



EDB

Postgres Enterprise Manager - User Guide

Release 8.5.0

EDB

Jul 15, 2022

CONTENTS

1	Accessing the web interface	3
1.1	The PEM menu bar	5
1.1.1	The File menu	5
1.1.2	The Object menu	6
1.1.3	The Management menu	7
1.1.4	The Dashboards menu	8
1.1.5	The Tools menu	8
1.1.6	The Help menu	9
1.2	Controlling and customizing charts, graphs and tables	9
2	Monitoring performance	11
2.1	Probes	13
2.1.1	System probes	13
2.1.2	BDR probes	18
2.1.3	Customizing probes	28
2.1.4	PEM probes – reference	38
2.2	Alerts	42
2.2.1	Viewing the alerts via Global dashboard	42
2.2.2	Viewing the alerts via Alerts dashboard	44
2.2.3	Customizing the Alerts dashboard	45
2.2.4	Defining a new alert template	47
2.2.5	Exporting or importing alert templates	50
2.2.6	Modifying or deleting an alert template	52
2.2.7	Templates applicable on agent	52
2.2.8	Templates applicable on server	54
2.2.9	Templates applicable on database	57
2.2.10	Templates applicable on schema	59
2.2.11	Templates applicable on table	60
2.2.12	Global templates	61
2.2.13	Audit log alerting	61
2.2.14	Schedule an alert blackout	71
2.3	Notifications	73
2.3.1	SMTP	74
2.3.2	Webhook	76
2.3.3	SNMP	81
2.3.4	Using PEM with Nagios	81
2.4	Capacity Manager	85
2.4.1	Capacity Manager templates	90
2.4.2	Capacity Manager metrics - reference	90
2.5	Audit Manager	93

2.5.1	Setting the EDB Postgres Advanced Server instance service ID	95
2.5.2	Setting the EDB Audit Configuration probe	96
2.5.3	Configuring audit logging with the Audit Manager	96
2.5.4	Viewing the log with the Audit Log dashboard	101
2.6	Log Manager	102
2.6.1	Reviewing the Server Log Analysis dashboard	109
2.6.2	Postgres Log Analysis Expert	111
2.6.3	Reviewing the Postgres Log Analysis Expert report	116
2.7	Charts	117
2.7.1	Creating a custom chart	118
2.7.2	Importing a Capacity Manager template	120
2.8	Dashboards	126
2.8.1	Dashboards overview	126
2.8.2	Dashboard configuration	128
2.8.3	Managing custom dashboards	129
2.9	Remote monitoring	133
3	Monitoring Barman	135
3.1	Prerequisites for monitoring Barman	135
3.2	Configuring a Barman server	135
3.2.1	Using PEM web client	135
3.2.2	Configure	135
3.2.3	Editing	137
3.2.4	Using pemworker command line	138
3.3	Viewing the Barman server details on a PEM dashboard	139
4	Managing BART	143
4.1	Prerequisites	143
4.2	Configuring a BART server	143
4.3	Associating the BART server with a database server	147
4.4	Viewing the BART server details on a PEM dashboard	149
4.5	Scheduling BART backups	150
4.6	Scheduling BART obsolete backups deletion	153
4.7	Restoring BART backups	155
5	Monitoring BDR	159
6	Monitoring Failover Manager	161
6.1	Replacing a primary node	162
6.2	Switchover EFM cluster	163
7	Monitoring Replication Server	165
8	Tuning performance	167
8.1	Tuning Wizard	167
8.2	Postgres Expert - best practice enforcement	173
8.2.1	Using the Postgres Expert wizard	173
8.2.2	Reviewing Postgres Expert recommendations	174
8.3	Performance Diagnostic	177
9	Profiling workloads	185
9.1	SQL Profiler	185
9.1.1	Installing the SQL Profiler plugin	185
9.1.2	SQL profiling and analysis	193
9.1.3	Uninstalling SQL Profiler	200

9.2	Using the Index Advisor	201
10	Viewing system reports	203
10.1	System Configuration Report	203
10.2	Core Usage report	204
11	PEM command line interface	209
11.1	Example	212

Welcome to Postgres Enterprise Manager (PEM) User Guide. This guide describes all the functions and features that users will use in PEM. PEM consists of components that provide the management and analytical functionality for your EDB Postgres Advanced Server or PostgreSQL database. PEM is based on the Open Source pgAdmin 4 project. PEM is a comprehensive database design and management system. PEM is designed to meet the needs of both novice and experienced Postgres users alike, providing a powerful graphical interface that simplifies the creation, maintenance, and use of database objects.

ACCESSING THE WEB INTERFACE

After installing a PEM server and agent, you can configure PEM to start monitoring and managing PostgreSQL or EDB Postgres Advanced Server instances. The PEM server installer installs the PEM web interface. You can use the interface to review information about objects that reside on monitored servers or to review statistical information gathered by the PEM server.

After the server installation completes, you can open the PEM interface in your choice of browser by navigating to:

`https://<ip_address_of_PEM_host>:8443/pem`

`ip_address_of_PEM_host` specifies the IP address of the host of the PEM server.



Fig. 1: The PEM Web Login page

Use the fields on the Postgres Enterprise Manager Login window to authenticate yourself with the PEM server:

- Provide the name of a `pem` database user in the **Username** field. For the first user connecting, this is the name provided when installing the PEM server.
- Provide the password associated with the user in the **Password** field.

After providing your credentials, select **Login** to connect to PEM.

The PEM web interface opens, displaying the Global Overview dashboard.

The Browser pane displays a tree that provides access to information about the database objects that reside on each server. The tree expands to display a hierarchical view of the servers and objects that the PEM server monitors.

Before you can use the PEM web interface to manage or monitor a database server, you must register that server with the PEM server. When you register a server, you describe the connection to the server, provide authentication information for the connection, and specify any management preferences (optionally binding an agent).

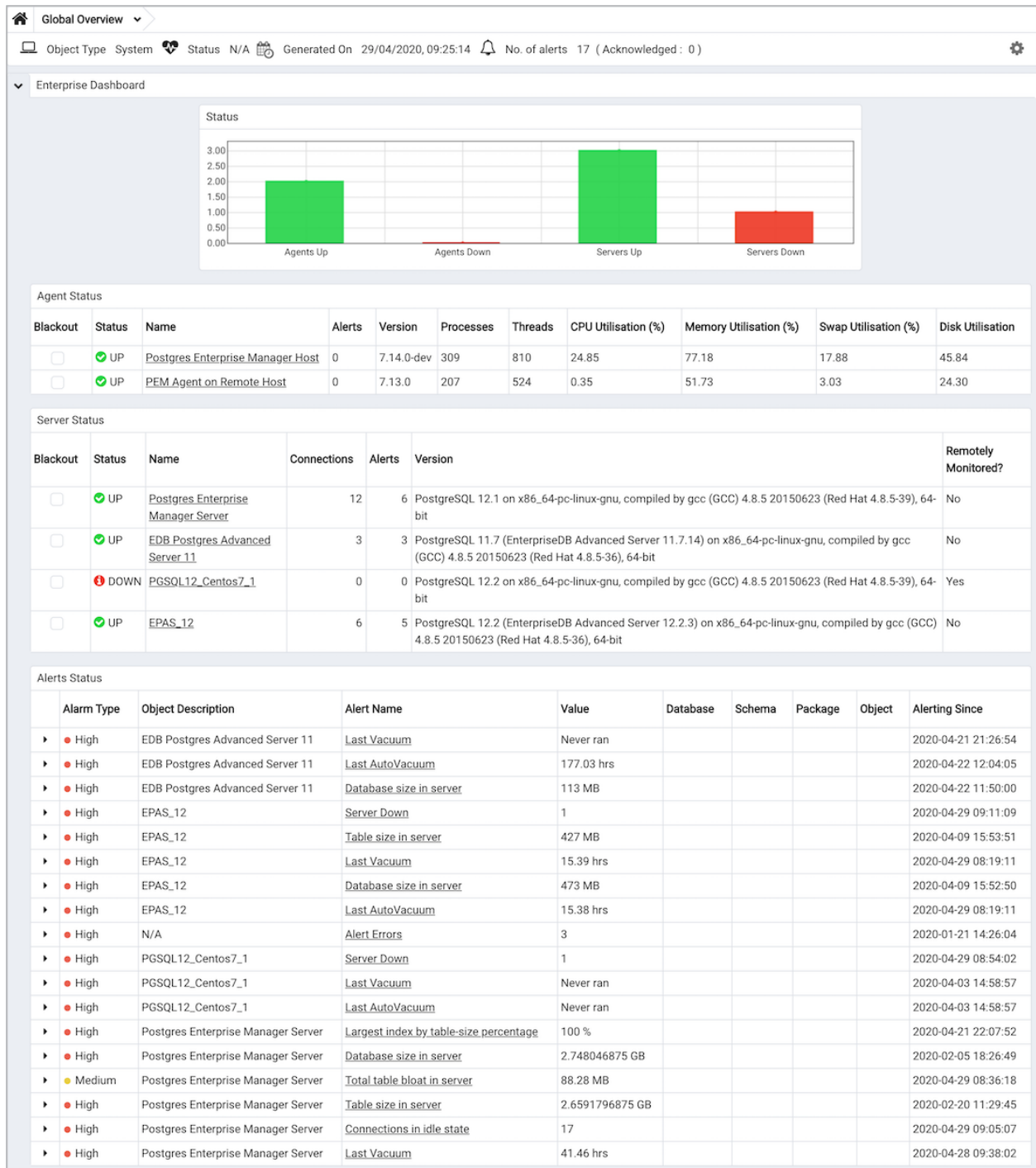


Fig. 2: The Global Overview Dashboard

A server can be managed or unmanaged:

- A *managed* server is bound to a PEM agent. The PEM agent monitors the server to which it's bound and performs tasks or reports statistics for display on the PEM dashboards. A managed server has access to extended PEM functionality such as custom alerting. When registering a server, you can also allow PEM to restart a managed server as required.
- An *unmanaged* server isn't bound to a PEM agent. You can create database objects on an unmanaged server, but extended PEM functionality (such as custom alerting) isn't supported on an unmanaged server.

You must also ensure the `pg_hba.conf` file of the server that you're registering allows connections from the host of the PEM web interface.

To access online help for the PEM web interface, select **Help** from the menu bar. The online help content is also available from the

EDB website .

1.1 The PEM menu bar

The PEM menu bar provides access to commands and features that you can use to manage your database servers and the objects that reside on those servers. If an option is disabled:

- The database server to which you are currently connected might not support the selected feature.
- The selected menu option might not apply to the current object.
- The role that you used to connect to the server might not have enough privileges to change the selected object.

Context-sensitive menus across the top of the PEM web interface allow you to customize your environment and provide access to the enterprise management features of PEM.

1.1.1 The File menu

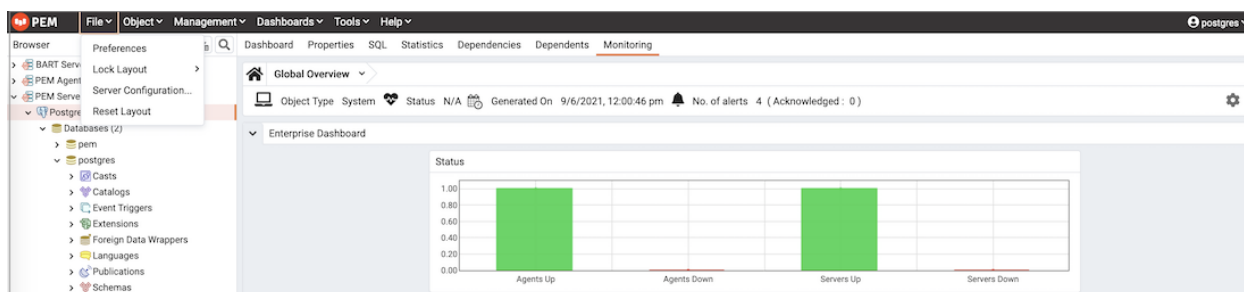


Fig. 3: The File Menu

Use the **File** menu to access the following options.

Option	Action
Preferences	Open the Preferences dialog box to customize your PEM client settings.
Lock Layout	Open a submenu to select the level for locking the UI layout.
Server Configuration	Open the Server Configuration dialog box and update your PEM server configuration settings.

continues on next page

Table 1 – continued from previous page

Option	Action
Reset Layout	If a workspace panel is popped out by mistake or intentionally, you can reset it to the default using Reset Layout .

1.1.2 The Object menu

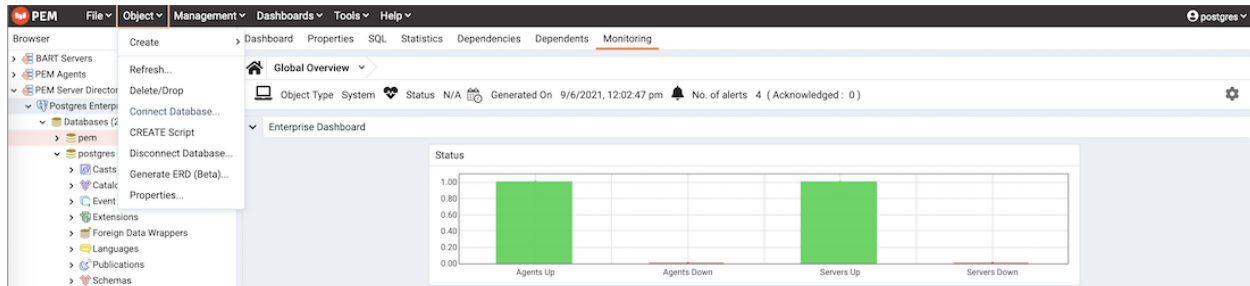


Fig. 4: The Object Menu

The **Object** menu is context sensitive. Use the **Object** menu to access the following options.

Option	Action
Create	Access a menu that provides context-sensitive selections.
Refresh	Refresh the currently selected object.
Delete/Drop	Delete the currently selected object from the server.
Connect Server	Open the Connect to Server dialog box to establish a connection with a server.
CREATE Script	Open the Query tool to edit or view the selected script.
Disconnect Server	Refresh the currently selected object.
BART	Access a menu that provides options for removing BART configuration, taking a BART backup, or revalidating the BART configuration.
Clear Saved Password	If you saved the database server password, clear the saved password. Enabled only after password is saved.
Clear SSH Tunnel Password	If you saved the ssh tunnel password, clear the saved password. Enabled only after password is saved.
Drop Cascade	Delete the currently selected object and all dependent objects from the server.
Hide	Hide the currently selected group. To view hidden groups, enable the Show hidden groups option in Preferences.
Properties	Review or modify the currently selected object's properties.
Trigger(s)	Disable or enable triggers for the currently selected table.
Truncate	Remove all rows from a table (Truncate) or remove all rows from a table and its child tables (Truncate Cascade).
View Data	Access a menu that provides several options for viewing data.

1.1.3 The Management menu

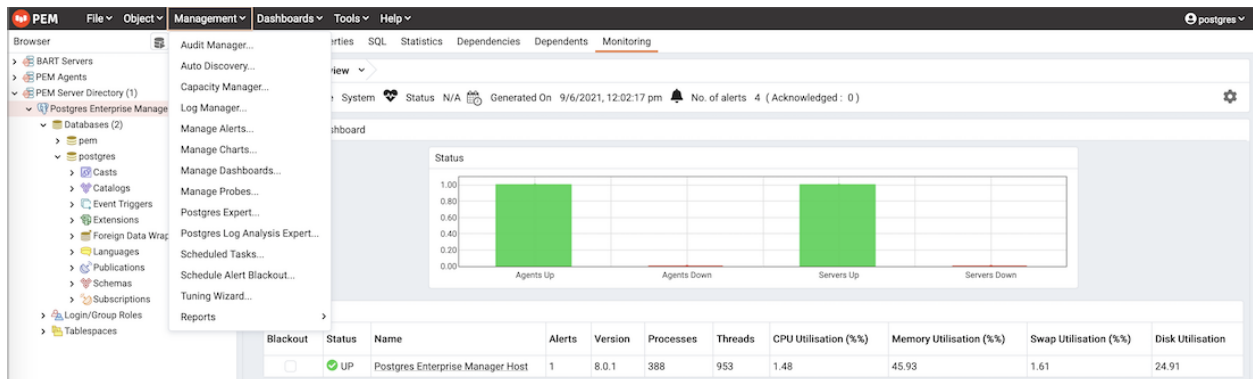


Fig. 5: The Management Menu

Use the **Management** menu to access the following PEM features.

Option	Action
Audit Manager	Open the Audit Manager and configure auditing on your monitored servers.
Auto Discovery	Open the Auto Discovery dialog box to configure a PEM agent to locate and bind monitored database servers.
Capacity Manager	Open the Capacity Manager dialog box and analyze historical or project future resource usage.
Log Manager	Open the Log Manager dialog box and configure log collection for a server.
Manage Alerts	Open the Manage Alerts tab and create or modify behavior for alerts.
Manage Charts	Open the Manage Charts tab to create or modify PEM charts.
Manage Dashboards	Open the Manage Dashboards dialog box to VACUUM, ANALYZE, REINDEX, or CLUSTER.
Manage Probes	Open the Manage Probes dialog box to VACUUM, ANALYZE, REINDEX, or CLUSTER.
Postgres Expert	Open the Postgres Expert wizard and perform a static analysis of your servers and databases.
Postgres Log Analysis Expert	Open the Postgres Log Analysis Expert dialog box to analyze log file contents for usage trends.
Scheduled Tasks	Open the Scheduled Tasks tab and review tasks that are pending or recently completed.
Tuning Wizard	Open the Tuning Wizard dialog box to generate a set of tuning recommendations for your server.
Reports	Open the Reports dialog box to generate the system configuration report and core usage report for your server.

1.1.4 The Dashboards menu

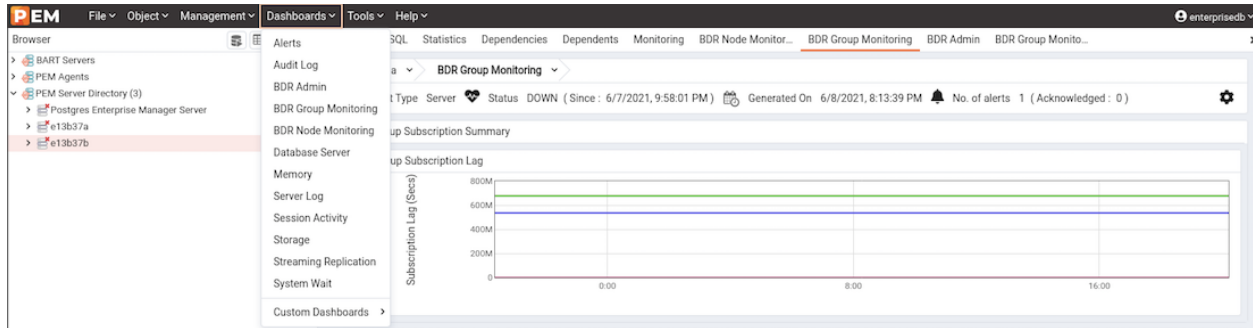


Fig. 6: The Dashboards menu

Use the context-sensitive **Dashboards** menu to access dashboards.

Option	Action
Alerts	Open the Alerts dashboard for the selected node.
Audit Log	Open the Audit Log Analysis dashboard for the selected node.
BDR Admin	Open the BDR Admin dashboard for the selected node.
BDR Group Monitoring	Open the BDR Group Monitoring dashboard for the selected node.
BDR Node Monitoring	Open the BDR Node Monitoring dashboard for the selected node.
Database Server	Open the Database Analysis dashboard for the selected node.
I/O Analysis	Open the I/O Analysis dashboard for the selected node.
Memory	Open the Memory Analysis dashboard for the selected node.
Object Activity	Open the Object Activity Analysis dashboard for the selected node.
Operating System	Open the Operating System Analysis dashboard for the selected node.
Probe Log	Open the Probe Log Analysis dashboard for the selected node.
Server Log	Open the Server Log Analysis dashboard for the selected node.
Session Activity	Open the Session Activity Analysis dashboard for the selected node.
Storage	Open the Storage Analysis dashboard for the selected node.
Streaming Replication	Open the Streaming Replication Analysis dashboard for the selected node.
System Wait	Open the System Wait Analysis dashboard for the selected node.

1.1.5 The Tools menu

Use the options on the **Tools** menu to access the following features.

Option	Action
Schema Diff	Open the Schema Diff dialog box to compare the schema objects between two database schemas.
Search objects	Open the Search Objects dialog box to search the database objects in a database.
Server	Access the various server-related tools such as Add Named Restore Point, Performance Diagnostics, Queue Server Startup, Queue Server Shutdown, Replace Cluster Primary, Switchover EFM Cluster, and SQL Profiler.
Query Tool	Open the Query tool for the currently selected object.
Reload Configuration	Update configuration files without restarting the server.
Pause replay of WAL	Pause replay of the WAL log.
Resume replay of WAL	Resume replay of the WAL log.

continues on next page

Table 5 – continued from previous page

Option	Action
Import/Export	Open the Import/Export Data dialog box to import or export data from a table.
Maintenance	Open the Maintenance dialog box to VACUUM, ANALYZE, REINDEX, or CLUSTER.
Backup	Open the Backup dialog box to back up database objects.
Backup Globals	Open the Backup Globals dialog box to back up cluster objects.
Backup Server	Open the Backup Server dialog box to back up a server.
Restore	Open the Restore dialog box to restore database files from a backup.
Grant Wizard	Open the Grant Wizard tool.
Schedule Backup	Open the Schedule Backup dialog box for BART backups.

1.1.6 The Help menu

Use the options on the **Help** menu to access the online help documents or to review information about the PEM installation.

Option	Action
Online Help	Open documentation for Postgres Enterprise Manager.
REST API Reference	Open the REST API Reference.
EDB Website	Open the EDB website in a browser window.
About Postgres Enterprise Manager	Locate versioning and user information for Postgres Enterprise Manager.

1.2 Controlling and customizing charts, graphs and tables

Use the icons in the upper-right corner of each graphic on a dashboard to control and customize the charts, graphs, and tables displayed in the PEM client for your current user session.

The following table gives the information about icons.

Icons	Information
! icon6	View information about the chart, graph, or table.
! icon1	Refresh the content of a chart, graph, or table.
! icon3	Personalize the chart, graph, or table settings for the current user.
! icon4	Download an image of the chart or graph.
! icon5	View the legends that are used in the chart, graph, or table.
! icon2	Expand the chart or graph to full-screen.

For more information about customizing the graphics displayed on the PEM dashboards, see the PEM client online help.

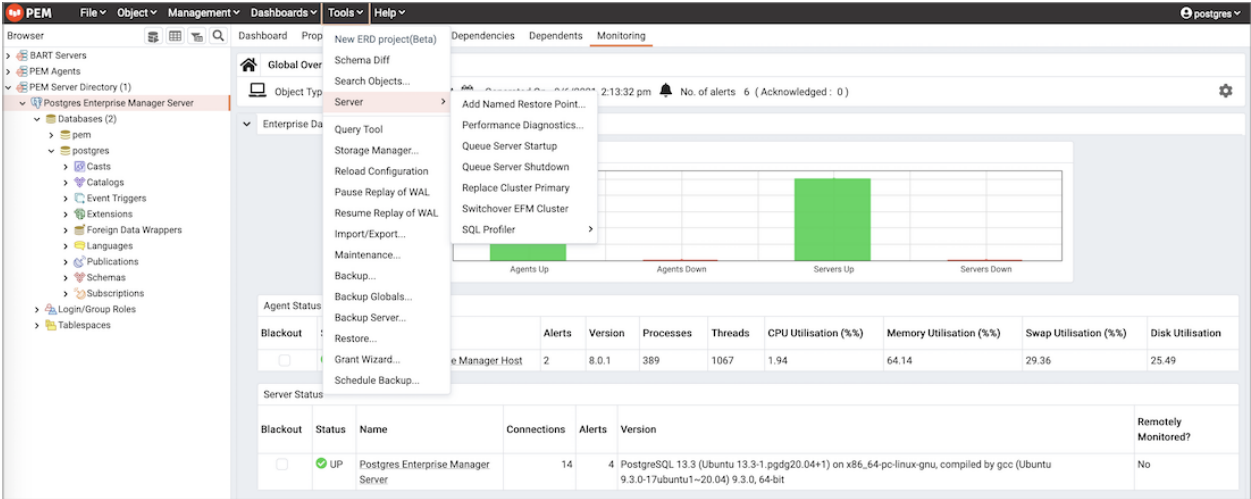


Fig. 7: The Tools menu

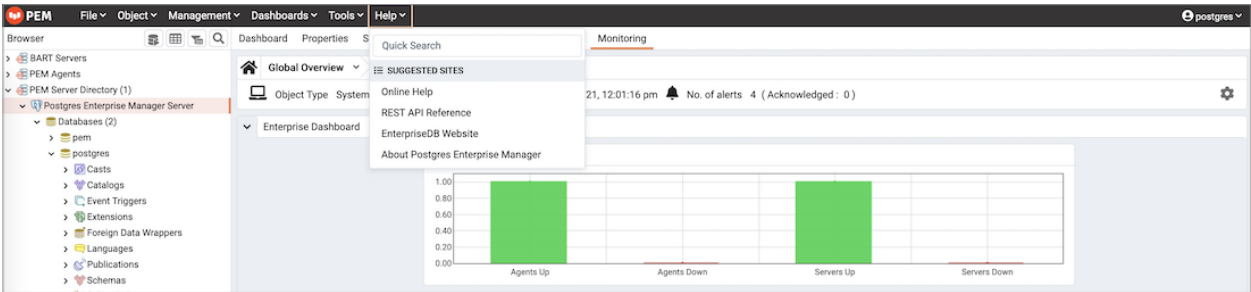


Fig. 8: The Help menu

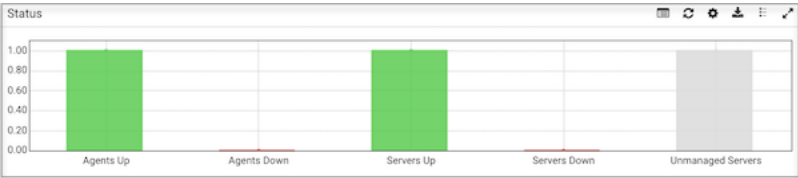


Fig. 9: The PEM Client chart control icons

MONITORING PERFORMANCE

PEM enables you to implement enterprise-wide performance monitoring of all managed servers.

The top-level dashboard is Global Overview. The Global Overview dashboard presents a status summary of all the servers and agents that are being monitored by the PEM server, a list of the monitored servers, and the state of any currently triggered alerts.

PEM provides a number of benefits not found in any other PostgreSQL management tool:

- **Management en masse design** — PEM is designed for enterprise database management and is built to tackle the management of large numbers of servers across geographical boundaries. Global dashboards visually keep you up to date on the up/down/performance status of all your servers.
- **Distributed architecture** — The PEM architecture maximizes the ability to gather statistical information and to perform operations remotely on machines regardless of operating system platform.
- **Graphical administration** — All aspects of database administration can be carried out in the PEM client using a graphical interface. You can handle server startup and shutdown, configuration management, storage and security control, object creation, performance management, and more from a single console.
- **Full SQL IDE** — PEM contains a robust SQL integrated development environment (IDE) that provides ad hoc SQL querying, stored procedure/function development, and a graphical debugger.
- **Enterprise performance monitoring** — PEM provides enterprise-class performance monitoring for all managed database servers. Lightweight and efficient agents monitor all aspects of each database server's operations as well as each machine's underlying operating system. They provide detailed statistics to easily navigate performance pages in the interface.
- **Proactive alert management** — PEM lets you create performance thresholds for each key metric, such as memory and storage, that are monitored around the clock. Any threshold violation results in an alert sent to a centralized dashboard that communicates the nature of the problem and the actions needed to prevent the situation from jeopardizing the overall performance of the server.
- **Simplified capacity planning** — All key performance-related statistics are collected and retained for a specified period in the PEM repository. The Capacity Manager utility allows you to select various statistics and perform trend analysis to understand things such as peak load periods and storage consumption trends. A forecasting mechanism in the tool allows you to also forecast resource usage so you can plan and budget accordingly.
- **Audit Manager** — Audit Manager configures audit logging on EDB Postgres Advanced Server instances. You can log activities such as connections to a database, disconnections from a database, and the SQL statements run against a database. You can then use the Audit Log dashboard to filter and view the log.
- **Log Manager** — The Log Manager wizard configures server logging parameters with (optional) log collection into a central table. Use the wizard to specify your preference for logging behaviors such as log file rotation, log destination, and the error message severity. Use the Server Log dashboard to filter and review the collected server log entries.

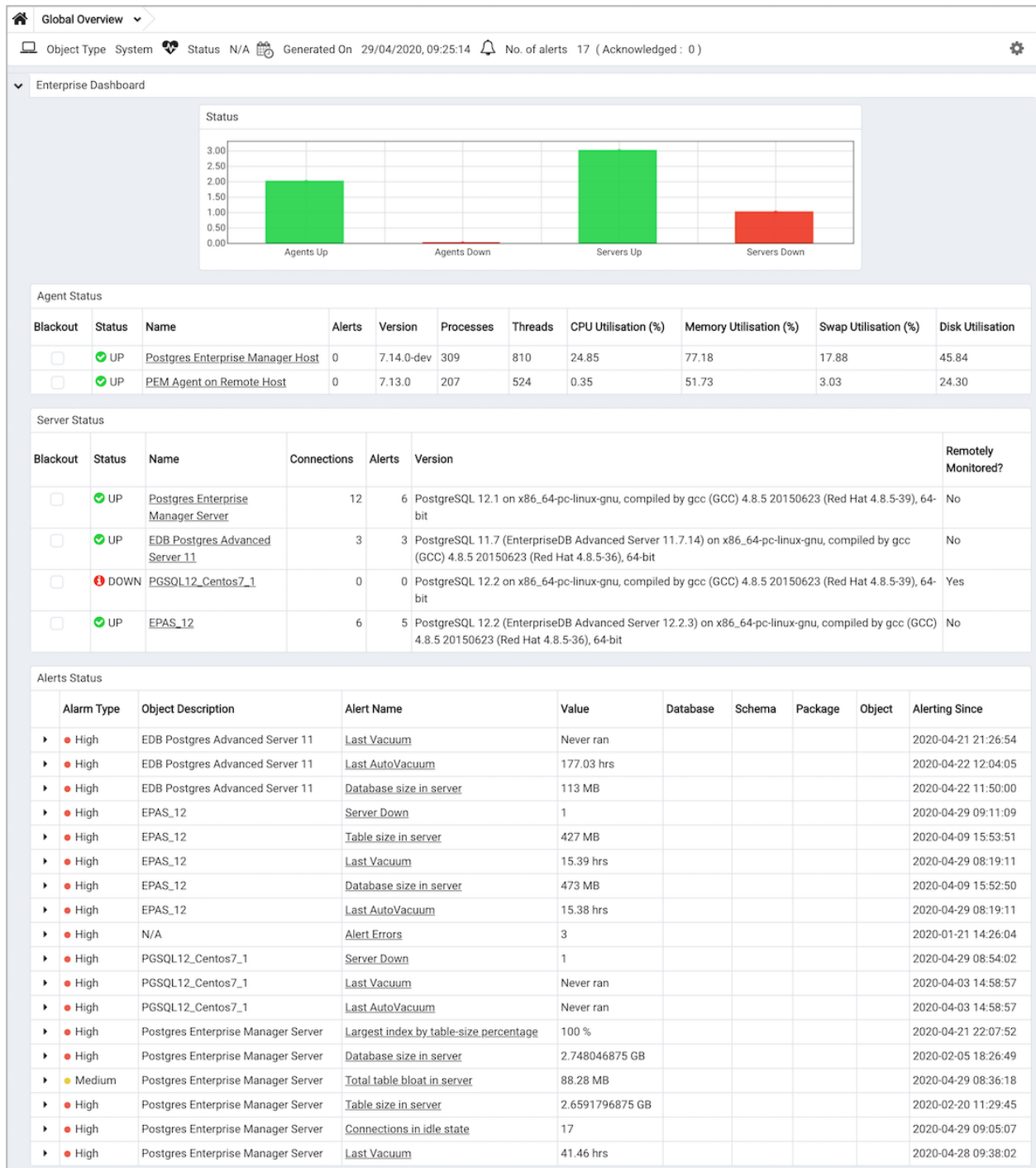


Fig. 1: The Global Overview dashboard

- SQL workload profiling — PEM contains a SQL profiling utility that allows you to trace the SQL statements that are executed against one or more servers. SQL profiling can be done either in an ad hoc or scheduled manner. You can then filter captured SQL statements so you can easily identify and tune SQL statements that are running poorly. You can also feed SQL statements into an index advisor on EDB Postgres Advanced Server that analyzes each statement and makes recommendations on new indexes to create to help performance.
- Expert database analysis — PEM includes the Postgres Expert utility. Postgres Expert analyzes selected databases for best practice enforcement purposes. Areas such as general configuration and security setup are examined. Any deviations from recommended best practices are reported back to you, along with an explanation of each issue and expert help on what to do to make things right.
- Streaming replication monitor — You can monitor the Streaming Replication dashboard or use options on the PEM client to promote a replica node to the primary node.
- Secure client connectivity — PEM supports secure client connections through an encrypted SSH tunnel. The full-featured PEM client includes an SSH Tunnel definition dialog box that allows you to provide connection information for a secure connection.
- Wide platform support — PEM supports most major Linux and Windows platforms.

2.1 Probes

A *probe* is a scheduled task that retrieves information about the database objects that are being monitored by the PEM agent. PEM uses the collected information to build the graphs displayed on each homepage. The **Management>ManageProbes** tab allows you to modify the data collection schedule and the length of time that PEM retains information returned by a specific probe.

2.1.1 System probes

Unless otherwise noted, Postgres Enterprise Manager enables the following probes at the server, database, schema, extension (starting with version 8.4), or agent levels:

Name	Information monitored by probe	Level
Background Writer Statistics	Monitors information about the background writer. The information includes the number of: Timed checkpoints Requested checkpoints Buffers written (by checkpoint) Buffers written (by background writer) Background writer cycles Background buffers written Buffers allocated	Server
Barman Configuration	Information about the Barman tool global configuration.	Agent
Barman Information	Information about the Barman tool.	Agent
Barman Server	Information about the respective database server configuration monitored by Barman.	Agent
Barman Server Status	Information about the respective database server status monitored by Barman.	Agent
Barman Server Backup	Information about the backups of the respective database servers.	Agent
Barman Server WAL Status	Information about the Barman server WAL files.	Agent

continues on next page

Table 1 – continued from previous page

Name	Information monitored by probe	Level
Blocked Session Information	Information about blocked sessions.	Server
CPU Usage	CPU usage information.	Agent
Data and Log File Analysis	Information about log files. The information includes: The name of the log file The directory in which the log file resides	Server
Database Frozen XID	The frozen XID of each database.	Server
Database Size	Information about the size of the monitored databases. The information includes: The time the information was gathered The database name The database size (in MB).	Server
Database Statistics	Database statistics. The information includes: The number of backends The number of transactions committed The number of transactions rolled back The number of blocks read The number of blocks hit The number of rows returned The number of rows fetched The number of rows inserted The number of rows updated The number of rows deleted	Server
Disk Busy Info	Information about disk activity. Note: This probe isn't supported on Mac OS X, Solaris, or HP-UX.	Agent
Disk Space	Information about disk space usage. The information includes: The amount of disk space used The amount of disk space available	Agent
EDB Audit Configuration	The audit logging configuration of EDB Postgres Advanced Server.	Server
Failover Manager Cluster Info	Monitors a Failover Manager cluster, returning information about the cluster. This probe is enabled when a cluster name and path of the Failover Manager binary is provided on the Server Properties dialog box.	Server
Failover Manager Node Status	Monitors a Failover Manager cluster, returning details about each node within the cluster. This probe is enabled when a cluster name and path of the Failover Manager binary is provided on the Server Properties dialog box.	Server
Function Statistics	Monitors a database, retrieving information about functions. The information includes: Function names Argument types Return values	Database
Index Size	Monitors a database, retrieving information about indexes. The information includes: The name of the index The time the data was gathered The size of the index (in MB)	Database
Index Statistics	Index statistics. The information includes: The number of index scans The number of rows read The number of rows fetched The number of blocks read The number of blocks hit	Database

continues on next page

Table 1 – continued from previous page

Name	Information monitored by probe	Level
Installed Packages	The packages that are currently installed. The information gathered includes: The name of the installed package The version of the installed package The date and time that the probe executed	Agent
IO Analysis	Disk I/O information. The information includes: The number of blocks read The number of blocks written The date and time that the probe executed Note: This probe isn't supported on Mac OS X.	Agent
Load Average	CPU load averages. The information includes: The 1-minute load average The 5-minute load average The 15-minute load average Note: This probe isn't supported on Windows.	Agent
Lock Information	Lock information. The information includes: The database name The lock type The lock mode The process holding the lock	Server
Memory Usage	Information about system memory usage. The information includes: Total RAM in MB Free RAM in MB Total swap memory in MB Free swap memory in MB Shared system memory in MB (used by tuning wizard to tune the memory parameters for the database server) - On non-Windows systems, it is the <i>shmmmax</i> value and read from <i>/proc/sys/kernel/shmmmax</i> . - On Windows, it is same as total memory.	Agent
Network Statistics	Network statistics. The information includes: The interface IP address The number of packets sent The number of packets received The number of bytes sent The number of bytes received The link speed (in MB/second)	Agent
Number of Prepared Transactions	Stores the number of prepared transactions.	Server
Number of WAL Files	The number of WAL files.	Server
Object Catalog: Database	Monitors a list of databases and their properties. The information includes: The database name The database encoding type If the database allows user connections or system connections	Server
Object Catalog: Foreign Key	Monitors a list of foreign keys and their properties. The information includes: The name of the table that contains the foreign key The name of the table that the foreign key references The name of the database in which the table resides The name of the schema in which the table resides	Schema

continues on next page

Table 1 – continued from previous page

Name	Information monitored by probe	Level
Object Catalog: Function	Monitors a list of functions and their properties. The information includes: The name of the function The name of the schema in which the function resides The name of the database in which the function resides	Schema
Object Catalog: Index	Monitors a list of indexes and their properties. The information includes: The name of the index The name of the table that the index is associated with The name of the database in which the indexed table resides	Schema
Object Catalog: Schema	Monitors a list of schemas and their associated databases and servers.	Database
Object Catalog: Sequence	Monitors a list of sequences and their properties.	Schema
Object Catalog: Table	Monitors a list of table information. The information includes: The table name The name of the schema in which the table resides The name of the database in which the schema resides A Boolean indicator that shows whether the table has a primary key	Schema
Object Catalog: Tablespace	Monitors a list of tablespaces.	Server
Operating System Information	Operating system details and boot time.	Agent
Package Catalog	The packages that are currently available for installation. The information gathered includes: The package name The package version	Agent
PG HBA Conf	Authentication configuration information from the <i>pg_hba.conf</i> file.	Server
Server Information	Server information.	Server
Session Information	Session information. The information includes: The name of the session user The date and time that the session connected to the server The status of the session at the time that the information was gathered (idle, waiting, and so on) The client address and port number	Server
Settings	Monitors the values currently assigned to GUC variables.	Server
SQL Protect	Monitors a server, retrieving information about SQL injection attacks.	Server
Slony Replication	Lag data for clusters replicated using Slony.	Database
Streaming Replication	Monitors a cluster that is using streaming replication, retrieving information about: The sent Xlog location (in bytes) The write Xlog location (in bytes) The flush Xlog location (in bytes) The replay Xlog location (in bytes) The Xlog lag (in segments) The Xlog lag (in pages)	Server

continues on next page

Table 1 – continued from previous page

Name	Information monitored by probe	Level
Streaming Replication Lag Time	Monitors a cluster that's using streaming replication, retrieving lag information about: Replication lag time (in seconds) Current status of replication (running/paused)	Server
Streaming Replication Database Conflicts	Monitors a database that's using streaming replication, retrieving information about any conflicts that arise. This includes information about queries that were canceled due to the number of: Drop tablespace conflicts Lock timeout conflicts Old snapshot conflicts Pinned buffer conflicts Deadlock conflicts	Server
Table Bloat	Information about the current table bloat. The information includes: The name of the table The name of the schema in which the table resides The estimated number of pages The estimated number of wasted pages The estimated number of bytes per row	Database
Table Frozen XID	Monitors the frozen XID of each table.	Schema
Table Size	Information about table size. The information includes: Table size (in MB) Total index size (in MB) Total table size, with indexes and TOAST (in MB)	Database
Table Statistics	Table statistics. The information includes: The number of sequential scans The number of sequential scan rows The number of index scans The number of index scan rows The number of rows inserted The number of rows updated The number of rows deleted The number of live rows The number of dead rows The last VACUUM The last auto-vacuum The last ANALYZE The last auto-analyze The number of pages estimated by ANALYZE The number of rows estimated by ANALYZE	Database
Tablespace Size	A list of tablespaces and their sizes.	Server
User Information	A list of the current users. The stored information includes: The user name The user type (superuser or non-superuser) The server to which the user is connected	Server
WAL Archive Status	Status of the WAL archive. The stored information includes: The number of WAL archives done The number of WAL archives pending The last archive time The number of WAL archives failed The time of the last failure	Server
xDB Replication	Lag data for clusters replicated using xDB replication.	Database

2.1.2 BDR probes

To monitor the BDR group using BDR dashboards, you must enable the following probes. Configure these probes at extension level.

Note: Prior to version 8.4, all these probes are available at the server level.

Probe name	Execution Frequency			Enabled?		Data Retention	
	Default?	Minutes	Seconds	Default?	Probe Enable?	Default?	Days
BDR Conflict History Summary	No	5	0	No	Yes	Yes	7
BDR Global Locks	No	5	0	No	Yes	Yes	30
BDR Group Camo Details	No	5	0	No	Yes	Yes	30
BDR Group Raft Details	No	5	0	No	Yes	Yes	30
BDR Group Replication Slots Details	No	5	0	No	Yes	Yes	30
BDR Group Subscription Summary	No	5	0	No	Yes	Yes	30
BDR Group Versions Details	No	5	0	No	Yes	Yes	30
BDR Monitor Group Raft Status	No	5	0	No	Yes	Yes	30
BDR Monitor Group Versions	No	5	0	No	Yes	Yes	30
BDR Node Replication Rates	No	5	0	No	Yes	Yes	30
BDR Node Slots	No	5	0	No	Yes	Yes	30
BDR Node Summary	No	5	0	No	Yes	Yes	30
BDR Stat Relation	No	5	0	No	Yes	Yes	30
BDR Stat Subscription	No	5	0	No	Yes	Yes	30

Fig. 2: BDR Probes

The user with `bdr_superuser` can view information from all the following probes.

All the following probes works with BDR Enterprise Edition.

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Conflict History Summary	Information about row conflicts per conflict type. The stored information includes: The local time of the conflict The type of the conflict	Yes	Yes

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Global Locks	Information about global locks in a BDR group. The stored information includes: The name of the node where the global lock originated The PID of the process holding the lock The type of lock (DDL or DML) The name of the locked relation (for DML Locks) or keys (for advisory locks) The internal state of the lock acquisition process The list of backends waiting for the same global lock The time when the global lock acquire was initiated by origin node The time when the local node started trying to acquire the local lock The time acquire_stage last changed	Yes	Yes

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Group Camo Details	Information about camos in a BDR group. The stored information includes: The name of the node The node name for whom this node is partner The node name for whom this node is origin The connection status The readiness status The number of pending or unresolved camo transactions The LSN of last applied WAL log The LSN of last received WAL log	No	No

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Group Replication Slot Details	Information about replication slots in a BDR group. The stored information includes: The name of the BDR group The name of the origin node The name of the target node The slot name on the origin node used by this subscription The active status The state of the replication (catchup, streaming, disconnected,...) The approximate lag time for reported write The approximate lag time for reported flush The approximate lag time for reported replay The bytes difference between sent_lsn and current WAL write position The bytes difference between write_lsn and current WAL write position The bytes difference between flush_lsn and current WAL write position	No	Yes

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Group Subscription Summary	Information about the summary of subscriptions in the BDR group. The stored information includes: The name of the origin of the subscription The name of the target of the subscription The timestamp of the last replayed transaction The lag between now and time of last replayed transaction 	No	Yes
BDR Monitor Group Raft	The status and message of a cluster-wide raft check.	Yes	Yes
BDR Group Raft Details	Information about raft consensus status from all the nodes in a BDR group. The stored information includes: The name of the node The raft worker state on the node The node id of the RAFT_LEADER The raft election internal id The raft snapshot internal id	Yes	Yes
BDR Monitor Group Versions	The status and message of cluster-wide version check.	Yes	Yes

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Group Versions Details	Information about version details of the installed postgres, pglogical, bdr, and bdr edition for each node in the BDR group. The stored information includes: The name of the node The installed postgres version on the node The installed pglogical version on the node The version of the BDR on the node The BDR edition (Standard/Enterprise)	No	Yes

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Node Replication Rates	Information about outgoing replication activity from a given node. The stored information includes: The name of the target peer node The latest sent position The latest position reported as replayed The approximate lag time for reported replay The bytes difference between <code>replay_lsn</code> and current WAL write position on origin The human-readable bytes difference between <code>replay_lsn</code> and current WAL write position Approximate time required for the peer node to catch up to all the changes that are yet to be applied	Yes	No

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Node Slots	Information about the mapping of local BDR nodes to replication slots, their status, and replication progress. The stored information includes: The name of the slot The name of the target node The name of the BDR group The database name on the target node The PID of the process attached to the slot The catalog XID needed by the slot The IP address of the client connection The latest sent position The latest position reported as replayed The approximate lag time for reported replay The bytes difference between <code>replay_lsn</code> and current WAL write position The human-readable bytes difference between <code>replay_lsn</code> and current WAL write position	Yes	Yes

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Node Summary	Information about all the nodes in the BDR group. The stored information includes: The name of the node The name of the BDR group the node is part of The consistent state of the node in human-readable form The state which the node is trying to reach (during join or promotion) The name of subscribed repsets	Yes	Yes

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Workers	Information about workers in a BDR node. The stored information includes: The PID of the worker process The worker query start timestamp The worker state change timestamp The worker wait event type The worker wait event The worker state The worker role name The worker commit timestamp The worker local timestamp The name of the origin node The receive LSN The receive commit LSN The last exact replay LSN The last exact flush LSN The last exact replay timestamp The worker query	Yes	Yes

continues on next page

Table 2 – continued from previous page

Probe name	Information monitored by probe	<i>bdr_monitor</i> role required?	Works with BDR SE?
BDR Work Errors	Information about the work errors in BDR node. The stored information includes: The process id of the worker causing the error The name of the BDR group the node is part of The name of the origin node The name of the source node The name of the target node The name of the subscription The internal identifier of the role of this worker The name of the role of this worker The date and time of the error The age of the error The description of the error The context in which the error happened The remote relation id The subscription writer id The subscription writer name	Yes	Yes

2.1.3 Customizing probes

A probe is a scheduled task that returns a set of performance metrics about a specific monitored object. A probe retrieves statistics from a monitored server, database, operating system, or agent. You can use the **ManageProbes** tab to override the default configuration and customize the behavior of each probe.

To open the **ManageProbes** tab, select **Management>ManageProbes**.

The **ManageProbes** tab provides a set of icons that you can use to create and manage probes:

- Select **ManageCustomProbes** to open the **CustomProbes** tab and create or modify a custom probe.
- Select **CopyProbes** to open the Copy Probe dialog box and copy the probe configurations from the currently selected object to one or more monitored objects.

Properties SQL Statistics Dependencies Dependents Monitoring Manage Probes ✕

Description

Manage Custom Probes: PEM uses probes to retrieve statistics from a monitored server, database, operating system or agent. You can view, reconfigure, delete, or create your own custom probes.

Copy Probes: PEM allows copying of probes from any chosen object recursively down through the object hierarchy. Click on Copy Probes to quickly copy the displayed probe configuration to a selected target.

Quick Links

 Manage Custom Probes
  Copy Probes
  Help

Probes

Probe name	Execution Frequency			Enabled?		Data Retention	
	Default?	Minutes	Seconds	Default?	Probe Enable?	Default?	Days
Background Writer Statistics	☒	5	0	☒	☒	☒	180
Blocked Session Information	☒	5	0	☒	☒	☒	180
Data and Log File Analysis	☒	0	10	☒	☒	☒	180
Database Frozen XID	☒	720	0	☒	☒	☒	180
Database Size	☒	30	0	☒	☒	☒	180
Database Statistics	☒	30	0	☒	☒	☒	90
Failover Manager Cluster Info	☒	5	0	☒	☐	☒	7
Failover Manager Node Status	☒	5	0	☒	☐	☒	7
Lock Information	☒	5	0	☒	☒	☒	180
Number of Prepared Transactions	☒	5	0	☒	☒	☒	180
Number of WAL Files	☒	5	0	☒	☒	☒	180
Object Catalog: Database	☒	5	0	☒	☒	☒	180
Object Catalog: Tablespace	☒	5	0	☒	☒	☒	180
PG HBA Conf	☒	30	0	☒	☒	☒	180
Server Information	☒	5	0	☒	☒	☒	180
Server log Configuration	☒	0	10	☒	☒	☒	180
Session Information	☒	5	0	☒	☒	☒	180
Settings	☒	5	0	☒	☒	☒	180
Streaming Replication	☒	5	0	☒	☐	☒	180
Streaming Replication Database Conflicts	☒	5	0	☒	☐	☒	180
Streaming Replication Lag Time	☒	5	0	☒	☐	☒	180
Tablespace Size	☒	30	0	☒	☒	☒	180
User Information	☒	30	0	☒	☒	☒	180
WAL Archive Status	☒	30	0	☒	☐	☒	180

Fig. 3: The Manage Probes tab

Note: Copy Probe isn't supported for the extension-level probes.

A probe monitors a unique set of metrics for each object type (server, database, database object, or agent). Select the name of an object in the tree to review the probes for that object.

To modify the properties associated with a probe, select the name of a probe and customize the settings that are displayed in the Probes table:

- Move the **Default** switch in the Execution Frequency columns to **N** to enable the **Minutes** and **Seconds** selectors, and specify a custom value for the length of time between executions of the probe.
- Move the **Default** switch in the Enabled? column to **No** to change the state of the probe and indicate if the probe is active or not.

Note: If data from a disabled probe is used in a chart, the chart displays an information icon in the upper-left corner that allows you to enable the probe by clicking the provided link.

- Move the **Default** switch in the Data Retention column to **No** to enable the **Day(s)** field and specify the number of days that information gathered by the probe is stored on the PEM server.

The **ManageProbes** tab might display information about probes that you can't modify from the current node. If you can't modify a probe from the current dialog box, the switches are disabled. Generally, you can modify a disabled probe from a node that's higher in the hierarchy of the PEM client tree control. Select another object in the tree control to modify the probes that are displayed or enabled in the **ManageProbes** tab.

Creating a custom probe

You can use the **PEMCustomProbes** tab to create a new probe or modify an existing custom probe. To open the **CustomProbes** tab, from the **ManageProbes** tab, select **ManageCustomProbes**.

The screenshot shows the 'Probes' tab in the 'ManageProbes' dialog. The 'Show System Probes?' switch is set to 'No'. The table below lists the probes:

	Probe name	Collection method	Target type	Execution frequency		Probe enabled?	Data retention
				Minutes	Seconds		
☒	Top_Five_Large_Tables	SQL	Table	1	0	Yes	1

Fig. 4: The Custom Probes dialog

Use the **ShowSystemProbes?** switch to show or hide the system probes on the **CustomProbes** tab.

You can use the **CustomProbes** tab to create a probe or modify an existing one. To create a probe:

1. Select **Add** in the upper-right corner of the tab.

2. Provide a name for the new probe in the Probe Name column.
3. Select **Edit** (located to the left of the probe name) to review or add the probe definition.

The screenshot shows the 'Probes' configuration window. At the top, there's a table with columns: Probe name, Collection method, Target type, Execution frequency (Minutes, Seconds), Probe enabled?, and Data retention. Below the table is the 'General' tab configuration form. The 'Probe name' field has a red error message: 'Please specify Probe name'. The 'Collection method' is set to 'SQL'. The 'Target type' is set to 'Server'. The 'Execution frequency' is set to 5 minutes and 0 seconds. The 'Probe enabled?' checkbox is checked. The 'Data retention' is set to 1. The 'Discard from history?' checkbox is unchecked. The 'Platform' is set to 'linux'.

Fig. 5: Defining a custom probe – the General tab

Use the fields on the **General** tab to modify the definition of an existing probe or to specify the properties of a new probe:

- Use the **ProbeName** field to provide a name for a new probe.
- Use the **Collectionmethod** field to specify the probe type. From the list, select:

```
- SQL &mdash; The probe gathers information by way of a SQL statement. - WMI &mdash;
↪ The probe gathers information by way of a Windows Management Instrumentation extension.
↪ - Batch &mdash; The probe uses a command-script or shell-script to gather
↪ information.
```

Before creating a batch probe on a Linux system, you must modify the `agent.cfg` file, setting the `allow_batch_probes` parameter to `true`. Then restart the PEM agent. The `agent.cfg` file is located in one of the following directories:

```
- If you installed PEM using graphical installer: `/opt/edb/pem/agent/etc/agent.cfg` -
↪ If you installed PEM using RPM: `/usr/edb/pem/agent/etc/agent.cfg`
```

On 64-bit Windows systems, agent settings are stored in the registry. Before creating a batch probe, modify the registry entry for the `AllowBatchProbes` registry entry and restart the PEM agent. PEM registry entries are located in `HKEY_LOCAL_MACHINE\Software\Wow6432Node\EnterpriseDB\PEM\agent`.

Batch probes are platform specific. If you specify a collection method of **Batch**, you must specify a platform type in the **Platform** field.

To invoke a script on a Linux system, you must modify the entry for the `batch_script_user` parameter of the `agent.cfg` file and specify the user to use to run the script. You can either specify a non-root user or root for this parameter. If you don't specify a user or the specified user doesn't exist, then the script doesn't run. Restart the agent after modifying the file.

To invoke a script on a Windows system, set the registry entry for `AllowBatchJobSteps` to `true` and restart the PEM agent.

- Use the **TargetType** list to select the object type for the probe to monitor. **TargetType** is disabled if **Collection-method** is WMI.
- Use the **Minutes** and **Seconds** selectors to specify how often the probe collects data.
- Use the **ProbeEnable?** switch to specify if the probe is enabled by default. Specify **Yes** to enable the probe by default or **No** to disable the probe by default.

Note: If data from a disabled probe is used in a chart, the chart displays an information icon in the upper-left corner that allows you to enable the probe by clicking the provided link.

- Use the **DataRetention** field to specify the number of days that gathered information is retained in the probe's history table.
- Use the switch next to **Discardfromhistory** to specify if the server creates a history table for the probe. Select **Yes** to discard probe history or **No** to retain the probe history in a table.
- Use the **Platform** list to specify the type of platform that the probe monitors. This field is enabled only when the **Collection** method is **Batch**.

Use the **Columns** tab to define the columns in which to store the probe data. To define a new column:

1. Navigate to the **Columns** tab and select **Add** in the upper-right corner.
2. Provide a column name in the **Name** field.
3. Select **Edit** (to the left of the new column name) to provide information about the column:
 - Provide a descriptive name for the column in the **Name** field.
 - The **InternalName** field isn't enabled for custom probes.
 - Use the **ColumnType** list to specify if the column is a key column (a primary key) or a non-key column. Non-key columns are generally metric items (values that can be graphed).
 - Use the **DataType** list to specify the type of data to store in the column.
 - Use the **Unit** field to specify the unit of measure that applies to the metric stored in the column. This unit is displayed on the y-axis of a custom chart or a Capacity Manager chart. This is an optional field.
 - Use the **Graphable** switch to specify if the defined metric can be graphed and to make the probe accessible from the Capacity Manager or Manage Charts dialog boxes.
 - Use the **IsPIT** switch to specify whether to store the metric by point-in-time.

Point-in-time metrics change (increase or decrease) at any given point of time. For example, database size is a point-in-time metric. At any given point-in-time, the size of the database is fluctuating. Metrics that aren't point-in-time (also referred to as cumulative metrics) are those whose size always increases over time. For example, Blocks Read and Tuples Read are cumulative metrics. The value stays the same or increases.

Probes

Show System Probes? ☐ Import

Probe name: Collection method: SQL Target type: Server Execution frequency: Minutes: 5 Seconds: 0 Probe enabled?: ☒ Data retention: 1

General Columns Code Alternate Code

Name	Internal name	Column type	Data type	Unit	Graphable?	Is PIT?	Calculate PIT?
		Non key	numeric		☒	☐	

General

Name ▲

Internal name

Column type: Non key
Use the Column Type drop-down to specify if the column is a Key column (a primary key) or a Non key column. Non-key columns are generally metric items (values that can be graphed).

Data type: numeric
Use the Data type drop-down to specify the type of data that will be stored in the column.

Unit
Unit field to specify the unit of measure that applies to the metric stored in the column. This unit is displayed on the Y-Axis of a custom chart or a Capacity Manager chart. This is an optional field.

Graphable? ☒
Set the Graphable switch to Yes to specify that the defined metric may be graphed, and that the probe should be accessible from the Capacity Manager or Manage Charts dialogs.

Is PIT? ☐
Set the Is PIT switch to Yes to specify that the metric is stored by point-in-time (by default). 'Point-in-time' metrics are those metrics that change (increase or decrease) at any given point of time. For example, database size is a point-in-time metric; at any given point-in-time, the size of the database is fluctuating. Metrics that are not point-in-time (also referred to as cumulative metrics) are metrics whose size always increases over time. For example, Blocks Read and Tuples Read are cumulative metrics; the value stays the same or increases.

Calculate PIT?
Set the Calculate PIT switch to Yes to specify that the server should calculate a point-in-time value for the metric data. Calculate PIT is disabled if Is PIT is Yes.

▲ Please specify column name

▲ Please specify Probe name

Fig. 6: The Columns tab of the Custom Probes dialog

- Use the **CalculatePIT** switch to specify for the server to calculate a point-in-time value for the metric data. **CalculatePIT** is disabled if **IsPIT** is **Yes**.

PEM allows you to store point-in time-values of cumulative metrics as well. PEM subtracts the last collected value of a cumulative metric from the current value and stores the difference as a point-in-time value.

Probes

Show System Probes? ☐ Import

Probe name: Collection method: SQL Target type: Server Execution frequency: Minutes: 5 Seconds: 0 Probe enabled?: ☒ Data retention: 1

General Columns Code Alternate Code

Code ▲

Use the Code tab to specify SQL SELECT statements invoked by the probe on the Code tab. The column names returned by the query must match the Internal name specified on the Columns tab. The number of columns returned by the query, as well as the column name, datatype, etc. must match the information specified on the Columns tab.

▲ Please specify Probe name

Fig. 7: The Code tab of the Custom Probes dialog

Use the **Code** tab to specify the default code for the propbe to execute:

- If the probe is a SQL probe, you must specify the **SQLSELECT** statement invoked by the probe on the **Code** tab. The column names returned by the query must match the **InternalName** specified on the **Columns** tab. The number of columns returned by the query, as well as the column name, data type, and so on, must match the information specified on the **Columns** tab.

- If the probe is a batch probe, you must specify the shell or .bat script to invoke when the probe runs. The output of the script is as follows:

- The first line must contain the names of the columns provided on the ****Columns**** tab. Each column name must be separated by a tab (t) character. From the second line onwards, each line must contain the data for each column, separated by a tab character.
- If a specified column is defined as a key column, make sure that the script doesn't produce duplicate data for that column across lines of output. The number of columns specified in the ****Columns**** tab and their names, data type, and so on must match with the output of the script output.

- If the probe is a WMI probe, you must specify the WMI query as a **SELECTWMI** query. The column name referenced in the **SELECT** statement must be same as the name of the corresponding column specified on the **Column** tab. The column names returned by the query must match the **InternalName** specified on the **Columns** tab. The number of columns returned by the query, as well as the column name, data type, and so on, must match the information specified on the **Columns** tab.

Fig. 8: The Alternate Code tab of the Custom Probes dialog

Use the **AlternateCode** tab to provide code to invoke if the probe fires on a specific version of the server. To provide version-specific code, move the **Appliestoanyserversversion?** switch to **No** and select **Add** . Then, use the **DatabaseVersion(s)** list to select a version, and select **Edit** (to the left of the version name) to provide the code to execute when the probe fires.

If you select a database version and leave the Probe Code column blank, PEM invokes the code specified on the **Code** tab when the probe executes on a server that matches that version.

When you've finished defining the probe, select **Save** (in the corner of the **CustomProbes** tab) to save the definition and make the probe data available for use on custom charts and graphs.

Exporting or importing a probe

To export the probe, select any probes from the **ManageCustomProbes** tab, and then select **Export** in the upper-right corner of the table. To generate the JSON file, select **SaveFile** and then select **OK**.

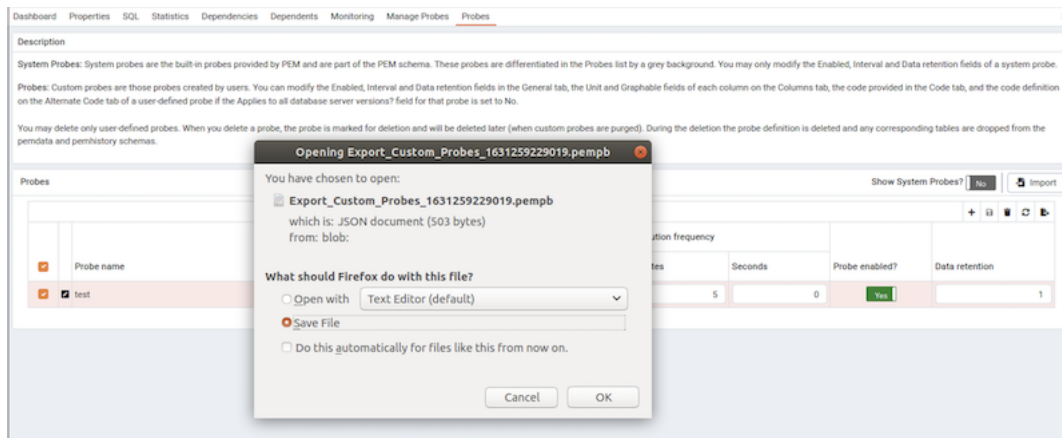


Fig. 9: Custom Probes - Export

To import the probe, on the **ManageCustomProbes** tab, select **Import** in the upper-right corner of the table.

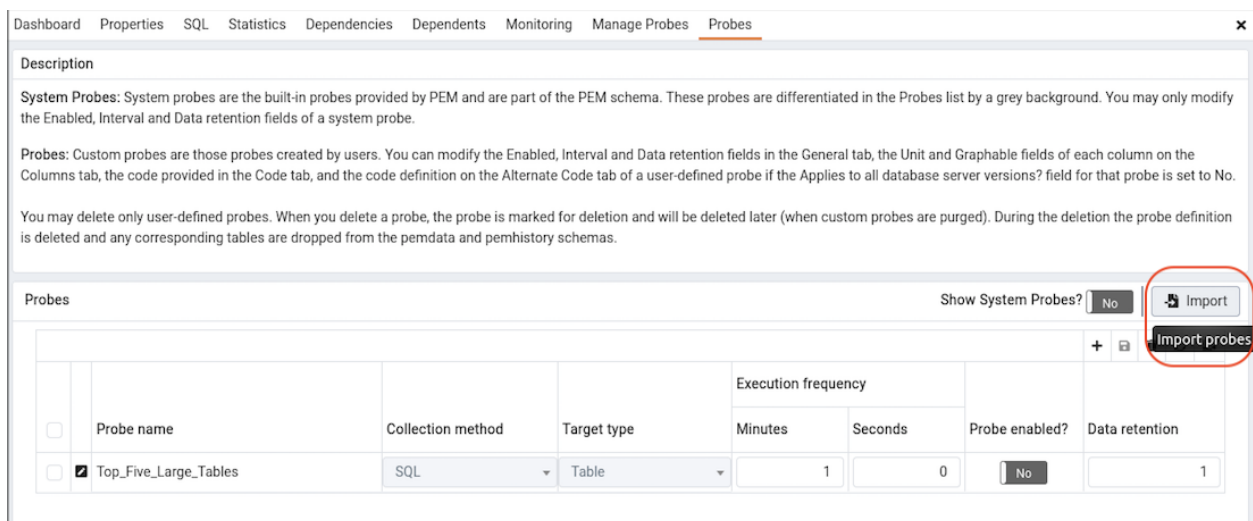


Fig. 10: Custom Probes Import

Select **Browse** to select the JSON file with the probe code to import, and then select **Import**.

After selecting the file to import, you can select the **skipexisting** check box. This option skips the probe if it already exists and displays the following message:

If the check box isn't selected and the probe already exists, then it doesn't import the probe. The following error appears:

Note: **Import** can't overwrite the existing probe as it might be configured to retain historical data as per the configured retention policy.

If the check box isn't selected and the probe doesn't exist but the corresponding table in the **pem** schema does, then it imports the probe successfully using the same table.

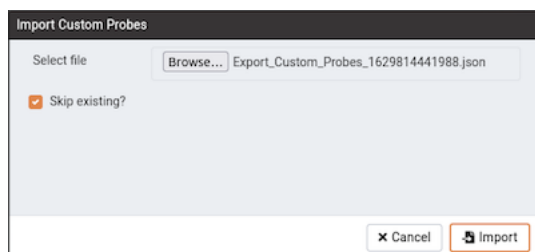


Fig. 11: Custom Probes - Import Browse

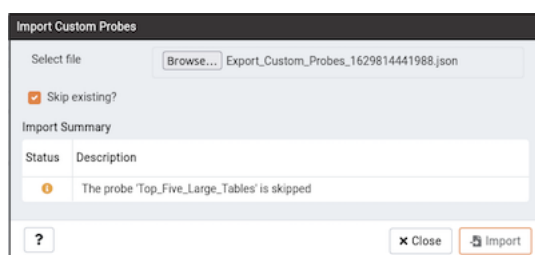


Fig. 12: Custom Probes - Import Skipped Message

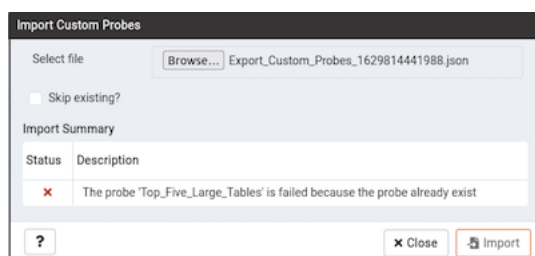


Fig. 13: Custom Probes - Import Error

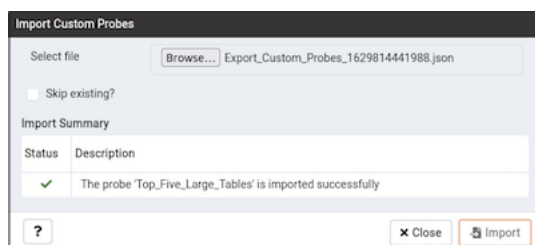


Fig. 14: Custom Probes Import Successful

Note: It's possible that the probe is deleted and not listed on **Manage Custom Probe** tab and for the table holding the data of that probe to exist in the *pem* schema.

Deleting a probe

You can delete only custom probes. To delete a probe, select the probe name in the probes table, and select **Delete** (located to the upper-right corner of the table). The probe history persists for the length of time specified in the **HistoryRetention** field in the probe definition. During the deletion, the probe definition is deleted and any corresponding tables are dropped from the *pemdata* and *pemhistory* schemas.

System probes are the built-in probes provided by PEM and are part of the PEM schema. You can only modify system probes. If you attempt to delete a system probe, PEM reports an error.



Fig. 15: Attempting to delete a system probe

Copying a probe

You can use the Copy Probe Configuration dialog box to copy probe definitions from one monitored object to one or more monitored objects of the same type. To open the Copy Probe Configuration dialog box:

1. In the PEM client tree, select the object from which you are copying probes.
2. Select **Management>ManageProbes**.
3. From the **ManageProbes** tab, select **CopyProbe**.

The dialog box copies the probe definitions from the object from which you opened the Copy Probe Configuration dialog box to the locations selected on the tree control.

If you specify a parent node in the **CopyProbeConfiguration** tree, PEM copies the probe configurations to each object of the same type that resides under that node in the tree. For example, to copy the probe definitions from one schema to all schemas that reside in a database, select only the parent database of the target schemas. A red warning symbol is displayed to the left of the name of a listed target object if that object is the source of the probe that's being copied.

After you select the target object or objects, select **ConfigureProbes** to copy the probe definitions to the location selected in the dialog box.

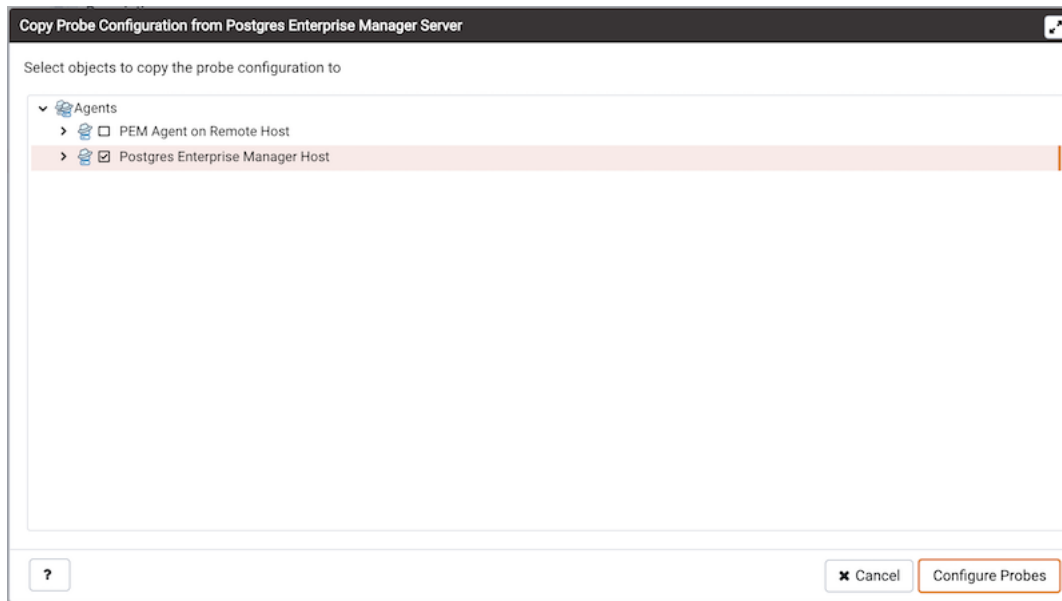


Fig. 16: The Copy Probe Configuration tree

2.1.4 PEM probes – reference

A probe is a scheduled task that retrieves information about the database objects that are being monitored by the PEM agent. PEM uses the collected information to build the graphs displayed on each dashboard. The **ManageProbes** tab (accessed from the **Management** menu) allows you to modify the data collection schedule and the length of time that PEM retains information returned by a specific probe.

Name	Information monitored	Level
Background Writer Statistics	Information about the background writer. The information includes: The number of timed checkpoints The number of requested checkpoints The number of buffers written (by checkpoint) The number of buffers written (by background writer) The number of background writer cycles The number of background buffers written The number of buffers allocated	Server
Blocked Session Information	Information about blocked sessions.	Server
CPU Usage	CPU Usage information.	Agent
Data and Log File Analysis	Information about log files. The information includes: The name of the log file The directory in which the log file resides	Server
Database Frozen XID	The frozen XID of each database.	Server
Database Size	Information about the size of the monitored databases. The information includes: The time the information was gathered The database name The database size (in MB).	Server

continues on next page

Table 3 – continued from previous page

Name	Information monitored	Level
Database Statistics	Database statistics. The information includes the number of: Backends Transactions committed Transactions rolled back Blocks read Blocks hit Rows returned Rows fetched Rows inserted Rows updated Rows deleted	Server
Disk Busy Info	Information about disk activity. Note: This probe isn't supported on Mac OS X, Solaris, or HP-UX.	Agent
Disk Space	Information about disk space usage. The information includes: The amount of disk space used The amount of disk space available	Agent
EDB Audit Configuration	The audit logging configuration of EDB Postgres Advanced Server.	Server
Failover Manager Cluster Info	Monitors a Failover Manager cluster, returning information about the cluster. This probe is enabled when you provide a cluster name and path of the Failover Manager binary in the Server Properties dialog box.	Server
Failover Manager Node Status	Monitors a Failover Manager cluster, returning detailed about each node in the cluster. This probe is enabled when you provide a cluster name and path of the Failover Manager binary in the Server Properties dialog box.	Server
Function Statistics	Monitors a database, retrieving information about functions. The information includes: Function names Argument types Return values	Database
Index Size	Monitors a database, retrieving information about indexes. The information includes: The name of the index The time the data was gathered The size of the index (in MB)	Database
Index Statistics	Index statistics. The information includes the number of: Index scans Rows read Rows fetched Blocks read Blocks hit	Database
Installed Packages	The packages that are currently installed. The information gathered includes: The name of the installed package The version of the installed package The date and time that the probe executed	Agent
IO Analysis	Monitors disk I/O information. The information includes: The number of blocks read The number of blocks written The date and time that the probe executed Note: This probe isn't supported on Mac OS X.	Agent
Load Average	CPU load averages. The information includes: The 1-minute load average The 5-minute load average The 15-minute load average Note: This probe isn't supported on Windows.	Agent
Lock Information	Monitors lock information. The information includes: The database name The lock type The lock mode The process holding the lock	Server

continues on next page

Table 3 – continued from previous page

Name	Information monitored	Level
Memory Usage	Information about system memory usage. The information includes: Total RAM in MB Free RAM in MB Total swap memory in MB Free swap memory in MB Shared system memory in MB (used by tuning wizard to tune the memory parameters for the database server) - On non-Windows system, it's the <i>shmmmax</i> value and read from <i>/proc/sys/kernel/shmmmax</i> . - On Windows, it's same as total memory.	Agent
Network Statistics	Network statistics. The information includes: The interface IP address The number of packets sent The number of packets received The number of bytes sent The number of bytes received The link speed (in MB/second)	Agent
Number of Prepared Transactions	Stores the number of prepared transactions.	Server
Number of WAL Files	The number of WAL files.	Server
Object Catalog: Database	Monitors a list of databases and their properties. The information includes: The database name The database encoding type If the database allows user connections or system connections	Server
Object Catalog: Foreign Key	Monitors a list of foreign keys and their properties. The information includes: The name of the table that contains the foreign key The name of the table that the foreign key references The name of the database in which the table resides The name of the schema in which the table resides	Schema
Object Catalog: Function	Monitors a list of functions and their properties. The information includes: The name of the function The name of the schema in which the function resides The name of the database in which the function resides	Schema
Object Catalog: Index	Monitors a list of indexes and their properties. The information includes: The name of the index The name of the table that the index is associated with The name of the database in which the indexed table resides	Schema
Object Catalog: Schema	Monitors a list of schemas and their associated databases and servers.	Database
Object Catalog: Sequence	Monitors a list of sequences and their properties.	Schema
Object Catalog: Table	Monitors a list of table information. The information includes: The table name The name of the schema in which the table resides The name of the database in which the schema resides A Boolean indicator that shows whether the table has a primary key	Schema
Object Catalog: Tablespace	Monitors a list of tablespaces.	Server

continues on next page

Table 3 – continued from previous page

Name	Information monitored	Level
Operating System Information	The operating system details and boot time.	Agent
Package Catalog	The packages that are currently available for installation. The information gathered includes: The package name The package version	Agent
PG HBA Conf	Authentication configuration information from the <i>pg_hba.conf</i> file.	Server
Server Information	Server information.	Server
Session Information	Session information. The information includes: The name of the session user The date and time that the session connected to the server The status of the session at the time that the information was gathered (idle, waiting, and so on) The client address and port number	Server
Settings	The values currently assigned to GUC variables.	Server
SQL Protect	Monitors a server, retrieving information about SQL injection attacks.	Server
Slony Replication	Lag data for clusters replicated using Slony.	Database
Streaming Replication	Monitors a cluster that's using streaming replication, retrieving information about: The sent Xlog location (in bytes) The write Xlog location (in bytes) The flush Xlog location (in bytes) The replay Xlog location (in bytes) The Xlog lag (in segments) The Xlog lag (in pages)	Server
Streaming Replication Lag Time	Monitors a cluster that's using streaming replication, retrieving lag information about: Replication lag time (in seconds) Current status of replication (running/paused)	Server
Streaming Replication Database Conflicts	Monitors a database that's using streaming replication, retrieving information about any conflicts that arise. This includes information about queries that were canceled due to the number of: Drop tablespace conflicts Lock timeout conflicts Old snapshot conflicts Pinned buffer conflicts Deadlock conflicts	Server
Table Bloat	Information about the current table bloat. The information includes: The name of the table The name of the schema in which the table resides The estimated number of pages The estimated number of wasted pages The estimated number of bytes per row	Database
Table Frozen XID	The frozen XID of each table.	Schema
Table Size	Information about table size. The information includes: Table size (in MB) Total index size (in MB) Total table size, with indexes and TOAST (in MB)	Database

continues on next page

Table 3 – continued from previous page

Name	Information monitored	Level
Table Statistics	Table statistics. The information includes: The number of sequential scans The number of sequential scan rows The number of index scans The number of index scan rows The number of rows inserted The number of rows updated The number of rows deleted The number of live rows The number of dead rows The last VACUUM The last auto-vacuum The last ANALYZE The last auto-analyze The number of pages estimated by ANALYZE The number of rows estimated by ANALYZE	Database
Tablespace Size	A list of tablespaces and their sizes.	Server
User Information	A list of the current users. The stored information includes: The user name The user type (superuser or non-superuser) The server to which the user is connected	Server
WAL Archive Status	The status of the WAL archive. The stored information includes: The number of WAL archives done The number of WAL archives pending The last archive time The number of WAL archives failed The time of the last failure	Server
xDB Replication	Monitors lag data for clusters replicated using xDB replication.	Database

2.2 Alerts

PEM continually monitors registered servers. It compares performance metrics against predefined and user-specified thresholds that specify good or acceptable performance for each statistic. Any deviation from an acceptable threshold value triggers an alert. An alert is a system-defined or user-defined set of conditions that PEM compares to the system statistics. Alerts tell you about conditions on registered servers that require your attention.

2.2.1 Viewing the alerts via Global dashboard

When your system statistics deviate from the boundaries specified for that statistic, the alert triggers. The alert displays a high (red), low (yellow), or medium (orange) severity warning in the left-most column of the Alert Status table on the Global Overview dashboard.

The PEM server includes a number of predefined alerts that are actively monitoring your servers. The alert definition might make details available about the cause of the alert. Select the down arrow to the right of the severity warning to open a dialog box that has details about the condition that triggered the alert.

PEM also provides an interface that lets you create customized alerts. Each alert uses metrics defined on an alert template. An alert template defines how the server evaluates the statistics for a resource or metric. The PEM server includes predefined alert templates, and you can create custom alert templates.

Alerts Status									
	Alarm Type	Object Description	Alert Name	Value	Database	Schema	Package	Object	Alerting Since
▶	● High	EDB Postgres Advanced Server 11	Database size in server	113 MB					2020-04-22 11:50:00
▶	● High	EDB Postgres Advanced Server 11	Last Vacuum	Never ran					2020-04-21 21:26:54
▶	● High	EDB Postgres Advanced Server 11	Last AutoVacuum	140.21 hrs					2020-04-22 12:04:05
▶	● High	EPAS_12	Table size in server	410 MB					2020-04-09 15:53:51
▶	● Medium	EPAS_12	Last Vacuum	5.18 hrs					2020-04-27 20:47:50
▶	● High	EPAS_12	Database size in server	455 MB					2020-04-09 15:52:50
▶	● Medium	EPAS_12	Last AutoVacuum	5.16 hrs					2020-04-27 20:47:50
▶	● High	N/A	Alert Errors	3					2020-01-21 14:26:04
▶	● High	PGSQL12_Centos7_1	Server Down	1					2020-04-27 20:48:50
▶	● High	PGSQL12_Centos7_1	Last Vacuum	Never ran					2020-04-03 14:58:57
▶	● High	PGSQL12_Centos7_1	Last AutoVacuum	Never ran					2020-04-03 14:58:57
▶	● High	Postgres Enterprise Manager Server	Largest index by table-size percentage	100 %					2020-04-21 22:07:52
▶	● High	Postgres Enterprise Manager Server	Database size in server	2.072265625 GB					2020-02-05 18:26:49
▶	● High	Postgres Enterprise Manager Server	Table size in server	1.9814453125 GB					2020-02-20 11:29:45
▶	● Medium	Postgres Enterprise Manager Server	Connections in idle state	12					2020-04-27 16:20:32
▶	● Medium	Postgres Enterprise Manager Server	Last Vacuum	4.99 hrs					2020-04-27 20:47:50

Fig. 17: The Alert Status table

Alert Details (Auto-refresh paused whilst rows are expanded. ⓘ)

	Ack'd	Alert Type	Name	Value	Agent	Server	Database	Schema	Package	Object	Alerting Since
▶	☐	● High	Table size in server	1.9814453125 GB		Postgres Enterprise Manager Server					2020-02-20 11:29:45

General

Parameters

Table name	Schema name	Database name	Total table size(MB)
table_statistics	pemhistory	pem	1087
server_logs	pemdata	pem	263
index_statistics	pemhistory	pem	237
session_info	pemhistory	pem	137
lock_info	pemhistory	pem	88

Fig. 18: Alert details

2.2.2 Viewing the alerts via Alerts dashboard

Use the **Dashboards** menu (on the **Monitoring** tab) to open the Alerts dashboard. The Alerts dashboard shows a summary of the active alerts and the status of each alert.

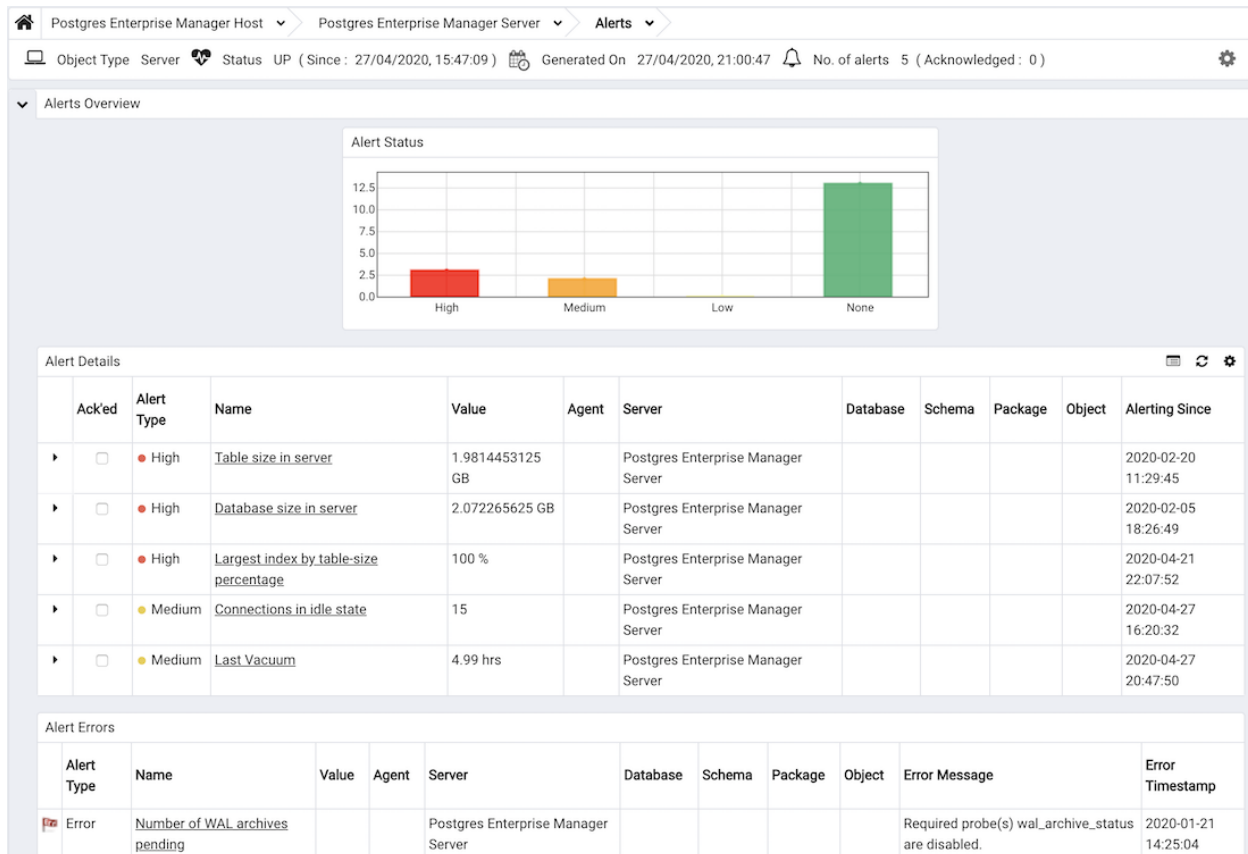


Fig. 19: The Alerts Dashboard

The Alerts dashboard header shows the date and time that the dashboard was last updated and the number of current alerts.

The Alerts Overview section shows a visual representation of the active alerts and a count of the current high, low, and medium alerts. The vertical bar on the left of the graph provides the count of the alerts displayed in each column. Hover over a bar to display the alert count for the selected alert severity in the upper-right corner of the graph.

The Alert Details table provides a list of the alerts that are currently triggered. The entries appear in order from high severity to low severity. Each entry includes information that lets you identify the alert and recognize the condition that triggered the alert. Select an alert to review detailed information about the alert definition.

The Alert Errors table shows configuration-related errors, such as accidentally disabling a required probe or improperly configuring an alert parameter. You can use the information provided in the Error Message column to identify and resolve the conflict that's causing the error.

2.2.3 Customizing the Alerts dashboard

You can customize tables and charts that appear on the Alerts dashboard. To customize a table or chart, select **Settings** in the upper-right corner.

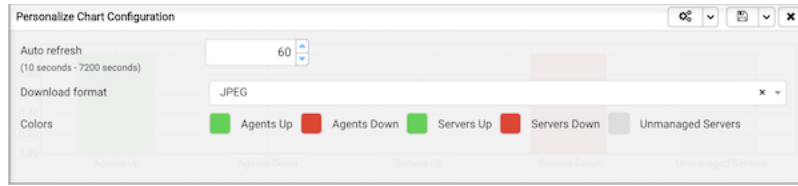


Fig. 20: Customizing a chart

Use fields on the Personalize Chart Configuration dialog box to provide your display preferences:

- Use the **AutoRefresh** field to specify the number of seconds between updates of the data displayed in the table or chart.
- Use the **Downloadas** field to indicate whether to download a chart as a JPEG image or as a PNG image.
- Use **Coloursselectors** to specify the colors to use on a chart.
- Set the **ShowAcknowledgedAlerts** switch to **Yes** if you want the table to display alerts that you acknowledged with a check box in the Ack'ed column. Set it to **No** to hide any acknowledged alerts. Acknowledged alerts are purged from the table content only when the time specified in the alert definition passes.

To save your customizations, select **Save** (a checkmark) in the upper-right corner. To delete any previous changes and revert to the default values, select **Delete**. Use the **Save** and **Delete** menus to specify whether to apply your preferences to all dashboards or to a selected server or database.

Managing alerts

Use the PEM client's **ManageAlerts** tab to define, copy, or manage alerts. To open the **ManageAlerts** tab, select **Management>ManageAlerts**.

Use the Quick Links toolbar to open dialog boxes and tabs for managing alerts:

- Select **CopyAlerts** to open the Copy Alert Configuration dialog box and copy an alert definition.
- Select **AlertTemplates** to open the **AlertTemplate** tab and modify or create an alert template.
- Select **EmailGroups** to open the **EmailGroups** tab and modify or create an email group.
- Select **Webhooks** to open the **Webhooks** tab and create or manage the webhooks endpoints.
- Select **ServerConfigurations** to open the Server Configuration dialog box and review or modify server configuration settings.
- Select **Help** to open the PEM online help in a new tab.

Use the table in the Alerts section of the **ManageAlerts** tab to create new alerts or manage existing alerts.

Quick Links

Copy Alerts

Alert Templates

Email Groups

Webhooks

Server Configuration

Help

Alerts

Manage Alerts

		Name	Auto created?	Template	Enable?	Interval		History retention		
						Default?	Minutes	Default?	Days	
☒		Average table bloat in server	Yes	Average table bloat in server	Yes	Yes		1	Yes	30
☒		Connections in idle-in-transaction state	Yes	Connections in idle-in-transactio...	Yes	Yes		1	Yes	30
☒		Connections in idle-in-transaction state, a...	Yes	Connections in idle-in-transactio...	Yes	Yes		1	Yes	30
☒		Connections in idle state	Yes	Connections in idle state	Yes	Yes		1	Yes	30
☒		Database size in server	Yes	Database size in server	Yes	Yes		1	Yes	30
☒		Highest table bloat in server	Yes	Highest table bloat in server	Yes	Yes		1	Yes	30
☒		Largest index by table-size percentage	Yes	Largest index by table-size perce...	Yes	Yes		1	Yes	30
☒		Last AutoVacuum	Yes	Last AutoVacuum	Yes	Yes		1	Yes	30
☒		Last Vacuum	Yes	Last Vacuum	Yes	Yes		1	Yes	30

Fig. 21: The Manage Alerts tab

Alert templates

An alert template is a prototype that defines the properties of an alert. An alert instructs the server to compare the current state of the monitored object to a threshold specified in the alert template to determine if a situation requires administrative attention.

You can use the **AlertTemplates** tab to define a custom alert template or view the definitions of existing alert templates. To open the **AlertTemplates** tab, select **Management>ManageAlerts**. From the **ManageAlerts** tab, on the Quick Links toolbar, select **AlertTemplates**.

Dashboard Properties SQL Statistics Dependencies Dependents Monitoring Manage Alerts Alert Templates											
Description											
Alert Template: An alert template is a prototype that defines the properties of a custom alert. An alert instructs the server to compare the current state of the monitored object to a threshold (of the type specified in the template that is associated with the alert) to determine if a situation exists that requires administrative attention. The Alert Templates tab provides an interface that allows you to define a custom alert template or view and modify the definitions of existing alert templates.											
Alert Templates											
										Show System Template: None	Import
	Template name	Description	Target type	Applies to server	Check frequency (minutes)						
☒	Test_alert_template	Testing	Server	ALL	1						

Fig. 22: The Alert Templates tab

Use the **ShowSystemTemplate** list to filter the alert templates that are displayed in the Alert Templates table. From the list, select a level of the PEM hierarchy to view all of the templates for that level.

2.2.4 Defining a new alert template

To define a new alert template, from the **ShowSystemTemplate** list, select **None** . Then click the plus sign (+) in the upper-right corner of the alert template table. The alert template editor opens.

The screenshot shows the 'General' tab of the alert template editor. It contains the following fields and controls:

- Template name:** A text input field with a red warning triangle icon.
- Description:** A text input field with a red warning triangle icon.
- Target type:** A dropdown menu currently set to 'Server'. Below it is a tooltip: 'Use the Target type field to select the type of object that will be the focus of the alert.'
- Applies to server:** A dropdown menu currently set to 'ALL'. Below it is a tooltip: 'Use the Applies to server field to specify the server to which the alert will be applied; specify a single server type or ALL.'
- History retention:** A numeric input field set to '30'. Below it is a tooltip: 'Use the History retention field to specify the number of days that the result of the alert execution will be stored on the PEM server.'
- Threshold unit:** A text input field. Below it is a tooltip: 'Use the Threshold unit field to specify the unit type of the threshold value.'
- Auto create:** A section containing an 'Auto create?' label and a slider control currently set to 'No'. Below the slider is a tooltip: 'Use the Auto create field to automatically create alerts using this template. Please provide default threshold values. If the value is "Yes" then all the added parameters in Parameters tab will be removed as we do not support parametrised auto alerts.'
- Operator:** A dropdown menu set to '>' and three buttons labeled 'Low', 'Med', and 'High'.
- Check frequency (minutes):** A numeric input field set to '1'. Below it is a tooltip: 'Use the Check frequency field to specify the number of minutes between alert executions. This value specifies how often the server will invoke the SQL code specified on the SQL tab and compare the result to the threshold value specified in the alert definition.'

Fig. 23: The General tab

Use fields on the **General** tab to specify general information about the template:

- Use the **Templatenam**e field to specify a name for the new alert template.
- Use the **Description** field to provide a description of the alert template.
- Use the **Targettype** list to select the type of object that is the focus of the alert.
- Use the **Appliesto**server list to specify the server type (EDB Postgres Advanced Server or PostgreSQL) to which to apply the alert. You can specify a single server type or **ALL** .
- Use the **Historyretention** field to specify the number of days to store the result of the alert execution on the PEM server.
- Use the **Thresholdunit** field to specify the unit type of the threshold value.
- Use fields in the **Autocreate** box to specify for PEM to use the template to generate an automatic alert. If you enable this option, PEM creates an alert when a new server or agent, as specified by the **Targettype** list, is added and deletes that alert when the target object is dropped.

- Move the **Autocreate?** slider to **Yes** to specify for PEM to create alerts based on the template. If you modify an existing alert template by changing the **Autocreate?** slider to **Yes** , PEM creates alerts on the existing agents and servers. If you change the slider from **Yes** to **No** , the default threshold values in existing alerts are erased, and you can't recover them. - Use the **Operator** list to select the operator for PEM to use when evaluating the current system values.

Select a greater-than sign (>) to trigger the alert when the system values are greater than the values entered in the **Thresholdvalues** fields.

Select a less-than sign (<) to indicate to trigger the alert when the system values are less than the values entered in the **Thresholdvalues** fields.

- Use the threshold fields to specify the values for PEM to compare to the system values to determine whether to raise an alert. You must specify values for all three thresholds (Low, Medium, and High).
- Use the **Checkfrequency** field to specify the default number of minutes between alert executions. This value specifies how often the server invokes the SQL code specified in the definition and compares the result to the threshold value specified in the template.

Fig. 24: The Probe Dependency tab of the Alert Templates dialog

Use the fields on the **ProbeDependency** tab to specify the names of probes referred to in the SQL query specified on the **SQL** tab:

- Use the **Probes** list to select from a list of the available probes. - To add the probe to the list of probes used by the alert template, select a probe name and select **Add** . - To remove a probe from the selected probes list, select the probe name and select **Delete** .

Fig. 25: The Parameters tab of the Alert Templates dialog

- Use the **Parameters** tab to define the parameters to use in the SQL code specified on the **SQL** tab. Select the plus sign (+). Then:

- Use the ****Name**** field to specify the parameter name.
- Use the ****Datatype**** list to specify the type of parameter.
- Use the ****Unit**** field to specify the type of unit specified by the parameter.

- Use the **Code** field on the **SQL** tab to provide the text of the SQL query for the server to invoke when executing the alert. The SQL query provides the result against which to compare the threshold value. If the alert result deviates from the specified threshold value, an alert is raised.

In the query, reference parameters defined on the **Parameters** tab sequentially by using the variable **param_x** . The **x** indicates the position of the parameter definition in the parameter list. For example, **param_1** refers to the first parameter in the parameter list, **param_2** refers to the second parameter in the parameter list, and so on.

The query can also include the following variables:

Fig. 26: The SQL tab of the Alert Templates dialog

Variable description	Variable name
agent identifier	'\${agent_id}'
server identifier	'\${server_id}'
database name	'\${database_name}'
schema name	'\${schema_name}'
Table	'\${object_name}'
index	'\${object_name}'
sequence	'\${object_name}'
function name	'\${object_name}'

- Use the **DetailedInformationSQL** field to provide a SQL query to invoke if the alert is triggered. The result set of the query might be displayed as part of the detailed alert information on the Alerts dashboard or Global Overview dashboard.

Note: If the specified query depends on one or more probes from different levels in the PEM hierarchy (server, database, schema, and so on), and a probe becomes disabled, any resulting alerts are displayed as follows:

- If the alert definition **and** the probe referenced by the query are **from the same** level **in** the PEM hierarchy, the server displays **any** alerts that reference the alert template on the Alert Error table of the Global Alert dashboard.
- If the alert definition **and** the probe referenced by the query are **from different** levels of the PEM hierarchy, the server displays **any** triggered alerts that reference the alert template on the Alert Details table of the hierarchy on which the alert was defined.

To save the alert template definition and add the template name to the Alert Templates list, select **Save** . After saving a custom alert template, you can use the Alerting dialog box to define an alert based on the template.

2.2.5 Exporting or importing alert templates

To export the alert template:

1. Select any alert template from the **AlertTemplates** tab.
2. Select **Export** in the upper-right corner of the table.
3. Select the **SaveFile** option.
4. To generate the JSON file, select **OK**.

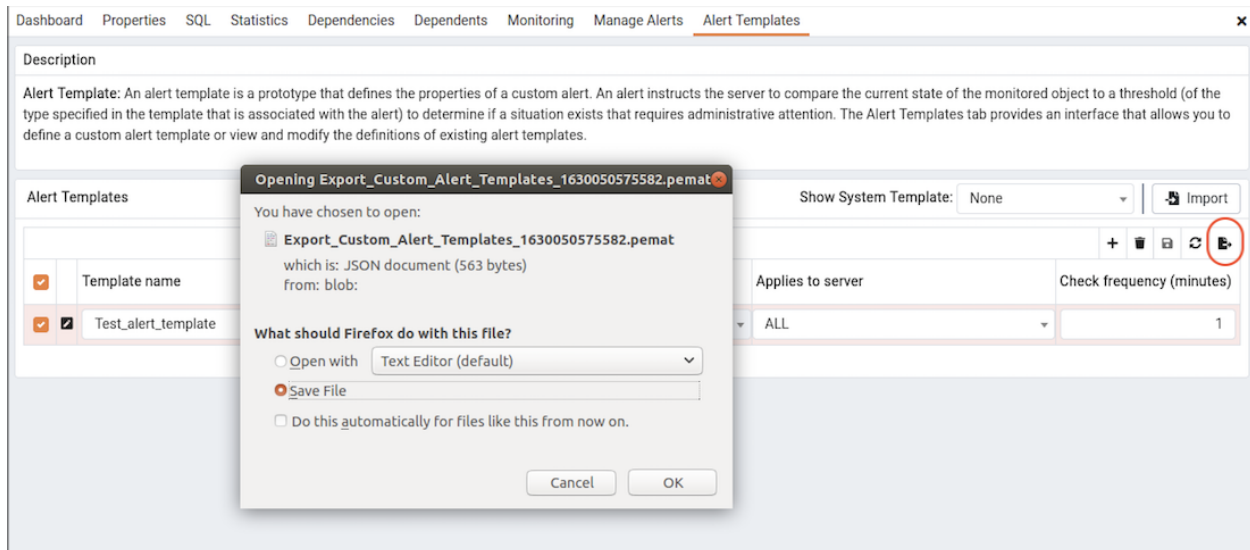


Fig. 27: Alert Templates Export

To import the Alert Template:

1. On the **AlertTemplates** tab, select **Import** in the upper-right corner.

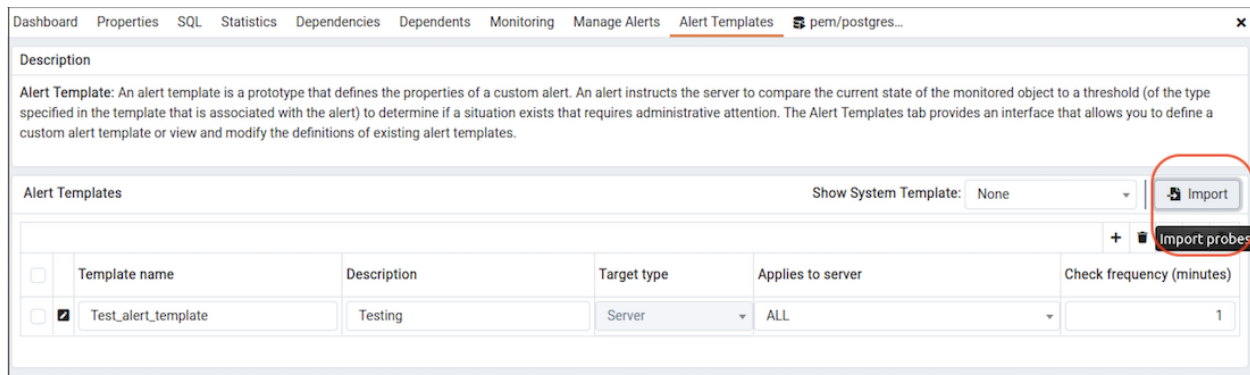


Fig. 28: Alert Templates Import

1. To select the JSON file with the code import, select **Browse**, and then select **Import**.
1. After selecting the file to import, you can select the following check boxes:
 - **skipexisting** — Skip the alert template if it already exists.

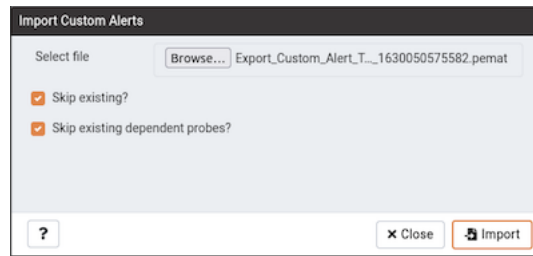


Fig. 29: Alert Templates Import Browse

- **skipexistingdependentprobe** — The alert templates depend on probes. Select this check box to skip the dependent probe if it already exists.

If both the check boxes are selected and the alert template already exists, then it skips importing the alert template with this message:

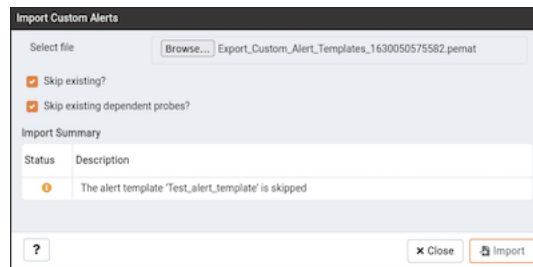


Fig. 30: Alert Templates Skip Both

If the **skipexisting** checkbox isn't selected, **skipdependentprobe** is selected, and the alert template already exists, then it imports the alert template successfully:

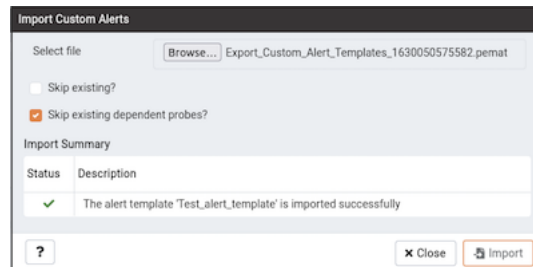


Fig. 31: Alert Templates Skip Dependent Probe

If both the check boxes are cleared and the alert template doesn't exist, then it successfully imports the alert template:

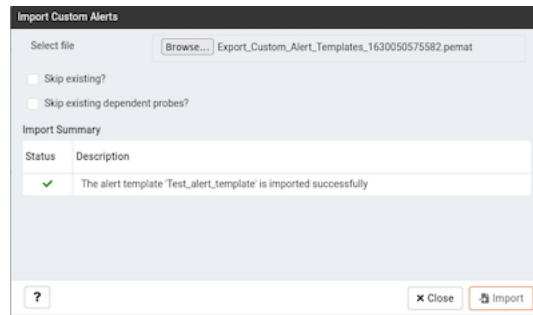


Fig. 32: Alert Template Import Successful

2.2.6 Modifying or deleting an alert template

To view the definition of an existing template (including PEM predefined alert templates), use the **ShowSystemTemplate** list to select the type of object monitored. When you select the object type, the Alert Templates table displays the alert templates that correspond with that object type.

Select a template name in the list, and select **Edit** at the left end of the row to review the template definition.

Use the tabs on the Alert Templates dialog box to view detailed information about the alert template:

- The **General** tab displays general information.
- The **ProbeDependency** tab lists the names of probes that provide data for the template.
- The **Parameters** tab lists the names of any parameters referred to in the SQL code.
- The **SQL** tab displays the SQL code that defines the behavior of the alert.

To delete an alert template, select the template name in the alert templates table and select **Delete**, located in the upper-right corner of the table. The alert history persists for the time specified in the **HistoryRetention** field in the template definition.

Predefined alert templates – reference

An alert definition contains a system-defined or user-defined set of conditions that PEM compares to the system statistics. If the statistics deviate from the boundaries specified for that statistic, the alert triggers, and the PEM client displays a warning on the Alerts Overview page and optionally sends a notification to a monitoring user.

The tables that follow list the system-defined alert templates that you can use to create an alert. This list is subject to change and can vary by system.

2.2.7 Templates applicable on agent

Template name	Description
Load Average (1 minute)	1-minute system load average
Load Average (5 minutes)	5-minute system load average
Load Average (15 minutes)	15-minute system load average
Load Average per CPU Core (1 minutes)	1-minute system load average per CPU core

continues on next page

Table 5 – continued from previous page

Template name	Description
Load Average per CPU Core (5 minutes)	5-minute system load average per CPU core
Load Average per CPU Core (15 minutes)	15-minute system load average per CPU core
CPU utilization	Average CPU consumption
Number of CPUs running higher than a	Number of CPUs running at greater than K% utilization threshold
Free memory percentage	Free memory as a percent of total system memory
Memory used percentage	Percentage of memory used
Swap consumption	Swap space consumed (in megabytes)
Swap consumption percentage	Percentage of swap area consumed
Disk Consumption	Disk space consumed (in megabytes)
Disk consumption percentage	Percentage of disk consumed
Disk Available	Disk space available (in megabytes)
Disk busy percentage	Percentage of disk busy
Most used disk percentage	Percentage used of the most utilized disk on the system
Total table bloat on host	The total space wasted by tables on a host, in MB
Highest table bloat on host	The most space wasted by a table on a host, in MB
Average table bloat on host	The average space wasted by tables on host, in MB
Table size on host	The size of tables on host, in MB
Database size on host	The size of databases on host, in MB
Number of ERRORS in the logfile on agent N in last X hours.	The number of ERRORS in the logfile on agent N in last X hours
Number of WARNINGS in the logfile on agent N in last X hours	The number of WARNINGS in the logfile on agent N in last X hours
Number of WARNINGS or ERRORS in the logfile on agent N in last X hours	The number of WARNINGS or ERRORS in the logfile on agent N in last X hours
Package version mismatch	Check for package version mismatch as per catalog
Total materialized view bloat on host	The total space wasted by materialized views on a host, in MB
Highest materialized view bloat on host	The most space wasted by a materialized view on a host, in MB
Average materialized view bloat on host	The average space wasted by materialized views on host, in MB
Materialized view size on host	The size of materialized views on host, in MB
Agent Down	Specified agent is currently down

2.2.8 Templates applicable on server

Template name	Description
Total table bloat in server	The total space wasted by tables in server, in MB
Largest table (by multiple of unbloated size)	Largest table in server, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB
Highest table bloat in server	The most space wasted by a table in server, in MB
Average table bloat in server	The average space wasted by tables in server, in MB
Table size in server	The size of tables in server, in MB
Database size in server	The size of databases in server, in MB
Number of WAL files	Total number of Write Ahead Log files
Number of prepared transactions	Number of transactions in prepared state
Total connections	Total number of connections in the server
Total connections as percentage of	Total number of connections in the server as a percentage of maximum
max_connections	connections allowed on server, settings
Unused, non-superuser connections	Number of unused, non-superuser connections on the server, user_info, settings
Unused, non-superuser connections as percentage of max_connections	Number of unused, non-superuser connections on the server as a percentage of max_connections of max_connections, user_info, settings
Ungranted locks	Number of ungranted locks in server
Percentage of buffers written by backends	The percentage of buffers written by backends vs. the total buffers written
Percentage of buffers written by checkpoint	The percentage of buffers written by the checkpoints vs. the total buffers written
Buffers written per second	Number of buffers written per second, over the last two probe cycles
Buffers allocated per second	Number of buffers allocated per second, over the last two probe cycles
Connections in idle state	Number of connections in server that are in idle state
Connections in idle-in-transaction state	Number of connections in server that are in idle-in-transaction state
Connections in idle-in-transaction state, as percentage of max_connections	Number of connections in server that are in idle-in-transaction state, as a percentage of maximum connections allowed on server, settings
Long-running idle connections	Number of connections in the server that have been idle for more than N seconds
Long-running idle connections and idle transactions	Number of connections in the server that have been idle or transactions idle-in-transaction for more than N seconds
Long-running idle transactions	Number of connections in the server that have been idle in transaction for more than N seconds
Long-running transactions	Number of transactions in server that have been running for more than N seconds
Long-running queries	Number of queries in server that have been running for more than N seconds
Long-running vacuums	Number of vacuum operations in server that have been running for more than N seconds

continues on next page

Table 6 – continued from previous page

Template name	Description
Long-running autovacuum	Number of autovacuum operations in server that have been running for more than N seconds
Committed transactions percentage	Percentage of transactions in the server that committed vs. that rolled-back over last N minutes
Shared buffers hit percentage	Percentage of block read requests in the server that were satisfied by shared buffers, over last N minutes
Tuples inserted	Tuples inserted into server over last N minutes
InfiniteCache buffers hit percentage	Percentage of block read requests in the server that were satisfied by InfiniteCache, over last N minutes
Tuples fetched	Tuples fetched from server over last N minutes
Tuples returned	Tuples returned from server over last N minutes
Dead Tuples	Number of estimated dead tuples in server
Tuples updated	Tuples updated in server over last N minutes
Tuples deleted	Tuples deleted from server over last N minutes
Tuples hot updated	Tuples hot updated in server, over last N minutes
Sequential Scans	Number of full table scans in server, over last N minutes
Index Scans	Number of index scans in server, over last N minutes
Hot update percentage	Percentage of hot updates in the server over last N minutes
Live Tuples	Number of estimated live tuples in server
Dead tuples percentage	Percentage of estimated dead tuples in server
Last Vacuum	Hours since last vacuum on the server
Last AutoVacuum	Hours since last autovacuum on the server
Last Analyze	Hours since last analyze on the server
Last AutoAnalyze	Hours since last autoanalyze on the server
Percentage of buffers written by backends over the last N minutes	The percentage of buffers written by backends vs. the total buffers backends over last N
Table Count	Total number of tables in server
Function Count	Total number of functions in server
Sequence Count	Total number of sequences in server
A user expires in N days	Number of days before a user's validity expires
Index size as a percentage of table size	Size of the indexes in server, as a percentage of their tables' size
Largest index by table-size percentage oc_index, table_size	Largest index in server, calculated as percentage of its table's size
Number of ERRORS in the logfile on server M in the last X hours	The number of ERRORS in the logfile on server M in last X hours
Number of WARNINGS in the logfile on server M in the last X hours	The number of WARNINGS in logfile on server M in the last X hours
Number of WARNINGS or ERRORS in the logfile on server M in the last X hours	The number of WARNINGS or ERRORS in the logfile on server M in the last X hours
Number of attacks detected in the last N minutes	The number of SQL injection attacks occurred in the last N minutes

continues on next page

Table 6 – continued from previous page

Template name	Description
Number of attacks detected in the last N minutes by username	The number of SQL injection attacks occurred in the last N minutes by username
Number of replica servers lag behind the primary by write location	Streaming Replication: number of replica servers lag behind the primary by write location
Number of replica servers lag behind the primary by flush location	Streaming Replication: number of replica servers lag behind the primary by flush location
Number of replica servers lag behind the primary by replay location	Streaming Replication: number of replica servers lag behind the primary by replay location
Replica server lag behind the primary by write location	Streaming Replication: replica server lag behind the primary by write location in MB
Replica server lag behind the primary by flush location	Streaming Replication: replica server lag behind the primary by flush location in MB
Replica server lag behind the primary by replay location	Streaming Replication: replica server lag behind the primary by replay location in MB
Replica server lag behind the primary by size (MB)	Streaming Replication: replica server lag behind the primary by size in MB
Replica server lag behind the primary by WAL segments	Streaming Replication: replica server lag behind the primary by WAL segments
Replica server lag behind the primary by WAL pages	Streaming Replication: replica server lag behind the primary by WAL pages
Total materialized view bloat in server	The total space wasted by materialized views in server, in MB
Largest materialized view (by multiple of unbloated size)	Largest materialized view in server, calculated as a multiple of its own estimated unbloated size; exclude materialized views smaller than N MB
Highest materialized view bloat in server	The most space wasted by a materialized view in server, in MB
Average materialized view bloat in server	The average space wasted by materialized views in server, in MB
Materialized view size in server	The size of materialized view in server, in MB
View Count	Total number of views in server
Materialized View Count	Total number of materialized views in server
Audit config mismatch	Check for audit config parameter mismatch
Server Down	Specified server is currently inaccessible
Number of WAL archives pending	Streaming Replication: number of WAL files pending to be replayed at replica
Number of minutes lag of replica server from primary server	Streaming Replication: number of minutes replica node is lagging behind the primary node
Log config mismatch	Check for log config parameter mismatch

2.2.9 Templates applicable on database

Template name	Description
Total table bloat in database	The total space wasted by tables in database, in MB
Largest table (by multiple of unbloated size)	Largest table in database, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB
Highest table bloat in database	The most space wasted by a table in database, in MB
Average table bloat in database	The average space wasted by tables in database, in MB
Table size in database	The size of tables in database, in MB
Database size	The size of the database, in MB
Total connections	Total number of connections in the database
Total connections as percentage of max_connections	Total number of connections in the database as a percentage of maximum connections allowed on server, settings
Ungranted locks	Number of ungranted locks in database
Connections in idle state	Number of connections in database that are in idle state
Connections in idle-in-transaction state	Number of connections in database that are in idle-in-transaction state
Connections in idle-in-transaction state, as percentage of max_connections	Number of connections in database that are in idle-in-transaction state, as a percentage of maximum connections allowed on server, settings
Long-running idle connections	Number of connections in the database that have been idle for more than N seconds
Long-running idle connections and idle transactions	Number of connections in the database that have been idle or idle-in-transaction for more than N seconds
Long-running idle transactions	Number of connections in the database that have been idle in transaction for more than N seconds
Long-running transactions	Number of transactions in database that have been running for more than N seconds
Long-running queries	Number of queries in database that have been running for more than N seconds
Long-running vacuums	Number of vacuum operations in database that have been running for more than N seconds
Long-running autovac-uums	Number of autovacuum operations in database that have been running for more than N seconds
Committed transactions percentage	Percentage of transactions in the database that committed vs. that rolled-back over last N minutes
Shared buffers hit percentage	Percentage of block read requests in the database that were satisfied by shared buffers, over last N minutes
InfiniteCache buffers hit percentage	Percentage of block read requests in the database that were satisfied by InfiniteCache, over last N minutes
Tuples fetched	Tuples fetched from database over last N minutes
Tuples returned	Tuples returned from database over last N minutes
Tuples inserted	Tuples inserted into database over last N minutes
Tuples updated	Tuples updated in database over last N minutes
Tuples deleted	Tuples deleted from database over last N minutes
Tuples hot updated	Tuples hot updated in database, over last N minutes

continues on next page

Table 7 – continued from previous page

Template name	Description
Sequential Scans	Number of full table scans in database, over last N minutes
Index Scans	Number of index scans in database, over last N minutes
Hot update percentage	Percentage of hot updates in the database over last N minutes
Live Tuples	Number of estimated live tuples in database
Dead Tuples	Number of estimated dead tuples in database
Dead tuples percentage	Percentage of estimated dead tuples in database
Last Vacuum	Hours since last vacuum on the database
Last AutoVacuum	Hours since last autovacuum on the database
Last Analyze	Hours since last analyze on the database
Last AutoAnalyze	Hours since last autoanalyze on the database
Table Count	Total number of tables in database
Function Count	Total number of functions in database
Sequence Count	Total number of sequences in database
Index size as a percentage of table size	Size of the indexes in database, as a percentage of their tables' size
Largest index by table-size percentage	Largest index in database, calculated as percentage of its table's size, oc_index, table_size
Database Frozen XID	The age (in transactions before the current transaction) of the database's frozen transaction ID
Number of attacks detected in the	The number of SQL injection attacks occurred in the last N minutes
Number of attacks detected in the	The number of SQL injection attacks occurred in the last N minutes by last N minutes by username
Queries that have been cancelled due to dropped tablespaces	Streaming Replication: number of queries that have been cancelled due to dropped tablespaces
Queries that have been cancelled due to lock time-outs	Streaming Replication: number of queries that have been cancelled due to lock time-outs
Queries that have been cancelled due to old snapshots	Streaming Replication: number of queries that have been cancelled due to old snapshots
Queries that have been cancelled due to pinned buffers	Streaming Replication: number of queries that have been cancelled due to pinned buffers
Queries that have been cancelled due to deadlocks	Streaming Replication: number of queries that have been cancelled due to deadlocks
Total events lagging in all slony clusters	Slony Replication: total events lagging in all slony clusters
Events lagging in one slony cluster	Slony Replication: events lagging in one slony cluster
Lag time (minutes) in one slony cluster	Slony Replication: lag time (minutes) in one slony cluster
Total rows lagging in xdb single primary replication	xDB Replication: Total rows lagging in xdb single primary replication
Total rows lagging in xdb multi primary replication	xDB Replication: Total rows lagging in xdb multi primary replication
Total materialized view bloat in database	The total space wasted by materialized views in database, in MB

continues on next page

Table 7 – continued from previous page

Template name	Description
Largest materialized view (by multiple of unbloated size)	Largest materialized view in database, calculated as a multiple of its estimated unbloated size; exclude materialized views smaller than N MB
Highest materialized view bloat in database	The most space wasted by a materialized view in database, in MB
Average materialized view bloat in database	The average space wasted by materialized views in database, in MB
Materialized view size in database	The size of materialized view in database, in MB
View Count	Total number of views in database
Materialized View Count	Total number of materialized views in database

2.2.10 Templates applicable on schema

Template name	Description
Total table bloat in schema	The total space wasted by tables in schema, in MB
Largest table (by multiple of unbloated size)	Largest table in schema, calculated as a multiple of its own estimated unbloated size; exclude tables smaller than N MB
Highest table bloat in schema	The most space wasted by a table in schema, in MB
Average table bloat in schema	The average space wasted by tables in schema, in MB
Table size in schema	The size of tables in schema, in MB
Tuples inserted	Tuples inserted in schema over last N minutes
Tuples updated	Tuples updated in schema over last N minutes
Tuples deleted	Tuples deleted from schema over last N minutes
Tuples hot updated	Tuples hot updated in schema, over last N minutes
Sequential Scans	Number of full table scans in schema, over last N minutes
Index Scans	Number of index scans in schema, over last N minutes
Hot update percentage	Percentage of hot updates in the schema over last N minutes
Live Tuples	Number of estimated live tuples in schema
Dead Tuples	Number of estimated dead tuples in schema
Dead tuples percentage	Percentage of estimated dead tuples in schema
Last Vacuum	Hours since last vacuum on the schema
Last AutoVacuum	Hours since last autovacuum on the schema
Last Analyze	Hours since last analyze on the schema
Last AutoAnalyze	Hours since last autoanalyze on the schema
Table Count	Total number of tables in schema
Function Count	Total number of functions in schema
Sequence Count	Total number of sequences in schema
Index size as a percentage of table size	Size of the indexes in schema, as a percentage of their table's size
Largest index by table-size percentage	Largest index in schema, calculated as percentage of its table's size, oc_index, table_size
Materialized View bloat	Space wasted by the materialized view, in MB
Total materialized view bloat in schema	The total space wasted by materialized views in schema, in MB

continues on next page

Table 8 – continued from previous page

Template name	Description
Materialized view size as a multiple of unbloated size	Size of the materialized view as a multiple of estimated unbloated size
Largest materialized view (by multiple of unbloated size)	Largest materialized view in schema, calculated as a multiple of its own estimated unbloated size; exclude materialized view smaller than N MB
Highest materialized view bloat in schema	The most space wasted by a materialized view in schema, in MB
Average materialized view bloat in schema	The average space wasted by materialized views in schema, in MB
Materialized view size	The size of materialized view, in MB
Materialized view size in schema	The size of materialized views in schema, in MB
View Count	Total number of views in schema
Materialized View Count	Total number of materialized views in schema
Materialized View Frozen XID	The age (in transactions before the current transaction) of the materialized view's frozen transaction ID

2.2.11 Templates applicable on table

Template name	Description
Table bloat	Space wasted by the table, in MB
Table size	The size of table, in MB
Table size as a multiple of unbloated size	Size of the table as a multiple of estimated unbloated size
Tuples inserted	Tuples inserted in table over last N minutes
Tuples updated	Tuples updated in table over last N minutes
Tuples deleted	Tuples deleted from table over last N minutes
Tuples hot updated	Tuples hot updated in table, over last N minutes
Sequential Scans	Number of full table scans on table, over last N minutes
Index Scans	Number of index scans on table, over last N minutes
Hot update percentage	Percentage of hot updates in the table over last N minutes
Live Tuples	Number of estimated live tuples in table
Dead Tuples	Number of estimated dead tuples in table
Dead tuples percentage	Percentage of estimated dead tuples in table
Last Vacuum	Hours since last vacuum on the table
Last AutoVacuum	Hours since last autovacuum on the table
Last Analyze	Hours since last analyze on the table
Last AutoAnalyze	Hours since last autoanalyze on the table
Row Count	Estimated number of rows in a table
Index size as a percentage of table size	Size of the indexes on table, as a percentage of table's size
Table Frozen XID	The age (in transactions before the current transaction) of the table's frozen transaction ID

2.2.12 Global templates

Template name	Description
Agents Down	Number of agents that haven't reported in recently
Servers Down	Number of servers that are currently inaccessible
Alert Errors	Number of alerts in an error state

2.2.13 Audit log alerting

PEM provides alert templates that let you use the Alerting dialog to create an alert that triggers when an **ERROR** or **WARNING** statement is written to a log file for a specific server or agent. To open the Alerting dialog, select the server or agent in the PEM client Object browser tree control, and select **Management>Alerting**.

To create an alert to notify you of error or warning messages in the log file for a specific server, create an alert that uses one of the following alert templates:

- Number of **ERRORS** in the logfile on server **M** in last **X** hours
- Number of **WARNINGS** in the logfile on server **M** in last **X** hours
- Number of **ERRORS** or **WARNINGS** in the logfile on server **M** in last **X** hours

To create an alert to notify you of error or warning messages for a specific agent, create an alert that uses one of the following alert templates. This functionality is supported only on EDB Postgres Advanced Server.

- Number of **ERRORS** in the logfile on agent **M** in last **X** hours
- Number of **WARNINGS** in the logfile on agent **M** in last **X** hours
- Number of **ERRORS** or **WARNINGS** in the logfile on agent **M** in last **X** hours

Defining a new alert

Use the PEM client **ManageAlerts** tab to define, copy, or manage alerts. To open the **ManageAlerts** tab, select **Management>ManageAlerts**.

The **ManageAlerts** tab displays a table of alerts that are defined on the object currently selected in the PEM client tree. You can use the Alerts table to modify an existing alert or to create a new alert.

To open the alert editor and create an alert, select the plus sign (+) in the upper-right corner of the table. The editor opens.

Use the fields on the **General** tab to provide information about the alert:

- Enter the name of the alert in the **Name** field.
- Use the **Template** list to select a template for the alert. An alert template is a function that uses one or more metrics or parameters to generate a value to which PEM compares user-specified alert boundaries. If the value returned by the template function evaluates to a value that's within the boundary of a user-defined alert as specified by the **Operator** and **Thresholdvalues** fields, PEM: - Raises an alert - Adds a notice to the Alerts overview display - Performs any actions specified on the template
- Use the **Enable?** switch to specify if the alert is enabled (**Yes**) or disabled (**No**).
- Use the **Interval** box to specify how often the alert confirms if the alert conditions are satisfied. Use the **Minutes** selector to specify an interval value. Use the **Default** switch to set or reset the **Minutes** value to the default (recommended) value for the selected template.

Quick Links

Copy Alerts

Alert Templates

Email Groups

Webhooks

Server Configuration

Help

Alerts

Manage Alerts

					Interval		History retention		
	Name	Auto created?	Template	Enable?	Default?	Minutes	Default?	Days	
☒	Average table bloat in server	Yes	Average table bloat in server	Yes	Yes		1	Yes	30
☒	Connections in idle-in-transaction state	Yes	Connections in idle-in-transactio...	Yes	Yes		1	Yes	30
☒	Connections in idle-in-transaction state, a...	Yes	Connections in idle-in-transactio...	Yes	Yes		1	Yes	30
☒	Connections in idle state	Yes	Connections in idle state	Yes	Yes		1	Yes	30
☒	Database size in server	Yes	Database size in server	Yes	Yes		1	Yes	30
☒	Highest table bloat in server	Yes	Highest table bloat in server	Yes	Yes		1	Yes	30
☒	Largest index by table-size percentage	Yes	Largest index by table-size perce...	Yes	Yes		1	Yes	30
☒	Last AutoVacuum	Yes	Last AutoVacuum	Yes	Yes		1	Yes	30
☒	Last Vacuum	Yes	Last Vacuum	Yes	Yes		1	Yes	30

Fig. 33: The Manage Alerts tab

General
Notification
Script execution

Name
Description
Template
Select from the list
A template uses metrics to generate a value to which PEM compares user specified alert boundaries. If the value returned by the template function evaluates to a value that is within the boundary of a user defined alert, PEM raises an alert.
Enable?
Yes
Select Yes to enable the alert, and No to disable the alert.
Interval
Default?
Yes
Minutes
1
Use fields in the Interval box to specify how often the alert should confirm that alert conditions are satisfied.
History retention
Default?
Yes
Days
30
Use fields in the History retention box to specify the number of days that PEM will store data collected by the alert.
Threshold values
Operator
>
Low
Medium
High
Unit
The fields in the Threshold values box work together to define the triggering criteria for the alert.
Auto created?
No
Parameter Options
Name
Value
server
Postgres Enterprise Manager Server

Fig. 34: The General tab of the alert editor

- Use the **Historyretention** box to specify the number of days that PEM stores data collected by the alert. Use the **Days** selector to specify the number of days to store the data. Use the **Default** switch to set or reset the **Days** value to the default value (30 days).
- Use controls in the **Thresholdvalues** box to define the triggering criteria for the alert. When the value specified in the **Thresholdvalues** fields evaluates to greater than or less than the system value (as specified with the Operator), PEM raises a Low, Medium or High alert level.
- Use the **Operator** list to select the operator for PEM to use when evaluating the current system values: - Select a greater-than sign (>) to trigger the alert when the system values are greater than the values entered in the **Thresholdvalues** fields. - Select a less-than sign (<) to trigger the alert when the system values are less than the values entered in the **Thresholdvalues** fields.
- Use the **Threshold** fields to specify the values for PEM to compare to the system values to determine whether to raise an alert. You must specify values for all three thresholds (Low, Medium, and High).

The Parameter Options table contains a list of parameters that are required by the selected template. The table displays both predefined parameters and parameters for which you must specify a value. You must specify a value for any parameter that displays a prompt in the **Value** column.

PEM can send a notification or execute a script if an alert is triggered or if an alert is cleared. Use the **Notification** tab to specify how PEM behaves if an alert is raised.

Fig. 35: The alert editor Notification tab

Use the **Emailnotification** box to specify the email group to receive an email notification if the alert is triggered at the specified level. Use the Email Groups tab to create an email group that contains the address of the users to notify when an alert is triggered. To access the **EmailGroups** tab, select **EmailGroups** located in the **QuickLinks** menu of the **ManageAlerts** tab.

- To instruct PEM to send an email when a specific alert level is reached, set the slider next to an alert level to **Yes**. Use the list to select the predefined user or group to notify.

You must configure the PEM server to use an SMTP server to deliver email before PEM can send email notifications.

Use the **Webhooknotification** box to specify one or multiple endpoints if the alert is triggered at the specified level. Use the webhooks tab to create a webhook endpoint to receive the notifications when an alert is triggered. To access the **Webhooks** tab, select **Webhooks** located in the **QuickLinks** menu of the **ManageAlerts** tab.

- Set **Enable?** to **Yes** to send the alert notifications to the webhook endpoint.
- Set **Overridedefaultconfiguration?** to **Yes** to set the customized alert levels as per the requirement. Once it's set to **Yes**, all the alert levels are enabled to configure.

Fig. 36: The alert editor Notification webhook tab

- Use the list to select a predefined endpoint to send a notification to for **Lowalerts?** , **Mediumalerts?** , **High-alerts?** , and **Clearedalerts?** .

Fig. 37: The alert editor Notification SNMP tab

Use the **Trapnotification** options to configure trap notifications for this alert:

- Set **Sendtrap** to **Yes** to send SNMP trap notifications when the state of this alert changes.
- Set **SNMPVer** to **v1** , **v2** , or **v3** to identify the SNMP version.
- Use the **Lowalert** , **Medalert** , and **Highalert** sliders to select the levels of alert to trigger the trap. For example, if you set the slider next to **Highalert** to **Yes** , PEM sends a notification when an alert with a high-severity level is triggered.

You must configure the PEM server to send notifications to an SNMP trap/notification receiver before notifications can be sent. For sending SNMP v3 traps, pemAgent uses 'User Security Model(USM)', which is in charge of authenticating, encrypting, and decrypting SNMP packets.

While sending SNMP v3 traps, the agent creates the `snmp_boot_counter` file. This file is created in the location

mentioned by the `batch_script_dir` parameter in `agent.cfg`. If this parameter isn't configured or if the directory isn't accessible due to authentication restrictions, then the file is created in the operating system temporary directory. If that's also not possible, then the file is created in your home directory.

Fig. 38: The alert editor Notification Nagios tab

Use the **Nagiosnotification** box to instruct the PEM server to notify Nagios network-alerting software when the alert is triggered or cleared. For more details, see [Using PEM with Nagios](#)

- Set the **SubmitpassiveservicecheckresulttoNagios** switch to **Yes** to instruct the PEM server to notify Nagios when the alert is triggered or cleared.

Use the **Scriptexecution** box to optionally define a script that executes if an alert is triggered and to specify details about the script execution.

- Set the **Executescript** slider to **Yes** to instruct PEM to execute the provided script if an alert is triggered.
- Set the **Executeonalertcleared** slider to **Yes** to instruct PEM to execute the provided script when the situation that triggered the alert is resolved.
- Use the **Executescripton** options to indicate for the script to execute on the PEM server or the monitored server.
- In the **Code** field, provide the script for PEM to execute. You can provide a batch/shell script or SQL code. In the script, you can use placeholders for the following:

%AlertName% — The name of the triggered alert.

%ObjectName% — The name of the server or agent on which the alert was triggered.

%ThresholdValue% — The threshold value reached by the metric when the alert triggered.

%CurrentValue% — The current value of the metric that triggered the alert.

%CurrentState% — The current state of the alert.

%OldState% — The previous state of the alert.

%AlertRaisedTime% — The time that the alert was raised or the most recent time that the alert state was changed.

To invoke a script on a Linux system, you must modify the entry for the `batch_script_user` parameter of the `agent.cfg` file and specify the user to use to run the script. You can either specify a non-root user or root for this parameter. If you don't specify a user or the specified user doesn't exist, then the script doesn't execute. Restart the agent after modifying the file.

General
Notification
Script execution

Script execution

Execute script?

No

Execute on alert cleared?

No

Execute script on

☐ PEM Server
☒ Monitored Server

Code

Use the fields in the Script execution box to (optionally) define a script that will be executed if an alert is triggered, and to specify details about the script execution.

- Set the Execute script slider to Yes to instruct PEM to execute the provided script if an alert is triggered.
- Set the Execute on alert cleared slider to Yes to instruct PEM to execute the provided script when the situation that triggered the alert has been resolved.
- Use the selector to indicate if the script should execute on the PEM Server or the Monitored Server.
- Provide the script that PEM should execute in the Code field. You can provide a batch/shell script, or SQL code. Within the script you can use the placeholders to replace the following:
 - %AlertID% - the id of the triggered alert.
 - %AlertName% - the name of the triggered alert.
 - %ObjectName% - the name of the server or agent on which the alert was triggered.
 - %ObjectType% - the type on which alert was generated.
 - %ThresholdValue% - the threshold value reached by the metric when the alert triggered.
 - %CurrentValue% - the current value of the metric that triggered the alert.
 - %CurrentState% - the current state of the alert.
 - %OldState% - the previous state of the alert.
 - %AlertRaisedTime% - the time that the alert was raised, or the most recent time that the alert state was changed.
 - %AgentID% - the id of the agent by which alert was generated.
 - %AgentName% - the name of the agent by which alert was generated.
 - %ServerID% - the id of the server on which alert was generated.
 - %ServerName% - the name of the server on which alert was generated.
 - %ServerIP% - the ip or address of the server on which alert was generated.
 - %ServerPort% - the the port of the server on which alert was generated.
 - %DatabaseName% - the name of the database on which alert was generated.
 - %SchemaName% - the name of the schema on which alert was generated.
 - %PackageName% - the name of the package on which alert was generated.
 - %DatabaseObjectName% - the name of the database object name like table name, function name etc on which alert was generated.
 - %Parameters% - the list of custom parameters used to generate the alert.

Fig. 39: The alert editor script execution tab

To invoke a script on a Windows system, set the registry entry for `AllowBatchJobSteps` to true and restart the PEM agent. PEM registry entries are located in `HKEY_LOCAL_MACHINE/Software/Wow6432Node/EnterpriseDB/PEM/agent`.

After you define the alert attributes, select **Edit** to close the alert definition editor and then **Save** in the upper-right corner of the Alerts table.

To discard your changes, select **Refresh**. A message prompts you to confirm that you want to discard the changes.

Note: Suppose you need to use the alert configuration placeholder values in an external script. You can do so either by passing them as the command-line arguments or exporting them as environment variables. The external script must have proper execution permissions.

- You can run the script with any of the placeholders as command-line arguments.

For example:

```
#!/bin/bash

bash <path_to_script>/script.sh "%AlertName% %AlertLevel% %AlertDetails%"
```

- You can define the environment variables for any of the placeholders and then use those environment variables in the script.

For example:

```
#!/bin/bash

export AlertName=%AlertName%
export AlertState=%AlertState%

bash <path_to_script>/script.sh
```

Modifying an alert

Use the Alerts table to manage an existing alert or create a new alert. Select an object in the PEM client tree to view the alerts that monitor that object.

You can modify some properties of an alert in the Alerts table:

- The Alert name column displays the name of the alert. To change the alert name, replace the name in the table and select **Save**.
- The Alert template column displays the name of the alert template that specifies properties used by the alert. You can use the list to change the alert template associated with an alert.
- Use the **Alertenable?** switch to specify if an alert is enabled (**Yes**) or disabled (**No**).
- Use the Interval column to specify how often PEM checks whether the alert conditions are satisfied. Set the **Default** switch to **No** and specify an alternate value, in minutes. Or set the **Default** switch to **Yes** to reset the value to its default setting. By default, PEM checks the status of each alert once every minute.
- Use the **Historyretention** field to specify the number of days that PEM stores data collected by the alert. Set the **Default** switch to **No** and specify an alternative value in days. Or set the **Default** switch to **Yes** to reset the value to its default setting. By default, PEM stores historical data for 30 days.

After modifying an alert, select **Save** (located in the upper-right corner of the table) to preserve your changes.

Alerts									
	Name	Auto created?	Template	Enable?	Interval		History retention		Days
					Default?	Minutes	Default?	Days	
☒	Audit config mismatch	☒	Audit config mismatch	☒	☒		☒		30
☒	Average table bloat in server	☒	Average table bloat in server	☒	☒		☒		30
☒	Connections in idle-in-transaction state	☒	Connections in idle-in-transactio...	☒	☒		☒		30
☒	Connections in idle-in-transaction state, as a ...	☒	Connections in idle-in-transactio...	☒	☒		☒		30
☒	Connections in idle state	☒	Connections in idle state	☒	☒		☒		30
☒	Database size in server	☒	Database size in server	☒	☒		☒		30
☒	Highest table bloat in server	☒	Highest table bloat in server	☒	☒		☒		30
☒	Largest index by table-size percentage	☒	Largest index by table-size perc...	☒	☒		☒		30
☒	Last AutoVacuum	☒	Last AutoVacuum	☒	☒		☒		30
☒	Last Vacuum	☒	Last Vacuum	☒	☒		☒		30
☒	Log config mismatch	☒	Log config mismatch	☒	☒		☒		30
☒	Number of prepared transactions	☒	Number of prepared transactions	☒	☒		☒		30
☒	Number of WAL archives pending	☒	Number of WAL archives pending	☒	☒		☒		30
☒	Number of WAL files	☒	Number of WAL files	☒	☒		☒		30
☒	Server Down	☒	Server Down	☒	☒		☒		30
☒	Table_Size_Alerts	☐	Table size in server	☒	☒		☒		30
☒	Table size in server	☒	Table size in server	☒	☒		☒		30
☒	Total connections	☒	Total connections	☒	☒		☒		30
☒	Total connections as percentage of max_con...	☒	Total connections as percentag...	☒	☒		☒		30
☒	Total table bloat in server	☒	Total table bloat in server	☒	☒		☒		30

Fig. 40: The Alerts table

To modify other alert attributes, select **Edit** to the left of an alert name to open an editor. The editor provides access to the complete alert definition.

Alert Details (Auto-refresh paused whilst rows are expanded. ⓘ)										
	Ack'd	Alert Type	Name	Value	Agent	Server	Database	Schema	Package	Object
▶	☐	High	Table size in server	1.9814453125 GB		Postgres Enterprise Manager Server				
Alerting Since: 2020-02-20 11:29:45										
General Parameters										
Table name	Schema name	Database name	Total table size(MB)							
table_statistics	perhistory	pem	1087							
server_logs	perdata	pem	263							
index_statistics	perhistory	pem	237							
session_info	perhistory	pem	137							
lock_info	perhistory	pem	88							

Fig. 41: The Alert details dialog

Use the Alert Details dialog box to modify the definition of the selected alert. After you modify the alert definition, select **Save**.

Deleting an alert

To mark an alert for deletion, select the alert name in the Alerts table. Then select **Delete** to the left of the name. The alert remains in the list in red strike-through font.

Delete is a toggle. You can undo the deletion by selecting it a second time. To permanently delete the alert definition, select **Save**.

Copying an alert

To speed up the deployment of alerts in the PEM system, you can copy alert definitions from one object to one or more target objects.

To copy alerts from an object, select the object in the PEM client tree on the main PEM window. Then, select **Management>CopyAlerts**. On the **ManageAlerts** tab, from the Quick Links toolbar, select **CopyAlerts**.

The Copy Alert Configuration dialog box copies all alerts from the object selected in the PEM client tree to the objects selected on the dialog box. Expand the tree to select nodes to specify as the target objects. The tree displays a red warning indicator next to the source object.

To copy alerts to multiple objects at once, select a parent node of the targets. For example, to copy the alerts from one table to all tables in a schema, select the check box next to the schema. PEM copies alerts only to targets that are the same type as the source object.

Select **Ignoreduplicates** to prevent PEM from updating any existing alerts on the target objects with the same name as those being copied. Select **Replaceduplicates** to replace existing alerts with alerts of the same name from the source object.

Select **ConfigureAlerts** to copy the alerts from the source object to all objects of the same type in or under those objects selected on the Copy Alert Configuration dialog box.

	Name	Auto created?	Template	Enable?	Interval		History retention	
					Default?	Minutes	Default?	Days
☒	Audit config mismatch	☒	Audit config mismatch	☒	☒	1	☒	30
☒	Average table bloat in server	☒	Average table bloat in server	☒	☒	1	☒	30
☒	Connections in idle-in-transaction state	☒	Connections in idle-in-transactio...	☒	☒	1	☒	30
☒	Connections in idle-in-transaction state, as a ...	☒	Connections in idle-in-transactio...	☒	☒	1	☒	30
☒	Connections in idle state	☒	Connections in idle state	☒	☒	1	☒	30
☒	Database size in server	☒	Database size in server	☒	☒	1	☒	30
☒	Highest table bloat in server	☒	Highest table bloat in server	☒	☒	1	☒	30
☒	Largest index by table-size percentage	☒	Largest index by table-size perc...	☒	☒	1	☒	30
☒	Last AutoVacuum	☒	Last AutoVacuum	☒	☒	1	☒	30
☒	Last Vacuum	☒	Last Vacuum	☒	☒	1	☒	30
☒	Log config mismatch	☒	Log config mismatch	☒	☒	1	☒	30
☒	Number of prepared transactions	☒	Number of prepared transactions	☒	☒	1	☒	30
☒	Number of WAL archives pending	☒	Number of WAL archives pending	☒	☒	1	☒	30
☒	Number of WAL files	☒	Number of WAL files	☒	☒	1	☒	30
☒	Server Down	☒	Server Down	☒	☒	1	☒	30
☒	Table_Size_Alerts	☐	Table-size-in-server	☒	☒	1	☒	30
☒	Table size in server	☒	Table size in server	☒	☒	1	☒	30
☒	Total connections	☒	Total connections	☒	☒	1	☒	30
☒	Total connections as percentage of max_con...	☒	Total connections as percentag...	☒	☒	1	☒	30
☒	Total table bloat in server	☒	Total table bloat in server	☒	☒	1	☒	30

Fig. 42: Deleting an alert

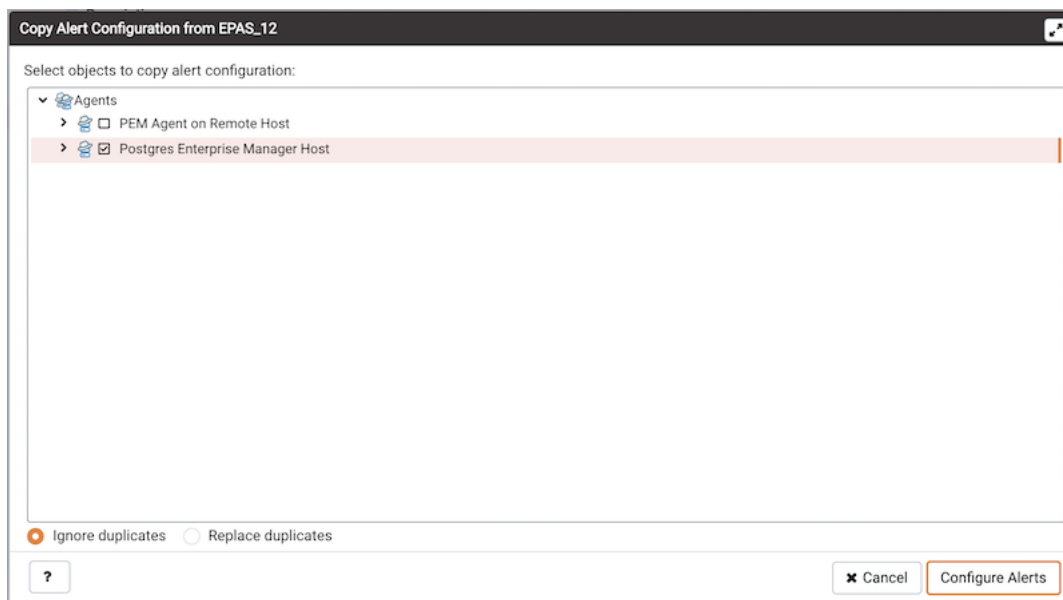


Fig. 43: The Copy Alert Configuration dialog

2.2.14 Schedule an alert blackout

You can use the **Management>ScheduleAlertBlackout** to schedule an alert blackout for your Postgres servers and PEM agents during maintenance. Alerts aren't raised during a defined blackout period.

To schedule an alert blackout, select **Management>ScheduleAlertBlackout**.

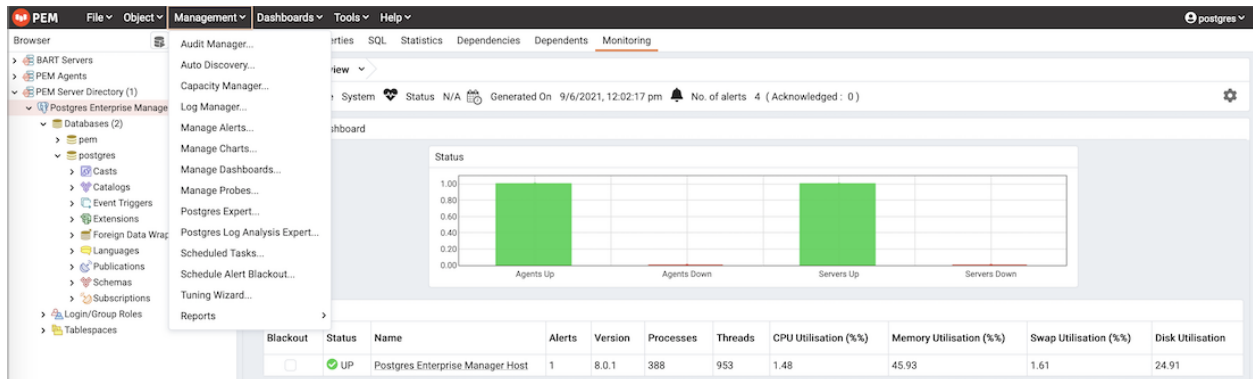


Fig. 44: PEM Management Menu

In the Schedule Alert Blackout dialog box, use the tabs to define the blackout period for servers and agents. On the **Server** tab, to add a row, select the plus sign (+) at the top-right corner.

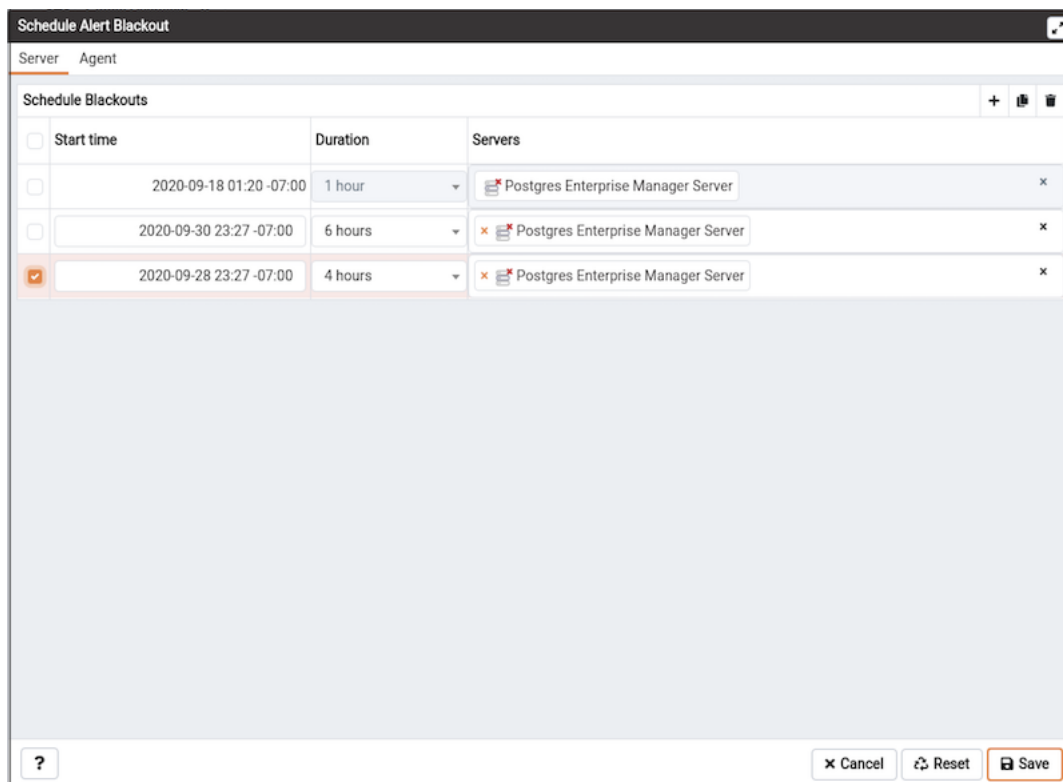


Fig. 45: Schedule Alert Blackout - Server tab

Use the **Server** tab to provide information about an alert blackout period. After you save the blackout period, you cannot edit it.

- Use the **Starttime** field to provide the date and time to start the alert blackout.
- Use the **Duration** field to provide the interval for which you want to black out the alerts.
- Use the **Servers** field to provide the server name for which you want to black out the alerts. You can also select multiple servers to black out the alerts for all of those servers.

After providing details, select **Save** . The alerts don't appear on the Alerts dashboard for the scheduled interval of that server.

You can also schedule a blackout period for PEM agents using the **Agent** tab on the dialog box. To add a row, on the **Agent** tab, select the plus sign (+) at the top-right corner.

Start time	Duration	Agents
2020-09-18 03:33 -07:00	1 hour	Postgres Enterprise Manager Host
2020-09-25 03:33 -07:00	4 hours	Agent2
2020-09-30 03:33 -07:00	23 hours	Postgres Enterprise Manager Host
2020-09-21 23:23 -07:00	6 hours	Agent2

Buttons: Cancel, Reset, Save

Fig. 46: Schedule Alert Blackout - Agent tab

Use the **Agent** tab to provide the information about an alert blackout period. After you save the blackout period, you can't edit it.

- Use the **Starttime** field to provide the date and time to start the alert blackout.
- Use the **Duration** field to provide the interval for which you want to black out the alerts.
- Use the **Agents** field to provide the agent name for which you want to black out the alerts. All server-level alerts for the servers bound to that agent black out.

After providing details, save the details by selecting **Save** . The alerts aren't displayed on the Alert dashboard for the scheduled interval for that PEM agent.

You can select **Clone** from the top-right corner of the dialog box to clone the scheduling of an alert blackout. To create the cloned copy of all the selected servers or agents, select the servers or agents you want to clone, and then select **Clone** . You can edit newly created schedules as needed, and then select **Save** .

Select **Delete** from the top-right corner of the dialog box to remove a scheduled alert blackout. Select the servers or agents and then select **Delete** .

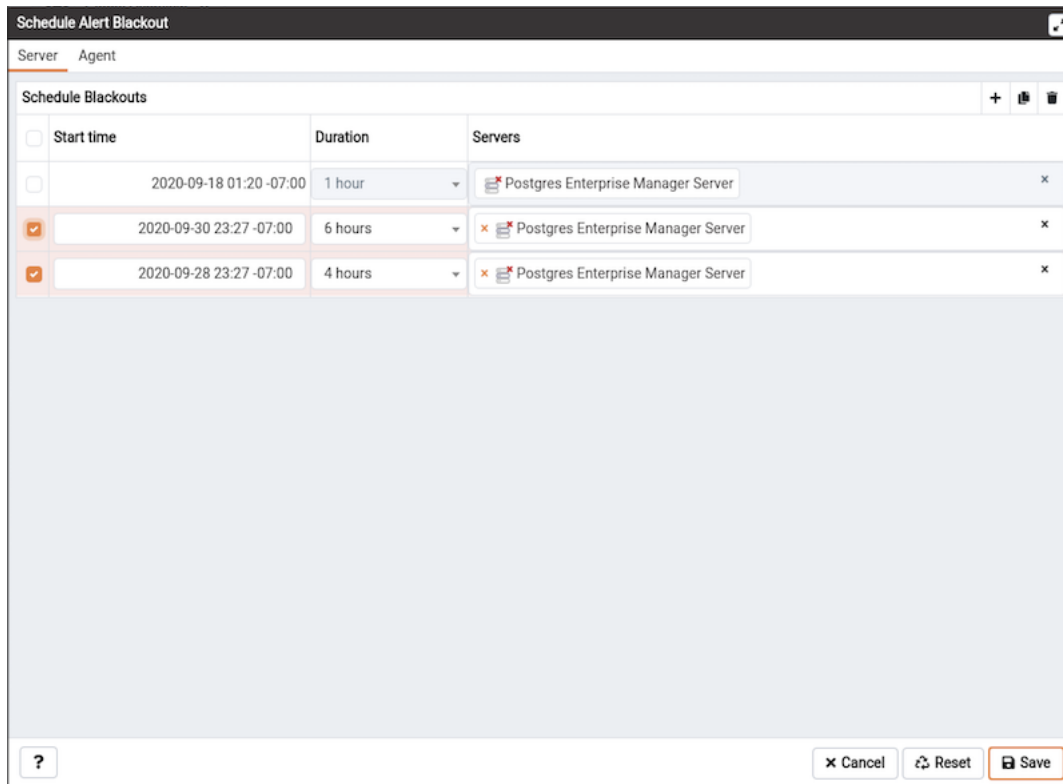


Fig. 47: Schedule Alert Blackout - Select servers

Select a server for which you want to delete the scheduled alert backout, and then select **Delete**. The server prompts for confirmation before deleting that row.

You can select **Reset** to reset the details on the Alert Blackout dialog box to the default settings. Saved blackouts aren't affected.

2.3 Notifications

PEM can send a notification or execute a script if an alert is triggered or cleared. You can send notifications using the following options:

- SMTP
- Webhooks
- SNMP
- Nagios

Use the **Notification** tab to specify PEM behavior when an alert is raised.

2.3.1 SMTP

You must configure the PEM server to use an SMTP server to deliver email before PEM can send email notifications.

Creating an email group

PEM monitors your system for conditions that require attention. You can use an email group to specify the email addresses of users for the server to notify if current values deviate from threshold values specified in an alert definition. An email group can notify multiple users or target specific users during the time periods you specify.

Use the **EmailGroups** tab to configure groups of SMTP email recipients. To open the **EmailGroups** tab, in the PEM client, select **Management>ManageAlerts**. When the **ManageAlerts** tab opens, select **EmailGroups** from the Quick Links toolbar.



Fig. 48: The Email Groups tab

The **EmailGroups** tab displays a list of the currently defined email groups. To modify an existing group, select a group name and select **Edit** at the far left end of the row.

To define a new email group, select the plus sign (+) in the upper-right corner of the **EmailGroups** table.

Use the Email Group dialog box to define an email group and its members.

Each row in the email group definition associates a set of email addresses with a specific time period. When an alert is triggered, the server evaluates the times specified in each row and sends the message to those group members whose definitions are associated with the time that the alert triggered.

1. Provide a name for the email group in the **GroupName** field.
2. To open the **Options** tab, select the plus sign (+) in the group members table.
3. Add the member addresses to receive notifications for the time period specified:
 - Enter a comma-delimited list of recipient addresses in the **ReplytoAddresses** field. - Enter a comma-delimited list of addresses to receive a copy of the email in the **CCAddresses** field. - Enter a comma-delimited list of addresses to receive a copy of the email without the knowledge of other recipients in the **BccAddresses** field. - Enter the email address to send the messages from in the **FromAddress** field. - Use the **Subjectprefix** field to provide a message to add to the start of each subject line when a notification is sent. - Use the **FromTime** and **ToTime** selectors to specify the time range for notifications to the group members that are identified on this row. Provide these values in the locale of the PEM client host. The PEM server translates the time into other time zones as required.
4. Select **Add** to add a row to the table, and specify another time period and the email addresses to notify during those hours.
5. When you've finished defining the email group, select **Save**.

Email Groups

	Group Name
☒	<Default>
☒	sales

Email Group

Group Name

sales

Email group options specify email notifications will be delivered to a specific group member (or members) during a selected time period.

- To addresses:** Enter a comma-delimited list of recipient addresses in the To addresses field.
- Reply to addresses:** Enter a comma-delimited list of recipient addresses in the Reply to addresses field.
- CC addresses:** Enter a comma-delimited list of addresses that will receive a copy of the email in the CC addresses field.
- BCC addresses:** Enter a comma-delimited list of addresses that will receive a copy of the email (without the knowledge of other recipients) in the BCC addresses field.
- From address:** Enter the email address that messages to this group should be sent from in the From address field.
- Subject prefix:** Enter the email subject prefix to this group in the Subject prefix field.
- From time/To time(HH:MM:SS):** Use the From time and To time hour selectors to select a time range for a group member (or members). When a notification is sent, the server will evaluate the times specified within the group list and send the message to those members whose group entries include the current time. Provide the From time and To time values in the locale of the PEM client host, and the PEM server will translate the time into other time zones as required.

To addresses	From address	From time	To time
☒ sales@enterprisedb.com	firstname.lastname@enterprisedb.com	00:00:00	23:59:59

Options

To addresses

sales@enterprisedb.com

Reply to addresses

CC addresses

BCC addresses

From address

firstname.lastname@enterprisedb.com

Subject prefix

From time

00:00:00

To time

23:59:59

Fig. 49: Adding an email group

2.3. Notifications

75

After creating the email group, you can use the **ManageAlerts** tab to set up the notification details for an alert to direct notifications to the group.

Deleting an email group

To delete an email group, in the Email Group table, select the name of the group and select **Delete**, located to the left of the group name.



Fig. 50: Deleting an email group

The group name appears in the Email Group table in red. Select **Save** to permanently remove the group from the table.

2.3.2 Webhook

You must configure the PEM Server to use webhooks to receive notification of alert events on threshold value violations in your configured applications.

Creating a webhook

PEM monitors your system for conditions that require user attention. You can use a webhook to create the endpoints to receive a notification if current values deviate from threshold values specified in an alert definition. Based on the events triggered, PEM sends a notification to multiple webhook endpoints or to specific target webhook endpoints.

Use the **Webhooks** tab to configure endpoint recipients. To open the **Webhooks** tab, select **Management>ManageAlerts**. From the **ManageAlerts** tab, on the Quick Links toolbar, select **Webhooks**.

The **Webhooks** tab displays a list of the currently defined recipient applications as endpoints. Select an endpoint and select **Edit** at the far left end of the row to modify an existing endpoint.

To define a new webhook, select the plus sign (+) in the upper-right corner of the table.

Use the **General** tab to define the basic details of the webhook:

- Provide a name for the webhook in the **Name** field.
- Specify a webhook URL to deliver all the notifications to in the **URL** field.
- Set the request method type used to make the call in the **RequestMethod** field: **POST** or **PUT**.
- By default, webhooks are enabled. To disable a webhook, set **Enable?** to **No**.

Note: The **Enable?** setting works only if the *enable_webhook* parameter is set to *true* in the *agent.cfg* file. By default, the *enable_webhook* parameter is set to *true* only for the agent running on the PEM server host. For all other agents running on other hosts, you need to set it to *true* manually.

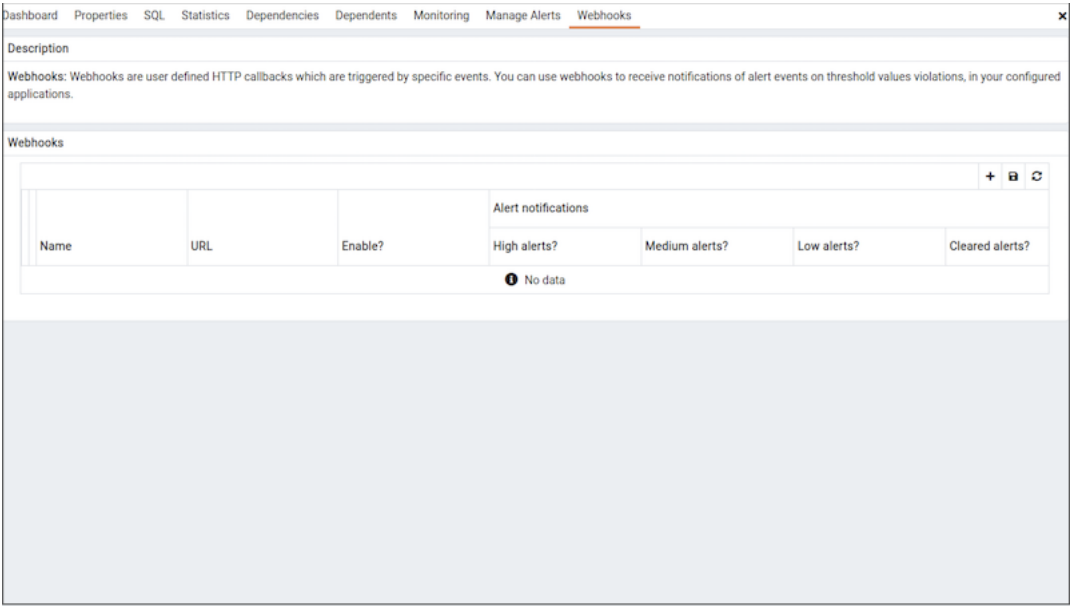


Fig. 51: Webhooks tab

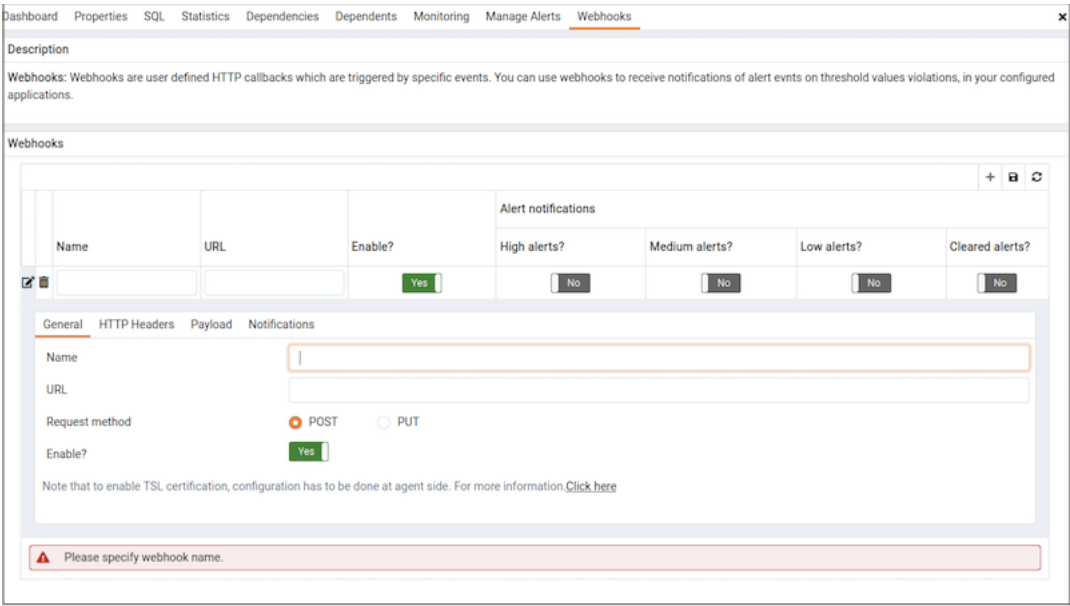


Fig. 52: Webhooks - Add New Webhook - General Tab

Defining webhook SSL configurations

You can define the webhook SSL parameters in the respective agent configuration file or registry in Windows. You can find the list of webhook SSL parameters in PEM agent configuration parameters . If you add or remove any of the agent configuration parameters, you must restart the agent to apply them.

- On 32-bit Windows systems, PEM registry entries for webhooks are located in HKEY_LOCAL_MACHINE\\Software\\EnterpriseDB\\PEM\\agent\\WEBHOOK .
- On 64-bit Windows systems, PEM registry entries for webhooks are located in HKEY_LOCAL_MACHINE\\Software\\Wow6432Node\\EnterpriseDB\\PEM\\agent\\WEBHOOK .
- On Linux systems, PEM configuration options for webhooks are stored in the agent.cfg file , located by default in /usr/edb/pem/agent/etc .

```
[WEBHOOK/Django]
webhook_ssl_key=<webhook_client_ssl_key_path>
webhook_ssl_cert=<webhook_client_ssl_certificate_path>
webhook_ssl_ca_cert=<webhook_server_ca_certificate_path>
webhook_ssl_crl=<crl_file_path_to_validate_webhook_server>
allow_insecure_webhooks=<true|false>
```

Fig. 53: Example - Webhook SSL Parameters in agent.cfg file

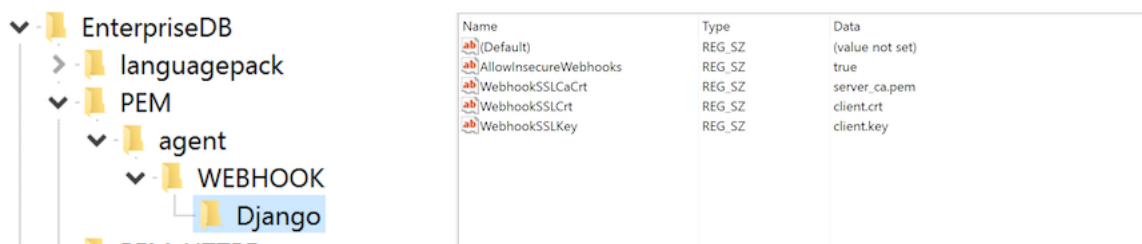


Fig. 54: Example - Webhook SSL Parameters in windows registry

Name	URL	Enable?	Alert notifications			
			High alerts?	Medium alerts?	Low alerts?	Cleared alerts?
Django	http://192.168.1.2:8000/public-w...	Yes	Yes	Yes	Yes	Yes

Key		Value
Content-type		application/json

Fig. 55: Webhooks - Add New Webhook - HTTP Headers Tab

Use the **HTTPHeaders** tab to define the header parameters to pass while calling the webhook endpoints:

- Specify all the values as a key and value pair.
- Specify a key parameter in the **Key** field and a value in the **Value** field.
- To add HTTP headers, select the plus sign (+) in the upper-right corner of the HTTP Headers table.

- To delete HTTP headers, select **Delete** to the left of **Key** . The header remains in the list but in strike-through font. Select **Save** to permanently delete the headers.
- To edit the HTTP headers, select **Edit** to the left of **Key** .

Webhooks

Name	URL	Enable?	Alert notifications
Django	http://192.168.1.23000/public-webhook	☒	High alerts? ☐ No Medium alerts? ☐ No Low alerts? ☐ No Cleared alerts? ☐ No

General HTTP Headers Payload Notifications

Type

Template

```

1 {
2   "payload": {
3     "version": "1.0",
4     "type": "alert",
5     "source": "edb-pes",
6     "alert": {
7       "AlertID": "%AlertID%",
8       "AlertName": "%AlertName%",
9       "Server/Agent": "%ObjectName%",
10      "ServerID": "%ServerID%",
11      "ServerName": "%ServerName%",
12      "ServerIP": "%ServerIP%",
13      "ServerPort": "%ServerPort%",
14      "DatabaseName": "%DatabaseName%",
15      "SchemaName": "%SchemaName%",
16      "PackageName": "%PackageName%",
17      "DatabaseObjectName": "%DatabaseObjectName%",
18      "Parameters": "%Parameters%",
19      "AlertInfo": "%AlertInfo%",
20      "Object Type": "%ObjectType%",
21      "Threshold Value": "%ThresholdValue%",
22      "Current Value": "%CurrentValue%",
23      "Current State": "%CurrentState%",
24      "Old State": "%OldState%",
25      "Alert Raised Time": "%AlertRaisedTime%",
26      "Info": "%AlertInfo%"
27    }
28  }
29 }

```

Test Connection

Provide the JSON that PEM should execute in the template field. Within the script you can use the placeholders to replace the following:

- %AlertID% - the id of the triggered alert.
- %AlertName% - the name of the triggered alert.
- %ObjectName% - the name of the server or agent on which the alert was triggered.
- %ObjectType% - the type on which alert was generated.
- %ThresholdValue% - the threshold value reached by the metric when the alert triggered.
- %CurrentValue% - the current value of the metric that triggered the alert.
- %CurrentState% - the current state of the alert.
- %OldState% - the previous state of the alert.
- %AlertRaisedTime% - the time that the alert was raised, or the most recent time that the alert state was changed.
- %AgentID% - the id of the agent by which alert was generated.
- %AgentName% - the name of the agent by which alert was generated.
- %ServerID% - the id of the server on which alert was generated.
- %ServerName% - the name of the server on which alert was generated.
- %ServerIP% - the ip or address of the server on which alert was generated.
- %ServerPort% - the port of the server on which alert was generated.
- %DatabaseName% - the name of the database on which alert was generated.
- %SchemaName% - the name of the schema on which alert was generated.
- %PackageName% - the name of the package on which alert was generated.
- %DatabaseObjectName% - the name of the database object name like table name, function name etc on which alert was generated.
- %Parameters% - the list of custom parameters used to generate the alert.
- %AlertInfo% - the detailed database object-level information of the alert.

Fig. 56: Webhooks - Add New Webhook - Payload Tab

Use the **Payload** tab to define the JSON data to send to the endpoint when an alert is triggered:

- ****Type**** specifies data to send **in** the **format type**, that **is**, JSON.
- Use **Template** to configure JSON data sent to endpoints. In the template, you can use placeholders for the following:

- ``%AlertID`` — The id of the triggered alert. - ``%AlertName`` — The name of the triggered alert. - ``%ObjectName`` — The name of the server or agent on which the alert was triggered. - ``%ObjectType`` — The type on which the alert was generated. - ``%ThresholdValue`` — The threshold value reached by the metric when the alert triggered. - ``%CurrentValue`` — The current value of the metric that triggered the alert. - ``%CurrentState`` — The current state of the alert. - ``%OldState`` — The previous state of the alert. - ``%AlertRaisedTime`` — The time that the alert was raised or the most recent time that the alert state changed. - ``%AgentID`` — The id of the agent by which the alert was generated. - ``%AgentName`` — The name of the agent by which the alert was generated. - ``%ServerID`` — The id of the server on which the alert was generated. - ``%ServerName`` — The name of the server on which the alert was generated. - ``%ServerIP`` — The ip or address of the server on which the alert was generated. - ``%ServerPort`` — The the port of the server on which the alert was generated. - ``%DatabaseName`` — The name of the database on which the alert was generated. - ``%SchemaName`` — The name of the schema on which the alert was generated. - ``%PackageName`` — The name of the package on which the alert was generated. - ``%DatabaseObjectName`` — The name of the database object, like table name or function name, on which the alert was generated. - ``%Parameters`` — The list of custom parameters used to generate the alert. - ``%AlertInfo`` — The detailed database object-level information.

(continued from previous page)

- Select **TestConnection** to test notification delivery to the mentioned endpoint.

The screenshot shows the 'Webhooks' management interface. At the top, there's a table with columns: Name, URL, Enable?, and Alert notifications (High alerts?, Medium alerts?, Low alerts?, Cleared alerts?). Below this, the 'Notifications' tab is selected, showing settings for 'All alerts?' and individual alert levels (High, Medium, Low, Cleared), all set to 'Yes'.

Fig. 57: Webhooks - Add New Webhook - Notifications Tab

Use the **Notifications** tab to specify an alert level for webhook endpoints:

- Set **Allalerts** to **Yes** to enable all alert levels to send notifications.
- To send a notification when a specific alert level is reached, set the slider next to an alert level to **Yes**. You must set **Allalerts** to **No** to configure an individual alert level.

Deleting a webhook

To mark a webhook for deletion, in the Webhooks table, select the webhook name and select **Delete** to the left of the name. The alert remains in the list but in strike-through font.

Webhooks				Alert notifications			
Name	URL	Enable?		High alerts?	Medium alerts?	Low alerts?	Cleared alerts?
☒ Django	http://192.168.1.2:8000/public-webh...	Yes		Yes	Yes	Yes	Yes
☒ Slack	https://hooks.slack.com/services/T0...	Yes		Yes	Yes	Yes	Yes

Fig. 58: Webhooks - Delete an existing webhook

Delete is a toggle. You can undo the deletion by selecting **Delete** a second time. Select **Save** to permanently delete the webhook definition.

2.3.3 SNMP

You must configure the PEM server to send the notifications to an SNMP trap/notification receiver before notifications can be sent. Set the SNMP ver to v1, v2, or v3 to identify the SNMP version.

Example - Configure SNMP V3 traps with net-snmp trap receiver

1. Set `snmp_security_engine_id` to `PEM_SNMP_AGENT` in plain text format in the Server Configuration dialog box.
2. Convert the plain text value to hexadecimal format to use it in `snmptrapd.conf` file. (You can have hexadecimal values of `snmp_security_engine_id` up to 32 octets length). `edb_notranlate_0`
3. Set the following parameters in the Server Configuration dialog box: - `snmp_security_nam` to `pem_snmp_user` - `snmp_authentication_protocol` to `MD5` - `snmp_authentication_password` to `pem_auth_pass` - `snmp_privacy_protocol` to `DES` - `snmp_privacy_password` to `pem_priv_pass`
4. The `snmptrapd.conf` file has the following values:

```
createUser -e 0x50454D5F534E4D505F4147454E540A pem_snmp_user MD5 pem_auth_pass DES pem_
↪priv_pass
authUser log pem_snmp_user
```

2.3.4 Using PEM with Nagios

The PEM server can send a passive alert result to Nagios network-alerting software when a user-defined alert is triggered. To instruct the PEM server to notify Nagios of a triggered alert, you must:

- Enable Nagios notification for each alert that triggers a notification from the PEM server to Nagios. You must configure PEM alerting before you create the `host.cfg` and `services.cfg` files.
- Configure Nagios-related behaviors of the PEM server.
- Create the `host.cfg` and `services.cfg` configuration files.
- If necessary, modify the Nagios configuration file and restart the Nagios server.
- Install the PEM agent on the system where the Nagios server is installed and register it with the PEM Server. Set `enable_nagios` configuration to `true` in the `agent.cfg` file for that agent, and restart the agent service.

After configuring the server to enable Nagios alerting, any triggered alerts send a passive check result to the Nagios service. The syntax of a passive alert is:

```
<timestamp> PROCESS_SERVICE_CHECK_RESULT; <host_name> ; <service_name> ; <service_status>
;
```

Where:

`timestamp` is the date and time that the alert was triggered.

`host_name` is the name of the server or agent.

`service_name` is the name of the alert.

`service_status` is the numeric service status value:

- 0 if the service status is OK
- 1 if the service status is WARNING
- 2 if the service status is CRITICAL

- 3 if the service status is UNKNOWN

The PEM server uses the following rules to evaluate the service status:

- If the PEM alert level is CLEARED, the warning message reads OK.
- If the PEM alert level is LOW, the warning message reads WARNING.
- If the `is_nagios_medium_alert_as_critical` flag (specified in the PEM server configuration dialog box) is set to FALSE and the alert level MEDIUM, the warning message reads WARNING.
- If the `is_nagios_medium_alert_as_critical` flag (specified in the PEM server configuration dialog box) is set to TRUE and the alert level is MEDIUM, the warning message reads CRITICAL.
- If the PEM alert level is HIGH, the warning message reads CRITICAL.

Enabling Nagios notification for an alert

The PEM server maintains a unique set of notification properties for each enabled alert. Use the **Notification** tab of the **ManageAlerts** tab to specify that, when triggered, a given alert sends an alert notice to Nagios.

To modify the notification properties of an alert, right-click the name of the object monitored by the alert, and select **Management>ManageAlerts** . On the **ManageAlerts** tab, select **Edit** to the left of the alert name in the Alerts list. When the edit pane opens, select the **Notification** tab.

To enable Nagios notification, move the slider next to **SubmitpassiveservicecheckresulttoNagios** to **Yes** . Then select **Save** .

Configuring Nagios-related behavior of the PEM Server

You can use the Server Configuration dialog box to provide information about your Nagios configuration to the PEM server. To open dialog box, select **Management>ServerConfiguration** .

Four server configuration parameters specify information about your Nagios installation and PEM server behavior related to Nagios:

- Use the `nagios_cmd_file_name` parameter to specify the location of the Nagios pipeline file that receives passive check alerts from PEM. The default value of this parameter is `/usr/local/nagios/var/rw/nagios.cmd` . If your `nagios.cmd` file resides elsewhere, specify the file location in the **Value** field.
- Move the slider in the `nagios_enabled` parameter to **Yes** to instruct the PEM server to send passive check alerts to Nagios.
- Use the `nagios_medium_alert_as_critical` slider to specify the warning severity that the PEM server passes to Nagios if a medium alert is triggered:

- If the ``is_nagios_medium_alert_as_critical`` flag is set to `**FALSE**` and the alert level is `**MEDIUM**` , the warning message reads WARNING.
- If the ``is_nagios_medium_alert_as_critical`` flag is set to `**TRUE**` and the alert level is `**MEDIUM**` , the warning message reads CRITICAL.

- Use the `nagios_spool_retention_time` parameter to specify the number of days of notification history to store on the PEM server. The default value is 7 days.

After modifying parameter values, select **Save** in the upper-right corner of the dialog box.

General
Notification

Email notification

All alerts?
☐ No

<Default>

Low alerts?
☐ No

<Default>

Medium alerts?
☐ No

<Default>

High alerts?
☐ No

<Default>

To configure notifications for an alert, use the fields in the Email notification box to specify the user or user group that will receive an email notification if the alert is triggered at the specified level. Use the drop-down listbox to select a pre-defined group that will be sent a notification if an alert of the selected level is triggered. Please note that you must configure the PEM Server to use an SMTP server to deliver email before PEM can send email notifications.

Trap notification

Send trap?
☐ No
SNMP version
v3
Low alert?
☐ No

Medium alert?
☐ No
High alert?
☐ No

Use the Trap notification options to configure trap notifications for this alert. Note that you must configure the PEM Server to send notifications to an SNMP trap/notification receiver before notifications can be sent.

Nagios notification

Submit passive service check result to Nagios?
☒ Yes

Set "Submit passive service check result to Nagios" to "Yes" to instruct the PEM server to notify Nagios when the alert is triggered or cleared.

Script execution

Execute script?
☐ No
Execute on alert cleared?
☐ No

Execute script on
☐ PEM Server ☒ Monitored Server

Code

Use the fields in the Script execution box to (optionally) define a script that will be executed if an alert is triggered, and to specify details about the script execution.

- Set the Execute script slider to Yes to instruct PEM to execute the provided script if an alert is triggered.
- Set the Execute on alert cleared slider to Yes to instruct PEM to execute the provided script when the situation that triggered the alert has been resolved.
- Use the selector to indicate if the script should execute on the PEM Server or the Monitored Server.
- Provide the script that PEM should execute in the Code field. You can provide a batch/shell script, or SQL code. Within the script you can use the placeholders to replace the following:
 - %AlertName% - the name of the triggered alert.
 - %ObjectName% - the name of the server or agent on which the alert was triggered.
 - %ThresholdValue% - the threshold value reached by the metric when the alert triggered.
 - %CurrentValue% - the current value of the metric that triggered the alert.
 - %CurrentState% - the current state of the alert.
 - %OldState% - the previous state of the alert.
 - %AlertRaisedTime% - the time that the alert was raised, or the most recent time that the alert state was changed.

Fig. 59: The Notification tab

Server Configuration

flapping_detection_state_change	3	
job_failure_notification	☐ False	t/f
job_notification_email_group	default	
job_retention_time	30	days
job_status_change_notification	☐ False	t/f
long_running_transaction_minutes	5	minutes
max_metrics_per_group_chart	16	
nagios_cmd_file_name	/usr/local/nagios/var/rw/nagios.cmd	
nagios_enabled	☒ True	t/f
nagios_medium_alert_as_critical	☐ False	t/f
nagios_spool_retention_time	7	days
probe_log_retention_time	30	days
reminder_notification_interval	24	hours
server_log_retention_time	30	days
show_data_points_on_graph	☐ False	t/f
show_data_tab_on_graph	☐ False	t/f
show_unmanaged_servers	☒ True	t/f

?

Fig. 60: Specify Nagios properties in the Server Configuration dialog

Creating the hosts.cfg and services.cfg file

The `templates.cfg` file (by default, located in `/usr/local/nagios/etc/objects`) specifies the properties of a generic host and generic service. The properties specify the parameters used in the `hosts.cfg` and `services.cfg` files.

In most cases (when PEM is installed in a default configuration), you don't need to modify the `templates.cfg` file before creating the `hosts.cfg` and `services.cfg` files. If necessary, you can modify the `templates.cfg` file to specify alternative values for parameters or to create new templates.

Before modifying the Nagios configuration file, use the following command to create a `hosts.cfg` file that contains information about the PEM hosts that reside on the local system:

```
psql -U postgres -p 5433 -d pem -A -t -c "select pem.create_nagios_host_config('generic-host')"
```

> `/usr/local/nagios/etc/objects/hosts.cfg`

Then, use the following command to create a `services.cfg` file that contains information about the PEM services that reside on the local system:

```
psql -U postgres -p 5433 -d pem -A -t -c "select pem.create_nagios_service_config('generic-service')"
```

> `/usr/local/nagios/etc/objects/services.cfg`

If you want to use a custom `template.cfg` file entry, specify the entry name in place of `generic-host` or `generic-service` in these commands.

Modifying the Nagios configuration file

After creating the `host.cfg` and `services.cfg` files, you must specify their location in the Nagios configuration file (by default, `/usr/local/nagios/etc/nagios.cfg`). Modify the configuration file, adding entries that specify the location of the files:

```
cfg_file=/usr/local/etc/objects/hosts.cfg
```

```
cfg_file=/usr/local/etc/objects/services.cfg
```

You can use the following command to confirm that Nagios is properly configured:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

After confirming that Nagios is configured correctly, restart the Nagios service:

```
/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

2.4 Capacity Manager

PEM's Capacity Manager analyzes collected statistics (metrics) to generate a graph or table that displays the historical usage statistics of an object. It can also project the anticipated usage statistics for an object. You can configure Capacity Manager to collect and analyze metrics for a specific host, server, database, or database object.

You can tailor the content of the Capacity Manager report by choosing:

- Specific metrics to include in the report
- The time range over which the metrics were gathered
- A high or low threshold for the metrics analyzed

You can also specify a start and end date for the Capacity Manager report. If the end date of the report specifies a time in the future, Capacity Manager analyzes the historical usage of the selected object to extrapolate the projected object usage in the future.

To open Capacity Manager, in the PEM client, select **Management>CapacityManager** . The Capacity Manager wizard opens, displaying a tree on the **Metrics** tab.

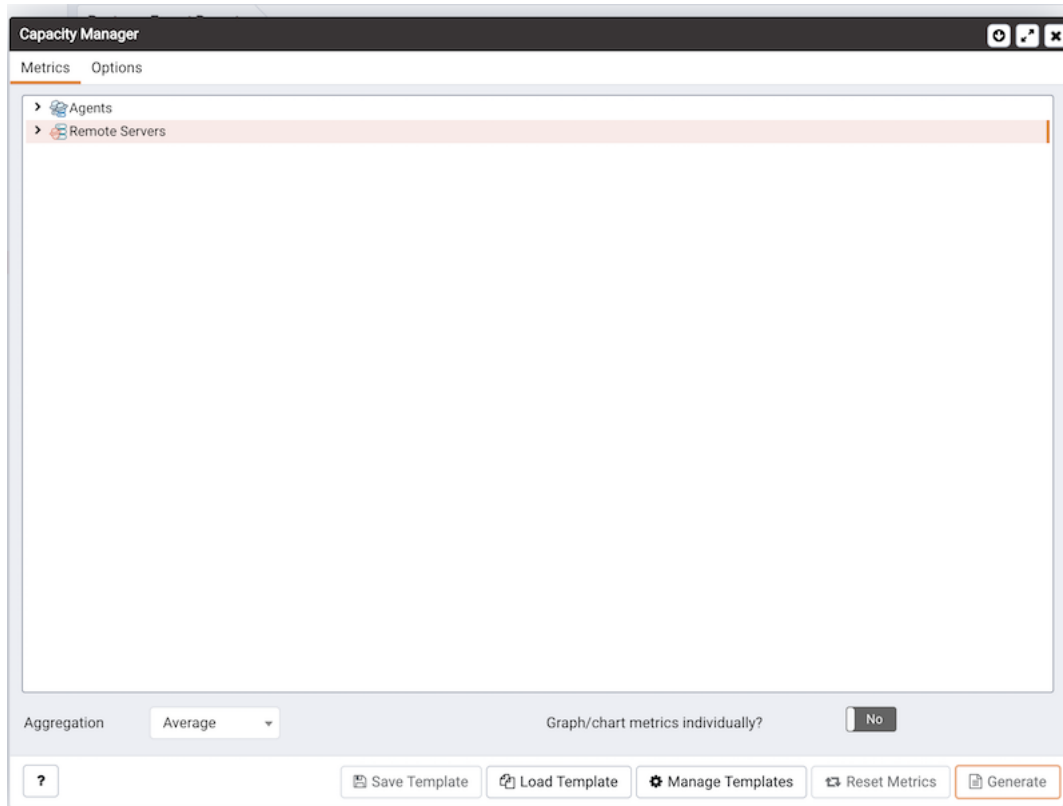


Fig. 61: Capacity Manager dialog

On the **Metrics** tab, expand the tree to review the metrics for the node that you want to analyze. Select the metric to include it in your report.

Capacity Manager uses the aggregation method specified with the **Aggregation** list. Capacity Manager uses the aggregation method to evaluate and plot the metric values. Select from:

- **Average** — Use the average of the values recorded during the time period.
- **Maximum** — Use the maximum value recorded during the time period.
- **Minimum** — Use the minimum value recorded during the time period.
- **First** — Use the first value recorded during the time period.

To remove a metric from the Capacity Manager report, clear the box to the left of the name of a metric.

Move the slider next to **Graph/chart metrics individually?** to **Yes** to produce a separate report for each metric selected on the **Metrics** tab. If the option is set to **No** , all selected metrics are merged into a single graph or table.

Select **Generate** to display the report onscreen (accepting the default configuration options), or use the **Options** tab to customize sampling boundaries, report type, and report destination. The times displayed on the **Options** tab are from the time zone where the PEM client resides.

Use the fields in the **TimePeriod** box to define the boundaries of the Capacity Manager report:

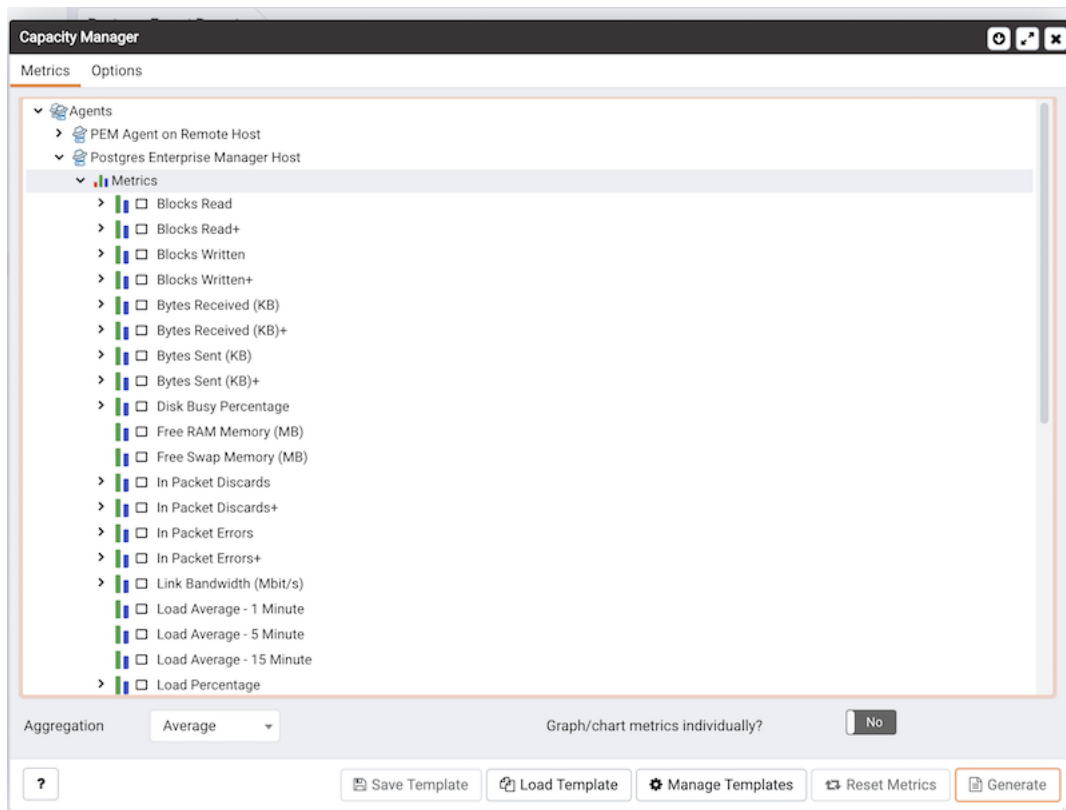
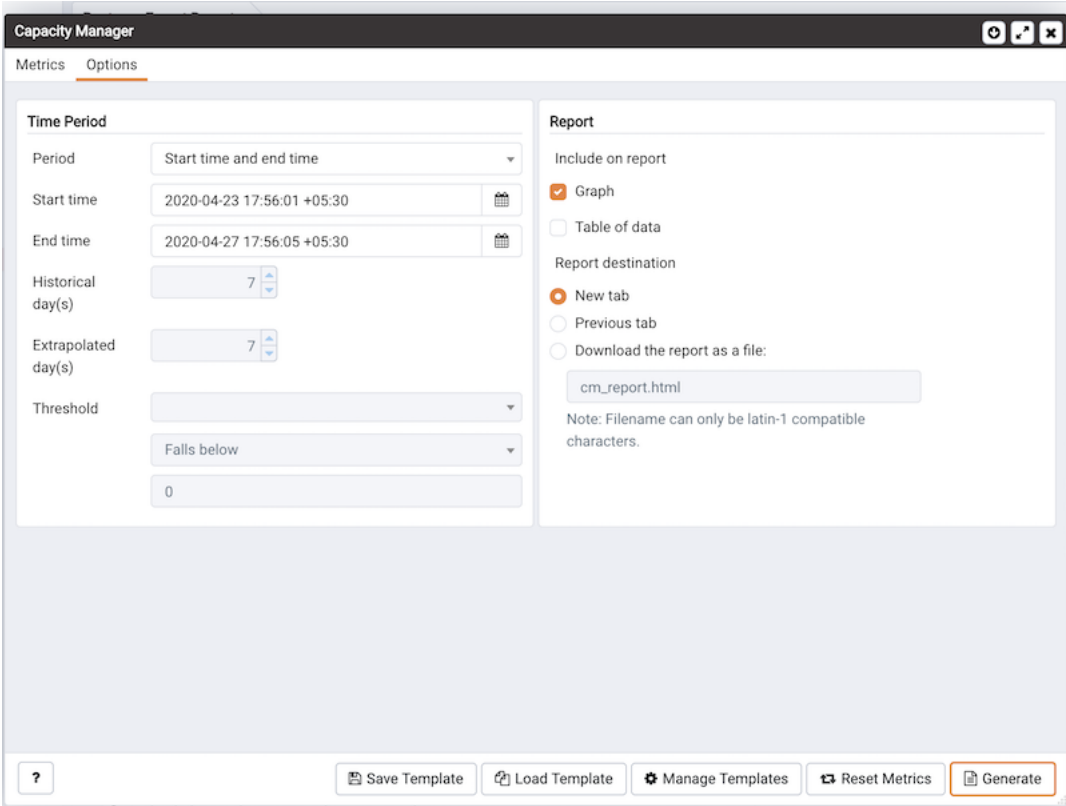


Fig. 62: Capacity Manager dialog - Metrics selection list



The screenshot shows the 'Capacity Manager' interface with the 'Options' tab selected. The interface is divided into two main sections: 'Time Period' and 'Report'.

Time Period Section:

- Period:** A dropdown menu set to 'Start time and end time'.
- Start time:** A text input field containing '2020-04-23 17:56:01 +05:30' with a calendar icon to its right.
- End time:** A text input field containing '2020-04-27 17:56:05 +05:30' with a calendar icon to its right.
- Historical day(s):** A numeric input field with the value '7' and up/down arrow controls.
- Extrapolated day(s):** A numeric input field with the value '7' and up/down arrow controls.
- Threshold:** A dropdown menu set to 'Falls below' with a numeric input field below it containing the value '0'.

Report Section:

- Include on report:** A group of radio buttons where 'Graph' is selected (checked).
- Report destination:** A group of radio buttons where 'New tab' is selected.
- Download the report as a file:** A radio button option that is not selected.
- Filename:** A text input field containing 'cm_report.html'.
- Note:** A text label stating 'Note: Filename can only be latin-1 compatible characters.'

Footer:

A row of buttons at the bottom includes: '?', 'Save Template', 'Load Template', 'Manage Templates', 'Reset Metrics', and 'Generate' (which is highlighted with an orange border).

Fig. 63: Specify the time period, type, and destination of the report

- Use the **Period** list to select the type of time period to use for the report. You can select:

Start time and end time	Specify a start date and an end date/time for the report
Start time and threshold	Specify a start date and time and a threshold to determine the end time and date for the report.
Historical days and extrapolated days	Specify a start date for the report that is a number of days in the past and an end date that is a number of days in the future. This option is useful for report templates that don't specify fixed dates.
Historical days and threshold	Specify a start date that is a number of days in the past, and end it when a threshold value is reached.

After specifying the type of time period for the report, select from other options in the **TimePeriod** box to define the time period for the report:

- Use the date and time selectors next to the **Starttime** field to specify the starting date and time of the sampling period. Or select the number of historical days of data to include in the report. The date and time specified in the **Starttime** field can't be later than the current date/time.

By default, Capacity Manager selects a start time one week prior to the current date and time.

- The end boundary for the report can be a time, a number of days in the future, or the point at which a selected metric reaches a specified threshold value. Use the date and time selectors next to **Endtime** to specify an end boundary for the report. Or select the number of extrapolated days of data to include in the report. The time specified in the **Endtime** field must be later than the time specified in the **Starttime** field.

If you select an end date and time in the future, Capacity Manager uses historical usage information to extrapolate anticipated future usage. Since the projected usage is based on the sampling of historical data, the accuracy of the future usage trend improves with a longer sampling period.

To specify a threshold value, use the list in the **Threshold** field to select a metric, an operator (**Exceeds** or **Fallsbelow**), and a target value for the metric. If you choose to define the end of the report using a threshold, the Capacity Manager report terminates when the value for the selected metric exceeds or falls below the specified value.

The `cm_max_end_date_in_years` configuration parameter defines a default time value for the end boundary of a Capacity Manager report. If you specify a threshold value as the end boundary of a report and the anticipated usage of the boundary isn't met before the maximum time passes, the report terminates at the time specified by the `cm_max_date_in_years` parameter. By default, `cm_max_end_date_in_years` is 5. You can use the Server Configuration dialog box to modify the value of `cm_max_end_date_in_years`.

The fields in the **Report** box specify the report type and destination. Use the **Includeonreport** options to specify the type of report produced by Capacity Manager:

- Select **Graph** to display the report in the form of a line graph in the PEM client window.
- Select **Tableofdata** to display a table containing the report data in the PEM client window.
- Select **Graphandtableofdata** to display both a line graph and a data table in the PEM client window.

Use the **Reportdestination** options to set where to display or save the report:

- Select **Newtab** to display the report on a new tab in the PEM client. You must select **Newtab** to display the first generation of a Capacity Manager report. For subsequent reports, you can select **Previoustab**.
- Select **Previoustab** to reuse a previously opened tab when displaying the report.
- Select **Downloadthereportasfile** and specify a file name to write the report to the specified file.

After you specify the report boundaries and select the type and destination of the Capacity Manager report, select **Generate** to create the report.

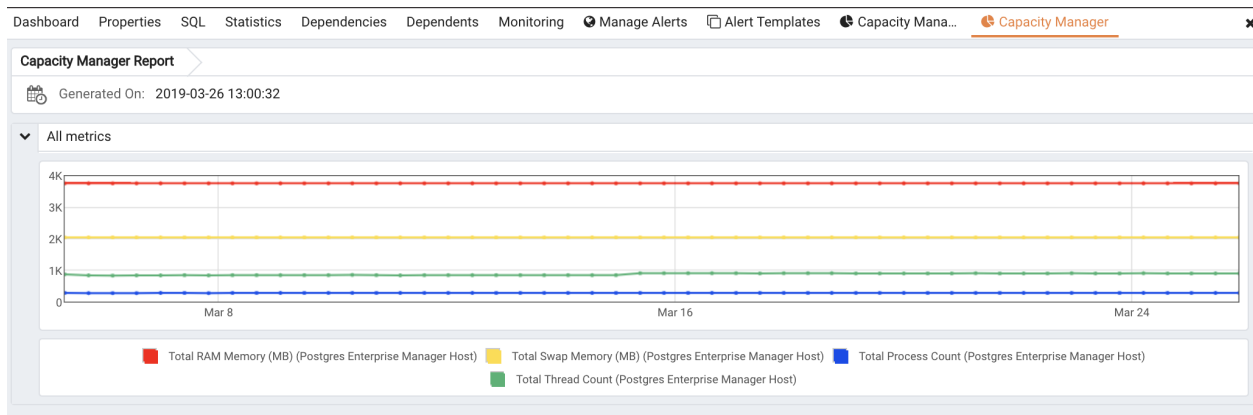


Fig. 64: The Capacity Manager report

Reports saved to file are stored as HTML. You can review a Capacity Manager report with any web browser that supports scalable vector graphics (SVG). Browsers that don't support SVG can't display a Capacity Manager graph and might include unwanted characters.

2.4.1 Capacity Manager templates

After defining a report, you can save the definition as a template for future reports. All PEM users can access Capacity Manager report templates. To save a report definition as a template:

1. Use the **Metrics** and **Options** tabs to define your report.
2. Select **Save** .
 1. In the **SaveTemplate** dialog box, Provide a report name in the **Title** field and select a location to store the template in the tree.
 2. Select **OK** .

When creating a report, you can use **LoadTemplate** to browse and open an existing template. Once opened, you can modify the report definition and save it again, either as a new template or by overwriting the original template.

Select **ManageTemplates** to open a dialog box that lets rename or remove unwanted templates.

2.4.2 Capacity Manager metrics - reference

The Capacity Manager metrics available vary by platform and are subject to change. The available metrics can include the metrics described in the table.

Metric name	Description
# Dead Tuples	The number of dead tuples in the selected table.
# Dead Tuples+	The cumulative number of dead tuples in the selected table.
# Heap Tuples Fetched by Index Scans	The number of heap tuples fetched by index scans.
# Heap Tuples Fetched by Index Scans	The cumulative number of heap tuples fetched by index scans.
# Idle Backends+	The cumulative number of currently idle backend clients.
# Index Scans	The number of index scans performed on the specified object.

continues on next page

Table 12 – continued from previous page

Metric name	Description
# Index Scans+	The cumulative number of index scans performed on the specified object.
# Index Tuples Read	The number of index tuples read.
# Index Tuples Read+	The cumulative number of index tuples read.
# Live Tuples	The number of tuples visible to transactions.
# Live Tuples+	The cumulative number of tuples visible to transactions.
# Pages Estimated by ANALYZE	The number of pages estimated by ANALYZE.
# Pages Estimated by ANALYZE+	The cumulative number of pages estimated by ANALYZE.
# Sequential Scans	The number of sequential scans performed on the specific table.
# Sequential Scans+	The cumulative number of sequential scans performed on the specific table.
# Sequential Scan Tuples	The number of tuples sequentially scanned in the specific table.
# Sequential Scan Tuples+	The cumulative number of tuples sequentially scanned in the specific table.
# Tuples Deleted	The number of tuples deleted.
# Tuples Deleted+	The cumulative number of tuples deleted.
# Tuples Estimated by ANALYZE	The number of live (visible) tuples estimated by ANALYZE.
# Tuples Estimated by ANALYZE+	The cumulative number of live tuples estimated by ANALYZE.
# Tuples HOT Updated	The number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry.
# Tuples HOT Updated+	The cumulative number of tuples HOT updated.
# Tuples Inserted	The number of tuples inserted into the specified table.
# Tuples Inserted+	The cumulative number of tuples inserted into the specified table.
# Tuples Updated	The number of tuples updated in the selected table.
# Tuples Updated+	The cumulative number of tuples updated in the selected table.
Blocks Hit	The number of blocks found in the cache.
Blocks Hit+	The cumulative number of blocks found in the cache.
Blocks Read	The number of blocks read.
Blocks Read+	The cumulative number of blocks read.
Blocks Read from InfiniteCache	The number of blocks read from InfiniteCache.
Blocks Read from InfiniteCache+	The cumulative number of blocks read from InfiniteCache.
Blocks Written	The number of blocks written.
Blocks Written+	The cumulative number of blocks written.
Buffers Allocated	The number of buffers allocated.
Buffers Allocated+	The cumulative number of buffers allocated.
Buffers Written - Backends	The number of buffer blocks written to disk by server processes (processes connected to a client application).
Buffers Written - Backends+	The cumulative number of buffer blocks written to disk by server processes.
Buffers Written - Checkpoint	The number of blocks written to disk by the checkpoint process.
Buffers Written - Checkpoint+	The cumulative number of blocks written to disk by the checkpoint process.
Buffers Written - Cleaning Scan	The number of blocks written to disk by the autovacuum process.
Buffers Written - Cleaning Scan+	The cumulative number of blocks written to disk by the autovacuum process.

continues on next page

Table 12 – continued from previous page

Metric name	Description
Bytes Received (KB)	The number of bytes received from the client (in kilobytes).
Bytes Received (KB)+	The cumulative number of bytes received (in kilobytes).
Bytes Sent (KB)	The number of bytes sent to the client (in kilobytes).
Bytes Sent (KB)+	The cumulative number of bytes sent (in kilobytes).
Checkpoints - Timed	The number of checkpoint operations triggered by the checkpoint interval.
Checkpoints - Timed+	The cumulative number of checkpoint operations triggered by the checkpoint interval.
Checkpoints - Untimed	The number of checkpoint operations triggered by checkpoint size.
Checkpoints - Untimed+	The cumulative number of checkpoint operations triggered by checkpoint size.
Database Size (MB)	The size of the specified database (in megabytes).
Free RAM Memory	The amount of free RAM memory (in megabytes).
Free Swap Memory	The amount of free swap space on disk (in megabytes).
Heap Blocks Hit	The number of heap blocks found in the cache.
Heap Blocks Hit+	The cumulative number of heap blocks found in the cache.
Heap Blocks Read	The number of heap blocks read.
Heap Blocks Read+	The cumulative number of heap blocks read.
Index Blocks Hit	The number of index blocks found in the cache.
Index Blocks Hit+	The cumulative number of index blocks found in the cache.
Index Blocks Read	The number of index blocks read.
Index Blocks Read+	The cumulative number of index blocks read.
Index Size (MB)	The size of the specified index (in megabytes).
In Packets Discards	The number of inbound packets discarded.
In Packets Discards+	The cumulative number of inbound packets discarded.
In Packets Errors	The number of inbound packets that contain errors.
In Packets Errors+	The cumulative number of inbound packets that contain errors.
Link Bandwidth (Mbit/s)	The speed of the network adapter (in megabits per second).
Load Average - 15 Minute	CPU saturation (in percent) - 15 minute sampling average.
Load Average - 1 Minute	CPU saturation (in percent) - 1 minute sampling average.
Load Average - 5 Minute	CPU saturation (in percent) - 5 minute sampling average.
Load Percentage	CPU saturation in percent.
Number of Prepared Transactions+	The cumulative number of prepared transactions.
Number of WAL Files+	The cumulative number of write-ahead log files.
Out Packets Discards	The number of outbound packets discarded.
Out Packets Discards+	The cumulative number of outbound packets discarded.
Out Packets Errors	The number of outbound packets that contain errors.
Out Packets Errors+	The cumulative number of outbound packets that contain errors.
Packets Received	The number of packets received.
Packets Received+	The cumulative number of packets received.
Packets Sent	The number of packets sent.
Packets Sent+	The cumulative number of packets sent.
Size (MB)	The total size of the disk (in megabytes).
Size of Indexes (MB)	The size of indexes on the specified table (in megabytes).
Space Available (MB)	The current disk space available (in megabytes).
Space Used (MB)	The current disk space used (in megabytes).
Table Size (MB)	The size of the specified table (in megabytes).
Tablespace Size (MB)	The size of the specified tablespace (in megabytes).
Temp Buffers (MB)	The size of temporary buffers (in megabytes).
Toast Blocks Hit	The number of TOAST blocks found in the cache.
Toast Blocks Hit+	The cumulative number of TOAST blocks found in the cache.

continues on next page

Table 12 – continued from previous page

Metric name	Description
Toast Blocks Read	The number of TOAST blocks read.
Toast Blocks Read+	The cumulative number of TOAST blocks read.
Total RAM Memory	The total amount of RAM memory on the system (in megabytes).
Total Swap Memory	The total amount of swap space on the system (in megabytes).
Total Table Size w/Indexes and Toast	The total size of the specified table (including indexes and associated oversized attributes).
Transactions Aborted	The number of aborted transactions.
Transactions Aborted+	The cumulative number of aborted transactions.
Transactions Committed	The number of committed transactions.
Transactions Committed+	The cumulative number of committed transactions.
Tuples Deleted	The number of tuples deleted from the specified table.
Tuples Deleted+	The cumulative number of tuples deleted from the specified table.
Tuples Estimated by ANALYZE	The number of visible tuples in the specified table.
Tuples Estimated by ANALYZE+	The cumulative number of visible tuples in the specified table.
Tuples Fetched	The number of tuples fetched from the specified table.
Tuples Fetched+	The cumulative number of tuples fetched from the specified table.
Tuples HOT Updated	The number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry.
Tuples HOT Updated+	The cumulative number of tuples HOT updated. In a HOT update, the new tuple resides in the same block as the original tuple and the tuples share an index entry.
Tuples Inserted	The number of tuples inserted into the specified table.
Tuples Inserted+	The cumulative number of tuples inserted into the specified table.
Tuples Returned	The number of tuples returned in result sets.
Tuples Returned+	The cumulative number of tuples returned in result sets.
Tuples Updated	The number of tuples updated in the specified table.
Tuples Updated+	The cumulative number of tuples updated in the specified table.
WAL Segment Size (MB)	The segment size of the write-ahead log (in megabytes).

Note: The + following the name of a metric means that the data for the metric is gathered cumulatively. Metrics that aren't followed by the '+' sign are collected as a point-in-time value.

2.5 Audit Manager

You can use the PEM Audit Manager to simplify audit log configuration for EDB Postgres Advanced Server instances. With Audit Manager, you can configure logging attributes such as:

- How often PEM collects log files
- The type of database activities that are included in the log files
- How often to rotate log files and when

Audit logs can include the following activities:

- All connections made to the database instance
- Failed connection attempts
- Disconnections from the database instance

Save Template

Title

template_capacity_memory

Location

▼  Templates

★ template_transaction_stats

✕ Cancel

 OK

Fig. 65: Saving a Capacity Manager Template

- All queries (SELECT statements)
- All DML statements (INSERT, UPDATE, DELETE)
- All DDL statements (for example, CREATE, DROP, ALTER)

Once the audit logs are stored on the PEM server, you can use the Audit Log dashboard to review the information in an easy-to-read form. The Audit Log dashboard allows you to filter the log file by:

- Timestamp range (when an activity occurred)
- The database on which the activity occurred
- The user performing the activity
- The type of command being invoked

2.5.1 Setting the EDB Postgres Advanced Server instance service ID

To configure logging for an EDB Postgres Advanced Server instance, the server must be a PEM-managed server with a bound agent, and the server registration must include the name of a service script. When registering a new server, include the service name in the **ServiceID** field on the **Advanced** tab of the New Server dialog box.

Before adding a service name to an existing, registered and connected server, you must disconnect the server:

1. Right-click the server name, and select **Disconnectserver** from the context menu.
2. Right-click the server name and select **Properties** from the context menu.
3. Select the **Advanced** tab, and add a service name to the **ServiceID** field.

The screenshot shows a configuration window titled 'EPAS_12'. The 'Advanced' tab is selected, showing various configuration fields. The 'Service ID' field contains the text 'edb-as-12'. Other visible fields include 'Host address', 'DB restriction', 'Password file', 'EFM cluster name', 'EFM installation path', and 'Connection timeout (seconds)' which is set to '10'. The window has a standard OS-style title bar and a bottom bar with 'Cancel', 'Reset', and 'Save' buttons.

Fig. 66: The Service ID of the Advanced Server instance

The Service ID field allows the PEM server to stop and start the service.

- The name of the EDB Postgres Advanced Server 11 service script is `edb-as-12`.
- The name of the EDB Postgres Advanced Server 11 service script is `edb-as-11`.

- The name of the EDB Postgres Advanced Server 10 service script is `edb-as-10` .
- The name of the EDB Postgres Advanced Server 9.6 service script is `edb-as-9.6` .
- The name of the EDB Postgres Advanced Server 9.5 (or prior) service script is `ppas-9.x` , where `x` specifies the version.
- The name of the PostgreSQL 9.6 service script is `postgresql-11` .
- The name of the PostgreSQL 9.6 service script is `postgresql-10` .
- The name of the PostgreSQL 9.6 service script is `postgresql-9.6` .

2.5.2 Setting the EDB Audit Configuration probe

Before configuring audit logging of EDB Postgres Advanced Server servers, you must ensure that the EDB Audit Configuration probe is enabled. To open the **ManageProbes** tab and check the status of the probe, right-click the name of a registered EDB Postgres Advanced Server server in the tree, and select **Management>ManageProbes** .

Ensure that the Enabled column in the Probe Configuration dialog box is set to **Yes** for the EDB Audit Configuration probe.

Quick Links

 Manage Custom Probes

 Copy Probes

 Help

Probes

Probe name	Execution Frequency			Enabled?		Data Retention	
	Default?	Minutes	Seconds	Default?	Probe Enable?	Default?	Days
Background Writer Statistics	Yes	5	0	Yes	Yes	Yes	180
Blocked Session Information	Yes	5	0	Yes	Yes	Yes	180
Data and Log File Analysis	Yes	0	10	Yes	Yes	Yes	180
Database Frozen XID	Yes	720	0	Yes	Yes	Yes	180
Database Size	Yes	30	0	Yes	Yes	Yes	180
Database Statistics	Yes	30	0	Yes	Yes	Yes	90
EDB Audit Configuration	Yes	2	0	Yes	Yes	Yes	180
Failover Manager Cluster Info	Yes	5	0	Yes	No	Yes	7
Failover Manager Node Status	Yes	5	0	Yes	No	Yes	

Fig. 67: The EDB Audit Configuration probe

If EDB Audit Configuration isn't enabled, use the **Enabled?** switch on the **ManageProbes** tab to enable it.

2.5.3 Configuring audit logging with the Audit Manager

To open the Audit Manager wizard, select **Management>AuditManager** . The Audit Manager - Welcome dialog box opens.

Select **Next** to continue.

Use the Select Servers tree to specify the servers to which to apply the auditing configuration. To make a server available in the tree, you must provide the Service ID on the **Advanced** tab of the Create – Server dialog box when registering a server for monitoring by PEM. Only EDB Postgres Advanced Server supports auditing. PostgreSQL servers don't appear in the tree.

Select **Next** to continue.

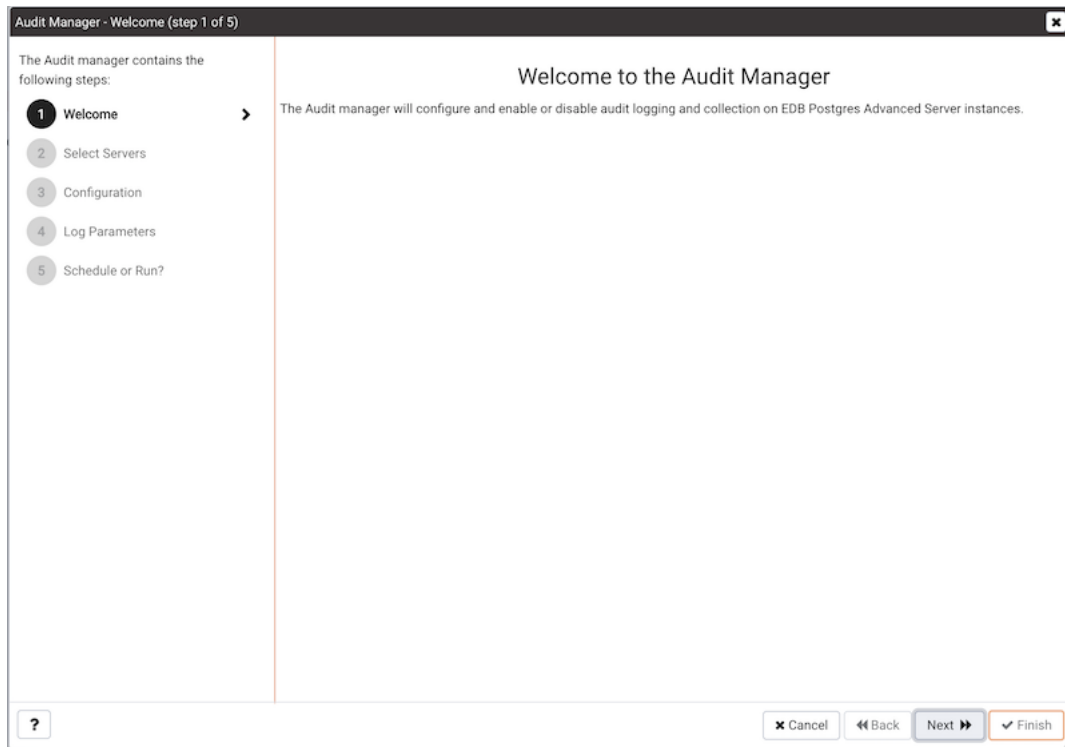


Fig. 68: The Audit Manager Welcome dialog

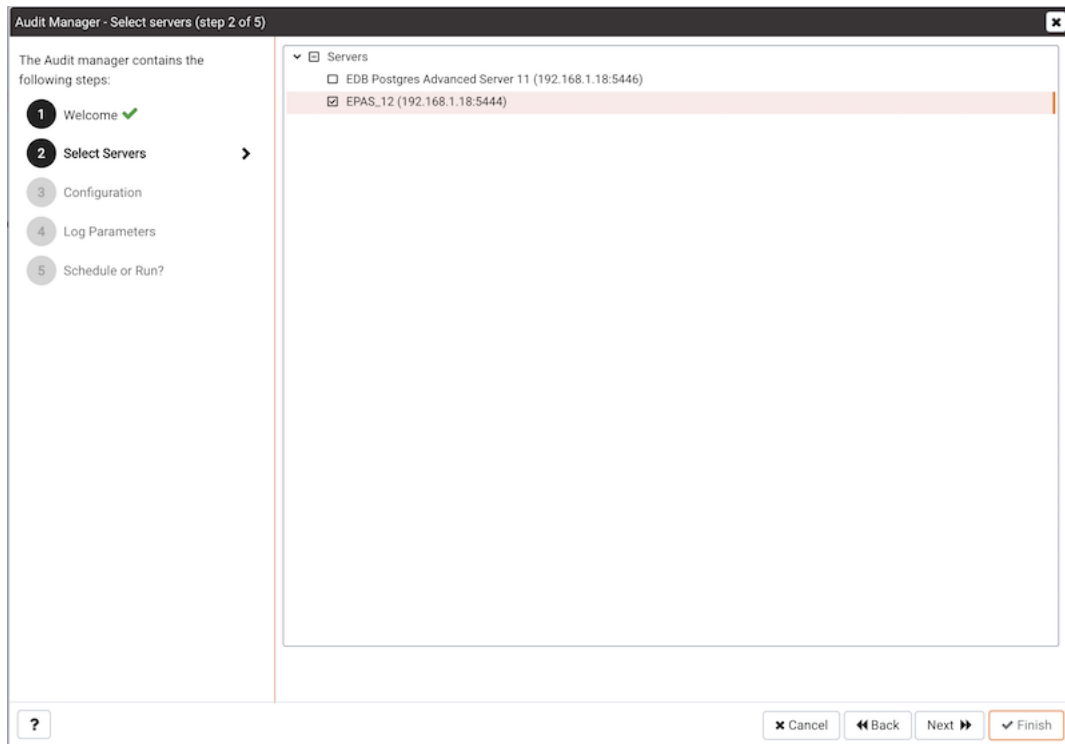


Fig. 69: Select the servers you wish to configure for auditing

The Auditing Parameters Configuration dialog box lets you enable or disable auditing and choose how often log records are collected into PEM.

Fig. 70: The Auditing Parameters Configuration dialog

Use the Auditing Parameters Configuration dialog box to specify auditing preferences:

- Use the **Auditing** switch to enable or disable auditing on the specified servers.
- Use the **Auditdestination** list to select a destination for the audit logs. Select **File** or **Syslog** . This feature is supported only on EDB Postgres Advanced Server 10 and later releases.
- Use the **ImportlogstoPEM** switch to periodically import log records from each server to the PEM Server. Set the switch to **Yes** to import log files. The default is **No** .
- Use the **Collectionfrequency** list to specify how often PEM collects log records from monitored servers when log collection is enabled.
- Use the **Logformat** list to select the raw log format to write on each server. If log collection is enabled, the PEM server uses CSV format.
- Use the **Filename** field to specify the format used when generating log file names. By default, the format is set to `audit-%Y-%m-%d_%H%M%S` where:

`audit` is the file name specified in the Audit Directory Name field `Y` is the year that the log was stored `m` is the month that the log was stored `d` is the day that the log was stored `H` is the hour that the log was stored `M` is the minute that the log was stored `S` is the second that the log was stored

- Select **ChangeLogDirectoryforselectedservers?** and use the **AuditDirectoryName** field to specify a directory name to which to write the audit logs. The directory resides beneath the data directory on the PEM server.
- Use the **Logdirectory** box to specify information about the directory in which the log files are saved:

- Move the ****Changelogdirectoryforselectedservers?**** switch to ****Yes**** to enable the ****Directoryname**** field.
- Use the ****Directoryname**** field to specify the name of the directory on each server to write audit logs to. The directory specified is created as a subdirectory of the data directory on the server.

Select **Next** to continue.

The Audit log configuration dialog box is available only if you enabled auditing on the Auditing Parameters Configuration dialog box.

The screenshot shows the 'Audit Manager - Audit log configuration (step 4 of 5)' dialog box. On the left, a progress bar indicates the current step is '4 Log Parameters'. The main area contains several configuration sections:

- Connection attempts:** A switch set to 'None' (checked) with options 'All' and 'Failed'.
- Disconnection attempts:** A switch set to 'None' (checked) with an option 'All'.
- Log statements:** A text field containing 'x Select x Error x DML x DDL' with a red box highlighting it.
- Audit tag:** An empty text field.
- Log rotation:** A section with 'Enable?' set to 'No', and fields for 'Day' (None), 'Size(MB)' (0), and 'Time(seconds)' (0).

At the bottom right, there are buttons for 'Cancel', 'Back', 'Next', and 'Finish' (highlighted with a red box).

Fig. 71: The Audit Log Configuration dialog

Use the Audit Log Configuration dialog box to specify log configuration details to apply to each server:

- Use the **Connectionattempts** switch to specify whether to log connection attempts. Specify:
 - None** to disable connection logging
 - All** to indicate that all connection attempts are logged
 - Failed** to log any connection attempts that fail
- Use the **Disconnectionattempts** switch to specify whether to log disconnections. Specify:
 - None** if you don't want to log disconnections
 - All** to enable disconnection logging
- Use the **Logstatements** field to specify the statement types that are logged. Select from:
 - Select** — Log all statements that include the SELECT keyword.
 - Error** — Log all statements that result in an error.

DML — Log all DML (data modification language) statements.

DDL — Log all DDL (data definition language) statements, that is, those that add, delete or alter data. Select **SelectAll** to select all statement types.

Select **UnselectAll** to clear all statement types.

- Use the **Audittag** field to specify a tracking tag for the collected logs. Audit tagging functionality is available only for supported versions of EDB Postgres Advanced Server.
- Use the **Logrotation** box to specify how to manage the log files on each server:

Use the **Enable?** switch to rotate log files. Use a new log file periodically to prevent a single file from becoming too large.

Use the **Day** list to select days on which to rotate the log file.

Use the **Size(MB)** field to specify a size in megabytes at which to rotate the log file.

Use the **Time(seconds)** field to specify the number of seconds between log file rotations.

Select **Next** to continue:

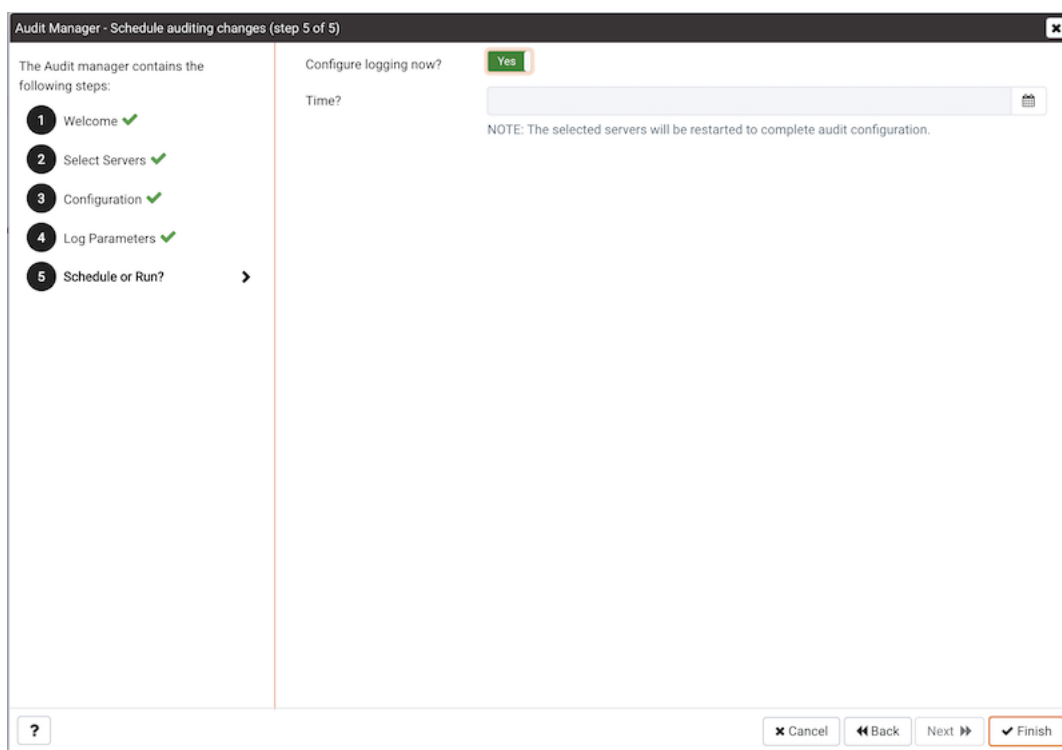


Fig. 72: The Schedule Auditing Changes dialog

Use the Schedule Auditing Changes dialog box to determine when auditing configuration changes take effect.

- Select **Configureloggingnow?** if you want the auditing configuration changes to take place immediately. The affected database servers restart so the auditing changes can take effect.
- Use the **Time?** selector to schedule the auditing configuration changes to take place at some point in the future. Select the desired date and time from the lists. The affected database servers restart at the specified date and time to put the auditing changes into effect.

Select **Finish** to complete the auditing configuration process.

Audit Manager schedules a job to apply the configuration to each server. The job consists of two tasks: one to update the audit logging configuration on the server and one to restart the server with the new configuration.

You can use the **ScheduledTasks** tab to review a list of scheduled jobs. To open the **ScheduledTasks** tab, select the name of a server or agent and select **Management>ScheduledTasks**.

2.5.4 Viewing the log with the Audit Log dashboard

Use the Audit Log dashboard to view the audit log from EDB Postgres Advanced Server database instances.

To open the Audit Log dashboard, right-click a server or agent node and select **AuditLogAnalysis** from the **Dashboards** menu. You can also open the Audit Log dashboard by navigating through the **Management>Dashboards** menu.

Id	Server	Timestamp	User Name	Database Name	Process ID	Session ID	Transaction ID	Connection From	Command	Message
7879	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26391	5ea902cf.6717	0	127.0.0.1:46780	idle	disconnection: session time: 0:00:00.014 user=enterprisedb database=postgres host=127.0.0.1 port=46780
7878	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26391	5ea902cf.6717	0	127.0.0.1:46780	idle	statement: SELECT setting FROM pg_settings WHERE name = 'edb_audit_rotation_seconds'
7877	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26391	5ea902cf.6717	0	127.0.0.1:46780	idle	statement: SELECT version();
7876	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26391	5ea902cf.6717	0	127.0.0.1:46780	authentication	connection authorized: user=enterprisedb database=postgres
7875	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26388	5ea902cf.6714	0	127.0.0.1:46774	idle	disconnection: session time: 0:00:00.007 user=enterprisedb database=postgres host=127.0.0.1 port=46774
7874	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26388	5ea902cf.6714	0	127.0.0.1:46774	authentication	connection authorized: user=enterprisedb database=postgres
7873	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26384	5ea902cf.6710	0	127.0.0.1:46766	idle	disconnection: session time: 0:00:00.029 user=enterprisedb database=postgres host=127.0.0.1 port=46766
7872	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26384	5ea902cf.6710	0	127.0.0.1:46766	idle	statement: SELECT setting FROM pg_settings WHERE name='log_temp_files'
7871	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26384	5ea902cf.6710	0	127.0.0.1:46766	idle	statement: SELECT setting FROM pg_settings WHERE name='log_autovacuum_min_duration'
7870	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26384	5ea902cf.6710	0	127.0.0.1:46766	idle	statement: SELECT setting FROM pg_settings WHERE name='log_min_duration_statement'
7869	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26384	5ea902cf.6710	0	127.0.0.1:46766	idle	statement: SELECT (setting::int/(24*60))::int FROM pg_settings WHERE name = 'log_rotation_age'
7868	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26384	5ea902cf.6710	0	127.0.0.1:46766	idle	statement: SELECT (setting::int/1024)::int FROM pg_settings WHERE name = 'log_rotation_size'
7867	EPAS_12	29/04/2020, 10:00:07	enterprisedb	postgres	26384	5ea902cf.6710	0	127.0.0.1:46766	idle	statement: SELECT upper(setting) FROM pg_settings WHERE name='syslog_facility';

Fig. 73: The Audit Log dashboard

The Audit Log dashboard displays the audit records in reverse chronological order.

You can use filtering to limit the number of audit records that are displayed. Select **ShowFilters** to expose the filters panel.

Use the filters panel to provide selection criteria for the audit records you want to display.

- Use the **Start** field to specify a start date for the report.
- Use the **End** field to specify an end date for the report.
- Use the **User** field to display only those entries where the activity was started by the given Postgres user.
- Use the **Database** field to display only those entries where the activity was issued on the given database.

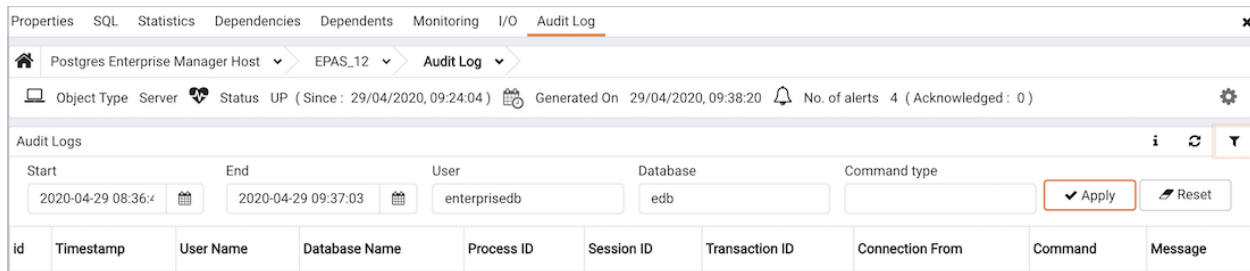


Fig. 74: The Audit Log dashboard filters panel

- Use the **Command type** field to display only those entries where the activity was of the given type. Command types you can specify are **idle**, **authentication**, and **SELECT**. (For viewing SQL statements from user applications, specify the idle command type.)

Select **Filter** to apply the filtering criteria to the log entries.

2.6 Log Manager

You can use the PEM Log Manager to simplify server log configuration for Postgres instances. With Log Manager, you can modify all of your server log parameters including:

- Where log files are written
- How often log files are written
- The type of information written to log files
- The format of log file entries
- Log rotation properties

To configure logging for a Postgres instance, you must register the server as a PEM-managed server, and the registration information must include the name of a service script.

To open Log Manager, in the PEM client, select **Management > Log Manager**. The wizard opens and welcome message appears.

Select **Next**.

The Server selection dialog box displays a list of the server connections monitored by PEM. Select the check boxes next to the names of servers to which you want the Log Manager wizard to apply the specified configuration. Log Manager is disabled for any server displaying a red exclamation mark to the left of its name in the Server selection tree. A server might not be enabled for several reasons:

- Log Manager can configure only a server that specifies a service ID on the **Advanced** tab of the Properties dialog box.

To provide a service ID: 1. In the tree, right-click the server name and select **Disconnect Server** from the context menu. 1. If prompted, provide a password. 1. From the context menu for the server, select **Properties**. 1. On the **Advanced** tab, enter the name of the service in the **ServiceID** field. 1. Select **Save**.

- If the PEM agent bound to the server doesn't have enough privileges to restart the server, the server is disabled.
- If the PEM agent bound to the server is an older version than the associated PEM server, the server is disabled.

Select **Next**.

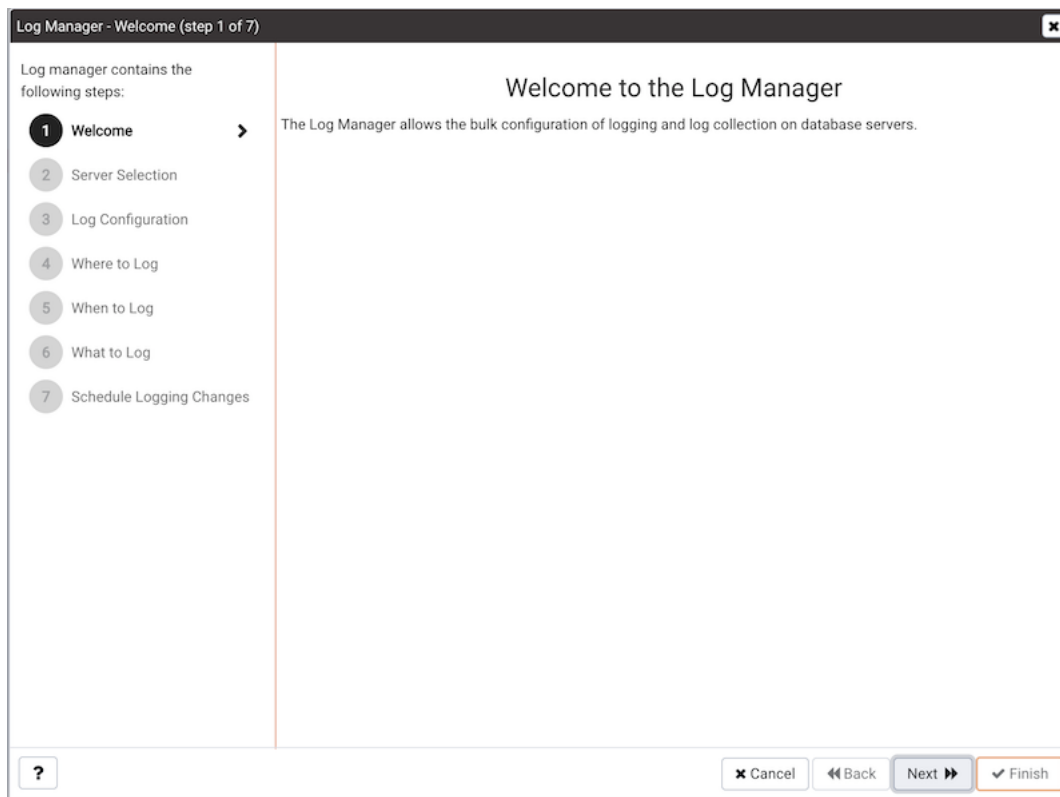


Fig. 75: The Log Manager welcome dialog

Use the options on the Log configuration dialog box to specify how often to import log files to PEM and to specify log rotation details.

Options in the **ImportLogs** box specify how often to import the log files to PEM:

- Use the switch next to **ImportlogstoPEM** to specify whether to import the log files to PEM and to display them on the Server Log Analysis dashboard.
- Use the **ImportFrequency** list to specify how often to import log files to PEM.

Use the fields in the **Logrotationconfiguration** box to specify the maximum length (lifespan or size) of a log file:

- Use the **RotationSize** field to specify the maximum size of a log file in megabytes. The default value is 10 MB. Set to 0 if you don't want to place a limit on the maximum size of a log file.
- Use the **RotationTime** field to specify the number of whole days to store in each log file. The default value is 1 day.

Use the **TruncationonRotation** switch to specify server behavior for time-based log file rotation:

- Select **On** to overwrite any existing log file that has the same name as a new file.
- Select **Off** to append any new log file entries to an existing log file with the same name as a new log file. This is the default behavior.

Select **Next**.

Use the Where to Log dialog box to specify where to write log files:

- Select an option from the **LogDestination** box to specify a destination for the server log output: - Set the **stderr** switch to **Yes** to write log files to stderr. - Set the **csvlog** switch to **Yes** to write log files in a comma-separated

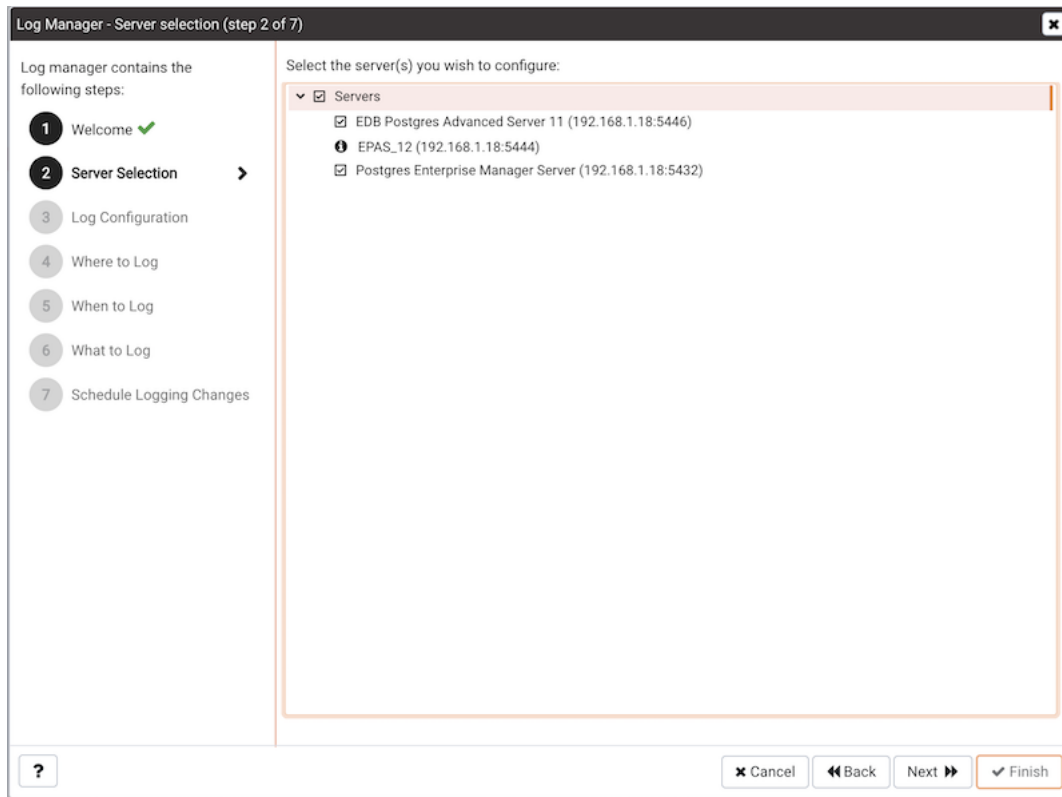


Fig. 76: The Log Manager Server selection dialog

value format. This option is enabled and not editable if you select **ImportlogstoPEM** on the Schedule dialog box. If you aren't importing server log files to PEM, you can edit this option. - Set the **syslog** switch to **Yes** to write log files to the system log files. - On Windows, set the **eventlog** switch to **Yes** to write log files to the event log.

- Use the options in the **Logcollection** box to specify your collection preferences: - Set the **LogCollector** switch to **Enable** to redirect captured log messages (directed to stderr) into log files. - Set the **LogSilentMode** switch to **Enable** to run the server silently in the background, disassociated from the controlling terminal.
- Use options in the **LogDirectory** box to specify log file location preferences: - Set the **Changelogdirectoryfors-electedservers?** switch to **Yes** to maintain each set of log files in a separate directory. - Use the **Directoryname** field to specify the directory to which to write log files. The directory resides beneath the **pg_log** directory under the installation directory of the monitored server.
- Use the **LogFileName** field to specify a format for the log file name. If set to **Default**, the format is **enterprisedb-%Y-%m-%d_%H%M%S**, where: - **enterprisedb** is the file name prefix - **Y** is the year that the log was stored - **m** is the month that the log was store - **d** is the day that the log was stored - **H** is the hour that the log was stored - **M** is the minute that the log was store - **S** is the second that the log was stored
- When logging to syslog is enabled:

- Use the ****SyslogFacility**** **list** to specify the syslog facility to use. - Use the ****SyslogIdent**** field to specify the program name that identifies EDB Postgres Advanced Server entries **in** system logs.

Select **Next**.

Use the When to Log dialog box to specify the events that initiate a log file entry. The severity levels in order of severity,

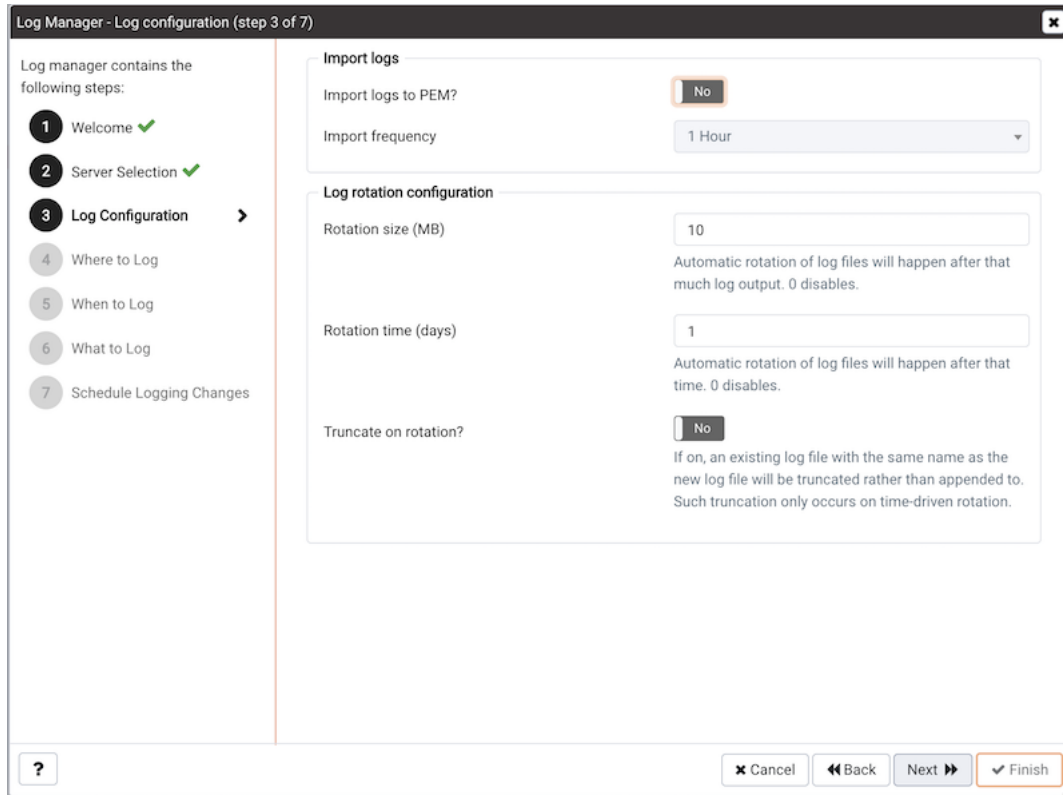


Fig. 77: The Log Manager Log configuration dialog

from most severe to least severe, are:

- ****panic**** — Errors that cause **all** database sessions to abort.
 - ****fatal**** — Errors that cause a session to abort.
 - ****log**** — Information messages of interest to administrators.
 - ****error**** — Errors that cause a command to abort.
 - ****warning**** — Error conditions **in** which a command completes but might **not** perform **as** expected.
 - ****notice**** — Items of interest to users. This **is** the default.
 - ****info**** — Information implicitly requested by the user.
 - ****debug5**** through ****debug1**** — Detailed debugging information useful to developers.
- Use the **Clientminmessages** list to specify the lowest severity level of message sent to the client application.
 - Use the **Logminmessages** list to specify the lowest severity level to write to the server log.

Log Manager - Where to log (step 4 of 7)

Log manager contains the following steps:

- Welcome ✓
- Server Selection ✓
- Log Configuration ✓
- Where to Log >
- When to Log
- What to Log
- Schedule Logging Changes

Log destination

stderr? ☒ Yes ☐ No csvlog? ☐ No ☒ No

syslog? ☐ No ☒ No eventlog? ☐ No ☒ No

Log collection

Log collector ☒ Enable ☐ Disable Log silent mode ☐ Disable ☒ Disable

Log directory

Change log directory for selected servers? ☐ No ☒ Yes

Directory name

Log file name

Syslog facility When logging to syslog is enabled, this parameter determines the syslog "facility" to be used.

Syslog ident When logging to syslog is enabled, this parameter determines the program name used to identify PostgreSQL messages in syslog logs.

? Cancel Back Next Finish

Fig. 78: The Where to Log dialog

- By default, when an error message is written to the server log, the text of the SQL statement that initiated the log entry isn't included. Use the **Logminerrorstatement** list to specify a severity level that triggers SQL statement logging. If a message is of the specified severity or higher, the SQL statement that produced the message is written to the server log.
- Use the **Logmindurationstatement** list to specify a statement duration, in milliseconds. Any statements that exceed the specified number of milliseconds are written to the server log. A value of **-1** disables all duration-based logging. A value of **0** logs all statements and their duration.
- Use the **Logtempfiles** field to specify a file size, in kilobytes. When a temporary file reaches the specified size, it's logged. Specify a value of **-1** (the default) to disable this functionality.
- Use the **LogautoVacuumminduration** field to specify a time length, in milliseconds. If auto-vacuuming exceeds the length of time specified, the activity is logged. Specify a value of **-1** (the default) to disable this functionality.

Select **Next**.

Use the fields on the What to Log dialog box to specify log entry options that are useful for debugging and auditing.

The switches in the **Debugoptions** box instruct the server to include information in the log files related to query execution that might be of interest to a developer:

- Set the **Parsetree** switch to **Yes** to include the parse tree in the log file.
- Set the **Rewriteroutput** switch to **Yes** to include query rewriter output in the log file.
- Set the **Executionplan** switch to **Yes** to include the execution plan for each executed query in the log file.

When the **IndentDebugOptionsOutputinLog** switch is set to **Yes**, the server indents each line that contains a parse tree entry, a query rewriter entry, or query execution plan entry. While indentation makes the resulting log file more

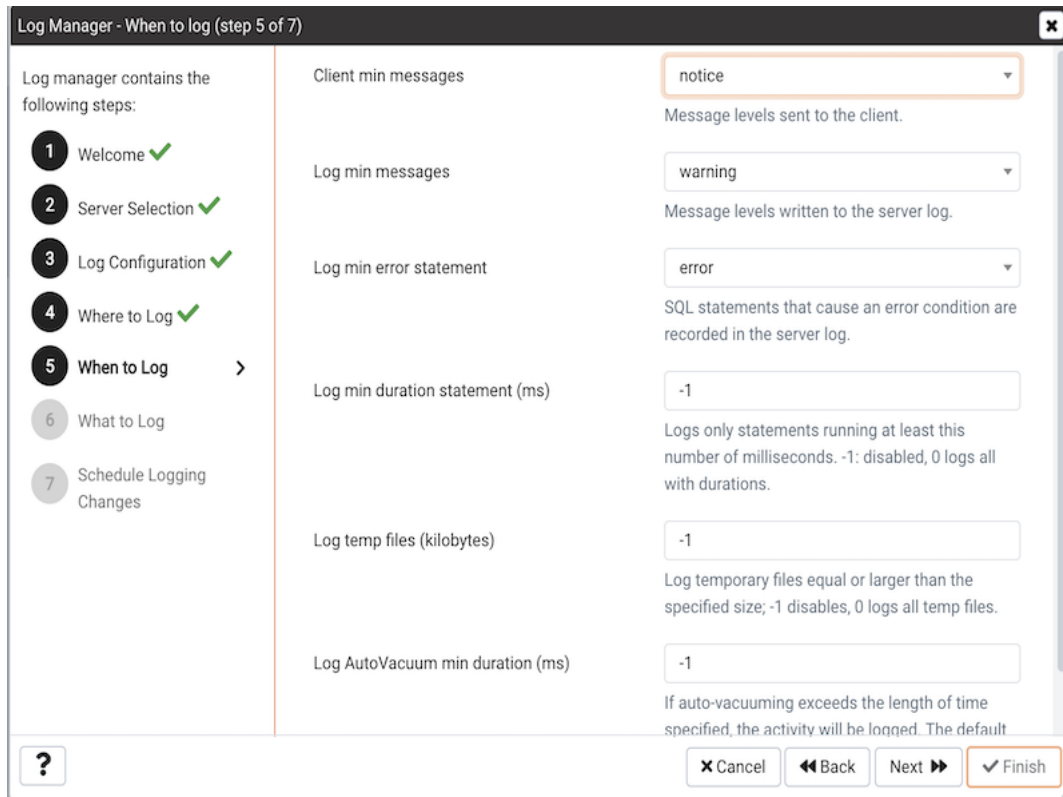


Fig. 79: The Log Manager When to Log dialog

readable, it results in a longer log file.

Use the switches in the **GeneralOptions** box to include auditing information in the log file:

- Set the **Checkpoints** switch to **Yes** to include checkpoints and restartpoints in the server log.
- Set the **Connections** switch to **Yes** to include each attempted connection to the server as well as successfully authenticated connections in the server log.
- Set the **Disconnections** switch to **Yes** to include a server log entry for each terminated session that provides the session information and session duration.
- Set the **Duration** switch to **Yes** to include the amount of time required to execute each logged statement in the server log.
- Set the **Hostname** switch to **Yes** to include both the IP address and host name in each server log entry. By default, only the IP address is logged. This might cause a performance penalty.
- Set the **LockWaits** switch to **Yes** to write a log entry for any session that waits longer than the time specified in the `deadlock_timeout` parameter to acquire a lock. This is useful when trying to determine if lock waits are the cause of poor performance.

Use the **Errorverbosity** list to specify the detail written to each entry in the server log:

- Select **default** to include the error message, DETAIL, HINT, QUERY, and CONTEXT in each server log entry.
- Select **terse** to log only the error message.
- Select **verbose** to include the error message, the DETAIL, HINT, QUERY, and CONTEXT error information, SQLSTATE error code and source code file name, the function name, and the line number that generated the error.

Log Manager - What to log (step 6 of 7)

Log manager contains the following steps:

- Welcome ✓
- Server Selection ✓
- Log Configuration ✓
- Where to Log ✓
- When to Log ✓
- What to Log** ▶
- Schedule Logging Changes

Debug options

Parse tree? ☐ No Rewriter output? ☐ No

Execution plan? ☐ No

Indent debug options output in log? ☒ Yes

General options

Checkpoints? ☐ No Connections? ☐ No

Disconnections? ☐ No Duration? ☐ No

Hostname? ☐ No Lock waits? ☐ No

Error verbosity: default
Level of message detail written to the server log.

Prefix string: %t
Use the Prefix String field to specify a printf-style string that is written at the beginning of each log file entry.

Statements: none
Controls which SQL statements are logged.

Buttons: ? Cancel Back Next Finish

Fig. 80: The Log Manager What to Log dialog

Use the **Prefixstring** field to specify a printf-style string that is written at the beginning of each log file entry. For information about the options supported, see the `log_line_prefix` documentation in the

[Postgres core documentation](#).

Use the **Statements** list to specify the SQL statements to include in the server log:

- Specify **none** (the default) to disable logging of SQL statements.
- Specify **ddl** to instruct to log ddl (data definition language) statements, such as CREATE, ALTER, and DROP.
- Specify **mod** to log all ddl statements as well as all dml (data modification language) statements, such as INSERT, UPDATE, DELETE, TRUNCATE, and COPY FROM.
- Specify **all** to log all SQL statements.

Select **Next**.

Use options on the Schedule logging changes dialog box to specify when logging applies configuration changes:

- Set the **Configureloggingnow** switch to **Yes** to enable your configuration preferences. The server restarts when you complete the Log Manager wizard.
- Set **Configureloggingnow** to **No** to use the **Scheduleitforsomeothertime** calendar selector to specify a convenient time to apply logging configuration preferences and for the server to restart.

When you apply the configuration changes specified by the Log Manager wizard, the server restarts, temporarily interrupting use of the database server for users.

Select **Finish** to exit the wizard. Either restart the server or schedule the server restart for the time specified on the scheduling dialog box.

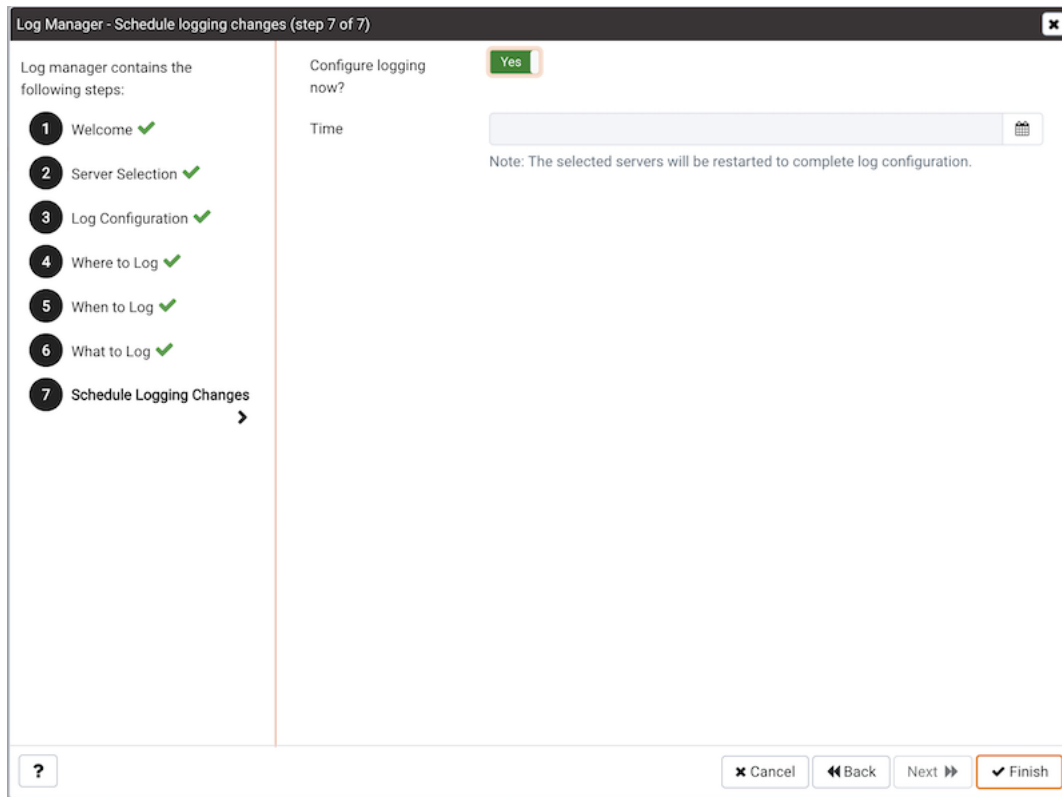


Fig. 81: The Schedule Logging Changes dialog

2.6.1 Reviewing the Server Log Analysis dashboard

After invoking the Log Manager wizard and importing your log files to PEM, you can use the Server Log Analysis dashboard to review the log files for a selected server. To open the Server Log Analysis dashboard, right-click the name of a monitored server in the PEM client tree and select **Dashboards>ServerLogAnalysis**.

The header information on the Server Log Analysis dashboard displays the date and time that the server was started, the date and time that the page was last updated, and the current number of triggered alerts.

Entries in the Server Log table appear in chronological order, with the most recent log entries first. Use the scroll bars to navigate through the log entries or to view columns that are off of the display.

Headings at the top of the server log table identify the information stored in each column. Hover over a column heading to view a tooltip that contains a description of the content of each column.

You can use filtering to limit the number of server log records that are displayed. Select **ShowFilters** to open the filters panel and define a filter.

Use the fields in the **filterdefinition** box to describe the selection criteria for selecting a subset of a report to display:

- Use the **From** field to specify a starting date for the displayed server log.
- Use the **To** field to specify an ending date for the displayed server log.
- Enter a role name in the **Username** field to display transactions performed only by that user.
- Enter a database name in the **Database** field to limit the displayed records to transactions that were performed against the specified database.
- Use the **CommandType** field to specify selection criteria for the commands that appear in the filtered report.

Properties SQL Statistics Dependencies Dependents Monitoring Server Log Postgres Enterprise Manager Host Postgres Enterprise Manager Server Server Log Object Type Server Status UP (Since : 27/04/2020, 15:47:09) Generated On 29/04/2020, 10:20:00 No. of alerts 6 (Acknowledged : 0)									
Server Logs									
id	Timestamp	User Name	Database Name	Process ID	Session ID	Transaction ID	Connection From	Command	Message
1870601	29/04/2020, 10:15:06	agent1	pem	64373	5ea8fbc6.fb75	5002111	127.0.0.1:55512	COPY	duration: 0.187 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE '~' CSV;
1870600	29/04/2020, 10:15:06	agent1	pem	64373	5ea8fbc6.fb75	0	127.0.0.1:55512	COMMIT	duration: 0.572 ms statement: END;
1870599	29/04/2020, 10:15:06	agent1	pem	64373	5ea8fbc6.fb75	5002110	127.0.0.1:55512	UPDATE	duration: 0.129 ms statement: UPDATE pem.log_configuration SET (last_read_filename, file_offset) = ('/var/lib/pgsql/12/data/log/postgresql-2020-04-29_004252.csv', 3135795) WHERE server_id = 1;
1870598	29/04/2020, 10:15:06	agent1	pem	64373	5ea8fbc6.fb75	5002110	127.0.0.1:55512	COPY	duration: 0.289 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE '~' CSV;
1870597	29/04/2020, 10:15:06	agent1	pem	64373	5ea8fbc6.fb75	0	127.0.0.1:55512	COMMIT	duration: 0.521 ms statement: END;
1870596	29/04/2020, 10:15:06	agent1	pem	64373	5ea8fbc6.fb75	5002109	127.0.0.1:55512	UPDATE	duration: 0.132 ms statement: UPDATE pem.log_configuration SET (last_read_filename, file_offset) = ('/var/lib/pgsql/12/data/log/postgresql-2020-04-29_004252.csv', 3134305) WHERE server_id = 1;
1870595	29/04/2020, 10:15:06	agent1	pem	64373	5ea8fbc6.fb75	5002109	127.0.0.1:55512	COPY	duration: 0.200 ms statement: BEGIN;COPY pemdata.server_logs(server_id, log_time, user_name, database_name, process_id, connection_from, session_id, session_line_num, command_tag, session_start_time, virtual_transaction_id, transaction_id, error_severity, sql_state_code, message, detail, hint, internal_query, internal_query_pos, context, query, query_pos, location, application_name) FROM STDIN WITH NULL AS 'NULL' QUOTE '~' CSV;
1870594	29/04/2020, 10:15:06	agent1	pem	64373	5ea8fbc6.fb75	0	127.0.0.1:55512	COMMIT	duration: 0.282 ms statement: END;

Fig. 82: The Server Log Analysis dashboard

Properties SQL Statistics Dependencies Dependents Monitoring Server Log Postgres Enterprise Manager Host Postgres Enterprise Manager Server Server Log Object Type Server Status UP (Since : 27/04/2020, 15:47:09) Generated On 29/04/2020, 10:23:01 No. of alerts 6 (Acknowledged : 0)									
Server Logs									
Start		End		User		Database		Command type	
2020-04-29 09:22:11		2020-04-29 10:22:21		postgres		pem			
								✓ Apply	Reset

Fig. 83: Defining a Server Log filter

After you specify the criteria for filtering the server logs, select **Filter** to display the filtered server log in the Server Log table.

2.6.2 Postgres Log Analysis Expert

The PEM Log Analysis Expert analyzes the log files of servers that are registered with Postgres Enterprise Manager and produces an analysis of your Postgres cluster's usage based on log file entries. You can use information on the Log Analysis Expert reports to make decisions about optimizing your cluster usage and configuration to improve performance.

Before using the PEM Log Analysis Expert, you must specify a **ServiceID** value on the **Advanced** tab of the Server Properties dialog box. You must also use the Log Manager wizard to enable log collection by the PEM server.

To open the Postgres Log Analysis Expert wizard, in the PEM client, select **Management>PostgresLogAnalysisExpert**. In the wizard's Welcome screen, select **Next**.

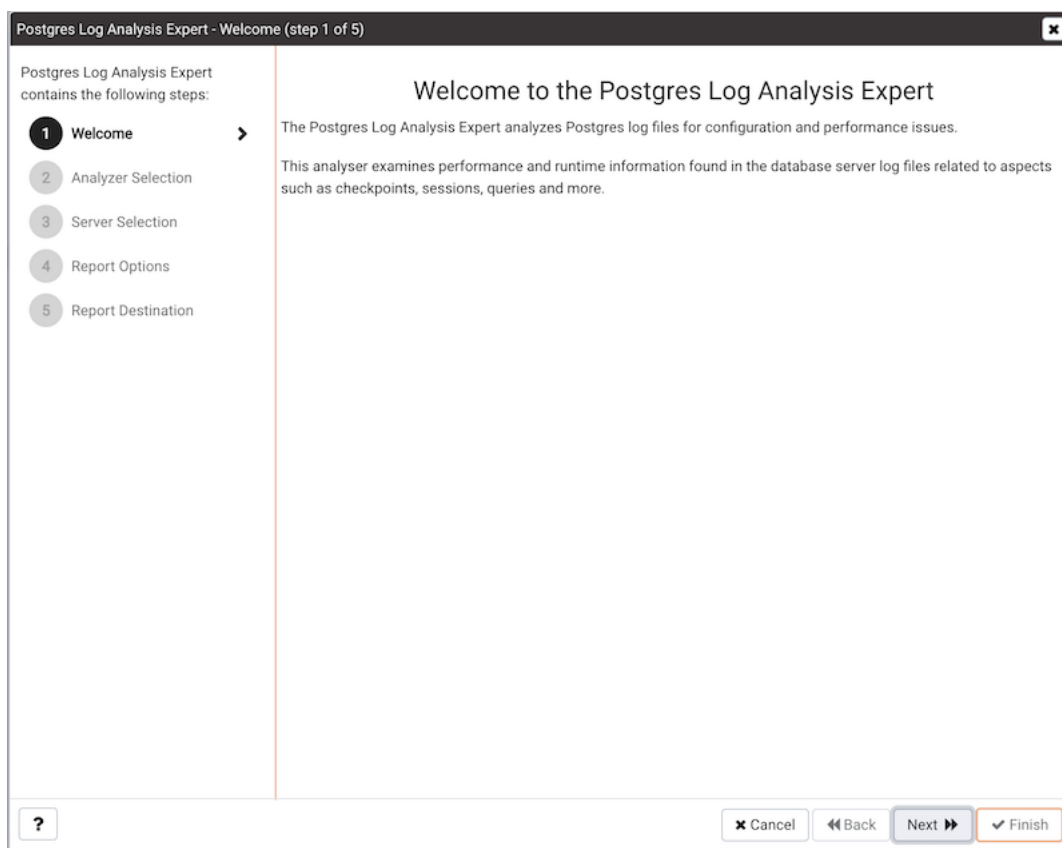


Fig. 84: The Log Analysis Expert Welcome dialog

The wizard's Analyzer selection dialog box displays a list of analyzers from which you can select. Each analyzer generates a corresponding table, chart, or graph that contains information gleaned from the log files.

Select the check box next to an analyzer to specify for the Log Analysis Expert to prepare the corresponding table, chart, or graph. After making your selections, select **Next** to continue to the Server selection tree.

Use the tree to specify the servers you want the Postgres Log Analysis Expert to analyze. If you select multiple servers, the resulting report contains the corresponding result set for each server in a separate but continuous list. Select **Next** to continue to the Report options dialog box.

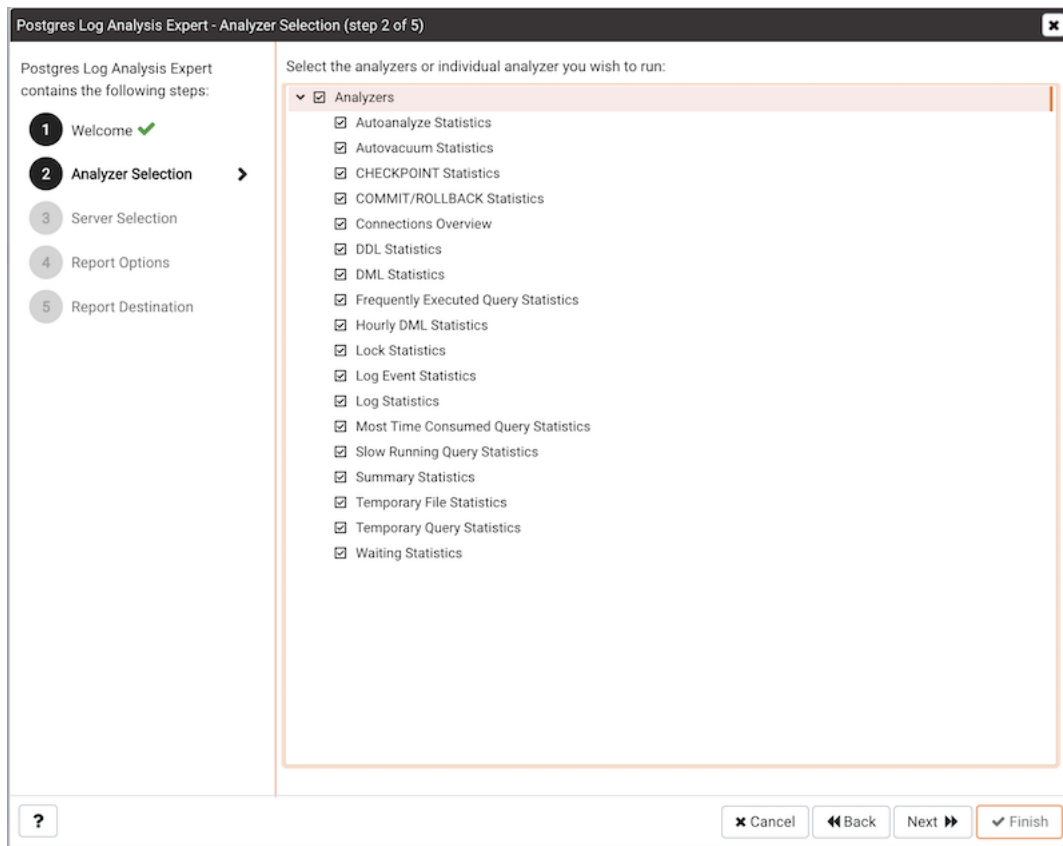


Fig. 85: The Analyzer selection dialog

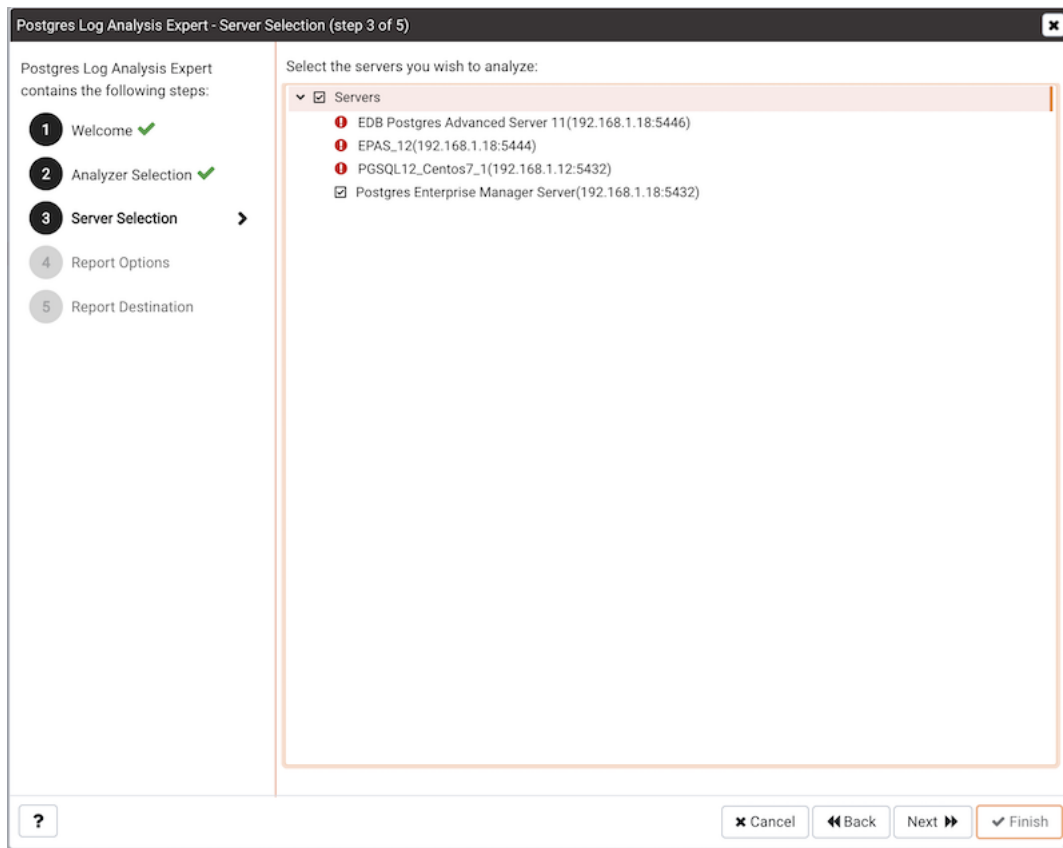


Fig. 86: The Server selection dialog

Postgres Log Analysis Expert - Report Options (step 4 of 5)

Postgres Log Analysis Expert contains the following steps:

- 1 Welcome ✓
- 2 Analyzer Selection ✓
- 3 Server Selection ✓
- 4 Report Options >
- 5 Report Destination

Time intervals

Relative days:

From:

To:

(+/-)From date:

Days before or after the 'From date' that should be included in the analysis.

Options

Aggregate method:

Method to consolidate data for the selected time span.

Time span:

Number of minutes that the analyzer will incorporate into each calculation for a point on a graph.

Rows limit:

Number of rows to include in a table.

?

Fig. 87: The Report options dialog

Use the **Options** section to specify the analysis method and the maximum length of any resulting tables:

- Use the **Aggregatemethod** list to select the method used by the Log Analysis Expert to consolidate data for the selected time span: - **SUM** calculates a value that is the sum of the collected values for the specified time span. - **AVG** calculates a value that is the average of the collected values for the specified time span. - **MAX** uses the maximum value that occurs in a specified time span. - **MIN** uses the minimum value that occurs in a specified time span.
- Use the **Timespan** field to specify the number of minutes that the analyzer incorporates into each calculation for a point on a graph. For example, if the time span is five minutes, and the aggregate method is **AVG**, each point on the given graph contains the average value of the activity that occurred in a five-minute time span.
- Use the **Rowslimit** field to specify the maximum number of rows to include in a table.

Use the **TimeIntervals** section to specify the time range for the Log Analysis Expert to analyze:

- Set **Relativedays** to **Yes** to enable the **(+/-)Fromdate** field and specify the number of days before or after the date and time selected in the **From** field.
- Use the **From** field to specify the starting date and time for the analysis.
- Use the **To** field to specify the ending date and time for the analysis.
- Use the **(+/-)Fromdate** selector to specify the number of days before or after the **From** date to include in the analysis.

After you specify the report options, select **Next** to continue to the Report destination dialog box.

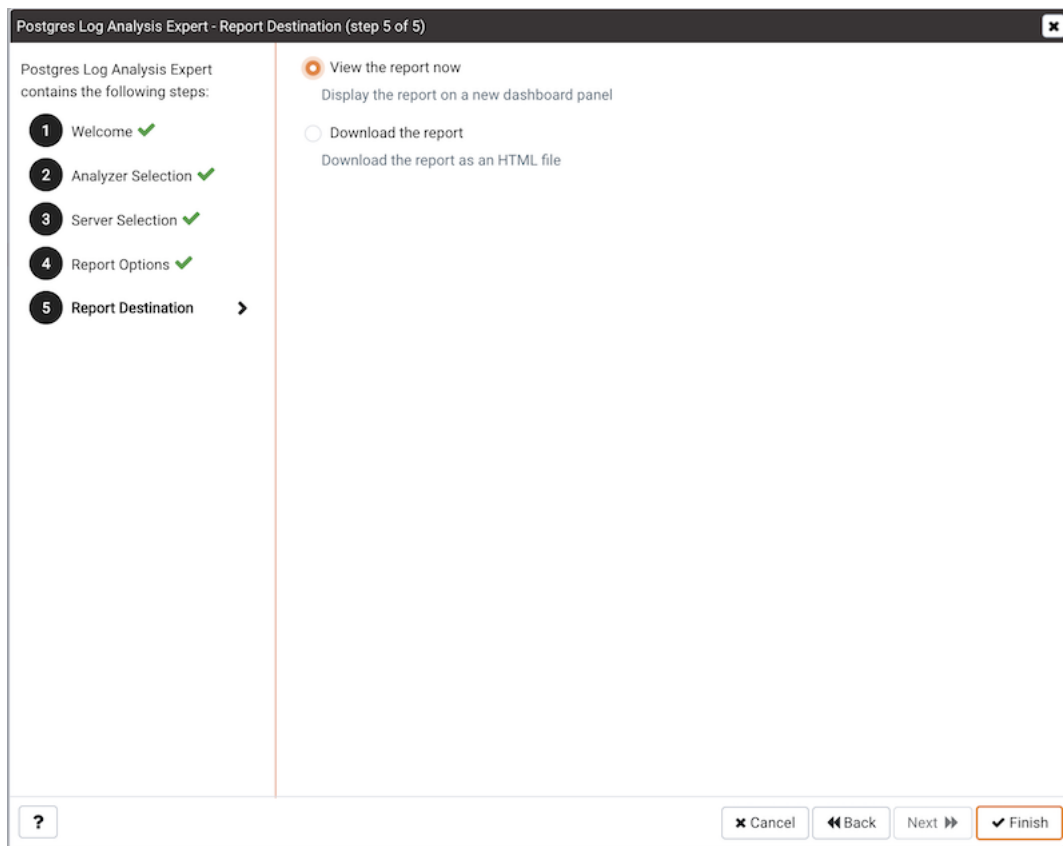


Fig. 88: The Report destination dialog

You can select the default option and select **Finish** to view the Log Analysis Expert report in the PEM client's tabbed browser. Or select **Download the report** to save a copy of the report to an HTML file for later use.

2.6.3 Reviewing the Postgres Log Analysis Expert report

If you choose to review the report immediately, the Postgres Log Analysis Expert report is displayed in the PEM Client window. The report header displays:

- The date and time that the report was generated
- The time period that the report spans
- The aggregation method specified when defining the report.

The name of the server for which information is displayed appears at the start of each section of the report.

The report displays the tables, graphs, and charts selected in the Log Analysis Expert wizard. Select **JumpTo**, located in the lower-right corner of the screen, to navigate to a specific graphic.

Postgres Log Analysis Expert

Interval: 2020-4-20 15:55:19 - 2020-4-27 15:55:19

Generated: 2020-04-27 16:00:51

Span: 5 Minutes

Aggregate: SUM

Go to: Postgres Enterprise Manager Server

▼ Postgres Enterprise Manager Server(192.168.1.18:5432)

Summary Statistics	
Settings	Values
Number of unique queries	151649
Total queries	155045
Total queries duration	
First query	27/04/2020 15:47:09.86 IST
Last query	27/04/2020 15:53:59.611 IST
Queries peak time	27/04/2020 15:49:38 IST queries 2821
Number of events	155045
Number of unique events	1
Total number of sessions	348
Total duration of sessions	
Average sessions duration	
Total number of connections	0
Total number of databases	0

Hourly DML Statistics						
Time	Database name	Statement	Count	Min duration	Max duration	Avg duration
27/04/2020 15:00	db01	SELECT	80	0.05	105.40	4.90
27/04/2020 15:00	edbstore_temp	SELECT	58	0.02	66.58	4.64
27/04/2020 15:00	hr	SELECT	48	0.01	29.26	1.87
27/04/2020 15:00	pem	COPY	1641	0.65	43.81	2.08
27/04/2020 15:00	pem	DELETE	73	0.44	8.74	1.04
27/04/2020 15:00	pem	INSERT	190	0.06	9.50	2.19

Fig. 89: The Postgres Log Analysis Expert Report

If the report contains an analysis of more than one monitored server, charts and tables are displayed in sets. First the graphs, tables, and charts that display statistics for one server appear. Then the graphics for the next server in the report appear.

2.7 Charts

You can use the **ManageCharts** tab to create or modify a custom line chart or table or to import a Capacity Manager template to use in a custom chart. After defining a chart, you can display the chart on a custom dashboard.

To open the **ManageCharts** tab, in the PEM client, select **Management>ManageCharts**.

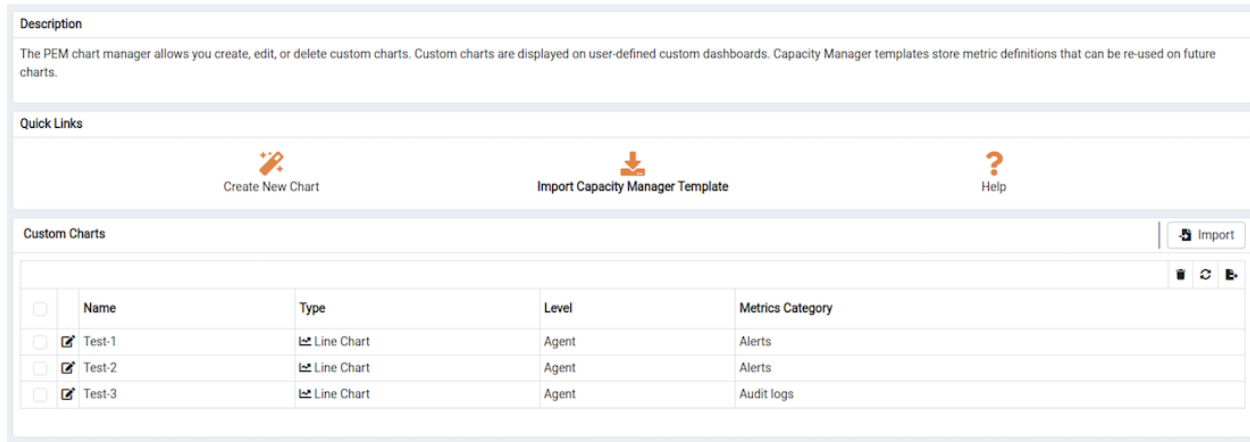


Fig. 90: The Manage Charts tab

The **ManageCharts** tab provides a Quick Links menu for accessing dialog boxes to:

- Create a new chart for use on a custom dashboard.
- Import a Capacity Manager template to use as a template for creating a custom chart.
- Access online help.

The Custom Charts table displays a list of charts you defined. When a chart is new, the font is green. When you add a chart or refresh the screen, the font is black.

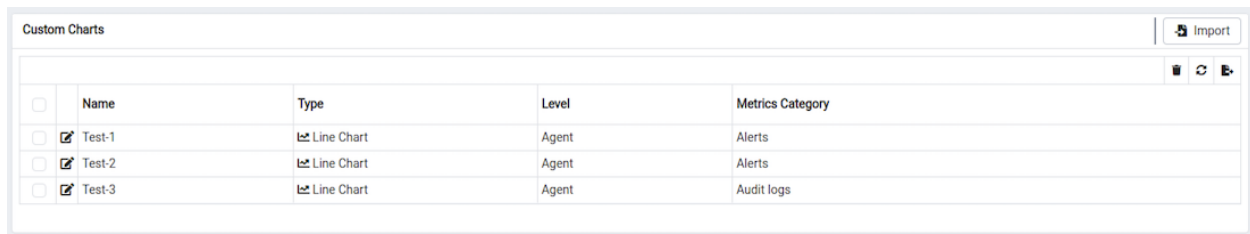


Fig. 91: The Custom Charts table

Use the search box in the upper-right corner of the Custom Charts table to search through your custom charts. Specify:

- Chart name
- Type
- Level
- Metrics category

Use icons to the left of a chart name in the Custom Charts table to manage a chart:

- Select **Edit** to open the Chart Configuration wizard and modify aspects of the chart or table.
- Select **Delete** to delete the selected chart.

2.7.1 Creating a custom chart

Select **CreateNewChart** in the Quick Links section of the **ManageCharts** tab to open the Create Chart wizard. The wizard takes you through the steps required to define a new chart.

The screenshot shows the 'Create Chart - Chart Configuration (step 1 of 4)' dialog box. On the left, a vertical progress bar indicates the current step is '1 Configure Chart', with subsequent steps being '2 Select Metrics', '3 Set Options', and '4 Set Permissions'. The main configuration area includes:

- Name:** A text field containing 'Top_Five_Table_Chart'.
- Category:** A dropdown menu showing 'Database Object Activity'.
- Type:** Radio buttons for 'Line chart' (selected) and 'Table'.
- Description:** A large, empty text area for providing a description.

 At the bottom of the dialog, there are four buttons: a help icon (?), 'Cancel', 'Back', 'Next', and 'Finish' (highlighted with an orange border).

Fig. 92: Specifying general information about the chart

Use the fields on the Configure Chart dialog box to specify general information about the chart:

- Specify the name of the chart in the **Name** field.
- Use the list in the **Category** field to specify the category in which to display this chart. When adding a custom chart to a custom dashboard, you can select the chart from the category specified.
- Use the **Type** field to specify if the chart is a line chart or a table.
- Provide a description of the chart in the **Description** field. The description is displayed on a custom dashboard when the user selects the information icon next to the chart.

After you complete the fields on the Configure Chart dialog box, select **Next**.

Use the fields on the Select Metrics dialog box to select the metrics to display on the chart.

- Use the **Metriclevel** list to specify the level of the PEM hierarchy from which you want to select metrics. You can specify **Agent**, **Database**, or **Server**. Each level offers access to a unique set of probes and metrics.
- Use the tree in the **Availablemetrics** box to select the metrics to display on the chart.

If you're creating a table, you can select metrics from only one probe. Each node of the tree lists the metrics returned by a single probe. Expand a node of the tree and select the check box next to a metric name to include that metric data in the table.

If you're creating a line chart, expand the nodes of the tree and double-click each metric that you want to include in the chart.

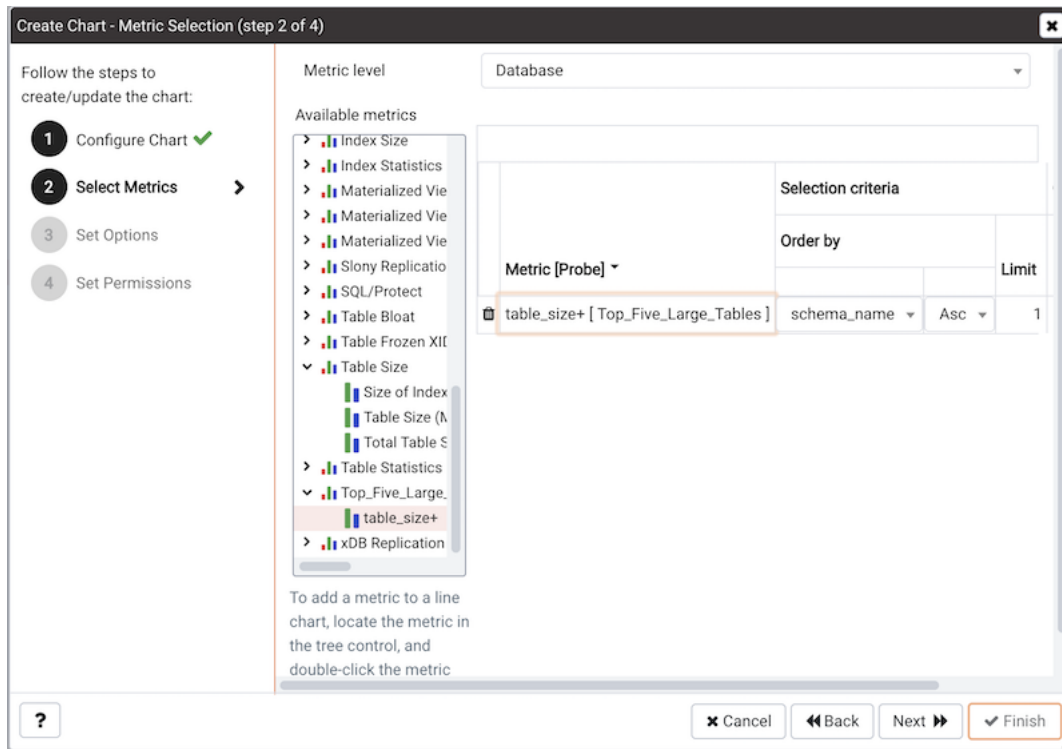


Fig. 93: Specifying the metrics to display

- Use the Selected Metrics panel to specify how the metric data appears in your chart. The selection panel displays the name of the metric in the nonmodifiable Metric [Probe] column. You can:

- Select the trash can to delete a metric from the list of selected metrics.
- Use the lists in the Selection Criteria column to specify the order of the data displayed.
- Use the ****Limit**** field to specify the number of rows in a table or lines in a chart: up to 32 lines and 100 rows.

- If you're creating a line chart, PEM supports comparisons of cross-hierarchy metrics.

- Select ****Compare**** to select one or more probe-specific attributes (such as CPUs, interfaces, and databases) to compare in the chart.
- Select ****Copy**** to apply your selections to all of the metrics for the same probe.

After you complete the fields on the Select Metrics dialog box, select **Next**.

Use the Set Options dialog box to specify display options for your chart:

- Use the **AutoRefresh** field to specify the number of minutes between chart updates. Choose a value from 1 to 120. The default auto refresh rate is 2 minutes.

Use fields under **Linechartoptions** to specify display preferences for a line chart:

- Use the **Pointstoplot** field to specify the maximum number of points to plot on the chart.
- Use the fields to the right of **Historicalspan** to specify how much historical data to display on the chart in days, hours, and minutes.

Use the fields in the **Dataextrapolation** box to specify whether PEM generates extrapolated data based on historical data:

Create Chart - Chart Options (step 3 of 4)

Follow the steps to create/update the chart:

- 1 Configure Chart ✓
- 2 Select Metrics ✓
- 3 Set Options ➔
- 4 Set Permissions

Auto refresh Minute(s)
Please specify the number of minutes between chart updates.

Line chart options

Points to plot
Please specify the maximum number of points to be plotted on the chart.

Historical span Day(s) Hour(s) Minute(s)
Please specify the number of minutes, hours or days of historical data that should be included on the chart.

Data extrapolation

☒ No Extrapolation ☐ Span ☐ Threshold

Data will not be extrapolated into the future.

Note: Extrapolated data is calculated using a linear trend analysis algorithm on the historical data. The accuracy of the extrapolated data depends on the available data in the historical span.

?

Fig. 94: Specifying chart options

- Select **NoExtrapolation** to omit extrapolated data from the chart.
- Select **Span** to use the **Days** and **Hours** selectors to specify the period of time spanned by the metrics on the chart.
- Select **Threshold** to use threshold selectors to specify a maximum or minimum value for the chart.

After you complete the fields in the Set Options dialog box, select **Next**.

Use the Set Permissions dialog box to specify display options for your chart.

- Set the **Sharewithall** slider to **Yes** to make the chart available to all authorized users. Select **No** to restrict access to the users or groups specified in the **Accesspermissions** field.
- Use the **Accesspermissions** field to select the groups you want to give access to the chart to.

After you finish defining the chart, select **Finish** to save your edits and add your chart to the list on the **ManageCharts** tab.

2.7.2 Importing a Capacity Manager template

Select **ImportCapacityManagerTemplate** in the Quick Links section of the **ManageCharts** tab to open the Create Chart dialog box and use a Capacity Manager template as a starting point for a chart or table.

When the Create Chart dialog box opens, provide information about the custom chart:

- Use the **Importcapacitytemplate** list to select the name of the template on which the chart is based.
- Specify the name of the chart in the **Name** field.

Create Chart - Chart Options (step 3 of 4)

Follow the steps to create/update the chart:

1

Configure Chart ✓

2

Select Metrics ✓

3

Set Options >

4

Set Permissions

Auto refresh

2

Minute(s)

Please specify the number of minutes between chart updates.

Line chart options

Points to plot

50

Please specify the maximum number of points to be plotted on the chart.

Historical span

7

0

0

Day(s)

Hour(s)

Minute(s)

Please specify the number of minutes, hours or days of historical data that should be included on the chart.

Data extrapolation

✓ No Extrapolation

Span

Threshold

Data will not be extrapolated into the future.

Note: Extrapolated data is calculated using a linear trend analysis algorithm on the historical data. The accuracy of the extrapolated data depends on the available data in the historical span.

?

Cancel

Back

Next

Finish

Fig. 95: Specifying access permissions

Description

The PEM chart manager allows you create, edit, or delete custom charts. Custom charts are displayed on user-defined custom dashboards. Capacity Manager templates store metric definitions that can be re-used on future charts.

Quick Links

Create New Chart

Import Capacity Manager Template

Help

Custom Charts

Q

Search chart by Name, Type, Level or Category

	Name	Type	Level	Metrics Category
✓	test1	Table	Database	Database Object Activity
✓	Test1	Line Chart	Agent	Alerts
✓	Top_Five_Table_Chart	Line Chart	Database	Database Object Activity

Fig. 96: The chart definition is displayed on the Manage Charts tab

Create Chart - Chart Configuration (step 1 of 4)

Follow the steps to create/update the chart:

- 1 Configure Chart** ➤
- 2 Select Metrics
- 3 Set Options
- 4 Set Permissions

Import capacity template: Templates.Cap_man_blks_read_write

Name: Cap_man_blks_read_write

Category: Database I/O

Type: ☒ Line chart ☐ Table

Description:

? Cancel Back Next Finish

Fig. 97: Importing a Capacity Manager template

- Use the **Category** list to specify the category in which to display this chart. When adding a custom chart to a custom dashboard, you can select the chart from this category.
- Use the **Type** field to specify if the chart is a line chart or a table.
- Provide a description of the chart in the **Description** field. The description is displayed on the custom dashboard when user selects the information icon next to the chart.

Select **Next**.

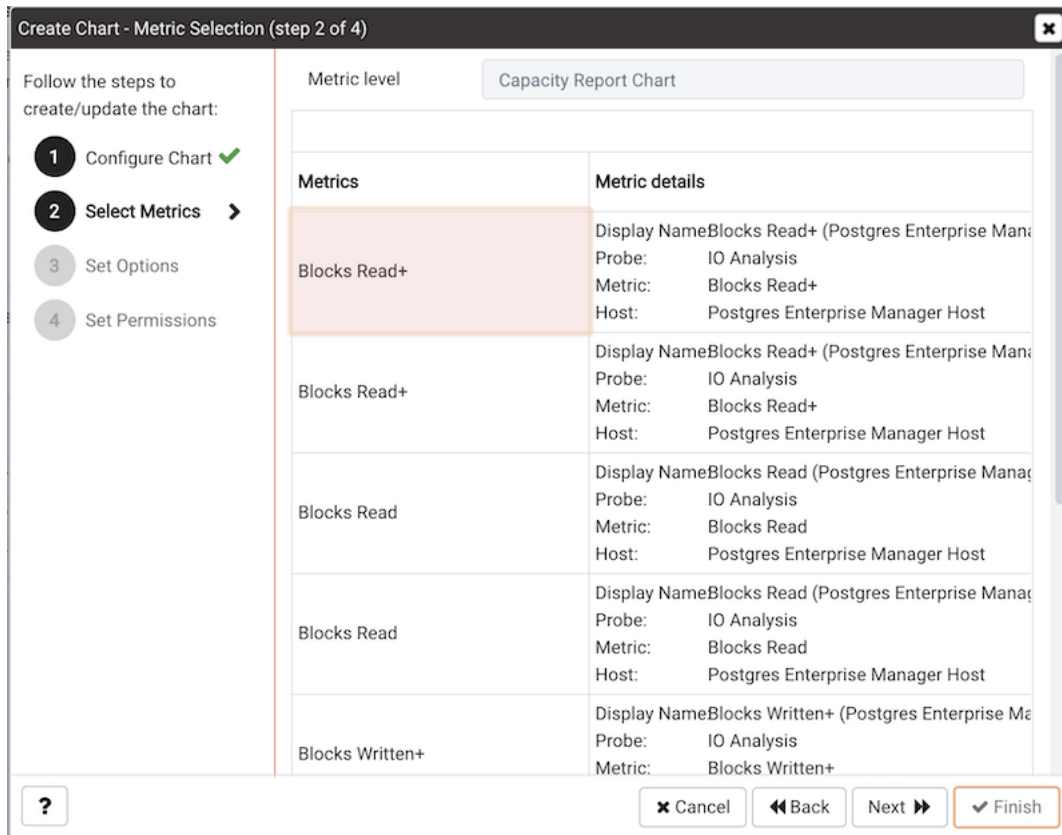


Fig. 98: The template metrics

The Select Metrics window allows you to review the metrics specified by the selected template. The bottom panel of the chart editor displays the metrics to include in the chart. You can't modify the metrics included in the chart using the chart editor. To modify the metrics, use the Capacity Manager utility to update the template.

After you review the metrics, select **Next**.

Use the Set Options window to specify display options for your chart:

- Use the **AutoRefresh** field to specify the number of minutes between chart updates. Choose a value from 1 to 120. The default auto refresh rate is 2 minutes.

Use the **Dataextrapolation** box to specify the time period covered by the chart. You can either:

- Select **Historicaldaysandextrapolateddays** and provide: - The number of days of historical data to chart in the **Historical** field. - The number of projected days to chart in the **Extrapolated** field.
- Select **Historicaldaysandthreshold** and provide: - The number of days of historical data to chart in the **Historical** field. - The threshold value at which the chart ends.

After you complete the Set Options window, select **Next**.

Create Chart - Chart Options (step 3 of 4)

Follow the steps to create/update the chart:

- 1 Configure Chart ✓
- 2 Select Metrics ✓
- 3 Set Options >
- 4 Set Permissions

Auto refresh Minute(s)
Please specify the number of minutes between chart updates.

Data extrapolation

☒ Historical days and extrapolated days ☐ Historical days and threshold

Historical Day(s)

Extrapolated Day(s)

? x Cancel << Back Next >> ✓ Finish

Fig. 99: Selecting chart options

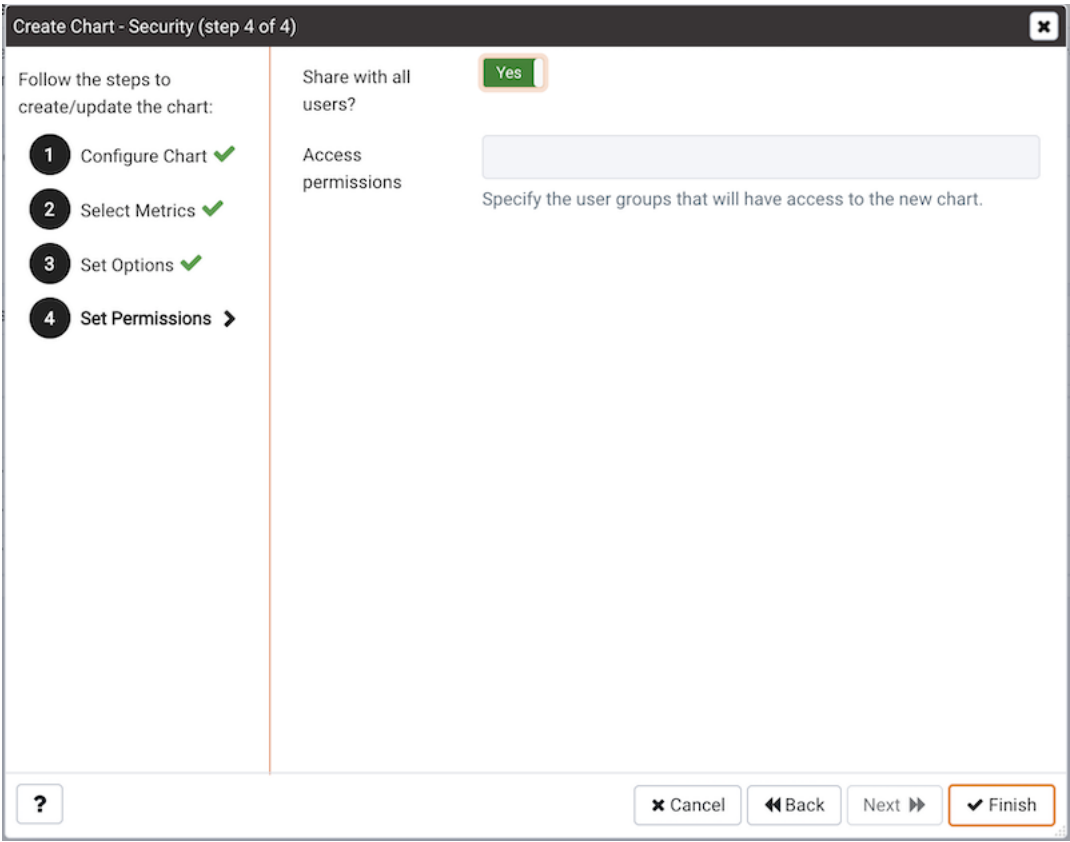


Fig. 100: Selecting permissions for the chart

Use the Set Permissions window to specify display options for your chart:

- Set the **Sharewithallslider** to **Yes** to make the chart available to all authorized users. Set it to **No** to restrict access to the users or groups specified in the **Accesspermissions** field.
- Use the **Accesspermissions** field to select the groups to give access to the chart to.

After you finish defining the chart, select **Finish** to save your edits and add your chart to the list on the **ManageCharts** tab.

2.8 Dashboards

PEM displays performance statistics through a number of dashboards. Each dashboard contains a series of summary views that contain charts, graphs, and tables that display the statistics related to the selected object.

2.8.1 Dashboards overview

The PEM client displays the Global Overview dashboard when it connects to the PEM server. Additional dashboards provide statistical information about monitored objects.

Opening dashboards

You can open a dashboard using either of these techniques:

- Select an active dashboard name from the **Management>Dashboards** menu.
- Right-click the name of a monitored object in the tree and select the name of the dashboard to review from the **Dashboards** menu.

Each dashboard is displayed on the **Monitoring** tab in the main panel of the client window. After opening a dashboard, you can navigate to other dashboards in the same tab.

Each dashboard header includes navigation menus that allow you to navigate to other dashboards. Use your browser's forward and back icons to scroll through previously viewed dashboards. Use **Refresh** to update the current dashboard.

Alerts dashboard

The Alerts dashboard displays the currently triggered alerts. If opened from the Global Overview, the dashboard displays the current alerts for all monitored nodes on the system. If opened from a node in a server, the report shows alerts related to that node and all monitored objects that reside below that object in the tree.

Audit Log Analysis dashboard

For EDB Postgres Advanced Server users, the Audit Log Analysis dashboard allows you to browse the audit logs that were collected from instances with audit logging and collection enabled.

BDR Admin dashboard

The BDR Admin dashboard displays the overview information about BDR node, BDR group, and BDR worker.

BDR Group Monitoring dashboard

The BDR Group Monitoring dashboard displays the information about BDR group subscription and BDR group replication slots.

BDR Node Monitoring dashboard

The BDR Node Monitoring dashboard displays the information about BDR node slots, BDR node replication rates, and BDR conflict history summary for the selected node.

Database Analysis dashboard

The Database Analysis dashboard displays performance statistics for the selected database.

I/O Analysis dashboard

The I/O Analysis dashboard displays I/O activity across various areas such as object DML activity and log operations.

Memory Analysis dashboard

The Memory Analysis dashboard supplies statistics concerning various memory-related metrics for the Postgres server.

Object Activity Analysis dashboard

The Object Activity Analysis dashboard provides performance details on tables/indexes of a selected database.

Operating System Analysis dashboard

The Operating System Analysis dashboard supplies information regarding the performance of the underlying machine's operating system.

Probe Log Analysis dashboard

The Probe Log Analysis dashboard displays any error messages returned by a PEM agent.

Server Analysis dashboard

The Server Analysis dashboard provides general performance information about the overall operations of a selected Postgres server.

Server Log Analysis dashboard

The Server Log Analysis dashboard allows you to filter and review the contents of server logs that are stored on the PEM server.

Session Activity Analysis dashboard

The Session Activity Analysis dashboard provides information about the session workload and lock activity for the selected server

Session Waits Analysis dashboard

The Session Waits Analysis dashboard provides an overview of the current DRITA wait events for an Advanced Server session.

Storage Analysis dashboard

The Storage Analysis dashboard displays space-related metrics for tablespaces and objects.

System Waits Analysis dashboard

The System Waits Analysis dashboard displays a graphical analysis of system wait information for an EDB Postgres Advanced Server session.

Streaming Replication Analysis dashboard

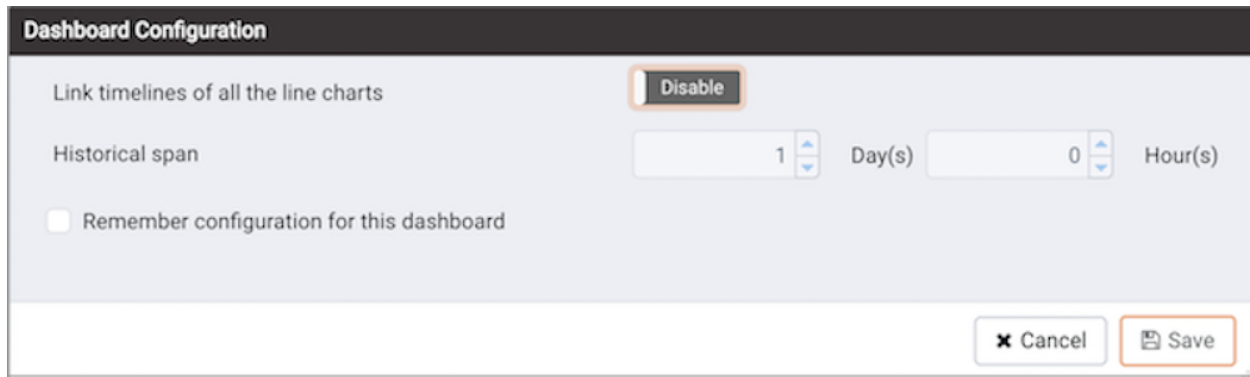
The Streaming Replication Analysis dashboard displays statistical information about WAL activity for a monitored server and allows you to monitor the status of Failover Manager clusters.

2.8.2 Dashboard configuration

Options on the Dashboard Configuration dialog box allow you to link the timelines of all of the line graphs on the dashboard. To open the Dashboard Configuration dialog box, select **Settings** in the dashboard header.

Use the Dashboard Configuration dialog box to control attributes of the charts displayed on the dashboard:

- Set **Linktimelinesofallthelinecharts** to **Enable** to apply the specified timeline to line graphs displayed on the dashboard. If set to **Disable**, your preferences are preserved for later use but don't modify the amount of data displayed.
- Use the **Days** selector to specify the number of days of gathered data to display on line graphs.
- Use the **Hour(s)** selector to specify the number of hours of gathered data to display on line graphs.
- Select **Rememberconfigurationforthisdashboard** to apply the customized time span to only the current dashboard only. Leave it cleared to apply the time span globally to line graphs on all dashboards.



Dashboard Configuration

Link timelines of all the line charts Disable

Historical span Day(s) Hour(s)

☐ Remember configuration for this dashboard

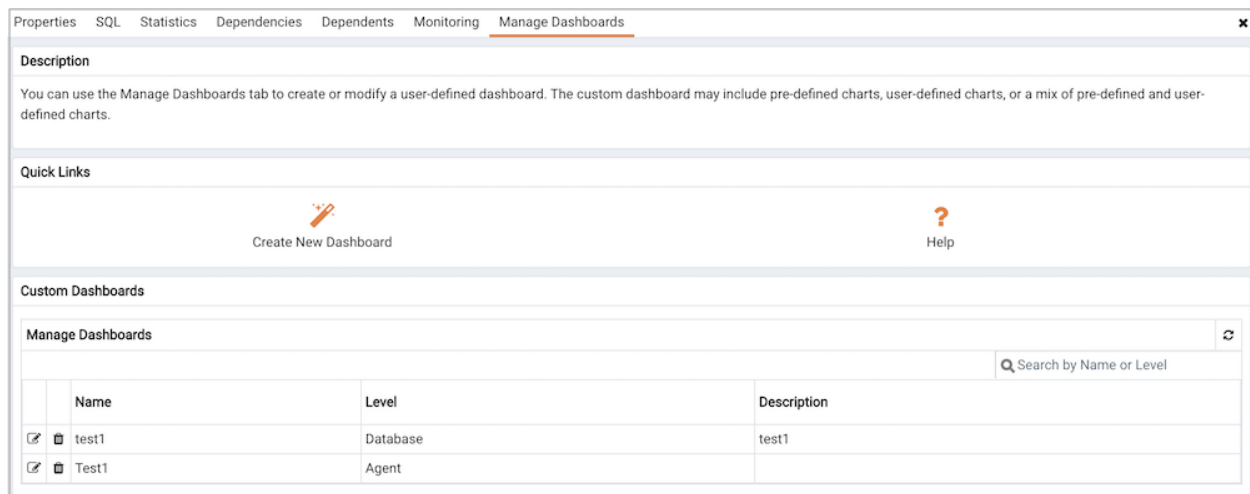
✕ Cancel Save

Fig. 101: The Dashboard Configuration dialog

Settings specified on the Dashboard Configuration dialog box are applied only to the current user's session.

2.8.3 Managing custom dashboards

PEM displays performance statistics through a number of system-defined dashboards. Each dashboard contains a series of summary views that contain charts, graphs, and tables that display statistics related to the selected object. You can use the **ManageDashboards** tab to create and manage custom dashboards that display the information that's most relevant to your system.



Properties SQL Statistics Dependencies Dependents Monitoring **Manage Dashboards**

Description

You can use the Manage Dashboards tab to create or modify a user-defined dashboard. The custom dashboard may include pre-defined charts, user-defined charts, or a mix of pre-defined and user-defined charts.

Quick Links

 Create New Dashboard  Help

Custom Dashboards

Manage Dashboards Search by Name or Level

	Name	Level	Description
☒	test1	Database	test1
☒	Test1	Agent	

Fig. 102: The Manage Dashboards tab

To create a custom dashboard, select **CreateNewDashboard** located in the Quick Links section of the **ManageDashboards** tab.

To modify an existing dashboard, select **Edit** next to the dashboard name. The dashboard editor opens, displaying the definition of the dashboard. After you finish modifying the dashboard's definition, select **Save**.

To delete a dashboard, select **Delete** next to the dashboard name.

Creating a custom dashboard

You can use the PEM dashboard editor to create or modify a custom dashboard. The custom dashboard can include predefined charts, custom charts, or a mix of predefined and custom charts.

Fig. 103: The Create Dashboard editor

Use the **Configure** section to specify general information about the dashboard:

- Specify a name for the dashboard in the **Name** field.
- Use the **Level** list to specify the level of the PEM hierarchy in the PEM client on which to display the dashboard. You can access a dashboard using the **Dashboards** menu on a global level, an agent level, the server level, or the database level. Each selected level in the list exposes a different set of metrics on which to base the custom dashboard's charts.
- Provide a description of the dashboard in the **Description** field.

Provide information in the **Opsdashboardoptions** box to use the dashboard as an Ops dashboard:

- Set **OpsDashboard?** to **Yes** to create a dashboard that's formatted to display on an Ops monitor.
- Set **ShowTitle?** to **Yes** to display the dashboard name at the top of the Ops dashboard.
- Use the **Font** list to select a custom font style for the title.
- Use the **Fontsize** list to select a custom font size for the title.

Use the **Permissions** box to specify the users who can view the new dashboard:

- Set **Sharewithall** to **Yes** to allow all teams to access the dashboard. Set it to **No** to enable the **Accesspermissions** field.
- Use the **Accesspermissions** field to specify the roles that can view the new dashboard. In the field, select from the list of users to add a role to the list of users with dashboard access.

After you complete the **ConfigureDashboard** section, select the arrow in the upper-right corner to close the section and access the **DashboardLayoutDesign** section.

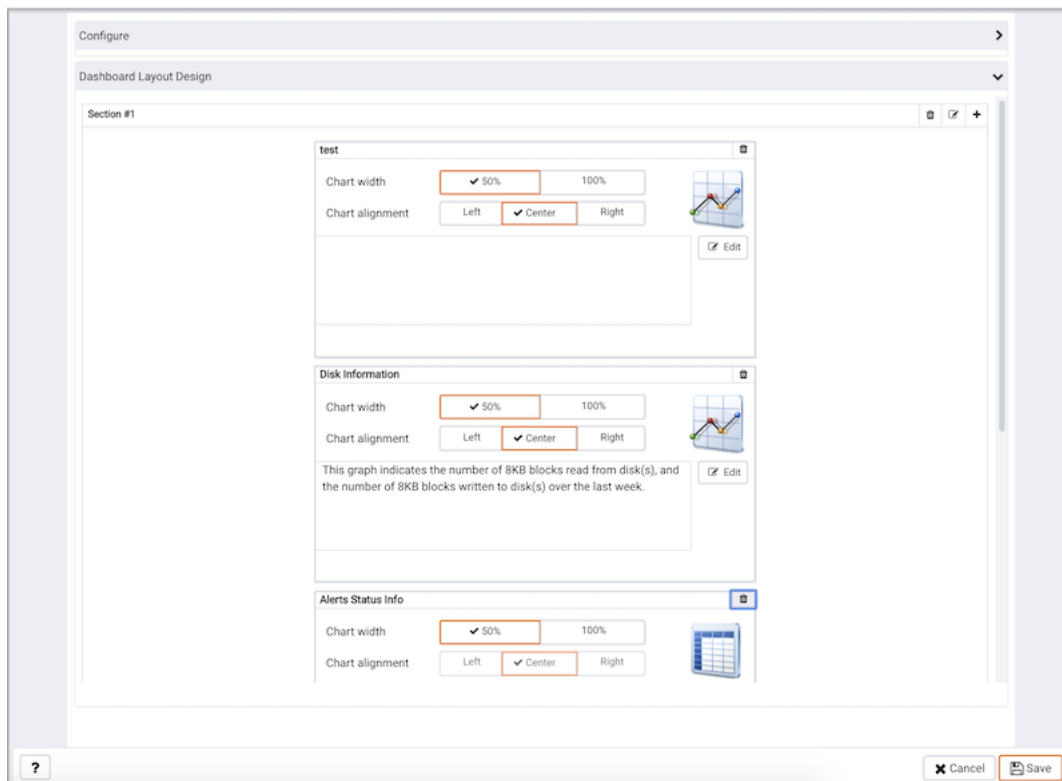


Fig. 104: Modifying a Section Header

Select **Edit** in a section header to specify a section name. Then, select the plus sign (+) to add a chart to the section.

Use the arrows to the right of each chart category to display the charts available and select a chart.

Use the chart detail selectors to specify placement details for the chart:

- Use the **Chartwidth** selector to indicate the width of the chart. Select **50%** to display the chart in half of the dashboard or **100%** to use the whole dashboard width.
- Use the **Chartalignment** selector to indicate the position of the chart in the section, left, center, or right justified.

Tables are always displayed centered.

When creating or editing a custom dashboard, you can drag to rearrange the charts in a section or to move a chart to a different section.

To add another chart to your dashboard, select the plus sign (+) in the section header. After you finish editing the dashboard, select **Save**.

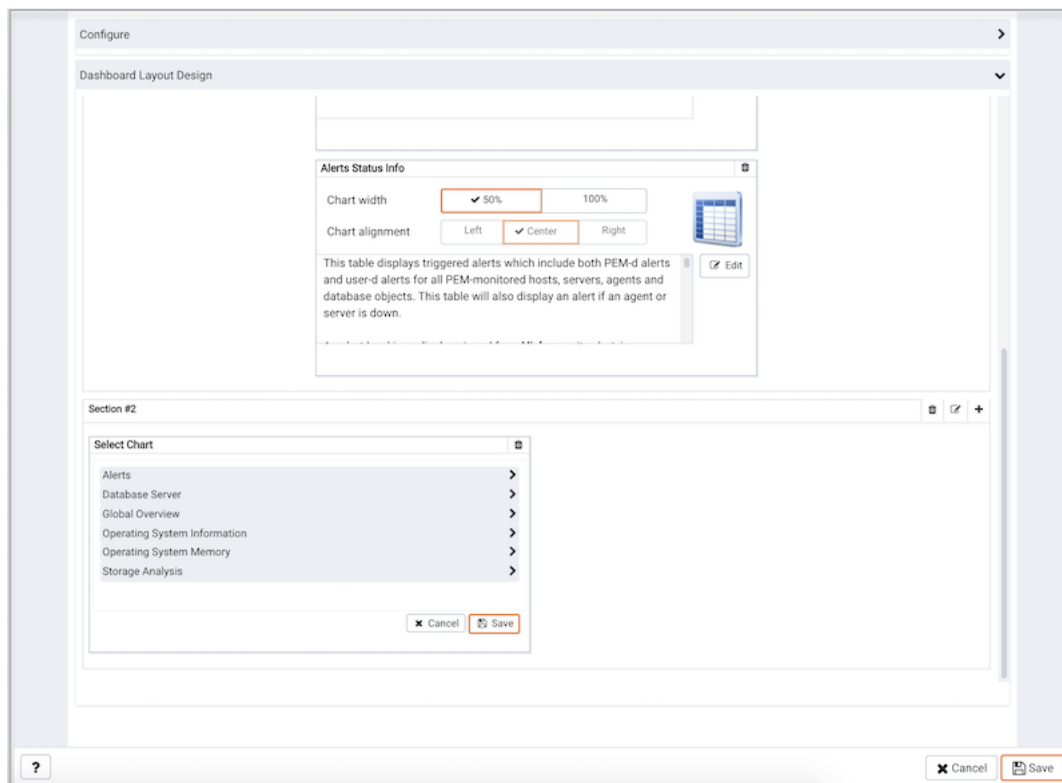


Fig. 105: Adding a Chart

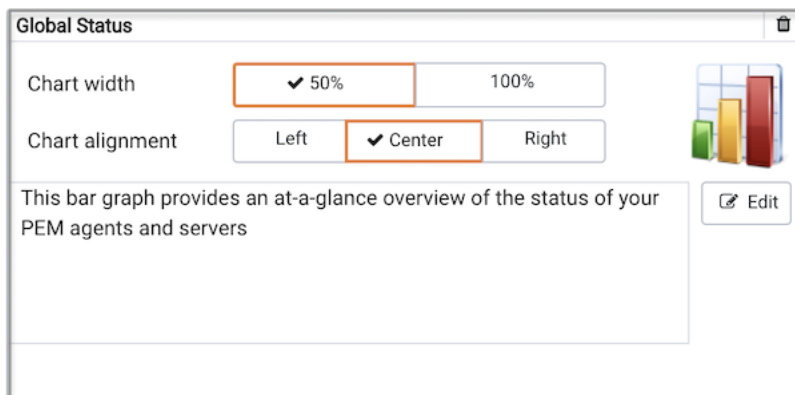


Fig. 106: Specifying placement details for a chart

Creating an Ops dashboard

You can use the PEM dashboard editor to create a custom dashboard formatted for display on an Ops monitor. An Ops dashboard displays the specified charts and graphs while omitting header information and minimizing extra banners, titles, and borders.

The screenshot shows a dialog box titled "Ops dashboard options". It has four input fields: "Ops dashboard?" with a "Yes" button, "Show title?" with a "Yes" button, "Font" with a dropdown menu showing "Arial", and "Font size" with a dropdown menu showing "14". At the bottom, there is a "Preview" button.

Fig. 107: Ops dashboard options

To create an Ops dashboard, provide detailed information about the Ops display in the **Opsdashboardoptions** section of the Create Dashboard dialog box.

- Set **OpsDashboard?** to **Yes** to create a dashboard that's formatted for display on an Ops monitor.
- Set **ShowTitle?** to **Yes** to display the dashboard name at the top of the Ops dashboard.
- Use the **Font** list to select a custom font style for the title.
- Use the **Fontsize** list to select a custom font size for the title.

After adding charts and tables to the Ops dashboard, select **Save**. You can then access the dashboard by navigating through the **Dashboards** menu of the hierarchy level specified in the **Level** field on the New Dashboard dialog box.

2.9 Remote monitoring

Remote monitoring is monitoring your Postgres cluster using a PEM Agent residing on a different host.

To remotely monitor a Postgres cluster with PEM, you must register the cluster with PEM and bind a PEM agent. See [Registering a server](#) for more information.

The following scenarios require remote monitoring using PEM:

- Postgres cluster running on AWS RDS
- Postgres cluster running on BigAnimal

PEM remote monitoring supports:

Feature Name	Remote monitoring supported?	Comments
Manage charts	Yes	
<i>Viewing system reports</i>	Yes	
<i>Capacity Manager</i>	Limited	There's no correlation between the Postgres cluster and operating system metrics.

continues on next page

Table 13 – continued from previous page

Feature Name	Remote monitoring supported?	Comments
Manage alerts	Limited	When you run an alert script on the Postgres cluster, it runs on the machine where the bound PEM agent is running and not on the actual Postgres cluster machine.
Manage dashboards	Limited	Some dashboards might not be able to show complete data. For example, the operating system information where the Postgres cluster is running isn't displayed as it isn't available.
Manage probes	Limited	Some of the PEM probes don't return information, and some of the functionality might be affected. For details about probe functionality, see PEM agent privileges.
<i>Postgres Expert - best practice enforcement</i>	Limited	The Postgres Expert provides partial information as operating system information isn't available.
Scheduled tasks	Limited	Scheduled tasks work only for Postgres clusters, and scripts run on a remote agent.
Core usage reports	Limited	The Core Usage reports don't show complete information. For example, the platform, number of cores, and total RAM aren't displayed.
<i>Audit Manager</i>	No	
<i>Log Manager</i>	No	
<i>Postgres Log Analysis Expert</i>	No	
<i>Tuning Wizard</i>	No	

MONITORING BARMAN

Barman (Backup and Recovery Manager) is an open-source administration tool for remote backups and disaster recovery of PostgreSQL servers in business-critical environments. It relies on PostgreSQL's point-in-time recovery technology, allowing DBAs to remotely manage a complete catalog of backups and the recovery phase of multiple remote servers from one location. For more information, see *Monitoring Barman*.

Starting with version 8.4, you can monitor a Barman server through the PEM console.

3.1 Prerequisites for monitoring Barman

Before adding a Barman server to the PEM console:

- You must manually install and configure Barman on the Barman host. For more information about installing and configuring Barman, see *Monitoring Barman*.
- Install the pg-backup-api tool on Barman host. For more information about installing, see pg-backup-api.

3.2 Configuring a Barman server

You can configure and edit your Barman server using:

- PEM web client
- `pemworker` command line

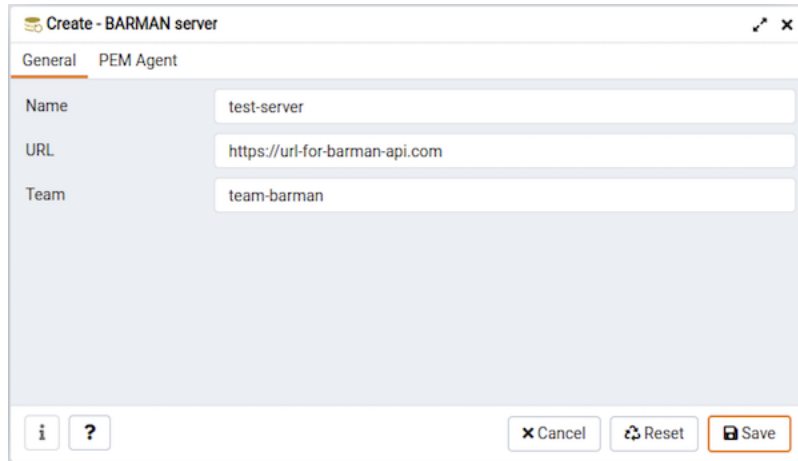
3.2.1 Using PEM web client

3.2.2 Configure

You can use the Create-BARMAN Server dialog box to register an existing Barman server with the PEM server. To open the dialog box, right-click the BARMAN Servers node and select **Create-BARMANServer**.

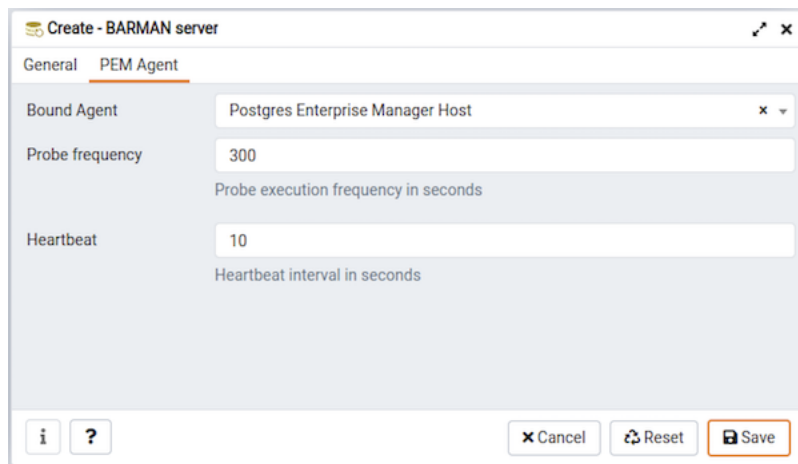
Use the **General** tab to describe the general properties of the Barman server:

- Use the **Name** field to specify a name for the server. The name identifies the server in the browser tree.
- Use the **URL** field to specify the URL of the host where Barman is installed.
- Use the **Team** field to specify a PostgreSQL role name. Only PEM users who are members of this role, who created the server initially, or have superuser privileges on the PEM server can see this server when they log on to PEM. If this field is left blank, all PEM users see the server.



The image shows a dialog box titled "Create - BARMAN server". It has two tabs: "General" (selected) and "PEM Agent". The "General" tab contains three input fields: "Name" with the value "test-server", "URL" with the value "https://url-for-barman-api.com", and "Team" with the value "team-barman". At the bottom of the dialog, there are three buttons: "Cancel", "Reset", and "Save".

Fig. 1: Create-BARMAN server dialog - General tab



The image shows the same dialog box, but with the "PEM Agent" tab selected. This tab contains three input fields: "Bound Agent" with the value "Postgres Enterprise Manager Host", "Probe frequency" with the value "300" (with a subtext "Probe execution frequency in seconds"), and "Heartbeat" with the value "10" (with a subtext "Heartbeat interval in seconds"). The "Cancel", "Reset", and "Save" buttons are also present at the bottom.

Fig. 2: Create-BARMAN server dialog - PEM Agent tab

Use the **PEMAgent** tab to specify connection details for the PEM Agent:

- Use the **BoundAgent** field to select the agent that you want to configure as a Barman server. Only those PEM agents that are supported for Barman are listed.
- Use the **ProbeFrequency** field to specify the number of seconds to execute the probes with the specified interval.
- Use the **Heartbeat** field to specify the interval to check the availability of PEM agent in seconds.

Note: After registering the Barman server, you need to restart the PEM agent.

3.2.3 Editing

To edit your Barman server, right-click the server from the browser tree and select **Properties**.

The screenshot shows a dialog box titled "test-5" with four tabs: General, PEM Agent, Information, and Configurations. The "General" tab is selected. It contains three input fields: "Name" with the value "test-5", "URL" with the value "http://test.com", and "Team" which is empty. At the bottom, there are buttons for "Cancel", "Reset", and "Save", along with information and help icons.

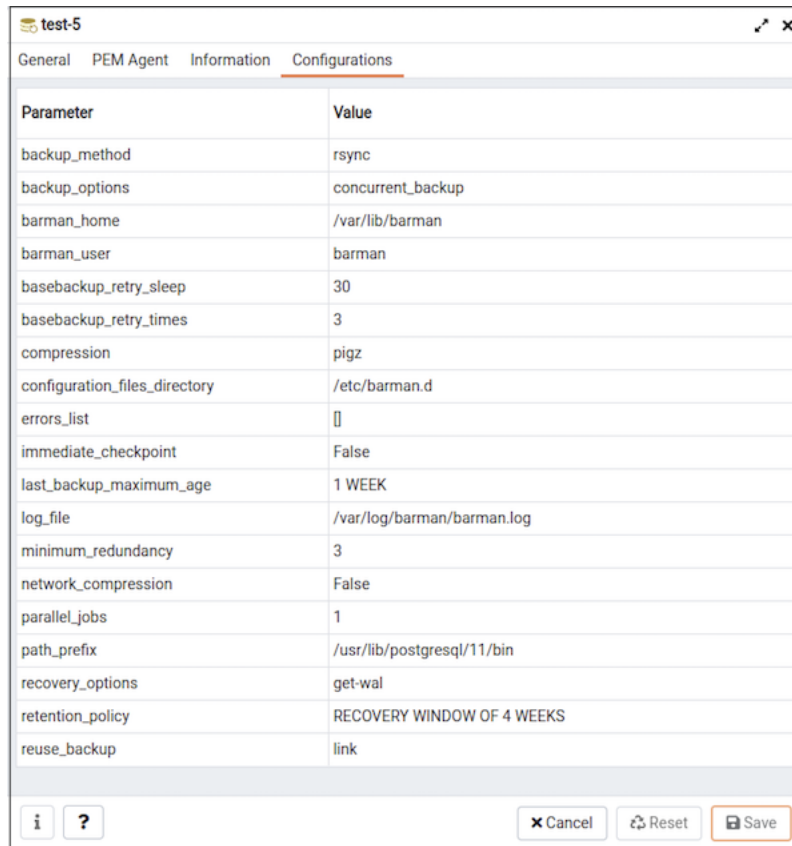
Fig. 3: BARMAN server properties - General tab

- Use the **PEMAgent** tab to modify the bound agent, probe frequency, and heartbeat. Only the owner of the Barman server can modify the fields on the **PEMAgent** tab.

The screenshot shows the same dialog box "test-5" but with the "Information" tab selected. It displays a table with two columns: "Parameter" and "Value". The table is currently empty. The "Save" button at the bottom is highlighted in orange.

Fig. 4: BARMAN server properties - Information tab

- Use the **Information** tab to view the detailed information about your Barman server. This tab gets populated whenever the Barman related probes are executed.



Parameter	Value
backup_method	rsync
backup_options	concurrent_backup
barman_home	/var/lib/barman
barman_user	barman
basebackup_retry_sleep	30
basebackup_retry_times	3
compression	pigz
configuration_files_directory	/etc/barman.d
errors_list	[]
immediate_checkpoint	False
last_backup_maximum_age	1 WEEK
log_file	/var/log/barman/barman.log
minimum_redundancy	3
network_compression	False
parallel_jobs	1
path_prefix	/usr/lib/postgresql/11/bin
recovery_options	get-wal
retention_policy	RECOVERY WINDOW OF 4 WEEKS
reuse_backup	link

Fig. 5: BARMAN server properties - Configuration tab

- Use the **Configuration** tab to view the configuration settings of your Barman server. This tab gets populated whenever the Barman related probes are executed.

3.2.4 Using pemworker command line

You can configure Barman server using `pemworker` command line options.

```
asheshvashi@pem:~/PEM/agent$ ./pemworker --update-barman --help
./pemworker --update-barman [barman-update-options]
```

barman-update-options:

```
--id <barman-id> (ID for the existing BARMAN API pg-backup-api)
--api-url <url> (URL of the BARMAN API pg-backup-api)
--probe-execution-frequency <interval> (Default: 30, Probe the BARMAN API pg-backup-
↪ api at regular interval in seconds and fetch the metrics.)
--heartbeat-interval <interval> (Default: 10, Ping the BARMAN API pg-backup-api status.
↪ API at a regular interval in seconds for checking its availability.)
--ssl-crt <certificate_file> (SSL certificate file for the BARMAN API.)
--ssl-key <key_file> (Private SSL key for the BARMAN API.)
--ssl-ca-cert <ca_file> (CA certificate to verify peer against the BARMAN API.)
```

(continues on next page)

(continued from previous page)

```

--config-file/-c <config_file> (Path to the agent configuration file.)

asheshvashi@pem:~/PEM/agent$ ./pemworker --unregister-barman --help
./pemworker --unregister-barman [barman-unregistration-options]

barman-unregistration-options:
  --id <barman-id> (ID for the existing BARMAN API, registered with the PEM Server.pg-
↳ backup-api)
  --config-file/-c <config_file> (Path to the agent configuration file.)

asheshvashi@pem:~/PEM/agent$ ./pemworker --register-barman --help
./pemworker --register-barman [barman-registration-options]

barman-registration-options:
  --api-url <url> (URL of the BARMAN API pg-backup-api)
  --description <name> (Description to show on the UI User interface for the BARMAN API.)
  --probe-execution-frequency <interval> (Default: 30, Probe the BARMAN API pg-backup-
↳ api at regular interval in seconds and fetch the metrics.)
  --heartbeat-interval <interval> (Default: 10, Ping the BARMAN API pg-backup-api status.
↳ API at a regular interval in seconds for checking its availability.)
  --ssl-crt <certificate_file> (SSL certificate file for the BARMAN API.)
  --ssl-key <key_file> (Private SSL key for the BARMAN API.)
  --ssl-ca-cert <ca_file> (CA certificate to verify peer against the BARMAN API.)
  --team <database-role> (Specify the name of the database group role, on the PEM.
↳ backend database server, that should have access to this BARMAN API Server.)
  --owner <database-user> (Specify the name of the database user, on the PEM backend.
↳ database server, who will own the BARMAN API Server.)
  --config-file/-c <config_file> (Path to the agent configuration file.)

```

Note: After registering the Barman server, you need to restart the PEM agent.

3.3 Viewing the Barman server details on a PEM dashboard

Once the Barman server is configured, you can see the entire backup- and server-related details for that Barman server on the PEM dashboard.

When you select a monitored Barman server, details of all the associated database servers along with their activities are displayed as a chart on the dashboard in the Barman Activities panel. You can select the activities on any criteria that you specify in the filter boxes (the database server, status, duration, or date).

The Servers panel displays a list of all the database servers managed by that Barman server along with the active status.

The Backups panel displays a list of all the database server backups managed by that Barman server. You can filter the list to display the details of any database server. You can also filter the list on any criteria that you specify in the filter box. Typically, this filter works with any kind of string value (excluding date, time, and size) listed under the columns. For example, you can enter tar to filter the list and display only those backups that are in tar format.

Backup details include the Backup ID, Server, Mode, Start time, End time, Size, Error, and Status column.

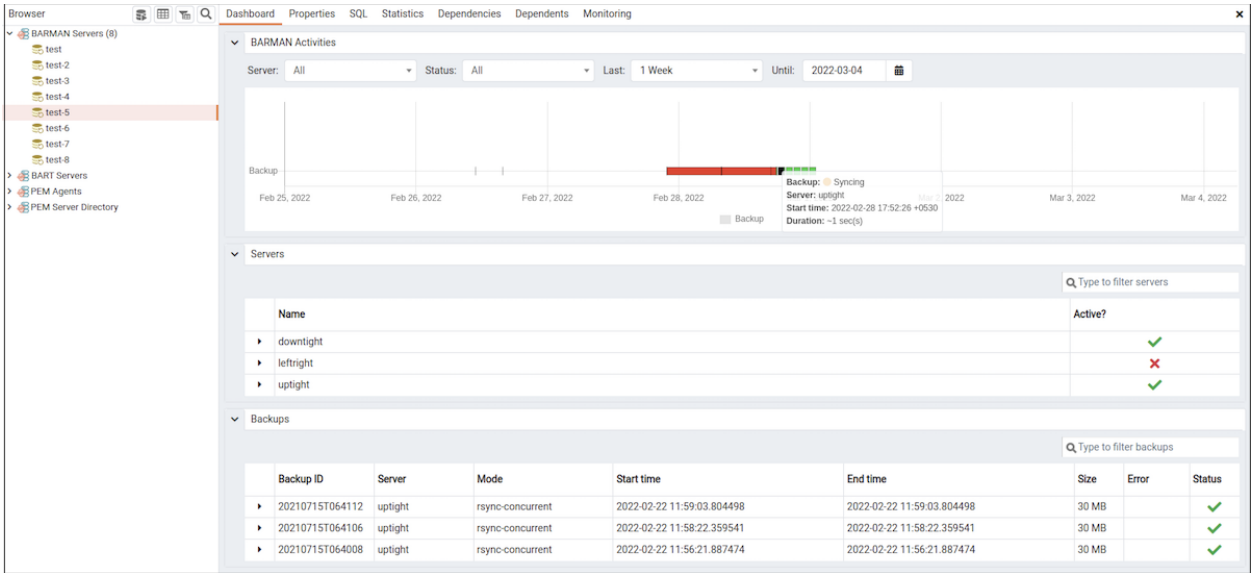


Fig. 6: BARMAN dashboard

Servers

Q Type to filter servers

Name	Active?
downtight	✓

Configuration

Parameter	Value
active	true
archiver	false
archiver_batch_size	0
backup_directory	/var/lib/barman/uptight
backup_method	rsync
backup_options	concurrent_backup
bandwidth_limit	
barman_home	/var/lib/barman
barman_lock_directory	/var/lib/barman
basebackup_retry_sleep	30
basebackup_retry_times	3
basebackups_directory	/var/lib/barman/uptight/base
check_timeout	30
compression	pgz
conninfo	host=uptight port=5432 dbname=postgres user=barman
create_slot	manual
custom_compression_filter	
custom_decompression_filter	
description	Backups from uptight
disabled	false
errors_directory	/var/lib/barman/uptight/errors
immediate_checkpoint	false
incoming_wals_directory	/var/lib/barman/uptight/incoming
last_backup_maximum_age	7 days
max_incoming_wals_queue	
minimum_redundancy	3
msg_list	[]
name	uptight
network_compression	false
parallel_jobs	1
path_prefix	/usr/lib/postgresql/11/bin
post_archive_retry_script	
post_archive_script	
post_backup_retry_script	
post_backup_script	
post_delete_retry_script	
post_delete_script	
post_recovery_retry_script	
post_recovery_script	
post_wal_delete_retry_script	
post_wal_delete_script	
pre_archive_retry_script	
pre_archive_script	
pre_backup_retry_script	
pre_backup_script	
pre_delete_retry_script	
pre_delete_script	
pre_recovery_retry_script	
pre_recovery_script	
pre_wal_delete_retry_script	
pre_wal_delete_script	
primary_ssh_command	
recovery_options	get-wal
retention_policy	window 4 w
retention_policy_mode	auto
reuse_backup	link
slot_name	backup_unarmed
ssh_command	ssh -q postgres@uptight
streaming_archiver	true
streaming_archiver_batch_size	0
streaming_archiver_name	barman_receive_wal
streaming_backup_name	barman_streaming_backup
streaming_conninfo	host=uptight port=5432 dbname=postgres user=streaming_barman
streaming_wals_directory	/var/lib/barman/uptight/streaming
tablespace_bandwidth_limit	
wal_retention_policy	simple-wal 4 w
wals_directory	/var/lib/barman/uptight/wals

leftright ✗

uptight ✓

Fig. 7: BARMAN dashboard - Servers panel

3.3. Viewing the Barman server details on a PEM dashboard

Backups

Q Type to filter backups

Backup ID	Server	Mode	Start time	End time	Size	Error	Status
20210715T064112	uptight	rsync-concurrent	2022-02-22 11:59:03.804498	2022-02-22 11:59:03.804498	30 MB		✓

Backup details

Parameter	Value
backup_id	20210715T064112
backup_label	START WAL LOCATION: 0/D000028 (file 00000001000000000000000D)\nCHECKPOINT LOCATION: 0/D000060\nBACKUP METHOD: streamed\nBACKUP FROM: master\...
begin_offset	40
begin_time	Thu Jul 15 06:41:12 2021
begin_wal	00000001000000000000000D
begin_xlog	0/D000028
config_file	/opt/postgres/data/postgresql.conf
copy_stats	{ 'analysis_time': 0.497281, 'analysis_time_per_item': { 'pgdata': 0.497281 }, 'copy_time': 0.482371, 'copy_time_per_item': { 'pg_control': 0.148338, 'pgdata': 0.332809 }, '...
deduplicated_size	41992
end_offset	248
end_time	Thu Jul 15 06:41:13 2021
end_wal	00000001000000000000000D
end_xlog	0/D0000F8
error	
hba_file	/opt/postgres/data/pg_hba.conf
ident_file	/opt/postgres/data/pg_ident.conf
included_files	/opt/postgres/data/conf.d/0000-tpa.conf, /opt/postgres/data/conf.d/0001-tpa_restart.conf, /opt/postgres/data/conf.d/11111-extensions.conf, /opt/postgres/data/conf...
mode	rsync-concurrent
pgdata	/opt/postgres/data
server_name	uptight
size	31975345
status	DONE
systemid	6985025709346482144
tablespaces	
timeline	1
version	110012
xlog_segment_size	16777216

20210715T064106	uptight	rsync-concurrent	2022-02-22 11:58:22.359541	2022-02-22 11:58:22.359541	30 MB		✓
20210715T064008	uptight	rsync-concurrent	2022-02-22 11:56:21.887474	2022-02-22 11:56:21.887474	30 MB		✓

Fig. 8: BARMAN dashboard - Backups panel

MANAGING BART

The EDB Backup and Recovery Tool (BART) is an administrative utility providing simplified backup and recovery management for multiple local or remote EDB Postgres Advanced Server and PostgreSQL database servers. For more information about BART, see Backup and Recovery Tool .

This feature is built into the Postgres Enterprise Manager (PEM) web interface and makes it easier for you to monitor and manage BART.

In PEM version 7.10 and later, you can manage a BART server and perform all the BART operations through the PEM console.

4.1 Prerequisites

- Before adding a BART server to the PEM console, you must manually install and configure BART on the BART host. For more information about installing and configuring BART, see BART installation .

Note: While integrating PEM with BART, the PEM agent creates the backup directory required for BART and ensures that the ownership and permissions on the directory are correct.

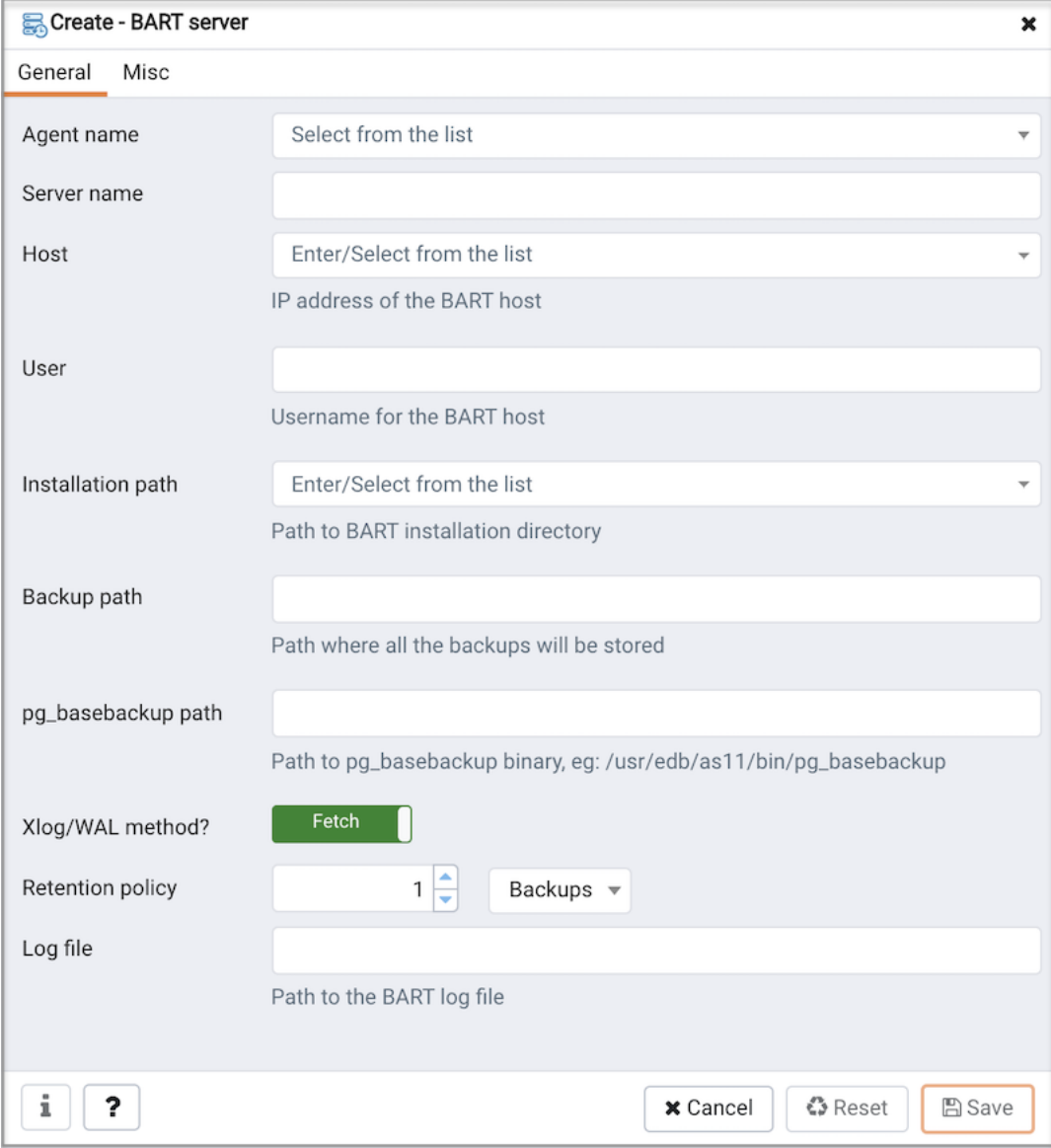
- Before associating a database server with a BART server, you must install SSH on the database server and the BART server.
- Before restoring a BART backup, you must install BART, a PEM agent, and SSH on the target server. SSH must also be installed on the BART server that you plan to use to restore.
- To take a backup of the replica database servers, ensure that the latest `pg_basebackup` utility is installed on the database server that you want to manage through BART.

4.2 Configuring a BART server

You can use the Create-BART Server dialog box to register an existing BART server with the PEM server. To open the dialog box, right-click the BART Servers node and select **Create-BARTServer** .

Use the **General** tab to describe the general properties of the BART server:

- Use the **AgentName** field to select the agent that you want to use for the BART server. Only those PEM agents that are supported for BART are listed.
- Use the **ServerName** field to specify a name for the server. The name specified identifies the server in the Browser tree.
- Use the **Host** field to specify the IP address of the host or agent where BART is installed.



The image shows a 'Create - BART server' dialog box with a 'General' tab selected. The dialog has a title bar with a close button (X) and two tabs: 'General' and 'Misc'. The 'General' tab contains the following fields and controls:

- Agent name:** A dropdown menu with the text 'Select from the list'.
- Server name:** A text input field.
- Host:** A dropdown menu with the text 'Enter/Select from the list'. Below it is the label 'IP address of the BART host'.
- User:** A text input field. Below it is the label 'Username for the BART host'.
- Installation path:** A dropdown menu with the text 'Enter/Select from the list'. Below it is the label 'Path to BART installation directory'.
- Backup path:** A text input field. Below it is the label 'Path where all the backups will be stored'.
- pg_basebackup path:** A text input field. Below it is the label 'Path to pg_basebackup binary, eg: /usr/edb/as11/bin/pg_basebackup'.
- Xlog/WAL method?:** A green button labeled 'Fetch' next to a small vertical slider.
- Retention policy:** A numeric input field with the value '1' and a 'Backups' dropdown menu.
- Log file:** A text input field. Below it is the label 'Path to the BART log file'.

At the bottom of the dialog, there are three buttons: 'Cancel' (with an X icon), 'Reset' (with a circular arrow icon), and 'Save' (with a floppy disk icon and an orange border). To the left of these buttons are two small icons: an information icon (i) and a help icon (?).

Fig. 1: The Create-BART server dialog - General tab

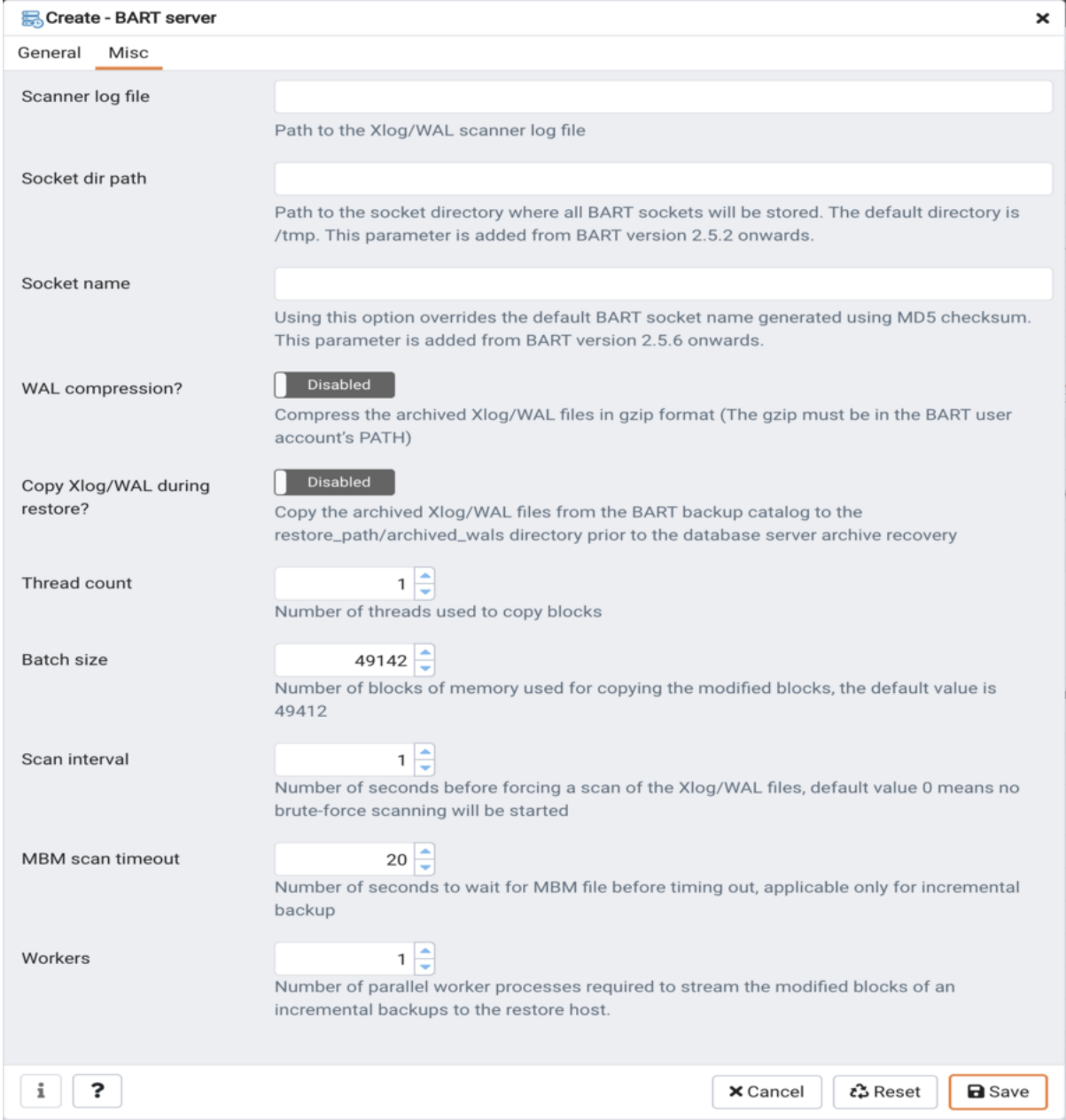
- Use the **User** field to specify the user name to use for performing all the BART operations. You can use the `enterprisedb` (for EDB Postgres Advanced Server) or `postgres` (for PostgreSQL) database user account, or you can create a new BART user account. This user must be an operating system user who owns the BART backup catalog directory.
- Use the **Installationpath** field to specify the directory path where BART is installed on the host or BART server.
- Use the **Backuppath** field to specify the file system parent directory to store all BART backups and archived WAL files.
- Use the **pg_basebackup_path** field to specify the path to the `pg_basebackup` utility.
- Use the **Xlog/WALmethod** field to specify how to collect the transaction log while `pg_basebackup` executes. The default option is **fetch**, which specifies to collect the transaction log files after the backup has completed. Set the **Xlog/WAL** method to **stream** to stream the transaction log in parallel with the full base backup creation. If streaming is used, the `max_wal_senders` configuration parameter in the `postgresql.conf` file for affected database servers must account for an additional session for the streaming of the transaction log. The setting must be a minimum of 2.

For more information about Xlog/WAL methods, see [PostgreSQL core documentation](#).

- Use the **Retentionpolicy** field to specify the retention policy for the backup. This policy determines when to mark an active backup as obsolete and hence a candidate for deletion. You can specify the retention policy in terms of number of backup or in terms of duration (days, weeks, or months).
- Use the optional **Logfile** field to specify the path to BART log file.

Use the **Misc** tab to describe the backup-related properties of the BART server:

- Use the **Scannerlogfile** field to specify the path to the Xlog/WAL scanner log file. BART creates a log scanner file only if you specify the path.
- Use the **Socketdirpath** field to specify the path to the socket directory to store all BART sockets. The default directory is `/tmp`. This parameter is added from BART version 2.5.2 and later.
- Use the **Socketname** field to specify the BART socket file name. Using this option overrides the default BART socket name generated using MD5 checksum. This parameter is added in BART version 2.5.6 and later.
- Use the **WALcompression?** switch to specify if you want to compress the archived Xlog/WAL files in Gzip format. To enable WAL compression, the `gzip` compression program must be present in the BART user account's `PATH`. Don't enable the WAL compression setting for database servers where you need to take incremental backups.
- Use the **CopyWALsduringrestore?** field to specify how the archived WAL files are collected when invoking the `RESTORE` operation. Set to **Enabled** to copy the archived WAL files from the BART backup catalog to the `restore_path/archived_wals` directory prior to the database server archive recovery. Set to **Disabled** to retrieve the archived WAL files directly from the BART backup catalog during the database server archive recovery. Enabling this option helps you save time during the restore operation.
- Use the **Threadcount** field to specify the number of worker threads for copying blocks or data files from the database server to the BART backup catalog. Specify a thread count of 1 if you want to take the backup using the `pg_basebackup` utility.
- Use the **Batchsize** field to specify the number of blocks of memory used for copying modified blocks. This applies only to incremental backups.
- Use the **Scaninterval** field to specify the number of seconds to wait before the WAL scanner scans the new WAL files.
- Use the **MBMscantimeout** field to specify the number of seconds to wait for MBM files before timing out. This applies only to incremental backups.



The image shows a 'Create - BART server' dialog box with a 'Misc' tab selected. The dialog contains several configuration options for the BART server, each with a text input field and a descriptive note. The options are: Scanner log file, Socket dir path, Socket name, WAL compression? (set to Disabled), Copy Xlog/WAL during restore? (set to Disabled), Thread count (set to 1), Batch size (set to 49142), Scan interval (set to 1), MBM scan timeout (set to 20), and Workers (set to 1). At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save', along with information and help icons.

Parameter	Value	Description
Scanner log file		Path to the Xlog/WAL scanner log file
Socket dir path		Path to the socket directory where all BART sockets will be stored. The default directory is /tmp. This parameter is added from BART version 2.5.2 onwards.
Socket name		Using this option overrides the default BART socket name generated using MD5 checksum. This parameter is added from BART version 2.5.6 onwards.
WAL compression?	Disabled	Compress the archived Xlog/WAL files in gzip format (The gzip must be in the BART user account's PATH)
Copy Xlog/WAL during restore?	Disabled	Copy the archived Xlog/WAL files from the BART backup catalog to the restore_path/archived_wals directory prior to the database server archive recovery
Thread count	1	Number of threads used to copy blocks
Batch size	49142	Number of blocks of memory used for copying the modified blocks, the default value is 49412
Scan interval	1	Number of seconds before forcing a scan of the Xlog/WAL files, default value 0 means no brute-force scanning will be started
MBM scan timeout	20	Number of seconds to wait for MBM file before timing out, applicable only for incremental backup
Workers	1	Number of parallel worker processes required to stream the modified blocks of an incremental backups to the restore host.

Fig. 2: The Create-BART server dialog - Misc tab

- Use the **Workers** field to specify the number of parallel worker processes required to stream the modified blocks of an incremental backups to the restore host.

4.3 Associating the BART server with a database server

After configuring the BART server, you must associate it with the database server whose backup you want to manage with BART. You can either:

- Use the PEM console to modify the properties of an existing monitored database server to map it to the newly configured BART server.
- Use the PEM console to create a new monitored database server and map it to the newly configured BART server.

To map the BART server to a new PEM database server, right-click the PEM Server Directory node and select **Create>Server** . Enter the details on all the generic tabs and then enter the BART-specific details on the **BART** tab.

Postgres Enterprise Manager Server

General Connection SSL SSH Tunnel Advanced PEM Agent **BART**

General Misc

For BART configuration, you need to install a PEM agent on the database server if Remote Monitoring is disabled for the agent. BART supports database server version 9.5 and above.

BART server: Select from the list

Server name:
Database server name that uniquely identifies an entry for database server in the server section of the configuration file

Backup name:
Template for backup name (may include %year, %month, %day, %hour, %minute, and %second)

Host address:
IP address of the database server to be configured for backup

Port:

User:

Password:

Cluster owner:
Operating system user that owns the database cluster

Override archive command? ☐ No
Overrides the archive_command settings in the postgresql.conf file.

Archive command:
Parameters for archive command (%p, %h, %a, %f). Ensure to bind a PEM agent and provide Service ID to reload or restart the database server.

Allow incremental backup? ☐ No

Setup passwordless SSH? ☐ No
Ensure to bind a PEM agent to setup passwordless SSH authentication between BART host and database server.

Buttons: [i] [?] [Cancel] [Reset] [Save]

Fig. 3: The Create Server dialog (BART - General tab)

Use the **General** tab to describe the general properties of the BART Server that map to the PEM server:

- Use the **BARTserver** field to select the BART server name. All the BART servers configured in the PEM console are listed.
- Use the **Servername** field to specify a name for the database server that you want to back up using the BART server. This name is stored in the BART configuration file.
- Use the **Description** field to specify the description of the database server.
- Use the **Backupname** field to specify a template for user-defined names to assign to the backups of the database server. If you don't specify a backup name template, then the backup can be referenced in BART sub-commands only by the BART-assigned, integer backup identifier.

- Use the **Hostaddress** field to specify the IP address of the database server that you want to configure for backup.
- Use the **Port** field to specify the port to use for the database that you want to backup.
- Use the **User** field to specify the user of the database that you want to back up using BART through the PEM console. If you want to enable incremental backups for this database server, then the user must be a superuser.
- Use the **Password** field to specify the password for the user of the database that you want to back up.
- Use the **ClusterOwner** field to specify the Linux operating system user account that owns the database cluster. This is typically `enterprisedb` for EDB Postgres Advanced Server database clusters installed in the Oracle databases compatible mode. It is `postgres` for PostgreSQL database clusters and for EDB Postgres Advanced Server database clusters installed in the PostgreSQL databases compatible mode.
- Use the **Archivecommand** field to specify the desired format of the archive command string to use in the `bart.cfg` file. Inputs provided for the archive command overwrite the database server's `Postgresql.conf` file. Once the server is added, the database server restarts or database configurations are reloaded.
- Use the **Archivepath** field to store the archived WAL files. The default location is the BART backup catalog. This parameter is in BART version 2.5.2 and later.
- Use the **Allowincrementalbackup?** switch to specify whether to enable incremental backup for this database server.
- Use the **SetuppasswordlessSSH?** switch to specify if you want to create SSH certificates to allow passwordless logins between the database server and the BART server. Be sure to bind a PEM agent before setting up the passwordless SSH authentication. Passwordless SSH doesn't work for a database server being remotely monitored by a PEM agent.

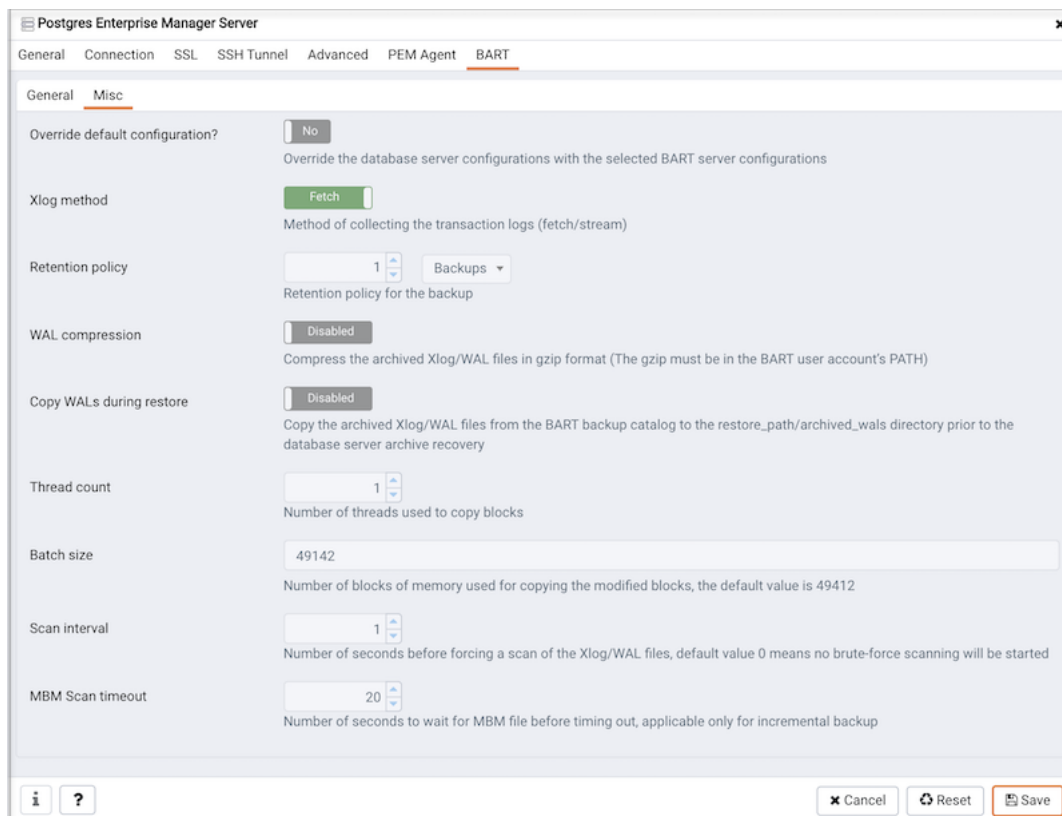


Fig. 4: The Create - Server dialog (BART - Misc tab)

Use the fields on the **Misc** tab to describe the miscellaneous properties of the BART server:

- Use the **Overridedefaultconfiguration?** switch to specify if you want to override the BART server configurations with the specific database server configurations.
- Use the **Xlog** method to specify how to collect the transaction log while `pg_basebackup` executes.
- Use the **Retentionpolicy** field to specify the retention policy for the backup. This policy determines when to mark an active backup as obsolete and hence a candidate for deletion. You can specify the retention policy in terms of number of backup or in terms of duration (days, weeks, or months).
- Use the **WALcompression** switch to specify if you want to compress the archived Xlog/WAL files in Gzip format. To enable WAL compression, the gzip compression program must be present in the BART user account's PATH. The `wal_compression` setting must not be enabled for those database servers where you need to take incremental backups.
- Use the **CopyWALsduringrestore** field to specify how the archived WAL files are collected when invoking the RESTORE operation. Set to enabled to copy the archived WAL files from the BART backup catalog to the `<;restore_path>/archived_wals` directory prior to the database server archive recovery. Set to disabled to retrieve the archived WAL files directly from the BART backup catalog during the database server archive recovery.
- Use the **Threadcount** field to specify the number of threads to copy the blocks. You must set the thread count to 1 if you want to take a backup with the `pg_basebackup` utility.
- Use the **Batchsize** field to specify the number of blocks of memory used for copying modified blocks. This option applies only to incremental backups.
- Use the **Scaninterval** field to specify the number of seconds for the WAL scanner to wait before scanning the new WAL files.
- Use the **MBMscantimeout** field to specify the number of seconds to wait for MBM files before timing out. This option applies only to incremental backups.
- Use the **Workers** field to specify the number of parallel worker processes required to stream the modified blocks of an incremental backups to the restore host.

4.4 Viewing the BART server details on a PEM dashboard

After associating the BART server with a database server, you can review the backup and restore details for that server on the PEM dashboard. You can also perform operations such as restoring or deleting a backup that's listed on the dashboard.

When you select a monitored BART server, details of all the associated database servers along with their backups and restore activities are displayed as a chart on the dashboard in the Backup and Restore Activities panel. You can filter the list of backups on criteria specified in the filter boxes (database server, activity, or duration).

The Managed Database Servers panel displays a list of all the database servers managed by a BART server along with their high-level details.

The Initiated Server Backups panel displays a list of all the backups of the database servers managed by a BART server. You can filter the list to display the details of any database server. You can also filter the list on criteria that you specify in the filter box. Typically, this filter works with any kind of string value (excluding date, time, and size) listed under the columns. For example, you can type `tar` to filter the list and display only those backups that are in tar format.

Backup details displayed include the Backup Name, Backup ID, Status, Server Name, Start Time, Type, Parent ID, Format, Duration, and Size. The Status column shows the status of the backups. Status can be one of the following: In Progress, Active, Keep, or Obsolete.

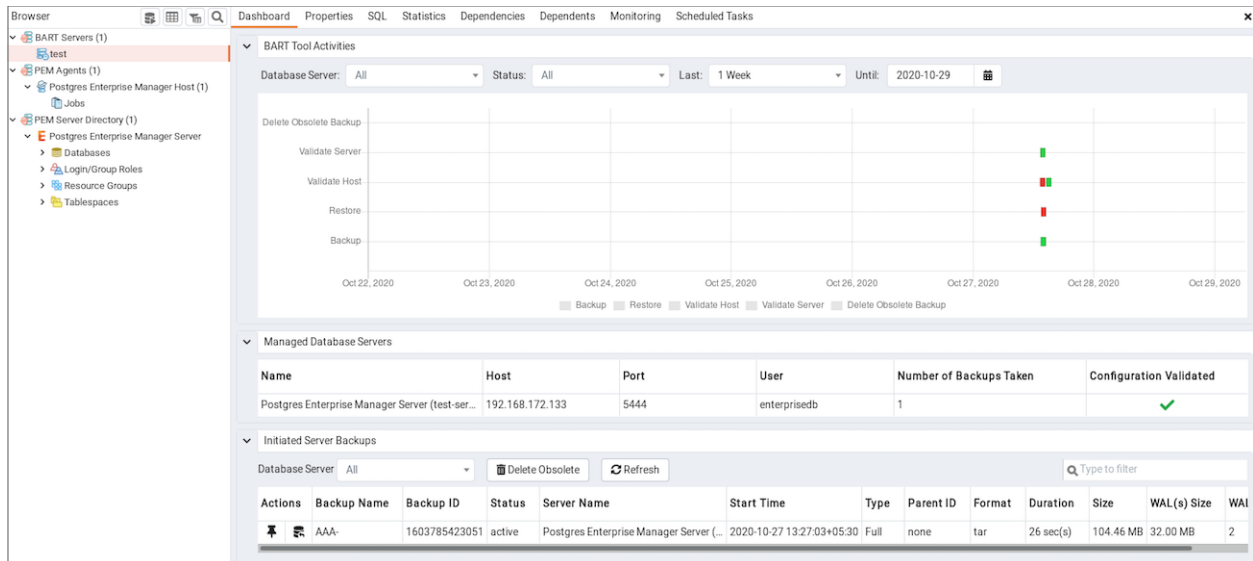


Fig. 5: The BART Backup Dashboard

A backup is marked as Obsolete when the backup retention period has passed or if the number of retained backups (specified as the retention policy of the BART server) is met. If you want to make an exception so that a particular backup doesn't get marked as Obsolete even after the retention policy expires, mark that backup as Keep. Similarly, if you mark a particular backup as NoKeep, the backup is reevaluated to determine whether to change its status back to Obsolete based on the current retention policy.

If any of the scheduled tasks for backup, restore, validate host, validate server, or delete obsolete backup for any of the BART server is deleted, it doesn't appear under the BART Tool Activities graph of the BART server's dashboard. However, it is listed under the **InitiatedServerBackups** list.

A pin in the first column under Actions indicates that you can mark a backup as Keep by selecting the pin. An inverted pin indicates that you can mark the backup as NoKeep. The second column under Actions displays the **Restore** icon, which you can select to perform the restore operation.

You can delete all the obsolete backups by selecting **DeleteObsolete**. You can also refresh the list of backups by selecting **Refresh**.

4.5 Scheduling BART backups

To create or manage a backup, select **Tools>ScheduleBackup**. The dialog box header displays general execution information about the backup. Select the plus sign (+) to add information about a scheduled backup. Enter the backup details in the schedule definition dialog:

Use the **General** tab to describe the general properties of the backup:

- Use the **DatabaseServer** field to specify the target database server that you want to back up.
- Use the **Backupname** to specify a name for the backup.
- Use the **Backuptype** switch to specify the backup type—full backup or incremental.
- Use the **Parentbackup** field to select the ID of the parent backup for incremental backup. This parent backup can be a full or an incremental backup.
- Use the **Format** switch to specify the output format of the backup—plain text or tar. For incremental backup, select plain text.

Fig. 6: The Schedule Backup dialog - General options

- Use the **Gzipcompression** switch to specify whether to enable gzip compression for the backup. This option applies only to tar format.
- Use the **Compressionlevel** field to specify the gzip compression level on the tar file output.
- Use the **Threadcount** field to specify the number of threads that copy the blocks.
- Use the **MBMscantimeout** field to specify the number of seconds to wait for required MBM files before timing out.
- Use the **Verifychecksum** field to specify if you want the application to verify the checksum of the backup.
- Use the **pg_basebackup** field to specify whether to use the pg_basebackup utility for the backup. Typically, you use the pg_basebackup utility only for backing up the replica servers since you can't use it for incremental backups.

Provide scheduling details for the backup on the **Schedule** tab:

- Use the **Enabled?** switch to indicate whether to enable the schedule.
- Use the calendar selector in the **Start** field to specify the starting date and time for the schedule.
- Use the calendar selector in the **End** field to specify the ending date and time for the schedule.

Use the **Repeat** tab to specify the details about the schedule in a cron-style format. The schedule executes on each date or time element selected on the **Repeat** tab. Click in a field to open a list of valid values for that field. Select a specific value to add that value to the list of selected values for the field.

Use the fields in the **Days** box to specify the days when the schedule executes:

- Use the **WeekDays** field to select the days when the schedule executes.
- Use the **MonthDays** field to select the numeric days when the schedule executes. Specify the **LastDay** value to perform the schedule on the last day of the month, regardless of the date.

The screenshot shows the 'Schedule Backups - bart_server' dialog box. The 'General' tab is selected. The 'Enabled?' checkbox is checked. The 'Start' date is set to '2020-04-27 10:52 +05:30'. The 'End' date is set to 'YYYY-MM-DD HH:mm Z'. The 'Format' is set to 'Tar'. The 'Verify checksum?' and 'Use pg_basebackup?' options are both set to 'No'.

Fig. 7: The Schedule Backup dialog - (Schedule : General)

The screenshot shows the 'Schedule Backups - bart_server' dialog box. The 'Repeat' tab is selected. The 'Days' section includes 'Week days', 'Month days', and 'Months'. The 'Times' section includes 'Hours' and 'Minutes'. The 'Repeat' section includes a 'Repeat' checkbox and a 'Repeat interval' dropdown menu. The 'Repeat' checkbox is checked. The 'Repeat interval' is set to 'Full'.

Fig. 8: The Schedule Backup dialog - (Schedule : Repeat)

- Use the **Months** field to select the months when the schedule executes.

Use the fields in the **Times** box to specify the times when the schedule executes:

- Use the **Hours** field to select the hour when the schedule executes.
- Use the **Minutes** field to select the minute when the schedule executes.

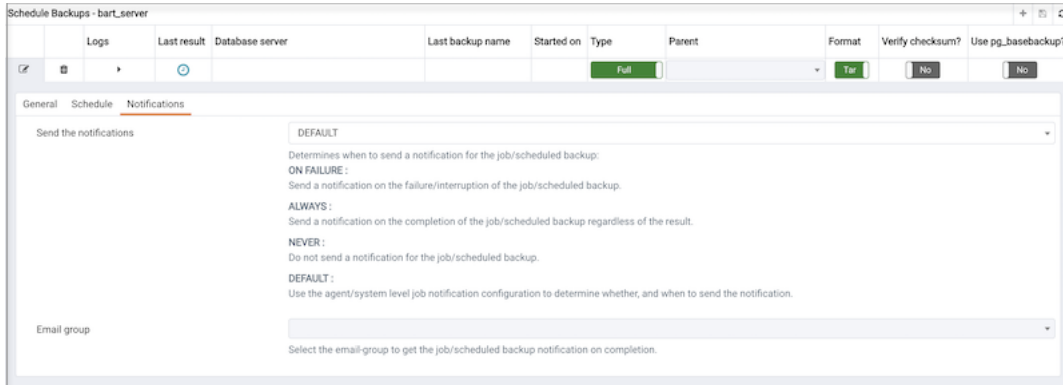


Fig. 9: The Schedule Backup dialog box - (Schedule : Notifications)

Use the **Notifications** tab to specify the email notification settings for a scheduled backup:

- Use the **Sendthenotifications** field to specify when to send the email notifications.
- Use the **Emailgroup** field to specify the email group to receive the email notification.

4.6 Scheduling BART obsolete backups deletion

Use the Schedule Obsolete Backup Deletion dialog box to schedule or modify a BART obsolete backup deletion.

Use the context menu from the database server where BART was configured.

Provide information on the **General** tab to describe the scheduling details:

- Use the **Enabled?** switch to indicate whether to enable the schedule.
- Use the calendar selector in the **Start** field to specify the starting date and time for the schedule.
- Use the calendar selector in the **End** field to specify the ending date and time for the schedule.

Use the **Repeat** tab to specify the details about the schedule in a cron-style format. The schedule executes on each date or time element selected on the **Repeat** tab. Click in a field to open a list of valid values. Select a value to add it to the list of selected values for the field.

Use the fields in the **Days** box to specify the days when the schedule executes:

- Use the **WeekDays** field to select the days when the schedule executes.
- Use the **MonthDays** field to select the numeric days when the schedule executes. Specify **LastDay** to perform the schedule on the last day of the month, regardless of the date.
- Use the **Months** field to select the months when the schedule executes.

Use the fields in the **Times** box to specify the times when the schedule executes:

- Use the **Hours** field to select the hour when the schedule executes.
- Use the **Minutes** field to select the minute when the schedule executes.

Schedule Obsolete Backup Deletion - Postgres Enterprise Manager Server

General Repeat

The delete obsolete backups job is disabled by default. You can enable it to delete obsolete backups automatically at scheduled intervals.

Enabled? ☒ Yes

Start 2020-08-04 17:21 +05:30

End YYYY-MM-DD HH:mm Z

? Cancel OK

Fig. 10: Schedule Obsolete Backup dialog - General Tab

Schedule - Obsolete Backup Deletion

General Repeat

Schedules are specified using a **cron-style** format.
 For each selected time or date element, the schedule will execute.
 e.g. To execute at 5 minutes past every hour, simply select '05' in the Minutes list box.
 Values from more than one field may be specified in order to further control the schedule.
 e.g. To execute at 12:05 and 14:05 every Monday and Thursday, you would click minute 05, hours 12 and 14, and weekdays Monday and Thursday.
 For additional flexibility, the Month Days check list includes an extra Last Day option. This matches the last day of the month, whether it happens to be the 28th, 29th, 30th or 31st.

Days

Week days ☒ Sunday ☒ Monday ☒ Tuesday ☒ Wednesday ☒ Thursday ☒ Friday ☒ Saturday

Month days Select the month days...

Months ☒ January ☒ February ☒ March ☒ April ☒ May ☒ June ☒ July ☒ August ☒ September ☒ October ☒ November ☒ December

Times

Hours ☒ 02

Minutes ☒ 00

? Cancel OK

Fig. 11: Schedule Obsolete Backup dialog - Repeat tab

4.7 Restoring BART backups

You can restore the backups that you created earlier using BART server on a target remote host. When you select a particular BART server, all the associated backups are listed in the dashboard under Initiated Server Backups.

To restore a backup, select **Restore** next to the backup that you want to restore.

The screenshot shows the 'Restore Backup' dialog box for a backup named '2020_10_21_11_56_03 (Postgres Enterprise Manager Server (as11))'. The 'General' tab is active. The fields are as follows:

- Target agent:** A dropdown menu with the text 'Select agent from the list'.
- Remote user:** A text input field with the placeholder 'Agent where you want to restore the backup'.
- Remote host address:** A dropdown menu with the text 'Enter/Select from the list'.
- SSH port:** A text input field with the value '22'.
- Restore path:** A text input field with the placeholder 'Path where you want to restore the backup'.
- Number of workers:** A spinner box with the value '1'.
- Setup passwordless SSH?:** A toggle switch currently set to 'No'.
- Verify checksum:** A toggle switch currently set to 'Enable'.
- Tablespace paths:** A table with two columns: 'Tablespace Name (OID)' and 'Path'.

At the bottom right, there are 'Cancel' and 'Restore' buttons.

Fig. 12: The Restore Backup dialog - General

In the Restore Backup dialog box, provide information in the fields on the **General** tab:

- Use the **Targetagent** field name to specify the name of the agent where you want to restore the backup.
- Use the **Remoteuser** field to specify the user account on the remote database server host where you want to restore the backup.
- Use the **Remotehostaddress** field to specify the IP address of the remote host where you want to restore the backup.
- Use the **SSHport** field to specify the SSH port to use for restoring the backup.
- Use the **Restorepath** field to specify the path where you want to restore the backup.
- Use the **Numberofworkers** field to specify processes to run in parallel to stream the modified blocks of an incremental backup to the restore location.
- Use the **SetuppaswordlessSSH?** switch to specify if you want to create SSH certificates to allow passwordless logins between the BART server and the target host for restore.

On the **Advanced** tab, specify your preferences for advanced options for restoring the backup:

- Use the **CopyWALstorestorepath?** switch to specify if you want to copy WAL files to the restore path.
- Use the **Pointintimerecovery** switch to specify if you want point-in-time recovery.

Fig. 13: The Restore Backup dialog - Advanced

- Use the **TimelineID** field to specify the timeline ID to use for replaying the archived WAL files for point-in-time recovery.
- Use the **TransactionID(XID)** field to specify the transaction ID for point-in-time recovery.
- Use the **Timestamp** field to specify the timestamp to use for restore.

Note: You can specify either **Transaction ID** or **Timestamp** for the point-in-time recovery.

Use the **Notifications** tab to specify the email notification settings for restoring the backup:

- Use the **Sendthenotifications** field to specify when you want to send the email notifications.
- Use the **Emailgroup** field to specify the email group to receive the email notification.

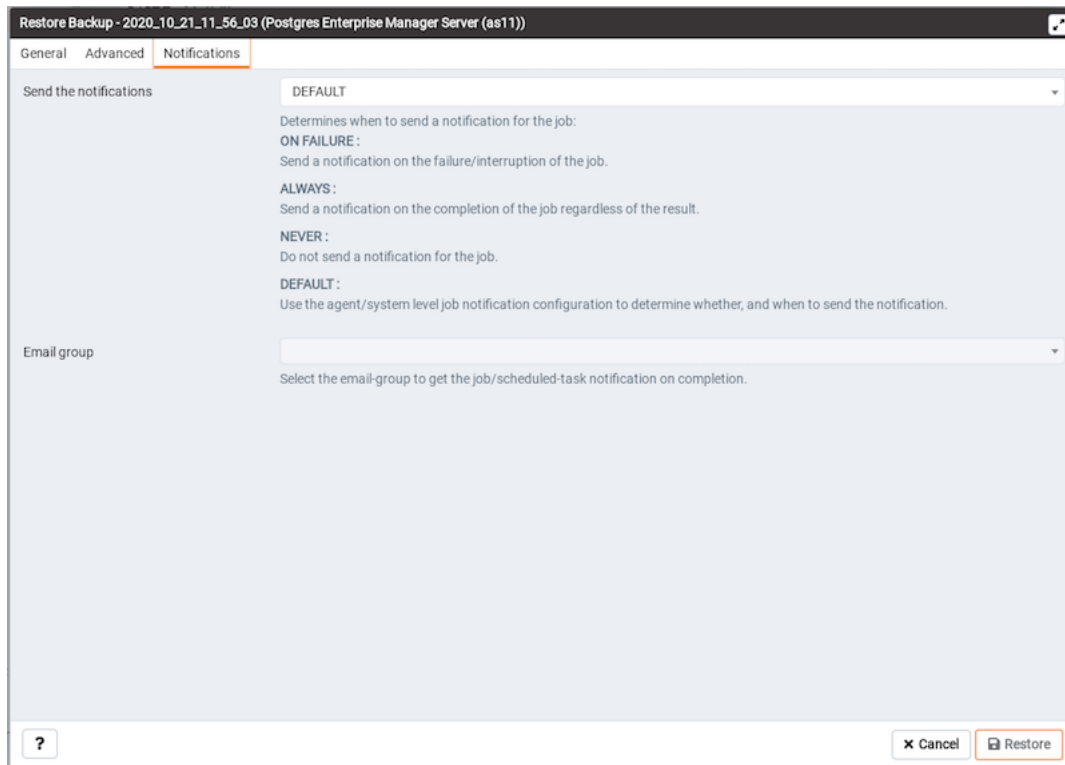


Fig. 14: The Restore Backup dialog - Notifications

MONITORING BDR

BDR (Bi-Directional Replication) is a PostgreSQL extension that provides multi-master replication between the PostgreSQL clusters. Before you monitor BDR through PEM console, you must ensure that your BDR nodes are up and running. For more information, see *BDR probes*.

You can configure PEM to display status information about one or more BDR nodes using dashboards in PEM version 8.1.0 and BDR version 3.7.9 and later.

To configure PEM to monitor BDR nodes, use the PEM web client to create a server definition. Use the tabs on the New Server Registration dialog box to specify general connection properties for the BDR node with the following exceptions:

- Specify the BDR-enabled database name in the **Database** field of the **PEMAgent** tab.
- Specify the user having the bdr_monitor or bdr_superuser role in the **username** field of the **PEMAgent** tab.

After saving the server definition, the BDR node is included in the list of servers under the PEM server directory in the PEM client object browser tree. You can monitor the BDR nodes from BDR dashboards.

To include monitoring information on the BDR dashboards, you must enable the relative probes for each BDR group. See the complete list of BDR

Probes.

To enable a probe, right-click the node name, and select **Management>ManageProbes**.

To monitor the BDR Node, right-click the name of the BDR node in the object browser tree. From the **Dashboards** menu, select the BDR Admin, BDR Group Monitoring, or BDR Node Monitoring dashboard.

MONITORING FAILOVER MANAGER

If you're using EDB Failover Manager to monitor your replication scenario, you must manually install and configure Failover Manager. For detailed information about installing Failover Manager, see the

[EDB website](#).

To monitor the status of a Failover Manager cluster on the Streaming Replication dashboard, you must provide the following information on the **Advanced** tab of the **ServerProperties** dialog box for each node of the cluster:

- Use the **EFMClusterName** field to specify the name of the Failover Manager cluster. The cluster name is the prefix of the name of the cluster properties file. For example, if your cluster properties file is named **efm.properties**, your cluster name is **efm**.
- Use the **EFMInstallationPath** field to specify the location of the Failover Manager binary file. By default, the Failover Manager binary file is installed in **/usr/efm-<X>/bin**, where **<X>** is the EFM version.

After registering your servers, the Streaming Replication Analysis dashboard displays status information about your EFM cluster near the bottom of the dashboard.

▼

Failover Manager Cluster Status

Failover Manager Cluster Information

Properties	Values
Cluster Name	efm
Failover Manager Agent Running Status	UP
Allowed Node List	172.16.177.194, 172.16.23.156
Standby Priority List	172.16.23.156
Missing Nodes	
Minimum Standbys	0
Membership Coordinator	172.16.177.194
Cluster Status Message	

Failover Manager Node Status

Agent Type	Address	Agent	DB	XLog Location	XLog Receive	Status Information	XLog Information	VIP	VIP Status
Primary	172.16.177.194	UP	UP	0/8000140					False
Standby	172.16.23.156	UP	UP	0/8000140	0/8000140				False

Fig. 1: The Failover Manager cluster status report

The **FailoverManagerClusterStatus** section of the Streaming Replication Analysis dashboard displays information about the monitored cluster.

The Failover Manager Cluster Information table provides information about the Failover Manager cluster:

- The Properties column displays the name of the cluster property.
- The Values column displays the current value of the property.

The Failover Manager Node Status table displays information about each node of the Failover Manager cluster:

- The Agent Type column displays the type of agent that resides on the node. The possible values are Primary, Replica, Witness, Idle, and Promoting.
- The Address column displays the IP address of the node.
- The Agent column displays the status of the agent that resides on the node.
- The DB column displays the status of the database that resides on the node.
- The XLog Location column displays the transaction log location of the database.
- The Status Information column displays any error-related information about the node.
- The XLog Information column displays any error-related information about the transaction log.
- The VIP column displays the VIP address that's associated with the node.
- The VIP Status column displays True if the VIP is active for the node, False if not.

6.1 Replacing a primary node

You can use the PEM client to replace the primary node of a Failover Manager cluster with a replica node. To start the failover process, select **Tools>Server>ReplaceClusterPrimary**. You are prompted to confirm that you want to replace the current primary node.

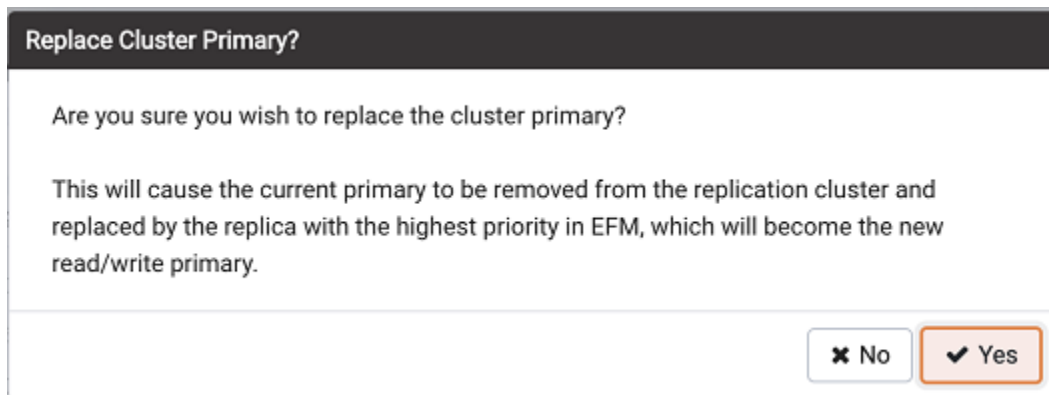


Fig. 2: Replacing the Primary node of a cluster

Select **Yes** to remove the current primary node from the Failover Manager cluster and promote a replica node to the role of read/write primary node in a Failover Manager cluster. The node with the highest promotion priority (defined in Failover Manager) becomes the new primary node. PEM reports the job status.

When the job completes and the Streaming Replication Analysis dashboard refreshes, you can review the Failover Manager Node Status table to confirm that a replica node was promoted to the role of primary in the Failover Manager cluster.

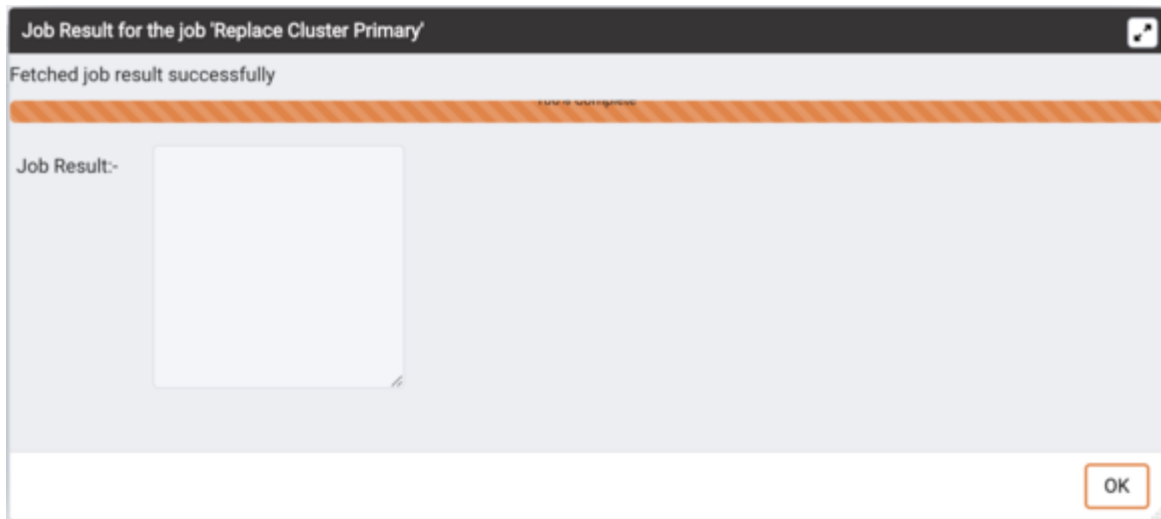


Fig. 3: Confirmation of the promotion

6.2 Switchover EFM cluster

You can use the PEM client to replace the primary node of a Failover Manager cluster with a replica node. To start the switchover process, select **Tools>SwitchoverEFMCluster** . You are prompted to confirm that you want to switch over EFM cluster.

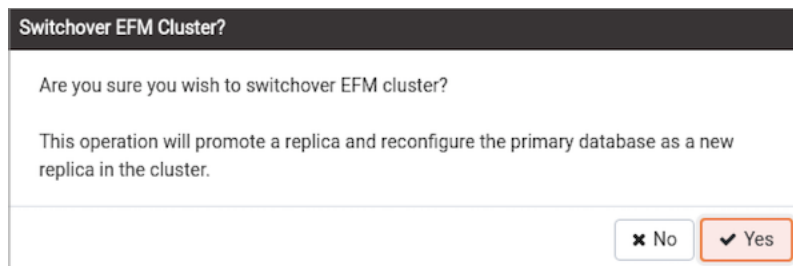


Fig. 4: Switchover EFM Cluster

Select **Yes** to:

- Start the Failover Manager switchover.
- Promote a replica node to the role of read/write primary node.
- Reconfigure the primary database as a new replica in a Failover Manager cluster.

The node with the highest promotion priority (defined in Failover Manager) becomes the new primary node. PEM reports the job status.

When the job completes and the Streaming Replication Analysis dashboard refreshes, you can review the Failover Manager Node Status table to confirm that a switchover occurred.

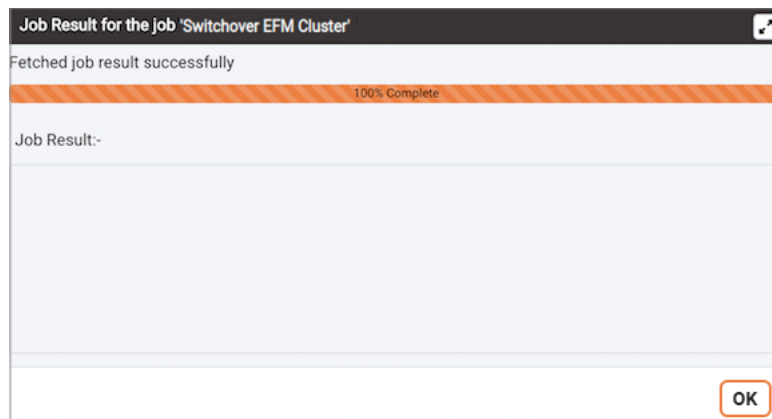


Fig. 5: Confirmation of the promotion

MONITORING REPLICATION SERVER

Before configuring PEM to retrieve statistics from an EDB Postgres Advanced Server or PostgreSQL database that's part of an xDB replication scenario, you must manually install and configure xDB replication. For more information about xDB replication solutions and documentation, see

Multi-master replication operation .

The PEM xDB Replication probe monitors lag data for clusters that use xDB multi-primary or single-primary replication that have an EDB Postgres Advanced Server or PostgreSQL publication database. If you configured replication between other proprietary database hosts (that is, Oracle or SQL Server) and EDB Postgres Advanced Server or PostgreSQL, the probe can't return lag information.

By default, the xDB Replication probe is disabled. To enable the xDB Replication probe:

1. Right-click the name of the server and select **Connect** from the context menu.
2. If prompted, provide authentication information.
3. After connecting, expand the server node of the tree and select the name of the replicated database.
4. Select **Management>ManageProbes** .

Use the **ManageProbes** tab to configure the xDB Replication probe:

- Set **Default** to **No** to modify the minutes and seconds between probe executions.
- Use the **Enabled?** slider to enable the xDB Replication probe.
- Set **Default** slider in the **DataRetention** field to **No** to modify the number of days for PEM to store the information retrieved by the probe.

After enabling the probe, you can use the metrics returned to create custom charts and dashboards in the PEM client.

Properties
SQL
Statistics
Dependencies
Dependents
Monitoring
Manage Probes

Description

Manage Custom Probes: PEM uses probes to retrieve statistics from a monitored server, database, operating system or agent. You can view, reconfigure, delete, or create your own custom probes.

Copy Probes: PEM allows copying of probes from any chosen object recursively down through the object hierarchy. Click on Copy Probes to quickly copy the displayed probe configuration to a selected target.

Quick Links

Manage Custom Probes

Copy Probes

Help

Probes

Probe name	Execution Frequency			Enabled?		Data Retention	
	Default?	Minutes	Seconds	Default?	Probe Enable?	Default?	Days
Background Writer Statistics	☒	5	0	☒	☒	☒	180
Blocked Session Information	☒	5	0	☒	☒	☒	180
Data and Log File Analysis	☒	0	10	☒	☒	☒	180
Database Frozen XID	☒	720	0	☒	☒	☒	180
Database Size	☒	30	0	☒	☒	☒	180
Database Statistics	☒	30	0	☒	☒	☒	90
Failover Manager Cluster Info	☒	5	0	☒	☐	☒	7
Failover Manager Node Status	☒	5	0	☒	☐	☒	7
Lock Information	☒	5	0	☒	☒	☒	180
Number of Prepared Transactions	☒	5	0	☒	☒	☒	180
Number of WAL Files	☒	5	0	☒	☒	☒	180
Object Catalog: Database	☒	5	0	☒	☒	☒	180
Object Catalog: Tablespace	☒	5	0	☒	☒	☒	180
PG HBA Conf	☒	30	0	☒	☒	☒	180
Server Information	☒	5	0	☒	☒	☒	180
Server log Configuration	☒	0	10	☒	☒	☒	180
Session Information	☒	5	0	☒	☒	☒	180
Settings	☒	5	0	☒	☒	☒	180
Streaming Replication	☒	5	0	☒	☐	☒	180
Streaming Replication Database Conflicts	☒	5	0	☒	☐	☒	180
Streaming Replication Lag Time	☒	5	0	☒	☐	☒	180
Tablespace Size	☒	30	0	☒	☒	☒	180
User Information	☒	30	0	☒	☒	☒	180
WAL Archive Status	☒	30	0	☒	☐	☒	180

Fig. 1: The Manage Probes tab

TUNING PERFORMANCE

The following features help in tuning the performance:

- *Tuning Wizard*
- *Postgres Expert - best practice enforcement*
- Performance Diagnostics

8.1 Tuning Wizard

The Tuning Wizard reviews your PostgreSQL or Advanced Server installation, and recommends a set of configuration options that will help tune the installation to best suit its anticipated workload. Please note that benchmarking systems or systems with a high work load may require additional manual tuning to reach optimum performance.

Before using the Tuning Wizard, you must specify the name of the service in the Service ID field on the Advanced tab of the server's Properties dialog. PEM will use the service name when restarting the service after tuning.

The Tuning Wizard can only make recommendations for those servers that reside on the same server as their bound PEM agent. If you have specified a value of Yes in the Remote monitoring field when defining your server, the server will not be displayed in the Tuning Wizard tree control.

To open the Tuning Wizard, select **Tuning Wizard...** from the **Management** menu of the PEM client. The Tuning Wizard opens, welcoming you.

Click **Next** to continue to the server selection dialog.

Expand the **Servers** node of the tree control to view a list of the servers that are currently monitored by PEM that are available for tuning. Check a box to the left of a server name to select the server for tuning.

Note: The Tuning Wizard displays a red warning symbol to the left of a server name in the tree control if the service name for that server is not provided on the server's Properties dialog.

Click **Next** to continue to the Configuration dialog.

Select an option in the **Machine utilization** field to specify the type of work performed by the selected servers. The type of work performed by the server determines how the tuning wizard will allocate system resources:

- Select **Dedicated** to dedicate the majority of the system resources to the database server.
- Select **Mixed use** to dedicate a moderate amount of system resources to the database server.
- Select **Developer workstation** to dedicate a relatively small amount of system resources to the database server.

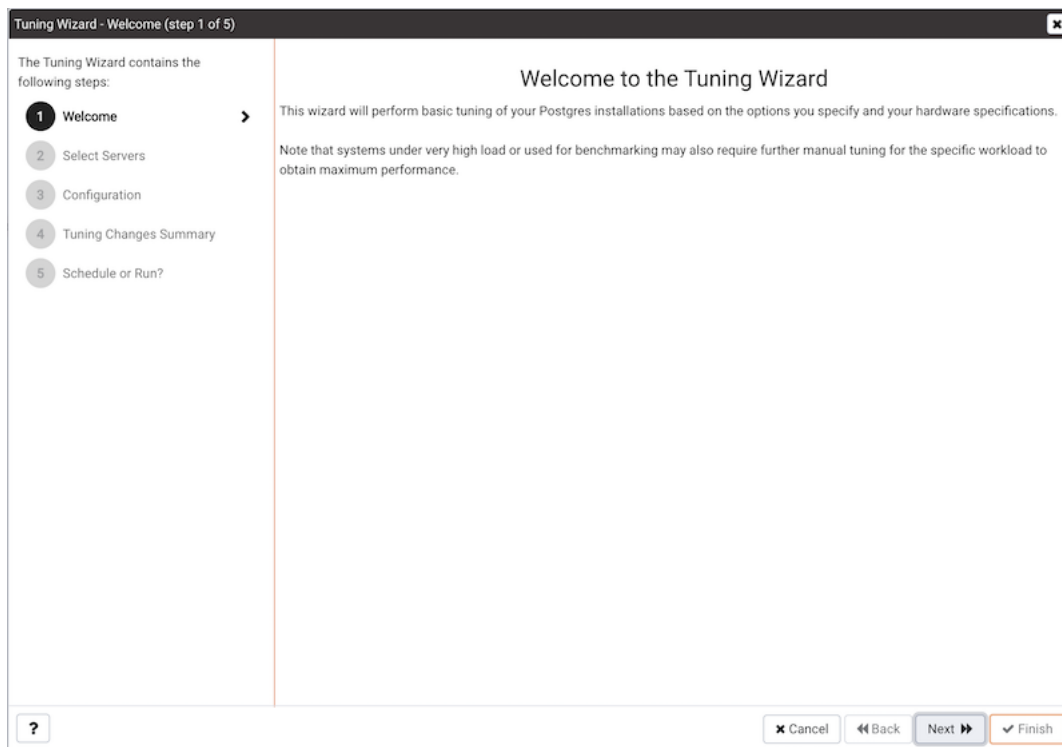


Fig. 1: The Tuning Wizard Welcome dialog

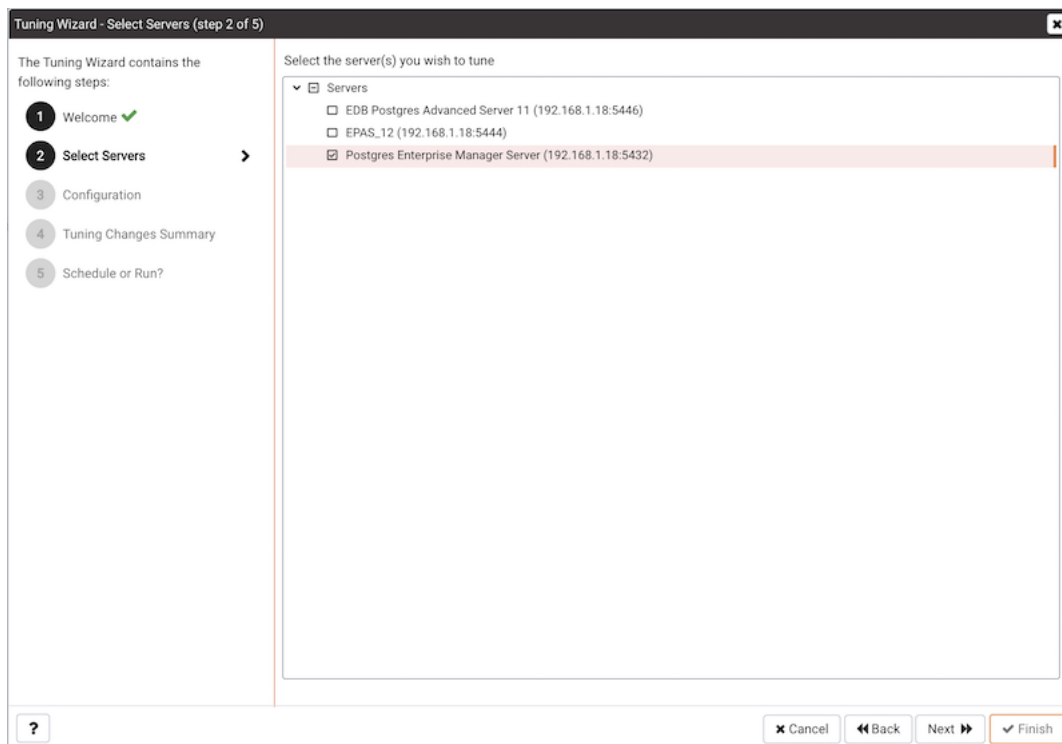


Fig. 2: The Select Servers dialog

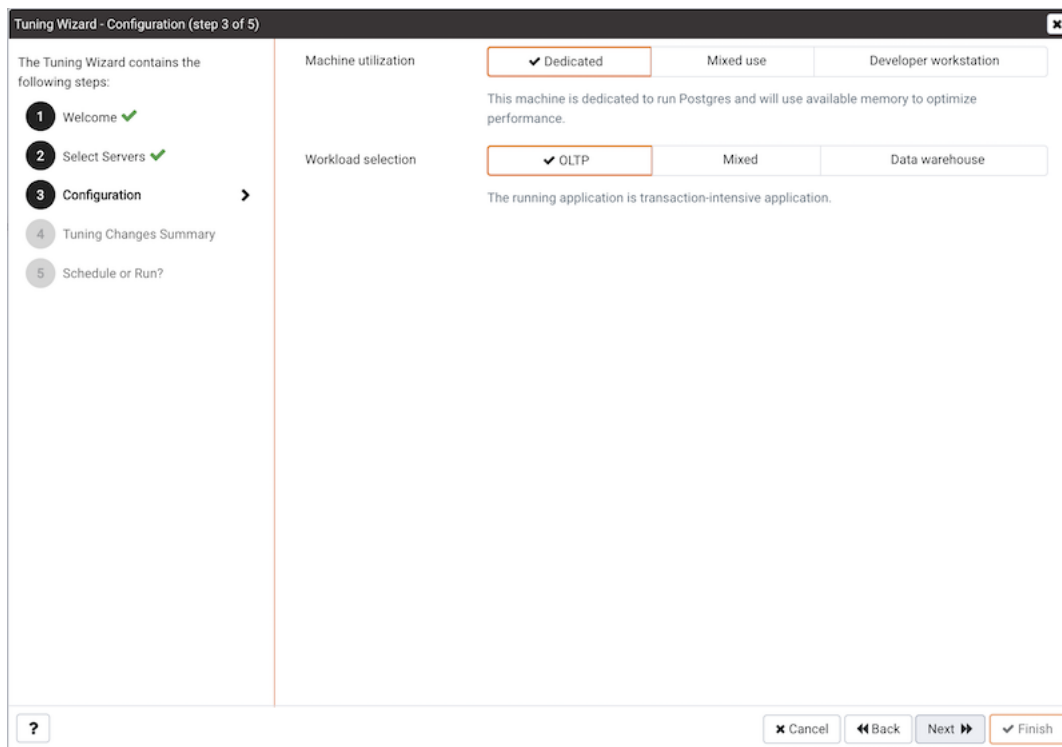


Fig. 3: The Configuration dialog

Select an option in the **Workload Selection** field to specify the type of workload typically performed on the selected server:

- Select **OLTP** if the selected server is used primarily to process online transaction workloads.
- Select **Mixed** if the selected server provides a mix of transaction processing and data reporting.
- Select **Data warehouse** if the server is used for heavy data reporting.

Click **Next** to continue to the **Tuning Changes Summary** dialog.

The tree control on the **Tuning Changes Summary** dialog displays the parameter setting modifications recommended for each server analyzed by the Tuning Wizard. Use the checkboxes next to a server or parameter name to select the recommendations that tuning wizard will either include in a preview report or apply:

- A checked box to the left of a parameter name specifies that the Tuning Wizard will include the parameter setting.
- A checked box to the left of a server name specifies that the Tuning Wizard will include all parameter setting recommendations for the specified server.

Specify which Tuning Wizard recommendations you wish to include in a report or apply, and click **Next** to continue.

Use the **Schedule or Run?** dialog to either specify a time that PEM will apply the changes, or generate a report that details the recommended changes.

The selected actions will apply to all of the changes noted on the **Tuning Changes Summary**. If you opt to generate a report, PEM will create a report that contains a list of the current values and recommended modifications to the configuration parameters selected on the **Tuning Changes Summary** dialog. Note that to implement changes, you will need to invoke the Tuning Wizard a second time, specifying the parameters you wish to modify on the **Tuning Changes Summary** dialog.

Select **Schedule** changes to view and specify your scheduling options.

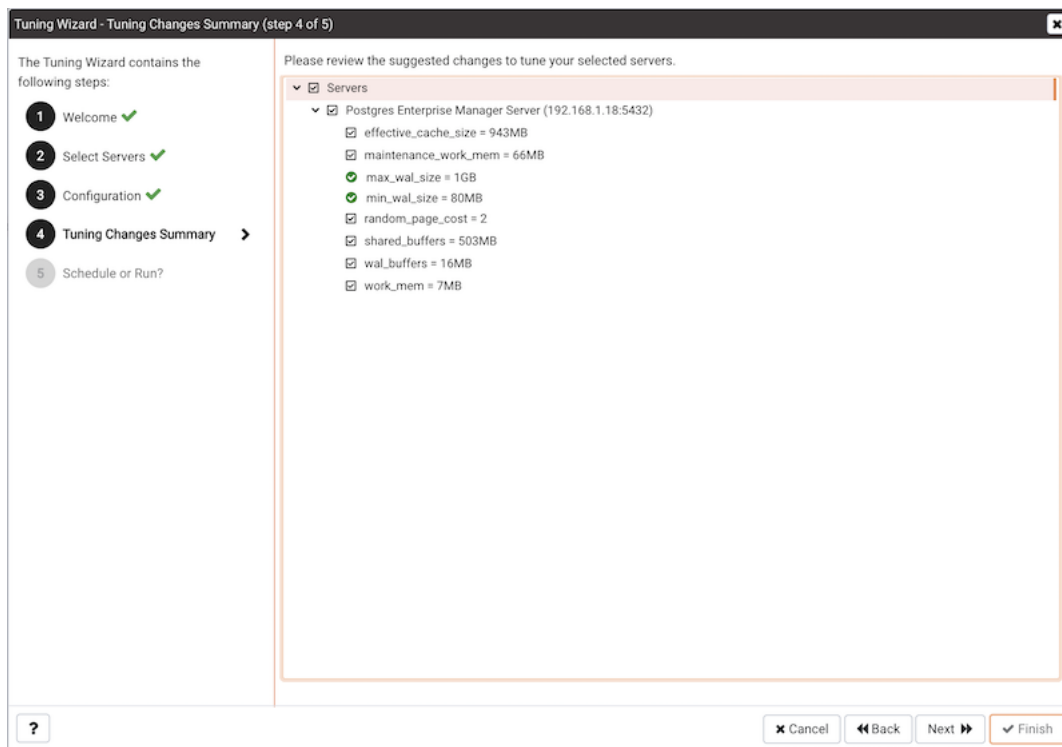


Fig. 4: The Tuning Changes Summary dialog

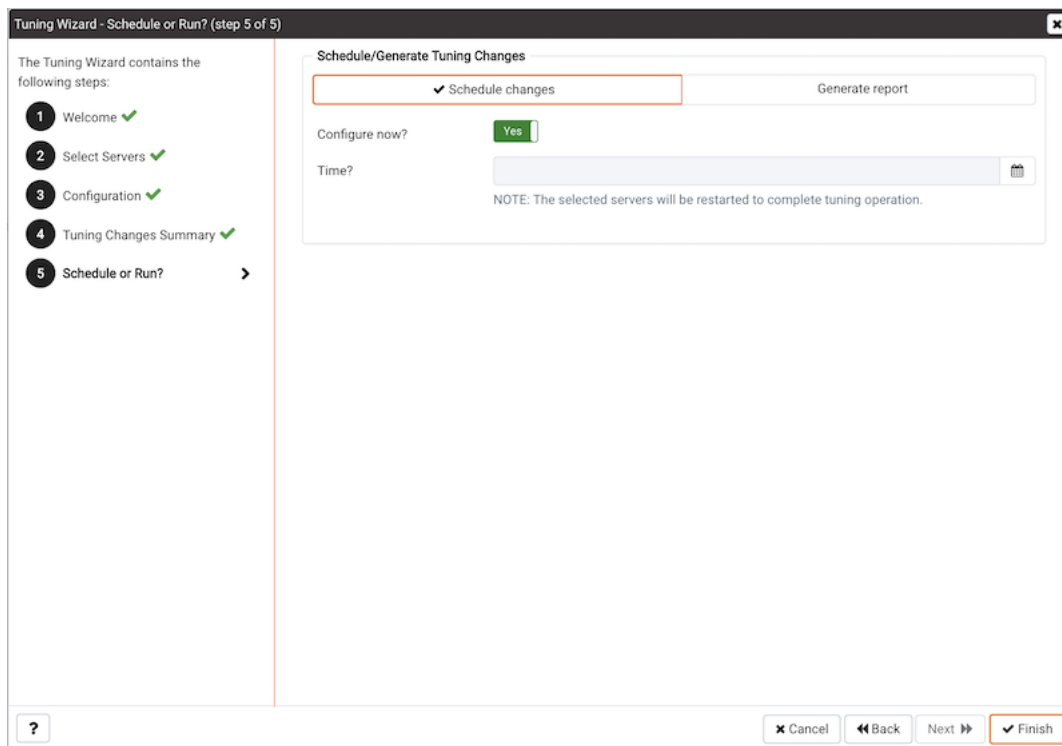


Fig. 5: The Schedule or Run? dialog

You can:

- Set the **Configuration now?** slider to **Yes** to apply the tuning wizard's recommendations and restart the server now.
- Set the **Configuration now?** slider to **No** to enable the **Time?** field and use the calendar selector to specify a time for PEM to apply the tuning wizard's recommendations and restart the server. Note that if you schedule a time for the changes to be applied, you will not be provided with a preview of the change recommendations.

Select **Generate report** to view your report options.

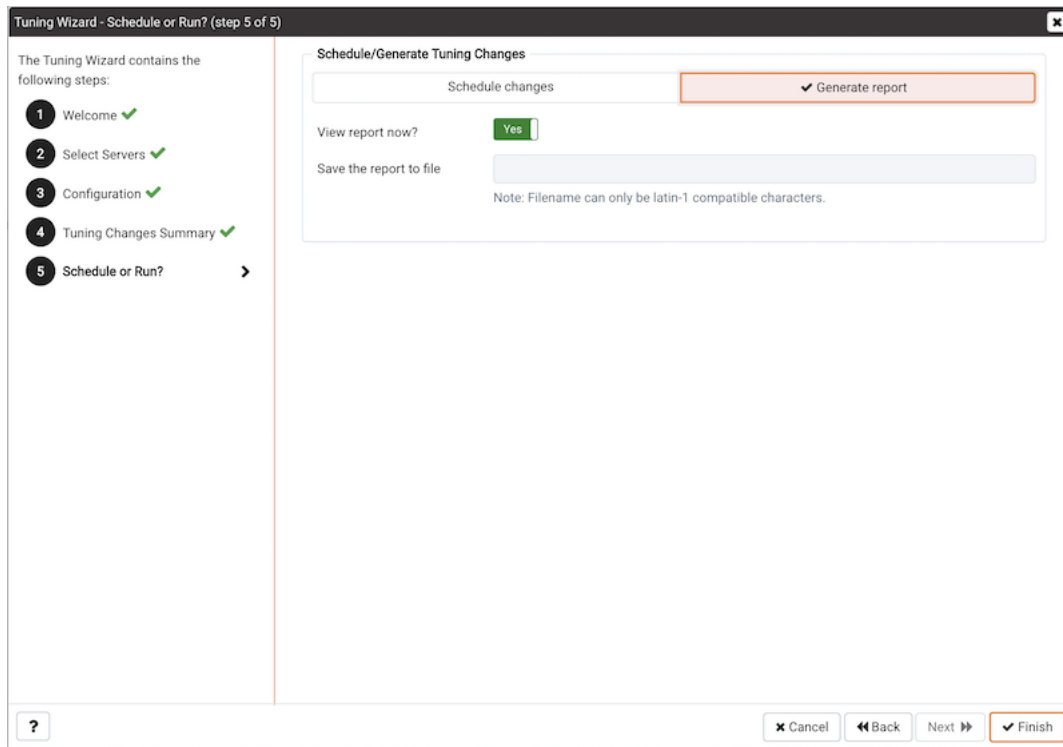


Fig. 6: The Schedule or Run? dialog

You can:

- Set the **View report now?** slider to **Yes** to display the Tuning Wizard report onscreen.
- Set the **View report now?** slider to **No** to enable the **Save the report to file** field and use the calendar selector to specify a file name and location to which PEM will write the Tuning Wizard report.

Click the **Finish** button to either apply the Tuning Wizard's modifications or generate a report and exit the Tuning Wizard.

You can confirm that Tuning Wizard has implemented the recommended changes by reviewing the `postgresql.conf` file for the modified server. The Tuning Wizard adds a comment above each modified parameter in the `postgresql.conf` file when the change is applied.

You can also confirm a parameter value by querying the server. For example, to confirm the value of the `shared_buffers` parameter, open a SQL command line using either the Query Tool (accessed through the Tools menu) or the `psql` client, and issue the command:

```
SHOW shared_buffers;
```

The value returned by the server will confirm that the parameter has been modified.

Tuning Wizard Report		
Generated On: 2020-04-27 16:19:52		Go to: Postgres Enterprise Manager Server
Summary		
Number of servers selected: 1 Machine utilization: Dedicated Workload profile: OLTP		
Server: Postgres Enterprise Manager Server (192.168.1.18:5432)		
GUC Parameter	Original Value	Recommended Value
effective_cache_size	4096MB	943MB
maintenance_work_mem	64MB	66MB
random_page_cost	4	2
shared_buffers	128MB	503MB
wal_buffers	4MB	16MB
work_mem	4MB	7MB

Fig. 7: The Tuning Wizard report

```

root@localhost:/opt/PostgresPlus/9.5AS/data
File Edit View Search Terminal Help
# The value for shared_buffers was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#shared_buffers = 196MB
# The value for shared_buffers was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:18:23
shared_buffers = 196MB

#huge_pages = try                # (change requires restart)
                                # on, off, or try
                                # (change requires restart)
#temp_buffers = 8MB              # min 800kB
#max_prepared_transactions = 0   # zero disables the feature
                                # (change requires restart)
# Note: Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.
#work_mem = 4MB                  # min 64kB
# The value for work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#work_mem = 3MB
# The value for work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:18:23
work_mem = 3MB
#maintenance_work_mem = 64MB     # min 1MB
# The value for maintenance_work_mem was changed by the Postgres Enterprise Manager Tuning Wizard at 2017-03-06 09:15:32
#maintenance_work_mem = 25MB
--More--(21%)

```

Fig. 8: Confirming a change in the postgresql.conf file

8.2 Postgres Expert - best practice enforcement

The Postgres Expert utility provides expert advice on how to best configure your Postgres servers for optimal performance, security, and more. Postgres Expert serves as a PostgreSQL ‘DBA in a box’ by analyzing your servers for deviations in best practices. Postgres Expert contains three specialized Experts:

- The Configuration Expert.
- The Schema Expert.
- The Security Expert.

You can select specific rules for each Expert to analyze, or accept all rules, and then review Postgres Expert reports detailing any best practice issues that require your attention.

8.2.1 Using the Postgres Expert wizard

To use the Postgres Expert wizard select the **Postgres Expert** option from the **Management** menu in the PEM client. When the wizard’s **Welcome** window opens, click **Next** to continue.

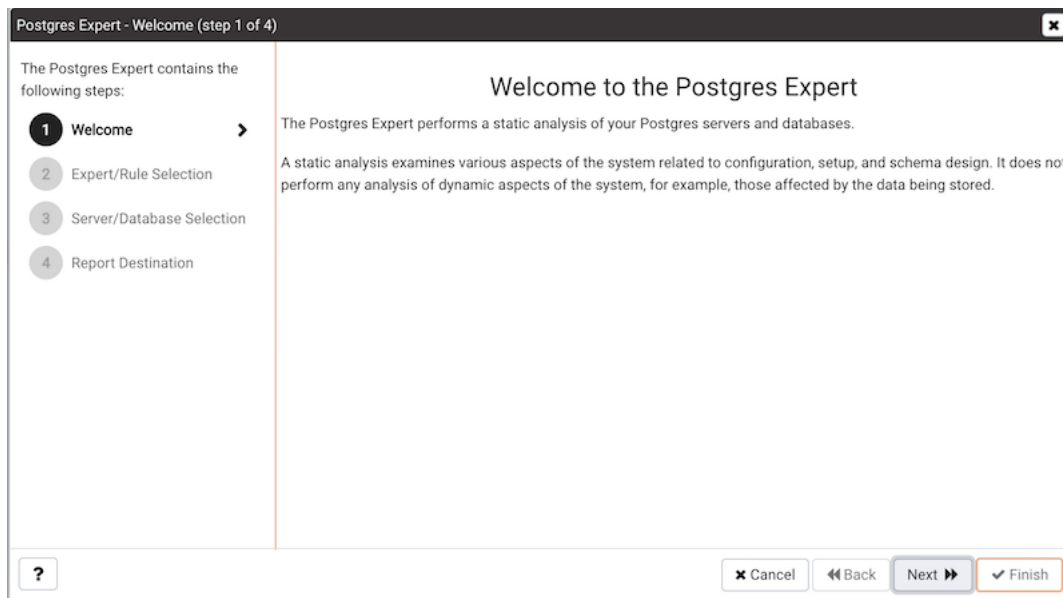


Fig. 9: The Postgres Expert Welcome dialog

The wizard displays a tree control that allows you to choose the **Experts** and **Rules** with which Postgres Expert will evaluate the specified server or database.

The tree control categorizes the rules under three Expert headings:

- Select from the **Configuration Expert** rules to analyze the parameter settings of the server or operating system to find any adjustments that might improve system performance.
- Select from the **Schema Expert** rules to analyze schema objects (locating missing primary keys, foreign keys without indexes, etc).
- Select from the **Security Expert** rules to review the system to find security vulnerabilities.

Use the checkmark indicator to the left of an expert or rule to indicate that the Postgres Expert should analyze the configuration of the selected servers for any best practice deviations related to the checked item.

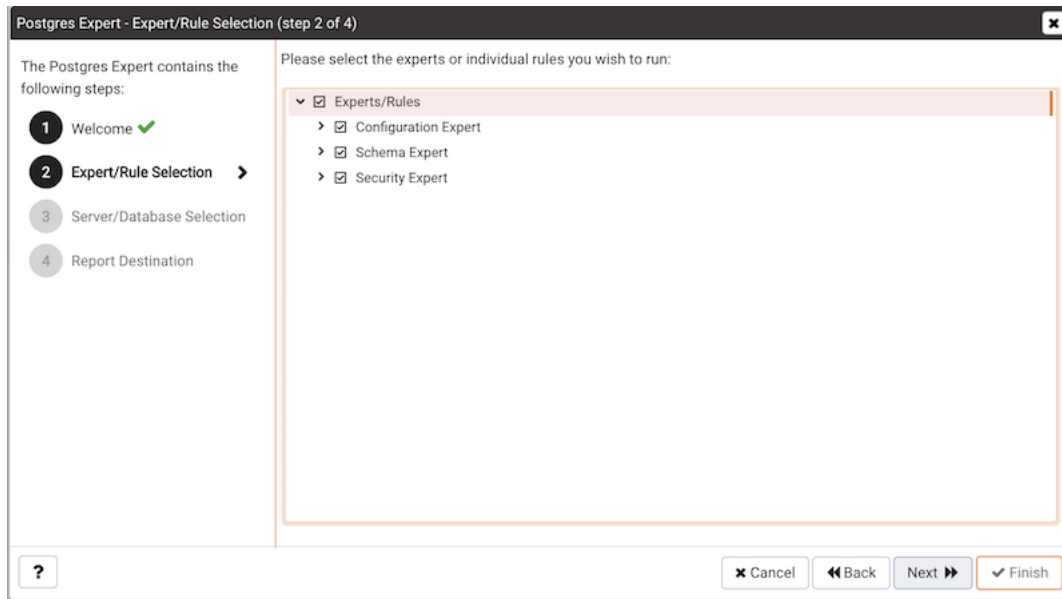


Fig. 10: The PEM Agent Installer's Welcome dialog

You can:

- Check the box next to the name of an expert to select or deselect all of the configuration items listed under that node of the tree control.
- Check the box next to Servers/Databases to instruct Postgres Expert to review the selected server for all of the items in the tree control.
- Deselect the box next to Servers/Databases to to un-check all of the rules; then, navigate through the tree control, specifying only the items that you wish Postgres Expert to evaluate.

After making your selections, click **Next** to continue to the Server/Databases tree control.

Select or de-select the servers and databases that you would like Postgres Expert to analyze. If you select multiple servers or databases, the resulting report will contain a separate analysis of each target. When you've finished, click **Next** to select a report destination.

You can select the default option and click **Finish** to view an onscreen report from Postgres Expert, or check the box next to Download the report to save a copy of the report to an HTML file for later use. If you choose to save the report to a file, the download will begin immediately. The file will be saved in your default download directory.

8.2.2 Reviewing Postgres Expert recommendations

Postgres Expert produces an easily navigated report that contains an analysis of the selected rules, categorized by high, medium, and low severities, for the selected servers.

The report header contains a summary of the report, and includes the date and time that the report was generated, the number of rules analyzed, and the number of deviations from best practices found by Postgres Expert. Use the Jump to drop-down listbox to select a server to navigate to the section of the report that targets recommendations for that server.

The body of the report contains the detailed findings for each server selected for analysis. The findings are sorted by Expert; within each Expert heading, any rule violations are ranked by Severity.

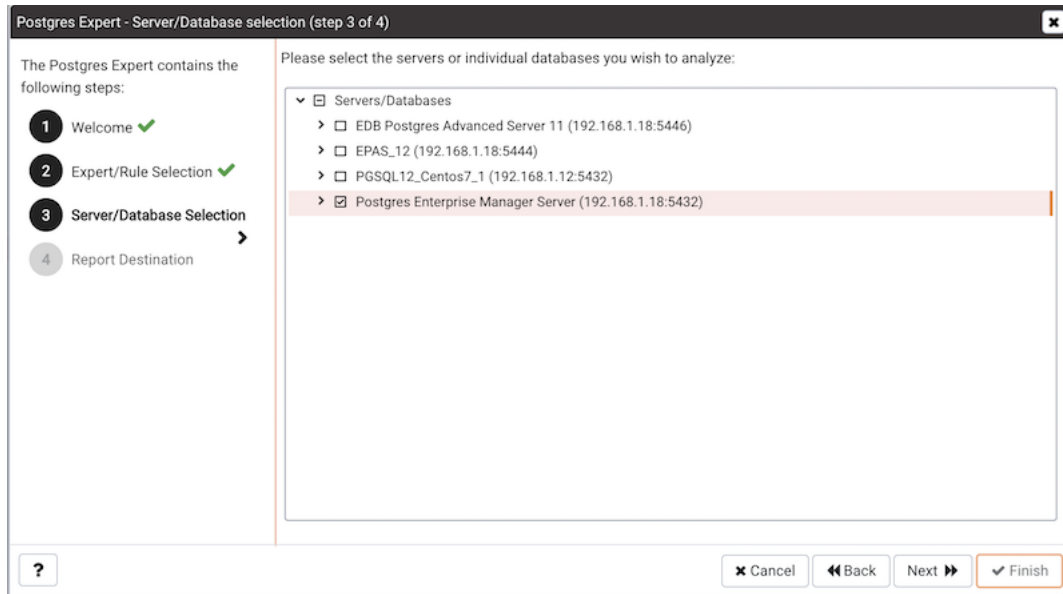


Fig. 11: The Servers/Databases dialog

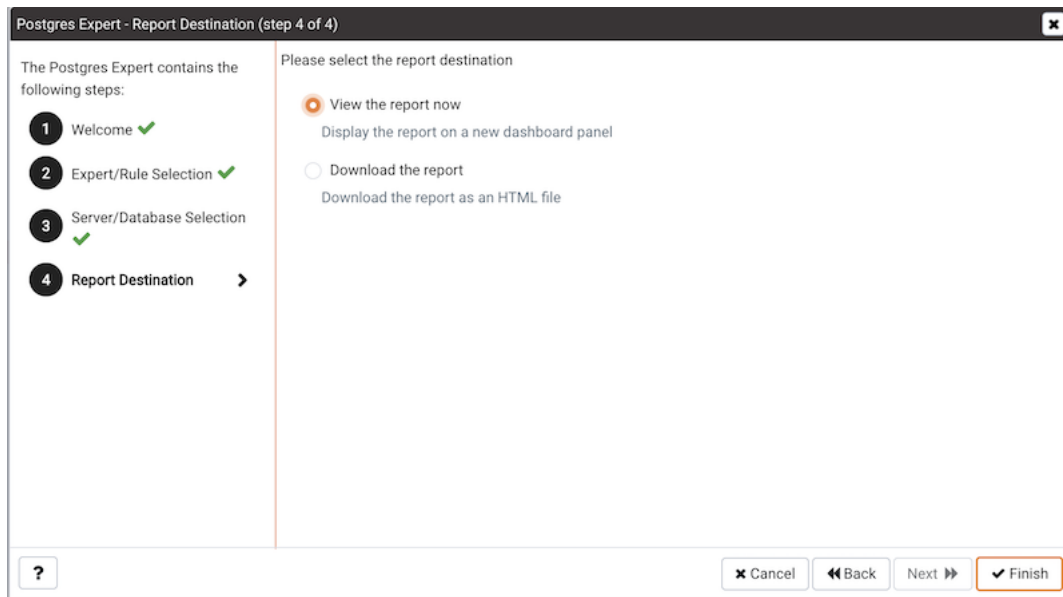


Fig. 12: Specify a report destination

Postgres Expert Report		
Generated On: 2020-04-27 16:57:47		Go to: Postgres Enterprise Manager Server
Summary		
Servers Tested: 1 Rules Checked: 31 High Alerts: 1 Medium Alerts: 3 Low Alerts: 2		
Server: Postgres Enterprise Manager Server (192.168.1.18:5432)		
Advisor: Configuration Expert		
Rule	Database	Severity
Check checkpoint_completion_target	-	Medium
Check effective_cache_size	-	Medium
Check effective_io_concurrency	-	Low
Check reducing_random_page_cost	-	Low
Advisor: Schema Expert		
Rule	Database	Severity
Check data and transaction log on same drive	-	High
Check for missing foreign key indexes	db01	Medium

Fig. 13: The Postgres Expert report

Postgres Expert Report

Generated On: 2020-04-27 16:57:47

Go to: Postgres Enterprise Manager Server

Summary

Servers Tested: 1 Rules Checked: 31 High Alerts: 1 Medium Alerts: 3 Low Alerts: 2

Server: Postgres Enterprise Manager Server (192.168.1.18:5432)

Advisor: Configuration Expert

Rule	Database	Severity
Check checkpoint_completion_target	-	Medium

Recommended Value: Consider adjusting checkpoint_completion_target.

Current Values:

Settings	Value
checkpoint_completion_target	0.5

Trigger: checkpoint_completion_target != 0.9

Description:

In order to ensure reliable and efficient crash recovery, PostgreSQL periodically writes all dirty buffers to disk. This process is called a checkpoint. Beginning in PostgreSQL 8.3, checkpoints take place over an extended period of time in order to avoid swamping the I/O system. checkpoint_completion_target controls the rate at which the checkpoint is performed, as a function of the time remaining before the next checkpoint is due to start. A value of 0 indicates that the checkpoint should be performed as quickly as possible, whereas a value of 1 indicates that the checkpoint should complete just as the next checkpoint is scheduled to start. It is usually beneficial to spread the checkpoint out as much as possible; however, if checkpoint_completion_target is set to a value greater than 0.9, unexpected delays near the end of the checkpoint process can cause the checkpoint to fail to complete before the next one needs to start. Because of this, the recommended setting is 0.9.

Check effective_cache_size	-	Medium
Check effective_io_concurrency	-	Low
Check reducing_random_page_cost	-	Low

Advisor: Schema Expert

Rule	Database	Severity
Check data and transaction log on same drive	-	High
Check for missing foreign key indexes	db01	Medium

Fig. 14: The detailed recommendation for a rule

Click on each rule in the Postgres Expert report to display details and recommendations for that rule. Within each rule, section headings display:

- The **Advisor** section lists the name of the Postgres Expert advisor that prompted the recommendation.
- The **Trigger** section displays a description of the rule that raised the alert.
- The **Recommended Value** section displays the value to which Postgres Expert recommends setting the selected parameter.
- The **Description** section displays information and advice about the parameter that caused the alert.
- The **Current Values** section displays the current value(s) of any parameter(s) that influence the Postgres Expert's evaluation.

8.3 Performance Diagnostic

You can use the Performance Diagnostic dashboard to analyze the database performance for Postgres instances by monitoring the wait events. To display the diagnostic graphs, PEM uses the data collected by EDB Wait States module.

Performance Diagnostic feature is supported for Advanced Server databases from PEM 7.6 version onwards and for PostgreSQL databases it is supported from PEM 8.0 onwards.

Note: For PostgreSQL databases, Performance Diagnostics is supported only for versions 10, 11, and 12 installed on the supported CentOS or RHEL platforms.

For more information on EDB Wait States, see [EDB wait states docs](#).

You can analyze the Wait States data on multiple levels by narrowing down your selection of data. Each level of the graph is populated on the basis of your selection of data at the higher level.

Prerequisite:

- For PostgreSQL, you need to install `edb_wait_states_<X>` package from `edb.repo` where `<X>` is the version of PostgreSQL Server. You can refer to [EDB Build Repository](#) for the steps to install this package. For Advanced Server, you need to install `edb-as<X>-server-edb-modules`, Where `<X>` is the version of Advanced Server.
- Once you ensure that EDB Wait States module of EDB Postgres Advanced Server is installed, then configure the list of libraries in the `postgresql.conf` file as below:

```
shared_preload_libraries = '$libdir/edb_wait_states'
```

Restart the database server, and then create the following extension in the maintenance database:

```
CREATE EXTENSION edb_wait_states;
```

- You must have super user privileges to access the Performance Diagnostic dashboard.

You get the following error while accessing the Performance Diagnostic dashboard if the above prerequisites are not met:

To open the Performance Diagnostic dashboard, select **Server** and then **Performance Diagnostic...** from the **Tools** menu of the PEM client.

By default, the top most Performance Diagnostic graph pulls the data of last one hour, starting from current date and time. This graph shows the time series containing the number of active sessions. Each point of this time series represents the active sessions and wait events at a particular time and last 15 seconds. These sessions may or may not be waiting for an wait event, or using the CPU at a particular point in time. This time series is generated based on the wait event samples collected by the `edb_wait_states` extension.

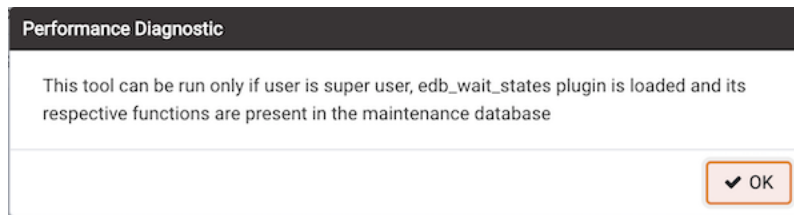


Fig. 15: Performance Diagnostic Error dialog

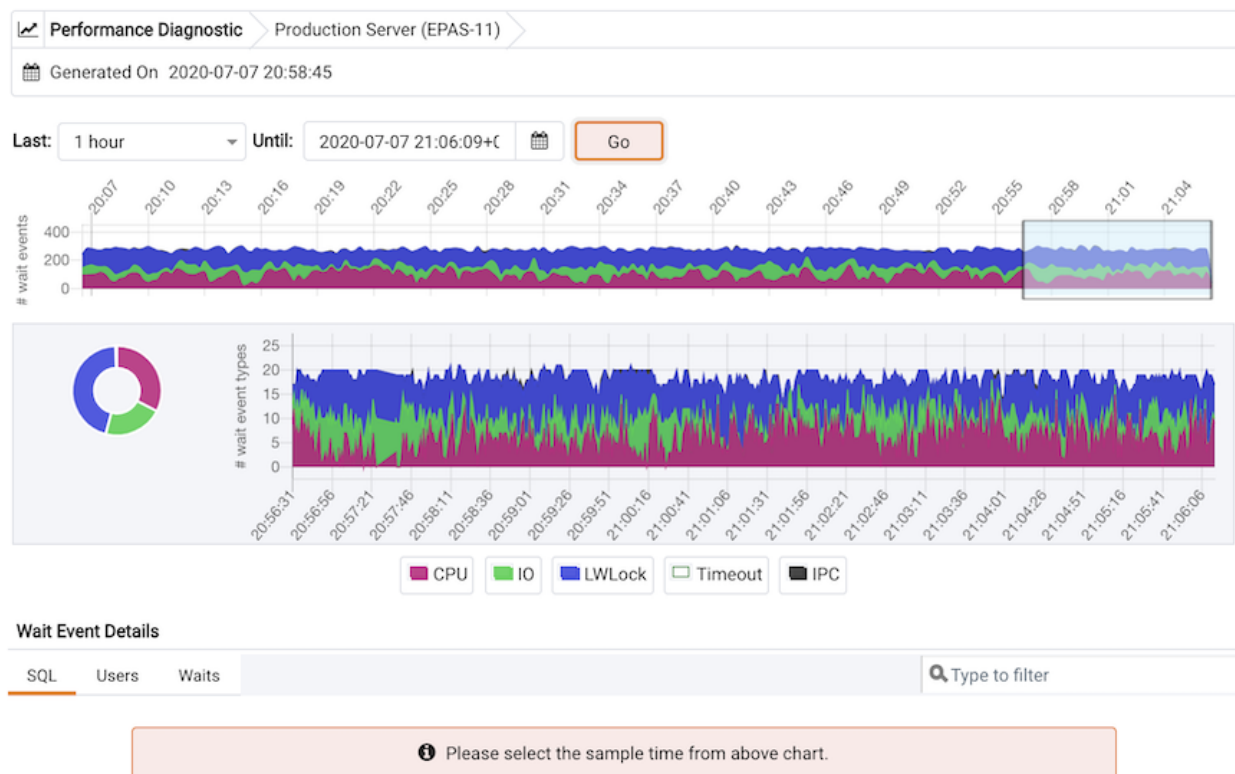


Fig. 16: Performance Diagnostic dashboard

You can also use the Preferences dialog to display Performance Diagnostic in a new browser tab. Use **Open in New Browser Tab?** to display the Performance Diagnostics dashboard in a new browser tab.

The range selection in the first graph is 10 minutes. You can use the **Last** drop-down list box to select the duration for which you want to see the graph: select the last 1 hour, last 4 hours, last 12 hours, or last 24 hours. You can also select the date and time through which you want the data to be displayed.

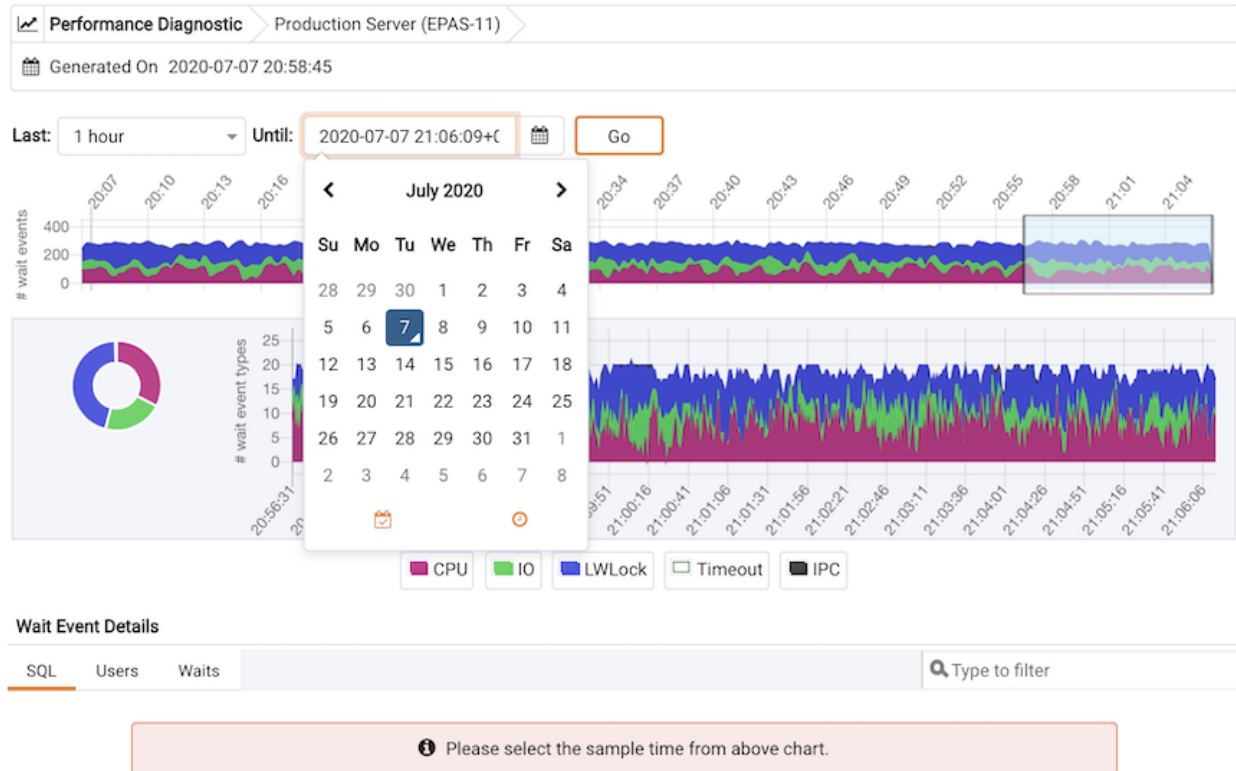


Fig. 17: Performance Diagnostic dashboard - Time Changes option

The first graph displays the number of active sessions (and - wait event types) for the selected time interval. You can narrow down the timeline in the first graph to analyze the data for a specific time period.

Next section plots the following graphs based on the selected time interval in the first graph:

1. Donut graph - It shows total wait event types according to the time range selection in the first graph. It helps you understand how much time was spent by those session on waiting for an event.
2. Line graph - It plots a time series with each point representing the active sessions for each sample time.

To differentiate each wait event types and the CPU usage clearly, the graph for each wait event type is displayed in a different color.

Select a particular time on the **Line** graph for which you wish to analyze the wait events; the third section displays the wait event details in the Performance Diagnostics dashboard on the basis of your selected particular time in the second graph. The third section displays wait event details on three tabs:

- The **SQL** tab displays the list of SQL queries having wait events for the selected sample time.
- The **Users** tab displays the details of the wait events grouped by users for selected sample time.
- The **Waits** tab displays the number of wait events belonging to each wait event type for the selected sample time.



Fig. 18: Performance Diagnostic dashboard - Time range selection in the first Wait event types graph

You can click on the graph legends to show or hide a particular wait event type in all the graphs. This will make the analysis of a specific wait event type easier.

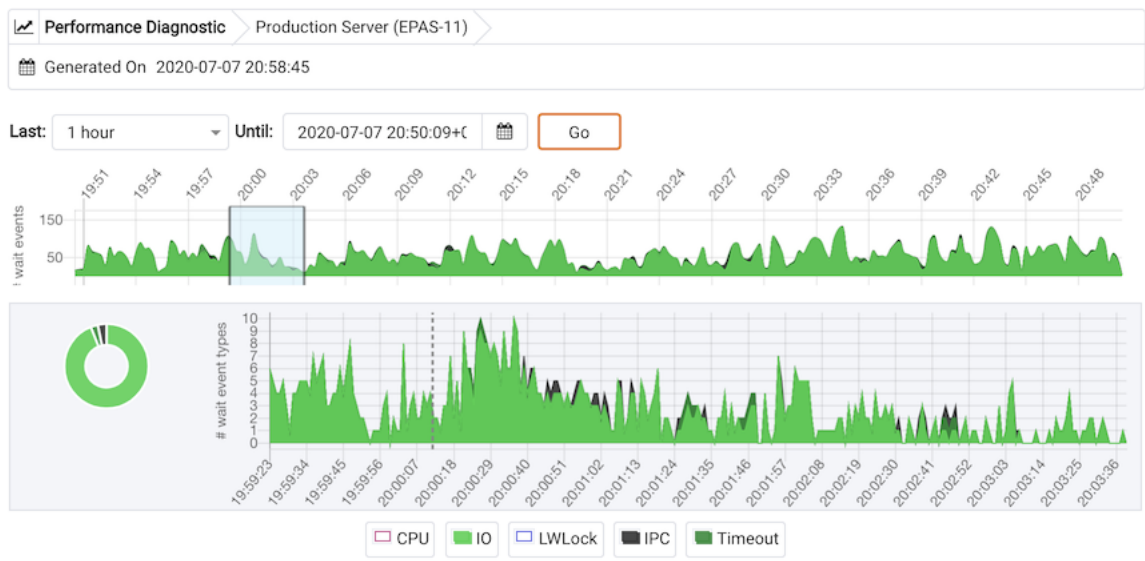


Fig. 19: Performance Diagnostic dashboard - Show and hide a particular wait event type by click the respective graph legend

You can filter the data displayed in the rows under all the three tabs. You can also sort the data alphabetically by clicking on the column headers.

SQL tab

Wait Event Details			
SQL	Users	Waits	
Load By Waits ▾		SQL	Number of sessions
👁		<code>select nation, o_year, sum(amount) as sum_profit from</code>	2
👁		<code>select o_year, sum(case when nation = \$1 then volume</code>	1
👁		<code>select l_shipmode, sum(case when o_orderpriority = \$1</code>	1
👁		<code>select supp_nation, cust_nation, l_year, sum(volume)</code>	1

Fig. 20: Performance Diagnostic - SQL tab with filter applied

Users tab

Waits tab

Click on the Eye icon in any row of the SQL tab to display a new tab with details of the query of that particular row. This page displays Query ID and its corresponding session IDs in a dropdown list at that particular selected sample time in the Query information section. You can select the session ID for the selected query for which you want to analyze the data. You will see the details corresponding to the selected session ID and query ID. The Query information table also displays the SQL query. If the SQL query is being displayed partially, click the down arrow at the bottom of the section to view the complete SQL query.

The Wait event types section displays the total number of wait event types for the selected session ID and query ID. It shows two type of graphs:



Fig. 21: Performance Diagnostic - Users tab

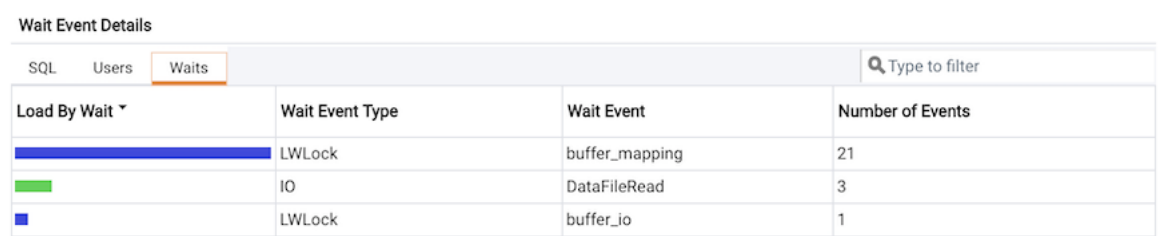


Fig. 22: Performance Diagnostic - Waits tab

1. Donut graph - It shows the proportions of categorical data, with the size of each piece representing the proportion of each wait event type.
 2. Timeline bar graph - It can be used to visualize trends in counts of wait event types over time.
- To differentiate clearly, each wait event type is represented by a different color in the bar graph.

The **Wait events** section has a table displaying all the wait events occurred during the query execution. It displays data in decreasing order by number of the wait events. Second table displays the wait event with sample time occurred over the period of whole query execution. It allows to analyze the wait events during the query execution over the period of time. It shows the actual samples collected by the EDB Wait States extension for that particular query ID and session ID.

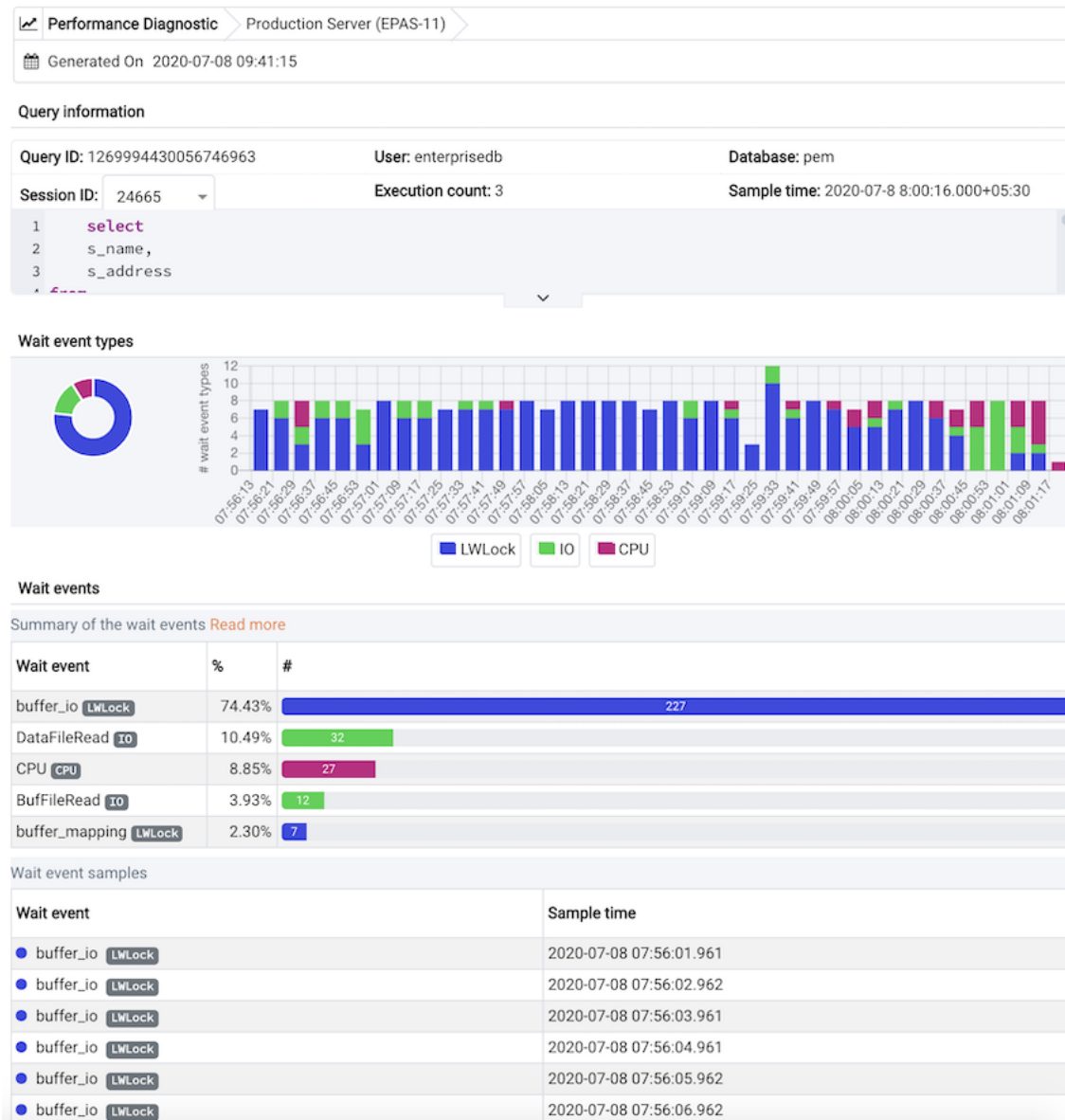


Fig. 23: Performance Diagnostic - Query dashboard

PROFILING WORKLOADS

The following features are available to profile the workloads:

- *SQL Profiler*
- *Using the Index Advisor*

9.1 SQL Profiler

Postgres Enterprise Manager (PEM) is designed to assist database administrators, system architects, and performance analysts when administering, monitoring, and tuning PostgreSQL and EDB Postgres Advanced Server database servers. PEM was designed to manage and monitor a single server or multiple servers from a single console, allowing complete control over monitored databases.

The SQL Profiler Plugin works with PEM to allow you to profile a server's workload. You can install the SQL Profiler plugin on servers with or without a PEM Agent. However you can run traces only in ad hoc mode on unmanaged servers, and you can schedule them only on managed servers.

SQL Profiler is officially supported only on the EDB distributions of PostgreSQL and EDB Postgres Advanced Server supported versions. The plugin is distributed using StackBuilder or as operating system-dependent packages in EDB's yum repositories. The plugin is also distributed and installed with the EDB Postgres Advanced Server installations.

9.1.1 Installing the SQL Profiler plugin

You must install the plugin on each server on which you want to use SQL Profiler. For example, if you have a host running PostgreSQL 9.6 and PostgreSQL 10, you must install two versions of the plugin, one for each server.

Install the plugin for PostgreSQL before configuring it. If you're using EDB Postgres Advanced Server, you can skip installation and go to

Configuring SQL Profiler .

You can use the graphical installer to install any version of SQL Profiler on the Windows platform. On Linux, use an RPM package to install the SQL Profiler. For detailed information about configuring the EDB repository for your host platform, see the Installing on Linux section.

Installing SQL Profiler on Windows

To invoke the SQL Profiler graphical installer, assume Administrator privileges, navigate to the directory that contains the installer, and double-click the installer icon. The SQL Profiler installer welcomes you to the Setup Wizard.

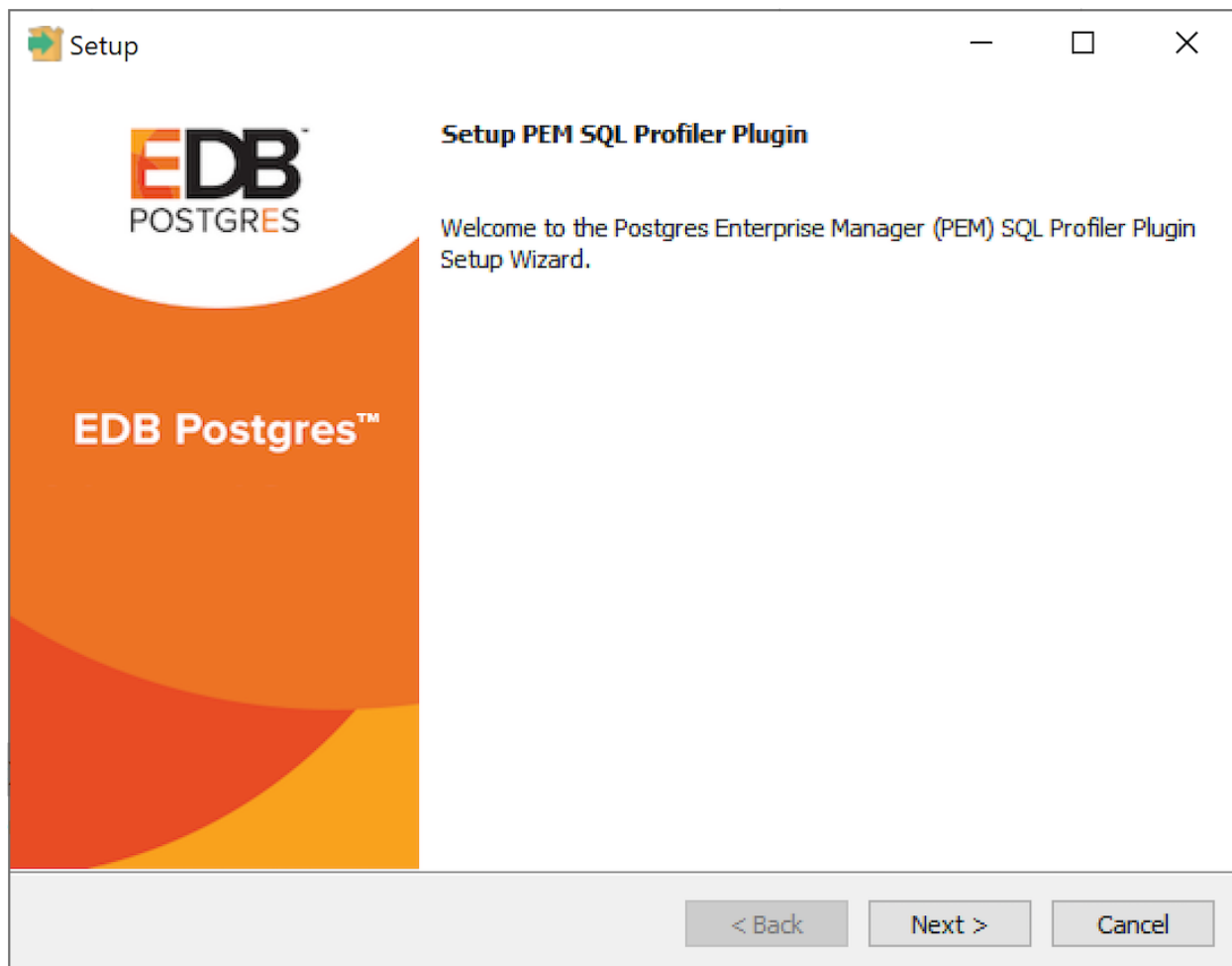


Fig. 1: The SQL Profiler Installer Welcome dialog

Select **Next** to continue to the license agreement.

Carefully review the license agreement before selecting the appropriate radio button and accepting the agreement. Select **Next** to continue to the Installation Directory dialog box.

Specify an alternative location for the installation directory, or accept the default location and select **Next**.

The wizard is now ready to install the SQL Profiler plugin. Select **Next**.

The SQL Profiler plugin installer displays progress bars as it copies files to your system.

When the installation is complete, the SQL Profiler plugin is ready to configure.

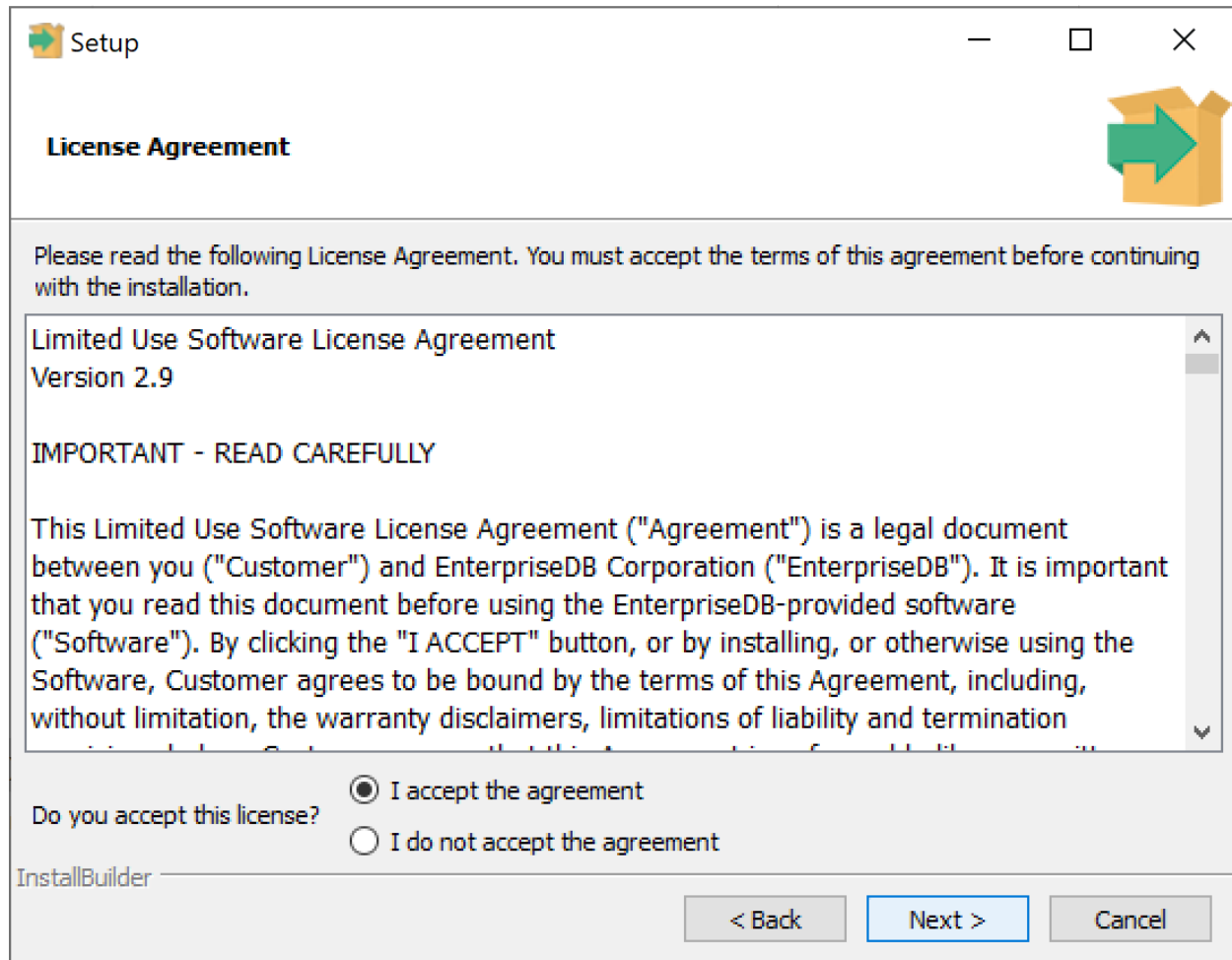


Fig. 2: The SQL Profiler License Agreement

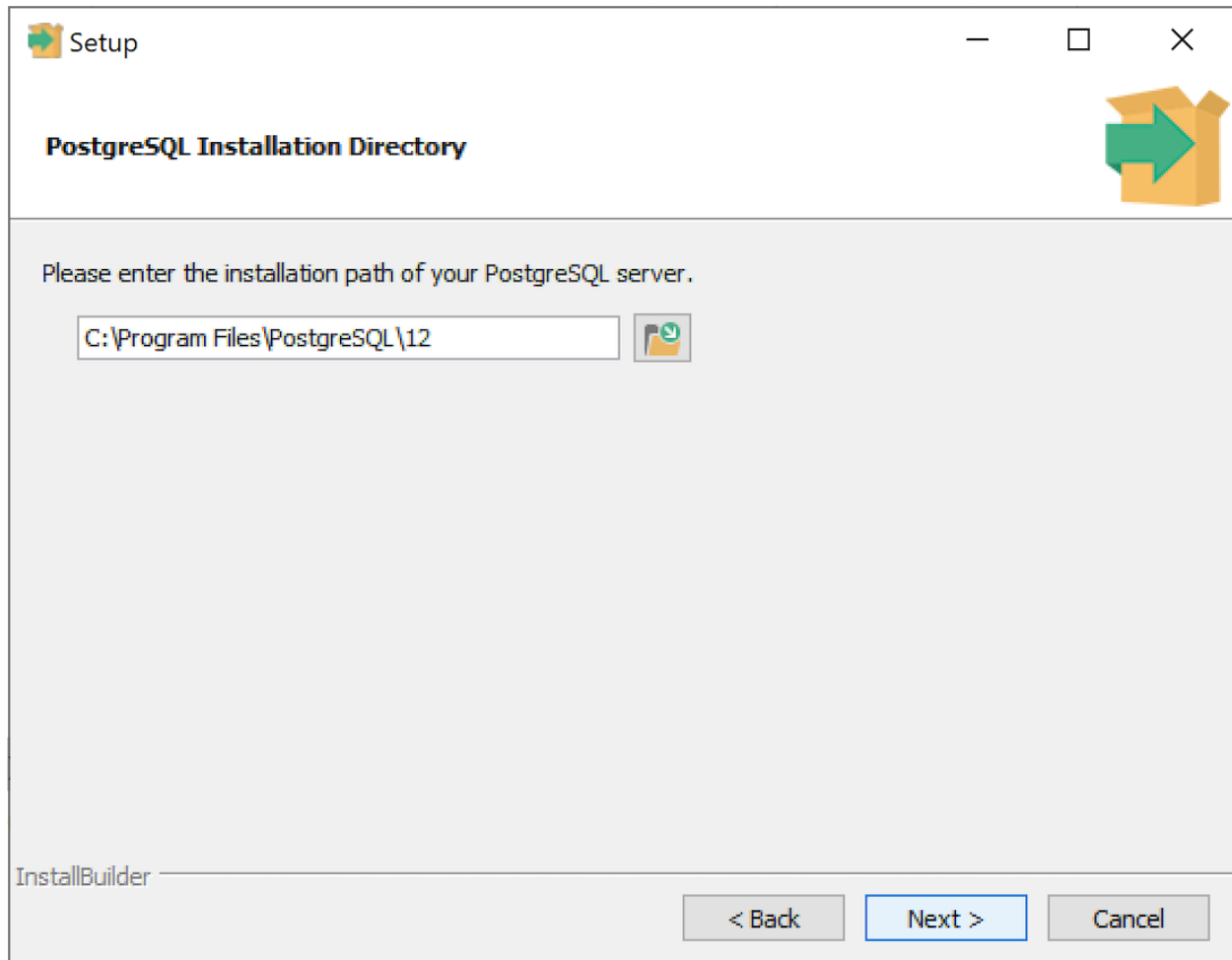


Fig. 3: The PostgreSQL Installation Directory

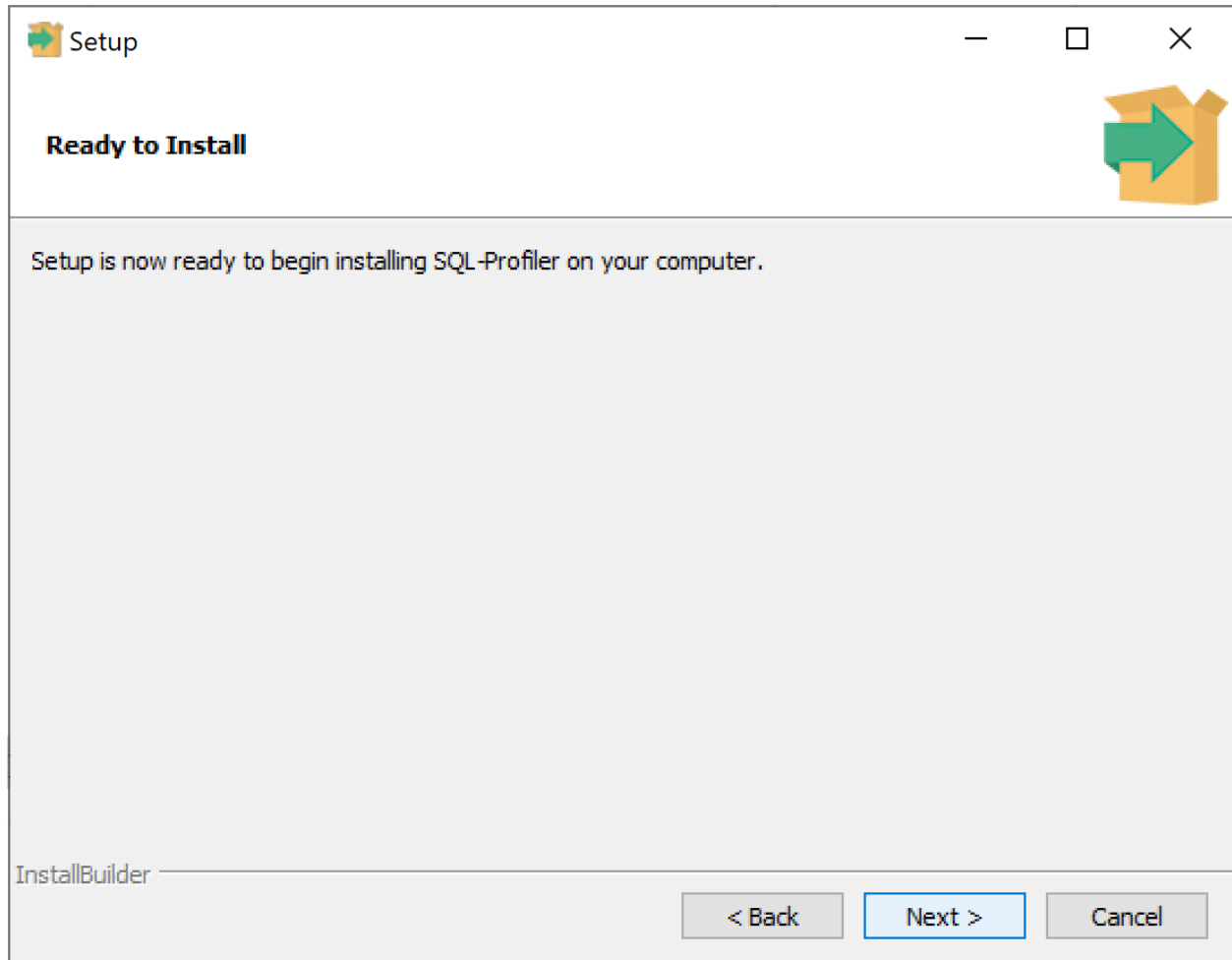


Fig. 4: Ready to Install

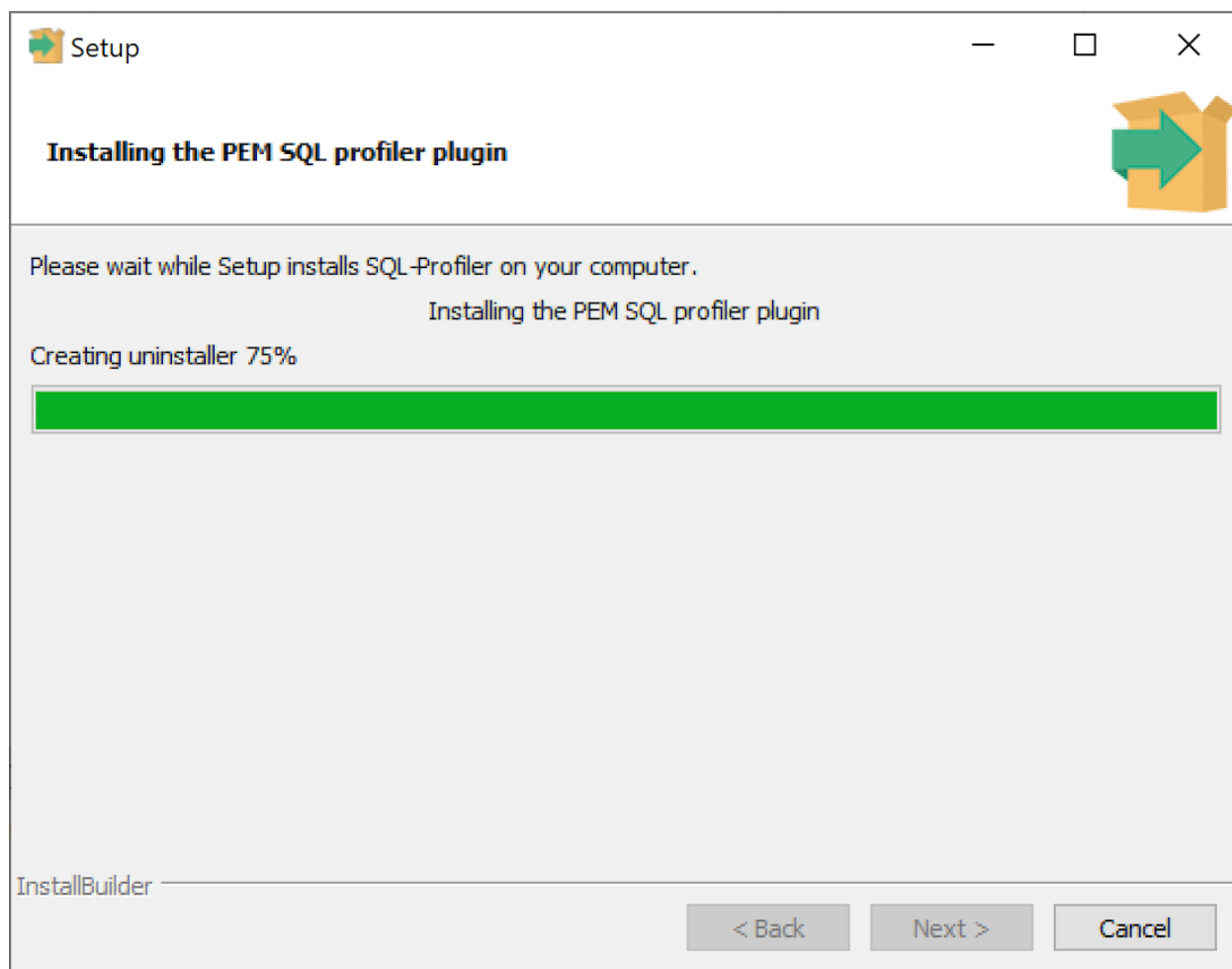


Fig. 5: Installing the SQL Profiler plugin

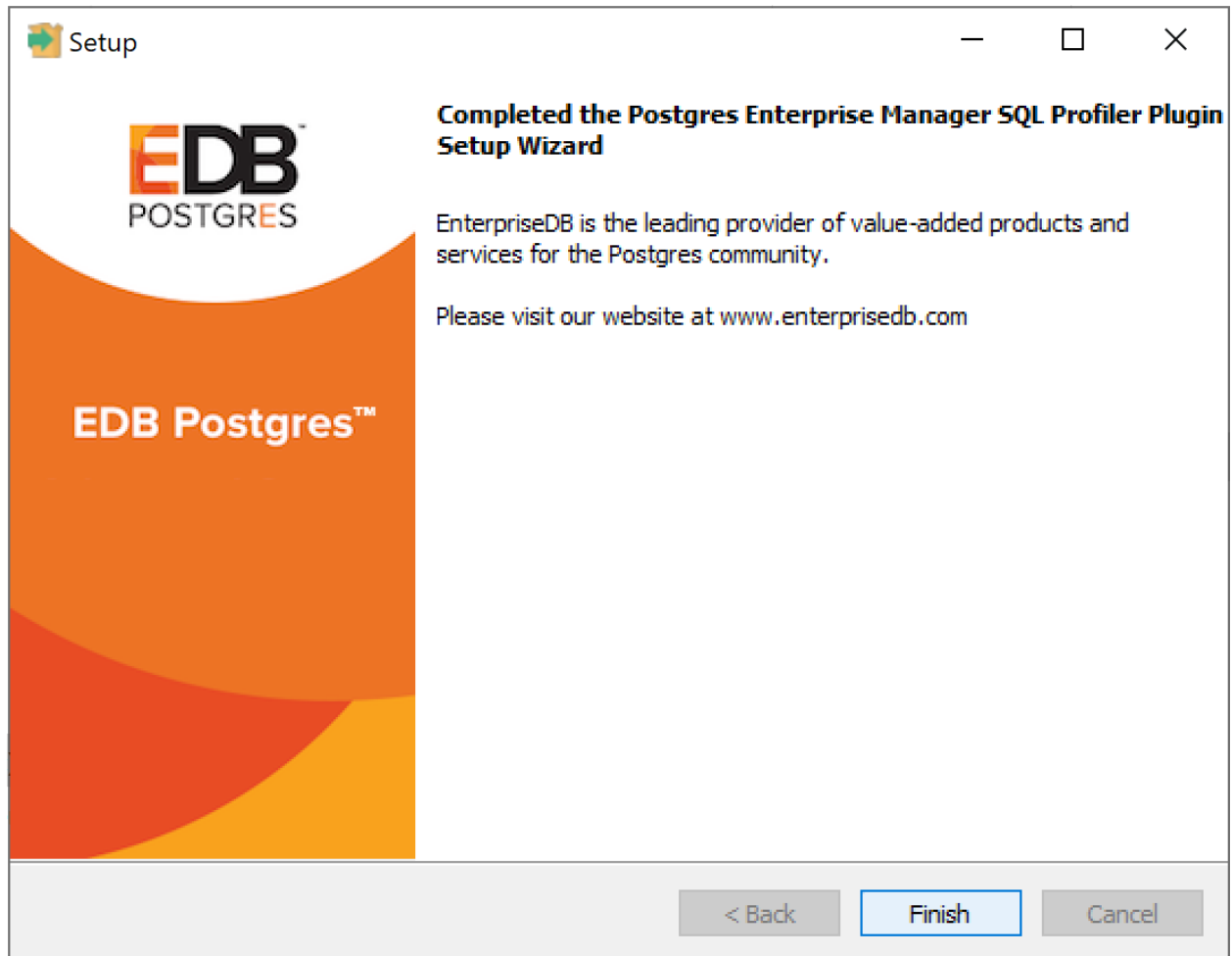


Fig. 6: The SQL Profiler installation is complete

Using an RPM Package to install SQL Profiler

Note: You might need to add the *sslutils* package to your PostgreSQL database servers before installing SQL Profiler.

If you already configured the EDB repository on your system, you can use `yum` or `dnf` to install SQL Profiler:

```
yum install postgresql<X>-sqlprofiler  
or  
dnf install postgresql<X>-sqlprofiler
```

Where `<X>` is the version of your Postgres installation.

For detailed information about configuring the EDB repository, see
Installing on Linux .

Installing SQL Profiler on Debian/Ubuntu

Note: You might need to add the *sslutils* package to your PostgreSQL database servers before installing SQL Profiler.

You can use an `apt` command to install SQL Profiler using DEB on Ubuntu 18; assume root privileges and enter:

```
apt install postgresql-<X>-sqlprofiler
```

Where `<X>` is the version of your Postgres installation.

When the installation is complete, the SQL Profiler plugin is ready to configure.

Configuring SQL Profiler

The SQL Profiler plugin isn't enabled when the installation process completes. This way you can restart the server at a convenient time. It also prevents the plugin from being loaded on systems where you don't need it all the time.

To enable the plugin:

1. Edit the `postgresql.conf` file on the server you want to profile, modifying the `shared_preload_libraries` parameter:

```
shared_preload_libraries = $libdir/sql-profiler
```

2. Restart the Postgres server.
3. Using the Query tool or the `psql` command line interface, run the `sql-profiler.sql` script in the maintenance database on the server you want to profile. If you are using:
 - PostgreSQL, the default maintenance database is `postgres` . - EDB Postgres Advanced Server, the default maintenance database is `edb` .

To use the PEM Query tool to run the script: 1. Select the maintenance database in the Browser tree and select **Tools>Querytool** . 1. When the Query tool opens, select **File>Open** and select the `sql-profiler.sql` script. By default, the `sql-profiler.sql` script is located in the `contrib` folder under your Postgres installation. 1. When the script opens in the SQL Editor panel of the Query tool, select the content of the script in the SQL Editor and select **Query>Execute** to invoke the script and configure SQL Profiler.

You can also use the psql command line to invoke the configuration script. The following command uses psql to invoke the sql-profiler.sql script on an EDB Postgres Advanced Server database on a Linux system:

```
$ /usr/edb/as<x>/bin/psql -U postgres postgres < /usr/edb/as<x>/share/contrib/sql-
↳ profiler.sql
```

Where <x> is the version of the EDB Postgres Advanced Server.

After configuring SQL Profiler, it's ready to use with all databases that reside on the server.

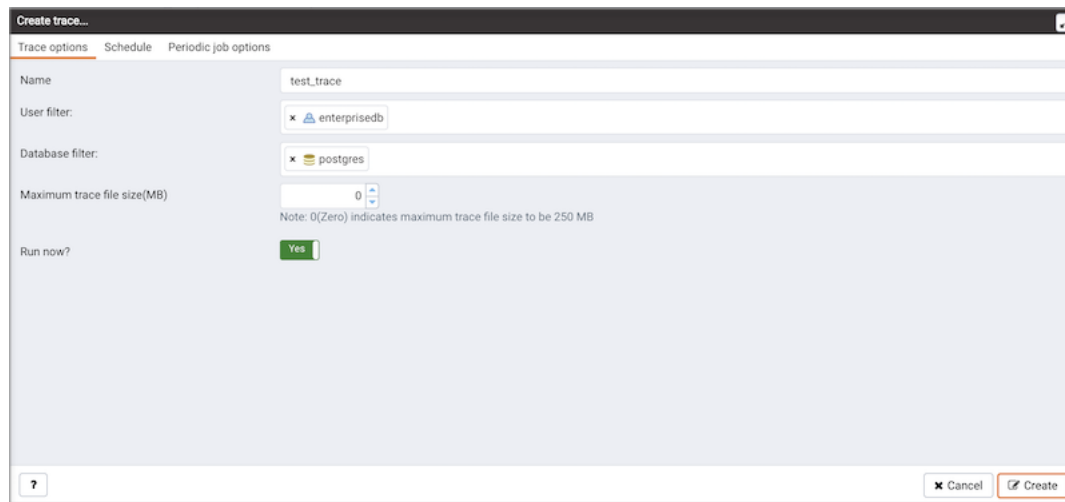


Fig. 7: Creating a new trace

To access SQL Profiler functionality, select the monitored Server/database in the PEM Browser tree. Select **Tools>Server>SQLProfiler**. Then select:

- ****Createtrace**** to define a new trace.
- ****Opentrace**** to open an existing trace.
- ****Deletetrace(s)**** to delete one or more traces.
- ****Viewscheduledtrace(s)**** to review a list of scheduled traces.

9.1.2 SQL profiling and analysis

Most RDBMS experts agree that inefficient SQL code is the leading cause of most database performance problems. The challenge for DBAs and developers is to locate that SQL code in large and complex systems and then optimize that code for better performance.

The SQL Profiler component allows a database superuser to locate and optimize inefficient SQL code. Microsoft's SQL Server Profiler is very similar to PEM's SQL Profiler in operation and capabilities. SQL Profiler is installed with each EDB Postgres Advanced Server instance. If you're using PostgreSQL, you must download the SQL Profiler installer and install the SQL Profiler product into each managed database instance you want to profile.

For each database monitored by SQL Profiler:

1. Edit the postgresql.conf file. You must include the SQL Profiler library in the shared_preload_libraries configuration parameter.

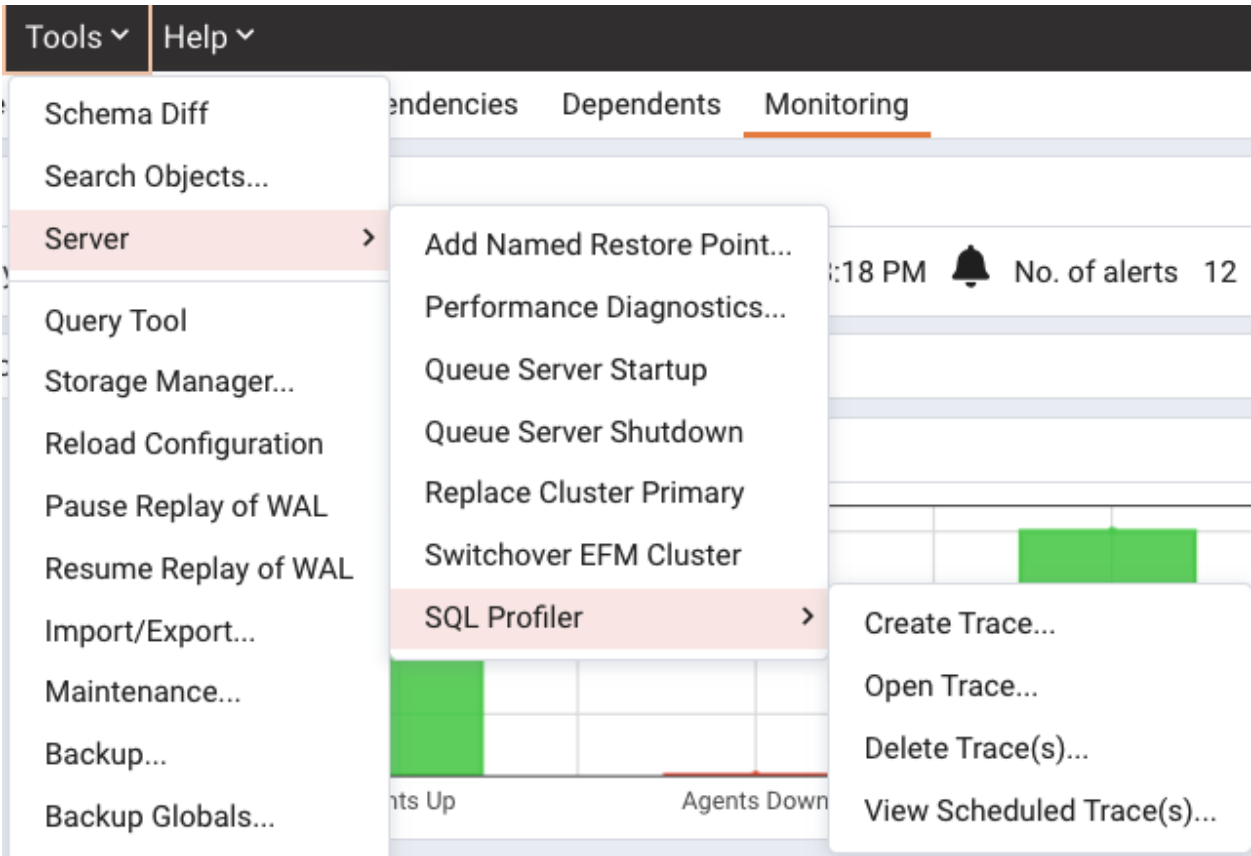


Fig. 8: Server > SQL Profiler menu

For Linux installations, the parameter value must include:

```
$libdir/sql-profiler
```

On Windows, the parameter value must include:

```
$libdir/sql-profiler.dll
```

2. Create the functions used by SQL Profiler in your database. The SQL Profiler installation program places a SQL script (named `sql-profiler.sql`) in the `share/postgresql/contrib` subdirectory of the main PostgreSQL installation directory on Linux systems. On Windows systems, this script is located in the `share` subdirectory. You must invoke this script on the maintenance database specified when registering the server with PEM.
3. Stop and restart the server to make the changes to take effect.

Note: If you connected to the PEM server with the PEM client before configuring SQL Profiler, you must disconnect and reconnect with the server to enable SQL Profiler functionality. For detailed information about installing and configuring the SQL Profiler plugin, see [Installing SQL Profiler on Windows](#).

Creating a new SQL trace

SQL Profiler captures and displays a specific SQL workload for analysis in a SQL trace. You can start and review captured SQL traces immediately or save captured traces for review later. You can use SQL Profiler to create and store up to 15 named traces.

Creating a trace

You can use the Create Trace dialog box to define a SQL trace for any database on which SQL Profiler was installed and configured. To open the dialog box, select the database in the PEM client tree and select **Management>SQLProfiler>Createtrace**.

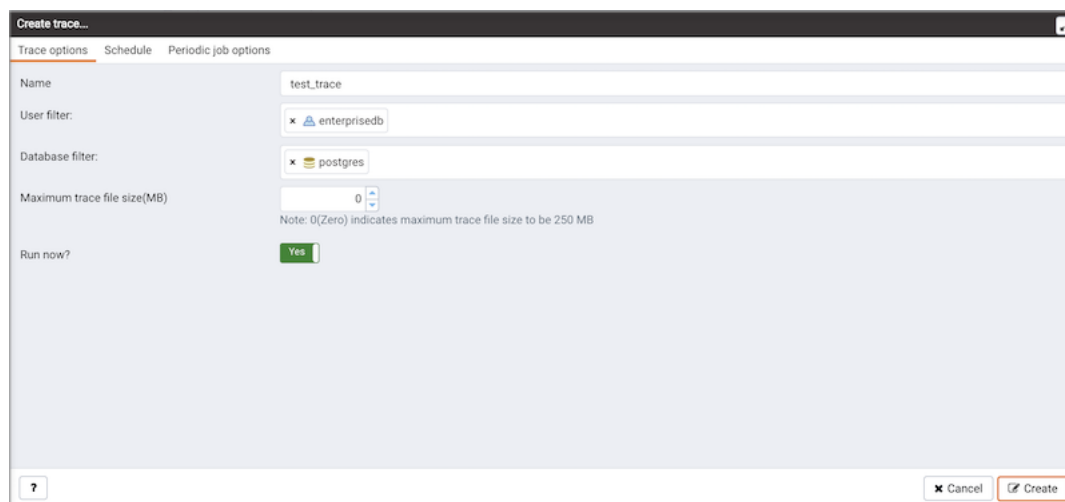


Fig. 9: The Trace options tab

Use the **Traceoptions** tab to specify details about the new trace:

- Provide a name for the trace in the **Name** field.

- Use the **Userfilter** field to specify the roles whose queries to include in the trace. Select **SelectAll** to include queries from all roles.
- Use the **Databasefilter** field to specify the databases to trace. Select **SelectAll** to include queries against all databases.
- Specify a trace size in the **MaximumTraceFileSize** field. SQL Profiler terminates the trace when it reaches approximately the size specified.
- Select **Yes** in the **RunNow** field to start the trace when you select **Create** . Select **No** to enable fields on the **Schedule** tab.

Fig. 10: The Create trace Schedule tab

Use the **Schedule** tab to specify scheduling details for the new trace:

- Use the **Starttime** field to specify the starting time for the trace.
- Use the **Endtime** field to specify the ending time for the trace.
- Select **Yes** in the **Repeat?** field to repeat the trace every day at the times specified. Select **No** to enable fields on the **Periodicjoboptions** tab.

Fig. 11: The Create trace Periodic job options tab

Use the **Periodicjoboptions** tab to specify scheduling details about a recurring trace. Use the **Days** section to specify the days when the job executes:

- Use the **Weekdays** field to select the days of the week for the trace to execute.
- Use the **Monthdays** field to select the days of the month for the trace to execute.
- Use the **Months** field to select the months when the trace executes.

Use the **Times** section to specify a schedule for the trace execution. Use the **Hours** and **Minutes** fields to specify the time when the trace executes.

After you complete the Create Trace dialog box, select **Create** to start the trace or to schedule the trace.

The screenshot displays the SQL Profiler tab with a table of query results. The table has columns for #, Start Time, Duration (ms), Query, Rows Affected, User, Database, PID, File System Read, File System Write, and Page Fault. Below the table, there are tabs for SQL Query, Metrics, Graphical Plan, and Text-Based Plan. The SQL Query tab is active, showing the SQL text of the selected query: SET DateStyle=ISO; SET client_min_messages=notice; SELECT set_config('bytea_output','escape',false);

#	Start Time	Duration (ms)	Query	Rows Affected	User	Database	PID	File System Read	File System Write	Page Fault
1	2020-04-24 01:54:57.556323-07	1.273133	SELECT heartbeat_info...	1	agent1	pern	91666	48	0	5
2	2020-04-24 01:54:57.627954-07	12.234771	/pgdash/ SELECT ...	5	enterprisedb	postgres	33485	0	0	0
3	2020-04-24 01:54:57.647487-07	0.00245	SELECT 1	1	enterprisedb	pern	30543	0	0	0
4	2020-04-24 01:54:57.659666-07	0.004035	SELECT pg_has_role(p...	1	enterprisedb	pern	30543	0	0	0
5	2020-04-24 01:54:57.660344-07	0.00083	SELECT 1	1	enterprisedb	pern	30543	0	0	0
6	2020-04-24 01:54:57.69393-07	0.917116	SET DateStyle=ISO; SE...	1	enterprisedb	postgres	33900	0	0	0
7	2020-04-24 01:54:57.696903-07	0.016561	SELECT db_oid as did, ...	1	enterprisedb	postgres	33900	0	0	0
8	2020-04-24 01:54:57.69824-07	0.027168	SELECT old as id, rolna...	1	enterprisedb	postgres	33900	0	0	0
9	2020-04-24 01:54:58.633668-07	12.019085	/pgdash/ SELECT ...	5	enterprisedb	postgres	33485	0	0	0
10	2020-04-24 01:54:58.814973-07	0.001999	SELECT 1	1	enterprisedb	pern	30543	0	0	0
11	2020-04-24 01:54:58.826576-07	0.050517	SELECT pg_has_role(p...	23	enterprisedb	pern	30543	0	0	0
12	2020-04-24 01:54:59.622713-07	11.296904	/pgdash/ SELECT ...	5	enterprisedb	postgres	33485	0	0	0
13	2020-04-24 01:55:00.137269-07	0.001639	SELECT 1	1	enterprisedb	postgres	32578	0	0	0
14	2020-04-24 01:55:01.250787-07	11.271025	/pgdash/ SELECT ...	5	enterprisedb	postgres	33485	0	0	0
15	2020-04-24 01:55:02.114481-07	0.627253	WITH agent_info AS (...)	1	agent1	pern	91666	680	0	52
16	2020-04-24 01:55:02.236089-07	12.07968	/pgdash/ SELECT ...	5	enterprisedb	postgres	33485	0	0	0
17	2020-04-24 01:55:02.28516-07	13.478983	SELECT * FROM (SEL...	3	agent1	pern	91666	1568	0	49
18	2020-04-24 01:55:03.232281-07	0.049985	SELECT * FROM pern.j...	1	agent1	pern	91666	0	0	0
19	2020-04-24 01:55:03.235696-07	0.011202	SELECT nextval(pern.j...	1	agent1	pern	91666	0	0	0
20	2020-04-24 01:55:03.239847-07	18.327522	/pgdash/ SELECT ...	5	enterprisedb	postgres	33485	0	0	0
21	2020-04-24 01:55:03.243198-07	0.128994	INSERT INTO pern.jobs...	1	agent1	pern	91666	32	0	1
22	2020-04-24 01:55:03.263871-07	0.03972	SELECT log_directory, l...	1	agent1	pern	91666	0	0	0

Fig. 12: The SQL Profiler tab, displaying the trace results

If you execute the trace immediately, the trace results appear in the PEM client.

Opening an existing trace

To view a previous trace, select the profiled database in the PEM client tree and select **Management>SQLProfiler>Opentrace**. The Open Trace dialog box opens.

Select an entry in the trace list and select **Open** to open the selected trace in the **SQLProfiler** tab.

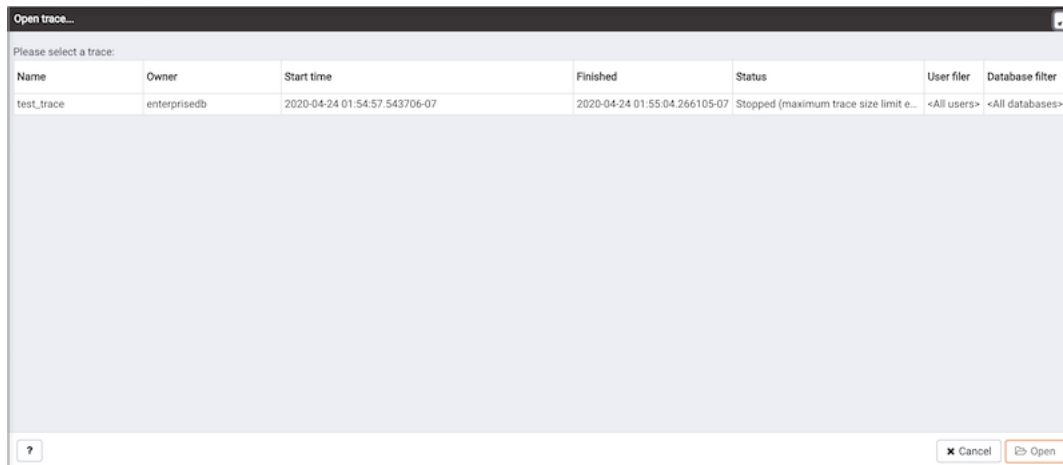


Fig. 13: Opening an existing trace

Filtering a trace

A filter is a named set of one or more rules, each of which can hide events from the trace view. When you apply a filter to a trace, the hidden events aren't removed from the trace but are excluded from the display.

Select **Filter** to open the Trace Filter dialog box and create a rule or set of rules that define a filter. Each rule screens the events in the current trace based on the identity of the role that invoked the event or the query type invoked during the event.

To open an existing filter, select **Open** . To define a new filter, select **Add(+)** to add a row to the table displayed on the **General** tab and provide rule details:

- Use the **Type** list to specify the trace field that the filter rule applies to.
- Use the **Condition** list to specify the type of operator for SQL Profiler to apply to the value when it filters the trace:
 - Select **Matches** to filter events that contain the specified value.
 - Select **Doesnotmatch** to filter events that don't contain the specified value.
 - Select **Isequalto** to filter events that contain an exact match to the string specified in the **Value** field.
 - Select **Isnotequalto** to filter events that don't contain an exact match to the string specified in the **Value** field.
 - Select **Startswith** to filter events that begin with the string specified in the **Value** field.
 - Select **Doesnotstartwith** to filter events that don't begin with the string specified in the **Value** field.
 - Select **Lessthan** to filter events that have a numeric value less than the number specified in the **Value** field.
 - Select **Greaterthan** to filter events that have a numeric value greater than the number specified in the **Value** field.
 - Select **Lessthanorequalto** to filter events that have a numeric value less than or equal to the number specified in the **Value** field.
 - Select **Greaterthanorequalto** to filter events that have a numeric value greater than or equal to the number specified in the **Value** field.
- Use the **Value** field to specify the string, number, or regular expression to search for.

After you finish defining a rule, select **Add(+)** to add another rule to the filter. To delete a rule from a filter, select the rule and select **Delete** .

Select **Save** to save the filter definition to a file without applying the filter. To apply the filter, select **OK** .

Deleting a trace

To delete a trace:

1. Select the profiled database in the PEM client tree.
2. Select **Management>SQLProfiler>Delete trace(s)** . The Delete Traces dialog box opens.

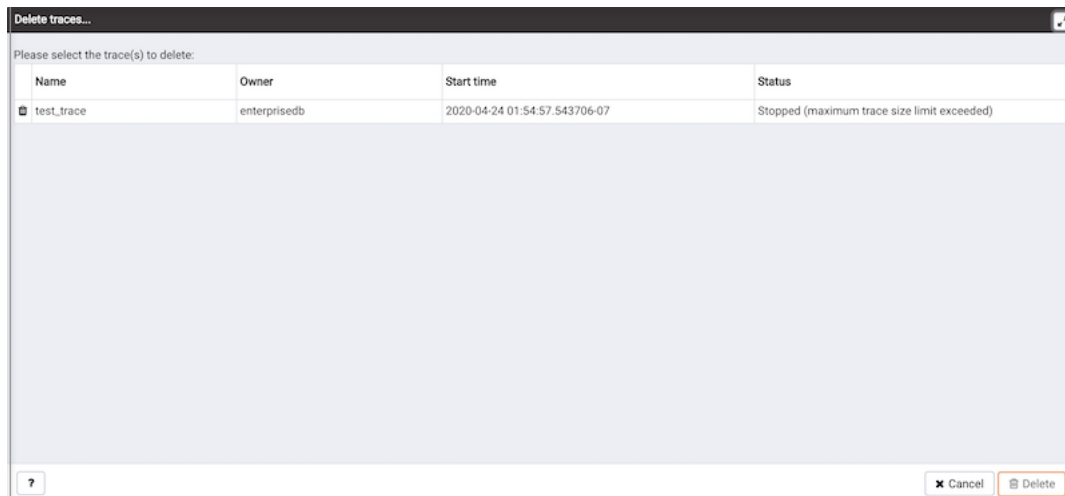


Fig. 14: The Delete traces... dialog

1. Select the icon to the left of a trace name to mark one or more traces for deletion.
2. Select **Delete** . The PEM client acknowledges that the selected traces were deleted.

Viewing scheduled traces

To view a list of scheduled traces, select the profiled database in the PEM client tree. Select **Management>SQLProfiler>Scheduledtraces** .

The Scheduled Traces dialog box displays a list of the traces that are awaiting execution. Select **Edit** to the left of a trace name to see detailed information about the trace:

- The **Status** field lists the status of the current trace.
- The **Enabled?** switch displays Yes if the trace is enabled, No if it is disabled.
- The **Name** field displays the name of the trace.
- The **Agent** field displays the name of the agent responsible for executing the trace.
- The **Lastrun** field displays the date and time of the last execution of the trace.
- The **Nextrun** field displays the date and time of the next scheduled trace.
- The **Created** field displays the date and time that the trace was defined.

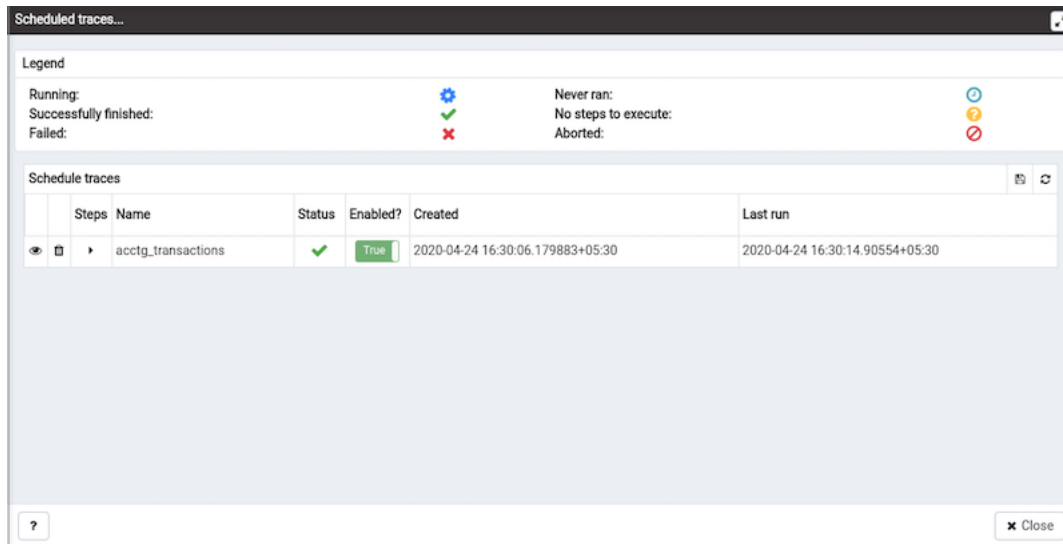


Fig. 15: Reviewing scheduled traces

9.1.3 Uninstalling SQL Profiler

The process of uninstalling SQL Profiler is platform specific.

Uninstalling SQL Profiler from Windows hosts

If you're using SQL Profiler on a Windows host, Windows locks any files that were executed or loaded into memory. To release any locked files, you must stop the Postgres server before performing an uninstall.

To stop the service:

1. In the Control Panel select **System and Security > Administrative Tools**.
2. Double-click **Services** icon.
3. In the Services dialog box, select the service name in the list, and use the option provided on the dialog box to stop the service.

After stopping the Postgres server, delete the existing SQL Profiler query set on each node by invoking the `uninstall-sql-profiler.sql` script. By default, the script resides in the `share\contrib` directory under your EDB Postgres Advanced Server or PostgreSQL installation.

Uninstalling SQL Profiler from CentOS or RHEL hosts

To uninstall a SQL Profiler installation that resides on a Linux host, delete the existing SQL Profiler query set on each node by invoking the `uninstall-sql-profiler.sql` script.

By default, if you're using EDB Postgres Advanced Server on a Linux host, the script resides in the `share/contrib` directory under the EDB Postgres Advanced Server installation. If you're using a PostgreSQL installation on a Linux host, the script resides in the `share/contrib` directory under the PostgreSQL installation.

9.2 Using the Index Advisor

Index Advisor is distributed with EDB Postgres Advanced Server version 9.0 and later. Index Advisor works with SQL Profiler by examining collected SQL statements and making indexing recommendations for any underlying tables to improve SQL response time. Index Advisor works on all DML (INSERT, UPDATE, DELETE) and SELECT statements that are invoked by a superuser.

Diagnostic output from the Index Advisor includes:

- Forecasted performance benefits from any recommended indexes
- The predicted size of any recommended indexes
- DDL statements you can use to create the recommended indexes

Before using Index Advisor, you must:

1. Modify the `postgresql.conf` file on each EDB Postgres Advanced Server host, adding the `index_advisor` library to the `shared_preload_libraries` parameter.
2. Install the `Index Advisor contrib` module. To install the module, use the `psql` client or PEM Query tool to connect to the database, and invoke the following command:

```
\i <complete_path>/share/contrib/index_advisor.sql
```

3. Restart the server to make your changes to take effect.

Index Advisor can make indexing recommendations based on trace data captured by SQL Profiler. To open Index Advisor, select one or more queries in the SQL Profiler Trace Data pane and select **IndexAdvisor** from the toolbar. For more information about configuring and using Index Advisor, see [Setting the EDB Postgres Advanced Server instance service ID](#).

Note: Index Advisor can't analyze statements invoked by a non-superuser. If you attempt to analyze statements invoked by a non-superuser, the server log includes the following error:

```
ERROR: access to library "index_advisor" is not allowed
```

Note: We recommend that you disable Index Advisor while using the `pg_dump` functionality.

VIEWING SYSTEM REPORTS

You can generate the System Configuration report and Core Usage report for all locally and remotely managed servers. To generate this report, select **Management>Reports**.

Reports has following options:

- System Configuration Report (JSON)
- System Configuration Report (HTML)
- Core Usage Report (JSON)
- Core Usage Report (HTML)

Only superusers or the users with the pem_admin role can download the System Configuration or Core Usage reports. Information in these reports shows the latest probe run time.

10.1 System Configuration Report

The System Configuration Report provides detailed information about the PEM Agents group, PEM Server Directory group, and custom groups listed under the browser tree. These groups can contain Postgres Enterprise Manager, PEM agent, and database servers. You can download this report in HTML and JSON formats.

The Postgres Enterprise Manager Summary provides details about:

- The Postgres Enterprise Manager backend database server version
- Application version
- User name accessing the application
- Python version
- Flask version
- Platform specific information

The summary provides information about the number of agents and servers.

The Group: PEM Agents panel provides details about the PEM agent, CPU cores, disk utilization, and memory information.

The Group: PEM Server Directory panel provides details about:

- Database server version
- Host
- Port

System Configuration Report	
Generated On: 2020-04-28 14:30:49	Go to: PEM Agents
Postgres Enterprise Manager Summary	
Parameter	Value
Name	Postgres Enterprise Manager
Backend version	PostgreSQL 12.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39), 64-bit
App version	7.14.0-dev (schema: 202003031)
User	postgres
Python version	3.7.5
Flask version	1.0.4
Platform	System : Darwin Node : Laptop358.pn.in Release : 18.7.0 Version : Darwin Kernel Version 18.7.0: Thu Jan 23 06:52:12 PST 2020; root:xnu-4903.278.2/RELEASE_ARM_T8020 Machine : x86_64 Processor : i386
Summary	
Parameter	Value
Agents	Windows : 0 Linux : 2
Servers	PG : 2 EPAS : 2 Unknown : 0 Locally Managed : 3 Remotely Managed : 1 Unmanaged : 0
Group: PEM Agents	
Group: PEM Server Directory	

Fig. 1: System Configuration Report - PEM Summary and Summary

- Database name
- Database size
- Tablespace size

The group server name depends on the group name to which the server is added.

10.2 Core Usage report

The Core Usage report provides detailed information about the number of cores specific to:

- Server type
- Database version
- Platform and group name

The report also gives detailed information about locally managed servers:

- Type
- Host
- Port
- Platform
- Cores
- RAM

System Configuration Report

Generated On: 2020-04-28 14:30:49 Go to: PEM Agents

> Postgres Enterprise Manager Summary

> Summary

▼ Group: PEM Agents

Agent: Postgres Enterprise Manager Host

▼ Agent Details

Parameter	Value
Platform	Linux-x64
OS	CentOS Linux release 7.5.1804 (Core)
Version	7.14.0-dev
Active	True
Hostname	localhost.localdomain
Domain Name	(none)
Bound Local Servers	- Postgres Enterprise Manager Server - EDB Postgres Advanced Server 11 - EPAS_12
Bound Remote Servers	(none)

▼ CPU

Total CPU Cores: 2
Average CPU Utilization (%): 25.31

Core ID	Load Percentage
CPU0	25.267327
CPU1	25.353135

▼ Disk Utilization

Total Disk Size (MB): 32098
Disk Space Used (MB): 13791
Disk Space Available (MB): 16669
Disk Utilization (%): 42.97

Mount Point	File System	Size (MB)	Space Used (MB)	Space Available (MB)
/	/dev/sda3	31622	13657	16352
/boot	/dev/sda1	476	134	317

▼ Memory Details

Parameter	Value
Free RAM (MB)	1050
Memory Usage Percentage	72.17
Total Swap Memory (MB)	7999
Free Swap Memory (MB)	6589
Swap Usage Percentage	17.63

Agent: localhost.localdomain

> Agent Details

> CPU

> Disk Utilization

> Memory Details

> Group: PEM Server Directory

Fig. 2: System Configuration Report - PEM Agents

System Configuration Report
Generated On: 2020-04-28 14:30:49
Go to: PEM Agents

Postgres Enterprise Manager Summary
Summary
Group: PEM Agents
Group: PEM Server Directory

Server: Postgres Enterprise Manager Server

Server Details

Parameter	Value
Agent	Postgres Enterprise Manager Host
Host	192.168.1.19
Port	5432
Database	postgres
Version	PostgreSQL 12.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39), 64-bit
Service Id	postgresql-12
Remote Monitored?	False
Active	True

Database Details

Name	Size (MB)	Tablespace Name
postgres	8	pg_default
edbstore_temp	8	pg_default
hr	8	pg_default
pem	2407	pg_default
testdb	27	pg_default
db01	8	pg_default

Tablespace Details

Name	Size (MB)
pg_global	0
pg_default	2482

Server: EPAS_12

Server Details

Database Details

Tablespace Details

Server: EDB Postgres Advanced Server 11

Server Details

Database Details

Tablespace Details

Server: PGSQL12_Centos7_1

Server Details

Database Details

Tablespace Details

Fig. 3: System Configuration Report - Group Server Name

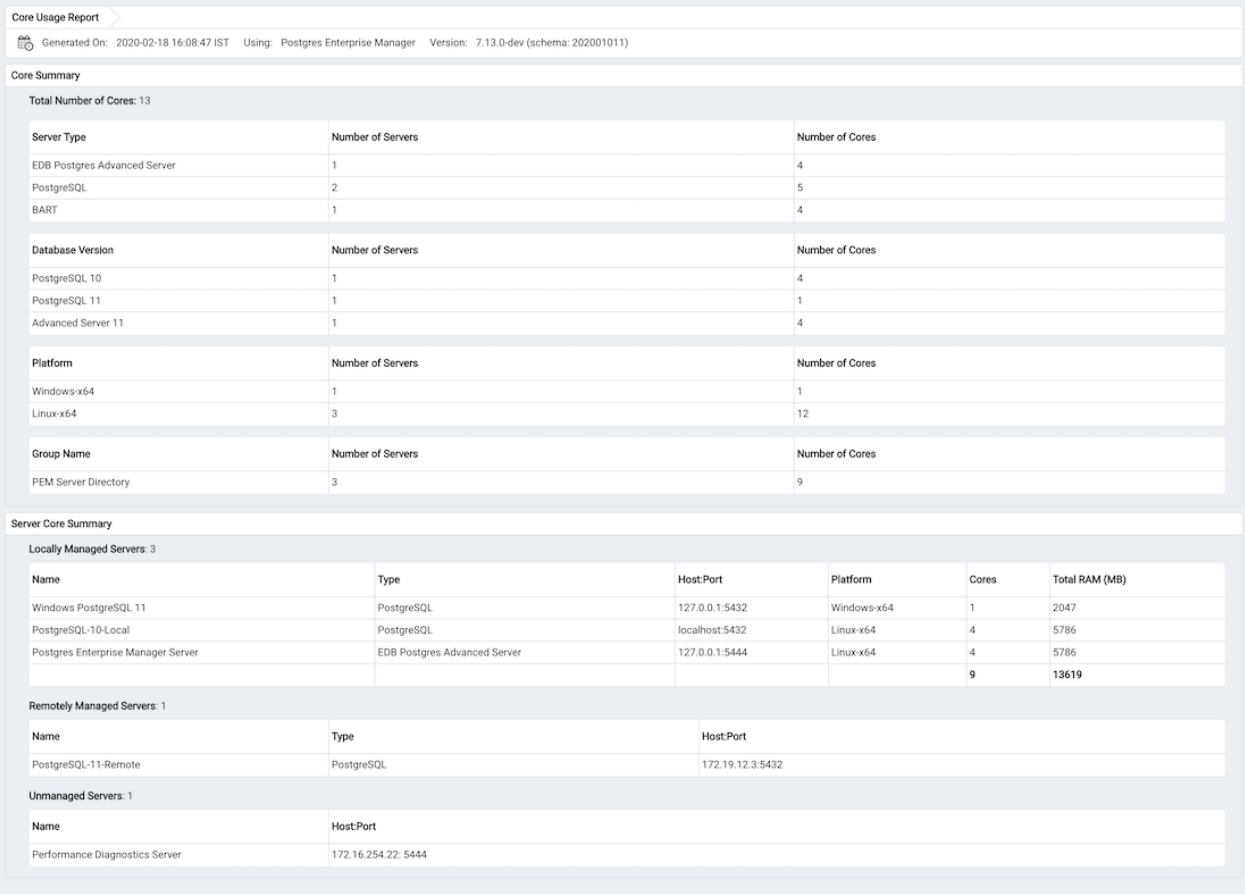


Fig. 4: Core Usage Report

PEM COMMAND LINE INTERFACE

The PEM CLI allows you to export and import the alert templates or probes from one PEM server to another.

You can install the PEM CLI using the package `edb-pem-cli-8.5.0-1-<platform_name>` separately from the `edb.repo` repository. It creates the binary file `/usr/edb/pem/cli/bin/pem` and the license file `/usr/edb/pem/edb-pem-cli-license.txt`.

After installing the PEM CLI package, use the `./pem -h` command to see all the available options.

```
$ ./pem -h
__OUTPUT__
PEM CLI (Postgres Enterprise Manager Command Line Interface)
Usage: ./pem [OPTIONS] [SUBCOMMAND]

Options:
  -h,--help                Print this help message and exit
  -v,--version              Show the app version and exit

Logging:
  --log-file [/home/asheshvashi/.pem-cli.log]
                              Set the log file

                              NOTE: Directory must exist

  -l,--log-level :value in {FATAL->1,ERROR->2,WARNING->3,DEBUG->5,INFO->4,VERBOSE->6} OR
  {1,2,3,5,4,6} [INFO]
                              Set the log level

PEM Information:
  -p,--pem-url REQUIRED (Env:PEM_SERVER_URL)
                              Set the URL for accessing Postgres Enterprise Manager
                              e.g.
                              https://127.0.0.1:8443/pem
                              https://pem_host:8443/pem

  -u,--pem-user REQUIRED (Env:PEM_API_USER)
                              User to access the PEM REST API

  -f,--password-file (Env:PEM_PASSWORD_FILE)
                              Location of the file, which contains the password.
                              The permission on a password file must disallow any access
  to world or group.
```

(continues on next page)

(continued from previous page)

NOTE:	
* The first non-empty line in the file will be treated as a password.	
* Environment variable PEM_API_PASSWORD can be used for providing the password instead of the password file	
-i,--insecure	Set the insecure connection
Subcommands:	
import-alert-templates	Import the alert template(s) into the PEM server.
list-alert-templates	List the alert templates in the PEM server.
export-alert-templates	Export the custom (user-defined) alert templates from the PEM server.
import-probes	Import the probe(s) into the PEM server.
export-probes	Export the custom (user-defined) probes from the PEM server.
list-probes	List the probes in the PEM server.

You can see all the available options with the `import-alert-templates` subcommand:

```
$ ./pem import-alert-templates -h
__OUTPUT__
Import the alert template(s) into the PEM server.
Usage: ./pem import-alert-templates [OPTIONS]

Options:
  -h,--help                Print this help message and exit
  -i,--in-file              Alert template file
  --ignore-existing         Ignore the existing alert template in the PEM server
  --ignore-existing-probes  Ignore the existing probes in the PEM server
```

You can see all the available options with the `list-alert-templates` subcommand:

```
$ ./pem list-alert-templates -h
__OUTPUT__
List the alert templates in the PEM server.
Usage: ./pem list-alert-templates [OPTIONS]

Options:
  -h,--help                Print this help message and exit
  -c,--alert-template-category :value in {ALL->2,CUSTOM->1,SYSTEM->0} OR {2,1,0} [CUSTOM]
                           List down the alert templates of this category
```

You can see all the available options with the `export-alert-templates` subcommand:

```
$ ./pem export-alert-templates -h
__OUTPUT__
Export the custom (user-defined) alert templates from the PEM server
Usage: ./pem export-alert-templates [OPTIONS]

Options:
```

(continues on next page)

(continued from previous page)

<code>-h,--help</code>	Print this help message and exit
<code>-i,--id INT ...</code>	List of id for the custom alert templates to be exported
<code>-o,--out-file [Standard output (console)]</code>	Output file name
<code>-w,--overwrite</code>	Overwrite the existing output file

You can see all the available options with the `import-probes` subcommand:

```
$ ./pem import-probes -h
__OUTPUT__
Import the probe(s) into the PEM server.
Usage: ./pem import-probes [OPTIONS]

Options:
  -h,--help                Print this help message and exit
  -i,--in-file              Probes file
  --ignore-existing         Ignore the existing custom probes in the PEM server
```

You can see all the available options with the `export-probes` subcommand:

```
$ ./pem export-probes -h
__OUTPUT__
Export the custom (user-defined) probes from the PEM server
Usage: ./pem export-probes [OPTIONS]

Options:
  -h,--help                Print this help message and exit
  -i,--probe-id INT ...    List of the custom probe-ids to be exported
  -o,--out-file [Standard output (console)]
                           Output file name
  -w,--overwrite           Overwrite the existing output file
```

You can see all the available options with the `list-probes` subcommand:

```
$ ./pem list-probes -h
__OUTPUT__
List the probes in the PEM server.
Usage: ./pem list-probes [OPTIONS]

Options:
  -h,--help                Print this help message and exit
  -c,--probe-category :value in {ALL->2,CUSTOM->1,SYSTEM->0} OR {2,1,0} [CUSTOM]
                           List down the probes of this category
```

11.1 Example

To list all the probes (system as well as custom probes) use the `-c all` option:

```
$ PEM_PASSWORD_FILE=/tmp/pass PEM_API_USER=postgres PEM_SERVER_URL=https://172.17.0.4/  
➔ pem ./pem -i list-probes -c all
```

Where:

- `PEM_PASSWORD_FILE=/tmp/pass` is the password file with the password of the postgres user.
- `PEM_API_USER=postgres` is the username through which you're running the PEM CLI.
- `PEM_SERVER_URL=https://172.17.0.4/pem` is the URL of the PEM server containing the probes.
- `-i list-probes` is the insecure option with the `list-probes` subcommand to allow insecure connection and to list the probes.
- `-c all` is the `-c` option with the `all` value to list all the system as well as custom probes.