • Teqwave Management Pack for PostgreSQL (Library)

Teqwave Management Pack for PostgreSQL on Linux (Monitoring)	• Data Warehouse Library • Health Library • Instance Group Library • Linux Operating System Library • Performance Library • System Center Core Library • System Library • Teqwave Management Pack for PostgreSQL (Library) • Unix/Linux Core Library • WS-Management Library
Teqwave Management Pack for PostgreSQL on Linux (Reports)	• Data Warehouse Library • System Library • Teqwave Management Pack for PostgreSQL (Library) • Teqwave Management Pack for PostgreSQL on Linux (Monitoring)
Teqwave Management Pack for PostgreSQL on Windows (Discovery)	• Instance Group Library • Microsoft System Center Operations Manager Library • Microsoft System Center Visualization Library • Performance Library • System Center Core Library • System Library • Teqwave Management Pack for PostgreSQL (Library) • Windows Core Library
Teqwave Management Pack for PostgreSQL on Windows (Monitoring)	• Data Warehouse Library • Health Library • Performance Library • System Center Core Library • System Library • Teqwave Management Pack for PostgreSQL (Library) • Teqwave Management Pack for PostgreSQL on Windows (Discovery)
Teqwave Management Pack for PostgreSQL on Windows (Reports)	• Microsoft Generic Report Library • System Library • Teqwave Management Pack for PostgreSQL on Windows (Discovery) • Teqwave Management Pack for PostgreSQL on Windows (Monitoring)

INSTALLING/UPGRADING POSTGRESQL MP

Teqwave PostgreSQL MP can monitor PostgreSQL servers running on Linux and Windows platforms. The first part of the installation/upgrade is the same for both platforms but the configuration is different. Please follow the configuration instructions for the specific platform. In case you are using a mixed environment (Linux and Windows PostgreSQL servers), you have to follow configuration instructions for both platforms.

To deploy the Teqwave PostgreSQL MP, follow these steps:

1. Install PostgreSQL MP on a Management Server
2. Import PostgreSQL MP Management Packs
3. Configure PostgreSQL MP
    a. (Linux platform only) Deploy PostgreSQL OMI Provider
    b. (Linux platform only) Configure PostgreSQL OMI Provider
    c. (Windows platform only) Configure PostgreSQL
4. License PostgreSQL MP Management Pack

If you are upgrading the Teqwave PostgreSQL MP from previous versions, follow these steps:

1. Install PostgreSQL MP on a Management Server

Note: You may be asked to close the System Center Operations Manager console.

2. Remove old Management Packs – please follow the detailed instructions for the specific version
3. Import new PostgreSQL MP Management Packs
4. (Linux platform only) Upgrade PostgreSQL OMI Provider

Please see the details of each step in the following sections.

BEFORE YOU BEGIN

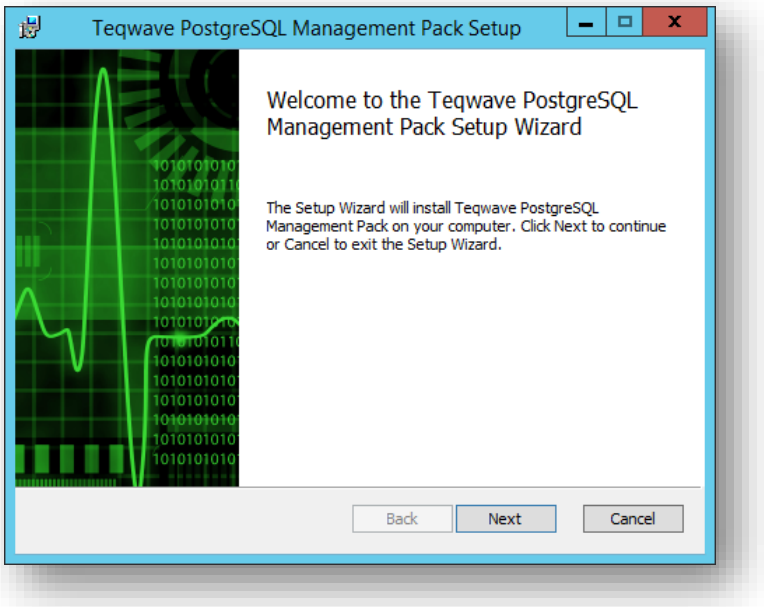
Before you begin the installation, take the following steps:

1. Make sure that your environment meets the prerequisite conditions described in section System Requirements.
2. Prepare PostgreSQL user account that will be used for monitoring – see Appendix A (Linux platform only) If you are using sudo-enabled accounts for Operations Manager monitoring, see Appendix B

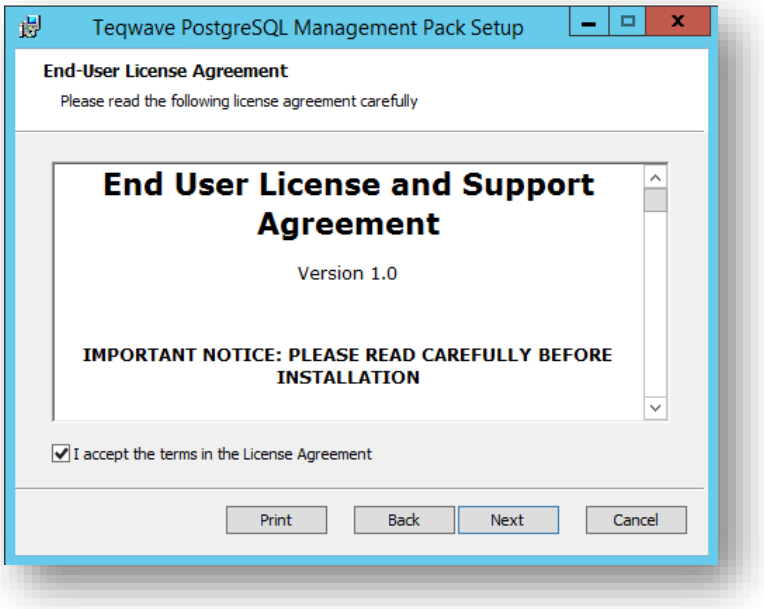
STEP 1 - INSTALL POSTGRESQL MP ON A MANAGEMENT SERVER

Log on to the Management Server using an account with local Administrator rights and launch the *Teqwave.PostgreSQL_MP.msi* setup package.

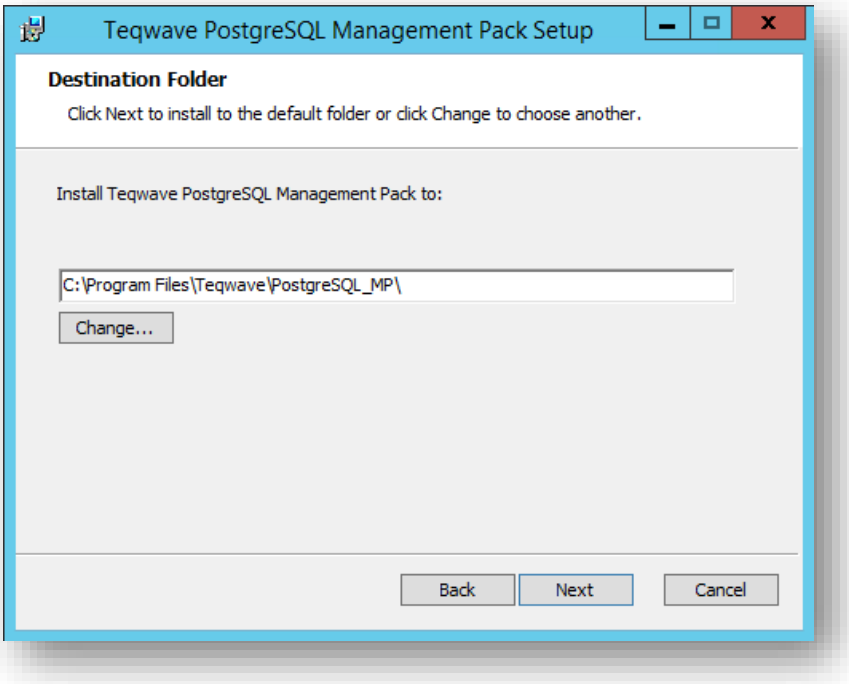
1. Click **Next** to start the installation.



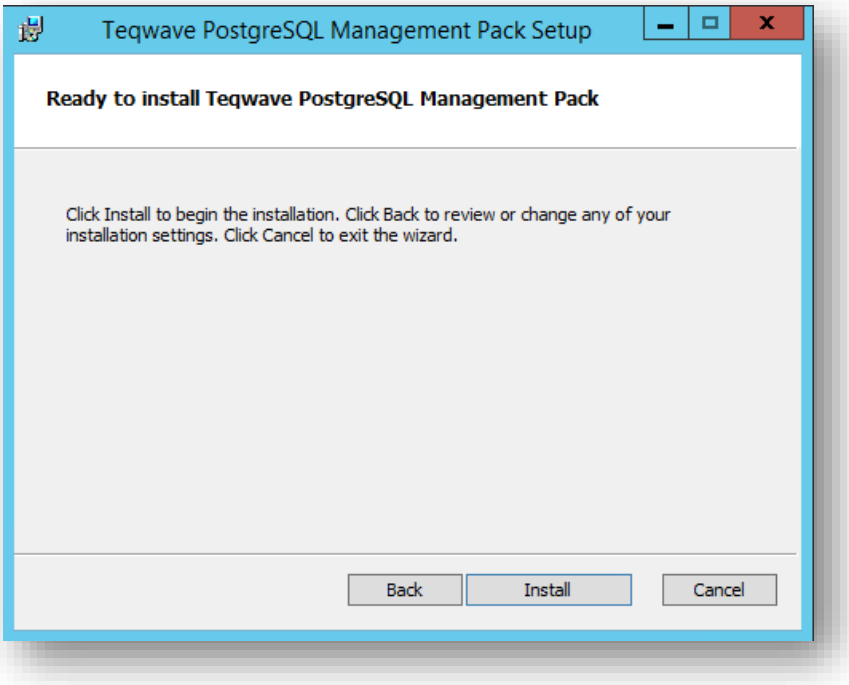
2. Read and accept the license agreement. If you reject the agreement, you will not be able to continue the installation.



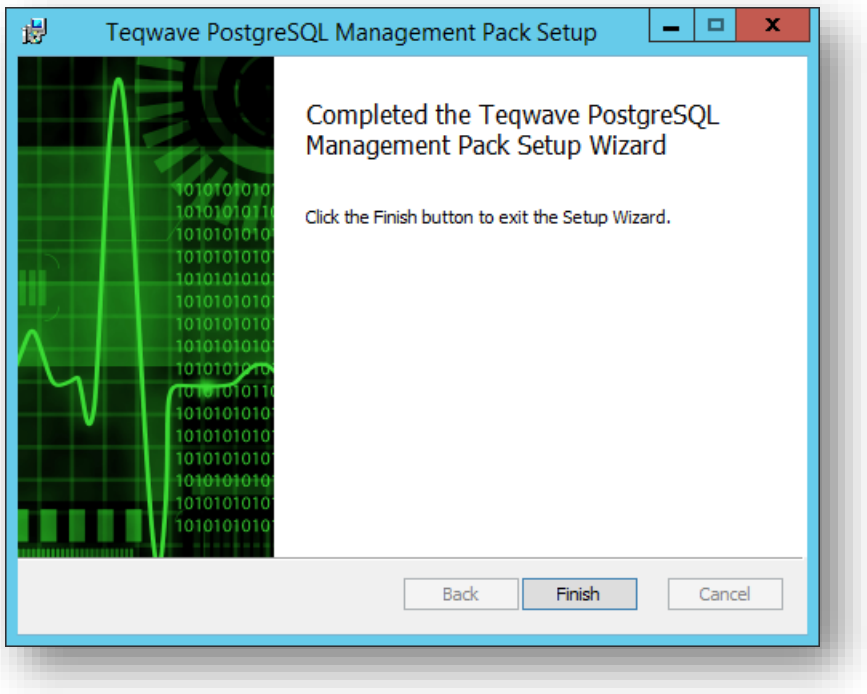
3. Click **Next** to install to the default folder or click **Change** to choose another folder.



4. Click **Install** to begin the installation.



5. Click **Finish**.



STEP 2 - IMPORT POSTGRESQL MANAGEMENT PACKS

Note: If you are upgrading Management Pack from versions 1.x or 2.x, please remove the following MPs from the SCOM before continuing with the instructions below:

- Teqwave Management Pack for PostgreSQL on Linux (Monitoring)
- Teqwave Management Pack for PostgreSQL on Unix/Linux (Reports)
- Teqwave Management Pack for PostgreSQL on Unix/Linux (Advanced Dashboards)
- Teqwave Management Pack for PostgreSQL on Unix/Linux (Presentation)

The following table describes the files included in this management pack.

File	Display name	Description
Teqwave.PostgreSQL.Library.mpb	Teqwave Management Pack for PostgreSQL (Library)	This management pack provides definitions of PostgreSQL core classes.

Teqwave.PostgreSQL.Linux.Core.mpb	Teqwave Management Pack for PostgreSQL on Linux (Core)	This is PostgreSQL management pack that provides Linux SCOM agent resources for monitoring PostgreSQL on Linux.
Teqwave.PostgreSQL.Presentation.mpb	Teqwave Management Pack for PostgreSQL (Presentation)	This management pack provides dashboards showing the PostgreSQL database and server summary.
Teqwave.PostgreSQL.PresentationAdv.mp	Teqwave Management Pack for PostgreSQL (Advanced Dashboards)	This management pack provides advanced dashboards showing PostgreSQL hosts, instances and databases.
Teqwave.PostgreSQL.Unix.Monitoring.mp	Teqwave Management Pack for PostgreSQL on Linux (Monitoring)	This management pack provides functionality for monitoring PostgreSQL on Linux.
Teqwave.PostgreSQL.Unix.Reporting.mp	Teqwave Management Pack for PostgreSQL on Linux (Reports)	This management pack contains PostgreSQL reports for Linux server instances.

Teqwave.PostgreSQL.Windows.Discovery.mpb	Teqwave Management Pack for PostgreSQL on Windows (Discovery)	This management pack discovers PostgreSQL components on Windows.
Teqwave.PostgreSQL.Windows.Monitoring.mpb	Teqwave Management Pack for PostgreSQL on Windows (Monitoring)	This management pack provides functionality for monitoring PostgreSQL on Windows.
Teqwave.PostgreSQL.Windows.Reporting.mp	Teqwave Management Pack for PostgreSQL on Windows (Reports)	This management pack contains PostgreSQL reports for Windows server instances.

To import the Teqwave PostgreSQL MPs to the System Center Operations Manager, perform the following steps:

1. On the **Management Server**, start the **System Center Operations Manager** console.
2. In the Operations console, click **Administration**.
3. Right-click the **Management Packs** node, and then click **Import Management Packs**.
4. The **Import Management Packs** wizard opens. Click **Add**, and then click **Add from disk**.
5. The **Select Management Packs to import** dialog box appears. Locate the management packs in the MP installation directory `%ProgramFiles%\Teqwave\PostgreSQL_MP`, select all MP files and then click **Open**.
6. On the **Select Management Packs** page, the management packs that you selected for import are listed. An icon next to each management pack in the list indicates the status of the selection, click **Install**.
7. The **Import Management Packs** page appears and shows the progress for each management pack. If there is a problem at any stage of the import process, select the management pack in the list to view the status details. Click **Close**.

Note: Import is possible only if all required management packs are available. Missing management packs can be imported from the System Center Operations Manager installation directory.

Note: If you are upgrading Management Pack from version 3.x, please remove the following MPs after all above steps have been completed:

- Teqwave Management Pack for PostgreSQL on Windows Server 2012 R2 (Discovery)
- Teqwave Management Pack for PostgreSQL on Windows Server 2016+ (Discovery)

These two Management Packs are not used anymore since version 3.2, therefore you can remove them from the SCOM system.

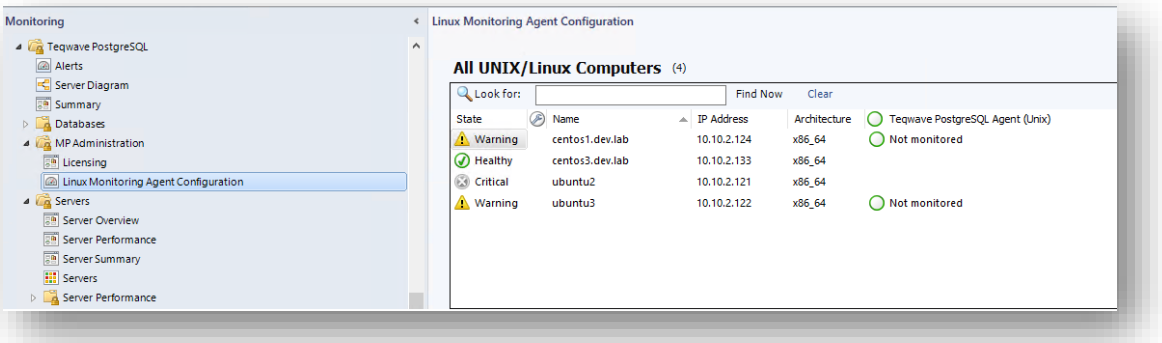
STEP 3A - DEPLOY/UPGRADE POSTGRESQL OMI PROVIDER (LINUX PLATFORM ONLY)

Note: This section is applicable only if you are running PostgreSQL server(s) on the Linux platform. You can skip this part if you are running PostgreSQL on Windows platform and continue with step 3C.

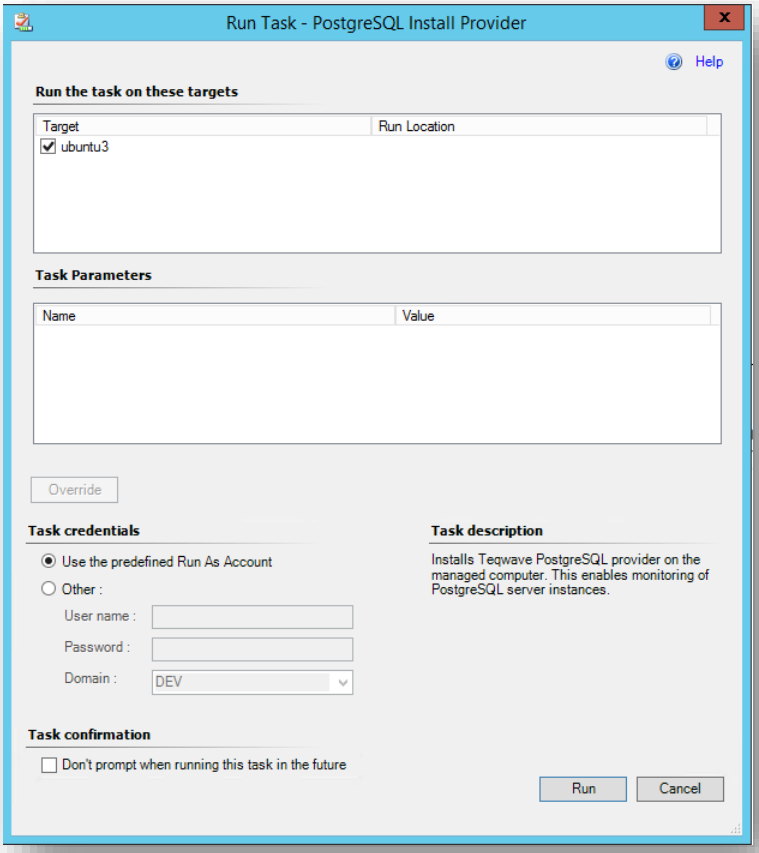
Monitoring of PostgreSQL Server requires that the corresponding OMI Provider is deployed to the managed computer. The PostgreSQL OMI provider uses a preconfigured PostgreSQL user and the PostgreSQL client library to retrieve performance and health data back to Operations Manager.

Note: If you are upgrading PostgreSQL OMI provider, it might take up to 12 hours for the provider installation package to be updated on the SCOM Management Servers. To speed this up, we recommend restarting Microsoft Monitoring Agent service on all SCOM Management Servers that are dedicated to monitoring PostgreSQL Linux servers.

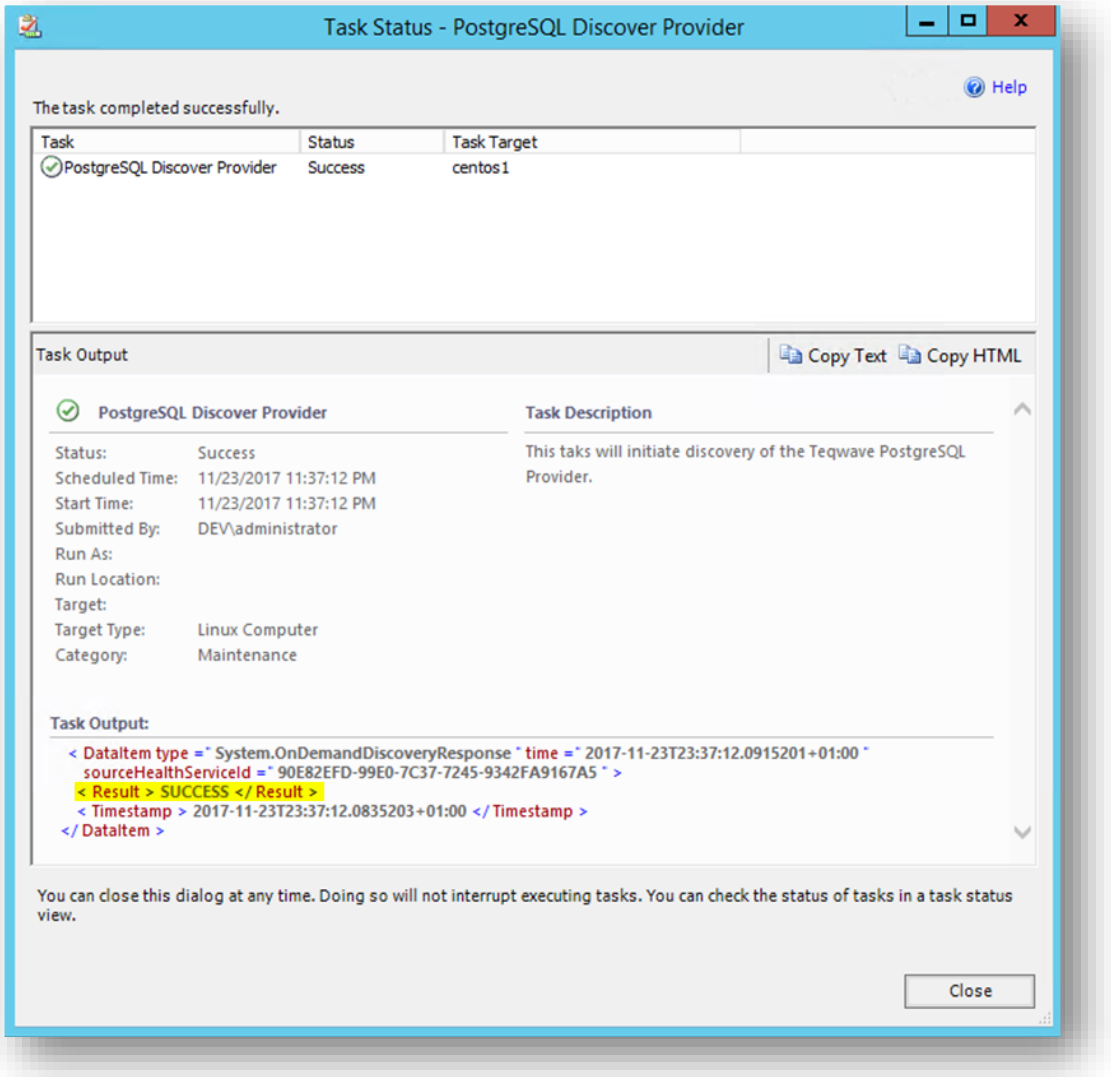
1. In SCOM Console navigate to **Monitoring** | **Teqwave PostgreSQL** | **MP Administration** | **Linux Monitoring Agent Configuration** | **All UNIX/Linux Computers** view and select servers on which you want to deploy PostgreSQL OMI provider



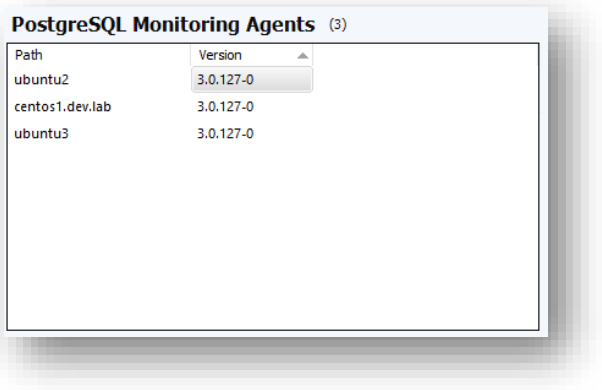
2. Run **PostgreSQL Install Provider** (or **PostgreSQL Upgrade Provider** if you are upgrading) task and check if there are any errors in the task output before continuing.



3. Run **PostgreSQL Discover Provider** task to discover the installed provider instantly. Check the task output for any errors.



4. If discover task is successful, refresh **PostgreSQL Monitoring Agents** view. The new host and the provider version should be listed in the view.

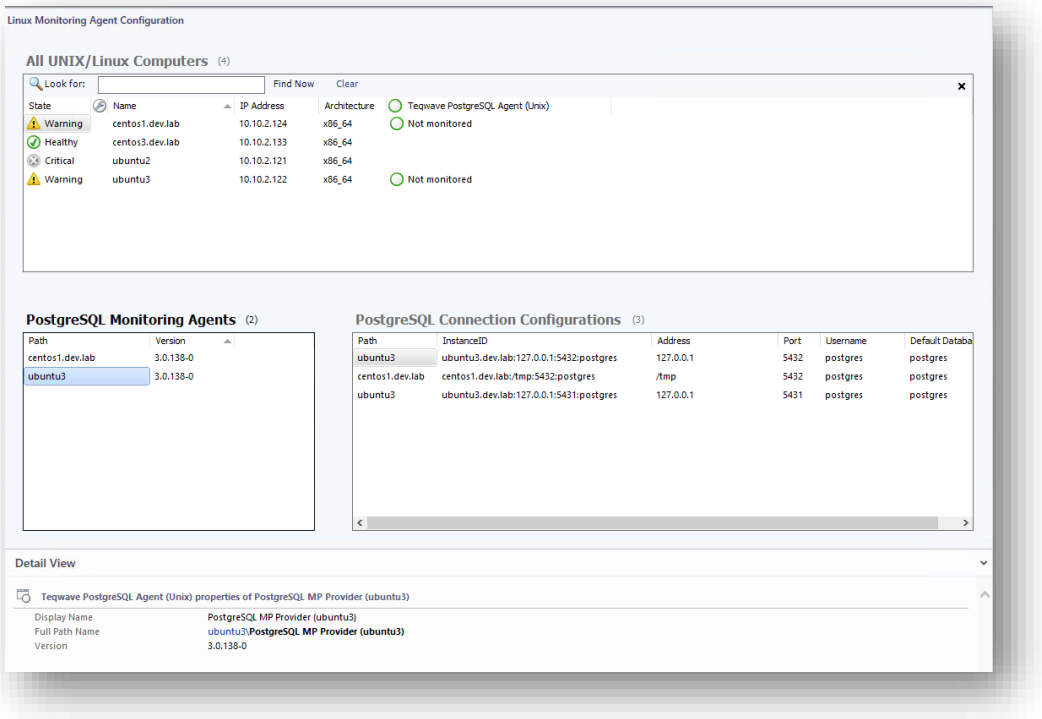


STEP 3B - CONFIGURE POSTGRESQL OMI PROVIDER (LINUX PLATFORM ONLY)

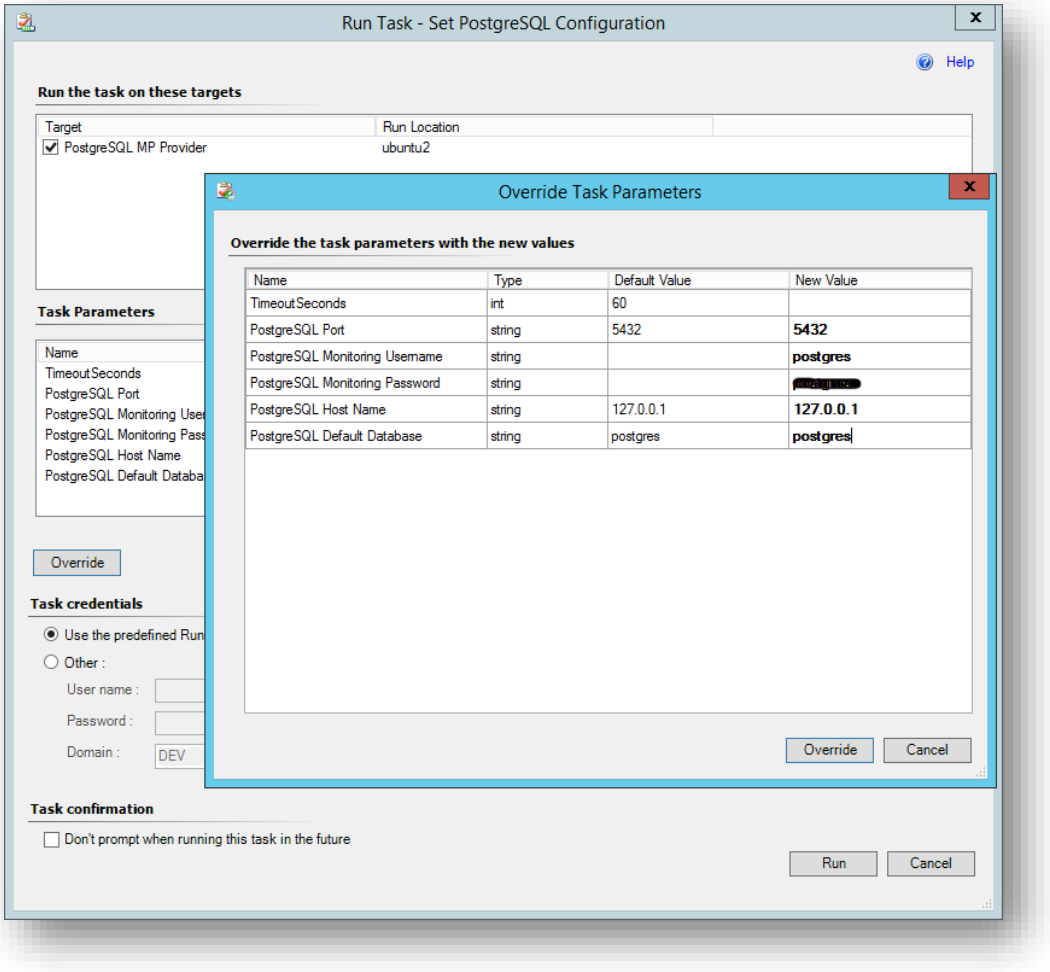
Note: This section is applicable only if you are running PostgreSQL server(s) on the Linux platform. You can skip this part if you are running PostgreSQL on Windows platform and continue with step 3C.

Note: You may also skip this part if you are upgrading the provider and you want to preserve old configuration.

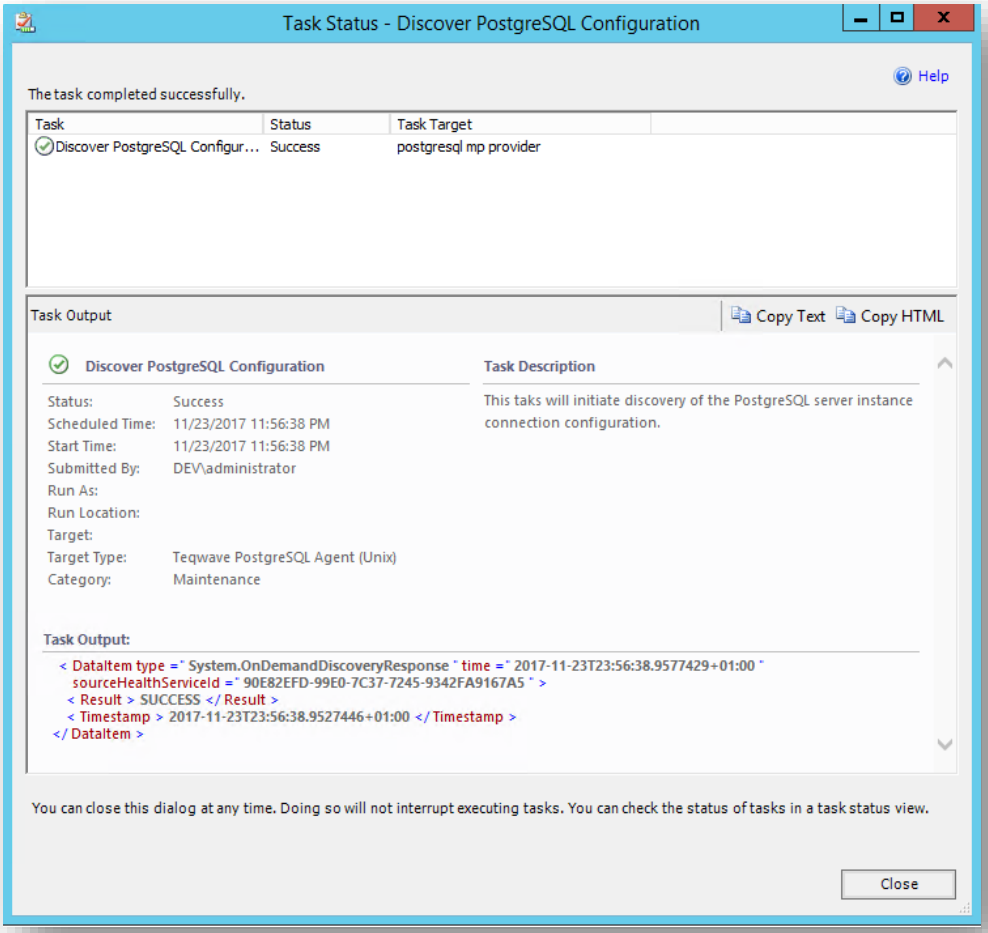
1. Navigate to **Monitoring** | **Teqwave PostgreSQL** | **MP Administration** | **Linux Monitoring Agent Configuration** | **PostgreSQL Monitoring Agents** view and select server on which you want to configure PostgreSQL OMI provider



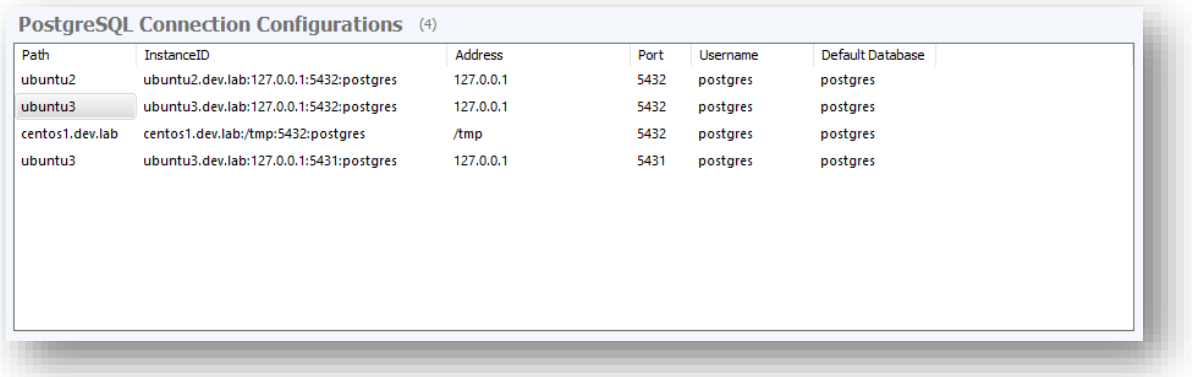
2. Run **Set PostgreSQL Configuration** task, click Override and enter **PostgreSQL Monitoring Username**, **PostgreSQL Monitoring Password,** and **PostgreSQL Database Server Host or Socket Directory** parameters.



3. Run **Discover PostgreSQL Configuration** task to discover the provider configuration instantly. Check the task output for any errors.



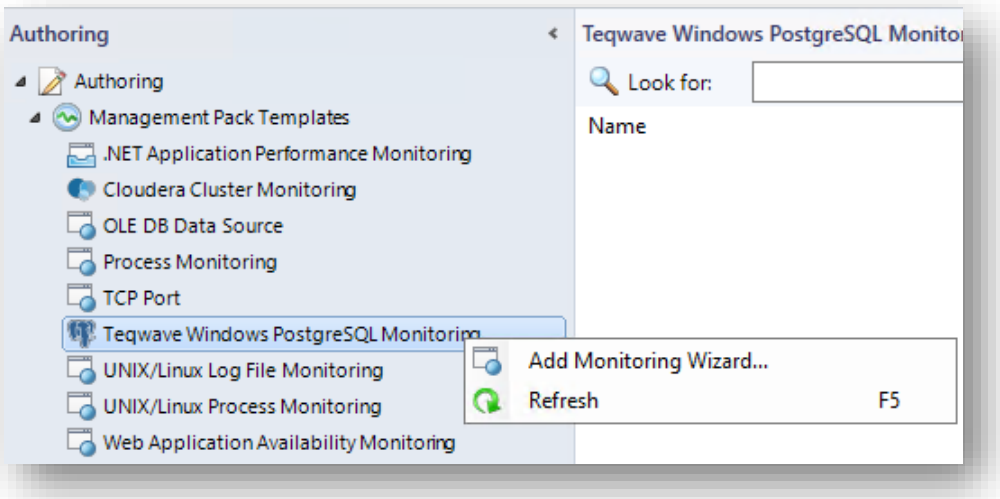
4. If discover was successful, you should be able to see the configuration details under the **PostgreSQL Connection Configurations** view.



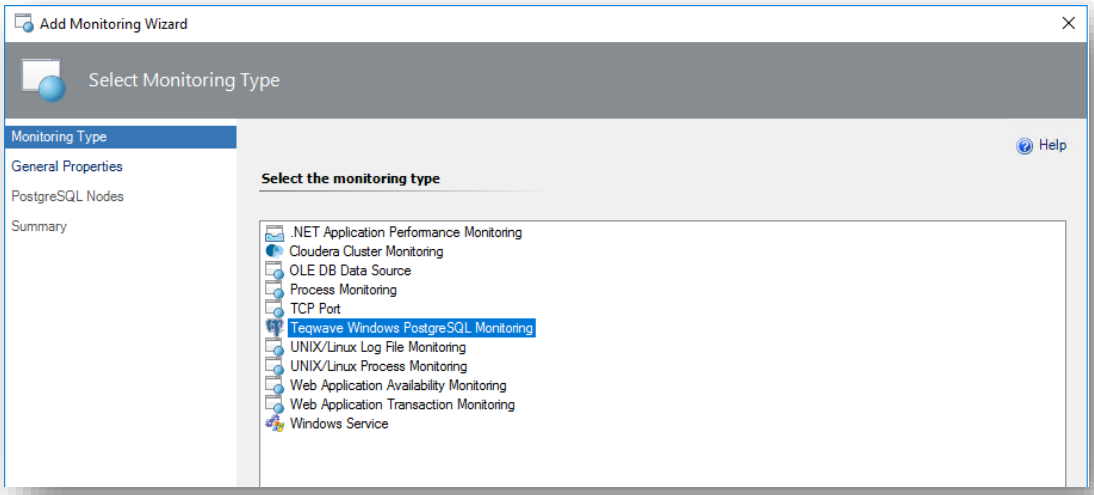
STEP 3C - CONFIGURE POSTGRESQL (WINDOWS PLATFORM ONLY)

Note: This section is applicable only if you are running PostgreSQL server(s) on the Windows platform. You can skip this part if you are running PostgreSQL on Linux platform.

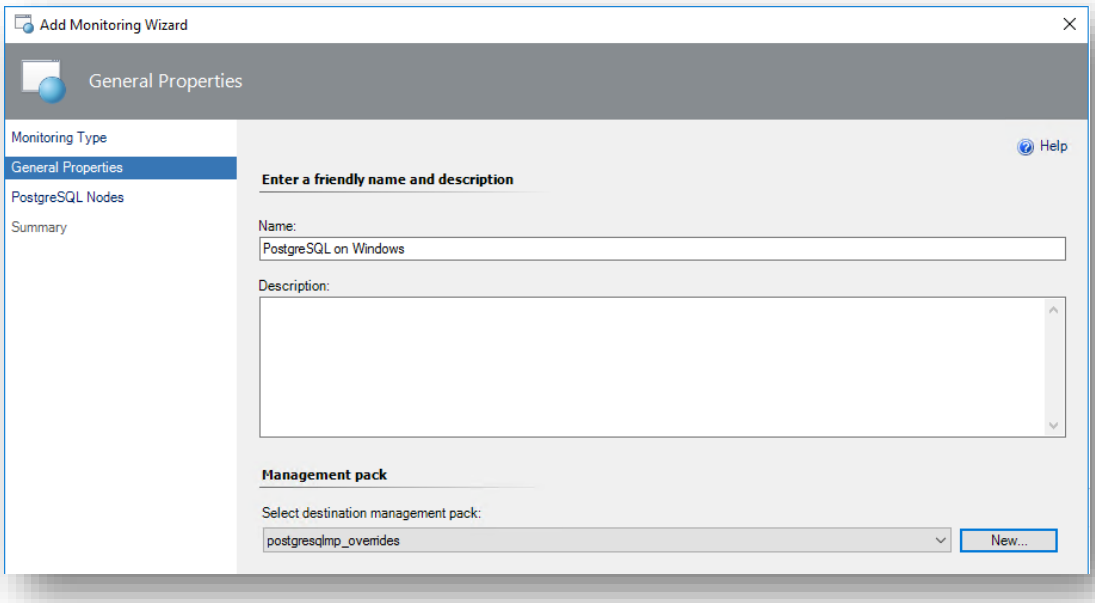
1. In SCOM Console navigate to **Authoring** | **Management Pack Templates** | **Teqwave Windows PostgreSQL Monitoring**, right-click it and select **Add Monitoring Wizard…**:



2. On the **Monitoring Type** page select **Teqwave Windows PostgreSQL Monitoring** and click **Next**:

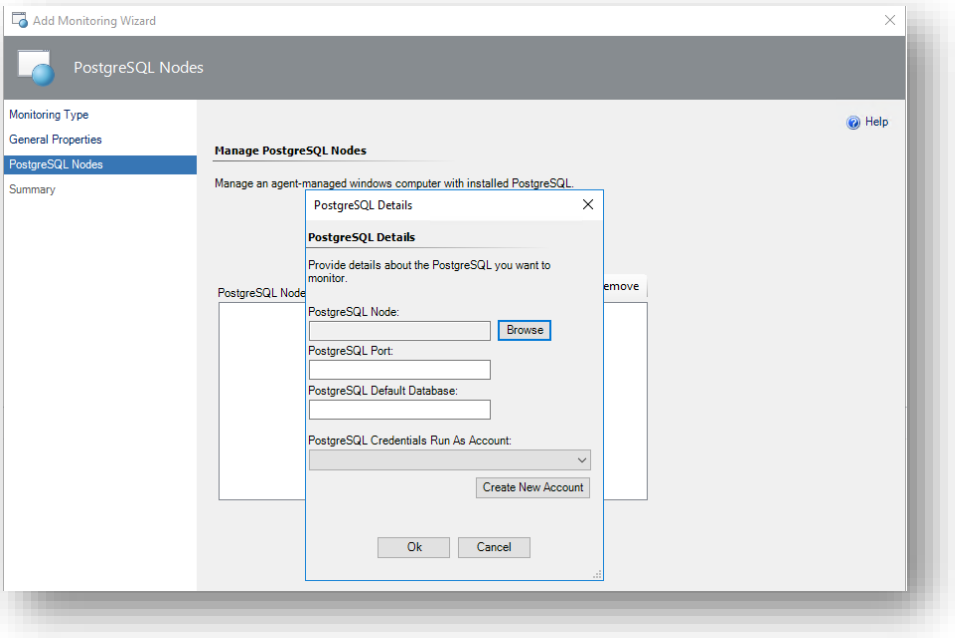


3. On the **General Properties** page provide **Name** and **Description** for your configuration template and select target **Management Pack** to store configuration to:

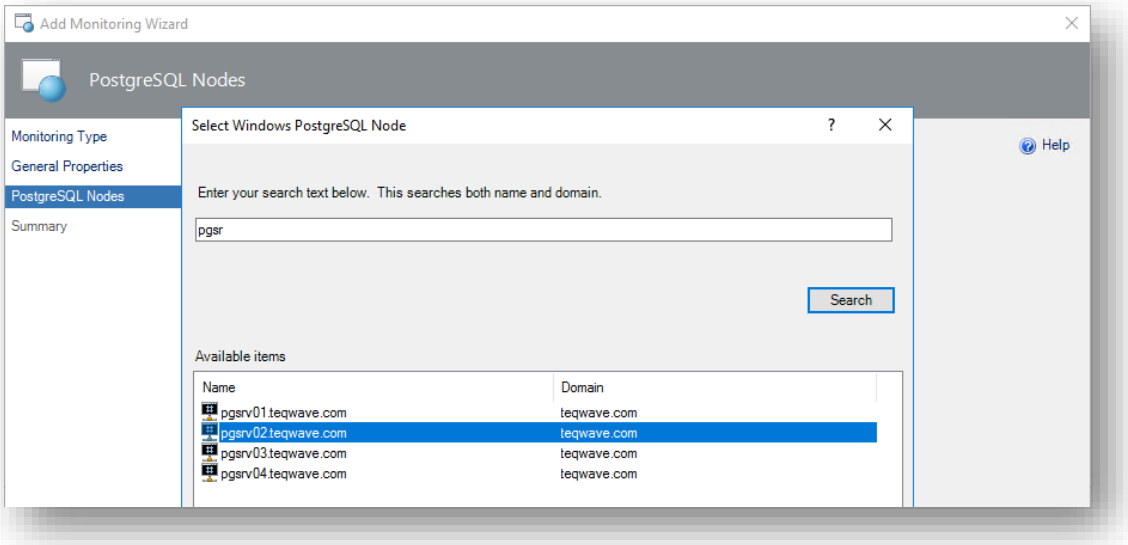


For more information about target Management Pack, please see Best Practice: Create a Management Pack for Customizations section. You can create a new management pack right from this wizard by clicking the **New** button, located next to Management Packs drop-down list.

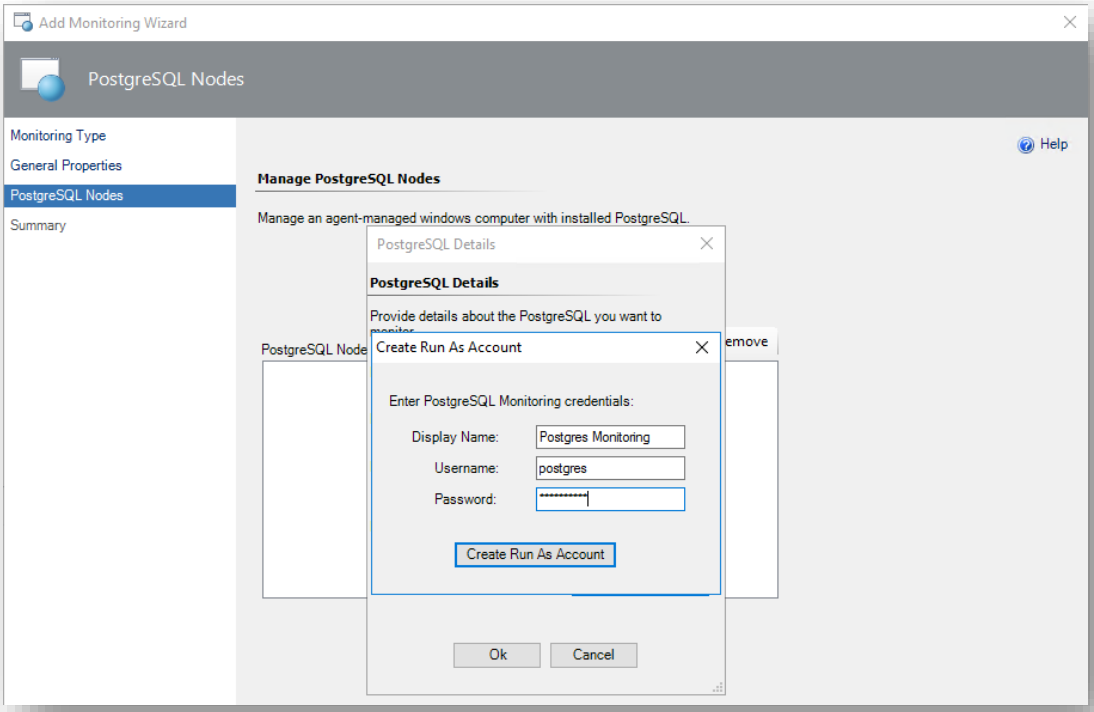
4. On the **PostgreSQL Nodes** page click on the **Add** button to add your **PostgreSQL Servers** that you want to monitor. A new window will open (**PostgreSQL Details**) where PostgreSQL server details have to be entered.

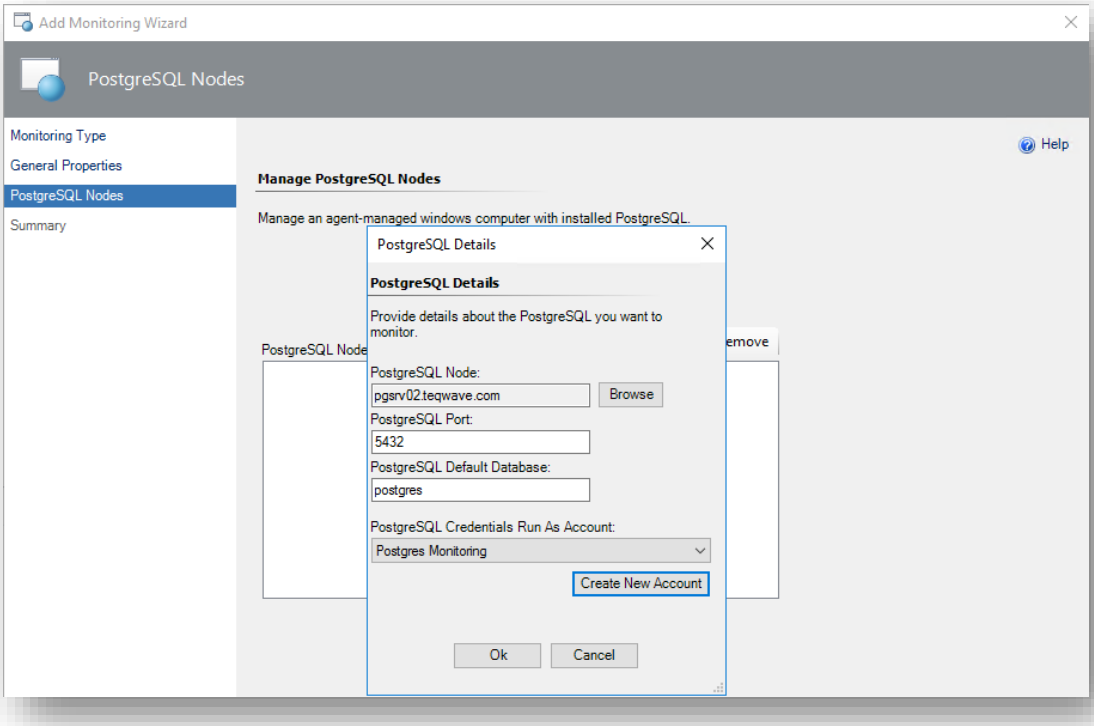


5. In **PostgreSQL Details** window click **Browse** and select PostgreSQL server node. Add **PostgreSQL Port** number and **Default Database**.

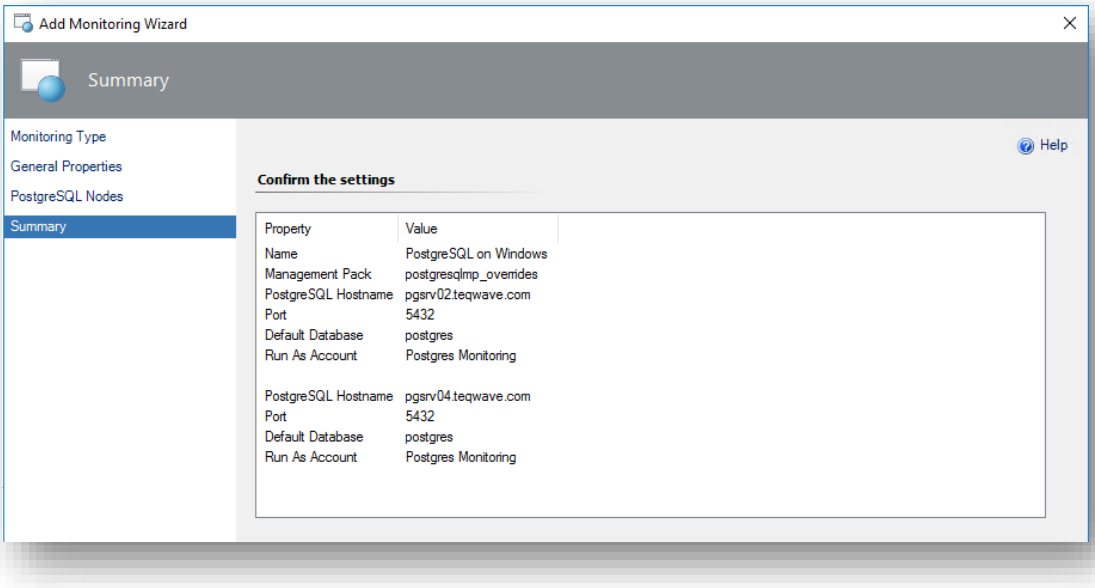


6. If the Run As account that will be used for PostgreSQL monitoring is not created yet, please click on **Create Run As Account** button and create one.





7. After adding all PostgreSQL server nodes that you want to monitor, check the configuration and create a monitoring template.



STEP 5 - LICENSE POSTGRESQL MP MANAGEMENT PACK

Teqwave PostgreSQL MP is licensed on a per PostgreSQL server instance basis. Every PostgreSQL server instance requires a valid license in order to be monitored. All databases running on the server instance will be automatically monitored.

Before using Teqwave PostgreSQL MP, a valid license must be obtained. There are two license types:

- Evaluation licenses and
- Permanent licenses

Evaluation licenses are time-limited and are used for product evaluations. Permanent licenses are issued after product purchase and have no expiration date.

Note: If you have upgraded the MP from the previous version, we recommend to restart health service on all SCOM Management Servers that are used for PostgreSQL monitoring to be sure that the new licensing module is loaded.

OBTAIN THE LICENSE

PERMANENT LICENSE

If you have purchased the PostgreSQL MP then the license file should already be sent to you electronically. You can skip this step.

EVALUATION LICENSE

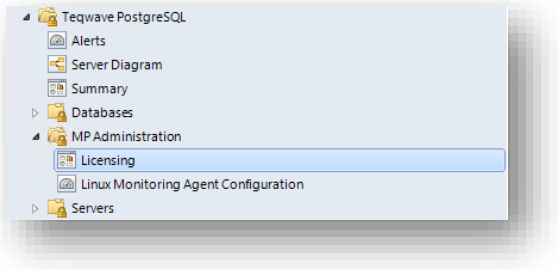
To obtain the evaluation license activation file, send an E-mail with the following details to the Teqwave licensing department at licensing@teqwave.com:

- Your company name
- Number of database instances you would like to monitor

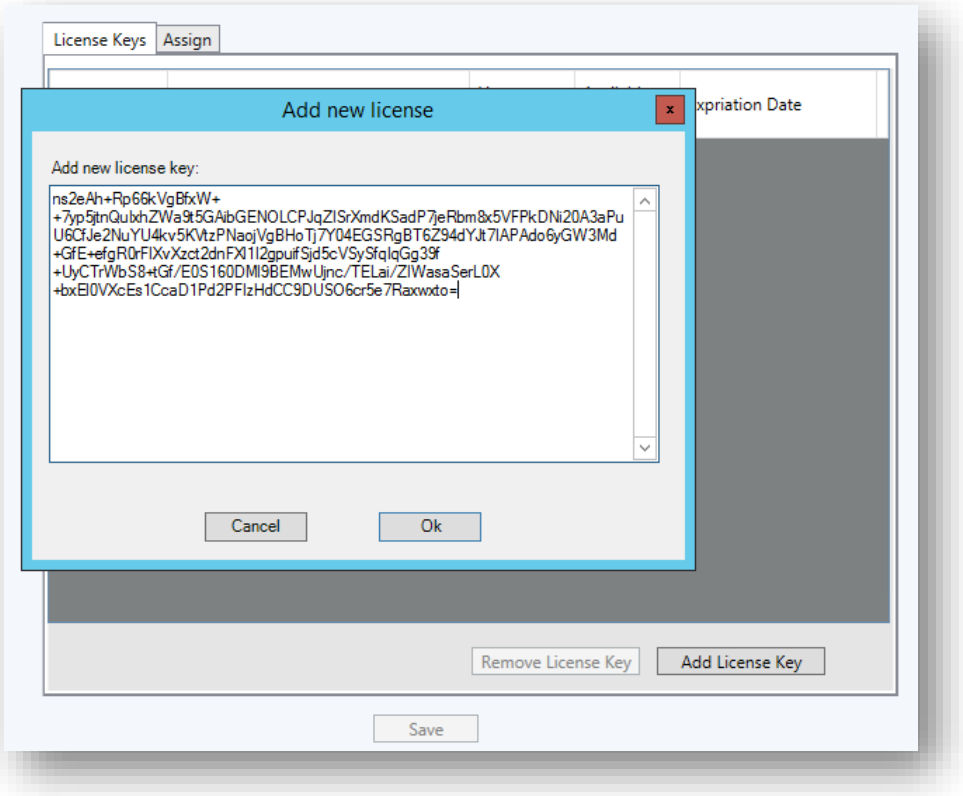
Use the same E-mail address if you have any questions about the licensing process.

APPLY THE LICENSE

To apply the license, open SCOM console on a SCOM server with admin privileges and navigate to **Teqwave PostgreSQL -> MP Administration -> Licensing** view.

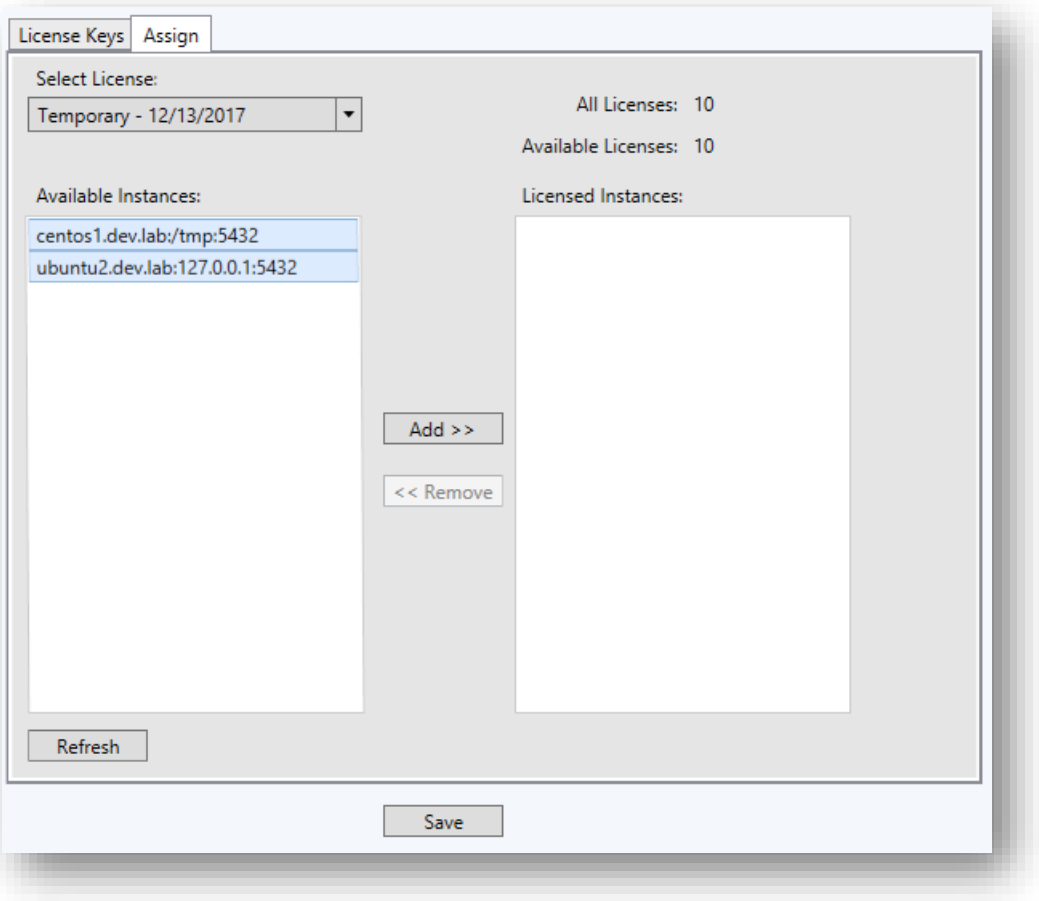


In the **License Keys** tab click on **Add License Key** button, copy the license key to the **Add new license** text box and press **Ok**.



Open **Assign** tab, select PostgreSQL server instances that you want to license from **Available Instances** table, press **Add** to move them to **Licensed Instances** table and press **Save.**

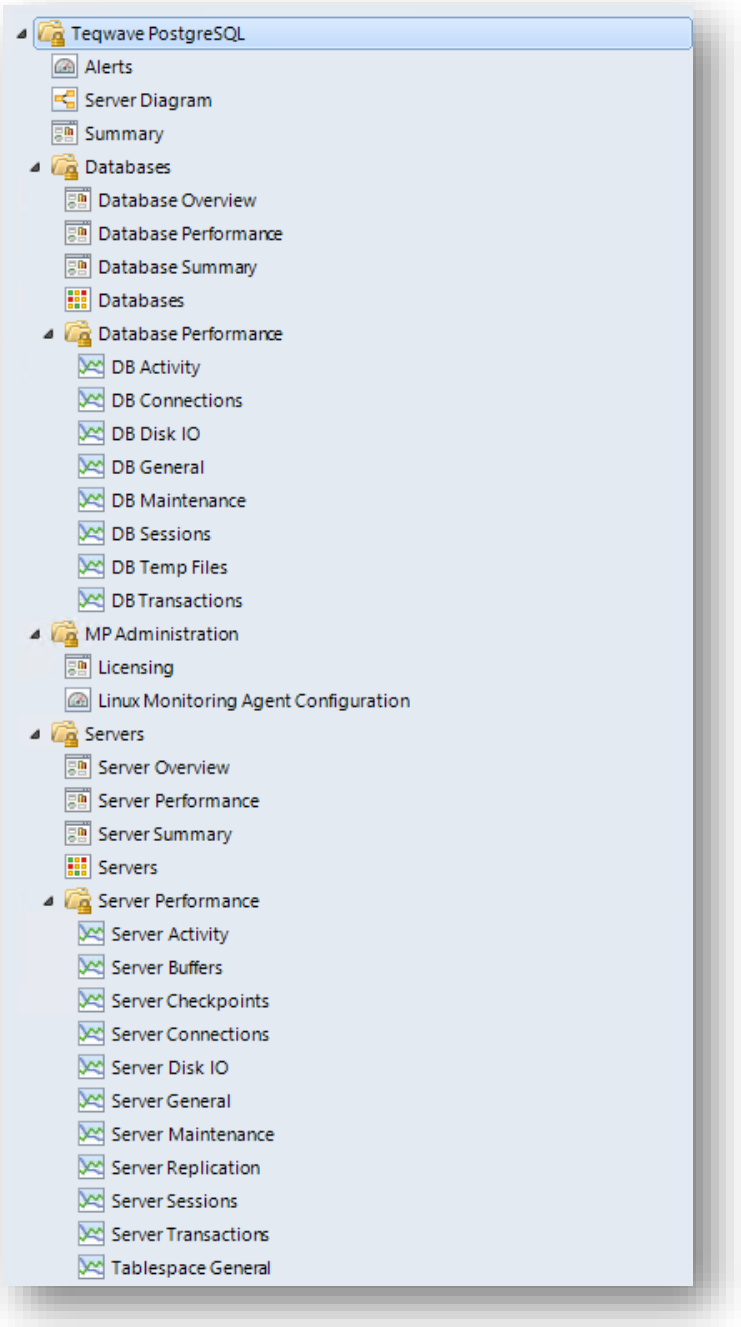
In case you have added multiple license keys, you can select from which license pool you are assigning the license by selecting the license from the top left corner (**Select License**).



Note: PostgreSQL server instances have to be discovered before applying the license. Please configure the MP and wait a couple of minutes for the initial discovery to finish and then apply the license.

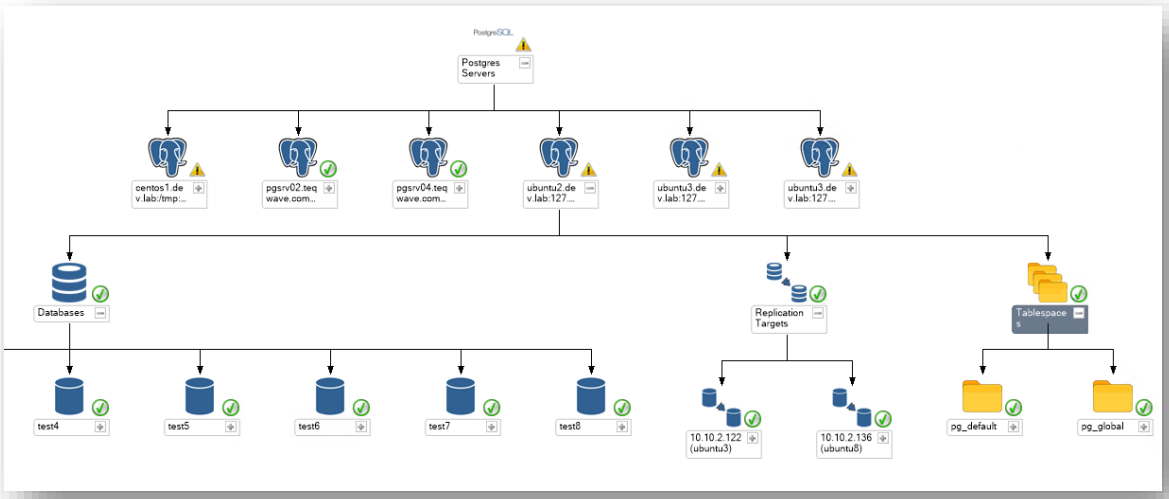
MONITORING WITH TEQWAVE POSTGRESQL MP

The Teqwave PostgreSQL MP includes a comprehensive set of views available under the Teqwave PostgreSQL folder in the Ops Mgr console Monitoring tree. The folder includes views for alerts, performance, state, and diagrams. Subfolders allow drill-down into filtered views for PostgreSQL servers and databases.



TOPOLOGY DIAGRAM VIEW

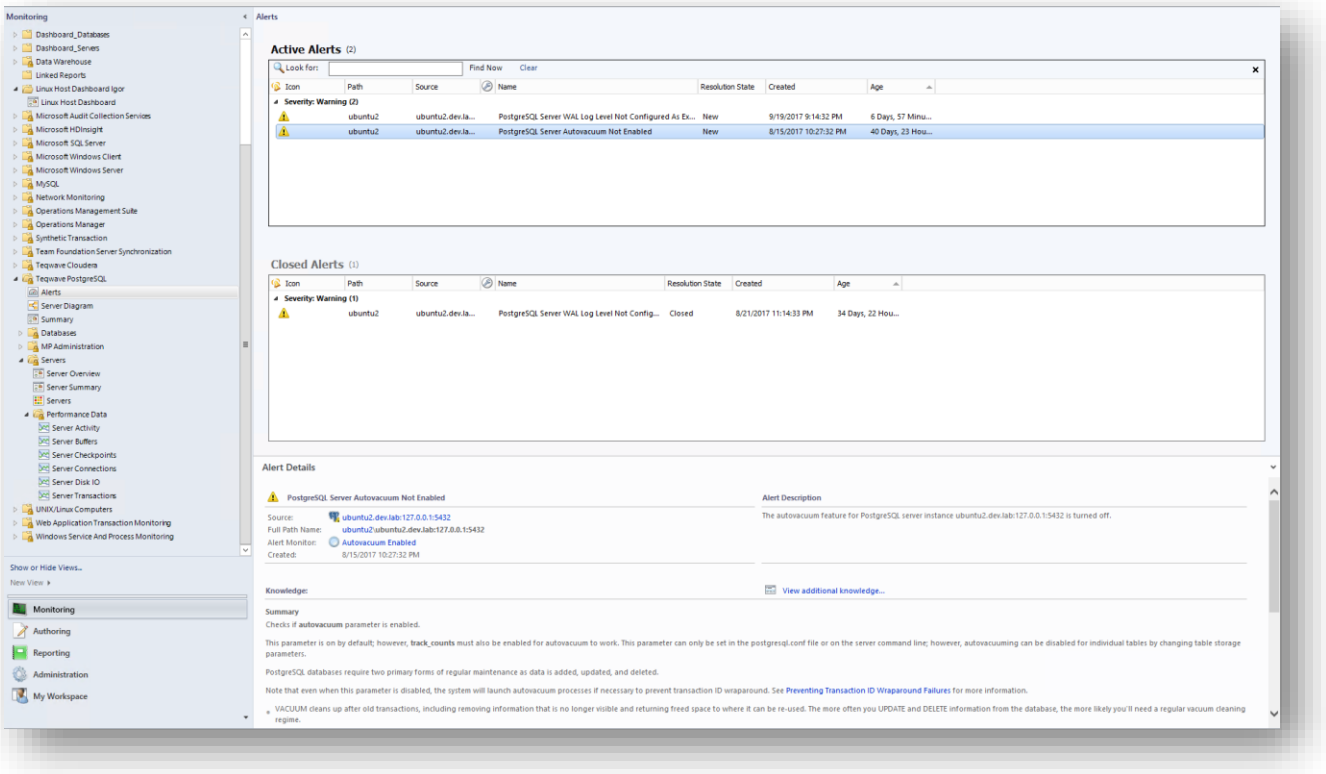
PostgreSQL MP topology shows PostgreSQL components including servers, instances, databases, and tablespaces.



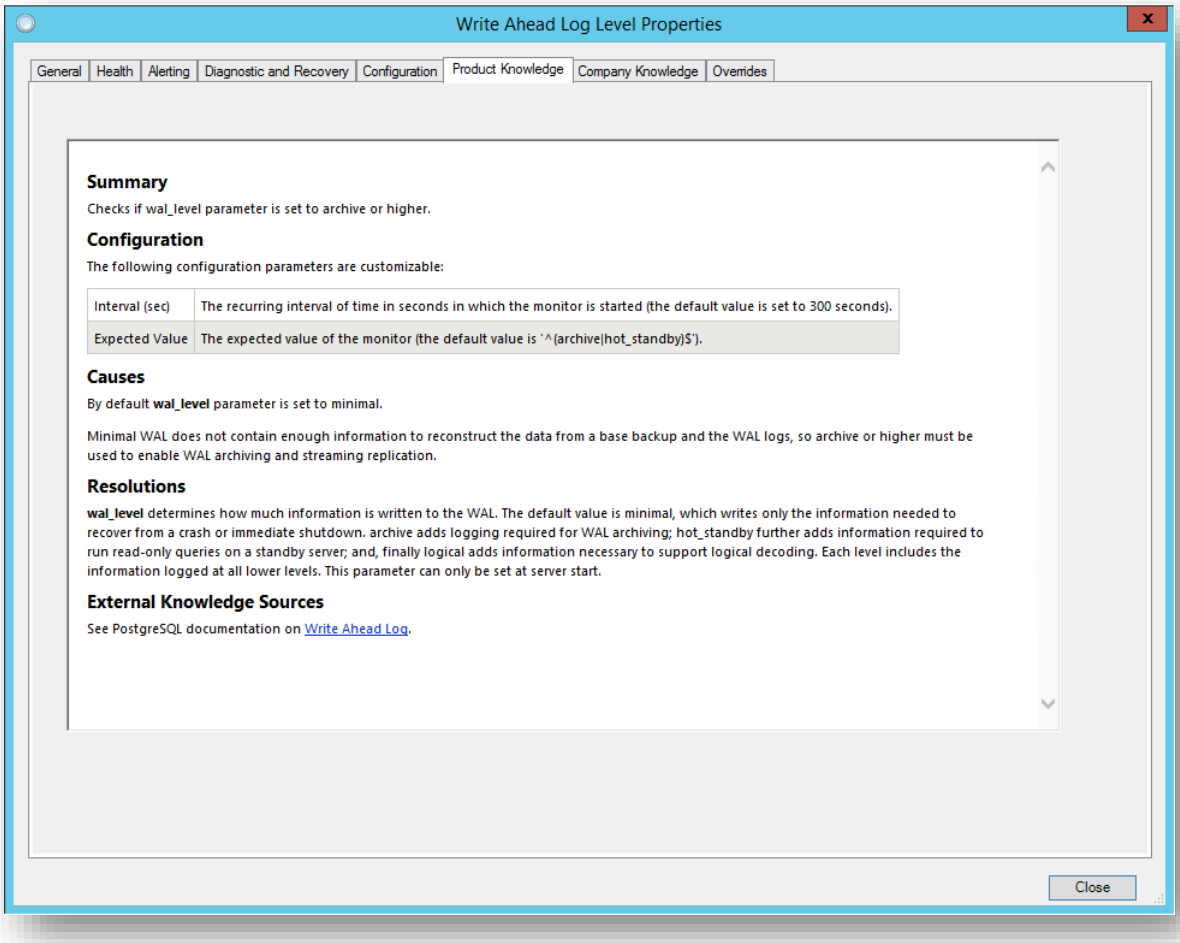
ALERTS AND KNOWLEDGE BASE

The Alert views reveal current issues in your PostgreSQL database.

Teqwave PostgreSQL MP includes **Alerts** view under the Teqwave PostgreSQL folder, showing all alerts generated by the Teqwave MP for PostgreSQL.



For each alert, you can view a knowledge base article that provides detailed information about the issue, possible cause description, resolution steps and links to external resources.



DASHBOARDS

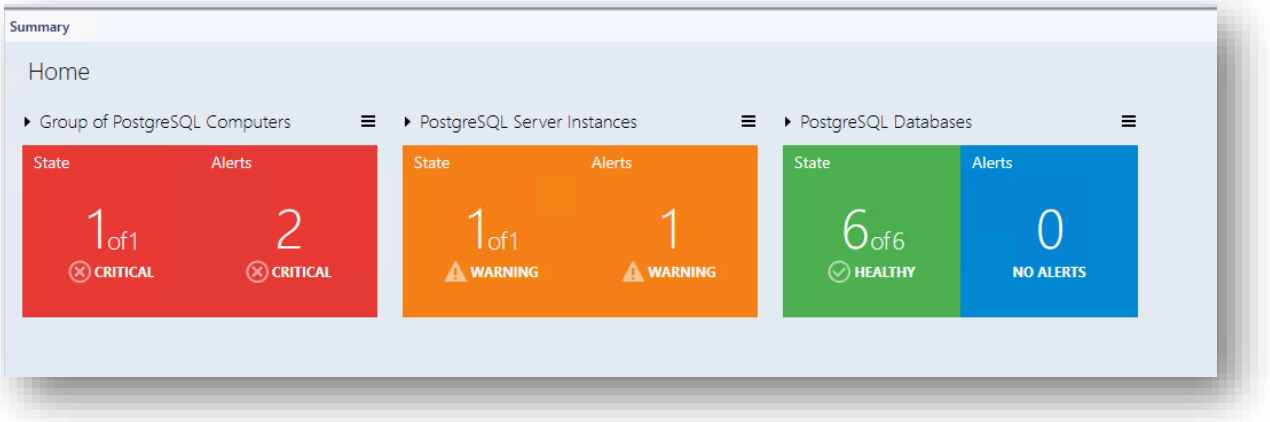
Dashboards help you quickly drill down into the root cause of a problem and speed up the troubleshooting process. These dashboards allow you to analyze the metric history for a specific performance area.

SUMMARY DASHBOARD

The summary dashboard shows a health status overview of all PostgreSQL components – Servers, Server Instances, and Databases.

In the top view, the dashboard shows the health state in an aggregated way. It is possible to drill down from the top view to the Instance view in order to investigate the root cause of the issue. You can return to the home page from any Instance view by clicking the Home part of the navigation pane.

All group tiles on the top view are collapsed by default. A tile consists of two parts; the left part displays the number of objects within a group in the worst state and the total number of objects. The right part of the widget displays the number of alerts with the highest severity.

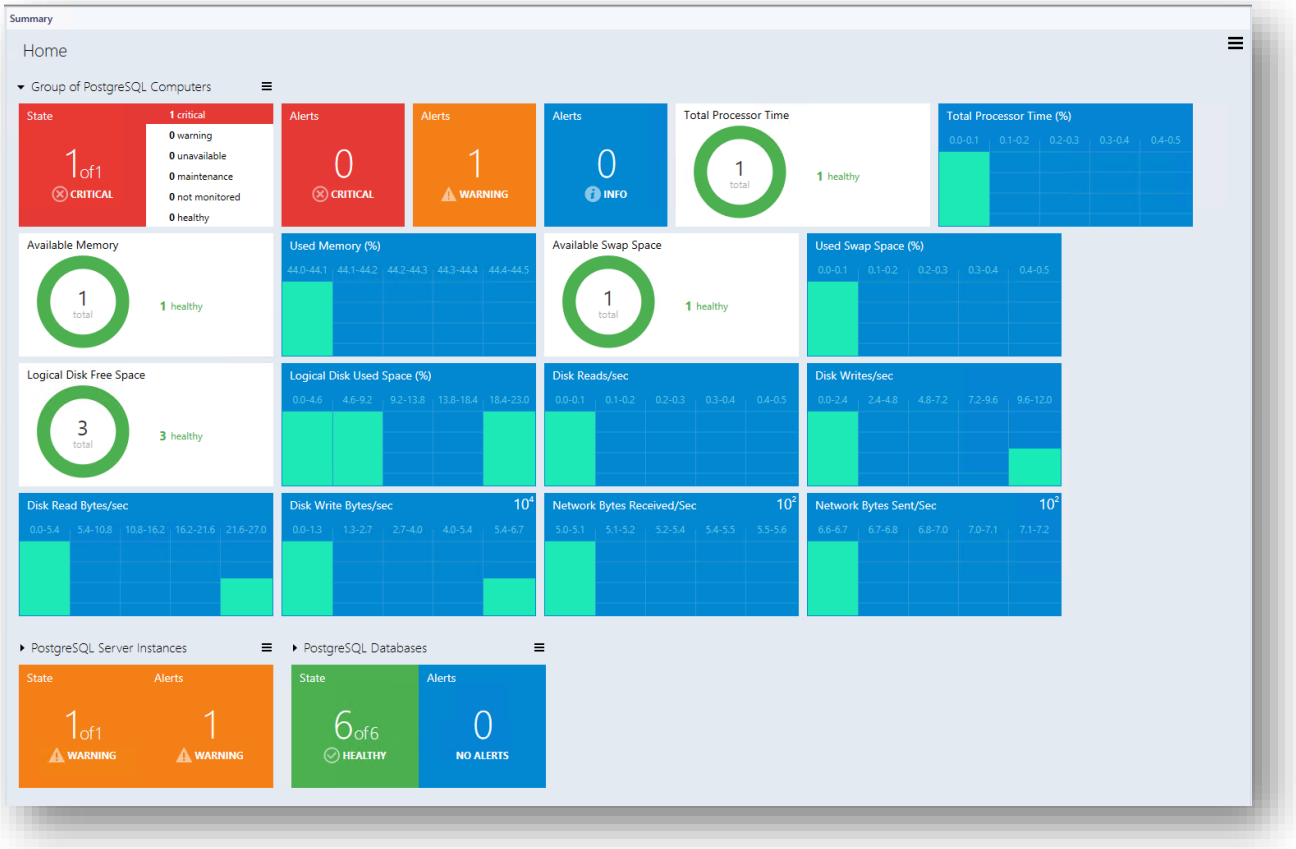


The expanded mode shows the number of objects in other states in addition to the data, which is displayed in the collapsed mode.

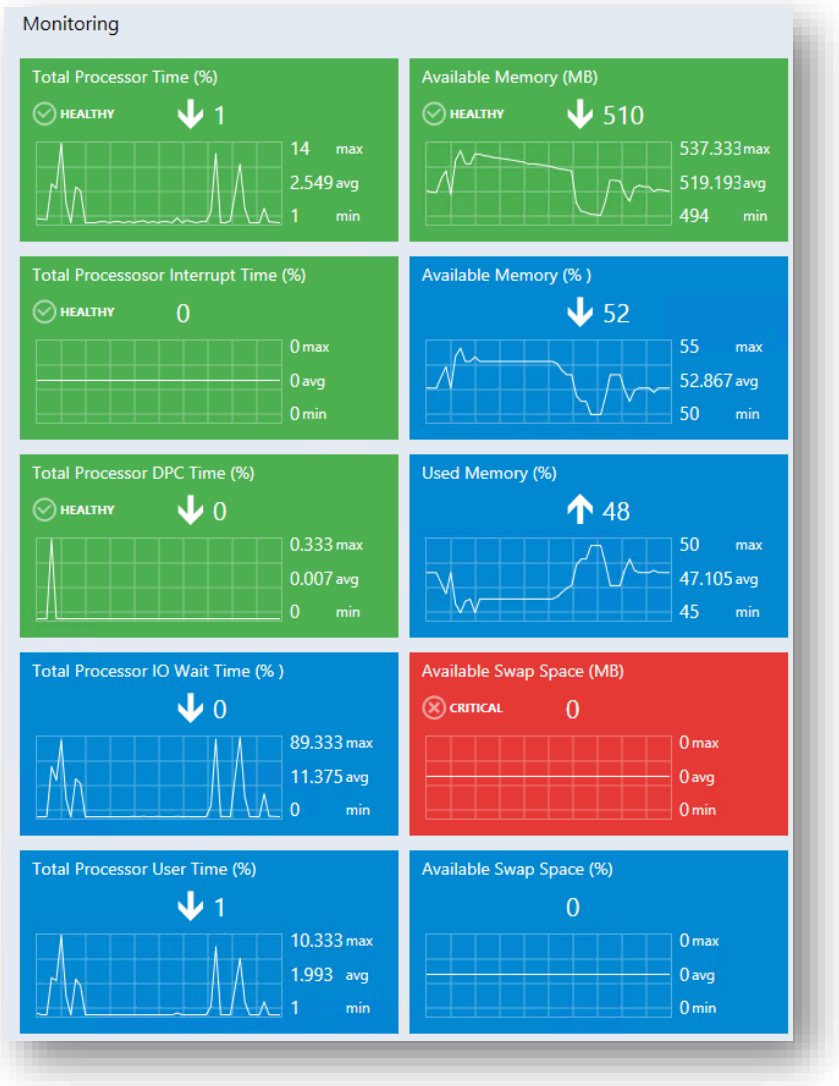
By default, the expanded mode displays three Alert widgets: Critical, Warning and Info. The number of alerts for each alert type is displayed within the corresponding widget.

The Aggregated state monitor tile provides the number of the objects' selected classes per state.

The Aggregated performance tile shows five columns each representing the number of the objects' selected classes in the current data range.



Instance view of the Dashboard opened while drilling into a group or an object from the previous Instance view or top dashboard view is provided below:

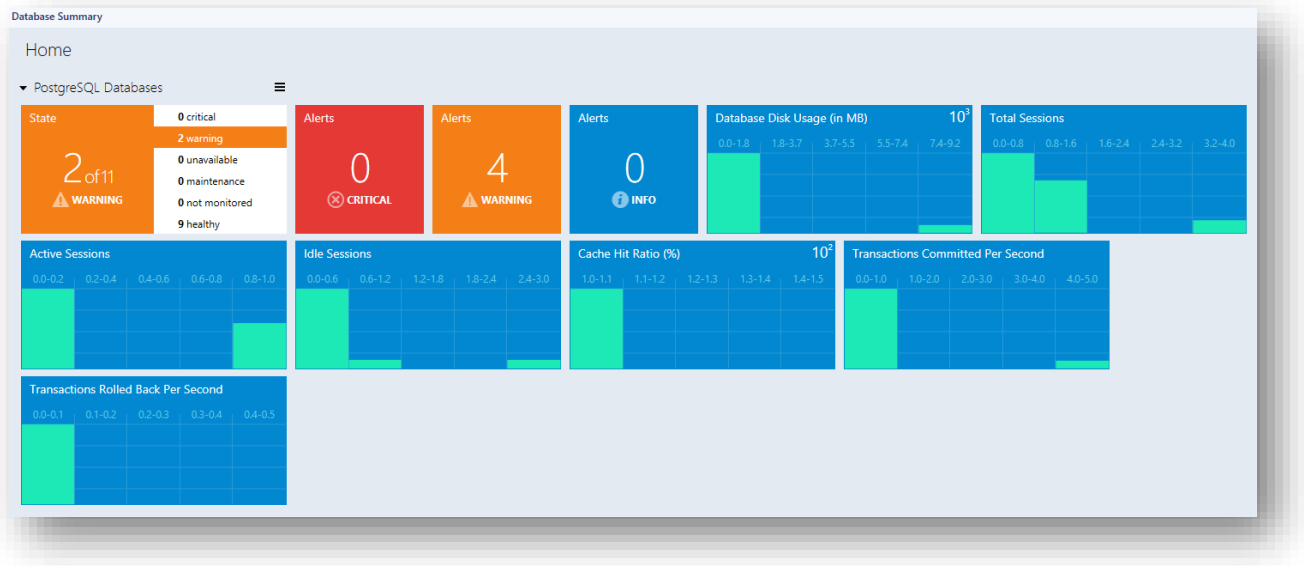


Note: Microsoft SQLServer Visualization Library MP version 6.6.0 is the prerequisite for this dashboard.

DATABASE SUMMARY

This dashboard shows a health status overview of all PostgreSQL databases.

In the top view, the dashboard shows the health state in an aggregated way. It is possible to drill down from the top view to the Instance view in order to investigate the root cause of the issue. You can return to the home page from any Instance view by clicking the Home part of the navigation pane.

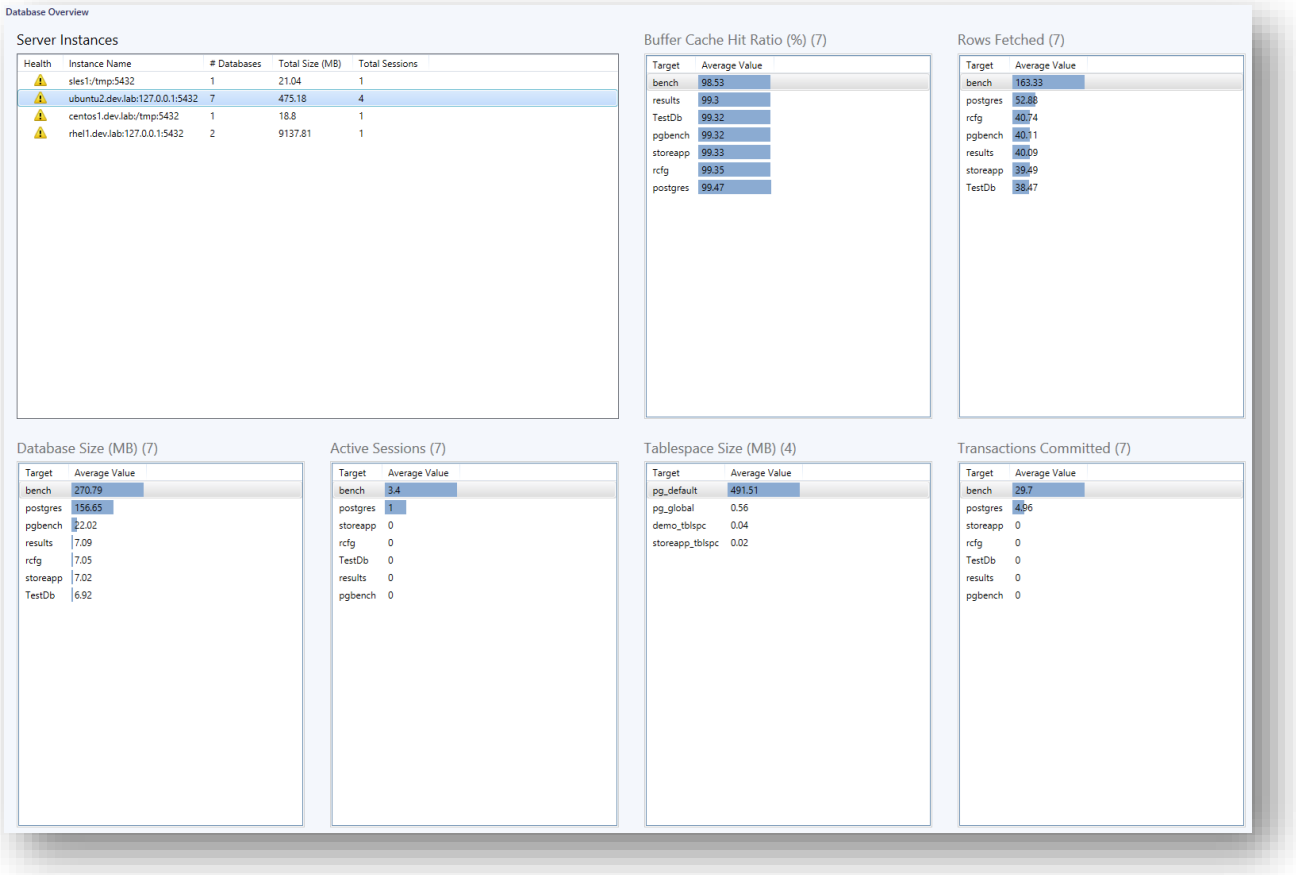


Note: Microsoft SQLServer Visualization Library MP version 6.6.0 is the prerequisite for this dashboard.

DATABASE OVERVIEW

Database Overview dashboard shows top PostgreSQL databases by:

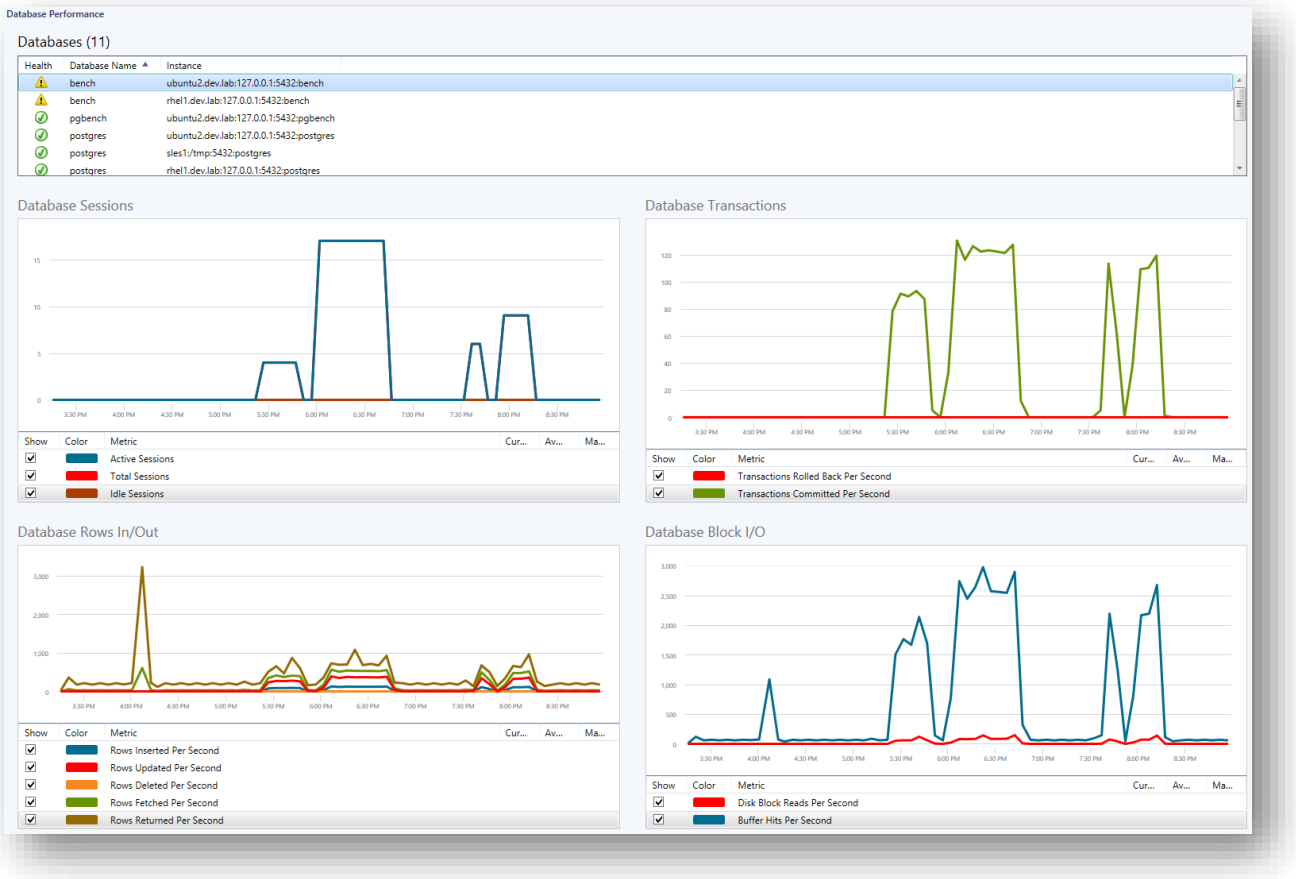
- Buffer Cache Hit Ratio (%)
- Rows Fetched
- Database Size (MB)
- Active sessions
- Tablespace size (MB)
- Transactions Committed



DATABASE PERFORMANCE

Database Summary dashboard shows important database metrics in a single view. Users can choose the database by clicking **Database Name** from the list of databases. After that user can see intuitively visualization of:

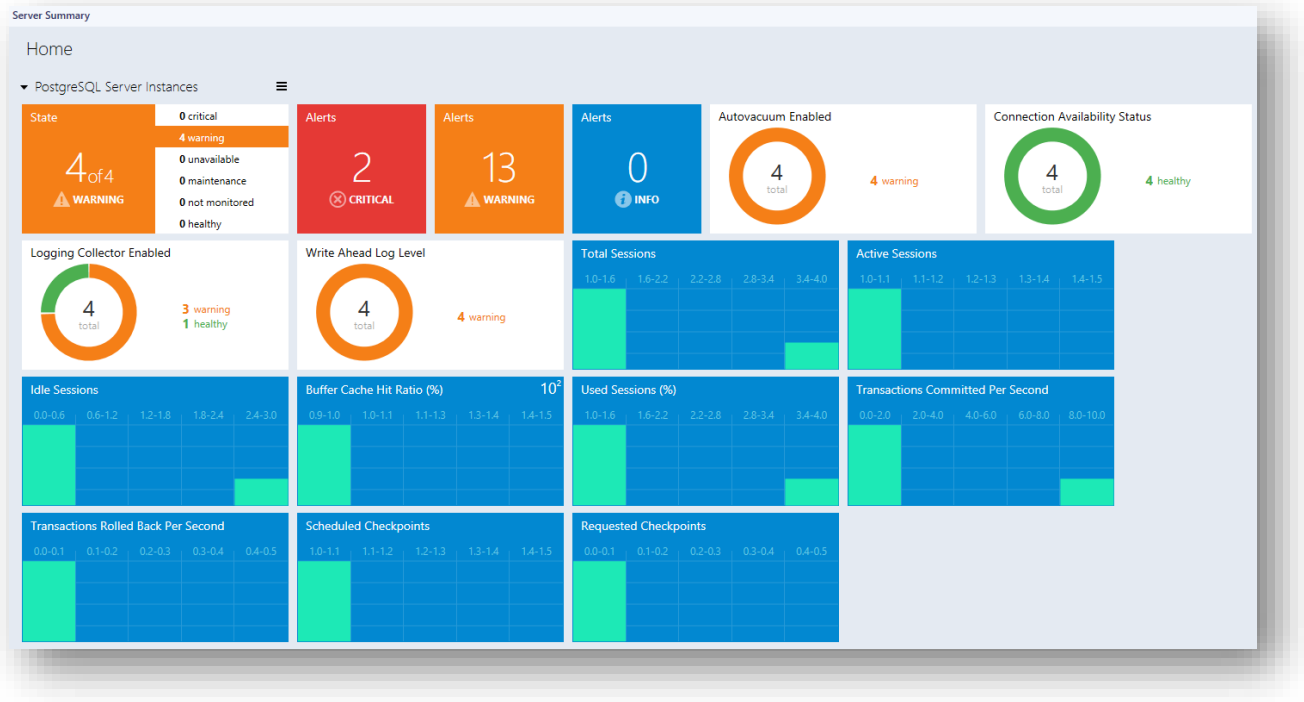
- Database Sessions
- Database Transactions
- Database Activity (Rows In/Out)
- Database I/O metrics



SERVER SUMMARY

This dashboard shows a health status overview of all PostgreSQL server instances.

In the top view, the dashboard shows the health state in an aggregated way. It is possible to drill down from the top view to the Instance view in order to investigate the root cause of the issue. You can return to the home page from any Instance view by clicking the Home part of the navigation pane.

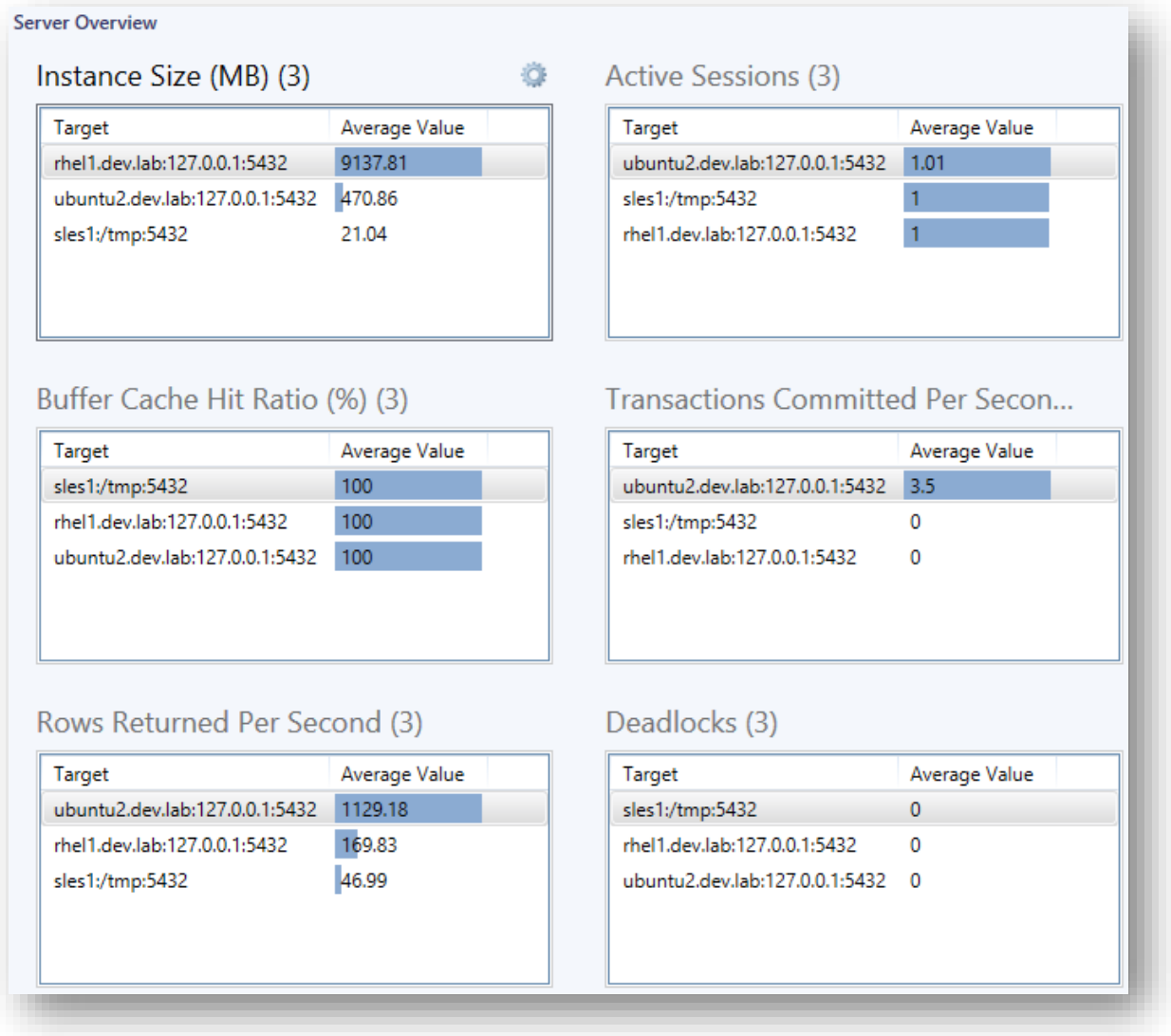


Note: Microsoft SQLServer Visualization Library MP version 6.6.0 is the prerequisite for this dashboard.

SERVER OVERVIEW

Server Overview dashboard shows top PostgreSQL server instances by:

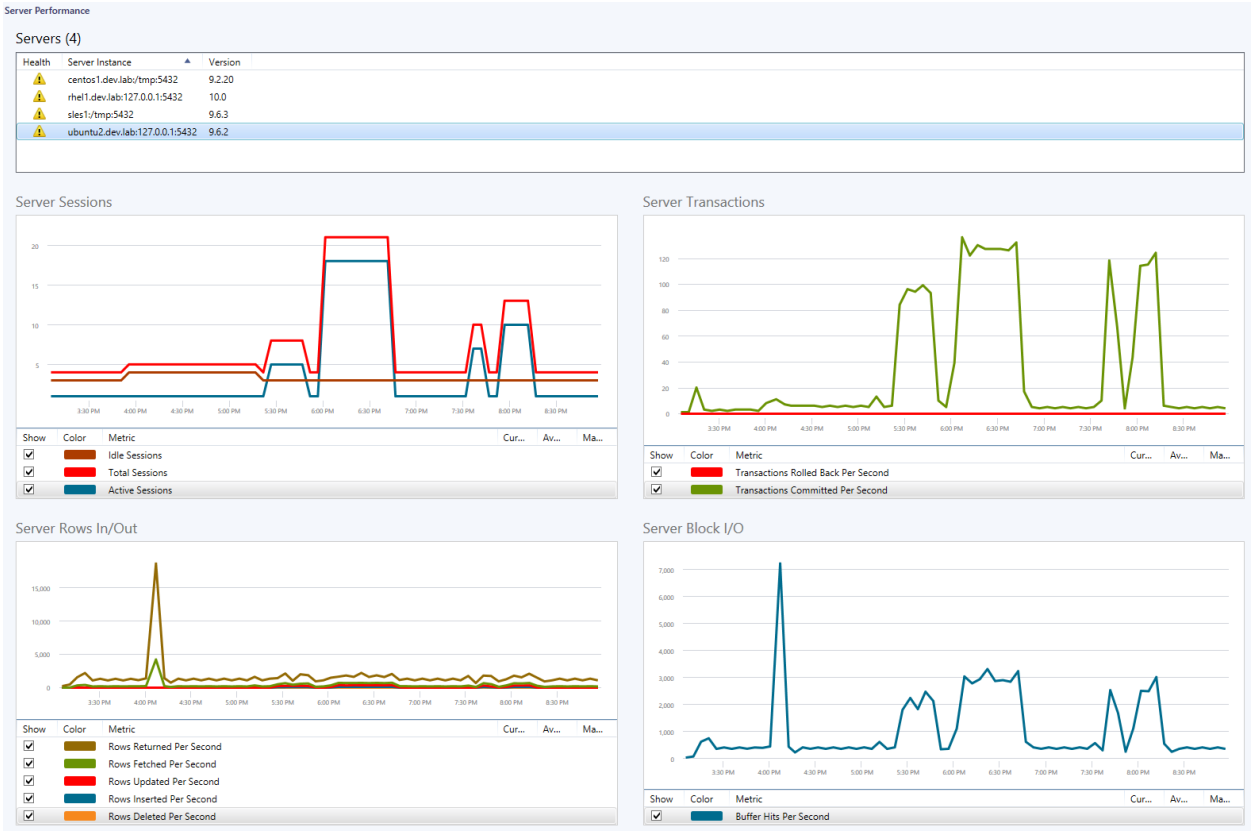
- Instance Size (MB)
- Active Sessions
- Buffer Cache Hit Ratio (%)
- Transactions Committed Per Second
- Rows Returned Per Second
- Deadlocks



SERVER PERFORMANCE

The server Performance dashboard shows important server metrics in a single view. Users can choose the server instance by clicking **Server Instance Name** from the list of servers. After that user can see intuitively visualization of:

- Server Sessions
- Server Transactions
- Server activity (Rows In/Out)
- Server I/O metrics



PERFORMANCE VIEWS

A set of performance views is available in view subfolders, showing the performance of discovered PostgreSQL servers and databases. The following performance views are available in the Teqwave PostgreSQL MP:

DATABASE PERFORMANCE VIEWS

- **DB Activity**
  Shows the following database metrics:
  - Rows Deleted Per second
  - Rows Returned Per second
  - Rows Updated Per second
  - Rows Fetched Per second
  - Rows Inserted Per second

- **DB Connections**
  Shows the following database metrics:
  - Oldest Running SQL (s)
  - Oldest Transaction (s)
  - Oldest Idle-In-Transaction (s)
  - Number of Lock Waits
  - Oldest Lock Wait (s)
  - User Lock Wait Count
  - Superuser Lock Wait Count
  - Oldest User Transaction (s)
  - Oldest Superuser Transaction (s)
  - Oldest Running User SQL (s)
  - Oldest Running Superuser SQL (s)
  - Oldest User Lock Wait (s)
  - Oldest Superuser Lock Wait (s)

- **DB Disk IO**
  Shows the following database metrics:
  - Disk Blocks Read Per Second
  - Buffer Hits Per Second
  - Buffer Cache Hit Ratio (%)
  - Time Spent Reading Data (ms)
  - Time Spent Writing Data (ms)

- **DB General**
  Shows the following database metrics:
  - DB Disk Usage (in MB)
  - Total Number of Conflicts
  - Total Number of Deadlocks
  - Active Locks

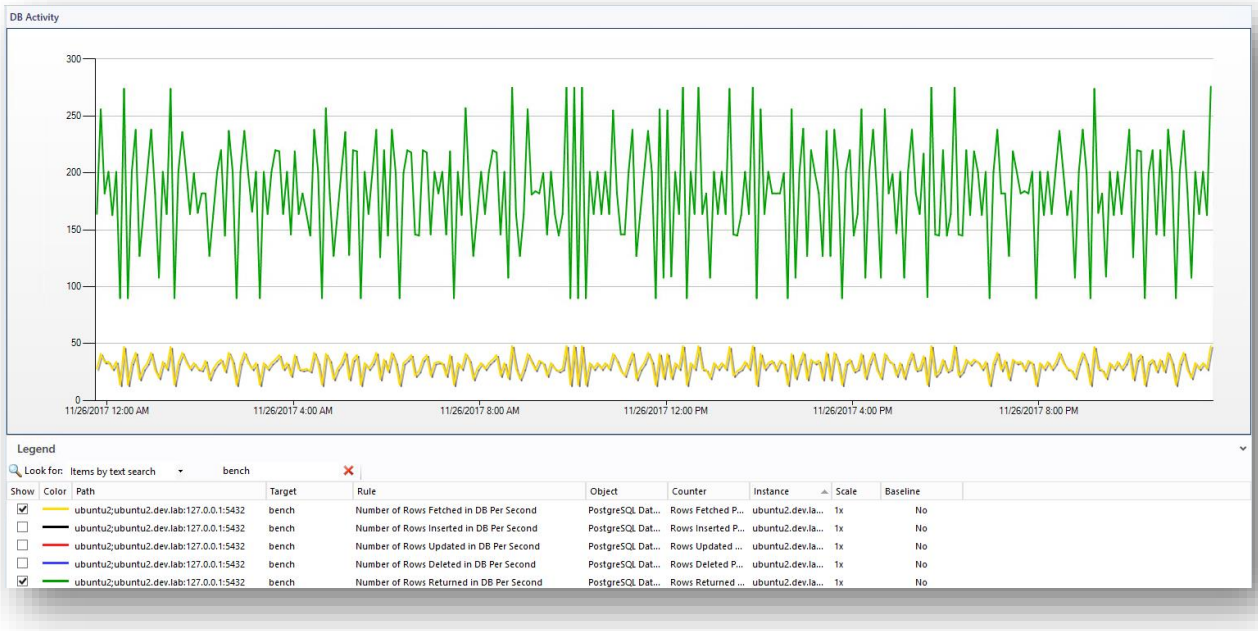
- **DB Maintenance**
  Shows the following database metrics:
  - Vacuums Active
  - Autovacuums Active
  - Analyzes Active
  - Autoanalyzes Active
  - Copies Active
  - Reindexes Active
  - Clusterings Active
  - Materialized View Refreshes Active

- **DB Sessions**
  Shows the following database metrics:
  - Total Sessions
  - Active Sessions
  - Idle Sessions
  - Idle Sessions in Transaction
  - Idle Sessions in Transaction (Aborted)
  - Sessions with Unknown State
  - Sessions Used (%)

- **DB Temp Files**
  Shows the following database metrics:
  - Temp Files Created
  - Temp Files Written (in KB)

- **DB Transactions**
  Shows the following database metrics:
  - Transactions Committed Per Second
  - Transactions Rolled Back Per Second

Example screenshot below shows the Database Activity Performance view.



SERVER PERFORMANCE VIEWS

- **Server Activity**
  Shows the following server metrics:
  - Rows Deleted Per Second
  - Rows Returned Per Second
  - Rows Updated Per Second
  - Rows Fetched Per Second
  - Rows Inserted Per Second

- **Server Buffers**
  Shows the following server metrics:
  - Buffers Allocated
  - Buffers Written Directly By Backend
  - Backend Fsync Calls
  - Buffers Written By Background Writer
  - Background Writer Stops
  - Buffers Written During Checkpoints

- **Server Checkpoints**
  Shows the following server metrics:
  - Scheduled Checkpoints
  - Requested Checkpoints
  - Checkpoint Write Time (ms)
  - Checkpoint Sync Time (ms)

- **Server Connections**
  Shows the following server metrics:
    - Oldest Running SQL (s)
    - Oldest Transaction (s)
    - Oldest Idle-In-Transaction (s)
    - Number of Lock Waits
    - Oldest Lock Wait (s)
    - User Lock Wait Count
    - Superuser Lock Wait Count
    - Oldest User Transaction (s)
    - Oldest Superuser Transaction (s)
    - Oldest Running User SQL (s)
    - Oldest Running Superuser SQL (s)
    - Oldest User Lock Wait (s)
    - Oldest Superuser Lock Wait (s)

- **Server Disk IO**
  Shows the following server metrics:
    - Blocks Read Per Second
    - Buffer Hits Per Second
    - Buffer Cache Hit Ratio (%)
    - Time Spent Reading Data (ms)
    - Time Spent Writing Data (ms)

- **Server General**
  Shows the following server metrics:
    - Instance Disk Usage (in MB)
    - Total Deadlocks

- **Server Maintenance**
  Shows the following server metrics:
    - Vacuums Active
    - Autovacuums Active
    - Analyzes Active
    - Autoanalyzes Active
    - Copies Active
    - Reindexes Active
    - Clusterings Active
    - Materialized View Refreshes Active

- **Server Replication**
  Shows the following server metrics:
    - Replication lag (in Bytes)

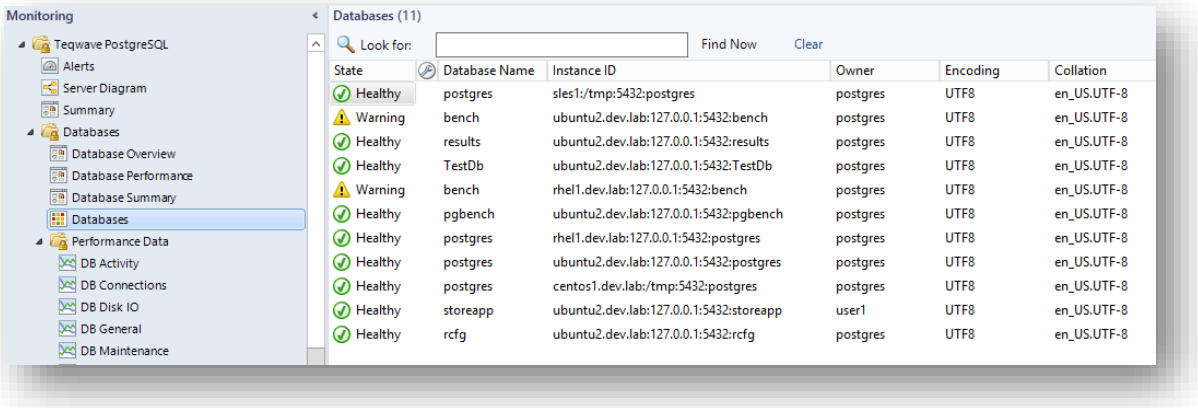
- **Server Sessions**
  Shows the following server metrics:
  - Total Sessions
  - Active Sessions
  - Idle Sessions
  - Idle Sessions in Transaction
  - Idle Sessions in Transaction (Aborted)
  - Sessions with Unknown State
  - Used Sessions (%)

- **Server Transactions**
  Shows the following server metrics:
  - Transactions Committed Per Second
  - Transactions Rolled Back Per Second

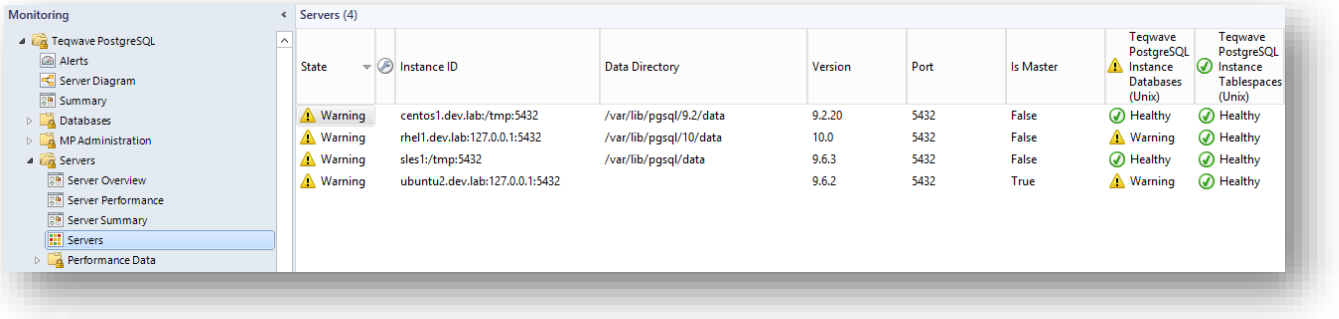
STATE VIEWS

A set of state views is available in view subfolders, showing the state and properties of discovered PostgreSQL components.

- The database state view shows the overview of all discovered PostgreSQL databases.



- The server state view shows the status of all discovered PostgreSQL server instances.



MONITORS

Teqwave PostgreSQL MP includes a set of availability and performance monitors to diagnose the state of PostgreSQL infrastructure components.

MONITORING SCENARIOS

Monitoring scenario	Description	Associated rules and monitors	Alert
Connection availability and sessions used	This scenario checks if PostgreSQL instance is ready to accept connections. In addition, it monitors the number of server connections and connection usage status. Warning and error alerts will be raised if: • connection usage is close to the maximum number of connections • connectivity is not possible.	Connection Availability Status Total Number of Sessions Number of Active Sessions Number of Idle Sessions Number of Idle Sessions in Transaction Number of Sessions with Unknown State Total Connections Used (in %)	True False False False False False True
Disk space usage	This scenario monitors the amount of used disk space by PostgreSQL databases, tablespaces and server instances. A warning alert will be raised if the used space is higher than normal.	Database Disk Usage (in GB) Tablespace Disk Usage (in GB) DB Disk Usage (in MB) Instance Disk Usage (in MB) Tablespace Disk Usage (in MB)	True True False False False

Temporary Files Created	This scenario monitors the number of temporary files created and the amount of data written to temporary files. An alert will be raised if a number of created temp files or amount of data written to the temp files is higher than normal.	Number of Temporary Files	True
		Check Amount of Data Written to Temporary Files (in KB)	True
		Number of Temp Files Created	False
		Temp Files Written (in KB)	False
		Number of Temp Files Created on Server Instance	False
		Temp Files Written (in KB) on Server Instance	False
Table and Index bloat	This scenario checks if there are bloated tables or indexes in the database.	Check if Bloated Tables Exist	True
		Check if Bloated Indexes Exist	True
Buffer Cache Hit Ratio	This scenario checks if the PostgreSQL database buffer cache hit ratio is lower than threshold.	Buffer Cache Hit Ratio (in %)	True
		Buffer Cache Hit Ratio (%) on Server Instance	False
Server settings	This scenario checks if specific PostgreSQL server settings are enabled or properly configured.	Autovacuum Enabled	True
		Logging Collector Enabled	True
		Write Ahead Log Level	True
Background Writer	This scenario checks how many times background writer stopped a cleaning scan because it had written too many buffers in the last monitoring interval.	Background Writer Stops	True
Replication Health & Performance	This scenario checks if replication from master server to standby is working and how far behind (measured in lag in Bytes) the replication server is.	Replication Status	True
		Replication Lag (in Bytes)	True

USING CUSTOM SQL QUERIES

Teqwave PostgreSQL MP enables you to configure monitors and rules with custom SQL queries. This feature is very helpful in case you want to collect additional performance metrics that are not included in the MP, or if you want to be notified if any of these metrics exceed a predefined threshold.

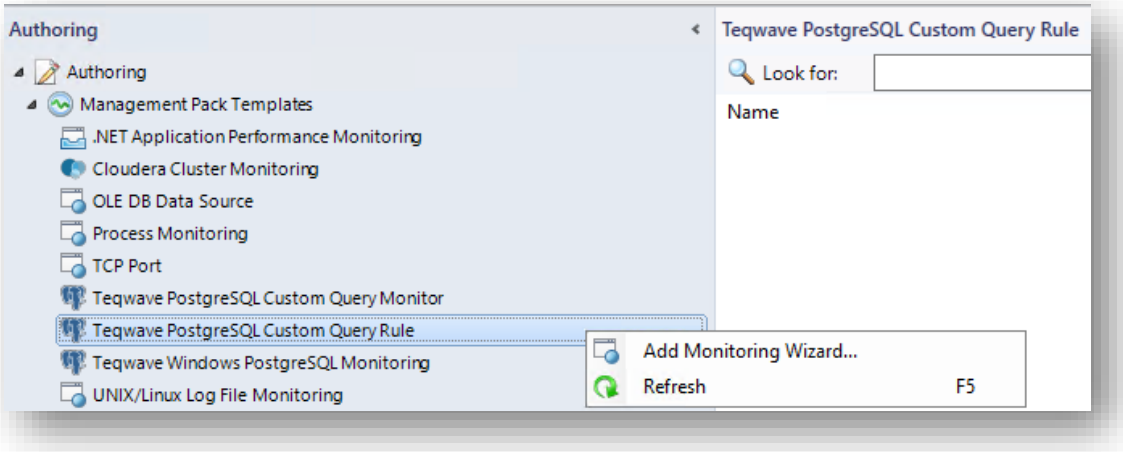
CREATING A RULE WITH CUSTOM SQL QUERY - EXAMPLE

This example shows how to create a rule that will use the below query and collect the total size of the pgbench_account table as an additional performance metric.

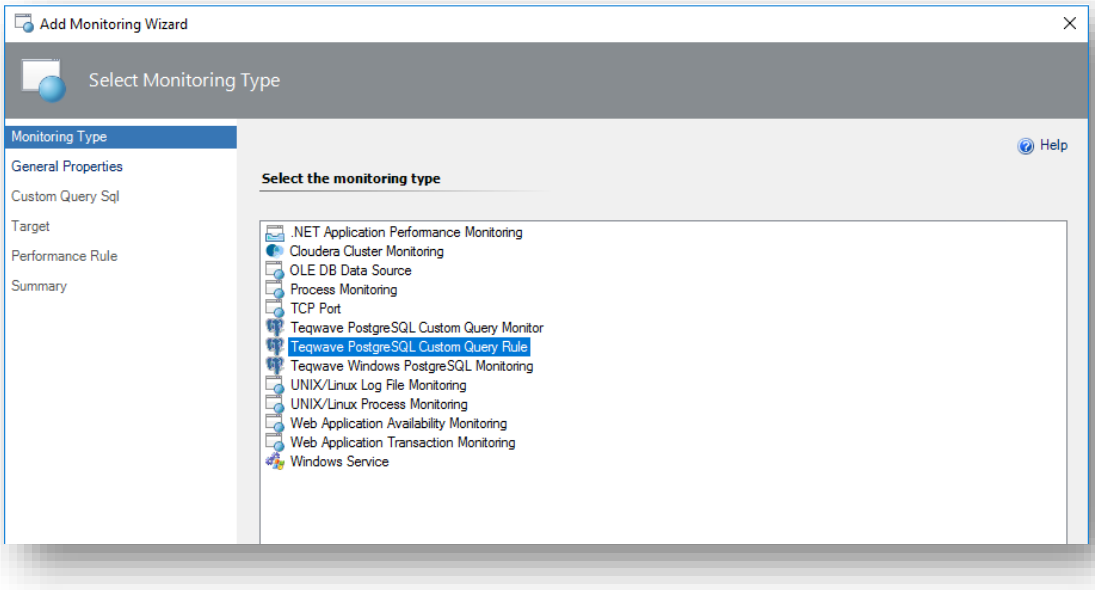
```
postgres=# SELECT relname AS relation, reltype AS type, pg_total_relation_size(C.oid) AS total_size FROM
pg_class C where relname = 'pgbench_accounts';
     relation      | type  | total_size
-------------------+-------+------------
 pgbench_accounts | 16402 |  196272128
(1 row)
```

To create a rule with custom SQL query, follow these steps

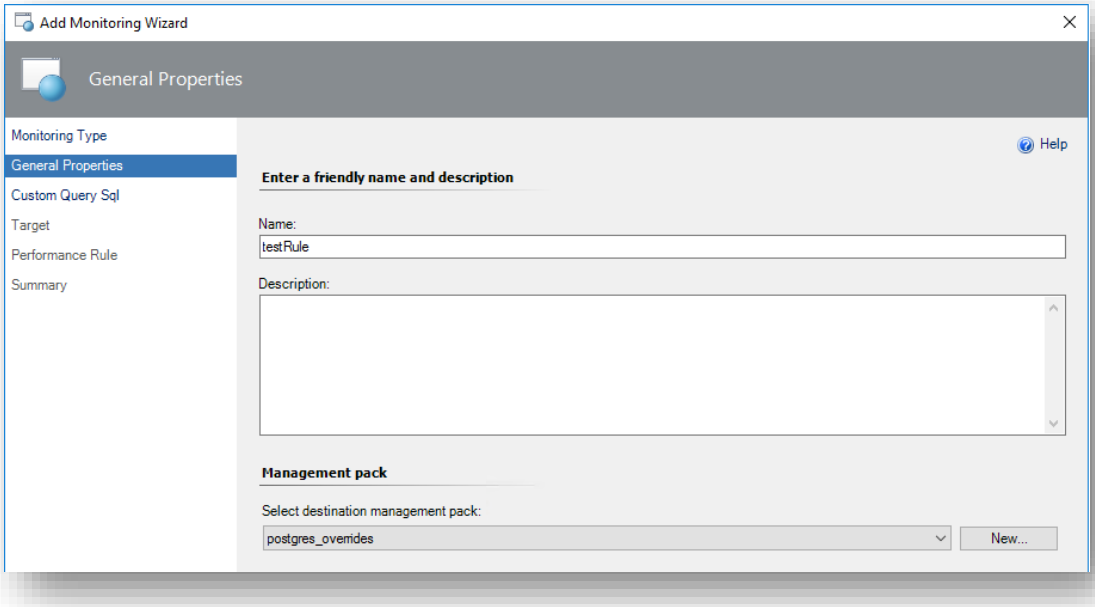
1. In SCOM Console navigate to **Authoring** | **Management Pack Templates** | **Teqwave PostgreSQL Custom Query Rule**, right-click it and select **Add Monitoring Wizard…**:



2. On the **Monitoring Type** page select **Teqwave PostgreSQL Custom Query Rule** and click **Next**:

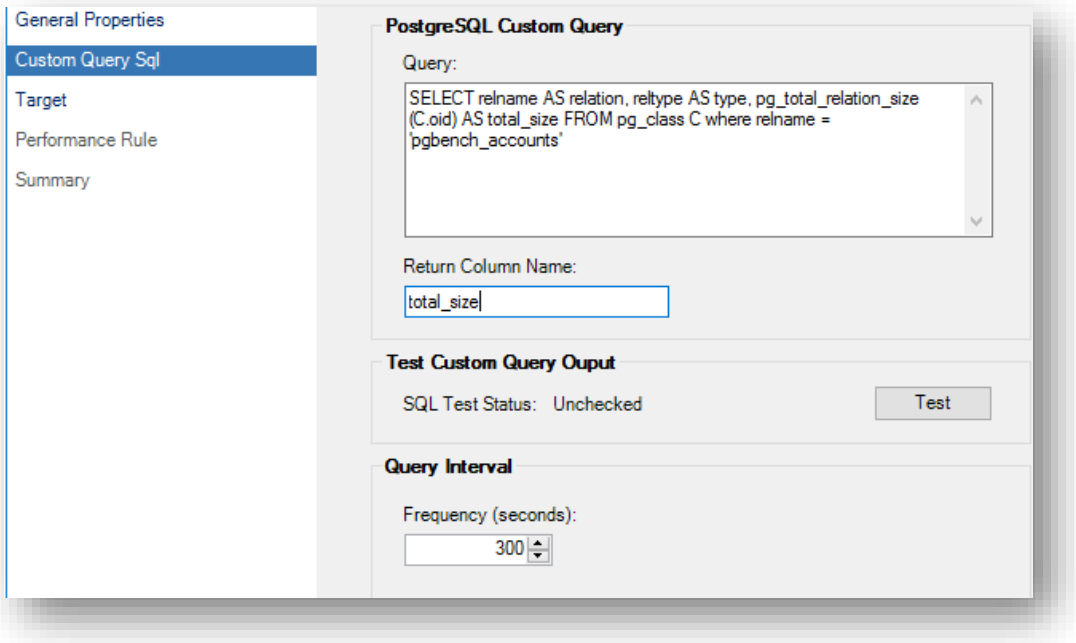


3. On the **General Properties** page provide **Name** and **Description** for your configuration template and select target **Management Pack** to store configuration to:

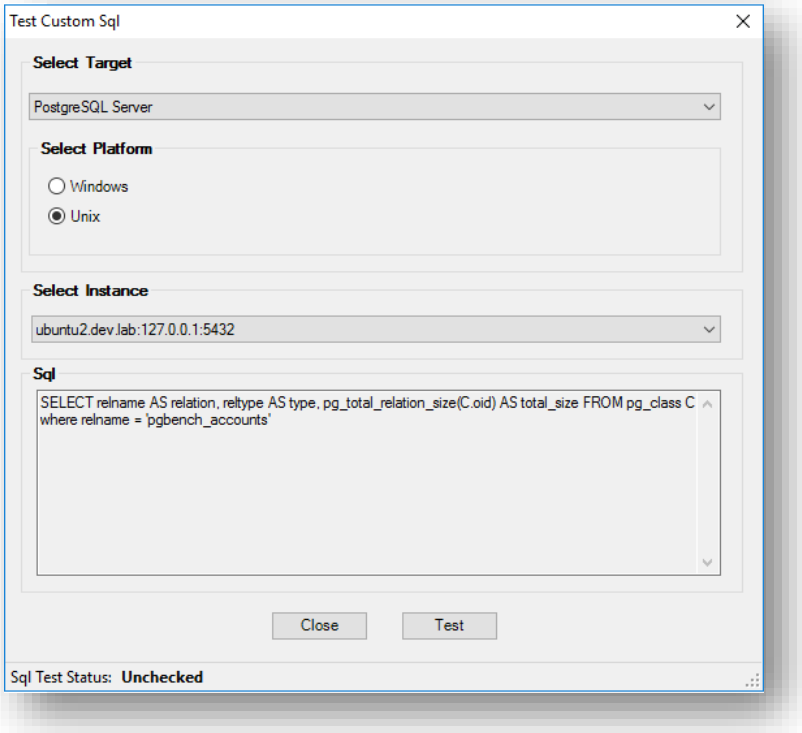


For more information about target Management Pack, please see Best Practice: Create a Management Pack for Customizations section. You can create a new management pack right from this wizard by clicking the **New** button, located next to Management Packs drop-down list.

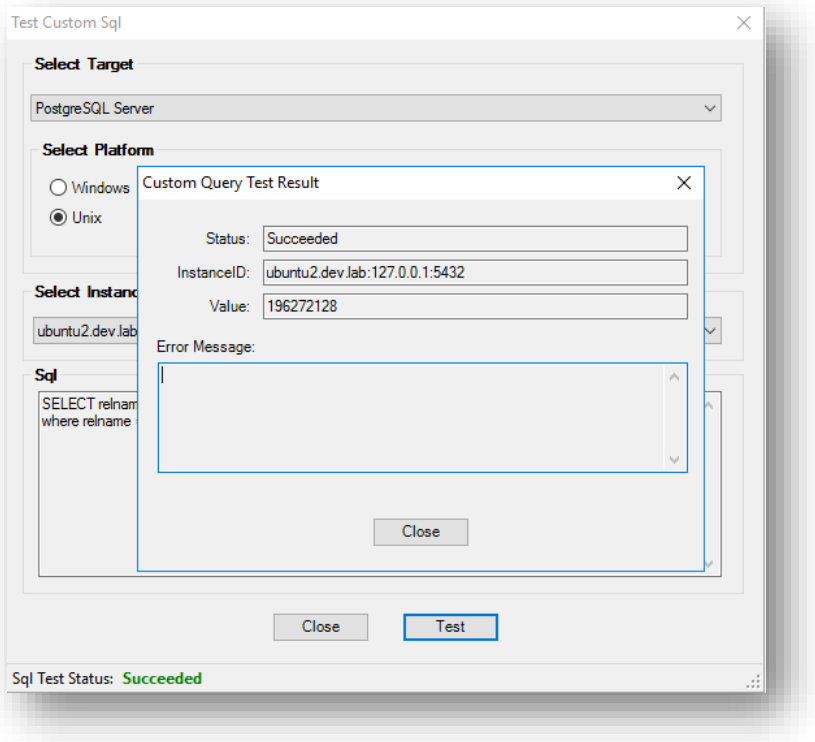
4. On the **Custom Query SQL** page enter your query in the **Query** text box, **Return Column Name** and the Rule frequency interval for the data collection. Please make sure that the specified query is valid by pressing the **Test** button.



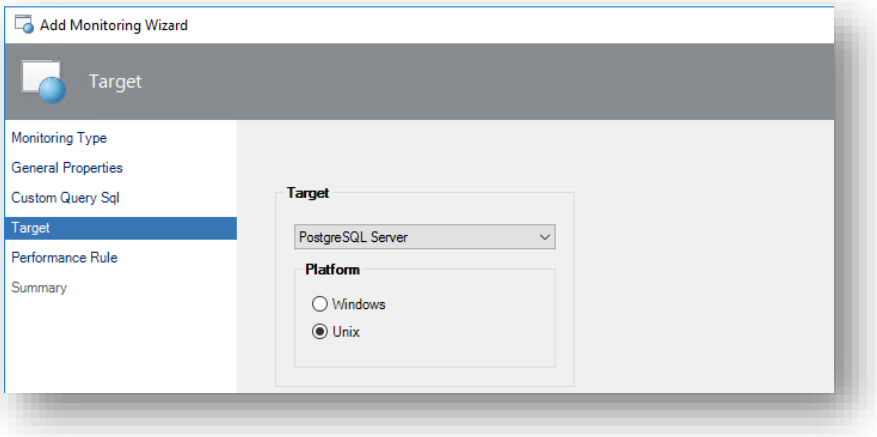
A new window will open where **Target**, **Platform**, and **Instance** have to be selected. The SQL test query will be executed using these paremeters. Press **Test** to see the results.



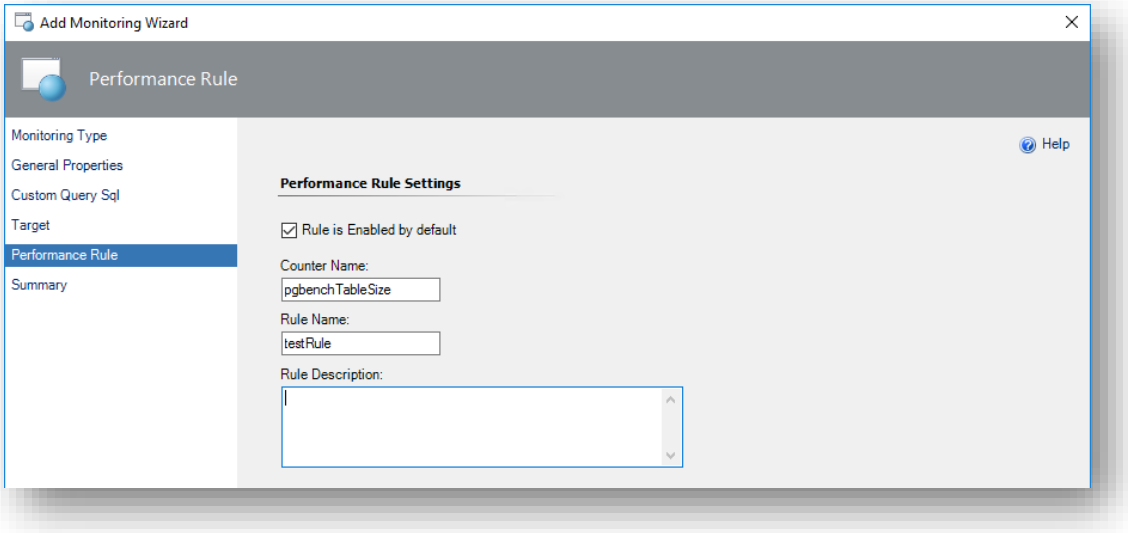
In case the test is successful, you will receive a **Succeeded** status. Please close the SQL test windows and continue with the next steps.



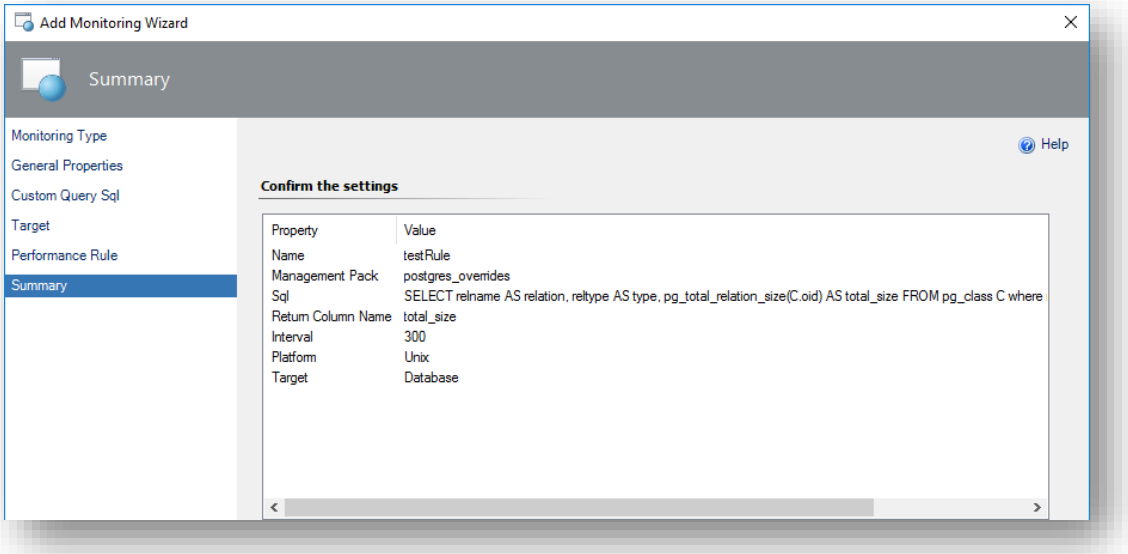
5. In the **Target** window, you have to select if you are querying server or database and select the **Platform** on which the PostgreSQL is running. Please select according to the test query in the previous step.



6. In **Performance Rule** window enter the **Counter Name**, **Rule Name,** and specify if you want to enable the rule for all instances.



7. Check the configuration in the **Summary** window and press the **Create** button to create a new Rule.



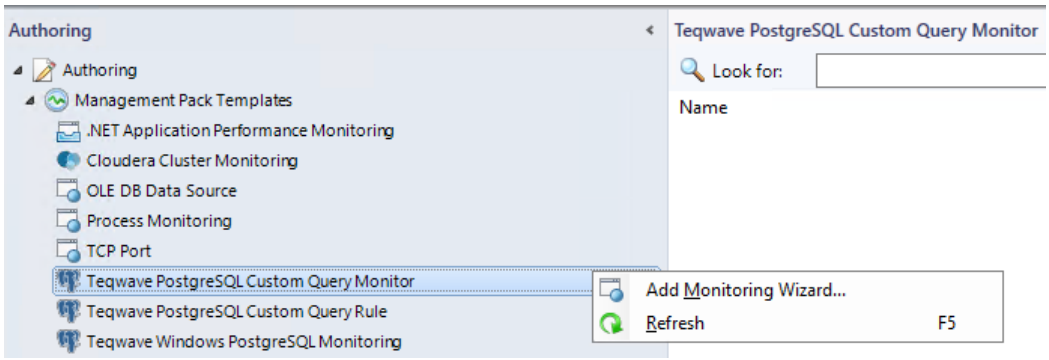
CREATING A MONITOR WITH CUSTOM SQL QUERY - EXAMPLE

This example shows how to create a monitor that will use the below query and send an Alert when the total size of the pgbench_account table exceeds the threshold.

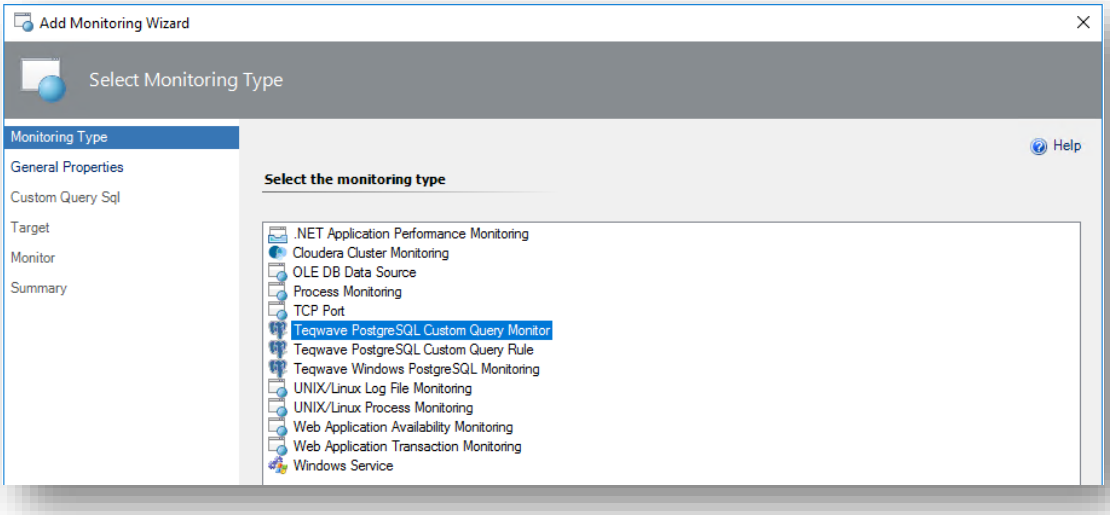
```
postgres=# SELECT relname AS relation, reltype AS type, pg_total_relation_size(C.oid) AS total_size FROM
pg_class C where relname = 'pgbench_accounts';
     relation      | type  | total_size
-------------------+-------+------------
 pgbench_accounts | 16402 |  196272128
(1 row)
```

To create the monitor with custom SQL query, follow these steps:

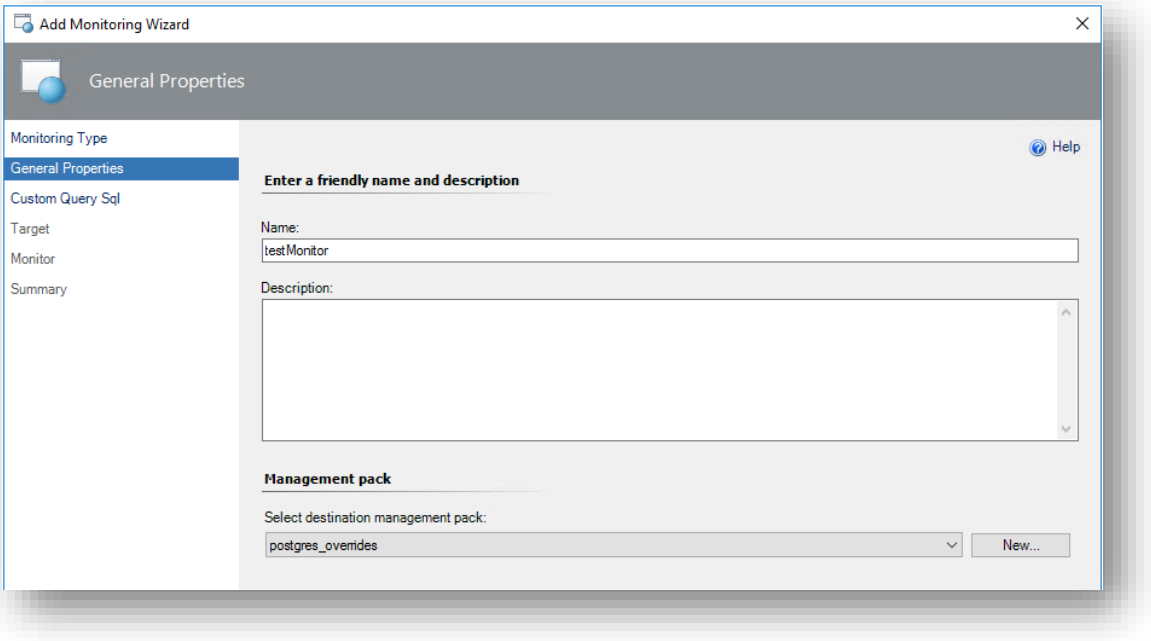
1. In SCOM Console navigate to **Authoring | Management Pack Templates | Teqwave PostgreSQL Custom Query Monitor**, right-click it and select **Add Monitoring Wizard…**:



2. On the **Monitoring Type** page select **Teqwave PostgreSQL Custom Query Monitor** and click **Next**:

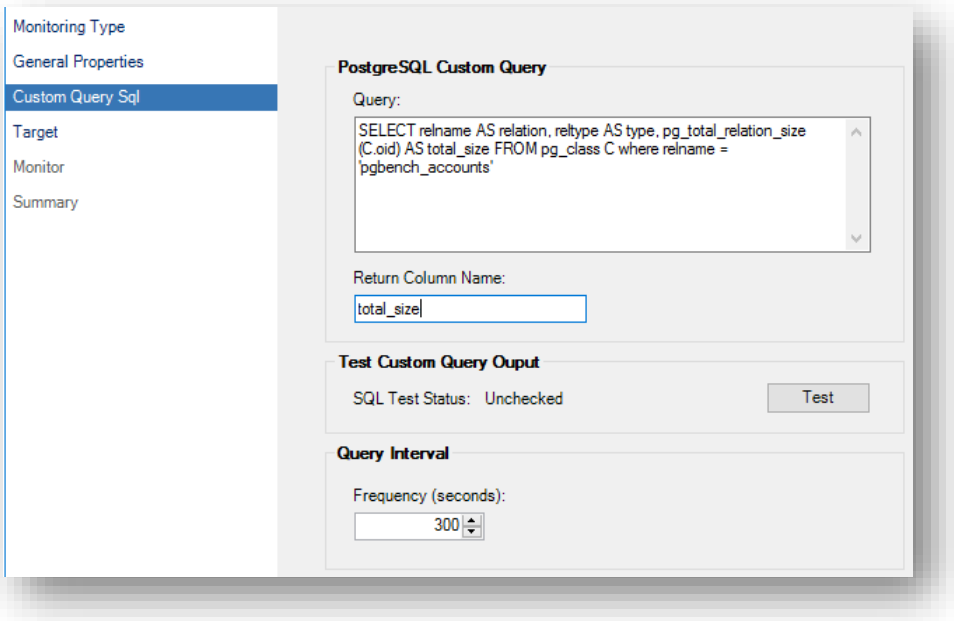


3. On the **General Properties** page provide **Name** and **Description** for your configuration template and select target **Management Pack** to store configuration to:

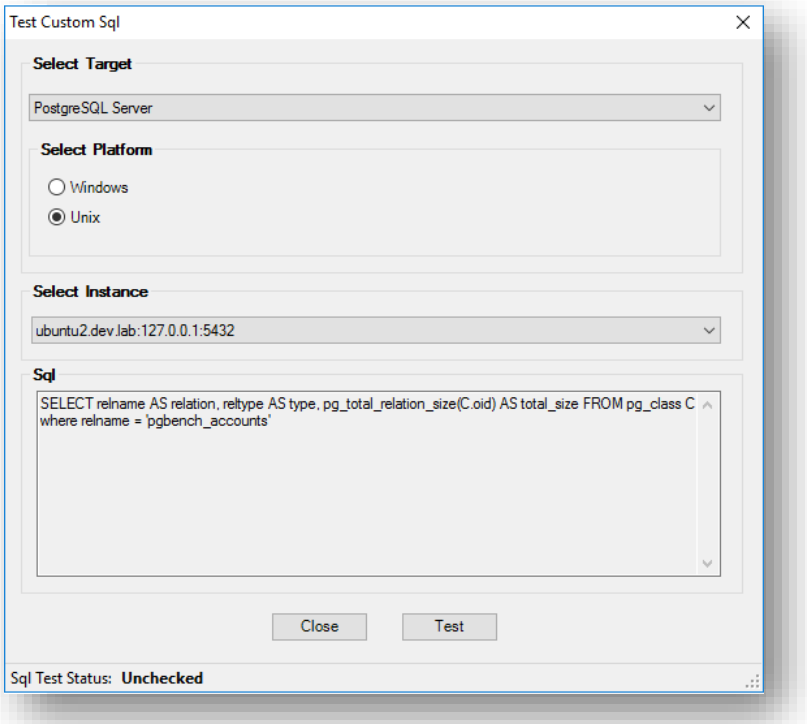


For more information about target Management Pack, please see Best Practice: Create a Management Pack for Customizations section. You can create a new management pack right from this wizard by clicking the **New** button, located next to Management Packs drop-down list.

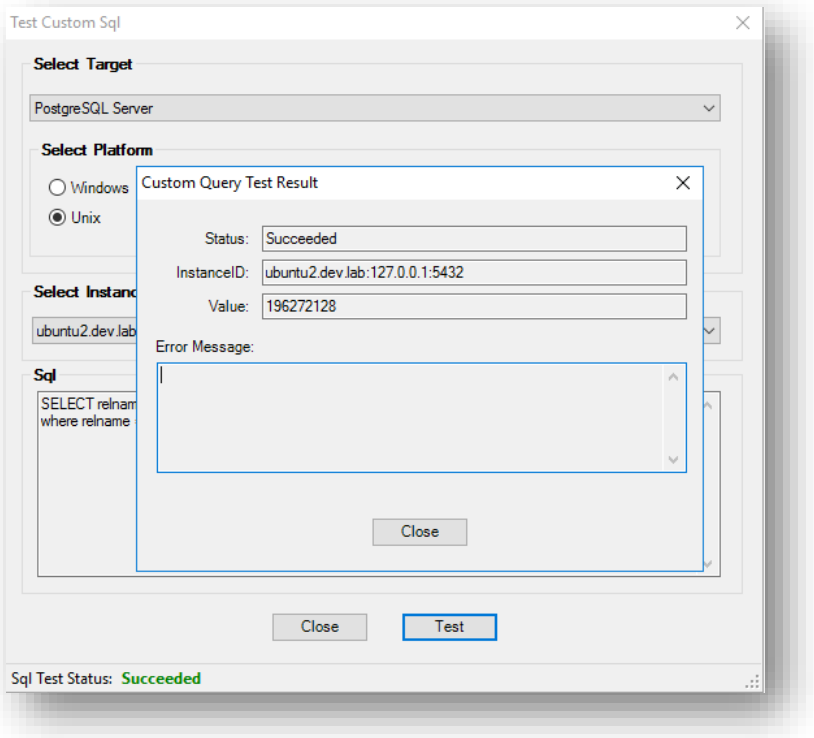
4. On the **Custom Query SQL** page enter your query in the **Query** text box, **Return Column Name** and the Monitor frequency interval. Please make sure that the specified query is valid by pressing the **Test** button.



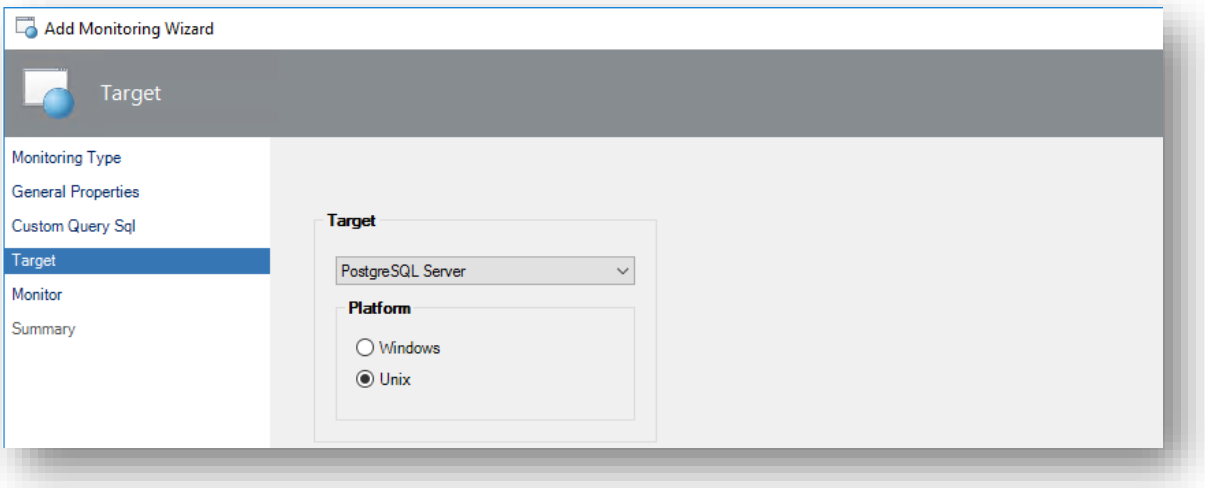
A new window will open where **Target**, **Platform**, and **Instance** have to be selected. The SQL test query will be executed using these paremeters. Press **Test** to see the results.



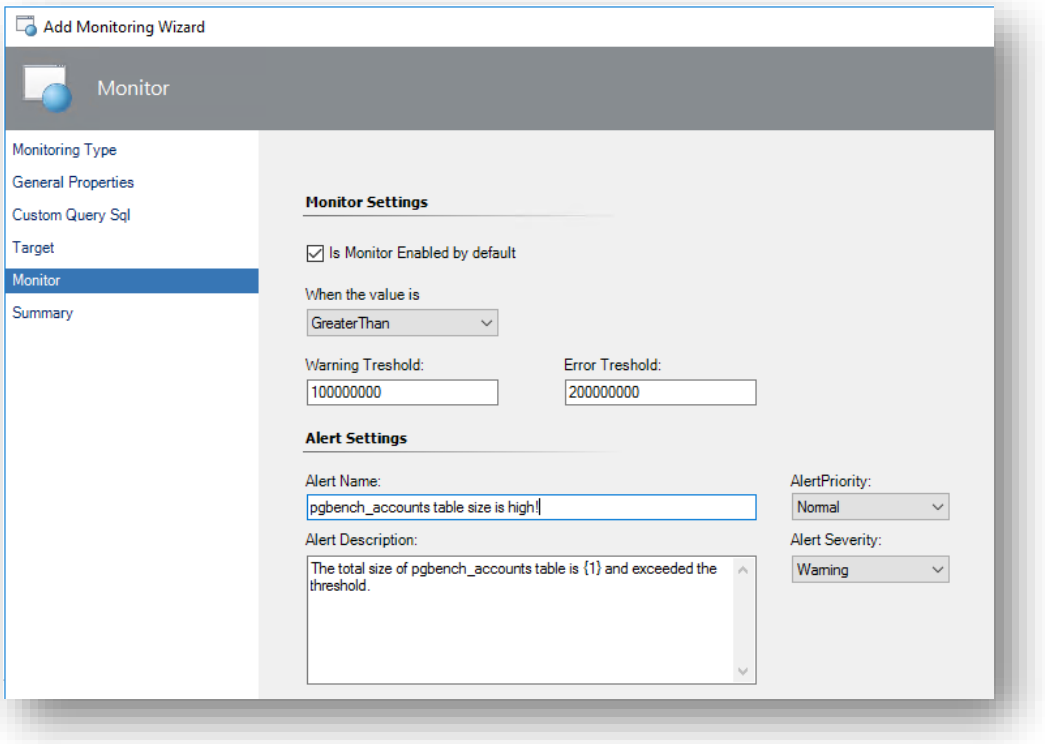
In case the test is successful, you will receive a **Succeeded** status. Please close the SQL test windows and continue with the next steps.



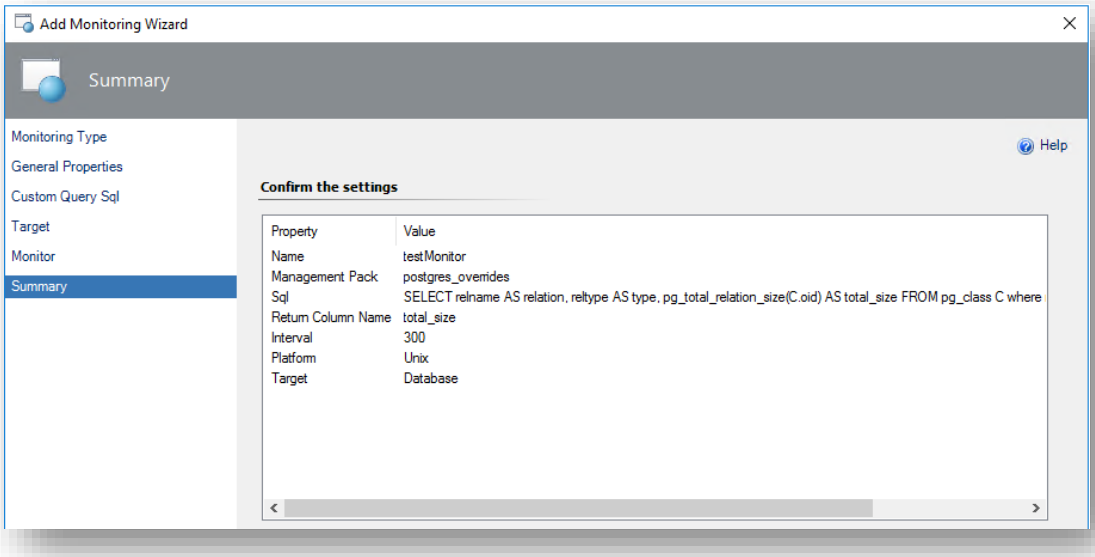
5. In the **Target** window, you have to select if you are querying server or database and select the **Platform** on which the PostgreSQL is running. Please select according to the test query in the previous step.



6. In the **Monitor** window specify if you want to enable the rule for all instances, select relational operator and enter **Warning Threshold**, **Error Threshold**, **Alert Name**, and **Alert Description**.

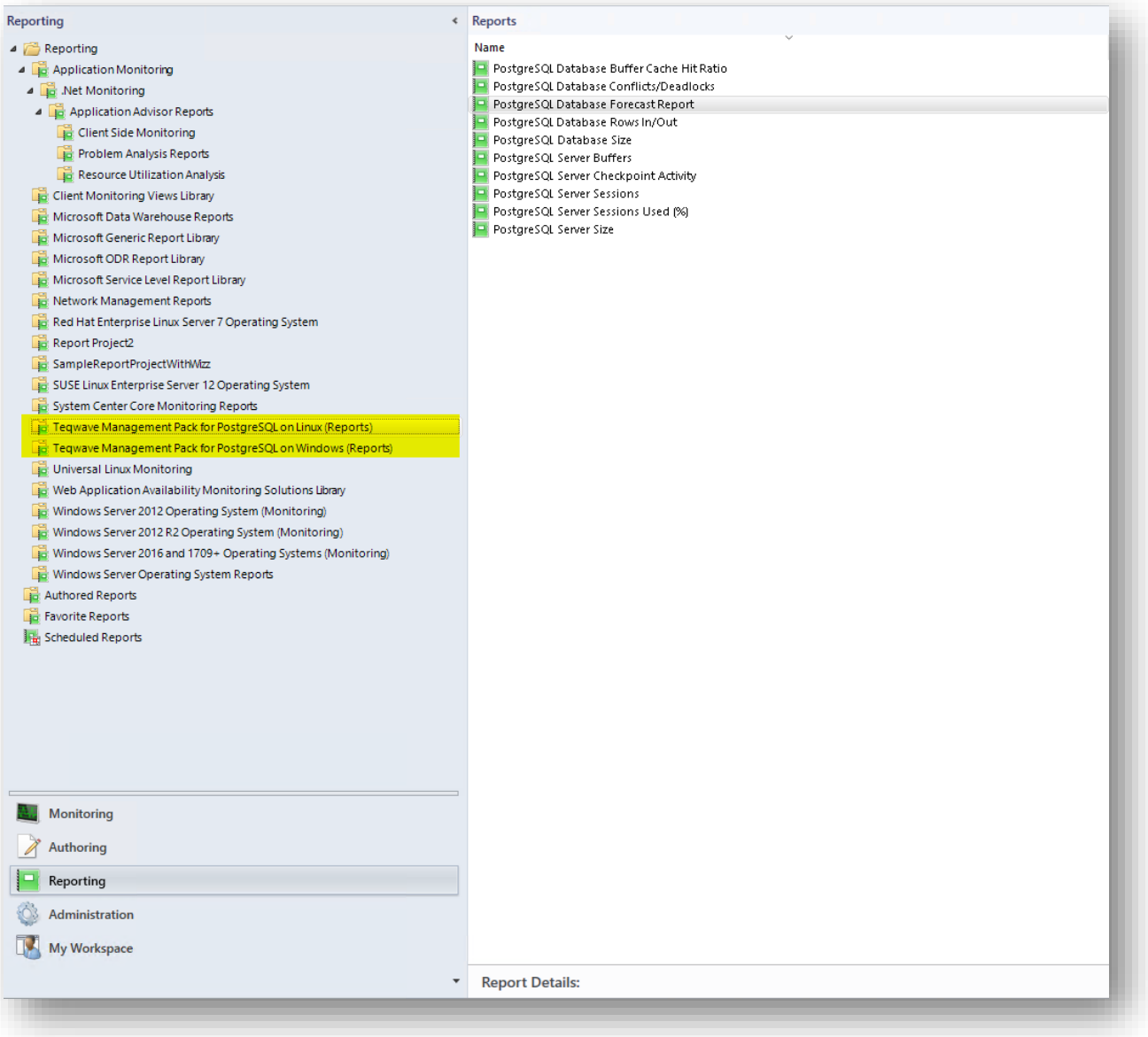


7. Check the configuration in the **Summary** window and press **Create** button to create a new Monitor.



REPORTS

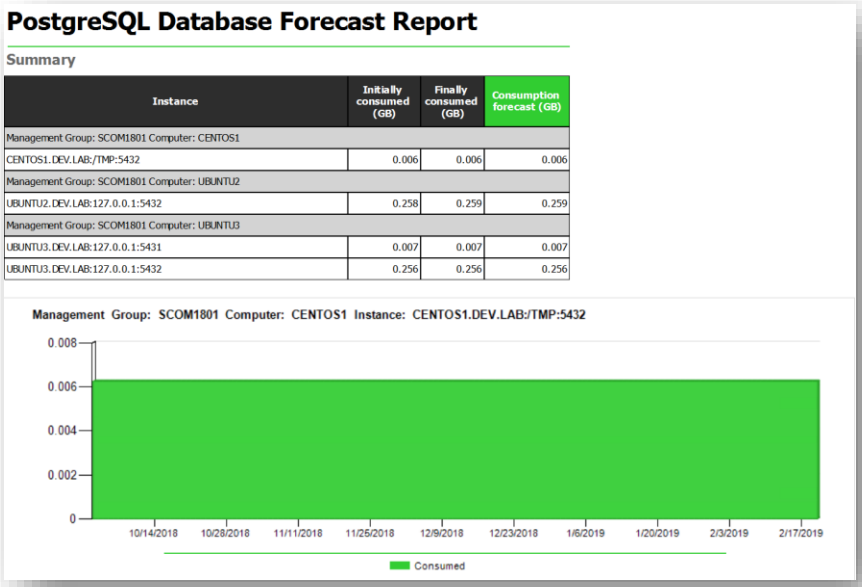
Teqwave PostgreSQL MP includes a comprehensive set of reports with rich functionality. The reports are available in the Reporting section of the Operations Manager.



DATABASE REPORTS

- **PostgreSQL Database Forecast Report**
  - The report displays several charts with the following performance items:
    - Initially consumed file space (GB)
    - Finally consumed file space (GB)
    - File space consumption forecast (GB)

The report displays a separate chart for every selected object or a group of objects.

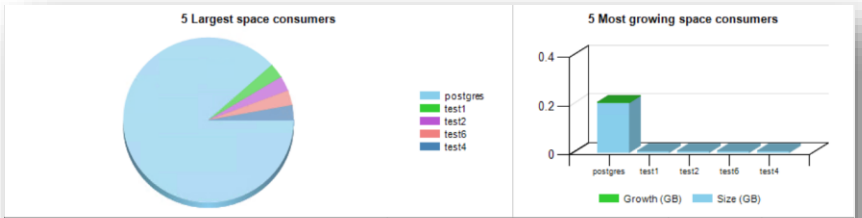


PostgreSQL Database Forecast Report

Summary

Instance	Initially consumed (GB)	Finally consumed (GB)	Consumption forecast (GB)
Management Group: SCOM1801 Computer: CENTOS1			
CENTOS1.DEV.LAB:/TMP:5432	0.006	0.006	0.006
Management Group: SCOM1801 Computer: UBUNTU2			
UBUNTU2.DEV.LAB:127.0.0.1:5432	0.258	0.259	0.259
Management Group: SCOM1801 Computer: UBUNTU3			
UBUNTU3.DEV.LAB:127.0.0.1:5431	0.007	0.007	0.007
UBUNTU3.DEV.LAB:127.0.0.1:5432	0.256	0.256	0.256

Management Group: SCOM1801 Computer: CENTOS1 Instance: CENTOS1.DEV.LAB:/TMP:5432



5 Largest space consumers

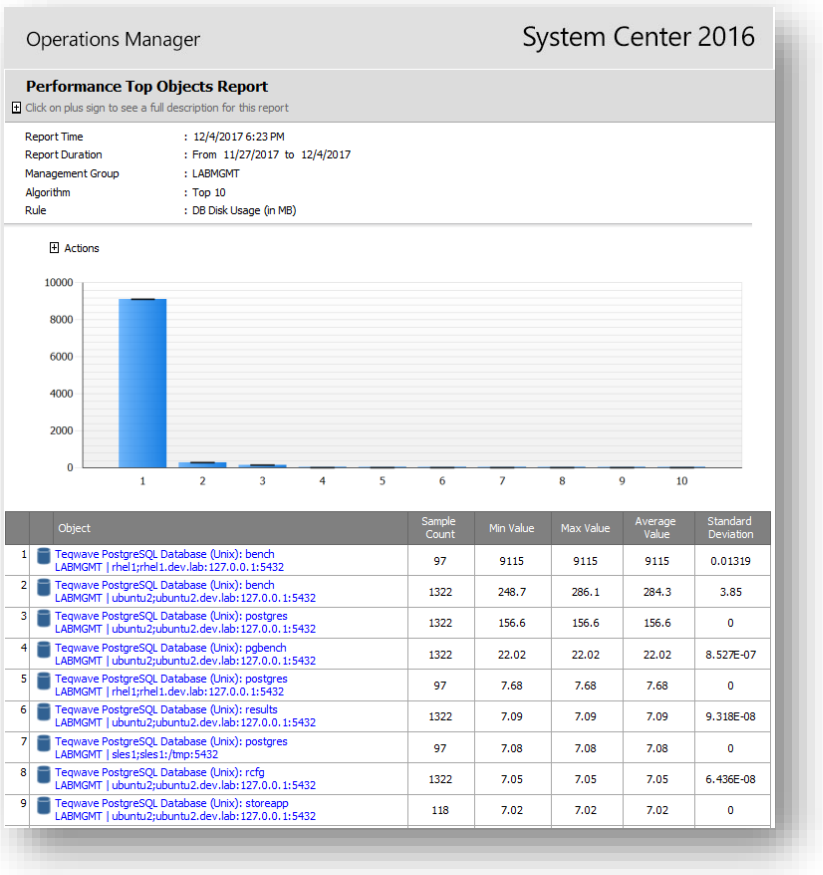
5 Most growing space consumers

Consumer	Size (GB)
postgres	0.205
test1	0.007
test2	0.007
test6	0.007
test4	0.007

Consumer	Size (GB)	Growth (GB)
postgres	0.205	0.000
test1	0.007	0.000
test2	0.007	0.000
test6	0.007	0.000
test4	0.007	0.000

Consumer	Initial size (GB)	Growth (GB)	Final size (GB)	Size forecast (GB)
postgres	0.204	0.000	0.205	0.205
test1	0.007	0.000	0.007	0.007
test2	0.007	0.000	0.007	0.007
test3	0.007	0.000	0.007	0.007
test4	0.007	0.000	0.007	0.007
test5	0.007	0.000	0.007	0.007
test6	0.007	0.000	0.007	0.007
test7	0.007	0.000	0.007	0.007

- **PostgreSQL Database Size**
  - Displays the top 10 PostgreSQL databases by size.



Operations Manager — System Center 2016

Performance Top Objects Report

Click on plus sign to see a full description for this report

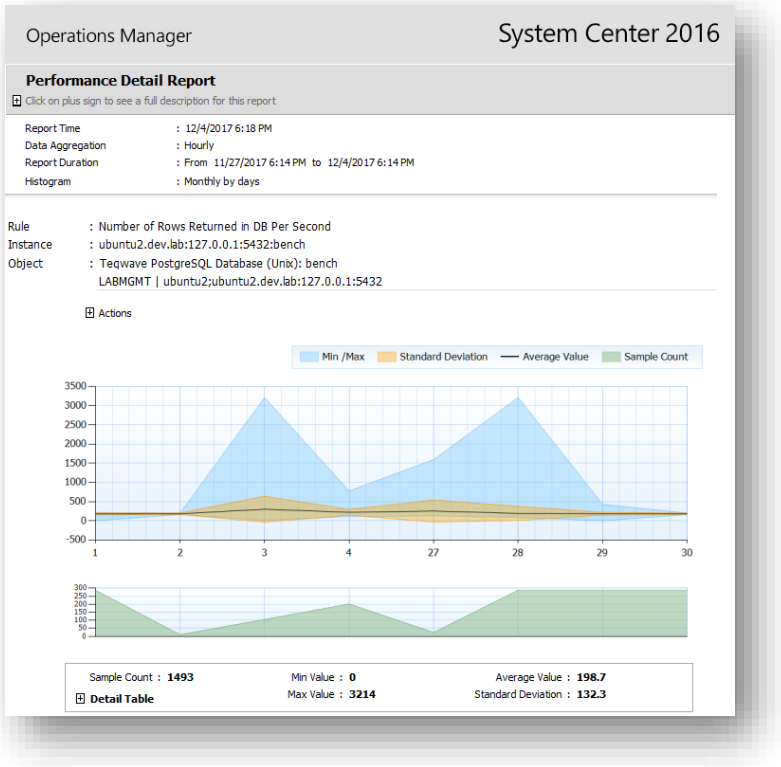
Report Time : 12/4/2017 6:23 PM
Report Duration : From 11/27/2017 to 12/4/2017
Management Group : LABMGMT
Algorithm : Top 10
Rule : DB Disk Usage (in MB)

Actions

	Object	Sample Count	Min Value	Max Value	Average Value	Standard Deviation
1	Teqwave PostgreSQL Database (Unix): bench LABMGMT | rhel1;rhel1.dev.lab:127.0.0.1:5432	97	9115	9115	9115	0.01319
2	Teqwave PostgreSQL Database (Unix): bench LABMGMT | ubuntu2;ubuntu2.dev.lab:127.0.0.1:5432	1322	248.7	286.1	284.3	3.85
3	Teqwave PostgreSQL Database (Unix): postgres LABMGMT | ubuntu2;ubuntu2.dev.lab:127.0.0.1:5432	1322	156.6	156.6	156.6	0
4	Teqwave PostgreSQL Database (Unix): pgbench LABMGMT | ubuntu2;ubuntu2.dev.lab:127.0.0.1:5432	1322	22.02	22.02	22.02	8.527E-07
5	Teqwave PostgreSQL Database (Unix): postgres LABMGMT | rhel1;rhel1.dev.lab:127.0.0.1:5432	97	7.68	7.68	7.68	0
6	Teqwave PostgreSQL Database (Unix): results LABMGMT | ubuntu2;ubuntu2.dev.lab:127.0.0.1:5432	1322	7.09	7.09	7.09	9.318E-08
7	Teqwave PostgreSQL Database (Unix): postgres LABMGMT | sles1;sles1:/tmp:5432	97	7.08	7.08	7.08	0
8	Teqwave PostgreSQL Database (Unix): rcfg LABMGMT | ubuntu2;ubuntu2.dev.lab:127.0.0.1:5432	1322	7.05	7.05	7.05	6.436E-08
9	Teqwave PostgreSQL Database (Unix): storeapp LABMGMT | ubuntu2;ubuntu2.dev.lab:127.0.0.1:5432	118	7.02	7.02	7.02	0

- **PostgreSQL Database Buffer Cache Hit Ratio**
  - Displays bottom 10 PostgreSQL databases by cache hit ratio.

- **PostgreSQL Database Rows In/Out**
  - Displays the following PostgreSQL database activity metrics:
    - Rows Deleted in DB Per Second
    - Rows Returned in DB Per Second
    - Rows Updated in DB Per Second
    - Rows Fetched in DB Per Second
    - Rows Inserted in DB Per Second



- **PostgreSQL Database Conflicts/Deadlocks**
  - Displays PostgreSQL database conflicts and deadlocks changing over the selected time period.

SERVER REPORTS

- **PostgreSQL Server Sessions**
  - Displays the following PostgreSQL Server metrics:
    - Total Sessions
    - Active Sessions
    - Idle Sessions
    - Idle Sessions in Transaction
    - Idle Sessions in Transaction (Aborted)

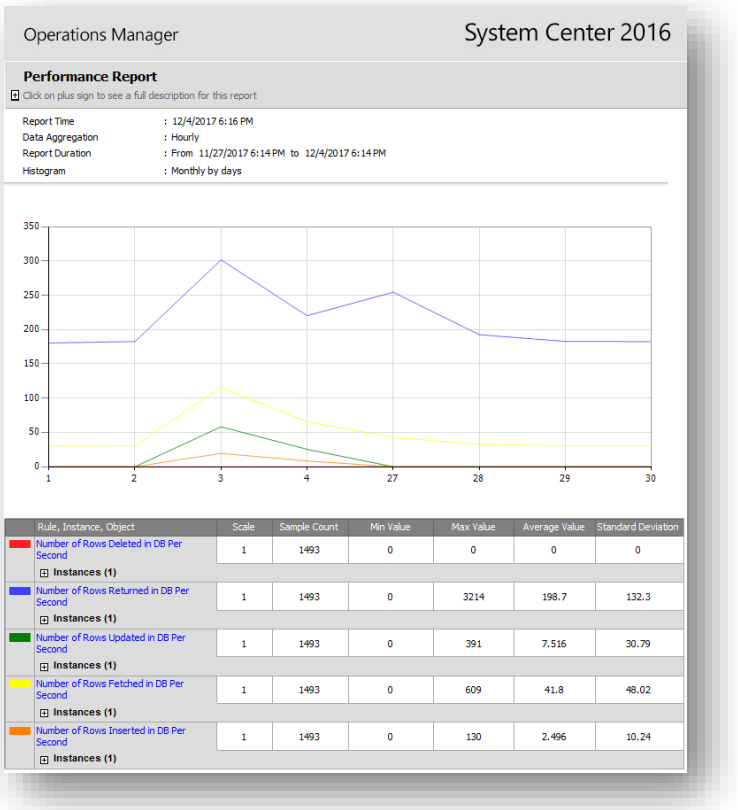
        - Sessions with Unknown State

- **PostgreSQL Server Sessions Used (%)**
    - Displays the percentage of used sessions over the selected time period.

- **PostgreSQL Server Checkpoint Activity**
    - Displays the number of scheduled and requested checkpoints in the selected time interval.

- **PostgreSQL Server Size**
    - Displays the top 10 PostgreSQL server instances by size.

- **PostgreSQL Server Buffers**
    - Displays a chart with the following PostgreSQL Server metrics:
        - Number of Buffers Allocated
        - Number of Buffers Written Directly by Backend
        - Number of Backend Fsync Calls
        - Number of Buffers Written by Background Writer
        - Number of Background Writer Stops
        - Number of Buffers Written During Checkpoints



APPENDIX A

RECOMMENDED SETTINGS FOR MONITORING USER

In order to take advantage of all functionality available the easiest way would be to use a superuser.

However, we know that you probably don't want to do that, so we prepared guidelines on how to setup a read only monitoring user for PostgreSQL 9.x or PostgreSQL 10.

In order for the index and table bloat monitoring functionality to work correctly, you will also need to configure additional privileges on each of the databases once the user is created. See Additional Privileges Required for Index and Table Bloat Monitoring.

Note: Note that the SQL statements must be run as a superuser (to create the SECURITY DEFINER function).

POSTGRESQL 9.X

Sample SQL to Create a Monitoring User - PostgreSQL 9.3 and up

```
CREATE SCHEMA IF NOT EXISTS teqwave;

CREATE OR REPLACE FUNCTION teqwave.get_stat_activity() RETURNS SETOF pg_stat_activity AS

$$

SELECT * FROM pg_catalog.pg_stat_activity;

$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


CREATE OR REPLACE FUNCTION teqwave.get_settings() RETURNS SETOF pg_settings AS

$$

SELECT * FROM pg_catalog.pg_settings;

$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


CREATE OR REPLACE FUNCTION teqwave.get_stats() RETURNS SETOF pg_stats AS

$$

SELECT schemaname, tablename, attname, inherited, null_frac, avg_width, n_distinct, NULL::anyarray, most
_common_freqs, NULL::anyarray, correlation, NULL::anyarray, most_common_elem_freqs, elem_count_histogram

FROM pg_catalog.pg_stats;

$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


CREATE OR REPLACE FUNCTION teqwave.get_tablespace_size(name text) RETURNS SETOF bigint AS
$$
SELECT * FROM pg_catalog.pg_tablespace_size(name);
$$ LANGUAGE sql VOLATILE SECURITY DEFINER;
```

```
CREATE OR REPLACE FUNCTION teqwave.get_replication() RETURNS TABLE (client_addr inet, client_hostname te
xt, state text, sent_location pg_lsn, replay_location pg_lsn, sync_state text) AS

$$

SELECT client_addr, client_hostname, state, sent_location, replay_location, sync_state FROM pg_stat_repl
ication WHERE state != 'backup';

$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


/** Create user for monitoring **/

CREATE USER monitoring_user WITH NOSUPERUSER NOINHERIT NOCREATEROLE NOCREATEDB LOGIN NOREPLICATION CONNE
CTION LIMIT 5 PASSWORD 'mypassword';

ALTER USER monitoring_user SET default_transaction_read_only TO 'true';

ALTER USER monitoring_user SET statement_timeout TO '2000';

ALTER USER monitoring_user SET lock_timeout TO '500';

ALTER USER monitoring_user SET temp_file_limit TO '0';


REVOKE ALL ON SCHEMA public FROM monitoring_user;

GRANT USAGE ON SCHEMA teqwave TO monitoring_user;
```

Sample SQL to Create a Monitoring User - PostgreSQL 9.2

```
CREATE SCHEMA teqwave;

CREATE OR REPLACE FUNCTION teqwave.get_stat_activity() RETURNS SETOF pg_stat_activity AS

$$

SELECT * FROM pg_catalog.pg_stat_activity;

$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


CREATE OR REPLACE FUNCTION teqwave.get_settings() RETURNS SETOF pg_settings AS

$$

SELECT * FROM pg_catalog.pg_settings;

$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


CREATE OR REPLACE FUNCTION teqwave.get_stats() RETURNS SETOF pg_stats AS

$$

SELECT schemaname, tablename, attname, inherited, null_frac, avg_width, n_distinct, NULL::anyarray, most
_common_freqs, NULL::anyarray, correlation, NULL::anyarray, most_common_elem_freqs, elem_count_histogram

FROM pg_catalog.pg_stats;

$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


CREATE OR REPLACE FUNCTION teqwave.get_tablespace_size(name text) RETURNS SETOF bigint AS
$$
```

```
SELECT * FROM pg_catalog.pg_tablespace_size(name);
$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


CREATE OR REPLACE FUNCTION teqwave.get_replication() RETURNS TABLE (client_addr inet, client_hostname te
xt, state text, sent_location pg_lsn, replay_location pg_lsn, sync_state text) AS

$$

SELECT client_addr, client_hostname, state, sent_location, replay_location, sync_state FROM pg_stat_repl
ication WHERE state != 'backup';

$$ LANGUAGE sql VOLATILE SECURITY DEFINER;


/** Create user for monitoring **/

CREATE USER monitoring_user WITH NOSUPERUSER NOINHERIT NOCREATEROLE NOCREATEDB LOGIN NOREPLICATION CONNE
CTION LIMIT 5 PASSWORD 'mypassword';

ALTER USER monitoring_user SET default_transaction_read_only TO 'true';

ALTER USER monitoring_user SET statement_timeout TO '2000';

ALTER USER monitoring_user SET temp_file_limit TO '0';


REVOKE ALL ON SCHEMA public FROM monitoring_user;

GRANT USAGE ON SCHEMA teqwave TO monitoring_user;
```

POSTGRESQL 10 OR LATER

Version 10 introduced a new monitoring role 'pg_monitor' which can be used to create a new restricted user that will be used for monitoring.

SQL to Create User - PostgreSQL 10 or newer

```
CREATE USER monitoring_user WITH PASSWORD 'mypassword' CONNECTION LIMIT 5;

GRANT pg_monitor TO monitoring_user;
```

ADDITIONAL PRIVILEGES REQUIRED FOR INDEX AND TABLE BLOAT MONITORING

After creating the user we need to set additional privileges on all databases in order for the index and table bloat monitoring to work correctly.

Sample SQL to Set Privileges on each Database for Index and Table Bloat

```
/** Revoke all on schema public **/

REVOKE ALL ON SCHEMA public FROM monitoring_user;

/** Allow the credential to look up tables and other database objects within the schema. **/

GRANT USAGE ON SCHEMA public TO monitoring_user;

/** Allow the credential to run SELECT on any table, view, materialized view, or foreign table in the sc
hema "public" (the default schema). You can add other schemas, separated by commas. **/

GRANT SELECT ON ALL TABLES IN SCHEMA public TO monitoring_user;
```

```
/** Allow the credential to automatically receive SELECT privileges on any new table, view, etc. created
 by the default user in the schema “public”. **/

ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT ON TABLES TO monitoring_user;
```

APPENDIX B

CONFIGURING SUDO ELEVATION FOR POSTGRESQL OMI PROVIDER

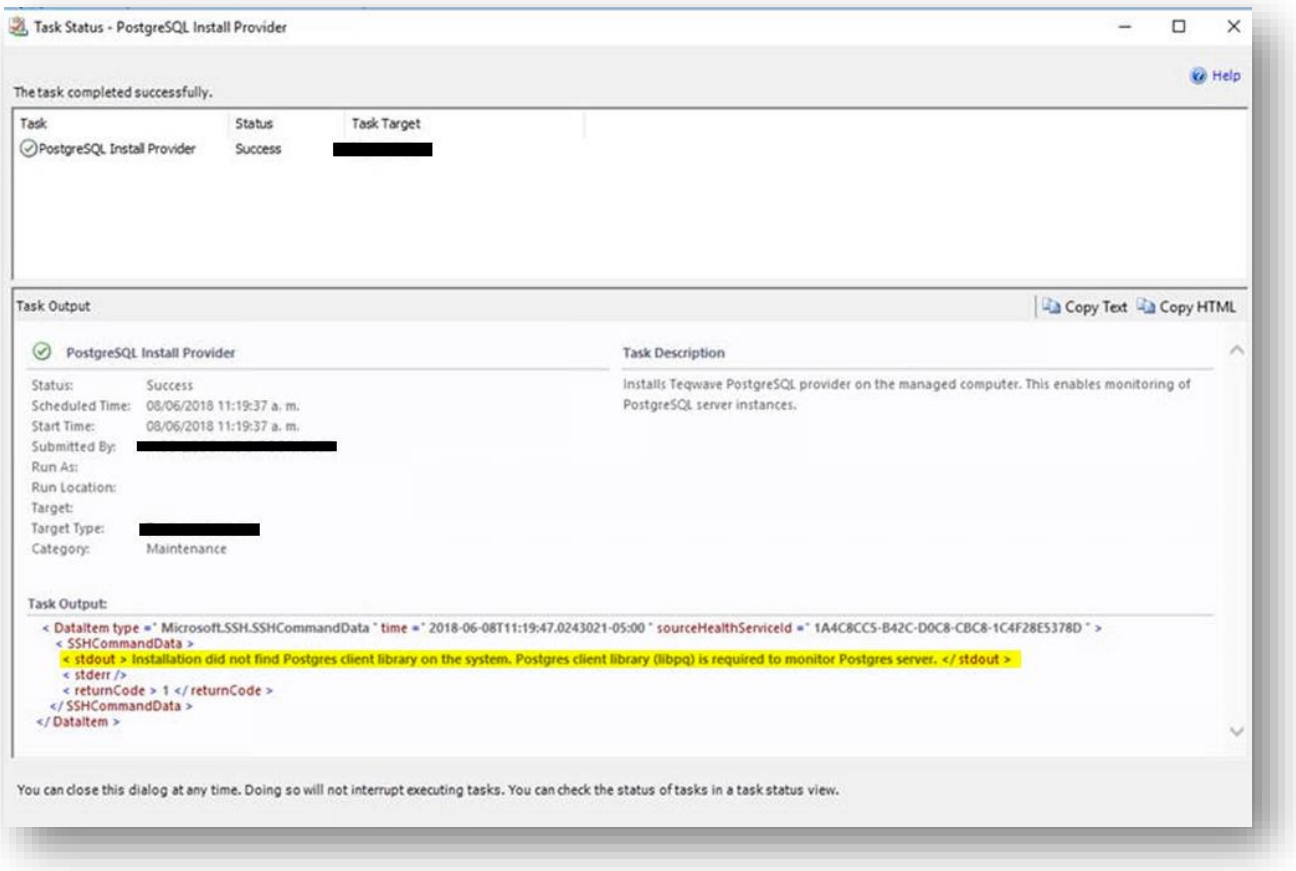
In case you are using sudo-enabled accounts for Operations Manager monitoring (for more details see https://social.technet.microsoft.com/wiki/contents/articles/7375.scom-1807-1801-2016-and-2012-configuring-sudo-elevation-for-unix-and-linux-monitoring.aspx), the sudoers (/etc/sudoers) file must be updated on each PostgreSQL computer where the PostgreSQL OMI Provider will be installed.

The sample configuration below provides "scomadm" user with the minimum necessary authorization to perform install/upgrade/remove activities.

```
##Install/upgrade Teqwave PostgreSQL Provider

scomadm ALL=(root) NOPASSWD: /bin/sh -c sh /tmp/scx-scomadm/teqwave-
postgresmp.x64.sh --install; EC=$?; cd /tmp; rm -rf /tmp/scx-scomadm; exit
$EC

scomadm ALL=(root) NOPASSWD: /bin/sh -c sh /tmp/scx-scomadm/teqwave-
postgresmp.x64.sh --upgrade; EC=$?; cd /tmp; rm -rf /tmp/scx-scomadm; exit
$EC


##Uninstall Teqwave PostgreSQL Provider

scomadm ALL=(root) NOPASSWD: /bin/sh -c sh /tmp/scx-scomadm/teqwave-
postgresmp.x64.sh --purge; EC=$?; cd /tmp; rm -rf /tmp/scx-scomadm; exit $EC
```

INSTALLATION OF POSTGRESQL MP PROVIDER ON LINUX FAILS (LIBPQ LIBRARY NOT FOUND)

Installation of PostgreSQL MP provider on Linux will fail if postgres client library (libpq) is not found.



To check if all required libraries are available, run the following command:

```
/$ldd /opt/teqwave/PostgresMP/lib/libPostgresProvider.so
```

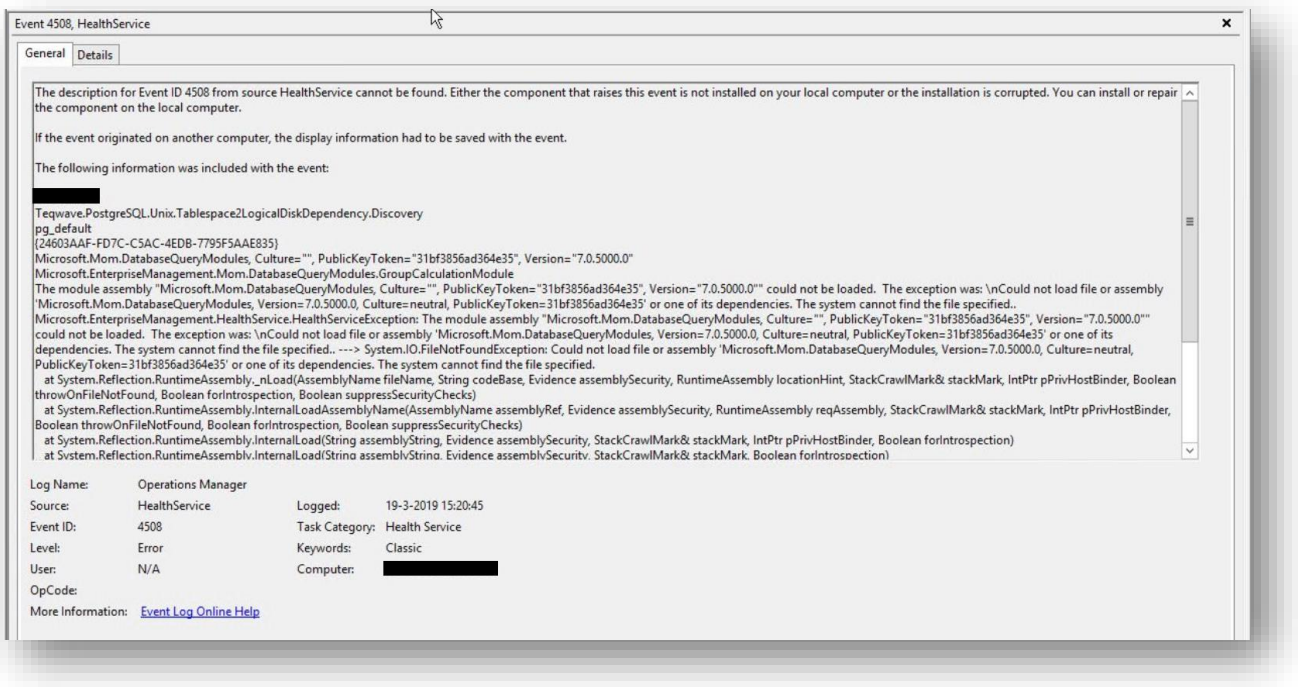
To find the libpq library on the system run:

```
find / -name libpq.so*
```

Create a file `/etc/ld.so.conf.d/libpsql.conf` and add path to the libpq library (e.g. `/usr/local/pgsql/lib/`) to the created file and run `ldconfig`

TABLESPACE TO OS LOGICAL DISK RELATIONSHIP DISCOVERY FAILS IF SCOM GATEWAY IS USED FOR MONITORING

If SCOM Gateway is used to monitor Linux PostgreSQL servers, then tablespace to OS disk relationship discovery will fail (see screenshot below) because of the missing libraries on the SCOM Gateway system.



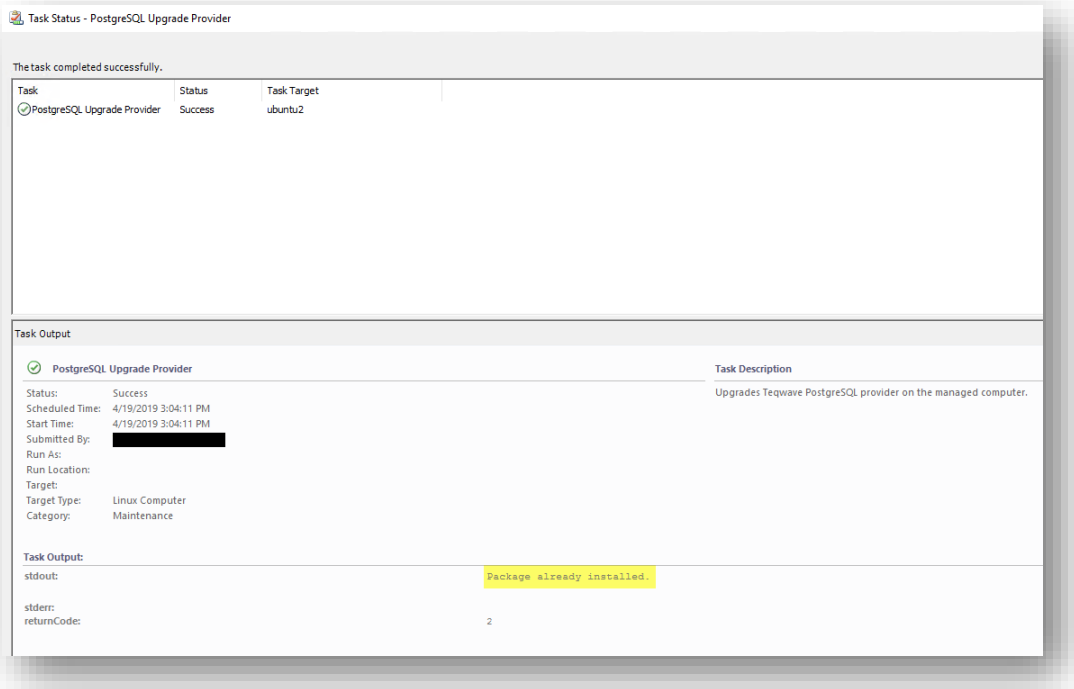
Additional discovery, which is disabled by default, has been added. This discovery has to be used only when the Linux PostgreSQL server is monitored with SCOM Gateway. Procedure to enable discovery:

- Disable the original discovery `PostgreSQL Tablespace To Logical Disk Relationship Discovery (Unix)` for all PostgreSQL servers that are monitored using a SCOM gateway

- Enable the optional discovery `PostgreSQL Tablespace To Logical Disk Relationship Discovery - Optional (Unix)` for all PostgreSQL servers that are monitored using a SCOM gateway

"PACKAGE ALREADY INSTALLED" ERROR WHEN UPGRADING POSTGRESQL OMI PROVIDER

If you are upgrading PostgreSQL OMI provider, it might take up to 12 hours for the provider installation package to be updated on the SCOM Management Servers. To speed this up, we recommend restarting Microsoft Monitoring Agent service on all SCOM Management Servers that are dedicated to monitoring PostgreSQL Linux servers.



To check if the latest OMI Provider installation package (`teqwave-postgresmp.x64.sh`) was refreshed on the SCOM Management Servers, please check the file modification date in `C:\ProgramData\Teqwave\PostgreSQL_MP` folder.