ADMINISTRATOR GUIDE

Database Performance Analyzer

Version 2024.3

SOLARWINDS

Last Updated: Tuesday, July 30, 2024

Table of Contents

DPA introduction

See the following topics to get an overview of DPA features and learn about DPA architecture:

- Introduction to DPA
- DPA architecture

Introduction to DPA

You can use Database Performance Analyzer (DPA) to monitor, diagnose, and resolve performance problems for many types of database instances, both self-managed and in the cloud.

DPA has agentless architecture and uses wait-based analytics for extended database monitoring. DPA uses less than one percent of resources on production systems.

ⓘ Get a walk-through of DPA functionality from the DPA Getting Started Guide.

Start monitoring database instances

Get the **licenses** you need, and then **register** the databases you want to monitor.

- DPA license requirements and license types
- Purchase and view DPA licenses
- Activate DPA licenses
- Allocate or deallocate DPA licenses
- Register a database instance for monitoring with DPA

Manage DPA user accounts

- DPA roles and privileges
- Create a DPA user account and assign privileges

Investigate performance issues

Use DPA to investigate queries with **long wait times**, **inefficient** queries and the tables they run against, and **anomalies** (unusually long wait times).

- Access DPA query, table, or index advisors
- View detailed information about a query
- Investigate inefficient queries running against a table

- Investigate violations of table tuning best practices
- Identify blocking sessions and deadlocks with DPA
- Find and investigate unusually long wait times (anomalies)
- About anomaly detection in DPA
- Add an annotation to document a change to the database

Search for a SQL statement

Do you want to see DPA's query performance analysis of a SQL statement that does not show up in the Trends charts? You can **search** for it based on what you know, such as when it ran, the application, or even part of the SQL text.

- Search for a SQL statement in DPA
- Find SQL search rules
- Enable or disable the DPA Find SQL feature

Configure alerts and reports

Use **alerts** to become aware of issues and address them proactively before they affect end users.

- Configure a DPA Wait Time alert
- Configure a DPA Administrative alert
- Configure a DPA Resources alert
- Configure a DPA Custom alert
- Send SNMP traps from DPA alerts
- Create or edit a custom email template for DPA alert notifications

Use **reports** to identify database trends, track the results of your performance tuning, and communicate those results to others.

- Create a DPA report
- Schedule a DPA report for delivery
- Create and manage a DPA report group

Link DPA servers together

For large or geographically separate environments, **link DPA servers** together.

- Set up a Central Server and add remote DPA servers
- Configure authentication for the DPA Central Server
- View information from remote servers on the DPA Central Server

Use the DPA REST API

Automate tasks with the DPA **REST API**.

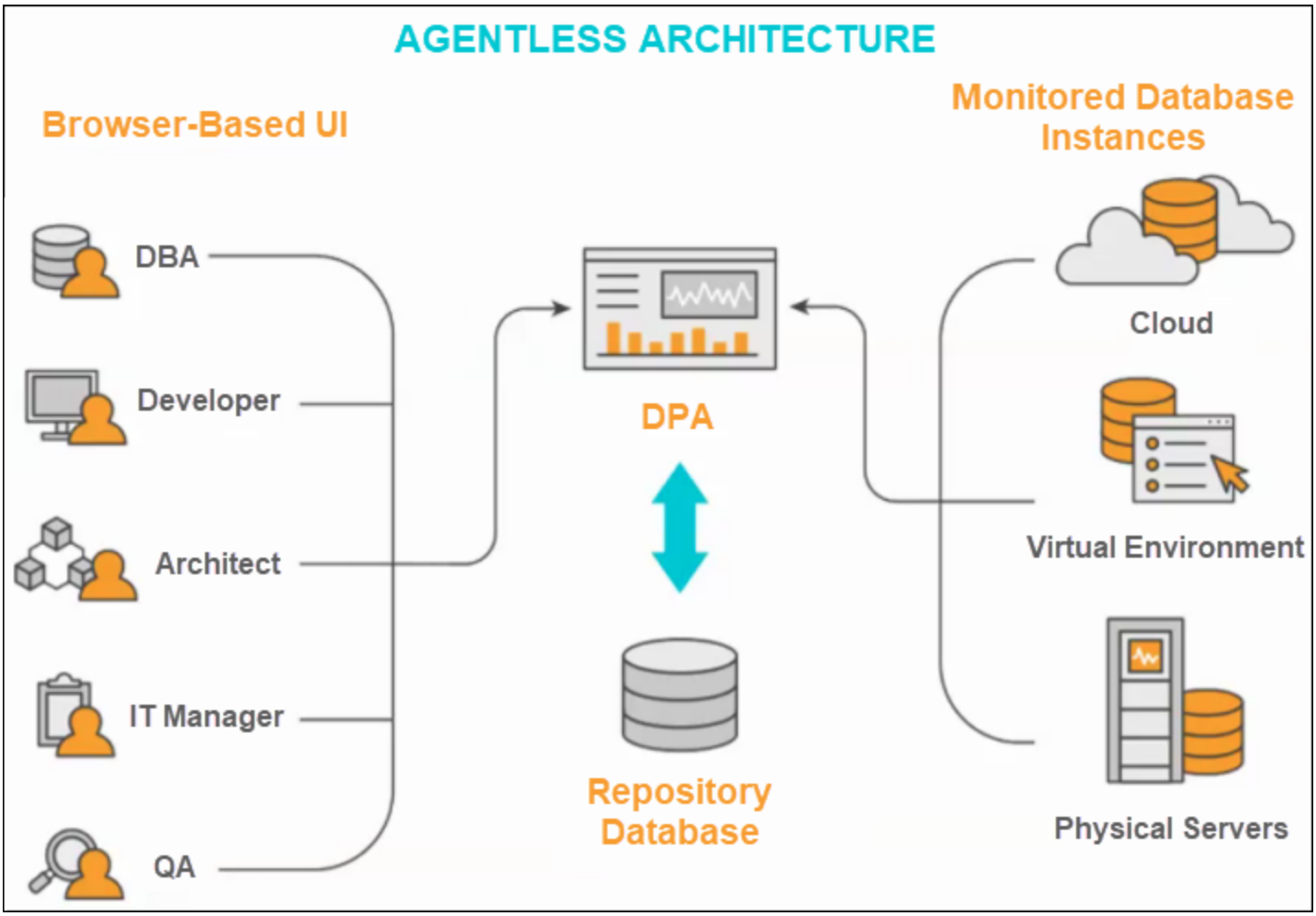
- Manage tokens used for authentication to the DPA API
- Learn about and experiment with the DPA API
- Examples of using Python scripts to make DPA API calls
- Examples of PowerShell scripts that make DPA API calls

DPA architecture

Database Performance Analyzer consists of:

- A DPA server
- A DPA repository database
- One or more database instances you want to monitor

The DPA server collects performance data from a set of database instances you choose to monitor. DPA stores this data in the repository database, and makes it available to users through its web-based interface.



For optimal performance, the DPA server, repository database, and the monitored database instances must reside on the same high-speed LAN. If your environment contains database instances that are on separate LANs, SolarWinds recommends creating a repository database on each LAN. For cloud monitoring, SolarWinds recommends deployment to the same region.

SolarWinds recommends installing one DPA instance on a computer. If you must install multiple instances on the same computer, submit a support ticket.

Key functions of the DPA server

The DPA server:

- Collects data from the monitored database instances and stores the data in the repository database.
- Provides a web-based interface that displays performance data from any computer with access to the DPA server. From this interface, you can monitor database activity, investigate performance issues, and configure alerts and reports.

Agentless monitoring for database instances and virtual environments

DPA remotely connects to each database instance using Java Database Connectivity (JDBC). DPA causes less than 1% overhead on the instance. No software is installed on the monitored server.

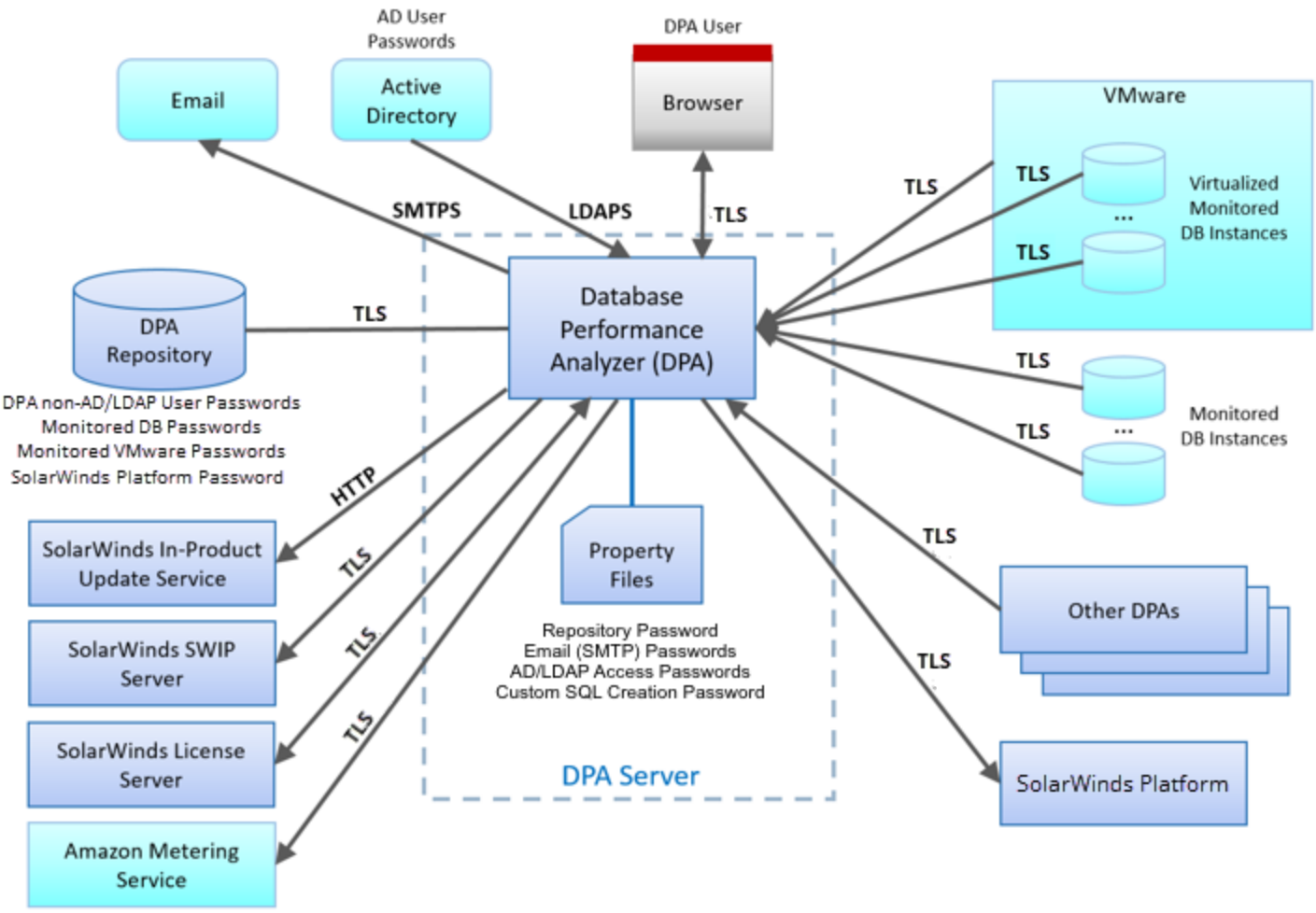
In a virtual environment, DPA can remotely connect to each VMware vCenter Server, ESX, or ESXi host. DPA causes less than 1% overhead on the monitored systems. No software is installed in the vCenter Server, ESX or ESXi host, or virtual machines.

DPA runs a query (the "quickpoll" query) on monitored database instances to collect information about wait events. By default, the quickpoll query runs once per second.

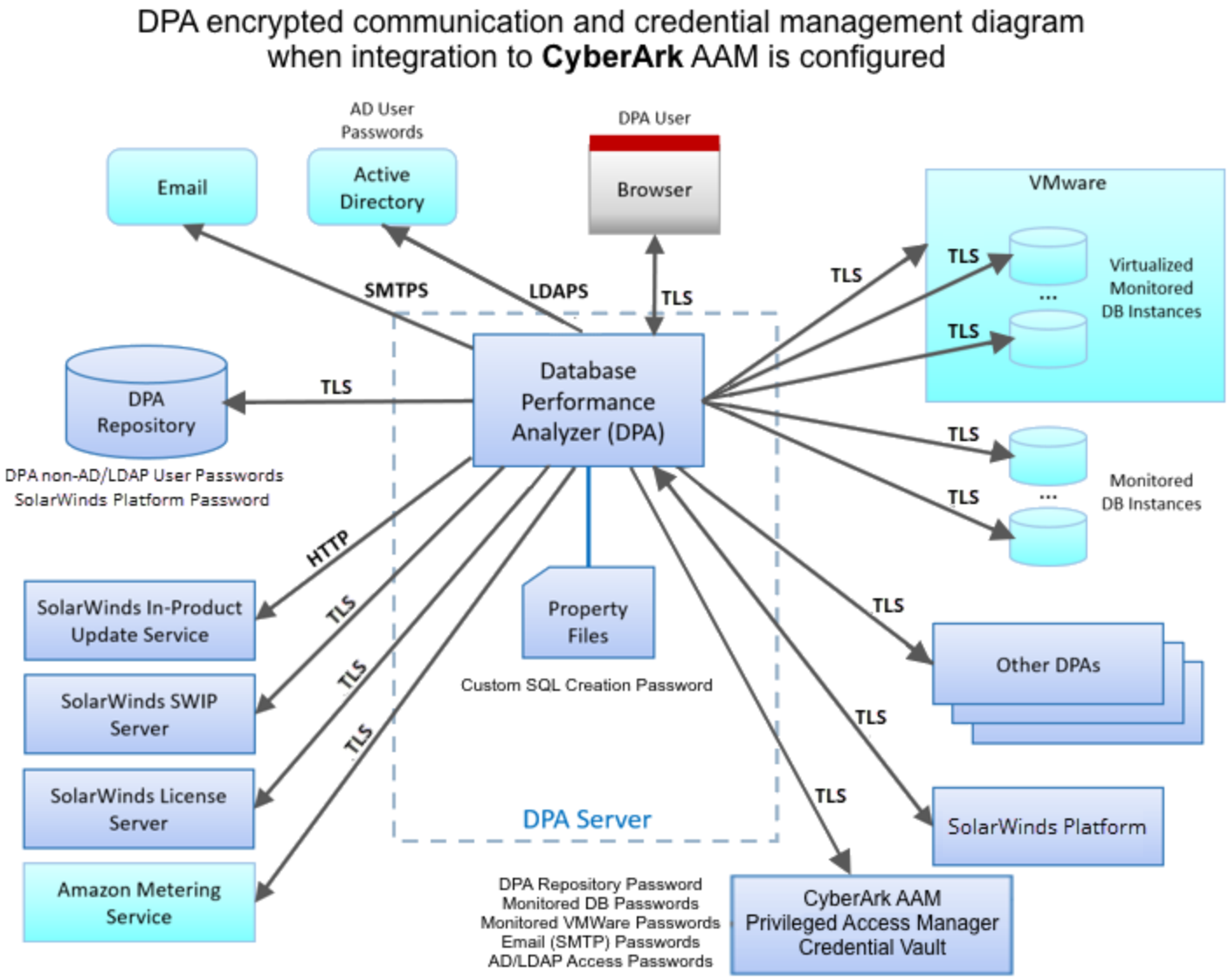
DPA encrypted communication and credential management diagrams

The following diagram provides information about DPA communication protocols and password storage when DPA is **not** configured to use credentials stored in CyberArk. The arrow direction represents connection establishment.

DPA encrypted communication and credential management diagram



The following diagram provides information about DPA communication protocols and password storage when DPA is configured to use credentials stored in CyberArk. The arrow direction represents connection establishment.



DPA licensing

See the following topics to learn more about DPA licensing:

- Learn about DPA license types and metering for DPA servers deployed in the Amazon Web Services (AWS) Marketplace.
- Learn about registration and licensing options for clustered environments.
- If you are monitoring a database instance that runs in a VM cluster, see the requirements to create a user.
- Purchase licenses and view your purchased licenses in the Customer Portal.
- Activate DPA licenses to make them available to a DPA server.
- Allocate licenses to the DPA database instances you want to monitor, or deallocate a license to make it available to another instance.
- If licenses are over-allocated, troubleshoot and resolve the issue.
- Deactivate DPA licenses to make them available to a different DPA server.

DPA license requirements and license types

License requirements depend on the type of DPA deployment:

- If you deploy DPA on a **self-managed server** or in the **Azure Marketplace**, you must purchase an individual license for each monitored database instance.
- If you deploy DPA in the **Amazon Web Services (AWS) Marketplace**, the AWS Marketplace Metering Service is used to calculate DPA charges.

Individual licenses

SolarWinds sells individual licenses by category according to the database edition they are authorized to monitor. You must purchase an individual license for each database instance you monitor. In addition, you can purchase virtual machine licenses to monitor the virtual infrastructure hosting a database instance.

> You can monitor database instances without licenses during the 14-day trial period. After the trial period, you must allocate a license for each database instance to continue monitoring.

The following table indicates the database types each DPA license type can monitor. Some database types require a Category 1 license, while others can be monitored with either a Category 1 or Category 2 license. For example, an IBM Db2 Express Edition instance can be monitored with either a Category 1 or Category 2 license. But any higher level Db2 edition requires a Category 1 license.

No additional functionality is provided by a Category 1 or Category 2 license. The only difference between the licenses in the following table is the types of databases they can monitor.

Database type	Category 1 license	Category 2 license	Azure SQL Database license
Oracle Enterprise Edition	✓		
Oracle Standard and Express Editions	✓	✓	
SQL Server, all editions	✓	✓	
MySQL, all editions	✓	✓	
Percona, all editions	✓	✓	
MariaDB, all editions	✓	✓	
PostgreSQL, all editions	✓	✓	
EDB Postgres, all editions	✓	✓	
IBM Db2, all editions	✓		
IBM Db2 Express Edition	✓	✓	
SAP ASE (Sybase), all editions	✓		
SAP ASE (Sybase) Express Edition	✓	✓	
Azure SQL Database, all editions including databases in elastic pools	✓	✓	✓

Monitoring VMs with VM Option licenses

If a database instance runs on a virtual machine (VM), you can apply an optional VM license in addition to the Category 1 or Category 2 license. When you apply a VM license, DPA collects performance metrics from the VM and the physical host on which the database instance runs. This information is displayed in the Virtualization view.

Perpetual and subscription licenses

Two types of individual licenses are available:

- Perpetual licenses can be used to monitor database instances indefinitely. Perpetual licenses do not expire.

- Subscription licenses have an expiration date. If a subscription license allocated to a database instance is not renewed before it expires, historical data is available for the instance but DPA does not continue to collect new data.

All individual licenses are floating

You can register more instances than you have licenses for. On the license allocation page, assign the licenses to the instances you want to monitor.

DPA does not collect data from registered database instances that are not licensed. However, you can view previously collected data on those database instances.

Clustered environments

For information about registering SQL Server AGs and Oracle RACs, see Registration and licensing options for clustered environments.

If you are monitoring a database instance that runs in a virtual machine (VM) cluster, a user with at least read-only permissions is required on the hosts and VMs that will be monitored.

AWS Marketplace Metering Service

When you deploy a DPA server from the AWS Marketplace, DPA uses the AWS Marketplace Metering Service to calculate charges. You can register database instances and immediately begin monitoring them **without** purchasing or allocating DPA licenses. DPA charges are based on the number of database instances you monitor each hour, and the charges are billed through Amazon. See the AWS Marketplace for details and pricing.

With the AWS Marketplace Metering Service, you can monitor any supported database type (like the Category 1 individual license). However, you cannot access the VM-related information that is available with a VM Option license.

> If you want to use individual DPA licenses in the Amazon cloud, you can deploy an EC2 instance, install DPA, and apply your licenses. You cannot use both individual DPA licenses and the AWS Marketplace Metering Service on a single DPA server.

Learn more

For more information about purchasing and allocating individual licenses, see:

- Purchase and view licenses
- Activate individual DPA licenses
- Allocate or deallocate individual DPA licenses

- Deactivate your licenses
- Troubleshoot over-allocated licenses

DPA registration and licensing options for clustered environments

To get the maximum value from DPA, SolarWinds recommends the following options for registering and licensing SQL Server Availability Groups (AGs) and Oracle Real Application Clusters (RACs).

Every environment is different, so talk with your SolarWinds representative for other possibilities.

SQL Server AGs

You can register SQL Server availability groups (AGs) using either of the following options:

- Register each SQL Server instance in the cluster
- Register the AG listener

DPA does **not** support monitoring distributed AGs (DAGs). DPA can monitor the SQL Server instances that participate in a distributed AG, but the AG monitoring features are not enabled for distributed AGs.

Register each SQL Server instance in the cluster

If there are multiple AGs in the cluster, this option is recommended because it ensures that DPA does not monitor the same instance more than once. DPA monitors all activity on each instance, including primary and secondary AG activity.

With this option, DPA does not follow AGs when they fail over. Monitoring all instances in the cluster ensures that you see all activity when AG failovers occur.

If you register each instance in the cluster, the following combination of AG connection settings is **not** supported:

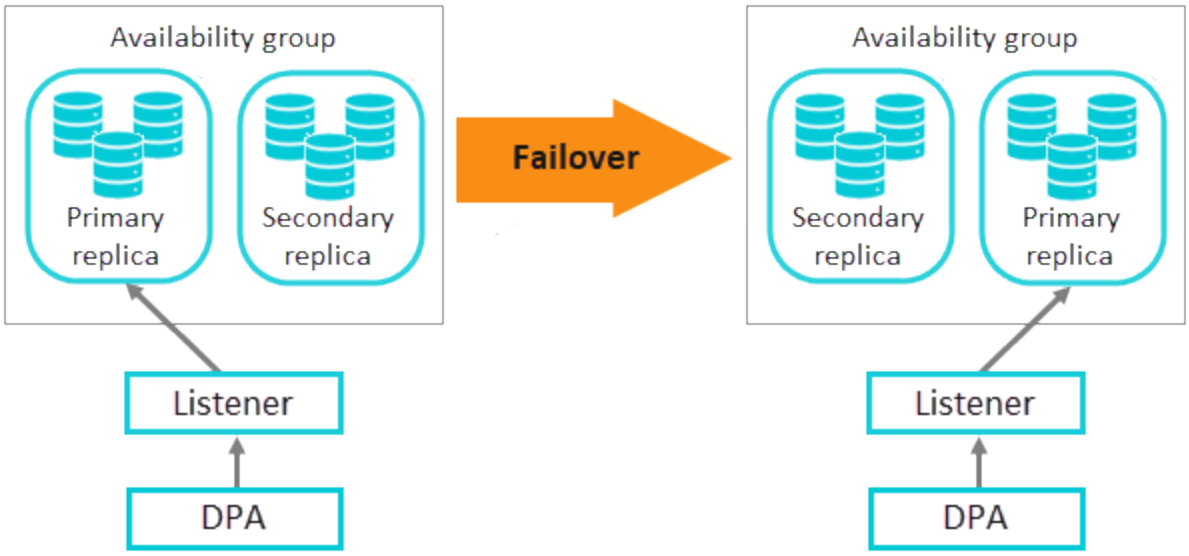
- Connections in primary role: Allow read/write connections
- Readable secondary: Read-intent only

To connect to an instance with Read-intent only selected, you must specify a connection property. The combination of settings listed above results in DPA being unable to connect to the instance part of the time:

- If the property **is** set, DPA can connect to the instance only when the instance is in **secondary** role.
- If the property is **not** set, DPA can connect to the instance only when it is in **primary** role.

Register the AG listener

Use this option if you want to monitor activity on the instance that contains the primary replica of an AG. When the AG fails over, DPA follows the listener and begins monitoring the SQL Server instance that now acts as the AG's primary replica.



SolarWinds recommends registering only one listener per cluster unless you can ensure that no instance in the cluster will act as the primary replica for multiple AGs. If you register multiple listeners and the same instance acts as the primary replica for more than one of the AGs, DPA monitors that instance multiple times. Duplicate monitoring is not recommended.

SQL Server logins are **not** automatically replicated. To enable DPA to continue monitoring after a failover, you must manually create the DPA login on all instances in the cluster that can act as the primary replica for the AG.

If Read-intent only is selected, you must register each instance instead of registering the listener.

Oracle RACs

For Oracle RAC (Real Application Clusters), register every instance in the cluster. Do not register the virtual IP that distributes load across the RAC instances.

For Oracle RAC with Data Guard, register both environments but monitor only the primary one. If a failover occurs, simply reassign the licenses to the instances in the secondary RAC environment.

When you register a RAC instance, listener configuration changes might be needed if you are not listening on the physical IP address. SolarWinds recommends:

- If you are registering pluggable databases (PDBs) on a RAC instance, register with the physical IP address of the host.

  For an Oracle multitenant container database (CDB), register each PDB contained in the CDB. You cannot register the CDB directly.

  When you register two or more Oracle PDBs in the same CDB, DPA automatically creates a group for the CDB. For more information, see About monitoring Oracle multitenent databases (CDBs).

- If you are registering a non-PDB RAC instance, register with the SID.
- If you are using the Service Name, use the physical IP address of the host. Do not use the virtual IP address (VIP) or the Oracle Single Client Access Name (SCAN) IP address.

Learn more

For more information about licensing, see the following topics:

- License types
- Purchase and view licenses
- Activate individual DPA licenses
- Allocate or deallocate individual DPA licenses

Requirements for monitoring a database instance running in a VM cluster

If you are using DPA to monitor a database instance that runs in a virtual machine (VM) cluster, a user with at least read-only permissions is required on the hosts and VMs that will be monitored. The monitored hosts and VMs include all of the following:

- The VMs that monitored database instances are running on.
- All hosts that those VMs could potentially run on (for example, all hosts in a DRS cluster).
- Other VMs on those hosts.

SolarWinds recommends giving the user read-only permissions on the entire vCenter Server or ESX/ESXi host so that DPA can dynamically monitor any entity as system changes take place.

Create a user on the vCenter Server or the ESX/ESXi host

Before you can assign user permissions, you must create the user:

- vCenter Server user: Authorized users for vCenter Server are those included in the Windows domain list referenced by vCenter Server or local Windows users on the vCenter Server system. To edit the user list or change user passwords, use the tools you use to manage your Windows domain or Active Directory.
- ESX/ESXi host user: Log in to an ESX/ESXi host as root using the vSphere Client. Then use the Users and Groups tab to add users, remove users, change passwords, set group membership, and assign the required permissions.

Assign user permissions to inventory objects

Use the vSphere Client to assign user permissions to inventory objects, such as the vCenter server, data center, host, or folder. Requirements and best practices:

- You must have modify permission on an object to be able to assign permissions to that object.
- SolarWinds recommends selecting the entire vCenter Server or ESX/ESXi host and assigning permissions to it.
- Make sure that the Propagate to Child Objects option is selected. This ensures that when new objects are inserted in to the inventory hierarchy, they inherit the required permissions.

Purchase and view DPA licenses

For DPA servers that use individual licenses, DPA has a 14-day evaluation license. During the evaluation period, you can monitor and view data for an unlimited number of database instances. After the evaluation period, to continue monitoring you must buy the correct quantity and type of licenses for your database instances.

- DPA has its own licensing and does not work with SolarWinds License Manager.
- If your DPA server is deployed in the Amazon Web Services (AWS) Marketplace, DPA uses the AWS Marketplace Metering Service to calculate charges. You do **not** need to purchase, activate, or allocate individual licenses.

Purchase licenses

Contact our sales team to purchase licenses directly from SolarWinds.

Only buy licenses for active database instances. Standby database instances used for disaster recovery or high availability do not need licenses.

- Online quote tool
- sales@solarwinds.com
- 866.530.8100

View purchased licenses

After you purchase individual licenses, you can view your DPA licenses in the SolarWinds Customer Portal.

1. Access the Customer Portal.
2. Click Licenses > Manage Licenses.

   The licenses for your DPA product are listed by license type.

Next steps

After you have bought licenses, you must activate them on a DPA server, register database instances for monitoring, and then allocate licenses to the registered instances.

Activate DPA licenses

After the DPA trial period ends, DPA monitors only licensed instances. If your DPA server uses individual licenses, you must activate a license for each database instance that you want to monitor. Make sure that you have the correct license types for the database instances you want to monitor.

- For information about licensing options for SQL Server Availability Groups and Oracle RAC environments, see Registration and licensing options for clustered environments.
- If your DPA server is deployed in the Amazon Web Services (AWS) Marketplace, DPA uses the AWS Marketplace Metering Service to calculate charges. You do **not** need to purchase, activate, or allocate individual licenses.

Activate licenses online

If the DPA server is connected to the Internet, you can activate licenses online.

To activate additional licenses with the **same** key on the **same** DPA server, see DPA licensing restriction.

1. Complete the following steps to retrieve your license activation key from the Customer Portal.

   If you are evaluating DPA and have received a license activation key from a SolarWinds representative, continue with step 2.

   a. Log in to the SolarWinds Customer Portal.

   b. Choose Licenses > Manage Licenses.

   c. Locate the license, and expand it.

   d. Copy the activation key.
2. On the DPA home page, click License Management. Then click License Manager.
3. Click Enter Activation Key.
4. Select Online Activation, and click Next.
5. On the Online Activation page, paste the activation key into the correct field.
6. In the Amount to Activate section, select All Available or Specify Amount.

   Unactivated licenses can be activated later. You can use the same activation key to activate licenses on a different DPA server. For example, if an activation key includes 10 licenses, you can enter the activation key on one DPA server and allocate 5 licenses, and

then enter it on a second DPA server and activate the remaining 5 licenses.

7. Enter the remaining information, and click Activate.

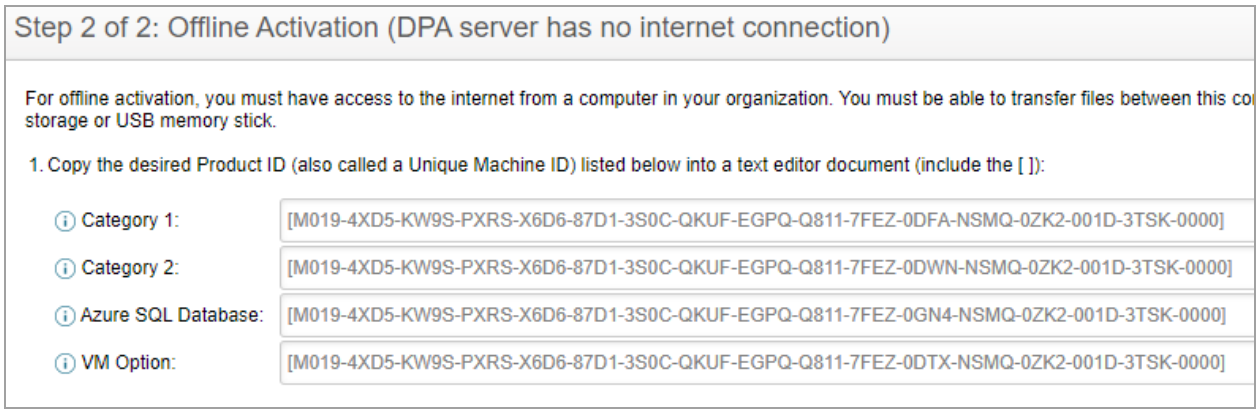
Activate licenses offline

Offline activation requires transferring files between the DPA server and a computer connected to the Internet. You can use email, shared storage, or a USB flash drive.

1. In DPA, click License Management > License Manager.
2. Click Enter Activation Key.
3. Select Offline Activation, and click Next.
4. Complete the following steps to obtain a license activation file from the Customer Portal.

   If you are evaluating DPA and have received a license activation file from a SolarWinds representative, continue with step 5.

   a. On the Offline Activation page, identify the type of license you want to activate (for example, Category 1 or Category 2).



   b. Copy the text string next to the license type you want to activate, and save it to a text file. This is the unique machine ID. Include the brackets. For example:

```
[7R12-X2QN-U8XM-WXTD-23H7-0TD7-59QH-6ERF-5BRN-2M17-328G-0DT2-MNMS-005C-000Z-04Q2-0000]
```

   Be sure to copy the unique machine ID for the correct license type. If you copy the wrong ID, you will receive the following error when you attempt to activate the license:

```
The UMID is for different product than one being activated.
```

   c. Transfer this text file to a computer with Internet access.

d. Log in to the SolarWinds Customer Portal.

e. Locate the license, and expand it.

f. Click Activate license manually.

g. Paste the text string into the Unique Machine ID field, and enter the other required information.

h. Click Generate License File to download the license file.

i. Transfer the license file to the DPA server.

5. On the Offline Activation page, click Choose File and browse to the license file.
6. Click Activate.

DPA licensing restriction

If you have used an activation key to activate licenses on a DPA server, you cannot use that activation key to activate additional licenses on the **same** server. If you try to use an activation key to activate additional licenses on the same server, DPA displays an error that includes the following text:

```
The activation key you provided has already been used. Provide an unused key and try again.
```

For example, you have an activation key that includes 10 licenses. You use it to activate 5 licenses on your DPA server. Later, you try to use that activation key to activate 2 more licenses on the same server, but DPA displays the message above.

To use an activation key to activate additional licenses on the same server:

1. Deactivate all licenses that were activated using that activation key.

   If the DPA server is online, the Customer Portal is updated with the deactivation amount.

   If the DPA server is offline, the DPA License Manager provides a deactivation receipt file that you upload to the Customer Portal. (See Deactivate offline.) When it has been uploaded, the activation key is reset.

2. Repeat the online or offline license activation task, and activate the total number of licenses that you want.

Next steps

When you activate a license, DPA automatically allocates the license to a registered database instance **if** you have enough licenses to monitor all registered instances in that license category. If you do not have enough licenses to monitor all registered instances, you must manually allocate licenses to the instances you want to monitor.

Allocate or deallocate DPA licenses

If your DPA server uses individual licenses, a license must be allocated to each registered database instance that you want to monitor. DPA starts monitoring new instances immediately after licenses are allocated.

> If your DPA server is deployed in the Amazon Web Services (AWS) Marketplace, DPA uses the AWS Marketplace Metering Service to calculate charges. You do **not** need to purchase, activate, or allocate individual licenses.

Category 1, Category 2, and Azure SQL Database licenses collect the data shown in the Performance view. VM licenses collect the data shown in the Virtualization view.

Automatic license allocation

When you register a database instance or activate a license, DPA determines if it can automatically allocate a license to each database instance.

DPA automatically allocates Category 1, Category 2, and Azure SQL licenses if there are enough activated licenses to cover all of the database instances in that license category.

Example 1:

1. You register **10** Oracle Enterprise Edition database instances, which require Category 1 licenses.
2. You activate **15** Category 1 licenses.

   Result: DPA automatically allocates **10** of the licenses to the Oracle Enterprise Edition database instances.
3. You register **5** additional Oracle Enterprise Edition database instances.

   Result: DPA automatically allocates the remaining **5** Category 1 licenses.

DPA automatically allocates VM licenses if there are enough VM licenses to cover all database instances that:

- Are linked to a VM
- Have been allocated a Category 1 or 2 license

Example 2:

- The 15 Category 1 licenses from the previous example are paired with **15** VM licenses.
- **10** of the registered database instances are linked to a VM.

  Result: DPA automatically allocates **10** VM licenses to the database instances linked to a VM.

If insufficient licenses are activated, DPA does not automatically allocate any licenses

If you have not activated enough licenses to cover all instances that require that license type, DPA does not allocate **any** of the licenses. You must manually allocate licenses to the database instances you want to monitor.

Example 3:

1. You register **15** MySQL database instances, which require Category 2 licenses.
2. You activate **10** Category 2 licenses.

   Result: DPA does not allocate **any** of the licenses. You can either:

   - Manually allocate licenses to the instances you want to monitor.

     > Instances without a license allocated to them remain registered with DPA, and you can view performance data that was collected in the past. You can deallocate a license from one registered instance and allocate it to another if necessary.

   - Make the number of activated licenses cover the number of registered instances. To do this, you can either activate 5 additional licenses, or unregister 5 database instances. When the number of activated licenses is sufficient to cover all registered instances, DPA automatically allocates the licenses.

DPA does not automatically allocate higher level licenses to instances that can be monitored with lower level licenses

Database types that require a Category 2 license (such as MySQL or SQL Server) can be monitored with a Category 1 license. Azure SQL Database instances require an Azure SQL Database license, but they can be monitored with either a Category 1 or Category 2 license. However, DPA does **not** automatically allocate higher level licenses to instances that don't require that license type.

Example 4:

1. You register **15** MySQL database instances, which require Category 2 licenses but can also be monitored with Category 1 licenses.
2. You activate **10** Category 2 licenses and **5** Category 1 licenses.

   Result: DPA does not allocate **any** of the licenses. If you want to use Category 1 licenses to monitor 5 of the MySQL instances, you must manually allocate those licenses.

3. You manually allocate the **5** Category 1 licenses to 5 of the MySQL instances.

   Result: DPA automatically allocates the **10** Category 2 licenses to the remaining 10 MySQL instances.

Manually allocate licenses to database instances

Use License Allocation to configure how your licenses are allocated to database instances.

View current license allocation

1. On the DPA home page, click License Management.
2. See the current license allocations in the summary boxes near the top of the License Allocation page.

Allocate licenses to database instances

1. On the License Allocation page, find the database instance you want in the list of registered database instances.
2. Select the Cat 1, Cat 2, or Azure checkbox next to the instance.
3. Click Save.

   The license count is updated after you allocate a license.

Allocate VM licenses to VM database instances

If a database instance runs on a virtual machine (VM), you can allocate a VM license to it in addition to a Category 1 or 2 license. When you allocate a VM license, DPA collects performance metrics from the VMware system (vCenter Server or ESX/ESXi Host) on which the database instance runs.

1. On the License Allocation page, locate a VM-hosted database instance that has a Category 1 or 2 license allocated to it.
2. Select the VM checkbox next to the instance.
3. Click Save.

ⓘ If you are monitoring a database instance that runs in a virtual machine (VM) cluster, a user with at least read-only permissions is required on the hosts and VMs that will be monitored.

Deallocate licenses

You can deallocate the license from one database instance to make it available to another database instance.

1. On the DPA home page, click License Management.
2. Clear the Cat 1, Cat 2, or Azure checkbox to deallocate licenses.

If you clear a Category 1 or 2 license from an instance that also has a VM license, DPA automatically clears the VM license as well.

Learn more

For more information about licensing, see the following topics:

- Purchase and view licenses
- Activate DPA licenses
- Troubleshoot over-allocated licenses

Troubleshoot over-allocated DPA licenses

The DPA home page displays a red banner if DPA is monitoring more registered database instances than you have licenses to monitor. This can happen in two situations:

- A license expires when you have unexpired licenses of the same type on the server.
- You deactivate a license and have other licenses of the same type on the server.

If DPA licenses are over-allocated, you cannot view or analyze your database instances until you deallocate the extra licenses. DPA continues monitoring the databases, so you will not lose data while you bring the allocated licenses to an allowable level.

If necessary, you can purchase and activate additional licenses.

To correct an issue of over-allocated licenses, deallocate database instances until you reach the proper number of licenses. If Category 2 licenses are over-allocated, assign available Category 1 licenses to cover the shortage. If Azure SQL Database licenses are over-allocated, assign Category 1 or 2 licenses to cover the shortage.

1. On the DPA home page, click License Management.
2. Locate the over-allocated license type on the allocations chart. Over-allocated license types are shown in red.
3. Clear Cat 1, Cat 2, Azure, or VM checkboxes until the chart is no longer red.
4. Click Save.

   You should now see your database instances in your views.

Deactivate your DPA licenses

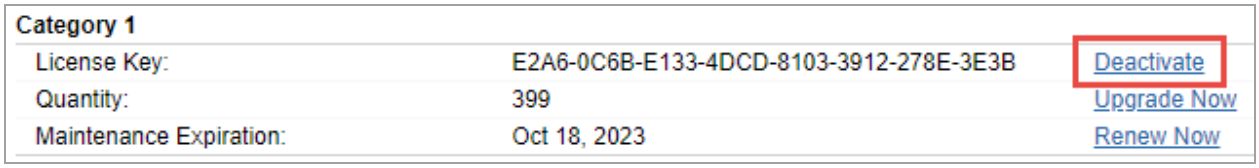
You can deactivate individual licenses on a DPA server to make the licenses available elsewhere.

If your DPA server has direct access to the internet, you can deactivate licenses online.

Evaluation licenses and temporary keys cannot be deactivated.

Deactivate online

1. On the DPA home page, click License Management.
2. On the License Allocation page, click License Manager.
3. In the Licenses section, locate the License Key you want to deactivate.
4. Click Deactivate in the right column.



Deactivate offline

To deactivate a license offline in DPA 10.0 or earlier, contact SolarWinds customer support.

To deactivate a license offline in DPA 10.1 and later:

1. From the SolarWinds DPA home page, click License Management > License Manager.
2. Click Deactivate next to the license.
3. Click Yes to confirm the offline deactivation, and download the deactivation receipt file.
4. Log in to the SolarWinds Customer Portal, and go to the License Management page.
5. Select the DPA instance, and click Deactivate license manually.
6. On the Manage License Deactivation page, browse for the deactivation receipt file, and click Upload.

Register a database instance for monitoring with DPA

Options for registering database instances include:

- Register multiple instances with the mass registration feature
- Register individual instances with a wizard
- Automate registration with the API

For more information about the database instances you can monitor, see Database instances DPA can monitor.

If the database instance runs on a virtual machine (VM), you can monitor VM performance data by registering the VMware ESX/ESXi Host or vCenter Server that the VM runs on.

Register multiple instances with the mass registration feature

If you are monitoring a large number of database instances, use the DPA mass registration feature to quickly register multiple databases.

Register individual instances with a wizard

To register a single database instance for monitoring using a wizard, select the type you want to register:

- Self-Managed:
  - Oracle
  - Microsoft SQL Server
  - SAP Sybase ASE
  - Db2
  - MySQL, Percona, or MariaDB
  - PostgreSQL or EDB Postgres

- Amazon RDS:
  - Amazon RDS for Oracle
  - Amazon RDS for SQL Server
  - Amazon RDS for MySQL or MariaDB
  - Amazon RDS for PostgreSQL
- Amazon Aurora:
  - MySQL-compatible Aurora
  - PostgreSQL-compatible Aurora
- Azure:
  - Azure SQL DB
  - Azure SQL Managed Instance
  - Azure Database for PostgreSQL
  - Azure Database for MySQL or MariaDB
- Google Cloud Platform:
  - PostgreSQL
  - SQL Server
  - MySQL

Automate registration with the API

You can register database instances using scripts that call the DPA API.

Database instances DPA can monitor

DPA can monitor database instances you manage on both physical and virtual servers or Amazon RDS instances hosted in the Amazon Elastic Compute Cloud (EC2). DPA can also monitor Azure SQL Database and Azure SQL Managed Instances. The server hosting DPA must be able to connect to the monitored instance.

ⓘ If a database version is no longer supported by the vendor, DPA ceases support for monitoring that version. Monitoring these versions may continue to function; however, SolarWinds does not provide support for any issues specific to an unsupported version.

Self-managed

For information about the privileges required for the privileged user, see the instructions for registering each database type.

Database type	Supported versions
Oracle	• 23c (single tenant and multitenant[1]) • 21 (single tenant and multitenant) • 19.x (single tenant and multitenant)
Microsoft SQL Server	• 2022 (Windows and Linux) • 2019 (Windows and Linux) DPA supports the latest SP unless otherwise noted.
SAP ASE (Sybase)	• 16
IBM Db2	• 11.5 • 11.1 • 10.5
MySQL[2]	• 8.3 • 8.0
Percona[2]	• 8.0
MariaDB[2]	• 10.6 • 10.5 • 10.4 • 10.3
PostgreSQL	• 16 • 15 • 14 • 13 • 12.0
EDB Postgres	• 16 • 15 • 14 • 13 • 12

[1] To monitor an Oracle multitenant container database (CDB), register each pluggable database (PDB) contained in the CDB. Register each PDB just as you would register an Oracle single tenant database. For more information, see Registration and licensing options for clustered environments.

[2] See Requirements for monitoring MySQL database instances with DPA.

Amazon RDS

DPA can monitor the following Amazon RDS database instances.

Database type	Supported versions
Amazon RDS for Oracle	• 19c
Amazon RDS for SQL Server	• 2019
Amazon RDS for MySQL	• 8
Amazon RDS for PostgreSQL	• 14 • 13 • 12.0
Amazon RDS for PostgreSQL EDB	• 13 • 12

Key differences for Amazon RDS for Oracle

Because of Amazon RDS access restrictions, some features that are available on Oracle self-managed database instances are not available for Amazon RDS for Oracle instances.

Category	Details
Unavailable alerts	Oracle Alert Log Error uses `V$DIAG_ALERT_EXT` instead of `X$DBGALERTEXT`.
Explain plans	Explain plans cannot be generated with a SYS account. You must specify a different account to generate the live plan.
Workarounds for not having a `SYS.UTL_CON` package	• To kill a real time session, use `RDSADMIN.RDSADMIN_UTIL.KILL`. • Trace session permissions granted through `START_TRACE_IN_SESSION` and `STOP_TRACE_IN_SESSION`.

Key differences for Amazon RDS for SQL Server

Because of Amazon RDS access restrictions, some features that are available on SQL Server self-managed database instances are not available for Amazon RDS for SQL Server instances.

Category	Details
Unavailable alerts	• SQL Server Windows Service Not Running • SQL Server Long Running Jobs • SQL Server Log Has Many Virtual Logs • SQL Server Job Failure • SQL Server Error Log Alert
Explain plans	The DPA monitoring user does not have a sysadmin role and may have limited access to objects. You can specify a different user to generate the live plan before you generate the plan.
Unavailable metrics	• CPU Queue Length • CPU Utilization • Disk Queue Length • Memory Paging Rate • Memory Utilization • Physical I/O Rate • Physical Read Rate • Physical Write Rate
Workaround for not having a SYSADMIN role	DPA user is a member of PROCESSADMIN role
Deadlock polling	The monitoring user and database administrator (DBA) do not have permission to create a custom Extended Events Session. Only the default `system_health` Extended Events Session can be used for deadlock polling.

About repointing database instances

You cannot transfer a registered Oracle or SQL Server database instance between Amazon RDS and a self-managed database and retain DPA historical data. An Oracle or SQL Server database instance transferred between Amazon RDS and a self-managed instance must be registered in DPA as a separate instance.

MySQL database instances can be repointed. After you transfer a MySQL database instance between Amazon RDS and self-managed, you can repoint DPA to the new instance and continue monitoring where you left off. To repoint, use the Update Connection Info wizard in DPA to update the connection details of the registered database instance to point to the new location.

Amazon Aurora

DPA can monitor the following Amazon Aurora database instances.

Database type	Supported versions
Amazon Aurora for MySQL-compatible	• 8.0
Amazon Aurora for PostgreSQL-compatible	• 14 • 13 • 12

Microsoft Azure

Database type	Required privileges	Supported versions
Azure SQL Database	db_owner role	V12
Azure Database for PostgreSQL - Flexible Server		• 13 • 12
Azure Database for MySQL - Single Server		• 8
Azure Database for MySQL - Flexible Server		• 5.7
Azure Database for MariaDB		• 10.3
Azure SQL Managed Instance (ASMI)	SYSADMIN role	V12

Key differences between self-managed SQL Server and Azure SQL Database

Category	Details
Unavailable Alerts	• Transaction Log Freespace • Windows Service Not Running – SQL Server • SQL Server Abnormal Mirroring Status • SQL Server Error Log Alerts • SQL Server Job Failure • SQL Server Log has Many Virtual Logs • SQL Server Long Running Jobs
Unavailable CPU Metrics	• Signal Waits • O/S CPU Utilization
Unavailable Memory Metrics	• Page Life Expectancy • O/S Memory Utilization • Plan Cache Size • Buffer Cache Size • Plan Cache Hit Ratio • Buffer Cache Hit Ratio • Log Bytes Flushed • Log Flushes • SQL Compilation • SQL Re-Compilations
Unavailable Disk Metrics	• Total I/O Wait Time • Total Read I/O Wait Time • Total Write I/O Wait Time • O/S Disk Queue Length • Page Reads • Page Writes • SQL Disk Read Latency • SQL Disk Write Latency
Unavailable Sessions Metrics	• Transaction Rate • Batch Requests
Unavailable License Compliance Metrics	• Core Count

Category	Details
Additional DTU metrics	• DTU Utilization • DTU Consumption • DTU Limit
Additional Memory metrics	• Memory Usage Utilization • XTP Storage Utilization
Additional Disk metrics	• Data I/O Utilization • Log Write Utilization • Database Storage Consumption • Database Size
Additional Sessions metrics	• Max Worker Utilization • Max Session Utilization

Key differences between self-managed SQL Server and ASMIs

Category	Details
Unavailable CPU metrics	• CPU Queue Length • Instance CPU Utilization
Unavailable Disk metrics	• Physical Read Rate • Physical Write Rate • Physical I/O Rate • O/S Disk Queue Length via WMI
Unavailable Memory metric	• Memory Paging Rate
Additional Disk metrics	• Data I/O Utilization • Log Write Utilization
Additional Memory metric	• XTP Storage Utilization
Additional Sessions metrics	• Max Worker Utilization • Max Session Utilization

About repointing database instances

Repointing database instances is not possible between an Azure SQL Database and a SQL Server database instance.

You can repoint a self-managed SQL Server instance to an ASMI. You can use this feature if you are migrating an existing self-managed SQL Server to an ASMI and you want to have DPA data collected from both the self-managed SQL Server and the ASMI associated with the same instance in DPA. However, be aware that ASMIs have different metrics and wait types than SQL Server database instances. Because of these differences, some historical data from the SQL Server database instance will not be displayed after it is repointed to an ASMI.

To retain all data, SolarWinds recommends registering the ASMI as a new instance and reassigning the license from the SQL Server instance. You will still be able to view historical data from the unlicensed SQL Server instance.

You cannot repoint an ASMI to a self-managed SQL Server instance.

Google Cloud SQL

Database type	Supported versions
Cloud SQL for PostgreSQL	• 14 • 13 • 12
Cloud SQL for SQL Server	• 2019 • 2017
Cloud SQL for MySQL	• 8

Register multiple database instances

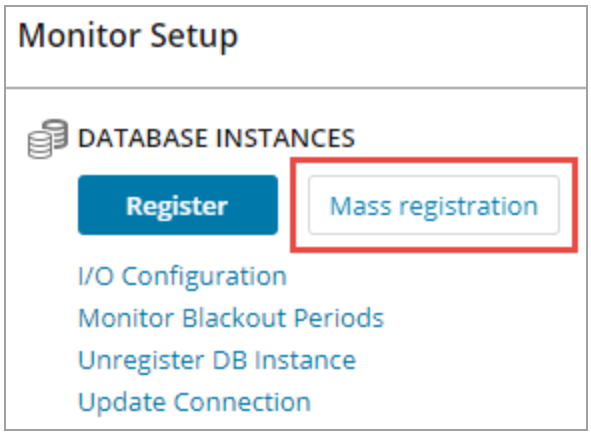
If you are monitoring a large number of database instances, use the DPA mass registration feature to quickly register multiple databases.

- You can also use a wizard to register a single database instance, or you can register database instances using scripts that call the DPA API.
- To register multiple Azure SQL databases using the Mass Registration feature, follow the instructions in this support article.
- To register SQL Server database instances using a Windows Computer Account (such as a Network Service Account), see this support article.

For DPA to collect metrics from a monitored SQL Server instance, Azure SQL instance, or ASMI, the SQL option `NUMERIC_ROUNDABOUT` must be set to `OFF`.

Complete the following steps to download a predefined template and enter the required information for all database instances.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Database Instances, click Mass registration.



3. Specify whether you want to edit and save the template on the DPA server or on your local computer.

   The instructions displayed in the right pane reflect your selection.

4. Under How to, click the link in step 1 to download the registration template used for all database types.

5. Edit the file to add information about each database instance.

   ⚠ Do **not** edit the header row.

- If you are registering multiple nodes in an Oracle RAC, manually create the monitoring user before you run mass registration. Enter `N` in the Create Monitoring User column of the template, and specify the manually created user for all nodes in the RAC.
- For an Azure SQL Managed Instance (ASMI), manually create the DPA monitoring user before you run mass registration, and enter `N` in the Create Monitoring User column.
- For PostgreSQL, manually create the DPA monitoring user before you run mass registration, and enter `N` in the Create Monitoring User column.

  You must also configure the database instances for monitoring before you run mass registration.

Column	DB instances	Description
Database Type	All	Enter one of the following values: • `SQL Server` • `Oracle` • `MySQL` • `Db2` • `Sybase` • `Azure SQL Database` • `Azure SQL Managed Instance` • `PostgreSQL`
Display Name	All	(Optional) Enter a display name to identify this database instance within DPA, or leave this field blank to use the default. You can change the display name later.
Server	All	Enter the server host name or IP address. • For SQL Server, if the SQL Server Browser service is running, enter the server name or IP address and the instance name in this format: Server\Instance. • For an Azure SQL database, enter the logical server name.

Column	DB instances	Description
Port	All	If the default port is not used, specify the port that DPA should use to connect to the database instance. The default ports are: • SQL Server, Azure SQL database, or Azure SQL Managed Instance: 1433 • Oracle: 1521 • MySQL: 3306 • PostgreSQL: 5432 • Sybase: 5000 • Db2: 50000
Service Name	Oracle	Specify the Service name of the database instance. If you complete this column, leave the SID column blank.
SID	Oracle	Specify the SID (System Identifier) of the database instance. If you complete this column, leave the Service Name column blank.
Database	Azure SQL Database, PostgreSQL, and Db2	For Azure SQL database, enter the name of the database. For PostgreSQL or Db2, enter the name of a database in the instance. The specified database is used during registration. DPA monitors **all** databases in the instance.

Column	DB instances	Description
Privileged User	All **except** Db2, PostgreSQL, and ASMI	(Optional) If you complete this field, during registration DPA uses the credentials of the privileged user to either create or configure the monitoring user. (The monitoring user enables DPA to collect information from the monitored instance.) DPA does **not** store the credentials of the privileged user. • If you want DPA to create or configure the monitoring user, enter the name of a user with the required privileges: ◦ Oracle: The `DBA` role ◦ SQL Server: The `SYSADMIN` role ◦ Sybase: The `sa_role` and the `sso_role` ◦ MySQL: The privileges listed here ◦ Azure SQL Database: The `db_owner` role • If you do **not** want to provide the credentials of a privileged user, or if DPA is integrated with CyberArk, leave this field blank and create the monitoring user yourself. See the instructions in the wizard topic for each database type.
Privileged User Password	All **except** Db2, PostgreSQL, and ASMI	(Optional) If you specified a privileged user, enter the password of the privileged user.
Monitoring User	All, if **not** integrated with CyberArk	Enter the name of the monitoring user that you created, or the name of the user that DPA will create during registration. If CyberArk is enabled, leave this column empty. For Azure SQL Database and ASMI database instances specifying a service principal instead of a user account, enter the service principal application ID.

Column	DB instances	Description
Monitoring User Password	All, if **not** integrated with CyberArk	Enter the password for the monitoring user that you created, or the password for the user that DPA will create during registration. If CyberArk is enabled, leave this column empty. For Azure SQL Database and ASMI database instances specifying a service principal instead of a user account, enter the value of the service principal secret.
Create Monitoring User (Y/N)	All	If DPA will create a new user during registration, enter `Y`. If you created the monitoring user, or if DPA will configure an existing user during registration, enter `N`.
Deployment	SQL Server, Oracle, MySQL, and PostgreSQL	If the instance runs in a self-managed environment, you can leave this column blank or enter `On-prem`. (On-prem is the default value.) If the instance is deployed in the cloud, enter one of the following values: • `Amazon` – The instance runs in Amazon RDS • `Azure` – The instance runs in Microsoft Azure • `Google` – The instance runs in the Google Cloud Platform
Repository Tablespace	DPA deployments with an Oracle repository	(Optional) If your repository database is Oracle, specify which tablespace in the repository database is used to store DPA performance data for this monitored instance. By default, performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.
Domain	SQL Server with NTLM	If the instance uses NTLM authentication, specify the domain of the monitoring user.

Column	DB instances	Description
Windows Authentication (Y/N)	SQL Server	If the monitoring user is a Windows user with the syntax DOMAIN\username, enter `Y`. The default value is `N`. If this column contains `Y`, then the Create Monitoring User (Y/N) column **must** contain `N`. DPA cannot create a Windows user.
SYS Password	Oracle	DPA requires a utility package to monitor Oracle database instances, and the monitoring user must have execute permissions for that package. DPA can create the utility package, or you can run a script to create it: • If you want DPA to create the utility package, enter the `SYS` password in this column. DPA does **not** store the `SYS` password. It is used only during registration and then forgotten. • If you prefer not to provide the `SYS` password, leave this column blank, and run a script to create the monitoring user and the utility package. For instructions, see Task 1: Create the monitoring user and utility package. If this column is blank, then the Create Monitoring User (Y/N) column must contain `N`. The script creates the monitoring user and gives it and execute permissions for the utility package .
Monitoring User Tablespace	Oracle	Specify a tablespace on the monitored database instance that is available for use by the monitoring user. DPA uses less than 5 MB of space.
Monitoring User Temp Tablespace	Oracle	Specify a temporary tablespace on the monitored database instance that is available for use by the monitoring user.

Column	DB instances	Description
SSL mode	PostgreSQL, SQL Server, Oracle, and Sybase	Enter one of the following values to specify the type of secure socket layer (SSL) connections established between the instance and the DPA server: • `disable` – SSL encryption is not used. • `require` – SSL is enabled, but no server certificate checks are performed. The server is trusted by default. Not supported for Oracle. • `verify-ca` – SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA). • `verify-full` – SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).
E-Business Suite (Y/N)	Oracle	Enter `Y` if the monitored instance contains the Oracle E-Business Suite and you want DPA to collect additional information about the suite. DPA can capture Oracle E-Business data to identify the screens, modules, and users generating the database requests. This gives you increased visibility into the causes of performance problems. The default value is `N`.
Additional JDBC URL Properties	All	(Optional) If additional JDBC URL properties are needed to enable DPA to connect to the monitored instance, enter them as name/value pairs delimited by semicolons (for example, `Property1=Value;Property2=Value`).
Additional Connection Properties	All	(Optional) If additional connection properties are needed to enable DPA to connect to the monitored instance, enter them as name/value pairs delimited by semicolons (for example, `Property1=Value;Property2=Value`).

Column	DB instances	Description
Database Group	All	(Optional) If you want to assign the instance to a group, enter the group name. ⓘ If an AG is registered via the listener, it might not be added to the group due to automatic instance naming.
CyberArk query	All, if integrated with CyberArk	If DPA is integrated with CyberArk, enter the query to retrieve the credentials of the monitoring user from CyberArk.
Authentication Method	AzureSQL Database, ASMI	Enter one of the following values to specify the authentication method used to register the instance: • `PASSWORD` – (Default) A SQL Server user password • `MEP` – A Microsoft Entra (formerly Azure AD) password • `MESP` – A Microsoft Entra service principal (formerly Azure service principal) ⓘ Do **not** specify the authentication method in the Additional Connection Properties column.

6. Save the file in `.csv` format.

   ⓘ If you selected From DPA server, the file **must** be saved in the following location:

   `<DPA_home>/iwc/tomcat/ignite_config/registration`

   The file name must be `massreg.csv`.

7. If you selected From local computer, click the Choose File button and select the file you saved.
8. Click Load Registration File.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Register an Oracle database instance

The following sections provide instructions for using a wizard to register a self-managed Oracle database instance for monitoring with DPA.

Alternatively, you can use mass registration to register multiple database instances, or you can register database instances using scripts that call the DPA API.

For an **Oracle RAC** (Real Application Cluster), SolarWinds recommends registering every physical instance in the cluster. Do not register the virtual IP that distributes load across the RAC instances.

If you choose to register the virtual IP load balancing listener, or to monitor only a subset of instances in the cluster, DPA will not have complete and consistent data. This will affect DPA's tuning and resource analysis.

For more information, see DPA registration and licensing options for clustered environments.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

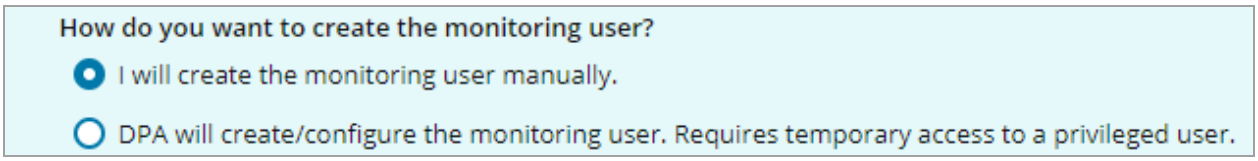
Registration wizard options

When you register an Oracle database instance for monitoring, the following changes are made to that instance:

- The DPA monitoring user is created or configured to enable DPA to monitor the database instance.
- A utility package is added, and execute permissions for that package are granted to the DPA monitoring user.

For detailed information about the utility package and the privileges granted to the monitoring user, see the Requirements Overview section of this article.

If DPA is **not** integrated with CyberArk, the following options in the registration wizard determine how the monitoring user and utility package are created or configured.



If DPA is integrated with CyberArk, these options are not displayed. You must create the monitoring user.

- **I will create the monitoring user manually.**

  With this option, you connect to the Oracle database as a user with the SYSDBA role (such as SYS) and run a script to create the monitoring user and utility package. Then you use the wizard to complete the registration.

  You must know the password of a user with the SYSDBA role to run the script, but you are not prompted to enter the SYS password or any database user credentials into DPA.

  To use this option, see Register an Oracle database instance and create the monitoring user yourself below.

- **DPA will create/configure the monitoring user.**

  With this option:

  - You must provide the credentials of a user with the DBA role so that DPA can create or configure the monitoring user.
  - If you want DPA to add the utility package, you must provide the SYS password. Otherwise, you can connect as a user with the SYSDBA role (such as SYS) and run a script to add the package.

  DPA does not store the user credentials or the SYS password. They are used only during registration and then forgotten.

  To use this option, see Register an Oracle database instance and let DPA create the monitoring user below.

Register an Oracle database instance and create the monitoring user yourself

Task 1: Create the monitoring user and utility package

1. Copy one of the following scripts to a file:

   - CreateMonUserOracleOptimized.sql

     This is the **performance-optimized** option. This script creates objects under the SYS schema. (See the Requirements Overview section of this article for more information about the objects created.)

   - CreateMonUserOracleReducedPermissions.sql

This is the **reduced-permission** option. This script does not create any objects under the `SYS` schema. With this option, DPA cannot retrieve the names of Oracle control files. If a wait event is associated with a control file, the Top Files trends chart cannot display the control file name. Instead, the chart displays the placeholder "Control File(s)". All other DPA functionality is available.

2. Edit the script to update the user name and password values.
3. Connect to the Oracle database as a user with the `SYSDBA` role (such as `SYS`), and run the script.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Self-managed, select Oracle.
3. If the monitoring user prompt is shown, select I will create the monitoring user manually. Then click Next.
4. Complete the Connection information panel:

a. Select the connection method, and then complete the fields required for the selected method:

Connection method	Required fields
Direct connect	• Enter the host name or IP address of the server that hosts the database instance. • Verify or update the port used for the connection. The default port is 1521. • Specify the SID (System Identifier) or Service name of the database instance.
TNS connect descriptor	In the TNS descriptor box, enter everything after `NAME=` in the `tnsnames.ora` file. The beginning `(DESCRIPTION=` is necessary. For example: `(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = myserver.mycompany.com)(PORT = 1521)))(CONNECT_DATA =(SERVICE_NAME = myserver)))`
TNS name	In the TNS name box, enter the `SERVICE_NAME` value from the `tnsnames.ora` file. ⓘ To use this option, Oracle Name Resolution must be configured. For instructions, see Connect to Oracle using name resolution.
LDAP	In the LDAP box, enter the LDAP distinguished name. ⓘ To use this option, Oracle Name Resolution must be configured. For instructions, see Connect to Oracle using name resolution.

b. Under SSL mode (if SSL mode is enabled), specify the type of secure socket layer (SSL) connections established between the instance and the DPA server.

ⓘ SSL connections are **not** available if LDAP is selected as the Connection method.

SSL mode	Description
Disable	SSL encryption is not used.

SSL mode	Description
Validate server certificate	SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA).
Validate server certificate and match hostname	SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).

c. Enter the user name and password of the monitoring user created previously. Or, if DPA is configured to use CyberArk, enter the CyberArk credentials query for the monitoring user.

d. Click Next.

e. DPA validates the connection information and the privileges of the monitoring user. If the validation is successful, the Instance options pane opens.

5. Specify the following Instance options.

ⓘ The instance name and group membership can be changed after registration.

a. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance.

By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

b. If the monitored instance contains the Oracle E-Business Suite, specify whether you want DPA to collect additional information about the suite.

DPA can capture Oracle E-Business data to identify the screens, modules, and users generating the database requests. This gives you increased visibility into the causes of performance problems in the Oracle E-Business Suite, Oracle Enterprise Resource Planning (ERP), and Oracle Applications environments.

c. Enter the name that DPA will display to identify this database instance.

The Display name field defaults to the name retrieved from the database instance.

d. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

ⓘ If no manual groups exist, this option is not shown.

e. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

> If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

f. Click Next.

6. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

Register an Oracle database instance and let DPA create the monitoring user

Task 1: Identify the privileged user

When you register a database instance using this option, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For self-managed Oracle database instances, the privileged user must be assigned the `DBA` role. It cannot be the repository database user.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Self-managed, select Oracle.
3. At the monitoring user prompt, select DPA will create/configure the monitoring user. Then click Next.

4. Enter the following connection information:

   a. Select the connection method, and then complete the fields required for the selected method:

Connection method	Required fields
Direct connect	• Enter the host name or IP address of the server that hosts the database instance. • Verify or update the port used for the connection. The default port is 1521. • Specify the SID (System Identifier) or Service name of the database instance.
TNS connect descriptor	In the TNS descriptor box, enter everything after `NAME=` in the `tnsnames.ora` file. The beginning `(DESCRIPTION=` is necessary. For example: `(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = myserver.mycompany.com)(PORT = 1521)))(CONNECT_DATA =(SERVICE_NAME = myserver)))`
LDAP or TNS name	In the LDAP/TNS name box, enter the LDAP distinguished name or the `SERVICE_NAME` value from the `tnsnames.ora` file. To use this option, Oracle Name Resolution must be configured. For instructions, see Connect to Oracle using name resolution.

   b. In the DBA Username and DBA Password fields, enter the name and password of the privileged user you identified previously.

   c. Click Next.

      DPA validates the connection information and the privileges of the privileged user. If the validation is successful, the Monitoring User pane opens.

5. Create or specify the account that DPA will use to gather information (the monitoring user).

   To ensure that the account has the required permissions, SolarWinds recommends creating a new account.

- To create a new account:
    a. Next to Create Monitoring User, select Yes.
    b. Enter the user name and password.
    c. Select a Tablespace and Temp Tablespace on the monitored database. This is primarily used for gathering Explain Plan data for monitored queries.
    d. Click Next.
- To specify an existing account:
    a. Next to Create Monitoring User, select No.
    b. Enter the user name and password.

       DPA uses the default Tablespaces for that user.

       DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will **not** cause load on the monitored instance.
    c. Click Next to open the Oracle Monitoring Information pane.

> If you are registering multiple Oracle Real Application Clusters (RAC) nodes, you may receive an error that the user already exists. You can create a different monitoring user or clear the Create a New Monitoring User checkbox and continue.

6. On the Oracle Monitoring Information pane, complete the following steps:
    a. Click Yes if the monitored instance contains the Oracle E-Business Suite and you want DPA to collect additional information about the suite.

       DPA can capture Oracle E-Business data to identify the screens, modules, and users generating the database requests. This gives you increased visibility into the causes of performance problems in the Oracle E-Business Suite, Oracle Enterprise Resource Planning (ERP), and Oracle Applications environments.
    b. Use one of the following options to install a utility package and grant execute permissions for that package to the DPA monitoring user:
        - Enter the `SYS` password to allow DPA to install the package and grant permissions. DPA does **not** store the `SYS` password.

> The SYS Password field is available only if remote login as `SYS` is enabled on the monitored Oracle instance.

- If you do not want to provide the `SYS` password, or the field is not available, complete the following steps:
    a. Click the link to open the Manual Steps for Monitored Database Instance Registration panel.
    b. Click Select All, copy the script, and paste it into a text file.
    c. As an Oracle Administrator, log in as `SYS` to the database instance to be monitored.
    d. Access the text file.
    e. Execute the script.

c. Click Next.

   If your repository database is Oracle, the Oracle Repository Tablespace panel opens. If not, continue with step 8.

7. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   > If your repository database is not Oracle, the wizard skips this step.

8. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

   > - If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
   > - Group membership can be changed after registration

9. Review the information and click Register Database Instance.
10. When the registration is complete, click Finish to return to the DPA home page.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register a SQL Server database instance

The following sections provide instructions for using a wizard to register a self-managed SQL Server database instance for monitoring with DPA.

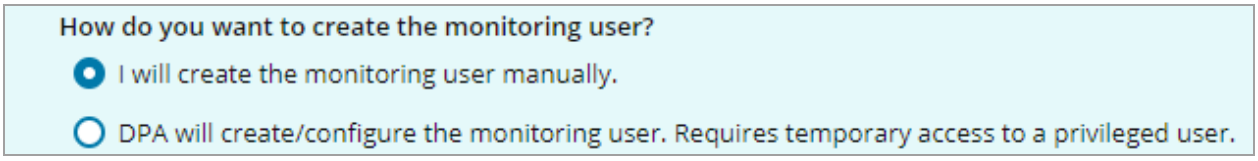
- Alternatively, you can use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.
- If you are monitoring a SQL Server Availability Group (AG), see Registration and licensing options for clustered environments.
- To register SQL Server database instances using a Windows Computer Account (such as a Network Service Account), see this support article.

For DPA to collect metrics from a monitored SQL Server instance, Azure SQL instance, or ASMI, the SQL option `NUMERIC_ROUNDABOUT` must be set to `OFF`.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Registration wizard options

The DPA monitoring user enables DPA to monitor a database instance. If DPA is **not** integrated with CyberArk, the following options in the registration wizard determine how the monitoring user and utility package are created or configured.



If DPA is integrated with CyberArk, these options are not displayed. You must create the monitoring user.

- **I will create the monitoring user manually.**

  With this option, you can create the monitoring user manually or by running a script. Then you use the wizard to complete the registration. You are **not** prompted to enter privileged user credentials into DPA.

  If you create the monitoring with one of the scripts provided, the DPA monitoring user is **not** assigned the `sysadmin` role. Because the monitoring user is not granted the `sysadmin` role, DPA has the following limitations:

- DPA cannot collect near-zero or zero cost plans.
- The 'SQL Server Log has Many Virtual Logs' alert does not work.
- The 'Windows Service Not Running - SQL Server' alert does not work.

> For detailed information about the privileges granted to the monitoring user, see this article.

To use this option, see Register a SQL Server database instance and create the monitoring user yourself below.

- **DPA will create/configure the monitoring user.**

  With this option:

  - You must provide the credentials of a user with the `sysadmin` role so that DPA can create or configure the monitoring user.
  - The monitoring user is assigned the `sysadmin` role.

  DPA does not store the user credentials. To use this option, see Register a SQL Server database instance and let DPA create the monitoring user below.

Register a SQL Server database instance and create the monitoring user yourself

Task 1: Create the monitoring user

Do one of the following:

- To avoid the limitations listed above, create the monitoring user manually and grant the `sysadmin` role to that user. For more information, see Create the DPA monitoring user for SQL Server and Azure SQL Managed Instance.
- To create the monitoring user without the `sysadmin` role, run a script:
  1. Copy one of the following scripts to a file:
     - For a **SQL Server 2014 or later** instance: CreateMonUserSqlServer2014orLater.sql
     - For a **SQL Server 2012** instance: CreateMonUserSqlServer2012.sql

     > These scripts are valid only if the default SQL Server permissions for system roles such as [Public] have not been altered with items revoked. If default system roles have been altered, DPA Support cannot help you find all items that are assumed to be allowed.

  2. Edit the script to update the user name and password values.

3. Connect to the SQL Server database instance and run the script. To ensure that the connected user has all the privileges needed to create the monitoring user, SolarWinds recommends connecting as `sysadmin` to run the script.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Self-managed, select Microsoft SQL Server.
3. If the monitoring user prompt is displayed, select I will create the monitoring user manually. Then click Next.
4. Complete the Connection information panel:
   a. Enter connection information for the SQL Server instance:
      - If the SQL Server Browser service is running, enter the server name or IP address and the instance name in this format: `Server\Instance`.
      - If the SQL Server instance contains one or more Availability Groups, click the Availability Groups link for instructions on how to register primary and secondary replicas.
      - Otherwise, enter the server name or IP address and the port number.

   ⓘ DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

   b. Under SSL mode, specify the type of secure socket layer (SSL) connections established between the instance and the DPA server.

   ⓘ If the `ForceEncryption` flag is set to Yes on the SQL Server database instance you are registering, all communication with the instance must be encrypted. Do **not** choose Disable as the SSL mode.

SSL mode	Description
Disable	SSL encryption is not used.
No certificate validation	SSL is enabled, but no server certificate checks are performed. This SSL configuration does not protect against man-in-the-middle attack because no certificate is required.
Validate server certificate	SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA).

SSL mode	Description
Validate server certificate and match hostname	SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).

c. Select the type of authentication you want to use. If Mixed Mode was selected during the SQL Server installation, you can choose either option.

d. Enter the user name and password of the monitoring user created previously. Or, if DPA is configured to use CyberArk, enter the CyberArk credentials query for the monitoring user.

e. Click Next.

DPA validates the connection information and the privileges of the monitoring user.

SSL is requested by default. If the server does not support SSL, a plain connection is used.

If you receive errors, see DPA for SQL Server installation troubleshooting.

5. Specify the following Instance options.

The instance name and group membership can be changed after registration.

a. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance.

By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

b. Enter the name that DPA will display to identify this database instance.

The Display name field defaults to the name retrieved from the database instance.

c. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

If no manual groups exist, this option is not shown.

d. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

e. Click Next.

6. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

Register a SQL Server database instance and let DPA create the monitoring user

Task 1: Identify the privileged user

When you register a database instance using this option, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For self-managed SQL Server database instances, the privileged user requires the `SYSADMIN` role. It cannot be the repository database user.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Self-managed, select SQL Server.
3. At the monitoring user prompt, select DPA will create/configure the monitoring user. Then click Next.
4. Complete the Enter Monitored Database Instance Connection Information panel:
    a. Enter connection information for the SQL Server instance:
        - If the SQL Server Browser service is running, enter the server name or IP address and the instance name in this format: `Server\Instance`.
        - If the SQL Server instance contains one or more Availability Groups, click the Availability Groups link for instructions on how to register primary and secondary replicas.
        - Otherwise, enter the server name or IP address and the port number.

        ⓘ DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

    b. Select the type of authentication you want to use. If Mixed Mode was selected during the SQL Server installation, you can choose either option.
    c. In the SYSADMIN Login and Password fields, enter credentials for the privileged user that DPA can use to register the instance.

- For Windows authentication, enter `<DOMAIN>\<username>` in the SYSADMIN Login field.
- For SQL Server authentication, enter the credentials that you enter on the Connect to Server dialog in SQL Server Management Studio (with Database Engine as the Server type).

DPA does not use or store these credentials after you complete the wizard.

d. Click Next.

DPA validates the connection information and the privileges of the privileged user.

SSL is requested by default. If the server does not support SSL, a plain connection is used.

If you receive errors, see DPA for SQL Server installation troubleshooting.

5. Create or specify the account that DPA will use to gather information (the monitoring user).

To ensure that the account has the required permissions, SolarWinds recommends creating a new account.

- To create a new account:
  a. Next to Create Monitoring User, select Yes.
  b. Select SQL Server as the authentication method. (DPA cannot create a new Windows account.)
  c. Enter a user name and password for the new account, or accept the default values.
  d. Click Next.
- To specify an existing account:
  a. Next to Create Monitoring User, select No.
  b. Select either authentication method.
  c. Enter the user name and password of an existing account.

  For Windows authentication, enter `<DOMAIN>\<username>` in the Monitoring User field.

  You can also authenticate using a Windows Computer Account.

  For SQL Server authentication, only the user name is required. Do not specify a domain.

DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will **not** cause load on the monitored instance.

d. Click Next.

If your repository database is Oracle, the Oracle Repository Tablespace panel opens. If not, continue with step 7.

6. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   If your repository database is not Oracle, the wizard skips this step.

7. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

   - If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
   - Group membership can be changed after registration

8. Review the information and click Register Database Instance.
9. When the registration is complete, click Finish to return to the DPA home page.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register a Sybase database instance

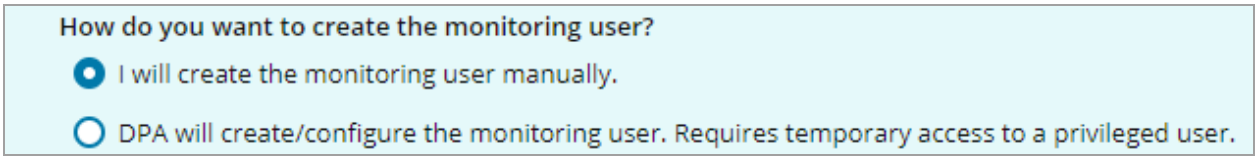
The following sections provide instructions for using a wizard to register a self-managed Sybase database instance for monitoring with DPA.

- You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.
- The Sybase Monitor Server does **not** need to be configured for DPA to monitor the instance.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Registration wizard options

The DPA monitoring user must be created to enable DPA to monitor a database instance. If DPA is **not** integrated with CyberArk, the following options in the registration wizard determine how the monitoring user and utility package are created or configured.



If DPA is integrated with CyberArk, these options are not displayed. You must create the monitoring user.

- **I will create the monitoring user manually.**

  With this option, you run a script to create the monitoring user and another script to configure the database instance. Then you use the wizard to complete the registration. You are **not** prompted to enter privileged user credentials into DPA. The DPA monitoring user is **not** assigned the `sa_role`.

  For detailed information about the privileges granted to the monitoring user and the changes made to the database instance, see this article.

  To use this option, see Register a Sybase database instance and create the monitoring user yourself below.

- **DPA will create/configure the monitoring user.**

  With this option:

  - You must provide the credentials of a user with the `sa_role` so that DPA can create or configure the monitoring user.
  - The monitoring user is assigned the `sa_role`.

  DPA does not store the user credentials. To use this option, see Register a Sybase database instance and let DPA create the monitoing user below.

Register a Sybase database instance and create the monitoring user yourself

Task 1: Create the monitoring user

1. Copy one of the following scripts to a file. These scripts create the monitoring user.
   - For a **Sybase 15.7 or later** instance: CreateMonUserSybase15-7orLater.sql
   - For a Sybase instance **earlier than 15.7**: CreateMonUserSybaseBefore15-7.sql
2. Edit the script to update the user name and password values.
3. Copy the following script to a file: ConfigureDbInstanceSybase.sql.

   This script configures the database instance. The same script is used for all versions.
4. Connect to the Sybase database instance and the first script and then the second script. To ensure that the connected user has all the privileges needed to create the monitoring user, SolarWinds recommends connecting as a user with the `sa_role` to run the script.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Self-managed, select SAP Sybase ASE.
3. If the monitoring user prompt is displayed, select I will create the monitoring user manually. Then click Next.
4. Complete the Connection information panel:
   a. Enter the host name or IP address of the server that hosts the database instance.
   b. Verify or update the port used for the connection.

      DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

   c. Under SSL mode (if SSL mode is enabled), specify the type of secure socket layer (SSL) connections established between the instance and the DPA server.

      SSL connections are **not** available if LDAP is selected as the Connection method.

SSL mode	Description
Disable	SSL encryption is not used.

SSL mode	Description
Validate server certificate	SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA).
Validate server certificate and match hostname	SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).

d. Enter the user name and password of the monitoring user created previously. Or, if DPA is configured to use CyberArk, enter the CyberArk credentials query for the monitoring user.

e. Click Next.

DPA validates the connection information and the privileges of the monitoring user.

5. Specify the following Instance options.

ⓘ The instance name and group membership can be changed after registration.

a. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance.

By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

b. Enter the name that DPA will display to identify this database instance.

The Display name field defaults to the name retrieved from the database instance.

c. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

ⓘ If no manual groups exist, this option is not shown.

d. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

ⓘ If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

e. Click Next.

6. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

Register a Sybase database instance and let DPA create the monitoring user

Task 1: Identify the privileged user

When you register a database instance using this option, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For self-managed Sybase database instances, the privileged user requires the `sa_role`. To create a new monitoring user, the `sso_role` is also required. The privileged user cannot be the repository database user.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Self-managed, select SAP Sybase ASE.
3. At the monitoring user prompt, select DPA will create/configure the monitoring user. Then click Next.
4. Complete the Enter Monitored Database Instance Connection Information panel:
   a. Enter the host name or IP address of the server that hosts the database instance.
   b. Verify or update the port used for the connection.

      DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

   c. In the Admin Username and Password fields, enter credentials for the privileged user that DPA can use to register the instance.

      DPA does not use or store these credentials after you complete the wizard.

   d. Click Next.

      DPA validates the connection information and the privileges of the privileged user.
5. Create or specify the account that DPA will use to gather information (the monitoring user).

   To ensure that the account has the required permissions, SolarWinds recommends creating a new account.

- To create a new account:
  a. Next to Create Monitoring User, select Yes.
  b. Enter a user name and password for the new account, or accept the default values.
  c. Click Next.
- To specify an existing account:
  a. Next to Create Monitoring User, select No.
  b. Select either authentication method.
  c. Enter the user name and password of an existing account.

     DPA requires the monitoring user to have `sa_role` and `mon_role` privileges for data collection.

     DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will **not** cause load on the monitored instance.
  d. Click Next.

     If your repository database is Oracle, the Oracle Repository Tablespace panel opens. If not, continue with step 7.

6. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   ⓘ If your repository database is not Oracle, the wizard skips this step.

7. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

   ⓘ
   - If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
   - Group membership can be changed after registration

8. Review the information and click Register Database Instance.
9. When the registration is complete, click Finish to return to the DPA home page.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register a Db2 database instance

Complete the following steps to register an individual Db2 database instance for monitoring with DPA.

> You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Create the DPA monitoring user

The DPA monitoring user enables DPA to monitor the database instance. The monitoring user must have the required permissions.

1. Create a user account and grant the following permissions:
   - `SYSMON` authority and connect privileges (required to monitor the Db2 instance)
   - `DBADM` authority (allows DPA to obtain access plans, and also allows DPA to explain the data access path for each specific SQL statement collected)
2. Run the following commands to grant that user `EXECUTE` privileges on the required functions:

```
grant execute on function SYSPROC.MON_GET_DATABASE to userName;
grant execute on function SYSPROC.MON_SAMPLE_WORKLOAD_METRICS to
userName;
grant execute on function SYSPROC.MON_GET_ACTIVITY to userName;
grant execute on function SYSPROC.MON_GET_BUFFERPOOL to userName;
grant execute on function SYSPROC.MON_GET_TABLESPACE to userName;
grant execute on function SYSPROC.MON_GET_TRANSACTION_LOG to userName;
grant execute on function SYSPROC.MON_GET_APPL_LOCKWAIT to userName;
grant execute on function SYSPROC.MON_GET_LOCKS to userName;
grant execute on function SYSPROC.MON_FORMAT_LOCK_NAME to userName;
```

```
grant execute on function SYSPROC.MON_GET_UNIT_OF_WORK to userName;
grant execute on function SYSPROC.MON_GET_AGENT to userName;
grant execute on function SYSPROC.MON_GET_PKG_CACHE_STMT to userName;
grant execute on function SYSPROC.MON_GET_CONNECTION to userName;
grant execute on function SYSPROC.MON_GET_MEMORY_POOL to userName;
```

3. To verify that the permissions were applied correctly, run the following command:

```
select substr(authid,1,20) as authid
    , authidtype
    , privilege
    , grantable
    , substr(objectschema,1,12) as objectschema
    , substr(objectname,1,30) as objectname
    , objecttype
from sysibmadm.privileges
where objectschema ='SYSPROC' AND AUTHID='userName';
```

Run the registration wizard

1. On the DPA home page, click Register DB Instance for Monitoring.
2. Under Self-managed, click IBM Db2. Then click Next.
3. Confirm the Db2 configuration settings.

   DPA requires the Db2 instance-wide parameter `{DFT_MON_STMT}` to be turned on to collect monitoring data. Follow the on-screen instructions to check and set the parameter.

   If `{DFT_MON_STMT}` is set to `OFF`, you can still use DPA to register the database instance. Later, you can set it to `ON` and restart the database instance during an approved maintenance window. In the meantime, the database shows a status of Idle.
4. Enter the following connection information:
   a. Enter the host name or IP address and port of the Db2 server.
   b. Specify a database that DPA can use to register the instance. DPA monitors all databases in the Db2 instance.

      For more information about monitoring a single database or all databases in the instance, see Switch to Db2 instance-wide monitoring.
   c. Enter the credentials for the DPA monitoring user account that you created earlier.

5. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   > If your repository database is not Oracle, the wizard skips this step.

6. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

   > - If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
   > - Group membership can be changed after registration

7. Review the information and click Register Database Instance.
8. When the registration is complete, click Finish to return to the DPA home page.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register a MySQL database instance

Complete the following steps to register an individual MySQL database instance for monitoring with DPA.

> You can use the registration wizard to register a **read/write** MySQL database instance. To register a **read-only** instance, see Monitor a read-only MySQL database instance in DPA.

To optimize DPA's reporting capabilities for a MySQL database instance, see the requirements for monitoring MySQL database instances.

> You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Identify the privileged user

If you choose to let DPA create or configure the account used to collect DPA data (the monitoring user), you must provide the credentials of a **privileged user**. (You can also choose to create the monitoring user yourself.) During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For self-managed MySQL database instances:

- The privileged user requires the following permission:

  ```
  CREATE USER
  ```

- The privileged user must be able to grant the following permissions:

  ```
  PROCESS on *.*
  SELECT & UPDATE on performance_schema.*
  ```

- To enable the retrieval of query execution plans, the privileged user must also be able to grant the following permissions:

  ```
  SELECT, INSERT, UPDATE, DELETE on *.*
  SYSADM
  ```

Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Self-managed, click MySQL. Then click Next.
3. Enter the following connection information:
   a. Enter the host name or IP address and port of the server.

      DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

   b. Create or specify the account that DPA will use to gather information (the monitoring user).

      > If you are registering a MySQL 8.0 instance, you must create the repository user manually, and you must include the following in the `CREATE USER` command:
      >
      > ```
      > IDENTIFIED WITH mysql_native_password BY 'yourPassword';
      > ```

      SolarWinds recommends creating a separate account for the monitoring user.

DPA ignores data generated by the monitoring user on the monitored database instance. For this reason, do not specify a user that causes load on the monitored instance.

- To let DPA create or configure the monitoring user:
  a. Select Let DPA create a new user or configure an existing user for me.
  b. Enter the credentials of an existing user with the required privileges.

     The credentials for the privileged user are not used or stored after the registration.

  c. Select a Tablespace and Temp Tablespace on the monitored database. This is primarily used for gathering Explain Plan data for monitored queries.
  d. Click Next.
- To create the monitoring user yourself:
  a. Select I'll create the database user.
  b. Click Monitoring User Creation Script.
  c. Copy the script to a file and edit it per the instructions.
  d. Copy the edited script to the MySQL console, and run it.

     The monitoring user is created.

  e. Enter this user's credentials in the Username and Password fields.

If your repository database is Oracle, the Oracle Repository Tablespace panel opens. If not, continue with step 5.

4. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   If your repository database is not Oracle, the wizard skips this step.

5. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

   - If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
   - Group membership can be changed after registration

6. Select a Typical or Custom configuration. SolarWinds recommends the Typical configuration

   - Typical is recommended. With this option:
     - The DPA Recommended option is used for Performance Schema setup.
     - `EXPLAIN` can be run on `SELECT` statements.
   - Select Custom to change the Performance Schema setup and to allow `EXPLAIN` to be run on different statements. Then specify what data the Performance Schema collects and maintains. This table shows which consumers and instruments each option enables.

     ⓘ The MySQL Performance Schema must be enabled. If you select Leave As Is, verify that Global Instrumentation and Thread Instrumentation are enabled in the existing Performance Schema configuration.

Option	Server default	DPA recommended	Detailed	Leave as is
Consumer Global Instrumentation	✓	✓	✓	NC*
Consumer Thread Instrumentation	✓	✓	✓	NC
Consumer Statement Digest	✓	✓	✓	NC
Consumer Statement (Current)	✓	✓	✓	NC
Consumer Wait (Current)		✓	✓	NC
Instrument Wait (Lock/*)		✓	✓	NC
Instrument Wait (I/O table) (I/O/file)		✓	✓	NC
Instrument Wait (I/O/socket)		✓	✓	NC
Instrument Wait (Synch/*)			✓	NC

*NC = No change. DPA does not change the existing Performance Schema configuration.

Values that are outside of the MYSQL_PERFORMANCE_SCHEMA configuration scope of DPA are not changed. For example, an instrument named `stage` exists in the MySQL Performance Schema. If you enable or disable that instrument, DPA will not change it.

7. If you specified a privileged user to create the DPA monitoring user, the Allow EXPLAIN to be run on section is displayed. Select what type of statements you want DPA to collect execution plans for. The monitoring user can run `EXPLAIN` on the selected statement types.
8. Review the information and click Register Database Instance.
9. When the registration is complete, click Finish to return to the DPA home page.

Register a MariaDB database instance

Placeholder -- content TBD.

Register a PostgreSQL database instance and prepare for monitoring

This registration procedure applies to the following deployment types:

- Self-managed
- Amazon RDS for PostgreSQL
- Amazon Aurora for PostgreSQL
- Azure Database for PostgreSQL
- Cloud SQL for PostgreSQL

Complete the following tasks to register a PostgreSQL database instance for monitoring with DPA.

> You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

Differences in registering a PostgreSQL database instance

Registering a PostgreSQL database instance is slightly different than registering other types of monitored database instances:

- You cannot use the wizard to create the DPA monitoring user. Create the monitoring user manually, as described below.
- If the DPA repository is an Oracle database, DPA stores performance data for monitored PostgreSQL database instances in the default tablespace of the repository user. You cannot change the default tablespace in the Register Instance Wizard. If you need to change the default tablespace, register the instance using mass registration.
- You must configure each PostgreSQL database instance, as described below.

Task 1: Create the DPA monitoring user

Use these instructions to manually create the user that DPA uses to monitor a PostgreSQL database instance. The user will have the necessary rights and privileges.

DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will **not** cause load on the monitored instance.

1. Run the following SQL statement on the PostgreSQL database instance to create the DPA monitoring user:

   ```
   CREATE USER dpa_user WITH ENCRYPTED PASSWORD 'password';
   ```

   where *dpa_user* is the user name and *password* is the password.

2. Grant privileges to the user.

   > There are dedicated `pg_read_all_stats` and `pg_read_all_settings` roles in PostgreSQL 10 and later. For earlier versions, the `SUPERUSER` privilege is required.

   - For PostgreSQL 10.*x* and later (including EDB Postgres 10.*x* and later) do **one** of the following:
     - Run the following commands to grant the minimum required privileges:

       ```
       GRANT pg_read_all_stats, pg_read_all_settings, pg_signal_backend
       TO dpa_user;
       ```

       ```
       GRANT ALL ON ALL TABLES IN SCHEMA public TO dpa_user;
       ```

       > The second `GRANT` command enables the DPA monitoring user to collect execution plans, which it needs to generate index and table advisors. At a minimum, the monitoring user must be granted `SELECT`, `INSERT`, `UPDATE`, and `DELETE` privileges. You can use a more granular command to list specific privileges and tables. Examples include:
       >
       > ```
       > GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE table_name TO
       > dpa_user;
       > ```
       >
       > ```
       > GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA
       > public TO dpa_user;
       > ```

     - Grant the monitoring user `SUPERUSER` privileges using the commands in the next two bullets.
   - For PostgreSQL 9.6.*x* in self-managed deployments:

     ```
     ALTER USER dpa_user WITH SUPERUSER;
     ```

- For PostgreSQL 9.6.*x* in Amazon RDS and Amazon Aurora deployments:

  ```
  GRANT rds_superuser TO dpa_user;
  ```

- For PostgreSQL 9.6.*x* in Azure deployments:

  ```
  GRANT azure_pg_admin TO dpa_user;

  GRANT ALL ON ALL TABLES IN SCHEMA public TO dpa_user;
  ```

  > The second `GRANT` command enables the DPA monitoring user to collect execution plans, which it needs to generate index and table advisors. At a minimum, the monitoring user must be granted `SELECT`, `INSERT`, `UPDATE`, and `DELETE` privileges. You can use a more granular command to list specific privileges and tables. Examples include:
  >
  > ```
  > GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE table_name TO dpa_
  > user;
  >
  > GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA
  > public TO dpa_user;
  > ```

3. If you are monitoring EDB Postgres version 10, you must give the DPA monitoring user access to the `pg_stat_statements` view.

   > For EDB Postgres version 10, granting the the `pg_read_all_stats` role does **not** give the DPA monitoring user access to the view `pg_stat_statements`.

   To grant access, create a dedicated DPA schema and make a synonym of `pg_stat_statements` in it:

   ```
   CREATE SCHEMA dpa_schema;
   CREATE VIEW dpa_schema.pg_stat_statements AS SELECT * FROM enterprisedb.pg_
   stat_statements;
   GRANT USAGE ON SCHEMA dpa_schema TO dpa_user
   GRANT SELECT ON dpa_schema.pg_stat_statements TO dpa_user;
   ```

Task 2: Configure PostgreSQL database instances for DPA monitoring

Determine which monitoring mode to use

DPA offers two modes of monitoring PostgreSQL database instances. The monitoring mode you choose determines what configuration steps are required.

- **Limited monitoring** queries only the `pg_stat_activity` view. The `pg_stat_activity` view is a system view containing information about database server processes activity. Limited monitoring:
  - Is sufficient for getting wait time information for queries.
  - Returns incomplete SQL texts, and query execution statistics might be missing.
- **Complete monitoring** queries both the `pg_stat_activity` and `pg_stat_statements` views. The `pg_stat_statements` view contains execution statistics for all SQL statements executed by a server. Complete monitoring:
  - Provides complete SQL texts and query execution statistics.
  - Requires additional `pg_stat_statements` extension configuration (described in the following section).
  - Enables DPA to generate table and index advisors.

> PostgreSQL is delivered as a set of mandatory and optional packages. The `pg_stat_statements` extension provides a means for tracking SQL statement execution statistics and is required for complete monitoring. This extension is included by default in PostgreSQL distributions for Linux and Windows OS. Installation of other extensions is platform-dependent. See https://www.postgresql.org/download/ for more information.

Configure each database instance

Complete the following steps to configure each PostgreSQL database instance that you want to monitor.

1. Enable remote access to the PostgreSQL instance.

   > Remote access is enabled by default for EDB Standard and EDB Enterprise editions. For those versions, you can skip step 1b below.

   a. Adjust firewall rules to allow an incoming connection from DPA to the monitored instance. Ensure that the port the PostgreSQL instances is listening on is open (port 5432 by default).

b. (For editions other than EDB Standard and EDB Enterprise) To configure PostgreSQL accessibility, edit the `postgresql.conf` configuration file and change the `listen_address` property value to:

```
listen_address = '*'
```

> Alternatively, you can append the IP address of the DPA server to a comma-separated list of addresses.

c. To configure authentication methods for the DPA user, edit the `pg_hba.conf` configuration file and add the following host record:

```
host all dpa_user all md5
```

where *`dpa_user`* is the DPA monitoring user name created previously.

d. If the `pg_hba.conf` configuration file restricts access to the monitored instance to a range of IP addresses, ensure that the DPA server is included in the IP address range.

2. If you want to perform complete monitoring, enable and configure the `pg_stat_statements` extension for Text Poll and Stats Poll functionality:

a. Run the following command to determine if the extension is installed:

```
SELECT * FROM pg_available_extensions WHERE name = 'pg_stat_statements';
```

If there is no installed version or you receive the error `pg_stat_statements does not exist`, you must load the extension (as described in the following step). The extension is loaded by adding `pg_stat_statements` entry to `shared_preload_libraries` because it requires additional shared memory.

b. To load the `pg_stat_statements` extension (if needed) and configure it, perform one of the following tasks:

- For on-premises deployments, edit the `postgresql.conf` file and add or modify the following entries:

```
shared_preload_libraries = 'pg_stat_statements'
track_activity_query_size = 4096
pg_stat_statements.track = top
```

> Optionally, you can enter `pg_stat_statements.track = all` instead of `pg_stat_statements.track = top`.

- For Amazon RDS deployments, use the AWS Console to modify your existing custom DB Parameter Group or create a new DB Parameter Group. Then enter the following parameter values:

Parameter name	Value
pg_stat_statements.track	ALL
shared_preload_libraries	pg_stat_statements
track_activity_query_size	4096

- For Azure Single Server or Hyperscale deployments, modify your Server parameters to include the parameter values listed in the previous table.
- For Azure Flexible Server deployments, allow-list the `pg_stat_statements` extension. For information about how to allow-list an extension, see PostgreSQL extensions in Azure Database for PostgreSQL - Flexible Server.

3. Restart the PostgreSQL server.
4. Create the `pg_stat_statements` extension and the `pgstattuple` extension in the database. These extensions are database-bound and must be created for each database.

> - The `pg_stat_statements` extension must be created in the database used to connect to DPA.
> - The `pgstattuple` extension is required to enable DPA to display the Index Bloat Metrics percentage on a table advisor,

To create the extensions:

a. Connect to the PostgreSQL database instance with the DPA user account or superuser (for EDB Enterprise edition).

b. Execute following commands:

```
CREATE EXTENSION pg_stat_statements;
CREATE EXTENSION pgstattuple;
```

Task 3: (Optional) Enable DPA to collect CPU metrics from a PostgreSQL instance

To enable DPA to collect CPU metrics from a PostgreSQL instance, the system_stats extension must be installed in the PostgreSQL instance. The system_stats extension is a library of stored procedures that provide access to database server metrics.

This option can be enabled either **during** or **after** registration. If you would like to collect CPU metrics for a PostgreSQL instance that is already registered, just perform the steps in this section.

1. Download and install the extension on your PostgreSQL database server using one of the following methods:

   - Install the extension on a **Linux or macOS** server using the **PGXS framework**.

     You can build the module using the PGXS framework, which is the PostgreSQL build infrastructure for extensions.

     a. Download the `tar.gz` file from the system_stats repository in Github.

     b. Move the downloaded file to an appropriate directory, and run a command such as the following to extract the contents.

     ```
     tar -zxvf system_stats-1.0.tar.gz
     ```

     c. Make sure the `PATH` environment variable includes that directory where you extracted the `.tar` file.

     d. Compile and install the code. For example:

     ```
     cd system_stats-1.0
     PATH="/usr/local/pgsql/bin:$PATH" make USE_PGXS=1
     sudo PATH="/usr/local/pgsql/bin:$PATH" make install USE_PGXS=1
     ```

   - Install the extension on a **Linux or macOS** server using an **RPM package**.

     a. Go to https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-latest-x86_64/ and download the following package:

     `system_stats_10-1.0-1.rhel8.x86_64.rpm`

     b. To install the RPM package, run the following command from a command prompt:

     `rpm -ivh packageName`

   - Install the extension on a **Windows** server.

     You can build the module using the Visual Studio project file.

a. Download the `.zip` file from the system_stats repository in Github.

b. Extract the `.zip` file to `$PGSRC\contrib\system_stats`.

c. Set the `PG_INCLUDE_DIR` and `PG_LIB_DIR` environment variables to make sure the PostgreSQL `include` and `lib` directories can be found for compilation. For example:

```
PG_INCLUDE_DIR=C:\Program Files\PostgreSQL\12\include
PG_LIB_DIR=C:\Program Files\PostgreSQL\12\lib
```

d. Open the Visual Studio project file `system_stats.vcxproj` in the `\system_stats` directory, and build the project.

2. Run the following SQL command to install the extension in the PostgreSQL database instance:

```
CREATE EXTENSION system_stats;
```

3. To give DPA access to the collected metrics, add the DPA user to the `monitor_system_stats` role:

```
GRANT monitor_system_stats to dpa_user;
```

(Optional) Task 4: Set up SSL communication for Google Cloud instances

If a PostgreSQL instance runs in the Google Cloud Platform and you want to use SSL communication, use the Google Cloud SQL Auth proxy to enable it. Run the Google Cloud SQL Auth proxy for that database instance on the DPA server to create a secure tunnel between DPA and the Cloud SQL instance. For more information, see About the Cloud SQL Auth proxy.

Task 5: Run the Register Instance Wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Click the PostgreSQL option under Self-managed, Amazon RDS and Aurora, Microsoft Azure, or Google Cloud SQL.
3. Click Next.
4. On the Connection Information panel, enter the server name or IP address of the database instance.
5. Enter the port number.
6. Enter the name of a database in this instance.

The specified database is used during registration. DPA monitors **all** databases in the instance.

7. Under SSL mode, specify the type of secure socket layer (SSL) connections established between the instance and the DPA server:
    - Disable: SSL connections are not used.
    - Require: SSL is enabled, but no certificate checks are performed.
    - Verify-CA: SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA).

      If you select this option and DPA cannot access a trusted certificate, you are prompted to import a certificate into the DPA trust store. Click the arrow on the right to view certificate details.

    - Verify-Full: SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).

      If you select this option and DPA cannot access a trusted certificate, you are prompted to import a certificate into the DPA trust store. Click the arrow on the right to view certificate details.
8. Select the authentication method used when the DPA monitoring user connects to this database instance.
9. Enter the user name and password for the monitoring user account that you created previously, and then click Next.

10. Complete the Instance Options panel:
    a. Enter the name that DPA will display to identify this database instance.

       The Display name field defaults to the name retrieved from the database instance.

    b. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

       ⓘ If no manual groups exist, this option is not shown.

    c. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

       ⓘ If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

    d. Click Next.
11. Review the Summary panel:
    a. Review the information. If necessary, click Back to make any corrections.
    b. When all information is correct, click Register.

ⓘ You can specify which database instances DPA collects execution plans from.

Troubleshooting the registration

`Certificate is not valid` error when Verify-CA or Verify-Full mode is selected

When Verify-CA or Verify-Full SSL mode is selected, the client checks the certificate chain up to a trusted certificate authority (CA). If the root certificate is signed by a custom CA, validation can fail with the message because the custom CA is not trusted:

```
Certificate is not valid. The SSL connection test failed.
```

Work-around: Import an intermediate certificate or a server certificate from the displayed certificate chain, or follow the instructions for creating a certificate chain in the PostgreSQL documentation topic PostgreSQL Secure TCP/IP Connections with SSL.

`Connection to the server failed` error when Verify-Full mode is selected

When Verify-Full SSL mode is selected, the connection to the database instance fails if the host name does not match the name specified in a certificate's Subject Alternative Name or CN (Common Name). Validation fails with the following message:

```
Connection to the server failed. If you are trying to connect to the server using SSL, verify that your SSL certificate is valid.
```

Work-around: Do either of the following:

- Create a new certificate on the database server that matches the required database host name in its Subject Alternative Name or Common Name.
- Use Verify-CA SSL mode instead.

Unable to connect to an instance when Verify-Ca SSL or Verify-Full mode is selected

When Verify-Ca or Verify-Full SSL mode is selected, in some environments DPA might be unable to connect to a PostgreSQL instance even after a trusted certificate is imported. This can happen because DPA uses its own SSL factory implementation by default for Verify-CA and Verify-Full modes.

Work-around: Do the following to override the default DPA behavior and use the PostgreSQL default SSL factory implementation instead:

1. Verify that the following files are in the required formats:
   - The sslcert (client certificate) must be in PEM format with a `*.crt` extension.
   - The sslkey (PKCS-8 client key) must be in PKCS8 or PKCS12 format with a `*.pk8` or `*.p12` extension.
   - The sslrootcert (root certificate) must be in PEM format with a `*.crt` extension.

   If your Certificate Authority certificates are not in one of those formats and you need to convert them, you can refer to the article Convert SSL Certificates into appropriate format using OpenSSL.

   For more help configuring PostgreSQL JDBC SSL clients, see the PostgreSQL.org document Using SSL. For complete command line examples of how to export certificates in different formats, PostgreSQL suggests viewing the certdir Makefile.

   > When you use a PKCS-12 client certificate, the name or alias in the command line **must** be the actual string `user`. For example: `openssl pkcs12 -export -name user ...`.

   For information about PostgreSQL JDBC connection parameters, see Connection Parameters.
2. In the DPA Registration Wizard, click Advanced Connection Properties.

3. Under JDBC URL Properties, enter the following properties, separated by a semicolon:

Property	Value
`sslmode`	The SSL mode: • `verify-ca` • `verify-full`
`sslcert`	The location and file name of the client certificate.
`sslkey`	The location and file name of the client key.
`sslrootcert`	The location and file name of the root certificate.

For example:

```
sslmode=verify-
ca;sslcert=D:\server.crt;sslkey=D:\server.key;sslrootcert=D:\root.crt
```

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register an Amazon RDS for Oracle database instance

The following sections provide instructions for using a wizard to register an Amazon RDS for Oracle database instance to be monitored by DPA.

Alternatively, you can use mass registration to register multiple database instances, or you can register database instances using scripts that call the DPA API.

For an **Oracle RAC** (Real Application Cluster), SolarWinds recommends registering every physical instance in the cluster. Do not register the virtual IP that distributes load across the RAC instances.

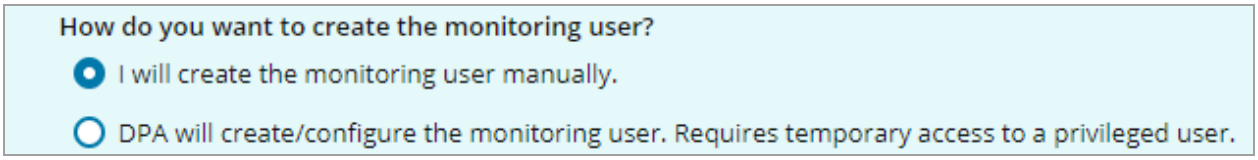
If you choose to register the virtual IP load balancing listener, or to monitor only a subset of instances in the cluster, DPA will not have complete and consistent data. This will affect DPA's tuning and resource analysis.

For more information, see DPA registration and licensing options for clustered environments.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Registration wizard options

The DPA monitoring user must be created to enable DPA to monitor a database instance. If DPA is **not** integrated with CyberArk, the following options in the registration wizard determine how the monitoring user and utility package are created or configured.



If DPA is integrated with CyberArk, these options are not displayed. You must create the monitoring user.

- **I will create the monitoring user manually.**

  With this option, you run a script to create the monitoring user. Then you use the wizard to complete the registration. You are **not** prompted to enter privileged user credentials into DPA.

  For detailed information about the privileges granted to the monitoring user, see this article.

  To use this option, see Register an Amazon RDS for Oracle database instance and create the monitoring user yourself below.

- **DPA will create/configure the monitoring user.**

  With this option you must provide the credentials of a user with the `DBA` role so that DPA can create or configure the monitoring user. DPA does not store the user credentials.

  To use this option, see Register an Oracle database instance and let DPA create the monitoing user below.

Register an Amazon RDS for Oracle database instance and create the monitoring user yourself

Task 1: Create the monitoring user

1. Copy the following script to a file: CreateMonUserAmazonOracle.sql
2. Edit the script to update the user name and password values.

3. Connect to the Amazon RDS for Oracle database, and run the script. To ensure that the connected user has all the privileges needed to create the monitoring user, SolarWinds recommends connecting as master user, which has the DBA role assigned to run the script.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Amazon RDS & Aurora, select Amazon RDS for Oracle.
3. If the monitoring user prompt is displayed, select I will create the monitoring user manually. Then click Next.

4. Complete the Connection information panel:

   a. Select the connection method, and then complete the fields required for the selected method:

Connection method	Required fields
Direct connect	• Enter the host name or IP address of the server that hosts the database instance. • Verify or update the port used for the connection. The default port is 1521. • Specify the SID (System Identifier) or Service name of the database instance.
TNS connect descriptor	In the TNS descriptor box, enter everything after `NAME=` in the `tnsnames.ora` file. The beginning `(DESCRIPTION=` is necessary. For example: `(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = myserver.mycompany.com)(PORT = 1521)))(CONNECT_DATA =(SERVICE_NAME = myserver)))`
TNS name	In the TNS name box, enter the `SERVICE_NAME` value from the `tnsnames.ora` file. ⓘ To use this option, Oracle Name Resolution must be configured. For instructions, see Connect to Oracle using name resolution.
LDAP	In the LDAP box, enter the LDAP distinguished name. ⓘ To use this option, Oracle Name Resolution must be configured. For instructions, see Connect to Oracle using name resolution.

   b. Under SSL mode, specify the type of secure socket layer (SSL) connections established between the instance and the DPA server.

ⓘ SSL connections are **not** available if LDAP is selected as the Connection method.

SSL mode	Description
Disable	SSL encryption is not used.

SSL mode	Description
Validate server certificate	SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA).
Validate server certificate and match hostname	SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).

c. Enter the user name and password of the monitoring user created previously. Or, if DPA is configured to use CyberArk, enter the CyberArk credentials query for the monitoring user.

d. Click Next.

DPA validates the connection information and the privileges of the monitoring user. If the validation is successful, the Instance options pane opens.

5. Specify the following Instance options.

ⓘ The instance name and group membership can be changed after registration.

a. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance.

By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

b. If the monitored instance contains the Oracle E-Business Suite, specify whether you want DPA to collect additional information about the suite.

DPA can capture Oracle E-Business data to identify the screens, modules, and users generating the database requests. This gives you increased visibility into the causes of performance problems in the Oracle E-Business Suite, Oracle Enterprise Resource Planning (ERP), and Oracle Applications environments.

c. Enter the name that DPA will display to identify this database instance.

The Display name field defaults to the name retrieved from the database instance.

d. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

ⓘ If no manual groups exist, this option is not shown.

e. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

> If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

f. Click Next.

6. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

Register an Amazon RDS for Oracle database instance and let DPA create the monitoring user

Task 1: Identify the privileged user

When you register a database instance using this option, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For Amazon RDS for Oracle database instances, the privileged user must be assigned the `DBA` role.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Amazon RDS & Aurora, select Amazon RDS for Oracle.
3. At the monitoring user prompt, select DPA will create/configure the monitoring user. Then click Next.

4. Enter the following connection information:

   a. Select the connection method, and then complete the fields required for the selected method:

Connection method	Required fields
Direct connect	• Enter the host name or IP address of the server that hosts the database instance. • Verify or update the port used for the connection. The default port is 1521. • Specify the SID (System Identifier) or Service name of the database instance.
TNS connect descriptor	In the TNS descriptor box, enter everything after `NAME=` in the `tnsnames.ora` file. The beginning `(DESCRIPTION=` is necessary. For example: `(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = myserver.mycompany.com)(PORT = 1521)))(CONNECT_DATA =(SERVICE_NAME = myserver)))`
LDAP or TNS name	In the LDAP/TNS name box, enter the LDAP distinguished name or the `SERVICE_NAME` value from the `tnsnames.ora` file. To use this option, Oracle Name Resolution must be configured. For instructions, see Connect to Oracle using name resolution.

   b. In the DBA Username and DBA Password fields, enter the name and password of the privileged user you identified previously.

   c. Click Next.

      DPA validates the connection information and the privileges of the privileged user. If the validation is successful, the Monitoring User pane opens.

      If registration fails because your DPA server cannot connect to the instance's server, see DPA database registration failure when attempting to register a database on an external network.

5. Create or specify the account that DPA will use to gather information (the monitoring user).

   To ensure that the account has the required permissions, SolarWinds recommends creating a new account.

- To create a new account:
    a. Next to Create Monitoring User, select Yes.
    b. Enter the user name and password.
    c. Select a Tablespace and Temp Tablespace on the monitored database. This is primarily used for gathering Explain Plan data for monitored queries.
    d. Click Next.
- To specify an existing account:
    a. Next to Create Monitoring User, select No.
    b. Enter the user name and password.

       DPA uses the default Tablespaces for that user.

       DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will **not** cause load on the monitored instance.
    c. Click Next to open the Oracle Monitoring Information pane.

6. On the Oracle Monitoring Information pane, complete the following steps:
    a. Click Yes if the monitored instance contains the Oracle E-Business Suite and you want DPA to collect additional information about the suite.

       DPA can capture Oracle E-Business data to identify the screens, modules, and users generating the database requests. This gives you increased visibility into the causes of performance problems in the Oracle E-Business Suite, Oracle Enterprise Resource Planning (ERP), and Oracle Applications environments.

       ⓘ The `SYS` password option is not available for Amazon RDS instances.
    b. Click Next.

       If your repository database is Oracle, the Oracle Repository Tablespace panel opens. If not, continue with step 8.
7. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   ⓘ If your repository database is not Oracle, the wizard skips this step.
8. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

- If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
- Group membership can be changed after registration

9. Review the information and click Register Database Instance.
10. When the registration is complete, click Finish to return to the DPA home page.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register an Amazon RDS for SQL Server database instance

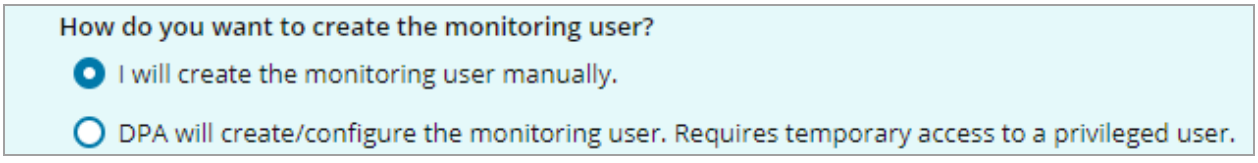
The following sections provide instructions for using a wizard to register an Amazon RDS for SQL Server database instance to be monitored by DPA.

You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Registration wizard options

The DPA monitoring user enables DPA to monitor a database instance. If DPA is **not** integrated with CyberArk, the following options in the registration wizard determine how the monitoring user and utility package are created or configured.



If DPA is integrated with CyberArk, these options are not displayed. You must create the monitoring user.

- **I will create the monitoring user manually.**

  With this option, you can create the monitoring user manually or by running a script. Then you use the wizard to complete the registration. You are **not** prompted to enter privileged user credentials into DPA.

  If you create the monitoring with one of the scripts provided, the DPA monitoring user is **not** assigned the `sysadmin` role. Because the monitoring user is not granted the `sysadmin` role, DPA has the following limitations:

  - DPA cannot collect near-zero or zero cost plans.
  - The 'SQL Server Log has Many Virtual Logs' alert does not work.
  - The 'Windows Service Not Running - SQL Server' alert does not work.

  > For detailed information about the privileges granted to the monitoring user, see this article.

  To use this option, see Register an Amazon RDS for SQL Server database instance and create the monitoring user yourself below.

- **DPA will create/configure the monitoring user.**

  With this option:

  - You must provide the credentials of a user with the `sysadmin` role so that DPA can create or configure the monitoring user.
  - The monitoring user is assigned the `sysadmin` role.

  DPA does not store the user credentials. To use this option, see Register an Amazon RDS for SQL Server database instance and let DPA create the monitoring user below.

Register an Amazon RDS for SQL Server database instance and create the monitoring user yourself

Task 1: Create the monitoring user

1. Copy the following script to a file: CreateMonUserAmazonSqlServer.sql

   > This script is valid only if the default SQL Server permissions for system roles such as [Public] have not been altered with items revoked. If default system roles have been altered, DPA Support cannot help you find all items that are assumed to be allowed.

2. Edit the script to update the user name and password values.

3. Connect to the SQL Server database instance and run the script. To ensure that the connected user has all the privileges needed to create the monitoring user, SolarWinds recommends connecting as master user name (a member of the `processadmin`, `public`, and `setupadmin` role) to run the script.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Amazon RDS & Aurora, select Amazon RDS for SQL Server.
3. If the monitoring user prompt is displayed, select I will create the monitoring user manually. Then click Next.
4. Complete the Connection information panel:
   a. Enter connection information for the SQL Server instance:
      - If the SQL Server Browser service is running, enter the server name or IP address and the instance name in this format: `Server\Instance`.
      - Otherwise, enter the server name or IP address and the port number.

      DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

   b. Under SSL mode, specify the type of secure socket layer (SSL) connections established between the instance and the DPA server.

SSL mode	Description
Disable	SSL encryption is not used.
No certificate validation	SSL is enabled, but no server certificate checks are performed. This SSL configuration does not protect against man-in-the-middle attack because no certificate is required.
Validate server certificate	SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA).
Validate server certificate and match hostname	SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).

   c. Select the type of authentication you want to use. If Mixed Mode was selected during the SQL Server installation, you can choose either option.

d. Enter the user name and password of the monitoring user created previously. Or, if DPA is configured to use CyberArk, enter the CyberArk credentials query for the monitoring user.

e. Click Next.

DPA validates the connection information and the privileges of the monitoring user.

SSL is requested by default. If the server does not support SSL, a plain connection is used.

> If registration fails because your DPA server cannot connect to the instance's server, see DPA database registration failure when attempting to register a database on an external network.

5. Specify the following Instance options.

> The instance name and group membership can be changed after registration.

a. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance.

By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

b. Enter the name that DPA will display to identify this database instance.

The Display name field defaults to the name retrieved from the database instance.

c. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

> If no manual groups exist, this option is not shown.

d. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

> If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

e. Click Next.

6. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

Register an Amazon RDS for SQL Server database instance and let DPA create the monitoring user

Task 1: Identify the privileged user

When you register a database instance using this option, you must provide the credentials of a **privileged user**. During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

The privileged user cannot be the repository database user.

Task 2: Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Amazon RDS & Aurora, select Amazon RDS for SQL Server.
3. At the monitoring user prompt, select DPA will create/configure the monitoring user. Then click Next.
4. Complete the Enter Monitored Database Instance Connection Information panel:
    a. Enter connection information for the SQL Server instance:
        - If the SQL Server Browser service is running, enter the server name or IP address and the instance name in this format: `Server\Instance`.
        - Otherwise, enter the server name or IP address and the port number.

        > DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

    b. Select the type of authentication you want to use. If Mixed Mode was selected during the SQL Server installation, you can choose either option.
    c. In the Login and Password fields, enter credentials for the privileged user that DPA can use to register the instance.
        - For Windows authentication, enter `<DOMAIN>\<username>` in the Login field.
        - For SQL Server authentication, enter the credentials that you enter on the Connect to Server dialog in SQL Server Management Studio (with Database Engine as the Server type).

        > DPA does not use or store these credentials after you complete the wizard.

    d. Click Next.

DPA validates the connection information and the privileges of the privileged user.

SSL is requested by default. If the server does not support SSL, a plain connection is used.

> If registration fails because your DPA server cannot connect to the instance's server, see DPA database registration failure when attempting to register a database on an external network.

5. Create or specify the account that DPA will use to gather information (the monitoring user).

   To ensure that the account has the required permissions, SolarWinds recommends creating a new account.

   - To create a new account:
     a. Next to Create Monitoring User, select Yes.
     b. Select SQL Server as the authentication method. (DPA cannot create a new Windows account.)
     c. Enter a user name and password for the new account, or accept the default values.
     d. Click Next.
   - To specify an existing account:
     a. Next to Create Monitoring User, select No.
     b. Select either authentication method.
     c. Enter the user name and password of an existing account.

        For Windows authentication, enter `<DOMAIN>\<username>` in the Monitoring User field.

        You can also authenticate using a Windows Computer Account.

        For SQL Server authentication, only the user name is required. Do not specify a domain.

        DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will **not** cause load on the monitored instance.

     d. Click Next.

        If your repository database is Oracle, the Oracle Repository Tablespace panel opens. If not, continue with step 7.

6. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   > If your repository database is not Oracle, the wizard skips this step.

7. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

   > - If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
   > - Group membership can be changed after registration

8. Review the information and click Register Database Instance.
9. When the registration is complete, click Finish to return to the DPA home page.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register an Amazon RDS for MySQL or MySQL-compatible Aurora database instance

Complete one of the following tasks to register an individual Amazon RDS for MySQL or MySQL-compatible Aurora database instance for monitoring with DPA.

> You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Register a read-only Amazon RDS for MySQL database instance

To register a **read-only** Amazon RDS for MySQL database instance, complete the following steps:

1. Register the corresponding read/write instance in DPA using the registration wizard.
2. Copy the user and permissions to the read-only instance.
3. Open the following file in a text editor:

   ```
   <DPA_Home>\iwc\tomcat\ignite_config\idc\system.properties
   ```

4. Add the following setting to the system.properties file and save it:

   ```
   com.confio.idc.wizard.allowDuplicateDatabaseRegistration=true
   ```

5. Use the registration wizard to register the instance:
   - For the monitoring user, choose Provide monitoring user. Then enter the credentials for the same user specified for the read/write instance.
   - On the Configuration for Monitoring panel, choose Leave As Is.

Use the registration wizard

To register an Amazon RDS for MySQL or MySQL-compatible Aurora database for DPA to monitor, complete the following tasks.

(Optional) Identify the privileged user

If you choose to let DPA create or configure the account used to collect DPA data (the monitoring user), you must provide the credentials of a **privileged user**. (You can also choose to create the monitoring user yourself.) During registration, the privileged user either creates the monitoring user or grants the required privileges to an existing user that you designate as the monitoring user. DPA does **not** store the credentials of the privileged user.

For self-managed MySQL, Percona, or Maria database instances:

- The privileged user requires the following permission:

  ```
  CREATE USER
  ```

- The privileged user must be able to grant the following permissions:

  ```
  PROCESS on *.*
  SELECT & UPDATE on performance_schema.*
  ```

- To enable the retrieval of query execution plans, the privileged user must also be able to grant the following permissions:

```
SELECT, INSERT, UPDATE, DELETE on *.*
SYSADM
```

Complete the registration wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Amazon RDS & Aurora, click Amazon RDS and Aurora for MySQL or MariaDB. Then click Next.
3. Enter the following connection information:
    a. Enter the host name or IP address and port of the server.

       DPA monitors all databases within the instance. If more than one instance exists on the server, you must register each instance separately in DPA.

    b. Create or specify the account that DPA will use to gather information (the monitoring user).

       SolarWinds recommends creating a separate account for the monitoring user.

       DPA ignores data generated by the monitoring user on the monitored database instance. For this reason, do not specify a user that causes load on the monitored instance.

       - To let DPA create or configure the monitoring user:
           a. Select Let DPA create a new user or configure an existing user for me.
           b. Enter the credentials of an existing user with the required privileges.

              ⓘ The credentials for the privileged user are not used or stored after the registration.

           c. Select a Tablespace and Temp Tablespace on the monitored database. This is primarily used for gathering Explain Plan data for monitored queries.
           d. Click Next.
       - To create the monitoring user yourself:
           a. Select I'll create the database user.
           b. Click Monitoring User Creation Script.

c. Copy the script to a file and edit it per the instructions.

d. Copy the edited script to the MySQL console, and run it.

e. Enter this user's credentials in the Username and Password fields.

If your repository database is Oracle, the Oracle Repository Tablespace panel opens. If not, continue with step 5.

4. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   ⓘ If your repository database is not Oracle, the wizard skips this step.

5. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

   ⓘ
   - If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
   - Group membership can be changed after registration

6. Select a Typical or Custom configuration. SolarWinds recommends the Typical configuration

   - Typical is recommended. With this option:
     - The DPA Recommended option is used for Performance Schema setup.
     - `EXPLAIN` can be run on `SELECT` statements.
   - Select Custom to change the Performance Schema setup and to allow `EXPLAIN` to be run on different statements. Then specify what data the Performance Schema collects and maintains. This table shows which consumers and instruments each option enables.

     ⓘ The MySQL Performance Schema must be enabled. If you select Leave As Is, verify that Global Instrumentation and Thread Instrumentation are enabled in the existing Performance Schema configuration.

Option	Server Default	DPA Recommended	Detailed	Leave as Is
Consumer Global Instrumentation	✓	✓	✓	NC*

Option	Server Default	DPA Recommended	Detailed	Leave as Is
Consumer Thread Instrumentation	✓	✓	✓	NC
Consumer Statement Digest	✓	✓	✓	NC
Consumer Statement (Current)	✓	✓	✓	NC
Consumer Wait (Current)		✓	✓	NC
Instrument Wait (Lock/*)		✓	✓	NC
Instrument Wait (I/O table) (I/O/file)		✓	✓	NC
Instrument Wait (I/O/socket)		✓	✓	NC
Instrument Wait (Synch/*)			✓	NC

*NC = No change. DPA does not change the existing Performance Schema configuration.

Values that are outside of the MYSQL_PERFORMANCE_SCHEMA configuration scope of DPA are not changed. For example, an instrument named `stage` exists in the MySQL Performance Schema. If you enable or disable that instrument, DPA will not change it.

7. If you specified a privileged user to create the DPA monitoring user, the Allow EXPLAIN to be run on section is displayed. Select what type of statements you want DPA to collect execution plans for. The monitoring user can run `EXPLAIN` on the selected statement types.
8. Review the information and click Register Database Instance.
9. When the registration is complete, click Finish to return to the DPA home page.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register an Amazon RDS for MariaDB database instance

Placeholder -- content TBD.

Register an Azure SQL Database

See the following sections to register an individual Azure SQL Database for monitoring with DPA.

> You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

> For DPA to collect metrics from a monitored SQL Server instance, Azure SQL instance, or ASMI, the SQL option `NUMERIC_ROUNDABOUT` must be set to `OFF`.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Options for creating the monitoring user

DPA requires an account with certain privileges to gather information from the database instance. This account is called the monitoring user. When you identify and authenticate the monitoring user in the registration wizard, you have the following options:

- Allow DPA to create the monitoring user or configure an existing user to give it the required privileges.

  > If you let DPA create the monitoring user, you must enter the credentials of a privileged user. These credentials are **not** saved after the registration is complete.

- Create the monitoring user manually by running a script.
- If you don't want to create, manage, and update user credentials, you can use a Microsoft Entra service principal (previously Azure service principle). See the prerequisites below.

Prepare to authenticate with a Microsoft Entra ID password

If you plan to use a Microsoft Entra ID (formerly Azure AD) password to authenticate the DPA monitoring user, follow the instruction in the support article Use Microsoft Entra ID (formerly Azure AD) authentication in DPA.

Prepare to authenticate with a service principal

If you plan to use a service principal to authenticate the DPA monitoring user, complete the following steps:

1. In the Azure portal (https://portal.azure.com) under Azure services, click App registrations. Then register an app.

   > You can use an existing app if you already have one.

2. Use the SQL Server Management Studio (SSMS) to connect to the Azure SQL Database at the database level, and run the following SQL statements to create a user and give it the `db_owner` role:

   ```
   CREATE USER appName FROM EXTERNAL PROVIDER;
   ALTER ROLE db_owner ADD member appName;
   ```

Complete the Register Instance Wizard

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Microsoft Azure, select Azure SQL Database. Then click Next.

   The Enter Monitored Database Instance Connection Information panel opens.

3. Enter the logical server name, port, and database name.

   > You cannot enter an IP address in the Server Name field.

4. Create or specify the monitoring user account. (See the options for the monitored user account above.) Then click Next.

   > To register a read-only geo-replica, you must create a monitoring account through the primary server first.

   - To let DPA create or configure the monitoring user:
     a. Select Let DPA create a new contained user.
     b. In the Username and Password fields, enter the credentials of a privileged user account. This account must be assigned the `db_owner` role.
     c. Click Next.

        DPA verifies the connection information and the credentials.

d. Enter the credentials of the monitoring user. This can be either:

   - An account that DPA creates.
   - An existing account that DPA configures to assign the required privileges.

e. Click Next.

- To create the monitoring user yourself:

  a. Select I'll create the contained user or login.

  b. Under Authentication Method, select either SQL User or Microsoft Entra Password (formerly Azure Active Directory password) as the type of authentication to use for the monitoring user account.

  c. Click Monitoring User Creation Script.

  d. Copy the script to a file and edit it per the instructions.

  e. Run the edited SQL statements on the Azure SQL database.

     The monitoring user is created.

  f. Enter this user's credentials in the Username and Password fields.

  g. Click Next.

- To specify a service principal:

  a. Select I'll create the contained user or login.

  b. Under Authentication Method, select Microsoft Entra Service Principal.

  > Do **not** specify the authentication method in the Connection Properties field.

  c. In the Username field, enter the service principal application ID.

  > You must enter the application ID, **not** the display name or object ID.

  d. In the Password field, enter the value of the service principal secret.

If your repository database is Oracle, the Oracle Repository Tablespace panel opens. If not, continue with step 6.

5. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance. Then click Next.

   By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

   > If your repository database is not Oracle, the wizard skips this step.

6. (Optional) Select the alert groups you want the new database instance to join. Then click Next.

   > - If no alert groups exist, or the existing groups do not match the database type of this instance, the wizard skips this step.
   > - Group membership can be changed after registration

7. Review the information and click Register Database Instance.
8. When the registration is complete, click Finish to return to the DPA home page.

Enable deadlocks for read-only geo-replicas

To enable the deadlock feature for read-only geo-replica Azure SQL Databases, you must create and enable an Extended Event Session (EES).

If you registered the primary server first, an EES is already created and synced. Skip to step 2.

Otherwise, connect to the primary server first to create an EES.

1. Run the following SQL statement:

```
CREATE EVENT SESSION [dpa_deadlock_capture] ON DATABASE
ADD EVENT sqlserver.xml_deadlock_report
ADD TARGET package0.ring_buffer(SET max_events_limit=(1000),
  max_memory=(256))
WITH (MAX_MEMORY = 256KB,
EVENT_RETENTION_MODE = ALLOW_SINGLE_EVENT_LOSS,
MAX_DISPATCH_LATENCY = 30 SECONDS,
MAX_EVENT_SIZE = 0KB,
MEMORY_PARTITION_MODE = NONE,
TRACK_CAUSALITY = OFF,
STARTUP_STATE = ON);
-- ALTER EVENT SESSION [dpa_deadlock_capture] ON DATABASE STATE = START;
```

2. Connect to the read-only replica database.

3. Click Extended Events > Sessions.
4. Enable the dpa_deadlock_capture session.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register an Azure SQL Managed Instance

Complete the following tasks to register a single Azure SQL Managed Instance (ASMI) for DPA to monitor.

You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

For DPA to collect metrics from a monitored SQL Server instance, Azure SQL instance, or ASMI, the SQL option `NUMERIC_ROUNDABOUT` must be set to `OFF`.

Registering an ASMI is slightly different than registering other types of monitored database instances:

- You cannot use the wizard to create the DPA monitoring user. Do one of the following:
  - Create the monitoring user manually.
  - If you don't want to create, manage, and update user credentials, you can use a Microsoft Entra service principal (previously Azure service principle). See the prerequisites below.
- If the DPA repository is an Oracle database, DPA stores performance data for monitored ASMIs in the default tablespace of the repository user. You cannot change the default tablespace in the Register Instance Wizard. If you need to change the default tablespace, register the instance using mass registration.

Create the monitoring user

If you are not using a Microsoft Entra service principal as the DPA monitoring user, you must create a user account to serve as the DPA monitoring user. DPA uses this account to register and monitor the instance.

DPA ignores data on the monitored database instance from the monitoring user. Make sure the monitoring user will **not** cause load on the monitored instance.

The monitoring user account must have the `SYSADMIN` role during registration. After you have registered the ASMI, you can (optionally) remove the SYSADMIN role from the DPA monitoring user. If you remove the SYSADMIN role, DPA functionality is limited. For example, the SQL Server Log Has Many Virtual Logs alert does not work.

1. Connect as a SYSADMIN user to the ASMI database instance you want to monitor.
2. Run the following SQL statements against the database you are registering. Replace *username* and *password* with the credentials for the user account. Replace the default database and language values if needed.

```
--Create DPA login

CREATE LOGIN username WITH
  PASSWORD=N'password',
  DEFAULT_DATABASE=master,
  DEFAULT_LANGUAGE=us_english,
  CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO

--Give SYSADMIN rights to the DPA user
EXEC sys.sp_addsrvrolemember @loginame = N'username', @rolename =
N'sysadmin'
GO
```

Prepare to authenticate with a Microsoft Entra ID password

If you plan to use a Microsoft Entra ID (formerly Azure AD) password to authenticate the DPA monitoring user, follow the instruction in the support article Use Microsoft Entra ID (formerly Azure AD) authentication in DPA.

Prepare to authenticate with a service principal

If you plan to use a service principal to authenticate the DPA monitoring user, complete the following steps:

1. In the Azure portal (https://portal.zaure.com) under Azure services, click App registrations. Then register an app.

   You can use an existing app if you already have one.

2. Use the SQL Server Management Studio (SSMS) to connect to the ASMI database instance at the server level, and run the following SQL statements to create a user and give it the `db_owner` role:

```
CREATE USER appName FROM EXTERNAL PROVIDER;
ALTER ROLE db_owner ADD member appName;
```

3. Run the following SQL statements to grant the `sysadmin` role.

```
CREATE LOGIN appName FROM EXTERNAL PROVIDER WITH
  DEFAULT_DATABASE=master,
  DEFAULT_LANGUAGE=us_english
ALTER SERVER ROLE sysadmin ADD MEMBER appName
```

Complete the Register Instance Wizard

1. On the DPA home page, click Register DB Instance for Monitoring.
2. Under Microsoft Azure, select Azure SQL Managed Instance. Then click Next.
3. On the Enter Monitored Database Instance Connection Information panel, enter the server name or IP address of the ASMI and the port number.
4. Under Authentication method, select the type of authentication to use for the monitoring user account:
   - SQL User
   - Microsoft Entra Password (formerly Azure Active Directory password)
   - Microsoft Entra Service Principal (formerly Azure service principal)

   Do **not** specify the authentication method in the Connection Properties field.
5. Specify the monitoring user credentials, and then click Next:

- To specify an account, enter the user name and password for the monitoring user account that you created previously.
- To specify a service principal:
  a. In the Username field, enter the service principal application ID.

  > You must enter the application ID, **not** the display name or object ID.

  b. In the Password field, enter the value of the service principal secret.

6. Specify the following Instance options.

   > The instance name and group membership can be changed after registration.

   a. Enter the name that DPA will display to identify this database instance.

      The Display name field defaults to the ASMI name retrieved from the instance.

   b. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

      > If no manual groups exist, this option is not shown.

   c. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

      > If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

   d. Click Next.

7. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register an Azure Database for MySQL

Complete the following steps to register an individual Azure Database for MySQL instance for monitoring with DPA.

You can use the registration wizard to register a **read/write** MySQL database instance. To register a **read-only** instance, see Monitor a read-only MySQL database instance in DPA.

To optimize DPA's reporting capabilities for an Azure Database for MySQL database instance, see the requirements for monitoring MySQL database instances.

You can also use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

If you register a database instance within the 14-day trial period, DPA begins monitoring the instance immediately. After the trial period, you must activate a license to monitor the database instance.

Task 1: Create the monitoring user

Complete the following steps to create the user that DPA will use to monitor the database instance. Alternatively, you can specify an Azure AD Service Principal if you don't want to create, manage, and update user credentials.

1. Create the DPA monitoring user in Azure Database for MySQL. For more information, see Create users in Azure Database for MySQL in the Azure documentation.
2. Run the following commands to grant the monitoring user the privileges required for monitoring:

```
GRANT SELECT ON performance_schema.* TO 'dpa_user'@'%';

GRANT UPDATE ON performance_schema.* TO 'dpa_user'@'%';

GRANT PROCESS ON *.* TO 'dpa_user'@'%';

GRANT SELECT ON *.* TO 'dpa_user'@'%';
```

3. Run the following commands to grant the monitoring user the privileges required to enable the retrieval of query execution plans:

```
GRANT INSERT ON *.* TO 'dpa_user'@'%';

GRANT UPDATE ON *.* TO 'dpa_user'@'%';

GRANT DELETE ON *.* TO 'dpa_user'@'%';
```

Task 2: Register the database instance

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Microsoft Azure, select Azure Database for MySQL. Then click Next.
3. On the Connection information panel, enter the host name or IP address of the database instance.
4. Enter the port number.
5. Under SSL mode, specify the type of secure socket layer (SSL) connections established between the instance and the DPA server.

SSL mode	Description
Disable	SSL encryption is not used.
Require	SSL is enabled, but no server certificate checks are performed. This SSL configuration does not protect against man-in-the-middle attack because no certificate is required.
Verify-CA	SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA). If you select this option and DPA cannot access a trusted certificate, you are prompted to import a certificate into the DPA trust store. Click the blue arrow › to view certificate details.
Verify-Identity	SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN). If you select this option and DPA cannot access a trusted certificate, you are prompted to import a certificate into the DPA trust store. Click the blue arrow › to view certificate details.

6. Enter the user name and password of the monitoring user created previously. Or, if DPA is configured to use CyberArk, enter the CyberArk credentials query for the monitoring user. Then click Next.

   DPA validates the connection information and the privileges of the monitoring user.

   ⓘ DPA does **not** support connecting to the instance through a gateway.

7. Specify the following Instance Options.

> The instance name and group membership can be changed after registration.

a. Enter the name that DPA will display to identify this database instance.

   The Display name field defaults to the name retrieved from the database instance.

b. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

   > If no manual groups exist, this option is not shown.

c. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

   > If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

d. Click Next.

8. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register a SQL Server instance running in the Google Cloud Platform

Task 1: Create the monitoring user with the necessary permissions

1. On the Google Cloud Platform (GCP) website, navigate to the SQL Server instance that you want to register.
2. Under Users, create a new user for DPA monitoring. Then run the following script using the default `sqlserver` or any other privileged account to assign the permissions needed for monitoring.

   The majority of the permissions are optional. DPA can monitor without them, but the monitoring will be limited. Limitations are described in the following sections.

The following script is valid only if the default SQL Server permissions for system roles such as `[Public]` have not been altered with items revoked. If default system roles have been altered, DPA Support cannot help you find all items that are assumed to be allowed.

```
--grant privileges to login
USE [master]
GRANT VIEW SERVER STATE TO [dpa_mon_user] AS CustomerDbRootRole
GRANT VIEW ANY DATABASE TO [dpa_mon_user] AS CustomerDbRootRole
GRANT VIEW ANY DEFINITION TO [dpa_mon_user] AS CustomerDbRootRole
--assign processaadmin role to allow kill sessions
ALTER SERVER ROLE [processadmin] ADD MEMBER [dpa_mon_user]
--create user on each database and grant db_datareader role
--in case there are databases not accessible by sqlserver user, you need to
grant the permissions by the database owner
DECLARE @DPA_User VARCHAR (50) = 'dpa_mon_user';
DECLARE @dbname VARCHAR(50);
DECLARE @SQL NVARCHAR(max);
DECLARE dbs CURSOR LOCAL FAST_FORWARD FOR
    SELECT name FROM master.dbo.sysdatabases where name NOT IN ('master',
'msdb', 'tempdb', 'model');
OPEN dbs;
FETCH NEXT FROM dbs INTO @dbname;
WHILE @@FETCH_STATUS = 0
BEGIN
    SET @SQL = 'use '+@dbname +';
    CREATE USER '+@DPA_User+' FOR LOGIN '+@DPA_User+';
    EXECUTE sp_addrolemember N''db_datareader'', '+@DPA_User+';'
    EXECUTE sp_executesql @SQL;
    FETCH NEXT FROM dbs INTO @dbname;
END;
CLOSE dbs;
DEALLOCATE dbs;
```

Task 2: Register the database instance

You can register the instance using the Register Database wizard, mass registration wizard, or the DPA API.

Register Database wizard

Complete the following steps to use the wizard to Register Database register one instance at a time.

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Google Cloud SQL, select Cloud SQL for SQL Server.
3. Complete the Connection information panel:
   a. Enter connection information for the SQL Server instance:
      - If the SQL Server Browser service is running, enter the server name or IP address and the instance name in this format: `Server\Instance`.
      - Otherwise, enter the server name or IP address and the port number.
   b. Under SSL mode, specify the type of secure socket layer (SSL) connections established between the instance and the DPA server.

SSL mode	Description
Disable	SSL encryption is not used.
No certificate validation	SSL is enabled, but no server certificate checks are performed. This SSL configuration does not protect against man-in-the-middle attack because no certificate is required.
Validate server certificate	SSL is enabled. The client verifies that the server is trustworthy by checking the certificate chain up to a trusted certificate authority (CA).
Validate server certificate and match hostname	SSL is enabled. The client verifies the certificate chain and also verifies that the server hostname matches its certificate's Subject Alternative Name or Common Name (CN).

   c. Enter the user name and password of the monitoring user created previously. Or, if DPA is configured to use CyberArk, enter the CyberArk credentials query for the monitoring user.
   d. Click Next.

      DPA validates the connection information and the privileges of the monitoring user.
4. Specify the following Instance options.

   The instance name and group membership can be changed after registration.

   a. If your repository database is Oracle, choose the tablespace in the repository database to store DPA performance data for this monitored instance.

      By default, the performance data is stored in the default tablespace of the repository user. However, data for monitored instances can be stored in separate tablespaces.

b. Enter the name that DPA will display to identify this database instance.

   The Display name field defaults to the name retrieved from the database instance.

c. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

   > If no manual groups exist, this option is not shown.

d. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

   > If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

e. Click Next.

5. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

Mass registration

Use the mass registration wizard to register multiple instances. The following shows an example of a spreadsheet for mass registration used to register a GCP SQL Server database instance.

- Leave the `Privileged User` and `Privileged User Password` columns empty.
- In the `Create Monitoring User (Y/N)` column, enter `N`.
- In the `Monitoring User` and `Monitoring User Password` columns, enter the credentials of the monitoring user created in the previous task.
- In the `Deployment` column, enter `Google`.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Database Type	Display Name	Server	Port	Service Na	Database	Privileged	Privileged	Monitoring User	Monitoring User Password	Create Monitoring User (Y/N)	Deployment
2	SQL Server	My GCP MsSql	34.122.1.2	1433					dpa_mon_user	Pa$$word1	N	Google

DPA REST API

Run scripts that call the DPA REST API to register instances. The following example is a segment of a JSON script that calls the DPA API to register a GCP SQL Server database instance. Change the values to reflect your monitored database and the credentials of the monitoring user created in the previous task.

```
{
  "databaseType": "SQLSERVER",
  "serverName": "34.122.1.2",
  "port": 1433,
  "monitoringUser": "dpa_mon_user",
```

```
  "monitoringUserPassword": "Pa$$word1",
  "deployment": "GOOGLE"
}
```

Monitoring user validation

During the registration, DPA validates that the monitoring user has the required permissions. By default, the registration fails if any mandatory permission is missing and displays a warning if optional permissions are missing. The same validations are also done when database connection details are updated.

This behavior can be overridden by setting the advanced Support option `MONITOR_VALIDATION_OVERRIDE` to `TRUE`. This allows the registration to pass regardless of which permissions are missing. All permissions are treated as optional and DPA displays a warning.

Impact of specific permissions on DPA monitoring

DPA validates that the monitoring login and user has the following permissions.

Permissions granted to login

Unless otherwise stated, the permissions are optional. (If optional permissions are missing, basic monitoring is possible but specific data or functionality is not available.)

Permission	Effect on DPA monitoring
`VIEW SERVER STATE` (mandatory)	Required to access the Dynamic Management Views used for polling. Without it, DPA cannot monitor the instance.
`VIEW ANY DATABASE`	Required to see any database. Without it, DPA can monitor only `master`, `tempdb`, and any databases that the DPA monitoring user owns.
`VIEW ANY DEFINITION`	Required to access the definitions of tables, indexes, and other database objects. Without it, Storage I/O data and Table Tuning Advisor current definitions are not available.

Role membership

Those roles are optional.

Role	Description
`processadmin`	Required to kill sessions. Without it, the Kill Session option in the Real Time Sessions view does not work.
`db_datareader`	Monitoring user needs to be created on each database before granting the role. Without it: • SQL texts will not be available. • Table Tuning Advisors cannot include current data, table size, and table churn. • 'Database Freespace' and 'Transaction Log Freespace' alerts will not function for all databases. ⓘ This has to be granted to any database created on the monitored instance after registration, because newly created databases will not have the user with this role created automatically. This is a limitation of GCP SQL Server, which is not able to grant the `CONNECT ANY DATABASE` privilege.

Limitations due to not having the sysadmin role in Cloud SQL for SQL Server

Some of the permissions that DPA uses can only be granted by granting the sysadmin role to the DPA monitoring user. Without the sysadmin role, the following alerts will not work:

- SQL Server Error Log Alert
- SQL Server Log has Many Virtual Logs
- SQL Server Long Running Jobs
- Windows Service Not Running - SQL Server

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Register a MySQL instance running in the Google Cloud Platform

The following sections provide instructions for using a wizard to register a MySQL instance running in the Google Cloud Platform for monitoring with DPA.

Alternatively, you can use mass registration to registrations to register multiple database instances, or you can register database instances using scripts that call the DPA API.

Task 1: Create the monitoring user

Complete the following steps to create the user that DPA will use to monitor the MySQL database instance.

1. Create the DPA monitoring user through the Google Cloud Platform. For more information, see Create a user in the Cloud SQL documentation.
2. Run the following commands to grant the monitoring user the privileges required for monitoring:

```
GRANT SELECT ON performance_schema.* TO 'dpa_user'@'%';

GRANT UPDATE ON performance_schema.* TO 'dpa_user'@'%';

GRANT PROCESS ON *.* TO 'dpa_user'@'%';

GRANT SELECT ON *.* TO 'dpa_user'@'%';
```

3. Run the following commands to grant the monitoring user the privileges required to enable the retrieval of query execution plans:

```
GRANT INSERT ON *.* TO 'dpa_user'@'%';

GRANT UPDATE ON *.* TO 'dpa_user'@'%';

GRANT DELETE ON *.* TO 'dpa_user'@'%';
```

Task 2: Register the database instance

1. In the upper-left corner of the DPA home page, click Register DB Instance for Monitoring.
2. Under Google Cloud SQL, select Cloud SQL for MySQL. Then click Next.
3. Complete the Connection information panel:

a. Enter the host name or IP address of the database instance.

b. Enter the port number.

c. Enter the user name and password of the monitoring user created previously. Or, if DPA is configured to use CyberArk, enter the CyberArk credentials query for the monitoring user.

d. Click Next.

DPA validates the connection information and the privileges of the monitoring user.

4. Specify the following Instance Options.

> The instance name and group membership can be changed after registration.

a. Enter the name that DPA will display to identify this database instance.

The Display name field defaults to the name retrieved from the database instance.

b. (Optional) If you have manually created instance groups, you can assign this database instance to one of the groups.

> If no manual groups exist, this option is not shown.

c. (Optional) If you have existing alert groups, you can assign this database instance to one or more groups.

> If no alert groups exist, or the existing groups do not match this instance's database type, this option is not shown.

d. Click Next.

5. Review the information on the Summary page. Click Back if you need to make changes. When the information is correct, click Register.

Task 3: Set up SSL communication for Cloud SQL for MySQL instances (Optional)

If you want to use SSL communication for a Cloud SQL for MySQL instance, use the Google Cloud SQL Auth proxy to enable it. Run the Google Cloud SQL Auth proxy for that database instance on the DPA server to create a secure tunnel between DPA and the Cloud SQL for MySQL instance. For more information, see About the Cloud SQL Auth proxy.

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Unregister a monitored database instance

If you want to remove one of your monitored database instances from DPA, you must unregister it.

⚠ If you unregister a monitored database instance, DPA stops monitoring the instance and removes all historical performance data from the repository.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Database Instances, click Unregister DB Instance.
3. Select a database instance, and click Next.
4. Determine which DPA objects (if any) to remove from the database instance:
   - If the database instance is **not** currently running or cannot be reached, do **not** select any objects. Click Next.

     ⓘ If you select objects and DPA cannot access the instance to remove them, DPA cannot unregister the instance.

   - If the database instance is running and can be reached, select the DPA objects to remove and then click Next.

     Depending on the database type, you can remove one or both of the following objects:

     - **Monitoring User:** You can remove the monitoring user if no other applications, including other installations of DPA, are using this user.
     - **DPA Database Objects:** This refers to tables that are created in the schema of the monitoring user. If you remove the monitoring user, these objects are removed since they are owned by the monitoring user. You can remove these objects if no other installations of DPA are monitoring this instance.

     ⓘ You cannot remove objects on certain database types, such as read-only replicas.
5. Confirm the unregistration information, and click Unregister Database Instance. This may take

several minutes.

6. Click Finish to complete the unregistration.

Register an Azure Database for MariaDB

Placeholder -- content TBD.

Database instance groups

See the following topics for information about creating and monitoring groups in DPA:

- About monitoring SQL Server Availability Groups with DPA describes the information that DPA provides about SQL Server AGs.
- About monitoring Oracle multitenent databases describes the information that DPA provides about Oracle CDBs.
- Manually group database instances in DPA describes how to create and modify custom groups.
- View information about a group of database instances explains how to view information about all instances in a group.

About monitoring SQL Server Availability Groups with DPA

DPA provides status information, annotations, and alerts for your SQL Server Availability Groups (AGs).

- For information about your choices for registering SQL Server AGs, see Registration and licensing options for clustered environments.
- DPA does **not** support monitoring distributed AGs (DAGs). DPA can monitor the SQL Server instances that participate in a distributed AG, but the AG monitoring features are not enabled for distributed AGs.

Automatic naming

When you register an AG listener, DPA automatically names the instance using the following format:

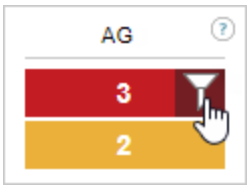
<PrimaryReplicaName> via <ListenerName>

When a failover occurs, the name is automatically updated to reflect the new primary replica.

If you manually change the display name of an AG that is registered via the listener, by default DPA overwrites the name each time the monitor starts. To change the default behavior and manually specify the name, edit the advanced Support option `AG_INSTANCE_NAME_UPDATE_ENABLED`.

AG information in DPA

On the DPA home page, the AG Status Summary box in the Status Summary area shows the number of database instances with partially healthy or not healthy AGs. As with other status boxes, click the filter next to a status to display only instances associated with the selected status.



The AG status icon AG identifies database instances that include AGs. The color of the dot provides status information (described in the following section). Possible statuses are:

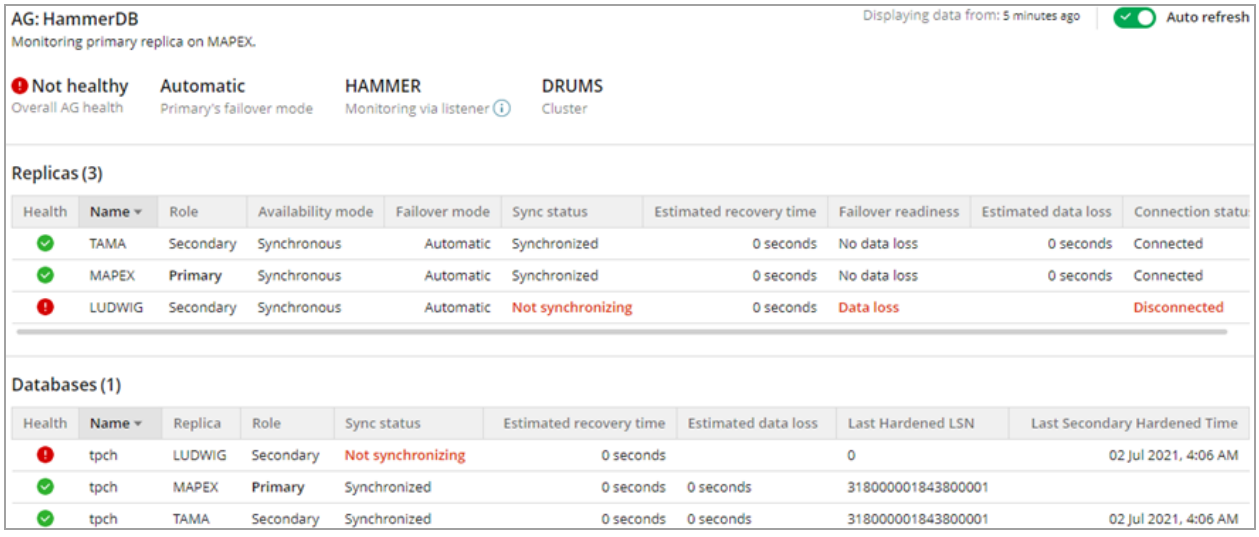
- Green for healthy
- Yellow for partially healthy
- Red for not healthy
- Gray for unknown

To view detailed information:

1. From the DPA home page, click an AG status icon to open the Availability Group Summary view.

   This view shows information about each AG in the database instance. DPA shows status information for primary replicas in the instance. For secondary replicas, the status of the primary replica is displayed if DPA is also monitoring the primary replica.

2. Click any link to view detailed information about the databases and replicas in the AG.



How DPA determines the AG status

If the instance is monitored directly and acting as a primary replica

If you are monitoring the instance directly (not through a listener), DPA does the following:

1. Determines the status of all AGs that the instance acts as the primary replica for.
2. Displays the worst status.

Example: An instance is acting as the primary replica for four availability groups. Their statuses are:

- AG1: Healthy
- AG2: Healthy
- AG3: Partially Healthy
- AG4: Not Healthy

DPA shows the status as **Not Healthy**.

AG1 (primary replica)	AG2 (primary replica)	AG3 (primary replica)	AG4 (primary replica)	**DPA status**
Healthy	Healthy	Partially Healthy	Not Healthy	**Not Healthy**

If the instance is monitored directly and acting as a secondary replica

If an instance is acting as a secondary replica for any AGs, that AG's status is Unknown. If the instance **also** acts as a primary replica for one or more AGs, the Unknown status is ignored.

Example: An instance acts as the primary replica for three availability groups. Their statuses are:

- AG1: Healthy
- AG2: Healthy
- AG3: Partially Healthy

The instance also acts as a secondary replica for one AG. Its status is Unknown.

DPA ignores the Unknown status, and shows the status as **Partially Healthy**.

AG1 (primary replica)	AG2 (primary replica)	AG3 (primary replica)	AG4 (secondary replica)	**DPA status**
Healthy	Healthy	Partially Healthy	Unknown	**Partially Healthy**

If DPA shows the AG status as Unknown, that typically indicates that the instance is acting as a secondary replica for all AGs.

If the instance is monitored via the listener

If you are monitoring the instance via the listener, by default DPA displays the aggregate status as described above. However, you can edit the advanced Support option `AG_STATUS_ROLLUP_USE_PRIMARY` to determine the status using only the AG associated with the listener.

AG alerts

DPA provides the following AG alerts: SQL Server Availability Group Failover and SQL Server Availability Group Status Change.

SQL Server Availability Group Failover

This alert is triggered when an AG failover occurs. DPA sends alerts based on how you registered instances:

- If you registered database instances directly (not through a listener), when a failover occurs DPA sends an alert for each instance involved in the failover that it is monitoring. For example, if an AG fails over from Instance1 to Instance2 and DPA is monitoring both instances, you receive two alerts. If DPA is monitoring only one of the instances, you receive only one alert.
- If you registered the AG through a listener and the AG associated with the listener fails over, DPA sends one alert, because the listener moves with the AG from Instance1 to Instance2.

If multiple AG failovers occur in a short period of time, DPA aggregates them into one alert per instance.

SQL Server Availability Group Status Change

This alert is triggered when an AG status changes to Partially Healthy or Not Healthy. DPA evaluates AG statuses every 10 minutes by default. You are alerted if the status changes from Healthy to Partially Healthy or Not Healthy between the evaluations. If the status changes from Healthy to another status and then back to Healthy during the same evaluation period, you are not alerted.

Example: In this example, DPA is monitoring an instance that acts as a primary replica for three AGs. The following table shows how the alerts would behave for each of DPA's alert policies. Changes to alert levels are shown in red.

Interval	AG status (change in red)	Policy: Notify when level not visited since normal	Policy: Notify when level changes	Policy: Notify when level is not normal
1	AG1 Healthy AG2 Healthy AG3 Healthy	No alert	No alert	No alert

Interval	AG status (change in red)	Policy: Notify when level not visited since normal	Policy: Notify when level changes	Policy: Notify when level is not normal
2	AG1 Healthy AG2 Partially Healthy AG3 Healthy	AG2 Partially Healthy	AG2 Partially Healthy	AG2 Partially Healthy
3	AG1 Healthy AG2 Partially Healthy AG3 Partially Healthy	AG3 Partially Healthy	AG3 Partially Healthy	AG2 Partially Healthy AG3 Partially Healthy
4	AG1 Healthy AG2 Not Healthy AG3 Not Healthy	AG2 Not Healthy AG3 Not Healthy	AG2 Not Healthy AG3 Not Healthy	AG2 Not Healthy AG3 Not Healthy
5	AG1 Partially Healthy AG2 Not Healthy AG3 Not Healthy	AG1 Partially Healthy	AG1 Partially Healthy	AG1 Partially Healthy AG2 Not Healthy AG3 Not Healthy
6	AG1 Healthy AG2 Partially Healthy AG3 Partially Healthy	No alert	AG1 Healthy AG2 Partially Healthy AG3 Partially Healthy	AG2 Partially Healthy AG3 Partially Healthy
7	AG1 Partially Healthy AG2 Healthy AG3 Healthy	AG1 Partially Healthy	AG1 Partially Healthy AG2 Healthy AG2 Healthy	AG1 Partially Healthy
8	AG1 Healthy AG2 Healthy AG3 Healthy	No alert	AG1 Healthy	No alert

Automatic annotations when AG failovers occur

Annotations are automatically added to wait time charts when an AG failover occurs. The annotations allow you to compare changes in performance before and after a failover.

The number of annotations depends on how you registered instances:

- If you registered database instances directly (not through a listener), when a failover occurs DPA adds an annotation for each instance involved in the failover that it is monitoring. For example, if an AG fails over from Instance1 to Instance2 and DPA is monitoring both instances, DPA adds two annotations. If DPA is monitoring only one of the instances, DPA adds only one annotation.
- If you registered the AG through a listener and the AG associated with the listener fails over, DPA adds one annotation.

If you do not want to add an annotation when an AG failover occurs, edit the advanced option `AG_EVENT_ANNOTATIONS_ENABLED`.

About monitoring Oracle multitenent databases

If you are using DPA to monitor the databases within an Oracle multitenent database (CDB), see the following information about registration, grouping, and annotations.

Registration and automatic grouping

Register each of the pluggable databases (PDBs) contained in the CDB. Register each PDB just as you would register an Oracle single tenant database. For more information, see Registration and licensing options for clustered environments.

You cannot register the CDB for monitoring. DPA monitors only the PDBs.

When you register two or more Oracle PDBs in the same CDB, DPA automatically creates a group for the CDB. This group is used for all registered PDBs from the CDB. If a DBA moves a PDB to a new CDB, DPA processes and groups the instance.

View the PDB load

On the DPA home page, you can:

- Click the CDB name to view summary data from all PDB instances in the group. Use this view to determine which PDB has the most wait time and what types of waits the PDBs are experiencing.

- Expand the CDB group and click a PDB name to view activity and investigate performance issues on that database instance.

Automatic annotations

Annotations are automatically added to wait time charts when a PDB is added, removed, or moved from one CDB to another. The annotations allow you to compare performance before and after the change.

Turn off automatic grouping of Oracle CDBs

If you do not want DPA to automatically group the PDBs within a CDB, you can turn automatic grouping off.

1. Click Options.
2. Under Administration > Configuration, click Advanced Options.
3. Click the ORACLE_CDB_AUTO_GROUP system option.
4. Select False from the New Value list, and click Update.

After you set this option to false, grouping of registered database instances does **not** change. Only newly registered or updated database instances are affected, and are not grouped.

Manually group database instances in DPA

DPA automatically groups Oracle Real Application Clusters (RAC) instances and Oracle multitenant container databases (CDB) containing pluggable databases (PDB). You can manually group other database instances so that they are displayed together on the DPA home page. For example, you can create groups based on type or location. When database instances are grouped, you can view information about all instances in the group.

A database instance can be included in only **one** group.

To move a database instance from one group to another, remove the instance from the group it is currently in. Then add it to the other group.

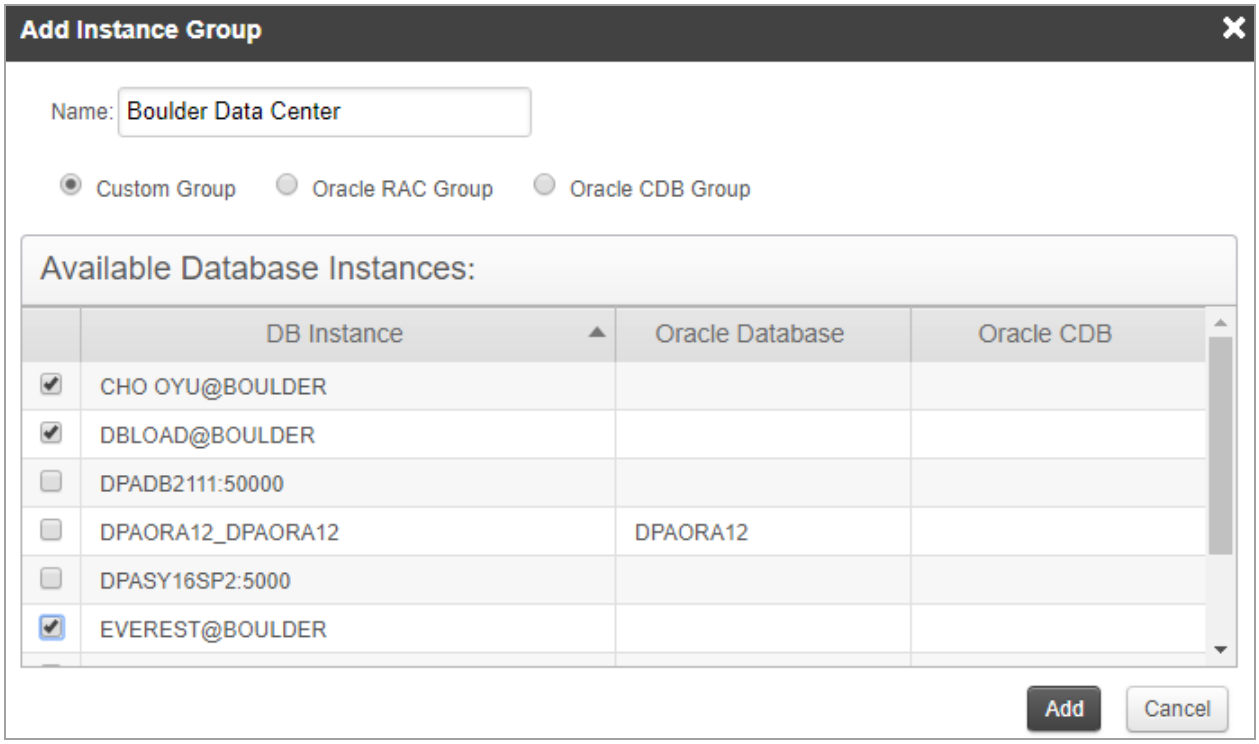
Create a custom group

1. On the DPA home page above the list of database instances, click Group Settings.

   The Manage Instance Groups dialog box lists the existing groups.
2. Click Add.

The Add Instance Group dialog box lists the database instances that are not members of an existing group.

3. Enter a name, select the database instances to include, and then click Add.



4. Click OK at the confirmation message.

Modify a group

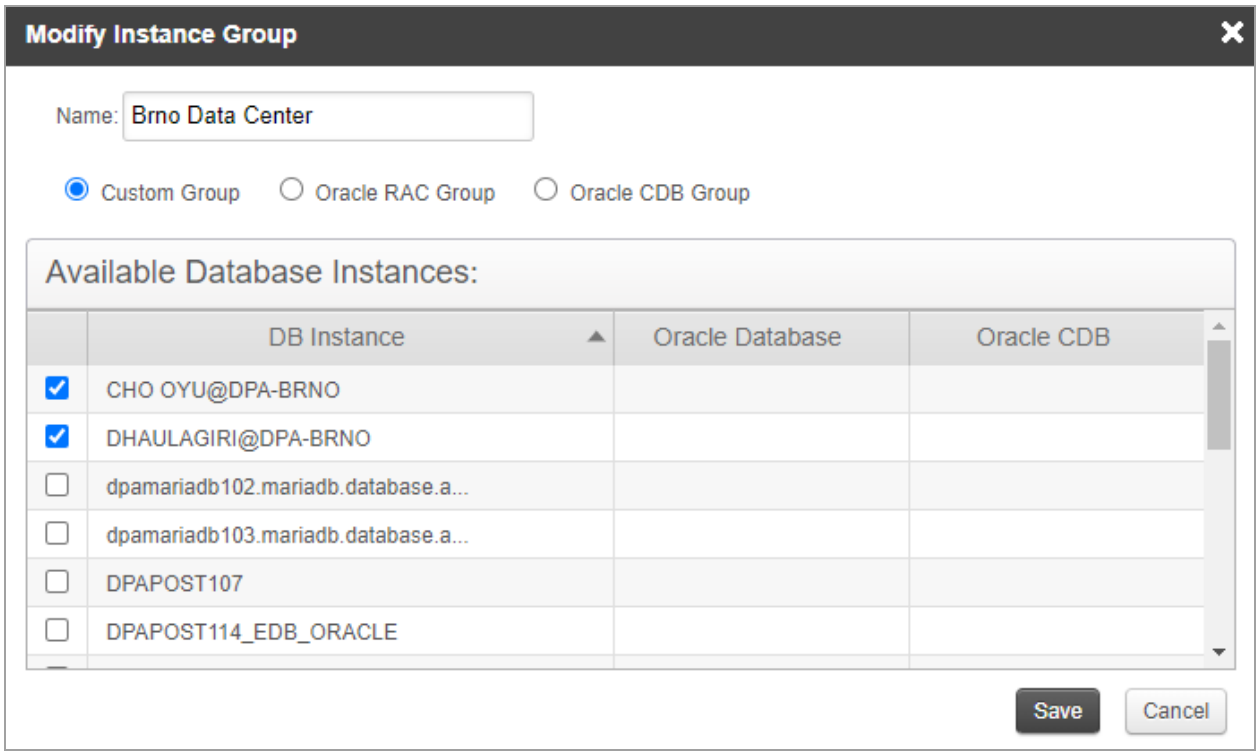
You can modify a group to change the name, add database instances, or remove database instances.

1. On the DPA home page above the list of database instances, click Group Settings.

   The Manage Instance Groups dialog box lists the existing groups.

2. Select the group that you want to modify, and click Modify.

   The Modify Instance Group dialog box lists the group's members (checked) as well as database instances that are not members of an existing group.



3. To change the group name, edit the text in the Name box.
4. To add an instance, select the instance.
5. To remove an instance, clear the checkbox next to the instance name.
6. Click Save.
7. Click OK at the confirmation message.

Delete a group

If you delete a group that contains database instances, the instances are ungrouped.

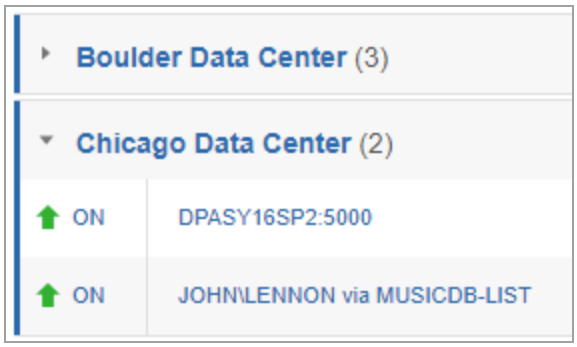
1. On the DPA home page above the list of database instances, click Group Settings.

   The Manage Instance Groups dialog box lists the existing groups.
2. Select the group you want to delete, and click Delete.
3. Click Yes at the confirmation message, and then click OK.

Show or hide groups on the DPA home page

Toggle the Show Groups button above the list of database instances to show or hide groups.

- When you show groups (the default), the DPA home page lists ungrouped database instances first, followed by groups in alphabetical order. You can expand or collapse each group. Click the group name to view information about all instances in the group.



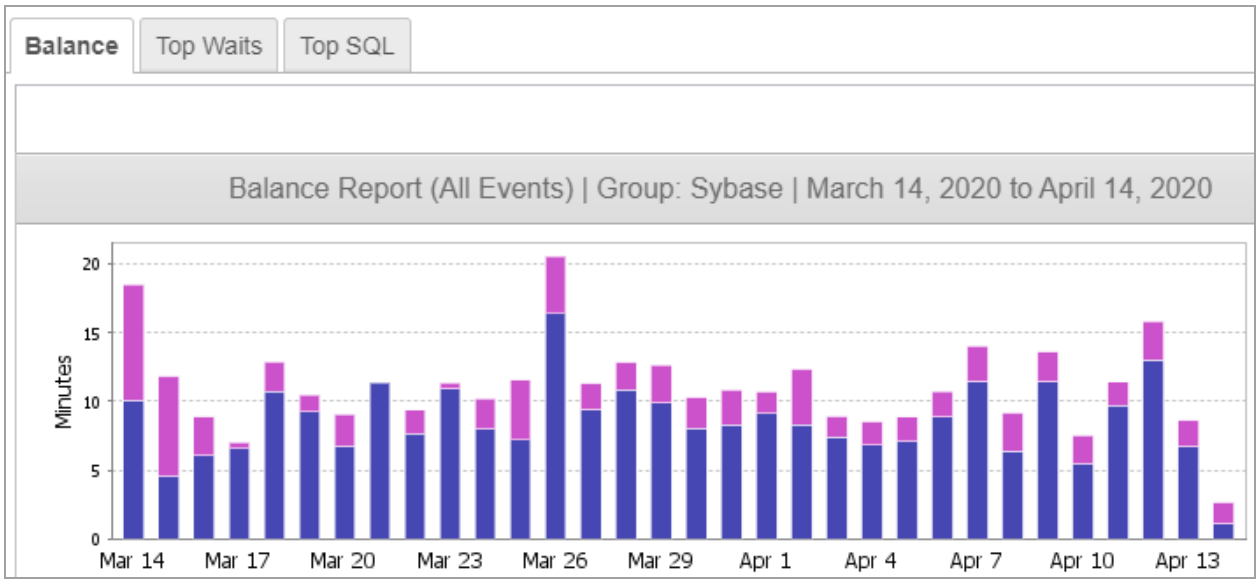
- When you hide groups, the DPA home page lists database instances alphabetically.

View information about a group of database instances

DPA automatically groups Oracle Real Application Clusters (RAC) instances and Oracle multitenant container databases (CDB) containing pluggable databases (PDB). You can also manually create groups of database instances. When databases are grouped, you can view information about how wait time is distributed throughout the group and top waits and top SQL statements for the entire group.

1. From the DPA home page, click the name of the group.

   The Balance Report bar graph shows the amount of wait time for each database instance in the group for the past month. Use this report to evaluate load distribution among the group members.



2. Click Top Waits to see the top 15 waits across all database instances in the group.
3. Click Top SQL to see the 15 SQL statements with the most wait time across all database instances in the group.

Manage connection information and monitoring

After you register database instances for monitoring, the DPA home page displays a list of the monitored database instances with status and wait time information. Click Action to start or stop monitoring.

Monitoring is always active after it is started. It is not necessary to restart the DPA monitor if the repository instance or the monitored database instance was unavailable for a period of time. Monitoring resumes when both are available again.

- If there is a period of time when monitoring should not occur, you can stop and then restart monitoring.
- If connection or user information changes with one of your monitored database instances, you can update that information.
- To monitor a virtual machine (VM), register the VMware ESX/ESXi Host or vCenter Server that the VM runs on.
- If necessary, update VMware connection information.
- If you are having problems connecting to or monitoring a database instance, see Troubleshooting tips.

For more information about using DPA to resolve issues on monitored instances, see Investigate performance issues with DPA.

Monitor VM performance data

When a database instance runs on a virtual machine (VM), you can monitor the VM in addition to the database instance. When you monitor a VM, you can view performance metrics from the VM, the physical host, and the storage system. DPA displays metrics from all these layers of infrastructure on a single screen. You can use this data to correlate events in the underlying layers with performance issues in the database instance.

To monitor a VM, you must register the VMware ESX/ESXi Host or vCenter Server that the VM runs on.

Monitoring a VM requires a VM license.

Set up a user on the VM host for monitoring

DPA requires a user with at least read-only permissions to monitor the VMs that run on a host (vCenter Server or ESX/ESXi Host).

First, create the DPA monitoring user:

- **vCenter Server User**: Authorized users for vCenter Server are those included in the Windows domain list referenced by vCenter Server or local Windows users on the vCenter Server system. To manipulate the user list or change user passwords, use the tools you use to manage your Windows domain or Active Directory.
- **Host User**: Log in to an ESX/ESXi host as root using the vSphere Client, then you can use the Users and Groups tab to add users, remove users, change passwords, set group membership, and configure permissions.

Give that user at least read-only permissions. SolarWinds recommends that you:

- Select the entire vCenter Server or ESX/ESXi Host and add the permission to that entity.
- When you add the permission, make sure the Propagate to Child Objects option is selected.

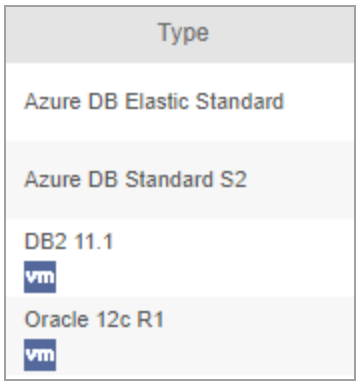
This approach ensures that any new objects inherit permissions and the DPA monitoring user can access them.

ⓘ You must have modify permission on an object to be able to assign permissions to that object.

Register a VM host

1. Near the top left of the DPA home page, click Register VMware for Monitoring.
2. Enter the connection credentials for a VMware ESX/ESXi Host or a vCenter Server.
3. Click Register.

   In most cases, DPA automatically detects the database instances running on the associated VMs. On the DPA home page, the VM icon is displayed in the Type column.



Manually link a database instance to a VM

If DPA does not automatically associate a database instance with the VM that it runs on, you can manually link it.

1. On the DPA home page, click the Action menu to the right of the database instance name.
2. Select Link to Virtual Machine.
3. On the Link Database Instance to Virtual Machine page, select the virtual machine and click Link.

   > By default, the Link Database Instance to Virtual Machine page lists a maximum of 250 VMs. If you have more than 250 VMs, edit the advanced Support option `LIMIT_OF_VM_ITEMS` to increase the limit.

4. Click Yes at the confirmation message, and then click OK.

   The database instance is listed under Running in a VM, and the VM icon is displayed in the Type column.

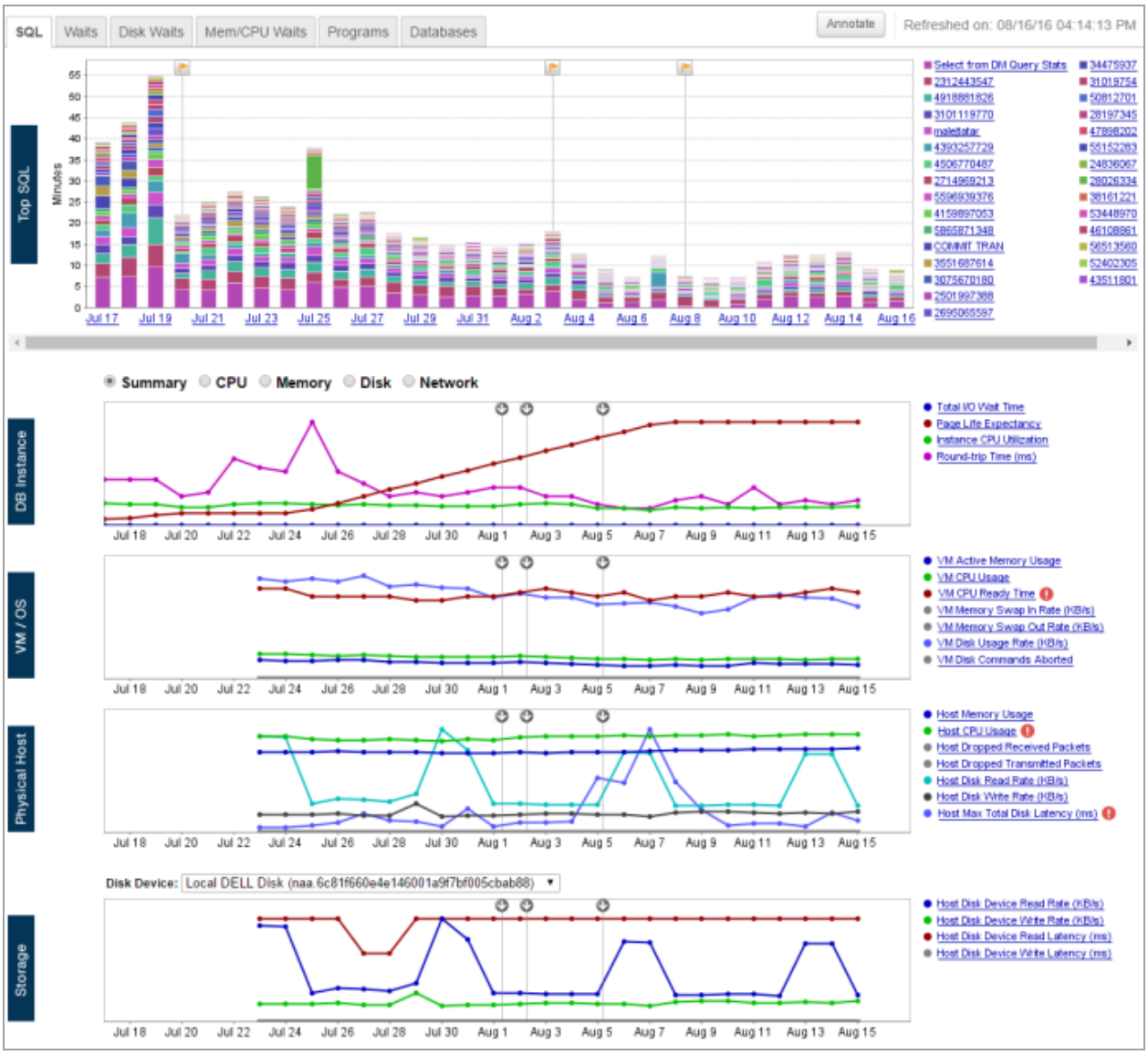
View VM performance data

After you register a VM host, you can view recent performance data immediately. DPA must gather data for a few days before it shows performance trends and baselines.

If the trial period is over, you must allocate a VM license to the database instance before you can view VM performance data.

1. At the top of the DPA home page, click Virtualization.
2. In the Database Instance list, click a database instance.

   DPA shows performance data from the database instance followed by metrics from the VM, the physical host, and the storage layer. Use these graphs to correlate waits in the database instance with events in the underlying layers.


SQL
Waits
Disk Waits
Mem/CPU Waits
Programs
Databases
Annotate
Refreshed on: 08/16/16 04:14:13 PM
Top SQL
Minutes
Summary
CPU
Memory
Disk
Network
DB Instance
Total I/O Wait Time
Page Life Expectancy
Instance CPU Utilization
Round-trip Time (ms)
VM / OS
VM Active Memory Usage
VM CPU Usage
VM CPU Ready Time
VM Memory Swap In Rate (KB/s)
VM Memory Swap Out Rate (KB/s)
VM Disk Usage Rate (KB/s)
VM Disk Commands Aborted
Physical Host
Host Memory Usage
Host CPU Usage
Host Dropped Received Packets
Host Dropped Transmitted Packets
Host Disk Read Rate (KB/s)
Host Disk Write Rate (KB/s)
Host Max Total Disk Latency (ms)
Disk Device:
Local DELL Disk (naa.6c81f660e4e146001a9f7bf005cbab88)
Storage
Host Disk Device Read Rate (KB/s)
Host Disk Device Write Rate (KB/s)
Host Disk Device Read Latency (ms)
Host Disk Device Write Latency (ms)

Update connection information for a monitored database instance

If connection information changes for one of your monitored database instances, you must update that information in DPA.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Database Instances, click Update Connection.
3. Select the database instance, and click Next.
4. Select the checkbox next to the property, update the value, and click Next.

   For database-specific connection information, see the following:
   - Oracle
   - SQL Server
   - Sybase
   - Db2
   - MySQL
   - PostgreSQL
   - Amazon RDS for Oracle
   - Amazon RDS for SQL Server
   - Amazon RDS for MySQL
   - Azure SQL Database
   - Azure SQL Managed Instance
   - Azure Database for MySQL or MariaDB
   - SQL Server running in the Google Cloud Platform
   - MySQL running in the Google Cloud Platform
5. Confirm the connection information, and click Update Connection.
6. Click Finish, or Update Another Database Instance to continue updating.

Update VMware connection information

If the connection information for a registered VMware ESX/ESXi Host or vCenter Server changes, you must update that information in DPA.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > VMware, click Update VM Connection.

3. Select the VMware ESX/ESXi Host or vCenter Server.

   The connection information is displayed.

4. Make the required changes, and click Update.
5. At the confirmation message, click Yes.

Stop monitoring a database instance for a period of time

A blackout is a period of time when DPA stops monitoring a certain database instance. DPA will not collect data or send alerts during this period.

> If you create a blackout period and then need to change it, delete it and create a new blackout period.

Create a new blackout period

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Database Instances, click Monitor Blackout Periods.
3. Select a database instance from the list on top.
4. Set a day and time to stop and restart monitoring.
5. Click Add New Blackout Period.

Delete a blackout period

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Database Instances, click Monitor Blackout Periods.
3. Select a database instance from the list on top.

   The existing blackout periods for that instance are listed.

4. Click Delete on the blackout period row.
5. Click Yes at the confirmation prompt.

DPA troubleshooting tips

Logs

DPA logs information about each monitored database instance. Use this information to help you determine why a database instance is not being monitored, or if data are missing.

Access log data through the DPA log viewer

Use the DPA Log Viewer to view log information for a specific database instance, or for all database instances and the DPA repository.

1. Open the Log Viewer:
    - To display log messages for a specific database instance:

      From the DPA home page, click Action > Log next to the database instance.
    - To display messages for all monitored database instances and the DPA repository:

      From the DPA menu in the upper-right corner, click Options. Then, under Support > Utilities, click Log Viewer.
2. Use any of the following options to locate information:
    - Use filters to help you find specific information. To change the filters, click Advanced and select the filter criteria. For example, you can filter by date range, a text string, or message level.
    - For any message above Info, click Details to view additional information from the log.
    - Click Log Files for Support to create a compressed file you can send to SolarWinds Support.

Open log files in a text editor

Log files are stored in the `DPA-install-dir/iwc/tomcat/logs/` directory on the DPA server.

Download log files to a DPA client

You can download some or all DPA log files from the Management Options page.

1. Log in to DPA as a user with administrative privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Support > Downloads, click one of the following options:

- To download all log files, click All Log Files (Zipped).
- To download the `idc.log` file, click Data Collection Log File.
- To download the `iwc.log` file, click Web Client Log File.
- To download the `wizard.log` file, click Wizard Log File.

Change the default ports

If the default ports of 8124 and 8127 are in use, change the ports that DPA uses.

SolarWinds recommends you do not change the default ports unless there is a conflict.

1. Open the following file in a text editor:

   DPA-install-dir`/iwc/tomcat/conf/server.xml`

2. Update one or both of the following lines with new port numbers:

   ```
   <Server port="8127" shutdown="SHUTDOWN">
                                  <Connector port="8124"/>
   ```

   You cannot use the same port for both of the lines.

3. Save the file and restart DPA.

When monitoring PostgreSQL databases with queries that return a large amount of data, DPA runs out of memory

The JDBC fetch size defines how many records are returned in one network call. If all data cannot be returned in one call because of the fetch size, multiple calls are issued.

By default, PostgreSQL does not set a fetch size, and attempts to return all data in one call. When a large amount of data needs to be sent to DPA, DPA can run out of memory. If this occurs, the `idc.log` file includes errors such as:

```
ERROR (2023-08-29T18:55:17,717-0400) [repositoryManager-thread-12] {name=MS-DB-PROD} PostgreSQLTextPollService:227 - Error getting data from pg_stat_statements.

org.springframework.dao.DataAccessResourceFailureException: PreparedStatementCallback; SQL [SELECT query FROM pg_stat_statements GROUP BY query ORDER BY SUM(total_time) DESC LIMIT ?]; Ran out of memory retrieving query results.; nested exception is org.postgresql.util.PSQLException: Ran out of memory retrieving query results.
```

Resolution: In new installations of DPA 2024.2 and later, the following property in the `system.properties` file sets a fetch size value for data that DPA retrieves:

```
com.confio.idc.database.service.statspoll.postgres.fetchsize=300
```

The default is 300, but this value is configurable. If you see errors such as the ones above:

- If the `system.properties` file on your DPA server does **not** include this property (for example, upgrading customers), add it.
- If the `system.properties` file includes the property but you still see the errors, decrease the value.

To add or change the property value:

1. Open the `system.properties` file in a text editor. This file is located in the following directory:

   DPA-install-dir`/iwc/tomcat/ignite_config/idc`

2. Add the following property, or decrease its value:

   ```
   com.confio.idc.database.service.statspoll.postgres.fetchsize=300
   ```

3. Save the file.
4. Restart DPA for the changes to take effect.

Connection attempts that use the HTTP connector port fail with a message that the DPA site can't be reached

For security reasons, HTTPS connections are required by DPA 2023.2 and later versions. Connection attempts that use the HTTP connector port are not accepted. To ensure that DPA is available to users who previously connected over HTTP, you can update the `server.xml` file to redirect traffic to the HTTP connector port (8123 by default) to the HTTPS/SSL connector port (8124 by default).

If the redirect is **not** added and users attempt to connect over HTTP, they will receive a message that the site can't be reached.

Resolution:

1. Open the following file in a text editor:

   DPA-install-dir`\iwc\tomcat\conf\server.xml`

2. Locate the `Connector` property below `<!--HTTPS/SSL connector>`, and note the `port` value. (By default, this is 8124.)

```
<!-- HTTPS/SSL connector -->
<!-- Note: AIX requires algorithm="ibmX509" attribute to be present in the connector -->
<Connector port="8124" maxHttpHeaderSize="20480" URIEncoding="UTF-8" scheme="https" secure="true" SSLEnabled="true"
           clientAuth="false" useServerCipherSuitesOrder="true" compression="on"
           compressionMinSize="2048" compressableMimeType=
           "text/html,text/xml,text/csv,text/css,application/javascript"
           keystoreType="PKCS12"
           keystoreFile="conf/.keystore"
           sslEnabledProtocols="TLSv1.2+TLSv1.3"
           ciphers="HIGH:!SHA1:!SHA256:!SHA384:!RSA:!DHE"
/>
```

3. Locate the `Connector` property below `<!--HTTP connector>`. Within the `Connector` property, add the following, where *`hpptsPortNumber`* is the `port` value noted in the previous step:

   `redirectPort="`*`hpptsPortNumber`*`"`

   For example:

   `redirectPort="8124"`

```
<!-- HTTP connector -->
<Connector port="8123" maxHttpHeaderSize="20480" URIEncoding="UTF-8" compression="on" redirectPort="8124"
           compressionMinSize="2048" compressableMimeType=
           "text/html,text/xml,text/csv,text/css,application/javascript"
/>
```

4. If you make these changes after the upgrade, restart DPA for the changes to take effect.

Access to a database instance

If DPA cannot access the server that hosts a database instance you want to monitor, make sure the port used for DPA connections is accessible. The port cannot be:

- Blocked by a firewall
- In use by another process

To determine if the port can be accessed, see Use PowerShell to test that a port is open on a server.

The port used for DPA connections is specified when you register the instance. If the instance is already registered, you can use the update wizard to view or update connection information.

Issues after the Oracle PDB that stores the repository is moved

If the DPA repository is created on an Oracle pluggable database (PDB), you might experience the following issues after the PDB is moved to a different container database (CDB).

DPA returns a connection error

The PBD moved to a CDB on a different server, and the connection string is incorrect.

Resolution: Update the connection string in the `repo.properties` file in the following location:

`DPA-install-dir``\iwc\tomcat\ignite_config\iwc\repo.properties`

DPA returns an invalid login error

Verify that the DPA monitoring user exists in the CDB. Common users (prefaced with C##) exist in only one CDB.

DPA does not start

Check the logs for error messages.

- If the logs do **not** have error messages, but you can't access DPA from a browser:
  - Make sure you do not have a firewall preventing DPA from listening on the ports that are specified in the following file:

    `DPA-install-dir/iwc/tomcat/conf/server.xml`
  - Make sure the client machine can ping the DPA server.
- If errors in the logs indicate that the default ports are in use, change the ports that DPA uses.

SSL

SSL is enabled by default. This includes an SSL connector in `DPA-install-dir/iwc/tomcat/conf/server.xml`, and a keystore file with a self-signed certificate in `DPA-install-dir/iwc/tomcat/conf/.keystore`.

On AIX, the following attribute must be added to the SSL connector in `DPA-install-dir/iwc/tomcat/conf/server.xml`:

```
algorithm="ibmX509"
```

For details on setting up SSL, see the knowledge base article Enable SSL for DPA.

Firefox browser

Logging out from a Firefox tab or browser will also log out of sessions you have open in other tabs or browsers. This is because Firefox shares the session cookie between browsers and tabs.

Investigate performance issues with DPA

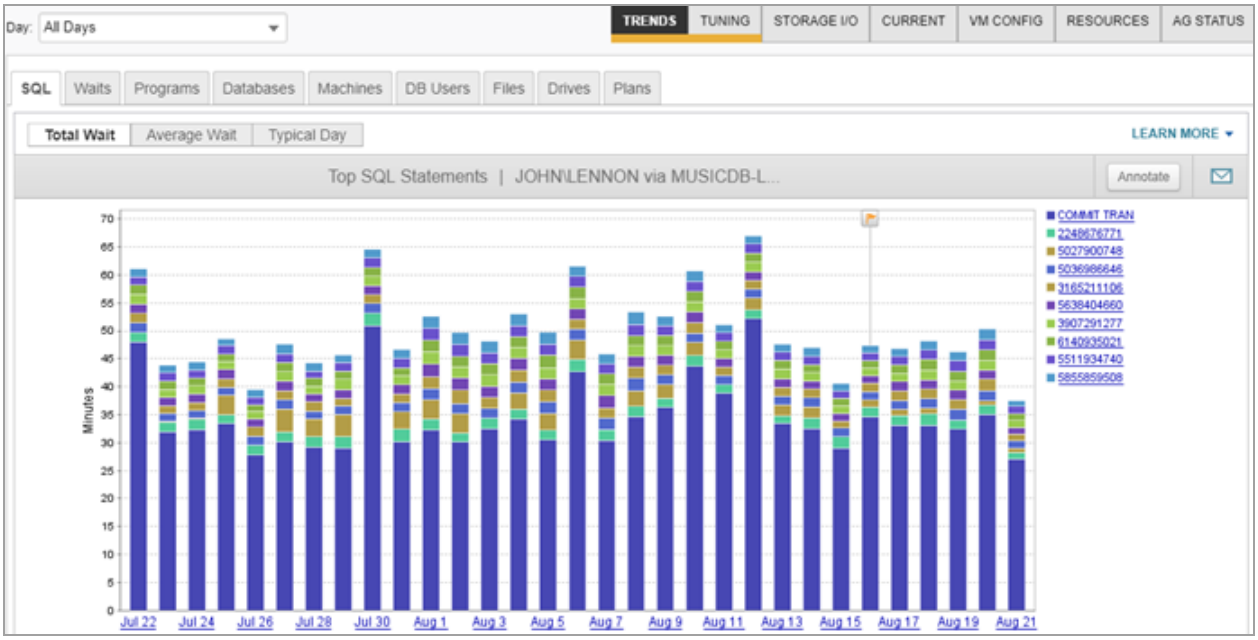
DPA uses an approach called wait-based analysis to help you focus on issues that provide the greatest performance improvements. Query advisors and table tuning advisors identify performance issues and help you find the root cause. Anomaly detection identifies queries with wait times significantly higher than normal.

The DPA Getting Started Guide includes walk-through examples of using DPA to investigate performance problems:

- Investigate an application performance problem
- Investigate an increase in wait time
- Investigate a wait time anomaly with DPA

Use the Trends charts to view data about wait times for an instance

From the DPA home page, click the name of a database instance to view information about that instance in the DPA Trends charts. The Total Wait chart on the SQL tab is displayed initially.



Examine SQL statements with the highest wait times

The Total Wait chart on the SQL tab identifies the SQL statements with the highest cumulative wait times during the selected time period. In many cases, these SQL statements have the highest performance impact on end users and applications.

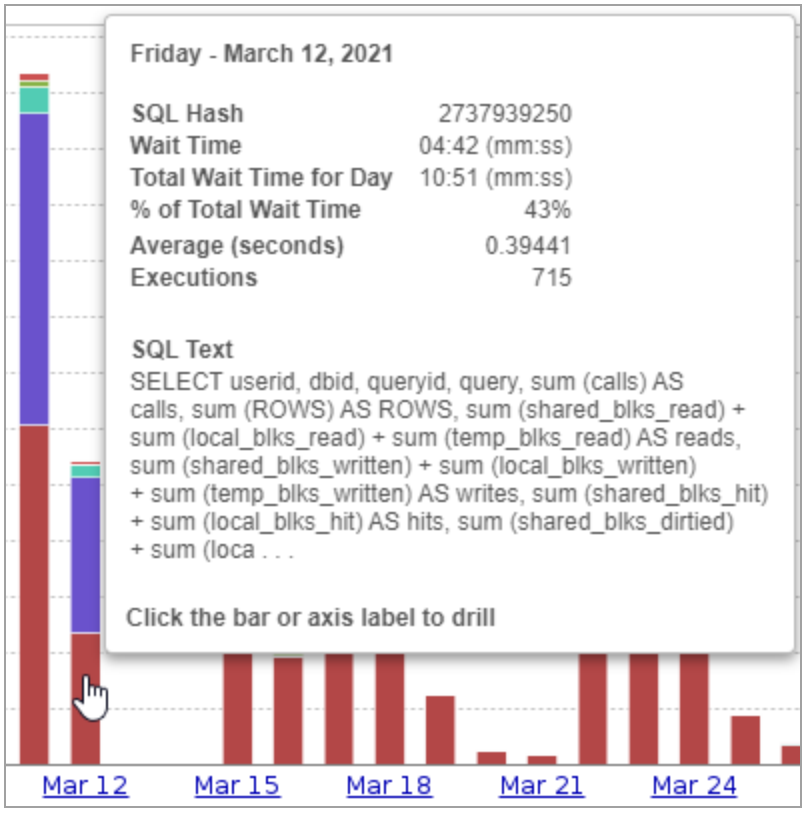
By default, the initial time period is the past 30 days of DPA data collection. To change the default for all Trends charts, edit the advanced Support option `TOP_INSTANCE_CHART_DAYS`.

- The vertical bars are a wait time summary for each day.
- The colored slices represent each SQL statement. The size of the slice represents the total time the SQL statement took to run, including all executions during the entire day. Bigger slices mean longer wait times for the users or applications that are waiting for the SQL statement to finish.

  To determine where to focus your tuning efforts, look for SQL statements that are consistently represented by large slices.

  Some long-running SQL statements (for example, maintenance jobs that run outside of business hours) are not candidates for tuning. The slices that represent these SQL statements can sometimes dominate the Trends charts. You can exclude these statements from the charts in order to focus on the waits that affect end users.

- Hover over a slice to see a summary of the SQL statement's performance that day.



- The legend on the right lists the SQL statements ordered by cumulative wait time. By default, the legend lists SQL statements by their hash values. You can name a SQL statement to make it easier to identify.
- A flag above a bar represents an annotation. Hover over or click the flag for more information.

  Annotations are shown on all Total Wait and Average Wait Trends charts.

Use this chart to identify SQL statements causing long waits that could potentially be tuned. When you identify a candidate for tuning, click the SQL statement name or hash in the legend to view in-depth analysis of that statement.

To walk through examples of using DPA to identify SQL statements for tuning, see Examples of investigating performance issues with DPA.

View the average wait times for long-running SQL statements

On the SQL tab, click Average Wait to see the average wait time for the longest running SQL statements.

- The vertical bars are a wait time summary for each day.
- The size of the colored slices represent the average wait time of each SQL statement.
- The legend on the right lists the SQL statements ordered by cumulative wait time (the same order as the Total Wait chart).

Use this chart to determine whether the performance of a SQL statement on a particular day was normal. Look for time periods where a query's average time was abnormally high, then drill down to see what was happening during that period to cause the anomaly.

To calculate the average wait, DPA takes the SQL statements with the most cumulative wait time over the latest 30 days of DPA data collection. It then divides the daily wait time of each SQL statement by its execution count.

See what time long-running SQL statements typically run

On the SQL tab, click Typical Day to see what time of day the top SQL statements typically run.

- The vertical bars represent the typical wait time for each hour of the day.
- The colored slices identify the SQL statements that typically run during each hour. The size of the slice represents the typical wait time for the SQL statement for all executions during that hour.
- The legend on the right lists the SQL statements ordered by cumulative wait time (the same order as the Total Wait chart).

Use this chart to determine if maintenance jobs are running during core hours and perhaps interfering with application query performance. If you see that SQL from maintenance jobs are bleeding into core business hours:

- Consider changing the schedule of the maintenance jobs.
- Consider tuning the SQL for maintenance jobs. Click the SQL hash value or name in the legend to open the Query Detail page and investigate the performance of the query.

You can also use this chart to understand usage patterns for the database. When you monitor your database instance with DPA at all times, you can see when critical applications are in use. Knowing this helps you determine when to schedule maintenance jobs to prevent them from affecting critical application performance.

For additional insight into what applications are running during each hour, click the Programs tab and select Typical Day. Typical Day is available for many of the DPA Trends charts.

To calculate the values on this chart, DPA takes the SQL statements with the most cumulative wait time over the latest 30 days of DPA data collection. For each hour of the day (for example, midnight to 1:00 AM or 1:00 AM to 2:00 AM), it sums the wait for each SQL statement that occurred during that hour (across all 30 days) and divides by 30.

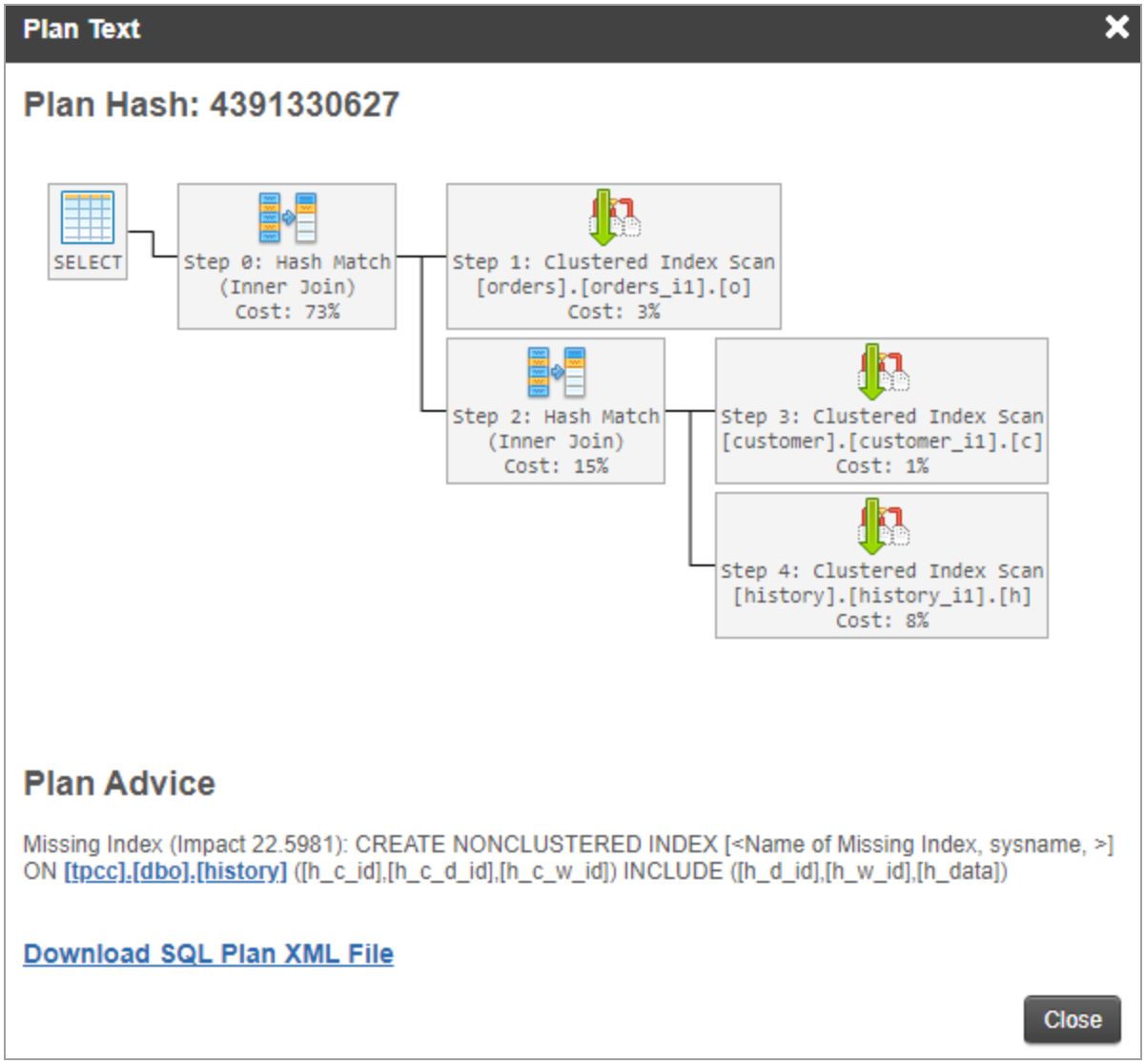
View the Trends charts for other dimensions

See wait time from different perspectives by clicking the other tabs. Use these tabs to get a complete picture of what the SQL statements are doing inside the database (for example, look at the Waits, Files, and Drives tabs) and who or what is generating SQL statements that are causing significant waits (for example, look at the Users, Programs, and Machines tabs).

The available Trends charts vary for different types of database instances. Examples include:

- The Waits chart shows what type of waits are affecting query performance. Knowing the type of wait causing a performance issue can help you fix the issue. Click a wait type name in the legend to display detailed information about that type of wait, including possible resolutions.
- The Programs charts shows which programs or applications generated the SQL statements that caused the longest waits. "No Program Name" means that the application did not set a program name in the database connection properties.
- The Databases chart shows how long users waited for data from each database. "No Database" means that the application did not specify a database context before issuing queries (that is, it did not connect to a specific database or issue a `USE` command).

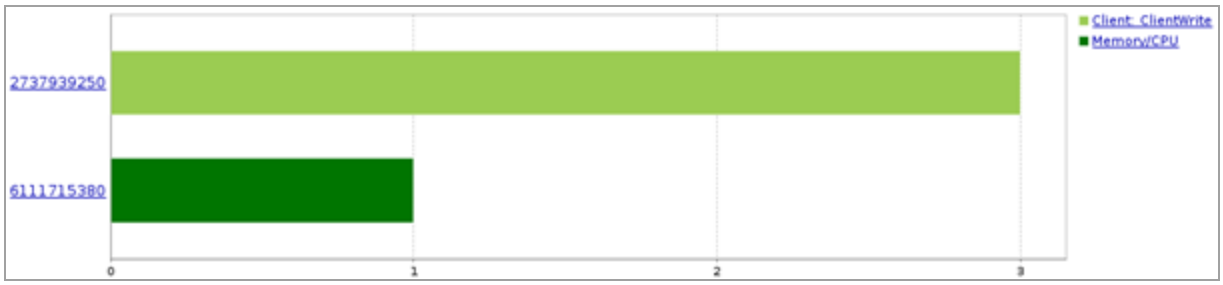
- The Machines chart shows which servers or workstations generated the SQL statements that caused the longest waits. In load-balanced environments, verify that the load is distributed correctly.
- The DB Users chart shows which users generated the SQL statements that caused the longest waits.
- The Plans tab shows the cached plans that had the most active wait time. Hover over a slice for details. Click a plan hash in the legend to display the plan text (and advice if available) from the database vendor. Examine the plan text to see which steps or operations have the highest cost. Click Download to download the plan.



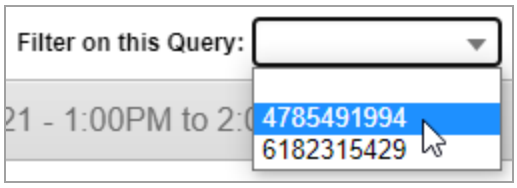
Drill in to a specific day or time period

If users complain about slow performance on a specific day or time, drill in to the Trends charts to find out more about what was happening during that period. On any Trends chart tab:

- From the 30-day chart, click a bar on the chart or a label on the x-axis to see the one-day chart. Each bar on the one-day chart represents a one-hour interval.
- From the one-day chart, click a bar on the chart to see the data for that hour. Each bar on the chart represents one entity, depending on the type of Trends chart. (For example, on the SQL chart, each bar represents one SQL statement. On the Programs chart, each bar represents one program.) The bars are color-coded by the wait type.



- From the one-hour chart, click the identifier to the left of a bar for more information about that entity. For example, on the SQL chart, click the SQL hash or name to view in-depth analysis of that statement. On the Waits chart, click the name of a wait type to see a list of SQL statements that ran during that period with that type of wait.
- When you drill down to a one-hour chart, the Timeslice tab becomes available. Click this tab to see the SQL statements that ran during each 10-minute interval within the hour. Use the Interval drop-down to change the length of the intervals.
- When you drill down to a one-hour chart, you can choose a SQL statement from the Filter on this Query drop-down to view SQL data for that SQL statement.



The available options depend on the type of monitored database instance. For example, on a MySQL or PostgreSQL instance, you can generate the execution plan.

View related data

Scroll down and click the Resources tab to view resource charts for the selected time period. For some database types, Blocking and Deadlocks tabs are also available. These charts can help you determine if resource contention, blocking, or deadlocks affected SQL performance.

Email a chart

When a chart illustrates a performance problem (or a tuning success), you can share it with non-DPA users by emailing it to them.

1. Click the email icon in the upper-right corner.
2. Enter the recipient, subject, and message.
3. Select the options, and click Send.

Access DPA query, table, or index advisors

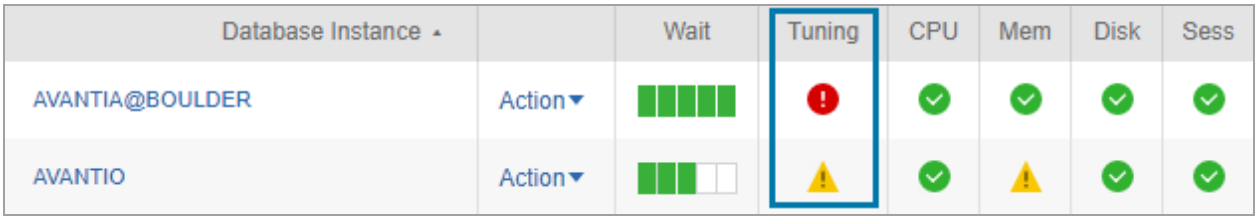
DPA provides the following types of advisors:

- **Query advisors** provide information to help you improve the performance of a specific query, including what type of waits were responsible for significant wait time, whether the statement was blocked by other sessions, whether execution plans include potentially expensive steps such as full table scans, and whether the execution plans for a query have changed.
- **Table advisors** are generated when inefficient queries run against a table. These advisors provide aggregated information about the table, the inefficient queries that ran against it, and any existing indexes.

  Table tuning advisors are available for Oracle, SQL Server, Azure SQL DB, and PostgreSQL database instances.
- **Index advisors** identify indexing opportunities and provide estimates of how much time adding the suggested index could potentially save.

  Index advisors are available for Oracle, SQL Server, Azure SQL DB, and PostgreSQL database instances.

The Tuning column on the DPA home page displays a warning or critical icon when advisors with a warning or critical status are available for a database instance. A green check mark in this column indicates that there are no advisors or that all advisors are informational.



You can specify which database instances DPA collects execution plans from.

View all advisors for a database instance

To view all advisors for a database instance, do either of the following to open the Tuning Advisors page:

- From the DPA home page, click the icon in the Tuning column.
- If you have drilled in to view information about a database instance, click the Tuning tab in the upper-right corner of the instance details page.

  A red or yellow bar on the Tuning tab indicates that critical or warning advisors are available.

The Tuning Advisors page displays the latest advisors:

- Query advisors are calculated every hour. The most recent query advisors are for the previous hour.
- Index and table tuning advisors are calculated once a day, at the end of the day. The most recent index and table advisors are for the previous day.

Use the drop-down menu at the top of the page to display advisors generated for a previous date.

View details

For detailed information to help you resolve performance issues:

- Click a row on the Index Advisors table to view details about the suggested index.
- Click a query advisor to open the Query Detail page, which displays detailed information about the query along with the most relevant statistics and metrics charts.
- Click a table advisor to open the Table Tuning Advisor page, which displays aggregated information about the table and the inefficient queries that ran against it.

View detailed information about a query

To help you investigate the root cause of a query's performance problems, DPA intelligently assembles the most relevant data about the query and displays it on the Query Details page. Use the Query Details page to:

- View waits, statistics, and metrics from any time period
- See what type of waits are affecting performance
- Review query and table tuning advisors
- Examine statistics and metrics charts to correlate query wait times with other events

See an example of using the Query Details page to investigate an increase in wait time.

Open the Query Details page

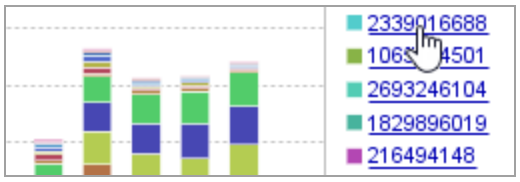
You can open the Query Details page in any of the following ways:

- From a query advisor.

  When DPA identifies a query with possible performance issues, it creates a query advisor. Open the Tuning Advisors page to view all advisors for a database instance. Click any query advisor to view information on the Query Details page.

- From a chart legend.

  When you are viewing data on a DPA Trends chart, click the SQL hash or name in the chart legend to view information on the Query Details page.



- From the Find SQL page.

  When you search for a SQL statement, click the blue arrow to the right of the Wait Time to display more information about a SQL statement in the search results. Then click the SQL hash or SQL name to view information on the Query Details page.

Select a time period

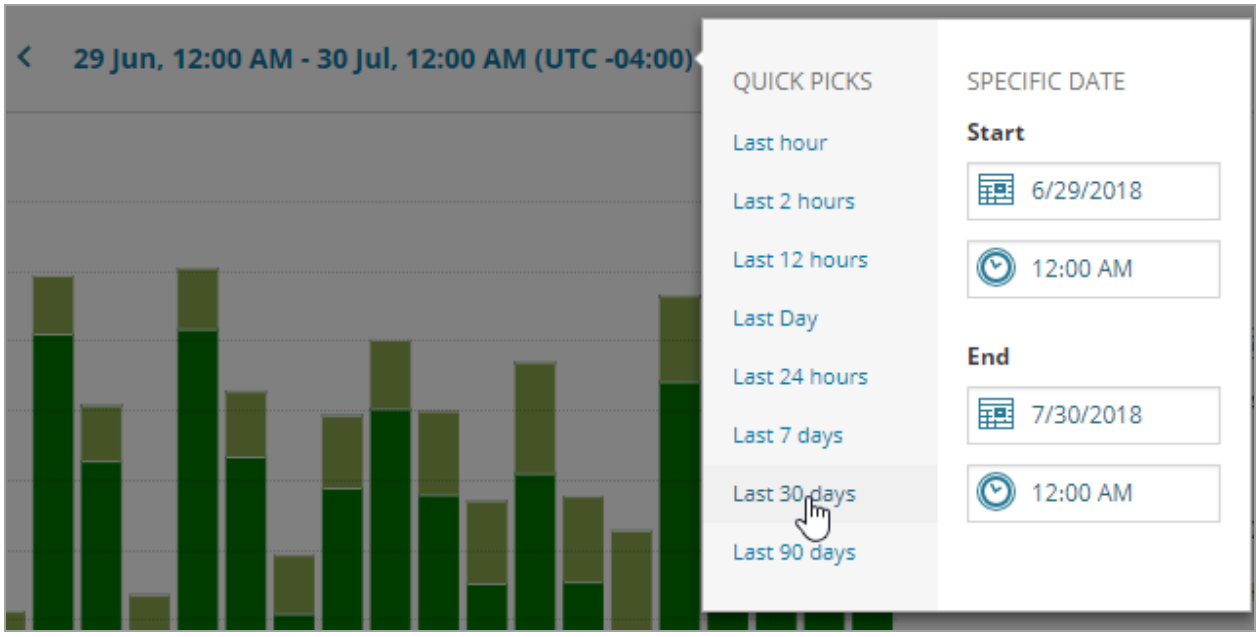
All data on the Query Details page reflects the selected time period, which is displayed at the top of the page.

When you open the Query Details page, it defaults to the time period selected for the previous chart. For example, if you open the Query Details page while viewing the Top SQL Statements for one day, the Query Details page shows data for that day.

To select a different time period, you can:

- Click a bar to drill in to that time period.
- Click the date range at the top of the page to open the date picker. Then select a predefined

time period, or enter specific dates.

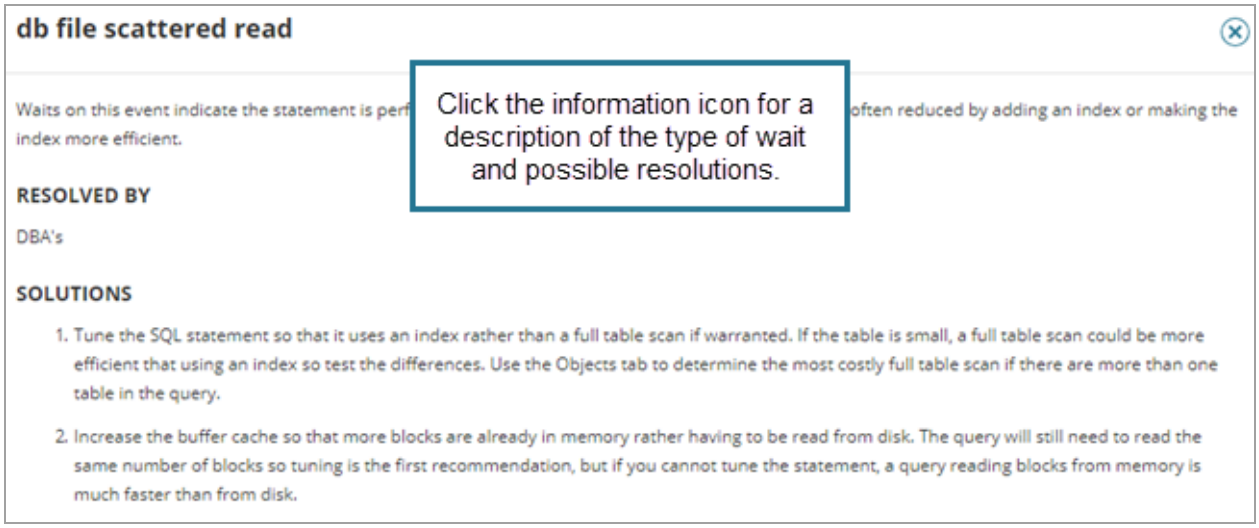


See what type of waits are affecting performance

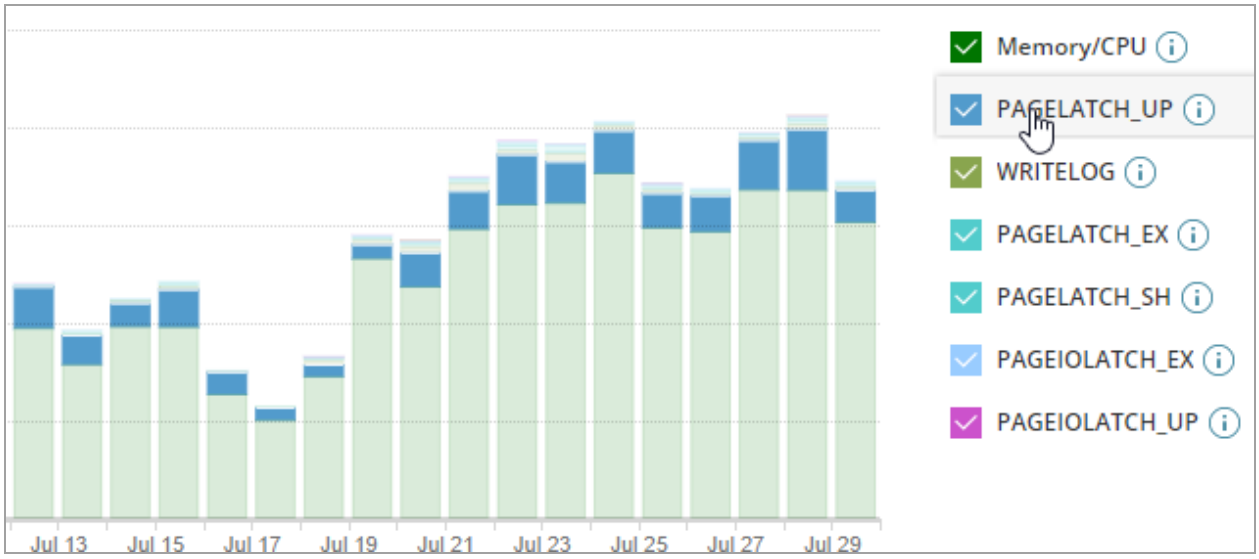
The Top Waits chart at the top of the page shows the query's execution time for the selected time period. The bars are color-coded by the type of wait. Knowing what type of waits are causing the performance issue can help you determine how to fix the issue.

On this chart, you can:

- Click the ⓘ next to an entry in the legend to display detailed information about that type of wait, including possible resolutions.



- Hover over an entry in the legend to dim other waits in the chart and better visualize the impact of this type of wait.



Review query and table tuning advisors

The Query Advisors section shows the latest advice for the selected time period. Query advisors provide information such as:

- What type of wait activities the SQL statement spend significant time on.
- Whether the statement was blocked by other sessions.
- Whether the statement took longer than normal to execute.
- If multiple execution plans were used, or if plans include potentially expensive steps such as full table scans.

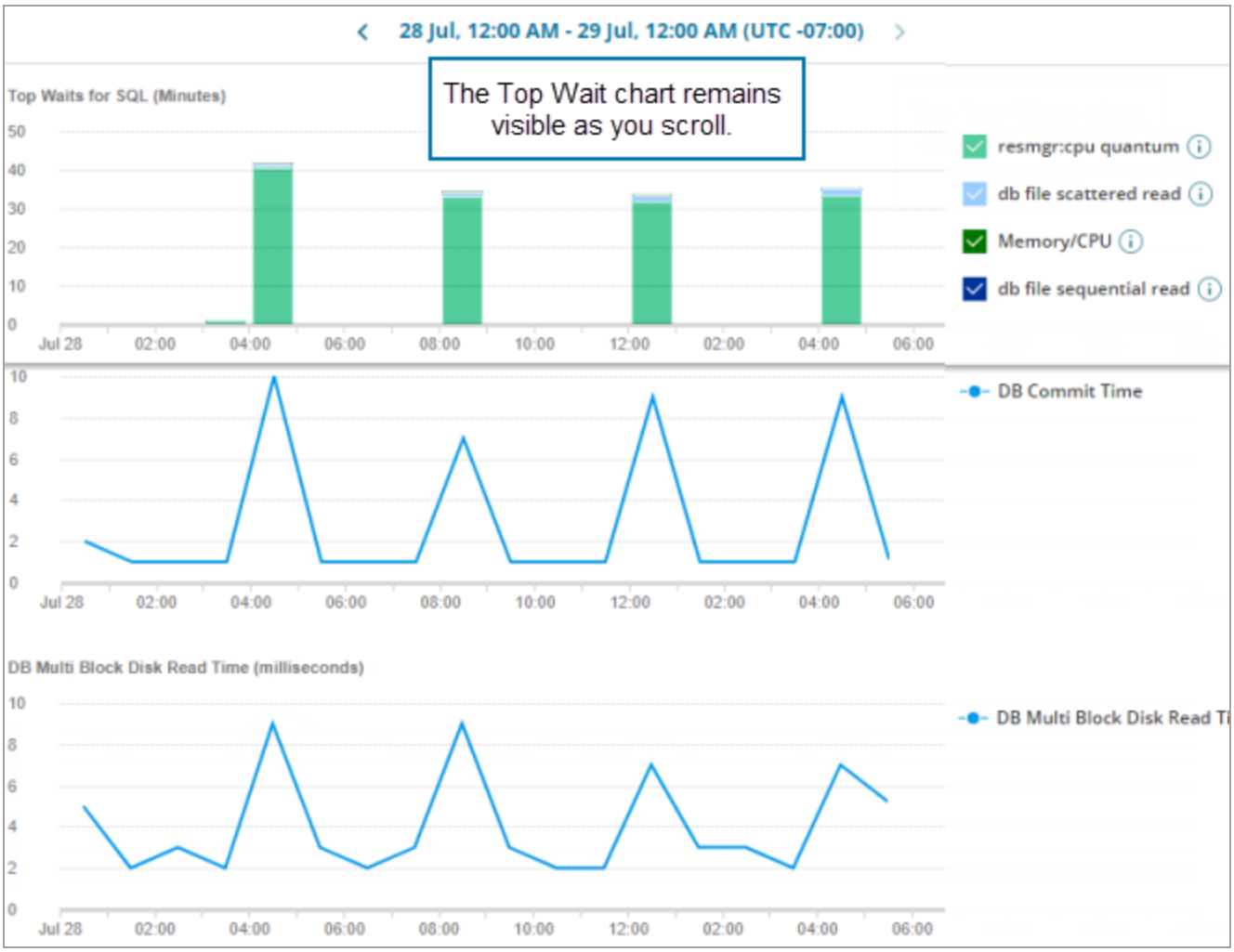
If any table tuning advisors included information about this query, you can click through for aggregated information about the table and all inefficient queries that ran on it.

Correlate query wait times with other events

To help you find the root cause of performance issues, the Query Details page includes the most relevant statistics, blocking, plan, and metrics charts. Sections with data to display are automatically expanded. Other sections are collapsed by default. For example, if there is no blocking data, the Blocking section is collapsed.

When you scroll down to view these charts, the Top Waits chart at the top of the page remains visible (by default) so you can correlate query wait times with other events during the same time period.

If you do not want the Top Waits chart to remain visible, click the pin in the top-right corner to unpin it.



DPA uses the predominant type of wait and other information to automatically select the most relevant charts. For example, if the predominant type for an Oracle database instance is Memory/CPU, DPA includes charts such as OS/CPU Utilization, CPU Utilization by DB, and Buffer Cache Hit Ratio.

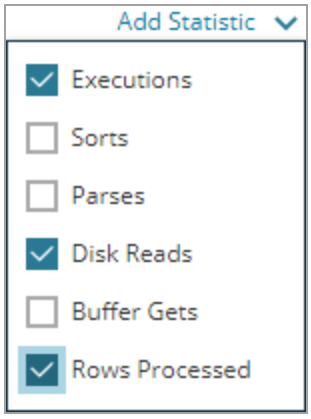
> The *predominant type of wait* is the type responsible for the majority of the time that a query spent waiting during the specified period.
>
> To be considered predominant, the type must be responsible for more than a certain percent of the total wait time for that period. By default, this threshold is 20%. You can change the threshold by changing the advanced option `PREDOMINANT_WAIT_THRESHOLD`.

You can manually select other statistics or metrics charts to include.

Display other statistics charts

1. If the Statistics section is collapsed, expand it.
2. On the right side of the Statistics section, click Add Statistic.

3. Select the statistics you want to include, and deselect any you want to remove.



4. Click outside the drop-down to close it.

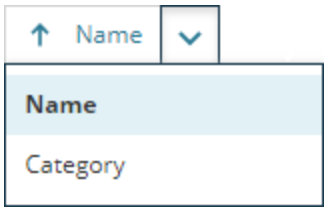
Display other metrics charts

1. If the Instance Resource Metrics section is collapsed, expand it.
2. On the right side of the section, click Add Metrics.

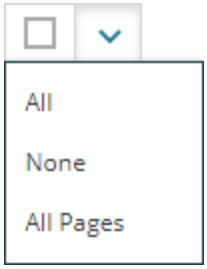
   The Add Metrics dialog box opens.
3. Filter or sort the list to locate the metrics you want to add:
   - Select one or more categories to filter by those categories.

   - Enter a string in the Search box to show only metric names containing that string. (Wildcards are not supported.)

   - Sort by name or by category.



4. Select one or more metrics. Use the selection drop-down menu to quickly select multiple metrics:
    - All: Selects all metrics on the current page.
    - All Pages: Select all metrics on all pages.



5. Click Save Changes to display the selected metrics.

Investigate inefficient queries running against a table

Inefficient queries—that is, queries that perform a large number of reads but return a relatively small number of rows—can lead to database performance issues. These queries do a large amount of work for little return. This type of inefficiency results in higher I/O, longer wait times, greater amounts of blocking, and increased resource contention.

Possible solutions include tuning the query, adding an index, or adding columns to an existing index. DPA's **table advisors** help you make informed decisions about the best course of action.

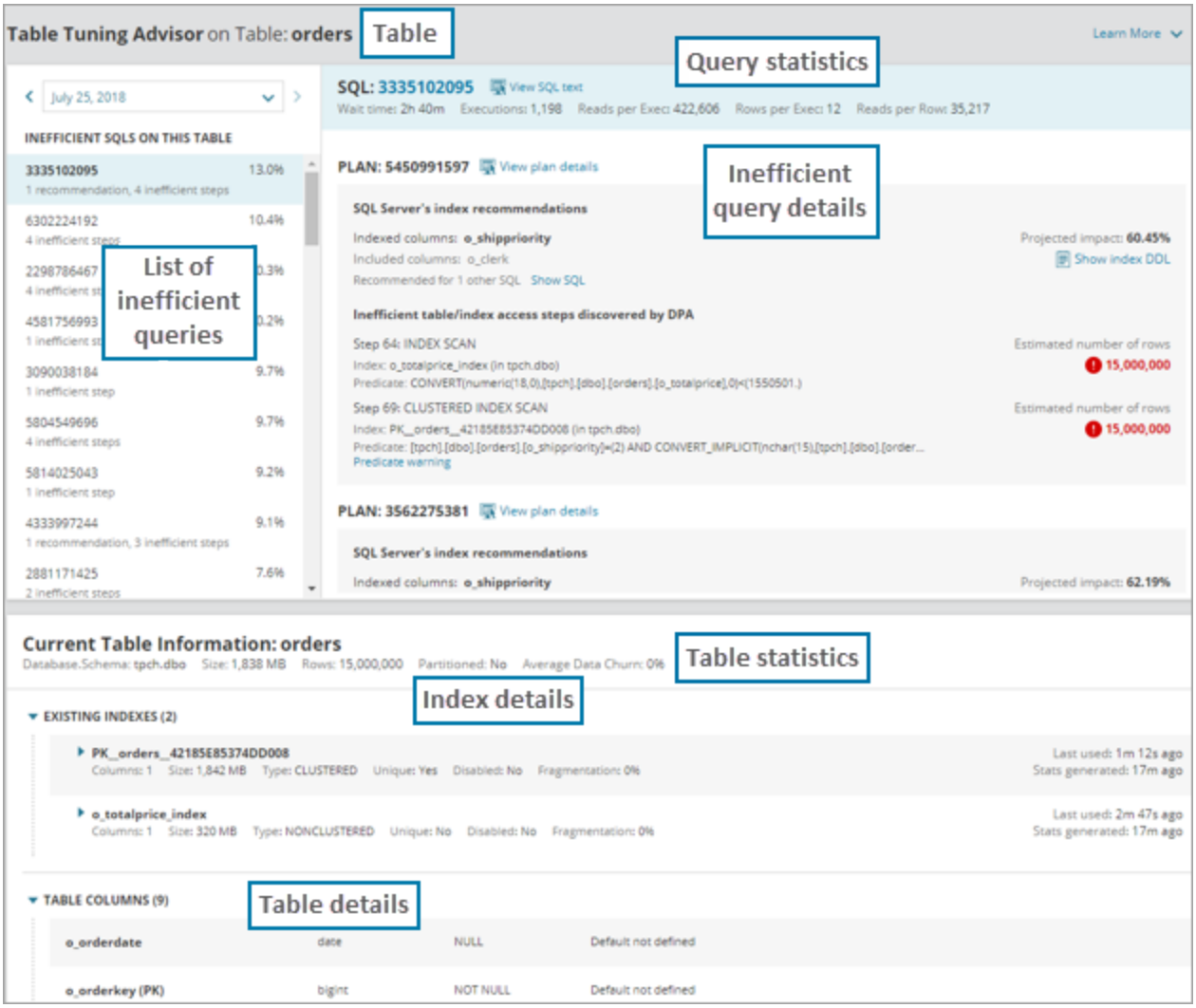
See the following sections for tips on using the information in each table advisor:

- What are table tuning advisors?
- Open a table tuning advisor
- Quick start
- Examine the list of inefficient queries
- Examine query details
- Examine table statistics
- Examine index details

What are table advisors?

At the end of each day, DPA runs an analysis to identify tables that had inefficient queries run against them during that day. For each of these tables, the Table Tuning Advisor page displays aggregated information about the table, the inefficient queries that ran against it, and any existing indexes. This information helps you optimize query performance while taking indexing trade-offs into account.

- Table advisors are available for Oracle, SQL Server, Azure SQL DB, and PostgreSQL database instances. The information DPA shows might vary slightly depending on the type of monitored instance.
- Table advisors are calculated at the end of each day. Therefore, the most recent table tuning advisors are for the previous day.



Open a table tuning advisor

The Tuning tab lists all table tuning advisors for the selected database instance. Click a table tuning advisor to open it.

Quick start

Each table tuning advisor provides detailed information, as described in the following sections. To get started, use these suggestions:

1. **Update statistics.**

   In the Existing Indexes section, look at the age of the index statistics. If the statistics are stale, especially if table churn is high, the optimizer does not have the best information to make good plan choices. Updating statistics is often a good first step before you do any further analysis.

2. **Evaluate indexes.**

   Click on several of the top inefficient queries and do the following:

   - Review the SQL text to learn more about the WHERE clauses and JOIN conditions that can affect query performance.
   - (SQL Server and Azure only) If plans with SQL Server's index recommendations are provided, consider adding them or extending existing indexes to satisfy them.
   - If plans with inefficient table or index access steps are provided:
     - Review each plan section and the predicates for each step. The columns in the predicates are candidates for indexes.
     - Check for warnings (shown as links below the step if they are detected) and consider their recommendations.
     - Consider indexing the candidate columns found across the SQL statements examined:
       - Is there an index that might benefit several queries?
       - Is there an existing index that could be extended to benefit one or more queries?

3. **Resolve fragmentation.**

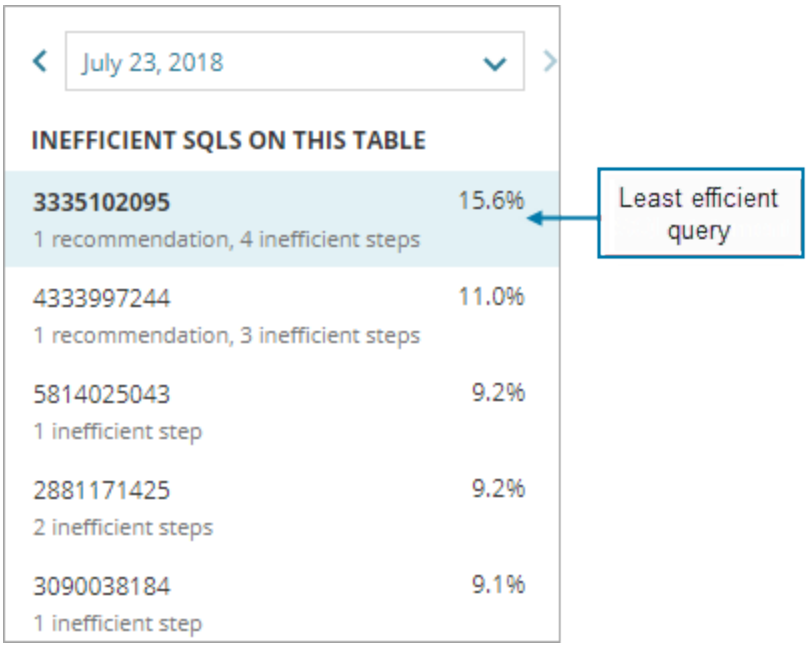
   Review the table's row count, churn, and index fragmentation. For larger tables, consider the following:

   - If fragmentation is high, defragmenting the indexes might help resolve performance problems when plan steps are using scan operations.
   - If churn is also high, consider defragmenting the index more frequently.

Examine the list of inefficient queries

The upper-left pane lists the inefficient queries that ran against the table on the selected day. DPA assigns a relative efficiency score to each query and uses this score to sort the list.

Select a query from this list to display detailed information about it.

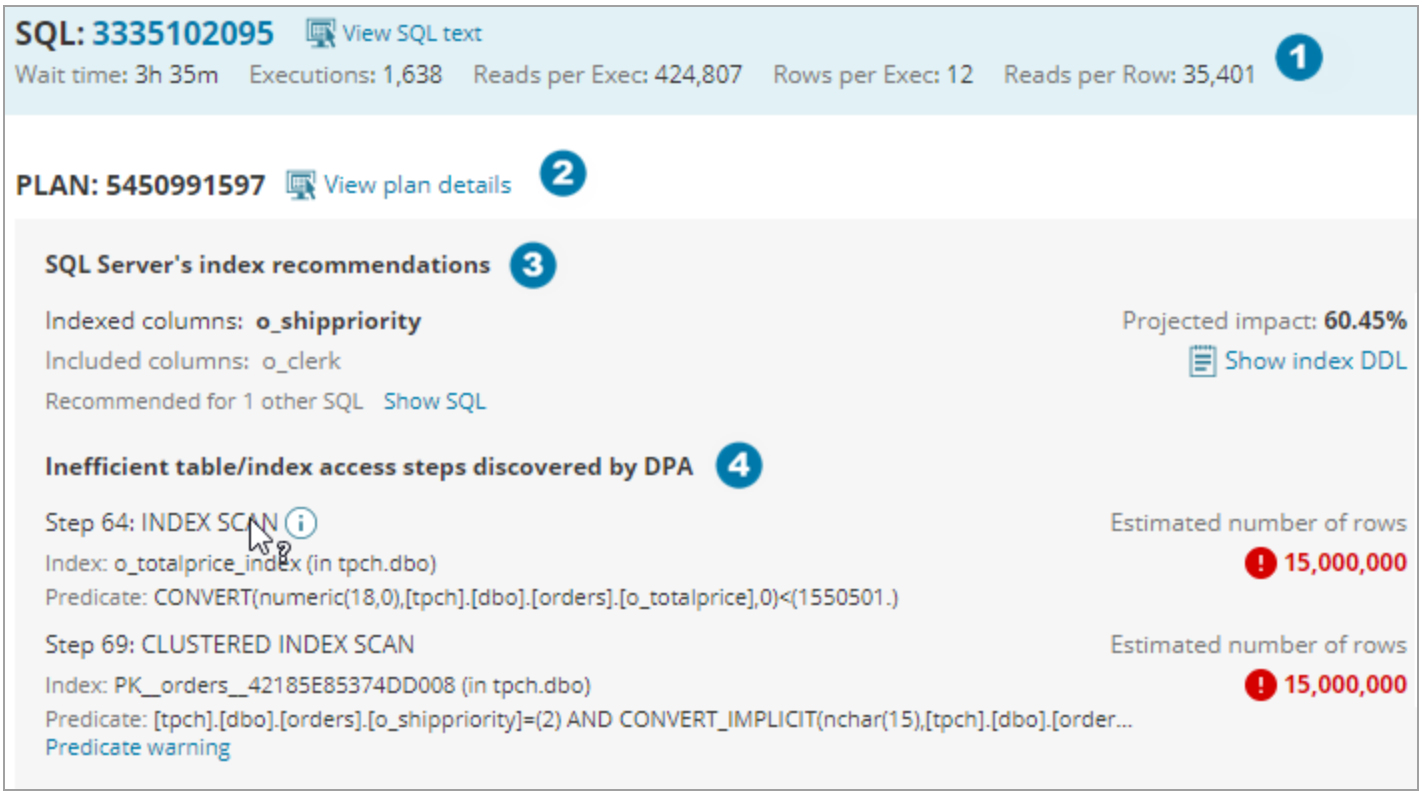


Tips for using this information

- Focus your tuning efforts on the queries at the top of the list, which are driving the most inefficient workload against this table.
- A large number of queries in the list could indicate a more widespread performance issue. Perhaps one good index could improve the performance of several similar queries.

Examine query details

The upper-right pane displays information about the selected query that can help you determine the source of read inefficiencies against this table.



1. The performance statistics at the top of the pane show the extent of the query's inefficiency:
   - Reads per Exec is the number of read I/O operations per execution, which indicates how much work the query is doing.
   - Rows per Exec is the number of rows the query returns.
   - The Reads per Row ratio is the number of reads the query needed to do to arrive at each row in the query's result set. Statements with the highest Reads per Row ratios could potentially benefit most from tuning.

   For more information about the query, click the SQL name or hash value to view DPA's query performance analysis, which shows when the query ran, the execution statistics, and the most relevant metrics charts.

2. DPA lists each execution plan that it finds. You can click the link to examine the full plan, but DPA lists the steps most likely to need attention below.

3. (SQL Server and Azure only) Index recommendations made by the SQL Server optimizer, if any, are listed. The Projected Impact is the cost reduction that the optimizer estimates the recommended index will have. Click Show index DDL to see the CREATE INDEX statement for the recommendation.

4. DPA analyzes the plan and lists steps with the most inefficient access paths.

   > These steps read data to be processed by subsequent "consumer" plan steps. While consumer steps (for example, sorts) can have a high plan cost, they are usually affected by a preceding step that read too much data.

   Information about each step includes:

   - The step number and the type of operation being performed in the step (for example, INDEX SCAN).
   - The index this step uses, if it uses an index.
   - Any predicates. These are snippets of the SQL that the plan step is acting on. They are typically portions of JOIN or WHERE clauses in which a table's column is being compared to another column or value.
   - Any warnings that apply to the step.
   - The number of rows the optimizer estimates this step will read. Critical and warning icons identify steps that read a high percentage of the table or index rows, and therefore have a greater need for tuning or an index.

Tips for using this information

- Before you add an index, weigh the projected impact or potential performance improvement against indexing trade-offs. Also consider the indexing needs of other queries.
- Click any step to get detailed information about the operation and recommendations for potentially reducing the amount of I/O.

- If predicates are listed, they often indicate which columns need to be indexed, or where the optimizer is not using an existing index. For example, if the query calls a function on the column, the plan will not use an index.
- If warnings are listed, click the warning for a detailed description of the condition that DPA has identified as a potential reason for concern.

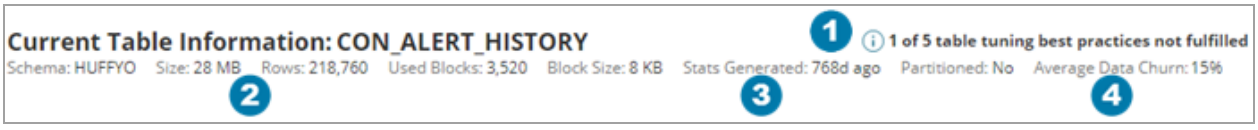
Warnings

DPA provides the following warnings:

- A **predicate warning** occurs when a column needs to be converted to a different data type before it can be used. For example, if a query has a JOIN clause that equates a numeric column to a varchar column, one of the columns will be implicitly converted to the other's type. The optimizer typically does not use an index on an implicitly converted column. This is often why the optimizer doesn't use an existing index that the query's author expected it to use.
- A **lookup warning** typically indicates that the database is doing an index lookup to identify the target rows, then doing an extra table access to get data not found in the indexed columns. To get better performance, consider adding a covering index, or extending an existing index to include columns needed to avoid the table lookup. However, remember that adding a large number of columns can increase the index size and maintenance overhead.
- A **spool warning** indicates that the step's result set is being stored for reuse later in the query's execution. While spool operations are often beneficial, the intermediate data storage can cause disk overhead and contention.
- A **parallel warning** indicates that DPA has detected a parallelism step later in this query's execution, implying that this step's intermediate result set is likely large enough to exceed parallel processing cost thresholds. Look for ways to rewrite the query to reduce the size of intermediate result sets earlier in the query. For example, look for a sub-select that could produce fewer rows or the nested loop join order if more than two tables are involved.

Examine table statistics

At the top of the Current table information section, DPA provides table statistics, such as the size of the table and the amount of churn.



Tips for using this information

1	**Best practices**: If the table or its indexes do not fulfill all of DPA's best practice recommendations, click the info icon to find out which recommendations are not met. Click here for information about correcting any violations.
2	**Size** and **Rows**: For large tables, indexing is often critical to good query performance, although an index on a large table uses large amounts of disk space. For small tables, full table scans sometimes offer better performance than the use of indexes.

3. **Stats Generated**: If the statistics are old and data churn is high, statistics should be updated frequently to provide the optimizer with the information it needs to make better plan decisions.

4. **Churn**: A table's churn is the daily number of insert and delete operation expressed as a percentage of the total number of table rows.

   Each insert and delete statement, as well as some update statements, incur a performance hit due to index maintenance. Generally, the higher the churn, the more caution you should take when adding an index. Before you add a new index, weigh the query execution time saved against the time spent on index maintenance.

Examine index details

DPA displays information about all existing indexes on the table, including the structure, the amount of fragmentation, how long ago the statistics were generated, and when the index was last used.

Where does DPA get the last used value for an index?

For SQL Server database instances, DPA shows when the index was last used for a seek, scan, or lookup operation, which is recorded in the sys.dm_db_index_usage_stats table. This value is not updated as a result of system activity.

For Oracle databases, DPA shows when the index was last included in an Oracle execution plan for a select, update, insert, or delete statement.

For PostgreSQL database instances, DPA shows when the index was last included in PostgreSQL execution plan for a select, update, insert, or delete statement.

Tips for using this information

Before you make any indexing decisions, first review the existing indexes. Consider the following questions.

ⓘ Take indexing trade-offs into account when you are considering adding or extending an index.

- Are the statistics stale? If the statistics are old and data churn is high, statistics should be updated frequently to provide the optimizer with the information it needs to make better plan decisions.

  If statistics are old and churn is high, consider updating the statistics before adding or modifying indexes.

- Is there an existing index that an inefficient query should be using?

  Look for ways to adjust the query so that it uses the index.

- If an inefficient query is using an existing index, are there inefficient table or index access steps on columns that aren't included in the index?

  Consider adding those columns to the existing index.

- If an inefficient query is using an existing index, are there inefficient table or index access steps that indicate a lookup warning?

  Consider adding those columns to the existing index to make it a covering index for the query.

- Is there no existing index that would improve an inefficient query's performance?

  Consider adding a new index.

- (Oracle and SQL Server) Are indexes fragmented? Fragmentation occurs as a result of numerous insert and delete statements. Fragmentation causes index data to become out of order on the disk, with gaps between index data. This is not a major concern for small tables, but for large tables this can cause slow performance when the index is read using a scan operation.

  Consider defragmenting your indexes on a regular basis for large tables, especially if data churn is high and many scans are occurring.

- (PostgreSQL) Is the Index Bloat Metrics percentage high? Bloated indexes can make inserts slower and negatively affect lookup performance.

  Consider rebuilding the index if the bloat percentage is high.

Correcting common index problems

After you determine what indexes are needed to improve query performance, look for additional benefits by identifying poor index usage, such as:

- **Unused indexes**: Can indexes be removed without negatively affecting query performance? To help you find unused indexes, DPA lists how long ago each index was used. However, before you remove an index:
  - Be aware that sometimes the Last Used value can show only the date since the monitored database instance was last started.
  - Consider whether queries that run infrequently (for example, monthly or quarterly) might use the index.

- **Too many indexes**: A large number of indexes on a table might be necessary for important queries to run quickly. However, you should also consider the performance overhead of index maintenance on other DML statements. Look for opportunities to:
    - Combine similar indexes.
    - Remove unused or rarely used indexes.
    - Remove indexes that were added for queries that are not performance sensitive.
- **Overlapping indexes**: Two indexes overlap if they both have the same leading edge columns in the exact same order, but one index has at least one additional column at the end. In this case, the larger index (with more columns) is all that you need, and you can remove the smaller, redundant index. Alternatively, you might choose to remove the larger one if the additional columns are not being used, or if the additional columns offer little benefit compared to the cost of index maintenance.
- **Questionable index structure**: The following might indicate a poorly constructed index:
    - Many columns: Indexes with many columns require more storage, and increase the cost of index maintenance. Perhaps the index was defined this way to make it a "covering index" for some queries. If not, consider removing trailing edge columns.
    - Wide columns: Some DBAs question the benefit of adding wide columns (for example, long varchars) to an index, because of the high amounts of storage needed for the index and the maintenance overhead. With this in mind, if your queries do a lot of searching on any column, consider indexing it.

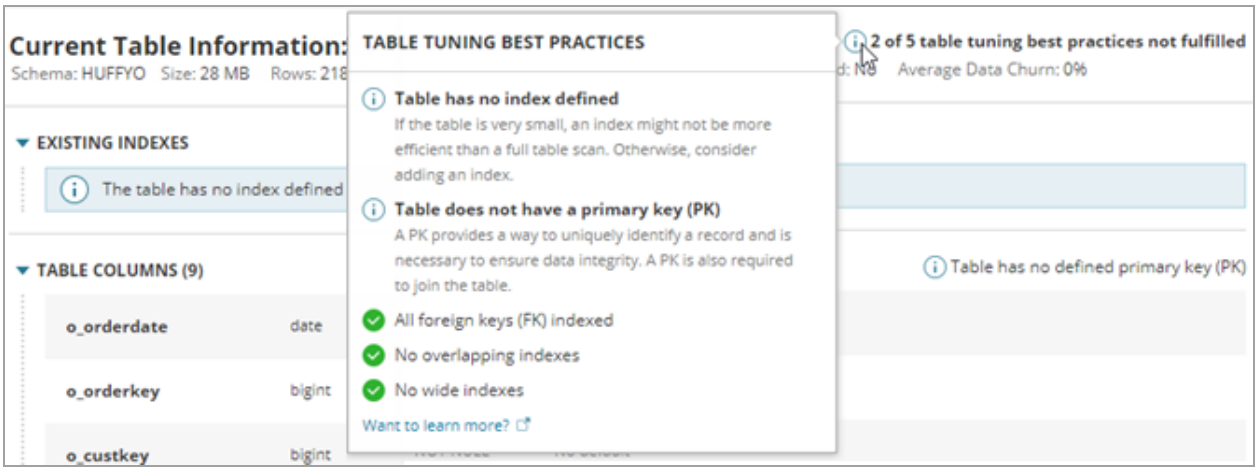
Indexing trade-offs

While indexes can provide performance benefits for some queries, consider the following trade-offs when making indexing decisions:

- **Index maintenance**: When a table row is inserted or deleted, the corresponding entry in each index must also be inserted or deleted. If an indexed column is updated, the associated entries in the index must also be updated. These operations on indexes increase the time an insert, delete, or update statement takes to run. The cost of index maintenance increases as the amount of data churn increases.
- **Disk space**: Indexes consume disk space. The larger the table and the more columns in the index, the more disk space it needs.

Investigate violations of table tuning best practices

When DPA generates a table tuning advisor, it evaluates the table and its indexes against a set of best practices. Any violations are listed in the Current Table Information section.



If violations are found, consider the following recommendations.

Edit the following advanced options to change the default values that DPA uses to check for best practices:

- To prevent DPA from checking for compliance to a best practice, change the corresponding `BEST_PRACTICES_<practiceName>` option to `false`.
- Use `BEST_PRACTICES_WIDE_INDEX_SIZE` to change the minimum size of a wide index.
- Use `BEST_PRACTICES_NUMBER_OF_COLUMNS_IN_WIDE_INDEX` to change the minimum number of columns in a wide index.
- Use `BEST_PRACTICES_NUMBER_OF_OVERLAPPING_COLUMNS` to change the minimum number of leading edge columns that indexes must share to be classified as overlapping.

Foreign key (FK) is not indexed

A foreign key in one table (the child table) refers to the primary key of another table (the parent table). Indexing each FK can improve the performance of queries that join the two tables. In addition, when FKs are not indexed, the database must perform a full table scan of the child table whenever a row is deleted or the primary key value is updated in the parent table.

Overlapping indexes found: At least two indexes have the same leading edge columns

Overlapping indexes have the same leading edge column (the first column defined). Because every index has a maintenance cost and consumes disk space, identifying and removing unneeded indexes can improve performance. Examine the overlapping indexes to determine if any can be removed. For example:

- If two indexes include the same columns in the same order but one includes additional columns, the smaller index is redundant and can be removed.
- If two indexes include the same columns but each has one additional column, modify one index to include all columns and remove the other index.

Wide index found: Index contains five or more columns or is more than 200 bytes

A wide index meets at least one of the following criteria:

- The index includes five or more columns.
- The index is more than 200 bytes (based on the amount of data that each column can hold).

Large indexes require more storage and increase the cost of index maintenance. If the index includes five or more columns because it is a covering index for multiple queries, the performance improvement might offset the additional overhead. However, if the index is **not** a covering index, the cost of maintaining the index could offset any performance improvement that the index provides. In this case, consider removing trailing edge columns.

To determine if an index is large enough to be classified as wide, DPA uses the size of the data that each column can hold, based on its datatype. For this reason, DPA might flag an index as wide even though the size shown in the Existing Indexes section is below 200 bytes. Including columns that are intended to hold large amounts of data is generally not a good indexing practice.

For example, an Oracle index includes a column with the `CLOB` datatype, which can store up to 4 GB. DPA identifies that index as wide, even though the size listed in the Existing Indexes section is 189 bytes.

Table has no defined indexes

The table is being queried, but no indexes exist. If the table is very small, an index might not be more efficient than a full table scan. For larger tables, consider adding an index.

Table does not have a primary key (PK)

A PK provides a way to uniquely identify a record and is necessary to ensure data integrity. A PK is also required to join the table. If no column or combination of columns provides a unique value, you can add an artificial PK such as an ID column.

View index recommendations

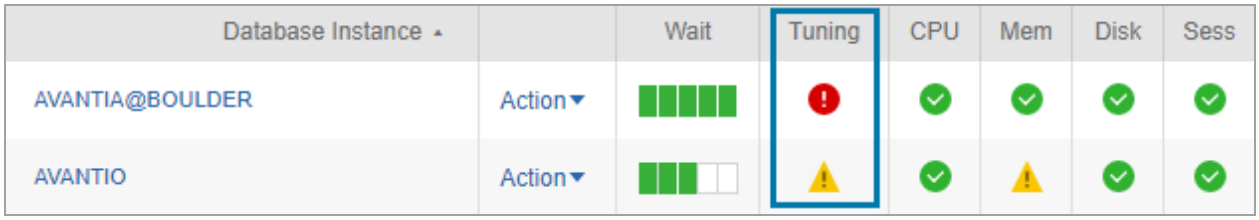
DPA identifies missing indexes that could improve the performance of specific queries. When DPA identifies a missing index, it creates a recommendation called an index advisor. Each index advisor includes information such as the index contents and estimated time savings.

> Index advisors are available for Oracle, SQL Server, Azure SQL DB, and PostgreSQL database instances. The information DPA shows might vary slightly depending on the type of monitored instance.

View the list of index advisors for a database instance

To view advisors for a database instance, open the Tuning Advisors page:

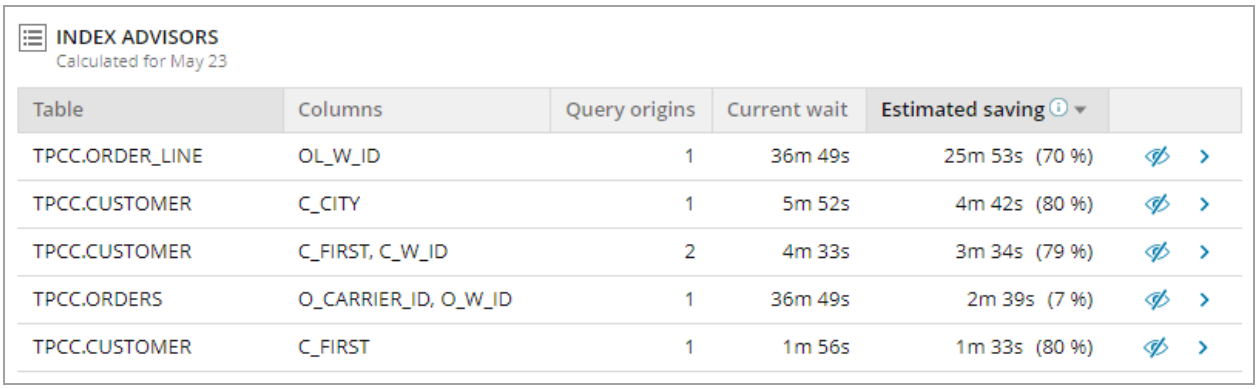
- From the DPA home page, click the icon in the Tuning column.



- If you have drilled in to view information about a database instance, click the Tuning tab in the upper-right corner of the instance details page.

The Index Advisors section lists any index advisors for the most recent time period. Select a date from the drop-down menu at the top of the page to view advisors for a different period.

> Index advisors are calculated once a day, at the end of the day. The most recent time period is the previous day.



For each index advisor, this section lists:

- The table and table columns to be indexed.
- The number of queries whose performance resulted in this index recommendation.
- The total wait time for all query executions during the selected time period.
- The estimated reduction in wait time as an amount and a percentage. This estimate is based on plan cost and step cost values.

Remove an index advisor from the list

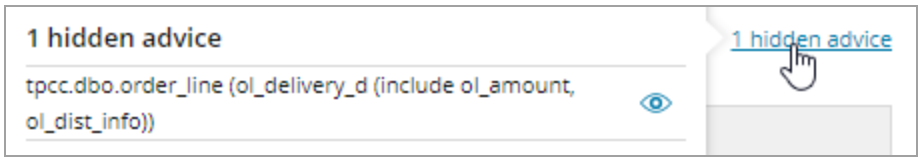
If you determine that a recommended index cannot or should not be created, you can dismiss the advisor to remove it from the list. The advisor is hidden from all DPA users across all dates.

1. Click the Hide this advice icon on the advisor row. The Dismiss advice dialog opens.
2. (Optional) Enter an explanation of why you want to remove the advisor.
3. Click Dismiss.

   In the upper-right corner, DPA shows the number of hidden advisors for this database instance.

Add a hidden advisor back to the list

1. Hover over the number of hidden advisors to display information.



2. Click the Show this advice icon on the advisor row.

   The advisor is added back to the list across all dates.

View index advisor details

Click the blue arrow in the right column to display more information about the recommendation:

- The Index DDL is the command to create the suggested index. To copy the command, click Copy to clipboard.
- The Query origins section provides information about the query or queries whose performance resulted in this index recommendation. For more information:

- To display all of the queries on the Find SQL page, click See affected queries.
- To display information about a query on the Query Details page, click the query name.
- To view the Oracle, SQL Server, or PostgreSQL execution plan for a query, click the plan hash number.

Advanced options for index advisors

The following advanced options are available to control how index advisors operate. Each option can be set globally or for a specific monitored instance.

Option	Description
INDEX_ADVISORS_BEST_PRACTICES_OVERRIDE	By default, index advisors do not recommend creating an index that is overlapping or wide, because these would violate table tuning best practices. To allow index advisors to recommend overlapping and wide indexes, set this option to True.
INDEX_ADVISORS_ENABLED	Index advisors are enabled by default. If you do not want DPA to generate index advisors, set this option to False.
INDEX_ADVISORS_WAIT_LIMIT	DPA does not generate index advisors for queries with negligible wait times (by default, less than 1 second). Edit this option if you want to change the default limit.

Configuration options and troubleshooting for PostgreSQL table and index advisors

See the following sections to configure DPA to generate table and index advisors.

Specify which PostgreSQL databases DPA collects plans from

By default, DPA collects plans from all databases. If you do not want to use plans from all databases, you can specify a list of databases to include or to exclude.

1. Set the advanced option POSTGRESQL_PLAN_COLLECTION_DB_NAMES to specify the list of databases you want to include or exclude.
2. Set the advanced option POSTGRESQL_PLAN_COLLECTION_DB_NAMES_FILTER to specify whether you want to include or exclude the list of databases.

These options can be set at both the global level and the monitored instance level. If they are set at the instance level, DPA considers both the global and instance-level values. If the values do not conflict, DPA applies all values. For example:

Global settings	Monitored instance settings	Result
Not Set	Include A, B	For monitored instances with the instance-level settings, DPA collects plans from A and B. For all other monitored instances, DPA collects plans from all databases.
Include A, B, C	Include D	For monitored instances with the instance-level settings, DPA collects plans from A, B, C, and D. For all other monitored instances, DPA collects plans from A, B, and C.
Exclude A, B, C	Exclude D, E, F	For monitored instances with the instance-level settings, DPA collects plans from all databases **except** A, B, C, D, E, and F. For all other monitored instances, DPA collects plans from all databases **except** A, B, and C.

When the global setting and the instance-level setting have different specifications for a specific database, the instance-level setting overrides the global setting for the specified instance. Otherwise, the global settings apply. For example:

Global setting	Monitored instance settings	Result
Include A, B, C	Exclude C	For monitored instances with the instance-level settings, DPA collects plans from A and B. For all other monitored instances, DPA collects plans from A, B, and C.
Exclude A, B, C	Include C	For monitored instances with the instance-level settings, DPA collects plans from all databases **except** A and B. For all other monitored instances, DPA collects plans from all databases **except** A, B, and C.

Troubleshooting PostgreSQL table and index advisors

Execution plans are missing for some SQL statements

To provide table and index advisors, DPA needs the execution plans that PostgreSQL generates. To determine if DPA is able to get the plan for a SQL statement, search for the SQL statement and get the SQL hash value. Then run the following query:

```
select * from CONPPT_ where sqlhash = 'SQLHash'
```

If no values are returned, check for the following issues:

- The DPA monitoring user does not have the required privileges on tables where the SQL statement runs.

  Check the `idc.log` file and look for the following message:

  `Failed to collect query plans due to: ERROR: permission denied for table` *`TableName`*`.`

  Resolution: To grant the required privileges, see the instructions for creating the monitoring user.

- The SQL statement is truncated in the `pg_stat_activity.query` field.

PostgreSQL stores currently running SQL statements in the `pg_stat_activity.query` field. If there is not enough memory allocated to store the entire SQL statement, it truncates the query. DPA cannot collect the execution plan for a truncated SQL statement. By default, PostgreSQL reserves 1024 bytes of memory.

Resolution: To avoid truncated queries, DPA recommends increasing the amount of reserved memory to 4096 bytes. To increase the reserved memory, modify the `track_activity_query_size` value as described in the instructions for configuring each database instance.

Index Bloat Metrics value is Unknown

On the Table Tuning Advisor page, the Index Bloat Metrics value is Unknown if the `pgstattuple` extension is not enabled on the database.

To enable the `pgstattuple` extension, connect to the database and run the following command:

```
CREATE EXTENSION pgstattuple;
```

Identify blocking sessions and deadlocks with DPA

DPA provides information to help you determine if blocking sessions and deadlocks are affecting performance, and to investigate the root cause of these issues. See the following sections:

- Identify blockers causing the longest waits
- Find the last activity of an idle blocker
- Investigate deadlocks on SQL Server instances

Identify blockers causing the longest waits

Are blocking sessions causing performance problems in your environment? Use the Blocking tab to identify the root blockers, find out which SQL statements are being blocked, and determine which blocking sessions are responsible for the longest overall waits. DPA shows the aggregated wait time for each blocker, which helps you focus your tuning efforts on blockers with the largest impact.

To view information about blocking sessions:

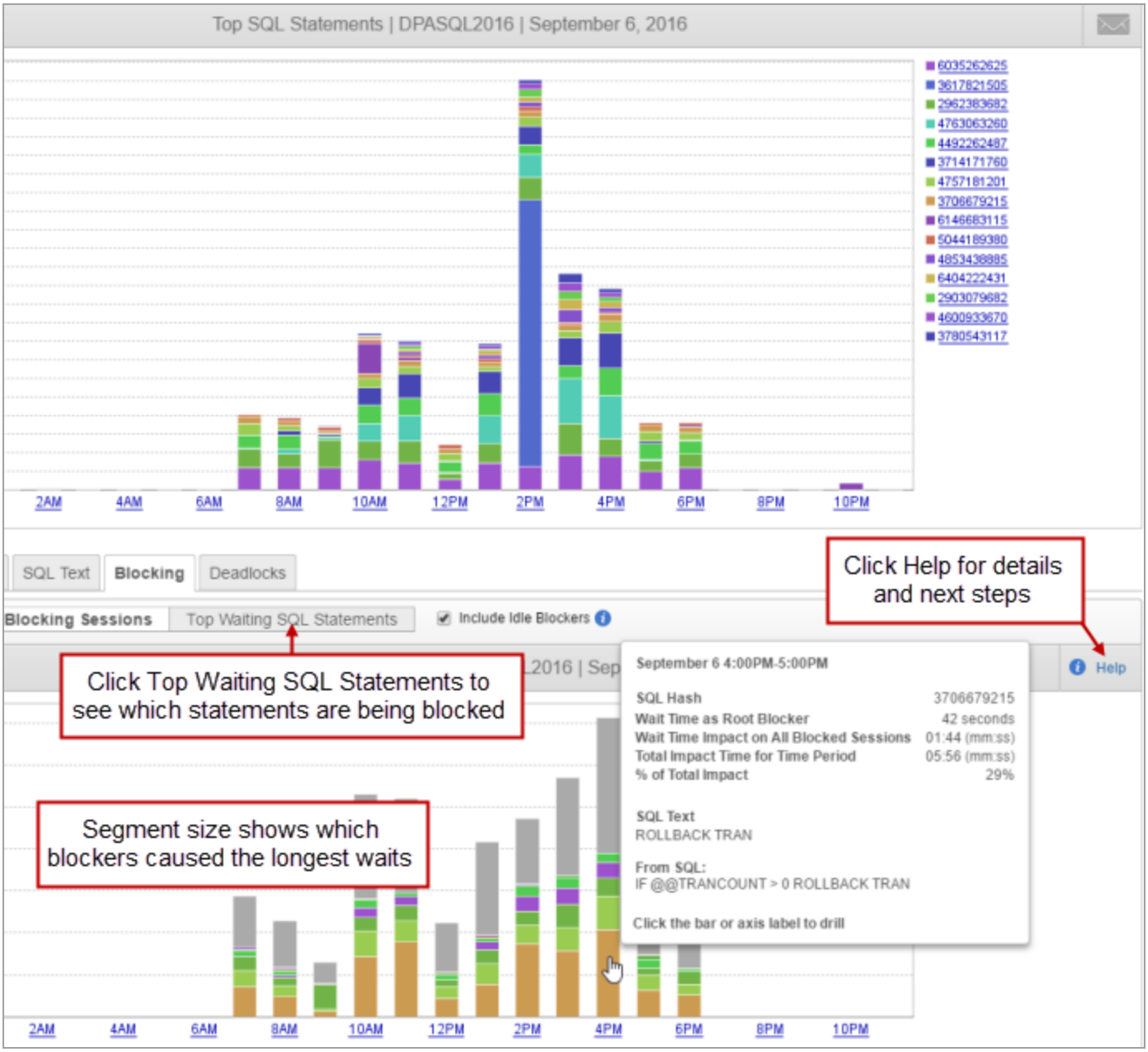
1. From the DPA home page, click a database instance name to display the Trends charts.
2. If necessary, click a bar on the chart to drill down to the time period you're interested in.

   By default, the initial time period is the past 30 days of DPA data collection. You can drill in to a Trends chart to view a more specific period.

3. Click the Blocking tab below any trends chart to view correlated information about blockers

during that time period.

The size of each segment in a bar provides a visual indicator of the waits that session caused.

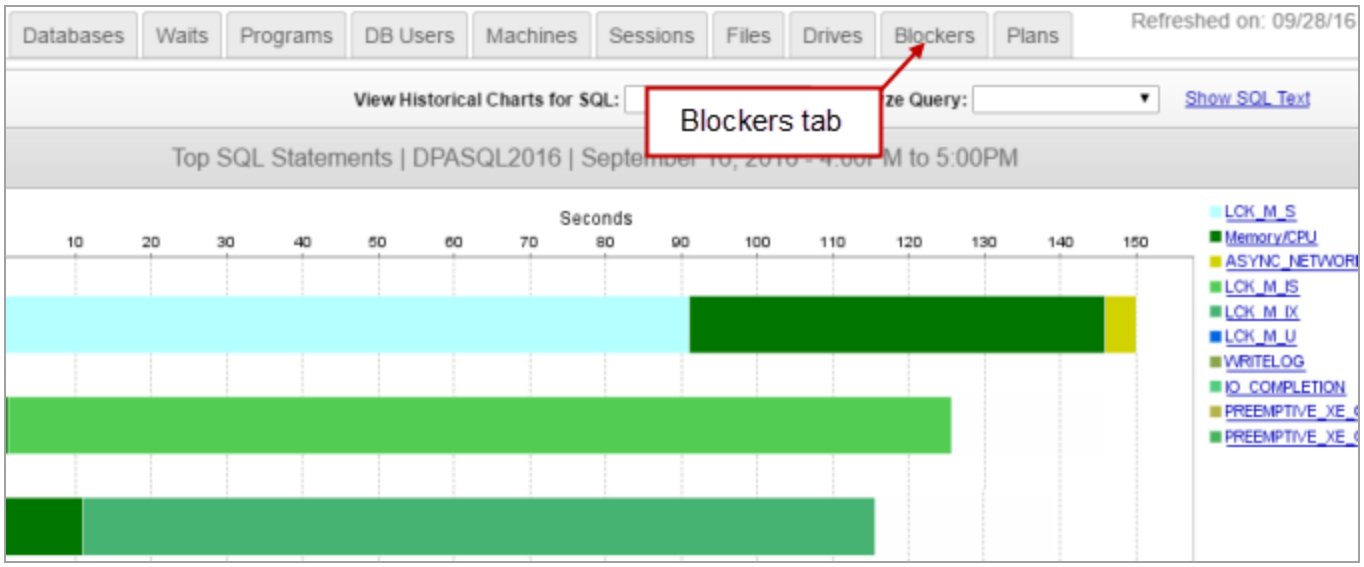


Find the last activity of an idle blocker

Idle blockers can be difficult to diagnose because they are currently not performing any activity in the database. To help you find and fix the root problem, use DPA to determine what that session was doing before it became idle.

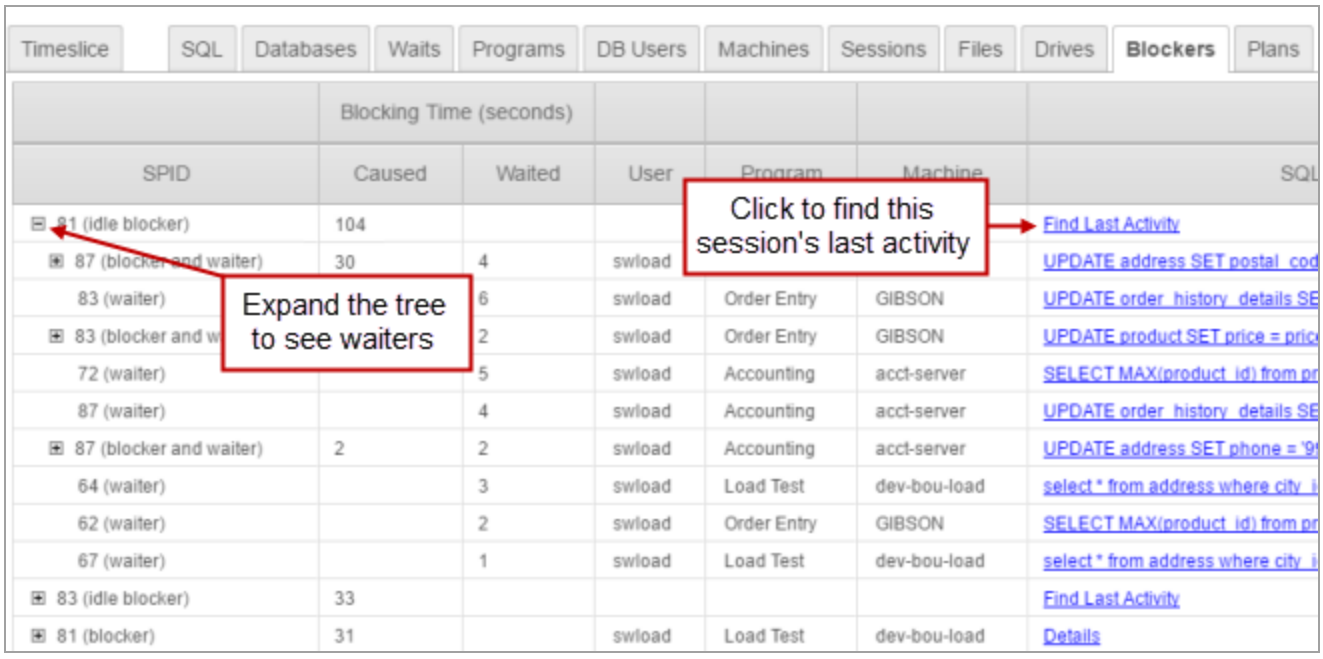
1. From the DPA home page, click a database instance name to display the Trends charts.
2. Click a bar on the Top SQL Statements chart to drill into a day, and then click a bar to drill into an hour.

   DPA displays information about the type of waits experienced during that hour.



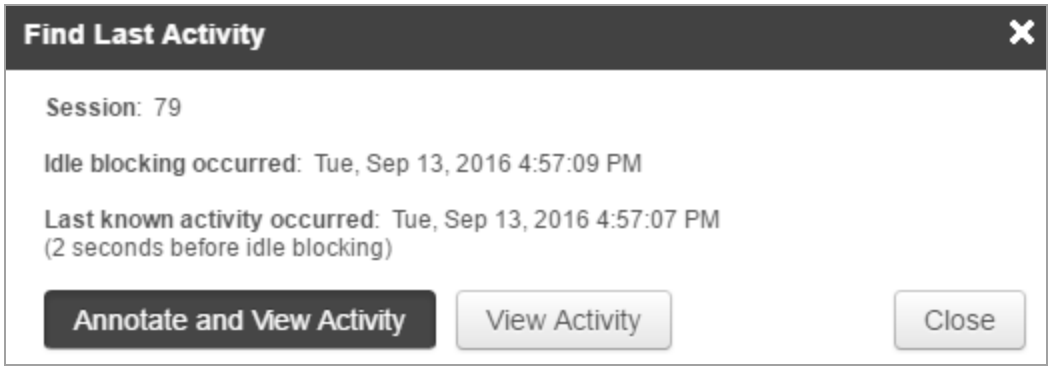
3. Click the Blockers tab above the chart to see a list of the blockers for that time period.
4. Expand a blocker to see information about the waits it caused.

   Each idle blocker row has a Find Last Activity link on the right.



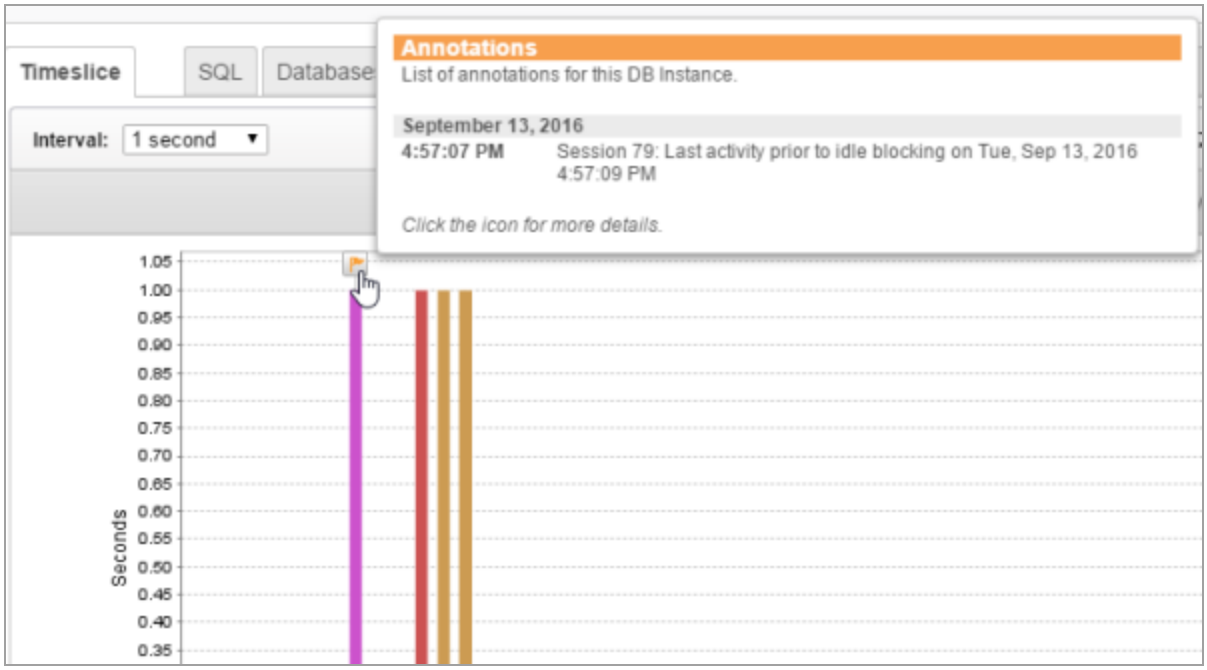
5. To find out what a blocking session was doing before it went idle, click Find Last Activity.

The Find Last Activity dialog tells you when the last activity occurred.



6. Click one of the following buttons:
   - Annotate and View Activity: Displays the activity and annotates the SQL statement to make it easy to find in the future.
   - View Activity: Displays the activity without annotating the SQL statement.

   The Timeslice tab shows a bar representing the last SQL statement executed by the idle blocker. You can drill in to investigate further.



Investigate deadlocks on SQL Server instances

Deadlocks occur when two sessions have a lock on different resources, and each session needs the resource of the other to complete its task. For example:

- Session 1 has a lock on Table A.
- Session 2 has a lock on Table B.

- Session 1 requests a lock on Table B, but it is blocked by Session 2.
- Session 2 requests a lock on Table A, but it is blocked by Session 1.

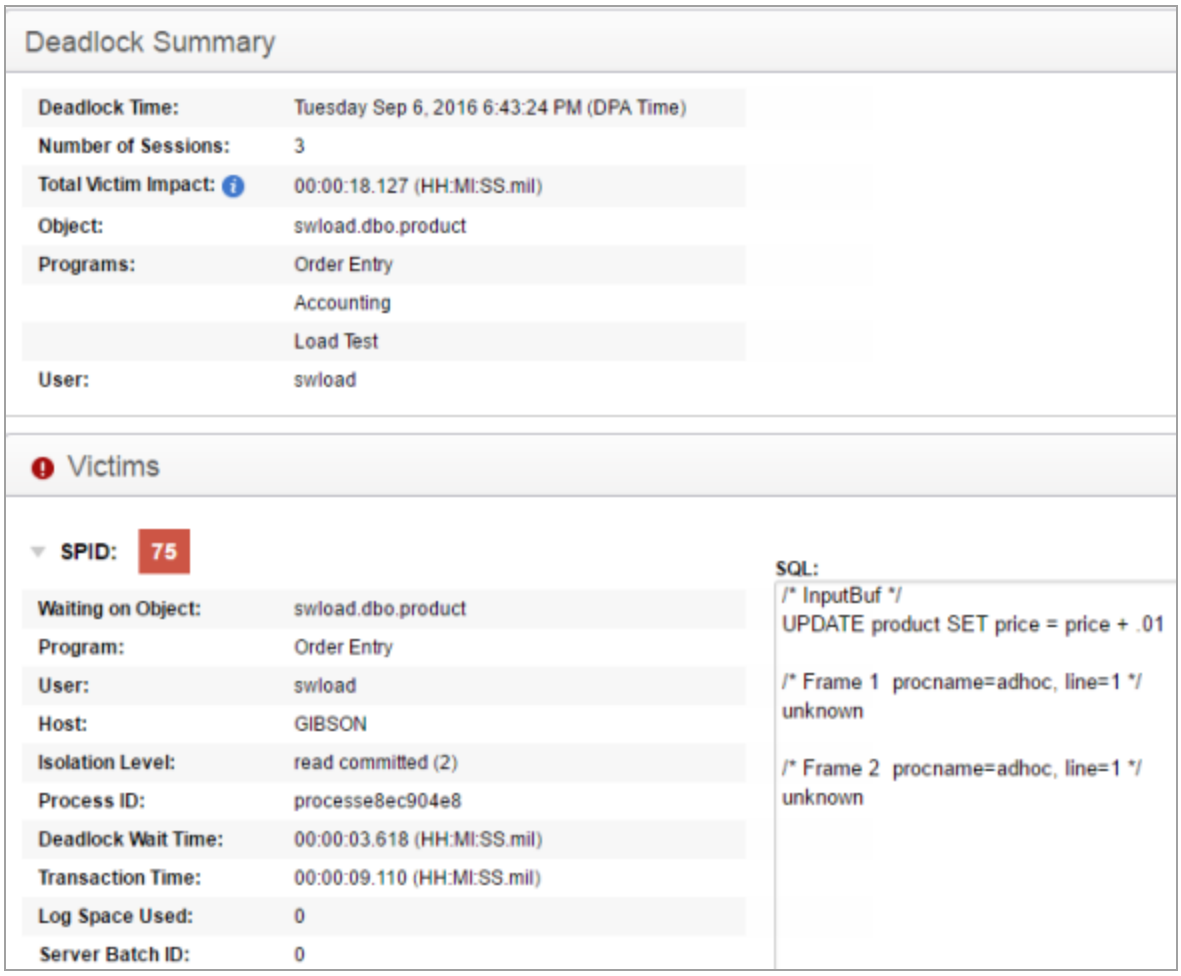
One session (the victim) eventually releases its lock and does not complete its task. The transaction time that the victim spent in contention is a good measure of the impact that the deadlock had on performance.

For monitored SQL Server database instances, DPA provides detailed information about deadlocks, including the Victim Impact (how long the deadlocked transactions ran).

1. From the DPA home page, click a database instance name to display the Trends charts.
2. If necessary, click a bar on the chart to drill down to the time period you're interested in.
3. Click the Deadlocks tab below any Trends chart to see the latest deadlocks for that time period.

4. Click the link in the Time column to open the Deadlock Details page, which includes the following sections:

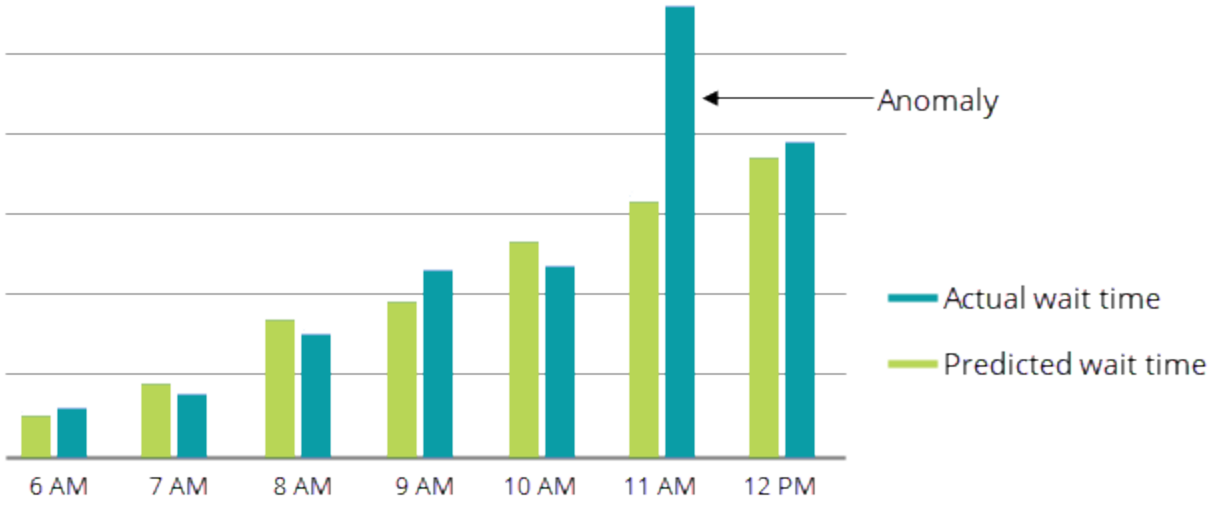
   - The Deadlock Summary section shows high-level information, including the Total Victim Impact.
   - The Victims section shows details about the queries that were rolled back.
   - The Survivor section shows details about the query that was completed.
   - The Deadlocked Resources section shows the type of lock and the lock mode. Click the links for expert advice.



For information about preventing deadlocks, see the "Resolve deadlocks" section of Deadlock analysis in DPA.

Find and investigate unusually long wait times (anomalies)

DPA's anomaly detection algorithm identifies unexpected increases in wait time. DPA collects historical data and uses it to "learn" what normal is. DPA's proprietary algorithm makes predictions based on this data. When wait times for a time period are higher than expected, DPA reports an anomaly.



Get notified when wait time is higher than expected

Configure the Database Instance Wait Time Anomaly alert to be notified whenever wait time is significantly higher than expected for a database instance. (To do this, configure a Wait Time alert and select Database Instance Wait Time Anomaly as the Alert Type.) This alert is triggered if the wait time for an instance was abnormally high during the most recently completed hour.

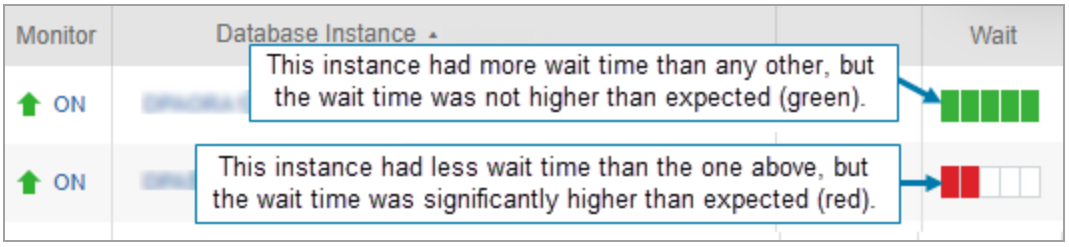
View information about wait time anomalies

The wait time meter on the DPA home page indicates recently detected anomalies. Drill in to a database instance to view more detailed information on the Anomaly Detection charts.

Wait time meter

On the DPA home page, the wait time meter for each database instance provides information about recent database activity:

- The bar **length** shows the amount of wait time for each database instance as compared to all other monitored instances. Use the bar length to quickly identify instances with the highest wait times.
- The bar **color** identifies instances where DPA detected higher-than-expected wait times (anomalies). Yellow indicates a warning status, and red indicates a critical status. (For information about these thresholds, see Anomaly thresholds.)



The wait time meter reflects recent database activity (a rolling one-hour time period). It is updated every 10 minutes to show the wait activity that occurred during the previous 60 minutes.

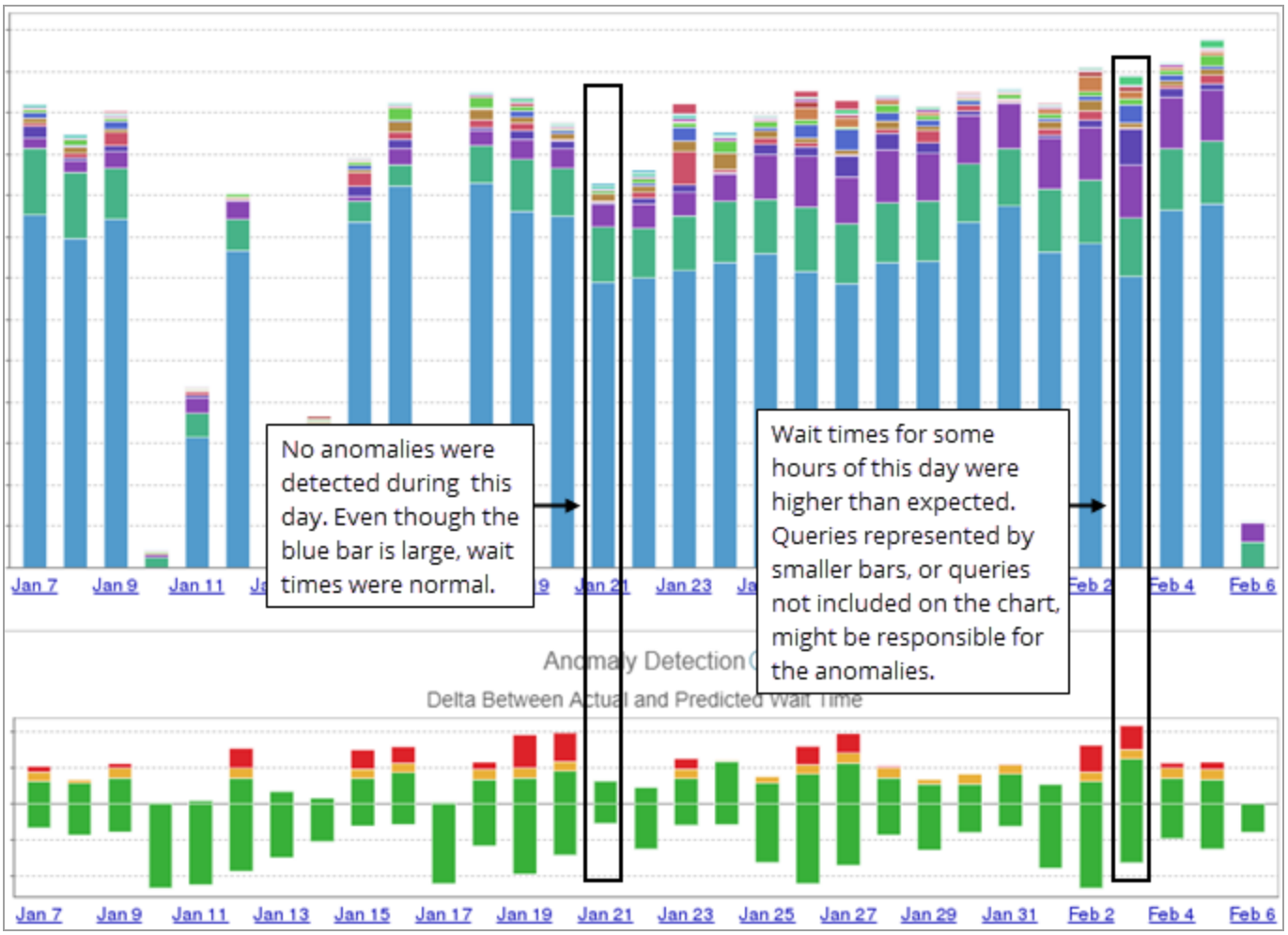
Anomaly Detection chart (30-day period)

If DPA detects wait time anomalies for a database instance, click the database instance on the DPA home page to drill in for more information. The Top SQL Statements chart and the Anomaly Detection chart show information from the past 30 days. These charts work together to help you understand the waits occurring in this database instance:

- The **Top SQL Statements chart** identifies the SQL statements with the highest wait times. In many cases, these are candidates for tuning. But in other cases, further tuning is not possible or the wait times are not a problem. The large bars are normal, and you are more interested in unexpected increases in wait time.

  > An anomaly is detected when the combined wait time for **all** SQL statements is higher than expected. The Top SQL Statements chart shows only the SQL statements with the highest waits, which might not be responsible for the anomaly.

- The **Anomaly Detection chart** identifies days when wait times were significantly higher than expected (wait time anomalies occurred).



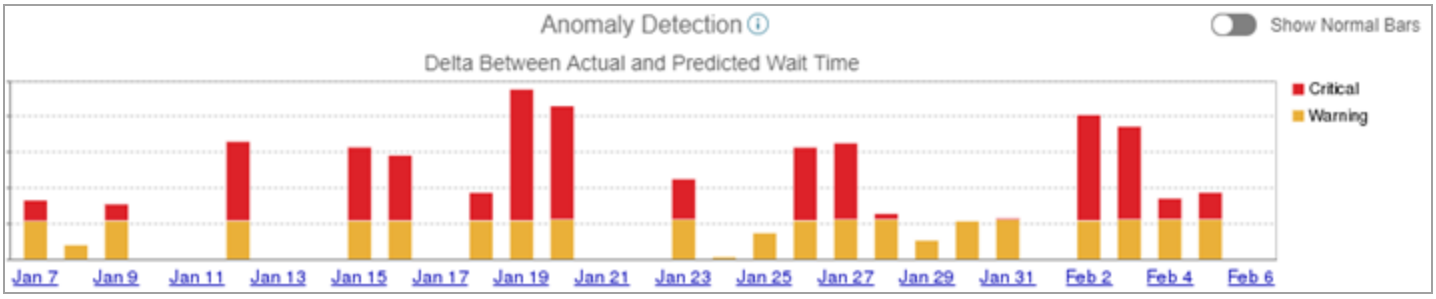
Each bar on the Anomaly Detection chart shows a roll-up of the amount of wait time that the database instance experienced during that day.

- Red segments indicate that wait times for one or more hours were much higher than expected (critical).
- Yellow segments indicate that wait times for one or more hours were higher than expected (warning).
- Green segments above the baseline (0) indicate that wait times for one or more hours were within the normal range, but slightly higher than expected.
- Green segments below the baseline indicate that wait times for one or more hours were lower than expected.

ⓘ DPA classifies all lower-than-expected wait times as normal, and does not alert on them.

Show only warning and critical segments

To focus only on segments that indicate wait time anomalies, you can deselect Show Normal Bars to hide the green bars.



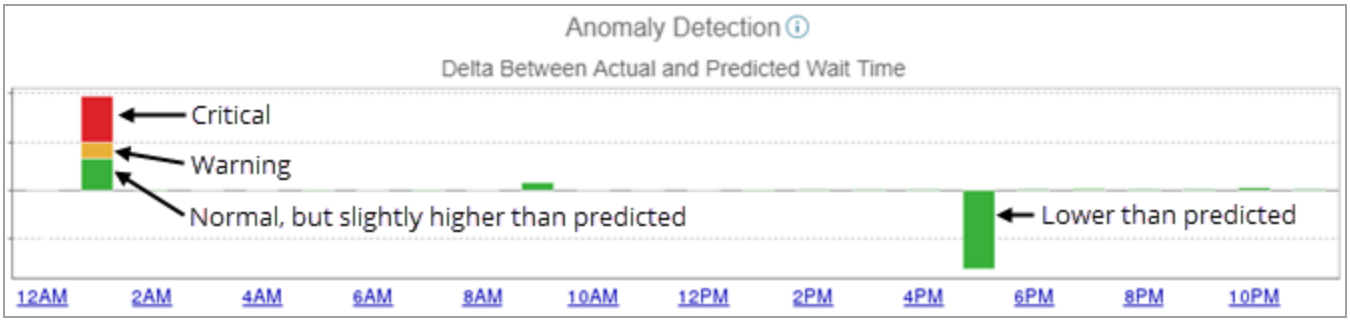
Drill in further

Click a bar that represents a day when anomalies occurred to display the Anomaly detection chart for that day.

Anomaly Detection chart (one-day period)

The Anomaly Detection chart for a one-day period shows the differences between the predicted wait times and actual wait times for each hour. The bar for the current hour shows the differences during the six most recent 10-minute intervals (a rolling one-hour time period).

The baseline (0) represents the predicted value for the hour.



Investigate higher-than-expected wait times

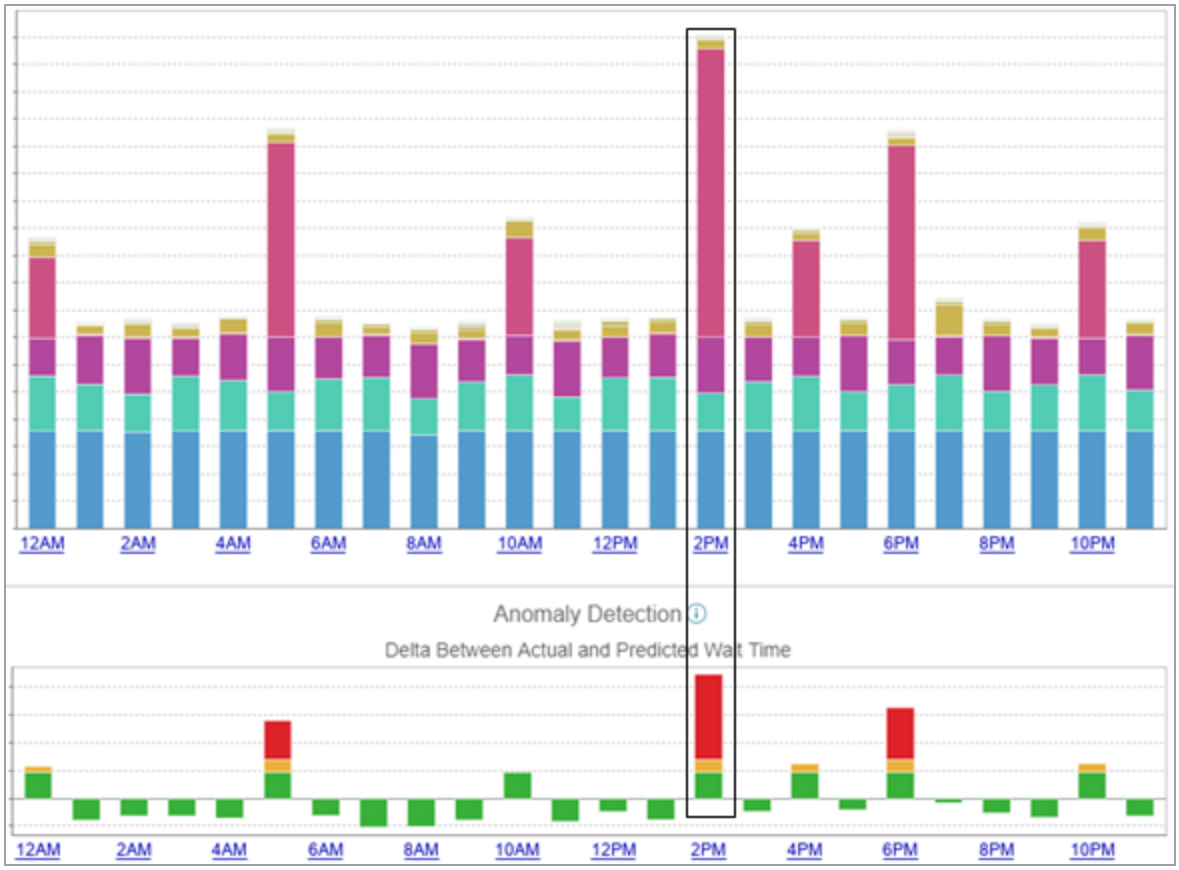
After you determine when anomalies are occurring, you can use either query performance analysis or DPA reports to help you determine which SQL statements are responsible for the anomalies.

Determine when anomalies occurred

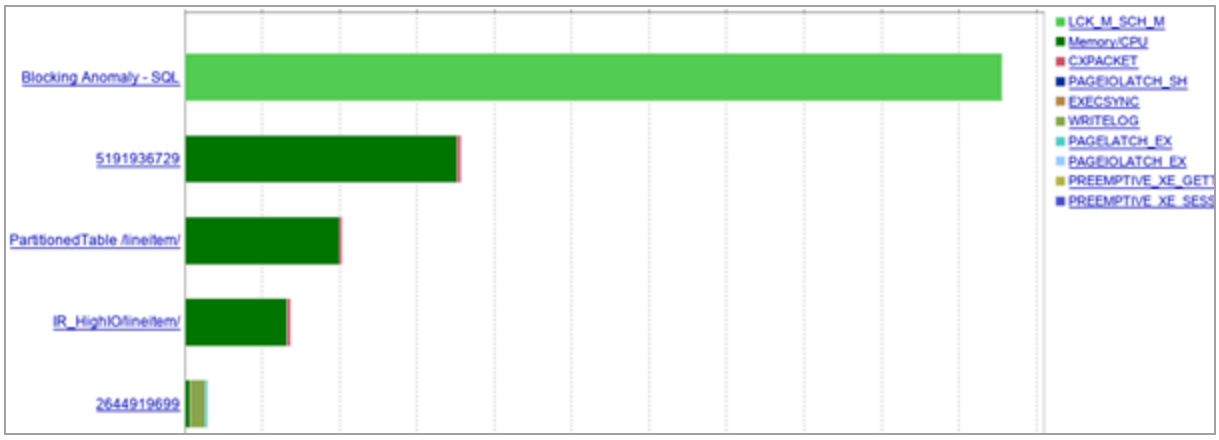
Use the Anomaly Detection charts to determine when anomalies occurred, and to see which SQL statements were running during that time period.

1. From the DPA home page, click the database instance that is experiencing anomalies to display the 30-day Anomaly Detection chart.
2. Click a bar that represents a day when wait times were much higher than expected.

   The one-day Anomaly Detection chart shows the hours when anomalies occurred. In this example, the 2 PM hour had the highest unexpected wait times.



3. Open the Anomaly Detection chart for a one-day period, and find the hours with large red segments. These are the hours when wait times were much higher than expected.
4. Click the bar that represents the hour, and view information about the SQL statements with high wait times that ran during that hour.



Display historic wait times and performance analysis for these queries

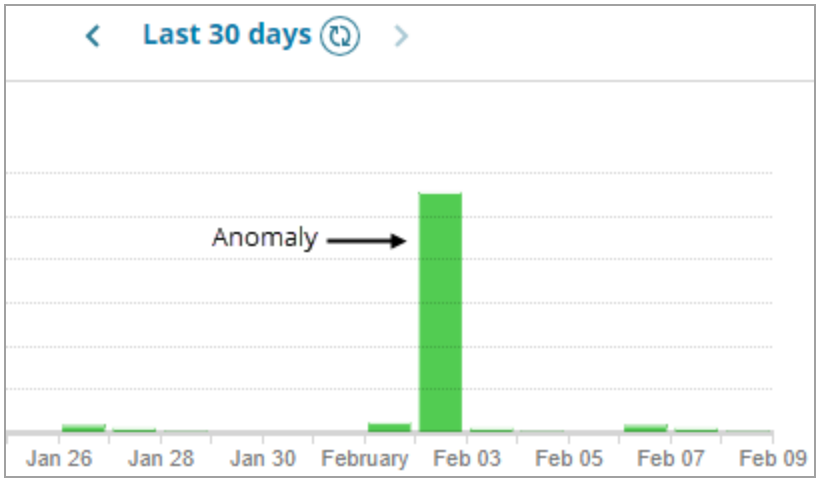
To determine which SQL statement is causing the anomaly, you can use the Query Detail page to view the historic wait times. It's usually a good idea to start with the bars at the top of the list. Also remember that more than one SQL statement might be causing the anomaly.

1. Click a bar that represents a SQL statement.

   The Query Details page displays wait times for that SQL statement during the selected one-hour time period.

2. Click the time period at the top of the page and change the time range. For example, select Last 30 days or Last 90 days.

   In this example, the wait times for February 2 are clearly an anomaly.



3. You can also scroll down to review DPA's query performance analysis for this SQL statement.

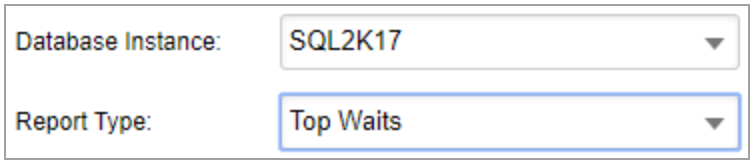
   In this example, we can see that the SQL statement was being blocked by other queries when

the anomaly occurred.

Use DPA reports to review wait or query details

After you determine when the anomalies are occurring, you can create a report to review the wait times for that hour during the last 30 days to look for unusually high wait times.

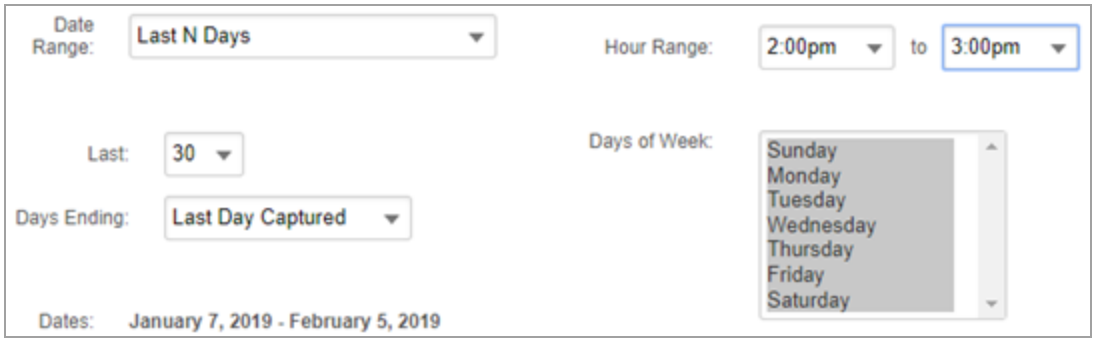
1. Click Reports.
2. Select the database instance that is experiencing anomalies.
3. Select Top Waits as the Report Type.



4. Click Report Options.
5. Under Waits to Display, select the Top 50 Waits.

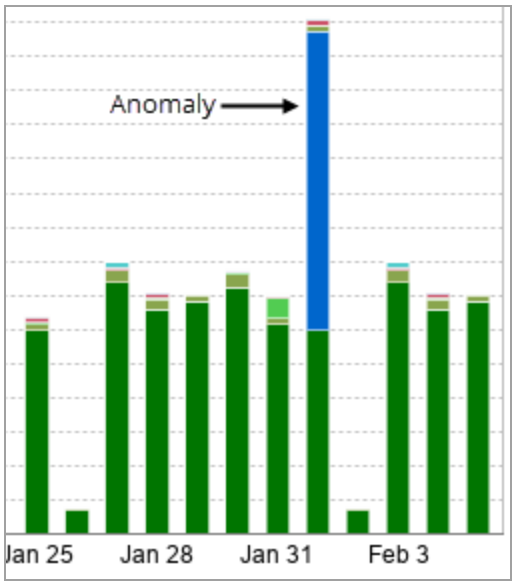
6. Under Dates to Display, select Last N Days as the Date Range, and leave 30 as the number of days.

7. Change the Hour Range to the time period when anomalies are occurring.



8. Click Display Report and review the wait times.

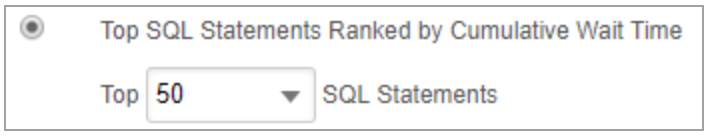
   In this example, the anomaly stands out.



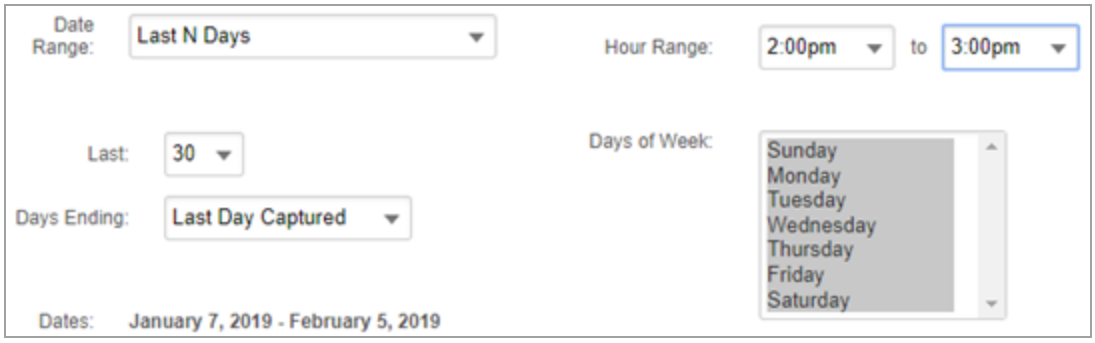
Investigate lower-than-expected wait times

If wait times are much lower than expected, consider investigating to determine whether any SQL statements that normally run during that time period are missing.

1. Open the Anomaly Detection chart for a one-day period, and find the hour with the largest green segment below the baseline. Note the date and hour.
2. Click Reports.
3. Select the database instance, and select Top SQLs as the Report Type.
4. Click Report Options.
5. Under SQL Statements to Display, select the Top 50 SQL Statements.



6. Under Dates to Display, select Last N Days as the Date Range, and leave 30 as the number of days.
7. Change the Hour Range to the time period when wait time was much lower than expected.



8. Click Display Report and review the SQLs that ran each day to help determine if anything is missing.

About anomaly detection in DPA

DPA uses an anomaly detection algorithm to determine if the wait times for a database instance are significantly higher than usual. In some cases, high wait times are normal and expected. With anomaly detection, DPA can alert you to unexpected increases in wait times, and help you investigate these anomalies.

How does DPA's anomaly detection work?

A machine learning algorithm uses wait time data that DPA collects to predict future wait times. DPA uses these predictions to detect wait times that are significantly higher than expected.

Step 1: Data collection	DPA gathers the data that the algorithm will use to learn what normal is and to predict future wait times. Up to 90 days of historical hourly data is used for learning. Anomaly detection requires a minimum of three days of learning data. DPA does not show any information about anomalies until it has collected at least three days of data. Predictions improve as more data is collected.

Step 2: Data analysis and predictions	Based on the learning data, the algorithm calculates: • The amount of wait time that the database instance is likely to experience during each 1-hour period for the next 30 days. • The standard deviation for the entire data set (which is used to calculate thresholds). When enough data is available, predictions include daily and weekly seasonality (patterns of predictable fluctuations): • Daily seasonality accounts for differences during each hour. For example, normal wait times at 2 AM are probably different than normal wait times at 2 PM. • Weekly seasonality accounts for differences during each day of the week. For example, normal wait times at 2 PM on Saturday are probably different than normal wait times at 2 PM on Wednesday. (Weekly seasonality requires at least 30 days of learning data.)
Step 3: Anomaly detection	For each hour, DPA compares the actual amount of wait time during that hour to the predicted value. If the actual amount of wait time is above the warning or critical threshold, DPA: • Changes the color of the wait time meter on the DPA home page. • Displays yellow or red segments on the bars in Anomaly Detection charts. • Triggers the Database Instance Wait Time Anomaly alert, if it has been configured.

How DPA determines the status of an incomplete hour

To determine if the wait time meter and hourly Anomaly Detection chart should show a warning or critical status for an incomplete hour, DPA uses the last 6 completed 10-minute intervals (a rolling one-hour interval). The status is updated every 10 minutes. For example, to determine the status of the 2:00 hour:

- From 2:00 to 2:09, DPA uses data from 1:00 to 1:59.
- From 2:10 to 2:19, DPA uses data from 1:10 to 2:09.
- From 2:20 to 2:29, DPA uses data from 1:20 to 2:19 (and so on).

For each 10-minute interval of the current hour, DPA uses a rolling one-hour interval to determine the status shown on the wait time meter. For example, 2:10 to 2:19 uses data from 1:10 to 2:09.

SQL statements excluded from the trend charts

The anomaly detection algorithm uses the total wait time for the database instance, including wait time from any SQL statements that you have excluded from the trend charts. In most cases, a statement is excluded from the trend charts because it always has high wait times and the large bar dominates the charts. If the statement runs on a regular schedule with the expected amount of wait time, no anomaly would be detected during that time period, because high wait times are normal during that period. An anomaly would be detected only if wait times during that period were significantly higher than normal, in which case you might want to investigate the change.

Does anomaly detection work well for all database instances?

DPA's anomaly detection algorithm, like most algorithms associated with workloads, works best when:

- The monitored database instances have a consistent workload executing against them.
- Daily and weekly seasonality is consistent. For example, database wait times are similar each Monday at 10 AM.
- DPA monitoring is always on (not shut down for hours or days at a time).

The algorithm might not work well when:

- The workload for a database instance is sporadic (for example, QA or reporting instances with inconsistent wait times).
- Daily and weekly seasonality is not consistent. For example, the workload on Monday at 10 AM varies from one week to the next, with no predictable pattern.
- DPA is not monitoring the instance consistently, and so it cannot get a good understanding of what normal is.

If anomaly detection does not work well for any of your monitored instances, SolarWinds recommends disabling anomaly detection for those instances.

Large gaps in the learning data

If monitoring stops for more than 30 days, the anomaly detection algorithm does **not** make predictions based on the stale learning data collected before the 30-day gap. DPA collects new learning data and, after three days, begins to make predictions based on the current data.

Anomaly thresholds

Anomalies are classified as warning and critical. The threshold for each classification is based on the standard deviation of the wait times for the associated time period.

Standard deviation is a measure of how dispersed the values in a data set typically are.

The default values for the thresholds are listed below. You can edit the associated advanced option to change the default values.

Classification	Default threshold	Advanced option
Warning	The predicted wait time for the hour + 2 standard deviations	`ANOMALY_DETECTION_THRESHOLD_WARNING`
Critical	The predicted wait time for the hour + 3 standard deviations	`ANOMALY_DETECTION_THRESHOLD_CRITICAL`

Specify the learning date after the load on a database instance changes

If the load on a database instance changes significantly (for example, because of changes in the network environment), the previously collected learning data is no longer accurate. To prevent this data from being used for anomaly detection, set the advanced Support option `ANOMALY_DETECTION_FORCE_LEARNING_DATE` to the date when the load change occurred. Wait time data collected before this date will not be used to predict future wait times.

Disable anomaly detection for a database instance

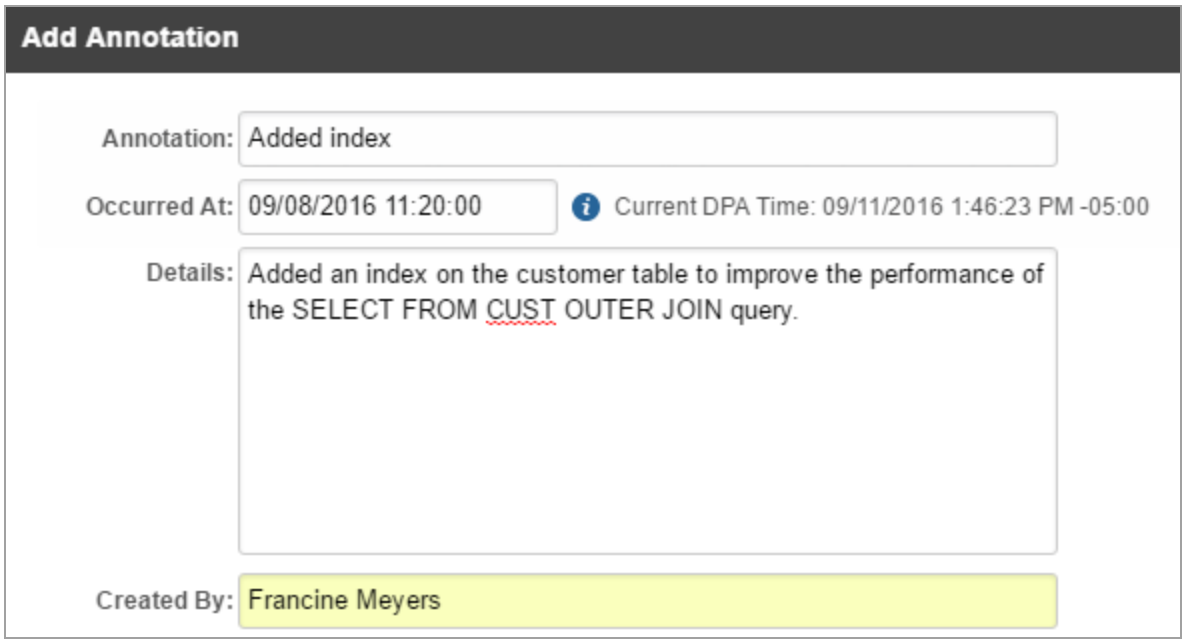
By default, anomaly detection is enabled for all database instances. To disable anomaly detection for a database instance that with an inconsistent workload or sporadic monitoring, set the advanced option `ANOMALY_DETECTION_ENABLED` to `False` for that instance.

Add an annotation to document a change to the database

When you make a change that could affect performance (such as adding an index, tuning a query, or adding resources), you can add an annotation in DPA to show when that change was made. The annotations are displayed on all trend and timeslice charts. By comparing performance data before and after the change, you can see what effect the change had.

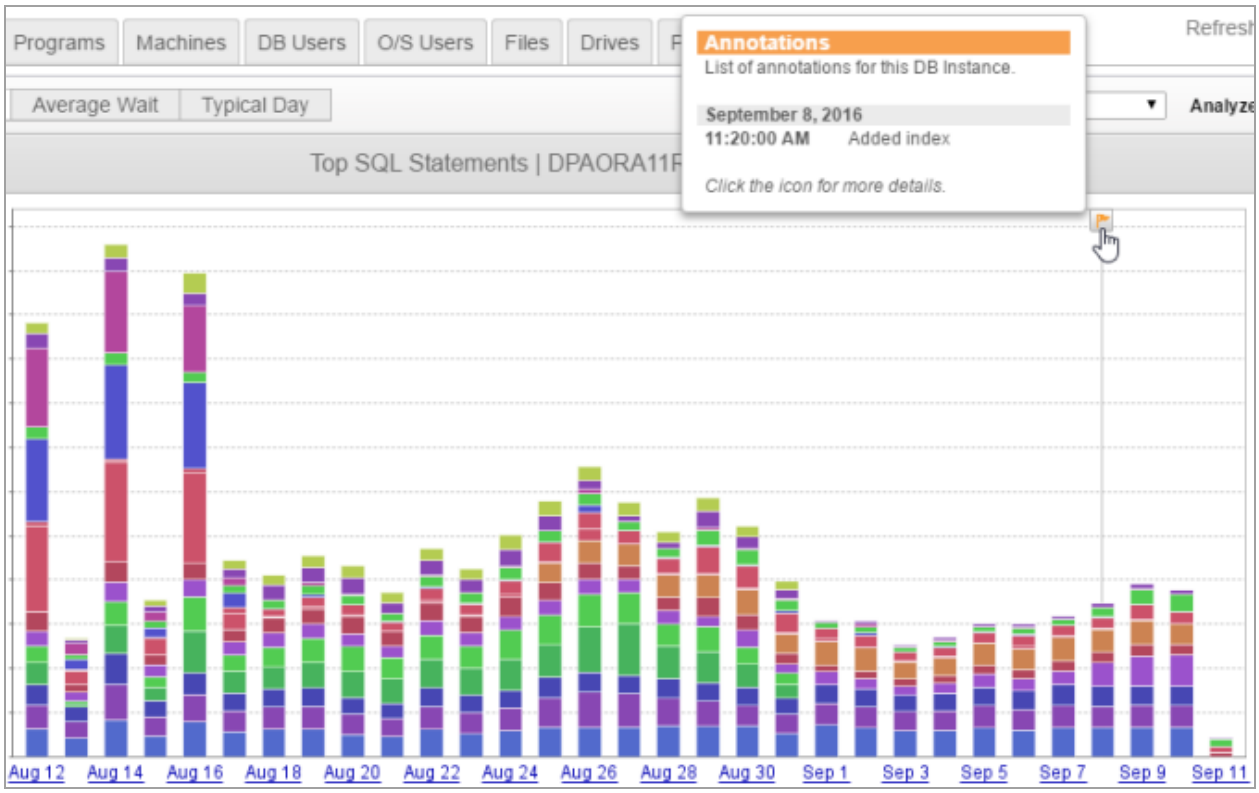
1. From the DPA home page, click the name of the database instance affected by the change.
2. Click Annotate in the upper-right corner of the trend chart.
3. Name the annotation, specify when it was added, and provide details about what change was made and why.

   If your DPA server is in a different time zone, enter the DPA server time.



4. Click Save.

   The annotation is displayed as a flag on the chart. Point to the flag to see a summary, or click it to see details.



Find SQL statements in DPA

Use the Find SQL feature to locate SQL statements that might not be displayed on the Trends charts.

Search for SQL statements in DPA

Use the Find SQL feature to search for any SQL statement based on what you know about it. You can specify a time range and then apply any combination of filters and search strings:

- Even if you don't know anything about the SQL text, you can apply filters to locate SQL statements that were run by a certain user, as part of a certain application, from a certain computer, or against a specific database.
- If you know something about the SQL text, you can enter search strings such as table names or the operation being performed.

Example use cases

Examples of possible use cases include:

- A user with the user name `jsmith` complained about the performance of the application `acme_ecommerce`, which he ran at about 10:00 AM yesterday. To investigate, set the time period, and then filter by user and application.
- You tuned a SQL statement, and it no longer appears in the Trends charts. You want to open the tuned version in the Query Details page to see the results of your tuning. Because you are familiar with the statement, you can search for terms or phrases from the SQL text.
- Deleting orders from a third-party application is suddenly very slow. You do not have access to the code, but you want to analyze `DELETE` statements related to the `ORDERS` table. You can search for SQL statements that include `DELETE` and `ORDERS`. To further narrow the results, you can also filter by the application name.

Open the Find SQL page

1. From the DPA home page, click the name of the database instance you want to search.

   The Trends tab lists the SQL statements with the highest wait times.
2. At the top of the page, click Find SQL.

The Find SQL page opens. If the Find SQL feature is not enabled for the selected database instance, the Find SQL page displays a message. You can enable the Find SQL feature for all database instances or for a specific database instance.

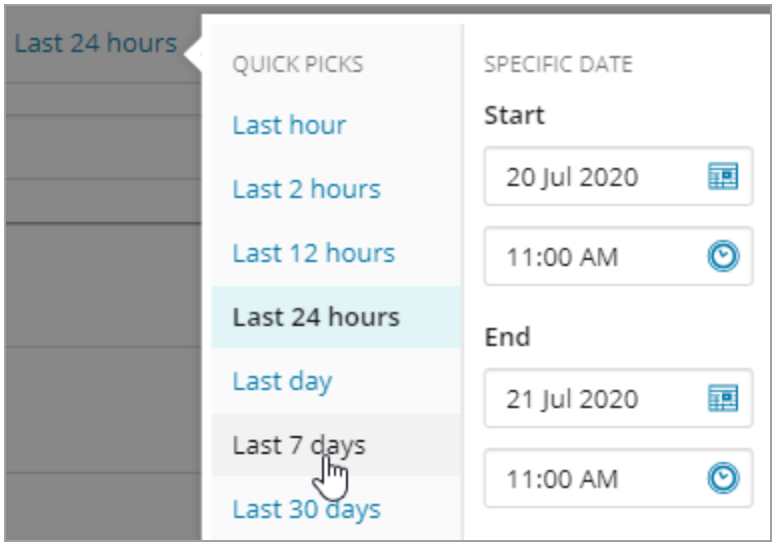
Select a time period

DPA searches for SQL statements that ran during the selected time period, which is displayed at the top of the page. When you open the Find SQL page, it defaults to the last 24 hours. You can select a different time period.

1. Click the date range at the top of the page to open the date picker.
2. Select a predefined time period or enter specific dates.

   The advanced option `CLEAN_DAYS_OF_FIND_SQL_DATA` specifies the maximum number of days Find SQL data is stored (by default, 30 days). On the Find SQL page, if you specify a time period that is larger than the `CLEAN_DAYS_OF_FIND_SQL_DATA` value, DPA changes the selected period to the maximum number of days.

   > If you increase the value of `CLEAN_DAYS_OF_FIND_SQL_DATA` to more than 30 days, be aware that larger values require more disk space and might affect performance. The maximum value is 90 days.



3. Click Search to repeat the current search in the selected time period.

Apply filters

Depending on what type of database instance is selected, some or all of the following filter categories are available:

- **Database user**: The user ID that ran the SQL statement.
- **Program**: The application that ran the SQL statement.
- **Database**: The database that the SQL statement queried.
- **Machine**: The computer from which the SQL statement ran.

To apply filters:

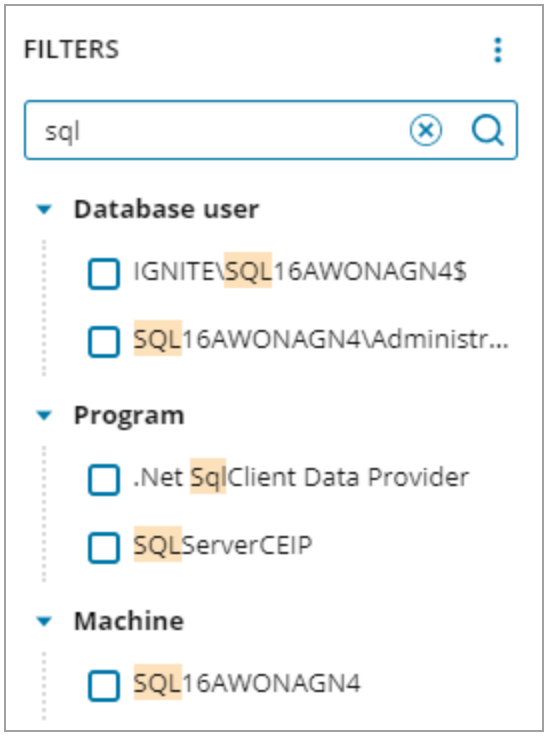
1. In the upper-left corner, click Filters.

   The filter categories available for the selected database instance are expanded by default.

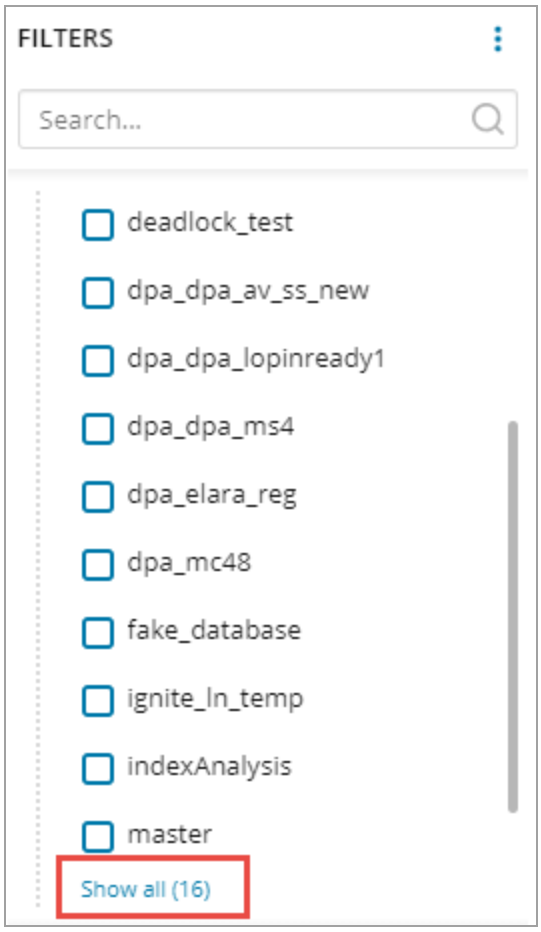
   > Click the vertical ellipsis (⋮) in the upper-right corner of the Filters box to expand or collapse all filter categories.

2. To search for a value, enter the search string in the Filters Search field.

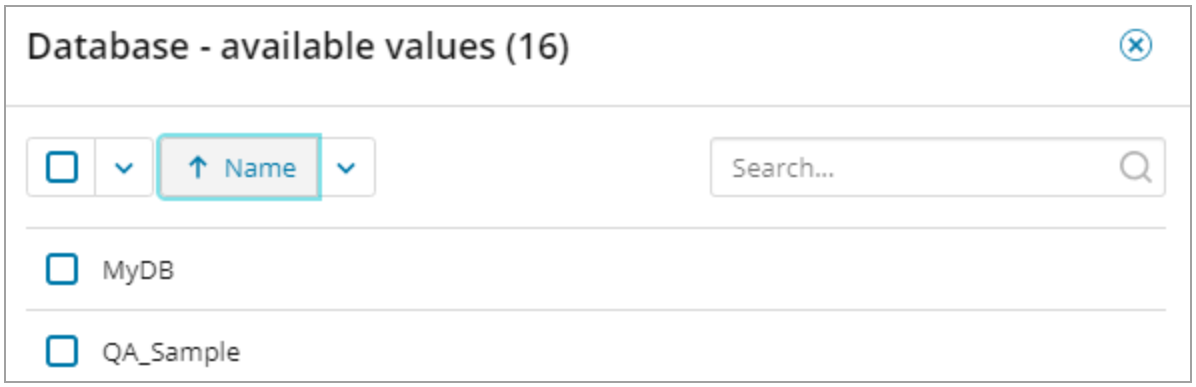
   Only values that include the search string are displayed.



3. If a filter category includes more than 10 items, click the Show All link.



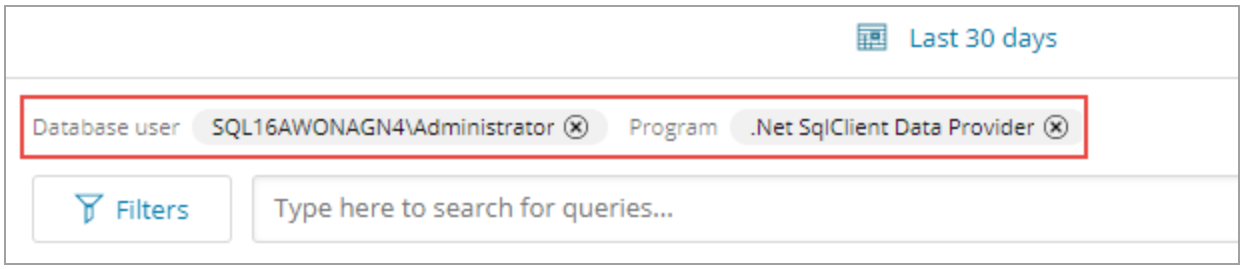
A dialog opens, from which you can page through to see all items, change the sort order, or search.



4. Select one or more filters, and click Search to apply them.

   The search results include only SQL statements that match all filters. If no search terms are applied with the filters, results are ordered by wait time.

   The applied filters are listed above the Filter button and Search bar.



Click the X beside a filter to remove it, or click Clear All to remove all filters.

Enter search strings

If you know any terms that are likely to be part of the SQL text, enter them in the Search box and click Search. For example, you can search for table names, column names, or the type of operation. If you have named the SQL statement, you can search for the SQL statement name. You can also search for the SQL statement hash.

When you enter search terms, the results are ordered by relevance. For example, if you enter multiple terms, SQL statements with all of the terms are listed before those with only some of the terms.

Simple and advanced search modes

When you enter search terms, you can choose between two modes:

- **Simple mode** is similar to an internet search engine. You can enter individual terms, or you can use double quotes to identify a phrase. In simple mode, you can enter multiple terms but no more than **one** phrase. All entries are separated by an implicit OR, and so statements that include any of the entries are returned.
- In **advanced mode**, you can use Boolean operators, wildcards, grouping, and other advanced features to refine your search. You can also enter multiple phrases.

For examples of phrases and terms you can enter in either mode, and for descriptions of the features available in advanced mode, see Find SQL search rules.

For example, in simple mode you can enter the following search terms to find information about `INSERT` and `SELECT` operations against the `ORDERS` table:

```
insert select orders
```

SQL statements with `INSERT` or `SELECT` operations against the `ORDERS` table would be ranked higher because they contain more of the search terms, but the results would also include SQL statements that performed `DELETE` operations against the `ORDERS` table, and SQL statements that performed `INSERT` or `SELECT` operations against other tables.

To narrow the search results, you can change the search mode to advanced. The following example uses grouping and Boolean operators to limit the results to SQL statements that include `INSERT` or `SELECT` and `ORDERS`:

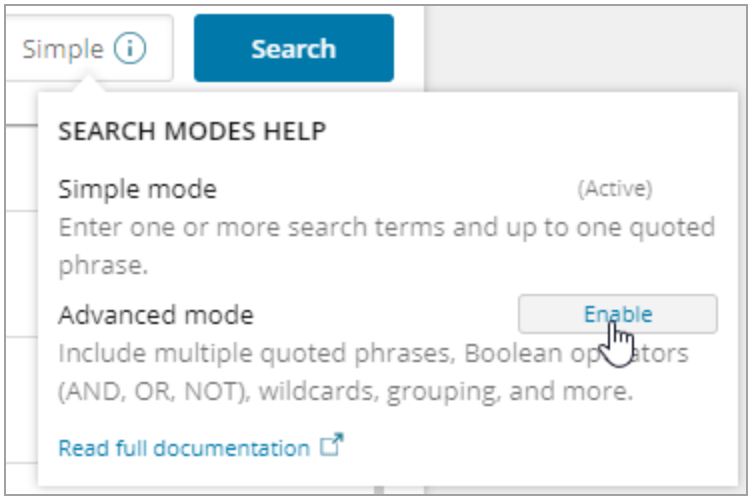
```
(insert OR select) AND orders
```

If you have contextual information such as what user, application, or machine executed the SQL statement, you can apply filters in addition to search terms to narrow the search results. Each time you add or change search terms and filters, click Search again to refresh the results.

Change the search mode

When you access the Find SQL page, simple mode is selected by default. To change the mode:

1. Move your mouse pointer over the mode name in the search box to display descriptions of each search mode.
2. In the description popup, click Enable to change modes.



Get more information about a SQL statement

Click the blue arrow to the right of the Wait Time to display statistics about the SQL statement and the complete SQL text.

Click the SQL hash or SQL name to display detailed information about the SQL statement on the Query Details page.

Share search parameters

Click Share at the top of the page to generate a link that other users can click to display the current search results for the selected database. The link is saved to the clipboard. You can paste it into an email or other document.

Find SQL search rules

When you search for a SQL statement from the Find SQL tab in DPA, the following rules apply:

- Rules for all searches
- Rules for searches in simple mode
- Rules for searches in advanced mode

Rules for all searches

The following rules apply to searches in both simple and advanced modes:

- Searches are not case sensitive. For example, `SELECT` and `select` return the same results.
- You can enter two types of search entries:
  - A **term** is a single word, such as `select`. You can enter multiple terms separated by spaces. For example:

    ```
    select where
    ```

    The terms do **not** have to occur together in the SQL statement.
  - A **phrase** contains multiple words surrounded by double quotes. For example:

    ```
    "select userid"
    ```

    Use phrases to search for terms that must occur together in a specific order.
- If you include multiple entries, there is an implicit `OR` between each entry. For example:
  - Entering `select userid` returns SQL statements that include either `select` **or** `userid`.
  - Entering `"select userid" count` returns SQL statements that contain either `select userid` **or** `count`.

  SQL statements that include all entries are ranked higher in the search results.
- Partial terms without wildcards are not matched. For example, searching for `sel` does **not** return SQL statements that contain `select`.

  Wildcards are supported only in advanced mode.

Rules for searches in simple mode

- Special characters do not need to be escaped in simple mode.
- You can enter multiple terms, but you cannot enter more than **one** phrase in simple mode. For example, you can enter:

  ```
  "select id" employees status active
  ```

  If you enter multiple phrases in simple mode, DPA treats everything between the first `"` and the last `"` as one phrase. For example, the following phrases are combined:

  ```
  "select userid" "from employees"
  ```

  SQL statements that contain `select id from employees` are returned, but SQL statements that contain (for example) `select id, lastname from employees` are **not** returned.
- Simple mode does not support Boolean operators, wildcards, grouping, or other options described in the following section.

Rules for searches in advanced mode

The following options are available in advanced mode:

- Multiple phrases
- Boolean operators
- Wildcards
- Fuzzy searches
- Proximity searches
- Grouping
- Escaping special characters

Multiple phrases

In advanced mode, you can enter multiple phrases separated by spaces or Boolean operators. For example:

```
"select count" "where type='user'"
```

Boolean operators

In advanced mode, you can use the following Boolean operators. Operators that are words (`OR`, `AND`, `NOT`) must be entered in all caps.

Operator	Description
OR	Returns SQL statements that include either term or phrase anywhere in the statement. The following example returns SQL statements that include either `select` or `insert`: `select OR insert` SQL statements that include both terms are at the top of the search results. ⓘ `OR` is the default operator. If you enter multiple terms without an operator, `OR` is used.
AND	Returns SQL statements that include both terms or phrases anywhere in the statement. The following example returns SQL statements that include both the phrase `select username` and the term `mgmt_user`: `"select username" AND mgmt_user`

NOT	Excludes SQL statements that include the term or phrase that immediately follows the operator `NOT`. The following example returns SQL statements that include `select` but do **not** include `username`:
	`select NOT username`
	You cannot use the `NOT` operator with only one search term or phrase. For example, the following search returns no results:
	`NOT username`

+	Indicates that the term or phrase after the `+` is required. Other terms or phrases not preceded by a `+` are optional. The following example returns all SQL statements that include `select`: `+select distinct` SQL statements that also include `distinct` are ranked higher in the search results.

Wildcards

In advanced mode, you can use the following wildcards:

Wildcard	Description
?	Replaces a single character in the search term. The following example returns SQL statements that contain `values` or `value$`: `value?`

Wildcard	Description
*	Replaces zero or more characters in the search term. The following example returns SQL statements that refer to any Oracle v$ view: `v$*`

Wildcards can be used in the following locations:

- You can place wildcards in a single search term, but **not** in a phrase. A phrase that includes a wildcard (for example `"select co?nt"`) returns no results.
- You can place a wildcard in the middle or at the end of a search term (for example `count*` or `cou?t`) A wildcard **cannot** be placed at the beginning of a term. If a wildcard is the first character of a term (for example, `select ?ount`), DPA displays the message `Unable to parse search request`.

Fuzzy searches

Fuzzy searches find terms that are similar in spelling to the specified term. Place a tilde (~) at the end of a single-word term to perform a fuzzy search. For example, the following returns SQL statements that include `account`, `accounts`, and `count`:

```
account~
```

Optionally, you can enter a number between 0 and 1 after the tilde to require more similarity than the default fuzzy search. Higher values require more similarity. For example, the following returns SQL statements that include `account` and `accounts`, but **not** `count`:

```
account~0.8
```

Proximity searches

Proximity searches find SQL statements that include all of the specified terms, but only when they are near each other. Place a tilde (~) followed by an integer greater than 0 after a phrase. The integer specifies the maximum number of words separating the terms. For example, the following returns SQL statements with `emp` and `insert` separated by no more than 10 words:

```
"insert emp"~10
```

Grouping

Use parentheses to group terms or phrases separated by Boolean operators. For example:

```
("owner = 'rdsadmin'" OR "owner = 'sys'") AND select
```

returns SQL statements that include any of the following combinations:

- `select` and `owner = 'rdsadmin'`
- `select` and `owner = 'sys'`
- `select` and both phrases

Escaping special characters

The following special characters must be escaped in advanced mode:

```
+ - && || ! ( ) { } [ ] ^ " ~ * ? : \
```

To escape these characters, place a backslash (`\`) before each special character. For example, to search for `SELECT count ( * )`, enter:

```
SELECT count \( \* \)
```

Move the Find SQL indexes

DPA indexes the SQL statements that it monitors on each database instance. The Find SQL feature uses this indexed data. By default, the indexes are created in the DPA directory.

DPA keeps indexed data for 30 days. In a large or busy DPA deployment, the Find SQL indexes can become large. In some cases, large indexes can affect DPA performance. If the Find SQL indexes are in the default location and the storage requirement reaches 5 GB, DPA displays a message. If you receive this message, you have the following options:

- If the DPA server has enough storage capacity and DPA is not experiencing performance problems, you can increase the recommended storage limit. To do this, edit the advanced Support option `FIND_SQL_INDEX_SIZE_LIMIT`.
- Move the indexes out of the DPA directory to a location with enough storage capacity for your indexed data.

See the following sections:

Space requirements for the Find SQL indexes

The amount of storage space required for the indexed search data is determined by the number of monitored database instances with Find SQL enabled. Index files for one instance can take up to 300 MB. So, for example, if you are monitoring 120 instances, reserve at least 36 GB to provide adequate disk space for the Find SQL indexes:

120 x 0.3 GB = 36 GB

Move the indexes to a custom location

1. Stop DPA.
2. Create a folder for the indexes in the new location, and set the privileges to allow the DPA service to read and write to the folder.
3. Configure DPA to write index data to this folder:
   a. Open the `system.properties` file in a text editor. This file is located in the following directory:

      DPA-install-dir\iwc\tomcat\ignite_config\idc

      In Windows, the default location is:

      C:\Program Files\SolarWinds\DPA\iwc\tomcat\ignite_config\idc

      In a Linux-based OS, the default location is:

      /home/solarwinds/dpa_*V_v*/DPA/iwc/tomcat/ignite_config/idc

   b. Add one of the following lines:

      - If you are going to move all indexes to the custom location, enter:

        com.solarwinds.dpa.findSql.index.location=*your_custom_location*

      - If you are going to move only indexes for specific database instances to the custom location, enter a line for each database instance, where *db_id* is the database ID:

        com.solarwinds.dpa.findSql.index.location.*db_id*=*your_custom_location*

        For example:

        com.solarwinds.dpa.findSql.index.location.5=/home/dpa/custom_location
4. Move indexes from the default location to your custom location. The default location is:

   DPA-install-dir/iwc/tomcat/ignite_config/lucene-index/*RepoID*/*db_id*

   where:

   - *RepoID* is the DPA repository ID.

     If you have more than one folder in this path (for example, because you changed the database used as the repository), open the *DPA-Install-Dir*/iwc/tomcat/ignite_config/idc/repo.properties file. The value of the `repo.guid.hash` property is the

currently used repository ID.

- *db_id* is the database ID.

An example of the command to move all indexes is:

```
mv DPA-install-dir/iwc/tomcat/ignite_config/lucene-index/ /home/dpa/custom_location
```

An example of the command to move an index for a specific database instance is:

```
mv DPA-install-dir/iwc/tomcat/ignite_config/lucene-index/<RepoID>/5 /home/dpa/custom_location/RepoID/
```

5. Start DPA.

Troubleshooting tip

If you followed the instructions to move the Find SQL indexes but DPA still displays a yellow banner with the message about the index size:

1. Verify that the indexes were moved to the custom location and that the `system.properties` file has the correct location.
2. Check the default index location in the DPA directory and verify that the indexes were removed. The indexes might have been copied instead of being moved. If the custom location is configured and the indexes were copied there, remove the index folders from the default location.

   For example, if the index for the database instance with the ID of 5 is configured in `system.properties`, remove the following folder:

   DPA-install-dir`/iwc/tomcat/ignite_config/lucene-index/`*RepoID*`/5`

   If all indexes are configured in `system.properties`, remove the following folder:

   DPA-install-dir`/iwc/tomcat/ignite_config/lucene-index`

Searches on the Find SQL tab do not return any data

If no results are returned when you search for a SQL statement, it could be for one of the following reasons:

- Data summarization has not had time to run
- DPA was not restarted after the Find SQL index files were moved
- DPA cannot create or modify the index
- The search criteria violates the Find SQL search rules

Data summarization has not had time to run

The Find SQL functionality was recently enabled, and data summarization has not run yet. Data summarization indexes the SQL statements to make them available for searching.

Resolution: Wait for data summarization to run. For a newly registered database instance, data summarization runs after at least one hour of data is collected. For a database instance with historical data, the process runs in 10 minute intervals.

DPA was not restarted after the Find SQL index files were moved

The Find SQL index files were moved, but DPA was not restarted afterward.

Resolution: Restart DPA and wait for data summarization to run.

DPA cannot create or modify the index

If DPA cannot create or modify the index, the following error can be found in the `error.log` file:

```
ERROR (2020-09-15T01:58:54,059-0700) [findSqlProcess-thread-358]
PeriodicLogger:190 - Error when getting index writer or index directory for text
index and database ID 64.
```

Check for the following issues:

- Check the amount of available space in the location where the Find SQL index files are stored. By default, the indexes are stored in the following location:

  DPA-install-dir`/iwc/tomcat/ignite_config/lucene-index`

  If you moved the files to a custom location, the location is specified by the `com.solarwinds.dpa.findSql.index.location` property in the following file:

  DPA-install-dir`/iwc/tomcat/ignite_config/idc/system.properties`

  Resolution: Increase the disk size or move the Find SQL indexes to a different location.

- Verify that the DPA service user has read and write permissions for the folder and files that store the Find SQL indexes. The DPA service user is LOCAL_SERVICE on Windows, and the user that runs DPA on a Linux-based OS.

  Resolution: Grant read and write permissions to the DPA service user.

- If there is enough space and permissions are correctly set, the index might be corrupted.

Resolution: Rebuild the index:

1. Stop DPA.
2. If only **one** database instance is identified in the error message, delete the Find SQL index folder for that database instance. The default location is:

   DPA-install-dir/iwc/tomcat/ignite_config/lucene-index/*RepoID*/*db_id*

   where:

   - *RepoID* is the DPA repository ID.

     If you have more than one folder in this path (for example, because you changed the database used as the repository), open the *DPA-install-dir*/iwc/tomcat/ignite_config/idc/repo.properties file. The value of the repo.guid.hash property is the currently used repository ID.

   - *db_id* is the database ID.

   If you moved the index files for all database instances to a custom location, the location is specified by the com.solarwinds.dpa.findSql.index.location property in the following file:

   DPA-install-dir/iwc/tomcat/ignite_config/idc/system.properties

   If you moved the index files for only some database instances to a custom location, the location is specified by the com.solarwinds.dpa.findSql.index.location.*db_id* property in that file.
3. If you receive error messages for **all** database instances, delete this folder to remove all Find SQL indexes:

   DPA-install-dir/iwc/tomcat/ignite_config/lucene-index/*RepoID*
4. Restart DPA.

The search criteria violates the Find SQL search rules

If you enter search terms that you believe should return data, but the Find SQL search does not show any results for that search, verify that your search terms follow the Find SQL search rules. For example, if you try to use wildcards in Simple mode (sel*) or you enter a partial term without a wildcard in either mode (sel), the search does not return any results.

Enable or disable the DPA Find SQL feature

When the Find SQL feature is enabled for a monitored database instance, you can search for any SQL statement that ran on the instance based on what you know about the SQL statement.

Enable the Find SQL feature

When you open the Find SQL page, it displays a message if it is not enabled for the selected database instance. You can enable the feature from the Find SQL page or by setting an advanced option. The feature can be enabled globally or for specific database instances.

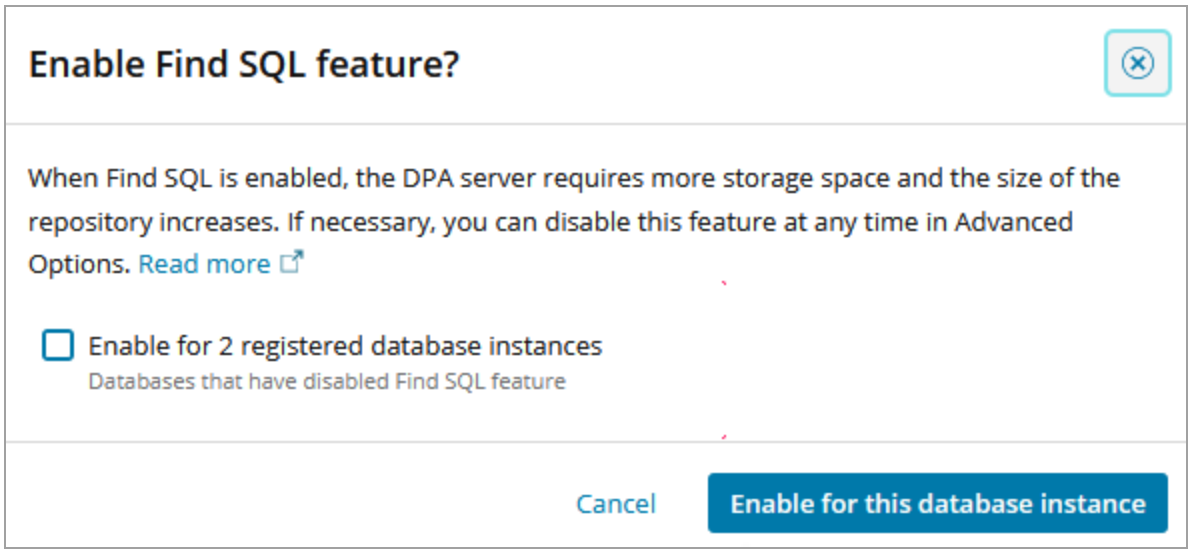
If the Find SQL feature is enabled, DPA requires additional storage space for the indexes. By default, DPA stores the indexes on the DPA server, but you can move them.

From the Find SQL page

To enable the Find SQL feature from the Find SQL page:

1. Below the message that Find SQL is not enabled, click Enable Find SQL feature.

   The Enable Find SQL feature dialog box opens.



2. Choose one of the following options:
   - To enable the Find SQL feature only for the currently selected database instance, click Enable for this database instance.
   - To enable the Find SQL feature for **all** monitored database instances that do not currently have it enabled:
     a. Select the Enable for *x* registered database instances checkbox. The text on the button changes.
     b. Click the Enable for *x* database instances button.

From advanced options

To enable the Find SQL feature using advanced options, set the advanced option `FIND_SQL_ENABLED` to `True`:

- Change the setting on the System Options tab to enable the feature globally.
- Change it on the DB Instance Options tab to enable the feature for a specific database instance.

> If you change the setting for a specific database instance, the global value is overridden for that instance.

Disable the Find SQL feature

When you disable the Find SQL feature, all associated indexes are removed. To disable the Find SQL feature, set the advanced option `FIND_SQL_ENABLED` to `False`:

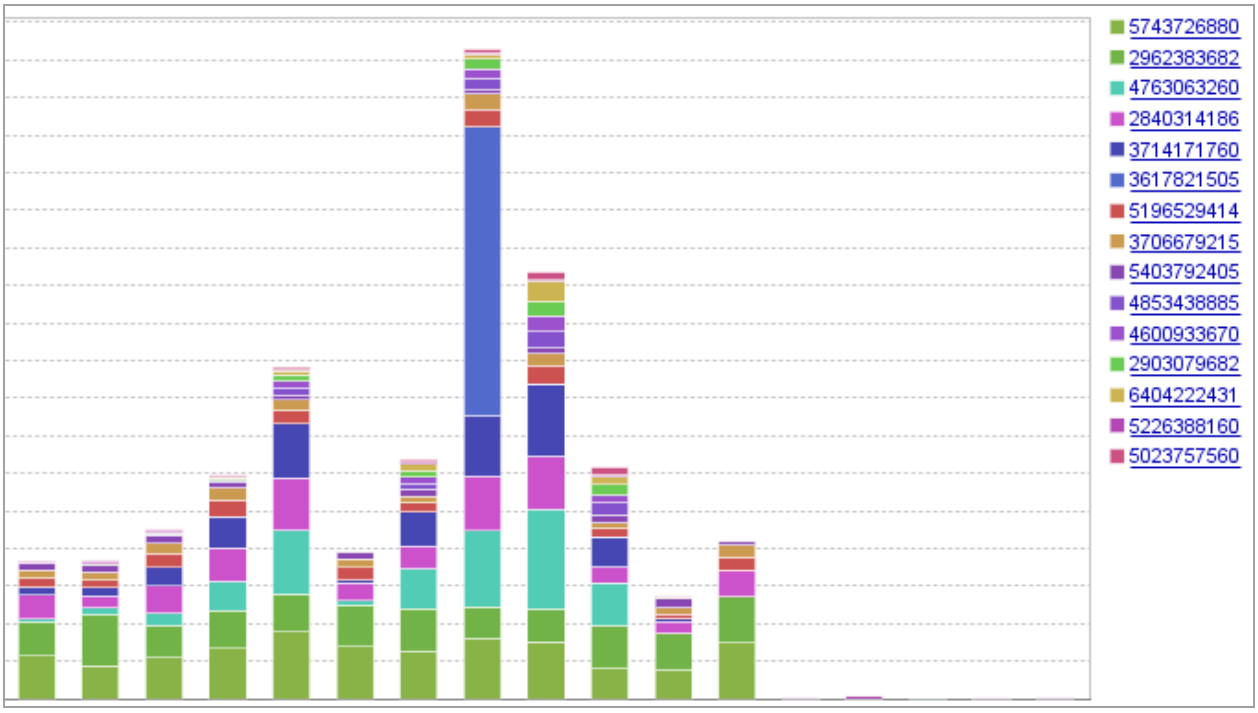
- Change the setting on the System Options tab to disable the feature globally.
- Change it on the DB Instance Options tab to disable the feature for a specific database instance.

Manage SQL statements

You can name SQL statements to make them easier to identify in reports and charts. You can also exclude SQL statements that are dominating the trends charts but cannot be tuned or do not cause issues.

Name SQL statements

On the right side of each DPA Trends chart, a legend identifies each SQL statement. By default, SQL statements are identified by their hash values.



When you are investigating a SQL statement, naming it makes it easier to identify. The name appears in reports and chart legends.

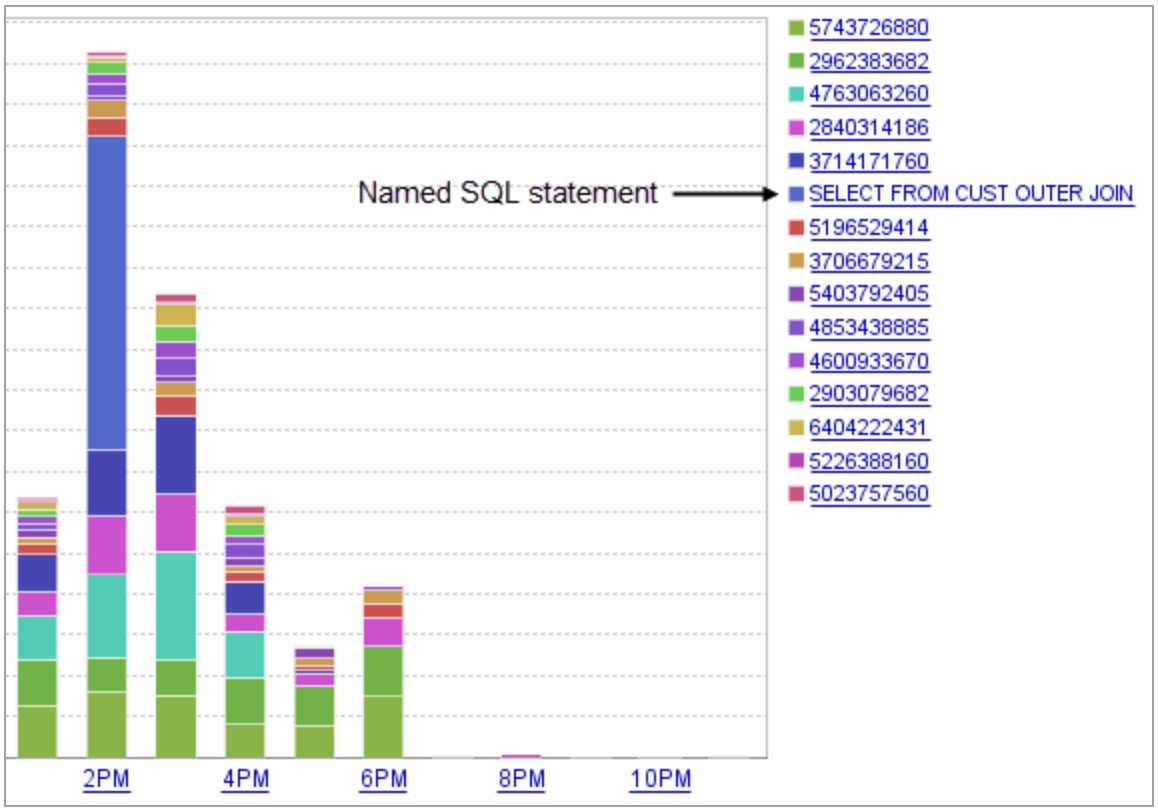
1. In the chart legend, click the hash value that represents the SQL statement.

   The Query Details page displays information about the SQL statement.
2. In the upper-right corner, click SQL Properties.
3. In the SQL Properties dialog, enter the name.

4. Click Save.
5. In the upper-left corner of the Query Details page, click Back to go back to the previous page.

   The legend displays the name instead of the hash value.

> The colors that represent the SQL statements on the chart might change after you name the SQL statement.



Exclude SQL statements from DPA

Certain long-running SQL statements might not be candidates for tuning (for example, SQL statements associated with database backups, replication, or data loads). To prevent these statements from dominating Trends charts or producing tuning advisors that are not actionable, you can exclude them from DPA.

Before excluding SQL statements, consider the possible impacts. If an excluded SQL statement begins affecting your database performance, you will not see the issue in DPA because of the exclusion.

Determine which option meets your needs

DPA provides three options for excluding SQL statements. Use this section to determine which option meets your needs, and then see the following sections for implementation instructions.

Summary	Option 1	Option 2	Option 3
Safe and easy	✓		
Excludes the SQL statement from both past and future charts and advisors	✓		
If the exclusion is reverted, DPA charts and advisors for the exclusion period show data about the excluded statement again	✓		
Excludes SQL statements from all DPA views and analysis, including reports and anomaly detection		✓	✓
Exclusion criteria is not restricted to the hash value or ID of one SQL statement			✓

Option 1: Exclude a specific SQL statement from DPA charts and analysis

Use the SQL Properties dialog box to exclude the selected SQL statement from DPA Trends charts, DPA tuning advisors, or both. When you exclude a statement from tuning advisors, DPA does not generate query advisors for it or consider it when generating table tuning advisors.

DPA recommends using this method if possible because it is safe (there is no risk of losing data) and easy (it is done through the interface).

Data about the statement is **excluded** from:	• All DPA Trends charts that represent one or more days, including charts that represent previous periods • All tuning advisors, including those generated for previous periods
Data about the statement is still **included** in:	• Charts that represent less than one day • Reports • Anomaly detection ⓘ If the statement runs on a regular schedule with a predictable amount of wait time, it would not cause DPA to detect an anomaly during the period when it runs. Higher wait times would be normal during that period.
If you **revert** the exclusion:	DPA charts and advisors for the exclusion period show data about the excluded statement again. With this method, DPA continues to collect and store data about the SQL statement, and so the data is available.

Option 2: Prevent DPA from storing data about a specific SQL statement

Run a statement against the DPA repository database that prevents DPA from storing the data that it collects about the specified SQL statement. The excluded SQL statement is identified by its DPA hash value.

This option for excluding SQL statements requires admin privileges.

Data about the statement is **excluded** from:	• Charts, tuning advisors, and reports that represent time periods **after** you ran the SQL statement • Anomaly detection
Data about the statement is still **included** in:	Charts, tuning advisors, and reports that represent periods **before** you ran the statement
If you **revert** the exclusion:	DPA charts and advisors for the exclusion period do not show data about the statement because that data is not stored in the DPA repository database.

Option 3: Exclude SQL statements from collection based on criteria in the WHERE clause

Modify the WHERE clause of the DPA quickpoll query to prevent DPA from collecting information about SQL statements that meet certain criteria. With this method, you can exclude statements that come from a certain program, user, host, or a combination of factors. For example, you can exclude a specific SQL statement when it is run by a certain user, or you can exclude all SQL statements coming from a user on a certain computer.

ⓘ This method cannot be used for statements that run on a Sybase monitored database instance.

⚠ In some cases, this method can improve the performance of the DPA quickpoll query on a busy database instance. However, adding too much logic to the WHERE clause can cause the query to run longer than expected or disrupt data collection.

If you use this option, DPA strongly recommends working with SolarWinds Support. Check CONTIME entries for QUICKPOLL_EXECUTE both before and immediately after you apply the change to determine if there is any difference in performance.

Data about the statement is **excluded** from:	• Charts, tuning advisors, and reports that represent time periods **after** you modified the WHERE clause • Anomaly detection
Data about the statement is still **included** in:	Charts, tuning advisors, and reports that represent periods **before** you modified the WHERE clause

If you **revert** the exclusion:	DPA charts and advisors for the exclusion period do not show data about the statement because that data was never collected.

Option 1: Exclude a specific SQL statement from DPA charts and analysis

1. In any chart legend, click the name or hash value that represents the SQL statement.

   The Query Details page displays information about the SQL statement.

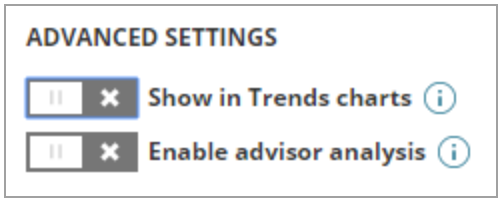
2. In the upper-right corner, click SQL Properties.

   The SQL Properties dialog opens.

3. Under Advanced Settings, clear one or both of the following options:

   - Clear the Show in Trends charts setting to remove the SQL statement from multi-day or one-day Trends charts. If you drill in to a time period less than one day, charts include the SQL statement.
   - Clear the Enable advisor analysis setting to exclude this statement from the analysis that DPA runs to generate query advisors and table tuning advisors. When analysis is disabled, DPA does not detect problems with the SQL statement.

     When you clear the Show in Trends charts setting, both options are cleared. DPA does not perform analysis on SQL statements that are not shown in the Trends charts.

4. Click Save.

You can add excluded SQL statements back to Trends charts and analysis at any time.

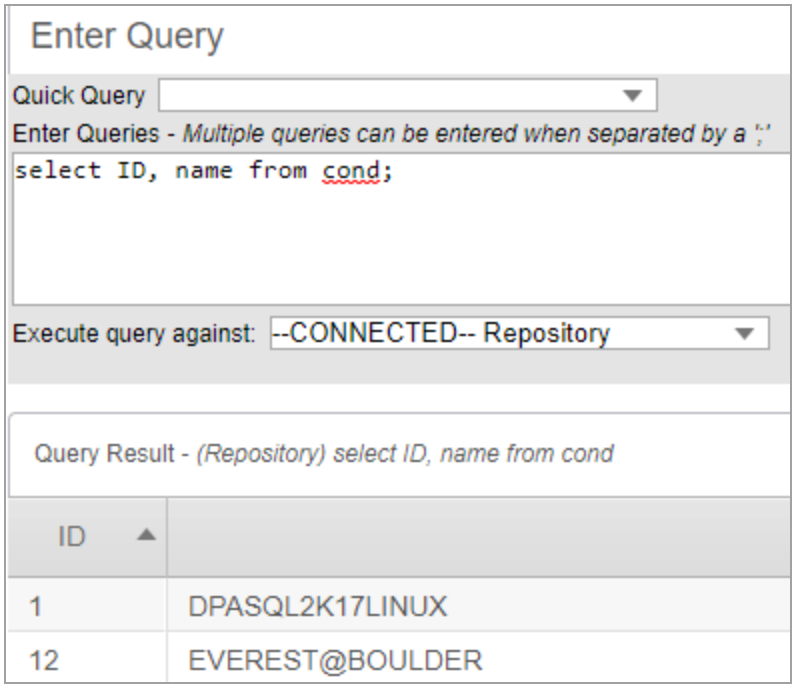
Option 2: Prevent DPA from storing data about a specific SQL statement

1. Log in to DPA using an account with admin privileges.
2. Get the hash value that identifies the SQL statement:
   a. In any DPA chart legend, click the name or hash value that represents the SQL statement.

      The Query Details page displays information about the SQL statement.
   b. In the upper-right corner, click SQL Properties.
   c. Copy the hash value from the top of the SQL Properties dialog box.
3. Open the DB query tool in DPA:
   a. From the DPA menu in the upper-right corner, click Options.
   b. Under Support > Utilities, click DB Query tool.
4. To get the database ID, enter the following query and click Execute Query:

   ```
   select ID, name from cond;
   ```

   The query returns the names and IDs of all monitored database instances.



5. Enter the following query (replacing the variables with your database ID and SQL hash value), and click Execute Query:

   ```
   INSERT INTO con_qp_exclude (dbid, type, value, origin) VALUES (databaseID,
   'H', 'sqlHash', 'U');
   ```

All future data collection excludes this SQL statement. Past data is not purged, so DPA charts and reports that represent previous time periods will still include the SQL statement.

To revert the exclusion, remove the SQL statement from the `con_qp_exclude` table by issuing a `DELETE FROM` statement:

```
DELETE FROM con_qp_exclude WHERE dbid=databaseID AND type='H' AND
value='sqlHash';
```

Option 3: Exclude SQL statements from collection based on criteria in the WHERE clause

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Configuration, click Advanced Options.

   The System Options tab lists options that apply to all database instances.
3. Click DB Instance Options and select the database instance on which the SQL statements run.
4. Select Support Options.
5. Click the name of the QUICKPOLL_WHERE_CLAUSE to open the Edit Option dialog.
6. Enter the phrase to include in the WHERE clause that specifies the SQL statements that should not be collected. Use the syntax appropriate for the database type. See the examples in the following sections.
7. Click Update, and then restart DPA.

   All future data collection excludes the SQL statements. Past data is not purged, so DPA charts and reports that represent previous time periods will still include the statements.

To revert the exclusion, repeat this procedure to remove the criteria from the quickpoll WHERE clause.

Examples for SQL Server

Example 1: Exclude all SQL statements from the TSQL program logging in from the server HPSERVER:

```
and not (s.program_name='TSQL' and s.hostname='HPSERVER')
```

Example 2: Exclude the specified SQL statement if it comes from a certain user, but do not exclude it if it comes from other users. Use the SQL_handle (not the DPA hash value) to identify the SQL statement.

```
and not (s.loginame='Bob' and s.sql_handle=0x00987097097897)
```

Example 3: Exclude all SQL statements executed in the master database by the dataload program:

```
and not (db_name(s.dbid)='master' and s.program_name='dataload')
```

Examples for Oracle

Example 1: Exclude all SQL statements from the SAP.exe program logging in from the server HPSERVER.

```
and not ("u".ksusepnm ='SAP.exe' and "u".ksusemnm='HPSERVER')
```

Example 2: Exclude the specified SQL statement if it comes from a certain user, but do not exclude it if it comes from other users.

```
and not ("u".ksuudlna='Bob' and "u".ksusesqh =97097897)
```

Examples for Db2

Example 1: Exclude all SQL statements from the SAP.exe program logging in from the server HPSERVER:

```
and not (ai.appl_name ='SAP.exe' and ai.client_nname='HPSERVER')
```

Example 2: Exclude a dynamic SQL statement if it comes from a certain user, but do not exclude it if it comes from other users.

```
and not (ai.auth_id='Bob' and s.stmt_text like '%insert into bad_table%')
```

Example 3: Exclude a static SQL statement if it comes from a certain user, but do not exclude it if it comes from other users.

```
and not (ai.auth_id='Bob' and st.text like '%insert into bad_table%')
```

Examples for MySQL

Example 1: Exclude all SQL statements from the SAP.exe program logging in from the server HPSERVER:

```
and not (program_name = 'SAP.exe' AND host='HPSERVER')
```

Example 2: Exclude a dynamic SQL statement if it comes from a certain user, but do not exclude it if it comes from other users:

```
and not (user = 'BOB' AND statement_sql like '%insert into bad_table%')
```

Examples for PostgreSQL

Example 1: Exclude the queries coming from a certain client address:

```
and a.client_addr != '10.140.66.28'
```

Example 2: Exclude a wait event type timeout that is coming from a client backend:

```
and a.wait_event_type != 'Timeout' and a.backend_type != 'client backend'
```

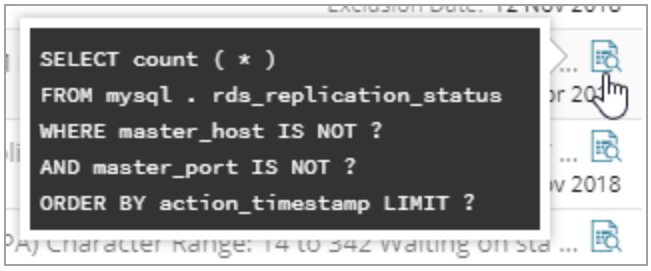
Add excluded SQL statements back to DPA Trends charts and analysis

If you excluded SQL statements from DPA Trends charts and analysis, you can add them back if needed.

Add excluded SQL statements back to Trends charts

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Configuration, click Excluded SQL Statements.

   The Excluded SQL Statements dialog box lists the SQL statements that are excluded from Trends charts.
3. Locate the SQL statement in the list:
   - Use the drop-down menu at the top to sort by exclusion date, database instance name, or SQL ID (name or hash).
   - Enter a string in the Search bar to show only SQL statements whose ID, database instance, or database type includes the search string.
   - Hold the mouse pointer over the icon to display the SQL.



4. Select the SQL statement.
5. Click Re-include Selected.
6. Click the x in the upper-right corner to close the dialog box.

Add excluded SQL statements back to analysis

1. If the SQL statement is currently excluded from Trends charts, add it back to Trends charts.
2. In any chart legend, click the name or hash value that represents the SQL statement.

   The Query Details page displays information about the SQL statement.
3. In the upper-right corner, click SQL Properties.

   The SQL Properties dialog opens.
4. Under Advanced Settings, select Enable advisor analysis.
5. Click Save.

Resource metrics in DPA

Use the resource metrics in DPA to monitor the health of your database and to correlate contention for resources with increases in database wait times.

View resource metrics in DPA

Resource metrics provide information about how resources (such as CPU, disk, and memory) are being used at specific points in time. These metrics show what was happening in the rest of your environment during database slow-downs, and can provide context to help you identify the root cause of performance problems.

DPA displays resource metrics in the following locations. You can:

- View all available metrics on the Resources tab
- Correlate wait time with resource metrics on the Trends tab
- View resource metrics related to the performance of a query

View all available metrics on the Resources tab

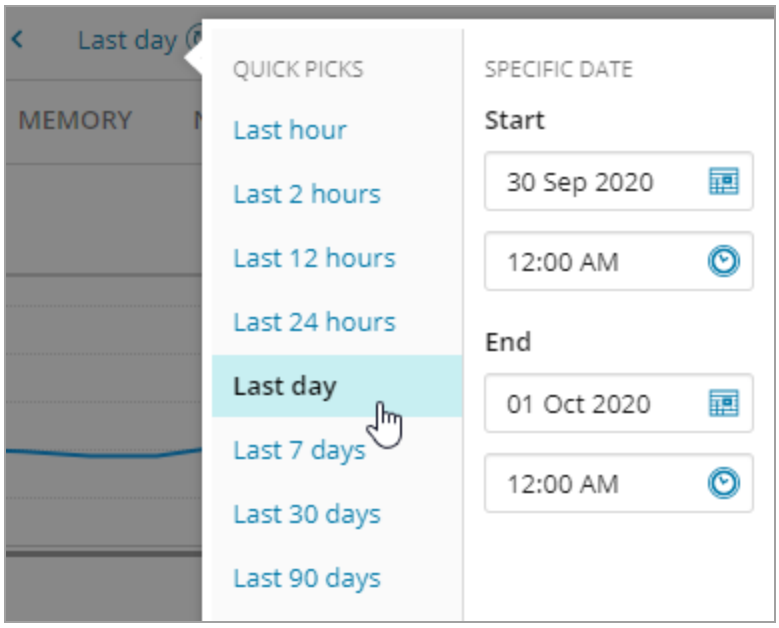
The Resources tab displays all available resource metrics for the selected database instance.

1. On the DPA home page, click a database instance to view detailed information.
2. In the upper-right corner, click the Resources tab.

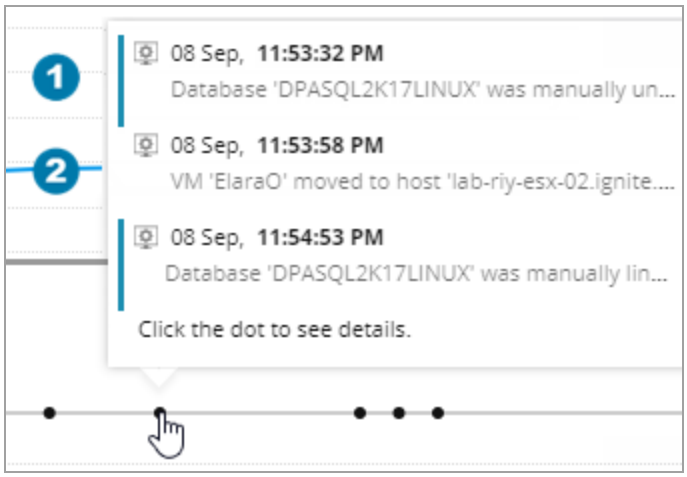
   The Resources tab displays all available resource metrics for the selected database instance. By default, charts show data for the last hour.

   - Click ⓘ next to a metric to display information about that metric.
   - Click ⚙ to view or change the thresholds for that metric.

3. To change the time range, click the date range at the top of the page to open the date picker. Then select a predefined time period or enter specific dates.



4. To view annotations or information about VMware events (for instances that run on a VM), hover over a dot on the line above a chart. These events can sometimes explain changes in resource metrics.



1	VMware events that occurred on the same VM have a blue line beside them.
2	VMware events that occurred on a different VM running on the same host do not have a line beside them.

You can choose which VMware events to display on charts.

Correlate wait time with resource metrics on the Trends tab

When you are viewing wait time charts on the Trends tab, you can scroll down to determine if unexpectedly long wait times correlate with resource contention.

1. On the DPA home page, click a database instance to view detailed information.
2. In the upper-right corner, click the Trends tab.
3. Scroll down and click the Resources tab below the wait time charts.

   The Resources tab displays a subset of the available resource metrics for the selected database instance. You can:

   - Click Add Resource Chart to include additional charts.
   - Click ⓘ to display information about a metric.
   - Click ⚙ to view or change the thresholds for that metric.
   - Use other icons to the right of a chart to remove it, change its location, or replace it with a different chart.
4. To view information about VMware events that might explain changes in behavior, hover over the blue or gray arrows at the top of a chart.

   Blue arrows indicate that events occurred on the VM where the database instance runs. Gray arrows indicate that events occurred on other VMs that run on the same host.

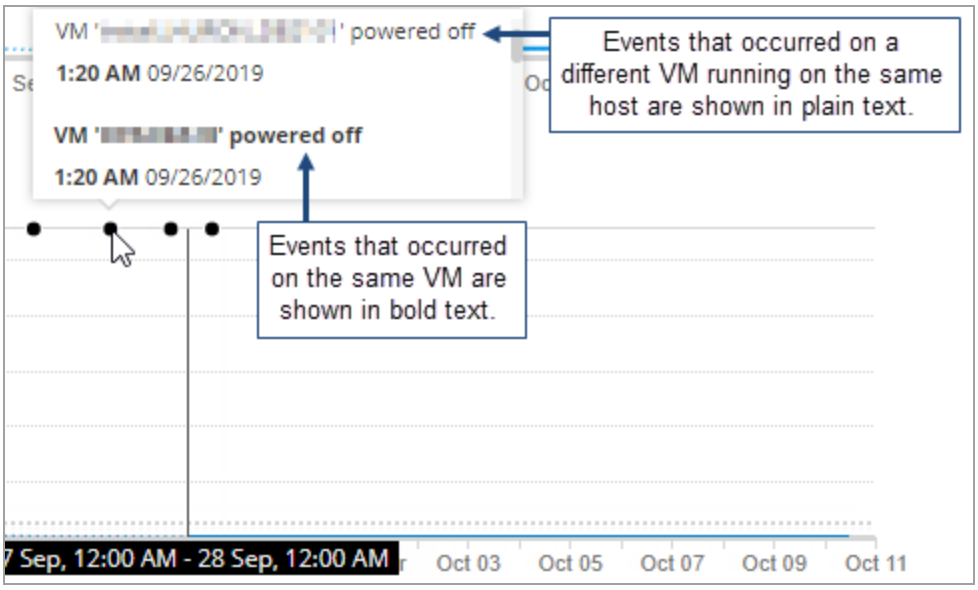
   When you hover over an arrow, events that occurred on other VMs are in shown in plain text. Those that occurred on the same VM are highlighted in bold, blue text.

   ⓘ You can choose which VMware events to display on charts.

View resource metrics related to the performance of a query

To help you find the root cause of long wait times for a query, the Query Details page includes the most relevant statistics, blocking, plan, and metrics charts. When you scroll down to view these charts, the Top Waits chart at the top of the page remains visible so you can correlate query wait times with other events during the same time period.

To view annotations or information about VMware events (for instances that run on a VM), hover over a dot on the line above a chart. These events can sometimes explain changes in resource metrics. VMware events that occurred on other VMs are shown in plain text. Those that occurred on the same VM are shown in bold text.



You can choose which VMware events to display on charts.

About DPA resource metric baselines

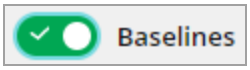
When you are viewing resource metrics on the Resources tab in DPA, you can display baselines to compare values from a specific period to historical norms. Baselines provide context for the current values. Metric values that are far above or below the baseline could indicate areas in need of tuning or reconfiguration.

Monitoring must be active for at least one day before baselines can be calculated, and baselines become more representative as more monitoring days pass.

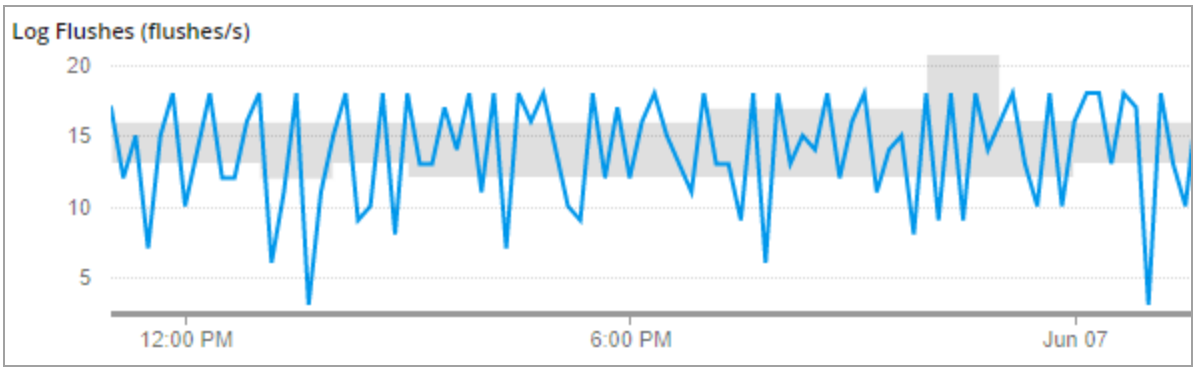
Baselines are not available for metrics collected for the VM Option.

Show or hide baselines

Baselines are available when the selected time period is one week or less. By default, baselines are not shown. Click the Baselines toggle switch in the upper-right corner of the Resources tab to show or hide baselines.



When you show baselines (and the time period is one week or less), a shaded area indicates the historic values between the 10th and 90th percentiles.



How are baselines calculated?

Baselines are calculated for each one-hour period. By default, baselines are calculated using data only from weekdays (Monday through Friday). Each baseline is calculated using data from the corresponding hour for all weekdays, so the value for a specific hour is the same across all days. (For example, the value for 1 - 2 PM is the same Monday through Friday.)

Baselines are calculated using historical data from **before** the earliest time shown on the chart. For example, if a chart covers one week and starts on May 10th, all baselines are calculated using data from May 9th and earlier. For this reason, one-week charts show repeating patterns for each day.

Change the days included in baseline calculations

To change the days included in baseline calculations, edit the advanced option `METRICS_BASELINE_TYPICAL_HOUR_CALCULATION`. This option can be set globally or for a specific monitored database instance.

Choose one of the following values:

Value	Description
Weekday Only (M-F)	Baselines are computed for each one-hour period using data from the corresponding hour on weekdays (Monday through Friday). This is the default.
All Days of the Week	Baselines are computed for each one-hour period using data from the corresponding hour on all days.
Same Day of Week	Baselines are computed for each one-hour period using data from the corresponding hour on the corresponding day. (For example, the value for 1 - 2 PM on Monday uses data from the corresponding hour on Mondays, and is therefore different than the value for 1 - 2 PM on Friday.) Be aware that this option increases the number of baselines per metric from 24 to 168.

View or change DPA resource metric thresholds

Resource metric charts in DPA indicate when the metric has exceeded a Warning or Critical threshold. You can create alerts based on resource metric thresholds.

You can change the default thresholds to meet the needs of your environment. If a metric does not have default thresholds, you can add them. The custom thresholds can apply to a specific database instance or all monitored instances.

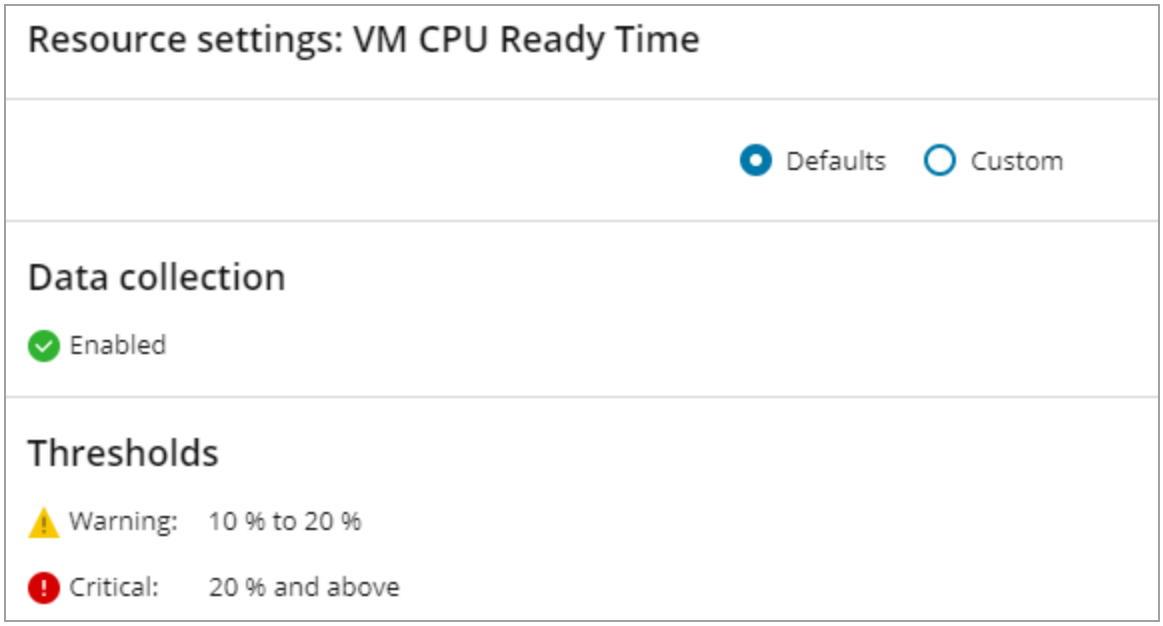
View the current thresholds

1. On the DPA home page, click the database instance whose resource metric thresholds you want to view.

   > If you are going to change the default thresholds for all database instances, you can click any instance.

2. Click the Resources tab.
3. Click the tab that displays the metric whose thresholds you want to view or change.
4. Locate the metric chart and click ⚙ to the right of the chart.

   The Resource Settings page displays the thresholds that are currently used for this metric.



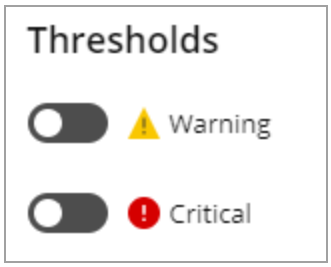
Change the thresholds

1. Select Custom.

   If the metric has default thresholds, the Custom dialog box initially shows the default values.

   If the metric does not have default thresholds, the Custom dialog box shows the Warning and Critical thresholds as disabled.

2. Enter the new threshold values:

   - If the metric does not have default thresholds, click the toggle switch next to each threshold level you want to enable.
   - If you enable both the Warning and Critical threshold levels, enter the same value at the intersection of the two levels:
     - For metrics that alert on **higher** values, enter the same number as the maximum Warning value and the minimum Critical value.

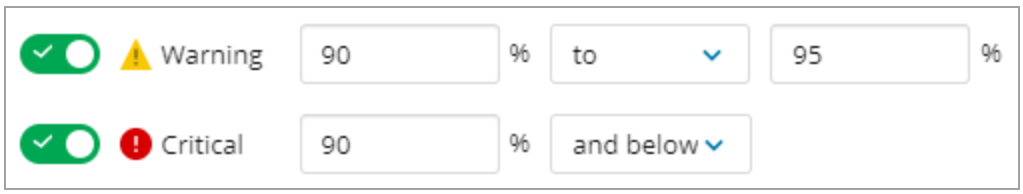
       DPA issues a Warning alert when the value is within the Warning range (inclusive). DPA issues a Critical alert when the value is greater than the minimum Critical threshold.

       Example: For the thresholds shown in step 1, DPA issues a Warning alert when the value is 10 through 20. DPA issues a Critical alert when the value is greater than 20.

     - For metrics that alert on **lower** values, enter the same number as the minimum Warning value and the maximum Critical value.

DPA issues a Warning alert when the value is within the Warning range (inclusive). DPA issues a Critical alert when the value is less than the minimum Critical threshold.

Example: For the thresholds shown below, DPA issues a Warning alert when the value is 90 through 95. DPA issues a Critical alert when the value is less than 90.



3. Do one of the following:
    - To use the new values only for **this** database instance, click Save.
    - To use the new values for **all** database instances, click Save As Default.

      When you Save As Default, the new default threshold values are used for all database instances unless custom thresholds have been specified for an instance. Any database instance with custom thresholds will continue to use those thresholds.

Show or hide VMware events on metric charts

For database instances that run on virtual machines (VMs), resource metric charts display VMware events. By default, the charts display events that occur on the VM where the database instance runs, as well as events on all other VMs that run on the same host. These events can sometimes explain changes in resource behavior.

- Events on the VM where the database instance runs are shown in bold text.
- Events on a different VM are shown in plain text.

You can change which VMware events DPA displays on metric charts. These settings apply when you open DPA on the current computer using the same type of browser (for example, Chrome).

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Display, click Display Options.
3. From the View Events drop-down menu, select which VMware events you want DPA to display on metric charts:
    - None
    - For this VM
    - For all VMs on host

Exclude SQL Server databases from backup metrics and backup alerts

DPA includes metrics and Administrative alerts for SQL Server instances to help you ensure that critical data is being backed up on schedule. However, some databases in a SQL Server instance might not need to be backed up on a regular basis (for example, databases used only for testing).

If a database has never been backed up, DPA automatically excludes that database when it calculates metric and alert values. To avoid displaying misleading metric values or triggering unnecessary alerts, you can exclude other databases that do not need to be tracked:

- **Metrics**: To specify which databases DPA tracks when it calculates backup metric values, set the advanced option `TRACK_BACKUPS_FOR_DBS`, as described below.
- **Alerts**: To specify which databases DPA tracks when it evaluates backup alerts, you can define a list within the alert definition or choose to use the advanced option `TRACK_BACKUPS_FOR_DBS`.

To set the advanced option `TRACK_BACKUPS_FOR_DBS`:

1. Open the Advanced Options page.
2. Click DB Instance Options, and select a SQL Server database instance.
3. Click `TRACK_BACKUPS_FOR_DBS` to open the Edit Option dialog.

   The default value is `ALL`, which tracks backups for all databases that were previously backed up.
4. To change the value, enter one of the following keywords followed by a comma-separated list of database names enclosed in parentheses:
   - `INCLUDE`, followed by a list of all databases in the instance whose backups DPA should track. For example:

     ```
     INCLUDE: (DB1, DB2, DB3)
     ```
   - `EXCLUDE`, followed by a list of all databases in the instance whose backups DPA should **not** track. For example:

     ```
     EXCLUDE: (DB4, DB5)
     ```
5. Click Update.

Disable the collection of resource metric data

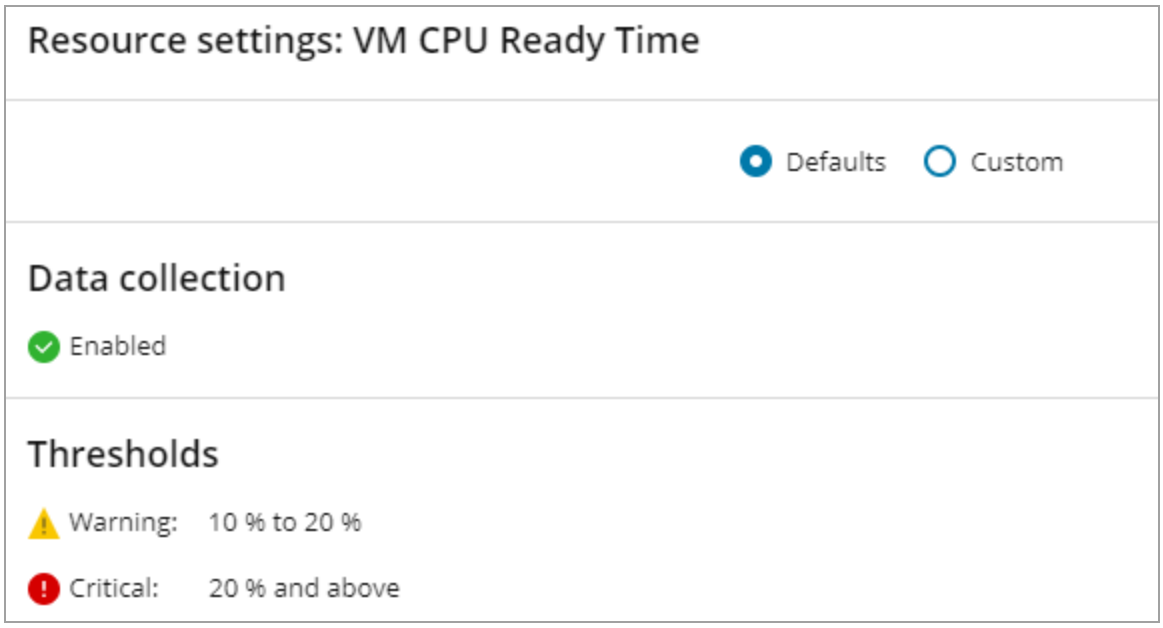
You can prevent DPA from collecting data for a resource metric. You can disable data collection for a specific database instance or for all monitored instances.

1. On the DPA home page, click the database instance.

   > If you are going to disable data collection for all database instances, you can click any instance.

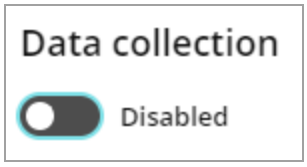
2. Click the Resources tab.
3. Click the tab that displays the metric that you do not want to collect data for.
4. Locate the metric chart and click to the right of the chart.

   The Resource Settings page displays the settings for the metric.



5. Click Custom.
6. Under Data collection, click the Enabled toggle switch to deselect it.

   The switch turns black and the label changes to Disabled.



7. Do one of the following:
   - To disable data collection for **this** database instance, click Save.
   - To disable data collection for **all** database instances, click Save As Default.

Create and manage custom resource metrics in DPA

DPA provides a set of default metrics, but you can also add custom metrics to track other values that are meaningful to your organization.

Types of custom metrics

You can create two types of custom metrics:

- Query-based custom metrics use a query to determine the metric value.
- Custom wait time metrics track a specific wait type or event.

Query-based custom metrics

For query-based custom metrics, the metric definition includes a query that returns a positive integer. The metric chart shows one of the following values:

- The value returned by the query
- The delta between values returned by subsequent query executions
- The rate of change between query executions
- The time it took to run the query

In addition to the query, the query definition also specifies how often the SQL runs, the units, and which tab on the Resources page displays the metric (either a default tab or a custom tab). You can also specify default thresholds. If thresholds are set, users can create resource alerts on these custom metrics just as they can on default metrics.

> To prevent unauthorized users from entering malicious SQL in a custom metric, you can configure password protection for this feature.

Examples of custom query-based metrics

Query-based custom metrics are flexible and can be used to track a wide variety of data. You can find examples of custom metrics that DPA users have created and shared through the DPA Content Exchange in THWACK. Some examples of how custom metrics can be used include:

- Track the fragmentation of a heavily used index. Fragmentation causes index data to become out of order on the disk, with gaps between index data. For large tables this can cause slow performance when the index is read using a scan operation.
- Monitor the number of buffer manager page reads per second in SQL Server database instances. If this value is high, the buffer manager is doing a lot of work flushing out old pages and reading in new pages, which can affect performance.

- Monitor the number of page splits occurring in SQL Server instances. Page splits occur when new data is being added to a page, and that page lacks the space to store it. A large number of page splits can affect performance.

Custom wait time metrics

Custom wait time metrics track the amount of time a monitored database instance spends on a specific wait type or event. DPA gets the data shown on these metric charts from the DPA repository database. The metric definition specifies the wait type or event, but it does not specify the SQL.

Some options that are available for query-based custom metrics cannot be set for custom wait time metrics:

- You cannot select which tab displays a custom wait time metric. These metrics are shown on the Waits tab, in alphabetical order below the default Total Instance Wait Time metric.
- The unit is seconds.
- Thresholds cannot be set.
- Frequency and timeout cannot be set.

Create a query-based custom metric

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Report, Metrics & Alerts, click Custom Metrics.

   The Custom Metrics page lists any existing custom metrics.
3. Click Create new custom metric.

   The Custom Resource Metric Configuration page opens.
4. Enter the Metric Properties:
   a. Leave Enabled selected if you want DPA to display the metric chart and collect data for this metric.

      If you disable the metric, the metric chart is not shown on the Resources page and DPA does not collect metric data. The metric definition is available and you can enable it later.
   b. Select the database type that the metric applies to.
   c. If the SQL to collect data for this metric cannot be run on all versions of the selected database type, open the Database Versions drop-down menu and choose Limit versions. Then specify one of the following:

- The minimum version and above
- The maximum version and below
- A range of versions (inclusive)

> The SQL Server version number does not always match the advertised version. For example, SQL Server 2017 has a version number of 14. For information about finding the version number, see How do I find SQL Server version.

d. As the Display Name, enter a name to identify the metric chart.

e. (Optional) Enter a description to be shown as a tooltip.

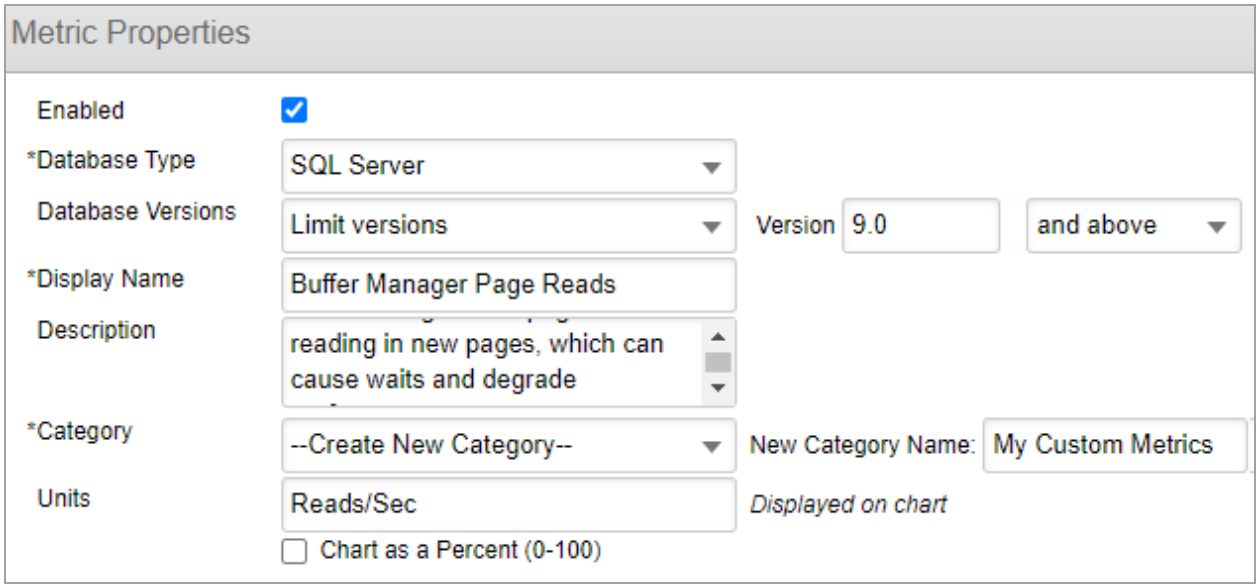
f. In the Category drop-down menu, specify which tab on the Resources page displays the metric chart:

- To add the metric to an existing tab, choose a category from the menu.
- To create a new category, choose Create New Category. Then enter the category name.

> When at least one enabled metric is assigned to a custom category, the Resources page displays a tab for the custom category. If all metrics assigned to a custom category are disabled, the tab is not shown.

g. If necessary, enter information in the Units text box to clarify the unit of measurement (for example, *MB*, *%*, *KBs/s*, or *days*). This information is shown in parentheses after the display name.

h. If the metric is a percentage, select Chart as a Percent.

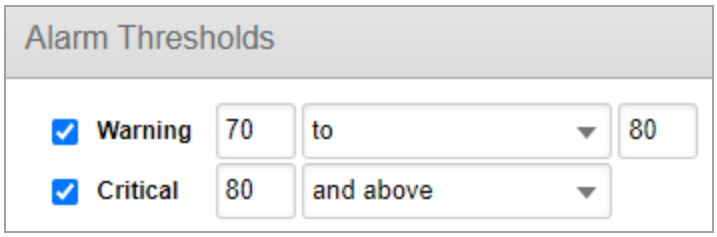


5. (Optional) If there are industry-standard or organizational thresholds that this metric value should not cross, specify the default thresholds in the Alarm Thresholds section.

    a. Select the levels you want to enable.

    b. For each level specify either the minimum value and above, the maximum value and below, or a range.

    If you enable both the Warning and Critical threshold levels, enter the same value at the intersection of the two levels (as you would with thresholds for default metrics).

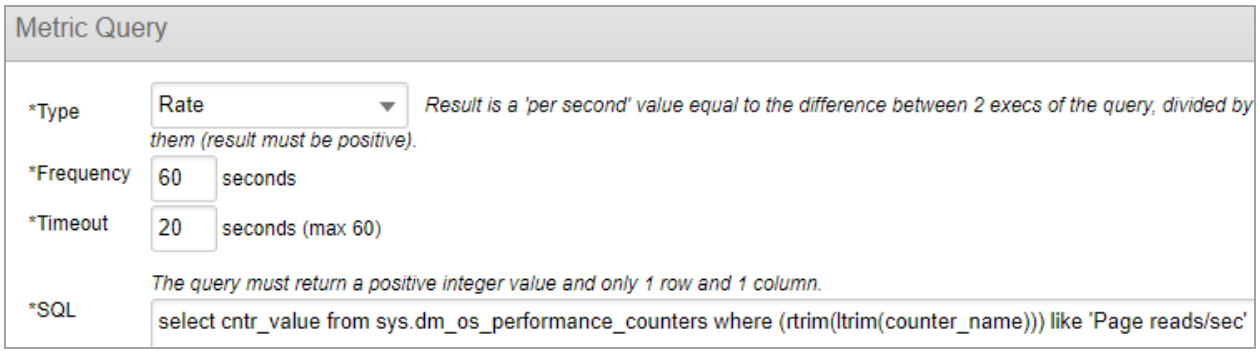


The resource metric chart indicates when these thresholds are violated. You can also create alerts on the metric based on the thresholds. The default thresholds can be overridden globally or for a specific database instance.

6. In the Metric Query section, define the query to run and the results to display:
    a. From the Type drop-down menu, select a type to specify how DPA determines the metric value.

Single Value	The metric value is the value returned by the query. DPA does not perform any calculations.
Delta	The metric value is the difference between the values returned for the latest query execution and the previous execution: `Metric_Value = Value_for_Latest_Exec - Value_for_Previous_Exec` The result must be positive. Use the Delta type for metric values that continue to go up, typically until the server is rebooted.
Rate Calculation	The metric value is the difference between the values returned for the latest query execution and the previous execution, divided by the number of seconds between query executions: `Metric_Value = (Value_for_Latest_Exec - Value_for_Previous_Exec) / Execution_Interval` The result must be positive. Use this type to show the rate of increase per second.
Timed Query	The metric value is the time in milliseconds that it took to run the query.

    b. In the Frequency text box, enter the number of seconds between query executions.
    c. In the Timeout text box, enter the number of seconds after a query execution that DPA should wait for results before it times out.
    d. In the SQL text box, enter the query to execute. The query must:
        - Contain valid SQL for the selected database type.
        - Return a positive integer.
        - Return only **one** value.



7. Test and save the metric:
    a. Click Test Metric.
    b. Select the monitored instance to test on. For Delta and Rate Calculation metrics, specify the interval to use for the test. Then click OK.

       DPA runs the SQL (twice for Delta and Rate Calculation metrics) and displays the results. If errors occur, correct the SQL and retest.
    c. Click OK to close the results dialog box.
    d. Click Save to save the metric definition, and then click OK at the confirmation dialog box.

Create a custom wait time metric

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Report, Metrics & Alerts, click Custom Metrics.

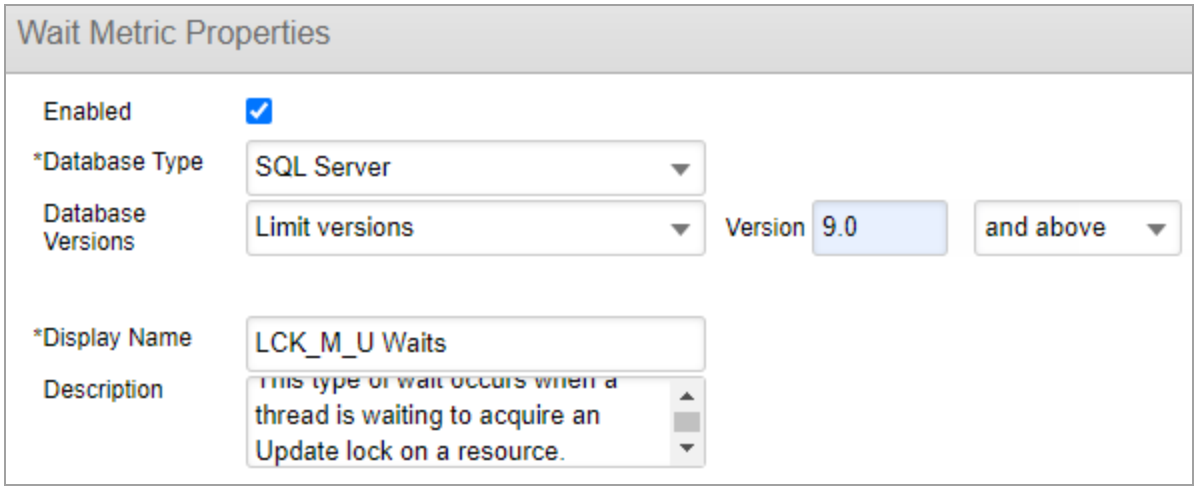
   The Custom Metrics page lists any existing custom metrics.
3. Click Create new wait time metric.

   The Wait Time Metric Configuration page opens.
4. Leave Enabled selected if you want DPA to display the metric chart.

   If you disable the metric, the metric chart is not shown on the Waits tab of the Resources page. The metric definition is available and you can enable it later.
5. Select the database type that experiences the wait type or event.
6. If all versions of the selected database type do not experience the wait type or event, open the Database Versions drop-down menu and choose Limit versions. Then specify one of the following:

- The minimum version and above
- The maximum version and below
- A range of versions (inclusive)

7. As the Display Name, enter a name to identify the metric chart.
8. (Optional) Enter a description to be shown as a tooltip.



9. In the Wait Name text box, identify the wait type or event.

> DPA does **not** verify that the Wait Name entry is valid. If the entry does not represent a valid wait type, or if the wait type does not occur on the selected database type, the metric chart always shows 0.

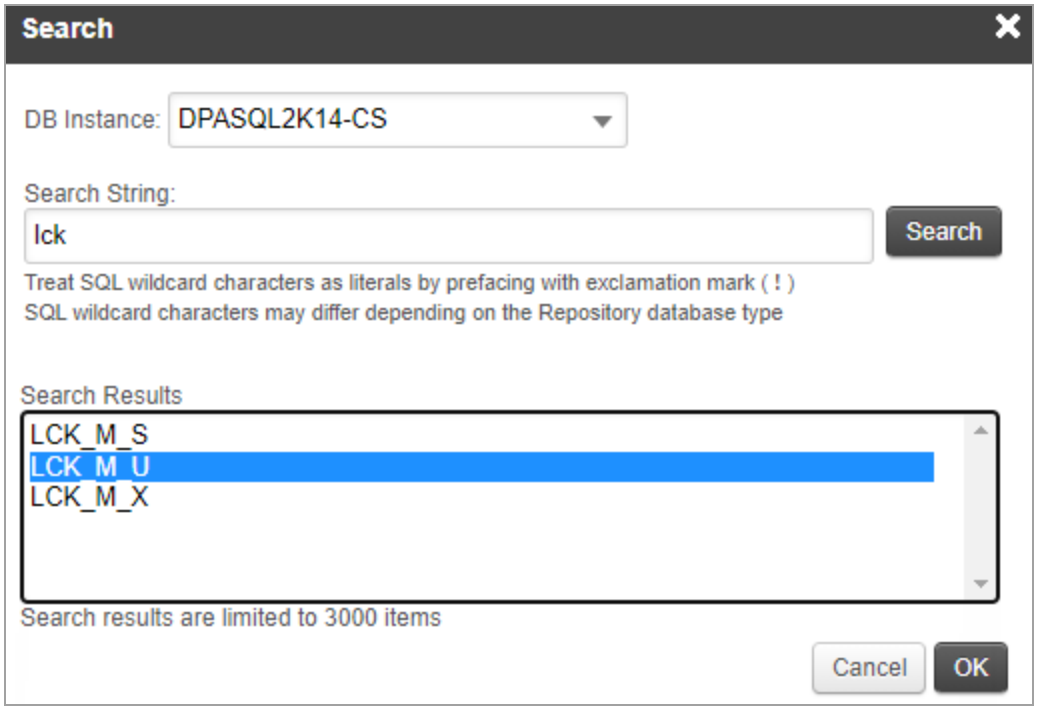
To search for a wait type:

a. Click Search next to the Wait Name text box.

b. Verify that the selected database is the correct type and version.

c. Enter part of the wait type name as a search string, and click Search.

d. Select an item from the Search Results and click OK.



10. Test and save the metric:

    a. Click Test Metric.

    b. Select the monitored instance to test on and click OK.

    DPA shows how much time the selected database instance spent on that wait type.

    c. Click OK to close the results dialog box.

    d. Click Save to save the metric definition, and then click OK at the confirmation dialog box.

Edit a custom metric definition

For query-based custom metrics, you can edit any values in the metric definition. For custom wait time metrics, you **cannot** change the Database Type or Wait Name values after the metric definition has been saved. If those values need to be changed, create a new wait time metric.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Report, Metrics & Alerts, click Custom Metrics.

   The Custom Metrics page lists existing custom metrics.
3. Click the name of the metric you want to edit.

The Custom Resource Metric Configuration page opens.

4. Make changes to the metric definition and click Save.

Delete a custom metric

1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Report, Metrics & Alerts, click Custom Metrics.

   The Custom Metrics page lists existing custom metrics.
3. Locate the metric and click the Delete button in that row. Then click Yes at the confirmation prompt.

Metrics collected by DPA

The following topics describe the metrics that DPA collects:

- Oracle metrics collected by DPA
- SQL Server metrics collected by DPA
- MySQL metrics collected by DPA
- Sybase metrics collected by DPA
- Db2 metrics collected by DPA
- Azure SQL database metrics collected by DPA
- ASMI metrics collected by DPA
- PostgreSQL metrics collected by DPA
- VM metrics collected by DPA

Oracle metrics collected by DPA

The following sections list the metrics that DPA collects for Oracle databases. Some metrics are not available for all Oracle deployments.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.
- If the database instance runs on a virtual machine (VM), metrics collected for the VM are described in VM metrics collected by DPA

ASM

Metric	Description
ASM Summary Reads	The total number of all I/O read requests.
ASM Summary Writes	The total number of all I/O write requests.
ASM Summary Read Time	The average I/O time per read request over all disks.
ASM Summary Write Time	The average I/O time per write request over all disks.
ASM Summary Write Rate	The total number of kilobytes written to disk every second.
ASM Summary Read Rate	The total number of kilobytes read from disk every second.

Connections

Metric	Description
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle).
Connected Users	The number of distinct users connected to this instance (even if the connection is idle).
Sessions	The number of sessions connected to this instance (even if the connection is idle).

CPU

Metric	Description
Core Count	The number of cores used by the instance.
CPU Utilization by DB	The percentage of CPU being utilized by the database instance, which is a subset of the CPU utilized by the entire system. Oracle supplies this value only if the database parameter `timed_statistics = TRUE`. If this is high, use DPA Trends charts to review queries waiting on CPU.
O/S CPU Utilization	The percentage of CPU being utilized by the entire system. If this is high, compare this utilization with the CPU Utilization By Oracle metric. If most of the CPU is being used by Oracle, use the DPA Trends charts to review queries waiting on CPU. If Oracle is not using a significant portion of total CPU, review other non-Oracle programs running at this time.

Disk

Metric	Description
DB Commit Time	The average number of milliseconds waiting for the `log file sync` event indicating commit times for this database.
DB Multi Block Disk Read Time	The average number of milliseconds waiting for the `db file scattered read` event in this database. If this is high, contact your system administrator to understand why these disk reads are slow. Use DPA to drill in to the `db file scattered read` waits and use the Files tab to show the disks involved.
DB Physical I/O Rate	The number of kilobytes being read and written to disk every second for this database. If this is high, drill in to the DPA Trends charts and review physical read and write wait events.
DB Physical Read Rate	The number of kilobytes being read from disk every second for this database. If this is high, drill in to the DPA Trends charts and review queries waiting on physical read wait events (for example, `db file scattered read` or `db file sequential read`).
DB Physical Write Rate	The number of kilobytes being written to disk every second for this database. If this is high, drill in to the DPA Trends charts and review queries waiting on write wait events (for example, `free buffer waits` or `direct path write temp`).
DB Single Block Disk Read Time	The average number of milliseconds waiting for the `db file sequential read` event in this database. If this is high, contact your system administrator to understand why disk reads are slow. Use DPA to drill in to the `db file sequential read` waits and use the Files tab to show the disks involved.

Exadata

Metric	Description
Cell Multiblock Physical Read Latency	The average number of milliseconds waiting for the `cell multiblock physical read` Exadata event in this database.

Metric	Description
Cell Single Block Physical Read Latency	The average number of milliseconds waiting for the `cell single block physical read` Exadata event in this database.
Cell Smart Table Scan Latency	The average number of milliseconds waiting for the `cell smart table scan` Exadata event in this database.
Flash Cache Hit Ratio	The amount of I/O operations satisfied by the Exadata Smart Flash Cache within the Storage Servers. Exadata Smart Flash Cache is one of the essential technologies of the Oracle Exadata Database Machine that enables the processing of up to 1.5 million random I/O operations per second (IOPS), and the scanning of data within Exadata storage at up to 75 GB/second. This metric helps you understand how much the cache is helping.
IO Saved by Storage Cell Offloading	The amount of physical I/O that has been saved by offloading it to the Exadata storage servers. Each of the storage servers might get a piece of the SQL statement to operate on, so the processing is also parallelized at the same time. This saves valuable database server processing cycles for other non-I/O related activities and can dramatically reduce response times. Smart Scan is another term that essentially means the same thing.
Smart Scan Efficiency	When the storage cells process full table scans they can apply columns filters and perform column projection so that not all blocks are returned to the database server, only the ones that are needed. This metric shows an efficiency of how well that is occurring. The data comes from v$sysstat. It looks at the '`cell IO uncompressed bytes`' (a), '`cell physical IO bytes saved by storage index`' (b) and '`cell physical IO interconnect bytes returned by smart scan`' (c) metrics. It then applies the formula of `100 * (a + b) / c` to get the percentage of data saved by the smart scans.

Memory

Metric	Description
Buffer Cache Hit Ratio	The rate at which this database finds the data blocks it needs in memory rather than having to read from disk. By itself, the buffer cache hit ratio is not very meaningful except for databases with undersized data buffer cache (db_cache_size parameter). Oracle provides the data buffer cache advisory utility v$db_cache_advice for assistance with sizing.

Metric	Description
Buffer Cache Size	The amount of memory allocated to all Oracle buffer caches.
DB Logical Read Rate	The number of memory reads (session logical reads statistic from v$sysstat) per second for this database.
Library Cache Hit Ratio	The library cache (a component of the shared pool) stores the executable (parsed or compiled) form of recently referenced SQL and PL/SQL code. Oracle tries to reuse this code. If the code has been executed previously and can be shared, Oracle will report a library cache hit. If Oracle is unable to use existing code, then a new executable version of the code must be built, which is known as a library cache miss.
PGA Cache Size	The amount of memory allocated to the PGA cache.
Shared Pool Size	The amount of memory allocated to the Oracle shared pool.

Network

Metric	Description
DB Round-trip Time	The round-trip time when running "select 1 from dual" (includes network time but not connect time) on this database. If this is high, contact your network administrator to understand network latency.
SQL*Net Received Rate	The throughput of SQL*Net bytes received from the clients in KB/second.
SQL*Net Sent Rate	The throughput of SQL*Net bytes sent to the clients in KB/second.

RAC

Metric	Description
Avg Current Block Flush Time	The average current block flush times being experienced from this instance across the RAC Interconnect. This value is calculated as follows: `(gc_current_block_flush_time * 10) / (gc_current_blocks_served)`
Avg Current Block Pin Time	The average current block pin times being experienced from this instance across the RAC Interconnect. This value is calculated as follows: `(gc_current_block_pin_time * 10) / gc_current_blocks_served as average_pin_time`
Avg Current Block Send Time	The average current block send times being experienced from this instance across the RAC Interconnect. This value is calculated as follows: `(gc_current_block_send_time * 10) / gc_current_blocks_served as average_send_time`
Avg GC CR Block Build Time	The average global cache CR block build times being experienced from this instance across the RAC Interconnect. The average time to build a consistent read block is calculated as follows: `(gc cr block build time * 10)/(gc cr blocks served)`
Avg GC CR Block Flush Time	The average global cache CR block flush times being experienced from this instance across the RAC Interconnect. The average time spent waiting for a redo log flush is calculated as follows: `(gc cr block flush time * 10)/(gc cr blocks served)`
Avg GC CR Block Receive Time	The average round-trip time or latency for all requests for a Consistent Read (CR) from this instance across the RAC Interconnect. If the transfer time is too high, or if one of the nodes in the cluster shows excessive transfer times, the RAC interconnect should be checked (using system level commands) to verify that it is functioning correctly. Calculation in (ms) is as follows: `(gc_current_block_receive_time)/(gc_cr_blocks_received) * 10`
Avg GC CR Block Send Time	The average global cache CR block send times being experienced from this instance across the RAC Interconnect. The average time to send a complete consistent read block is calculated as follows: `(gc cr block send time * 10)/(gc cr blocks served)`

Metric	Description
Avg GC Current Block Receive Time	The average round-trip time or latency for all processing requests for Current Mode Block from this instance across the RAC Interconnect. Calculation in (ms) is as follows: `(gc_current_block_receive_time)/(gc_current_block_receive_time) * 10`
Current Block Service Time	The average Current Block Service Time (ms) is calculated as follows: `(gc current block pin time)+(gc current block flush time)+(gc current block send time)/(gc current blocks served) * 10`
LMS Service Time	The average LMS Service Time measures overall latency for a Consistent Read. This includes queue, build, flush, and send time. The Lock Manager Server (LMS) process, also called the GCS (Global Cache Services) process, is used to transport blocks across the nodes for cache-fusion requests. If there is a Consistent Read request, the LMS process rolls back the block, makes a Consistent Read image of the block, and then ships this block across the HSI (High Speed Interconnect) to the process requesting from a remote node. LMS must also check constantly with the LMD background process (or GES process) to get the lock requests placed by the LMD process.

Sessions

Metric	Description
DB Active Sessions	The number of sessions actively performing work or waiting for a resource (excludes idle sessions) for this database.
DB Blocked Sessions	The number of sessions that are blocked because another session is using a needed resource on this database.
DB Transaction Rate	The number of Transactions (user commits + user rollbacks statistics from v$sysstat) being executed every second for this database.

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the database.

SQL Server metrics collected by DPA

The following sections list the metrics that DPA collects for SQL Server database instances. Some metrics are not collected for every instance.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.
- If the database instance runs on a virtual machine (VM), metrics collected for the VM are described in VM metrics collected by DPA

For DPA to collect metrics from a monitored SQL Server instance, Azure SQL instance, or ASMI, the SQL option `NUMERIC_ROUNDABOUT` must be set to `OFF`.

Backups

When database backups fail or are not performed regularly, organizations run the risk of losing valuable data. Use these metrics to make you aware of any issues and ensure that backups are performed on schedule.

Metric	Description
Active Backup Jobs	The number of currently running backup jobs for the instance. If this number is higher than expected, it can have performance implications or indicate issues with the scheduled backups.
Longest Time for a DB without a Successful DB Backup (Diff or Full)	The longest time that any database in a SQL Server has gone without a successful differential or full backup. Use the "Longest Time" metric values to determine if Service Level Objectives for backup frequency are being met, and use the historical values of these metrics to identify whether recent delays are a one-time problem or a recurring problem (for example, nightly backups aren't happening every Tuesday). If a metric value is higher than expected: • Review the backup schedules for full backups to verify that they are correct and not disabled. • Review the backup results to determine if any errors are causing backup failures to occur. ⓘ To limit the databases that are included in the metric results, you can exclude SQL Server databases from backup metrics.

Metric	Description
Longest Time for a DB without a Successful Full Backup	The longest time that any database in a SQL Server has gone without a successful full backup. See Longest Time for a DB without a Successful DB Backup (Diff or Full) for recommendations.
Longest Time for a DB without a Successful Transaction Log Backup	The longest time that any database in a SQL Server has gone without a successful transaction log backup. See Longest Time for a DB without a Successful DB Backup (Diff or Full) for recommendations.
Size of Transaction Logs Not Yet Archived	The size of all transaction logs in MB that have not yet been archived to free up space for logging future transactions.
Sum of All Backup Assets Required for Recovery of All DBs	The cumulative size, in GB, of all the backup assets for all databases in the SQL Server instance that are required to recover to the current point in time. For each DB, this is the size of the last full backup plus the last differential backup plus all transaction logs created after the most recent full or differential backup. Use this metric to track changes to the minimum required storage space needed to do a complete recovery of the SQL Server. It is also important to understand how much temporary free space could be required to restore all the backup assets for a complete recovery.

Connections

Metric	Description
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle).
Connected Users	The number of distinct users (that is, login names) connected to this instance (even if the connection is idle).
Sessions	The number of sessions connected to this instance (even if the connection is idle).

CPU

Metric	Description
Core Count	The number of cores used by the instance.
Instance CPU Utilization	The CPU Utilization for this specific SQL Server instance. This is a subset of the O/S CPU Utilization metric.
O/S CPU Queue Length via WMI	The number of O/S threads waiting to access the CPU for the entire system (includes all instances on this machine).
O/S CPU Utilization	The percentage of CPU being used for the entire system (includes all instances on this machine). Potential solutions to a CPU bottleneck are to reduce the server load by tuning the queries waiting on CPU, get faster CPUs, or get more CPUs.
Signal Waits	The percentage of total waits that are runnable and waiting for an available CPU. Anything over 20% indicates that there is a possible CPU resource bottleneck. Examine the overall wait events for the server as a whole. A high signal wait percentage could be due to an increased number of sessions, so examine the overall workload for the server as well. Take steps to either reduce the overall runtime for queries or reduce the total number of sessions.

Disk

Metric	Description
O/S Disk Queue Length	The number of I/O operations waiting for disk drives to become available for the entire system (includes all instances on this machine). Spikes of high disk queue length may be normal, but if this is high for an extended period, you could have an I/O bottleneck. Drill in to DPA Trends charts to examine queries with I/O wait types during the timeframe.
O/S Disk Queue Length via WMI	The number of I/O operations waiting for disk drives to become available for the entire system (includes all instances on this machine).
Page Reads	The number of SQL Server physical reads from disk to memory. OLTP workloads are typically about 80-90 per second with higher values (or spikes) being an indication of insufficient storage performance, insufficient indexing, or not enough memory.

Metric	Description
Page Writes	The number of SQL Server physical writes from memory to disk. OLTP workloads are typically about 80-90 per second. If this is high (or spikes) it needs to be cross checked with lazy-writes/sec and checkpoints in order to determine if the issue might be due to low memory.
Physical Read Rate via WMI	The number of kilobytes being read from disk every second for the entire system (includes all instances on this machine). If this is high, drill in to the DPA Trends charts and review queries waiting on physical read wait types, such as `PAGEIOLATCH_SH` or `PAGEIOLATCH_EX`.
Physical Write Rate via WMI	The number of kilobytes being written to disk every second for the entire system (includes all instances on this machine). If this is high, drill in to the DPA Trends charts and review queries waiting on write wait types, such as `IO_COMPLETION` or `PAGEIOLATCH`.
Physical I/O Rate via WMI	The number of kilobytes being read and written to disk every second for the entire system (includes all instances on this machine). If this is high, drill in to the DPA Trends charts and review physical read and write wait types.
SQL Disk Read Latency	Disk read latency from `dm_io_virtual_file_stats` DMO.
SQL Disk Write Latency	Disk write latency from `dm_io_virtual_file_stats` DMO.
Total I/O Wait Time	The sum of all I/O activity for all database files. If this is high: 1. Examine the current physical structure of databases on the server to see if it is possible to reduce I/O load by redistributing the database files to distinct disks. 2. Examine queries and database design to determine if they can be tuned to reduce I/O.
Total Read I/O Wait Time	The sum of all read I/O activity for all database files.
Total Write I/O Wait Time	The sum of all write I/O activity for all database files.

Memory

Metric	Description
Buffer Cache Hit Ratio	The rate at which SQL Server finds the data blocks it needs in memory rather than having to read from disk for this instance. By itself, the buffer cache hit ratio is not very meaningful except for servers with undersized memory settings. Tuning queries and performing index optimization is the best way to increase buffer cache hit ratios. To see the current metrics for the buffer cache, run the following query: `select * from master..sysperfinfo where object_name like 'Buffer Manager'`
Buffer Cache Size	The current size of the SQL Server Buffer Cache.
Log Bytes Flushed	The number of bytes of information being flushed per second.
Log Flushes	The number of log flushes that occur per second.
Memory Paging Rate via WMI	The number of pages read from or to the disk to resolve memory references to pages that were not in memory at the time of the reference. This metric is for the entire system (includes all instances on this machine). High rates may indicate excessive memory contention (thrashing).
O/S Memory Utilization	The percentage of memory being used for the entire system (includes all instances on this machine). If this is high and the Memory Paging Rate metric is high, you might need to increase the amount of physical RAM in the server, reduce the load on the server, or change the server memory configuration. Run `sp_configure` and review settings for "max server memory" and "min server memory" to determine amount of memory allocated to SQL Server.
Page Life Expectancy	The number of seconds a page will stay in the buffer pool without references. A lower value (for example, under 300) indicates the buffer pool is under memory pressure and you should add more memory to the system (enable AWE on 32-bit systems) or find the process in Task Manager that is consuming outside of SQL Server.
Plan Cache Size	The current size of the SQL Server Plan Cache.

Metric	Description
Procedure Cache Hit Ratio	The percentage of time when SQL Server looks for an execution plan in the procedure cache and finds it for this instance. If this is low, try to write more reusable code or consider increasing the size of the procedure cache. To see current metrics for the procedure cache, run the following query: `select * from master..sysperfinfo where object_name like '%Plan Cache%';`
SQL Compilations	The number of compilations performed by SQL Server per second. Compilations are a natural part of SQL Server operations but do utilize CPU and other resources. Compare this to the Batch Requests/sec metric to understand if this metric is too high. Minimizing compilations will help overall performance. For more information, see the following Microsoft Knowledgebase article: http://support.microsoft.com/kb/243588.
SQL Re-Compilations	The number of re-compilations performed by SQL Server per second. Re-compilations occur for many reasons but this number should typically be low.

Network

Metric	Description
Round-trip Time	The round-trip time when running "select 1" against this instance (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

Sessions

Metric	Description
Active Sessions	The number of sessions in this instance actively performing work or waiting for a resource (excludes idle sessions).
Batch Requests	The number of batches being executed by SQL Server every second.
Blocked Sessions	The number of sessions that are blocked in this instance because another session is using a needed resource.
Transaction Rate	The number of transactions being executed every second in this instance (the Transactions/sec statistic from sysperfinfo for the instance).

TempDB

Space required by the TempDB database fluctuates based on the number of queries running and the nature of those queries. If TempDB fills up and cannot autogrow, the performance of all queries is affected. Use TempDB metrics to monitor the amount of space required and determine what types of objects require the most space.

Metric	Description
TempDB Free Space	The amount of free space in TempDB. Spaced used in TempDB fluctuates based on the nature and volume of the SQL statements that are currently running. If TempDB fills up and there is not enough disk space for it to autogrow (or it is not set to autogrow), the performance of all SQL statements will be affected as they wait for access to TempDB.
TempDB Internal Objects	The amount of space in TempDB used by internal objects. Internal objects are created by SQL Server to process queries. For example, internal objects can be used for spooling operations, for sort space, or for hash tables. Queries that process large amounts of data can increase the space required for internal objects in TempDB.
TempDB Log File % Free Space	The percentage of space allocated to the TempDB log file that is not currently being used.
TempDB Log File Free Space	The amount of space allocated to the TempDB log file that is currently free.
TempDB Log File Utilized Space	The amount of space allocated to the TempDB log file that is currently being used.
TempDB Mixed Extents	The amount of space in TempDB used by mixed extents. Mixed extents are shared by up to eight objects.
TempDB User Objects	The amount of space in TempDB used by user objects. User objects are temporary objects explicitly created by users. They include temporary tables and indexes, temporary stored procedures, table variables, and cursors.

Metric	Description
TempDB Version Store	The amount of space in TempDB used by the version store. While a table row is being updated or deleted, the version store contains the committed version of that row. `SELECT` operations that need to access the row being updated or deleted are not blocked because they can read the row in the version store. When the transaction is committed, the row is removed from the version store. Long-running or orphaned transactions can increase the size of the version store. A large version store can affect database performance because of the overhead of reading the large version store.
Total TempDB Log File Size	The amount of disk space allocated for the log file in the TempDB database over time. Each time SQL Server is restarted, TempDB is re-created, and the log file is created using the default size or the size specified by the DBA. If the TempDB log file requires more space, by default it autogrows as needed. However, autogrowth can affect performance because the TempDB log file cannot be used during autogrowth, and because autogrowth can lead to file fragmentation. Use the Total TempDB Log File Size metric to: • Determine the size that the TempDB log file typically grows to over time and specify an initial size that prevents excessive autogrowth. • Identify sudden growth spikes and investigate what queries could have caused the spikes.

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the instance.

MySQL metrics collected by DPA

The following sections list the metrics that DPA collects for MySQL database instances.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.
- If the database instance runs on a virtual machine (VM), metrics collected for the VM are described in VM metrics collected by DPA

Disk

This data reflects activity for sessions accessing tables managed by the InnoDB (or an InnoDB-based) storage engine.

Metric	Description
InnoDB Data Read Ops Rate	The number of InnoDB data read operations per second.
InnoDB Data Write Ops Rate	The number of InnoDB data write operations per second.
InnoDB Log Write Rate	The number of requests per second to write to the InnoDB redo log. The general recommendation is to set the combined size of log files to about 25% through 100% of the buffer pool size to avoid unnecessary buffer pool flush activity on log file overwrite. ⓘ A larger log file size will increase the time needed for a recovery process. If this is one of your top metrics, consider increasing the `innodb_log_file_size` in `my.cnf` and `my.ini` and then restarting MySQL.
InnoDB fsync Call Rate	The number of InnoDB fsync() system calls per second made to flush both the data and log files to disk.

InnoDB Logical I/O

This data reflects activity for sessions accessing tables managed by the InnoDB (or an InnoDB-based) storage engine.

Metric	Description
InnoDB Buffer Pool Read Rate	The number of logical read requests per second from the InnoDB buffer pool. High values usually indicate high load on the system. Reads from the buffer pool are efficient reads, so high rates only rarely indicate a performance problem.
InnoDB Buffer Pool Write Rate	The number of requests per second to write to the InnoDB buffer pool.

Metric	Description
InnoDB Row Read Rate	The number of rows that are read from InnoDB tables per second. An occasional spike in this rate can indicate that a mysqldump backup task is running. If you see a high InnoDB Row Read Rate that you believe is contributing to slow performance, consider optimizing the SQL to reduce the number of rows being read: 1. Go to the Trends page for the time frame and look at the statements with the highest wait time. 2. Determine which of the statements have the highest 'Rows Examined' value on the SQL Data tab. 3. For these statements, consider the following: • Use summary tables where possible to limit the number of rows processed. • Rewrite complicated queries to assist in processing fewer rows. • Evaluate WHERE clauses to ensure you process only rows that are required

Memory

This data reflects activity for sessions accessing tables managed by the InnoDB (or an InnoDB-based) storage engine.

Metric	Description
InnoDB % of Dirty Buffer Pool Pages	The percentage of InnoDB buffer pool data pages that have been changed in memory but have not yet been written (flushed) to disk.

Metric	Description
InnoDB Buffer Pool Consumed Space	The percentage of the InnoDB buffer pool that contains data. Innodb_buffer_pool_size is a very important parameter for InnoDB performance. A rule of thumb is to set the `innodb_buffer_pool_size` up to the total size of the database but not to exceed about 70% of total RAM. A general good practice is to size the buffer pool such that it is mostly full. By doing this, it indicates that you are not wasting memory and that queries are finding the majority of their data in the buffer pool. If this metric is either too high or too low, consider the following: • If the Consumed Space is consistently low, this indicates that your buffer pool is too big and memory is unnecessarily allocated to the buffer pool. Investigate lowering the `innodb_buffer_pool_size` variable. • If the Consumed Space is consistently very high (99% or higher), this may indicate that the size of the buffer pool is too low. Check the resource metric InnoDB Buffer Pool Hit Ratio. If this metric is periodically or consistently low, investigate increasing the `innodb_buffer_pool_size` variable. • If the Consumed Space is low, but it is on the rise, this indicates that the buffer is being initially populated with data. No action is needed at this point.
InnoDB Buffer Pool Data Pages	The number of pages that contain data in the InnoDB buffer pool. This includes both dirty and clean pages.

Metric	Description
InnoDB Buffer Pool Flushed Page Rate	The number of requests per second to flush pages from the InnoDB buffer pool to the data file. Flushing pages to disk is a normal InnoDB operation. InnoDB tries to do this activity in the background when the total load is low. If the flush rate is too high, consider the following: • If the InnoDB log files are too small, this forces a checkpoint operation that flushes buffer pool pages to disk. Check the InnoDB Log Write Rate metric. If you see a lot of log writes that correspond to high InnoDB Buffer Pool Flushed Page Rate values, increase the `innodb_log_file_size` variable. • A buffer pool size that is too small can cause frequent flushes. A rule of thumb is to set the `innodb_buffer_pool_size` up to the total size of the database but not to exceed about 70% of total RAM. ⓘ If you have MyISAM tables, balance the `key_buffer_size` and `innodb_buffer_pool_size` values to best utilize memory for your MySQL instance. • Check the load, mostly writes, on the system and investigate ways to decrease the load. Although SELECTs can also cause pages to be flushed from the buffer pool to disk, writes usually cause higher flush rates. • Optimize SQL to reduce the number of rows being written: 1. Go to the Trends page for the timeframe and look at the UPDATE, DELETE, and INSERT statements with the highest wait time. 2. Determine which statements have the highest Rows Affected or Sent value on the SQL Data tab. 3. For these statements, consider evaluating WHERE clauses to ensure you process only rows that are required.
InnoDB Buffer Pool Hit Ratio	The rate at which the InnoDB engine finds the data blocks it needs in memory rather than having to read from disk. Innodb_buffer_pool_size is a very important parameter for InnoDB performance. A rule of thumb is to set the innodb_buffer_pool_size up to the total size of the database but not to exceed about 70% of total RAM. ⓘ If you have MyISAM tables, you want to balance the key_buffer_size and the innodb_buffer_pool_size to best utilize memory for your MySQL instance. If the hit ratio is lower than 90%, investigate increasing the buffer pool in `my.cnf` and `my.ini` by updating the `innodb_buffer_pool_size` system variable and then restarting MySQL.

Network

Metric	Description
Bytes Received	Throughput of bytes received by MySQL from clients. If Bytes Received has an abnormal spike or if it is higher than normal in general, consider: 1. Examine the Bytes Received (KB/s) metric together with Bytes Sent (KB/s) to gain a more complete story of network traffic. 2. Check the LOAD DATA infile statements which can contribute to the network traffic. 3. Enlist the assistance of your network admin to evaluate network traffic, with a focus on the traffic between the Application server and the MySQL server.
Bytes Sent	Throughput of bytes sent from MySQL to clients. If Bytes Sent has an abnormal spike or if it is higher than normal, consider the following: 1. Examine the Bytes Received (KB/s) metric together with Bytes Sent (KB/s) to gain a more complete story of network traffic. 2. Optimize SQL to reduce network traffic. Go to the Trends page to identify which SQL statements have the highest wait time. Determine which of these statements have the highest Rows Affected or Sent statistic on the SQL Data tab. For these statements: • Evaluate WHERE clauses to ensure you are processing only rows that are required. • Eliminate columns from your result set that you don't need. • Use summary tables where possible to limit the number of rows processed/returned. • Rewrite complicated queries to assist in processing fewer rows. 3. Enlist the assistance of your network admin to evaluate network traffic, with a focus on the traffic between the Application server and the MySQL server.
Round-trip Time	The round-trip time when running "select 1" against this instance (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

Objects

Metric	Description
Table Cache Filled	The percentage of the cache that is filled with "file descriptors" (that is, an `.frm` file that contains a table's underlying format).
Table Cache Hit Ratio	The percentage of time that MySQL used an available cached "file descriptor" (that is, an `.frm` file that contains a table's underlying format). Whenever MySQL needs to access a table, it needs the table structure. The structures of previously opened tables are stored in the table cache. If a table's structure has not been cached, MySQL needs to load the structure from disk into cache, negatively affecting database performance. The lower this ratio is, the more the database has to load table structures from disk. Table structures are stored in .frm files on disk (tableName.frm). If the Table Cache Hit Ratio is low, increase the `table_open_cache` variable in `my.cnf` and `my.ini`. Recommendations: • Set `table_open_cache` to the total number of tables in the database. • A typical range for the `table_open_cache` is from 2000 (default) to 100,000.

Sessions

Metric	Description
Active Threads	The number of active threads in the database instance to support client connections. This metric is based on the MySQL Global Status variable `threads_running`. MySQL associates each client connection with a dedicated thread that handles all requests for that connection. This means that there are as many threads as there are clients currently connected. MySQL employs a thread cache to reduce the performance penalties associated with creating and destroying threads. The size of the thread cache is governed by the `thread_cache_size` system variable. When a connection is established, MySQL creates a new thread if an available thread cannot be found in the thread cache. When a connection ends, its thread is returned to the thread cache unless the cache is full. To monitor how many threads are being created because no cached thread is available, look at the Created Threads (sessions) metric. Each thread has some overhead in the form of server and kernel resources, including stack space, that affects the ability to scale to handle large numbers of connections. If you need to handle a large number of simultaneous connections, a common solution is to decrease the thread stack size. Doing so will limit memory-consuming activities conducted by the thread. If Active Threads is too high, consider the following: • Use connection pooling in your applications to reduce the number of simultaneous queries. • Use the MySQL master/slave architecture and move some or all SELECT queries to a slave. • MySQL may be incurring excess overhead such as memory. If you feel that this is a problem, you can decrease the thread stack size, but you need to realize that this will limit memory-consuming activities conducted by the thread. In other words, it limits complexity of SQL statements and stored program recursion depth. To set the stack size, start the server with `--thread_stack=N` where `N` is in bytes. • If the Active Threads value is higher than the thread cache size, MySQL may be incurring excess expense due to the creation and destruction of threads. If you feel that this overhead is a problem, you can try increasing the size of the thread cache by increasing the value of the `thread_cache_size` system variable.

Metric	Description
Blocked Threads	The number of threads that are blocked because another thread is holding a lock on an object, typically a table or an index. Drill down in the Trend page to locate additional details about what the blocking sessions are doing. Tune the queries you find by adding indexes or rewriting queries to minimize the time the locks are held.
Connection Attempts	The number of connection attempts in the given time interval (successful or not). This metric is based on the MySQL Global Status variable connections. If the Connection Attempts value is high, investigate the connection attempts in the logs. Enable logging of the connection attempts in the following ways: • If you are only interested in aborted attempts, make sure that the value (level) of the `log_warning` system variable is 2, and then check the error log. • If you are interested in successful and aborted connections, make sure that the general query log is enabled by checking the `general_log` system variable. The location of the general query log file is in the `general_log_file` system variable. Enabling the general query log can decrease the performance of the MySQL server, as every connection attempt and SQL statement will be logged.
Created Threads	The number of created threads in the database instance to support client connections in the given interval. This metric is based on the MySQL Global Status variable `threads_created`. MySQL associates each client connection with a dedicated thread that handles all requests for that connection. This means that there are as many threads as there are clients currently connected. Because thread creation and disposal can be expensive, MySQL employs a thread cache. When a connection is established, MySQL will create a new thread if an available thread cannot be found in the thread cache. When a connection ends, its thread is returned to the thread cache unless the cache is full. To monitor how many threads are currently running (cached or not), look at the Active Threads (sessions) metric. • If the Thread Creation Rate value is high and the thread cache is not full, this generally means that the cache is being filled, which is a normal situation. To see how many threads are in the cache and the size of the thread cache, look at the `threads_cached` and `thread_cache_size` system variables. • If the Thread Creation Rate value is high and the thread cache is full, this means that available threads are not being found in the thread cache, causing new connections to create new threads, which can be an expensive operation. If you think this overhead is causing problems, you can try increasing the size of the thread cache by increasing the value of the `thread_cache_size` system variable. • Consider using connection pooling in your application(s).

Sorts/Joins

Metric	Description
Joins By Table Scan	The number of joins that performed table scans (that is, joins that did not use indexes).
On-Disk Temp Table Creation Rate	The number of internal on-disk temporary tables created per second while executing statements.
Row Sort Rate	The number of rows sorted per second while executing statements. If MySQL cannot use an index to retrieve presorted rows, it performs a sort that increments the `sort_rows` counter. If this metric is high, consider these solutions: • Check the Sort Merge Passes resource metric and determine if there is a need to increase `sort_buffer_size`. • Optimize the SQL to reduce sorting. 1. Go to the Trends page for the time frame and look at the statements with the highest wait time. 2. Find the statements with the highest Rows Sorted value on the SQL Data tab. 3. For these statements, consider using a combined or covered index with the same columns in the same order as the ORDER BY clause. In some cases, MySQL can use an index to satisfy an ORDER BY clause without performing any extra sorting.

Metric	Description
Sort Merge Passes	The number of merge passes per second performed by the sort algorithm. A Sort Merge Pass occurs if sorting large amounts of data using a limited amount of space. Performance suffers when these sorts can not be performed in memory. When the sort buffer overflows, MySQL creates temporary files on disk to use in the file sorting and merging algorithm. The data is sorted in multiple passes to first sort small chunks of data before merging the results together. • Look at the Row Sort Rate metric. If there is a lot of sorting happening in this time frame, follow the suggested solutions. • Increase the global `sort_buffer_size` system variable to improve the performance of queries that sort a lot of data. • Increase the `sort_buffer_size` at the session level with a SET statement. Add the statement to your application code before running these kinds of queries. For example: `SET session sort_buffer_size = 8M` • Use an index with columns in the ORDER BY clause. MySQL might use this index to satisfy an ORDER BY clause without extra sorting.
Temp Table Creation Rate	The number of internal temporary tables created per second while executing statements. MySQL creates internal temporary tables to process operations such as SELECT ... GROUP BY / ORDER BY and SELECT DISTINCT. Unfortunately, temporary tables larger than the sizes specified in `tmp_table_size` and `max_heap_table_size` have to be converted to a slow, disk-based MyISAM temporary table. Likewise, if the query uses TEXT or BLOB fields, MySQL always has to use slow, disk-based temporary tables because in-memory temporary tables don't support those fields. If this metric is high, you run the risk of temporary tables being created on disk. Consider the following: 1. Go to the Trends page to identify which SQL statements have the highest wait time. 2. Find the statements with the highest Temp Tables Created statistic on the SQL Data tab. 3. For these statements: • Use a combined or covered index that has the same columns in the same order as the ORDER BY clause. In some cases, MySQL can use an index to satisfy an ORDER BY clause without doing any extra sorting. • Remove TEXT/BLOB fields if they are not needed for the query. Consider also increasing the `tmp_table_size` or `max_heap_table_size` values to reduce the number of internal temporary tables that have to be written to disk.

Statements

Metric	Description
Delete Statement Rate	The number of times a DELETE statement has been executed per second.
Insert Statement Rate	The number of times an INSERT statement has been executed per second.
Select Statement Rate	The number of times a SELECT statement has been executed per second.
Statements Execution Rate	The number of statements executed per second, not including those executed from stored programs. This is only a problem if your users complain about poor performance. Consider the following: • Identify and tune the queries with the highest wait time. • Look to see if there are a high number of executions of a SQL Statement. Look for possible ways to modify your application to decrease the number of executions, such as caching. • If you believe poor performance is due to the volume of statements being executed, consider implementing a MySQL master and slave architecture and move some or all SELECT queries to a slave.
Statements Execution Rate from Stored Programs	The number of statements executed per second from programs. This is only a problem if your users complain about poor performance. Consider these measures: • Identify and tune the queries with the highest wait time. • If you believe poor performance is due to the volume of statements being executed, consider implementing a MySQL master and slave architecture and move some or all SELECT queries to a slave.
Update Statement Rate	The number of times an UPDATE statement has been executed per second.

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the instance.

Sybase metrics collected by DPA

The following sections list the metrics that DPA collects for Sybase database instances.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.
- If the database instance runs on a virtual machine (VM), metrics collected for the VM are described in VM metrics collected by DPA

Connections

Metric	Description
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle).
Connected Users	The number of distinct users connected (that is, distinct logins) to this instance (even if the connection is idle).
Sessions	The number of sessions connected to this instance (even if the connection is idle).

CPU

Metric	Description
CPU Utilization By Sybase	The percentage of CPU being used by the database instance, which is a subset of the CPU used by the entire system. If this is high, use DPA's Trends charts to review queries with the wait type "waiting on run queue after sleep".
Core Count	The number of cores used by the instance.

Disk

Metric	Description
DB APF Read Rate	The number of Asynchronous Prefetch (APF) reads this Sybase server performed from disk every second. If this is high, drill in to DPA Trends charts and review queries with wait types of "waiting for an APF buffer read to complete".
DB Physical I/O Rate	The number of read and write operations that this Sybase server performed to or from disk every second. If this is high, drill in to the DPA Trends charts and review physical read and write wait types.
DB Physical Read Rate	The number of read operations this Sybase server performed from disk every second. If this is high, drill in to the DPA Trends charts and review queries with physical read wait types, such as "waiting for i/o (read or write) to complete".

Metric	Description
DB Physical Write Rate	The number of write operations this Sybase server performed to disk every second. If this is high, drill in to the DPA Trends charts and review queries with write wait types, such as "waiting for disk write to complete".
Disk I/O Access Time	The average time to read from or write to disk.

Memory

Metric	Description
Buffer Cache Hit Ratio	The rate at which Sybase finds the data blocks it needs in memory rather than having to read from disk. By itself, the buffer cache hit ratio is not very meaningful except for servers with undersized memory settings. Tuning queries and performing index optimization is the best way to increase buffer cache hit ratios. To determine the current sizes of the data caches, use the `sp_helpcache` command.
Procedure Cache Hit Ratio	The percentage of time Sybase finds an available plan already in cache. If this is low, try to write more reusable code and/or consider increasing the size of the procedure cache. To determine the current size of the procedure cache, review the value of the `procedure cache size` parameter.

Network

Metric	Description
DB Network Receive Rate	The number of bytes received over the network every second for this database. If this is high, drill in to DPA Trends charts and review queries with network wait types (for example, "waiting for incoming network data").

Metric	Description
DB Network Send Rate	The number of bytes sent over the network every second for this database. If this is high, drill in to the DPA Trends charts and review queries with network wait types (for example, "waiting for network send to complete").
DB Round-trip Time	The round-trip time when running "select 1" (includes network time but not connect time) on this database. If this is high, contact your network administrator to understand network latency.

Sessions

Metric	Description
DB Active Sessions	The number of sessions actively performing work or waiting for a resource (excludes idle sessions).
DB Blocked Sessions	The number of sessions that are blocked because another session is using a needed resource.

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the instance.

Db2 metrics collected by DPA

The following sections list the metrics that DPA collects for Db2 self-managed database instances.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.
- If the database instance runs on a virtual machine (VM), metrics collected for the VM are described in VM metrics collected by DPA

Connections

Metric	Description
Connected Devices	The number of distinct client machines connected (even if the connection is idle).
Connected Users	The number of distinct users connected (even if the connection is idle).

Metric	Description
Sessions	The number of sessions connected (even if the connection is idle).

CPU

Metric	Description
Core Count	The number of cores used by the instance.
O/S CPU Utilization	The percentage of CPU being used for the entire system (includes all databases on this machine). Potential solutions to a CPU bottleneck are to reduce the server load (tune those queries), get faster CPUs, or get more CPUs.

Disk

Metric	Description
DB Physical I/O Rate	The number of read and write operations performed to/from disk every second. If this is high, drill in to the DPA Trends charts and review queries waiting on physical read and write wait events. ⓘ Db2 might supply this value only if the DFT_MON_BUFPOOL monitoring switch is ON.
DB Physical Read Rate	The number of reads performed from disk every second. If this is high, drill in to the DPA Trends charts and review queries waiting on physical read wait events. ⓘ Db2 might supply this value only if the DFT_MON_BUFPOOL monitoring switch is ON.
DB Physical Write Rate	The number of writes performed to disk every second. If this is high, drill in to the DPA Trends charts and review queries waiting on write wait events. ⓘ Db2 might supply this value only if the DFT_MON_BUFPOOL monitoring switch is ON.

Memory

Metric	Description
DB Buffer Pool Hit Ratio	The rate at which Db2 finds the data it needs in memory rather than having to read from disk. By itself, the buffer pool hit ratio is not very meaningful except for databases with undersized memory settings. Db2 might supply this value only if the DFT_MON_BUFFERPOOL monitoring switch is ON. Tuning queries and performing index optimization is the best way to increase buffer pool hit ratios.
DB Catalog Cache Hit Ratio	The percentage of time when Db2 looks for an execution plan in the catalog cache and finds it. A low hit ratio indicates the `catalogcache_sz` parameter should be increased.
DB Package Cache Hit Ratio	The percentage of time when Db2 looks for an execution plan in the package cache and finds it. A low hit ratio indicates the `pckcachesz` parameter should be increased.
O/S Memory Utilization	The percentage of system memory being used for the entire system (includes all databases on this machine). If this is high, you might need to increase the amount of physical RAM in the server, reduce the load on the server, or change your server memory configuration.
Virtual Memory Utilization	The percentage of virtual memory being used.

Some metrics are not available for all Oracle deployments.

Network

Metric	Description
DB Round-trip Time	The round-trip time when running "select 1 from sysibm.sysdummy1" against the database specified during registration (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

Sessions

Metric	Description
DB Blocked Sessions	The number of sessions that are waiting on lock waits because another session is using a needed resource.

Metric	Description
DB Connections Currently Executing	The number of sessions that are actively performing work or waiting for a resource (excludes idle sessions).
DB Transaction Rate	The number of transactions being executed every second: `commit_sql_stmts + int_commits + rollback_sql_stmts + int_rollbacks`

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the instance.

Azure SQL database metrics collected by DPA

The following sections list the metrics that DPA collects for Azure SQL database instances.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.
- If the database instance runs on a virtual machine (VM), metrics collected for the VM are described in VM metrics collected by DPA

Connections

Metric	Description
Connected Machines	The number of distinct client machines connected to this database (even if the connection is idle).
Connected Users	The number of distinct users (that is, login names) connected to this database (even if the connection is idle).
Sessions	The number of sessions connected to this database (even if the connection is idle).

CPU

Metric	Description
CPU Utilization	The percentage of CPU being used based on the DTU limit. If CPU is near the upper limit, use DPA to determine which queries are contributing to high CPU usage, and determine if the queries can be tuned.

Disk

Metric	Description
Data I/O Utilization	The percentage of data I/O utilization based on the DTU limit.
Database Size	The size of the database in GB (rounded up to the nearest GB).
Database Storage Consumption	The percent of the storage available to the database that is currently used.
Log Write Utilization	The percentage of log write Utilization based on the DTU limit.

DTU

Metric	Description
DTU Consumption	The total number of DTUs (Database Transaction Units) being used. ⓘ A DTU represents a combination of CPU, memory, data I/O and transaction log I/O. Microsoft places limits on these resources based on what service tier a database is in. When a database exceeds its limit for any resource, Microsoft restricts throughput, which slows performance.
DTU Limit	The DTU limit for this database instance.
DTU Utilization	The percentage of available DTUs being used. Use this value to determine the appropriate service tier for your needs.

Memory

Metric	Description
Memory Usage Utilization	The percentage of memory being used. ⓘ It is not unusual for this metric to be high. Data being used by applications is stored in memory to improve performance.

Metric	Description
XTP Storage Utilization	The percentage of XTP storage utilization based on the DTU limit. This resource is available only for databases running on the Premium service tier. Zero percent is returned for the Basic and Standard service tiers.

Network

Metric	Description
Round-trip Time	The round-trip time when running "select 1" against this database (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

Sessions

Metric	Description
Active Sessions	The number of sessions in this database actively performing work or waiting for a resource (excludes idle sessions).
Blocked Sessions	The number of sessions in this database that are blocked because another session is using a needed resource.
Max Session Utilization	The percentage of Max Session Utilization based on the database limit.
Max Worker Utilization	The percentage of Max Worker Utilization based on the database limit.

Waits

Metric	Description
Total Instance Wait Time	Total wait time for the database instance.

ASMI metrics collected by DPA

The following sections list the metrics that DPA collects for Azure SQL managed instances (ASMIs).

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.
- If the database instance runs on a virtual machine (VM), metrics collected for the VM are described in VM metrics collected by DPA

Connections

Metric	Description
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle).
Connected Users	The number of distinct users (that is, login names) connected to this instance (even if the connection is idle).
Sessions	The number of sessions connected to this instance (even if the connection is idle).

CPU

Metric	Description
Core Count	The number of cores used by the instance.
CPU Utilization	The amount of CPU being used as a percentage of the limit of the service tier.
Signal Waits	The percentage of overall time that sessions are waiting for a CPU to become available. Anything over 20% indicates a possible CPU resource bottleneck. Examine the overall wait events for the server as a whole. A high signal wait percentage could be due to an increased number of sessions, so examine the overall workload for the server as well. Take steps to either reduce the overall runtime for queries or reduce the total number of sessions.

Disk

Metric	Description
Data I/O Utilization	The average data I/O utilization as a percentage of the service tier limit.
Log Write Utilization	The average transaction log writes as percentage of the service tier limit.

Metric	Description
O/S Disk Queue Length	The number of I/O operations waiting for disk drives to become available for the entire system (includes all instances on this machine). Spikes of high disk queue length may be normal, but if this is high for an extended period, you could have an I/O bottleneck. Drill in to DPA Trends charts to examine queries with I/O wait types during the timeframe.
Page Reads	The number of SQL Server physical reads from disk to memory. OLTP workloads are typically about 80-90 per second with higher values (or spikes) being an indication of insufficient storage performance, insufficient indexing, or not enough memory.
Page Writes	The number of SQL Server physical writes from memory to disk. OLTP workloads are typically about 80-90 per second. If this is high (or spikes) it needs to be cross checked with lazy-writes/sec and checkpoints in order to determine if the issue might be due to low memory.
SQL Disk Read Latency	Disk read latency from dm_io_virtual_file_stats DMO.
SQL Disk Write Latency	Disk write latency from dm_io_virtual_file_stats DMO.
Total I/O Wait Time	The sum of all I/O activity for all database files. If this is high: 1. Examine the current physical structure of databases on the server to see if it is possible to reduce I/O load by redistributing the database files to distinct disks. 2. Examine queries and database design to determine if they can be tuned to reduce I/O.
Total Read I/O Wait Time	The sum of all read I/O activity for all database files.
Total Write I/O Wait Time	The sum of all write I/O activity for all database files.

Memory

Metric	Description
Buffer Cache Hit Ratio	The rate at which SQL Server finds the data blocks it needs in memory rather than having to read from disk for this instance. By itself, the buffer cache hit ratio is not very meaningful except for servers with undersized memory settings. Tuning queries and performing index optimization is the best way to increase buffer cache hit ratios. To see the current metrics for the buffer cache, run the following query: `select * from master..sysperfinfo where object_name like 'Buffer Manager'`
Buffer Cache Size	The current size of the SQL Server Buffer Cache.
Log Bytes Flushed	The number of bytes of information being flushed per second.
Log Flushes	The number of log flushes that occur per second.
O/S Memory Utilization	The percentage of memory being used for the entire system (includes all instances on this machine). If this is high and the Memory Paging Rate metric is high, you might need to increase the amount of physical RAM in the server, reduce the load on the server, or change the server memory configuration. Run `sp_configure` and review settings for "max server memory" and "min server memory" to determine amount of memory allocated to SQL Server.
Page Life Expectancy	The number of seconds a page will stay in the buffer pool without references. A lower value (for example, under 300) indicates the buffer pool is under memory pressure and you should add more memory to the system (enable AWE on 32-bit systems) or find the process in Task Manager that is consuming outside of SQL Server.
Plan Cache Size	The current size of the SQL Server Plan Cache.
Procedure Cache Hit Ratio	The percentage of time when SQL Server looks for an execution plan in the procedure cache and finds it for this instance. If this is low, try to write more reusable code or consider increasing the size of the procedure cache. To see current metrics for the procedure cache, run the following query: `select * from master..sysperfinfo where object_name like '%Plan Cache%';`

Metric	Description
SQL Compilations	The number of compilations performed by SQL Server per second. Compilations are a natural part of SQL Server operations but do utilize CPU and other resources. Compare this to the Batch Requests/sec metric to understand if this metric is too high. Minimizing compilations will help overall performance. For more information, see the following Microsoft Knowledgebase article: http://support.microsoft.com/kb/243588.
SQL Re-Compilations	The number of re-compilations performed by SQL Server per second. Re-compilations occur for many reasons but this number should typically be low.
XTP Storage Utilization	The percentage of available XTP Storage being used.

Network

Round-trip Time	The round-trip time when running "select 1" against this instance (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

Sessions

Metric	Description
Active Sessions	The number of sessions in this instance actively performing work or waiting for a resource (excludes idle sessions).
Batch Requests	The number of batches being executed by SQL Server every second.
Blocked Sessions	The number of sessions that are blocked in this instance because another session is using a needed resource.
Max Session Utilization	Maximum concurrent sessions as a percentage of the limit of the database's service tier.
Max Worker Utilization	Maximum concurrent workers (requests) as a percentage of the limit of the database's service tier.
Transaction Rate	The number of transactions being executed every second in this instance (the Transactions/sec statistic from sysperfinfo for the instance).

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the database instance.

PostgreSQL metrics collected by DPA

The following sections list the metrics that DPA collects for PostgreSQL databases.

Learn how to view these metrics and change the thresholds.

Cache Eviction

Metric	Description
Dirty Buffers Evicted by Background Writer	The number of buffers written and freed (evicted) due to the PostgreSQL background writer.
Dirty Buffers Evicted by Background Writer Ratio	The ratio of buffers written and freed (evicted) due to the PostgreSQL background writer to the total number of evictions.
Dirty Buffers Evicted by Checkpoints	The number of buffers written and freed (evicted) due to a checkpoint execution. Higher values indicate an increased need for checkpoints.
Dirty Buffers Evicted by Checkpoints Ratio	The ratio of buffers written and freed (evicted) due to a checkpoint execution to the total number of evictions. Higher values indicate an increased need for checkpoints.
Dirty Buffers Evicted by Client Backends	The number of times client backends were delayed by being forced to write and free (evict) buffers themselves, instead of the buffers being evicted asynchronously by the background writer.
Dirty Buffers Evicted by Client Backends Ratio	The ratio of buffers written and freed (evicted) by a client backend to the total number of evictions.
Total Dirty Buffers Evicted	The total number of dirty buffers evicted by checkpoints, background writer, and client backends.

Checkpoints

Metric	Description
Requested Checkpoints	The number of unscheduled checkpoints requested by client statements because the WAL size has reached its threshold (`max_wal_size`). Requested checkpoints can cause client backend waits. Consider reconfiguration of checkpoint related settings (`checkpoint_timeout`, `checkpoint_completion_target`, and `max_wal_size`).
Requested Checkpoints Ratio	The ratio of requested checkpoints to total checkpoints (requested and scheduled). The percentage should be low—optimally 0%.
Scheduled Checkpoints	The number of scheduled checkpoints processed in the background, without affecting client statements. Scheduled checkpoints should not cause client backend waits.

Connections

Metric	Description
Connected Devices	The number of distinct client machines connected to this database instance, even if the connection is idle.
Connected Users	The number of distinct users (that is, login names) connected to this database instance, even if the connection is idle.

CPU

To collect CPU metrics from a PostgreSQL instance, DPA requires the system_stats extension to be installed in the PostgreSQL instance.

Metric	Description
Host CPU Utilization	The percentage of CPU being used by the entire database server host. If this is high, compare this metric to the Instance CPU Utilization metric. If the instance is **not** using a significant portion of total CPU, review other programs running at this time.
Instance CPU Utilization	The percentage of CPU being used by the database instance, which is a subset of the CPU used by the host. If this is high, use DPA Trends charts to review queries waiting on CPU.

Disk

Metric	Description
Blocks Hit	The number of times disk blocks were found already in the buffer cache, so that a read was not necessary. This includes only hits in the PostgreSQL buffer cache, not the operating system's file system cache.
Blocks Read	The number of disk blocks read in this database.
Blocks Read Time	The average amount of block read I/O during the specified time interval. If the `track_io_timing` parameter is off, the value of this metric is always 0. For more information, see this support article.
Blocks Write Time	The average amount of block write I/O during the specified time interval. If the `track_io_timing` parameter is off, the value of this metric is always 0. For more information, see this support article.
Temp Bytes Written	The total amount of data in kilobytes written to temporary files by queries in this database. All temporary files are counted, regardless of why the temporary file was created, and regardless of the `log_temp_files` setting.
Temp Files	The number of temporary files created by queries in this database. All temporary files are counted, regardless of why the temporary file was created (for example, sorting or hashing), and regardless of the `log_temp_files` setting.
Write-ahead Log (WAL) Rate	The rate of the Write-ahead Log creation as a result of database transaction activity in MB per second.

Memory

Metric	Description
Buffer Cache Hit Ratio	The rate at which PostgreSQL finds the data blocks it needs in memory rather than having to read from disk.

Network

Metric	Description
DB Round-trip Time	The round-trip time when running "select 1" (includes network time but not connect time) on this database.

Replication

Metric	Description
Replication Lag	The replication lag between the primary database and all replica databases. An increase in the replication lag indicates a growing number of transactions that are not yet replicated and at risk of not being replicated if the primary database fails.

Rows

Metric	Description
Fetched vs. Returned Rows	Of the total number of rows that were scanned (Rows Returned), the percentage that contained data needed to execute the query (Rows Fetched). High values indicate that the database is executing queries efficiently. Low values indicate that the database is performing extra work because it is scanning a large number of rows that aren't required to process the query. For example, 10% means that the database is scanning 10 rows to use 1 row. Low values could indicate inefficient queries or missing indexes.
Rows Operations	The number of rows inserted, updated, and deleted by queries in this database instance.
Rows Deleted	The number of rows deleted by queries in this database instance.
Rows Fetched	The subset of scanned rows (Rows Returned) that contained data needed to execute the query. For example, take the following query: `SELECT * FROM customers WHERE country = 'Spain';` The `customers` table has 10,000 rows, and `country = 'Spain'` in 100 rows. The column is not indexed, and so a full table scan is required. The Rows Returned value is 10,000, but the Rows Fetched value is only 100. ⓘ The Rows Fetched value is different than the number of rows returned to the client.
Rows Inserted	The number of rows inserted by queries in this database instance.
Rows Returned	The total number of rows scanned by queries executed against this database instance. ⓘ This value indicates rows returned by the storage layer to be scanned, **not** rows returned to the client.

Metric	Description
Rows Updated	The number of rows updated by queries in this database instance.

Sessions

Metric	Description
Active Sessions	The number of transactions in the following state: The backend is executing a query.
All Sessions	All sessions, regardless of state.
Blocked Sessions	The number of sessions in a blocked state.
Deadlocks	The number of deadlocks detected in this database instance.
Fastpath Function Call Sessions	The number of sessions in which the backend is executing a fast-path function.
Idle Sessions	The number of sessions in the following state: The backend is waiting for a new client command.
Idle in Transaction (Aborted) Sessions	This state is similar to Idle in Transaction Sessions, except one of the statements in the transaction caused an error.
Idle in Transaction Sessions	The number of sessions in the following state: The backend is in a transaction, but is not currently executing a query.
Percentage of Idle in Transaction	The percentage of allowed connections that are in the idle-in-transaction state. Transactions that are in the idle-in-transaction state for a significant amount of time can cause the connection pool to fill to the limit, and they can cause other database operations such as VACUUM to fail to complete. You might also see `SELECT waiting` messages, which indicates that an HTTP request was made but the SELECT operation to get the data never completed (probably because the user waited so long that they abandoned the task).
Recovery Conflicts	The number of queries canceled due to conflicts with recovery in this database instance. Conflicts occur only on standby servers.

Metric	Description
Transaction Commit Rate	The number of transactions being committed every second in this database instance.
Transaction Rate	The number of transactions being executed every second in this database instance.
Transaction Rollbacks	The number of transactions in this database that have been rolled back.
track_ activities Disabled Sessions	This state is reported if `track_activities` is disabled in this backend.

Vacuum

Metric	Description
Autovacuum Worker Utilization	An indication of how busy the set of vacuum worker processes are. The `pg_stat_ progress_vacuum` view provides information about current vacuuming processes. If this value is consistently high, consider increasing the value of the `autovacuum_ max_workers` parameter.
Multixact ID Space Taken	The percentage of space available to store Multixact IDs (MXIDs) that is currently filled. this is the highest value across all databases in the database server.
Transaction ID Space Taken	The percentage of space available to store Transaction IDs (XIDs) that is currently filled. this is the highest value across all databases in the database server.

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the database instance.

Amazon RDS for SQL Server metrics collected by DPA

The following sections list the metrics that DPA collects for Amazon RDS for SQL Server database instances. Some metrics are not collected for every instance.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.

Backups

When database backups fail or are not performed regularly, organizations run the risk of losing valuable data. Use these metrics to make you aware of any issues and ensure that backups are performed on schedule.

Metric	Description
Active Backup Jobs	The number of currently running backup jobs for the instance. If this number is higher than expected, it can have performance implications or indicate issues with the scheduled backups.
Longest Time for a DB without a Successful DB Backup (Diff or Full)	The longest time that any database in a SQL Server has gone without a successful differential or full backup. Use the "Longest Time" metric values to determine if Service Level Objectives for backup frequency are being met, and use the historical values of these metrics to identify whether recent delays are a one-time problem or a recurring problem (for example, nightly backups aren't happening every Tuesday). If a metric value is higher than expected: • Review the backup schedules for full backups to verify that they are correct and not disabled. • Review the backup results to determine if any errors are causing backup failures to occur. ⓘ To limit the databases that are included in the metric results, you can exclude SQL Server databases from backup metrics.
Longest Time for a DB without a Successful Full Backup	The longest time that any database in a SQL Server has gone without a successful full backup. See Longest Time for a DB without a Successful DB Backup (Diff or Full) for recommendations.

Metric	Description
Longest Time for a DB without a Successful Transaction Log Backup	The longest time that any database in a SQL Server has gone without a successful transaction log backup. See Longest Time for a DB without a Successful DB Backup (Diff or Full) for recommendations.
Size of Transaction Logs Not Yet Archived	The size of all transaction logs in MB that have not yet been archived to free up space for logging future transactions.
Sum of All Backup Assets Required for Recovery of All DBs	The cumulative size, in GB, of all the backup assets for all databases in the SQL Server instance that are required to recover to the current point in time. For each DB, this is the size of the last full backup plus the last differential backup plus all transaction logs created after the most recent full or differential backup. Use this metric to track changes to the minimum required storage space needed to do a complete recovery of the SQL Server. It is also important to understand how much temporary free space could be required to restore all the backup assets for a complete recovery.

Connections

Metric	Description
Connected Devices	The number of distinct client machines connected to this instance (even if the connection is idle).
Connected Users	The number of distinct users (that is, login names) connected to this instance (even if the connection is idle).
Sessions	The number of sessions connected to this instance (even if the connection is idle).

CPU

Metric	Description
Core Count	The number of cores used by the instance.
Instance CPU Utilization	The CPU Utilization for this specific SQL Server instance.

Metric	Description
Signal Waits	The percentage of total waits that are runnable and waiting for an available CPU. Anything over 20% indicates that there is a possible CPU resource bottleneck. Examine the overall wait events for the server as a whole. A high signal wait percentage could be due to an increased number of sessions, so examine the overall workload for the server as well. Take steps to either reduce the overall runtime for queries or reduce the total number of sessions.

Disk

Metric	Description
SQL Disk Read Latency	Disk read latency from `dm_io_virtual_file_stats` DMO.
SQL Disk Write Latency	Disk write latency from `dm_io_virtual_file_stats` DMO.
Total I/O Wait Time	The sum of all I/O activity for all database files. If this is high: 1. Examine the current physical structure of databases on the server to see if it is possible to reduce I/O load by redistributing the database files to distinct disks. 2. Examine queries and database design to determine if they can be tuned to reduce I/O.
Total Read I/O Wait Time	The sum of all read I/O activity for all database files.
Total Write I/O Wait Time	The sum of all write I/O activity for all database files.

Memory

Metric	Description
Buffer Cache Hit Ratio	The rate at which SQL Server finds the data blocks it needs in memory rather than having to read from disk for this instance. By itself, the buffer cache hit ratio is not very meaningful except for servers with undersized memory settings. Tuning queries and performing index optimization is the best way to increase buffer cache hit ratios. To see the current metrics for the buffer cache, run the following query: `select * from master..sysperfinfo where object_name like 'Buffer Manager'`
Buffer Cache Size	The current size of the SQL Server Buffer Cache.
Log Bytes Flushed	The number of bytes of information being flushed per second.
Log Flushes	The number of log flushes that occur per second.
Page Life Expectancy	The number of seconds a page will stay in the buffer pool without references. A lower value (for example, under 300) indicates the buffer pool is under memory pressure and you should add more memory to the system (enable AWE on 32-bit systems) or find the process in Task Manager that is consuming outside of SQL Server. The default threshold for Page Life Expectancy (PLE) is 300. For modern database systems, DBAs recommend using a formula such as the following to calculate an appropriate PLE threshold: `DataCacheSizeInGB / 4 GB * 300`
Plan Cache Size	The current size of the SQL Server Plan Cache.
Procedure Cache Hit Ratio	The percentage of time when SQL Server looks for an execution plan in the procedure cache and finds it for this instance. If this is low, try to write more reusable code or consider increasing the size of the procedure cache. To see current metrics for the procedure cache, run the following query: `select * from master..sysperfinfo where object_name like '%Plan Cache%';`

Metric	Description
SQL Compilations	The number of compilations performed by SQL Server per second. Compilations are a natural part of SQL Server operations but do utilize CPU and other resources. Compare this to the Batch Requests/sec metric to understand if this metric is too high. Minimizing compilations will help overall performance. For more information, see the following Microsoft Knowledgebase article: http://support.microsoft.com/kb/243588.
SQL Re-Compilations	The number of re-compilations performed by SQL Server per second. Re-compilations occur for many reasons but this number should typically be low.

Network

Metric	Description
Round-trip Time	The round-trip time when running "select 1" against this instance (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

Sessions

Metric	Description
Active Sessions	The number of sessions in this instance actively performing work or waiting for a resource (excludes idle sessions).
Batch Requests	The number of batches being executed by SQL Server every second.
Blocked Sessions	The number of sessions that are blocked in this instance because another session is using a needed resource.
Transaction Rate	The number of transactions being executed every second in this instance (the Transactions/sec statistic from sysperfinfo for the instance).

TempDB

Space required by the TempDB database fluctuates based on the number of queries running and the nature of those queries. If TempDB fills up and cannot autogrow, the performance of all queries is affected. Use TempDB metrics to monitor the amount of space required and determine what types of objects require the most space.

Metric	Description
TempDB Free Space	The amount of free space in TempDB. Spaced used in TempDB fluctuates based on the nature and volume of the SQL statements that are currently running. If TempDB fills up and there is not enough disk space for it to autogrow (or it is not set to autogrow), the performance of all SQL statements will be affected as they wait for access to TempDB.
TempDB Internal Objects	The amount of space in TempDB used by internal objects. Internal objects are created by SQL Server to process queries. For example, internal objects can be used for spooling operations, for sort space, or for hash tables. Queries that process large amounts of data can increase the space required for internal objects in TempDB.
TempDB Log File % Free Space	The percentage of space allocated to the TempDB log file that is not currently being used.
TempDB Log File Free Space	The amount of space allocated to the TempDB log file that is currently free.
TempDB Log File Utilized Space	The amount of space allocated to the TempDB log file that is currently being used.
TempDB Mixed Extents	The amount of space in TempDB used by mixed extents. Mixed extents are shared by up to eight objects.
TempDB User Objects	The amount of space in TempDB used by user objects. User objects are temporary objects explicitly created by users. They include temporary tables and indexes, temporary stored procedures, table variables, and cursors.
TempDB Version Store	The amount of space in TempDB used by the version store. While a table row is being updated or deleted, the version store contains the committed version of that row. `SELECT` operations that need to access the row being updated or deleted are not blocked because they can read the row in the version store. When the transaction is committed, the row is removed from the version store. Long-running or orphaned transactions can increase the size of the version store. A large version store can affect database performance because of the overhead of reading the large version store.

Metric	Description
Total TempDB Log File Size	The amount of disk space allocated for the log file in the TempDB database over time. Each time SQL Server is restarted, TempDB is re-created, and the log file is created using the default size or the size specified by the DBA. If the TempDB log file requires more space, by default it autogrows as needed. However, autogrowth can affect performance because the TempDB log file cannot be used during autogrowth, and because autogrowth can lead to file fragmentation. Use the Total TempDB Log File Size metric to: • Determine the size that the TempDB log file typically grows to over time and specify an initial size that prevents excessive autogrowth. • Identify sudden growth spikes and investigate what queries could have caused the spikes.

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the instance.

Amazon RDS for MySQL metrics collected by DPA

The following sections list the metrics that DPA collects for Amazon RDS for MySQL database instances. Some metrics are not collected for every instance.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.

Disk

This data reflects activity for sessions accessing tables managed by the InnoDB (or an InnoDB-based) storage engine.

Metric	Description
InnoDB Data Read Ops Rate	The number of InnoDB data read operations per second.

Metric	Description
InnoDB Data Write Ops Rate	The number of InnoDB data write operations per second.
InnoDB Log Write Rate	The number of requests per second to write to the InnoDB redo log. The general recommendation is to set the combined size of log files to about 25% through 100% of the buffer pool size to avoid unnecessary buffer pool flush activity on log file overwrite. A larger log file size will increase the time needed for a recovery process. If this is one of your top metrics, consider increasing the `innodb_log_file_size` in `my.cnf` and `my.ini` and then restarting MySQL.
InnoDB fsync Call Rate	The number of InnoDB fsync() system calls per second made to flush both the data and log files to disk.

InnoDB Logical I/O

This data reflects activity for sessions accessing tables managed by the InnoDB (or an InnoDB-based) storage engine.

Metric	Description
InnoDB Buffer Pool Read Rate	The number of logical read requests per second from the InnoDB buffer pool. High values usually indicate high load on the system. Reads from the buffer pool are efficient reads, so high rates only rarely indicate a performance problem.
InnoDB Buffer Pool Write Rate	The number of requests per second to write to the InnoDB buffer pool.

Metric	Description
InnoDB Row Read Rate	The number of rows that are read from InnoDB tables per second. An occasional spike in this rate can indicate that a mysqldump backup task is running. If you see a high InnoDB Row Read Rate that you believe is contributing to slow performance, consider optimizing the SQL to reduce the number of rows being read: 1. Go to the Trends page for the time frame and look at the statements with the highest wait time. 2. Determine which of the statements have the highest 'Rows Examined' value on the SQL Data tab. 3. For these statements, consider the following: • Use summary tables where possible to limit the number of rows processed. • Rewrite complicated queries to assist in processing fewer rows. • Evaluate WHERE clauses to ensure you process only rows that are required

Memory

This data reflects activity for sessions accessing tables managed by the InnoDB (or an InnoDB-based) storage engine.

Metric	Description
InnoDB % of Dirty Buffer Pool Pages	The percentage of InnoDB buffer pool data pages that have been changed in memory but have not yet been written (flushed) to disk.

Metric	Description
InnoDB Buffer Pool Consumed Space	The percentage of the InnoDB buffer pool that contains data. Innodb_buffer_pool_size is a very important parameter for InnoDB performance. A rule of thumb is to set the `innodb_buffer_pool_size` up to the total size of the database but not to exceed about 70% of total RAM. A general good practice is to size the buffer pool such that it is mostly full. By doing this, it indicates that you are not wasting memory and that queries are finding the majority of their data in the buffer pool. If this metric is either too high or too low, consider the following: • If the Consumed Space is consistently low, this indicates that your buffer pool is too big and memory is unnecessarily allocated to the buffer pool. Investigate lowering the `innodb_buffer_pool_size` variable. • If the Consumed Space is consistently very high (99% or higher), this may indicate that the size of the buffer pool is too low. Check the resource metric InnoDB Buffer Pool Hit Ratio. If this metric is periodically or consistently low, investigate increasing the `innodb_buffer_pool_size` variable. • If the Consumed Space is low, but it is on the rise, this indicates that the buffer is being initially populated with data. No action is needed at this point.
InnoDB Buffer Pool Data Pages	The number of pages that contain data in the InnoDB buffer pool. This includes both dirty and clean pages.

Metric	Description
InnoDB Buffer Pool Flushed Page Rate	The number of requests per second to flush pages from the InnoDB buffer pool to the data file. Flushing pages to disk is a normal InnoDB operation. InnoDB tries to do this activity in the background when the total load is low. If the flush rate is too high, consider the following: • If the InnoDB log files are too small, this forces a checkpoint operation that flushes buffer pool pages to disk. Check the InnoDB Log Write Rate metric. If you see a lot of log writes that correspond to high InnoDB Buffer Pool Flushed Page Rate values, increase the `innodb_log_file_size` variable. • A buffer pool size that is too small can cause frequent flushes. A rule of thumb is to set the `innodb_buffer_pool_size` up to the total size of the database but not to exceed about 70% of total RAM. ⓘ If you have MyISAM tables, balance the `key_buffer_size` and `innodb_buffer_pool_size` values to best utilize memory for your MySQL instance. • Check the load, mostly writes, on the system and investigate ways to decrease the load. Although SELECTs can also cause pages to be flushed from the buffer pool to disk, writes usually cause higher flush rates. • Optimize SQL to reduce the number of rows being written: 1. Go to the Trends page for the timeframe and look at the UPDATE, DELETE, and INSERT statements with the highest wait time. 2. Determine which statements have the highest Rows Affected or Sent value on the SQL Data tab. 3. For these statements, consider evaluating WHERE clauses to ensure you process only rows that are required.
InnoDB Buffer Pool Hit Ratio	The rate at which the InnoDB engine finds the data blocks it needs in memory rather than having to read from disk. Innodb_buffer_pool_size is a very important parameter for InnoDB performance. A rule of thumb is to set the innodb_buffer_pool_size up to the total size of the database but not to exceed about 70% of total RAM. ⓘ If you have MyISAM tables, you want to balance the key_buffer_size and the innodb_buffer_pool_size to best utilize memory for your MySQL instance. If the hit ratio is lower than 90%, investigate increasing the buffer pool in `my.cnf` and `my.ini` by updating the `innodb_buffer_pool_size` system variable and then restarting MySQL.

Network

Metric	Description
Bytes Received	Throughput of bytes received by MySQL from clients. If Bytes Received has an abnormal spike or if it is higher than normal in general, consider: 1. Examine the Bytes Received (KB/s) metric together with Bytes Sent (KB/s) to gain a more complete story of network traffic. 2. Check the LOAD DATA infile statements which can contribute to the network traffic. 3. Enlist the assistance of your network admin to evaluate network traffic, with a focus on the traffic between the Application server and the MySQL server.
Bytes Sent	Throughput of bytes sent from MySQL to clients. If Bytes Sent has an abnormal spike or if it is higher than normal, consider the following: 1. Examine the Bytes Received (KB/s) metric together with Bytes Sent (KB/s) to gain a more complete story of network traffic. 2. Optimize SQL to reduce network traffic. Go to the Trends page to identify which SQL statements have the highest wait time. Determine which of these statements have the highest Rows Affected or Sent statistic on the SQL Data tab. For these statements: • Evaluate WHERE clauses to ensure you are processing only rows that are required. • Eliminate columns from your result set that you don't need. • Use summary tables where possible to limit the number of rows processed/returned. • Rewrite complicated queries to assist in processing fewer rows. 3. Enlist the assistance of your network admin to evaluate network traffic, with a focus on the traffic between the Application server and the MySQL server.
Round-trip Time	The round-trip time when running "select 1" against this instance (includes network time but not connect time). If this is high, contact your network administrator to understand network latency.

Objects

Metric	Description
Table Cache Filled	The percentage of the cache that is filled with "file descriptors" (that is, an `.frm` file that contains a table's underlying format).
Table Cache Hit Ratio	The percentage of time that MySQL used an available cached "file descriptor" (that is, an `.frm` file that contains a table's underlying format). Whenever MySQL needs to access a table, it needs the table structure. The structures of previously opened tables are stored in the table cache. If a table's structure has not been cached, MySQL needs to load the structure from disk into cache, negatively affecting database performance. The lower this ratio is, the more the database has to load table structures from disk. Table structures are stored in .frm files on disk (tableName.frm). If the Table Cache Hit Ratio is low, increase the `table_open_cache` variable in `my.cnf` and `my.ini`. Recommendations: • Set `table_open_cache` to the total number of tables in the database. • A typical range for the `table_open_cache` is from 2000 (default) to 100,000.

Sessions

Metric	Description
Active Threads	The number of active threads in the database instance to support client connections. This metric is based on the MySQL Global Status variable `threads_running`. MySQL associates each client connection with a dedicated thread that handles all requests for that connection. This means that there are as many threads as there are clients currently connected. MySQL employs a thread cache to reduce the performance penalties associated with creating and destroying threads. The size of the thread cache is governed by the `thread_cache_size` system variable. When a connection is established, MySQL creates a new thread if an available thread cannot be found in the thread cache. When a connection ends, its thread is returned to the thread cache unless the cache is full. To monitor how many threads are being created because no cached thread is available, look at the Created Threads (sessions) metric. Each thread has some overhead in the form of server and kernel resources, including stack space, that affects the ability to scale to handle large numbers of connections. If you need to handle a large number of simultaneous connections, a common solution is to decrease the thread stack size. Doing so will limit memory-consuming activities conducted by the thread. If Active Threads is too high, consider the following: • Use connection pooling in your applications to reduce the number of simultaneous queries. • Use the MySQL master/slave architecture and move some or all SELECT queries to a slave. • MySQL may be incurring excess overhead such as memory. If you feel that this is a problem, you can decrease the thread stack size, but you need to realize that this will limit memory-consuming activities conducted by the thread. In other words, it limits complexity of SQL statements and stored program recursion depth. To set the stack size, start the server with `--thread_stack=N` where `N` is in bytes. • If the Active Threads value is higher than the thread cache size, MySQL may be incurring excess expense due to the creation and destruction of threads. If you feel that this overhead is a problem, you can try increasing the size of the thread cache by increasing the value of the `thread_cache_size` system variable.

Metric	Description
Blocked Threads	The number of threads that are blocked because another thread is holding a lock on an object, typically a table or an index. Drill down in the Trend page to locate additional details about what the blocking sessions are doing. Tune the queries you find by adding indexes or rewriting queries to minimize the time the locks are held.
Connection Attempts	The number of connection attempts in the given time interval (successful or not). This metric is based on the MySQL Global Status variable connections. If the Connection Attempts value is high, investigate the connection attempts in the logs. Enable logging of the connection attempts in the following ways: • If you are only interested in aborted attempts, make sure that the value (level) of the `log_warning` system variable is 2, and then check the error log. • If you are interested in successful and aborted connections, make sure that the general query log is enabled by checking the `general_log` system variable. The location of the general query log file is in the `general_log_file` system variable. Enabling the general query log can decrease the performance of the MySQL server, as every connection attempt and SQL statement will be logged.
Created Threads	The number of created threads in the database instance to support client connections in the given interval. This metric is based on the MySQL Global Status variable `threads_created`. MySQL associates each client connection with a dedicated thread that handles all requests for that connection. This means that there are as many threads as there are clients currently connected. Because thread creation and disposal can be expensive, MySQL employs a thread cache. When a connection is established, MySQL will create a new thread if an available thread cannot be found in the thread cache. When a connection ends, its thread is returned to the thread cache unless the cache is full. To monitor how many threads are currently running (cached or not), look at the Active Threads (sessions) metric. • If the Thread Creation Rate value is high and the thread cache is not full, this generally means that the cache is being filled, which is a normal situation. To see how many threads are in the cache and the size of the thread cache, look at the `threads_cached` and `thread_cache_size` system variables. • If the Thread Creation Rate value is high and the thread cache is full, this means that available threads are not being found in the thread cache, causing new connections to create new threads, which can be an expensive operation. If you think this overhead is causing problems, you can try increasing the size of the thread cache by increasing the value of the `thread_cache_size` system variable. • Consider using connection pooling in your application(s).

Sorts/Joins

Metric	Description
Joins By Table Scan	The number of joins that performed table scans (that is, joins that did not use indexes).
On-Disk Temp Table Creation Rate	The number of internal on-disk temporary tables created per second while executing statements.
Row Sort Rate	The number of rows sorted per second while executing statements. If MySQL cannot use an index to retrieve presorted rows, it performs a sort that increments the `sort_rows` counter. If this metric is high, consider these solutions: • Check the Sort Merge Passes resource metric and determine if there is a need to increase `sort_buffer_size`. • Optimize the SQL to reduce sorting. 1. Go to the Trends page for the time frame and look at the statements with the highest wait time. 2. Find the statements with the highest Rows Sorted value on the SQL Data tab. 3. For these statements, consider using a combined or covered index with the same columns in the same order as the ORDER BY clause. In some cases, MySQL can use an index to satisfy an ORDER BY clause without performing any extra sorting.

Metric	Description
Sort Merge Passes	The number of merge passes per second performed by the sort algorithm. A Sort Merge Pass occurs if sorting large amounts of data using a limited amount of space. Performance suffers when these sorts can not be performed in memory. When the sort buffer overflows, MySQL creates temporary files on disk to use in the file sorting and merging algorithm. The data is sorted in multiple passes to first sort small chunks of data before merging the results together. • Look at the Row Sort Rate metric. If there is a lot of sorting happening in this time frame, follow the suggested solutions. • Increase the global `sort_buffer_size` system variable to improve the performance of queries that sort a lot of data. • Increase the `sort_buffer_size` at the session level with a SET statement. Add the statement to your application code before running these kinds of queries. For example: `SET session sort_buffer_size = 8M` • Use an index with columns in the ORDER BY clause. MySQL might use this index to satisfy an ORDER BY clause without extra sorting.
Temp Table Creation Rate	The number of internal temporary tables created per second while executing statements. MySQL creates internal temporary tables to process operations such as SELECT ... GROUP BY / ORDER BY and SELECT DISTINCT. Unfortunately, temporary tables larger than the sizes specified in `tmp_table_size` and `max_heap_table_size` have to be converted to a slow, disk-based MyISAM temporary table. Likewise, if the query uses TEXT or BLOB fields, MySQL always has to use slow, disk-based temporary tables because in-memory temporary tables don't support those fields. If this metric is high, you run the risk of temporary tables being created on disk. Consider the following: 1. Go to the Trends page to identify which SQL statements have the highest wait time. 2. Find the statements with the highest Temp Tables Created statistic on the SQL Data tab. 3. For these statements: • Use a combined or covered index that has the same columns in the same order as the ORDER BY clause. In some cases, MySQL can use an index to satisfy an ORDER BY clause without doing any extra sorting. • Remove TEXT/BLOB fields if they are not needed for the query. Consider also increasing the `tmp_table_size` or `max_heap_table_size` values to reduce the number of internal temporary tables that have to be written to disk.

Statements

Metric	Description
Delete Statement Rate	The number of times a DELETE statement has been executed per second.
Insert Statement Rate	The number of times an INSERT statement has been executed per second.
Select Statement Rate	The number of times a SELECT statement has been executed per second.
Statements Execution Rate	The number of statements executed per second, not including those executed from stored programs. This is only a problem if your users complain about poor performance. Consider the following: • Identify and tune the queries with the highest wait time. • Look to see if there are a high number of executions of a SQL Statement. Look for possible ways to modify your application to decrease the number of executions, such as caching. • If you believe poor performance is due to the volume of statements being executed, consider implementing a MySQL master and slave architecture and move some or all SELECT queries to a slave.
Statements Execution Rate from Stored Programs	The number of statements executed per second from programs. This is only a problem if your users complain about poor performance. Consider these measures: • Identify and tune the queries with the highest wait time. • If you believe poor performance is due to the volume of statements being executed, consider implementing a MySQL master and slave architecture and move some or all SELECT queries to a slave.
Update Statement Rate	The number of times an UPDATE statement has been executed per second.

Waits

Metric	Description
Total Instance Wait Time	The total wait time for the instance.

Amazon RDS for PostgreSQL metrics collected by DPA

The following sections list the metrics that DPA collects for Amazon RDS for PostgreSQL database instances.

Learn how to view these metrics and change the thresholds.

Cache Eviction

Metric	Description
Dirty Buffers Evicted by Background Writer	The number of buffers written and freed (evicted) due to the PostgreSQL background writer.
Dirty Buffers Evicted by Background Writer Ratio	The ratio of buffers written and freed (evicted) due to the PostgreSQL background writer to the total number of evictions.
Dirty Buffers Evicted by Checkpoints	The number of buffers written and freed (evicted) due to a checkpoint execution. Higher values indicate an increased need for checkpoints.
Dirty Buffers Evicted by Checkpoints Ratio	The ratio of buffers written and freed (evicted) due to a checkpoint execution to the total number of evictions. Higher values indicate an increased need for checkpoints.
Dirty Buffers Evicted by Client Backends	The number of times client backends were delayed by being forced to write and free (evict) buffers themselves, instead of the buffers being evicted asynchronously by the background writer.
Dirty Buffers Evicted by Client Backends Ratio	The ratio of buffers written and freed (evicted) by a client backend to the total number of evictions.
Total Dirty Buffers Evicted	The total number of dirty buffers evicted by checkpoints, background writer, and client backends.

Checkpoints

Metric	Description
Requested Checkpoints	The number of unscheduled checkpoints requested by client statements because the WAL size has reached its threshold (`max_wal_size`). Requested checkpoints can cause client backend waits. Consider reconfiguration of checkpoint related settings (`checkpoint_timeout`, `checkpoint_completion_target`, and `max_wal_size`).
Requested Checkpoints Ratio	The ratio of requested checkpoints to total checkpoints (requested and scheduled). The percentage should be low—optimally 0%.
Scheduled Checkpoints	The number of scheduled checkpoints processed in the background, without affecting client statements. Scheduled checkpoints should not cause client backend waits.

Connections

Metric	Description
Connected Devices	The number of distinct client machines connected to this database instance, even if the connection is idle.
Connected Users	The number of distinct users (that is, login names) connected to this database instance, even if the connection is idle.

Disk

Metric	Description
Blocks Hit	The number of times disk blocks were found already in the buffer cache, so that a read was not necessary. This includes only hits in the PostgreSQL buffer cache, not the operating system's file system cache.
Blocks Read	The number of disk blocks read in this database.
Blocks Read Time	The average amount of block read I/O during the specified time interval. If the `track_io_timing` parameter is off, the value of this metric is always 0. For more information, see this KB article.
Blocks Write Time	The average amount of block write I/O during the specified time interval. If the `track_io_timing` parameter is off, the value of this metric is always 0. For more information, see this KB article.

Metric	Description
Temp Bytes Written	The total amount of data in kilobytes written to temporary files by queries in this database. All temporary files are counted, regardless of why the temporary file was created, and regardless of the `log_temp_files` setting.
Temp Files	The number of temporary files created by queries in this database. All temporary files are counted, regardless of why the temporary file was created (for example, sorting or hashing), and regardless of the `log_temp_files` setting.

Memory metric

Metric	Description
Buffer Cache Hit Ratio	The rate at which PostgreSQL finds the data blocks it needs in memory rather than having to read from disk.

Network metric

Metric	Description
DB Round-trip Time	The round-trip time when running "select 1" (includes network time but not connect time) on this database.

Replication metric

Metric	Description
Replication Lag	The replication lag between the primary database and all replica databases. An increase in the replication lag indicates a growing number of transactions that are not yet replicated and at risk of not being replicated if the primary database fails.

Rows metrics

Metric	Description
Fetched vs. Returned Rows	Of the total number of rows that were scanned (Rows Returned), the percentage that contained data needed to execute the query (Rows Fetched). High values indicate that the database is executing queries efficiently. Low values indicate that the database is performing extra work because it is scanning a large number of rows that aren't required to process the query. For example, 10% means that the database is scanning 10 rows to use 1 row. Low values could indicate inefficient queries or missing indexes.

Metric	Description
Rows Operations	The number of rows inserted, updated, and deleted by queries in this database instance.
Rows Deleted	The number of rows deleted by queries in this database instance.
Rows Fetched	The subset of scanned rows (Rows Returned) that contained data needed to execute the query. For example, take the following query: `SELECT * FROM customers WHERE country = 'Spain';` The `customers` table has 10,000 rows, and `country = 'Spain'` in 100 rows. The column is not indexed, and so a full table scan is required. The Rows Returned value is 10,000, but the Rows Fetched value is only 100. ⓘ The Rows Fetched value is different than the number of rows returned to the client.
Rows Inserted	The number of rows inserted by queries in this database instance.
Rows Returned	The total number of rows scanned by queries executed against this database instance. ⓘ This value indicates rows returned by the storage layer to be scanned, **not** rows returned to the client.
Rows Updated	The number of rows updated by queries in this database instance.

Sessions metrics

Metric	Description
Active Sessions	The number of transactions in the following state: The backend is executing a query.
All Sessions	All sessions, regardless of state.
Blocked Sessions	The number of sessions in a blocked state.
Deadlocks	The number of deadlocks detected in this database instance.

Metric	Description
Fastpath Function Call Sessions	The number of sessions in which the backend is executing a fast-path function.
Idle Sessions	The number of sessions in the following state: The backend is waiting for a new client command.
Idle in Transaction (Aborted) Sessions	This state is similar to Idle in Transaction Sessions, except one of the statements in the transaction caused an error.
Idle in Transaction Sessions	The number of sessions in the following state: The backend is in a transaction, but is not currently executing a query.
Percentage of Idle in Transaction	The percentage of allowed connections that are in the idle-in-transaction state. Transactions that are in the idle-in-transaction state for a significant amount of time can cause the connection pool to fill to the limit, and they can cause other database operations such as VACUUM to fail to complete. You might also see `SELECT waiting` messages, which indicates that an HTTP request was made but the SELECT operation to get the data never completed (probably because the user waited so long that they abandoned the task).
Recovery Conflicts	The number of queries canceled due to conflicts with recovery in this database instance. Conflicts occur only on standby servers.
Transaction Commit Rate	The number of transactions being committed every second in this database instance.
Transaction Rate	The number of transactions being executed every second in this database instance.
Transaction Rollbacks	The number of transactions in this database that have been rolled back.
track_ activities Disabled Sessions	This state is reported if `track_activities` is disabled in this backend.

Vacuum metrics

Metric	Description
Autovacuum Worker Utilization	An indication of how busy the set of vacuum worker processes are. The `pg_stat_progress_vacuum` view provides information about current vacuuming processes. If this value is consistently high, consider increasing the value of the `autovacuum_max_workers` parameter.
Multixact ID Space Taken	The percentage of space available to store Multixact IDs (MXIDs) that is currently filled. this is the highest value across all databases in the database server.
Transaction ID Space Taken	The percentage of space available to store Transaction IDs (XIDs) that is currently filled. this is the highest value across all databases in the database server.

Waits metric

Metric	Description
Total Instance Wait Time	The total wait time for the database instance.

VM metrics collected by DPA

The following sections list the metrics that DPA collects from virtual machines (VMs). For database instances that run on a VM, these metrics are displayed in addition to the metrics collected for the database type.

DPA collects VM metrics only if you register the VM for monitoring.

- Learn how to view these metrics and change the thresholds.
- For detailed information about resolving issues, click the ⓘ next to the metric on the Resources tab. The Information link is not available for all metrics.
- If the database instance runs on a virtual machine (VM), metrics collected for the VM are described in VM metrics collected by DPA

CPU

Metric	Description
Host CPU Usage	Actively used CPU as a percentage of the total available CPU on the machine. If this metric is high, determine if the VM CPU Ready Time is also high.

Metric	Description
VM Co-Stop	The percentage of time the VM has been waiting on physical CPU resources allocated to other VMs. If this value is above 3%, consider the actions listed above to reduce co-stop delays.
VM CPU Ready Time	The percentage of time that the virtual machine was ready to use CPU resources, but could not get scheduled to run on the physical CPU. This value is the average across all CPUs. If this metric is high, check the Host CPU Usage: • If Host CPU Usage is normal, the VM could have under-allocated CPU resources. If the VM has been configured with a CPU limit, consider raising or removing the limit. Or use resource controls to give higher priority to this VM, which will allocate more CPU resources to it. • If Host CPU Usage is high, this could indicate a host over-commitment of CPU resources. Consider: ◦ Reducing the number of VMs running on the host. ◦ Increasing the available CPU resources by adding the host to a DRS cluster. ◦ Increasing the efficiency with which VMs use CPU resources by tuning SQL statements and non-database applications. ◦ Using resource controls to direct available resources to critical VMs.
VM CPU Usage MHz	The average amount of CPU (in MHz) actively used by the VM (for all vCPUs configured for the VM). This is the host's view of the CPU usage, not the guest operating system view, so the values may differ. Typically the host view of CPU is more accurate.

Metric	Description
VM CPU Usage Percentage	Actively used CPU as a percentage of the total available virtual CPU in the virtual machine. Note that this is the host's view of the CPU usage, not the guest O/S view, so the values may differ. If this value is high, check the VM CPU Ready Time: • If VM CPU Ready Time is also high, the host has under-allocated CPU resources to the VM. (See VM CPU Ready Time below.) • If VM CPU Ready Time is not high and you are not experiencing a performance problem, high CPU usage values are not a cause for concern. • If VM CPU Ready Time is not high but you are experiencing a performance problem, you can address the issue in either of the following ways: ◦ Increase the CPU resources provided to the VM. To do this, you can add vCPUs to the VM, migrate the VM to a host with more powerful processors, or add additional VMs running the same application and then balance the workload. ◦ Increase the efficiency with which the VM uses CPU resources. To do this, you can tune the queries with long Memory/CPU wait types or tune the non-database applications using the most CPU.
VM Total Co-Stop Time	The amount of wait time incurred because the VM in which the database is running has to wait on physical CPU resources allocated to other VMs. If the database instance is on a VM configured to use multiple vCPUs, co-stop delays can cause long Memory/CPU wait times. Co-stop delays occur when a VM is not being scheduled to run consistently because it has to wait on vCPU resources to be freed from other VMs contending for those vCPUs. If this value is not near 0, consider taking one of the following actions to reduce co-stop delays: • Decrease the number of vCPUs on the VM. • Add additional CPUs to the pool available to the VMs. • Use vMotion to migrate other VMs to a different host to reduce contention.
VM Total CPU Usage Time	The total amount of time (in milliseconds) that the VM spent using the virtual CPUs (that is, the sum of time spent on each virtual CPU during the time period).

Disk

Metric	Description
Host Disk Read Rate	The average rate at which data is read from each LUN on the host. `read rate = # blocks read per second x block size` If your database instance is suffering from disk I/O performance related issues, it's possible that another VM on the same host is consuming high amounts of disk resources and causing delays for this VM. To understand that relationship, check the Physical I/O rate from the database instance compared to this metric. If this metric is much higher than the database metric, another VM might be causing the issue. If not, this VM might be putting to many demands on the underlying disk devices.
Host Disk Write Rate	The average rate at which data is written to each LUN on the host. `write rate = # blocks written per second x block size`
Host Max Total Disk Latency	The highest latency value across all disks used by the host. Latency measures the time taken to process a disk command issued by the guest OS to the virtual machine. High latency is a key indicator of slow storage.
VM Disk Bus Resets	The number of disk bus reset commands issued. This typically occurs when storage demand is excessively high, or when storage is not properly configured to handle the I/O load. Bus Resets occur when the disk subsystem times out and commands are canceled and retried. This happens when the HBA device is overloaded or its queue depth is exhausted.
VM Disk Commands	The number of disk commands issued by the virtual machine. High disk usage could be due to guest swapping, which you can investigate using OS analysis tools. VMs configured with insufficient memory can also cause excessive guest swapping and, in turn, high disk usage.
VM Disk Commands Aborted	The number of disk commands that were aborted. This typically occurs when storage demand is excessively high, or when storage is not properly configured to handle the I/O load. Beyond re-balancing load, there is typically little that can be done from within vSphere to solve problems related to slow or overloaded storage. Follow the guidelines from your storage vendor to monitor the demand being placed on the storage device, and follow the vendor-specific configuration recommendations to configure the device for the demand. If the device is not capable of satisfying the I/O demand with good performance, distribute the load among multiple devices, or obtain faster storage.

Metric	Description
VM Disk Read Rate	The average rate at which data is read from each virtual disk on the virtual machine. `read rate = # blocks read per second x block size`
VM Disk Usage Rate	The average disk I/O rate across all virtual disks on the virtual machine.
VM Disk Write Rate	The average rate at which data is written to each virtual disk on the virtual machine. `write rate = # blocks read per second x block size`

Disk Device

Metric	Description
Host Disk Device Read Latency	The average time taken to process a SCSI read command issued from the Guest OS to the virtual machine (across all VMs). Expected disk latencies depend on the nature of the storage workload (for example, read/write mix, randomness, and I/O size) and the capabilities of the storage subsystems.
Host Disk Device Read Rate	The average rate at which data is read from a specific LUN on the host (across all VMs on the host).
Host Disk Device Write Latency	The average time taken to process a SCSI write command issued from the Guest OS to the virtual machine (across all VMs).
Host Disk Device Write Rate	The average rate at which data is written to a specific LUN on the host (across all VMs on the host).

Memory

Metric	Description
Host Memory Usage	The actual memory usage on the host (total consumed memory / total machine memory). High host memory usage is not necessarily a problem, but could indicate host memory over-commitment (or looming over-commitment). Check to see if memory swapping is occurring by looking at memory swap in/out rates, which is a clear indicator of host memory over-commitment.

Metric	Description
VM Active Memory Usage	Memory that is actively in use (that is, used currently or in the recent past) as a percentage of virtual machine configured memory. While a VM may have been allocated large amounts of memory, it is possible that the OS and applications are only using a small percentage of what the VM was assigned. Inactive memory is subject to being "ballooned" (reclaimed by other VMs) when memory is scarce. When the active memory for all VMs exceeds the total host memory, it indicates host memory saturation. As a result, host-level memory swapping typically occurs.
VM Memory Balloon	The percentage of the virtual machine memory that is currently claimed by the balloon driver. This is not necissarily a performance problem, but shows the host starting to take memory from VMs that need less memory and assigning it to VMs with large amounts of active memory. If high amounts of ballooning are occurring, check for high Memory Swap In/Out Rates which would indicate performance problems.
VM Memory Balloon Size	The amount of virtual machine memory that is currently claimed by the balloon driver. If high amounts of ballooning are occurring, check for high Memory Swap In/Out Rates which would indicate performance problems.
VM Memory Granted	Memory that has been given to the virtual machine by the host, not including overhead. Typically, VMs are granted increasing amounts of memory over time until reaching the configured VM memory size.
VM Memory Overhead	The amount of memory used to run the virtual machine. Configuring a virtual machine with excess memory or excess virtual CPUs will unnecessarily increase the overhead.
VM Memory Swap In Rate	The rate at which memory is swapped in from disk. A value greater than 0 indicates that performance is suffering due to lack of memory. This is typically caused by memory being previously swapped out, memory over-commitment (many virtual machines with high amounts of active memory), or a problem with the balloon driver. Consider the following possible solutions: • Reduce the level of memory over-commit. • Enable the balloon driver in all VMs. • Reduce memory reservations. • Use resource controls to dedicate memory to critical VMs.

Metric	Description
VM Memory Swap Out Rate	The rate at which memory is swapped out to disk. High values indicate a problem with lack of memory that is causing performance to suffer. This is typically caused by either memory over-commitment (many virtual machines with high amounts of active memory) or a problem with the balloon driver.

Network

Metric	Description
Host Dropped Received Packets	The number of dropped received packets across all physical NICs on the host.
Host Dropped Transmitted Packets	The number of dropped transmitted packets across all physical NICs on the host. The following problems can cause the guest OS to fail to retrieve packets quickly enough from the virtual NIC: • High CPU utilization • Guest OS driver configuration Solutions are all related to ways of improving the ability of the guest OS to quickly retrieve packets from the virtual NIC. You can: • Increase the CPU resources provided to the VM. • Increase the efficiency with which the VM uses CPU resources. • Tune network stack in the Guest OS. • Add additional virtual NICs to the VM and spread network load across them.
VM Data Receive Rate	The average rate at which data is received on the virtual machine. This represents the receive bandwidth of the network.
VM Data Transmit Rate	The average rate at which data is transmitted on the virtual machine. This represents the transmit bandwidth of the network.
VM Network Packets Received	The number of packets received across all vNICs on the virtual machine.
VM Network Packets Transmitted	The number of packets transmitted across all vNICs on the virtual machine.

DPA user accounts

See the following topics to create user accounts, assign privileges, and specify how users will log in to DPA.

DPA roles and privileges

When you add user accounts in DPA, you assign each user a role. The role determines the user's privileges.

Administrator role

By default, administrators have access to all DPA functionality, including all setup, administration, and support options. You have the option of removing user administration privileges from the Administrator role in order to limit user management to the User Manager role.

DPA requires at least one Administrator account, which is created during installation.

Only administrators can perform certain actions, such as:

- Register and unregister database instances and VMs
- Allocate licenses
- Run advanced support utilities
- Edit system-wide Advanced Options
- Start and stop all monitors
- Create, edit, and delete report schedules
- Create, edit, and delete alert groups
- Create, edit, and delete email templates for alert notifications
- Configure the mail server
- Create and manage contacts and contact groups
- Create and manage custom properties
- View logs

User Manager role

Accounts with the User Manager role have privileges to create and manage DPA user accounts, but they cannot view data collected by DPA or perform any other DPA tasks.

By default, accounts with the Administrator role also have privileges to create and manage DPA user accounts. To enforce a strict separation of duties, you can remove user account management privileges from the Administrator role.

ⓘ The Repository Owner always retains user account management privileges.

Read Only on All Instances role

Users with the Read Only role can perform the following actions for **all** database instances:

- View performance data and metrics
- Run reports and view existing report groups
- View existing alerts
- View annotations

Custom Privileges role

The Custom Privileges role specifies which privileges a user has, and which database instances these privileges apply to. Use this role to:

- Prevent users from seeing data about certain database instances
- Give users privileges to manage monitoring options, alerts, and reports without granting them full administrative privileges

When you assign this role to a user, you can grant any of the following privileges. Privileges can apply to all database instances or only selected instances.

Privilege	Actions allowed against selected database instances
View Data	• View performance data and metrics • Run reports and view existing report groups • View annotations
Manage Reports	Create, edit, and delete report groups
View Alerts	View existing alerts
Manage Alerts	• Create, edit, and delete alerts • View existing alert groups

Privilege	Actions allowed against selected database instances
Manage Monitoring	• Create, edit, and delete blackout periods for monitoring • Manage I/O configuration • Update Advanced Options for a specific database instance • Add annotations • Exclude SQL statements from trend charts • Start and stop individual monitors ⓘ Users with Manage Monitoring permissions cannot see the charts at the top of the DPA home page.

Create a DPA user account and assign privileges

You must add a user account for each person who needs to log in to DPA. Each user is assigned a role, which determines the user's permissions.

ⓘ Optionally, you can integrate DPA with your company's Active Directory (AD) or LDAP service. If you do this:

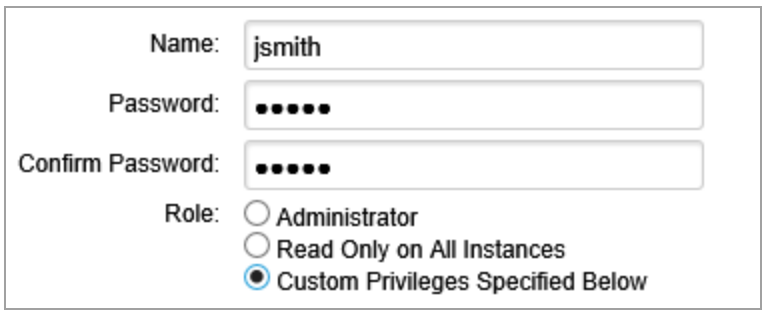
- Users can log in to DPA with their domain accounts.
- You can add AD or LDAP groups to DPA and assign privileges to each group.

Before you add users, determine who needs access to DPA and which privileges each user needs. For more information about the available options, see DPA roles and privileges.

ⓘ Only users with the Administrator or User Manager role can create and manage user accounts.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Users & Contacts, click User Administration.
3. On the User Administration page, click Create User.
4. Enter a unique user name.

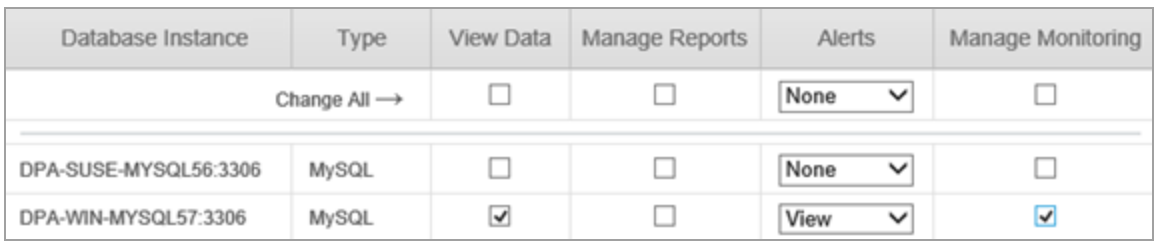
5. Enter a password that meets the following requirements:
   - Must begin with an alphabetic character.
   - Can contain **only** the following characters:
     - Upper- and lowercase alphabetic characters from the English alphabet
     - Numbers 0 through 9
     - The following special characters: `#$*_-~/.,+^@!<>\`
   - Cannot contain spaces, single quotation marks, double quotation marks, or any other special characters that are not listed above.
6. Specify the user's privileges:
   - To assign privileges associated with predefined roles, select Administrator or Read Only on All Instances.
   - To assign custom privileges:
     a. Select Custom Privileges Specified Below.



     b. To grant access to data from all database instances, select privileges in the top row. To limit access, select privileges for each database instance.

> The View Data privilege is automatically selected when you select any higher privilege.

In the example below, the user can access data from only one database instance. This user can make changes to monitoring options, run reports, and view existing alerts for the selected instance.



7. Click Save.

If you configured DPA to point to your Active Directory or LDAP server, you will see an option to either create a user or a group. The group corresponds to a group in Active Directory or LDAP.

Limit user account management to the User Manager role

By default, both the Administrator role and the User Manager role have privileges to create and manage DPA user accounts. To enforce a strict separation of duties, you can remove user account management privileges from the Administrator role. Only users with the User Manager role will be able to manage user accounts.

1. Open the following file in a text editor:

   `DPA-install-dir\iwc\tomcat\ignite_config\idc\system.properties`

2. Set the following property to `true`:

   `com.solarwinds.dpa.userManagement.strictSeparationOfDuties=true`

3. Save the file and restart DPA.
4. Assign the User Manager role to one or more DPA user accounts.

When the `strictSeparationOfDuties` property is set to `true` and at least one DPA user is assigned the User Manager role, the Administrator role does **not** have privileges to create and manage user accounts.

After you complete this procedure, Administrator role privileges vary depending on whether the User Manager role is assigned. This ensures that at least one DPA user has account management privileges:

- If any DPA user is assigned the User Manager role, then the Administrator role does **not** have privileges to create and manage user accounts.
- If the User Manager role is **not** assigned to any user, then the Administrator role **does** have user account management privileges.

ⓘ The Repository Owner always retains user account management privileges, regardless of whether the User Manager role is assigned.

DPA user authentication

DPA offers the user authentication options described in the following sections.

AD and LDAP

DPA supports Active Directory (AD) and Lightweight Directory Access Protocol (LDAP) authentication. Using your existing authentication infrastructure eliminates the need to duplicate your user accounts in DPA. After you configure AD or LDAP authentication, users can log in with their domain account or a custom user account created by DPA.

AD user authentication

DPA integrates with Windows Active Directory (AD). DPA uses the security group information from AD to assign permissions to groups. To configure DPA user authentication and permissions using AD, see Configure Active Directory or LDAP.

If your repository database is Azure SQL and you are monitoring one or more Azure SQL databases, you can Microsoft Entra ID (previously Azure AD) authentication in DPA. To configure DPA user authentication and permissions using Microsoft Entra ID, see Use Azure AD authentication in DPA.

LDAP user authentication

DPA integrates with most LDAP implementations to assign permissions to groups. To configure DPA user authentication and permissions using LDAP, see Configure Active Directory or LDAP.

Single sign-on

Using single sign-on (SSO), your AD users can log in to DPA without re-entering the domain credentials they used to log in to their operating system. Before you configure DPA for SSO, configure DPA for AD authentication.

Common Access Cards

You can use a Common Access Card (CAC) to log in to Windows and DPA. Before using a CAC, configure DPA for AD, and then for SSO as described in the sections above.

SAML authentication

SAML authentication in DPA offers single sign-on (SSO) and the opportunity to use different credential storage or multifactor authentications using third-party providers like Okta, Microsoft Entra ID, or Keycloak.

What is SAML?

The Security Assertion Markup Language (SAML) is an open standard for exchanging *authentication* and *authorization* data between an identity provider (IdP) and a service provider (SP). The most common use of SAML is web browser single sign-on (SSO). DPA supports SAML 2.0.

ⓘ Authentication is determining that the users are who they claim to be. Authorization is determining if users have the right to access certain systems or content.

Configure DPA to use Active Directory or LDAP

To use AD or LDAP user authentication in DPA:

1. Gather the following information from your domain administrator:
    - Directory service type: AD or LDAP
    - Domain name
    - Port number: Used to connect to the directory service
    - If DPA is configured to use credentials stored in CyberArk, the CyberArk credentials query
    - If DPA is **not** configured to use credentials stored in CyberArk:
        - User: The domain user DPA uses to query the directory for users and groups
        - Password: The password of the domain user, preferably one that does not expire
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Configure AD/LDAP.
4. Select the type of directory service you have: Active Directory or LDAP.
5. Click Next.

Connection information

Domain name

Enter the domain name.

ⓘ SolarWinds recommends using a domain name, not the name of a specific domain controller.

Do you have multiple domains?

If your domain users authenticate from a different domain other than the domain specified here, you must connect to the global catalog ports 3268 or 3269. The domain users must belong to a universal group, and that universal group must be added under Options > Administration > Users & Contacts > User Administration.

Port

Select the port number.

If you use a unique port, select Other non-standard port. Enter the port number, and select SSL if required.

User and Password

ⓘ If DPA is configured to use credentials stored in CyberArk, the User and Password fields are **not** displayed. The Credential query field is displayed instead.

DPA uses this user to search the directory service for users and groups.

Active Directory user name

For the AD user name, use one of the following formats:

- Distinguished Name (DN): `cn=BobSmith,cn=Users,dc=domain,dc=local`
- User Principal Name (UPN): `bsmith@domain.local`

ⓘ See this article for information about valid characters for Active Directory user names.

LDAP user name

For the LDAP user name, use the following format:

- Distinguished Name (DN): `cn=BobSmith,cn=Users,dc=domain,dc=local`

Credential query

If DPA is configured to use credentials stored in CyberArk, enter the CyberArk credentials query.

Did the connection test fail?

If you use an SSL port and the verification fails, DPA must import its certificate. Click Yes on the confirmation window to try again.

Base search location

Base DN

Use the default

SolarWinds recommends selecting the default, so DPA uses the detected base DN from the previous step.

Example of default base DN: `dc=east,dc=acme,dc=com`

Use a custom value

You may use a value other than the default base DN. For example: You use a global catalog that supports multiple domains, and you want to broaden the scope of the search.

Example for multiple domains: `dc=acme,dc=com`

Advanced settings

If this is your first time using this wizard, do not use the advanced settings.

Only use advanced settings if you completed this wizard and you experience slow domain user logins or group searches.

Are domain user logins slow?

Set the User Search Base value if domain user logins take a long time.

If your company has one domain, specify the location in the directory tree that contains all of the domain users that will use DPA.

If you do not know what to put here, ask the domain administrator of your company the following questions:

"What folder, or organization unit (OU), in the directory tree of the domain contains all of the users? I must specify a search base for users. What is the distinguished name of the folder?"

Example: `cn=users OR ou=users`

Are domain group searches slow?

Set the Group Search Base value if domain group searches in User Administration take a long time.

Specify the location in the directory tree that contains all of the groups to which DPA users belong.

If your company has multiple domains, you can enter the group search bases individually. After you add groups to DPA using the group search base from one domain, update this wizard to specify a group search base in another domain.

If you do not know what to put here, ask your the domain administrator of your company the following:

"What folder, or organization unit (OU), in the directory tree of the domain contains all of the groups? I must specify a search base for groups. What is the distinguished name of the folder?"

Example: `cn=groups OR ou=groups`

Summary

Confirm the information for configuring DPA with your directory service, and click Finish.

> You must restart DPA for the settings to take effect.

Configure authentication and permissions for groups of users

After you have set up DPA to use Active Directory or LDAP, do the following:

1. In AD or LDAP, determine which groups contain the users that you want to grant access to DPA. You may need to create a group if a suitable group does not exist.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click User Administration.
4. Click Add Active Directory Group or Add LDAP Group.
5. Click Search for a Group.
6. Find and select the group you want and click Save.
7. Assign privileges to the group, just as you would for a user. This assigns those permissions to the domain users who are members of the group.

   > DPA does not support single sign-on (SSO) for individual accounts. It only supports AD or LDAP groups.

8. Click Save.

   All domain users in the selected group can log in to DPA using their domain credentials. The users have the privileges you set up for the group in DPA.

You can add multiple AD or LDAP groups in DPA. If a domain user is a member of more than one group, DPA grants them the combined privileges from all of their groups.

Log in to DPA

When you enter the domain user name and password in the DPA login screen, DPA searches your directory service for a matching user name, and then authenticates using the password. If the domain user belongs to one of the groups that you configured as a DPA custom user, the login succeeds.

Name formats for AD login

DPA supports three types of login name formats for Active Directory:

- SAM account name: `username`
- User principal name: `username@domain.local`
- NT/AD: `domain\username`

User name for LDAP

The user name used by DPA is the LDAP user object `uid` attribute.

Configure DPA to use SAML authentication with Okta

SAML authentication in DPA offers single sign-on (SSO) and the opportunity to use different credential storage or multifactor authentications using third-party providers like Okta, Microsoft Entra ID (previously Azure AD), or Keycloak. Complete the following tasks to configure SAML authentication and single sign-on with Okta as the identity provider.

> If DPA is running behind a load balancer (or API Gateway) and you want to enable SAML SSO authentication in DPA, you **must** enable SSL communication between the load balancer (or API Gateway) and DPA.

(Optional) Configure SAML keystore properties

By default, the keystore file from the classpath resource (`saml.keystore`) is used for SAML authentication. If you use the default keystore file, you do **not** need to modify the SAML keystore properties.

If you would like to use a different keystore file, specify values for the following properties in the `system.properties` file.

1. Open the following file in a text editor:

   DPA-install-dir`\iwc\tomcat\ignite_config\idc\system.properties`

2. Add or uncomment the following properties and specify the values.

Property	Value
com.confio.security.saml.keystore.path	The keystore file path.
com.confio.security.saml.keystore.password	The password of the keystore file.
com.confio.security.saml.keystore.privatekey.alias	The alias of the private key stored in the keystore file.
com.confio.security.saml.keystore.privatekey.password	The password of the private key certificate added in the keystore file. The value for this property can be the same as the value for the property com.confio.security.saml.keystore.password. If the private key certificate does not have a password, uncomment or add the property but do not enter a value.
com.confio.security.saml.keystore.type	The file type of the keystore file. This property is optional. If the custom keystore file is not JKS or PKCS12, use this property to specify the type.

3. Restart DPA for the new properties to take effect.

Prepare the identity provider (IdP): Okta

When configuring Okta to communicate with DPA, you will be working with both Okta and DPA at the same time. You must keep both systems open to copy information from one system into the other.

Before you start

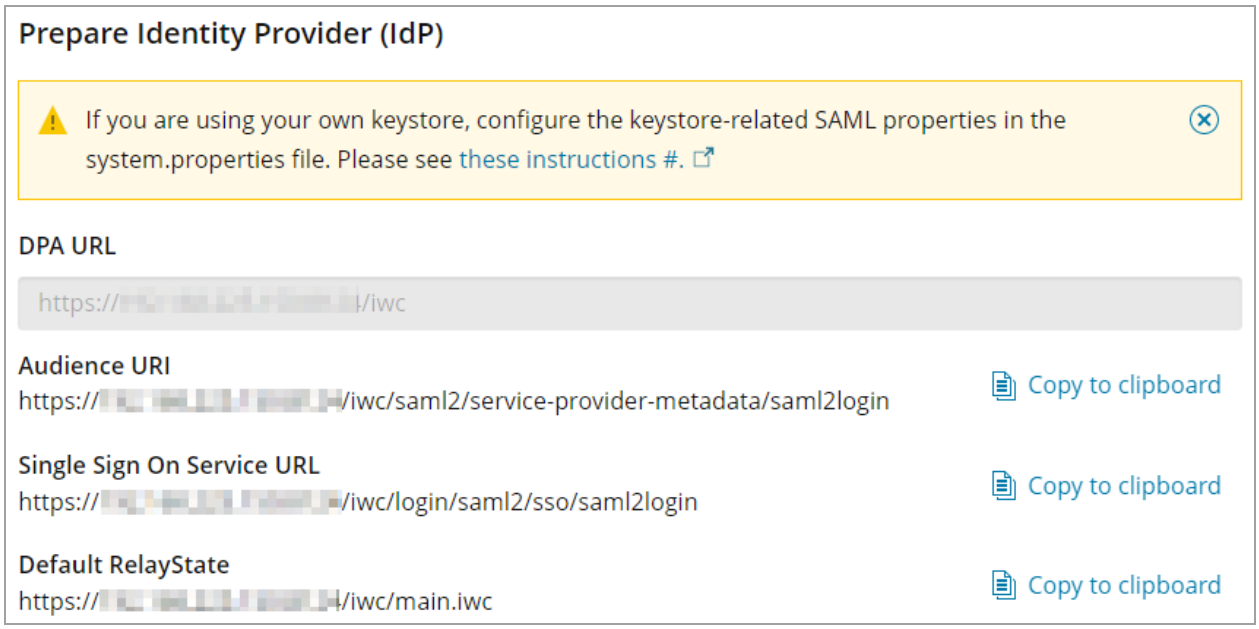
- DPA must be configured to use SSL to protect data during transmission. To enable SSL for DPA, see Configure DPA to use a custom certificate for SSL/TLS.
- DPA must be running on an HTTPS connection.
- If you do not want to use the default keystore file (`saml.keystore`), configure the SAML keystore properties in the `system.properties` file.

Task 1: In DPA, obtain the identity provider URL and URI

1. Log in to DPA as a user with administrative privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Configure SAML.

   On the Prepare Identify Provider (IdP) page, the following information is automatically added:

   - DPA URL
   - Audience URI
   - Single Sign On Service URL
   - Default RelayState

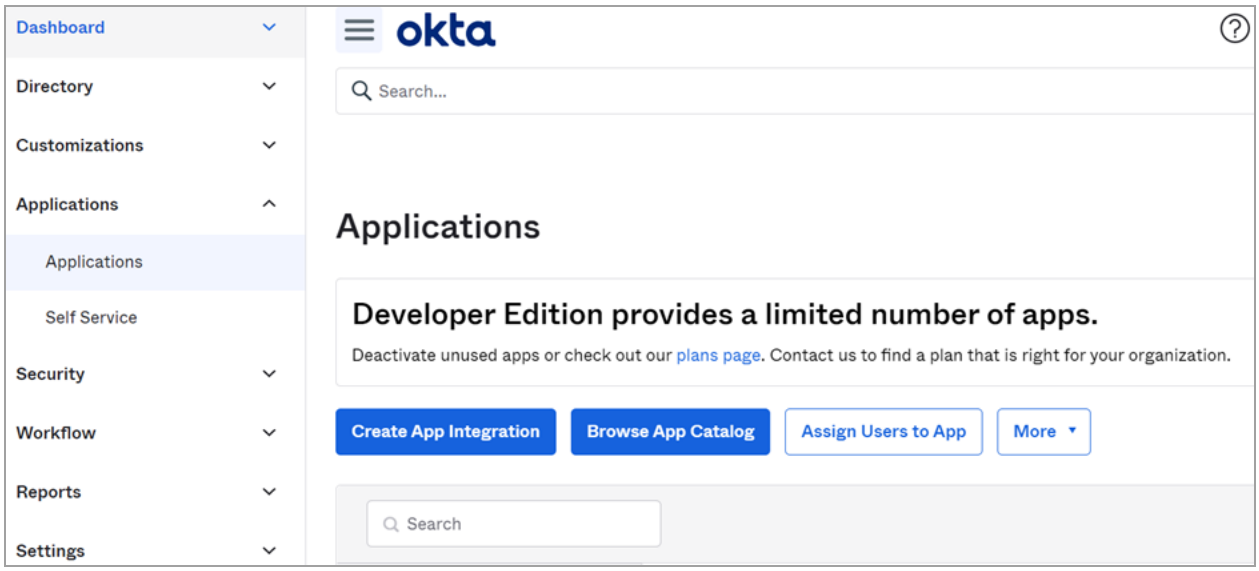


   You will copy and paste this information into the configuration in Okta.

4. Keep DPA open, and continue in Okta.

Task 2: In Okta, create the SAML application, configure URLs and the URI, and specify users for SAML login

1. Log in to your Okta organization with administrative privileges.
2. In the left pane of the Admin Console, click Applications > Applications.
3. Click Create App Integration.

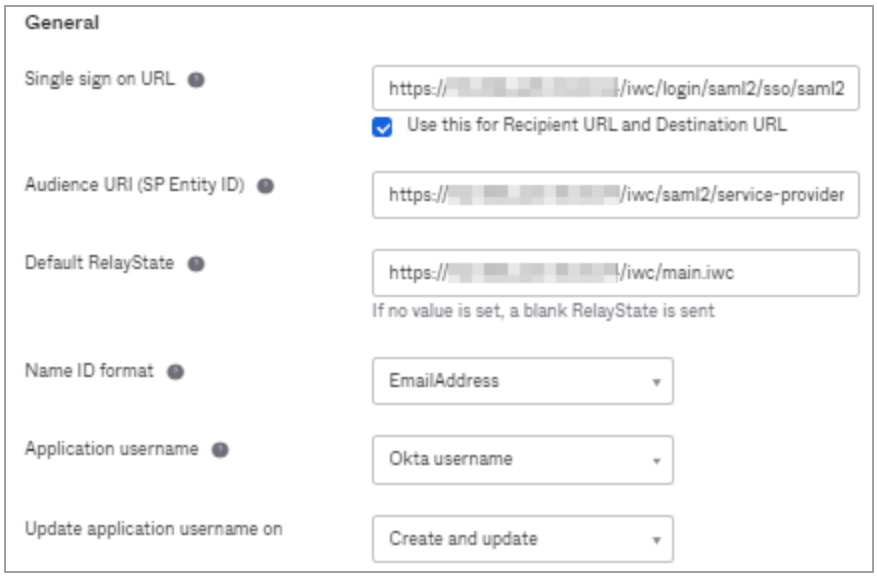


4. Select the SAML 2.0 option, and click Next.
5. In General Settings, enter a name for your SAML integration, and click Next.

6. In the SAML Settings section, make the following changes:

   a. In the General section, paste the following values from DPA into Okta:

DPA field	Okta field	Notes
Single Sign On Service URL	Single sign on URL	This SAML URL is used for the Recipient URL and Destination URL. This is a location where the SAML assertion is sent with an HTTP POST. This is often referred to as the SAML Assertion Consumer Service (ACS) URL for your application.
Audience URI	Audience URI (SP Entity ID)	This is the intended audience of the SAML assertion. This is most often the SP Entity ID of your application.
Default RelayState	Default RelayState (Optional)	This is default landing page in the IDP initiated flow.

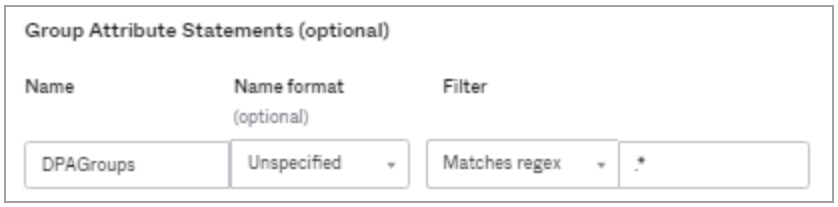


   b. In the Attribute Statements section, add the following attribute statements:

Name	Name format	Value
Email	Unspecified	user.email
FirstName	Unspecified	user.firstName
LastName	Unspecified	user.lastName

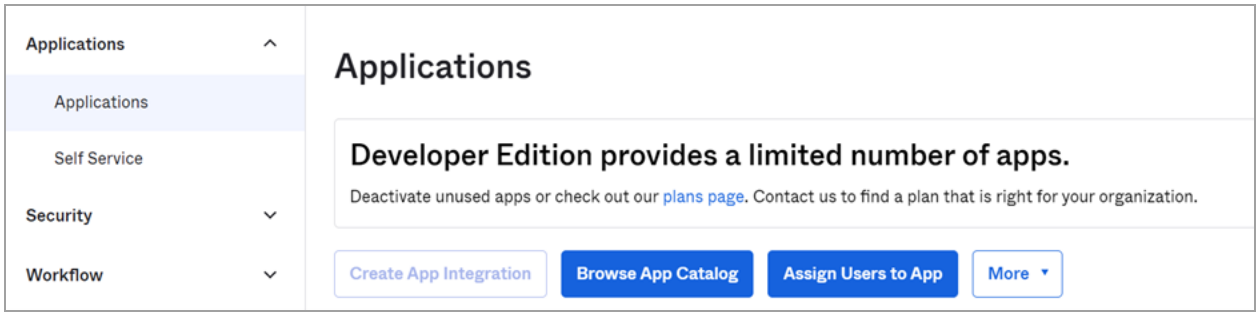
c. In the Group Attributes Statements section, add following attribute statements:

Name	Name format	Filter	Value
DPAGroups	Unspecified	Matches regex	.*



d. Click Next, provide the requested background information about yourself, and click Finish.

7. Specify the users to access DPA through SAML login:
   a. In the left pane of the Admin Console, click Applications > Applications.
   b. Click Assign Users to App.



   c. Select the applications and people, click Next, and click Confirm Assignments.

8. On the Sign On tab, click the View Setup Instructions button in the Sign On Methods section. Keep the tab open so that you can copy and paste the information into DPA.

Complete the identity provider configuration in DPA

In DPA, the Add Identify Provider wizard is still open on the Prepare IdP page.

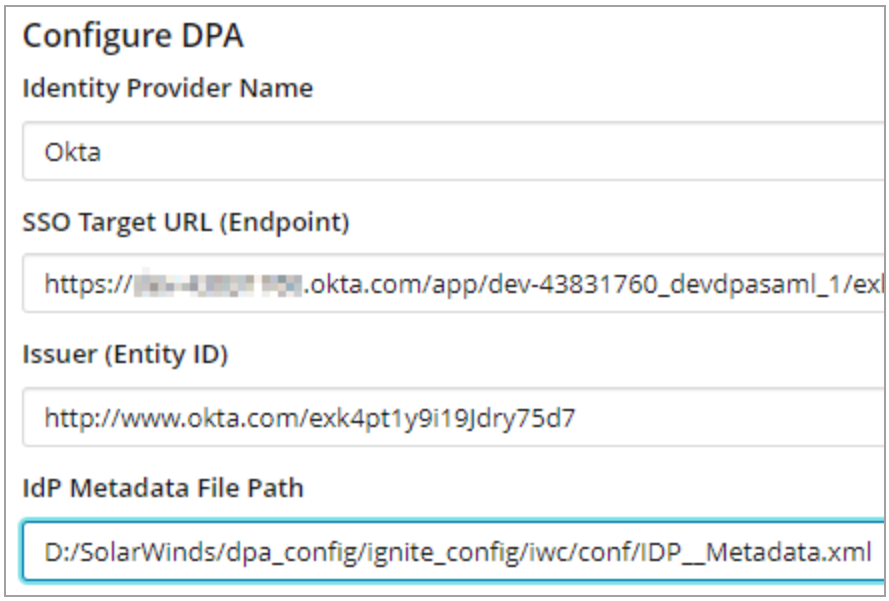
1. Click Next to open the Configure DPA page.
2. Enter `Okta` in the Identity Provider Name field.

3. Paste the following values from Okta into DPA:

Okta field	DPA field
Identity Provider Single Sign-On URL	SSO Target URL (Endpoint)
Identity Provider Issuer	Issuer (Entity ID)

4. In the IdP Metadata File Path in DPA, enter one of the following from Okta:
   - From the Optional section, enter the path of the XML file where content is saved.

   - Download the `metadata.xml` file from the Identity Provider metadata link and enter that absolute file path.



5. Click Next.
6. On the Summary page, review the information and click Configure.
7. At the confirmation message, click Finish and go to Options page.
8. Restart DPA for the settings to take effect.

   When the configuration is complete, the DPA `system.properties` file includes the following entries:

- com.confio.security.saml.sso.targetUrl
- com.confio.security.saml.entityId
- com.confio.saml.sso.idp.metaData
- com.confio.security.saml.enabled
- com.confio.security.saml.defaultIDP
- com.confio.security.saml.identityProviderName
- com.confio.security.saml.dpaUrl

Create groups of users and configure user permissions

After you have set up DPA to use SAML authentication, do the following:

1. In Okta, assign application to users.
2. Assign an app integration to a group.
3. Determine which groups contain the users that you want to grant access to DPA. You may need to create a group if a suitable group does not exist.
4. From the DPA menu in the upper-right corner, click Options.
5. Under Administration > Users & Contacts, click User Administration.
6. Click Add SAML Group.
7. Add the same group name that was added in Okta.
8. Assign privileges to the group, just as you would for a user.
9. Click Save.

ⓘ You can add multiple groups in DPA. If a user is a member of more than one group, DPA grants them the combined privileges from all their groups.

Log in to DPA

When the SAML configuration is complete, the DPA login dialog includes an additional button: Login with SAML SSO.

Instead of entering credentials at the DPA login dialog, click Login with SAML SSO. The first time you log in, the Okta website opens and you can enter your Okta credentials.

When you are already logged in to Okta, DPA opens when you click Login with SAML SSO. You are not prompted for credentials unless you are logged out of Okta during a browser session, or you close the browser.

If the Okta admin user is also the DPA user, you are not prompted for credentials when you click Login with SAML SSO.

(Optional) Enable assertion encryption

SAML assertion encryption is optional. It's an extra level of security in addition to the security provided by HTTPS. By default, assertion encryption is not enabled.

1. Log in to your Okta organization with administrative privileges.
2. In the left pane of the Admin Console, click Applications.
3. Click the General tab.
4. In SAML Settings, click Edit. Then click Next.
5. On the Configure SAML page, click Show Advanced Settings.
6. From the Assertion Encryption drop-down, select Encrypted.
7. Upload the encryption certificate. If you use DPA's default `saml.keystore` file, the encryption certificate is available in the following location:

   DPA-install-dir`\iwc\tomcat\webapps\iwc\WEB-INF\classes\DefaultPublicCertForSaml.crt`
8. Click Next and then Finish to exit the wizard.

Configure DPA to use SAML authentication with Microsoft Entra ID

SAML authentication in DPA offers single sign-on (SSO) and the opportunity to use different credential storage or multifactor authentications using third-party identity providers like Microsoft Entra ID (previously Azure AD), Okta, or Keycloak. Complete the following tasks to configure SAML authentication and single sign-on with Microsoft Entra ID as the identity provider.

If DPA is running behind a load balancer (or API Gateway) and you want to enable SAML SSO authentication in DPA, you **must** enable SSL communication between the load balancer (or API Gateway) and DPA.

(Optional) Configure SAML keystore properties

By default, the keystore file from the classpath resource (`saml.keystore`) is used for SAML authentication. If you use the default keystore file, you do **not** need to modify the SAML keystore properties.

If you would like to use a different keystore file, specify values for the following properties in the `system.properties` file.

1. Open the following file in a text editor:

   DPA-install-dir\iwc\tomcat\ignite_config\idc\system.properties

2. Add or uncomment the following properties and specify the values.

Property	Value
com.confio.security.saml.keystore.path	The keystore file path.
com.confio.security.saml.keystore.password	The password of the keystore file.
com.confio.security.saml.keystore.privatekey.alias	The alias of the private key stored in the keystore file.
com.confio.security.saml.keystore.privatekey.password	The password of the private key certificate added in the keystore file. The value for this property can be the same as the value for the property com.confio.security.saml.keystore.password. If the private key certificate does not have a password, uncomment or add the property but do not enter a value.
com.confio.security.saml.keystore.type	The file type of the keystore file. This property is optional. If the custom keystore file is not JKS or PKCS12, use this property to specify the type.

3. Restart DPA for the new properties to take effect.

Prepare the identity provider (IdP): Microsoft Entra ID

When configuring Microsoft Entra ID to communicate with DPA, you will be working with both Microsoft Entra ID and DPA at the same time. You must keep both systems open to copy information from one system into the other.

Before you start

- DPA must be configured to use SSL to protect data during transmission. To enable SSL for DPA, see Configure DPA to use a custom certificate for SSL/TLS.

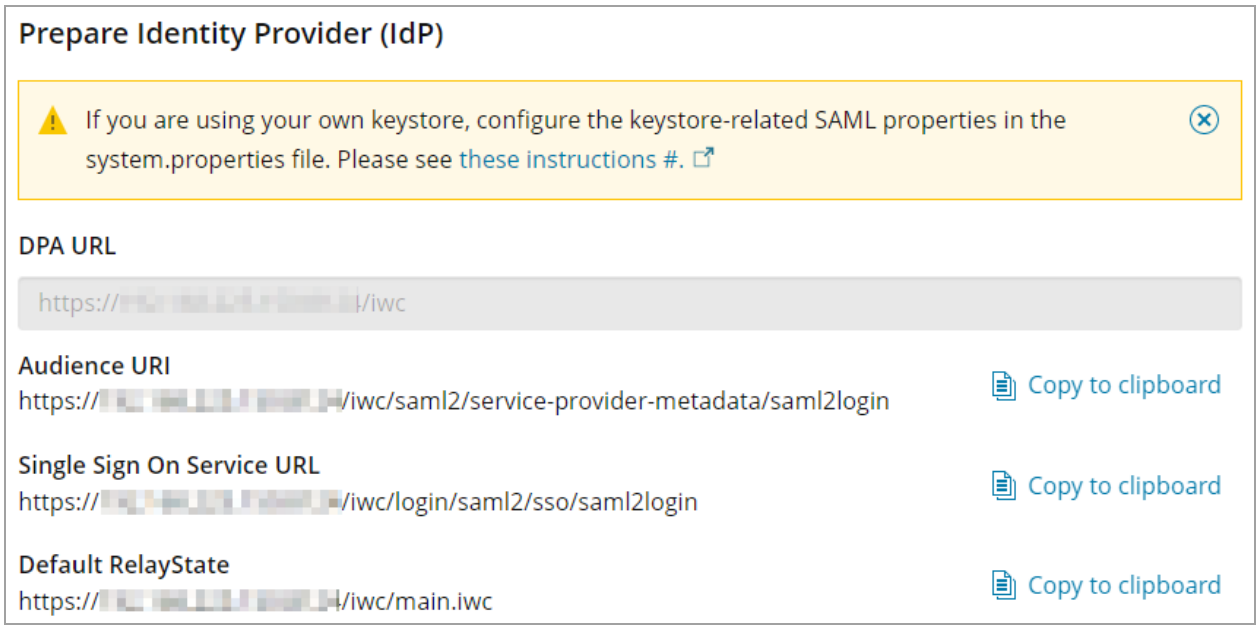
- DPA must be running on an HTTPS connection.
- If you do not want to use the default keystore file (`saml.keystore`), configure the SAML keystore properties in the `system.properties` file.

Task 1: In DPA, obtain the DPA endpoint URLs for the identity provider

1. Log in to DPA as a user with administrative privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Configure SAML.

   On the Prepare Identify Provider (IdP) page, the following information is automatically added:

   - DPA URL
   - Audience URI
   - Single Sign On Service URL
   - Default RelayState

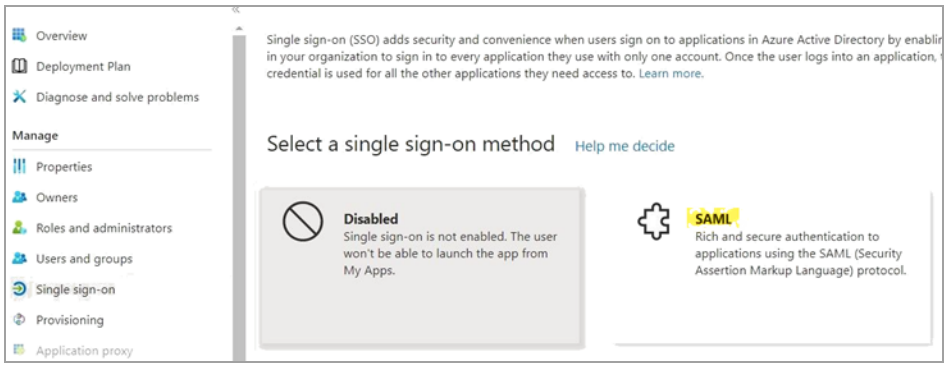


   You will copy and paste this information into the configuration in Azure AD.

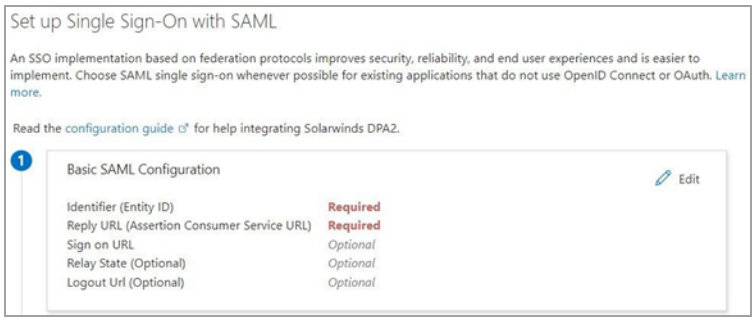
4. Keep DPA open, and continue in Azure.

Task 2: In Microsoft Entra ID, create the SAML application, configure URLs and the URI, and specify users for SAML login

1. Log in to your Microsoft Entra ID portal with administrative privileges to add the application for SAML authentication.
2. Create the enterprise application and then select SAML as the single sign-on method.

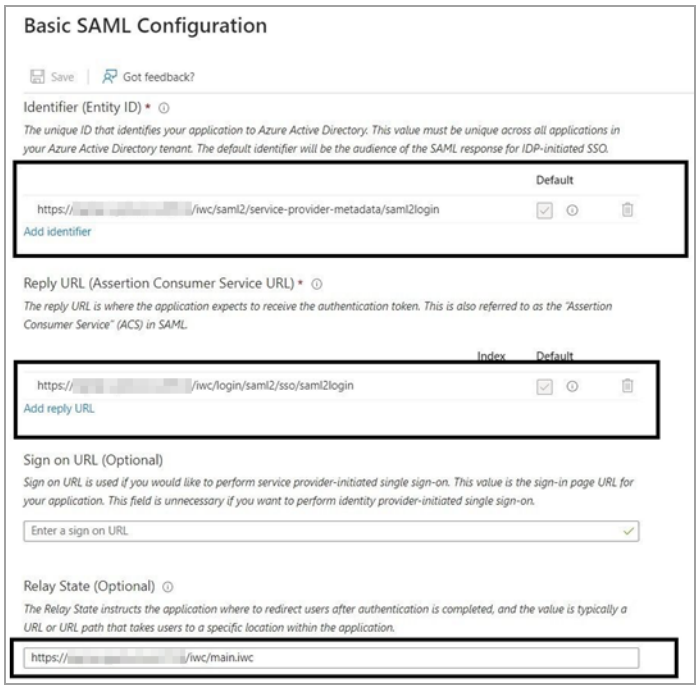


The Set up Single Sign-On with SAML screen opens.

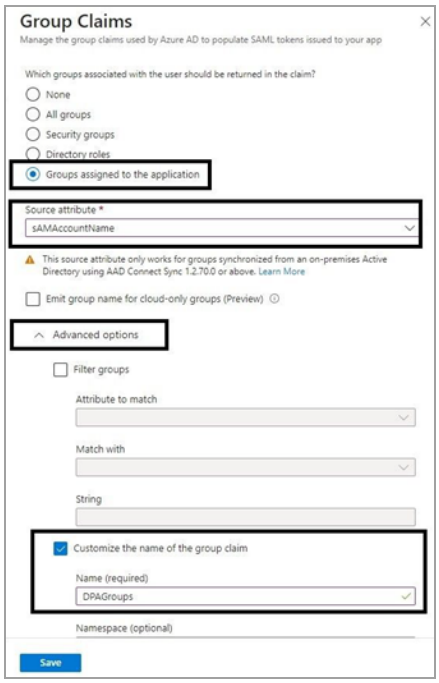


3. In Basic SAML Configuration, click the Edit icon.
4. Paste the following values from DPA into Microsoft Entra ID:

DPA field	Microsoft Entra ID field	Notes
Audience URI	Identifier (Entity ID)	This is the intended audience of the SAML assertion. This is most often the SP Entity ID of your application.
Single Sign On Service URL	Reply URL (Assertion Consumer Service URL)	The reply URL is where the application expects to receive the authentication token. This is also referred to as the “Assertion Consumer Service” (ACS) in SAML.
Default RelayState	Default RelayState	This is default landing page in the IDP initiated flow.

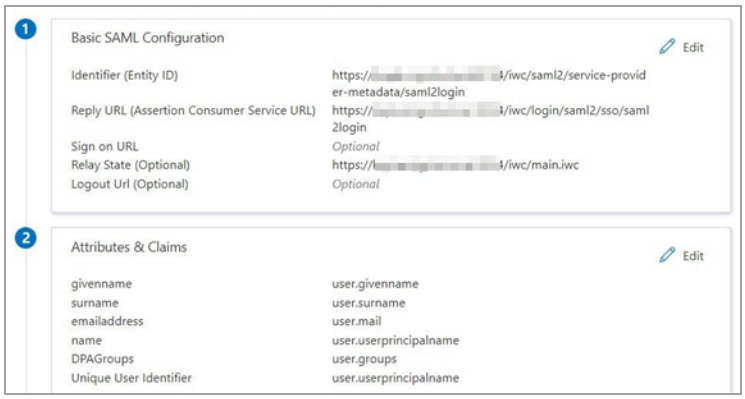


5. In Attributes & Claims, click the Edit icon. Then click Add a group claim.
6. In the Group Claims panel, specify the following:
    a. Under the Which groups associated with the user option, select Groups assigned to the application.
    b. Under Source attribute, select sAMAccountName.
    c. Expand Advanced options. Then select Customize the name of the group claim, and enter DPAGroups in the Name box.



d. Click Save.

The Set up Single Sign-On with SAML displays the information you entered.



Complete the identity provider configuration in DPA

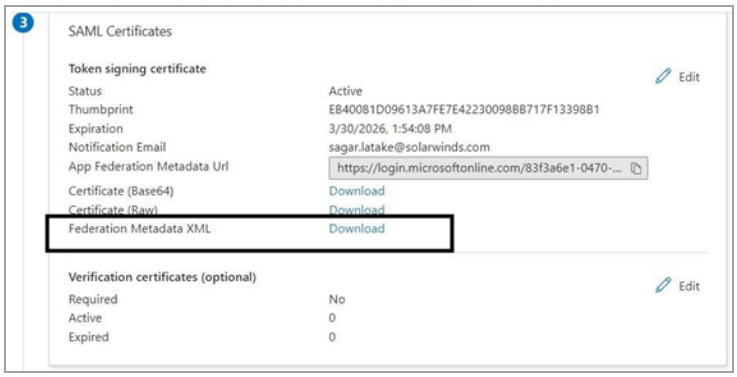
In DPA, the Add Identify Provider wizard is still open on the Prepare IdP page.

1. Click Next to open the Configure DPA page.
2. Enter an identifying name such as `Microsoft Entra ID` in the Identity Provider Name field.

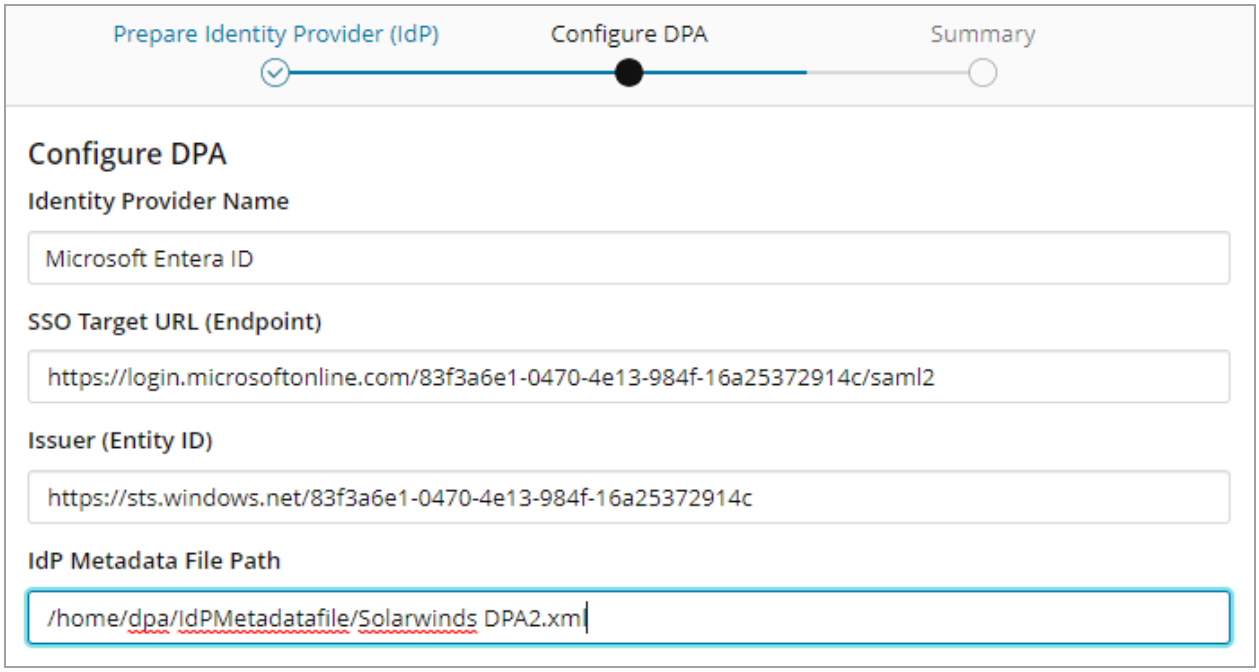
3. Paste the following values from Microsoft Entra ID into DPA:

Microsoft Entra ID field	DPA field
Login URL	SSO Target URL (Endpoint)
Microsoft Entra ID Identifier	Issuer (Entity ID)

4. Specify the IdP Metadata File Path:
   a. In Microsoft Entra ID, click the Download option next to Federation Metadata XML.



   b. In DPA, enter the absolute file path of downloaded Federation Metadata XML file.



5. Click Next.
6. On the Summary page, review the information and click Configure.
7. At the confirmation message, click Finish and go to the Options page.

8. Restart DPA for the settings to take effect.

   When the configuration is complete, the DPA `system.properties` file includes the following entries:

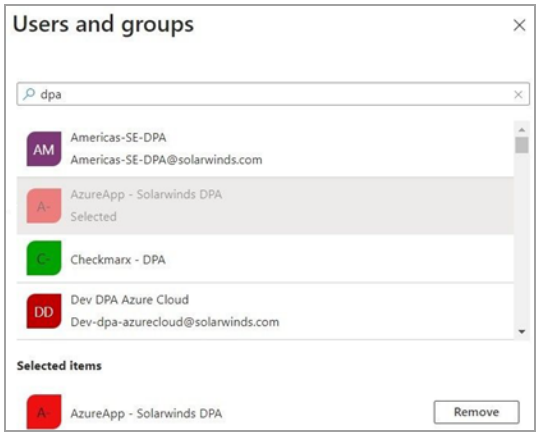
   - com.confio.security.saml.sso.targetUrl
   - com.confio.security.saml.entityId
   - com.confio.saml.sso.idp.metaData
   - com.confio.security.saml.enabled
   - com.confio.security.saml.defaultIDP
   - com.confio.security.saml.identityProviderName
   - com.confio.security.saml.dpaUrl

Create groups of users and configure user permissions

After you have set up DPA to use SAML authentication, do the following:

1. In the left pane of the SAML configuration page in the Azure AD portal, click Users and groups.
2. Click Add user/group.
3. Under Users and groups, click None Selected.
4. In the Users and groups panel, search for a group and select it.

   > If the group does not exist, then create the group in Azure AD and add the users to that group.

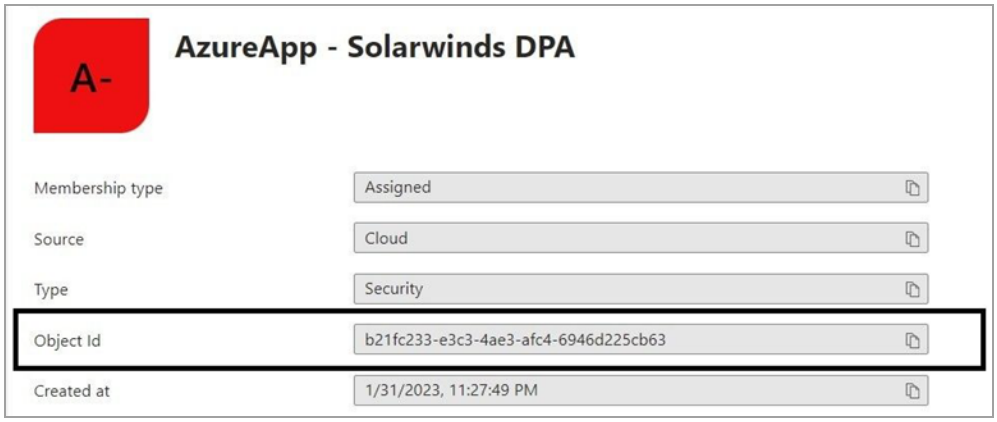


5. Save the configuration and click Assign.

   The assigned groups are shown on the Users and groups screen.

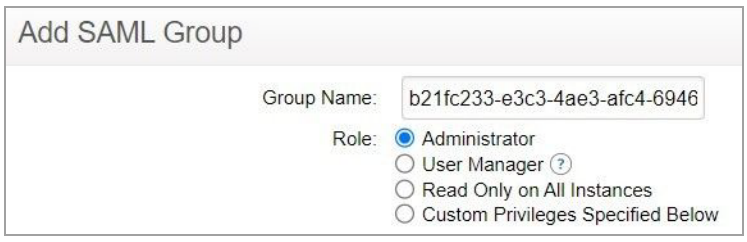
6. Copy either the group name or ID (you will need this to create the SAML group in DPA):

- If the group was created in an on-premises product and copied to Azure AD, copy the group name.
- If the group was created directly in Azure AD, click the group name and copy the object ID of the group from the selected group page.



> If you do not know where the group was created, copy the object ID to enter into DPA. If that is not correct, users will receive the following message when they attempt to log in with SAML authentication: `Domain user has no permissions in DPA`. If this occurs, access DPA logs, open the `auth.log` file, and look for the entry `Groups received in SAML response`. If this entry contains the group name instead of the object ID, you must recreate the SAML group with the group name.

7. In DPA, create the SAML group:
   a. From the DPA menu in the upper-right corner, click Options.
   b. Under Administration > Users & Contacts, click User Administration.
   c. Click Add SAML Group.
   d. As the Group Name, enter the group name or object ID copied in the previous step.
   e. Assign privileges to the group, just as you would for a user.



f. Click Save.

> You can add multiple groups in DPA. If a user is a member of more than one group, DPA grants them the combined privileges from all their groups.

Log in to DPA

When the SAML configuration is complete, the DPA login dialog includes an additional button: Login with SAML SSO.

Instead of entering credentials at the DPA login dialog, click Login with SAML SSO. The first time you log in, you will need to enter the credentials on the Microsoft Entra ID login page.

When you are already logged in to the Microsoft Entra ID portal or any other SAML application, you are automatically logged in to DPA when you click Login with SAML SSO. You are not prompted for credentials unless you are logged out of Microsoft Entra ID during a browser session, or you close the browser.

(Optional) Enable assertion encryption

SAML assertion encryption is optional. It's an extra level of security in addition to the security provided by HTTPS. By default, assertion encryption is not enabled.

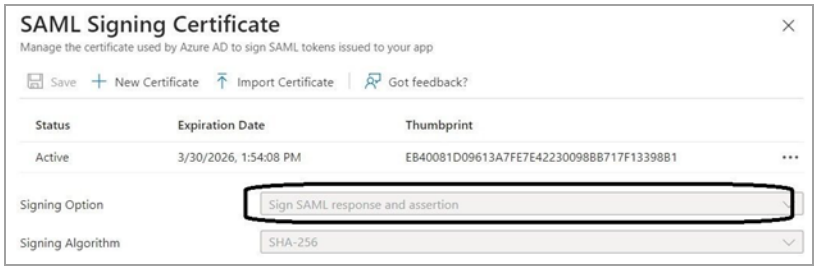
1. Log in to the Microsoft Entra ID portal with administrative privileges.
2. Select the DPA application that you want to configure assertion encryption for.
3. In the left pane under Security, click Token encryption.
4. Click Import Certificate, and upload the encryption certificate. If you use DPA's default `saml.keystore` file, the encryption certificate is available in the following location:

   DPA-install-dir`\iwc\tomcat\webapps\iwc\WEB-INF\classes\DefaultPublicCertForSaml.crt`

   When the certificate is uploaded, it is shown in the list.
5. Click the three dots on the left side of the list item, and then click Activate token encryption certificate.
6. Click Yes to confirm the activation.

   When the assertion encryption is enabled, the Status column displays Active.
7. In the SAML Certificates section, click the Edit icon. Then verify that the Signing Option box

contains the value Sign SAML response and assertion.



Configure DPA to use SAML authentication with Keycloak

SAML authentication in DPA offers single sign-on (SSO) and the opportunity to use different credential storage or multifactor authentications using third-party providers like Okta, Microsoft Entra ID (previously Azure AD), or Keycloak. Complete the following tasks to configure SAML authentication and single sign-on with Keycloak as the identity provider.

If DPA is running behind a load balancer (or API Gateway) and you want to enable SAML SSO authentication in DPA, you **must** enable SSL communication between the load balancer (or API Gateway) and DPA.

Task 1: Enable SSL/HTTPS for the Keycloak server

By default, Keycloak is not set up to handle SSL/HTTPS traffic. Complete the following steps to enable HTTPS.

1. Generate a Java keystore that contains the private key and certificate for SSL/HTTPS traffic:
    a. Open a command prompt and run the following command:

    ```
    keytool -genkey -alias localhost -keyalg RSA -keystore keycloak.jks -validity 10950
    ```

    b. Respond to the prompts, and enter `yes` if the summarized information is correct.

```
$ keytool -genkey -alias localhost -keyalg RSA -keystore keycloak.jks -validity 10950
    Enter keystore password: secret
    Re-enter new password: secret
    What is your first and last name?
    [Unknown]:  localhost
    What is the name of your organizational unit?
    [Unknown]:  Keycloak
    What is the name of your organization?
    [Unknown]:  Red Hat
    What is the name of your City or Locality?
    [Unknown]:  Westford
    What is the name of your State or Province?
    [Unknown]:  MA
    What is the two-letter country code for this unit?
    [Unknown]:  US
    Is CN=localhost, OU=Keycloak, O=Test, L=Westford, ST=MA, C=US correct?
    [no]:  yes
```

The certificate is created in the directory from which you ran the command.

c. Optionally, copy the file to a different location on the Keycloak server.

2. Configure the Keycloak server to use the keypair and certificate generated in the previous step.

a. Locate the `keycloak.conf` file in the `keycloak-`*`version`*`\conf` folder, and open the file in a text editor.

b. Add the following lines to the the `keycloak.conf` file, where:

- *`Path`* is the location of the `Keycloak.jks` file.
- *`Password`* is the password that you added when you created the certificate.
- *`Port`* is the HTTPS port number. This line is required only if port 8443 (the default HTTPS port for Keycloak) is in use.

```
https-key-store-file=Path
https-key-store-password=Password
https-key-store-type=jks
https-port=Port
```

c. Save the file.

Task 2: Configure a new realm in Keycloak

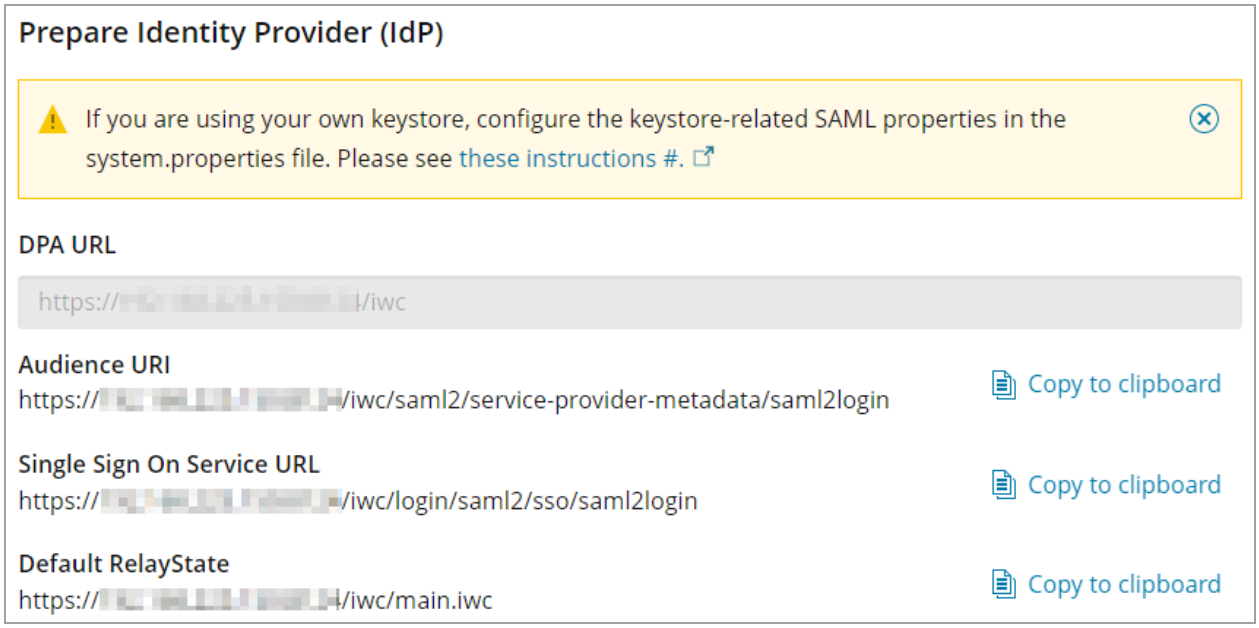
1. Log in to the Keycloak admin console, and click Create Realm.
2. Name the realm and keep the default settings.
3. Click the link SAML 2.0 Identity Provider Metadata option on the Realm settings page.
4. Save the metadata to a file with a recognizable file name, such as `IDP_Metadata.xml`.
5. Open the file in a text editor and verify that the value of `WantAuthnRequestsSigned` is `false`.

Task 3: Configure DPA to use SAML

1. Log in to DPA as a user with administrative privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Configure SAML.

   On the Prepare Identify Provider (IdP) page, the following information is automatically added:

   - DPA URL
   - Audience URI
   - Single Sign On Service URL
   - Default RelayState



4. Click Next to open the Configure DPA page.
5. Enter Keycloak in the Identity Provider Name field.

6. Enter the following values from the metadata file saved in Task 2:

DPA field	Value
SSO Target URL (Endpoint)	In the metadata file saved in Task 2:, locate the `SingleSignOnService` tag and enter the `location` value.
Issuer (Entity ID)	In the metadata file, locate the `EntityDescriptor` tag and enter the `entityID` value.
IdP Metadata File Path	Enter the path and file name of the metadata file.

7. Click Next.
8. On the Summary page, review the information and click Configure.
9. At the confirmation message, click Finish and go to the Options page.
10. Restart DPA for the settings to take effect.

    When the configuration is complete, the DPA `system.properties` file includes the following entries:

    - com.confio.security.saml.sso.targetUrl
    - com.confio.security.saml.entityId
    - com.confio.saml.sso.idp.metaData
    - com.confio.security.saml.enabled
    - com.confio.security.saml.defaultIDP
    - com.confio.security.saml.identityProviderName
    - com.confio.security.saml.dpaUrl

Task 4: Configure the client

1. In DPA, download the service provider (DPA) metadata file:
    a. Under Administration > Users & Contacts, click Configure SAML to open the Prepare Identify Provider (IdP) page.
    b. On the Audience URI line, click Copy to clipboard.
    c. Paste the URI into the address bar of a browser, and press Enter.

    The `spring_saml_metadata.xml` is downloaded to the `Downloads` folder or your computer.
2. In the Keycloak left pane, click Clients. Then click Import client.

3. Click Browse, browse to the location of the `spring_saml_metadata.xml` file, and select it to upload.

   The contents of the file are shown in the Resource file box.

4. Click Save.

   Information is automatically added to the Client details tabs.

5. Add a mapper:

   a. On the Client scopes tab, click the existing automatically populated Assigned client scope name that starts with DPA URL (for example, `https://DPA_URL/iwc/saml2/service-provider-metadata/saml2login-dedicated`).

      The Dedicated scopes page opens.

   b. On the Mappers tab, click Configure a new mapper.

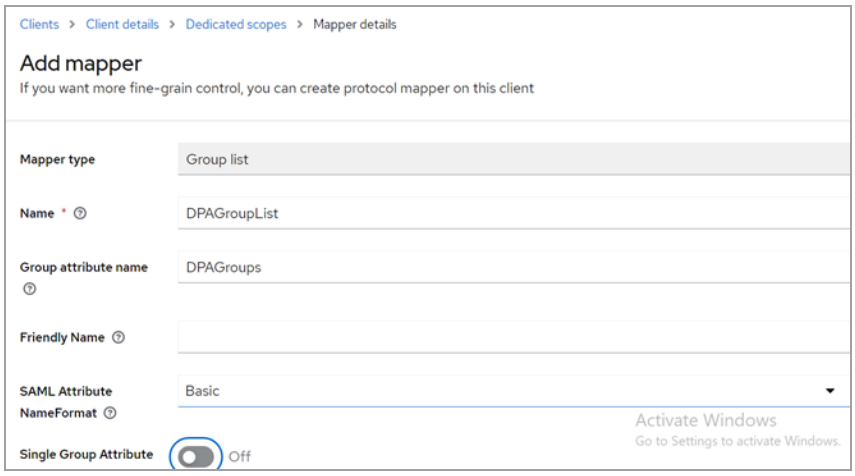
      The Mapper details page opens.

   c. As the Mapper type, select Group list, and enter a name.

   d. As the Group attribute name, you **must** enter `DPAGroups`.

6. As the SAML Attribute Name Format, select Basic.

7. Toggle the Single Group Attribute and Full group path to Off.



8. Click Save.

Task 5: Configure users and groups

1. In Keycloak, add one or more groups:
    a. In the Keycloak left pane, click Groups.
    b. Click Create group.
    c. Enter a name and click Create.
2. In Keycloak, create one or more users, and add each user to a group:
    a. In the left pane, click Users.
    b. Click Add user.
    c. Enter the user name and email address, and click Create.
    d. On the Credentials tab, click Set password.
    e. Enter the password, click Save, and click Save password at the confirmation message.
    f. On the Groups tab, click Join Group.
    g. Select the group you created in step 2, and click Join.
3. In DPA, create a SAML group for each group you created in Keycloak:
    a. From the DPA menu in the upper-right corner, click Options.
    b. Under Administration > Users & Contacts, click User Administration.
    c. Click Add SAML Group.
    d. As the Group Name, enter the same name you entered for the group in Keycloak.
    e. Assign privileges to the group, just as you would for a user.

f. Click Save.

> You can add multiple groups in DPA. If a user is a member of more than one group, DPA grants them the combined privileges from all their groups.

Log in to DPA

When the SAML configuration is complete, the DPA login dialog includes an additional button: Login with SAML SSO.

Instead of entering credentials at the DPA login dialog, click Login with SAML SSO. The first time you log in, The Keycloak login page opens and you can enter your Keycloak credentials.

When you are already logged in to Keycloak, DPA opens when you click Login with SAML SSO. You are not prompted for credentials unless you are logged out of Keycloak during a browser session, or you close the browser.

Define contacts for DPA alert notifications and reports

When you create an alert or schedule a report, the list of contacts determines who receives the alert notifications or the scheduled report.

Contacts store the information that DPA needs to send emails, post to a Teams or Slack channel, or send SNMP traps. Contact groups allow you to quickly select multiple recipients. As your organization changes, you can edit or delete contacts or contact groups.

ⓘ Only DPA administrators can create and manage contacts.

Create email contacts

Email contacts are people who can receive email notifications when an alert is triggered, or who can receive scheduled reports through email. When you define an alert or schedule a report, you can select the recipient from the list of available contacts.

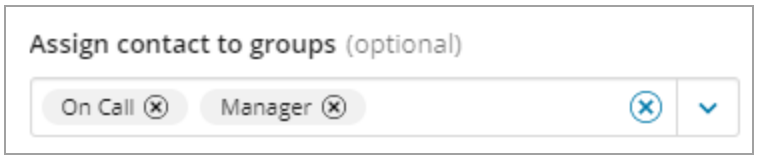
ⓘ You can also send alert notifications to Slack or Teams channels and send SNMP traps from DPA alerts.

1. Log in to DPA using an account with administrator privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Contact Management.
4. On the Contact management page, click Create contact.

   ⓘ If any contacts are selected, the Create contact button is not displayed.
5. Under Contact type, select Email.
6. Enter the contact's name and email address.

7. To test the email address, click Send test email. Then verify that the person received the email.
8. (Optional) Add the contact to one or more groups.
    a. Click the down-arrow under Assign contact to groups.
    b. Click a group name to select the group.

       The group is shown in the Assign contacts to groups box.
    c. Repeat these steps to add the contact to more groups.



9. Click Create.

   The contact is added to the list of contacts.

Create webhook contacts for Slack or Teams notifications

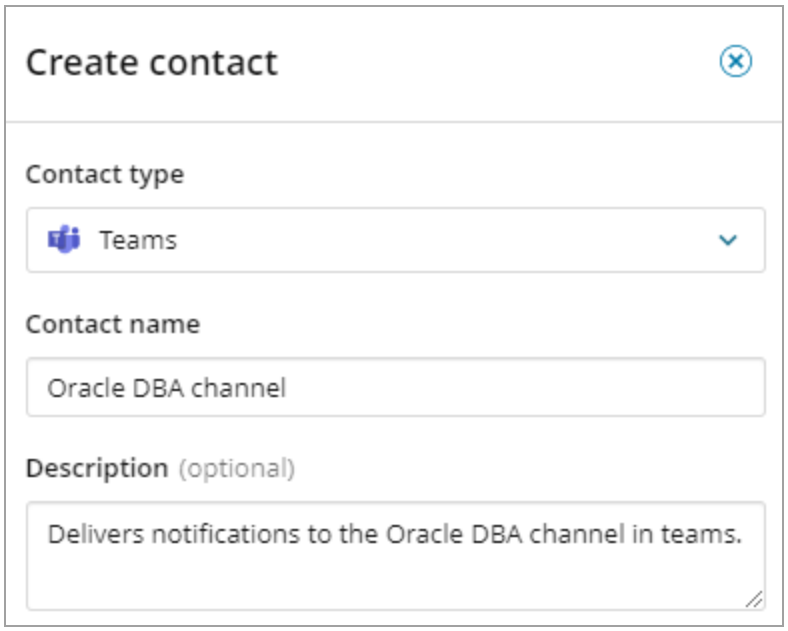
Webhook contacts can be used to post alert notifications in a Slack or Teams channel. When you create a webhook contact, you can select it as a recipient when you define an alert.

You can also send alert notifications through email and send SNMP traps from DPA alerts.

1. Create a webhook in the Slack or Teams channel that you want to send alert notifications to. Then copy the webhook URL.

   For more information about creating webhooks, see the Microsoft or Slack documentation.
2. Log in to DPA using an account with administrator privileges.
3. From the DPA menu in the upper-right corner, click Options.
4. Under Administration > Users & Contacts, click Contact Management.
5. On the Contact management page, click Create contact.

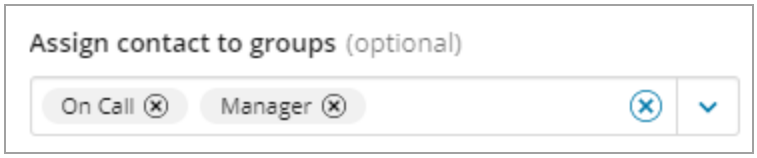
   ⓘ If any contacts are selected, the Create contact button is not displayed.
6. Under Contact type, select Slack or Teams.
7. Enter a name to identify this contact and, optionally, a description.



8. In the Address box, enter the webhook URL that you copied in step 1.
9. To test the webhook URL, click Send test webhook. Then verify that a test notification from DPA was posted in the Slack or Teams channel.
10. (Optional) Add the contact to one or more groups.
    a. Click the down-arrow under Assign contact to groups.
    b. Click a group name to select the group.

       The group is shown in the Assign contacts to groups box.

c. Repeat these steps to add the contact to more groups.



11. Click Create.

    The contact is added to the list of contacts.

Restrict where DPA can send Slack or Teams notifications

By default, DPA can sent alert notifications to any Slack or Teams webhook contact that you create. You can configure DPA to send notifications only to the webhook URLs that are allowed in the `system.properties` file.

1. Open the following file in a text editor:

   DPA-install-dir`\iwc\tomcat\ignite_config\idc\system.properties`

2. Uncomment or add one or more lines to identify all allowed URLs:

   `com.solarwinds.dpa.notifications.webhooks.service.whitelist.urlPattern=(`*`allowedUrlPattern`*`)`

   where *`allowedUrlPattern`* is the allowed URL or a regular expression that matches multiple allowed URLs.

3. Save the file, and then restart DPA.

Throttling properties

DPA includes the following properties that affect the number of webhook notifications it can send in a given time period.

Property	Description	Default value
`delayMS`	The minimum number of milliseconds between notifications.	1000
`buffer.size`	The maximum number of notifications that DPA can store in memory while waiting to send them. If the buffer is full, additional webhook notifications are discarded.	3600
`timeoutMS`	The number of milliseconds DPA waits for a response from Teams or Slack before resending the notification.	20000

To prevent denial of service (DOS) attacks, Slack and Teams also limit the number of messages that each channel can receive.

If DPA generates more notices than it can send, create multiple Slack or Teams channels to receive additional notifications. Then create multiple "profiles" in the `system.properties` file. Each profile identifies a URL pattern and specifies the throttling property values for notifications sent to URLs matching that pattern.

If the URL in a webhook contact does not match any of the URL patterns defined in the `system.properties` file, the default limits apply.

1. Open the following file in a text editor:

   DPA-install-dir\iwc\tomcat\ignite_config\idc\system.properties

2. For each profile, add the following line to identify the profile and specify which webhook URLs it affects:

   com.solarwinds.dpa.notifications.webhooks.service.*n*.urlPattern=*UrlPattern*

   where:

   - *n* is a number that identifies each profile. For example, the first profile would contain `service.1` and the next would contain `service.2`.
   - *UrlPattern* is the URL that the properties in this profile apply to, or a regular expression that matches multiple URLs.

     Special characters in the regular expression must have **two** escape characters in front of them (//).

     The regular expression in the following example matches all Teams URLs:

     ```
     com.solarwinds.dpa.notifications.webhooks.service.1.urlPattern=http
     s:\\/\\/[a-zA-Z0-9_\\-\\.]+\.office\\.com\\/webhookb2\\/[a-zA-Z0-9_\\-
     @\\/]+
     ```

     The regular expression in the following example matches all Slack URLs:

     ```
     com.solarwinds.dpa.notifications.webhooks.service.2.urlPattern=https:\
     \/\\/hooks.slack.com\\/services\\/[A-Za-z0-9_\\-\\/]+
     ```

3. Add the following lines to each profile:

   com.solarwinds.dpa.notifications.webhooks.service.1.delayMs=*Value*

   com.solarwinds.dpa.notifications.webhooks.service.1.buffer.size=*Value*

`com.solarwinds.dpa.notifications.webhooks.service.1.timeoutMs=`*`Value`*

4. Save the file, and then restart DPA.

Send SNMP traps from DPA alerts

You can configure DPA alerts to send SNMPv2c traps to an SNMP-enabled Network Management System (NMS) when an alert level is reached and when the alert level returns to Normal. The trap contains the name of the monitored database instance, alert name, alert level, and response instructions.

To configure an alert to send an SNMP trap, complete the following tasks:

1. Import the DPA MIB file into your NMS.
2. Create one or more SNMP contacts. The SNMP contact defines the response instructions included in the trap, and so you must create different contacts for alerts with different response instructions.
3. In the alert definition, add the SNMP contact as recipient for the alert level that you want to send a trap.

The DPA MIB file

DPA contains a Management Information Base (MIB) file that defines the trap and the associated data sent with each trap. The MIB file defines the following:

- Private Enterprise Number
- One Trap Definition (NOTIFICATION-TYPE)
- Four string objects bound to each trap: database name, alert name, alert level, and response instructions

Before configuring DPA to send SNMP traps, provide the MIB file to the person responsible for importing MIB files into the NMS. The MIB file is in the following location:

`*DPA-install-dir*/iwc/CONFIO-MIB.mib`

Create an SNMP contact

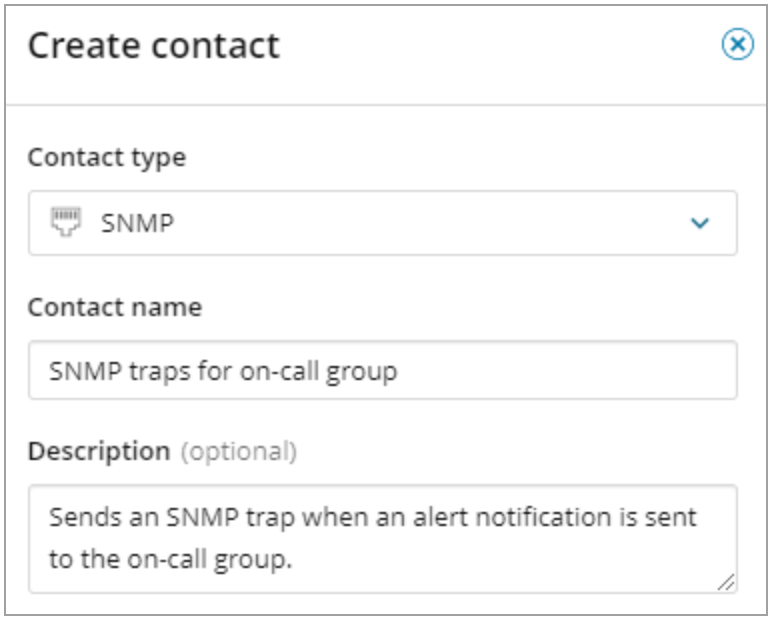
The NMS that receives the trap is represented as an SNMP contact in DPA.

1. Log in to DPA using an account with administrator privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Contact Management.

4. On the Contact management page, click Create contact.

   If any contacts are selected, the Create contact button is not displayed.

5. Under Contact type, select SNMP.
6. Enter a name and description to identify the associated alerts.



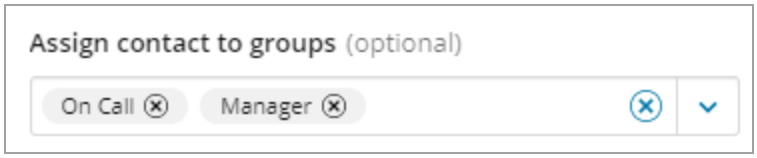
7. Identify the NMS to send the trap to:
   a. In the Trap receiver host field, enter the host name or IP address of the server where the NMS is running.
   b. In the Trap receiver port field, enter the port number where the NMS host is receiving traps. The default is 162.
   c. In the Community string field, enter the community string used by the NMS for traps.
8. Enter the response instructions to be included in the trap.
9. To test the configuration, click Send test SNMP trap. Then verify that the NMS received the trap.
10. (Optional) Add the contact to one or more groups.

    For example, when an alert reaches a certain level, you might want to send an email to the on-call personnel and send a trap to the NMS. You can add the SNMP contact to the On Call group.

    a. Click the down-arrow under Assign contact to groups.
    b. Click a group name to select the group.

       The group is shown in the Assign contacts to groups box.

c. Repeat these steps to add the contact to more groups.



11. Click Create.

    The contact is added to the list of contacts.

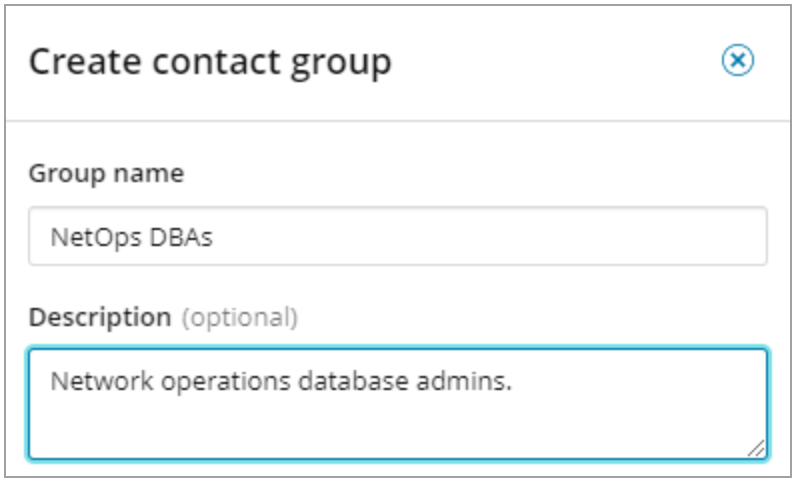
Create contact groups

Use contact groups to send alert notifications or scheduled reports to multiple contacts. DPA provides several default contact groups, but you can create other groups.

1. Log in to DPA using an account with administrator privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Contact Management.
4. On the Contact management page, click the Contact Groups tab.
5. Click Create group.

   ⓘ If any contact groups are selected, the Create group button is not displayed.

6. Enter a name to identify the group and, optionally, a description.



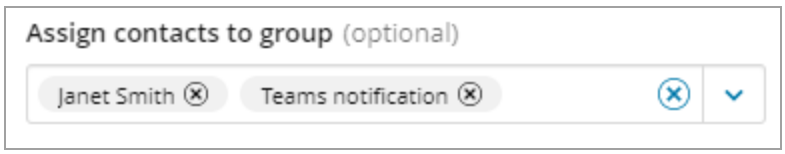
7. Add existing contacts to the group:

a. Click the down-arrow under Assign contacts to group.

b. Click a contact name to select the contact.

   The contact is shown in the Assign contacts to group box.

c. Repeat these steps to add more contacts to the group.



8. Click Create.

   The contact group added to the list of groups.

Update contacts and contact groups

You can edit the definition of a existing contact or contact group (for example, to update an email address or add group members). You can also deactivate contacts or activate them again.

Edit a contact or contact group definition

1. Log in to DPA using an account with administrator privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Contact Management.
4. To edit a contact group definition, click the Contact Groups tab.
5. To open the item, either:
   - Click the vertical ellipsis (⋮) on the right, and click Edit.
   - Select the contact or contact group, and click the Edit contact or Edit group button.
6. Make the necessary changes. For more information about the available fields, see:
   - Create email contacts
   - Create contacts for Slack or Teams notifications
   - Send SNMP traps from DPA alerts
   - Create contact groups

   ⓘ You cannot change the contact type of an existing contact.
7. Click Save.

Deactivate or activate a contact

On the Contact management page, the green toggle button on the right indicates that a contact is active. Contacts are active by default.

When an email contact or a webhook contact is inactive, DPA does not sent alert notifications or (for email contacts) scheduled reports to that contact. When an SMTP contact is inactive, alerts associated with this contact do not send traps to the NMS.

To deactivate or reactivate contacts:

- Click the toggle button to the right of a contact to make it active or inactive .
- Select one or more contacts, and click Activate or Deactivate.

Delete contacts and contact groups

1. Log in to DPA using an account with administrator privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Users & Contacts, click Contact Management.
4. To delete a contact group, click the Contact Groups tab.
5. Do either of the following:
   - Click the vertical ellipsis (⋮) to the right of a contact or contact group, and click Delete.
   - Select one or more contacts or contact groups, and click the Delete button.
6. On the confirmation dialog, click Yes.

DPA reports

To work with DPA reports, see the following topics:

- Learn about the available report types and the differences between report data and chart data.
- Access and run existing DPA reports.
- Create a new DPA report.
- Search for a SQL statement to include in a report.
- Schedule a DPA report for email delivery.
- Create a DPA report group.

About DPA reports

Use reports to communicate the long-term performance of your databases and supply evidence to support your work. Reports can capture the results of performance tuning and highlight database trends. You can send reports to managers, team members, and customers.

Differences between report data and detailed chart data displayed in the DPA interface

The data shown in reports differs from the detailed chart data shown in the DPA interface in the following ways:

- Storage period and granularity:
    - **Reports** can show data captured over longer intervals and display long-term trends. To generate reports, DPA summarizes repository data to make long-term information available in a manageable size.

      The previous 90 days of data are summarized by hour. After 90 days, data are summarized by day. This information is available for five years.
    - Detailed **chart** data is available for a shorter period, typically 30 days.

      Charts can show information down to the second.

- Data collection period:
  - **Reports** can be generated after a one-hour data collection period. SolarWinds recommends allowing a 24-hour data collection period before you create a report.
  - **Charts** display data after a 10-minute data collection period.

Report types

DPA has many standard reports that include the most commonly used wait time statistics. You can customize each report by selecting the database instance, time interval, and the items included (for example, the wait types or SQL statements). The following types of reports are available:

- Average Wait

  Reports in this category show the average times for a single SQL statement or multiple SQL statements.

- Top <element>

  These reports show the files, SQL statements, users, or other elements that are experiencing the longest waits. For example, the Top Files report shows the busiest files ranked by total I/O wait time.

- Typical Day of <element> Wait

  These reports show the times of day when files, SQL statements, users, or other elements are experiencing the longest waits.

Learn more

To work with DPA reports, see the following topics:

- Access and run DPA reports
- Create a DPA report
- Search for a SQL statement to report on
- Schedule a DPA report for delivery
- Create and manage a DPA report group

Access and run DPA reports

From the Reports tab, you can view existing reports and create new reports.

1. On the DPA menu, click Reports.

   The Reports section lists the reports that have been created on this DPA server.

2. In the Reports section, you can:

   - Choose a database instance from the drop-down menu in the upper-right corner to filter the list of reports.
   - Click Show to run and open a report.
   - Click a column heading to sort the list of reports.
   - In the right column, click Delete to delete a report.

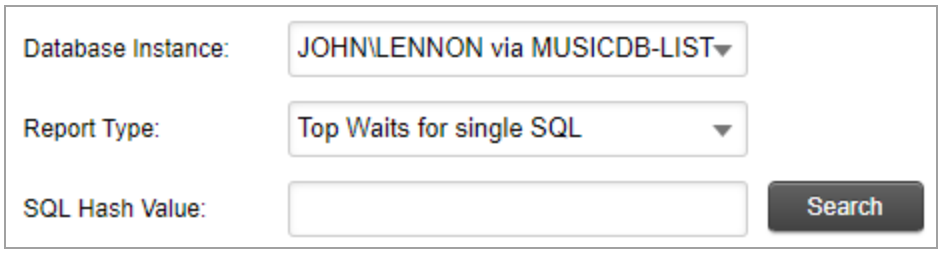
Create a DPA report

Use DPA reports to identify database trends and track the results of your performance tuning.

1. On the DPA menu, click Reports.
2. Select the Database Instance and the Report Type.

3. If the report type shows information about a **single** SQL, plan, or wait (for example, the Top Waits for single SQL report), identify the SQL, plan, or wait:

   a. Click Search next to the field that is added to the Create a New Report pane.



   b. Locate the SQL, plan, or wait and click OK to add it.

      To find a SQL statement, see Search for a SQL statement to report on.

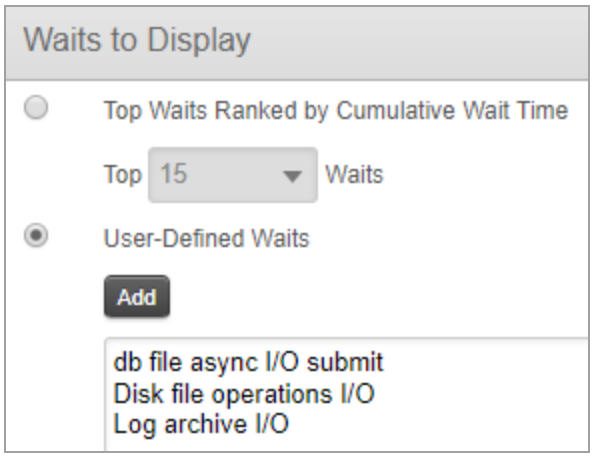
4. Click Report Options.

The Report - Advanced Options page opens.

5. Depending on the report type, specify which waits, SQL statements, or other elements to display in the report.

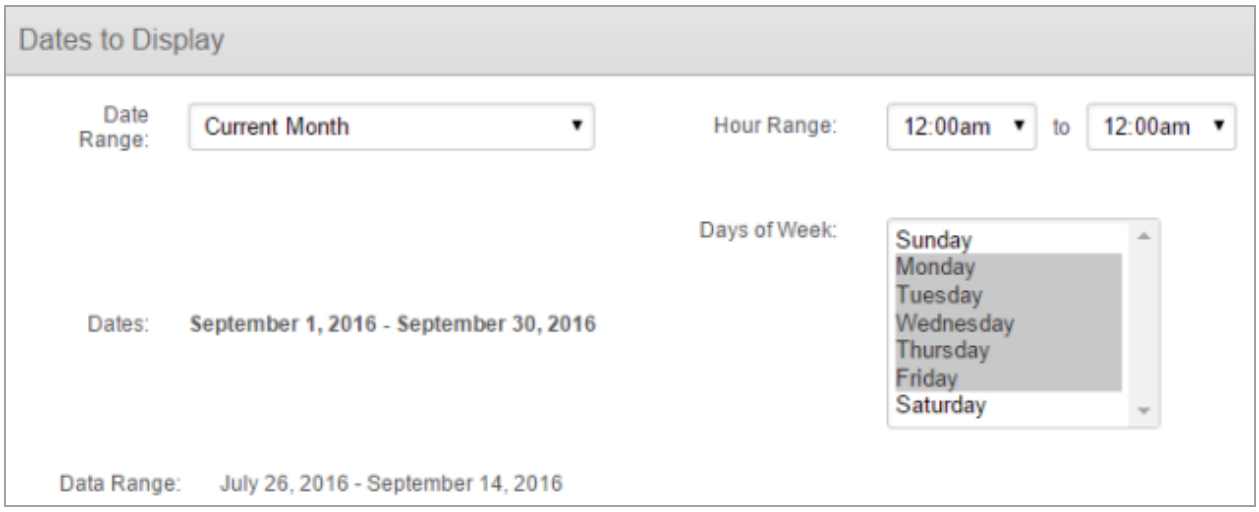
   By default, the report includes the elements with the highest wait time. To include specific elements, select User-Defined, click Add, and then use the Search box to locate and add up to 50 elements.

   To find a SQL statement, see Search for a SQL statement to report on.



6. Under Dates to Display, specify the dates that the report should include.

   The Data Range at the bottom of this section shows the time period for which data is available.



7. Under General, complete the following fields.

Report Name	Enter a unique name to identify this report in the report list.
Report Title	(Optional) Enter a title to display at the top of the report. If you leave this field blank, the report title defaults to the report type, database instance, and time period.
Report Description	(Optional) Enter a description to explain the report's content or purpose.

8. In the New Report section at the top of the window, click Display Report.

   The report opens.

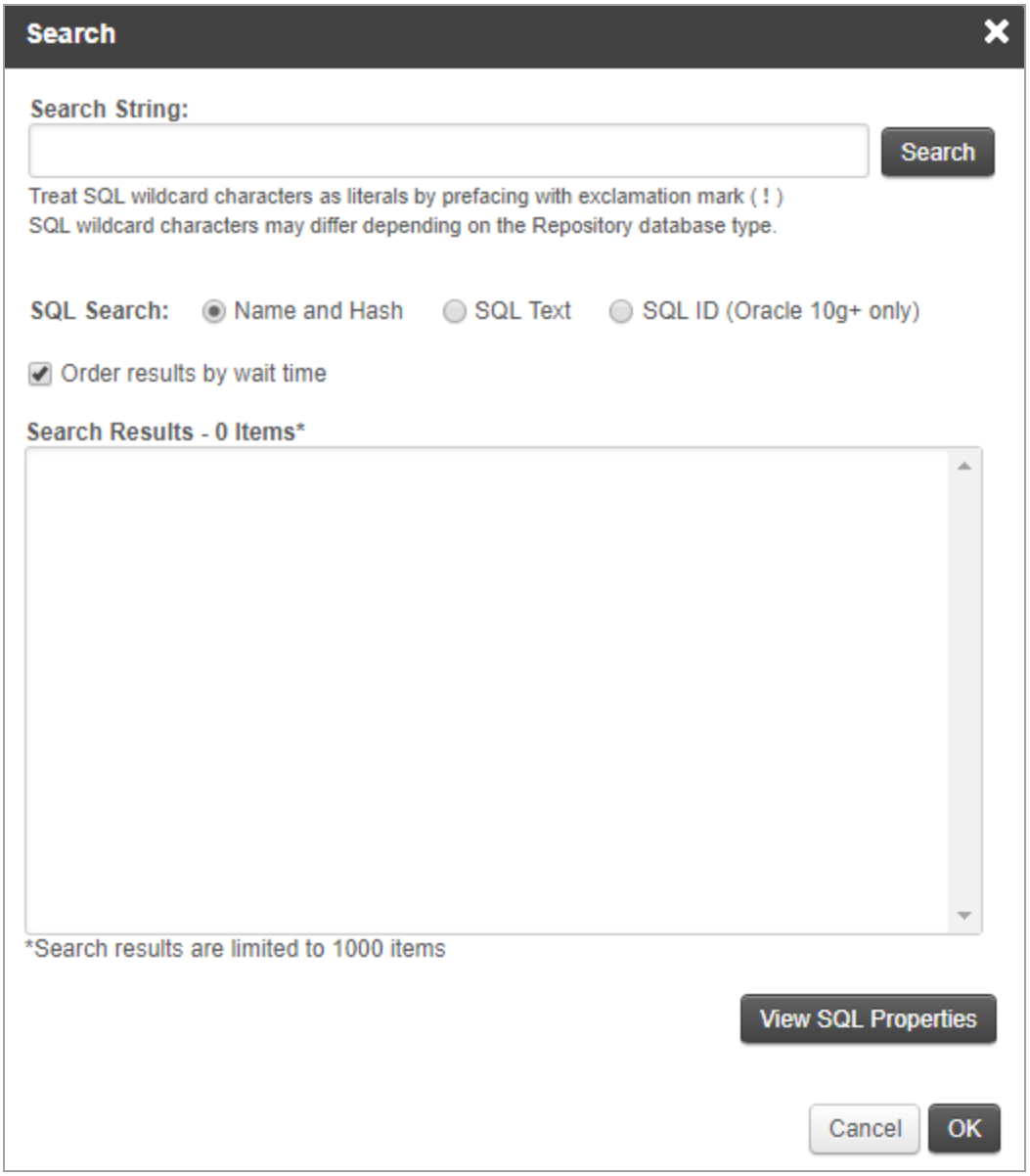
9. Choose one of the following options:

Click	If you want to
Save	Save the report with the name you entered previously.
Save As	Save the report with a different name.
Edit	Return to the Report - Advanced Options page and make changes.
Email Report	Send the report to one or more users.

You can view the report from the Reports tab at any time, or schedule the report to run automatically and be emailed to a group of recipients.

Search for a SQL statement to report on

When you create a report to show information about a single SQL statement or a group of SQL statements, you must identify the SQL statements. When you click the Search button on the Reports page or the Select Items to Display dialog, the following Search dialog opens:



To search for SQL statements, complete the following steps.

1. Select a SQL Search option, and then enter a string in the Search String:
   - **Name and Hash** (selected by default)

     If you named the SQL statement, enter part of the name. If not, enter part of the hash value that DPA uses to identify unnamed SQL statements.
   - **SQL Text**

     Enter a string that is included in the SQL statement. For example, entering `where` returns all SQL statements that include a `WHERE` clause (up to the limit of 1000 search results).

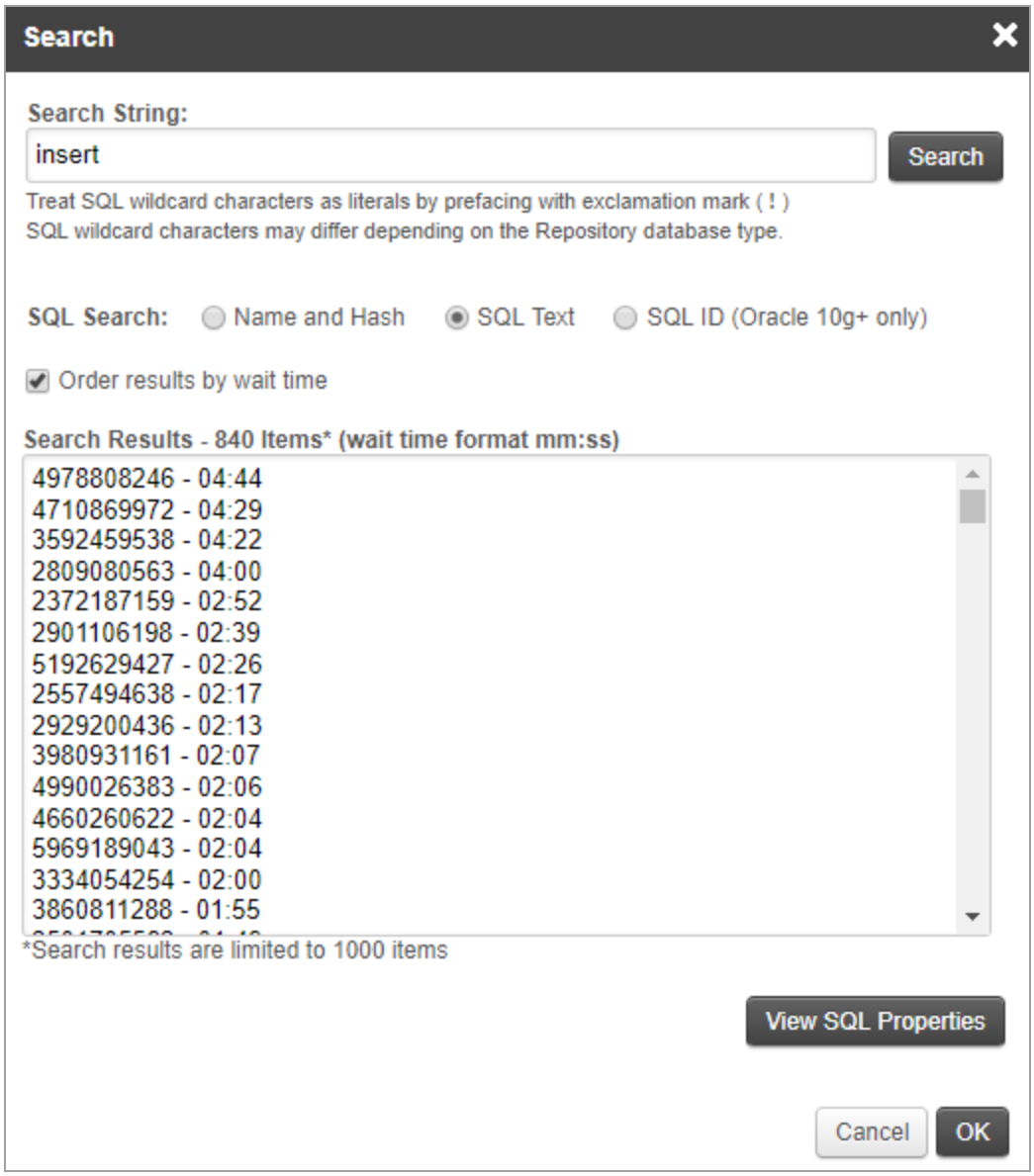
     The search string can include SQL wildcard characters. For example, the following search string uses `%` as a substitute for 0 or more characters:

     ```
     select%Project%where%Status
     ```

     It returns all `SELECT` statements against a table named `Project` that include a column named `Status` in the `WHERE` clause.
   - **SQL ID**

     For SQL statements that run against an Oracle 10g or later database, enter part of the SQL ID.
2. Click Search.

   By default, the Search Results box lists the name or hash value of each SQL statement, as well as the SQL statement's total wait time for the last seven days. The results are ordered by wait time with the highest waits first.



- To change the amount of wait time shown in the search results, set the advanced option `REPORT_SEARCH_WAIT_TIME_DAYS`.
- To list the search results in alphanumeric order without wait times, deselect Order results by wait time. Then click Search again.

3. Locate the SQL statement you want to report on, and select it.

   For more information about any SQL statement, select it and click View SQL Properties to see the SQL text.

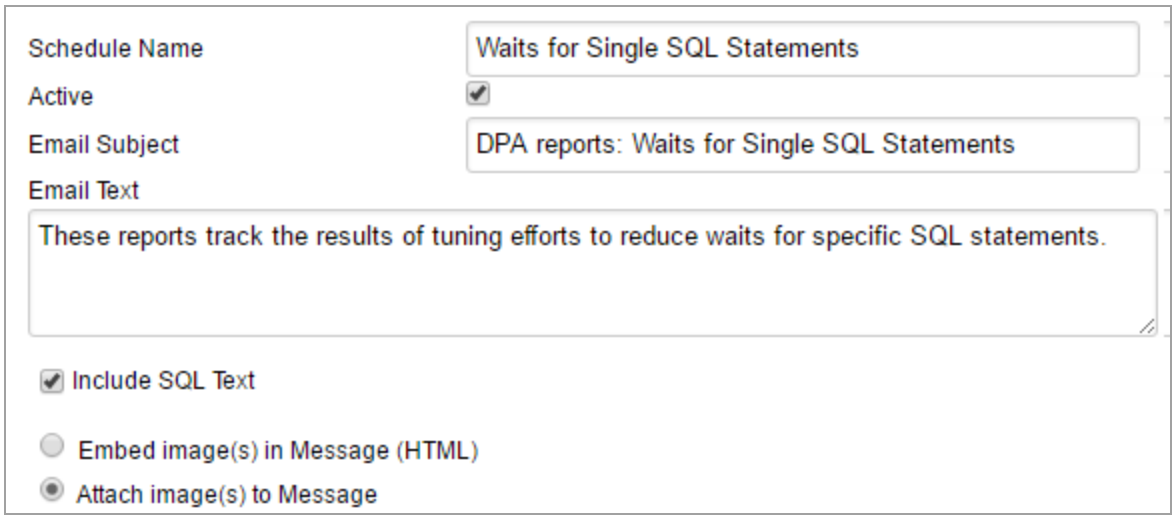
4. Click OK to add the SQL statement to the report.

Schedule a DPA report for delivery

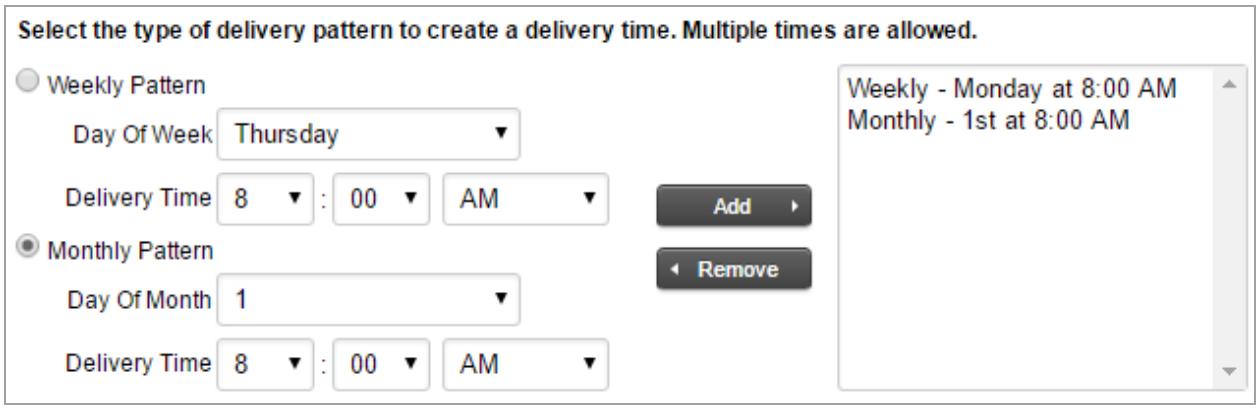
Report schedules automatically email reports (or report groups) at regular intervals. You can send reports to managers, team members, and customers. Use report schedules to communicate database trends to people who do not have direct access to DPA, and to and highlight performance improvements across your organization.

- Only DPA administrators can create report schedules.
- The report recipients must be added as DPA contacts.

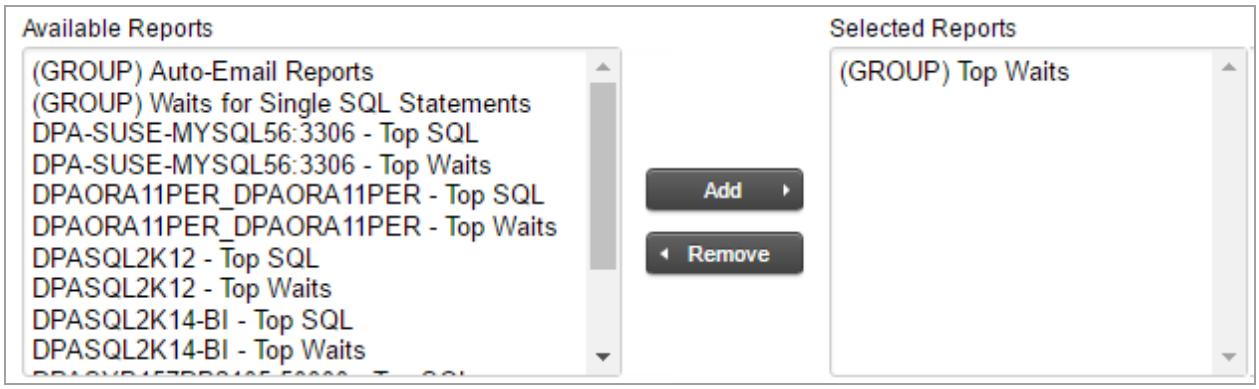
1. Log in to DPA using an account with administrator privileges.
2. On the DPA menu, click Reports.
3. Click the Report Schedules tab.
4. Click Create Schedule.
5. Name the schedule and enter an email subject and body text.



6. Specify when you want the report delivered, and click Add. You can specify multiple delivery times.



7. Under Available Reports, select the reports or report groups, and click Add.



8. If you want to review the email that will be sent when the schedule runs, click Send Test Email and enter an email address.
9. Under Available Contacts, select the recipients of the report, and click Add.

> If you have not added the recipients as contacts in DPA, click Add Contact or Add Contact Group.

10. Click Create Schedule.

    Your schedule is added to the list of report schedules.

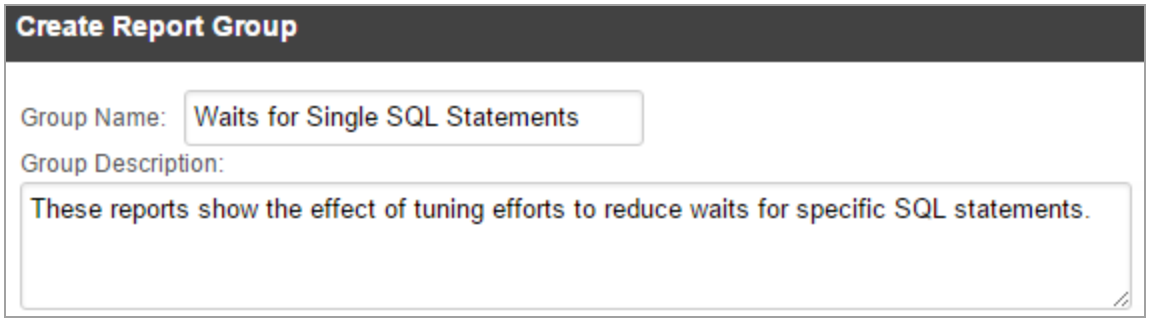
Does your network or firewall require an internal SMTP server? If so, see SMTP mail server for outgoing email.

Create and manage a DPA report group

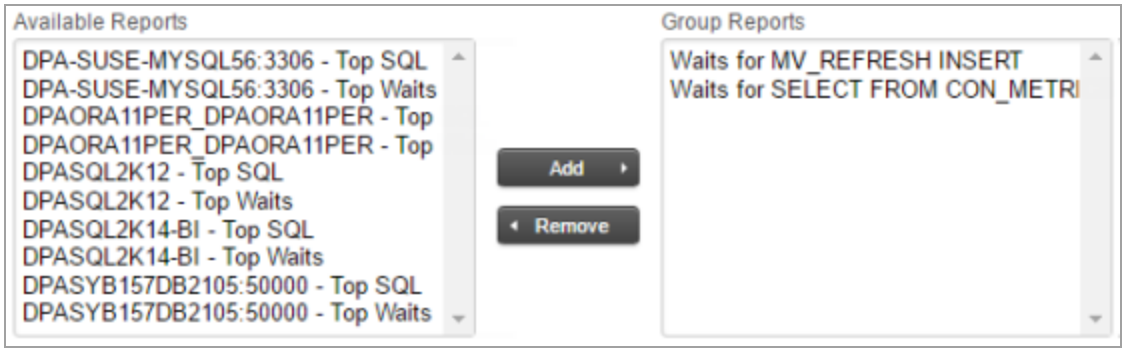
Use report groups to display data from related reports on the same page. With report groups, you can quickly run or schedule multiple reports.

Create a report group

1. On the DPA menu, click Reports.
2. Click the Report Groups tab.
3. Click Create Report Group.
4. Give the group a name and (optionally) a description.



5. Select the reports to include in this group and click Add.

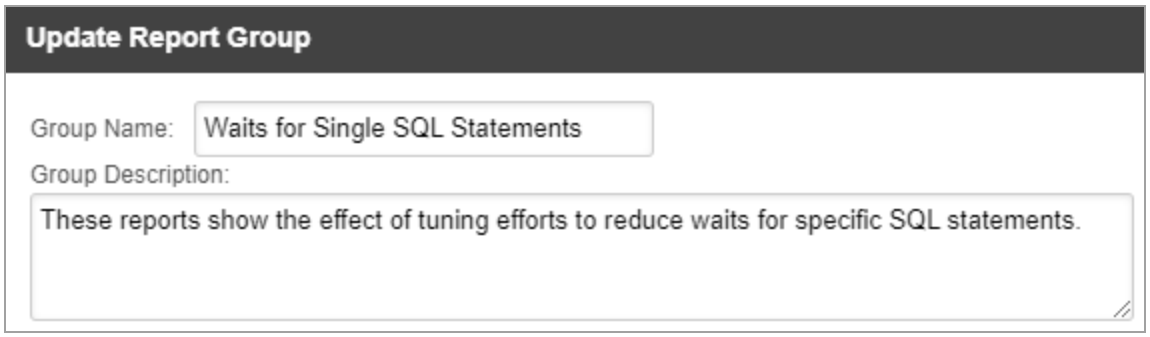


6. Click OK, and then click OK at the confirmation message.

   This group is added to the list of report groups.

Edit a report group

1. On the DPA menu, click Reports.
2. Click the Report Groups tab.
3. Click the name of the report to open the Update Report Group dialog.



4. Select the reports you want to add or remove, and then click the Add or Remove button.
5. Click OK, and then click OK at the confirmation message.

   This group is updated.

Delete a report group

1. On the DPA menu, click Reports.
2. Click the Report Groups tab.
3. Click the Delete button on the line of the report you want to delete.
4. Click Yes at the confirmation message.

   This group is removed from the list of report groups.

DPA alerts

Use DPA alerts to become aware of issues and address them proactively before they affect end users. Set thresholds on key wait time statistics, resource metrics, or standard administration indicators. The result is improved customer service, fewer help desk tickets, and increased compliance with database service-level agreements.

For information about creating or editing the email template that defines the contents of alert notifications, see Create or edit a custom email template for DPA alert notifications.

For information about setting up the email server that DPA uses to send notifications, see Configure the mail server used to send DPA emails.

View the status and history of DPA alerts

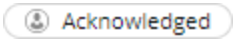
You can view the status of currently active alerts, and drill in to see alert details and history.

1. From the DPA menu in the upper-right corner, click Alerts.

   The Current alert status page shows information about DPA alerts that are currently active.

   - An icon indicates the current status (for example, High, Medium, Normal, or Broken).

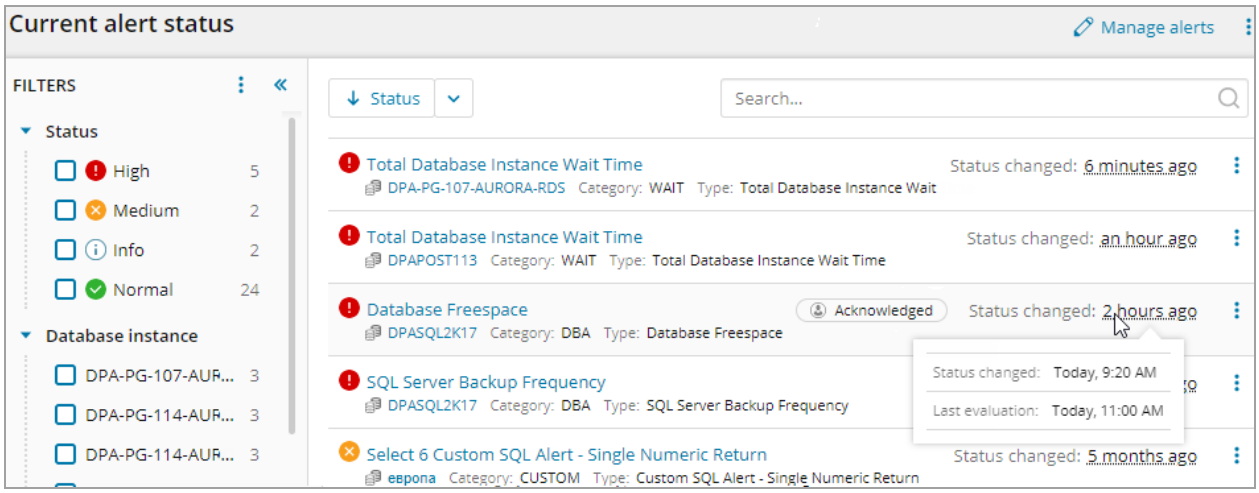
     > This page does not show alerts with the following statuses: Not Run, Unknown, and Inactive.

   - The 

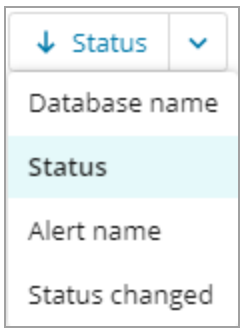
 icon identifies alerts that have been acknowledged. Hover over the icon to see who acknowledged it, when, and any comments.

     > Alerts with a status of Normal cannot be acknowledged.

   - Hover over the Status changed value to see when the alert was last evaluated.



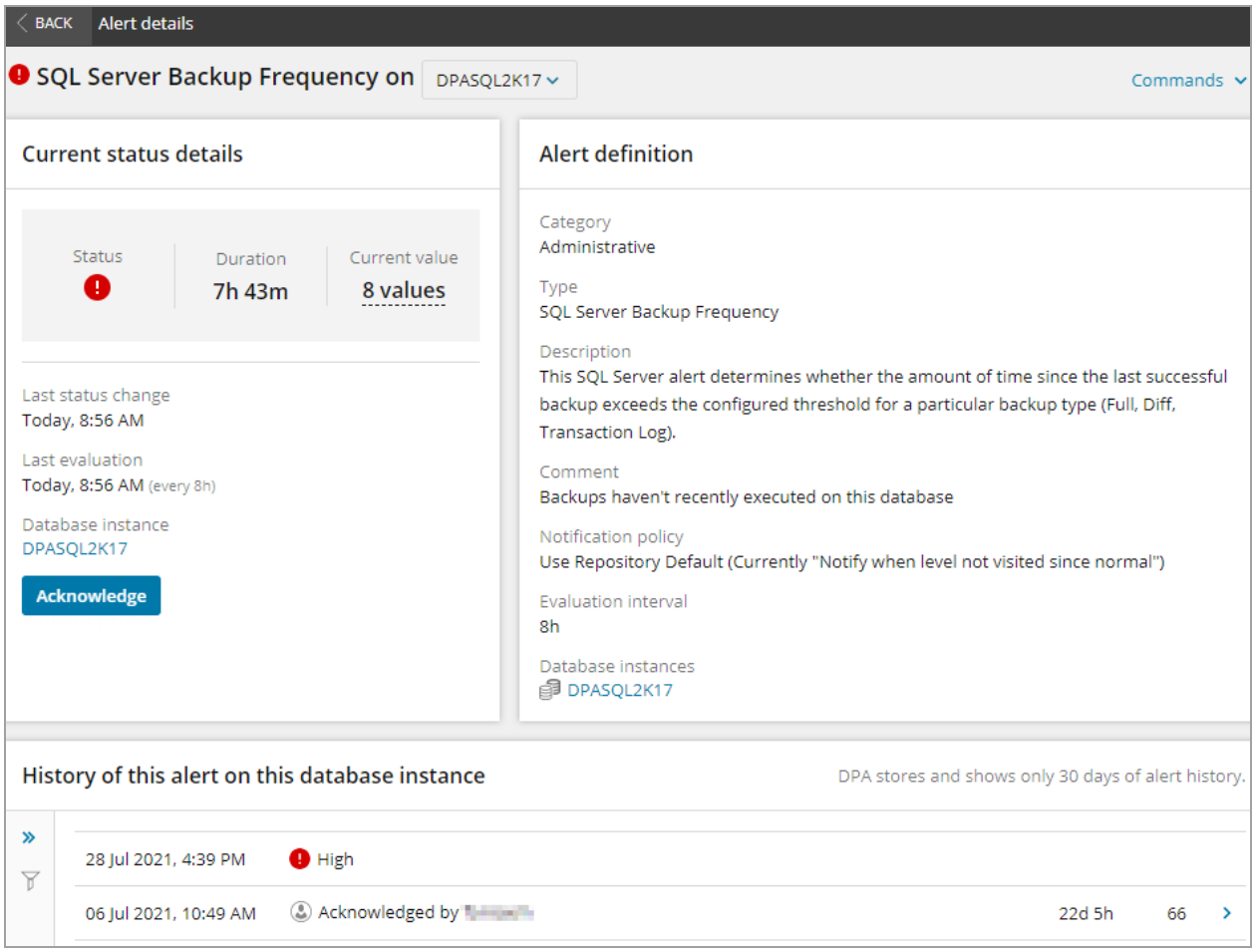
2. To find the alerts you are interested in:
   - Select one or more filters from the Filters list.
   - Enter a string in the Search text box to search for an alert name or database instance name.
   - Select an option from the drop-down menu above the list to change the sort order. By default, alerts are sorted by status.



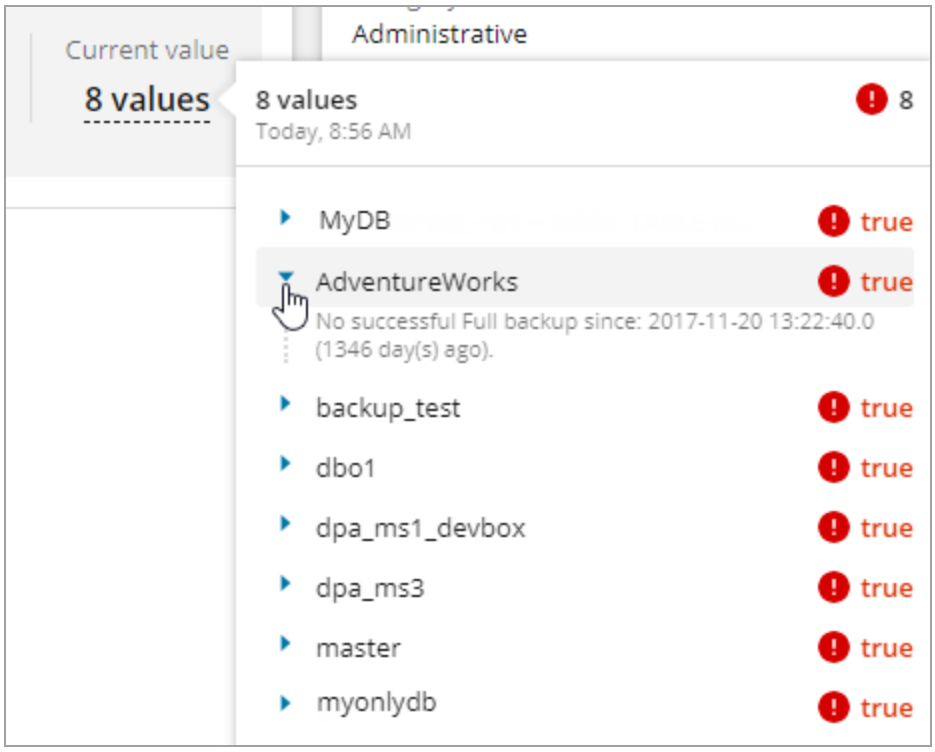
View alert details and history

1. To display additional information about the alert, do one of the following from the Current alerts status page:
   - Click the alert name.
   - Click the vertical ellipsis (⋮) on the right and choose Details and history.

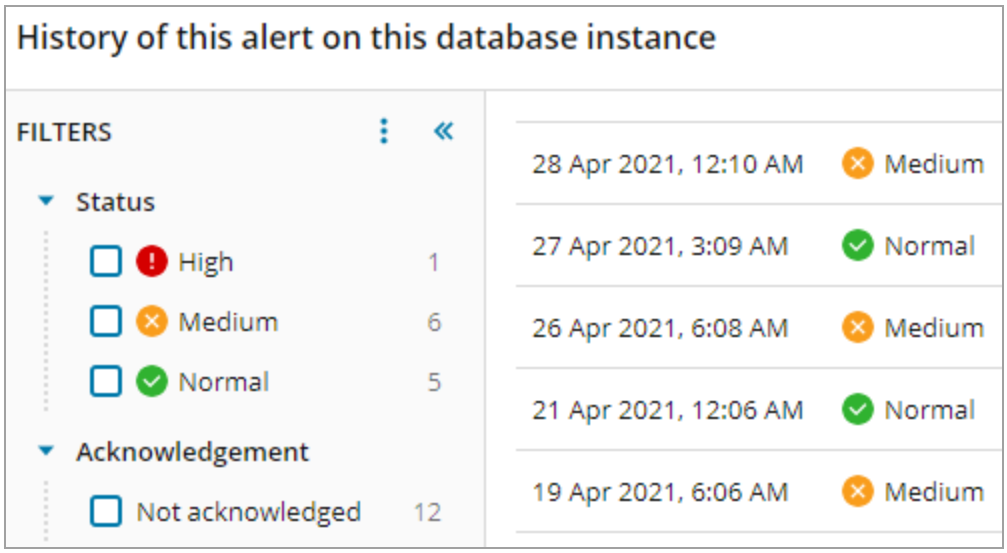
   The Alert details page displays the alert's current status, definition, and history. The History section lists the dates and times when the alert status changed or when the alert was acknowledged or unacknowledged.



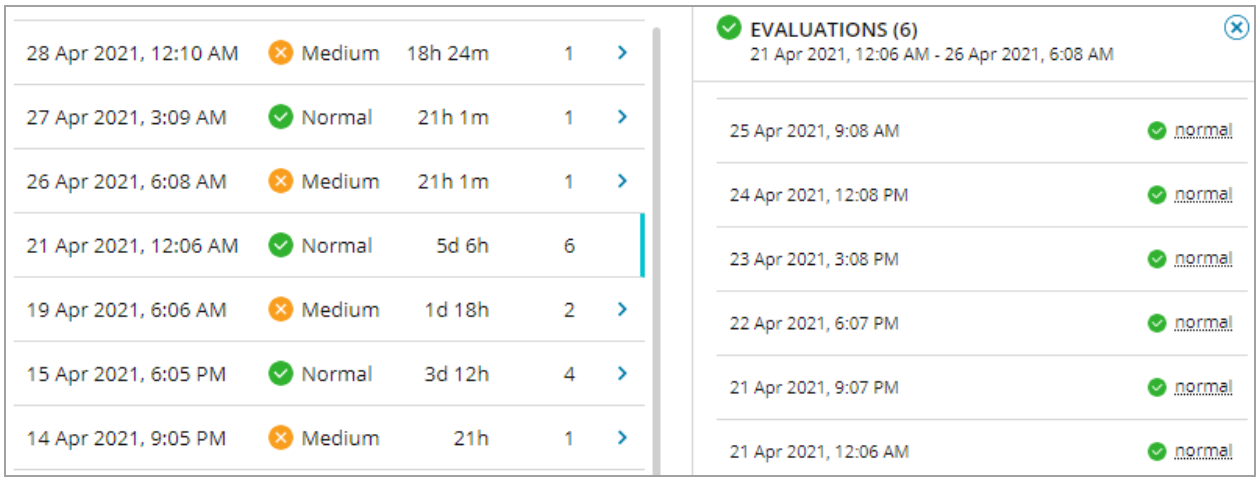
2. If there are multiple values, hover the Current value to display the values for each database instance that triggered the alert. Click a line with a blue arrow to display details about that value.



3. To filter items in the History section, expand the Filters pane on the left.



4. Click the > on the left side of an item in the History section to see the list of evaluations at this status.



Acknowledge or unacknowledge a DPA alert

After an alert is triggered, you can acknowledge it to indicate that the appropriate people are aware of the issue and it is being addressed. Acknowledging an alert prevents DPA from sending further notifications for that triggered alert.

> If the alert status returns to Normal and then the alert is triggered again, DPA sends a new notification.

DPA automatically records when the alert was acknowledged and the account that acknowledged it. You can also add notes that other users can read. In addition to preventing further notifications, acknowledging an alert can provide an audit trail, help DBAs focus on alerts that still require attention, and prevent multiple people from working on the same issue.

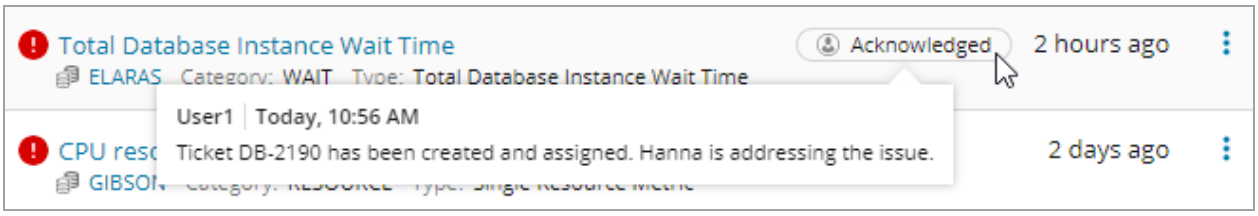
If you acknowledge an alert in error, or if the situation changes, you can unacknowledge the alert. When an alert is unacknowledged, DPA begins sending notifications again based on the specified notification policy.

Determine if an alert has been acknowledged and view notes

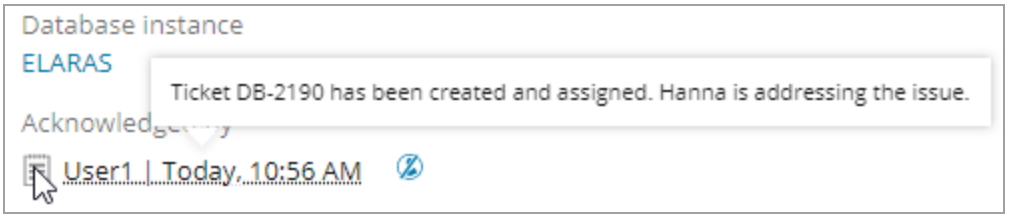
On the DPA menu, click Alerts to open the Current alert status page. This page shows information about each DPA alert that is currently triggered. Acknowledged alerts are labeled.

> To quickly find acknowledged or unacknowledged alerts, apply one of the Acknowledgement filters.

Hover over the Acknowledged label to see when it was acknowledged, what user account acknowledged it, and any notes.



If you click the alert title to open the Alert details page, the Acknowledged by field shows when the alert was acknowledged and what user account acknowledged it. Hover over the text or the notes icon to display any notes.



Acknowledge an alert

You can acknowledge an alert from the Current alert status page or the Alert details page.

If the alert status is currently Normal, the options to acknowledge the alert are **not** shown. You can acknowledge an alert only if the status is not Normal.

- From the Current alert status page:
  1. Click the vertical ellipsis (⋮) on the alert row, and choose Acknowledge alert.

     The Acknowledge alert dialog box opens.
  2. Enter a note (optional), and then click Acknowledge.
- From the Alert details page:
  1. In the Current status details pane, click Acknowledge.

     The Acknowledge alert dialog box opens.
  2. Enter a note (optional), and then click Acknowledge.

Unacknowledge an alert

After an alert has been acknowledged, you can unacknowledge it from the Current alert status page or the Alert details page.

- From the Current alert status page:
  1. Click the vertical ellipsis (⋮) on the alert row, and choose Unacknowledge alert.

     The Unacknowledge alert dialog box opens.
  2. Enter a note (optional), and then click Unacknowledge.
- From the Alert details page:
  1. In the Current status details pane, click the Unacknowledge alert icon (⊘) to the right of the acknowledgment information.

     The Unacknowledge alert dialog box opens.
  2. Enter a note (optional), and then click Unacknowledge.

DPA alert categories and types

DPA provides a wide range of alert types in four categories.

DPA alert categories

DPA provides the following alert categories:

- Wait Time alerts are triggered when wait time exceeds a user-defined threshold, or when wait time is much higher than expected (an anomaly).
- Resources alerts are triggered when a resource metric, such as CPU utilization or memory usage, exceeds its threshold.
- Administrative alerts are used to monitor the health of the database system.
- Custom alerts are user-defined SQL statements or stored procedures that are run against the monitored database or the DPA repository.

DPA Wait Time alert types

For Wait Time alerts, DPA evaluates the amount of wait time that occurred during each alert execution interval. An alert is triggered if the wait time during an interval exceeds the specified threshold.

ⓘ To create a Wait Time alert, see Configure a DPA Wait Time alert.

Alert type	DB type	Description
Database Instance Wait Time Anomaly	All	Alerts you when the wait time of an instance was abnormally high during the most recently completed hour. The wait time status is calculated at the top of each hour using the DPA anomaly detection algorithm.
Total Database Instance Wait Time	All	Alerts you when the total wait time for an entire database instance exceeds the threshold.
Total SQL Wait Time for a Single SQL	All	Alerts you when the total execution time for the specified SQL statement exceeds the threshold.
Average Wait Time for a Single SQL	All	Alerts you when the average execution time for the specified SQL statement exceeds the threshold.
Total SQL Wait Time for Single Wait	All	Alerts you when the total wait time for the specified wait type or event exceeds the threshold.
Total SQL Wait Time - Program	All	Alerts you when the total execution time for SQL statements executed by the specified program or application exceeds the threshold.
Total SQL Wait Time - Database User	All	Alerts you when the total execution time for SQL statements executed by the specified database user exceeds the threshold.
Total SQL Wait Time - O/S User	Oracle	Alerts you when the total execution time for SQL statements executed by the specified operating system (OS) user exceeds the threshold.
Total SQL Wait Time - Machine	All	Alerts you when the total execution time for SQL statements executed on the specified computer exceeds the threshold.
Total SQL Wait Time - Database	All **except** Oracle	Alerts you when the total execution time for the specified database in an instance exceeds the threshold.

Alert type	DB type	Description
Total SQL Wait Time - Custom for Oracle	Oracle	Alerts you when the total wait time for any combination of the following exceeds the threshold: • Specified wait type or event • Specified SQL statement • SQL statements executed by the specified program or application • SQL statements executed by the specified database user • SQL statements executed by the specified OS user • SQL statements executed on the specified computer
Total SQL Wait Time - Custom for SQL Server/Azure SQL MI/Sybase	SQL Server, ASMI, Sybase	Alerts you when the total wait time for any combination of the following exceeds the threshold: • Specified wait type or event • Specified SQL statement • SQL statements executed by the specified program or application • SQL statements executed by the specified database user • SQL statements executed on the specified computer • Specified database in an instance
Total SQL Wait Time - Custom for Azure SQL Database	Azure SQL Database	Alerts you when the total wait time for any combination of the following exceeds the threshold: • Specified wait type or event • Specified SQL statement • SQL statements executed by the specified program or application • SQL statements executed by the specified database user • SQL statements executed on the specified computer

Alert type	DB type	Description
Total SQL Wait Time - Custom for MySQL	MySQL	Alerts you when the total wait time for any combination of the following exceeds the threshold: • Specified wait type or event • Specified SQL statement • SQL statements executed by the specified program or application • SQL statements executed by the specified database user • SQL statements executed on the specified computer • Specified database in an instance • Wait events associated with the specified wait instrument • SQL statements that perform the specified operation, such as select or fetch
Total SQL Wait Time - Custom for Db2	Db2	Alerts you when the total wait time for any combination of the following exceeds the threshold: • Specified wait type or event • Specified SQL statement • SQL statements executed by the specified program or application • SQL statements executed on the specified computer • Specified database (the Database field is ignored when a single database is being monitored)
Total SQL Wait Time - Custom for PostgreSQL	PostgreSQL	Alerts you when the total wait time for any combination of the following exceeds the threshold: • Specified wait type or event • Specified SQL statement • SQL statements executed by the specified program or application • SQL statements executed by the specified database user • SQL statements executed on the specified computer • Specified database in an instance

Alert type	DB type	Description
RAC Overhead Wait Time	Oracle	Alerts you when the total wait time for RAC events exceeds the threshold.
Total Blocking Wait Time	Oracle, SQL Server, Azure SQL DB, ASMI, Sybase, MySQL, PostgreSQL	Alerts you when the amount of time that sessions waited due to blocking exceeds the threshold. Wait time due to blocking is shown on the Blocking tab.

DPA Resources alert types

For Resources alerts, DPA looks at the metric values collected during each execution interval and applies the calculation that you specify in the alert definition (percentage, average, median, maximum, or minimum). An alert is triggered if the resulting calculated value exceeds the specified threshold.

Resource alert types apply to any database type.

To create a Resources alert, see Configure a DPA Resources alert.

Alert type	DB type	Description
Single Resource Metric	All	Alerts you if the calculated value for a specific resource metric exceeds the threshold.
All Metrics in a Category	All	Alerts you if the calculated value for all resource metrics a category exceeds the threshold.

DPA Administrative alert types

Alerts in the Administrative category are triggered when DPA detects certain conditions or events.

To create an Administrative alert, see Configure a DPA Administrative alert.

Alert type	DB type	Description
Database Instance Availability	All	Alerts you if a monitored database instance is not available. DPA determines availability by attempting to connect to the instance.

Alert type	DB type	Description
Database Freespace	SQL Server, Azure SQL DB, ASMI, Sybase, Db2	Alerts you if the percentage of free space in any database in the instance (or in the single database for Azure SQL databases) falls below the threshold.
Database Instance Parameter Changes	All **except** MySQL	Alerts you if any database instance parameter settings are changed.
Tablespace Freespace	Oracle, Db2	Alerts you if the percentage of free space in any tablespace in the monitored instance falls below the threshold. For Db2, only database-managed tablespaces (DMS) are evaluated.
Transaction Log Freespace	SQL Server, ASMI, Sybase, Db2	Alerts you if the percentage of free space in the transaction log of any database in the instance falls below the threshold.
Oracle PDB Move	Oracle	Alerts you when a monitored PDB database instance is moved to or from a CDB container.
Oracle PDB Database Instance Availability	Oracle	Alerts you if a monitored PDB database instance is not available. DPA determines availability by attempting to connect to the instance.
Oracle Alert Log Entries	Oracle	Alerts you when the alert log contains more than the minimum number of occurrences of the specified string. DPA searches for the specified string in the alert log (from the `x$dbgalertext` table) and returns all unique matching entries and the count of each entry. The thresholds for each alert level specify the minimum and maximum number of occurrences for that level.
Oracle Long Running Transaction	Oracle	Alerts you when a transaction runs for more than the number of seconds specified by the threshold.
Oracle Percent Redo Logs Unarchived	Oracle	Alerts you when the percentage of unarchived redo logs exceeds the threshold.

Alert type	DB type	Description
Oracle Redo Log Switching Frequency	Oracle	Alerts you if the number of redo log switches during an execution interval exceeds the threshold. To avoid alerts during periods when frequent log switches are expected, you can specify a time range to include or exclude.
Oracle Session Limit	Oracle	Alerts you if the percentage of active sessions exceeds the threshold. To determine the percentage of active sessions, DPA compares the number of active sessions to the maximum number of sessions. The maximum number of sessions is configured in the `v$parameter` 'sessions' row.
Oracle Stale Statistics	Oracle	Alerts you if tables or indexes have stale or empty statistics. You can specify schemas to include or exclude. The alert notification lists all tables and indexes (in included schemas) that have stale or empty statistics. To avoid repetitive alerts, SolarWinds recommends that you increase the execution interval for this alert to at least one day.
Oracle Archiver Errors	Oracle	Alerts you if the archiver process receives an error while trying to archive a redo log or if the online log files are not being archived. If the problem is not resolved quickly, the database stops executing transactions. This alert is typically triggered when the destination device is out of space to store the redo log file.
Windows Service Not Running - SQL Server	SQL Server	Alerts you if the specified Windows service is not running in the selected SQL Server database instances.
SQL Server Abnormal Mirroring Status	SQL Server	Alerts you if the SQL Server mirroring status is anything other than Synchronized.
SQL Server Availability Group Failover	SQL Server	Alerts you if an availability group has failed over from one instance to another.
SQL Server Availability Group Status Change	SQL Server	Alerts you if an availability group has had a change in status to Partially Healthy or Not Healthy. An Alert Level of High indicates Not Healthy and Medium indicates Partially Healthy.

Alert type	DB type	Description
SQL Server Deadlocks	SQL Server, ASMI	Alerts you if the number of deadlocks that occurred on an instance exceeds the threshold.
SQL Server Error Log Alert	SQL Server, ASMI	Alerts you if error logs contain a specified string or pair of strings.
SQL Server/Azure SQL Ineffective Statistics	SQL Server, ASMI, Azure SQL DB	Alerts you if indexes have ineffective statistics. It uses criteria such as time since last stats update, percent of rows changed, and table size. You can specify database to include or exclude. The alert notification lists all indexes (in included databases) with ineffective statistics. To avoid repetitive alerts, SolarWinds recommends that you increase the execution interval for this alert to at least one day.
SQL Server Job Failure	SQL Server, ASMI	Alerts you if a job fails. The alert notification reports all jobs that failed since the last time this alert was executed.
SQL Server Backup Frequency	SQL Server	Alerts you if the number of days since the last backup of the specified type (full, differential, or transaction log) exceeds the threshold.
SQL Server Recovery Backup Assets Size	SQL Server	Alerts you if the size of all backup assets required to recover a database exceeds the threshold.
SQL Server Backup Jobs Running	SQL Server	Alerts you if the number of currently running backup jobs for a specified database instance exceeds the threshold.
SQL Server Backup Time Allotted	SQL Server	Alerts you if the time required to complete the last backup of the specified type (full, differential, or transaction log) exceeds the threshold.
SQL Server Log has Many Virtual Logs	SQL Server, ASMI	Alerts you if the number of Virtual Logs in any database exceeds the threshold. To limit the result set and reduce the required DPA Repository space, set the Minimum number of virtual logs.
SQL Server Long Running Jobs	SQL Server, ASMI	Alerts you if any jobs (in SQL Agent) are running longer than two standard deviations from the mean execution time based on previous execution times.

Alert type	DB type	Description
DPA Database Instance Monitor Errors	All	Alerts you if any errors occurred while monitoring a database instance.
DPA Resource Collection Errors	All	Alerts you if any errors occurred while collecting resource data for a database instance.
MySQL Temporary Tables Creation Rate	MySQL	Alerts you if the number of temporary tables created per second exceeds the threshold.
MySQL Temporary Tables on Disk Creation Rate	MySQL	Alerts you if the number of temporary tables created on disk per second exceeds the threshold.
MySQL Schema Freespace	MySQL	Alerts you if the percentage of free space in any schema exceeds the threshold. Free space is occupied by the MySQL database, but it is not being used.
MySQL Table Freespace	MySQL	Alerts you if the percentage of free space in any table exceeds the threshold. Free space is occupied by the MySQL database table, but it is not being used.
MySQL Oversized Index	MySQL	Alerts you if any table has defined indexes that occupy more overall space than the percentage specified as the threshold.
MySQL Tables Missing Primary Key	MySQL	This alert determines if any of the MySQL tables does not have PK defined
MySQL Latest Deadlock Alert	MySQL	Alerts you if there is an unseen deadlock in a database.
MySQL InnoDB Buffer Pool Utilization Alert	MySQL	Alerts you if the percentage of free buffer pages is below the threshold. Data reflects activity against tables managed by the InnoDB (or an InnoDB-based) storage engine.
MySQL InnoDB Log File Size Alert	MySQL	Alerts you if the REDO log file size is below the threshold. Data reflects activity against tables managed by the InnoDB (or an InnoDB-based) storage engine.

Alert type	DB type	Description
MySQL File Sorts on Disk Alert	MySQL	Alerts you if a database instance frequently sorts to disk instead of performing the sort in memory. The alert is triggered if the number of disk sorts during an execution interval exceeds the threshold.
MySQL Replication Seconds Behind Master Alert	MySQL	Alerts you if the replication (slave) instance is more than the specified number of seconds behind the master.
MySQL Replication Threads Availability Alert	MySQL	Alerts you if the replication (slave) MySQL database threads (I/O and slave) are unavailable.
MySQL Redundant Indexes Alert	MySQL	Alerts you if any tables in a database instance contain redundant indexes.
MySQL Duplicate Indexes Alert	MySQL	Alerts you if any tables in a database instance contain duplicate indexes.
PostgreSQL Autovacuum Status	PostgreSQL	Alerts you if the autovacuum process is `OFF`. The autovacuum process helps prevent table bloat.
PostgreSQL Track Counts Status	PostgreSQL	Alerts you if the `track_counts` setting is `OFF`. The `track_counts` setting must be `ON` to allow PostgreSQL to collect statistics on database activity. The autovacuum process requires these statistics.
PostgreSQL Track Activities Status	PostgreSQL	Alerts you if the `track_activities` setting is `OFF`. The `track_activities` setting enables tracking of currently executing SQL statements, and it must be `ON` to allow DPA to monitor the database instance.

Alert type	DB type	Description
PostgreSQL Last Analyze	PostgreSQL	If the PostgreSQL autovacuum process is disabled, it does not automatically trigger an analyze operation to update statistics. This alert warns you if the time period since the last manually run analyze operation exceeds a threshold. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.
PostgreSQL Last Auto Analyze	PostgreSQL	The PostgreSQL autovacuum process can automatically trigger an analyze operation, which updates the statistics used to determine query plans. This alert warns you if the time period since the last automatically triggered analyze operation exceeds a threshold. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.
PostgreSQL Last Vacuum	PostgreSQL	If the PostgreSQL autovacuum process is disabled, it does not run automatically to remove dead tuples (outdated versions of rows that are no longer needed). This alert warns you if the time period since the last manual vacuum operation exceeds a threshold. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.
PostgreSQL Last AutoVacuum	PostgreSQL	The PostgreSQL autovacuum process runs automatically to remove dead tuples (outdated versions of rows that are no longer needed). This alert warns you if the time period since the last autovacuum exceeds a threshold. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.

Alert type	DB type	Description
PostgreSQL Long Running Vacuum	PostgreSQL	This alert notifies you when a vacuum operation runs for longer than the specified threshold. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.
PostgreSQL Long Running Query	PostgreSQL	This alert notifies you when any query other than the autovacuum process runs for longer than the specified threshold. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.
PostgreSQL Dead Tuple	PostgreSQL	This alert monitors the row count or percentage of dead tuples in the database instance. A high percentage can indicate that the PostgreSQL vacuuming process is not healthy. A dead tuple is an outdated version of a row that was updated or deleted. It is no longer needed by any transaction, and the vacuuming process should remove it so the space can be used for new rows. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.
PostgreSQL Total Idle in Transaction Connections	PostgreSQL	Each PostgreSQL instance has a maximum number of connections (max_connections in pg_settings). This alert warns you when a high percentage of the available connections are “idle in transaction”, meaning that an application or query started a transaction but the transaction is now idle, possibly waiting on something else. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.

Alert type	DB type	Description
PostgreSQL Total Connections	PostgreSQL	Each PostgreSQL instance has a maximum number of connections (max_connections in pg_settings). This alert warns you when a high percentage of the available connections are in use. It includes connections in all states. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.
PostgreSQL User Role Expiry	PostgreSQL	This alert notifies you when a user role will expire within the specified number of days or has already expired. This alert can use data from multiple databases in the cluster. To enable DPA to collect data from multiple PostgreSQL databases, you must configure a foreign data wrapper.
PostgreSQL Total Table Bloat in Database	PostgreSQL	This alert warns you if the percentage of bloat (unused space that was not reclaimed) exceeds a threshold.
PostgreSQL Collect Database Size	PostgreSQL	This alert collects the size of each database in a PostgreSQL cluster. This alert is never triggered. The collected data is used as input for the PostgreSQL Database/Table Percentage Growth alert.
PostgreSQL Collect Relation Table Size	PostgreSQL	This alert collects the size of each relation (table) in a PostgreSQL cluster. This alert is never triggered. The collected data is used as input for the PostgreSQL Database/Table Percentage Growth alert.
PostgreSQL Database/Table Percentage Growth	PostgreSQL	This alert warns you if the size of the database or the size of all relations increase by more than the specified percentage during an execution interval. Depending on which size you are monitoring, the PostgreSQL Collect Database Size alert or the PostgreSQL Collect Relation (Table) Size alert must also be configured.

DPA Custom alert types

Custom alerts are triggered based on the value(s) returned by a user-defined SQL statement or stored procedure. Custom alerts apply to all database types.

To create a Custom alert, see Configure a DPA Custom alert.

Alert type	DB type	Description
Custom SQL Alert - Single Numeric Return	All	Executes a user-defined SQL statement that returns a single numeric value. The alert is triggered if the value exceeds a threshold.
Custom SQL Alert - Multiple Numeric Return	All	Executes a user-defined SQL statement that returns one or more rows with a string in the first column and a numeric value in the second column. The alert is triggered if any numeric value exceeds a threshold.
Custom SQL Alert - Single Boolean Return	All	Executes a user-defined SQL statement that returns a single string value of `TRUE` or `FALSE` (not case-sensitive). The alert is triggered if the SQL statement returns `TRUE`.
Custom SQL Alert - Single Alert Status Return	All	Executes a user-defined SQL statement that returns a single string value that specifies the alert status. Valid values are `NORMAL`, `INFO`, `LOW`, `MEDIUM`, and `HIGH` (not case-sensitive).
Custom Procedure Alert - Single Numeric Return	All	Executes a user-defined stored procedure that returns a single numeric value. The alert is triggered if the value exceeds a threshold.
Custom Procedure Alert - Single Boolean Return	All	Executes a user-defined stored procedure that returns a single string value of `TRUE` or `FALSE` (not case-sensitive). The alert is triggered if the stored procedure returns `TRUE`.
Custom Procedure Alert - Single Alert Status Return	All	Executes a user-defined stored procedure that returns a single string value that specifies the alert status. Valid values are `NORMAL`, `INFO`, `LOW`, `MEDIUM`, and `HIGH` (not case-sensitive).

Create a DPA Wait Time alert

Wait Time alerts notify you when the amount of time users or applications waited on the database was high. These alerts are triggered when wait time exceeds a user-defined threshold, or when wait time is much higher than expected (an anomaly).

1. From the DPA menu in the upper-right corner, click Alerts.
2. In the upper-right corner, click Manage alerts.
3. Select Wait Time as the Alert Category, select the Alert Type, and then click Create Alert.

   > To find out more about each alert type, see DPA alert categories and types. Or select a type to display a description on the right.

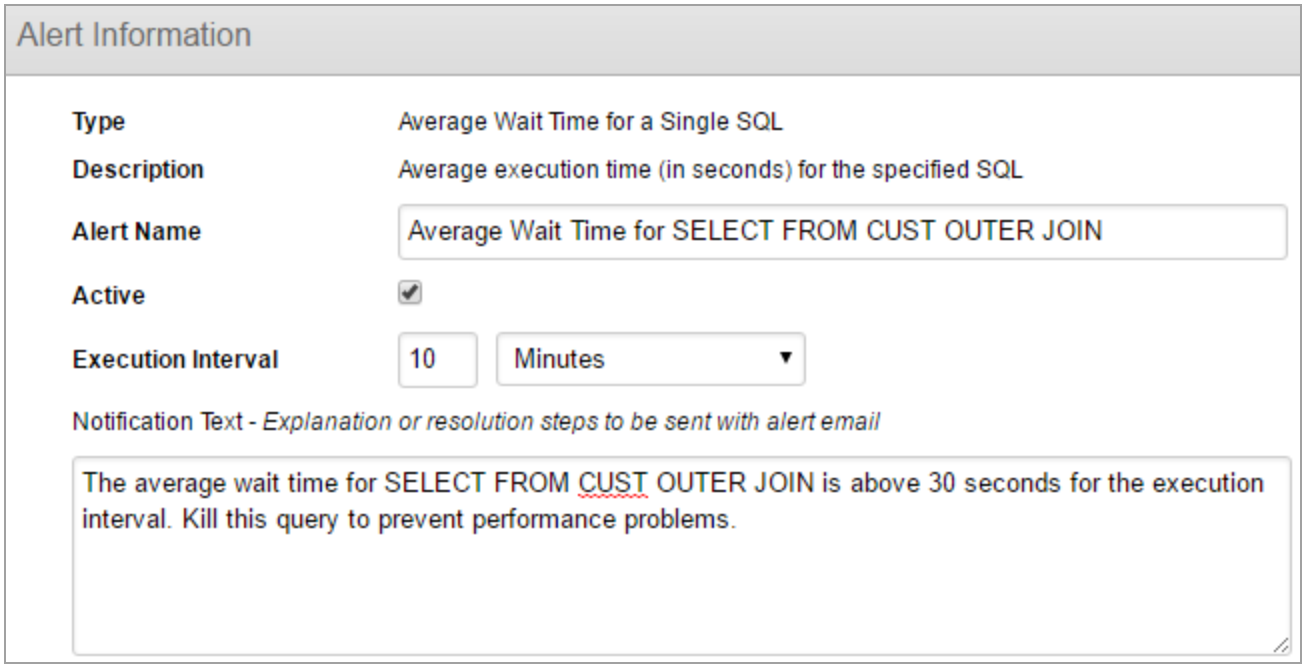
4. In the Alert Information section:
   a. Enter a unique name.
   b. If you want to disable the alert, clear the Active checkbox.
   c. Select the execution interval.

      The execution interval specifies how often the alert runs and the amount of data that DPA examines. For example, if the execution interval is 10 minutes, DPA executes the alert every 10 minutes and examines the last 10 minutes of data to determine whether to trigger the alert. DPA recommends an execution interval of **at least 10 minutes** to prevent unnecessary alerts from a single slow execution or temporary condition.

   d. Enter the notification text to be sent with the email notification. Include an explanation of the issue and the suggested resolution.

      > If you apply a custom email template to this alert, the email notification includes the notification text if the email template contains one of the following variables:
      >
      > - Alert Notification text: `[=alert.notificationText]`

- Results: `[=dpa.body]` (included by default)



Under Database Instances, specify the database instances that the alert applies to. You can use a rule to identify instances that meet one or more conditions, or you can manually select the instances.

5.

- When you select a rule, DPA uses the rule conditions to determine which instances the alert monitors. If your environment changes, the list of instances is automatically updated.

  a. Click Use rule.

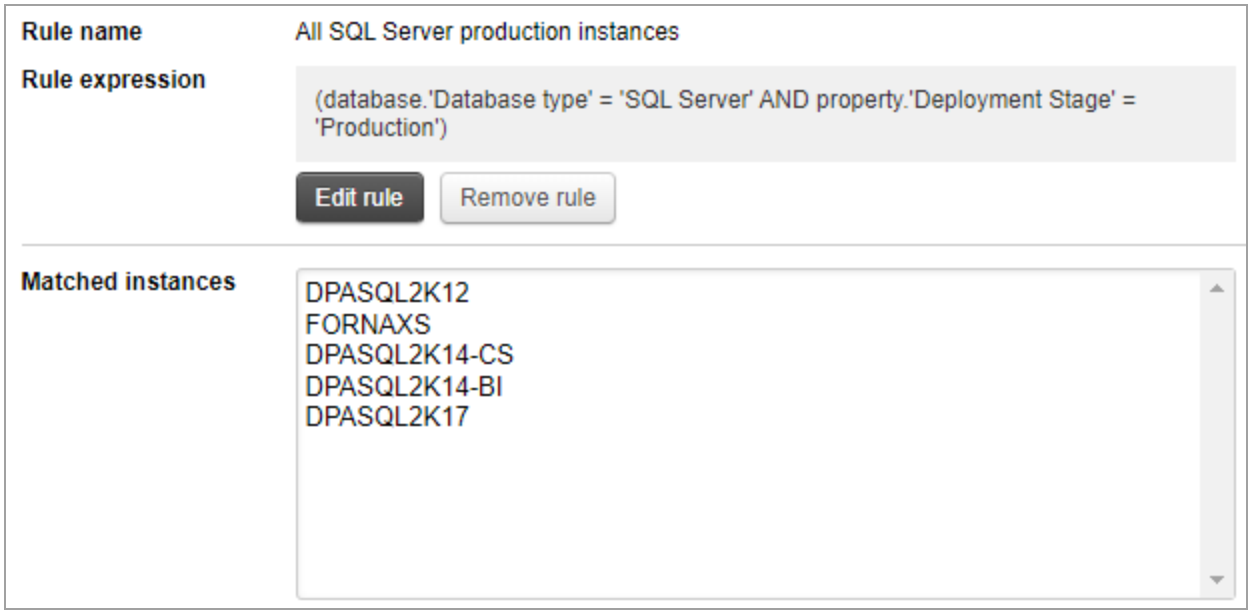
     The Rules page lists the existing rules.

  b. Select an existing rule, or create a new rule and then select it.

  c. Click Assign rule.

     The alert definition shows the selected rule name, rule expression, and a list of

instances that currently meet the rule conditions.



- If you manually select the database instances, the list is static.
  a. Click Select database instances.

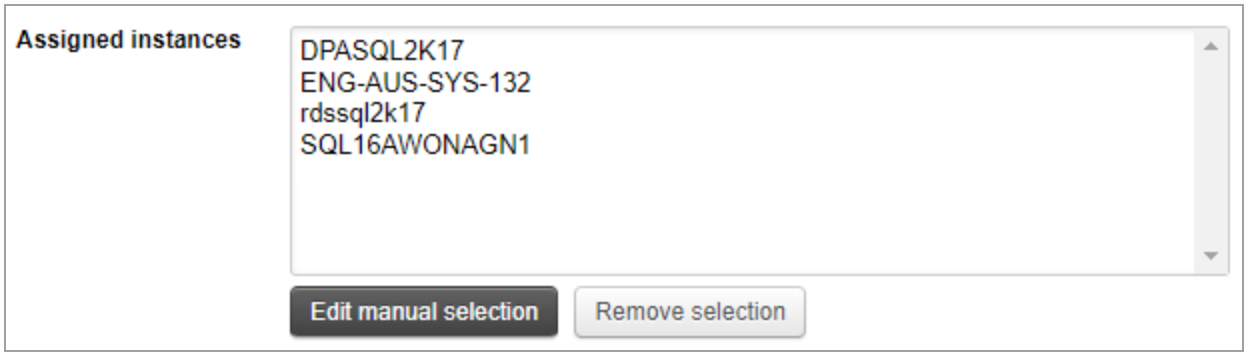
     The Available database instances page lists database instances. If the alert type is specific to one type of database, the page lists only instances of that type.

  b. Use the Search bar to locate instances, or apply filters to refine the list.

  c. To select all instances in the list, click the checkbox above the list. To select individual instances, click the checkbox next to each instance.

  d. Click Assign and go back.

     The alert definition shows the list of selected instances.



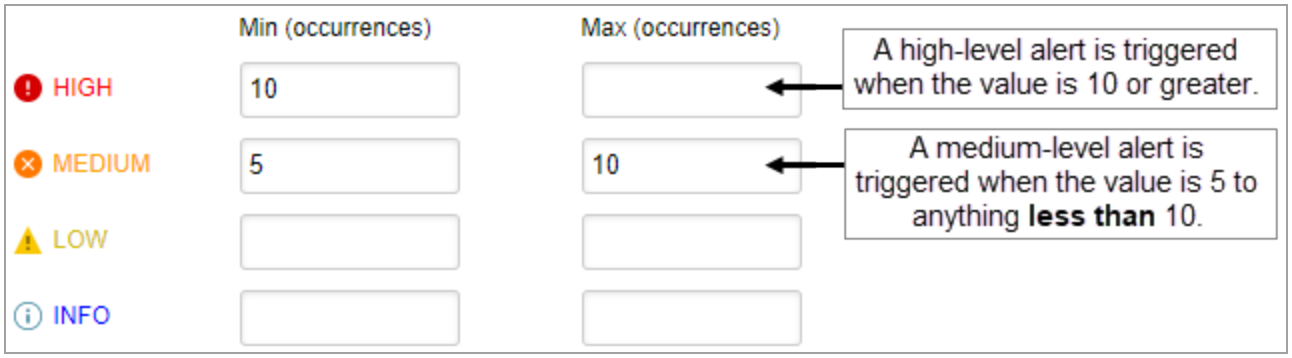
6. If the alert type requires parameters, under Alert Parameters:
   a. Click Search.
   b. If necessary, select a database instance at the top of the Search dialog box.
   c. Enter a search string (for example, part of the SQL statement name or wait type).

      By default, search results for SQL statements are sorted by wait time, with the highest waits first. To list them in alphanumeric order by name or hash value, clear the Order results by wait time checkbox, and then click Search again.

   d. Select a value and click OK.
7. For all Wait Time alerts **except** the Database Instance Wait Time Anomaly alert, specify the thresholds for each alert level you want to enable.

   Alert thresholds for anomalies have default values that can be changed through advanced configuration options.

   - Leave the Max value for the highest level **blank** to alert on anything above the minimum value for that level.
   - If you configure multiple levels, the Max value for lower levels must **equal** the Min value for the next higher level.
   - When you enter a Max value for a level, DPA alerts at that level when the value is **greater than or equal to** the Min value but **less than** the Max level. For example, if the Min value is 5 and the Max value is 10, DPA alerts at that level when the value is 5 or more, but less than 10.



Select the person or group who gets notified when each alert level is triggered and when the alert is broken. (The alert status is set to Broken if an error occurs during execution.)

To send notifications when the alert returns to Normal, select a recipient for Normal.

- If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group and create the contact or group.
- Select an SNMP contact to send SNMP traps when the alert is triggered. Select an SNMP contact for Normal to send a clearing notification when the alert status returns to Normal.

8. Verify or change the notification policy.

   To send notifications when the alert returns to Normal, the notification policy must be `Notify when level changes.`

9. Select the email template that defines the contents of the email notifications sent by this alert.
10. Click Email Preview to see an example of the email that will be generated using the selected email template and contact information.

    If the alert applies to multiple database instances, select an instance in the Email Preview dialog box and click OK. After reviewing the email, you can select a different database instance or click Cancel to close the Email Preview dialog box.

    The email sent to users might not exactly match the preview because some alert parameters cannot be evaluated during a preview.

11. Click Test Alert to test the alert and view the current alert level. The test does not generate an email.
12. Click Save.

Create a DPA Resources alert

Resources alerts are triggered when a resource metric, such as CPU utilization or memory usage, exceeds its threshold. A Resources alert can monitor a single resource metric (such as Buffer Cache Hit Ratio) or all metrics in a resource category (such as Memory).

Task 1: Verify or set resource thresholds

Thresholds for Resources alerts are specified in the resource settings, **not** within the alert. Some metrics have default thresholds, and other do not. Before you create an alert that is triggered when one or more resource metric thresholds are exceeded, make sure that:

- Each metric has a threshold.
- The threshold is appropriate for your environment.

See View or change DPA resource metric thresholds.

Task 2: Create a Resources alert

1. From the DPA menu in the upper-right corner, click Alerts.
2. In the upper-right corner, click Manage alerts.
3. Select Resources as the Alert Category, select the Alert Type, and then click Create Alert.

   > To find out more about each alert type, see DPA alert categories and types. Or select a type to display a description on the right.

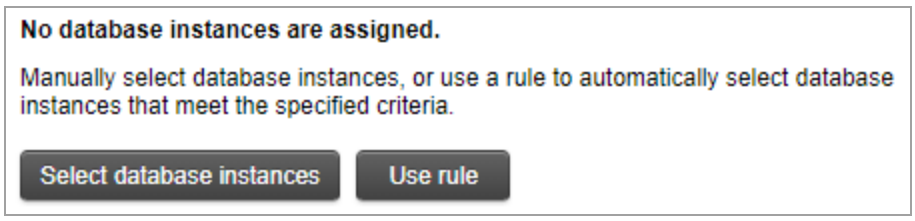
4. In the Alert Information section:
   a. Enter a unique name.
   b. If you want to disable the alert, clear the Active checkbox.
   c. Select the execution interval.

      The execution interval specifies how often the alert runs and the amount of data that DPA examines. For example, if the execution interval is 10 minutes, DPA executes the alert every 10 minutes and examines the last 10 minutes of data to determine whether to trigger the alert. DPA recommends an execution interval of **at least 10 minutes** to prevent unnecessary alerts from a temporary condition.

   d. Enter the notification text to be sent with the email notification. Include an explanation of the issue and the suggested resolution.

      > If you apply a custom email template to this alert, the email notification includes the notification text if the email template contains one of the following variables:
      >
      > - Alert Notification text: `[=alert.notificationText]`
      > - Results: `[=dpa.body]` (included by default)

   Under Database Instances, specify the database instances that the alert applies to. You can use a rule to identify instances that meet one or more conditions, or you can manually select the instances.



5.

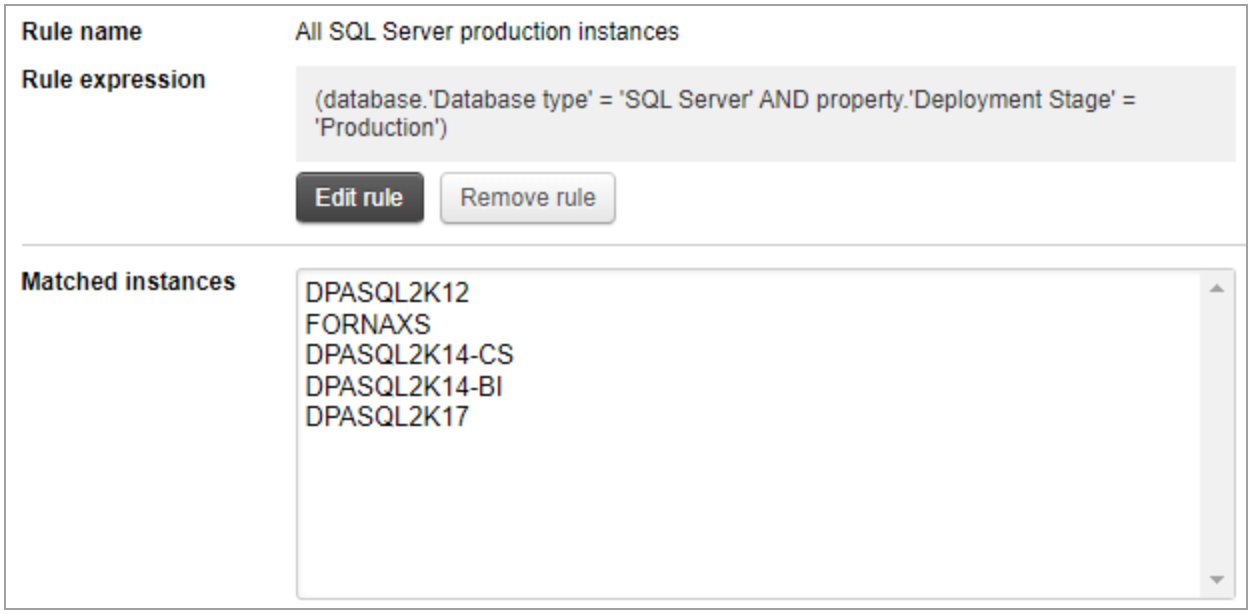
- When you select a rule, DPA uses the rule conditions to determine which instances the alert monitors. If your environment changes, the list of instances is automatically updated.
    a. Click Use rule.

       The Rules page lists the existing rules.

    b. Select an existing rule, or create a new rule and then select it.

    c. Click Assign rule.

       The alert definition shows the selected rule name, rule expression, and a list of instances that currently meet the rule conditions.



- If you manually select the database instances, the list is static.
    a. Click Select database instances.

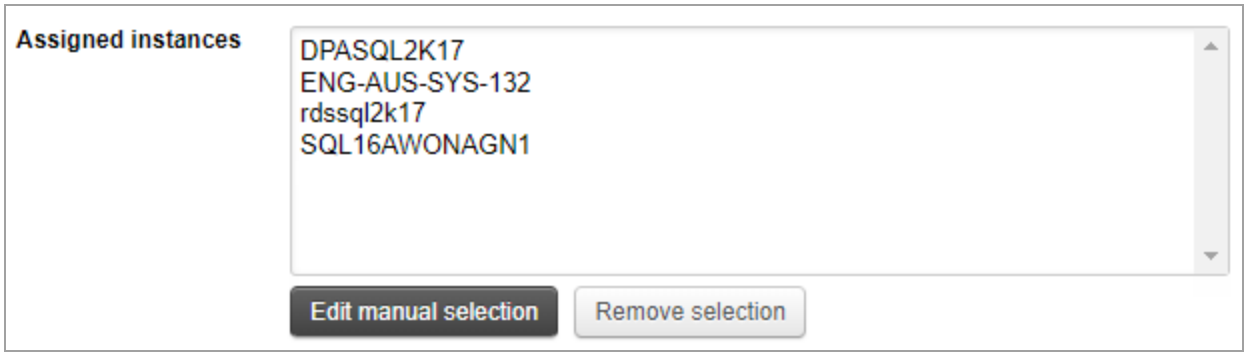
       The Available database instances page lists database instances. If the alert type is specific to one type of database, the page lists only instances of that type.

    b. Use the Search bar to locate instances, or apply filters to refine the list.

    c. To select all instances in the list, click the checkbox above the list. To select individual instances, click the checkbox next to each instance.

    d. Click Assign and go back.

The alert definition shows the list of selected instances.



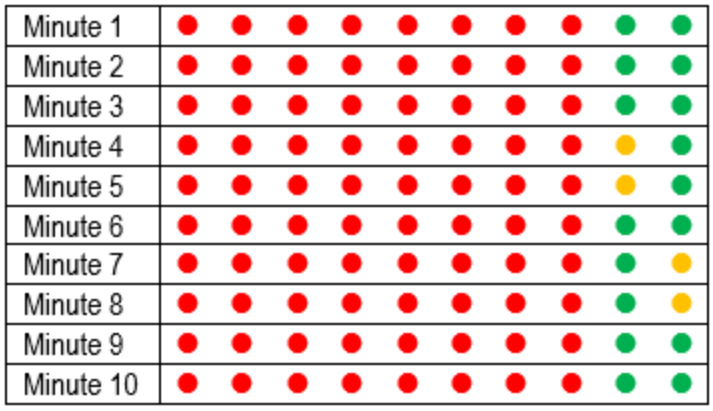
6. Under Alert Parameters:
    a. Select the resource metric(s) that trigger the alert:
        - For alerts against all metrics in a category, select the Category.
        - For alerts against a single metric, select a Category if you want to filter list of metrics. Then select a resource metric from the Resource drop-down menu.

        The Category drop-down menu lists only categories that contain at least one metric with a defined threshold. The Resource drop-down menu lists only resource metrics with defined thresholds. You can set or change thresholds for any resource metric.

    b. Specify the Calculation that is used to determine the alert level for an execution interval.

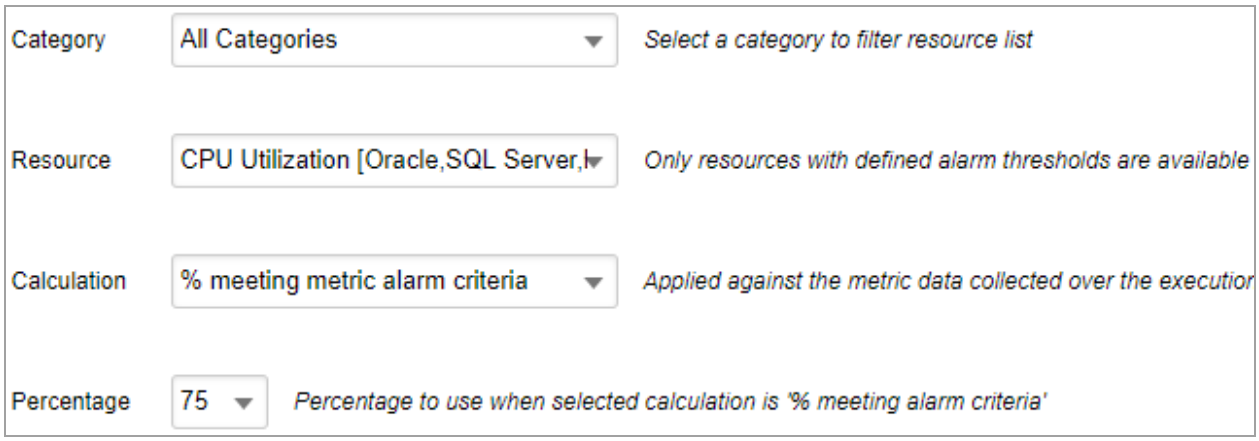
        To determine the alert level, DPA looks at the values collected during an execution interval and applies the specified calculation.

        Example: For a single resource alert, if the metric value is collected once each minute and the execution interval is 10 minutes, DPA looks at the 10 values collected for an interval and applies one of the following calculations.

Calculation	Description
% meeting metric alarm criteria	The alert is triggered when a certain percentage of the values collected during an execution interval meet or exceed the warning or critical threshold for the metric or metrics. Use the Percentage drop-down to specify the percentage. **Example**: The alert is for a category that contains 10 metrics. DPA collects the metric values once each minute, and the alert is set to run once every 10 minutes. Therefore, during each interval, DPA collects 100 values. If the Percentage is 75%: • The alert is triggered at the warning level if 75 or more of these values exceed the warning threshold for the associated metric, but fewer than 75 exceed the critical threshold. • The alert is triggered at the critical level if 75 or more of these values exceed the critical threshold for the associated metric. The following circles represent the values of the 10 metrics collected during the 10-minute alert interval. The red circles indicate that 80 of the 100 alert values exceeded the critical threshold. Because 80% is above the specified Percentage of 75%, the alert is triggered at the critical level. 
Average	DPA uses the average of the values collected during an interval to assign the alert level for that interval.
Median	DPA uses the median value collected during an interval to assign the alert level for that interval.

Calculation	Description
Maximum	DPA uses the maximum value collected during an interval to assign the alert level for that interval.
Minimum	DPA uses the minimum value collected during an interval to assign the alert level for that interval.

c. If the alert is against all metrics in a category **and** you selected % meeting metric alarm criteria, specify the Percentage to use in the calculation.



7. Select the person or group who gets notified when each alert level is triggered and when the alert is broken. (The alert status is set to Broken if an error occurs during execution.)

   To send notifications when the alert returns to Normal, select a recipient for Normal.

   - If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group and create the contact or group.
   - Select an SNMP contact to send SNMP traps when the alert is triggered. Select an SNMP contact for Normal to send a clearing notification when the alert status returns to Normal.

8. Verify or change the notification policy.

   To send notifications when the alert returns to Normal, the notification policy must be `Notify when level changes.`

9. Select the email template that defines the contents of the email notifications sent by this alert.
10. Click Email Preview to see an example of the email that will be generated using the selected email template and contact information.

If the alert applies to multiple database instances, select an instance in the Email Preview dialog box and click OK. After reviewing the email, you can select a different database instance or click Cancel to close the Email Preview dialog box.

> The email sent to users might not exactly match the preview because some alert parameters cannot be evaluated during a preview.

11. Click Test Alert to test the alert and view the current alert level. The test does not generate an email.
12. Click Save.

Create a DPA Administrative alert

Administrative alerts are used to monitor the health of the database system. For example, you can configure an alert that is triggered when a database instance is not accessible or when any database parameter changes.

> Some PostgreSQL administrative alerts can use data from multiple databases. Those alerts are identified here. To enable DPA to collect data from multiple databases in a PostgreSQL cluster, you must configure a foreign data wrapper.

1. From the DPA menu in the upper-right corner, click Alerts.
2. In the upper-right corner, click Manage alerts.
3. Select Administrative as the Alert Category, select the Alert Type, and then click Create Alert.

   > To find out more about each alert type, see DPA alert categories and types. Or select a type to display a description on the right.

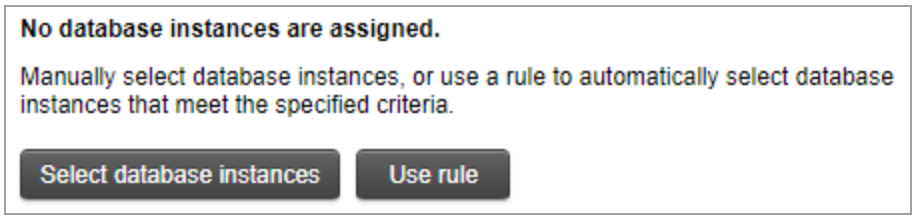
4. In the Alert Information section:
   a. Enter a unique name.
   b. If you want to disable the alert, clear the Active checkbox.
   c. Select the execution interval.

      The execution interval specifies how often the alert runs and the amount of data that DPA examines. For example, if the execution interval is 10 minutes, DPA executes the alert every 10 minutes and examines the last 10 minutes of data to determine whether to trigger the alert. DPA recommends an execution interval of **at least 10 minutes** to prevent unnecessary alerts from a temporary condition.

d. Enter the notification text to be sent with the email notification. Include an explanation of the issue and the suggested resolution.

> If you apply a custom email template to this alert, the email notification includes the notification text if the email template contains one of the following variables:
>
> - Alert Notification text: `[=alert.notificationText]`
> - Results: `[=dpa.body]` (included by default)

Under Database Instances, specify the database instances that the alert applies to. You can use a rule to identify instances that meet one or more conditions, or you can manually select the instances.



5.

- When you select a rule, DPA uses the rule conditions to determine which instances the alert monitors. If your environment changes, the list of instances is automatically updated.

  a. Click Use rule.

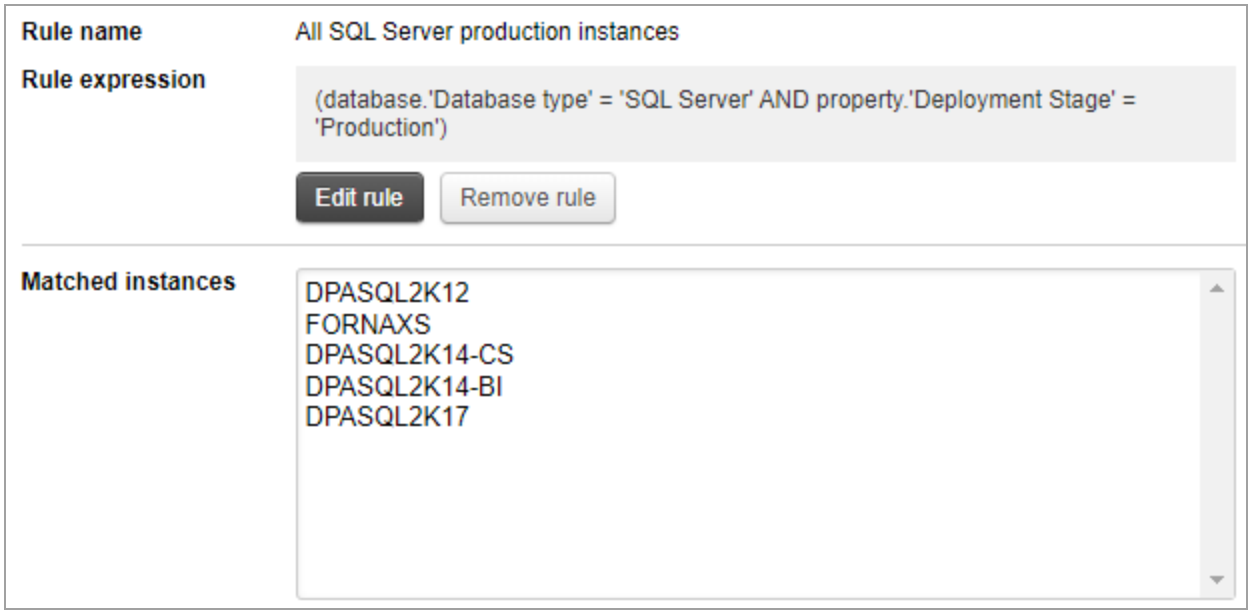
     The Rules page lists the existing rules.

  b. Select an existing rule, or create a new rule and then select it.

  c. Click Assign rule.

     The alert definition shows the selected rule name, rule expression, and a list of

instances that currently meet the rule conditions.



- If you manually select the database instances, the list is static.
  a. Click Select database instances.

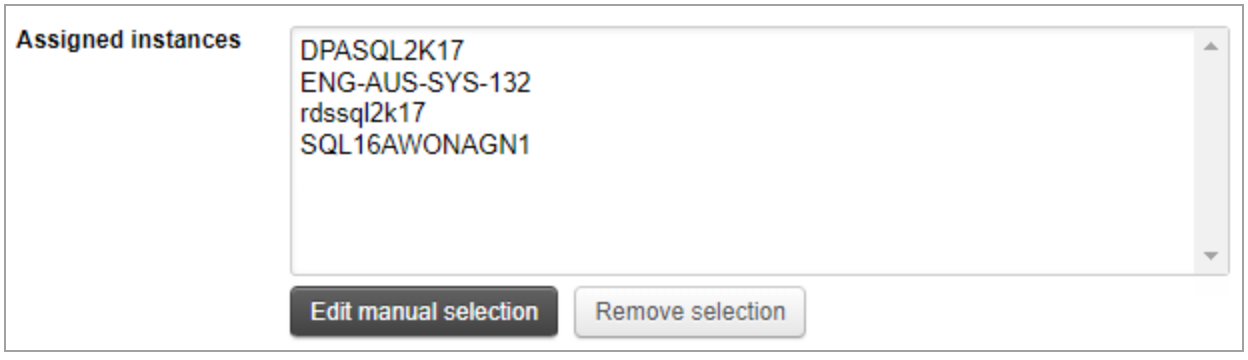
     The Available database instances page lists database instances. If the alert type is specific to one type of database, the page lists only instances of that type.

  b. Use the Search bar to locate instances, or apply filters to refine the list.

  c. To select all instances in the list, click the checkbox above the list. To select individual instances, click the checkbox next to each instance.

  d. Click Assign and go back.

     The alert definition shows the list of selected instances.



6. If any Alert Parameters are required for the alert type, enter the required value.

For the SQL Server Backup Frequency, SQL Server Recovery Backup Assets Size, and SQL Server Backup Time Allotted alerts, select an Evaluation Option if you do **not** want DPA to include all previously backed up databases when it evaluates the alert. (For example, you might want to exclude databases used for testing that do not need regular backups.)

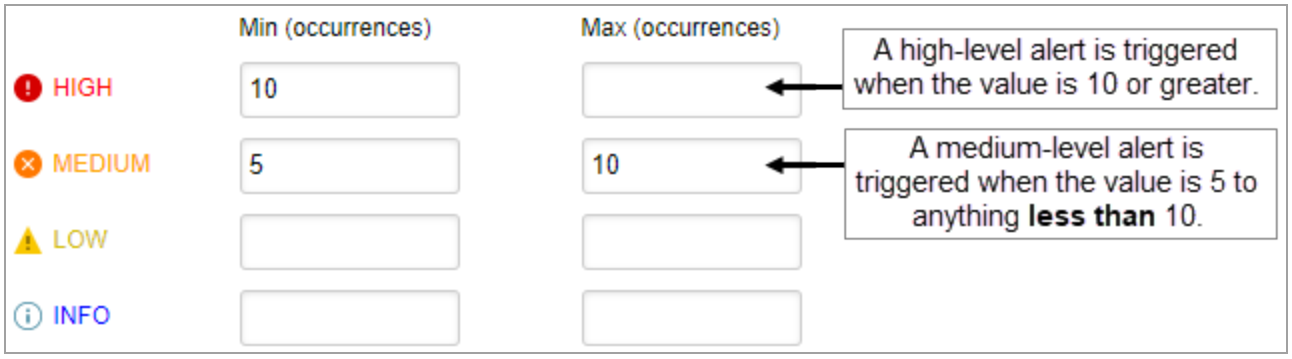
- If you select Use TRACK_BACKUPS_FOR_DBS, set the `TRACK_BACKUPS_FOR_DBS` advance option for each selected database instance.
- If you select Exclude DBs or Include DBs, enter a comma-separated list of the databases to exclude or include.

If a PostgreSQL Administrative alerts can use data from multiple databases, the Alert Parameters section includes the Schema Name option. Leave the default value of Catalog to collect data from a single database. To include data from multiple databases, specify the schema or schemas that include the data. See Specifying schemas in an alert definition.

7. Specify the thresholds for each alert level you want to enable.

   Some Administrative alerts have only one level.

   - Leave the Max value for the highest level **blank** to alert on anything above the minimum value for that level.
   - If you configure multiple levels, the Max value for lower levels must **equal** the Min value for the next higher level.
   - When you enter a Max value for a level, DPA alerts at that level when the value is **greater than or equal to** the Min value but **less than** the Max level. For example, if the Min value is 5 and the Max value is 10, DPA alerts at that level when the value is 5 or more, but less than 10.



8. Select the person or group who gets notified when each alert level is triggered and when the alert is broken. (The alert status is set to Broken if an error occurs during execution.)

   To send notifications when the alert returns to Normal, select a recipient for Normal.

- If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group and create the contact or group.
- Select an SNMP contact to send SNMP traps when the alert is triggered. Select an SNMP contact for Normal to send a clearing notification when the alert status returns to Normal.

9. Verify or change the notification policy.

   To send notifications when the alert returns to Normal, the notification policy must be `Notify when level changes.`

10. Select the email template that defines the contents of the email notifications sent by this alert.
11. Click Email Preview to see an example of the email that will be generated using the selected email template and contact information.

    If the alert applies to multiple database instances, select an instance in the Email Preview dialog box and click OK. After reviewing the email, you can select a different database instance or click Cancel to close the Email Preview dialog box.

    The email sent to users might not exactly match the preview because some alert parameters cannot be evaluated during a preview.

12. Click Test Alert to test the alert and view the current alert level. The test does not generate an email.
13. Click Save.

Create a DPA Custom alert

Use Custom alerts to execute SQL statements or stored procedures against the monitored database or DPA repository to check for conditions not covered by other DPA alerts. Each SQL statement or procedure returns a number (or set of numbers) that can trigger an alert depending on user-defined thresholds. Custom alerts can be used to alert against a wide variety of conditions. Any parameter that can be returned to DPA using a SQL statement or stored procedure can be used as the basis for a custom alert.

To create a custom alert, see the following sections:

- Custom alert types and expected return values
- Requirements for stored procedures
- Create a Custom alert
- Custom tags

Examples of custom alerts can be found on the DPA content exchange page in THWACK.

Custom alert types and expected return values

Depending on what type of custom alert you select, the SQL statement or stored procedure must return one of the following values.

Alert type	Expected return values
Single Numeric Return	The SQL statement or stored procedure returns a single numeric value. The alert is triggered if the value exceeds the defined High, Medium, Low, and Info thresholds.

Multiple Numeric Return	(SQL statements only.) The SQL statement returns one or more rows of data. Each row contains a string in the first column and a numeric value in the second column. For example, the query could return database names and the amount of free space for each one:
	`DB1  120`
	`DB2  840`
	`DB2  35`
	The alert is triggered if any value exceeds the defined High, Medium, Low, and Info thresholds.

Single Boolean Return	The SQL statement or stored procedure returns a string value of `TRUE` or `FALSE` (not case-sensitive). The alert is triggered if `TRUE` is returned.
Single Alert Status Return	The SQL statement or stored procedure returns a string value that specifies the alert status. Valid values are `NORMAL`, `INFO`, `LOW`, `MEDIUM`, and `HIGH` (not case-sensitive).

Requirements for stored procedures

When you create a custom alert that calls a stored procedure, the stored procedure must include **two** output parameters. These output parameters must be in the following order relative to each other, and no other output parameters can be included:

1. `AlertValue OUT VARCHAR2`

   The value of this parameter must be one of the expected return values for the selected alert type. (For example, if the alert type is Custom Procedure Alert - Single Boolean Return, this output parameter must be `TRUE` or `FALSE`.)

Use the custom tag `#ALERTVALUE#` to include this output parameter.

2. `AlertString OUT VARCHAR2`

   The value of this parameter is a description of the result of the stored procedure.

   Use the custom tag `#ALERTSTRING#` to include this output parameter.

The stored procedure can include any number of input parameters. The input parameters can be interspersed with the output parameters, as long as the output parameters are in the correct order relative to each other. For example:

```
myproc('inputParam1', #ALERTVALUE#, 'inputParam2', #ALERTSTRING#, '#DBLINK#')
```

Create a Custom alert

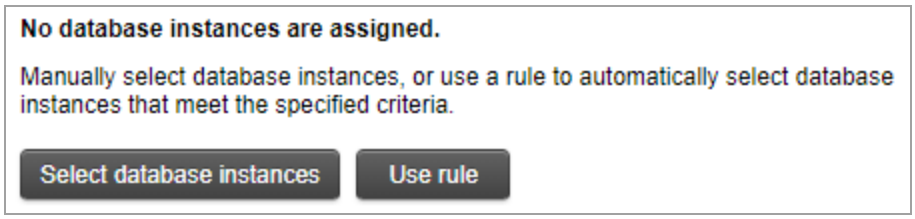
1. From the DPA menu in the upper-right corner, click Alerts.
2. In the upper-right corner, click Manage alerts.
3. Select Custom as the alert category, select the alert type, and then click Create Alert.

   > To find out more about each alert type, see DPA alert categories and types. Or select a type to display a description on the right.

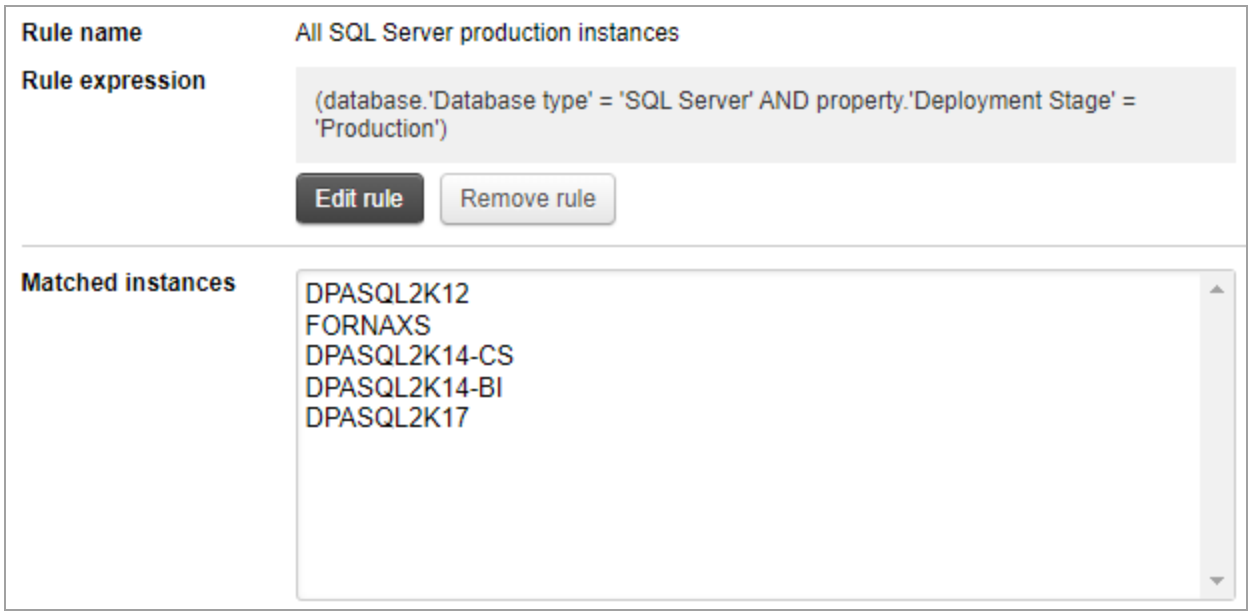
4. In the Alert Information section:
   a. Enter a unique name.
   b. If you want to disable the alert, clear the Active checkbox.
   c. Select the execution interval. (DPA recommends an execution interval of at least 10 minutes.)
   d. Enter the notification text to be sent with the email notification. Include an explanation of the issue and the suggested resolution.

      > If you apply a custom email template to this alert, the email notification includes the notification text if the email template contains one of the following variables:
      >
      > - Alert Notification text: `[=alert.notificationText]`
      > - Results: `[=dpa.body]` (included by default)

5. To run the SQL statement or stored procedure against monitored database instances (instead of the DPA repository), specify the database instances that the alert applies to. You can use a rule to identify instances that meet one or more conditions, or you can manually select the instances.



- When you select a rule, DPA uses the rule conditions to determine which instances the alert monitors. If your environment changes, the list of instances is automatically updated.
  a. Click Use rule.
  b. The Rules page lists the existing rules.
  c. Select an existing rule, or create a new rule and then select it.
  d. Click Assign rule.
  e. The alert definition shows the selected rule name, rule expression, and a list of instances that currently meet the rule conditions.

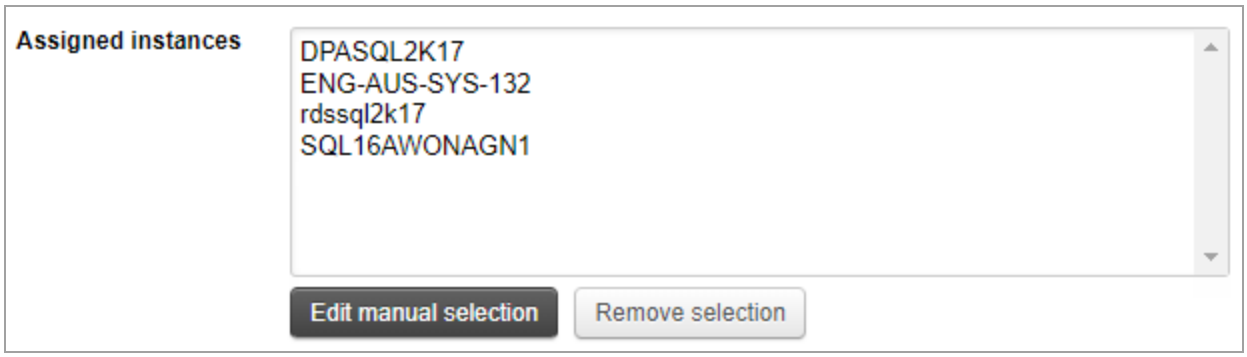


- If you manually select the database instances, the list is static.
  a. Click Select database instances.
  b. The Available database instances page lists database instances. If the alert type is specific to one type of database, the page lists only instances of that type.
  c. Use the Search bar to locate instances, or apply filters to refine the list.

d. To select all instances in the list, select the checkbox above the list. To select individual instances, select the checkbox next to each instance.

e. Click Assign and go back.

f. The alert definition shows the list of selected instances.



6. In the Alert Parameters section:

   a. Enter the SQL statements to execute, or enter a call to a stored procedure.

      Use custom tags to include variables such as the database ID and to include the required output parameters for stored procedures.

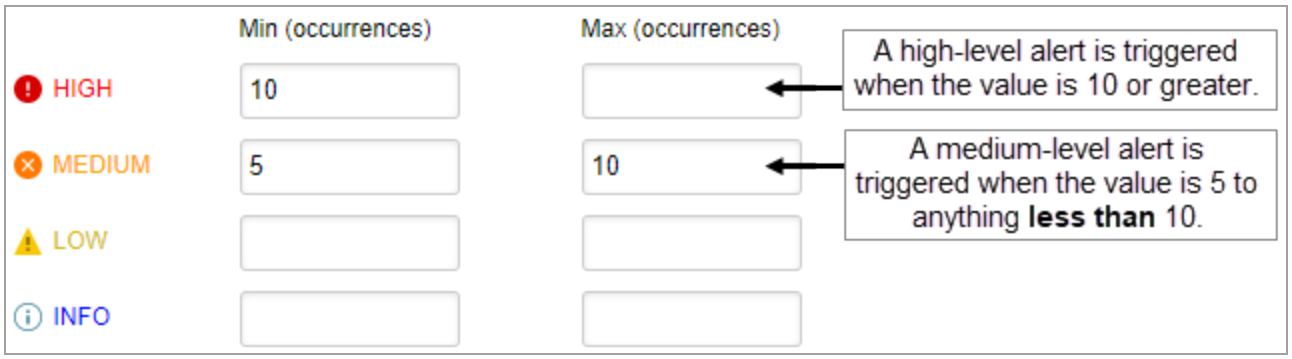
   b. In the Execute Against drop-down, indicate if the SQL statement or stored procedure should be executed against the selected database instances or against the DPA repository database.

   c. If the Description field is available, you can enter a custom description for the alert. This description replaces the DPA default description for the alert type when the Description parameter is included in the email template.

   d. If the alert returns a numeric value, specify the Units for the returned value.

7. If the alert returns a numeric value, specify the thresholds for each alert level you want to enable.

   - Leave the Max value for the highest level **blank** to alert on anything above the minimum value for that level.
   - If you configure multiple levels, the Max value for lower levels must **equal** the Min value for the next higher level.
   - When you enter a Max value for a level, DPA alerts at that level when the value is **greater than or equal to** the Min value but **less than** the Max level. For example, if the Min value is 5 and the Max value is 10, DPA alerts at that level when the value is 5 or more, but less than 10.



8. Select the person or group who gets notified when each alert level is triggered and when the alert is broken. (The alert status is set to Broken if an error occurs during execution.)

   To send notifications when the alert returns to Normal, select a recipient for Normal.

   - If you have not added the person or group as a contact in DPA, click Add Contact or Add Contact Group and create the contact or group.
   - Select an SNMP contact to send SNMP traps when the alert is triggered. Select an SNMP contact for Normal to send a clearing notification when the alert status returns to Normal.

9. Verify or change the notification policy.

   To send notifications when the alert returns to Normal, the notification policy must be `Notify when level changes.`

10. Select the email template that defines the contents of the email notifications sent by this alert.
11. Click Email Preview to see an example of the email that will be generated using the selected email template and contact information.

    If the alert applies to multiple database instances, select an instance in the Email Preview dialog box and click OK. After reviewing the email, you can select a different database instance or click Cancel to close the Email Preview dialog box.

    The email sent to users might not exactly match the preview because some alert parameters cannot be evaluated during a preview.

12. Click Test Alert to test the alert and view the current alert level. The test does not generate an email.
13. Click Save.

Custom tags

You can include the following custom tags in your SQL statements or stored procedure calls. DPA replaces these tags at runtime with the appropriate values.

Tag	Description

`#DBID#`	The internal DPA ID for the monitored database instance.
	• Data type: `VARCHAR2(50)`
	• SQL statement usage example:
	`select mycol from mytable where dbid=#DBID#`
	• Stored procedure usage example:
	`myproc(..., #DBID#, ...)`

`#DBLINK#`	A database link used to connect to an Oracle monitored database.
	• Data type: `VARCHAR2(50)`
	• SQL statement usage example:
	`select mycol from myschema.mytable@#DBLINK#`
	• Stored procedure usage example:
	`myproc(..., '#DBLINK#', ...)`

`#ALERTVALUE#`	(Stored procedures only.) The first required output parameter for stored procedures. It returns one of the expected values based on the alert type. It must appear in the parameter list before `#ALERTSTRING#`. • Data type: `VARCHAR2(500)` • Stored procedure usage example: `myproc(..., #ALERTVALUE#, ..., #ALERTSTRING#)`
`#ALERTSTRING#`	(Stored procedures only.) The second required output parameter for stored procedures. It returns a description of the alert condition. This value is sent as the content or body of the alert. • Data type: `VARCHAR2(4000)` • Stored procedure usage example: `myproc(..., #ALERTVALUE#, ..., #ALERTSTRING#)`

Tag	Description

`#FREQUENCY#`	The execution interval for the alert, in minutes.
	• Data type: `NUMBER`
	• SQL statement usage example:
	`select mycol from myschema.mytable@#DBLINK# where mydate > SYSDATE - (#FREQUENCY#/1440)`
	• Stored procedure usage example:
	`myproc(..., #FREQUENCY#, ...)`

Create and manage rules to determine which database instances are assigned to alerts

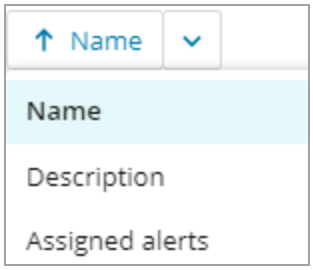
When you are creating or updating an alert definition, you can manually select the assigned database instances, or you can use a rule to identify instances that meet one or more conditions. For example, you could create a rule that assigns all production Oracle database instances owned by the eCommerce Business Unit.

If you choose Use rule, you must select an existing rule or create a new rule that identifies the group of instances you want the alert to monitor. As you add or removed monitored instances, or as the characteristics of the instances change (for example, the version is upgraded or custom property values change), DPA automatically updates the list of instances.

View information about existing rules

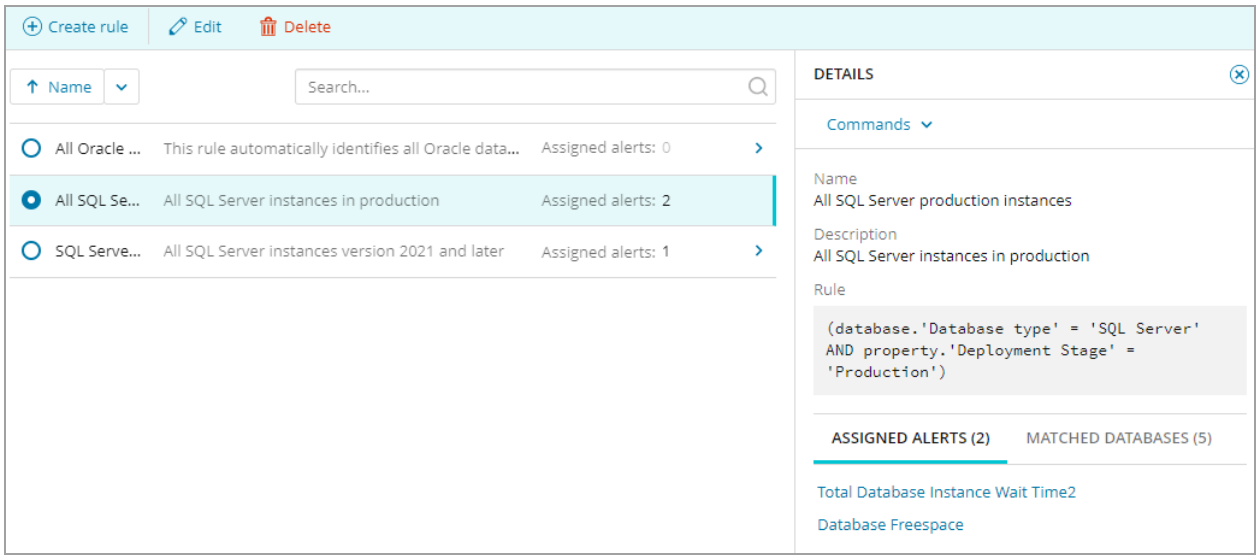
1. Open the list of existing rules:
   - If you are creating an alert definition, click Use rule.
   - If you are **not** creating an alert definition:
     a. From the DPA menu in the upper-right corner, click Options.
     b. Under Monitor Setup > Report, Metrics, & Alerts, click Manage Rules.

   The Rules page shows information about each existing rule, such as the name, description, and number of alerts that use the rule.
2. To find the rule you are interested in:
   - Enter a string in the Search text box to search for terms in a rule name or description.
   - Select an option from the drop-down menu above the list to change the sort order. For example, sort by assigned alerts to locate any unused rules. By default, rules are sorted by name.

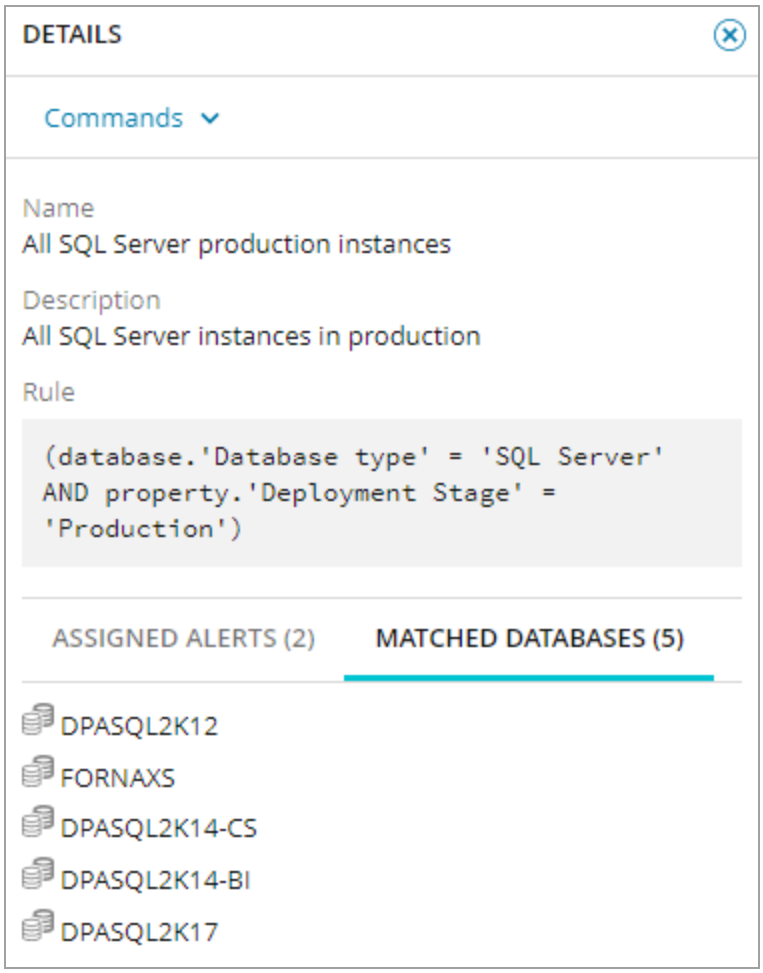


3. To see additional information about a rule, click the > to the right of "Assigned alerts" on that row.

   The Details pane shows information such as the rule definition and the alerts that use the rule.



4. Click Matched Databases to see the database instances that meet the rule conditions.



Create a new rule

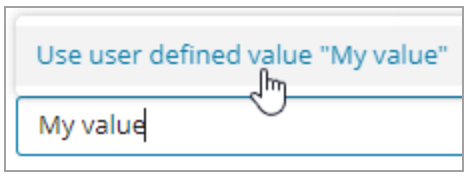
1. Open the list of existing rules.
2. Click Create rule.

   The Rule builder page opens.
3. Enter a unique rule name and, optionally, a description of what instances this rule assigns.

4. Under Assign database instance, add the first condition:
    a. Click ⊕.
    b. Select a database property or a custom property.
    c. Select an operator. Available operators include:

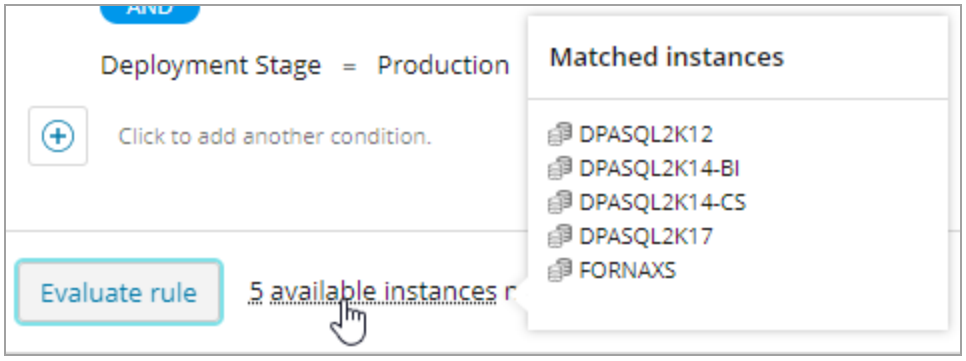
contains	The selected property must contain the specified value.
does not contain	The selected property must not contain the specified value.
=	The selected property must exactly match the specified value.
!=	The selected property must not match the specified value.
<=	(Numeric properties only.) The selected property must be less than or equal to the specified value.
>=	(Numeric properties only.) The selected property must be greater than or equal to the specified value.
<	(Numeric properties only.) The selected property must be less than the specified value.
>	(Numeric properties only.) The selected property must be greater than the specified value.

    d. Select a value from the drop-down menu, or type a value and click Use user defined value.



    e. Click Add condition.

5. Add other conditions as needed. See the following sections for more information:
   - Change the AND operator to an OR operator
   - Group or ungroup conditions
   - Rearrange conditions
   - Edit a condition
   - Remove a condition
6. To see how many instances that meet the rule conditions, click Evaluate rule. Hold the mouse pointer over the number to display the list of instances.

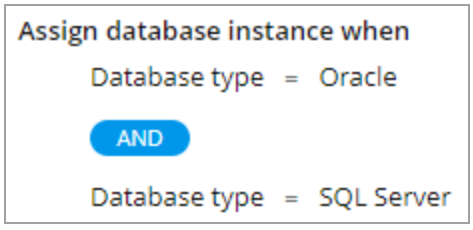


7. Click Save to save the rule and return to the Rules page.

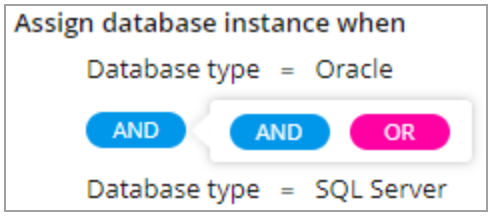
Change the AND operator to an OR operator

When you add an additional condition, the Boolean operator joining the conditions is AND by default. To use an OR operator:

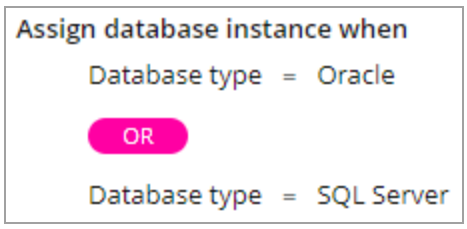
1. Add an additional condition. The connecting operator is AND.



2. Click the AND operator. The available operators are displayed to the right.

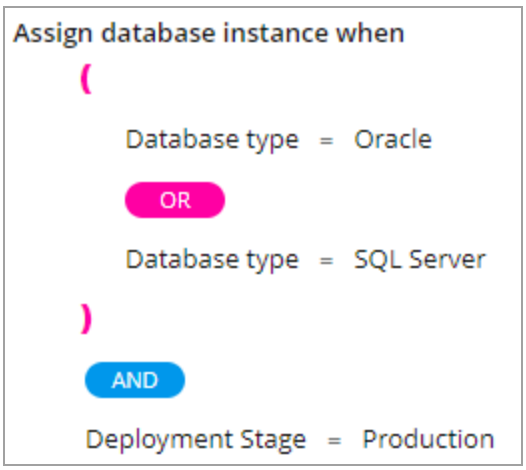


3. Click the OR operator. The connecting operator changes to OR.



Group and ungroup conditions

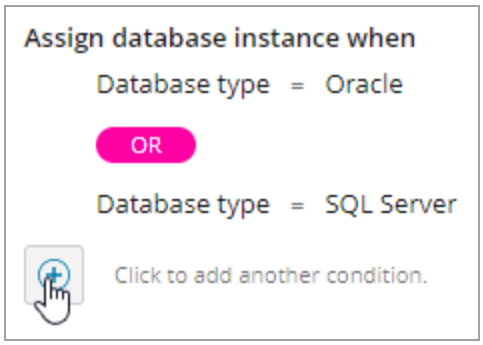
Grouped conditions are surrounded by parentheses, and they are evaluated together. For example, the following rule would identify instances that 1) are either Oracle or SQL Server and 2) have a value of Production for the custom property Deployment Stage.



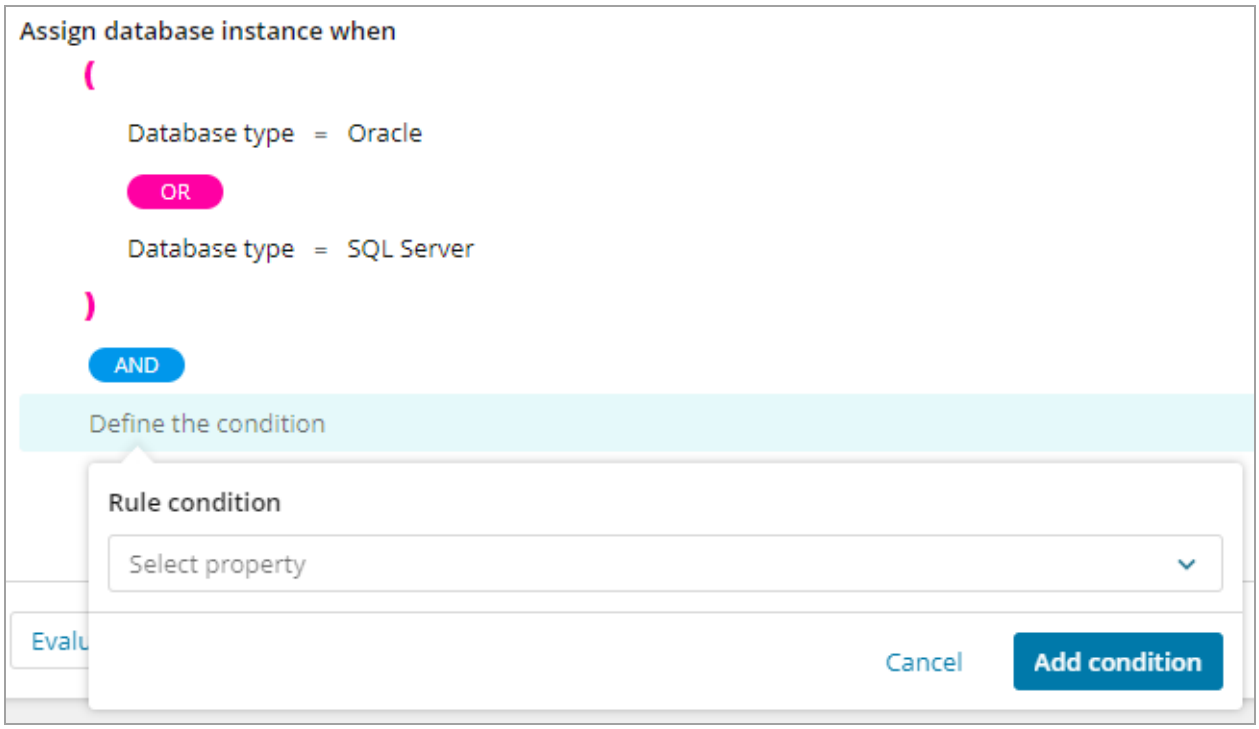
Group conditions

You can group conditions in either of the following ways:

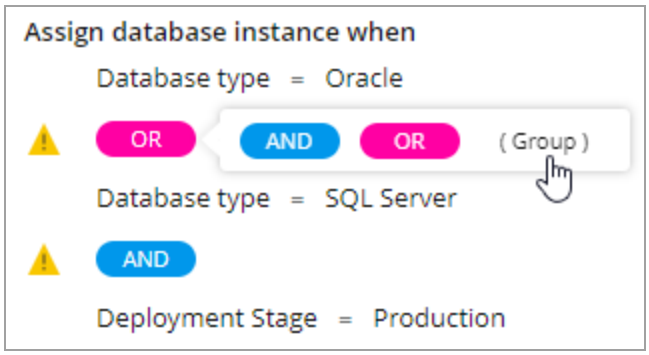
- If you have two or more conditions, you can automatically group all existing conditions when you add a new condition. Click the ⊕ below the existing conditions.



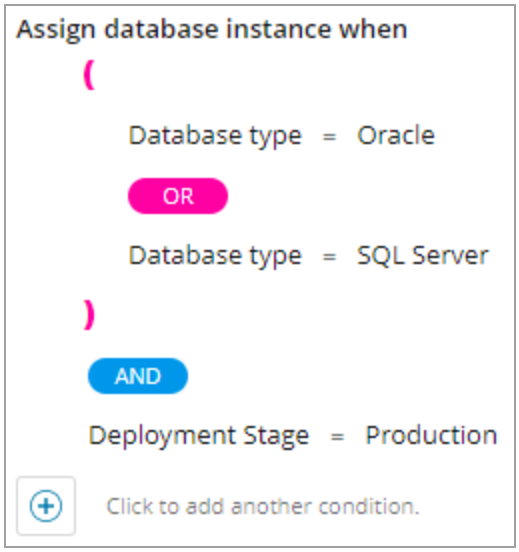
  DPA places parentheses around all existing conditions to group them, and the Rule condition dialog box opens.



- To group a pair of existing conditions **without** adding a new condition:
  1. Click the operator between the conditions you want to group. Options are displayed on the right.



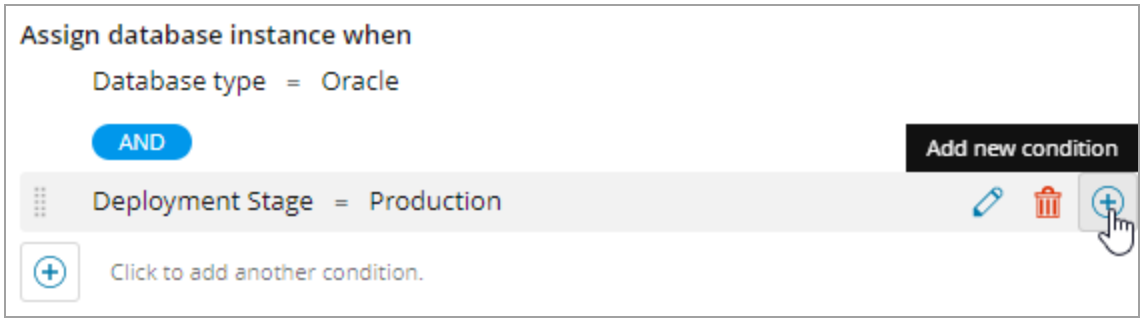
2. Click (Group). DPA places parentheses around the conditions to group them.



Add a condition without grouping existing conditions

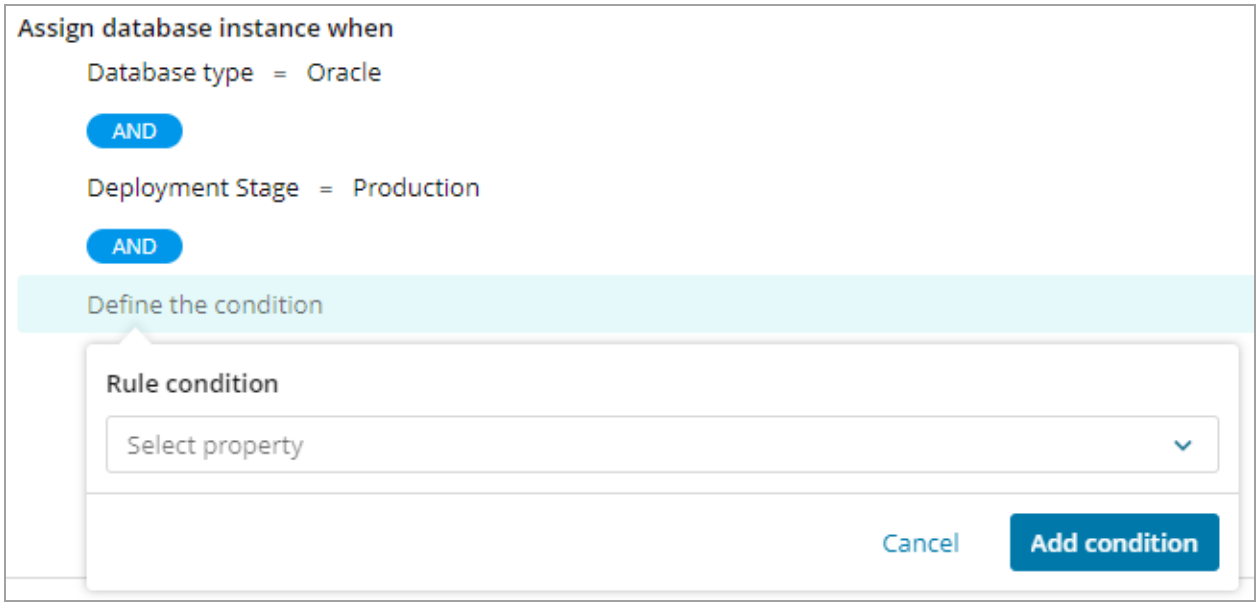
If you have two or more conditions, complete the following steps to add another condition **without** grouping the existing conditions.

1. Hold the mouse pointer over any condition.
2. Click the ⊕ to the right of the condition.



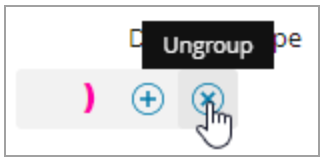
The new condition is added below the selected condition, and the existing conditions are not

grouped.



Ungroup conditions

1. Hold the mouse pointer over either parenthesis.
2. Click the ⊗ to the right of the parenthesis.



Rearrange conditions

You can change the order of the conditions or move a condition into or out of a group.

1. Hold the mouse pointer over a condition.
2. Click the ⋮⋮ and drag the condition to a different location. As you drag the condition, blue lines indicate where you can drop it.

Edit a condition

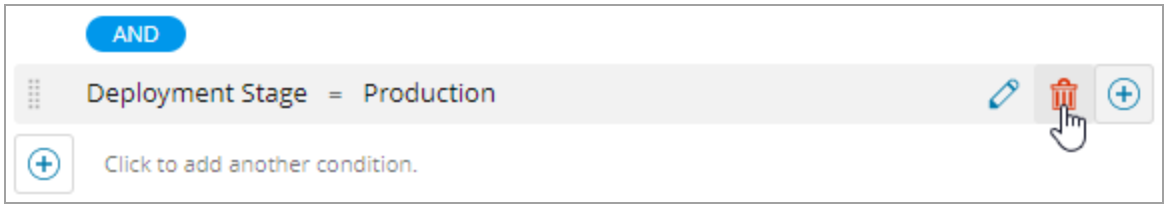
1. Hold the mouse pointer over the condition.
2. Click the pencil icon to the right of the condition.

The Rule condition dialog box opens.

3. Change one or more values, and click Edit condition.

Remove a condition

1. Hold the mouse pointer over the condition.
2. Click the trash can icon to the right of the condition.



The condition is removed. If all remaining conditions were grouped, the grouping is removed.

Edit a rule

1. Open the list of existing rules.
2. Select the rule, and click Edit. Or from the Details pane, click Commands > Edit rule.
3. Change any fields. See the previous sections for more information.
4. Click Save to save the rule and return to the Rules page.

Delete a rule

1. Open the list of existing rules.
2. Look at the Assigned alerts value. If any alerts use this rule, open the Details pane and note which ones use it so that you can update the alert definitions.
3. Select the rule, and click Delete. Or from the Details pane, click Commands > Delete.

   A confirmation prompt warns you that the rule will be removed from any alert definitions that use it.
4. Click Delete at the confirmation prompt.

The rule is deleted and removed from any alert definitions. The alert definitions now have no instances assigned to them. Edit the alert definitions to choose a different rule or manually assign instances.

Configure a foreign data wrapper to collect data from multiple PostgreSQL databases

Some PostgreSQL Administrative alerts can be configured to collect data from multiple databases. The default PostgreSQL implementation does not support cross-database queries. If you want an alert to be triggered based on data from multiple databases, you must configure a foreign data wrapper. A foreign data wrapper is an extension available in PostgreSQL that allows you to access data from multiple databases.

For more information about foreign data wrappers, see the PostgreSQL documentation.

DPA provides scripts to help you implement foreign data wrappers.

1. Download fdw_scripts.zip and extract the contents on the database server. This file contains the following scripts for Linux and Windows operating systems, respectively:
   - fdw_linux.sh
   - fdw_win.bat
2. Decide on the value of the parameters. The following parameters are required for both Linux and Windows:

Parameter	Description
v_dbcount	Use `1` as the value.
v_dbname	Specify the name of the database for which the foreign data wrapper is being created.
v_port	Specify the PostgreSQL server port. The default port is 5432.
v_hostip	Specify the host IP address.
v_username	Specify the PostgreSQL user for which the foreign data wrapper is mapped by default. The superuser `postgres` can be used.
v_userpwd	Specify the password for the user in the previous parameter.
v_schemaname	Specify the name of the schema to which foreign views will be imported. Use a unique schema for each foreign data wrapper.database.

Parameter	Description
v_servername	Specify the server name where the server is created. Use a unique server name for each foreign data wrapper.database.
v_psqlpath	Specify the PostgreSQL path.

3. Execute the appropriate script from a command prompt on the database server, specifying the values for the parameters above. For example:
   - For Linux:

```
./fdw_linux.sh 1 mytest 5432 10.199.8.32 postgres root123
foreignschema1 foreign_server1 /var/lib/bin
```

   - For Windows:

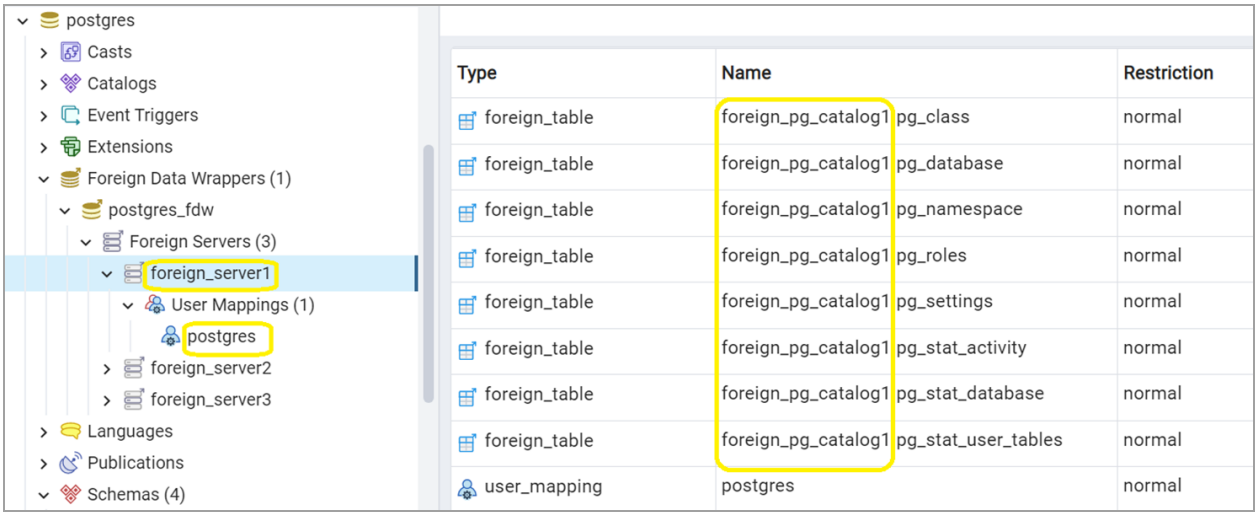
```
fdw_win.bat 1 mytest 5432 10.199.8.32 postgres root123
foreignschema1 foreign_server1 "C:\Program Files\PostgreSQL\14\bin"
```

Specifying schemas in an alert definition

When you create an alert and you want to include data from multiple databases, you will specify which schemas to get the data from. The available schemas can be seen in the client console under Foreign Data Wrappers. For example, let's say the following command was used to create the foreign data wrapper, with `foreign_pg_catalog1` specified as the schema name:

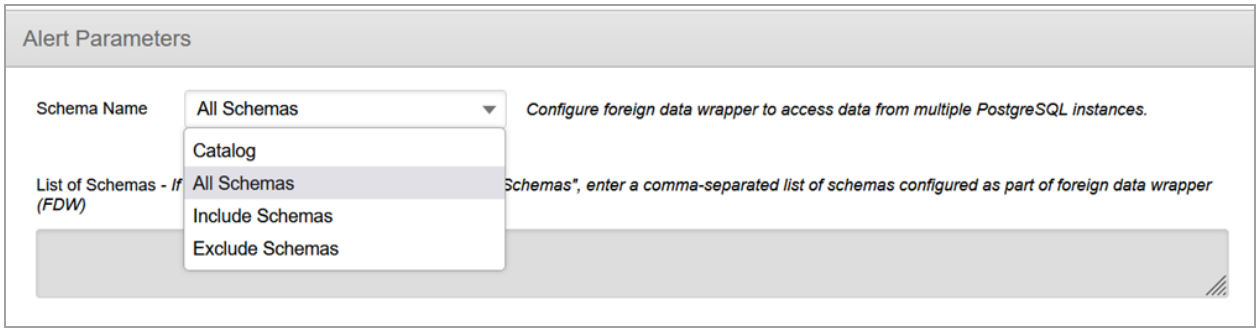
`fdw_win.bat 1 foreign_db1 5432 10.199.8.22 postgres admin` **`foreign_pg_catalog1`** `foreign_server1 "C:\Program Files\PostgreSQL\14\bin"`

This schema can be seen in the console here:

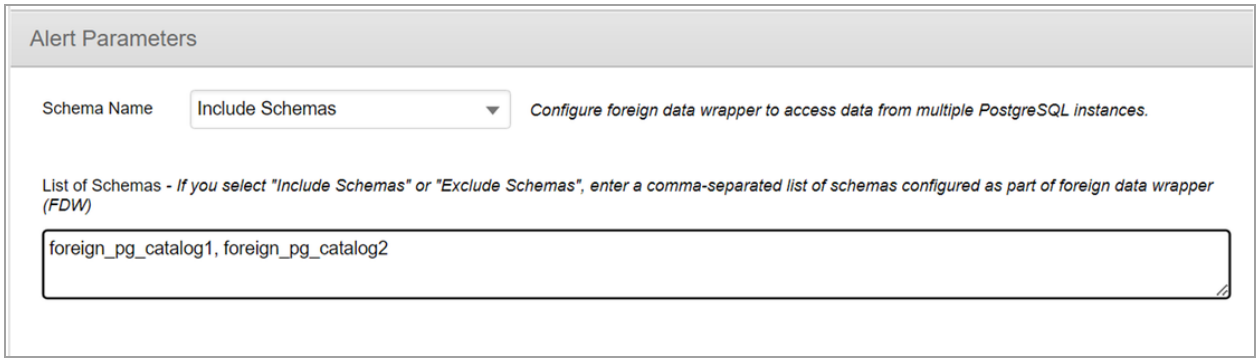


If an alert can use data from multiple databases, the Alert Parameters section includes the Schema Name option. To define which schemas to use:

1. Select an option from the Schema Name drop-down menu:
    - Catalog: Uses data from a single database, and does not require a foreign data wrapper
    - All Schemas: Uses data from all existing schemas
    - Include Schemas: Uses data from the schemas you specify
    - Exclude Schemas: Uses data from all existing schemas **except** the ones you specify



2. If you selected Include Schemas or Exclude Schemas, enter the names of the schemas to include or exclude.



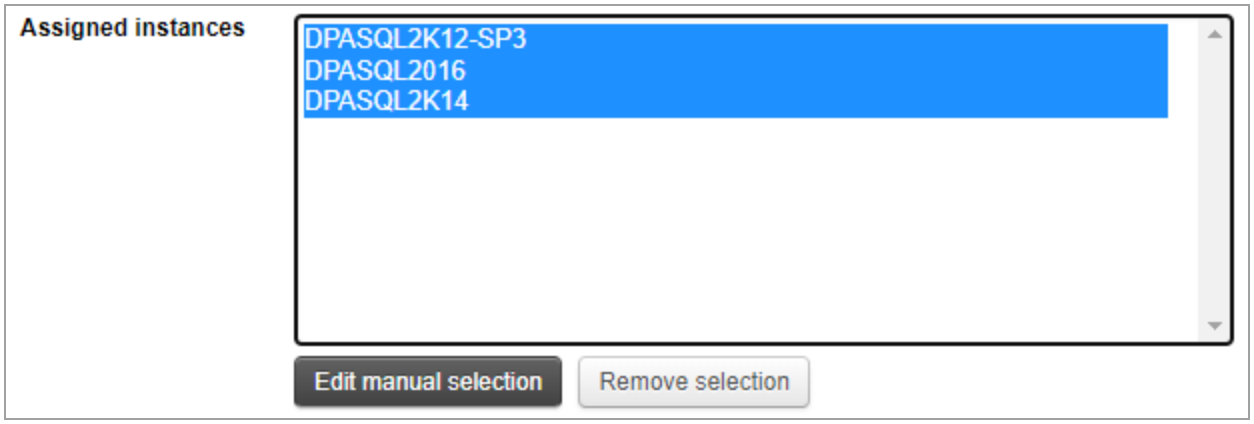
The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Edit the definition of an existing DPA alert

You can edit an existing alert definition to change anything **except** the alert type and description.

1. Open the alert definition:
   - From **anywhere** in DPA:
     a. From the DPA menu in the upper-right corner, click Alerts.
     b. In the upper-right corner, click Manage alerts.

        The Manage Alerts page lists all existing alert definitions.
     c. Click the name of the alert definition you want to edit.
   - From the Alert details page of the alert you want to edit:
     a. Click the Commands menu in the upper-right corner.
     b. Choose Edit alert definition.
2. To change the assigned instances, do either of the following:
   - If the instances were manually selected, click the Edit manual selection button to change the manual selection. Or complete the following steps to remove the manually selected instances and use a rule instead:
     a. In the Assigned instances box, select all instances.



     b. Click Remove selection.

        The option to use a rule becomes available.

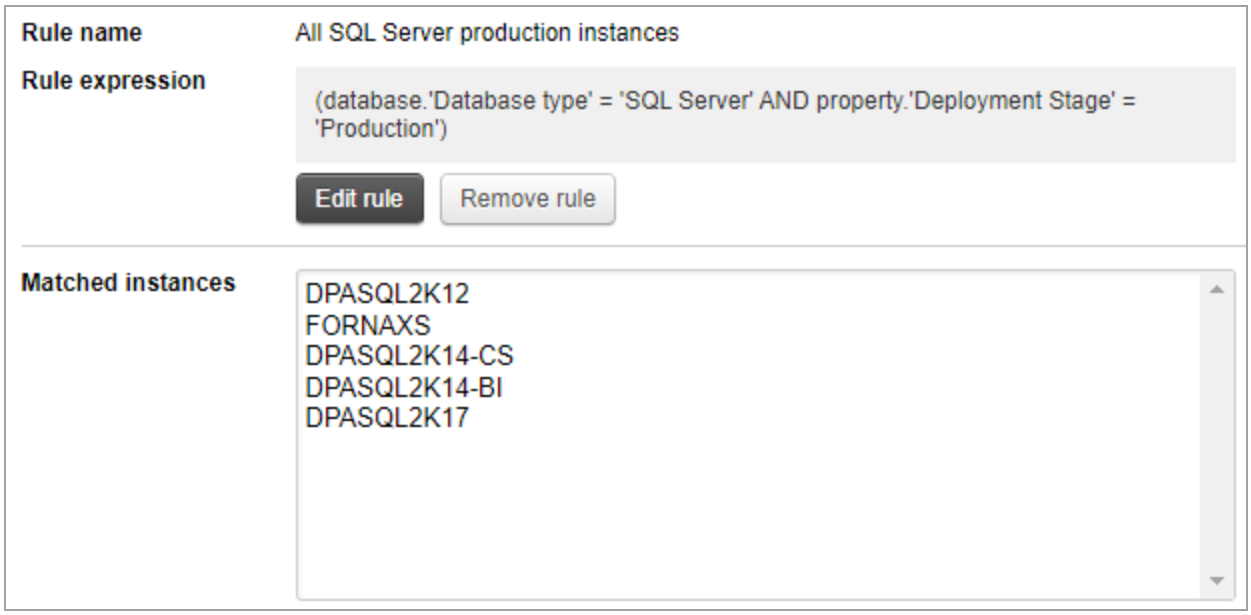
     c. Click Use auto-assignment rule.

The Rules page lists the existing rules.

d. Select an existing rule, or create a new rule and then select it.

e. Click Assign rule.

The alert definition shows the selected rule name, rule expression, and a list of instances that currently meet the rule conditions.



- If the instances were selected using a rule:
  - To edit the rule, click Edit rule.
  - To use a different rule or manually select instances, click Remove rule.

3. Edit any other fields you want to change. For more information about the available fields, see one of the following topics:
   - Create a DPA Wait Time alert
   - Create a DPA Resources alert
   - Create a DPA Administrative alert
   - Create a DPA Custom alert

Stop DPA alerts for a period of time

To stop alerting for a period of time, create an alert blackout. For example, you can create an alert blackout to suppress alerts during a maintenance window.

Open the Alert Blackouts tab

To open the Alert Blackouts tab from **anywhere** in DPA:

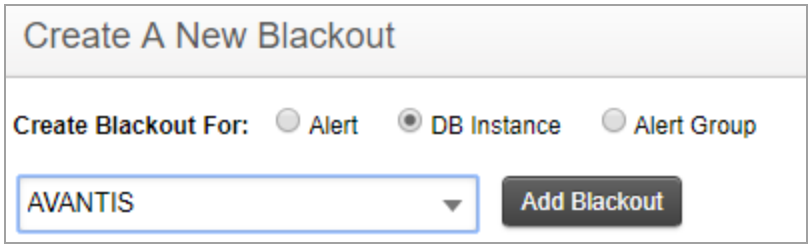
1. From the DPA menu in the upper-right corner, click Alerts.
2. In the upper-right corner, click Manage alerts.
3. Click the Alert Blackouts tab.

If you opened the Alert details page to view information about the alert, you can open the Alert Blackouts tab from there:

1. Click the Commands menu in the upper-right corner.
2. Choose Blackouts.

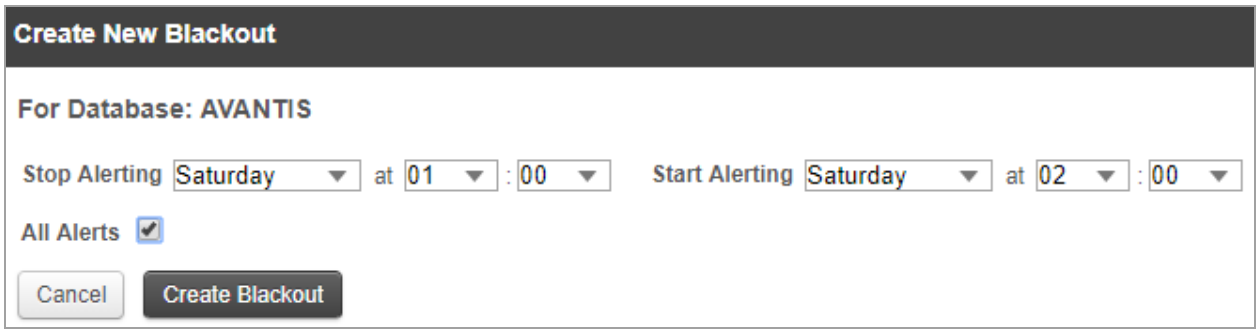
Create an alert blackout

1. Open the Alert Blackouts tab.
2. Specify whether you want to create a blackout for an alert, a database instance, or an alert group.
3. Select the alert, database instance, or alert group, and then click Add Blackout.



4. Specify the beginning and end of the blackout period. Times are based on a 24-hour clock.
5. If the blackout period is for an alert, select All Databases or specify the database instances that this alert should not run on.

   If the blackout period is for a database instance, select All Alerts or specify the alerts that should not run.



6. Click Create Blackout.

The blackout is effective each week on the specified day and time until it is deleted.

Edit an alert blackout

1. Open the Alert Blackouts tab.
2. Under Existing Blackout Periods, locate the blackout period and click Edit.
3. Update the schedule and the affected alerts or database instances.

   > You cannot change the original target of the blackout period. For example, if the blackout period was created to suppress all alerts on a database instance, you cannot change it to suppress a specific alert on multiple database instances. To make that type of change, delete the blackout period and create a new one.

4. Click Save Blackout.

Delete an alert blackout

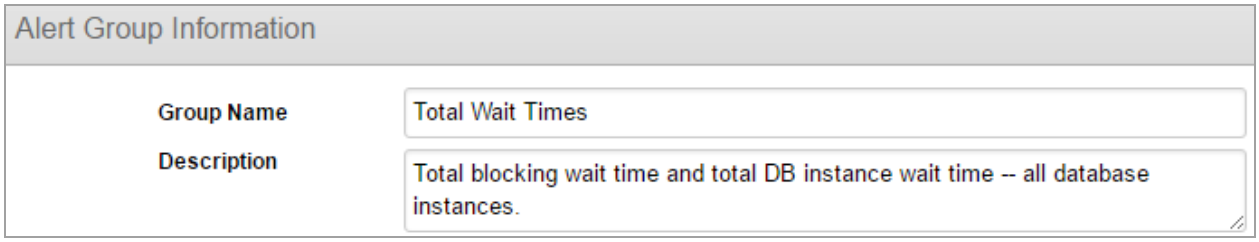
1. Open the Alert Blackouts tab.
2. Under Existing Blackout Periods, locate the blackout period and click Delete.
3. At the confirmation prompt, click Yes.

Create a DPA alert group

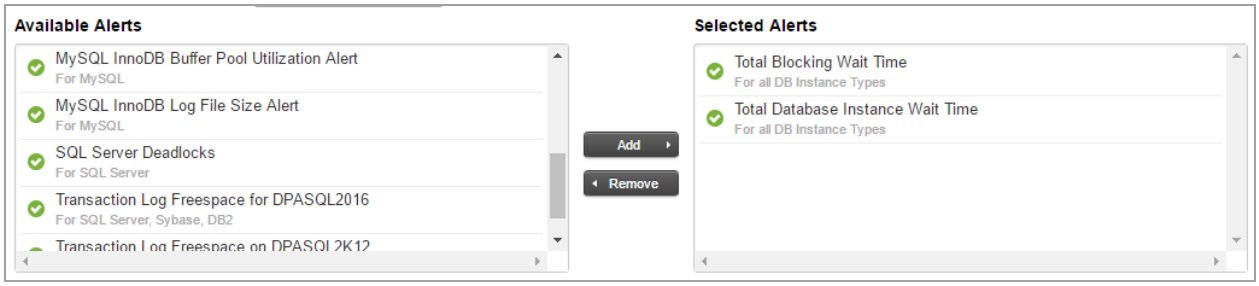
An alert group defines a set of alerts to be run against a set of database instances. Alert groups simplify alert configuration and help make alerting more consistent across the monitored database instances. When you add alerts to an alert group, you do not have to select database instances within each alert definition. Instead, you select the database instances just once for the entire group. If the list of instances changes, you can update it in only one place.

> If you select database instances within an alert definition and you select other database instances for that alert within an alert group, **all** selected databases will trigger the alert.

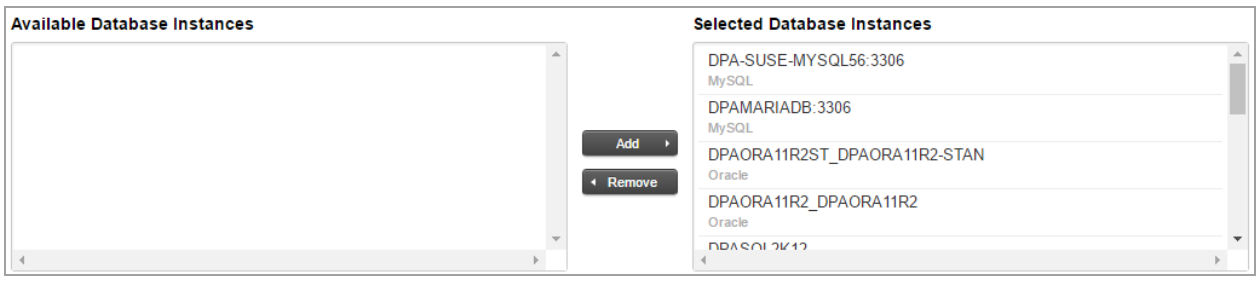
1. Open the Alert Groups tab:
   - From **anywhere** in DPA:
     a. From the DPA menu in the upper-right corner, click Alerts.
     b. In the upper-right corner, click Manage alerts.
     c. Click the Alert Groups tab.
   - From the Alert details page:
     a. Click the Commands menu in the upper-right corner.
     b. Choose Groups.
2. Click Create Alert Group.
3. Enter a unique name and a brief description.



4. Select the alerts to include in this group.



5. Select the database instances on which to execute these alerts.



6. Click Save.

Notification policy for DPA alerts

When you create an alert in DPA, you can accept the default notification policy or apply a different policy to that alert. The notification policy determines when notifications about the alert are sent. The following sections describe each policy.

When an alert level changes, DPA sends a notification only if a recipient is selected for that level in the alert definition. For example, if the notification policy is `Notify when level changes`, you must specify a recipient for Normal if you want DPA to send a notification when the alert level returns to Normal. The examples in the following tables assume that recipients are specified for all alert levels.

Notify when level not visited since normal

A notification is sent if the alert status is not Normal and the alert has not been in this status since the last time the status was Normal. If the alert returns the same status for multiple polling periods without returning to Normal, you are notified only once for each status. For example:

Execution Interval	Alert Level	Notification Sent?
1	Normal	No
2	Medium	Yes
3	High	Yes
4	High	No (this alert level was returned previously)
5	Medium	No (this alert level was returned previously)
6	Low	Yes
7	Normal	No
8	Low	Yes

This is DPA's default notification policy. A DPA administrator can change the default policy for your DPA deployment by setting the advanced option `ALERT_NOTIFICATION_TRIGGER`.

Notify when level changes

A notification is sent if the alert status has changed since the previous execution interval, even if the change is that it returned to Normal. You are notified each time the level changes, but only once for each change. For example:

Execution Interval	Alert Level	Notification Sent?
1	Normal	No (the alert was not triggered during the previous execution interval)
2	Medium	Yes
3	High	Yes
4	High	No (the alert status has not changed)
5	Medium	Yes
6	Low	Yes
7	Normal	Yes
8	Low	Yes

Notify when level is not normal

A notification is sent if the alert status is not Normal, regardless of the alert's previous status. For example:

Execution Interval	Alert Level	Notification Sent?
1	Normal	No
2	Medium	Yes
3	High	Yes
4	High	Yes
5	Medium	Yes
6	Low	Yes
7	Normal	No
8	Low	Yes

Send DPA alert notifications to a third-party notification service through SolarWinds Observability

Some organizations use third-party notification services, such as PagerDuty and ServiceNow, to view alerts from multiple monitoring tools in one place. When you connect DPA to SolarWinds Observability with Platform Connect, you can choose to send DPA alert notifications to notification services that are configured in SolarWinds Observability. Currently, DPA supports sending alert notifications to PagerDuty and ServiceNow.

See the following sections:

How does Platform Connect secure the information communicated from DPA to SolarWinds Observability?

For data in transit from DPA to SolarWinds Observability, DPA encrypts the data using the TLS 1.2 protocol with a 2048-bit RSA certificate.

Requirements

- DPA users with subscription licenses are eligible to connect to SolarWinds Observability. Users doing a 14-day evaluation and users with temporary licenses (for a fixed term) are also eligible.
- You must have connectivity between your DPA server and your cloud service provider for communication with SolarWinds Observability.
- When DPA is connected to SolarWinds Observability, an agent is installed on the DPA server. During this installation, the `swagent` user is created. The `swagent` user requires root privileges.

Known limitations

- Platform Connect is not supported on a server where both DPA and SQL Sentry are installed.
- If CyberArk vault is being used, configuring DPA to send ServiceNow and PagerDuty notifications using Platform Connect is not supported.
- If you have an evaluation license for DPA, a SolarWinds Observability trial account is created for you when you connect to SolarWinds Observability. You can use this trial account for 30 days. If you change your DPA license from an evaluation license to a temporary or subscription license, you must contact Sales or Support to extend the SolarWinds Observability trial account.
- If you install DPA with Platform Connect on a DPA AWS AMI server, a SolarWinds Observability trial account is created that you can use for 30 days. If you want to extend the SolarWinds Observability trial account, contact Sales or Support.
- If you have used Platform Connect and you want to uninstall DPA, you must first uninstall the SolarWinds Observability Agent.
- If the `repo.properties` file in the DPA installation directory is updated, you must copy and paste the updated version to the following folder on the DPA server:
  - Linux: `/opt/solarwinds/uamsclient/dpa/`
  - Windows: `C:\ProgramData\SolarWinds\UAMSClient\dpa\`

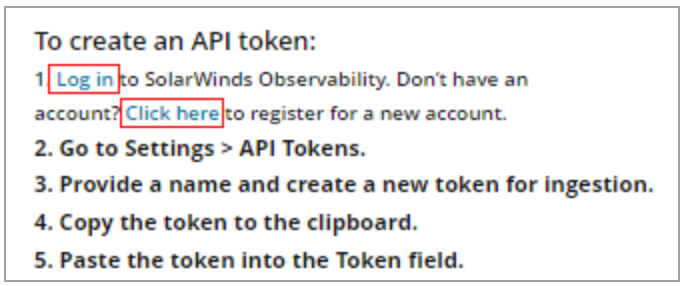
  The `repo.properties` file might be updated, for example, when you change the location of the Find SQL indexes.

Connect DPA to SolarWinds Observability

1. Log in to DPA as an administrator.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Configuration, click Platform Connect.

   The Platform Connect wizard opens.
4. On the Welcome screen, click Next.
5. If you accept the license agreement, select the check box and click Next.
6. If you have a SolarWinds Observability account, click the Log in link on the Create Account/Configure Token page to open SolarWinds Observability. Then continue with the next step.

If you do **not** have a SolarWinds Observability account, complete the following steps to create a new trial account.



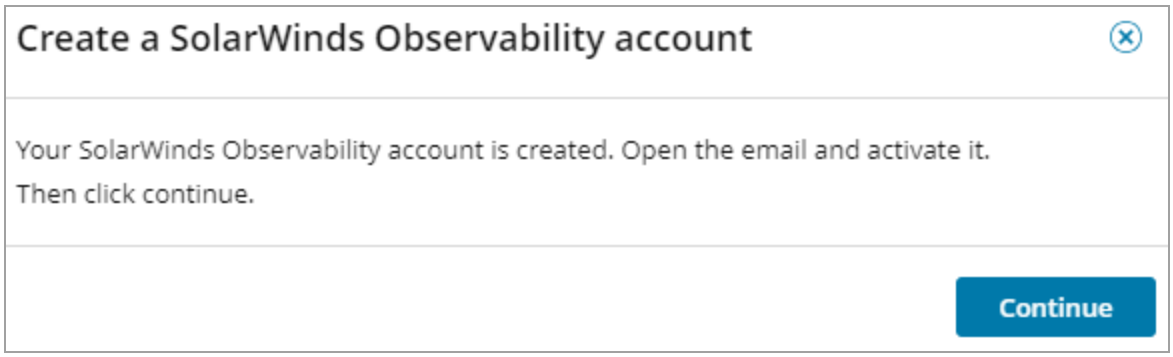
With a trial account, you can evaluate Platform Connect and SolarWinds Observability for 30 days.

a. Click the Click here link on the Create Account/Configure Token page.

   The Create a SolarWinds Observability account dialog opens.

b. Enter your information and select a Cloud Provider. Then click Create.

   A message tells you that your account has been created.



c. Check your email. When you receive an email from solarwinds.com, open it.

   The email provides information about SolarWinds Observability.

d. Click the link in the email to open the SolarWinds Observability login page.

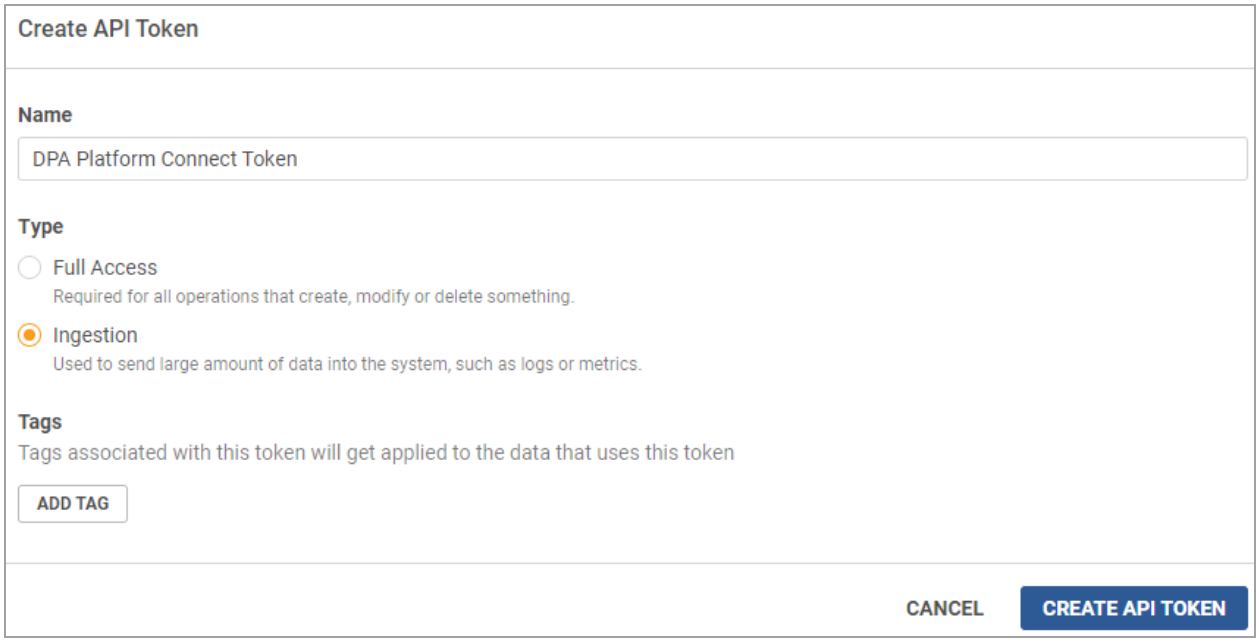
e. Enter a password for the account, confirm the password, and click Save New Password.

   The SolarWinds Observability Home page opens.

f. If the dialog prompting you to add entities is open, click Skip to close it.

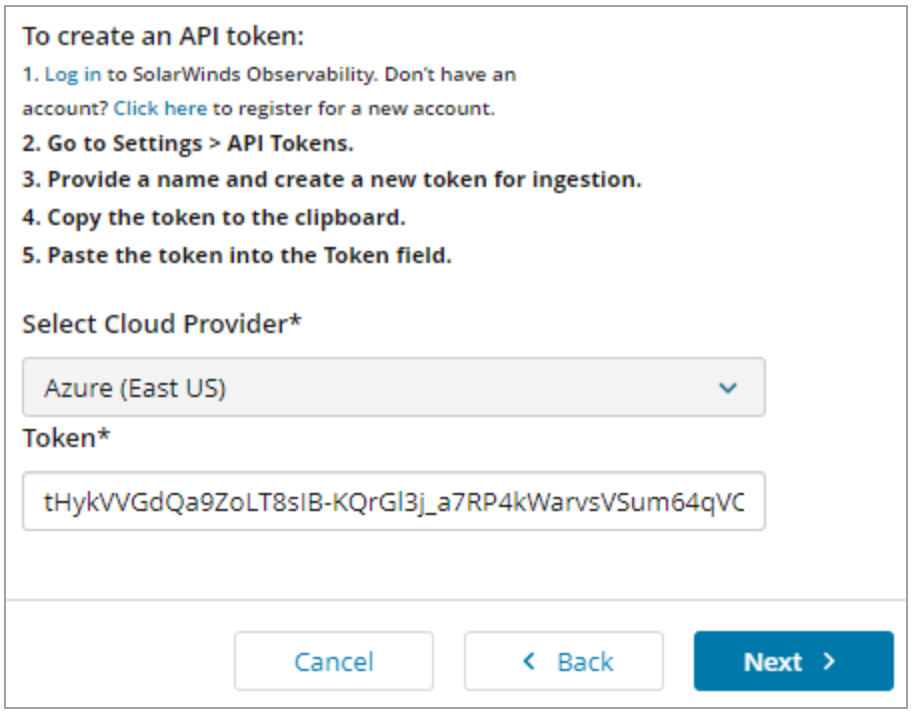
7. In the SolarWinds Observability left menu, click Settings.
8. In the My Settings menu, click API Tokens.
9. Click Create API Token.

10. Enter a name, verify that Ingestion is selected, and click Create API Token.



    A dialog displays the API token.

11. Click the clipboard icon to copy the token to the clipboard. Then click Close.
12. Copy the token to the clipboard. Paste the token into the Token field.
13. Return to the browser tab in which DPA is running. On the Create a SolarWinds Observability account dialog, click Continue to close the dialog.
14. Select a cloud provider, and paste the copied token into the Token field. Then click Next.



The Install Agent page opens.

15. Click Start Installer to begin installing the SolarWinds Observability Agent on your DPA server.

    The agent is the behind-the-scenes service that enables the connection between DPA and SolarWinds Observability.

    DPA displays a message when the installation is complete.

16. Click Finish.

    The connection between DPA and SolarWinds Observability is configured, and the Manage Platform Connect Service page opens.

Configure a connection service in SolarWinds Observability

If you have not done so already, set up a PagerDuty or ServiceNow integration in SolarWinds Observability. For details, see Notification Services settings.

Create a contact to send notifications to a notification service

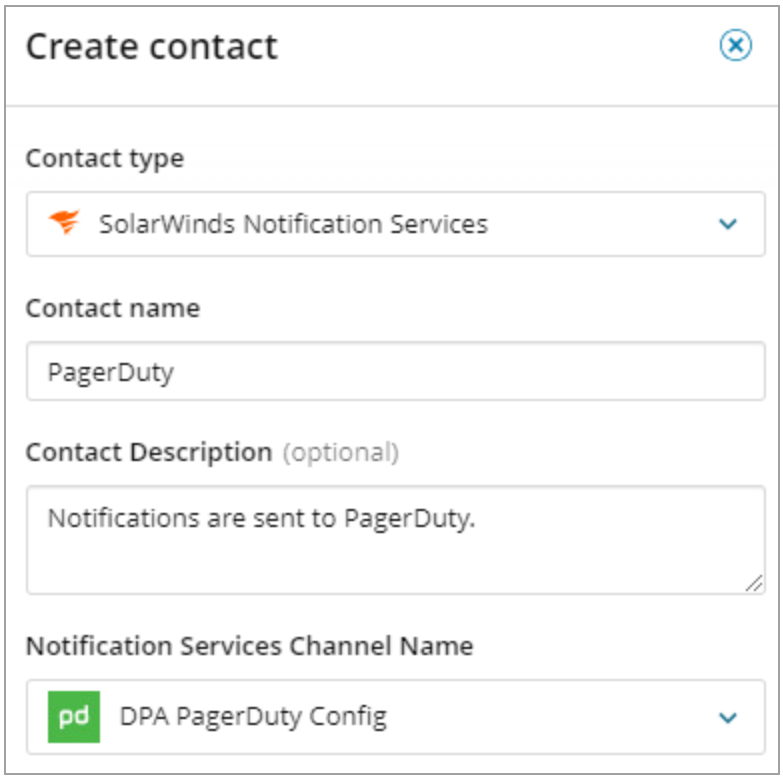
When you create an alert definition and select this contact to receive notifications, the notifications are sent to the associated notification service.

1. Log in to DPA using an account with administrator privileges.
2. From the DPA menu in the upper-right corner, click Options.

3. Under Administration > Users & Contacts, click Contact Management.
4. On the Contact management page, click Create contact.

   If any contacts are selected, the Create contact button is not displayed.

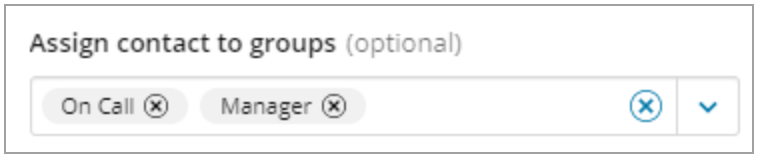
5. Under Contact type, select SolarWinds Notification Services.
6. Enter a name and, optionally, a description.
7. Under Notification Services Channel Name, select the name associated with the notification service you want to send notifications to.



8. To test the configuration, click Send Test Notification and verify that the test was sent to the notification service.
9. (Optional) Add the contact to one or more groups.
   a. Click the down-arrow under Assign contact to groups.
   b. Click a group name to select the group.

The group is shown in the Assign contacts to groups box.

c. Repeat these steps to add the contact to more groups.



10. Click Create.

    The contact is added to the list of contacts.

Mapping an alert definition to the contact

When you create an alert definition, select the contact you created in the previous section from the Select a recipient drop-down menu. When an alert is triggered at the associated alert level, DPA sends notifications to the notification service.

Indication that the notification was forwarded

To confirm that an alert notification was forwarded to ServiceNow or PagerDuty, view the alert history.

1. From the DPA menu in the upper-right corner, click Alerts.

   The Current alert status page shows information about DPA alerts that are currently active.

2. Click the name of the alert. The Alert details page opens.

3. Scroll down to the History of this alert on this database instance page.

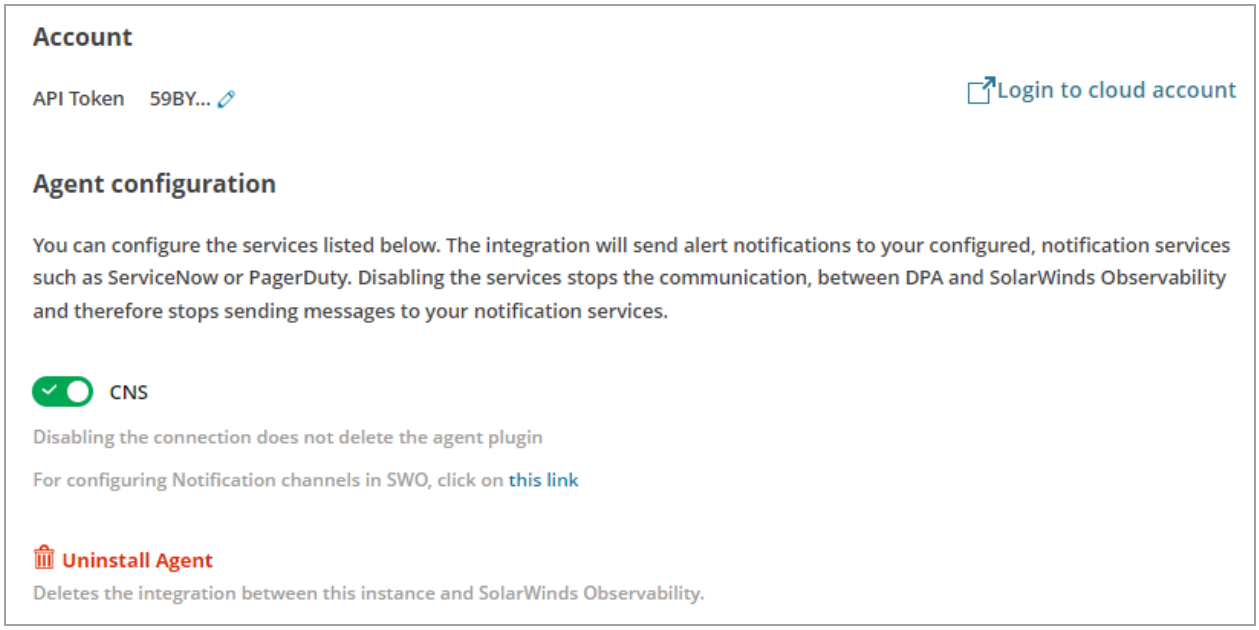
   A PagerDuty or ServiceNow icon is displayed for notifications that were forwarded to a notification service.

Disable Platform Connect

If you want DPA to stop sending notifications to SolarWinds Observability, disable the connection between DPA and SolarWinds Observability.

1. Log in to DPA as an administrator.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Configuration, click Platform Connect.

   The Manage Platform Connect Service page opens.



4. Click the green Notification Services button to deselect it.

   The button turns dark gray, and a message informs you that the service is disabled.

Define email templates for alert notifications

When an alert is triggered, DPA sends an email to notify the designated recipients. Email templates define the contents of the email notification. DPA provides a DPA System Template, but you can create custom templates and assign them to alert definitions.

Admin privileges are required to create or manage custom email templates.

Create or edit a custom email template for DPA alert notifications

Use custom email templates to customize the contents of the email notifications that DPA sends when alerts are triggered. You can create multiple custom email templates for different types of alerts.

Templates allow you to include additional information in email alerts. DPA provides a set of default variables (such as the alert value and database name), and you can also include custom properties as variables. Receivers of email alert notifications can set up rules to prioritize, forward, or otherwise process incoming notifications based on this information.

1. From the DPA menu in the upper-right corner, click Alerts.
2. In the upper-right corner, click Manage alerts.
3. Click the Email Templates tab.
4. Do one of the following:
    - To create a new email template, click Create email template.

      The Create email template page opens. It includes the system-defined template definition as a starting point.
    - To edit an existing email template, click the email template name.

      The Edit email template page displays the existing template definition.
5. Enter a unique name and, optionally, a description.

6. Specify the content and formatting of email notifications based on this template:
    - To add content:
        - Drag and drop variables from the right panel into the Subject line or body (or type the variables). These variables represent information about the triggered alert or links to additional information in DPA. For information about the available variables, see Email template variables.
        - Type static text into the Subject line or body.
    - To remove content, delete variables or static text from the Subject line or body.
    - To format body text with bold, italics, or line breaks:
        a. Under Body message format, select HTML.

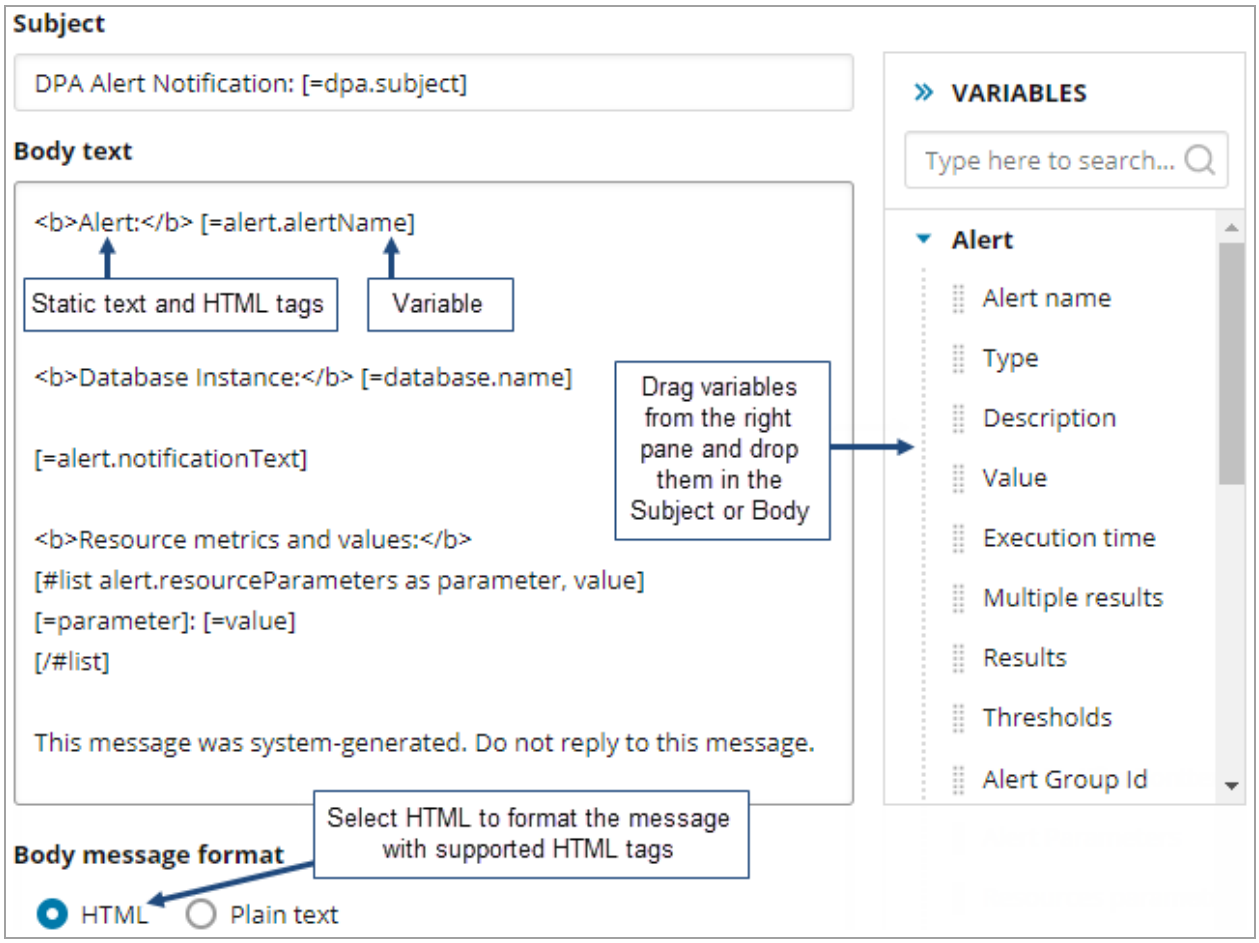
        > If Plain text is selected, any HTML tags are treated as text and shown in the alert notification email.

        b. Enter the following tags to format the body text:

        ```
        <b> </b>
        <i> </i>
        ```

 or

        > All other tags are unsupported. You cannot save a template that contains unsupported tags.



7. Click Save to save your changes and close the page.
8. Apply the email template to alerts in either of the following ways:
   - To apply the template to a specific alert, edit the alert definition and select the template from the Email Template drop-down menu.
   - To apply the template to all alerts that use the default email template, specify this template as the default.

Email template variables

The following variables are available:

- Alert variables
- Database variables
- Link variables
- DPA alert variables
- Custom properties

Alert variables

Use these variables to include information about the alert that was triggered.

Name	Variable	Description
Alert name	[=alert.alertName]	The user-defined name that identifies the alert that was triggered.
Type	[=alert.type]	The type of alert.
Description	[=alert.description]	DPA's description of the alert type.
Value	[=alert.value]	The value that triggered the alert.
Execution time	[=alert.executionTime]	The date and time when the alert was triggered.
Multiple results	[=alert.multiReturn]	A value of true or false to indicate whether the alert returns multiple values.
Results	[#list alert.results as result] [=result.category]: [#if result.parameterName??] [=result.parameterName] [#else] [/#if] [=result.label]: [=result.value] [#if result.units??] [=result.units][#else] [/#if] [=result.description] [/#list]	For Custom alerts that return multiple values, the parameter name, value, units (if specified in the alert parameters), and description (if specified in the alert parameters) for each returned value. ⓘ For custom alerts executed against a stored procedure, you can add the alert.results tag to include the `#ALERTSTRING#`.
Thresholds	[#list alert.threshold as t] Threshold level: [=t.level] * min: [#if t.levelMin??] [=t.levelMin][#else]N/A [/#if] * max: [#if t.levelMax??] [=t.levelMax][#else]N/A [/#if] [/#list]	The minimum and maximum values for threshold levels that are specified in the alert definition. ⓘ This variable does not return threshold information for Resources alerts because the thresholds are defined on the resource instead of in the alert definition.

Name	Variable	Description
Alert Notification text	[=alert.notificationText]	The text from the Notification Text field in the alert definition.
Alert Parameters	[#list alert.alertParameters as parameter, value] [=parameter]: [=value] [/#list]	For Wait Time and Administrative alerts, the name and value of the each parameter specified for the alert. If the alert type does not require parameters, this is blank.
Resources parameters	[#list alert.resourceParameters as parameter, value] [=parameter]: [=value] [/#list]	For Resources alerts, the name and value of each parameter specified for the alert.
Alert Group ID	[=alert.group.id]	The ID of the alert group that the alert belongs to.
Alert Group Name	[=alert.group.name]	The name of the alert group that the alert belongs to.
Alert Group Description	[=alert.group.description]	The user-defined description of the alert group that the alert belongs to.
Alert Status Value	[=alert.status.value]	The status of the alert (for example, High or Broken).
Alert Error Message	[=alert.status.message]	When the alert is Broken, the error message generated when the alert is triggered.
Single Alert	[=alert.singleAlert]	A value of true or false to indicate whether multiple result values are sent in a single message. True indicates that multiple results are sent in one message, and false indicates that they are sent separately.

Database variables

Use these variables to include information about the monitored database instance on which the alert was triggered.

Name	Variable	Description
Name	[=database.name]	The DPA display name of the monitored database instance.

Name	Variable	Description
Type	[=database.databaseType]	The type of monitored database instance.
IP address	[=database.ipAddress]	The IP address of the database server.
Hostname	[=database.hostname]	The host name of the database server.
Port	[=database.port]	The port used by the monitored database instance.
Version	[=database.databaseVersion]	Version of the monitored database instance.
Full type	[=database.databaseFullType]	The full type of the monitored database instance.

Link variables

Use these variables to include links to related information in DPA.

Name	Variable	Description
Alert Status	[=links.alertStatus]	A link to the Alert Status tab, from which you can view the status and history of the alert.
Alert Trends	[=links.alertTrends]	A link to the 1-day Trends chart for the day on which the alert was triggered.
Alert History	[=links.alertHistory]	A link to the detailed history of the alert on the database instance where it was triggered.
Instance Alerts	[=links.instanceAlerts]	A link to the Alert Status tab filtered to show only the alerts configured to run on the database instance where the alert was triggered.
Notification	[#list links.notification as label, url] [=label]: [=url] [/#list]	A link to the DPA chart associated with the alert type. For example, the link for a Database Instance Wait Time Anomaly alert opens the Anomaly Detection chart for the day when the alert was triggered. If no chart is associated with the alert type, this is blank.

DPA alert variables

These variables define the default DPA alert content.

Name	Variable	Description

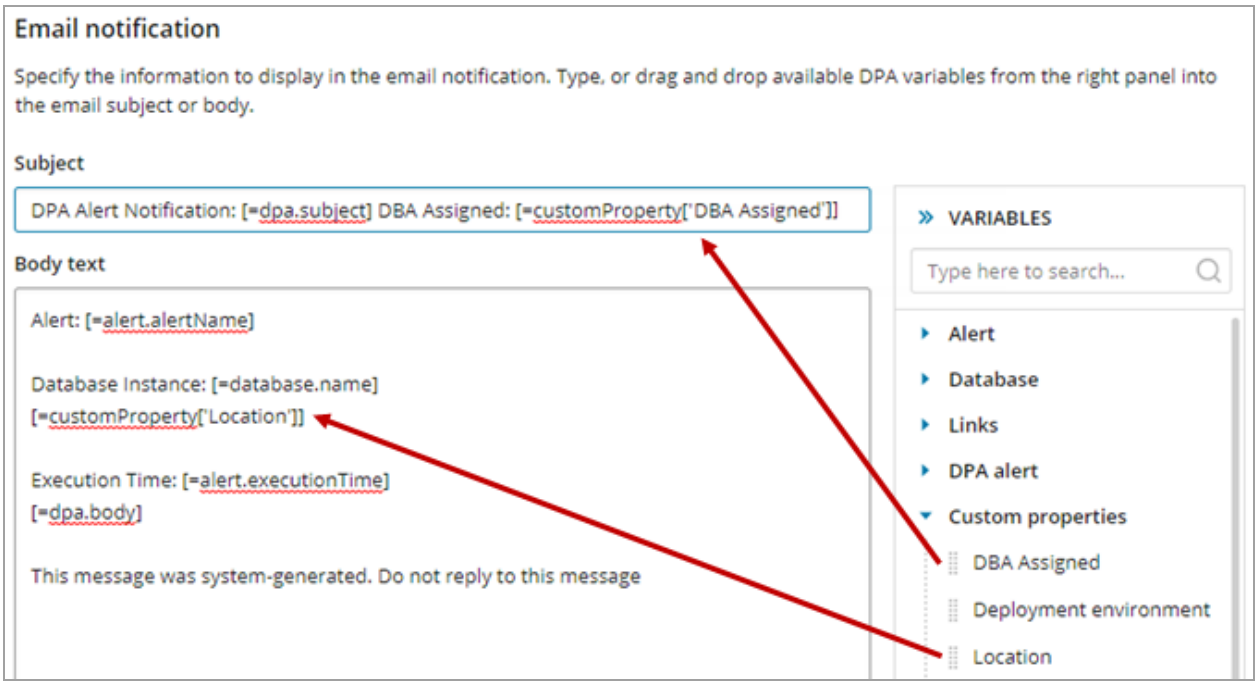
Subject	[=dpa.subject]	The default subject line of an email alert. This variable includes the alert name, the database instance, and the alert level in the following format:
		`Alert Name (Database Instance) - ALERT LEVEL`
		For example:
		`Total SQL Wait Time for Memory/CPU Waits (MyDatabaseInstance) - HIGH`

Results	[=dpa.body]	The default body text of an email alert. This variable includes the alert status link, the alert notification text, and the value that triggered the alert. Each element is a separate paragraph. For example:
		`View Alert`
		`Status: http://xxxxxxxx:8124/iwc/alertMain.iwc`
		`The total wait time for Memory/CPU waits has exceeded a threshold.`
		`Value: 600 seconds`

Custom properties

In addition to the predefined variables listed above, you can create custom properties to include other information in alert notification email templates. If any custom properties have been created for this DPA server, they are displayed in the Custom properties section in the Variables list.

The following example includes the custom properties DBA Assigned and Location. All DBAs receive these email notifications, but the DBAs can set up rules to process them. For example, a DBA named Bill Smith has a rule to delete DPA alert notifications if the Subject does **not** contain `DBA Assigned: Bill Smith`. If the Subject **does** contain that string, he has another rule to flag the email as important.



Delete a custom email template

If you delete a custom email template that is assigned to one or more alerts, DPA assigns the default template to those alerts.

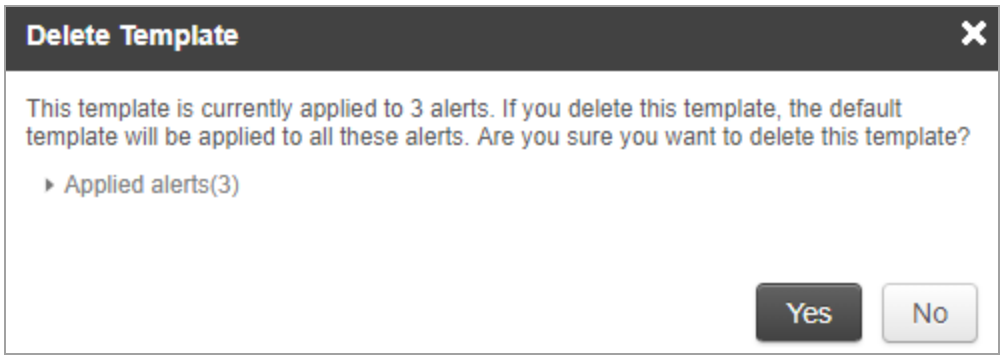
- You cannot delete the DPA System Template.
- You cannot delete a custom template that is currently designated as the default template.

To delete a custom template:

1. From the DPA menu in the upper-right corner, click Alerts.
2. In the upper-right corner, click Manage alerts.
3. Click the Email Templates tab.
4. If the template you want to delete is currently designated as the default template, designate a different template as the default template.
5. Locate the table row that shows the template you want to delete, and click Delete.

   If the template is not assigned to any alerts, DPA displays a simple confirmation message.

   If the template is assigned to one or more alerts, DPA displays the following confirmation message. To see which alerts use the template, click Applied alerts.

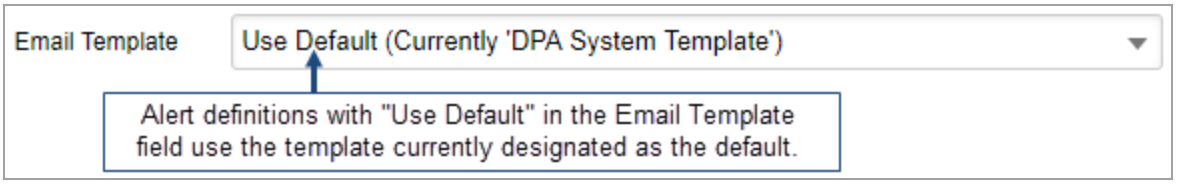


6. Click Yes at the confirmation message to delete the template.

   Any alerts that previously used that template now use the default template.

Change the default email template for DPA alert notifications

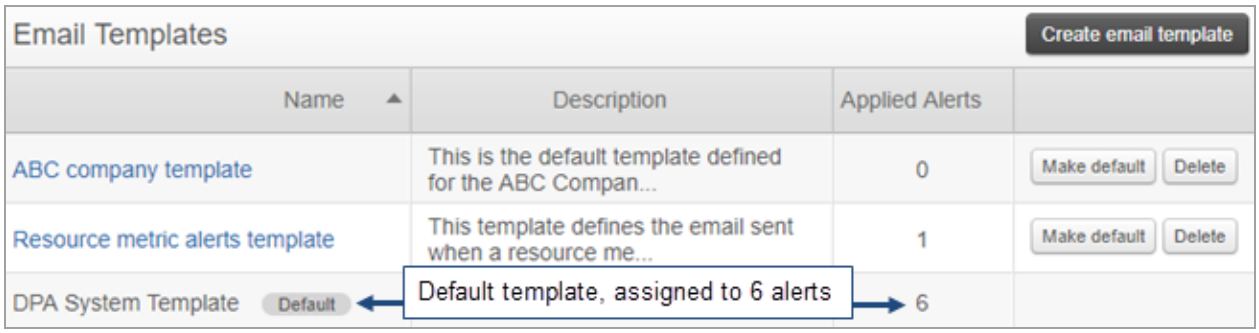
If an alert definition does not assign a specific email template to use for alert notifications, the default email template is used.



Initially, the DPA System Template is the default template. You can create a custom template and designate it as the default.

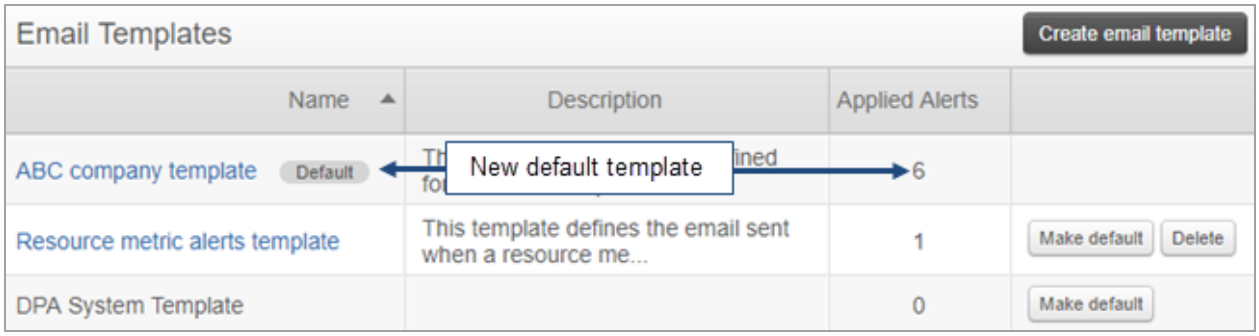
1. From the DPA menu in the upper-right corner, click Alerts.
2. In the upper-right corner, click Manage alerts.
3. Click the Email Templates tab.

   "Default" next to the template name identifies the default template.



4. Locate the template you want to designate as the default, and click Make default. Then click Yes at the confirmation prompt.

"Default" appears next to the template's name, and the number of Applied Alerts is updated. All alert definitions with "Use Default" in the Email Template field now use this template for email notifications.



The Delete button is no longer displayed for the new default template, because you cannot delete a template while it is designated as the default.

Create and manage custom properties

In addition to the physical properties of a database instance (such as database vendor, version, and IP address), you can create custom properties. Custom properties are used to associate business or organizational attributes to monitored database instances. For example, you could use custom properties to specify the location of a monitored database instance or the DBAs responsible for an instance.

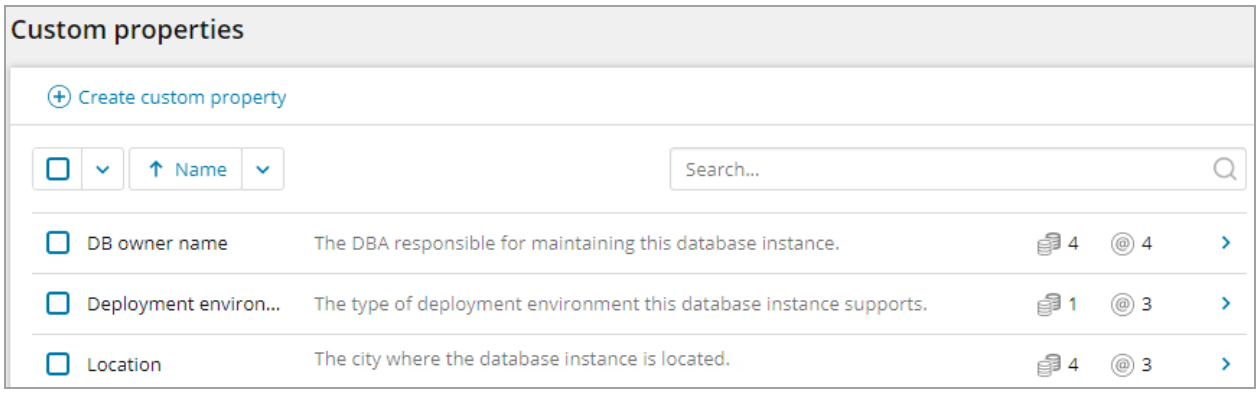
You can include custom properties as variables in email templates for alert notifications. Receivers of email alert notifications can set up rules to prioritize, forward, or otherwise process incoming notifications based on the custom property values in the emails. You can also include custom properties in alert assignment rules.

Only DPA administrators can create and manage custom properties.

Access the Custom properties page and view property details

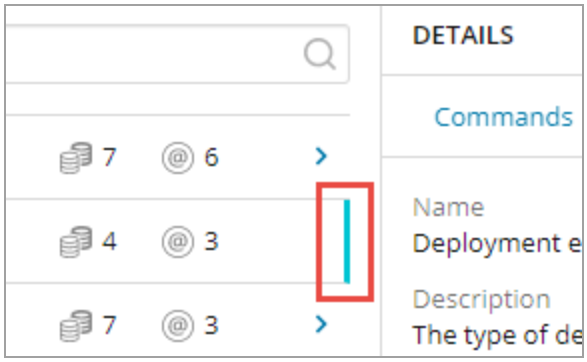
1. From the DPA menu in the upper-right corner, click Options.
2. Under Monitor Setup > Database Instances, click Custom properties.

   The Custom properties page lists existing custom properties. It shows the number of database instances each property is assigned to and the number of values defined for the property.

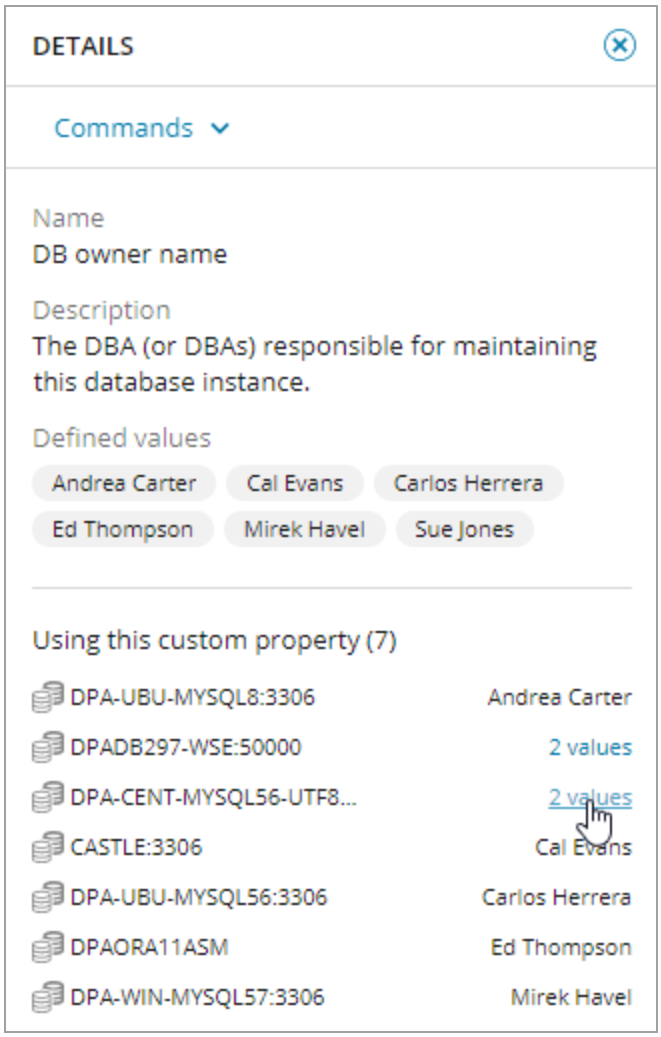


3. To see more information about a property, click the blue arrow on the right (›) to open the Details pane.

   When the Details pane is open, the blue arrow is replaced with a vertical blue bar to indicate which custom property is associated with the details pane.



   The Details pane shows all defined property values, the database instances using this property, and the value assigned to each database instance. If more than one value is assigned to a database instance, click the number to display the list of values assigned to that instance.

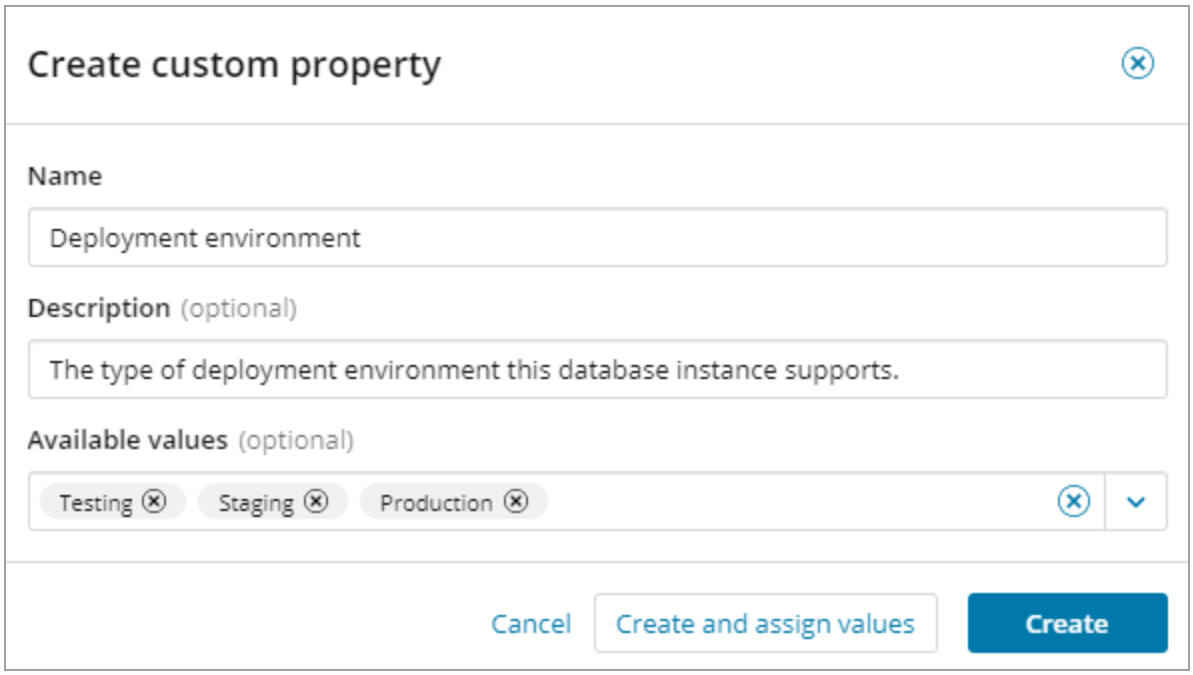


Create a custom property

1. At the top of the Custom properties page, click Create custom property.

   The Create custom property dialog box opens.
2. Enter a unique name to identify the property.
3. (Optional) Enter a description.

4. (Optional) Define one or more values for this custom property. To define each value:
    a. Type a value in the Available values field.
    b. Press Enter.



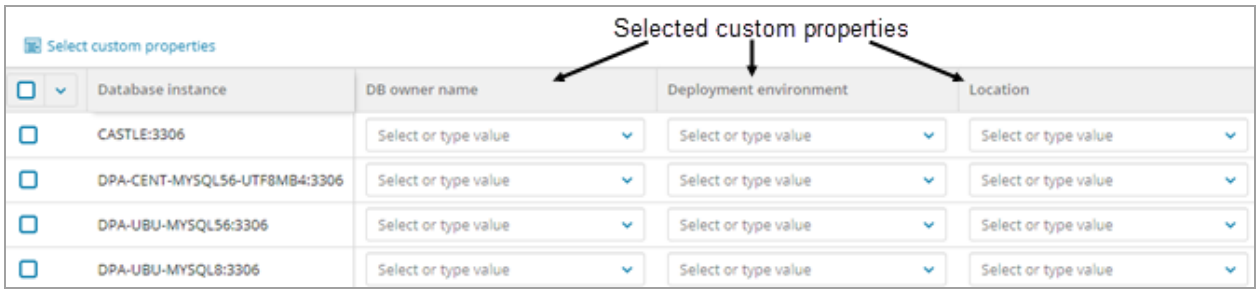
5. Do one of the following:
    - To save the property and close the dialog box, click Create.
    - To save the property and open the Assign custom property values page, click Create and assign values.

Assign property values to database instances

If a custom property applies to a database instance, assign the appropriate property values to that instance. You can assign multiple values for the same property.

1. On the Custom properties page, select the custom properties whose values you want to assign. Then click Assign values.

    The Assign custom property values page lists all database instances. It includes a column for each selected property.



You can also open this page from the Details pane by clicking Commands > Assign values.

2. (Optional) If you change your mind about the property values you want to assign, update the properties shown on the Assign custom property values page:

   a. Click Select custom properties.

      The Available custom properties dialog box opens. The currently selected properties are checked.

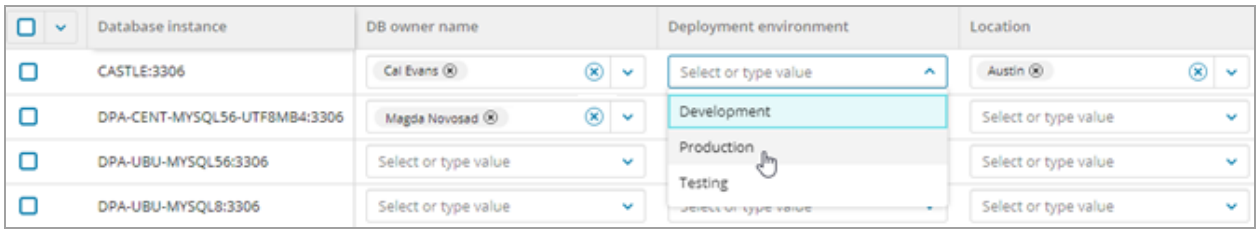
   b. Select or deselect properties to specify which properties you want to include.

   c. Click Apply to update the properties shown on the Assign custom property values page.

3. (Optional) Use the Filters pane on the left to reduce number of database instances in the list. For example, select a database type to list only instances of that type.

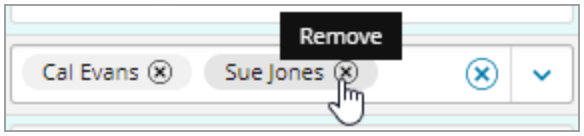
4. To assign or unassign property values for individual database instances, do the following:

   - To assign a value, click the down arrow and select the value from the drop-down list. If the value you need is **not** in the list, add it by typing the value in the text box and pressing Enter.

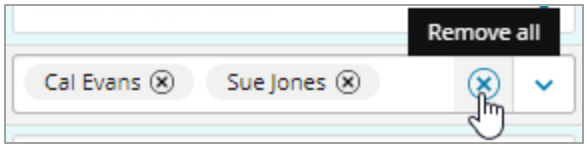


You can assign multiple values to the same property.

   - To remove **one** value assigned to a property, click the x next to the assigned property.



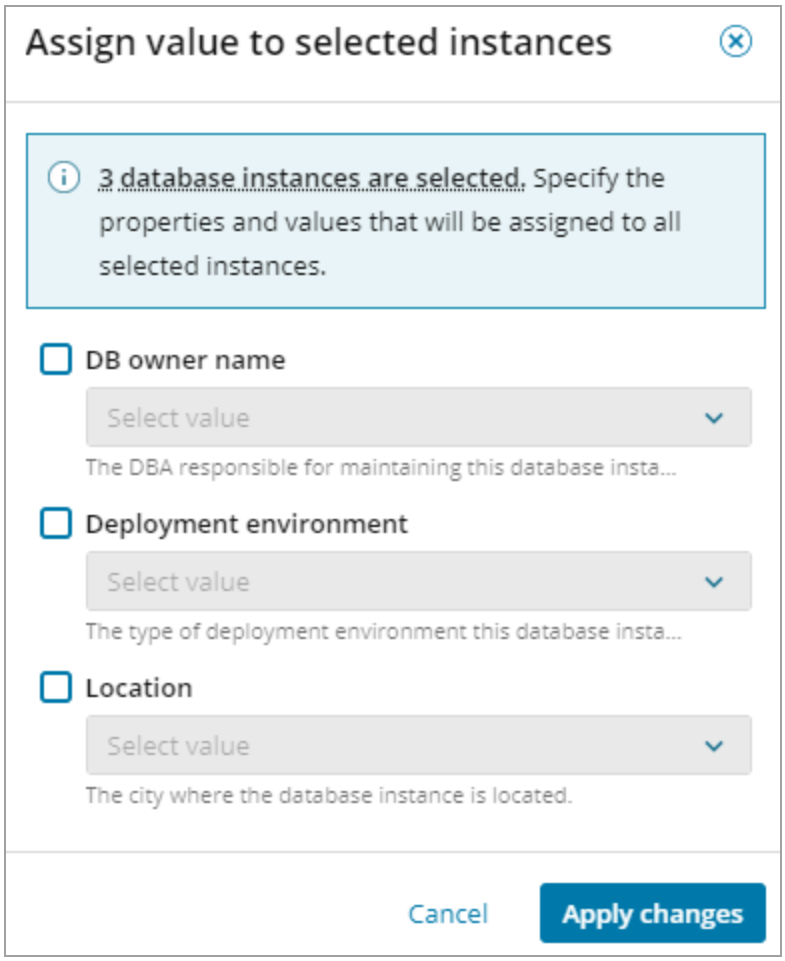
- To remove **all** values assigned to a property, click the x on the right side of the property box.



> Removing an assigned value from a property on a database instance does **not** delete the value. It is still available to be assigned to other database instances. To delete the value and remove it from all assigned instances, you must edit the custom property definition.

5. To assign the same property value to **multiple** database instances:
   a. Select the database instances.
   b. Click Assign to selected instances.

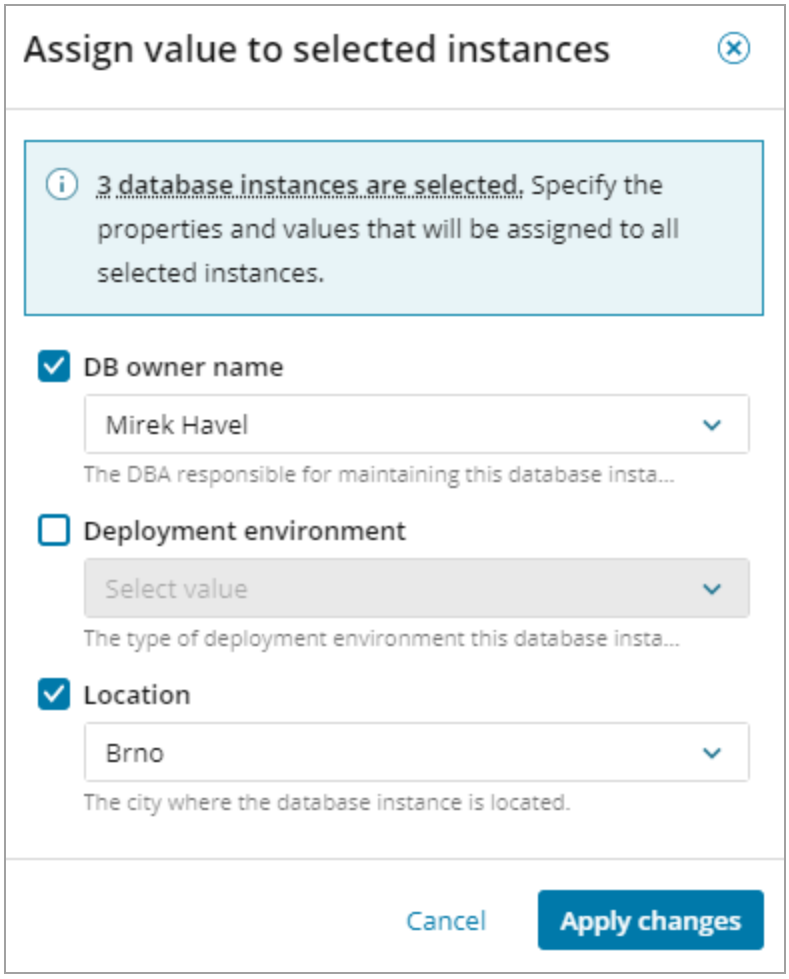
   The Assign value to selected instances dialog box opens.



When you use this dialog box to assign values to multiple instances:

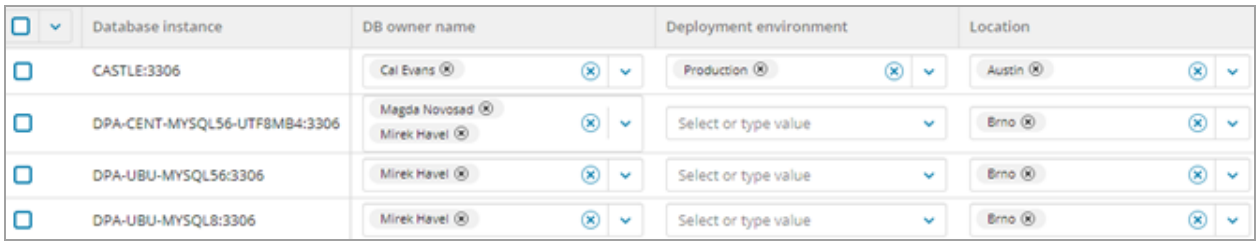
- You can only select values that were previously defined for the property. You cannot add new values.
- This dialog box does not display the values that are currently assigned to a property.
- If a value is currently assigned and you use this dialog box to assign a new value, the existing value is not replaced. Both values are assigned.

c. Select each property whose value you want to assign, and then choose a value from the drop-down list.



d. Click Apply changes.

Your changes are shown on the Select custom properties page.



6. Click Save to save your updates.

> To return to the Custom properties page, click Back .

Edit a custom property definition

1. On the Custom properties page, select a custom property and click Edit custom property. Or, in the Details pane for that property, click Commands > Edit custom property.

   The Edit custom property dialog box opens.
2. Edit the Name and Description as needed.
3. To add a value, type the value name in the Available values box and press Enter.
4. To remove a value, click the x on the oval that shows the value name.
5. If you remove values by accident, you can add them back:
   a. Click the down arrow on the Available values box.

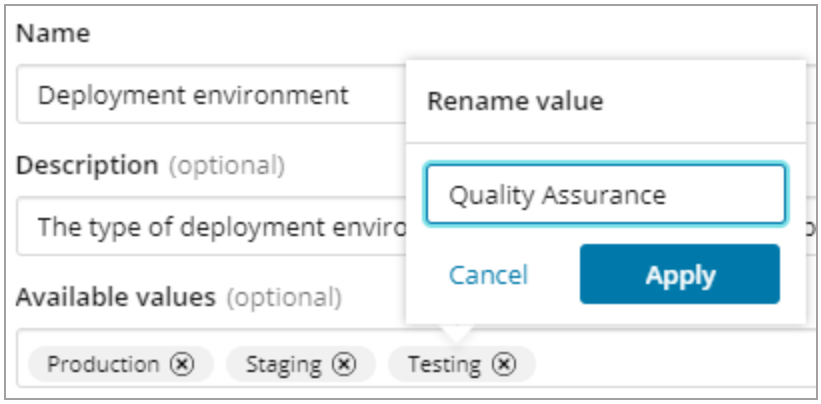
      The drop-down shows all values deleted during the current editing session.

   b. Click a line to add that value back.
   c. Repeat these steps to add other deleted values back.
6. To change the name of a value:

a. On the oval that shows the value name, click anywhere **except** the x.

b. Type the new name and click Apply.



7. Do one of the following:

   - To save your changes and close the dialog box, click Save.
   - To save your changes and open the Assign custom property values page, click Save and assign values.

> If you removed a value that is currently assigned to one or more database instances, a confirmation message is displayed.
>
> - If you do **not** want to delete the values that are in use, click Cancel to return to the Edit custom property dialog box. Then complete the steps to add that value back.
> - If you want to delete the values that are in use, click Delete.

Delete a custom property

When you delete a custom property, any property values that were assigned to database instances are automatically removed.

1. On the Custom properties page, select one or more properties and click Delete. Or, in the Details pane for a property, click Commands > Delete.
2. On the confirmation prompt, click Delete.

Import and export custom definitions and entities

With DPA 2022.4 and later, you can export custom definitions and entities to a file. You can then import all data in the file, or select a subset of the data to import. Use this capability to:

- Create a backup of your customizations. If customizations are subsequently lost or changed, use the backup to restore the customizations that existed at the time of the export.
- In an environment with multiple DPA servers, replicate the custom definitions and entities configured on one server to other DPA servers.

You can import and export the following custom definitions and entities:

- Alert definitions, which raise an alert whenever certain contacts exist.

  When you import an alert definition, some information (such as the contact for each alert level) is not included and must be specified after the import.

- Alert assignment rules, whose logic is used to determine which database instances are assigned to any alert that uses the rule.
- Custom property definitions, which are used to associate business or organizational attributes to monitored instances. Custom properties can be used in alert assignment rules and in alert notification email templates.

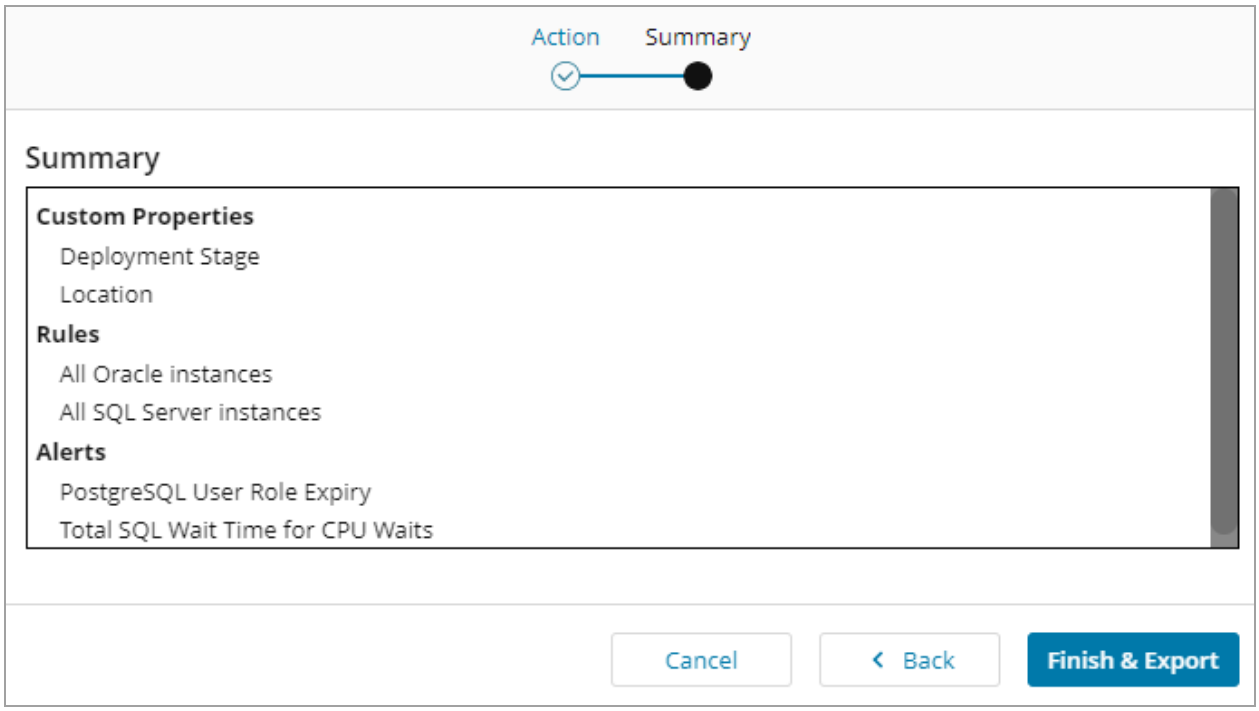
Exporting custom definitions and entities

The export includes all the alert definitions, alert assignment rules, and custom property defitions that are currently configured on the DPA server.

1. Log in to DPA as a user with administrative privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Support > Utilities, click Export Custom Entities.

   The Export Custom Entities wizard opens.
4. Click Next.

   The Summary page lists the custom properties, rules, and alerts that will be exported.



5. Click Finish & Export.

   The custom entity definitions are saved in a file for each entity type. These files are compressed to a `.zip` file named `CustomEnties_YYYY-MM-DD-timestamp.zip`. The `.zip` file is placed in the default download directory on your client computer.

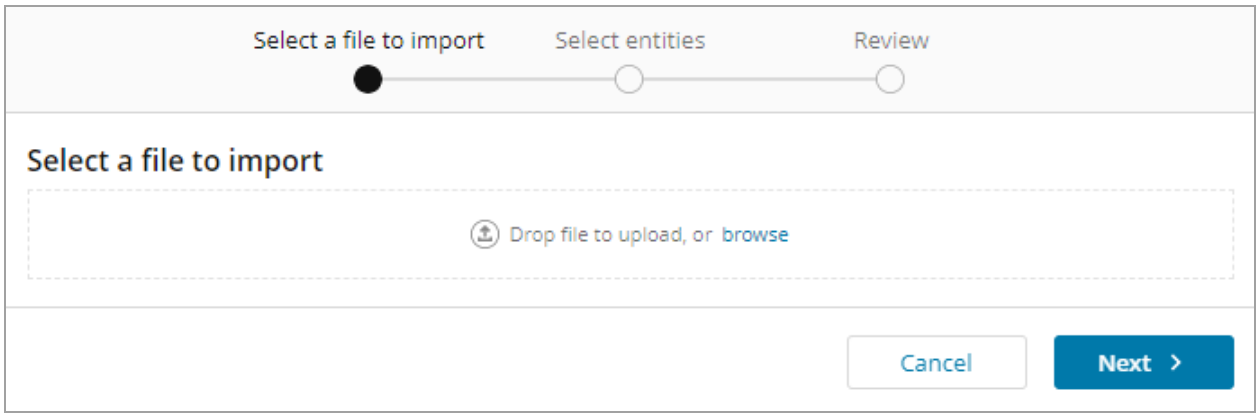
   ⚠ Do **not** edit these files with a text editor. The files cannot be imported if they have been edited.

Import custom definitions and entities

During the import, you can select which exported definitions and entities you want to import and whether DPA should overwrite existing entities with the same name.

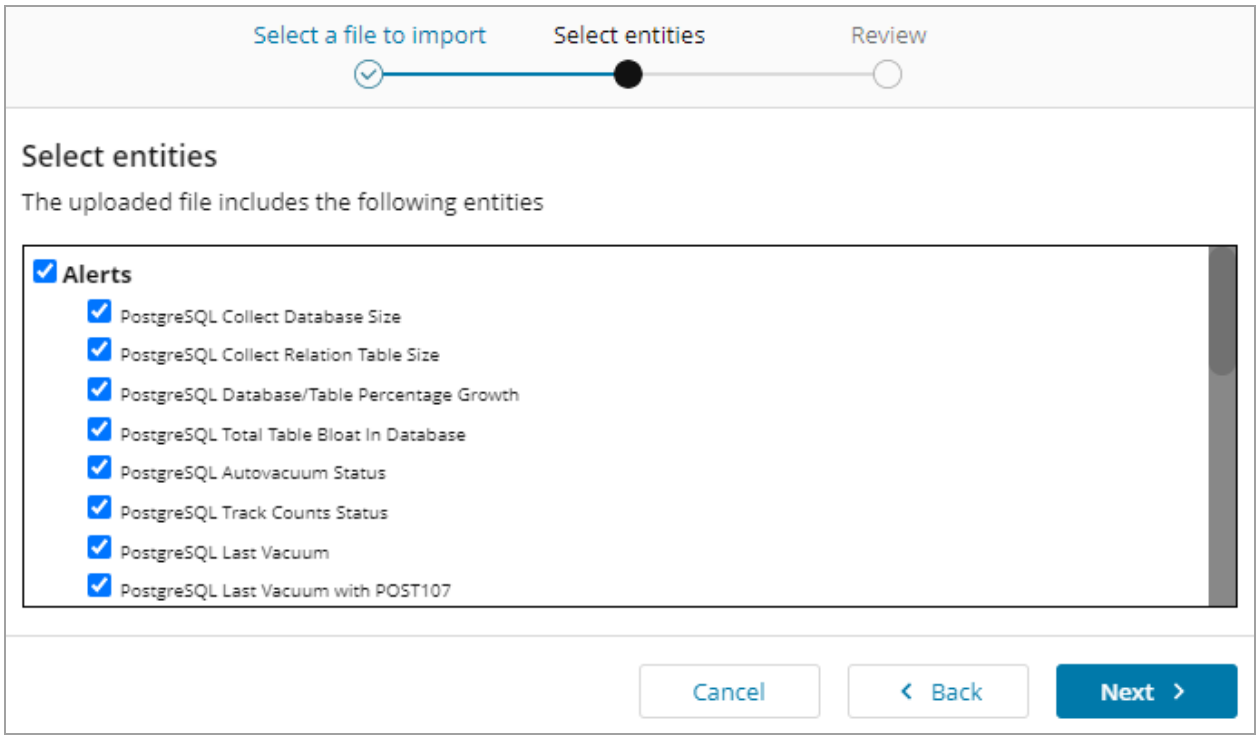
1. Log in as an administrator to the DPA instance where you want to import content from a DPA export file.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Support > Utilities, click Import Custom Entities.

   The Import Custom Entities wizard opens.



4. Browse to select the exported `.zip` file, or drag and drop the file on the wizard. Click Next.

   The wizard lists the entities included in the file. By default, all entities are selected.



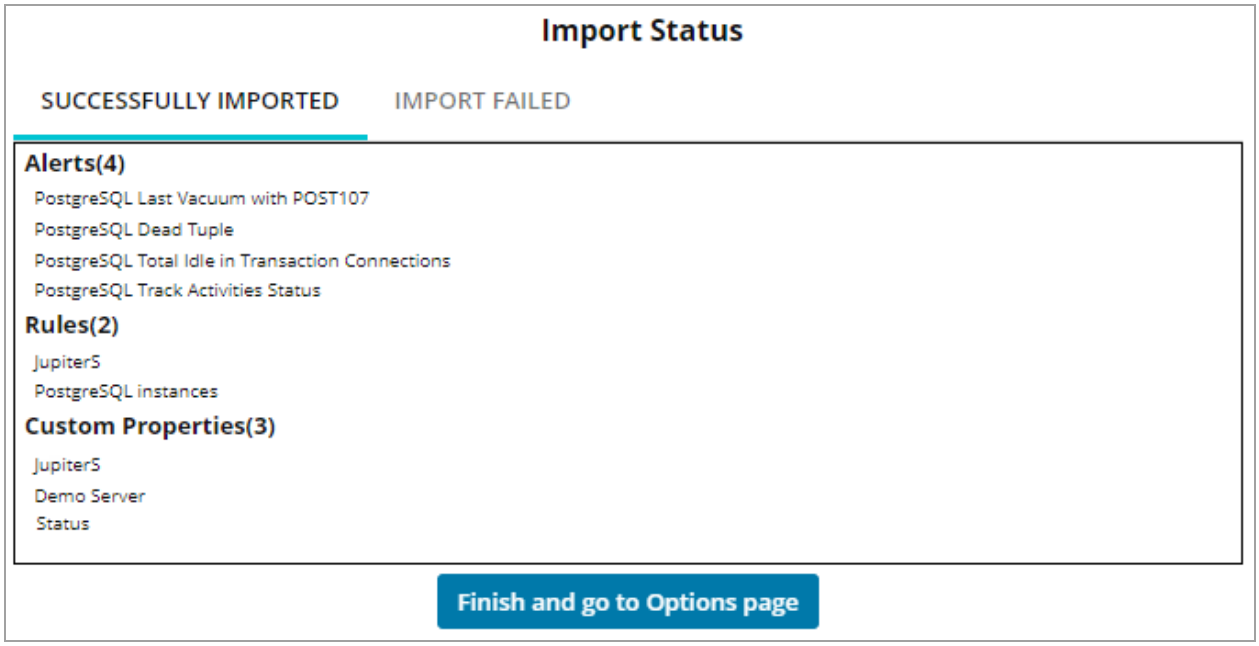
5. Clear the checkbox next to any entity that you do not want to import. Click Next.
6. Review the entities to be imported.

   The Existing Entities tab lists any existing entities that have the same type and name as entities selected for import. By default, existing entities are **not** overwritten during an import. If you want to update existing entities, select Overwrite existing entities.

   ⚠ If you choose to overwrite an existing entity, that entity is deleted and replaced by the entity defined in the export file.

7. Click Import to import the entities.

   The Import Status page shows the result of the import.



8. To close the dialog, click Finish and go to Options page.
9. If you imported alert definitions, edit each definition to specify the following information:
   - If the associated database instances were manually selected (instead of being automatically determined by an alert assignment rule), or if the associated rule was not imported and did not already exist, specify the database instances that the alert applies to. (You can specify instances by manually selecting them or by applying a rule.)
   - Specify the group or individual to notify for each alert level.
   - To use a notification policy other than the default, select the notification policy.

Link together separate DPA servers

Use Central Server mode to link separate DPA servers together and to view summarized information from multiple servers in one place. This is useful if query volumes have exceeded one server's capacity, if your monitored databases are distributed geographically, or if you want to separate monitored instances by team or business unit.

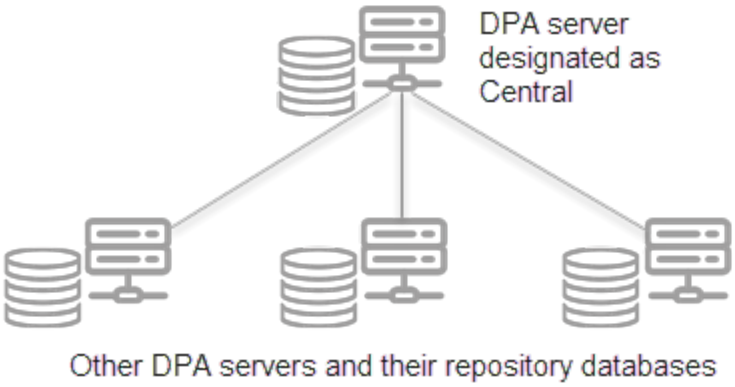
Set up a Central Server and add remote DPA servers

Use Central Server mode to link separate DPA servers together. Examples of why you might want to do this include:

- The infrastructure resources supporting DPA (for example, storage availability, I/O throughput, RAM, and CPU) reach capacity as the volume of queries being analyzed increases.
- Your monitored instances are distributed geographically, and the network lag time to distant instances is high. You can install separate DPA servers in each location.
- You want to enable separate teams or business units to manage their subset of database instances.

The Central Server collects information from your remote servers and consolidates the data into a single interface.

Each DPA server has its own repository. The Central Server has low overhead and no additional information from other DPA installations is added to its repository database.



Set up a Central Server

1. Install DPA on a server. This will be your Central Server.
2. Log in to that instance as an administrator.

3. From the DPA menu in the upper-right corner, click Options.
4. Under Administration > Display, click Manage Central.

   Your DPA server should be listed as the Central DPA Server in the list of Registered Servers.

Add remote DPA servers

The user credentials for the Central and remote DPA servers must match. See Configure authentication for the DPA Central Server for more information.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Display, click Manage Central.
3. Click Add Server.
4. Enter information about the remote DPA server.
5. Click Test connection, and click Save.

   > A successful test indicates that DPA can communicate with the remote server through the provider host and port. It does not indicate that DPA can authenticate users.

   If the test fails, check the host name in the Server Name field. Does it contain an underscore (_) character? An underscore is not valid for host names. If you cannot rename the host, enter the IP address.
6. Repeat steps 1 - 4 for the remaining remote DPA servers.

The details of your remote DPA servers are not stored in the repository, but in a file on your Central Server, located here:

`DPA-install-dir/iwc/tomcat/ignite_config/iwc/central/RemoteRepositories.json`

This is a plain-text JSON file. No sensitive data is stored in this file.

Configure authentication for the DPA Central Server

You can authenticate to the Central Server and the remote servers using one account. The account must be added to each server as a DPA user, or through an Active Directory (AD) or LDAP group.

Log in with a DPA user

You must create the user on the Central Server and each remote server. See Create a DPA user account and assign privileges for more information.

> The password must match on all servers.

Read-only permissions are sufficient to view data from the remote repositories.

Log in with an Active Directory or LDAP user

You must first set up AD or LDAP on the Central Server and each remote server. See DPA user authentication for more information.

Next, create the AD or LDAP user group on the Central Server and each remote server.

Read-only permissions are sufficient to view data from the remote repositories.

View information from remote servers on the DPA Central Server

In a DPA Central Server, the Central page displays summarized information about all connected DPA servers.

Open the Central page

The default home page of a Central Server is the DPA home page. Open the Central page to see database information from all connected DPA servers.

1. Log in to your DPA Central Server as an administrator.
2. From the DPA menu in the upper-right corner, click Central.

View the number of alarms for each server

On the Central page, click the Alarm Summary tab to see a summary of how many critical or warning level alerts or alarms are reported by each DPA server. The number on each row represents the number of issues reported for all instances monitored by that DPA server.

Server (version)	Alerts	Wait	Tuning	CPU	Mem	Disk	Sess	Net
10.199.8.205 (2024.2.0.500)	1 0 0 0 0	0 0	1 0	1 0	0 1	0 0	0 0	0 0
10.199.8.146 (2023.5.0.1550)	0 5 0 0 0	0 0	2 0	1 0	1 1	0 0	0 0	0 0
10.199.8.185 (2024.2.0.8)	0 1 0 0 0	0 0	2 0	1 0	0 1	0 0	0 0	0 0
DPA-NesoA (2023.4.300.502)	? ? ? ? ?	0 0	5 7	1 1	3 1	0 0	0 0	0 2
Totals	1 6 0 0 0	0 0	10 7	4 1	4 4	0 0	0 0	0 2

The Alarm Summary displays the following information:

- Alerts: The number of broken alerts, and the number of alerts at each level that are triggered but not acknowledged. When the table is sorted by the Alerts column in descending order, broken alerts have the highest severity, followed by high-level alerts, followed by each subsequent alert level.

  > Alert information is available in DPA 2024.2 and later. For DPA servers running earlier versions, a question mark is displayed. Upgrade your DPA servers to 2024.2 or later to see alert information for all servers.

- Wait: The number of database instances where the wait time during the last hour was higher or much higher than normal.
- Tuning: The number of query, index, or table tuning advisors with a warning or critical status.
- CPU: The number of CPU metrics whose value exceeded the warning or critical threshold.
- Mem: The number of memory metrics whose value exceeded the warning or critical threshold.
- Disk: The number of disk metrics whose value exceeded the warning or critical threshold.
- Sess: The number of session metrics whose value exceeded the warning or critical threshold.
- Net: The number of network metrics whose value exceeded the warning or critical threshold.

To see more detailed information, click a DPA server name or IP address in the Server column to open the DPA home page for that server.

View the number of registered databases not being monitored

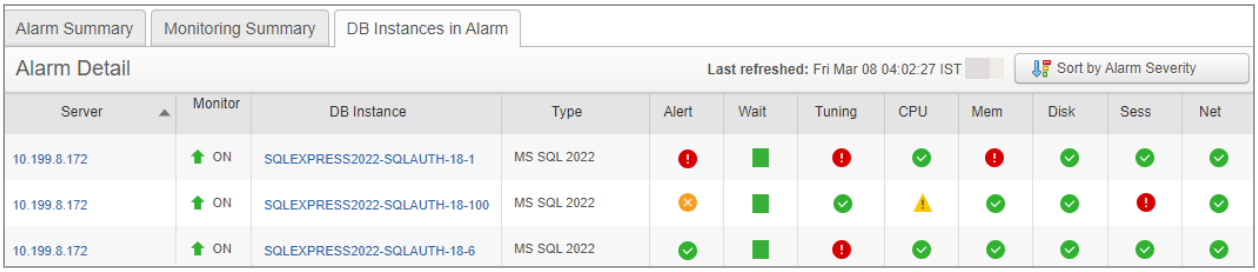
On the Central page, click the Monitoring Summary tab to see how many instances registered by each DPA server are running. Each column represents a database type. A green up arrow indicates that some or all of the registered instances are running, and a blue down arrow indicates that none of the registered instances are running.

To see more detailed information, click a DPA server host name or IP address in the Server column to open the DPA home page for that server.

View alarm information for each monitored database instance

On the Central page, click the DB Instances in Alarm tab to see which monitored database instances have triggered alerts or alarms. The color of the square or icon indicates severity.

The DB Instances in Alarm tab lists a maximum of 200 database instances. If you are monitoring more than 200 instances, sort by severity to see the instances with the most severe issues.



The following columns indicate whether any alerts or alarms were detected for each monitored instance:

- Alert: If the instance has any broken alerts, this column contains the broken alert icon. Otherwise, it contains an icon to indicate the highest severity alert that is triggered but not acknowledged. When the table is sorted by the Alerts column in descending order, rows with broken alerts are first, followed by rows with high alerts, followed by each subsequent alert level.

  Alert information is available in DPA 2024.2 and later. For DPA servers running earlier versions, a question mark is displayed. Upgrade your DPA servers to 2024.2 or later to see the alert status for all monitored instances.

- Wait: A colored box to indicate whether anomalies (higher than usual waits) were detected for the instance during the last hour.
- Tuning: An icon to indicate if query, index, or table tuning advisors with a warning or critical status were detected for the instance.
- CPU: An icon to indicate if the values of CPU metrics exceeded the warning or critical threshold.
- Mem: An icon to indicate if the values of memory metrics exceeded the warning or critical threshold.
- Disk: An icon to indicate if the values of disk metrics exceeded the warning or critical threshold.
- Sess: An icon to indicate if the values of session metrics exceeded the warning or critical threshold.
- Net: An icon to indicate if the values of network metrics exceeded the warning or critical threshold.

To see more detailed information, click a DPA server name or IP address in the Server column to open the DPA home page for that server. Or click the instance name in the DB Instance column to open the Trends charts for that instance.

Sort by severity

By default, the Alarm Detail table on the DB Instances in Alarm tab is sorted by the DPA server name or IP address. To put the instances with the most severe issues at the top of the list, click Sort by Alarm Severity.

When multiple rows have the same severity score, issues in the Alert column are given more weight, followed by the Wait column, the Tuning column, and then all other columns.

Search for a monitored database instance

From any tab, you can search for a monitored database instance by name.

1. In the Search by DB instance name box, enter a string that is included in the instance name as it is displayed in DPA.
2. To limit the search to one database type, select a type from the drop-down menu.
3. Click Search.

   The search results are listed on a separate tab, which opens automatically. It displays the same information about each instance as the DB Instances in Alarm tab.

Change the default refresh rate

When users are viewing the DPA Central page, the displayed data is refreshed every 300 seconds by default. You can change the default refresh rate.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Display, click Display Options.

   The Display Options page opens.
3. In the DPA Central Page Refresh box, enter the number of seconds between DPA Central data refreshes.

Manually refresh cached data

To manually refresh the DPA Central cache and update the displayed data:

1. From the DPA menu in the upper-right corner, click Options.
2. Under Support > Utilities, click Refresh Central Cache.

Advanced configuration for the DPA Central Server

You may need to change the Central Server configuration to make it run more efficiently in your environment.

To change the default behavior:

1. Open the `system.properties` file in a text editor. This file is located in the following directory on your Central Server:

   DPA-install-dir`/iwc/tomcat/ignite_config/idc`

2. Add one or more of the settings described below, and save the file.

General Central Server settings

Setting	Value	Description
com.confio.iwc.central.enabled	true (default) false	Enables or disables the use of Central Server mode.
com.confio.iwc .token.login.supported	true (default) false	Enables or disables the use of encrypted login tokens when jumping from the Central Server to a remote instance. If true, a web service call authenticates the user and creates a temporary token to identify the incoming user and bypass the login process. If false, the user is always prompted to log in to the remote instance.
com.confio.iwc.show.all.errors	true false (default)	Determines which users see failures in the Unavailable DPA Servers section. If true, all users see failures for all instances. If false, only administrators see failures. Set this option to false if you do not want all users to know about other DPA instances in the organization.
com.confio.iwc.automatic.update	true (default) false	Enables or disables a process that performs simple checks on the file when DPA starts. For example, flagging any local instances as the Central Server.

Setting	Value	Description
com.confio.iwc.alarm.level	Warning	The minimum message level to include on the Alarm Details tab. Valid values are below. If (empty) is set, details are disabled. • Critical • Warning • Normal • (empty)
com.confio.iwc.alarm.count	200	The number of detail rows to show on the Alarm Details tab.

Thread pool settings

These settings control the number of threads that are used by the Central Server to make web service calls to other remote servers. The default settings are set for a few concurrent users hitting up to 100 remote instances. If you have more than 100 instances or many concurrent users, SolarWinds recommends adjusting these settings higher.

Setting	Value	Description
com.confio.iwc.centralServiceTaskExecutor.corePoolSize	50	The core number of threads that Central Server uses to make web service calls to remote servers.
com.confio.iwc.centralServiceTaskExecutor.maxPoolSize	150	The maximum number of threads that Central Server uses to make web service calls to the remote servers. Central Server adds more threads only when all core threads are in use and the task queue is full.

Setting	Value	Description
com.confio.iwc.centralServiceTaskExecutor.queueCapacity	5000	The maximum number of requests in the queue before Central Server either creates new threads to help with the work or rejects the request. Tasks are rejected if all 40 threads cannot keep up with the requests being made.
com.confio.iwc.centralServiceTaskExecutor.keepAliveSeconds	120	The number of seconds to keep an idle thread before removing it.

Client factory cache

A client factory creates web service clients that talk to remote instances on a per-user basis. One client factory is created per host:port combination (not per user), so the same factory is used to create individual clients for different users. Factory creation is resource-intensive because an initial handshake is done between the client and server, and kept in a cache for reuse.

Setting	Value	Description
com.confio.iwc.client.factory.cache.size	100	The maximum number of client factories held in the cache. The default is 100, which equates to 100 unique remote DPA instances. Increase this value if you are connecting to more than 100 remote instances.
com.confio.iwc.client.factory.cache.timeout	1800	The number of seconds a client factory remains in the cache without being used. The default is 1800 seconds, which is equal to 30 minutes.

Setting	Value	Description
com.confio.iwc.client.factory.connection.timeout	15	The number of seconds a client attempts to establish a connection before it times out. The default is 15. Zero (0) specifies that the client will continue to attempt to open a connection indefinitely.
com.confio.iwc.client.factory.read.timeout	30	The number of seconds the client waits for a response before it times out. The default is 30 seconds. Zero (0) specifies that the client will wait indefinitely.
com.confio.iwc.client.factory.enable.chunking	true false (default)	Enables or disables HTTP chunking. False is the safer option.
com.confio.iwc.client.factory.enable.log	true (default) false	Enables logging of inbound and outbound messaging to capture the web service calls. Log levels are still controlled in the `log4j.xml` file. Set this value to false to disable logging.

View and manage trusted certificates

You can view and manage the following types of trusted certificates in DPA:

- Certificates in the **DPA trust store**

  DPA can use certificates in the DPA trust store to connect to any database instance or LDAP server. These certificates are used only by DPA.

  You can view, import, and delete certificates in the DPA trust store.

- Certificates in the **Java trust store**

  DPA can use certificates in the Java trust store to connect to any database instance or LDAP server. These certificates are not managed through DPA. You can use DPA to view the alias and expiration date of these certificates.

  > The `cacerts` file that contains these certificates is included in the JDK installed with DPA. The `cacerts` file is replaced each time DPA is upgraded, and any changes to this file are overwritten.

- **DB certificates**

  A DB certificate is associated with a specific PostgreSQL or EDB Postgres database instance, and DPA uses it to connect to that instance. They are not shared with database instances that they haven't been assigned to, and they are not used by other services such as LDAP. DB certificates are used only by DPA.

  You can use DPA to view, import, and remove DB certificates.

Manage trusted certificates in the DPA trust store

DPA can use certificates in the DPA trust store to connect to any database instance or LDAP server. These certificates are used only by DPA. You can use DPA to view, import, and delete these certificates.

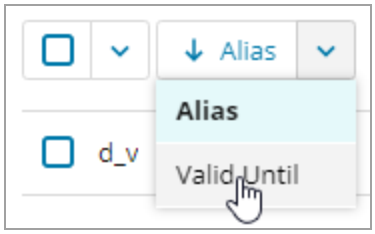
The DPA trust store is located, by default, in the *DPA-install-dir*`\iwc\tomcat\ignite_config\security` directory.

View information about certificates in the DPA trust store

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Configuration, click Trusted Certificate Management.

The Trusted Certificate Management page opens. The DPA Trust Store tab lists the alias and expiration data of each certificate in the DPA trust store. You can:

- Sort by alias or expiration date. Click the down arrow to select the attribute you want to sort by. Click the sort button to reverse the sort order.



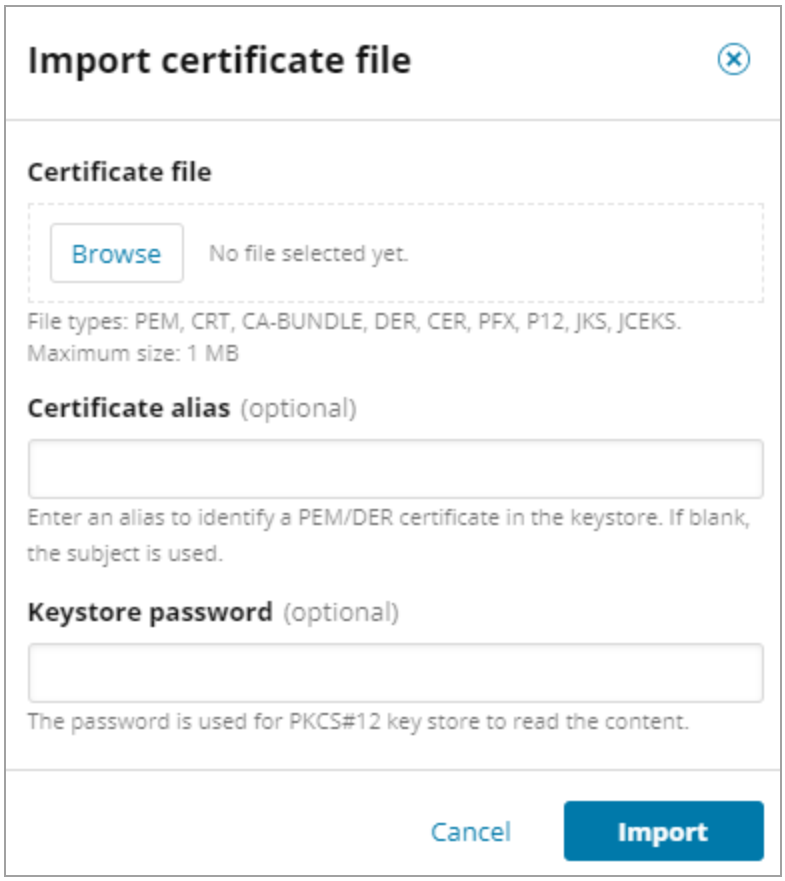
- Enter a string in the search field to search for a certificate by alias name. The list is filtered to show only certificates whose name includes the search string.

Import a certificate into the DPA trust store

You can import a file that contains a single certificate, or you can import an entire keystore.

1. Open the DPA Trust Store tab.
2. Click Import certificate(s).

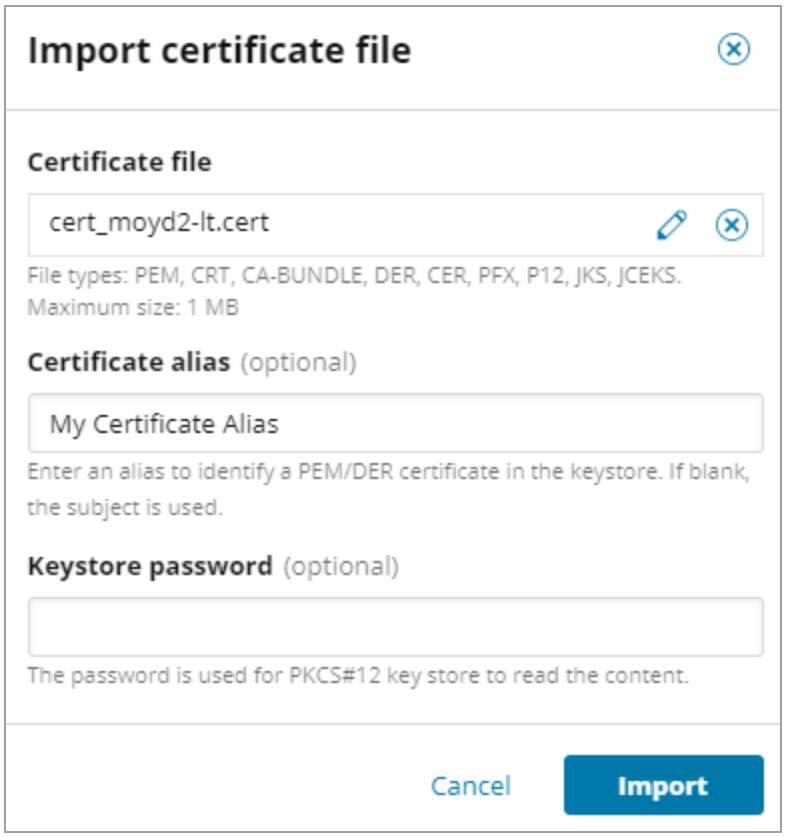
   The Import certificate file dialog box opens.



3. Click Browse and select the certificate file.
4. If you are importing a PKCS#12 file, enter the keystore password.

   If you are importing a PEM or DER file, enter a unique alias to identify this certificate.

   > If you do not enter an alias, the subject from the certificate is used. If the alias is not unique within the DPA trust store, DPA appends a number to make it unique.



5. Click Import.

   DPA displays the import status. To display the status of each certificate, click Show complete results.

Delete a certificate from the DPA trust store

You can remove certificates that are expired or no longer needed.

1. Open the DPA Trust Store tab.
2. Select one or more certificates to delete.

3. Click Delete, and then click Delete in the confirmation dialog box.

   The certificate is removed from the DPA trust store.

View trusted certificates in the Java trust store

Certificates in the Java trust store can be used by any Java application. DPA can use these certificates to connect to any database instance or LDAP server.

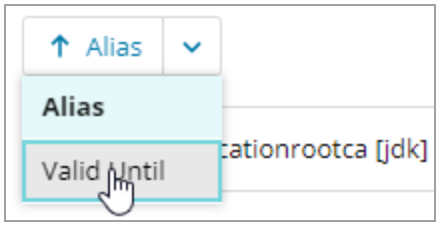
These certificates are located (by default) at *DPA-install-dir*\iwc\jre\lib\security\cacerts, and they are not managed through DPA. You can use DPA to view the names and expiration dates of these certificates.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Configuration, click Trusted Certificate Management.

   The Trusted Certificate Management page opens.
3. Click Java Trust Store.

   The Java Trust Store tab lists the alias and expiration date of each certificate in the Java trust store. You can:

   - Sort by alias or expiration date. Click the down arrow to select the attribute you want to sort by. Click the sort button to reverse the sort order.



- Enter a string in the search field to search for a certificate by alias name. The list is filtered to show only certificates whose name includes the search string.

Manage DB certificates

A DB certificate is associated with a specific PostgreSQL or EDB Postgres database instance, and DPA can use it to connect to that instance. DB certificates are used only by DPA, and they are stored in the DPA repository database.

You can use DPA to view information about DB certificates, import a DB certificate and associate it with a database instance, or remove a DB certificate.

View information about DB certificates

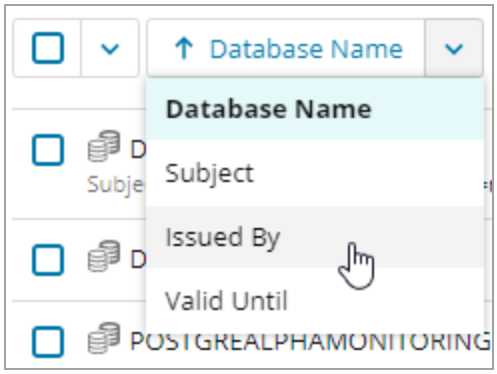
1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Configuration, click Trusted Certificate Management.

   The Trusted Certificate Management page opens.
3. Click DB Certificates.

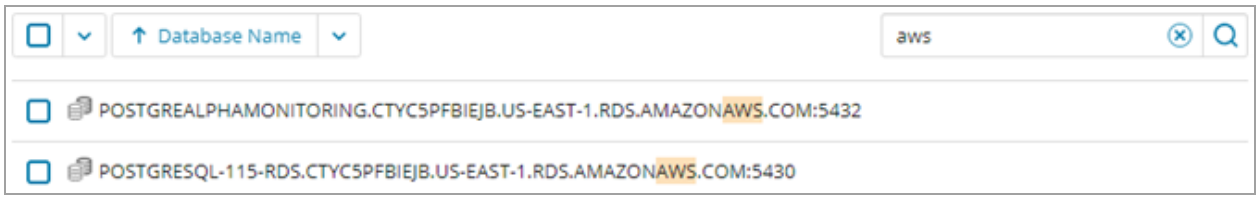
   The DB Certificates tab lists all PostgreSQL or EDB Postgres database instances monitored by DPA. Instances that are associated with a DB certificate have a certificate icon next to the name, and information about the certificate is displayed.

4. You can:

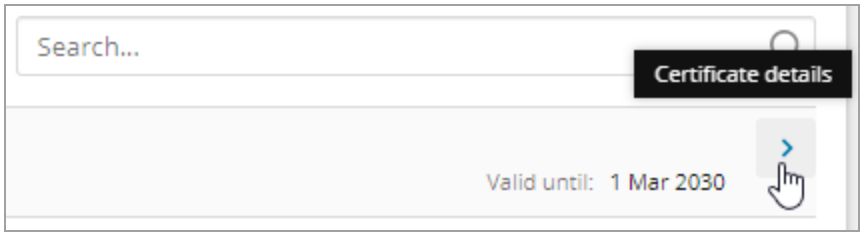
   - Sort by database instance name, subject, issuer, or expiration date. Click the down arrow to select the attribute you want to sort by. Click the sort button to reverse the sort order.



   - Enter a string in the search field to find database names, subjects, or certificates that include the string. The list is filtered to show only database instances that meet the search criteria.

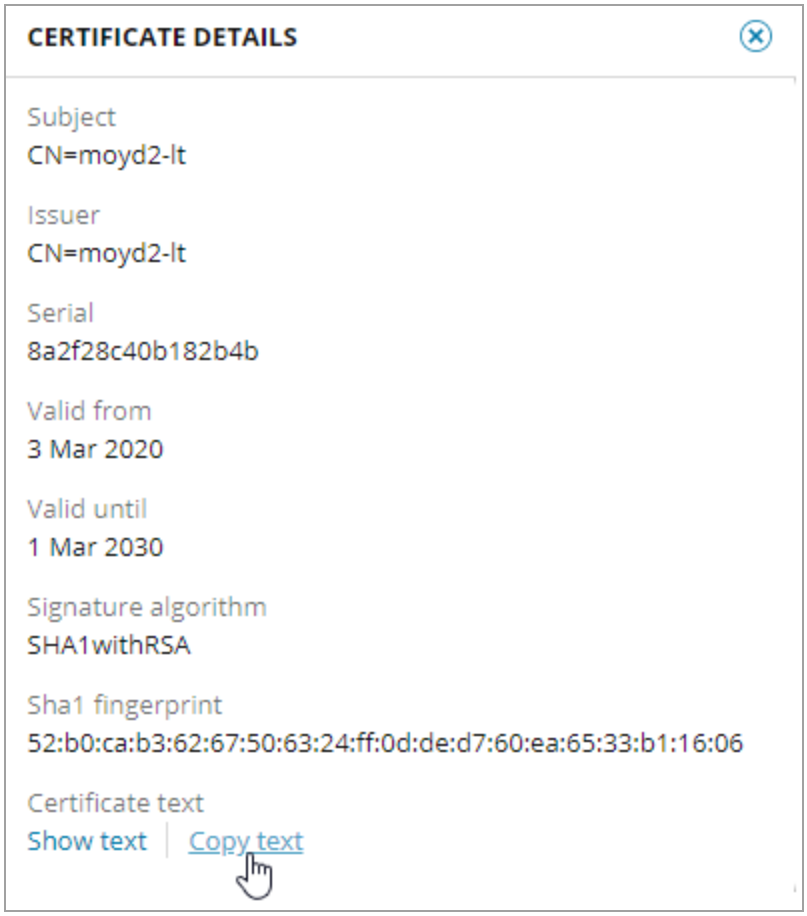


   - Click the arrow on the right to view certificate details.



     The Certificate Details panel opens. Click a link at the bottom to view or copy the

certificate text.

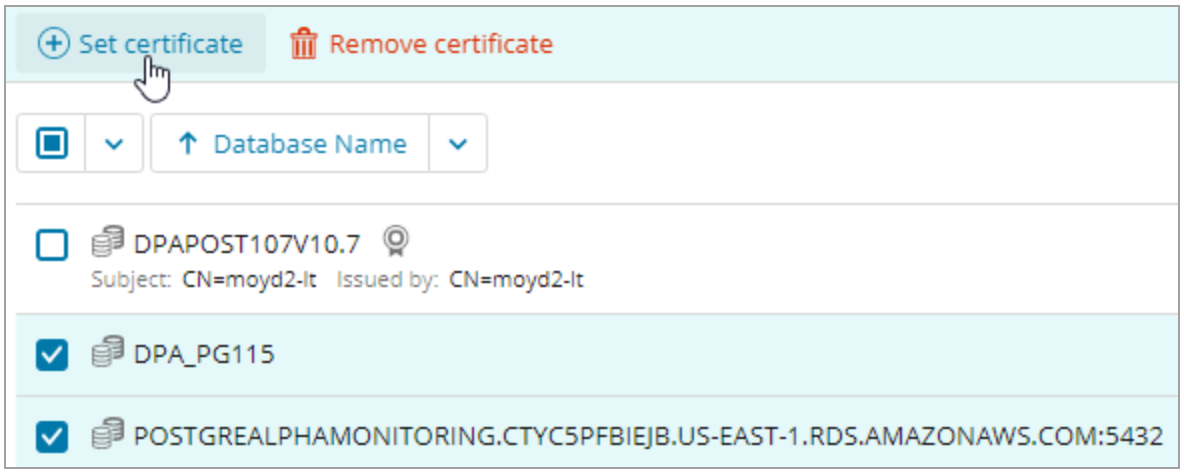


Import a DB certificate and associate it with a PostgreSQL database instance

When you import a certificate through the DB Certificates tab, the certificate is stored in the DPA repository database. It is associated with one or more PostgreSQL database instances during the import process, and DPA uses it to connect to those instances.

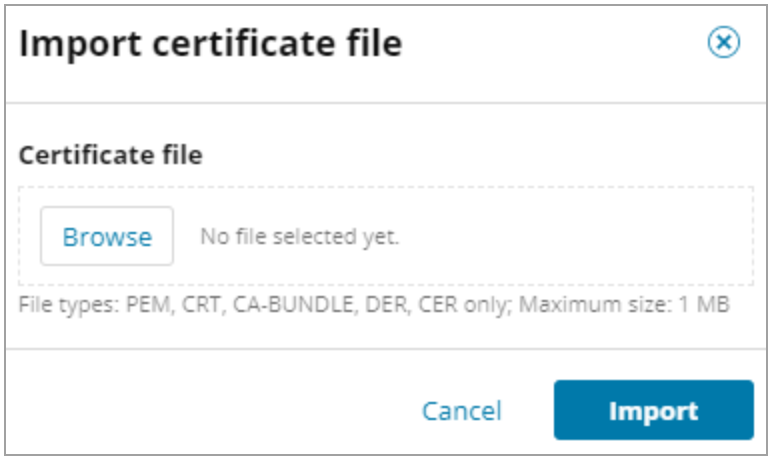
1. Open the DB Certificates tab.
2. Select the database instances you want to associate with a certificate.

   The Set certificate and Remove certificate buttons are displayed.

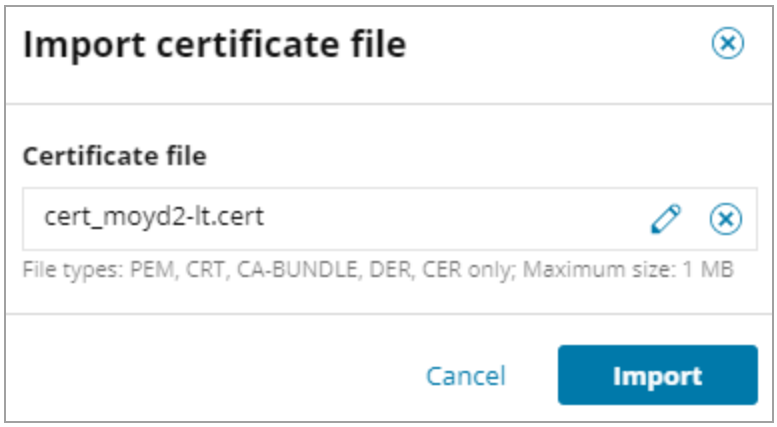


3. Click Set certificate.

   The Import certificate file dialog box opens.



4. Click Browse and select the certificate file.



5. Click Import.

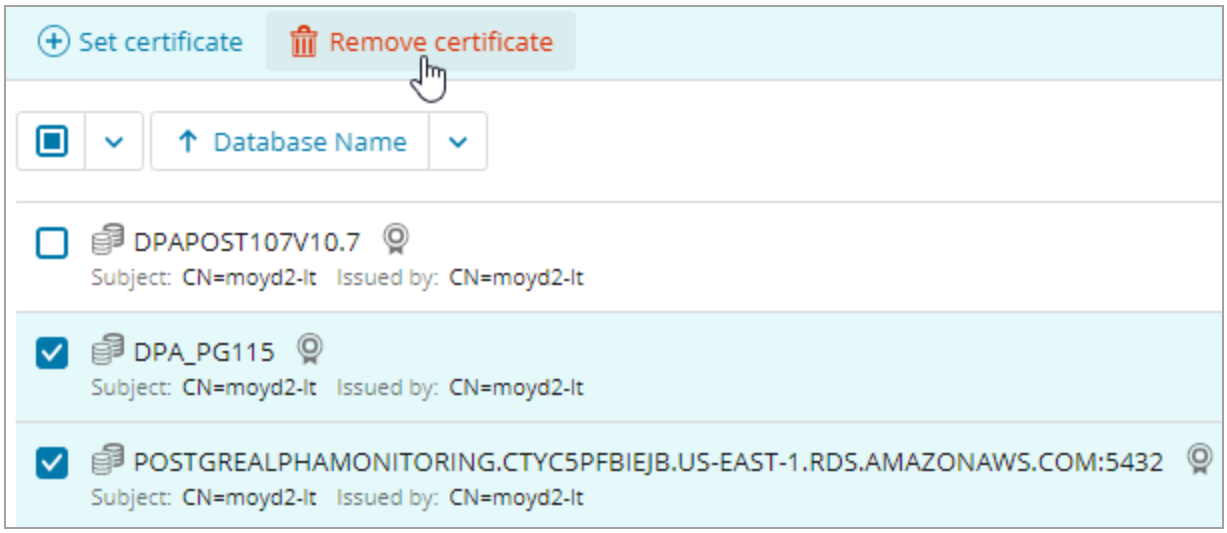
   The certificate is imported and associated with the selected database instances.

Remove a DB certificate

You can disassociate a DB certificate from a database instance and remove it from the DPA repository database. For example, you can remove a certificate when it expires.

1. Open the DB Certificates tab.
2. Select the database instances that are associated with certificates you want to remove.

   The Set certificate and Remove certificate buttons are displayed.



3. Click Remove certificate, and then click Remove on the confirmation dialog box.

   The selected database instances no longer have certificates associated with them, and the certificates are removed from the DPA repository database.

CyberArk integration

If your company uses CyberArk to store login credentials, you can configure DPA to authenticate using credentials stored in CyberArk.

To create a new repository that uses credentials stored in CyberArk, see Create a DPA repository that uses CyberArk for authentication.

Configure DPA to use credentials stored in CyberArk

DPA can use credentials stored in CyberArk for authentication to:

- The DPA repository
- Monitored database instances
- Monitored VMware ESX/ESXi Hosts or vCenter Servers
- LDAP/AD servers
- Mail servers

You can configure DPA to use CyberArk credentials globally, for authentication to all of the entities listed above. Or you can configure DPA to use CyberArk credentials for certain types of authentication. For example, DPA could use CyberArk credentials for authentication to the DPA repository and monitored instances, but not to VMware, LDAP/AD, or mail servers.

DPA uses the REST API to call the CyberArk Central Credential Provider (CCP) and client certificates authentication to authenticate to it, which means that your CyberArk CCP must be configured to use SSL.

Complete the following tasks to configure DPA to use CyberArk to as a credential provider.

Gather the required information

To configure CyberArk integration in DPA, you must have the following information:

- The URL of the CyberArk Central Credential Provider
- The client certificate that will be used to authenticate to the CyberArk CCP
- The CA root certificate of CyberArk CCP server if you are using a private Certification Authority to create trust between DPA and CyberArk CCP

If your client certificate is in PEM format, convert it to PKCS12 format

DPA does not support PEM format. If your client certificate is in PEM format, complete the following steps to convert it to PKCS12 format.

1. Download and install the OpenSSL toolkit. For Windows you can download it from here.
2. Open a command prompt and run the following command, where *yourClientCertificate*.pem is your client certificate file in PEM format.

   ```
   openssl pkcs12 -export -inkey yourClientCertificate.pem -in yourClientCertificate.pem -out CA_keystore.p12
   ```

3. When prompted, enter the password protecting the PEM file (if there is one).
4. When prompted, set a password for the new PKCS12 file. This is the password you will use for the `keystore.password` and `key.password` properties in the `cyberark.properties` file.

   OpenSSL uses the same password for both the keystore and the key itself.

A new file called `CA_keystore.p12` is created. It stores the converted client certificate in PKCS12 format. Specify this file as the `keystore.location` property in the `cyberark.properties` file.

Configure the cyberark.properties file

1. Open the following file in a text editor:

   DPA-install-dir`\iwc\tomcat\ignite_config\idc\cyberark.properties`

2. Enter the values for the following properties, and save the file.

base.uri

The common part of requests to get credentials from the CyberArk CCP. All requests made by DPA to the CCP consist of this URL concatenated with a suffix specific to the given connection. For example, if your CCP URL is `https://cyberark.prod.com` and your AppID for DPA is `DPA`, the `base.uri` is specified as:

```
base.uri=https://cyberark.prod.com/AIMWebService/api/Accounts?AppID=DPA
```

When you register database instances for monitoring (for example), you will use the suffix query that is specific to the monitored instance. This type of query might look like this:

```
&safe=mondb&object=sqlserverprod1
```

In this example, DPA would use the following URL to get the credentials for this monitored instance:

```
https://cyberark.prod.com/AIMWebService/api/Accounts?AppID=DPA&safe=mondb&object=sqlserverprod1
```

DPA provides flexibility for structuring your CCP requests. You can use any field documented by CyberArk.

request.timeout.millis

The connection timeout of the CCP requests in milliseconds. If this value is not specified, the default value of 30000 (30 seconds) is used. If you are using the `ConnectionTimeout` property as part of the CCP URL, the lower value of the two properties is used.

disable.hostname.verification

Disables hostname verification. If set to `true`, DPA does **not** verify that the CyberArk CCP server certificate matches the CyberArk CCP hostname. DPA recommends the default of `false` (hostname verification is enabled).

keystore.location

The path to the keystore that holds your client certificate used to authenticate DPA requests to the CyberArk CCP. Be aware that if your location string contains a backslash (`\`) you must escape it with another backslash, as shown in the following example:

```
keystore.location= C:\\security\\ca_keystore.p12
```

keystore.type

The type of keystore. Valid values are `jks` and `pkcs12`.

DPA supports the following file types:

- PKCS12: The file usually has a `.p12` extension.
- JKS: The file usually has a `.jks` extension.

If your file is in a **different** format, take one of the following actions:

- PEM: If your client certificate is in PEM format, convert it to PKCS12 format.
- PXF: A file in PXF format can be used as-is, because the PXF and PKCS12 formats are generally interchangeable. Enter `pkcs12` as the `keystore.type`.

keystore.password

The password to the keystore. After DPA starts, this value is encrypted based on the settings specified in DPA password encryption settings.

key.password

The password to the client certificate in the keystore. After DPA starts, this value is encrypted based on the settings specified in DPA password encryption settings.

Establish trust between DPA, the CyberArk CCP, and the client certificate

Because your CyberArk deployment is using SSL, you must ensure that DPA is configured to trust the CyberArk CCP server certificate. To do this, import the CA certificate of the CyberArk CCP into the DPA trust store.

Enable auto-invalidation of monitored instance connections when credentials have changed

DPA polls credentials from CyberArk every 10 minutes (by default). If new credentials are detected, all connections created after the change will respect the new credentials.

If CyberArk stops providing the credentials (for example, because the account name in CyberArk changed so the query is no longer valid, or CyberArk is down for some reason), DPA will postpone the connection invalidations to help you overcome short CyberArk outages. By default, the delay in connection invalidations is 30 minutes. You can configure this value for every connection type by setting following properties in the `system.properties` file, with the value in minutes:

- For the repository connection:

  ```
  com.solarwinds.dpa.cyberark.checker.repo.periodicClearCredentialsMinutes=30
  ```

- For monitored database instance connections:

  ```
  com.solarwinds.dpa.cyberark.checker.periodicClearCredentialsMinutes=30
  ```

- For monitored VMware ESX/ESXi Host or vCenter Server connections:

  ```
  com.solarwinds.dpa.cyberark.checker.vsphere.periodicClearCredentialsMinutes
  =30
  ```

- For an LDAP/AD server connection:

  ```
  com.solarwinds.dpa.cyberark.checker.ldap.periodicClearCredentialsMinutes=30
  ```

- For a mail server connection:

  ```
  com.solarwinds.dpa.cyberark.checker.mail.periodicClearCredentialsMinutes=30
  ```

A value of 30 minutes means that DPA will invalidate existing connections if CyberArk is not able to provide credentials for a period longer than 30 minutes.

To reschedule the DPA credentials polling job (which runs every 10 minutes by default), set the following properties in the `system.properties` file, with the value in milliseconds:

- For the repository connection:

  ```
  com.solarwinds.dpa.cyberark.checker.repo.periodicIntervalMillis=600000
  ```

- For monitored database instance connections:

  `com.solarwinds.dpa.cyberark.checker.periodicIntervalMillis=600000`

- For monitored VMware ESX/ESXi Host or vCenter Server connections:

  `com.solarwinds.dpa.cyberark.checker.vsphere.periodicIntervalMillis=600000`

- For an LDAP/AD server connection:

  `com.solarwinds.dpa.cyberark.checker.ldap.periodicIntervalMillis=600000`

- For a mail server connection:

  `com.solarwinds.dpa.cyberark.checker.mail.periodicIntervalMillis=600000`

Enter connection credentials into CyberArk and test them

Add all connection credentials used by DPA into CyberArk. SolarWinds recommends using the DPA REST API to test the connection queries before you enable the integration.

Domain names

If a domain needs to be specified during authentication, DPA uses the value in either the `LogonDomain` field or the `UserName` field of the CyberArk object:

- If the `UserName` field includes a backslash (`\`), DPA uses the value in that field as the `DOMAIN_NAME\Username`. (The `LogonDomain` field is ignored.) For example, if the CyberArk object includes:

  ```
  "UserName": "MYDOMAIN\MyUserName",
  ```

  DPA uses `MYDOMAIN\MyUserName` for authentication.

- If the `LogonDomain` field contains a value and the `UserName` field does **not** contain a backslash, DPA prepends the domain name and a backslash to the user name. For example, if the CyberArk object includes:

  ```
  "LogonDomain": "MYDOMAIN",
  "UserName": "MyUserName",
  ```

  DPA prepends the `LogonDomain` value to the `UserName` value, and the value used for authentication is `MYDOMAIN\MyUserName`.

Enable the DPA CyberArk integration

After you have edited the `cyberark.properties` file and updated the DPA trust store with the required certificates, complete the following steps to enable CyberArk mode so that DPA will start to use those settings.

1. Open the following file in a text editor:

   DPA-install-dir\iwc\tomcat\ignite_config\idc\system.properties

2. Uncomment or add one of the following, and save the file:

   - If **all** credentials that DPA uses for authentication are stored in CyberArk, uncomment or add the following line:

     ```
     com.solarwinds.dpa.credentials.provider.type=CYBERARK_REMOTE
     ```

     > The general property `com.solarwinds.dpa.credentials.provider.type` is the default and includes all credential types. If you include both `com.solarwinds.dpa.credentials.provider.type` and one or more of the type-specific properties listed below, the value for the type-specific property takes precedence.

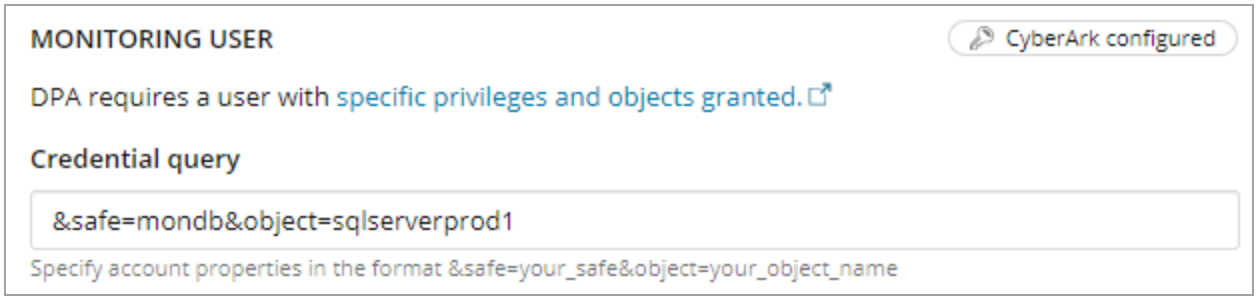
   - If only some of the credentials that DPA uses are stored in CyberArk, uncomment or add the following lines as needed. Enter a value of either `DPA` or `CYBERARK_REMOTE` to specify where the credentials are stored:

     ```
     com.solarwinds.dpa.credentials.provider.type.database = [DPA ||
     CYBERARK_REMOTE]

     com.solarwinds.dpa.credentials.provider.type.repository = [DPA ||
     CYBERARK_REMOTE]

     com.solarwinds.dpa.credentials.provider.type.vsphere = [DPA ||
     CYBERARK_REMOTE]

     com.solarwinds.dpa.credentials.provider.type.mail = [DPA || CYBERARK_
     REMOTE]

     com.solarwinds.dpa.credentials.provider.type.ldap = [DPA || CYBERARK_
     REMOTE]
     ```

3. Restart DPA.

   After the restart, the affected DPA wizards and other interfaces display a field for the CyberArk

credentials query instead of the user name and password fields.



Update an existing DPA installation to use credentials stored in CyberArk

If any of the following were configured before the CyberArk integration, you must update the configurations to use credentials stored in CyberArk:

- The DPA repository database
- Monitored database instances
- Monitored VMware ESX/ESXi Hosts or vCenter Servers
- AD or LDAP server
- Your company mail server that is used to send emails from DPA

For new installations without a repository database, see Create a DPA repository that uses CyberArk for authentication.

1. To configure the DPA repository database to use credentials stored in CyberArk:
    a. Open the following file in a text editor:

    `DPA-install-dir\iwc\tomcat\ignite_config\idc\repo.properties`

    b. Remove the `repo.user` and `repo.password` properties.
    c. Add the following setting, where *`cyberArkQueryForRepoCredentials`* is the CyberArk query that DPA will use to obtain the credentials to connect to the repository database:

    `repo.cyberarkQuery=cyberArkQueryForRepoCredentials`

    For example:

    `repo.cyberarkQuery=&safe=mondb&object=dparepo`

    d. Save the file.
    e. Restart DPA for the changes to take effect.

2. For each monitored database instance, use the Update Connection Wizard to enter the CyberArk credentials query. Or use the cyberArkQuery parameter in the DPA REST API to update existing monitored database instances.
3. For each monitored VMware ESX/ESXi Host or vCenter Server, open the Update VMware Connection page and enter the CyberArk credentials query.
4. If DPA uses your company's mail server to send email, update the mail server configuration and enter the CyberArk credentials query.

   If DPA uses the default mail server or the embedded mail server, no changes are needed.

5. If DPA is configured to use AD or LDAP user authentication, use the Configure AD/LDAP wizard to enter the CyberArk credentials query. Then restart DPA for the changes to take effect.

Register additional database instances for monitoring

When DPA is started in CyberArk mode, you must specify the CyberArk query instead of entering a user name and password when you register monitored database instances. You can use the Register Database Instance wizard, mass registration, or the REST API to register instances:

- To use the Register Database Instance wizard, follow the instructions in one of the following topics for creating the monitoring user and completing the wizard:
  - Register an Oracle database instance
  - Register a SQL Server database instance
  - Register a Sybase database instance
  - Register a Db2 database instance
  - Register a MySQL, Percona, or Maria database instance
  - Register a PostgreSQL database instance and prepare for monitoring
  - Register an Amazon RDS for Oracle database instance
  - Register an Amazon RDS for SQL Server database instance
  - Register an Amazon RDS for MySQL or MySQL-compatible Aurora database instance
  - Register an Azure SQL database
  - Register an Azure SQL Managed Instance
- To use mass registration or the REST API:
  1. Follow the instructions in one of the topics listed above to create the monitoring user.
  2. Access the REST API, or open the Mass registration page and choose All database types.
  3. See the following table for API parameters or registration file template values.

If you use mass registration or the REST API, enter the required values in the mass registration form or in the REST API call. Values specific to CyberArk implementation are described in the following table.

Mass registration field	REST API parameter	Value
Monitoring User Monitoring User Password Privileged User Privileged User Password SYS Password	monitoringUser monitoringUserPassword sysAdminUser sysAdminPassword sysPassword	Do **not** enter a value.
CyberArk query	cyberArkQuery	Enter information that can be concatenated with the `base.uri` value from the `cyberark.properties` file to create a valid request to the CyberArk CCP. For example: `&safe=mondb&object=oracleprod1` DPA uses values in the `UserName` field, the `Content` field, and (in some cases) the `LogonDomain` field from the CyberArk response as credentials for monitoring the database.
Create Monitoring User	monitoringUserIsNew	Enter `N` on the mass registration form. Enter `false` as the REST API parameter value.

The following example shows a REST API request to register Oracle instance:

```
{    "databaseType": "ORACLE",
     "serverName": "127.0.0.1",
     "serviceNameOrSID": "DPA_ORA11R1",
     "port": "1521",
     "monitoringUserIsNew": false,
     "displayName": "DPA_ORA11R1",
     "cyberArkQuery: "&safe=mondb&object=oracleprod1"
}
```

Secure your CyberArk integration

SolarWinds strongly recommends securing information used to authenticate to CyberArk.

Use a strong encryption algorithm and passphrase

Password fields in `cyberark.properties` are automatically encrypted after DPA startup. To use a strong encryption algorithm and passphrase:

1. Open DPA in your browser.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Administration > Configuration, click Password Encryption Settings.
4. Under Choose an encryption scheme, select AES 256-bit.
5. Enter your Encryption Passphrase.

   ⚠ Do not share the passphrase with anyone.
6. Click Update.

   Passwords stored in DPA configuration files or in the DPA repository are re-encrypted using your selection.

Limit access to the files storing sensitive information

Make sure that, except for the user running DPA service (Windows) or process (Linux), no one has read permission to the CyberArk configuration file (`cyberark.properties`) and CyberArk client certificate (referenced from `cyberark.properties`).

For the user running the DPA service or process, read permission is sufficient for the CyberArk client certificate. Both read and write permissions are needed for the `cyberark.properties` file (so DPA can encrypt the password fields).

When you install DPA on a Linux server, make sure you do not install DPA under the root user. Using a dedicated DPA user instead of the root user gives you flexibility to set the permissions as described above.

When you install DPA on a Windows server, the Ignite_PI service runs under the "Local System Account" by default. To use a dedicated DPA user instead, complete the following steps:

1. Open a command prompt, and enter `services.msc`.
2. Right-click the Ignite PI Server service and select Properties.
3. On the Log On tab, click This Account and enter the credentials for the account you want to use.

Be aware that after you upgrade DPA, the Ignite_PI service is recreated and "Local System Account" is used again. You must manually update the service after every DPA upgrade.

Troubleshoot the CyberArk integration

The following situations can occur when DPA is not properly configured to use CyberArk. Any misconfiguration of CyberArk should not prevent DPA from starting.

I am not able to create the repository, register a database or VM instance, or integrate the LDAP/AD or mail server

When you attempt to perform any of these actions, you see the following message:

```
Failed to retrieve credentials. Error sending the credential request.
```

To find the root cause of the problem, open *DPA_install_dir*`\iwc\tomcat\logs\errors.log` and search for any of the errors described in CyberArk configuration errors. See that section for steps to resolve the issue.

Monitoring does not start for some instances, but it does for others

The `cyberark.properties` file is configured correctly, but there is a mistake in the CyberArk query you provided when you registered the database instances. The CyberArk query is always validated during registration, but a change might have been made to your CyberArk CPP so the path has changed.

For each instance that is not monitored, use the Update Connection Wizard to correct the CyberArk query.

Monitoring does not start for any instances

The `cyberark.properties` file is probably not configured correctly. Open *DPA_install_dir*`\iwc\tomcat\logs\errors.log` and search for any of the errors described in CyberArk configuration errors. See that section for steps to resolve the issue.

CyberArk configuration errors

This section explains the most common CyberArk configuration errors that can appear in the *DPA_install_dir*`\iwc\tomcat\logs\errors.log` file.

- `PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target`

  Problem: The CyberArk CC sever certificate is not trusted by DPA.

  Resolution: Your CyberArk CC sever certificate is not signed by a public CA. Therefore you must import the CA certificate of the CyberArk CCP into the DPA trust store.

- `The keystore could not be loaded: keystore password was incorrect`

  Problem: The password to the keystore holding your private key to authenticate to CyberArk is incorrect.

  Resolution:

  1. Open the following file in a text editor:

     DPA_install_dir`\iwc\tomcat\ignite_config\idc\cyberark.properties`

  2. Change the value of the `keystore.password` property to the correct keystore password, and save the file.
  3. Restart DPA. The password is encrypted when DPA restarts.

- `Error initializing the key manager factory: Get Key failed: Given final block not properly padded. Such issues can arise if a bad key is used during decryption.`

  Problem: The password to the private key to authenticate to CyberArk is incorrect.

  Resolution:

  1. Open the following file in a text editor:

     DPA_install_dir`\iwc\tomcat\ignite_config\idc\cyberark.properties`

  2. Change the value of the `key.password` property to the correct key password, and save the file.
  3. Restart DPA. The password is encrypted when DPA restarts.

- `The keystore file could not be opened: … (The system cannot find the file specified)`

  Problem: DPA cannot find the keystore to authenticate to CyberArk.

  Resolution:

1. Open the following file in a text editor:

   DPA_install_dir\iwc\tomcat\ignite_config\idc\cyberark.properties

2. Change the value of the `keystore.location` property to the correct location of the keystore, and save the file.

- `Failed to retrieve credentials for INSTANCE_NAME using request 'CYBERARK_QUERY' … UnknownHostException: No such host is known (cyberark-vaultl.ignite.local)`

  Problem: The CyberArk CCP host name is incorrect.

  Resolution:

  1. Open the following file in a text editor:

     DPA_install_dir\iwc\tomcat\ignite_config\idc\cyberark.properties

  2. Update the value of the `base.uri` property with the correct CyberArk CCP host name, and save the file.

- `Failed to retrieve credentials for INSTANCE_NAME using request 'CYBERARK_QUERY' … Password object matching query [CYBERARK_QUERY] was not found`

  Problem: The path to the CyberArk CCP to get the credentials is incorrect.

  Resolution: Use the Update Connection Wizard to correct the CyberArk query for the affected instance.

- `ERROR (2022-06-01T18:07:25,740-0400) [localhost-startStop-1] SslServiceImpl:147 - Failed to create empty custom DPA trust store.`

  `java.io.IOException: keystore password was incorrect`

  Problem: The keystore for the custom DPA trust store is incorrect.

  Resolution:

  1. Open the following file in a text editor:

     DPA_install_dir\iwc\tomcat\ignite_config\idc\system.properties

  2. Change the value of the `com.confio.security.trustStorePassword=` property to the correct keystore password, and save the file.

     > Make sure that `com.confio.security.trustStore` is pointed to the correct keystore file. By default, it should be pointed to `.keystore`.

  3. Restart DPA. The password is encrypted when DPA restarts.

Revert the CyberArk integration

If you have configured DPA to use credentials stored in CyberArk, complete the following steps if you need to revert the integration and have DPA manage the credentials instead.

1. Delete all the CyberArk-related certificates from the DPA trust store.
2. Change the credential provider type:
   a. Open the following file in a text editor:

      DPA-install-dir\iwc\tomcat\ignite_config\idc\system.properties

   b. Change the value of the following properties (if they exist) from `CYBERARK_REMOTE` to `DPA`, and save the file:

      ```
      com.solarwinds.dpa.credentials.provider.type=DPA

      com.solarwinds.dpa.credentials.provider.type.database=DPA

      com.solarwinds.dpa.credentials.provider.type.repository=DPA

      com.solarwinds.dpa.credentials.provider.type.vsphere=DPA

      com.solarwinds.dpa.credentials.provider.type.mail=DPA

      com.solarwinds.dpa.credentials.provider.type.ldap=DPA
      ```

3. Remove the keystore containing the CyberArk client certificate. The location of the keystore is in the following file:

   DPA-install-dir\iwc\tomcat\ignite_config\idc\cyberark.properties

4. Delete the `cyberark.properties` file.
5. Restart DPA.

   After the restart, DPA wizards and other interfaces display fields for credentials instead of the CyberArk credentials query.

6. For each monitored database instance, use the Update Connection Wizard to enter the user name and password for the DPA monitoring user.
7. For each monitored VMware ESX/ESXi Host or vCenter Server, open the Update VMware Connection page and update the credentials.

8. To stop using CyberArk credentials for the DPA repository database:
    a. Open the following file in a text editor:

    `DPA-install-dir\iwc\tomcat\ignite_config\idc\repo.properties`

    b. Remove the line that starts with `repo.cyberarkQuery`,
    c. Add or uncomment the `repo.user` and `repo.password` properties.
    d. Save the file.
    e. Restart DPA for the changes to take effect.
9. If DPA uses your company's mail server to send email, update the user name and password required to access the mail server.

    ⓘ If DPA uses the default mail server or the embedded mail server, no changes are needed.
10. If DPA is configured to use AD or LDAP user authentication, use the Configure AD/LDAP wizard to update the user name and password that DPA uses to query the directory for users and groups. Then restart DPA for the changes to take effect.

Automate tasks with the DPA REST API

Use the DPA REST API to securely connect to the DPA server and issue commands. DPA API calls can retrieve information and automate management tasks, such as registering database instances, stopping and starting monitors, adding annotations, and allocating licenses.

Manage tokens used for authentication to the DPA API

About refresh tokens and access tokens

Token-based authentication allows users to access the API without entering credentials to authenticate every request. Two types of tokens are required to authenticate requests to the DPA API:

- An **access token** is a secure string required to authenticate requests to access the API.

  By default, an access token expires after 900 seconds. (You can change the default lifespan by editing the advanced option `API_ACCESS_TOKEN_EXPIRATION`.) Access tokens have short lifespans for security reasons. The short lifespan makes it less likely for the access token to be accessed by a malicious actor.

  > Access tokens also expire if the DPA server is rebooted, or if the issuing refresh token expires or is deleted.

- A **refresh token** is used to issue access tokens. Refresh tokens typically have long lifespans. They are generated in DPA and stored in a secure location. Before a script or application calls the API, it uses a refresh token to obtain an access token. If the access token expires before the activity is complete, the refresh token can automatically request a new access token.

To obtain the tokens that are required to make calls to the API:

1. An administrator creates a refresh token through the DPA interface and stores it in a secure location.

   When you create a refresh token, you can specify the expiration date or set it to never expire. The default expiration date is 90 days from the creation date. You can change the default by editing the advanced option `API_REFRESH_TOKEN_EXPIRATION`.

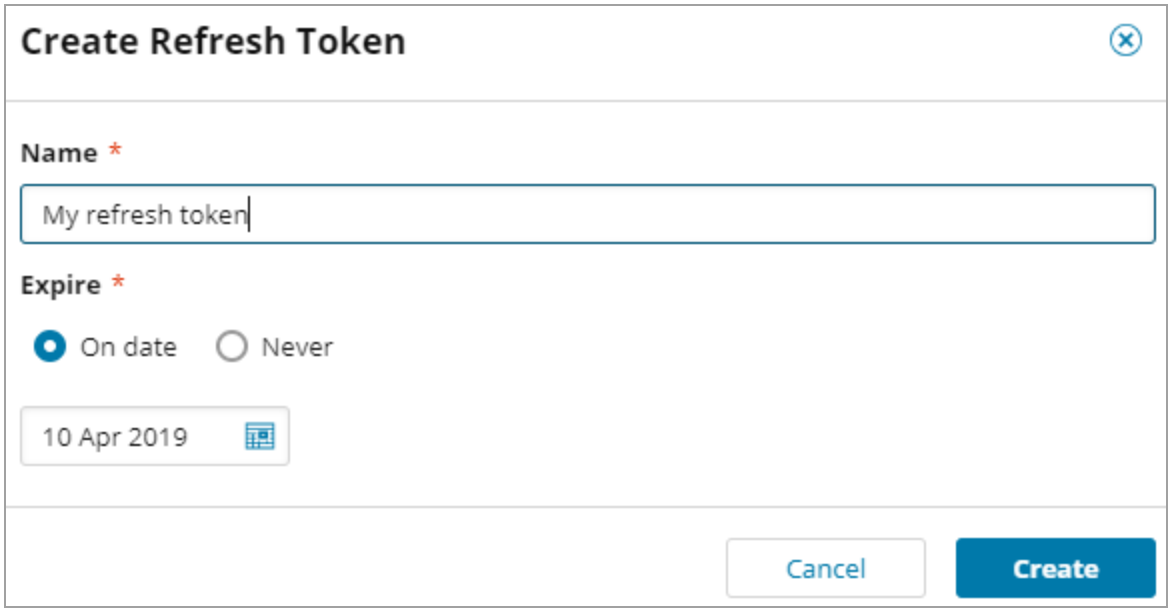
2. Before a script or application makes calls to the API, it uses the refresh token to obtain an access token. The script or code can also include a function to verify that the access token is still valid and acquire a new access token if necessary.

   The Python and PowerShell script examples show how to use the refresh token to obtain an access token.

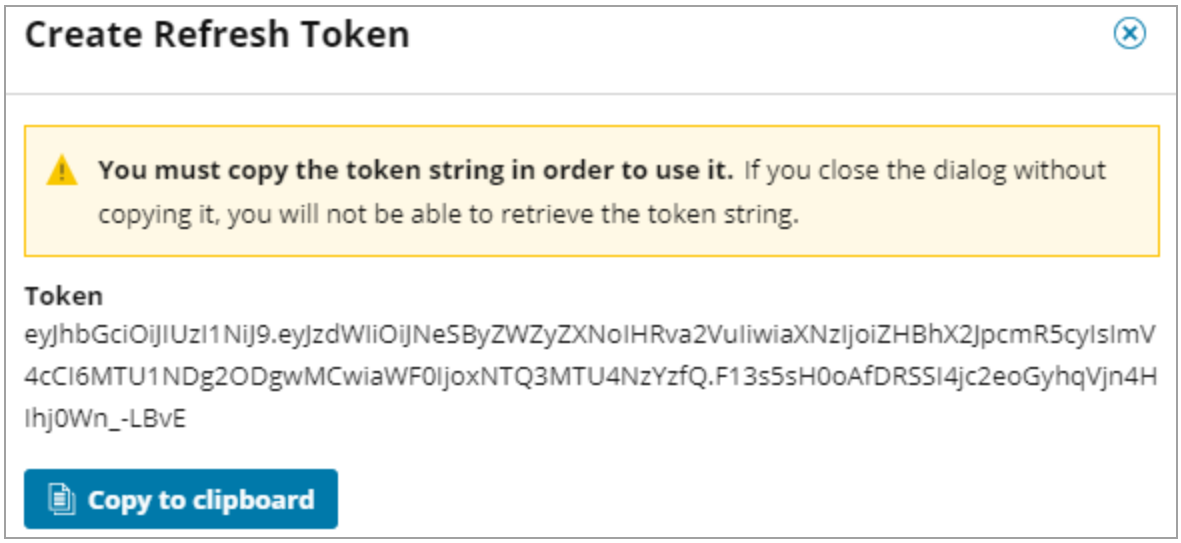
Create a refresh token

1. Log in to DPA as a user with administrative privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Users & Contacts, click Refresh Token Management.
4. On the API Refresh Token Management page, click Create token.
5. Enter a name and specify when the token expires.

   By default, refresh tokens for the DPA API expire after 90 days. However, you can choose to create refresh tokens that never expire.



6. Click Create. The token string is displayed.



7. Click Copy to clipboard, and then click Close.

If you create a refresh token and fail to copy the string or lose the copied string, the refresh token cannot be used. Delete that token and create a new one.

About storing refresh tokens

Store refresh tokens in a secure location, such as a password-protected file system or an encrypted database. Limit access to users who need the tokens to make API calls.

If you believe that a refresh token has been accessed by an unauthorized user, delete it and create a new one.

Delete a refresh token

You can delete a refresh token at any time. For example, you can delete refresh tokens that have expired. If you delete a refresh token that has not expired, any access tokens obtained using that refresh token are invalidated and can no longer be used.

1. Log in to DPA as a user with administrative privileges.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Users & Contacts, click Refresh Token Management.
4. On the API Refresh Token Management page, select one or more tokens.

5. Click Delete.

Learn about and experiment with the DPA API

The DPA API is documented in the Swagger interface. Use this interactive interface to explore the available API endpoints and try out API calls.

You can use scripts to call the API outside of the Swagger interface. See examples of Python and PowerShell scripts that call the DPA API.

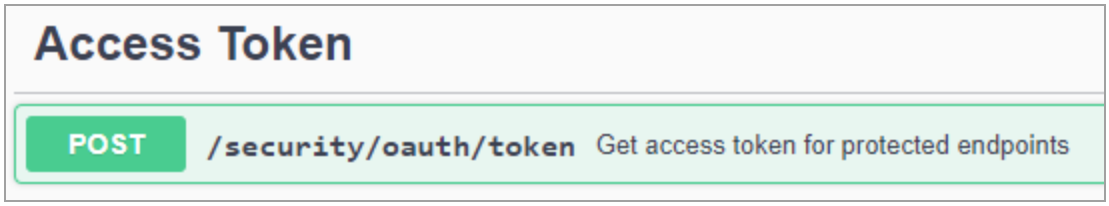
Access the DPA API documentation and get authorization to make API calls

You can access the URL and review the DPA API documentation without being authorized to make API calls. However, an access token is required to make API calls. Complete the following steps to access the API documentation and authenticate with an access token.

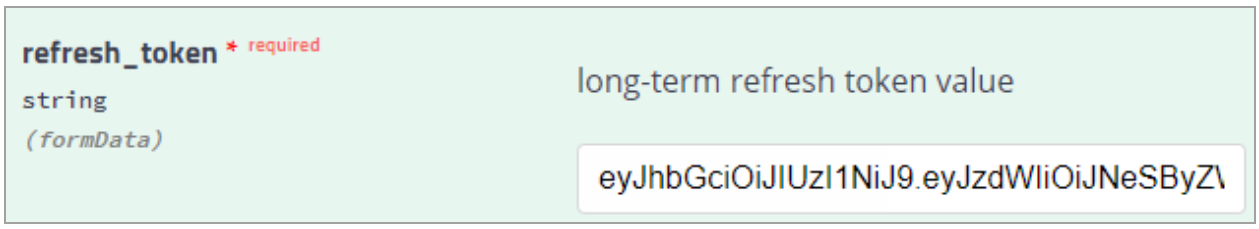
1. Create a new refresh token and copy it to the clipboard, or copy an existing refresh token that your organization has stored in a secure location.
2. From the DPA menu in the upper-right corner, click Options.
3. Under Support > Utilities, click Management API Documentation.

   The Swagger interface opens.

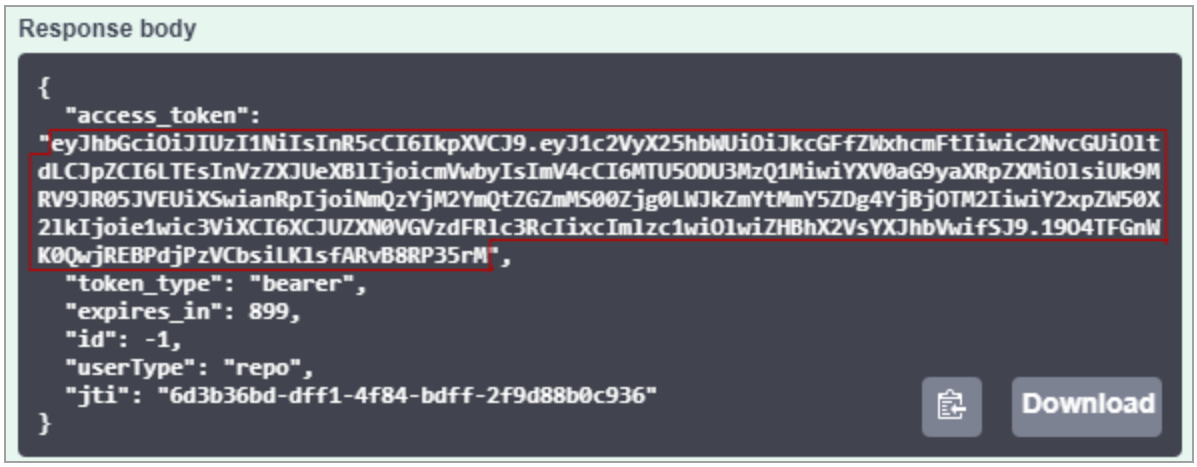
4. Use the refresh token to obtain an access token:
   a. Click Access Token to expand it.



   b. Click Post to expand that section.
   c. Click the Try it out button.
   d. Paste the refresh token value you copied in step 1 into the refresh_token box.

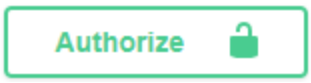


   e. Click Execute.
   f. Copy the access token within the quotation marks in the Response body. (Do not include the quotation marks.)

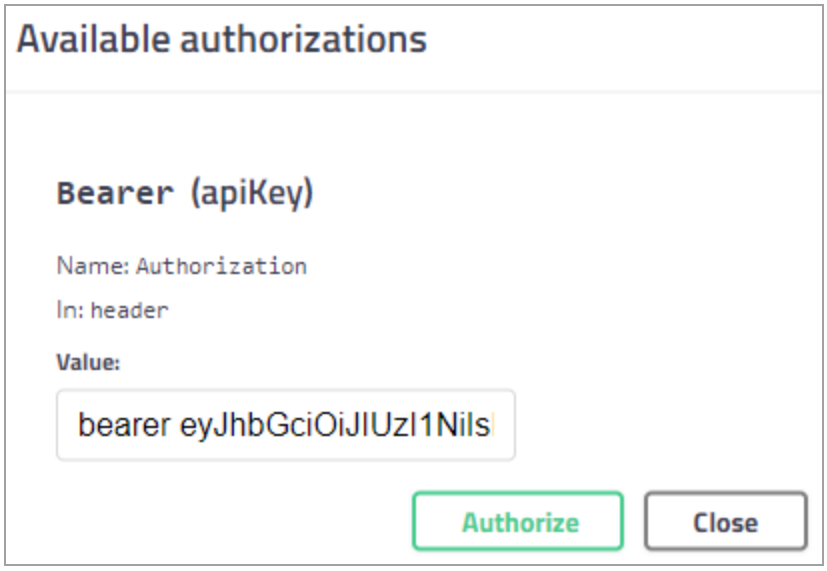


5. Authenticate with the access token:

a. In the upper-right corner, click Authorize.

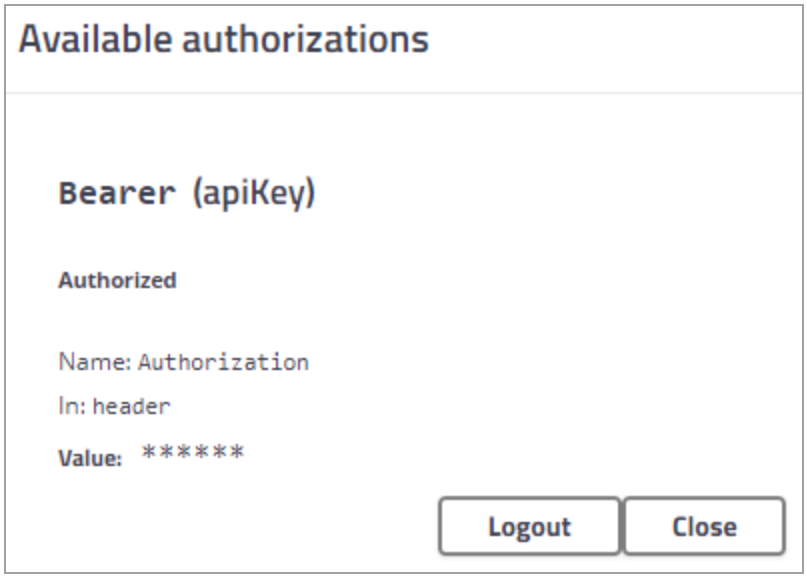


b. In the Available authorizations dialog, type `bearer` followed by a space, and then paste the access token.



c. Click Authorize.

If the authorization is successful, the following dialog is displayed.



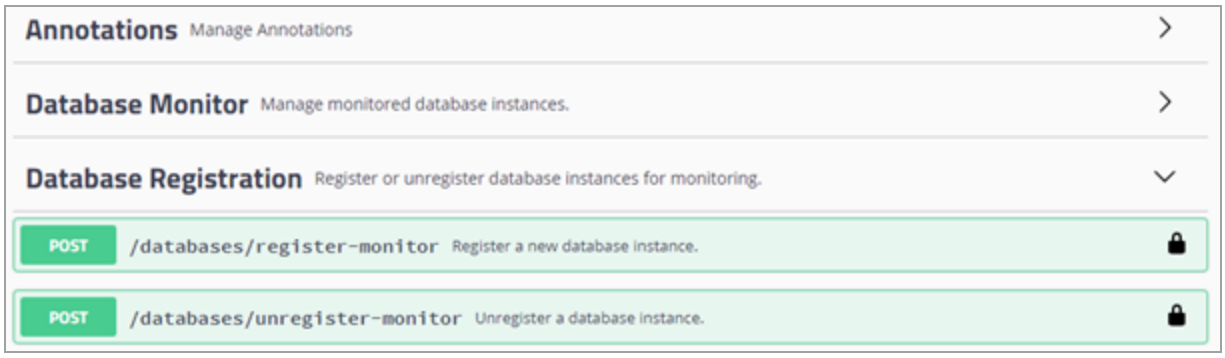
d. Click Close.

You can now use the Swagger interface to learn about and execute the available API commands.

View the DPA API documentation

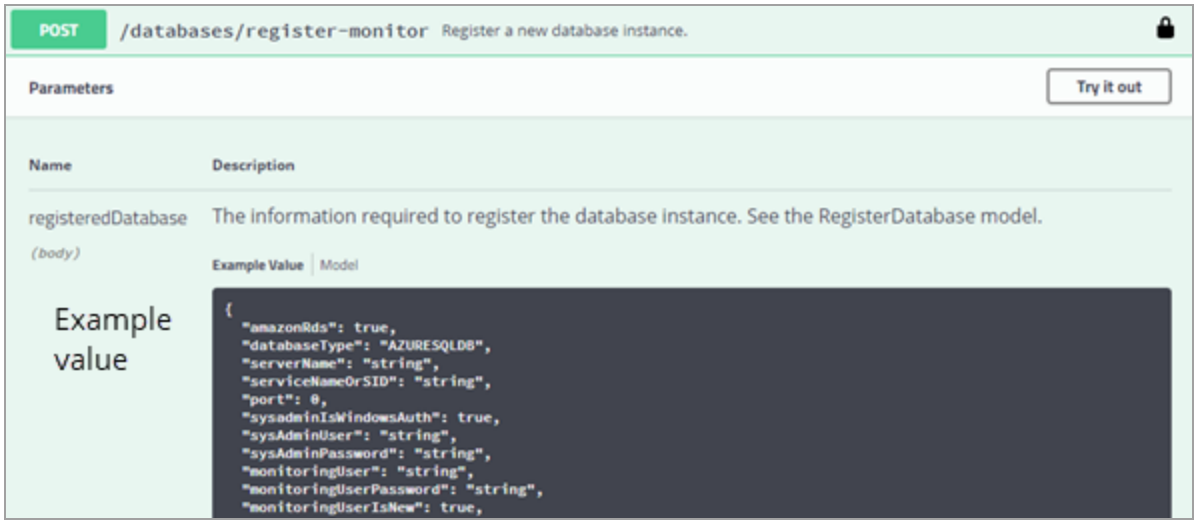
The Management API spec provides detailed information about each API endpoint. Endpoints are grouped by function.

1. Click any group to display the endpoints within it.

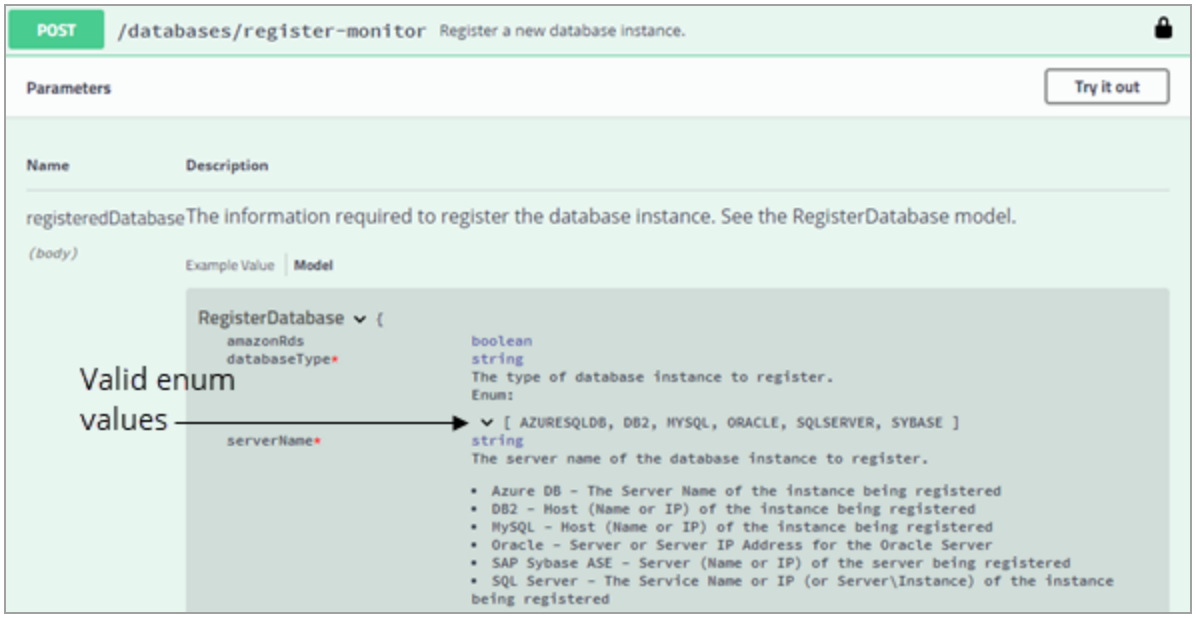


2. Click the endpoint to display its parameters and responses.

   Complex parameters and responses include an Example Value | Model section. The example value is shown by default.



3. Click Model to display additional information, including the valid values for enumerations.

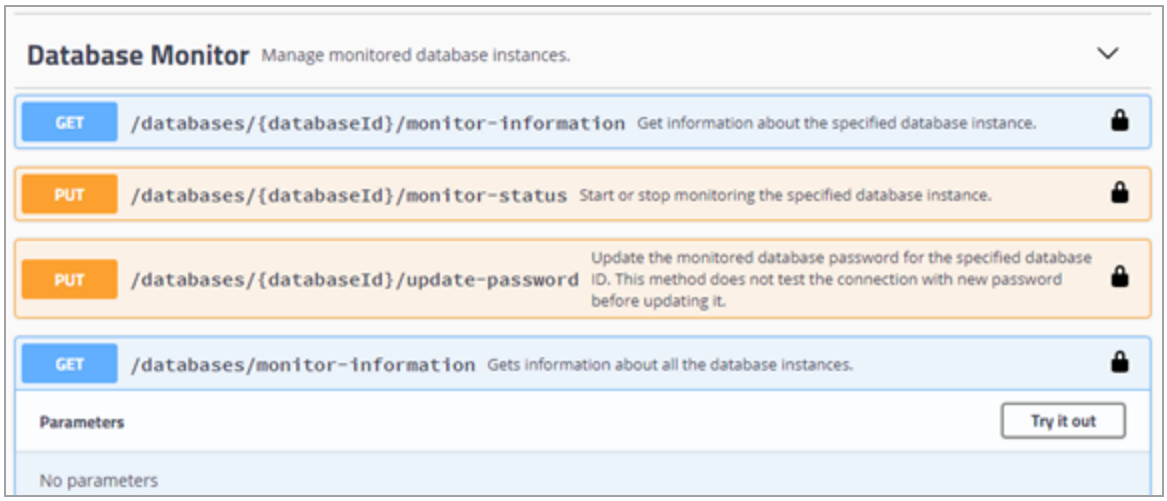


Make an API call from the Swagger interface

The following example shows how to make a call to get the current license allocation for a monitored database instance.

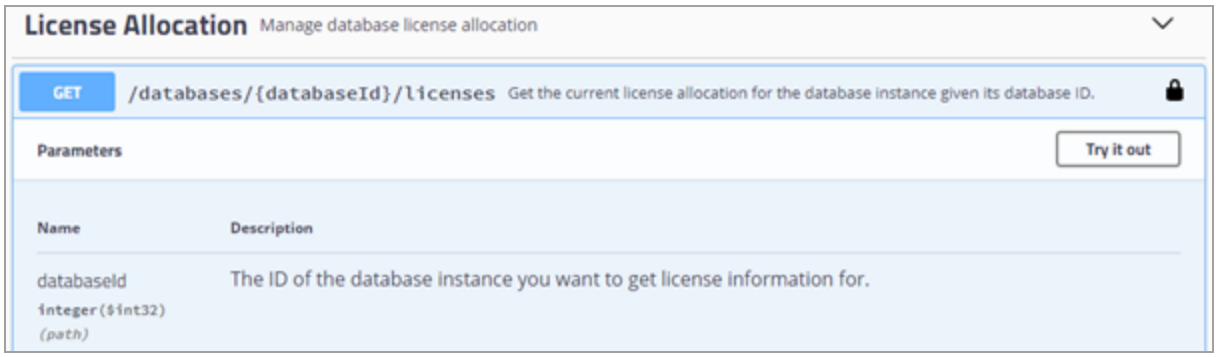
> When you make an API call through the Swagger interface, the call affects your DPA server in the same way as it would if it were issued through a command or script.

1. If you do not know the database ID, complete the following steps to get it:
   a. Click Database Monitor to display the endpoints.
   b. Click GET/databases/monitor-information to expand it.



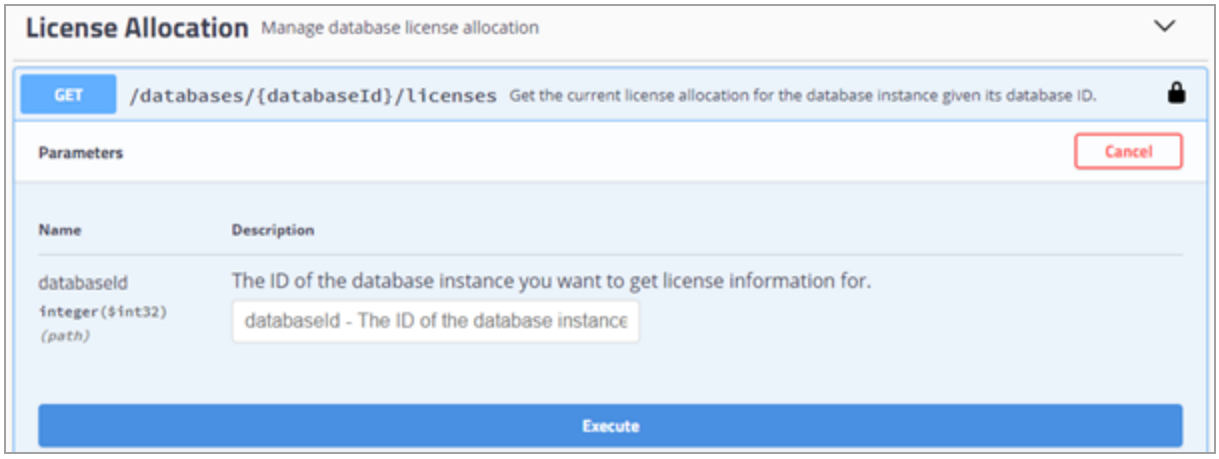
   c. Click Try it out, and then click Execute.
   d. Scroll through the Response body, find the database name, and make a note of the associated ID.

2. Click License Allocation to display the endpoints.
3. Click the GET/databases/{databaseId}/licenses endpoint to expand it.



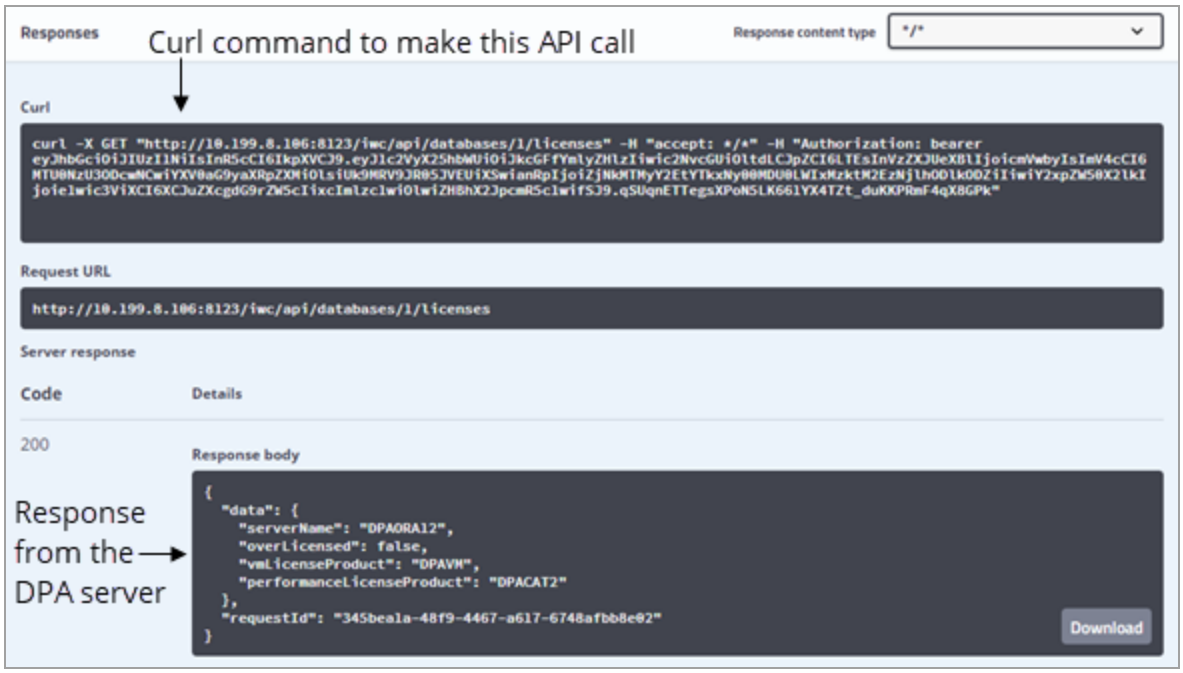
4. Click Try it out.

   The interface displays a field for the parameter value and an Execute button.



5. Enter the database ID and click Execute.

   The Response body section shows the response from the DPA server, and the Curl section shows the Curl command (including the access token) that could be run to make this API call.



Examples of using Python scripts to make DPA API calls

The following examples show Python scripts that call the DPA API to retrieve information and perform DPA management functions. The first examples are snippets that demonstrate each API call individually. The last example is a full script that shows how to put the snippets together into a working script.

> You can call the DPA API with any programming language that can send HTTP requests. See this topic for PowerShell script examples.

Prerequisites

- Before you can use scripts to make API calls, you must create a refresh token.
- These examples use the Requests HTTP library for Python. This library must be installed for these examples to work.

If your DPA server does not use HTTPS or your certificates are self-signed

The examples all use HTTPS, which can cause problems if your DPA server is not configured to use HTTPS or if your certificates are self signed. If this is the case, you can do either of the following:

- Run the examples using HTTP.
- Change the `verify_cert` value to `False` in the configuration section to prevent verifying the server's TLS certificate.

```
# ============================================================
# Configure the variables below for the DPA Host
# ============================================================
base_url = "https://localhost:8124/iwc/api/"
refresh_token = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."
verify_cert = False
# ============================================================
```

Get an access token

The first step in using the API is to get an access token. An access token is required to make any API calls. This call POSTs the refresh token to DPA, which returns an access token to be used by all other API calls.

- If the call is successful, it prints out the data that was returned from DPA, including the `access_token`, and then goes on to create HTTP Headers that will contain the access token and other information to be used on subsequent calls.
- If the call is not successful it prints out the error message.

You must set the `base_url` and the `refresh_token` variables to match your environment.

```
# ============================================================
# Configure the variables below for the DPA Host
# ============================================================
base_url = "https://localhost:8124/iwc/api/"
refresh_token = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."
verify_cert = True
# ============================================================

# ============================================================
# Get Access Token
# ============================================================
def get_access_header(prefix_url, rfrsh_token):
  """
  Given a base url and a refresh token retrieve the access token
  and return a header object with it.
  :param prefix_url: the base url
  :param rfrsh_token: refresh token used to get access token
  :return: the request header that contains the access token
  :rtype: dict
  """
```

```
  auth_token_url = prefix_url + "security/oauth/token"
  grant_type = "refresh_token"

  payload = {"grant_type": grant_type, "refresh_token": rfrsh_token}
  try:
    # get an access token
    resp = requests.post(auth_token_url, data=payload, verify=verify_cert)
    resp.raise_for_status()
    resp_json = resp.json()

    token_type = resp_json["token_type"]
    access_code = resp_json["access_token"]

    headers = {"authorization": f"{token_type} {access_code}",
      "content-type": "application/json;charset=UTF-8",
      "accept": "application/json"
      }

    return headers

  except requests.exceptions.HTTPError as ex:
    print(ex)
    print(ex.response.text)
    # print(json.dumps(json.loads(ex.response.text), indent=2))
    return None  # requests is bad return None, can't get access_code

# get the header that contains access token for authentication
header = get_access_header(base_url, refresh_token)
if header is None:
  sys.exit(0)
```

Database Monitor examples

The following examples show how to use Database Monitor calls.

Get information about one monitored database instance

```
# Get information about a single monitored database instance
database_id = 1
monitor_url = f"{base_url}databases/{database_id}/monitor-information"
```

```
single_monitor = None
try:
  print(f"\n*** Get Monitor Information for database with id of {database_id}
***")
  response = requests.get(monitor_url, headers=header, verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  single_monitor = response_json["data"]
  print(json.dumps(single_monitor, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Get Monitor Information for database with id of 1 ***
{
  "dbId": 1,
  "name": "DEV-DPA\SQLEXPRESS",
  "ip": "127.0.0.1",
  "port": "1433",
  "jdbcUrlProperties": "applicationIntent=readOnly",
  "connectionProperties": null,
  "databaseType": "SQL Server",
  "databaseVersion": "12.0.6205.1",
  "databaseEdition": "Enterprise Edition: Core-based Licensing (64-bit)",
  "monitoringUser": "ignite_next",
  "defaultDbLicenseCategory": "DPACAT2",
  "assignedDbLicenseCategory": "DPACAT2",
  "assignedVmLicenseCategory": null,
  "monitorState": "Monitor Stopped",
  "oldestMonitoringDate": "2018-12-09T00:00:00.000-07:00",
  "latestMonitoringDate": "2019-01-07T00:00:00.000-07:00",
  "agListenerName": null,
  "agClusterName": null,
  "agName": null,
  "racInfo": null,
  "rac": false,
  "rds": false,
  "ebusiness": false,
  "linkedToVirtualMachine": false,
  "pdb": false
```

```
}
```

Start and stop monitoring a database instance given its database ID

```
database_id = 1
monitor_url = f"{base_url}databases/{database_id}/monitor-status"
try:
  # Start monitoring a database instance given its database ID.
  print(f"*** Start Monitor for database {database_id} ***")
  body = {"command": "START"}
  response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))

  print("Waiting 15 seconds...")
  time.sleep(15)

  # Stop monitoring a database instance given its database ID.
  print(f"*** Stop Monitor for database {database_id} ***")
  body = {"command": "STOP"}
  response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
  print("Waiting 15 seconds...")
  time.sleep(15)

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Start Monitor for database 1 ***
"SUCCESS"
Waiting 15 seconds...

*** Stop Monitor for database 1 ***
```

```
"SUCCESS"
Waiting 15 seconds...
```

Get information about all monitored database instances

```
database_id = 1
monitor_url = f"{base_url}databases/monitor-information"
try:
  print("*** Get Information for a all database instances ***")
  response = requests.get(monitor_url, headers=header, verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  print(json.dumps(data, indent=2))

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Get information for all database instances ***
[
  {
    "dbId": 1,
    "name": "DEV-DPA\SQLEXPRESS",
    "ip": "127.0.0.1",
    "port": "1433",
    "jdbcUrlProperties": "applicationIntent=readOnly",
    "connectionProperties": null,
    "databaseType": "SQL Server",
    "databaseVersion": "12.0.6205.1",
    "databaseEdition": "Enterprise Edition: Core-based Licensing (64-bit)",
    "monitoringUser": "ignite_next",
    "defaultDbLicenseCategory": "DPACAT2",
    "assignedDbLicenseCategory": "DPACAT2",
    "assignedVmLicenseCategory": null,
    "monitorState": "Monitor Stopped",
    "oldestMonitoringDate": "2018-12-09T00:00:00.000-07:00",
    "latestMonitoringDate": "2019-01-07T00:00:00.000-07:00",
    "agListenerName": null,
    "agClusterName": null,
```

```
    "agName": null,
    "racInfo": null,
    "rac": false,
    "rds": false,
    "ebusiness": false,
    "linkedToVirtualMachine": false,
    "pdb": false
  },
  {
    "dbId": 2,
    "name": "DEV-MYSQL",
    "ip": "127.0.0.1",
    ...
  }
]
```

Stop and start monitoring for all database instances

```
# Start monitoring all database instances.
monitor_url = f"{base_url}databases/monitor-status"
try:
  print("*** Starting all Monitors ***")
  body = {"command": "START"}
  response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
  print("Waiting 30 seconds...")
  time.sleep(30)

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Stop monitoring all database instances.
try:
  print("*** Stopping all Monitors ***")
  body = {"command": "STOP"}
  response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
```

```
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
  print("Waiting 30 seconds...")
  time.sleep(30)

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Starting all Monitors ***
"SUCCESS"
Waiting 30 seconds...

*** Stopping all Monitors ***
"SUCCESS"
Waiting 30 seconds...
```

Update the user password for a monitored database instance

```
database_id = 1
monitor_url = f"{base_url}databases/{database_id}/update-password"
try:
  print(f"*** Update the Monitor password for database {database_id} ***")
  body = {"password": "NewPassword!"}
  response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Update the Monitor password for database 1 ***
"SUCCESS"
```

License Allocation examples

The examples below show how to use License Allocation calls.

Get information about currently installed licenses

```
license_url = f"{base_url}databases/licenses/installed"
try:
  print("\n*** Getting Installed license information with total amounts
available for use and total amounts used ***")
  response = requests.get(license_url, headers=header, verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  print("licenseProduct licenseCategory licensesAvailable licensesConsumed")
  print("-------------- --------------- ----------------- ----------------")
  for i in range(len(data)):
    print('{:<15s}{:<16s}{:>17d}{:>17d}'.format(data[i]["licenseProduct"],
data[i] ["licenseCategory"],
                                                                    data[i]["
data[i]["licensesConsumed"]))

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Getting Installed license information with total amounts available for
use and total amounts used ***
licenseProduct licenseCategory licensesAvailable licensesConsumed
-------------- --------------- ----------------- ----------------
DPACAT1        DPA_DB                        100               22
DPACAT2        DPA_DB                        100               16
DPAAzureSQL    DPA_DB                          0                0
DPAVM          DPA_VM                        100               12
```

Get license information for a single database instance

```
database_id = 1
license_url = f"{base_url}databases/{database_id}/licenses"
try:
  print(f"\n*** Getting current license information for the database instance
```

```
with database ID of {database_id} ***")
  response = requests.get(license_url, headers=header, verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Getting current license information for the database instance with
database ID of 1 ***
{
  "serverName": "DEV-DPA",
  "overLicensed": false,
  "performanceLicenseProduct": "DPACAT2",
  "vmLicenseProduct": "DPAVM"
}
```

Update license information for a database instance

```
database_id = 1
license_url = f"{base_url}databases/{database_id}/licenses"

# Add a DPACAT2 and a DPAVM license
body = {"performanceLicenseProduct": "DPACAT2",
       "vmLicenseProduct": "DPAVM"}
try:
  print(f"\n*** Updating license for database id {database_id} ***")
  response = requests.put(license_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Remove the DPAVM license
body = {"performanceLicenseProduct": "DPACAT2",
```

```
        "vmLicenseProduct": "REMOVE"}
try:
  print(f"\n*** Updating license for database id {database_id} ***")
  response = requests.put(license_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Updating license for database id 1 ***
{
  "serverName": "DEV-BOU-CALLEN",
  "overLicensed": false,
  "performanceLicenseProduct": "DPACAT2",
  "vmLicenseProduct": DPAVM
}
*** Updating license for database id 1 ***
{
  "serverName": "DEV-BOU-CALLEN",
  "overLicensed": false,
  "performanceLicenseProduct": "DPACAT2",
  "vmLicenseProduct": null
}
```

Annotation examples

The examples below show how to use Annotation calls.

Get a list of annotations for the last 30 days

```
# Gets a List of annotations for the last 30 days
database_id = 1
annotation_url = f"{base_url}databases/{database_id}/annotations"

# Dates are in ISO 8601 format ( 2018-12-31T12:00:00.000-07:00 )
end_time = datetime.datetime.now()
start_time = end_time + datetime.timedelta(days=-30)
```

```
args = {"startTime": start_time.astimezone().isoformat(),
        "endTime": end_time.astimezone().isoformat()}

try:
  print("\n*** Getting Annotations for the last 30 days ***")
  response = requests.get(annotation_url, params=args, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Getting Annotations for the last 30 days ***
[
  {
    "id": 112,
    "title": "Test Title API",
    "description": "Test Event created by DPA API",
    "createdBy": "DPA API",
    "time": "2018-12-11T10:01:35-07:00",
    "type": "API"
  },
  {
    "id": 113,
    "title": "Test Title API",
    "description": "Test Event created by DPA API",
    "createdBy": "DPA API",
    "time": "2018-12-12T15:00:40-07:00",
    "type": "API"
  },
  {
    ...
  }
]
```

Create a new annotation

```
database_id = 1
annotation_url = f"{base_url}databases/{database_id}/annotations"

# Dates are in ISO 8601 format, no millis ( 2018-12-31T12:00:00-07:00 )
create_time = datetime.datetime.now().replace(microsecond=0)
body = {"title": "API Test Title",
  "description": "API Test Description",
  "createdBy": "Test API User",
  "time": create_time.astimezone().isoformat()}
try:
  print("\n*** Creating Annotation ***")
  response = requests.post(annotation_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Creating Annotation ***
{
  "id": 171,
  "title": "API Test Title",
  "description": "API Test Description",
  "createdBy": "Test API User",
  "time": "2019-01-09T11:04:33-07:00",
  "type": "API"
}
```

Delete an annotation

```
database_id = 1
annotation_id = 171
annotation_url = f"{base_url}databases/{database_id}/annotations/{annotation_
id}"
try:
  print(f"\n*** Deleting Annotation with id of {annotation_id} ***")
```

```
  response = requests.delete(annotation_url, headers=header, verify=verify_
cert)
  response.raise_for_status()
  if response.status_code == 204:
    print(f"Annotation with id of {annotation_id} deleted")
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Deleting Annotation with id of 171 ***
Annotation with id of 171 deleted
```

Database Registration examples

The examples below show how to use Database Registration calls.

Register and unregister a SQL Server database instance for monitoring

This example registers a new SQL Server database instance, waits 60 seconds, and then unregisters the database instance.

```
# ------------------------------------------------------------
# Register a SQL Server database instance for monitoring.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "SQLSERVER",
        "serverName": "127.0.0.1",
        "port": "1433",
        "sysAdminUser": "sa",
        "sysAdminPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "displayName": "DPA_SQL2K12"}

new_db_id = None
try:
  print("\n*** Register SQL Server database ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
```

```
  response.raise_for_status()
  responseJson = response.json()
  data = responseJson["data"]
  new_db_id = data["databaseId"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)

# ------------------------------------------------------------
# Unregister the SQL Server database instance.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "sa",
        "sysAdminPassword": "Password"}
try:
  print(f"\n*** Unregister SQL Server database [{new_db_id}] ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Register SQL Server database ***
{
  "databaseId": 77,
  "result": "SUCCESS"
}
Waiting 60 seconds...

*** Unregister SQL Server database [77] ***
```

```
{
  "databaseId": 77,
  "result": "SUCCESS"
}
```

Register and unregister an Oracle database instance for monitoring

This example registers a new Oracle database instance, waits 60 seconds, and then unregisters the database instance.

```
# ------------------------------------------------------------
# Register an Oracle database instance for monitoring.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "ORACLE",
        "serverName": "127.0.0.1",
        "serviceNameOrSID": "DPA_ORA11R1",
        "port": "1521",
        "sysAdminUser": "system",
        "sysAdminPassword": "Password",
        "sysPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "monitoringUserTableSpace": "USERS",
        "monitoringUserTempTableSpace": "TEMP",
        "oracleEBusinessEnabled": False,
        "displayName": "DPA_ORA11R1"}

new_db_id = None
try:
  print("\n*** Register Oracle database ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  data = responseJson["data"]
  new_db_id = data["databaseId"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
```

```
  print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)

# ------------------------------------------------------------
# Unregister the Oracle database instance.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "system",
        "sysAdminPassword": "Password"}
try:
  print(f"\n*** Unregister Oracle database [{new_db_id}] ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Register Oracle database ***
{
  "databaseId": 78,
  "result": "SUCCESS"
}
Waiting 60 seconds...

*** Unregister Oracle database [78] ***
{
  "databaseId": 78,
  "result": "SUCCESS"
}
```

Register and unregister a MySQL database instance for monitoring

This example registers a new MySQL database instance, waits 60 seconds, and then unregisters the database instance.

```
# ------------------------------------------------------------
# Register a MySQL database instance for monitoring.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "MYSQL",
        "serverName": "127.0.0.1",
        "port": "3306",
        "sysAdminUser": "root",
        "sysAdminPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "displayName": "DPA_MYSQL56"}

new_db_id = None
try:
  print("\n*** Register MySQL database ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  data = responseJson["data"]
  new_db_id = data["databaseId"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)

# ------------------------------------------------------------
# Unregister the MySQL database instance.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
```

```
        "removeDatabaseObjects": True,
        "sysAdminUser": "root",
        "sysAdminPassword": "Password"}
try:
  print(f"\n*** Unregister MySQL database [{new_db_id}] ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will print out data like this:
*** Register MySQL database ***
{
  "databaseId": 79,
  "result": "SUCCESS"
}
Waiting 60 seconds...

*** Unregister MySQL database [79] ***
{
  "databaseId": 79,
  "result": "SUCCESS"
}
```

Database Custom Properties examples

The examples below show how to use Database Custom Properties calls. Custom property values can be included in custom email templates for alert notifications.

Create a custom property

This script creates a custom property and defines its name and description.

```
property_name = "Location"
property_description = "Location of the database server"
create_property_url = f"{base_url}databases/properties"
body = {
```

```
  "name": property_name,
  "description": property_description
}

property_id = None

try:
  print("\n*** Creating custom property ***")
  response = requests.post(create_property_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  property_id = data["id"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

#This will print out data like this:
{
  "id": 1,
  "name": "Location",
  "description": "Location of the database server"
}
```

Create a custom property value

This script creates a value for the custom property created by the previous script.

```
property_id = 1
property_value = "New York"
create_value_url = f"{base_url}databases/properties/" + str(property_id) +
"/values"
body = property_value

property_value_id = None

try:
  print("\n*** Creating custom property value ***")
  response = requests.post(create_value_url, data=body, headers=header,
```

```
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  property_value_id = data["id"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

#This will print out data like this:
{
  "id": 1,
  "value": "New York"
}
```

Assign a property value to a monitored database instance

This script assigns a property value to a monitored database instance.

```
property_id = 1
property_value_id = 1
database_id = 1
assign_property_value_url = f"{base_url}databases/" + str(database_id) +
"/properties/" + str(property_id) + "/values/" + str(property_value_id)

try:
  print("\n*** Assigning custom property value ***")
  response = requests.post(assign_property_value_url, headers=header,
verify=verify_cert)
  response.raise_for_status()
  if response.status_code == 200:
  print(f"Custom property value assigned to the DB with ID: {database_id}")
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

#This will print out data like this:
#Custom property value assigned to the DB with ID: 1
```

Get all information about properties

This script returns information about all custom properties and their values.

```
get_properties_url = f"{base_url}databases/properties?require=assignment"

try:
  print("\n*** Getting custom property information ***")
  response = requests.get(get_properties_url, headers=header, verify=verify_
cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

#This will print out data like this:
[
  {
    "id": 1,
    "name": "Location",
    "description": "Location of the database server",
    "values": [
      {
        "id": 1,
        "value": "New York",
        "assignment": [
          1
        ]
      }
    ],
    "unassigned": [
      2
    ]
  }
]
```

Delete a custom property

This script deletes a custom property.

```
property_id = 1
delete_property_url = f"{base_url}databases/properties/" + str(property_id)

try:
  print("\n*** Deleting custom property ***")
  response = requests.delete(delete_property_url, headers=header,
verify=verify_cert)
  response.raise_for_status()
  if response.status_code == 204:
  print(f"Custom property with ID of {property_id} deleted")
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

#This will print out data like this:
#Custom property with ID of 1 deleted
```

Full working script

The following script combines all of the examples shown above into a script that can be run.

```
import json
import sys
import time
import datetime
import requests

# ============================================================
# Configure the variables below for the DPA Host
# ============================================================
base_url = "http://localhost:8124/iwc/api/"
refresh_token = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."
verify_cert = True
# ============================================================

# ============================================================
# Get Access Token
# ============================================================
def get_access_header(prefix_url, rfrsh_token):
  """
  Given a base url and a refresh token retrieve the access token
```

```
  and return a header object with it.
  :param prefix_url: the base url
  :param rfrsh_token: refresh token used to get access token
  :return: the request header that contains the access token
  :rtype: dict
  """

  auth_token_url = prefix_url + "security/oauth/token"
  grant_type = "refresh_token"

  payload = {"grant_type": grant_type, "refresh_token": rfrsh_token}
  try:
    # get an access token
    resp = requests.post(auth_token_url, data=payload, verify=verify_cert)
    resp.raise_for_status()
    resp_json = resp.json()

    token_type = resp_json["token_type"]
    access_code = resp_json["access_token"]

    headers = {"authorization": f"{token_type} {access_code}",
               "content-type": "application/json;charset=UTF-8",
               "accept": "application/json"
    }

    return headers

  except requests.exceptions.HTTPError as ex:
    print(ex)
    print(ex.response.text)
    # print(json.dumps(json.loads(ex.response.text), indent=2))
    return None  # requests is bad return None, can't get access_code

# get the header that contains access token for authentication
header = get_access_header(base_url, refresh_token)
if header is None:
  sys.exit(0)

# ============================================================
# Database Monitor Examples
# ============================================================
```

```
# Calls for individual monitors...

# Get Monitor Information for a single database instance
database_id = 1
monitor_url = f"{base_url}databases/{database_id}/monitor-information"
single_monitor = None
try:
  print(f"\n*** Get Monitor Information for database with id of {database_id}
***")
  response = requests.get(monitor_url, headers=header, verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  single_monitor = response_json["data"]
  print(json.dumps(single_monitor, indent=2))

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Start or Stop monitoring a database instance given its database ID.
# If it is already running stop it and then restart it
# If it is not running start it and then stop it
if single_monitor is not None:
  monitor_url = f"{base_url}databases/{database_id}/monitor-status"
  if single_monitor["monitorState"] == "Monitor Running":
    change_command = "STOP"
    revert_command = "START"
  elif single_monitor["monitorState"] == "Monitor Stopped":
    change_command = "START"
    revert_command = "STOP"
  else:
    change_command = None
    revert_command = None

  if change_command is not None:
    try:
      print(f"\n*** {change_command} Monitor for database {database_id} ***")
      body = {"command": change_command}
      response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
```

```
      response.raise_for_status()
      response_json = response.json()
      print(json.dumps(response_json["data"], indent=2))

      print("Waiting 15 seconds...")
      time.sleep(15)

      print(f"\n*** {revert_command} Monitor for database {database_id} ***")
      body = {"command": revert_command}
      response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
      response.raise_for_status()
      response_json = response.json()
      print(json.dumps(response_json["data"], indent=2))

      print("Waiting 15 seconds...")
      time.sleep(15)

    except requests.exceptions.HTTPError as e:
      print(e)
      print(e.response.text)

# Calls for all monitors...

# Get Monitor Information for all database instances
database_id = 1
running_ids = []
monitor_url = f"{base_url}databases/monitor-information"
try:
  print("\n*** Get Information for a all database instances ***")
  response = requests.get(monitor_url, headers=header, verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  print(json.dumps(data, indent=2))

  # Keep a list of running or started monitors to be used later
  for monitor in data:
    state = monitor["monitorState"]
    if state == "Monitor Running" or state == "Monitor Start No License" or
'Start' in state:
```

```
      running_ids.append(monitor["dbId"])

  print(f"Running Monitors: {running_ids}")

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Start monitoring all database instances.
monitor_url = f"{base_url}databases/monitor-status"
try:
  print("\n*** Starting all Monitors ***")
  body = {"command": "START"}
  response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
  print("Waiting 30 seconds...")
  time.sleep(30)

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Stop monitoring all database instances.
try:
  print("\n*** Stopping all Monitors ***")
  body = {"command": "STOP"}
  response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
  print("Waiting 30 seconds...")
  time.sleep(30)

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)
```

```
# Try to put it back the way we found it by restarting the ones that were
running
for db_id in running_ids:
  try:
    print(f"\n*** Starting Monitor for database {db_id} ***")
    monitor_url = f"{base_url}databases/{db_id}/monitor-status"
    body = {"command": "START"}
    response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
    response.raise_for_status()
    response_json = response.json()
    print(json.dumps(response_json["data"], indent=2))
  except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# Update the monitor database user password (Un-comment to use)
#monitor_url = f"{base_url}databases/{database_id}/update-password"
#try:
#  print(f"*** Update the Monitor password for database {database_id} ***")
#  body = {"password": "Password"}
#  response = requests.put(monitor_url, json=body, headers=header,
verify=verify_cert)
#  response.raise_for_status()
#  response_json = response.json()
#  print(json.dumps(response_json["data"], indent=2))

#except requests.exceptions.HTTPError as e:
#  print(e)
#  print(e.response.text)

# ============================================================
# Licensing Examples
# ============================================================

# Get the currently installed license information
license_url = f"{base_url}databases/licenses/installed"
try:
  print("\n*** Getting Installed license information with total amounts
available for use and total amounts used ***")
  response = requests.get(license_url, headers=header, verify=verify_cert)
```

```
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  print("licenseProduct licenseCategory licensesAvailable licensesConsumed")
  print("-------------- --------------- ----------------- ----------------")
  for i in range(len(data)):
    print('{:<15s}{:<16s}{:>17d}{:>17d}'.format(data[i]["licenseProduct"],
data[i]["licenseCategory"],
    data[i]["licensesAvailable"], data[i]["licensesConsumed"]))

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Get License Information for a single database
license_url = f"{base_url}databases/{database_id}/licenses"
license_info = None
try:
  print(f"\n*** Getting current license information for the database instance
with database ID of {database_id} ***")
  response = requests.get(license_url, headers=header, verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  license_info = response_json["data"]
  print(json.dumps(license_info, indent=2))

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# This will Update License Information for a single database setting the
# Performance License and the VM License to what it currently is.
# It should succeed but it should make no changes.
if license_info is not None:
  database_id = 1
  license_url = f"{base_url}databases/{database_id}/licenses"
  db_product = license_info["performanceLicenseProduct"]
  vm_product = license_info["vmLicenseProduct"]
  body = {"performanceLicenseProduct": db_product,
    "vmLicenseProduct": vm_product}
try:
```

```
  print(f"\n*** Updating license for database id {database_id} ***")
  response = requests.put(license_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))

except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# ============================================================
# Annotation Examples
# ============================================================

# Gets a List of annotations for the last 30 days
annotation_url = f"{base_url}databases/{database_id}/annotations"

# Dates are in ISO 8601 format ( 2018-12-31T12:00:00.000-07:00 )
end_time = datetime.datetime.now()
start_time = end_time + datetime.timedelta(days=-30)
args = {"startTime": start_time.astimezone().isoformat(),
        "endTime": end_time.astimezone().isoformat()}

try:
  print("\n*** Getting Annotations for the last 30 days ***")
  response = requests.get(annotation_url, params=args, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  print(json.dumps(response_json["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Create a new annotation
annotation_url = f"{base_url}databases/{database_id}/annotations"
annotation_id = None

#Dates are in ISO 8601 format, no millis ( 2018-12-31T12:00:00-07:00 )
create_time = datetime.datetime.now().replace(microsecond=0)
```

```
body = {"title": "API Test Title",
        "description": "API Test Description",
        "createdBy": "Test API User",
        "time": create_time.astimezone().isoformat()}
try:
  print("\n*** Creating Annotation ***")
  response = requests.post(annotation_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  annotation_id = data["id"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Delete an annotation
if annotation_id is not None:
  annotation_url = f"{base_url}databases/{database_id}/annotations/
{annotation_id}"
  try:
    print(f"\n*** Deleting Annotation with id of {annotation_id} ***")
    response = requests.delete(annotation_url, headers=header, verify=verify_
cert)
    response.raise_for_status()
    if response.status_code == 204:
      print(f"Annotation with id of {annotation_id} deleted")
  except requests.exceptions.HTTPError as e:
    print(e)
    print(e.response.text)

# ============================================================
# Registration Examples
# ============================================================

# ------------------------------------------------------------
# Register a SQL Server database instance for monitoring.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "SQLSERVER",
```

```
        "serverName": "127.0.0.1",
        "port": "1433",
        "sysAdminUser": "User",
        "sysAdminPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "displayName": "DPA_SQL2K12"}

new_db_id = None
try:
  print("\n*** Register SQL Server database ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  data = responseJson["data"]
  new_db_id = data["databaseId"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)

# ------------------------------------------------------------
# Un-register the SQL Server database instance.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "User",
        "sysAdminPassword": "Password"}
try:
  print(f"\n*** Unregister SQL Server database [{new_db_id}] ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
```

```
  print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# ------------------------------------------------------------
# Register an Oracle database instance for monitoring.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "ORACLE",
        "serverName": "127.0.0.1",
        "serviceNameOrSID": "DPA_ORA11R1",
        "port": "1521",
        "sysAdminUser": "User",
        "sysAdminPassword": "Password",
        "sysPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "monitoringUserTableSpace": "USERS",
        "monitoringUserTempTableSpace": "TEMP",
        "oracleEBusinessEnabled": False,
        "displayName": "DPA_ORA11R1"}

new_db_id = None
try:
  print("\n*** Register Oracle database ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  data = responseJson["data"]
  new_db_id = data["databaseId"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)
```

```
# ------------------------------------------------------------
# Un-register the Oracle database instance.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "User",
        "sysAdminPassword": "Password"}
try:
  print(f"\n*** Unregister Oracle database [{new_db_id}] ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# ------------------------------------------------------------
# Register a MySQL database instance for monitoring.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/register-monitor"
body = {"databaseType": "MYSQL",
        "serverName": "127.0.0.1",
        "port": "3306",
        "sysAdminUser": "User",
        "sysAdminPassword": "Password",
        "monitoringUser": "dpa_test_m",
        "monitoringUserPassword": "Password",
        "monitoringUserIsNew": True,
        "displayName": "DPA_MYSQL56"}

new_db_id = None
try:
  print("\n*** Register MySQL database ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
```

```
  data = responseJson["data"]
  new_db_id = data["databaseId"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

print("Waiting 60 seconds...")
time.sleep(60)
# ------------------------------------------------------------
# Un-register the MySQL database instance.
# ------------------------------------------------------------
registration_url = f"{base_url}databases/unregister-monitor"
body = {"databaseId": new_db_id,
        "removeMonitoringUser": True,
        "removeDatabaseObjects": True,
        "sysAdminUser": "User",
        "sysAdminPassword": "Password"}
try:
  print(f"\n*** Unregister MySQL database [{new_db_id}] ***")
  response = requests.post(registration_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  responseJson = response.json()
  print(json.dumps(responseJson["data"], indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)
# ============================================================
# Custom Property Examples
# ============================================================

# Create custom property
property_name = "Location"
property_description = "Location of the database server"
create_property_url = f"{base_url}databases/properties"
body = {
  "name": property_name,
  "description": property_description
}
```

```
property_id = None

try:
  print("\n*** Creating custom property ***")
  response = requests.post(create_property_url, json=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  property_id = data["id"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Create value of the custom property
property_value = "New York"
create_value_url = f"{base_url}databases/properties/" + str(property_id) +
"/values"
body = property_value

property_value_id = None

try:
  print("\n*** Creating custom property value ***")
  response = requests.post(create_value_url, data=body, headers=header,
verify=verify_cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  property_value_id = data["id"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Assign property value to DB
assign_property_value_url = f"{base_url}databases/" + str(database_id) +
"/properties/" + str(property_id) + "/values/" + str(property_value_id)

try:
```

```
  print("\n*** Assigning custom property value ***")
  response = requests.post(assign_property_value_url, headers=header,
verify=verify_cert)
  response.raise_for_status()
  if response.status_code == 200:
  print(f"Custom property value assigned to the DB with ID: {database_id}")
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Get all information about properties (including DB assignment)
get_properties_url = f"{base_url}databases/properties?require=assignment"

try:
  print("\n*** Getting custom property information ***")
  response = requests.get(get_properties_url, headers=header, verify=verify_
cert)
  response.raise_for_status()
  response_json = response.json()
  data = response_json["data"]
  print(json.dumps(data, indent=2))
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)

# Delete custom property
delete_property_url = f"{base_url}databases/properties/" + str(property_id)

try:
  print("\n*** Deleting custom property ***")
  response = requests.delete(delete_property_url, headers=header,
verify=verify_cert)
  response.raise_for_status()
  if response.status_code == 204:
  print(f"Custom property with id of {property_id} deleted")
except requests.exceptions.HTTPError as e:
  print(e)
  print(e.response.text)
```

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Examples of PowerShell scripts that make DPA API calls

The following examples show PowerShell scripts that call the DPA API to retrieve information and perform DPA management functions. The first examples are snippets that demonstrate each API call individually. The last example is a full script that shows how to put the snippets together into a working script.

ⓘ You can call the DPA API with any programming language that can send HTTP requests. See this topic for Python script examples.

See the following sections:

Prerequisite

Before you can use scripts to make API calls, you must create a refresh token.

If your DPA server does not use HTTPS or your certificates are self-signed

The examples all use HTTPS, which can cause problems if your DPA server is not configured to use HTTPS or if your certificates are self signed. If this is the case, you can do either of the following:

- Run the examples using HTTP.
- Add the following code below the configuration section.

```
#------------------------------------------------------------
# Adding certificate exception to prevent API errors
#------------------------------------------------------------

add-type @"
  using System.Net;
  using System.Security.Cryptography.X509Certificates;
  public class TrustAllCertsPolicy : ICertificatePolicy {
    public bool CheckValidationResult(
      ServicePoint srvPoint, X509Certificate certificate,
      WebRequest request, int certificateProblem) {
      return true;
    }
  }
"@
[System.Net.ServicePointManager]::CertificatePolicy = New-Object
TrustAllCertsPolicy
```

Get an access token

The first step in using the API is to get an access token. An access token is required to make any API calls. This call POSTs the refresh token to DPA, which returns an access token to be used by all other API calls.

- If the call is successful, it prints out the data that was returned from DPA, including the `access_token`, and then goes on to create HTTP Headers that will contain the access token and other information to be used on subsequent calls.
- If the call is not successful it prints out the error message.

You must set the `$baseURL` and the `$refreshToken` variables to match your environment.

```
#------------------------------------------------------------
# Configure the variables below for the DPA Host
#------------------------------------------------------------
$baseURL = "https://localhost:8124/iwc/api/"
$refreshToken = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."

#------------------------------------------------------------
# Get an access token
#------------------------------------------------------------
$authTokenURL = $baseURL + "security/oauth/token"
```

```
$body = @{"grant_type" = "refresh_token"
          "refresh_token" = "$refreshToken"}
Try {
  Write-Host "Getting Access Token..."
  $dpaAuthResponse = Invoke-RestMethod -Uri $authTokenURL -Method POST -Body $body
  $dpaAuthResponse | Format-List
}
Catch {
  $_.Exception.ToString()
  return
}

# If successful we will create our headers to be used for all API calls
$tokenType = $DpaAuthResponse.token_type
$accessToken = $DpaAuthResponse.access_token
$dpaHeader = @{}
$dpaHeader.Add("Accept", "application/json")
$dpaHeader.Add("Content-Type", "application/json;charset=UTF-8")
$dpaHeader.Add("Authorization", "$tokenType $accessToken")

# This will print out data like this:

Getting Access Token...

access_token :
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX25hbWUiOiJpZ25pdGUiLCJzY29wZSI6W10sI
mlkIjotMSwidXNlcl...
token_type   : bearer
expires_in   : 365
id           : -1
userType     : repo
jti          : e0d51295-2010-4ed4-b5ea-982a4e6ae1c5
```

Database Monitor examples

The following examples show how to use all Database Monitor calls.

Get information about one monitored database instance

```
# Get Monitor Information for a single database
$databaseId = 1
```

```
$monitorURL = $baseURL + "databases/$databaseId/monitor-information"
Try {
  Write-Host "Get Monitor Information for database with id of $databaseId..."
  $monitorJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers $dpaHeader
-TimeoutSec 60
  $monitor = $monitorJSON.data
  $monitor | Format-List
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Get Monitor Information for database with id of 1...
dbId                       : 1
name                       : DEV-DPA\SQLEXPRESS
ip                         : 127.0.0.1
port                       : 1433
jdbcUrlProperties          : applicationIntent=readOnly
connectionProperties       :
databaseType               : SQL Server
databaseVersion            : 12.0.6205.1
databaseEdition            : Enterprise Edition: Core-based Licensing (64-bit)
monitoringUser             : ignite_next
defaultDbLicenseCategory   : DPACAT2
assignedDbLicenseCategory  : DPACAT2
assignedVmLicenseCategory  :
monitorState               : Monitor Running
oldestMonitoringDate       : 2018-12-04T00:00:00.000-07:00
latestMonitoringDate       : 2019-01-02T00:00:00.000-07:00
agListenerName             :
agClusterName              :
agName                     :
racInfo                    :
rac                        : False
linkedToVirtualMachine     : False
rds                        : False
pdb                        : False
ebusiness                  : False
```

Start and stop monitoring a database instance given its database ID

```
$databaseId = 1
$monitorURL = $baseURL + "databases/$databaseId/monitor-status"

# Start monitoring a database instance given its database ID.
Try {
  Write-Host "Start Monitor for database $databaseId..."
  $command = @{"command" = "START"} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
  $result = $monitorJSON.data
  Write-Host "Result: $result"
  Write-Host "Waiting 15 seconds...`r`n"
  Start-Sleep -s 15
}
Catch {
  $_.Exception.ToString()
}

# Stop monitoring a database instance given its database ID.
Try {
  Write-Host "Stop Monitor for database $databaseId..."
  $command = @{"command" = "STOP"} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
  $result = $monitorJSON.data
  Write-Host "Result: $result"
  Write-Host "Waiting 15 seconds...`r`n"
  Start-Sleep -s 15
}
Catch {
  $_.Exception.ToString()
}
# This will print out data like this:
Start Monitor for database 1...
Result: SUCCESS
Waiting 15 seconds...

Stop Monitor for database 1...
Result: SUCCESS
Waiting 15 seconds...
```

Get information about all monitored database instances

```
# Get Monitor Information for all databases
$monitorURL = $baseURL + "databases/monitor-information"
Try {
  Write-Host "Get Monitor Information for all databases..."
  $monitorListJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers
$dpaHeader -TimeoutSec 60
  $monitorList = $monitorListJSON.data
  $monitorList | Format-Table -AutoSize
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Get Monitor Information for all databases...

dbId name                 ip          port  jdbcUrlProperties
connectionProperties  databaseType  databaseVersion  ...
---- ----                 --          ----  -----------------           -----------
---------  ------------  ---------------  ---
   1 DEV-DPA\SQLEXPRESS  10.10.10.1  1433  applicationIntent=readOnly
          SQL Server    12.0.6205.1      ...
   3 DEVORA11_DEVORA11   10.10.10.2  1521
          Oracle        11.2.0.1.0       ...
  10 DEV-MYSQL:3306      10.10.10.3  3306  dumpQueriesOnException=true
          MySQL         5.7.19           ...
etc.
```

Stop and start monitoring for all database instances

```
$monitorURL = $baseURL + "databases/monitor-status"

# Start monitoring all database instances.
Try {
  Write-Host "Starting all Monitors..."
  $command = @{"command" = "START"} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
  $result = $monitorJSON.data
```

```
  Write-Host "Result: $result"
  Write-Host "Waiting 30 seconds...`r`n"
  Start-Sleep -s 30
}
Catch {
  $_.Exception.ToString()
}

# Stop monitoring all database instances.
Try {
  Write-Host "Stopping all Monitors..."
  $command = @{"command" = "STOP"} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
  $result = $monitorJSON.data
  Write-Host "Result: $result"
  Write-Host "Waiting 30 seconds...`r`n"
  Start-Sleep -s 30
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Starting all Monitors...
Result: SUCCESS
Waiting 30 seconds...

Stopping all Monitors...
Result: SUCCESS
Waiting 30 seconds...
```

Update the user password for a monitored database instance

```
$databaseId = 1
$monitorURL = $baseURL + "databases/$databaseId/update-password"
Try {
  Write-Host "Update the Monitor password for database $databaseId..."
  $command = @{"password" = "NewPassword!"} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
```

```
  $result = $monitorJSON.data
  Write-Host "Result: $result`r`n"
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Update the Monitor password for database 1...
Result: SUCCESS
```

License Allocation examples

The examples below show how to use all License Allocation calls.

Get information about currently installed licenses

```
$licenseURL = $baseURL + "databases/licenses/installed"

Try {
  Write-Host "Getting Installed license information with total amounts available
for use and total amounts used..."
  $licenseListJSON = Invoke-RestMethod -Method Get -Uri $licenseURL -Headers
$dpaHeader -TimeoutSec 60
  $licenseList = $licenseListJSON.data
  $licenseList | Format-Table -AutoSize
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Getting Installed license information with total amounts available for use and
total amounts used...

licenseProduct licenseCategory licensesAvailable licensesConsumed
-------------- --------------- ----------------- ----------------
DPACAT1        DPA_DB                        100               22
DPACAT2        DPA_DB                        100               16
DPAAzureSQL    DPA_DB                          0                0
DPAVM          DPA_VM                        100               12
```

Get license information for a single database instance

```
$databaseId = 1
$licenseURL = $baseURL + "databases/$databaseId/licenses"
Try {
  Write-Host "Getting current license information for the database instance with
database ID of $databaseId."
  $licenseListJSON = Invoke-RestMethod -Method Get -Uri $licenseURL -Headers
$dpaHeader -TimeoutSec 60
  $licenseInfo = $licenseListJSON.data
  $licenseInfo | Format-Table -AutoSize
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Getting current license information for the database instance with database ID of
1.

serverName overLicensed vmLicenseProduct performanceLicenseProduct
---------- ------------ ---------------- -------------------------
DEV-DPA           False DPAVM            DPACAT2
```

Update license information for a database instance

```
$databaseId = 1
$licenseURL = $baseURL + "databases/$databaseId/licenses"

# Add a DPACAT2 and a DPAVM license
$licenseAllocation = @{"performanceLicenseProduct" = "DPACAT2";
                       "vmLicenseProduct" = "DPAVM"} | ConvertTo-Json
Try {
  Write-Host "Updating license for database id $databaseId..."
  $licenseResultJSON = Invoke-RestMethod -Method Put -Uri $licenseURL -Body
$licenseAllocation -Headers $dpaHeader -TimeoutSec 60
  $licenseResult = $licenseResultJSON.data
  Write-Host "New License Allocation result for the database instance with database
ID of $databaseId."
  $licenseResult | Format-Table -AutoSize
}
```

```
Catch {
  $_.Exception.ToString()
}

# Remove the DPAVM license
$licenseAllocation = @{"performanceLicenseProduct" = "DPACAT2";
                       "vmLicenseProduct" = "REMOVE"} | ConvertTo-Json
Try {
  Write-Host "Updating license for database id $databaseId..."
  $licenseResultJSON = Invoke-RestMethod -Method Put -Uri $licenseURL -Body
$licenseAllocation -Headers $dpaHeader -TimeoutSec 60
  $licenseResult = $licenseResultJSON.data
  Write-Host "New License Allocation result for the database instance with database
ID of $databaseId."
  $licenseResult | Format-Table -AutoSize
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Updating license for database id 1...
New License Allocation result for the database instance with database ID of 1.

serverName overLicensed vmLicenseProduct performanceLicenseProduct
---------- ------------ ---------------- -------------------------
DEV-DPA           False DPAVM            DPACAT2

Updating license for database id 1...
New License Allocation result for the database instance with database ID of 1.

serverName overLicensed vmLicenseProduct performanceLicenseProduct
---------- ------------ ---------------- -------------------------
DEV-DPA           False                  DPACAT2
```

Annotation examples

The examples below show how to use all Annotation calls.

Get a list of annotations for the last 30 days

```
$databaseId = 1
$annotationURL = $baseURL + "databases/$databaseId/annotations"

# Dates are in ISO 8601 format ( 2018-12-31T12:00:00.000-07:00 )
$endTime = Get-Date
$startTime = $endTime.AddDays(-30)
$startTime = [System.Web.HttpUtility]::UrlEncode($startTime.ToString("yyyy-MM-
ddTHH\:mm\:ss.fffzzz"))
$endTime = [System.Web.HttpUtility]::UrlEncode($endTime.ToString("yyyy-MM-
ddTHH\:mm\:ss.fffzzz"))

$request = [System.UriBuilder]$annotationURL
$request.Query = "startTime=$startTime&endTime=$endTime"
$annotationURL = $request.Uri

Try {
  Write-Host "Getting Annotations for the last 30 days..."
  $annotationListJSON = Invoke-RestMethod -Method Get -Uri $annotationURL -Headers
$dpaHeader -TimeoutSec 60
  $annotationList = $annotationListJSON.data
  $annotationList | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}

# This will print out data like this:
Getting Annotations for the last 30 days...

 id title          description                  createdBy time
 type
 -- -----          -----------                  --------- ----
 ----
 98 Test Title API Test Event created by DPA API DPA API   2018-12-04T18:13:04-
07:00 Custom
 99 Test Title API Test Event created by DPA API DPA API   2018-12-04T18:14:27-
07:00 Custom
100 Test Title API Test Event created by DPA API DPA API   2018-12-04T18:16:46-
07:00 Custom
etc.
```

Create a new annotation

```
$databaseId = 1
$annotationURL = $baseURL + "databases/$databaseId/annotations"
$createTime = Get-Date
# Dates are in ISO 8601 format, no millis ( 2018-12-31T12:00:00-07:00 )
$createTime = $createTime.ToString("yyyy-MM-ddTHH\:mm\:sszzz")
$body = @{"title" = "API Test Title";
                    "description" = "API Test Description";
                    "createdBy" = "Test API User";
                    "time" = "$createTime"} | ConvertTo-Json
Try {
  Write-Host "Creating Annotation..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $annotationURL -Body $body -Method POST
-Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $annotationId = $dpaResponse.id
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Creating Annotation...

 id title          description          createdBy     time
type
 -- -----          -----------          ---------     ----                       ---
-
148 API Test Title API Test Description Test API User 2019-01-03T15:20:36-07:00 API
```

Delete an annotation

```
$databaseId = 1
$annotationId = 148
$annotationURL = $baseURL + "databases/$databaseId/annotations/$annotationId"
Try {
  Write-Host "Deleting Annotation with id of $annotationID..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $annotationURL  -Method DELETE -Headers
$dpaHeader -TimeoutSec 60
```

```
  Write-Host "Annotation with id of $annotationID deleted`r`n"
}
Catch {
  $_.Exception.ToString()
}

# This will print out data like this:
Deleting Annotation with id of 148...
Annotation with id of 148 deleted
```

Database Registration examples

The examples below show how to use all Database Registration calls.

Register and unregister a SQL Server database instance for monitoring

This example registers a new SQL Server database instance, waits 60 seconds, and then unregisters the database instance.

```
#---------------------------------------------------------
# Register a SQL Server database instance for monitoring.
#---------------------------------------------------------
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "SQLSERVER";
          "serverName" = "127.0.0.1";
          "port" = "1433";
          "sysAdminUser" = "sa";
          "sysAdminPassword" = "Password";
          "monitoringUser" = "dpa_test_m";
          "monitoringUserPassword" = "Password";
          "monitoringUserIsNew" = $true;
          "displayName" = "DPA_SQL2K12"} | ConvertTo-Json
Try {
  Write-Host "Registering Database..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $newDbId = $dpaResponse.databaseId
}
Catch {
```

```
  $_.Exception.ToString()
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-------------------------------------------------------------
# Un-register the SQL Server database instance.
#-------------------------------------------------------------
if ($newDbId) {
  $registrationURL = $baseURL + "databases/unregister-monitor"
  $body = @{"databaseId" = $newDbId;
            "removeMonitoringUser" = $true;
            "removeDatabaseObjects" = $true;
            "sysAdminUser" = "sa";
            "sysAdminPassword" = "Password"} | ConvertTo-Json
  Try {
    Write-Host "Registering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
  }
  Catch {
    $_.Exception.ToString()
  }
}

# This will print out data like this:
Registering Database...

databaseId result
---------- ------
        70 SUCCESS

Waiting 60 seconds...

Unregistering Database...

databaseId result
---------- ------
```

```
        70 SUCCESS
```

Register and unregister an Oracle database instance for monitoring

This example registers a new Oracle database instance, waits 60 seconds, and then unregisters the database instance.

```
#-------------------------------------------------------------
# Register an Oracle database instance for monitoring.
#-------------------------------------------------------------
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "ORACLE";
          "serverName" = "127.0.0.1";
          "serviceNameOrSID" = "DPA_ORA11R1";
          "port" = "1521";
          "sysAdminUser" = "system";
          "sysAdminPassword" = "Password";
          "sysPassword" = "Password";
          "monitoringUser" = "dpa_test_m";
          "monitoringUserPassword" = "Password";
          "monitoringUserIsNew" = $true;
          "monitoringUserTableSpace" = "USERS";
          "monitoringUserTempTableSpace" = "TEMP";
          "oracleEBusinessEnabled" = $false;
          "displayName" = "DPA_ORA11R1"} | ConvertTo-Json
Try {
  Write-Host "Registering Database..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $newDbId = $dpaResponse.databaseId
}
Catch {
  $_.Exception.ToString()
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-------------------------------------------------------------
```

```
# Un-register the Oracle database instance.
#-----------------------------------------------------------
if ($newDbId) {
  $registrationURL = $baseURL + "databases/unregister-monitor"
  $body = @{"databaseId" = $newDbId;
            "removeMonitoringUser" = $true;
            "removeDatabaseObjects" = $true;
            "sysAdminUser" = "system";
            "sysAdminPassword" = "Password"} | ConvertTo-Json
  Try {
    Write-Host "Unregistering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
  }
  Catch {
    $_.Exception.ToString()
  }
}

# This will print out data like this:
Registering Database...

databaseId result
---------- ------
        71 SUCCESS

Waiting 60 seconds...

Unregistering Database...

databaseId result
---------- ------
        71 SUCCESS
```

Register and unregister a MySQL database instance for monitoring

This example registers a new MySQL database instance, waits 60 seconds, and then unregisters the database instance.

```
#------------------------------------------------------------
# Register a MySQL database instance for monitoring.
#------------------------------------------------------------
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "MYSQL";
          "serverName" = "127.0.0.1";
          "port" = "3306";
          "sysAdminUser" = "root";
          "sysAdminPassword" = "Password";
          "monitoringUser" = "dpa_test_m";
          "monitoringUserPassword" = "Password";
          "monitoringUserIsNew" = $true;
          "displayName" = "DPA_MYSQL56"} | ConvertTo-Json
Try {
  Write-Host "Registering Database..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $newDbId = $dpaResponse.databaseId
}
Catch {
  $_.Exception.ToString()
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#------------------------------------------------------------
# Un-register the MySQL database instance.
#------------------------------------------------------------
if ($newDbId) {
       $registrationURL = $baseURL + "databases/unregister-monitor"
       $body = @{"databaseId" = $newDbId;
              "removeMonitoringUser" = $true;
              "removeDatabaseObjects" = $true;
              "sysAdminUser" = "root";
              "sysAdminPassword" = "Password"} | ConvertTo-Json
  Try {
    Write-Host "Registering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
```

```
POST -Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
  }
  Catch {
    $_.Exception.ToString()
  }
}

# This will print out data like this:
Registering Database...

databaseId result
---------- ------
        72 SUCCESS

Waiting 60 seconds...

Unregistering Database...

databaseId result
---------- ------
        72 SUCCESS
```

Database Custom Properties examples

The examples below show how to use Database Custom Properties calls. Custom property values can be included in custom email templates for alert notifications.

Create a custom property

This script creates a custom property and defines its name and description.

```
$propertyName = "Location"
$propertyDescription = "Location of the database server"
$createPropertyURL = $baseURL + "databases/properties"
$body = @{"name" = $propertyName; "description" = $propertyDescription;} |
ConvertTo-Json

Try {
  Write-Host "Creating custom property ..."
```

```
  $dpaResponseJSON = Invoke-RestMethod -Uri $createPropertyURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $propertyId = $dpaResponse.id
} Catch {
    $_.Exception.ToString()
}

#This will print out data like this:
id    name     description
--    ----     -----------
10434 Location Location of the database server
```

Create a custom property value

This script creates a value for the custom property created by the previous script.

```
property_id = 1
property_value = "New York"
$createValueURL = $baseURL + "databases/properties/" + $propertyId + "/values"
$body = $propertyValue | ConvertTo-Json

Try {
  Write-Host "Creating custom property value ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $createValueURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
} Catch {
  $_.Exception.ToString()
}

#This will print out data like this:
id value
-- -----
1  "New York"
```

Assign a property value to a monitored database instance

This script assigns a property value to a monitored database instance.

```
property_id = 1
property_value_id = 1
$assignPropertyValueURL = $baseURL + "databases/" + $databaseId + "/properties/" +
$propertyId + "/values/" + $propertyValueId;

Try {
  Write-Host "Assigning custom property value ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $assignPropertyValueURL -Method POST -
Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  Write-Host "Custom property value assigned to the DB with ID: $databaseId`r`n"
} Catch {
  $_.Exception.ToString()
}

#This will print out data like this:
Custom property value assigned to the DB with ID: 1
```

Get all information about properties

This script returns information about all custom properties and their values.

```
$getPropertiesURL = $baseURL + "databases/properties?require=assignment"

Try {
  Write-Host "Getting custom property information ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $getPropertiesURL -Method GET -Headers
$dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
} Catch {
  $_.Exception.ToString()
}

#This will print out data like this:
Getting custom property information ...
id name     description                    values
    unassigned
-- ----     -----------                    ------
    ----------
```

```
1  Location Location of the database server {@{id=1; value="New York"; assignment=
[1]}} {4, 8}
```

Delete a custom property

This script deletes a custom property.

```
property_id = 1
$deletePropertyURL = $baseURL + "databases/properties/" + $propertyId

Try {
  Write-Host "Deleting custom property ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $deletePropertyURL -Method DELETE -
Headers $dpaHeader -TimeoutSec 60
  Write-Host "Custom property with id of $propertyID deleted`r`n"
} Catch {
  $_.Exception.ToString()
}

#This will print out data like this:
Custom property with id of 1 deleted
```

Full working script

The following script combines all of the examples shown above into a script that can be run.

```
#---------------------------------------------------------
# Examples:
# - Get an access token
# - Database Monitor Examples
#   - Get Monitor Information for a single database
#   - Start or Stop monitoring a database instance given its database ID
#   - Get Monitor Information for all databases
#   - Start monitoring for all database instances
#   - Stop monitoring for all database instances
#   - ERROR: Get Monitor Information for a database that doesn't exist
#   - ERROR: Start a database that doesn't exist
# - Licensing Examples
#   - Get the currently installed license information
#   - Get License Information for a single database
#   - Update License Information for a single database
```

```
# - Annotation Examples
#   - Gets a List of annotations for the last 30 days
#   - Create a new annotation
#   - Delete an annotation
# - Registration Examples
#   - Register a MySQL database instance for monitoring
#   - Un-register the MySQL database instance
#-------------------------------------------------------------

#-------------------------------------------------------------
# Configure the variables below for the DPA Host
#-------------------------------------------------------------
$baseURL = "https://localhost:8124/iwc/api/"
$refreshToken = "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJNeVRva2VuIiwiaXN..."
$databaseId = 1
#-------------------------------------------------------------
# Nothing to configure below this line
#-------------------------------------------------------------

#-------------------------------------------------------------
# Function to parse the Response Data from DPA and print
# out the error information
#-------------------------------------------------------------
Function handleError ($thisError) {
  Write-Host "-------------------------------------------------------------------------"
-ForegroundColor Red
  Write-Host "Caught Exception at line:" $_.InvocationInfo.ScriptLineNumber -
ForegroundColor Red
  if ($_.Exception.Response) {
    $streamReader = [System.IO.StreamReader]::new
($_.Exception.Response.GetResponseStream())
    $errResp = $streamReader.ReadToEnd()
    $streamReader.Close()
  }
  if ($errResp) {
    # This will format the JSON
    $errResp = $errResp | ConvertFrom-Json | ConvertTo-Json -Depth 100
    Write-Host $thisError.Exception.Message -ForegroundColor Red
    Write-Host "Response:`r`n$errResp" -ForegroundColor Red
  }
  else {
```

```
    Write-Host $_.Exception.ToString() -ForegroundColor Red
  }
  Write-Host "------------------------------------------------------------------------"
-ForegroundColor Red
}

#-------------------------------------------------------------
# Adding certificate exception to prevent API errors
# Uncomment this if you are getting trust errors and would
# like to run with self-signed certificates.
#-------------------------------------------------------------
# add-type @"
#   using System.Net;
#   using System.Security.Cryptography.X509Certificates;
#   public class TrustAllCertsPolicy : ICertificatePolicy {
#     public bool CheckValidationResult(
#       ServicePoint srvPoint, X509Certificate certificate,
#       WebRequest request, int certificateProblem) {
#       return true;
#     }
#   }
# "@
# [System.Net.ServicePointManager]::CertificatePolicy = New-Object
TrustAllCertsPolicy

#-------------------------------------------------------------
# Get an access token
#-------------------------------------------------------------
$authTokenURL = $baseURL + "security/oauth/token"
$body = @{"grant_type" = "refresh_token"
          "refresh_token" = "$refreshToken"}
Try {
  Write-Host "Getting Access Token..."
  $dpaAuthResponse = Invoke-RestMethod -Uri $authTokenURL -Method POST -Body $body
  $dpaAuthResponse | Format-List
}
Catch {
  handleError $Error[0]
  Write-Host 'Error getting authentication token, cannot continue' -ForegroundColor
Red
  return
```

```
}

# If successful we will create our headers to be used for all API calls
$tokenType = $dpaAuthResponse.token_type
$accessToken = $dpaAuthResponse.access_token
$dpaHeader = @{}
$dpaHeader.Add("Accept", "application/json")
$dpaHeader.Add("Content-Type", "application/json;charset=UTF-8")
$dpaHeader.Add("Authorization", "$tokenType $accessToken")

#-------------------------------------------------------------
# Database Monitor Examples
#-------------------------------------------------------------

# Get Monitor Information for a single database
$monitorURL = $baseURL + "databases/$databaseId/monitor-information"
Try {
  Write-Host "Get Monitor Information for database with id of $databaseId..."
  $monitorJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers $dpaHeader
-TimeoutSec 60
  $monitor = $monitorJSON.data
  $monitor | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}

# Start or Stop monitoring a database instance given its database ID.
# If it is already running stop it and then restart it
# If it is not running start it and then stop it
$monitorURL = $baseURL + "databases/$databaseId/monitor-status"
if ($monitor.monitorState -eq "Monitor Running") {
  $changeCommand = "STOP"
  $revertCommand = "START"
}
elseif ($monitor.monitorState -eq "Monitor Stopped") {
  $changeCommand = "START"
  $revertCommand = "STOP"
}
Try {
  Write-Host "$changeCommand Monitor for database $databaseId..."
```

```
  $command = @{"command" = $changeCommand} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
  $result = $monitorJSON.data
  Write-Host "Result: $result"
  Write-Host "Waiting 15 seconds...`r`n"
  Start-Sleep -s 15

  Write-Host "$revertCommand Monitor for database $databaseId..."
  $command = @{"command" = $revertCommand} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
  $result = $monitorJSON.data
  Write-Host "Result: $result"
  Write-Host "Waiting 15 seconds...`r`n"
  Start-Sleep -s 15
}
Catch {
  handleError $Error[0]
}

# Get Monitor Information for all databases
$monitorURL = $baseURL + "databases/monitor-information"
Try {
  Write-Host "Get Monitor Information for all databases..."
  $monitorListJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers
$dpaHeader -TimeoutSec 60
  $monitorList = $monitorListJSON.data
  $monitorList | Format-Table -AutoSize

  # Keep a list of running or started monitors to be used later
  $runningIds = @()
  foreach ($monitor in $monitorList) {
    if ($monitor.monitorState -eq "Monitor Running" -or
      $monitor.monitorState -eq "Monitor Start No License" -or
      $monitor.monitorState -like '*Start*')
    {
      $runningIds += $monitor.dbId
    }
  }
  Write-Host "Running Monitors: $runningIds`r`n"
```

```
}
Catch {
  handleError $Error[0]
}

# Start monitoring all database instances.
$monitorURL = $baseURL + "databases/monitor-status"
Try {
  Write-Host "Starting all Monitors..."
  $command = @{"command" = "START"} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
  $result = $monitorJSON.data
  Write-Host "Result: $result"
  Write-Host "Waiting 30 seconds...`r`n"
  Start-Sleep -s 30
}
Catch {
  handleError $Error[0]
}

# Stop monitoring all database instances.
Try {
  Write-Host "Stopping all Monitors..."
  $command = @{"command" = "STOP"} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
  $result = $monitorJSON.data
  Write-Host "Result: $result"
  Write-Host "Waiting 30 seconds...`r`n"
  Start-Sleep -s 30
}
Catch {
  handleError $Error[0]
}

# Try to put it back the way we found it by restarting the ones that were running
$command = @{"command" = "START"} | ConvertTo-Json
foreach ($dbId in $runningIds) {
  Try {
    $monitorURL = $baseURL + "databases/$dbId/monitor-status"
```

```
    Write-Host "Starting Monitor for database $dbId..."
    $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
    $result = $monitorJSON.data
    Write-Host "Result: $result`r`n"
  }
  Catch {
    handleError $Error[0]
  }
}

# Update the monitor database user password (Un-comment to use)
# $monitorURL = $baseURL + "databases/$databaseId/update-password"
# Try {
#   Write-Host "Update the Monitor password for database $databaseId..."
#   $command = @{"password" = "NewPassword!"} | ConvertTo-Json
#   $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
#   $result = $monitorJSON.data
#   Write-Host "Result: $result`r`n"
# }
# Catch {
#   handleError $Error[0]
# }

# Try to cause some errors...

# Get Monitor Information for a database that doesn't exist
$monitorURL = $baseURL + "databases/-1/monitor-information"
Try {
  Write-Host "Get Monitor Information for invalid database..."
  $monitorJSON = Invoke-RestMethod -Method Get -Uri $monitorURL -Headers $dpaHeader
-TimeoutSec 60
}
Catch {
  handleError $Error[0]
}

# Start a database that doesn't exist
$monitorURL = $baseURL + "databases/-1/monitor-status"
Try {
```

```
  Write-Host "START Monitor for invalid database..."
  $command = @{"command" = "START"} | ConvertTo-Json
  $monitorJSON = Invoke-RestMethod -Method Put -Uri $monitorURL -Body $command -
Headers $dpaHeader -TimeoutSec 60
}
Catch {
  handleError $Error[0]
}

#-------------------------------------------------------------
# Licensing Examples
#-------------------------------------------------------------

# Get the currently installed license information
$licenseURL = $baseURL + "databases/licenses/installed"

Try {
  Write-Host "Getting Installed license information with total amounts available
for use and total amounts used..."
  $licenseListJSON = Invoke-RestMethod -Method Get -Uri $licenseURL -Headers
$dpaHeader -TimeoutSec 60
  $licenseList = $licenseListJSON.data
  $licenseList | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}

# Get License Information for a single database
$licenseURL = $baseURL + "databases/$databaseId/licenses"
Try {
  Write-Host "Getting current license information for the database instance with
database ID of $databaseId."
  $licenseListJSON = Invoke-RestMethod -Method Get -Uri $licenseURL -Headers
$dpaHeader -TimeoutSec 60
  $licenseInfo = $licenseListJSON.data
  $licenseInfo | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}
```

```
# This will Update License Information for a single database setting the
# Performance License and the VM License to what it currently is.
# It should succeed but it should make no changes.
$dbProduct = $licenseInfo.performanceLicenseProduct
$vmProduct = $licenseInfo.vmLicenseProduct
$licenseAllocation = @{"performanceLicenseProduct" = $dbProduct;
                       "vmLicenseProduct" = $vmProduct} | ConvertTo-Json
Try {
  Write-Host "Updating license for database id $databaseId..."
  $licenseResultJSON = Invoke-RestMethod -Method Put -Uri $licenseURL -Body
$licenseAllocation -Headers $dpaHeader -TimeoutSec 60
  $licenseResult = $licenseResultJSON.data
  Write-Host "New License Allocation result for the database instance with database
ID of $databaseId."
  $licenseResult | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}

#------------------------------------------------------------
# Annotation Examples
#------------------------------------------------------------

# Gets a List of annotations for the last 30 days
$annotationURL = $baseURL + "databases/$databaseId/annotations"

# Dates are in ISO 8601 format ( 2018-12-31T12:00:00-07:00 )
$endTime = Get-Date
$startTime = $endTime.AddDays(-30)
$startTime = [System.Web.HttpUtility]::UrlEncode($startTime.ToString("yyyy-MM-
ddTHH\:mm\:ss.fffzzz"))
$endTime = [System.Web.HttpUtility]::UrlEncode($endTime.ToString("yyyy-MM-
ddTHH\:mm\:ss.fffzzz"))

$request = [System.UriBuilder]$annotationURL
$request.Query = "startTime=$startTime&endTime=$endTime"
$annotationURL = $request.Uri

Try {
```

```
  Write-Host "Getting Annotations for the last 30 days..."
  $annotationListJSON = Invoke-RestMethod -Method Get -Uri $annotationURL -Headers
$dpaHeader -TimeoutSec 60
  $annotationList = $annotationListJSON.data
  $annotationList | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}

# Create a new annotation
# Dates are in ISO 8601 format, no millis ( 2018-12-31T12:00:00-07:00 )
$annotationURL = $baseURL + "databases/$databaseId/annotations"
$createTime = Get-Date
$createTime = $createTime.ToString("yyyy-MM-ddTHH\:mm\:sszzz")
$body = @{"title" = "API Test Title";
          "description" = "API Test Description";
          "createdBy" = "Test API User";
          "time" = "$createTime"} | ConvertTo-Json
Try {
  Write-Host "Creating Annotation..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $annotationURL -Body $body -Method POST
-Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $annotationId = $dpaResponse.id
}
Catch {
  handleError $Error[0]
}

# Delete an annotation
if ($annotationId) {
  $annotationURL = $baseURL + "databases/$databaseId/annotations/$annotationId"
  Try {
    Write-Host "Deleting Annotation with id of $annotationID..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $annotationURL  -Method DELETE -
Headers $dpaHeader -TimeoutSec 60
    Write-Host "Annotation with id of $annotationID deleted`r`n"
  }
  Catch {
```

```
    handleError $Error[0]
    }
  }

#-------------------------------------------------------------
# Registration Examples
#-------------------------------------------------------------

#-------------------------------------------------------------
# Register a SQL Server database instance for monitoring.
#-------------------------------------------------------------
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "SQLSERVER";
          "serverName" = "127.0.0.1";
          "port" = "1433";
          "sysAdminUser" = "sa";
          "sysAdminPassword" = "Password";
          "monitoringUser" = "dpa_test_m";
          "monitoringUserPassword" = "Password";
          "monitoringUserIsNew" = $true;
          "displayName" = "DPA_SQL2K12"} | ConvertTo-Json
Try {
  Write-Host "Registering Database..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $newDbId = $dpaResponse.databaseId
}
Catch {
  handleError $Error[0]
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-------------------------------------------------------------
# Un-register the SQL Server database instance.
#-------------------------------------------------------------
if ($newDbId) {
  $registrationURL = $baseURL + "databases/unregister-monitor"
```

```
  $body = @{"databaseId" = $newDbId;
            "removeMonitoringUser" = $true;
            "removeDatabaseObjects" = $true;
            "sysAdminUser" = "sa";
            "sysAdminPassword" = "Password"} | ConvertTo-Json
  Try {
    Write-Host "Unregistering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
  }
  Catch {
    handleError $Error[0]
  }
}

#------------------------------------------------------------
# Register an Oracle database instance for monitoring.
#------------------------------------------------------------
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "ORACLE";
          "serverName" = "127.0.0.1";
          "serviceNameOrSID" = "DPA_ORA11R1";
          "port" = "1521";
          "sysAdminUser" = "system";
          "sysAdminPassword" = "Password";
          "sysPassword" = "Password";
          "monitoringUser" = "dpa_test_m";
          "monitoringUserPassword" = "Password";
          "monitoringUserIsNew" = $true;
          "monitoringUserTableSpace" = "USERS";
          "monitoringUserTempTableSpace" = "TEMP";
          "oracleEBusinessEnabled" = $false;
          "displayName" = "DPA_ORA11R1"} | ConvertTo-Json
Try {
  Write-Host "Registering Database..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
```

```
  $newDbId = $dpaResponse.databaseId
}
Catch {
  handleError $Error[0]
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#-------------------------------------------------------------
# Un-register the Oracle database instance.
#-------------------------------------------------------------
if ($newDbId) {
       $registrationURL = $baseURL + "databases/unregister-monitor"
       $body = @{"databaseId" = $newDbId;
              "removeMonitoringUser" = $true;
              "removeDatabaseObjects" = $true;
              "sysAdminUser" = "system";
              "sysAdminPassword" = "Password"} | ConvertTo-Json
  Try {
    Write-Host "Unregistering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
  }
  Catch {
    handleError $Error[0]
  }
}

#-------------------------------------------------------------
# Register a MySQL database instance for monitoring.
#-------------------------------------------------------------
$registrationURL = $baseURL + "databases/register-monitor"
$body = @{"databaseType" = "MYSQL";
          "serverName" = "127.0.0.1";
          "port" = "3306";
          "sysAdminUser" = "root";
          "sysAdminPassword" = "Password";
          "monitoringUser" = "dpa_test_m";
```

```
          "monitoringUserPassword" = "Password";
          "monitoringUserIsNew" = $true;
          "displayName" = "DPA_MYSQL56"} | ConvertTo-Json
Try {
  Write-Host "Registering Database..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $newDbId = $dpaResponse.databaseId
}
Catch {
  handleError $Error[0]
}

Write-Host "Waiting 60 seconds...`r`n"
Start-Sleep -s 60

#------------------------------------------------------------
# Un-register the MySQL database instance.
#------------------------------------------------------------
if ($newDbId) {
  $registrationURL = $baseURL + "databases/unregister-monitor"
  $body = @{"databaseId" = $newDbId;
            "removeMonitoringUser" = $true;
            "removeDatabaseObjects" = $true;
            "sysAdminUser" = "root";
            "sysAdminPassword" = "Password"} | ConvertTo-Json
  Try {
    Write-Host "Unregistering Database..."
    $dpaResponseJSON = Invoke-RestMethod -Uri $registrationURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
    $dpaResponse = $dpaResponseJSON.data
    $dpaResponse | Format-Table -AutoSize
  }
  Catch {
    handleError $Error[0]
  }
}

#------------------------------------------------------------
```

```
# Custom Property Examples
#-------------------------------------------------------------
# Create custom property

$propertyName = "Location"
$propertyDescription = "Location of the database server"
$createPropertyURL = $baseURL + "databases/properties"
$body = @{"name" = $propertyName; "description" = $propertyDescription;} |
ConvertTo-Json

Try {
  Write-Host "Creating custom property ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $createPropertyURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $propertyId = $dpaResponse.id
}
Catch {
  handleError $Error[0]
}

#Create value of the custom property

$propertyValue = "New York"
$createValueURL = $baseURL + "databases/properties/" + $propertyId + "/values"
$body = $propertyValue | ConvertTo-Json

Try {
  Write-Host "Creating custom property value ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $createValueURL -Body $body -Method
POST -Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  $propertyValueId = $dpaResponse.id
}
Catch {
  handleError $Error[0]
}

#Assign property value to DB
```

```
$assignPropertyValueURL = $baseURL + "databases/" + $databaseId + "/properties/" +
$propertyId + "/values/" + $propertyValueId;

Try {
  Write-Host "Assigning custom property value ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $assignPropertyValueURL -Method POST -
Headers $dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
  Write-Host "Custom property value assigned to the DB with ID: $databaseId`r`n"
}
Catch {
  handleError $Error[0]
}

#Get all information about properties (including DB assignment)

$getPropertiesURL = $baseURL + "databases/properties?require=assignment"

Try {
  Write-Host "Getting custom property information ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $getPropertiesURL -Method GET -Headers
$dpaHeader -TimeoutSec 60
  $dpaResponse = $dpaResponseJSON.data
  $dpaResponse | Format-Table -AutoSize
}
Catch {
  handleError $Error[0]
}

#Delete custom property

$deletePropertyURL = $baseURL + "databases/properties/" + $propertyId

Try {
  Write-Host "Deleting custom property ..."
  $dpaResponseJSON = Invoke-RestMethod -Uri $deletePropertyURL -Method DELETE -
Headers $dpaHeader -TimeoutSec 60
  Write-Host "Custom property with id of $propertyID deleted`r`n"
}
```

```
Catch {
  handleError $Error[0]
}

# End of script
```

The scripts are not supported under any SolarWinds support program or service. The scripts are provided AS IS without warranty of any kind. SolarWinds further disclaims all warranties including, without limitation, any implied warranties of merchantability or of fitness for a particular purpose. The risk arising out of the use or performance of the scripts and documentation stays with you. In no event shall SolarWinds or anyone else involved in the creation, production, or delivery of the scripts be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use the scripts or documentation.

Restart or configure DPA

This section provides information about administrative tasks in DPA.

Stop and start DPA

Stop and start DPA on a Windows server

- To stop DPA, use the Windows Control Panel to stop the `Ignite PI Server` service.
- To restart DPA, use the Windows Control Panel to start the `Ignite PI Server` service again.

> If the service does not start or does not continue running when you start the service, you can also start it by running the `startup.bat` script file from the DPA directory. However, if you start DPA using this script, you **must** leave the command window open. Closing the command window stops DPA.
>
> If you have issues with the `Ignite PI Server` service, contact SolarWinds Support.

Stop and start DPA on a server with a Linux-based OS

- To stop DPA, run the following command from the DPA directory:

  ```
  shutdown.sh
  ```

  For example, using the default DPA directory:

  ```
  /home/solarwinds/dpa_V_v/shutdown.sh
  ```

- To restart DPA, run the following command from the DPA directory:

  ```
  startup.sh
  ```

  For example, using the default DPA directory:

  ```
  /home/solarwinds/dpa_V_v/startup.sh
  ```

  > DPA must be running at all times to collect monitoring data. If you plan to log out of the account from which you start SolarWinds DPA, run the following command instead. This prevents the SolarWinds DPA process from exiting when you log out.
  >
  > ```
  > nohup ./startup.sh
  > ```

Set advanced DPA configuration options

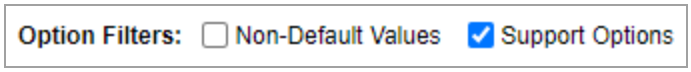
You can use advanced options to change DPA's default behavior. For example, you can:

- Change the default expiration times for access and refresh tokens
- Change the Warning and Critical thresholds for anomaly detection
- Change the default values that DPA uses to check for table tuning best practices

In most cases, you will change these options when instructed to do so by SolarWinds Support or based on information from another topic of this administration guide (such as the linked topics above).

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Configuration, click Advanced Options.

   The System Options tab lists options that apply to all database instances. The list includes a description of each option.
3. If you are setting an option that applies to a single database instance, click DB Instance Options and select the database instance.
4. If the option is a Support option, select Support Options in the upper-right corner to display those options.

   Support options are not displayed by default. In most cases, they should be changed only if you are instructed to do by a Support representative.
5. Click the name of an option to open the Edit Option dialog.
6. To change an option value, enter the New Value and click Update.

Enable SNMP Monitoring in SCOM

You can set up DPA to use SNMP to monitor System Center Operations Manager (SCOM).

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Users & Contacts, click Contact Management.
3. Click Create SNMP Contact.
4. Enter the SCOM host IP address and port in the Trap Receiver fields. The default port is 162.

5. Enter the community string that was set up on the SNMP Service on the SCOM host.

   This string is case sensitive.

6. On the DPA server, make sure the SNMP service is running and the community string set matches the string you entered in the SNMP Contact window.

Configure password protection for DPA features that allow custom SQL

To prevent unauthorized users from entering malicious SQL, you can configure password protection for DPA features that allow users to enter custom SQL. These features include:

- Custom metrics
- Custom alerts
- The database query tool
- The Update DB Instance Connection Wizard

Enable password protection

When password protection is enabled, users are prompted for the specified password when they test or save a custom metric or custom alert, and when they open the database query tool or the Updated DB Instance Connection Wizard.

1. Open the `sqlauth.xml` file in a text editor. This file is located in the following directory:

   DPA-install-dir`\iwc\tomcat\ignite_config\iwc\security`

   The default location is:

   `C:\Program Files\SolarWinds\DPA\iwc\tomcat\ignite_config\iwc\security`

2. Enter the password as the value of the `<entry key="sql.authentication.password">` setting. The password must contain:

   - At least 7 characters
   - At least 1 numeric character

   For example:

   ```
   <entry key="sql.authentication.password">MyPassword1</entry>
   ```

If the password includes special characters (such as `&`, `<`, or `>`), enclose the password with CDATA as follows:

```
<entry key="sql.authentication.password"><![CDATA[My&Password1]]></entry>
```

3. Save the file.

   Changes take effect immediately. The password in the `sqlauth.xml` file is encrypted the first time DPA prompts a user to enter it.

Disable password protection

Password protection is disabled by default. If you enable it and then want to disable it again, complete the following steps.

1. Open the `sqlauth.xml` file in a text editor. This file is located in the following directory:

   DPA-install-dir`\iwc\tomcat\ignite_config\iwc\security`

   The default location is:

   `C:\Program Files\SolarWinds\DPA\iwc\tomcat\ignite_config\iwc\security`

2. Remove the value of the `<entry key="sql.authentication.password">` setting. For example:

   ```
   <entry key="sql.authentication.password"></entry>
   ```

3. Save the file.

   Changes take effect immediately.

Configure the mail server used to send DPA emails

To send email messages (such as alert notifications or scheduled reports), DPA must be able to connect to a mail (SMTP) server. By default, DPA uses a third-party SMTP email service (AuthSMTP). Alternatively, you can choose to use:

- An embedded mail server that runs inside of DPA.

  > In some environments, this server might be blocked from sending email by firewalls or other SMTP restrictions.

- Your company's mail server.

To change the mail server or update connection information, complete the following steps.

1. From the DPA menu in the upper-right corner, click Options.
2. Under Administration > Configuration, click Configure Mail Server.

   The Mail Server Configuration page opens.
3. Under Choose a Mail Server, select the mail server you want DPA to use.

4. If you selected Company Mail Server, enter connection information under Company Mail Server Settings.

   If DPA is configured to use credentials stored in CyberArk, this page displays a field for the CyberArk credentials query instead of fields for a user name and password.
5. Click Send Test E-mail to test the settings.
6. Click Save.

You do **not** need to restart DPA after you configure the mail server.