

# Alibaba Cloud

## AnalyticDB for PostgreSQL Instances

Document Version: 20220223

# Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

# Document conventions

| Style  | Description   | Example   |
|--|---|---|
|  <b>Danger</b>  | A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results. |  <b>Danger:</b><br>Resetting will result in the loss of user configuration data.                                       |
|  <b>Warning</b> | A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results. |  <b>Warning:</b><br>Restarting will cause business interruption. About 10 minutes are required to restart an instance. |
|  <b>Notice</b>  | A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.      |  <b>Notice:</b><br>If the weight is set to 0, the server no longer receives new requests.                              |
|  <b>Note</b>  | A note indicates supplemental instructions, best practices, tips, and other content.  |  <b>Note:</b><br>You can use Ctrl + A to select all files.  |
| >  | Closing angle brackets are used to indicate a multi-level menu cascade.   | Click <b>Settings</b> > <b>Network</b> > <b>Set network type</b> .  |
| <b>Bold</b>  | Bold formatting is used for buttons, menus, page names, and other UI elements.  | Click <b>OK</b> .   |
| Courier font   | Courier font is used for commands   | Run the <code>cd /d C:/window</code> command to enter the Windows system folder.  |
| <i>Italic</i>  | Italic formatting is used for parameters and variables.   | <code>bae log list --instanceid</code><br><i>Instance_ID</i>  |
| [ ] or [a b]   | This format is used for an optional value, where only one item can be selected.   | <code>ipconfig [-all -t]</code>   |
| { } or {a b}   | This format is used for a required value, where only one item can be selected.  | <code>switch {active stand}</code>  |

---

# Table of Contents

|  |    |
|--|----|
| 1.Create an AnalyticDB for PostgreSQL instance .....                     | 06 |
| 2.Node configuration change .....  | 11 |
| 2.1. Change the specifications of compute nodes .....                    | 11 |
| 3.Create a Basic Edition instance .....                                  | 15 |
| 4.Create a database account .....  | 17 |
| 5.Configure an IP address whitelist .....                                | 19 |
| 6.Use client tools to connect to an instance .....                       | 22 |
| 7.Enable SQL audit .....   | 30 |
| 8.Instance locking .....   | 32 |
| 8.1. Instance locking caused by disk usage .....                         | 32 |
| 9.Restart an AnalyticDB for PostgreSQL instance .....                    | 34 |
| 10.Configure disk encryption for an AnalyticDB for PostgreSQL i... ..... | 35 |
| 11.Network Connection .....  | 37 |
| 11.1. Manage public endpoints .....                                      | 37 |
| 11.2. Release an Internet IP address .....                               | 38 |
| 11.3. Endpoints of an instance and its primary coordinator nod... .....  | 39 |
| 12.Version Management .....  | 41 |
| 12.1. Upgrade the engine version .....                                   | 41 |
| 12.2. Query the minor engine version .....                               | 41 |
| 12.3. Compatibility comparisons between AnalyticDB for Postgr... .....   | 42 |
| 12.4. Check for incompatibility between AnalyticDB for Postgre... .....  | 47 |
| 13.Backup and restoration .....  | 54 |
| 13.1. Overview .....   | 54 |
| 13.2. Back up data .....   | 54 |
| 13.3. Restore data .....   | 56 |
| 14.Tags .....  | 59 |

---

|   |    |
|---|----|
| 14.1. Add tags to instances .....                                 | 59 |
| 14.2. Unbind a tag from an AnalyticDB for PostgreSQL instanc..... | 60 |
| 14.3. Filter AnalyticDB for PostgreSQL instances by tag .....     | 61 |
| 15. Diagnostics and optimization .....                            | 62 |
| 15.1. View slow SQL queries .....                                 | 62 |
| 15.2. View SQL distribution statistics .....                      | 64 |
| 16. CloudMonitor .....  | 66 |
| 16.1. View monitoring data and configure an alert rule .....      | 66 |
| 16.2. Configure an alert template .....                           | 68 |
| 16.3. Metrics and system events .....                             | 70 |
| 17. Release an instance .....                                     | 74 |
| 18. Configure parameters .....                                    | 75 |

# 1. Create an AnalyticDB for PostgreSQL instance

This topic describes how to create an instance in the AnalyticDB for PostgreSQL console.

## Prerequisites

- An Alibaba Cloud account is created. If you do not have an Alibaba Cloud account, visit the [Alibaba Cloud official website](#) to create an account.
- If this is the first time that you create an AnalyticDB for PostgreSQL instance, make sure that a service-linked role is created. For more information, see [Service linked role](#).
  - i. Log on to the [AnalyticDB for PostgreSQL console](#).
  - ii. In the upper-right corner, click **Create Instance**.
  - iii. In the **Create Service Linked Role** message, click **OK**.

## Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-right corner, click **Create Instance** to go to the buy page.
3. Configure the **Product Type** parameter. Valid values:
  - **Subscription** : You must pay a subscription fee when you create an instance. This billing method is more cost-effective than the pay-as-you-go billing method and is suitable for long-term use. Larger discounts are provided for longer subscription periods.
  - **Pay-as-you-go**: You are charged on an hourly basis for the resources that you use. The pay-as-you-go billing method is suitable for short-term use. If you no longer need a pay-as-you-go AnalyticDB for PostgreSQL instance, you can release the instance to reduce costs.

 **Note** You can change the billing method of an instance from pay-as-you-go to subscription. However, you cannot change the billing method from subscription to pay-as-you-go. For more information, see [Change the billing method from pay-as-you-go to subscription](#).

4. Configure the parameters that are described in the following table.

| Parameter | Description |
|-----------|-------------|
|-----------|-------------|

| Parameter              | Description   |
|------------------------|---|
| Region                 | <p>Select the region where you want to create the instance.</p> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p> <b>Note</b></p> <ul style="list-style-type: none"> <li>◦ After you complete the purchase, you cannot change the region of the instance. To accelerate access from users to the instance, we recommend that you select a region that is close to the geographical location of the users.</li> <li>◦ Make sure that the instance is deployed in the same region as the Elastic Compute Service (ECS) instance to which you want to connect. Otherwise, the AnalyticDB for PostgreSQL instance and the ECS instance can communicate only over the Internet. As a result, the performance of the instance may be compromised.</li> </ul> </div> |
| Zone                   | <p>A zone is an independent geographical location in a region. All zones in a region provide the same level of service performance.</p> <p>You can create the instance in the same zone as the ECS instance to which you want to connect or in a different zone.</p>  |
| Network Type           | <p>Select the network type of the instance. You can select only <b>VPC</b>.</p>   |
| VPC                    | <p>Select the ID of the virtual private cloud (VPC).</p> <p>A VPC is an isolated virtual network that provides higher security and higher performance than the classic network. Before you configure this parameter, make sure that a VPC and a vSwitch are created in the region where you want to create the instance. For more information, see <a href="#">Create and manage a VPC</a>.</p>   |
| vSwitch                | <p>Select a vSwitch in the specified VPC.</p>   |
| Instance Resource Type | <p>Select the resource type of the instance. Valid values:</p> <ul style="list-style-type: none"> <li>◦ <b>Serverless</b>: You can specify only the required computing resources without the need to reserve storage resources.</li> <li>◦ <b>Flexible Storage Mode</b>: You can separately expand disks and smoothly scale out instances online.</li> <li>◦ <b>Reserved Storage Mode</b>: You cannot separately expand disks or smoothly scale out instances.</li> </ul> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfe2f3;"> <p> <b>Note</b> The resource types that are available vary based on the region and zone. If the current region and zone do not provide the required instance resource type, change the region and zone.</p> </div>                  |

| Parameter   | Description   |
|---|---|
| <b>Edition</b>  | <p>Select an edition. Valid values: <b>High-availability</b> and <b>High Performance (Basic Edition)</b>.</p> <p> <b>Note</b> You can select <b>High Performance (Basic Edition)</b> only if you select <b>Reserved Storage Mode</b> for the Instance Resource Type parameter. For more information about Basic Edition, see <a href="#">Basic Edition</a>.</p>  |
| <b>Coordinator Nodes</b>                                | <p>Select the number of coordinator nodes. Default value: 1.</p> <p>The maximum number of coordinator nodes that you can select in the console is 2. If you require more coordinator nodes, <a href="#">Submit a ticket</a> to contact technical support.</p>   |
| <b>Compute Node Specifications</b><br><b>Node Cores</b> | <p>Select specifications for the compute nodes. The available specifications vary based on the <b>resource type</b> and <b>edition</b>. For more information, see <a href="#">Instance specifications</a>.</p> <p> <b>Note</b> During the public preview from January 01, 2022 to January 31, 2022, you can select only 4C16G for instances in Serverless mode.</p>  |
| <b>Nodes</b>  | <p>Select the number of compute nodes. The instance performance increases with the number of compute nodes. The number of compute nodes that you can select varies based on the <b>resource type</b> and <b>edition</b>:</p> <ul style="list-style-type: none"> <li>◦ <b>Serverless</b> <p><b>High-availability:</b> Valid values: 2 to 512. The number must be an integral multiple of two.</p> <p> <b>Note</b> During the public preview from January 01, 2022 to January 31, 2022, you can select up to 12 compute nodes for instances in Serverless mode.</p> </li> <li>◦ <b>Elastic Storage Mode</b> <ul style="list-style-type: none"> <li>▪ <b>High-availability:</b> Valid values: 4 to 512. The number must be an integral multiple of four.</li> <li>▪ <b>High Performance (Basic Edition):</b> Valid values: 2 to 512. The number must be an integral multiple of two.</li> </ul> </li> <li>◦ <b>Reserved Storage Mode</b> <p><b>High-availability:</b> Valid values: 2, 4, 8, 12, 16, 24, 32, 64, 96, and 128.</p> </li> </ul> |

| Parameter                         | Description   |
|-----------------------------------|---|
| Storage Disk Type<br>Storage Type | <p>Select a disk type. The available disk types vary based on the <b>resource type</b> and <b>edition</b>:</p> <ul style="list-style-type: none"> <li>◦ <b>Serverless</b>: You do not need to select a disk type.</li> <li>◦ <b>Elastic Storage Mode</b> <ul style="list-style-type: none"> <li>▪ <b>High-availability</b>: You can select <b>Enhanced SSD (ESSD)</b> or <b>Ultra Disk</b>.</li> <li>▪ <b>High Performance (Basic Edition)</b>: You can select only <b>Enhanced SSD (ESSD)</b>.</li> </ul> </li> <li>◦ <b>Reserved Storage Mode</b>: You can select only <b>SSD</b>.</li> </ul> |
| Encryption Type                   | <p>Select <b>Not Encrypted</b> or <b>Disk Encryption</b>.</p> <p> <b>Note</b> You can configure this parameter only if you select <b>Elastic Storage Mode</b>. Disk encryption is not supported for instances in <b>Serverless mode</b> or <b>reserved storage mode</b>.</p>   |
| CMK                               | <p>Select a disk encryption key.</p> <p>This parameter is required only if you select <b>Elastic Storage Mode</b> and <b>Disk Encryption</b>.</p>   |
| Single Node Storage Capacity      | <p>Select the exclusive storage capacity for each node in the instance. Valid values: 50 to 4,000. Unit: GB. The value must be an integral multiple of 50.</p> <p> <b>Note</b> You can configure this parameter only if you select <b>Elastic Storage Mode</b>.</p>  |
| Purchase Plan                     | <p>Select a subscription duration for the instance.</p> <p> <b>Note</b> This parameter is required only if you select <b>Subscription</b>.</p>   |

5. Click **Buy Now**.
6. On the **Confirm Order** page, perform the following operations based on the value of the **Product Type** parameter:
  - **Subscription**
    - a. Confirm information in the **Parameters** section, select **Terms of Service**, and then click **Pay**.
    - b. On the **Purchase** page, click **Subscribe**.
  - **Pay-as-you-go**

On the **Confirm Order** page, confirm the information in the **Parameters** section, select **Terms of Service**, and then click **Activate Now**.
7. After the instance is created, you can view the instance on the **Instances** page.

 **Note** It requires a specific amount of time to initialize an AnalyticDB for PostgreSQL instance. You can perform operations only when the instance is in the **Running** state.

## Related API operations

| API                                 | Description   |
|-------------------------------------|---|
| <a href="#">CreateDBInstance</a>    | Creates an AnalyticDB for PostgreSQL instance.<br>You can call this operation to create an AnalyticDB for PostgreSQL instance in Serverless mode, elastic storage mode, or reserved storage mode. |
| <a href="#">CreateECSDBInstance</a> | Creates an AnalyticDB for PostgreSQL instance in elastic storage mode.  |

## 2. Node configuration change

### 2.1. Change the specifications of compute nodes

When you use AnalyticDB for PostgreSQL, the data size and computing workload may surge and the data processing speed may hit a bottleneck due to insufficient computing resources such as CPU, memory, disk space, and data processing nodes. AnalyticDB for PostgreSQL allows you to change instance specifications and scale instances in a dynamic manner. This topic describes how to change the specifications of compute nodes.

#### Precautions

The specification changes that are supported vary based on the **resource type** of an AnalyticDB for PostgreSQL instance.

| Operation                     | Serverless    | Elastic storage mode | Reserved storage mode |
|-------------------------------|---------------|----------------------|-----------------------|
| Upgrade node specifications   | Not supported | Supported            | Supported             |
| Downgrade node specifications | Not supported | Supported            | Not supported         |
| Add compute nodes             | Supported     | Supported            | Supported             |
| Remove compute nodes          | Supported     | Not supported        | Not supported         |

#### Time required to change specifications

- Elastic storage mode and reserved storage mode

The amount of time required to upgrade specifications can range from 30 minutes to several hours. The amount of time varies based on factors such as the number of tables, the number of partitioned tables, the number of indexes, compression status, total data size, and instance specifications. During the upgrade, two service interruptions occur and the instances remain read-only to ensure data consistency. We recommend that you take precautions. After you complete the upgrade, the instance returns to the Running state and is updated to the latest minor version. You can connect to the instance.

- Serverless mode

You can scale AnalyticDB for PostgreSQL instances in Serverless mode within a few minutes without the need to migrate data. The scale speed varies based on the amount of time required to apply for resources and is not affected by the data size. The following scaling performance is provided for reference:

- Up to 60 seconds are required to scale an instance that has 16 or fewer nodes.
- Up to 5 minutes are required to scale an instance that has more than 16 nodes.

#### Upgrade node specifications

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the **Database Configuration Information** section of the **Basic Information** page, choose **Change Compute Node Specifications > Upgrade Node Specifications**.
5. On the **Upgrade/Downgrade** page, configure the parameters described in the following table.

| Parameter                           | Description   |
|-------------------------------------|---|
| <b>Instance Resource Type</b>       | The resource type of the current instance is displayed and cannot be modified.  |
| <b>Compute Node Specifications</b>  | Select node specifications based on your business requirements.   |
| <b>Single Node Storage Capacity</b> | Select a storage capacity per node based on your business requirements.<br><br><div style="background-color: #e1f5fe; padding: 5px; border: 1px solid #cfcfcf;"> <span style="font-size: 1.2em; color: #0070c0;">?</span> <b>Note</b> You cannot downgrade the storage capacity.         </div> |

6. Read and select the Terms of Service, and then click **Buy Now**.
7. Return to the **Instances** page and wait for the instance to change to the **Running** state.

## Downgrade node specifications

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the **Database Configuration Information** section of the **Basic Information** page, choose **Change Compute Node Specifications > Upgrade Node Specifications**.
5. On the **Upgrade/Downgrade** page, configure the parameters described in the following table.

| Parameter                          | Description  |
|------------------------------------|--|
| <b>Instance Resource Type</b>      | The resource type of the current instance is displayed and cannot be modified.   |
| <b>Compute Node Specifications</b> | Select node specifications based on your business requirements.<br><br><div style="background-color: #e1f5fe; padding: 5px; border: 1px solid #cfcfcf;"> <span style="font-size: 1.2em; color: #0070c0;">?</span> <b>Note</b> You can downgrade AnalyticDB for PostgreSQL instances in elastic storage mode. You cannot downgrade AnalyticDB for PostgreSQL instances in reserved storage mode.         </div> |

| Parameter                    | Description   |
|------------------------------|---|
| Single Node Storage Capacity | Select a storage capacity per node based on your business requirements. <div style="border: 1px solid #ccc; background-color: #e0f2f7; padding: 5px; margin-top: 10px;"> <span style="color: #0070c0;">?</span> <b>Note</b> You cannot downgrade the storage capacity.                 </div> |

6. Read and select the Terms of Service, and then click **Buy Now**.
7. Return to the **Instances** page and wait for the instance to change to the **Running** state.

## Add compute nodes

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the **Database Configuration Information** section of the **Basic Information** page, choose **Change Compute Node Specifications > Add Compute Node**.
5. In the **Info** dialog box, select **I am aware of the impact of this operation and agree to continue it.** and click **OK**.

🔔 **Notice** After you submit the specification change request, the SQL statement that is being executed will be interrupted and cannot be resumed even after the specifications are changed.

6. On the **Upgrade/Downgrade** page, select a value for the **Nodes** parameter based on your business requirements, read and select the Terms of Service, and then click **Buy Now**.
7. Return to the **Instances** page and wait for the instance to change to the **Running** state.
8. (Optional) You can execute the following SQL statement to prefetch data from tables that have high performance requirements to accelerate data access.

```
SELECT count(*) FROM <hot_table>;
```

? **Note**

- Only AnalyticDB for PostgreSQL instances in Serverless mode can prefetch data.
- During an upgrade or a downgrade, five steps are performed: the resources are initialized, the metadata information about system tables is synchronized, the resources are locked to modify the data distribution information, the resources are unlocked and cleared, and the local cache is asynchronously restored. The hit ratio of the local cache remains low for a short period of time because the local cache is asynchronously restored. You can prefetch data to accelerate data access.

## Remove compute nodes

1. Log on to the [AnalyticDB for PostgreSQL console](#).

2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the **Database Configuration Information** section of the **Basic Information** page, choose **Change Compute Node Specifications > Remove Compute Node**.
5. In the **Info** dialog box, select **I am aware of the impact of this operation and agree to continue it.** and click **OK**.

 **Notice** After you submit the specification change request, the SQL statement that is being executed will be interrupted and cannot be resumed even after the specifications are changed.

6. On the **Upgrade/Downgrade** page, select a value for the **Nodes** parameter based on your business requirements, read and select the Terms of Service, and then click **Buy Now**.
7. Return to the **Instances** page and wait for the instance to change to the **Running** state.
8. (Optional) You can execute the following SQL statement to prefetch data from tables that have high performance requirements to accelerate data access.

```
SELECT count(*) FROM <hot_table>;
```

 **Note**

- Only AnalyticDB for PostgreSQL instances in Serverless mode can prefetch data.
- During an upgrade or a downgrade, five steps are performed: the resources are initialized, the metadata information about system tables is synchronized, the resources are locked to modify the data distribution information, the resources are unlocked and cleared, and the local cache is asynchronously restored. The hit ratio of the local cache remains low for a short period of time because the local cache is asynchronously restored. You can prefetch data to accelerate data access.

## 3. Create a Basic Edition instance

This topic describes how to create an AnalyticDB for PostgreSQL Basic Edition instance.

### Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-right corner, click **Create Instance** to go to the AnalyticDB for PostgreSQL buy page.
3. Select a billing method and configure parameters.

To create a Basic Edition instance, you must specify the parameters described in the following table based on their descriptions. For information about the other parameters on the instance buy page, see [Create an AnalyticDB for PostgreSQL instance](#).

| Parameter                     | Description  |
|-------------------------------|--|
| <b>Region and Zone</b>        | Basic Edition is supported in the following regions and zones: <ul style="list-style-type: none"><li>◦ China:<ul style="list-style-type: none"><li>▪ China (Beijing): Zone I</li><li>▪ China (Hangzhou): Zone J</li><li>▪ China (Shanghai): Zone L</li><li>▪ China (Shenzhen): Zone F</li></ul></li><li>◦ Asia Pacific:<ul style="list-style-type: none"><li>Singapore: Zone C</li></ul></li></ul> |
| <b>Instance Resource Type</b> | Only <b>Elastic Storage Mode</b> is supported for Basic Edition.   |
| <b>Edition</b>                | Select <b>Basic</b> .  |

4. After you configure the parameters, click **Buy Now**.
5. On the **Confirm Order** page, select Terms of Service and click **Pay** or **Activate Now** to complete the payment.
6. Go back to the **Instances** page to view the created instance.

 **Note** It takes a while for AnalyticDB for PostgreSQL instances to initialize. After **Running State** of the instance changes to **Running**, you can perform subsequent operations on the instance.

### Related operations

| Operation                                   | Description  |
|---|--|
| <a href="#">CreateECSDBInstance</a>         | Creates an AnalyticDB for PostgreSQL instance in elastic storage mode. |
| <a href="#">DescribeDBInstanceAttribute</a> | Queries the details of an AnalyticDB for PostgreSQL instance.          |

---

| Operation  | Description   |
|--|---|
| <a href="#">DescribeDBInstanceOnECSAttribute</a> | Queries the details of an AnalyticDB for PostgreSQL instance in elastic storage mode. |
| <a href="#">DescribeDBInstances</a>              | Queries the AnalyticDB for PostgreSQL instances located within a specific region.     |

## 4. Create a database account

This topic describes how to create a database account for an AnalyticDB for PostgreSQL instance.

### Context

AnalyticDB for PostgreSQL provides the following types of database accounts:

- **Privileged account:** an account that has all permissions on all databases. The first account created for an instance in the console is a privileged instance.
- **Standard accounts:** accounts that have all permissions only on their authorized databases.

 **Note** Permissions include SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES, and TRIGGER. For information about how to configure permissions, see [Manage users and permissions](#).

### Precautions

- Before you use databases in an AnalyticDB for PostgreSQL instance, you must create a privileged account.
- Standard accounts cannot be created by using the console. For information about how to create standard accounts, see [Execute SQL statements to create accounts](#).
- After you create a privileged account for an instance, you cannot delete the privileged account.

### Create a privileged account

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, click **Account Management**.
5. Click **Create Account**.
6. In the **Create Account** panel, specify the following parameters.

| Parameter      | Description   |
|----------------|---|
| <b>Account</b> | <p>The name of the privileged account.</p> <ul style="list-style-type: none"><li>◦ The name can contain lowercase letters, digits, and underscores (_).</li><li>◦ The name must start with a lowercase letter and end with a lowercase letter or a digit.</li><li>◦ The name cannot start with gp.</li><li>◦ The name must be 2 to 16 characters in length.</li></ul> |

| Parameter               | Description   |
|-------------------------|---|
| <b>New Password</b>     | <p>The password of the privileged account.</p> <ul style="list-style-type: none"> <li>The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.</li> <li>The following special characters are supported:<br/>! @ # \$ % ^ &amp; * ( ) _ + - =</li> <li>The password must be 8 to 32 characters in length.</li> </ul> |
| <b>Confirm Password</b> | Enter the password of the account again.  |

7. Click **OK**.

 **Notice** After the account is created, you can click **Reset Password** in the **Actions** column to modify the account password. To ensure data security, we recommend that you change your password on a regular basis and do not use passwords that you have used before.

## Execute SQL statements to create accounts

- Create a privileged account.

```
CREATE ROLE <Account name> WITH LOGIN ENCRYPTED PASSWORD <Password> RDS_SUPERUSER;
```

Example:

```
CREATE role admin0 WITH LOGIN ENCRYPTED PASSWORD '111111' rds_superuser;
```

- Create a standard account.

```
CREATE ROLE <Account name> WITH LOGIN ENCRYPTED PASSWORD <'Password'>;
```

Example:

```
CREATE role test1 WITH LOGIN ENCRYPTED PASSWORD '111111';
```

## Related operations

| Operation                                | Description  |
|--|--|
| <a href="#">CreateAccount</a>            | Creates a privileged account.  |
| <a href="#">DescribeAccounts</a>         | Queries the accounts created on an AnalyticDB for PostgreSQL instance. |
| <a href="#">ModifyAccountDescription</a> | Modifies the name of an account for an instance.                       |
| <a href="#">ResetAccountPassword</a>     | Resets the password of an account for an instance.                     |

# 5. Configure an IP address whitelists

To ensure the security and stability of AnalyticDB for PostgreSQL databases, AnalyticDB for PostgreSQL instances block access from all IP addresses by default. Before you use an AnalyticDB for PostgreSQL instance, you must add IP addresses or CIDR blocks that are used to access the AnalyticDB for PostgreSQL instance to the whitelists of the instance. A properly configured IP address whitelists can make your AnalyticDB for PostgreSQL instance more secure. We recommend that you maintain IP address whitelists on a regular basis.

## Prerequisites

Before you configure a whitelists for an AnalyticDB for PostgreSQL instance, you must obtain the IP address of the client based on its installation location by using the following methods.

- ECS instances:
  - Linux instance  
Run the `ifconfig` command to view NIC information. You can view the IP addresses, subnet masks, gateways, DNS servers, and MAC addresses in the command output.
  - Windows instance  
In Command Prompt, run the `ipconfig /all` command to view NIC information. You can view the IP addresses, subnet masks, gateways, DNS servers, and MAC addresses in the command output.
- On-premises devices or third-party clouds:
  - Linux operating system: Run the `curl ipinfo.io |grep ip` command on the on-premises device to obtain its public IP address.
  - Windows operating system: Visit [ipinfo](#) on the on-premises device to obtain its public IP address.

## Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, click **Security Controls**.
5. On the **Security Controls** page, perform the following operations:
  - Create a whitelists.
    - a. Click **Create Whitelists**.

b. In the **Create Whitelist** dialog box, specify the following parameters.

| Parameter             | Description  |
|-----------------------|--|
| <b>Whitelist Name</b> | <p>The name of the new whitelist.</p> <ul style="list-style-type: none"> <li>▪ The name can contain lowercase letters, digits, and underscores (_).</li> <li>▪ The name must start with a lowercase letter and end with a lowercase letter or a digit.</li> <li>▪ The name must be 2 to 32 characters in length.</li> </ul>  |
| <b>IP Addresses</b>   | <p>The IP addresses or CIDR blocks that are allowed to access the instance.</p> <ul style="list-style-type: none"> <li>▪ Separate multiple IP addresses with commas (,). A maximum of 999 unique IP addresses can be specified.</li> <li>▪ Supported formats are specific IP addresses such as 10.23.12.24 and <b>CIDR blocks</b> such as 10.23.12.24/24. /24 indicates the length of the IP address prefix. An IP address prefix can be 1 to 32 bits in length.</li> <li>▪ If you set the prefix length to 0, for example, 0.0.0.0/0 or 127.0.0.1/0, all IP addresses are allowed to access the instance. This poses a high security risk. Proceed with caution.</li> <li>▪ The IP address 127.0.0.1 indicates that no external IP addresses are allowed to access the instance.</li> </ul> |

c. Click **OK**.

o **Modify a whitelist.**

a. Click **Modify** to the right of the whitelist name.

b. Add or remove IP addresses or CIDR blocks in the **IP Addresses** section.

 **Note** The **Whitelist Name** of the default whitelist cannot be modified.

c. Click **OK**.

o **Delete a whitelist.**

 **Note** The default whitelist cannot be deleted.

a. Click **Delete** to the right of the whitelist name.

b. In the **Delete Whitelist** message, click **OK**.

o **Clear the default whitelist.**

a. Click **Clear** to the right of the default whitelist.

b. In the **Clear Whitelist** message, click **OK**.

The default whitelist contains only 127.0.0.1 after it is cleared.

## Related operations

---

| Operation                                     | Description  |
|---|--|
| <a href="#">DescribeDBInstanceIPArrayList</a> | Queries the IP addresses that are allowed to access an instance. |
| <a href="#">ModifySecurityIps</a>             | Modifies an IP address whitelist of an instance.                 |

# 6. Use client tools to connect to an instance

This topic describes how to connect to an AnalyticDB for PostgreSQL instance by using tools that support the message-based protocol of PostgreSQL.

## Background information

AnalyticDB for PostgreSQL is fully compatible with the message-based protocol of PostgreSQL. You can connect to an AnalyticDB for PostgreSQL instance by using tools that support the protocol and GUI tools.

- Tools that support the message-based protocol of PostgreSQL include psql, libpq, Java Database Connectivity (JDBC), Open Database Connectivity (ODBC), and psycopg2.
- GUI tools include Alibaba Cloud Data Management (DMS) and DBeaver 6.0.0 or later.

 **Note** AnalyticDB for PostgreSQL V4.3 is built based on PostgreSQL 8.3. AnalyticDB for PostgreSQL V6.0 is built based on PostgreSQL 9.4.

## DMS

**DMS** allows you to manage relational, online transaction processing (OLTP), online analytical processing (OLAP), and NoSQL database services. Relational database services include ApsaraDB RDS for MySQL, ApsaraDB RDS for SQL Server, ApsaraDB RDS for PostgreSQL, and ApsaraDB RDS for PPAS. OLTP database services include Distributed Relational Database Service (DRDS). OLAP database services include AnalyticDB for PostgreSQL and Data Lake Analytics (DLA). NoSQL database services include ApsaraDB for MongoDB and ApsaraDB for Redis. DMS offers an integrated solution to manage data, schemas, and servers. You can also use DMS to authorize users, audit security, view BI charts and data trends, track data, and optimize performance.

This section describes how to use DMS to log on to an AnalyticDB for PostgreSQL instance.

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. (Optional) Create an AnalyticDB for PostgreSQL instance. For more information, see [Create an AnalyticDB for PostgreSQL instance](#).  
If an AnalyticDB for PostgreSQL instance has already been created, skip this step.
3. In the upper-left corner of the console, select the region where the instance resides.
4. Find the instance that you want to manage and click the instance ID.
5. (Optional) Create a database account. For more information, see [Create a database account](#).  
If an initial account has already been created, you can use this account to log on to the instance.
6. In the upper-right corner, click **Login Database**.
7. In the **Login instance** dialog box, specify the **Database Account** and **Database password** parameters, and click **Login**.

 **Note** AnalyticDB for PostgreSQL

For more information about DMS, see [Overview](#).

## psql

psql is a common command-line tool used together with Greenplum and provides a variety of commands. Its binary files are located in the bin directory of Greenplum.

Download client tools:

For Red Hat Enterprise Linux (RHEL) 6 or 7 and CentOS 6 or 7, you can download client tools from the following links and decompress the tool package:

- Download links of client tools for AnalyticDB for PostgreSQL V4.3:
  - RHEL 6 or CentOS 6: [ADBPG\\_client\\_package\\_el6](#)
  - RHEL 7 or CentOS 7: [ADBPG\\_client\\_package\\_el7](#)
- Download links of client tools for AnalyticDB for PostgreSQL V6.0:
  - RHEL 6 or CentOS 6: [ADBPG\\_client\\_package\\_el6](#)
  - RHEL 7 or CentOS 7: [ADBPG\\_client\\_package\\_el7](#)

After you download the client tool package, you must decompress the package and install the tools. This section describes how to install the client tools for AnalyticDB for PostgreSQL V6.0 on CentOS 7.

1. Run the following command to decompress the client tool package in its directory:

```
tar -xzvf adbpg_client_package.el7.x86_64.tar.gz
```

2. Switch to the *bin* directory and run the following command:

```
cd adbpg_client_package/bin
```

3. The *bin* directory includes client tools such as psql and pg\_dump. Run the corresponding commands based on the reference documentation of each tool.
  - For more information about how to use psql to connect to an AnalyticDB for PostgreSQL instance, see [psql](#).
  - pg\_dump is a logical backup tool for PostgreSQL. For more information about how to use pg\_dump, see [pg\\_dump](#).

To use the psql tool to connect to an instance, perform the following steps:

1. Use one of the following methods to connect to an instance:
  - Connection string

```
psql "host=yourgpdbaddress.gpdb.rds.aliyuncs.com port=5432 dbname=postgres user=gpdba  
ccount password=gpdbpassword"
```

- Specified parameters

```
psql -h yourgpdbaddress.gpdb.rds.aliyuncs.com -p 5432 -d postgres -U gpdbaccount
```

Parameter description:

- -h: the host address.
- -p: the port number used to connect to the instance.
- -d: the name of the database. The default value is postgres.
- -U: the account used to connect to the instance.

 **Note** You can run the `psql --help` command to view more options. You can also run the `\?` command to view the commands supported in psql.

2. Enter the password to go to the psql command interface.

References:

- For more information about how to use psql in Greenplum, see [Greenplum psql](#).
- AnalyticDB for PostgreSQL also supports psql for PostgreSQL. Take note of the differences of psql commands between Greenplum and PostgreSQL. For more information, see [PostgreSQL psql](#).

You can use the Docker toolbox for AnalyticDB for PostgreSQL, as shown in the following sample code. For more information about how to install Docker, visit the [Docker official website](#).

- Run the tool image corresponding to AnalyticDB for PostgreSQL V4.3.

```
docker run -idt --name=adbpgcli aliadbpg/adbpgcli:v4.3.0
docker exec -it adbpgcli /bin/bash -l
```

- Run the tool image corresponding to AnalyticDB for PostgreSQL V6.0.

```
docker run -idt --name=adbpgcli aliadbpg/adbpgcli:v6.3.0
docker exec -it adbpgcli /bin/bash -l
```

## DBeaver

DBeaver can run on major operating systems such as Windows, macOS, and Linux. This section describes how to use DBeaver on Windows to connect to an AnalyticDB for PostgreSQL instance.

1. Download and run the installation package of DBeaver Community Edition. For more information about the download link, visit the [DBeaver official website](#).
2. Start DBeaver and choose **Database > New Database Connection**.
3. In the **Connect to a database** dialog box, select the **Greenplum** database type and click **Next**.

 **Note** The first time you connect to an instance, you must download the corresponding database driver file.

4. On the **Main** tab of the **Connection Settings** page, set the following parameters.

| Parameter | Description | Example |
|-----------|-------------|---------|
|-----------|-------------|---------|

| Parameter | Description  | Example   |
|-----------|--|---|
| Host      | The endpoint and port number used to connect to the AnalyticDB for PostgreSQL instance. To obtain this information, perform the following steps:   | gp-bp1g*****-master.gpdbmaster.rds.aliyuncs.com |
| Port      | <ol style="list-style-type: none"> <li>i. Log on to the <a href="#">AnalyticDB for PostgreSQL console</a>.</li> <li>ii. In the upper-left corner of the console, select the region where the instance resides.</li> <li>iii. Find the instance that you want to manage and click the instance ID.</li> <li>iv. On the <b>Basic Information</b> page, view <b>Internal Endpoint</b> and <b>Internal Port</b> or <b>Public Address</b> and <b>Public Port</b> in the <b>Database Connection</b> section.</li> </ol>  | 5432  |
| Database  | The name of the database.  | postgres  |
| Username  | The database account used to connect to the AnalyticDB for PostgreSQL instance. For more information about how to create an account, see <a href="#">Create a database account</a> .   | testuser  |
| Password  | The password of the database account used to connect to the AnalyticDB for PostgreSQL instance.  | PassW0rd  |

5. Click **Test Connection**. A **Connected** message indicates that the connection is established.

 **Note** If an error message of `org.postgresql.Driver` is returned, you can perform the following operations to **download** or **update** the driver library of Greenplum:

- i. On the **Main** tab of the **Connection Settings** page, click **Edit Driver Settings**.
- ii. On the **Libraries** tab, click **Download/Update**.

6. Click **Finish**.

## JDBC

You can use the PostgreSQL or Greenplum JDBC driver to connect to AnalyticDB for PostgreSQL.

Use one of the following methods to download the JDBC driver:

- Click [PostgreSQL JDBC Driver](#) to download the PostgreSQL JDBC driver and add it to an environment variable.
- Obtain the Greenplum JDBC driver from the Greenplum official website. For more information, see [Greenplum Database 4.3 Connectivity Tools for UNIX](#) or [Greenplum Client and Loader Tools Package](#).

**Sample code:**

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class gp_conn {
    public static void main(String[] args) {
        try {
            Class.forName("org.postgresql.Driver");
            Connection db = DriverManager.getConnection("jdbc:postgresql://mygpdbpub.gpdb.r
ds.aliyuncs.com:5432/postgres","mygpdb","mygpdb");
            Statement st = db.createStatement();
            ResultSet rs = st.executeQuery("select * from gp_segment_configuration;");
            while (rs.next()) {
                System.out.print(rs.getString(1));
                System.out.print(" | ");
                System.out.print(rs.getString(2));
                System.out.print(" | ");
                System.out.print(rs.getString(3));
                System.out.print(" | ");
                System.out.print(rs.getString(4));
                System.out.print(" | ");
                System.out.print(rs.getString(5));
                System.out.print(" | ");
                System.out.print(rs.getString(6));
                System.out.print(" | ");
                System.out.print(rs.getString(7));
                System.out.print(" | ");
                System.out.print(rs.getString(8));
                System.out.print(" | ");
                System.out.print(rs.getString(9));
                System.out.print(" | ");
                System.out.print(rs.getString(10));
                System.out.print(" | ");
                System.out.println(rs.getString(11));
            }
            rs.close();
            st.close();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

For more information about the JDBC driver, see [The PostgreSQL JDBC Interface](#).

## Python

You can use `psycopg2` to connect to Greenplum or PostgreSQL in Python. To use `psycopg2`, perform the following steps:

1. Install psycopg2. If you use CentOS, you can use one of the following installation methods:

- Method 1: Run the `yum -y install python-psycopg2` command.
- Method 2: Run the `pip install psycopg2` command.
- Method 3: Use the source code.

```
yum install -y postgresql-devel*
wget http://initd.org/psycopg/tarballs/PSYCOPG-2-6/psycopg2-2.6.tar.gz
tar xf psycopg2-2.6.tar.gz
cd psycopg2-2.6
python setup.py build
sudo python setup.py install
```

2. Set the PYTHONPATH environment variable for subsequent variable reference. Sample code:

```
import psycopg2
sql = 'select * from gp_segment_configuration;'
conn = psycopg2.connect(database='gpdb', user='mygpdb', password='mygpdb', host='mygpdbpub.gpdb.rds.aliyuncs.com', port=5432)
conn.autocommit = True
cursor = conn.cursor()
cursor.execute(sql)
rows = cursor.fetchall()
for row in rows:
    print row
conn.commit()
conn.close()
```

Sample command output:

```
(1, -1, 'p', 'p', 's', 'u', 3022, '192.168.2.158', '192.168.2.158', None, None)
(6, -1, 'm', 'm', 's', 'u', 3019, '192.168.2.47', '192.168.2.47', None, None)
(2, 0, 'p', 'p', 's', 'u', 3025, '192.168.2.148', '192.168.2.148', 3525, None)
(4, 0, 'm', 'm', 's', 'u', 3024, '192.168.2.158', '192.168.2.158', 3524, None)
(3, 1, 'p', 'p', 's', 'u', 3023, '192.168.2.158', '192.168.2.158', 3523, None)
(5, 1, 'm', 'm', 's', 'u', 3026, '192.168.2.148', '192.168.2.148', 3526, None)
```

## libpq

libpq is the C language interface to PostgreSQL. You can use the libpq library to connect to and manage PostgreSQL databases in a C program. If Greenplum or PostgreSQL is deployed, you can find both the static and dynamic libraries of libpq in the *lib* directory.

- For more information about examples of libpq, see [Example Programs](#).
- For more information about libpq, see [libpq - C Library](#).

## ODBC

The PostgreSQL ODBC driver is an open source tool licensed based on the GNU Lesser General Public License (LGPL) protocol. You can download the driver from the [PostgreSQL official website](#).

To use the ODBC driver to connect to an AnalyticDB for PostgreSQL instance, perform the following steps:

1. Run the following commands to install the ODBC driver:

```
yum install -y unixODBC.x86_64
yum install -y postgresql-odbc.x86_64
```

## 2. Run the following command to configure the data source:

```
vim /etc/odbc.ini
```

Add connection information. Example:

```
[mygpdb]
Description = Test to gp
Driver = PostgreSQL
Database = ****
Servername = ****.gpdb.rds.aliyuncs.com
UserName = ****
Password = ****
Port = ****
ReadOnly = 0
```

 **Note** Replace `****` in the sample code with the corresponding connection information.

## 3. Test the connectivity. Example:

```
echo "select count(*) from pg_class" | isql mygpdb
+-----+
| Connected! |
| |
| sql-statement |
| help [tablename] |
| quit |
| |
+-----+
SQL> select count(*) from pg_class
+-----+
| count |
+-----+
| 388 |
+-----+
SQLRowCount returns 1
1 rows fetched
```

## 4. After the ODBC driver is connected to an instance, connect your application to the driver. For more information, see [PostgreSQL ODBC driver](#) and [psqlODBC HOWTO - C#](#).

## Other client tools

- GUI client tools

You can use other GUI client tools supported by Greenplum, such as [DBeaver](#), to connect to an AnalyticDB for PostgreSQL instance.

- Greenplum client tools

The Greenplum official website provides a tool package that includes JDBC, ODBC, and libpq. This package is easy to install and use. For more information, see the [Greenplum documentation](#).

## References

- [Pivotal Greenplum Database 4.3.33 Documentation](#)
- [Pivotal Greenplum 6.3 Documentation](#)
- [PostgreSQL ODBC driver](#)
- [Compiling psqlODBC on UNIX](#)
- [Greenplum ODBC Driver](#)
- [Greenplum JDBC Driver](#)

## 7.Enable SQL audit

AnalyticDB for PostgreSQL allows you to use the SQL audit feature to view SQL details and audit SQL queries on a regular basis. The SQL audit feature does not affect instance performance.

### Background information

SQL audit records all DML and DDL operations. The system captures such operations by analyzing data transmitted over network protocols. A few records may be lost when a large number of SQL queries are sent to database instances.

### Precautions

SQL audit logs are retained for 30 days.

### Pricing

For more information, see [AnalyticDB for PostgreSQL Pricing](#).

### Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click the instance ID.
4. In the left-side navigation pane, click **SQL Audit**.
5. On the **SQL Audit** tab, search SQL information by **Database Name**, **Execution Duration**, **Select Time Range**, **Database User**, **Source IP**, **Execution Status**, **Operation Category**, **Operation Type**, and **Text**.

 **Note** You can copy and export SQL information on the current page from SQL details.

### Related operations

| Operation                                  | Description  |
|--|--|
| <a href="#">DescribeSlowLogRecords</a>     | Queries the slow query details of a database in an instance within a specified period of time. |
| <a href="#">DescribeSlowSQLLogs</a>        | Queries the slow query logs of an instance within a specified period of time.                  |
| <a href="#">DescribeSQLCollectorPolicy</a> | Queries whether the SQL collection feature is enabled for an instance.                         |
| <a href="#">DescribeSQLLogByQueryId</a>    | Queries slow query logs by query ID.   |
| <a href="#">DescribeSQLLogFiles</a>        | Queries a list of SQL audit log files of an instance.  |
| <a href="#">DescribeSQLLogRecords</a>      | Queries the details of SQL audit logs of an instance within a specified period of time.        |

---

| Operation                                | Description   |
|--|---|
| <a href="#">ModifySQLCollectorPolicy</a> | Enables or disables the SQL collection feature for an instance. |

# 8.Instance locking

## 8.1. Instance locking caused by disk usage

This topic describes how a large amount of disk usage can cause instance locking for AnalyticDB for PostgreSQL instances in elastic storage mode.

### Introduction

When the disk usage of a node of an AnalyticDB for PostgreSQL instance exceeds a specific threshold, the whole instance is locked and becomes read-only.

To prevent business disruptions, you can upgrade instance specifications or add nodes before the disk usage reaches the threshold. For more information, see [Change the specifications of compute nodes](#).

### Check whether an instance is about to be locked

You can use the following methods to check the disk usage of an instance to determine whether the instance is about to be locked:

- View the disk usage of the instance:

You can view the **Instance Maximum Storage Water Level** value in the console. When the value exceeds 45%, the instance is locked. For information about how to view the **Instance Maximum Storage Water Level** value, see the [View the disk usage of an instance](#) section in this topic.

- View the disk usage of nodes:

You can view the disk usage of all nodes of the instance in the console. When the disk usage of a single node reaches 90%, the instance is locked. For information about how to view the disk usage of the nodes of an instance, see the [View the disk usage of nodes](#) section in this topic.

In the following example, an instance that has 50 GB of disk space in each compute node is used. The threshold that triggers instance locking is calculated based on the following formulas:

- High-availability Edition:  $(50 \text{ GB} \times 2) \times 90\% = 90 \text{ GB}$

When a compute node of the instance contains 90 GB of data, the instance is locked.

 **Note** High-availability Edition instances are deployed in a dual-replica architecture. Therefore, the disk usage value needs to be doubled when you calculate the threshold for instance locking.

- Basic Edition:  $(50 \text{ GB} \times 1) \times 90\% = 45 \text{ GB}$

When a compute node of the instance contains 45 GB of data, the instance is locked.

### View the disk usage of an instance

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the **Configuration Information** section, view the values to the right of **Storage Usage**.

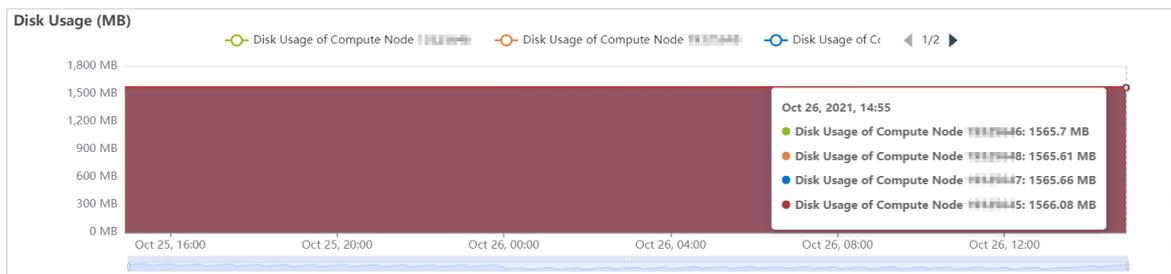
| Configuration Information |   | <a href="#">Add Master Node</a> <a href="#">Reduce Master Node</a> <a href="#">Compute Node Expansion</a> <a href="#">Computing Node Upgrade</a> |  |
|---------------------------|---|--|--|
| Database Type             | AnalyticDB for PostgreSQL 6.0   | System Version   | Standard Edition   |
| Multi-Slave Configuration | Not Enabled   | Storage Type   | ESSD Cloud Disk  |
| Node Specifications       | 2 Cores, 8 GB Memory, 50 GB Available Storage, 50 GB Total Physical Storage (Single-copy)       | Nodes  | 2  |
| Storage Watermark         | Instance Maximum Storage Water Level <b>0.6442%</b> , Instance Storage Watermark <b>1.2883%</b> | Total Instance Resources   | 4 Cores, 16 GB Memory, 100 GB ESSD Cloud Disk Total Physical Storage (Single-copy) |

- o **Instance Maximum Storage Water Level:** the disk usage of the node that contains the largest volume of data among all the nodes of the instance.
- o **Instance Storage Watermark:** the average disk usage of all the nodes of the instance.

## View the disk usage of nodes

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. Click the **Compute Node Monitoring** tab and view the **Disk Usage (MB)** section.

Move the pointer over the chart to view the disk usage of compute nodes at the specific point in time.



## Solution

If your AnalyticDB for PostgreSQL instance is locked due to overwhelming disk usage, you can use the following methods to unlock the instance:

- **Upgrade instance specifications (recommended)**  
You can upgrade the specifications of the instance to add more disk space.
- **Increase the number of nodes**  
You can add compute nodes to the instance. After this operation, the data of the instance is redistributed across all compute nodes, which may decrease the disk usage to a value lower than the threshold. If your instance is still locked after you add nodes, [Submit a ticket](#).

For information about how to upgrade instance specifications and increase the number of nodes, see [Change the specifications of compute nodes](#).

# 9.Restart an AnalyticDB for PostgreSQL instance

This topic describes how to restart an AnalyticDB for PostgreSQL instance in the AnalyticDB for PostgreSQL console.

## Notice

An AnalyticDB for PostgreSQL instance is unavailable while it is being restarted. We recommend that you restart your AnalyticDB for PostgreSQL instance during off-peak hours.

## Procedure

To restart an AnalyticDB for PostgreSQL instance, follow these steps:

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the top navigation bar, select the region where the target instance resides. For example, select the China (Hangzhou) region.
3. Find the target instance and click its ID. The **Basic Information** page appears.
4. In the upper-right corner, click **Restart Instance**. In the message that appears, click **OK**. If your account is bound to your mobile phone number, you must provide a verification code.

## Notice

In most cases, the restart process takes 3 minutes to 30 minutes. The instance is unavailable while it is being restarted. We recommend that you make arrangements for your business before you restart the instance. After the instance is restarted, its status becomes **Running**. This means it is available again.

After you complete the preceding steps, check the status of the instance. If the instance is restarted, its status shows **Running**. Otherwise, its status shows **Restarting**.

## Related operations

| Operation                         | Description                                     |
|-----------------------------------|---|
| <a href="#">RestartDBInstance</a> | Restarts an AnalyticDB for PostgreSQL instance. |

# 10. Configure disk encryption for an AnalyticDB for PostgreSQL instance

This topic describes how to configure disk encryption for an AnalyticDB for PostgreSQL instance in elastic storage mode. The disk encryption feature encrypts the data on each data disk of your AnalyticDB for PostgreSQL instance by using Elastic Block Storage (EBS). This way, your data cannot be cracked even if your data is leaked.

## Introduction

After an encrypted disk is created and attached to an ECS instance, the ECS instance encrypts the following data:

- Static data on disks.
- Data transmitted between disks and instances. Data is not encrypted again within the instance operating system.
- All snapshots created from encrypted disks. Such snapshots are encrypted snapshots.

## Precautions

- You can enable disk encryption only when your AnalyticDB for PostgreSQL instance is being created. After your instance is created, you cannot enable disk encryption.
- You must select Enhanced SSD or Ultra Disk storage type for your instance when you create the instance.
- Disk encryption cannot be disabled after it is enabled.
- After you enable disk encryption for your instance, both the snapshots generated by the instance and the new instances created from those snapshots are automatically encrypted.
- Disk encryption does not interrupt your business and you do not need to modify your application.

## Billing

The disk encryption feature is free of charge for AnalyticDB for PostgreSQL instances. You do not need to pay additional fees for the read/write operations you perform on encrypted disks.

For information about key hosting fees and API operation call fees, see [Billing](#).

## Procedure

When you create an instance, perform the following operations. For more information, see [Create an instance](#).

1. Set **Instance Resource Type** to Elastic Storage Mode.
2. Set **Storage Type** to Enhanced SSD or Ultra Disk.
3. Set **Encryption Type** to Disk Encryption.
4. Select a key that is used for encryption. If no key is created, you must activate Key Management Service (KMS) and create a key.

 **Note**

- Disk encryption supports only keys that are manually created. When you create keys, you must select **Disable** from the **Rotation Period** drop-down list. For more information, see [Create a CMK](#).
- If you authorize the instance to access KMS, ActionTrail records the action. For more information, see [Use ActionTrail to query KMS event logs](#).

5. Click **Buy Now** to configure disk encryption for the instance.

# 11. Network Connection

## 11.1. Manage public endpoints

If your application is deployed on an ECS instance that is located within the same region and has the same **network type** as your AnalyticDB for PostgreSQL instance, you do not need to apply for a public endpoint. If your application is deployed on a third-party system or on an ECS instance that is located within a different region or has a different network type from your AnalyticDB for PostgreSQL instance, you must apply for a public endpoint.

 **Note** If your application is deployed on an ECS instance that is located within the same region (but maybe in different zones) and has the same network type as your AnalyticDB for PostgreSQL instance, the application can connect to the AnalyticDB for PostgreSQL instance over the internal network.

### Scenarios

Internal and public endpoints apply to the following scenarios:

- Scenarios where only an internal endpoint is required:
  - Your application is deployed on an ECS instance that is located within the same region and has the same **network type** as your AnalyticDB for PostgreSQL instance.
- Scenarios where only a public endpoint is required:
  - Your application is deployed on an ECS instance that is located within a region different from your AnalyticDB for PostgreSQL instance.
  - Your application is deployed on a third-party system.
- Scenarios where both an internal endpoint and a public endpoint are required:
  - Some modules of your application are deployed on an ECS instance that is located within the same region and have the same **network type** as your AnalyticDB for PostgreSQL instance, but other modules are deployed on an ECS instance that is located within a region different from your AnalyticDB for PostgreSQL instance.
  - Some modules of your application are deployed on an ECS instance that is located within the same region and have the same **network type** as your AnalyticDB for PostgreSQL instance, but other modules are deployed on a third-party system.

### Precautions

- Before you connect your application to your AnalyticDB for PostgreSQL instance, you must add the IP address or the corresponding CIDR block of the application to a whitelist of the instance. For more information, see [Configure an IP address whitelist](#).
- If you connect to the instance by using the public endpoint, security is compromised. Proceed with caution. To increase security and accelerate data transfer, we recommend that you migrate your application to an ECS instance that is located within the same region as your AnalyticDB for PostgreSQL instance.

### Apply for a public endpoint

1. Log on to the [AnalyticDB for PostgreSQL console](#).

2. In the upper-left corner of the page, select the **region** where the desired instance resides from the drop-down list.
3. Find the instance for which you want to apply for a public endpoint, and click its ID.
4. On the **Basic Information** page, click **Apply for Public Endpoint**. Alternatively, in the left-side navigation pane, click **Database Connection**.
5. On the **Database Connection** page, click **Apply for Public Endpoint**.
6. A public endpoint is generated.

On the **Database Connection** page, you can click **Release Public Endpoint** to release the public endpoint.

## Release a public endpoint

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the page, select the **region** where the desired instance resides from the drop-down list.
3. Find the instance for which you want to release the public endpoint and click the instance ID.
4. In the left-side navigation pane, click **Database Connection**.
5. On the **Database Connection** page, click **Release Public Endpoint**.  
If you have not applied for a public endpoint, only the **Apply for Public Endpoint** option is displayed on the **Database Connection** page.
6. In the message that appears, click **OK** to release the public endpoint.

## Related API operations

| API  | Description  |
|--|--|
| <a href="#">AllocateInstancePublicConnection</a> | Applies for a public endpoint for an AnalyticDB for PostgreSQL instance. |
| <a href="#">ReleaseInstancePublicConnection</a>  | Releases the public endpoint of an AnalyticDB for PostgreSQL instance.   |

# 11.2. Release an Internet IP address

If the network environment changed after the Internet address is allocated, you can release the Internet address on AnalyticDB for PostgreSQL console if you don't need it any more. After releasing the Internet address, make sure to change the application configurations which related to this address.

Before performing this operation, please read the following scenarios.

## Scenarios

Internet IP addresses and intranet IP addresses are used in the following scenarios:

- Use an intranet IP addresses only:
  - Your application is deployed on an ECS instance in the same region as your AnalyticDB for PostgreSQL instance and shares the same **network type** with the ECS instance.
- Use an Internet IP addresses only:

- The ECS instance where your application is deployed and your AnalyticDB for PostgreSQL instance are in different regions.
- Your application is deployed in a third-party system other than Alibaba Cloud.
- Use both Internet and intranet IP addresses:
  - Some application modules are deployed on an ECS instance in the same region with the same **network type**, while other modules are deployed on an ECS instance in a different region.
  - Some modules of the application are deployed on an ECS instance in the same region with the same **network type**, while other modules are deployed in systems other than Alibaba Cloud.

## Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. Select the **Region** of the instance.
3. Click the ID of the instance to go to its **Basic Information** page.
4. Click **Database Connection** on the left-side navigation.
5. On the **Database Connection** page, Click **Release Internet Address**.

If you haven't applied for an Internet address since you created an instance, there is only **Apply for Internet address** on the **Database Connection** page.

6. Click **OK** on the dialog box to release the Internet address.

## Related API

| API   | Description                                  |
|---|--|
| <a href="#">ReleaseInstancePublicConnection</a> | Release the Internet address of an instance. |

# 11.3. Endpoints of an instance and its primary coordinator node

Starting from February 8, 2021, AnalyticDB for PostgreSQL allows you to configure multiple coordinator nodes.

You can add multiple coordinator nodes to an instance to push beyond the limits of its original single-coordinator node architecture. If the compute nodes permit, the number of connections and the I/O capabilities linearly increase with the number of coordinator nodes, and overall system performance improves. However, if you use multiple coordinator nodes within an instance, the endpoints of the instance have limits. For your convenience, AnalyticDB for PostgreSQL provides endpoints for both the primary coordinator node and the instance. You can choose an endpoint to use based on your compatibility and performance considerations. If you have questions, join the DingTalk group for technical support or [submit a ticket](#).

If you set more than one coordinator node when you create an AnalyticDB for PostgreSQL instance, separate endpoints are created for the primary coordinator node and the instance.

**Endpoints of the primary coordinator node:** Requests initiated from this type of endpoints are forwarded to the primary coordinator node. Secondary coordinator nodes do not handle requests. If you choose to use this type of endpoints, the capabilities of the system are fully compatible with an AnalyticDB for PostgreSQL instance that has a single coordinator node.

**Endpoints of the instance:** Requests initiated from this type of endpoints are forwarded to the primary coordinator node and secondary coordinator nodes because Server Load Balancer (SLB) is automatically connected.

# 12. Version Management

## 12.1. Upgrade the engine version

To better meet your requirements, AnalyticDB for PostgreSQL has its minor engine version updated on a regular basis. By default, the latest minor version is used when you create an AnalyticDB for PostgreSQL instance. After a new minor version is released, you can update your instance to use the updated features. This topic describes how to update the minor version of an instance.

### Precautions

- The console allows you to update the minor version of an instance, but you cannot upgrade the major version from 4.3 to 6.0. For more information about how to upgrade the major version from 4.3 to 6.0, see [Compatibility comparisons between AnalyticDB for PostgreSQL V4.3 and V6.0](#) and [Check for incompatibility between AnalyticDB for PostgreSQL V4.3 and V6.0](#).
- When you update the minor version of an instance, the instance restarts and is unavailable during the restart. We recommend that you update the minor version during off-peak hours.

### Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. On the **Basic Information** page, click **Upgrade Minor Version** in the upper-right corner.
5. In the **Upgrade Minor Version** message, click **OK**.

If your account is bound to your mobile phone number, you must provide a verification code.

 **Note** Minor version updates take 3 to 15 minutes to complete. When an instance is being updated, it is unavailable. We recommend that you prepare in advance before you update the instance. After the update is complete, the instance enters the Running state, and you can access the instance databases.

6. After you complete the preceding steps, check the state of the instance. If the update is complete, the instance is in the **Running** state. If the instance is still being updated, it remains in the **Upgrading version** state.

Before the update, the system checks the minor version of your instance. If the latest minor version is used, the system skips the update and restart operations.

### Related operations

| Operation                        | Description                               |
|----------------------------------|---|
| <a href="#">UpgradeDBVersion</a> | Updates the minor version of an instance. |

## 12.2. Query the minor engine version

This topic describes how to query the minor engine version of an AnalyticDB for PostgreSQL instance.

## Prerequisites

You are connected to an AnalyticDB for PostgreSQL instance. For more information about how to connect to an instance, see [Use client tools to connect to an instance](#).

## Query the minor version number

Execute the following statement to query the minor version number of an AnalyticDB for PostgreSQL instance:

```
show adbpq_version;
```

Sample query result:

```
adbpq_version
-----
6.3.4.0
(1 row)
```

## Query the minor version release date

Execute the following statement to query the minor version release date of an AnalyticDB for PostgreSQL instance:

```
show rds_release_date;
```

Sample query result:

```
rds_release_date
-----
20210713
(1 row)
```

## References

- For information about how to update the minor version, see [Upgrade the engine version](#).
- For information about updated features of minor versions, see [Release notes](#).

# 12.3. Compatibility comparisons between AnalyticDB for PostgreSQL V4.3 and V6.0

This topic describes the compatibility comparisons between AnalyticDB for PostgreSQL V4.3 and V6.0. If you upgrade your AnalyticDB for PostgreSQL instances from V4.3 to V6.0, you must make necessary adjustments.

## Query optimizers

| Item                    | V4.3   | V6.0 |
|-------------------------|--------|------|
| Default query optimizer | Legacy | ORCA |

| Item | V4.3 | V6.0 |
|------|------|------|
|------|------|------|

Both AnalyticDB for PostgreSQL V4.3 and V6.0 support Legacy and ORCA query optimizers. For more information about these query optimizers, see [Choose a query optimizer](#).

## Escape characters

- In AnalyticDB for PostgreSQL V6.0, `backslashes (\)` within strings are not used as escape characters.
- You can execute the following statement to use backslashes (\) as escape characters. However, we recommend that you do not perform this operation.

```
set standard_conforming_strings = off;
```

 **Note** The preceding statement is available only for sessions. If you want to use this statement for AnalyticDB for PostgreSQL instances, [submit a ticket](#) and request technical support of AnalyticDB for PostgreSQL to configure the escape characters based on your requirements.

## Data type conversion

- AnalyticDB for PostgreSQL V6.0 does not automatically convert a string of the `YYYYMMDDHH24MISS` format to a timestamp. To perform such conversion, you can use the `to_timestamp/to_char` built-in function.
- Compared with V4.3, AnalyticDB for PostgreSQL V6.0 does not implicitly convert numeric data types to TEXT. After you upgrade AnalyticDB for PostgreSQL from V4.3 to V6.0, you must add functions to the original SQL statements to convert numeric data types to TEXT. Examples:

```
create or replace function substr(numeric, integer, integer) returns text as $$
select substr($1::text, $2, $3);
$$ language sql IMMUTABLE strict;
```

```
create or replace function pg_catalog.btrim(str numeric) returns text as $$
return $_[0];
$$ language plperl IMMUTABLE strict;
```

```
create or replace function to_date(timestamp, text) returns date as $$
select to_date($1::text, $2);
$$ language sql IMMUTABLE strict;
```

```
create or replace function to_date(integer, text) returns date as $$
select to_date($1::text, $2);
$$ language sql IMMUTABLE strict;
```

- You must manually rewrite the SQL statements or functions in which the data type needs to be implicitly converted to TEXT.

## Error logs of external tables

In AnalyticDB for PostgreSQL V6.0, you cannot use the `INTO error_table` clause in the `CREATE EXTERNAL TABLE` or `COPY` statement. Instead, you can use a built-in function to manipulate error logs of external tables.

```
gp_read_error_log('$external_table')
gp_truncate_error_log('$external_table')
```

## Data types

- If you change the storage format of NUMERIC files, the corresponding disk space is affected.
- If you change the MONEY data type from 32-bit to 64-bit, the corresponding disk space is affected.
- In AnalyticDB for PostgreSQL V6.0, you cannot use the following data types for distribution keys:
  - abstime
  - reltime
  - tinterval
  - money
  - anyarray

## Keywords

AnalyticDB for PostgreSQL V6.0 adds, modifies, and deletes some keywords. Names of database objects cannot be the same as keywords.

The use of keywords varies between different categories. You can execute the following statement to view all keywords and their categories in both AnalyticDB for PostgreSQL V4.3 and V6.0:

```
select * from pg_get_keywords();
```

The following table describes keyword categories.

| Category Code | Category                                     | Description  |
|---------------|--|--|
| U             | unreserved                                   | Unreserved. The keywords of this category can be used as names for all objects including views, tables, functions, indexes, fields, and types. |
| C             | unreserved (cannot be function or type name) | Unreserved. The keywords of this category can be used as names of objects except for functions and types.                                      |
| T             | reserved (can be function or type name)      | Reserved. The keywords of this category cannot be used as names of objects except for functions and types.                                     |
| R             | reserved                                     | Reserved. The keywords of this category cannot be used as names of objects.  |

The following table lists the keywords that fall into different categories between AnalyticDB for PostgreSQL V4.3 and V6.0.

| Keyword                | V4.3   | V6.0   |
|------------------------|--|--|
| <b>between</b>         | reserved                                     | unreserved (cannot be function or type name) |
| <b>collation</b>       | None   | reserved (can be function or type name)      |
| <b>concurrently</b>    | unreserved                                   | reserved (can be function or type name)      |
| <b>convert</b>         | unreserved (cannot be function or type name) | None   |
| <b>filter</b>          | reserved                                     | unreserved                                   |
| <b>lateral</b>         | N/A  | reserved                                     |
| <b>new</b>             | reserved                                     | None   |
| <b>off</b>             | reserved                                     | unreserved                                   |
| <b>old</b>             | reserved                                     | None   |
| <b>percentile_cont</b> | unreserved (cannot be function or type name) | None   |
| <b>percentile_disc</b> | unreserved (cannot be function or type name) | None   |
| <b>range</b>           | reserved                                     | unreserved                                   |
| <b>reindex</b>         | unreserved                                   | reserved                                     |
| <b>rows</b>            | reserved                                     | unreserved                                   |
| <b>sort</b>            | reserved                                     | reserved (can be function or type name)      |
| <b>variadic</b>        | None   | reserved                                     |

## System tables

Some system tables differentiate between AnalyticDB for PostgreSQL V4.3 and V6.0. If your business logic references the following system tables, you must modify the referenced system tables to avoid errors.

| V4.3                  | V6.0 | Description   |
|-----------------------|------|---|
| pg_class.reltoastixid | None | AnalyticDB for PostgreSQL V6.0 does not support this table. |

| V4.3  | V6.0   | Description   |
|---|--|---|
| pg_stat_activity.procpid  | pg_stat_activity.pid                             | In AnalyticDB for PostgreSQL V6.0, the column name is changed from procpid to pid.  |
| pg_stat_activity.current_query  | pg_stat_activity.state<br>pg_stat_activity.query | AnalyticDB for PostgreSQL V6.0 displays the current activity of a server process in two columns. The state column displays the current overall state of the backend. The query column displays the currently executing query. |
| gp_distribution_policy.attnums  | gp_distribution_policy.distkey                   | In AnalyticDB for PostgreSQL V6.0, the column name is changed from attnums to distkey, and the data type of this column is changed to int2vector.   |
| sesion_level_memory_consumption.__gp_localid<br>sesion_level_memory_consumption.__gp_masterid | None   | AnalyticDB for PostgreSQL V6.0 does not support these tables.   |
| pg_filespace<br>pg_filespace_entry  | None   | AnalyticDB for PostgreSQL V6.0 does not support these tables.   |

## Parameters of built-in functions

AnalyticDB for PostgreSQL V6.0 changes the parameters of some built-in functions.

| V4.3                           | V6.0                              | Description  |
|--------------------------------|-----------------------------------|--|
| int4_avg_accum(bytea, integer) | int4_avg_accum(bigint[], integer) | None   |
| string_agg(expression)         | string_agg(expression, delimiter) | In AnalyticDB for PostgreSQL V6.0, you can use the string_agg function to convert an expression to a string. |

## Other comparisons

| Item  | V4.3          | V6.0      |
|---|---------------|-----------|
| LEFT() function                               | Not supported | Supported |
| UPDATE operations on distribution key columns | Not supported | Supported |

## 12.4. Check for incompatibility between AnalyticDB for PostgreSQL V4.3 and V6.0

AnalyticDB for PostgreSQL V6.0 has configurations that are incompatible with V4.3. To upgrade your AnalyticDB for PostgreSQL instances from V4.3 to V6.0, you must preprocess the incompatible configurations. This topic describes how to run a shell script to check for incompatibility between AnalyticDB for PostgreSQL V4.3 and V6.0. In this example, a Linux operating system is used. If you use Windows, you can refer to the SQL statements in the script.

### Precautions

The check items do not include SQL statements for your business or custom stored procedures, functions, or views. You can verify the excluded items in AnalyticDB for PostgreSQL V6.0 instances.

### Sample shell script

```
#!/bin/bash
#
# Copyright (c) 2020, Alibaba Group, Inc.
#
# Description:
check unsupported items before upgrade instance.
# Usage:  sh 4x_to_6x_check.sh <PGHOST> <PGPORT> <PGUSER> <PGPASSWORD>
# CheckList:
#       1) Check the version of the AnalyticDB for PostgreSQL instance.
#       2) Check the libraries that are not transferred.
#       3) Check the unsupported distribution keys for tables.
#       4) Check the unsupported field types for tables.
#       5) Check the unsupported extensions.
#       6) Check the unsupported stored procedures and functions.
#       7) Check the unsupported views.
# Notice:      If one or more error messages are returned, you must modify the correspond
ing configurations.
#
if [[ $# -lt 4 ]]; then
    echo "Usage:
$0 <PGHOST> <PGPORT> <PGUSER> <PGPASSWORD>"
    exit 1
fi
export PGHOST=$1
export PGPORT=$2
export PGUSER=$3
export PGPASSWORD=$4
db_ver=`psql -d postgres -c "copy (select version()) to stdout"`
db_names=`psql -d postgres -c "copy (select sodddatname from gp_toolkit.gp_size_of_database
) to stdout"`
db_names=${db_names}
db_len=${#db_names[@]}
unsupport6x_ext="('feature_extractor','varbitx')"
unsupport6x_disted_type="('money','tinterval')"
unsupport6x_type="('unknown')"
# Check the version of the AnalyticDB for PostgreSQL instance.
```

```

check_version()
{
    echo ''
    echo $db_ver
    echo ''
    echo '***** check base version...'
    base_time=`date -d "2020-08-31" +%s`
    db_verdate=${db_ver##*compiled on}
    seconds=`date -d "$db_verdate" +%s`
    if [[ $seconds -lt $base_time ]]; then
        echo 'ERROR: please upgrade minor version...'
    else
        echo 'pass.....'
    fi
}
# Check the libraries that are not transferred.
check_libraries()
{
    echo ''
    echo '***** check untransferred libraries...'
    count=`psql -d postgres -c "copy (select count(1) from pg_catalog.pg_library) to stdout"`
    if [[ $count -gt 0 ]]; then
        psql -d postgres -c "select name,laname language from pg_catalog.pg_library;"
        echo "WARN: please transfer libraries manually..."
    else
        echo 'pass.....'
    fi
}
# Check the unsupported distribution keys for tables.
check_table_did()
{
    echo ''
    echo '***** check unsupported table distributedId types...'
    count=0
    if [[ $db_ver == *8.2*4.3* ]]; then
        for ((i=0; i<$db_len; ++i)); do
            sql="select count(1) from pg_catalog.pg_class c,pg_catalog.pg_attribute a,pg_catalog.
pg_type t,pg_catalog.gp_distribution_policy p where
a.atttypid=t.oid and a.attrelid=c.oid and p.localoid=c.oid and a.attnum=any(p.attrnum
s) and a.attnum>0 and t.typname in $unsupport6x_disted_type"
            count1=`psql -d ${db_names[$i]} -c "copy ($sql) to stdout"`
            count=$((count + count1))
            if [[ $count1 -gt 0 ]]; then
                sql="select '${db_names[$i]}' dbname,n.nspname schema,c.relname table_name,a.attnam
e distributed_field,t.typname field_type from
pg_catalog.pg_namespace n,pg_catalog.pg_class c,pg_catalog.pg_attribute a,pg_catalo
g.pg_type t,pg_catalog.gp_distribution_policy p
where a.atttypid=t.oid and n.oid=c.relnamespace and a.attrelid=c.oid and p.localoid
=c.oid and a.attnum=any(p.attrnums) and a.attnum>0 and t.typname in $unsupport6x_disted_typ
e order by schema,table_name;"
                psql -d ${db_names[$i]} -c "$sql"
            fi
        done
    fi
}

```

```

if [[ $count -gt 0 ]]; then
    echo 'ERROR: please alter table distributedId types manually...'
else
    echo 'pass.....' fi
}
# Check the unsupported field types for tables.
check_table_fstype()
{
    echo ''
    echo '***** check unsupported table field types...'
    count=0
    if [[ $db_ver == *8.2*4.3* ]]; then
        for ((i=0; i<$db_len; ++i)); do
            sql="select count(1) from pg_catalog.pg_class c,pg_catalog.pg_attribute a,pg_catalog.
pg_type t where
            a.atttypid=t.oid and a.attrelid=c.oid and a.attnum>0 and t.typname in $unsupport6x_ty
pe"
            count1=`psql -d ${db_names[$i]} -c "copy ($sql) to stdout"`
            count=$((count + count1))
            if [[ $count1 -gt 0 ]]; then
                sql="select '${db_names[$i]}' dbname,n.nspname schema,c.relname table_name,a.attnam
e field_name,t.typname field_type from
                pg_catalog.pg_namespace n,pg_catalog.pg_class c,pg_catalog.pg_attribute a,pg_catalo
g.pg_type t
                where a.atttypid=t.oid and n.oid=c.relnamespace and a.attrelid=c.oid and a.attnum>0
and t.typname in $unsupport6x_type order by schema,table_name;"
                psql -d ${db_names[$i]} -c "$sql"
            fi
        done
    fi
    if [[ $count -gt 0 ]]; then
        echo 'ERROR: please alter table field types manually...'
    else
        echo 'pass.....'
        fi
}
# Check the unsupported extensions.
check_extensions()
{
    echo ''
    echo '***** check unsupported extensions...' count=0
    if [[ $db_ver == *8.2*4.3* ]]; then
        for ((i=0; i<$db_len; ++i)); do
            count1=`psql -d ${db_names[$i]} -c "copy (select count(1) from pg_catalog.pg_extensio
n where extname in $unsupport6x_ext) to stdout"`
            count=$((count + count1))
            if [[ $count1 -gt 0 ]]; then
                psql -d ${db_names[$i]} -c "select '${db_names[$i]}' dbname,extname,extversion from
pg_catalog.pg_extension where extname in $unsupport6x_ext;"
            fi
        done
    fi
    if [[ $count -gt 0 ]]; then
        echo 'WARN: please drop useless extensions manually...'
        echo 'REF DROP EXTENSION SQL: drop extension <name>'
    fi
}

```

```

echo REF DROP EXTENSION SQL: drop extension $name/
else
    echo 'pass.....'
fi
}
# Check the unsupported stored procedures and functions.
check_procs()
{
    echo ''
    echo '***** check unsupported procs...'
    count=0
    clause="lower(p.prosrc) like '%pg_stat_activity%'
    and ( lower(p.prosrc) like '%.procpid%' or lower(p.prosrc) like '%.current_query%' or low
er(p.prosrc) like '%.waiting%' )
    and n.nspname not in ('gp_toolkit','information_schema')"
    if [[ $db_ver == *8.2*4.3* ]]; then
        for ((i=0; i<$db_len; ++i)); do
            count1=`psql -d ${db_names[$i]} -c "copy (select count(1) from pg_catalog.pg_proc p j
oin pg_catalog.pg_namespace n on p.pronamespace = n.oid where $clause) to stdout"`
            count=$((count + count1))
            if [[ $count1 -gt 0 ]]; then
                psql -d ${db_names[$i]} -c "select '${db_names[$i]}' dbname,n.nspname schemaname,p.
priname from pg_catalog.pg_proc p join pg_catalog.pg_namespace n on p.pronamespace = n.oid
where $clause;"
            fi
        done
    fi
    if [[ $count -gt 0 ]]; then
        echo 'WARN: please drop/repair proc/function manually after transferred...'
    else
        echo 'pass.....'
    fi
}
# Check the unsupported views.
check_views()
{
    echo ''
    echo '***** check unsupported views...'
    count=0
    clause="lower(definition) like '%pg_stat_activity%'
    and ( lower(definition) like '%.procpid%' or lower(definition) like '%.current_query%' or
lower(definition) like '%.waiting%' )
    and schemaname not in ('gp_toolkit','information_schema')"
    if [[ $db_ver == *8.2*4.3* ]]; then
        for ((i=0; i<$db_len; ++i)); do
            count1=`psql -d ${db_names[$i]} -c "copy (select count(1) from pg_catalog.pg_views wh
ere $clause) to stdout"`
            count=$((count + count1))
            if [[ $count1 -gt 0 ]]; then
                psql -d ${db_names[$i]} -c "select '${db_names[$i]}' schemaname,viewname from pg_ca
talog.pg_views where $clause;"
            fi
        done
    fi
    if [[ $count -gt 0 ]]; then

```

```

    echo 'WARN: please drop useless views manually...'
else
    echo 'pass.....'
fi
}
check_version
check_libraries
check_table_did
check_table_ftype
check_extensions
check_procs
check_views

```

### Parameters

| Parameter    | Description   |
|--------------|---|
| <PGHOST>     | The endpoint that is used to connect to the AnalyticDB for PostgreSQL V4.3 instance.    |
| <PGPORT>     | The port number that is used to connect to the AnalyticDB for PostgreSQL V4.3 instance. |
| <PGUSER>     | The username that is used to connect to the AnalyticDB for PostgreSQL V4.3 instance.    |
| <PGPASSWORD> | The password for the user.  |

### Procedure

1. Run the following command to install a PostgreSQL client on your Linux device:

```
sudo yum install postgresql
```

2. View the public IP address of the Linux device. Add the public IP address of the Linux device to an IP address whitelist for the AnalyticDB for PostgreSQL V4.3 instance in the [AnalyticDB for PostgreSQL console](#). For more information about how to configure an IP address whitelist, see [Configure an IP address whitelist](#).

3. Connect to the AnalyticDB for PostgreSQL V4.3 instance by using the Linux device.

```
psql -h <PGHOST> -p <PGPORT> -U <PGUSER>
```

4. Save the edited shell script as a script file such as 4x\_to\_6x.sh. In this example, run the following command to run the 4x\_to\_6x\_check.sh script for incompatibility check:

```
sh 4x_to_6x_check.sh <PGHOST> <PGPORT> <PGUSER> <PGPASSWORD>
```

5. Modify the incompatible items of the AnalyticDB for PostgreSQL V4.3 instance based on returned error messages. After you modify the items, run the script again to verify whether the check items are compatible with AnalyticDB for PostgreSQL V6.0.

### Sample check results

*A "pass" message without "ERROR" indicates that the check items are compatible with*

**AnalyticDB for PostgreSQL V6.0. The following code shows that all check items are compatible with AnalyticDB for PostgreSQL V6.0:**

```

***** check base version...
pass.....
***** check untransferred libraries...
pass.....
***** check unsupported table distributedId types...
pass.....
***** check unsupported table field types...
pass.....
***** check unsupported extensions...
pass.....
    
```

**If the results contain one or more error messages, you must modify the incompatible configurations. The following code shows that all check items are incompatible with AnalyticDB for PostgreSQL V6.0:**

```

PostgreSQL 8.2.15 (Greenplum Database 4.3.99.00 build dev) compiled on May 2 2020 09:35:15
***** check base version...
ERROR: please upgrade minor version...
***** check untransferred libraries...
name  | language
-----+-----
 select_1 | plpgsql
(1 row)
WARN: please transfer libraries manually...
***** check unsupported table distributedId types...
dbname | schema | table_name | distributed_field | field_type
-----+-----+-----+-----+-----
 adbpq  | public | test1      | id                 | money
(1 row)
ERROR: please alter table distributedId types manually...
***** check unsupported table field types...
dbname | schema | table_name | field_name | field_type
-----+-----+-----+-----+-----
 adbpq  | public | test2      | name        | unknown
(1 row)
ERROR: please alter table field types manually...
***** check unsupported extensions...
dbname | extname | extversion
-----+-----+-----
 adbpq  | varbitx | 1.0
(1 row)
WARN: please drop useless extensions manually...
REF DROP EXTENSION SQL: drop extension <name>
    
```

| Error message | Modification method |
|---------------|---------------------|
|---------------|---------------------|

| Error message  | Modification method   |
|--|---|
| <b>ERROR: please upgrade minor version...</b>                    | <b>Upgrade the minor version</b> of the AnalyticDB for PostgreSQL V4.3 instance in the <a href="#">AnalyticDB for PostgreSQL console</a> . For more information, see <a href="#">Upgrade the engine version</a> .                                       |
| <b>WARN: please transfer libraries manually...</b>               | Migrate the libraries that are used in the AnalyticDB for PostgreSQL V4.3 instance.   |
| <b>ERROR: please alter table distributedId types manually...</b> | Modify the incompatible distribution keys in the AnalyticDB for PostgreSQL V4.3 instance.   |
| <b>ERROR: please alter table field types manually...</b>         | Modify the incompatible field types in the AnalyticDB for PostgreSQL V4.3 instance.   |
| <b>WARN: please drop useless extensions manually...</b>          | If you need to use the tables or stored procedures that are relevant to the incompatible extensions, modify the configurations of the extensions. If the incompatible extensions are not used in AnalyticDB for PostgreSQL V6.0, delete the extensions. |

# 13. Backup and restoration

## 13.1. Overview

AnalyticDB for PostgreSQL provides periodic base backup and log backup to prevent data loss.

### Principles

AnalyticDB for PostgreSQL uses base backup and log backup to restore instance data to a specific point in time and keep the state and data of a distributed instance consistent.

- **Base backup:** backs up all data of an instance. Base backup compresses a snapshot of all instance data and stores the snapshot to offline storage media. Base backup does not block your read and write operations. The operation logs generated during the base backup operation are also backed up to ensure the integrity of base backup.
- **Log backup** (also called incremental backup): backs up log files generated from an instance and stores them to offline storage media. Log files record the DML and DDL operations performed on databases.

AnalyticDB for PostgreSQL uses a full base backup and continuous log backups to restore data to new instances to a specific point in time and ensure data security within a period of time.

After the base backup data is restored, all nodes execute the data change statements recorded in the log backup files in sequence until the point in time for restoration. The point in time ensures consistency among all nodes when the data restoration of each node is complete.

### Scenarios

The backup and restoration feature can be used in the following scenarios:

- Instance data is lost for reasons such as accidental changes.
- The source instance data is corrupted or unavailable.
- An instance needs to be cloned with the same data.
- The instance specifications except for the number of nodes need to be changed.

### Limits

Data can be restored from the source instance only when the source instance has at least one backup set in the successful state.

### Billing

During public preview, AnalyticDB for PostgreSQL provides the backup and restoration feature free of charge. You may be charged for this feature in the future.

## 13.2. Back up data

This topic describes how to view backup data and modify backup settings in AnalyticDB for PostgreSQL.

### View and modify backup polices

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.

3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, click **Backup recovery**.
5. On the page that appears, click the **Backup settings** tab to view backup settings.
6. To modify the backup settings, click **Modify backup configuration**.
7. In the **Backup configuration** panel, set the following parameters.

| Parameter                         | Description  |
|-----------------------------------|--|
| <b>Data backup retention days</b> | The retention period of data backups. Valid values: 1 to 7. Unit: days.<br>Data backups can be retained for up to seven days during public preview.  |
| <b>Data backup cycle</b>          | The backup cycle. Select at least one day of the week for data backup.   |
| <b>Data backup start time</b>     | The time range within which you want to create a data backup. Example: <b>05:00-06:00</b> . We recommend that you back up data during off-peak hours.  |
| <b>Recovery point settings</b>    | Specifies whether to enable point-in-time restoration.<br>If this feature is disabled, data cannot be restored by point in time.   |
| <b>Recovery point frequency</b>   | The frequency at which you want to perform point-in-time restoration. Valid values: <ul style="list-style-type: none"> <li>◦ <b>Hourly</b></li> <li>◦ <b>Every two hours</b></li> <li>◦ <b>Every four hours</b></li> <li>◦ <b>Every eight hours</b></li> </ul> |

8. Click **OK**.

## View base backup sets

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, click **Backup recovery**.
5. On the page that appears, click the **Data backup** tab to view base backup sets.

Base backup data indicates a full copy of all database data. Base backup data is stored in Object Storage Service (OSS) and retained for seven days by default.

Information of base backup sets is displayed on this tab, as shown in the preceding figure. The information includes **Backup start time**, **Backup end time**, **Backup status (Success or Failure)**, **Backup size**, and **Consistency point in time**. **Consistency point in time** indicates the point in time to which you can restore instance data to ensure data consistency among databases.

## View restoration points in time

1. Log on to the [AnalyticDB for PostgreSQL console](#).

2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, click **Backup recovery**.
5. On the page that appears, click the **Recovery point** tab.
6. View restoration points in time on the **Recovery point** tab.

A restoration point in time indicates the point in time to which you can restore instance data.

The name and timestamp of each restoration point in time are displayed on this tab, as shown in the preceding figure. You can restore instance data to a restoration point in time.

## View log backups

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, click **Backup recovery**.
5. On the page that appears, click the **Log backup** tab.
6. View log backups on the **Log backup** tab.

Log files record all data changes to databases. An instance can be restored to a previous state by using its corresponding log files. All log files are stored in OSS.

# 13.3. Restore data

This topic describes how to restore backup data to a new instance in AnalyticDB for PostgreSQL.

## Precautions

- Both the source and destination instances must be in **elastic storage mode**.
- The source and destination instances must reside within the same region.
- The storage capacity of the destination instance must be greater than or equal to that of the source instance.
- The amount of time consumed by instance data restoration is determined by the volume of data stored in the source instance. Typically, the data of an instance is restored within a few hours.

## Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, click **Backup recovery**.
5. Click the **Data backup** or **Recovery point** tab. Find a backup that you want to use to restore data and click **Recovery** in the **Actions** column.
6. On the **Clone Instance** page, configure the parameters described in the following table.

| Parameter                           | Description  |
|-------------------------------------|--|
| <b>Source Instance</b>              | The source instance whose data you want to restore.  |
| <b>Restore Mode</b>                 | The mode that is used to restore data. It is automatically set to <b>Backup Set</b> .  |
| <b>Backup Set</b>                   | The point in time to which you want to restore data.   |
| <b>Region</b>                       | The region in which to clone the instance. It is automatically set to the region in which the source instance resides.   |
| <b>Zone</b>                         | The zone in which to clone the instance.   |
| <b>Network Type</b>                 | The network type of the destination instance. It is automatically set to <b>VPC</b> .  |
| <b>VPC</b>                          | The ID of the virtual private cloud (VPC).   |
| <b>VSwitch</b>                      | The vSwitch in the specified VPC.  |
| <b>Instance Resource Type</b>       | The resource type of the destination instance. It is automatically set to <b>Elastic Storage Mode</b> .  |
| <b>Edition</b>                      | <p>The edition of the destination instance. Set the edition to <b>High Performance (Basic Edition)</b> or <b>High-availability</b> based on your business requirements.</p> <p>Basic Edition is available only in some regions and zones. For more information about the specific regions and zones, see <a href="#">Scope</a>.</p> <p>For more information about Basic Edition, see <a href="#">Basic Edition</a>.</p>      |
| <b>Compute Node Specifications</b>  | The specifications of compute nodes in the destination instance.   |
| <b>Nodes</b>                        | The number of compute nodes in the destination instance.   |
| <b>Single Node Storage Capacity</b> | The storage capacity per compute node in the destination instance. The total storage capacity of compute nodes in the destination instance must be greater than or equal to that in the source instance.   |
| <b>Storage Disk Type</b>            | <p>The disk type. Valid values:</p> <ul style="list-style-type: none"> <li>◦ <b>Enhanced SSD (ESSD)</b></li> <li>◦ <b>Ultra Disk</b></li> </ul> <div style="background-color: #e0f2f7; padding: 5px; margin-top: 10px;"> <p> <b>Note</b> Only <b>Enhanced SSD (ESSD)</b> is available if you set the edition to Basic Edition.</p> </div> |

---

| Parameter              | Description   |
|------------------------|---|
| <b>Encryption Type</b> | The encryption type.<br>If you set the encryption type to <b>Disk Encryption</b> , you must also specify <b>CMK</b> . |

7. Read and select **Terms of Service**.
8. Click **Buy Now**.

 **Note** A new instance is created with the data of the source instance restored. The amount of time consumed by instance data restoration is determined by the volume of data stored in the source instance. Typically, the data of an instance is restored within a few hours.

# 14.Tags

## 14.1. Add tags to instances

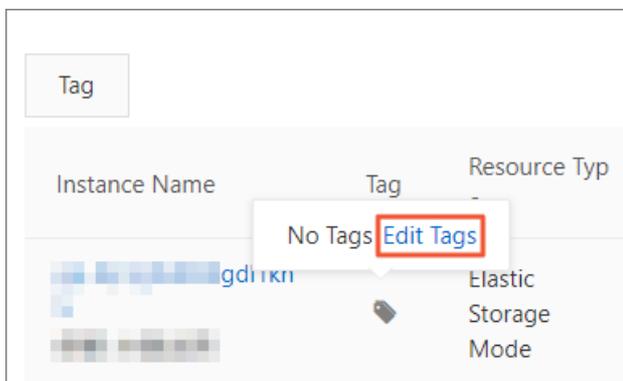
This topic describes how to add tags to one or more AnalyticDB for PostgreSQL instances. You can use tags to classify a large number of instances. Each tag consists of a key and a value. You can use tag keys and values to further classify instances.

### Limits

- You can add up to 20 tags to each instance. Each tag must have a unique key. If two tags have the same key, the tag that is created later overwrites the earlier tag.
- You can add tags to up to 50 instances at a time.
- Instances in different regions do not share the same tag namespace.
- After you remove a tag from an instance, AnalyticDB for PostgreSQL checks whether the tag is added to other instances. If not, AnalyticDB for PostgreSQL deletes the tag.

### Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance to which you want to add tags and move the pointer over the  icon in the **Tag** column.
4. Click **Edit Tags**.



5. In the **Edit Tags** dialog box, click **Create Tag**.

6. Enter a key in **Tag Key** and a value in **Tag Value**. Click **OK**.

? **Note** To add multiple tags, repeat this step.

7. After you add a tag, click **OK**.

## 14.2. Unbind a tag from an AnalyticDB for PostgreSQL instance

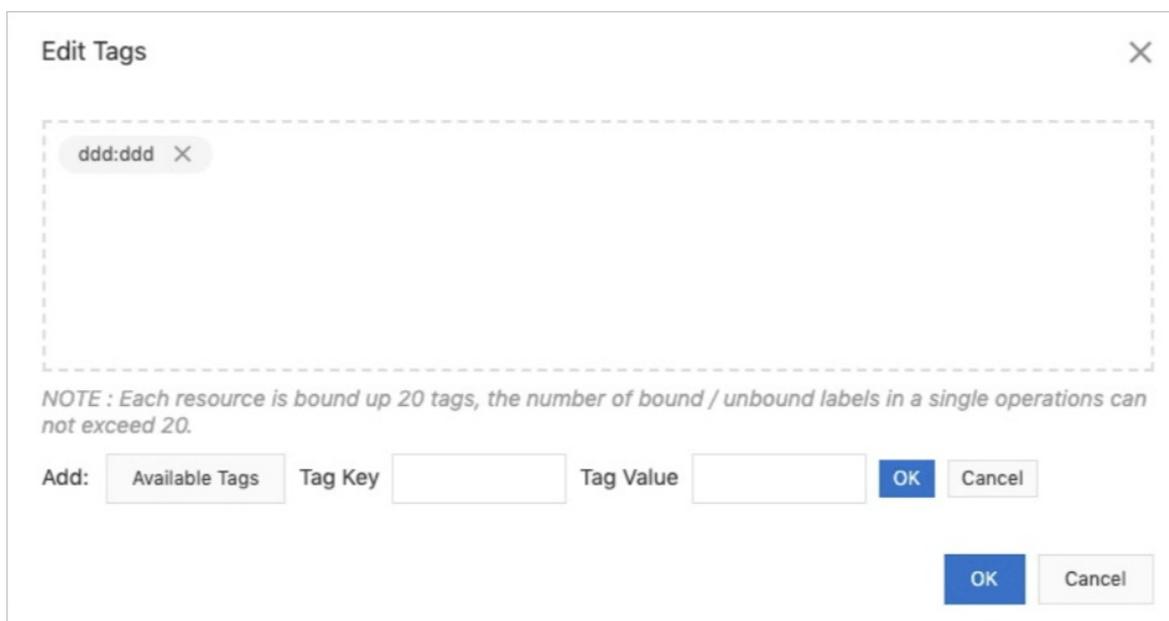
This topic describes how to unbind a tag from an AnalyticDB for PostgreSQL instance after the instance no longer needs the tag.

## Limits

- You can unbind a maximum of 20 tags at a time.
- After you unbind a tag from an instance, the system deletes the tag if the tag is not bound to any other instances.

## Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. Find the target instance, move the pointer over the icon in the Tag column, and click Edit Tags.
3. Click the X icon next to the tag you want to unbind from the instance.

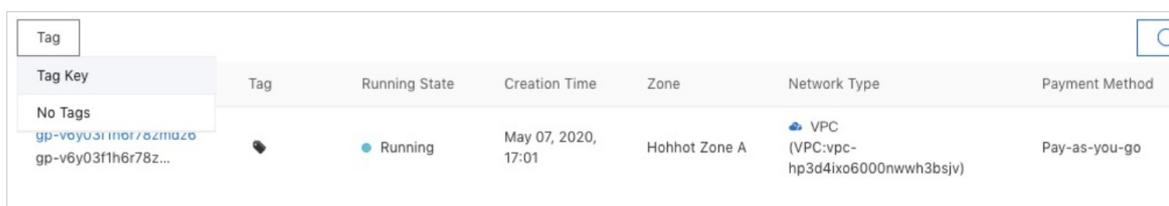


4. Click OK.

## 14.3. Filter AnalyticDB for PostgreSQL instances by tag

This topic describes how to filter AnalyticDB for PostgreSQL instances by tag after you bind tags to them.

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the **Instance List** page, click the **Tag** button. Then select a **tag key** and a **tag value** to filter instances.



# 15. Diagnostics and optimization

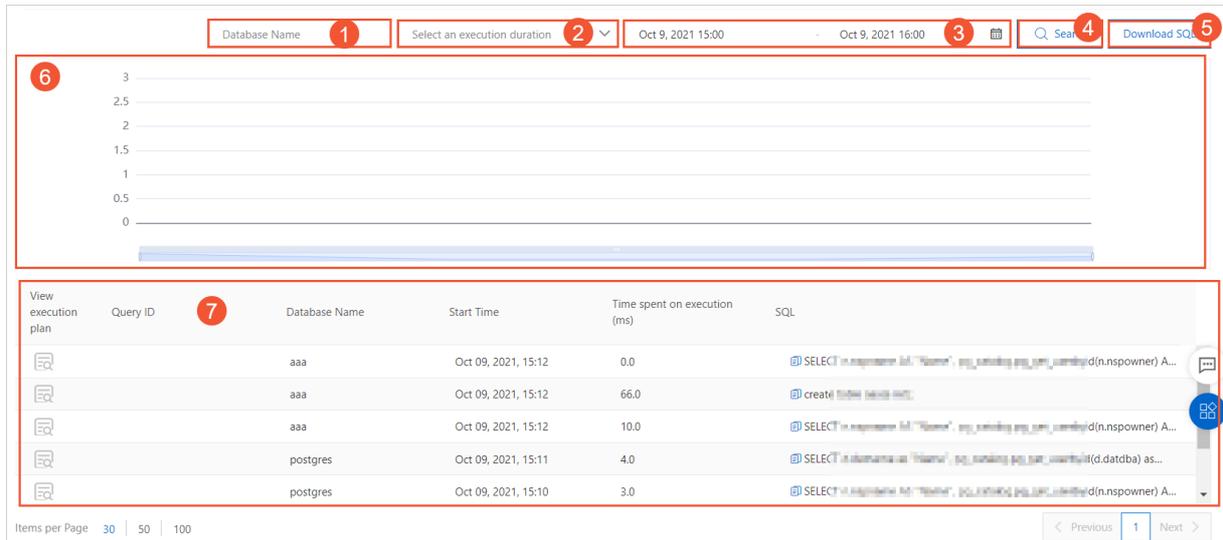
## 15.1. View slow SQL queries

You can analyze SQL queries executed within a specific time range, view execution plan details, and identify root causes of slow SQL queries in the AnalyticDB for PostgreSQL console.

### Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, choose **Diagnostics and Optimization > Slow SQL Queries**.
5. Select a time range and click **Search** to view the search results.

### View slow SQL queries



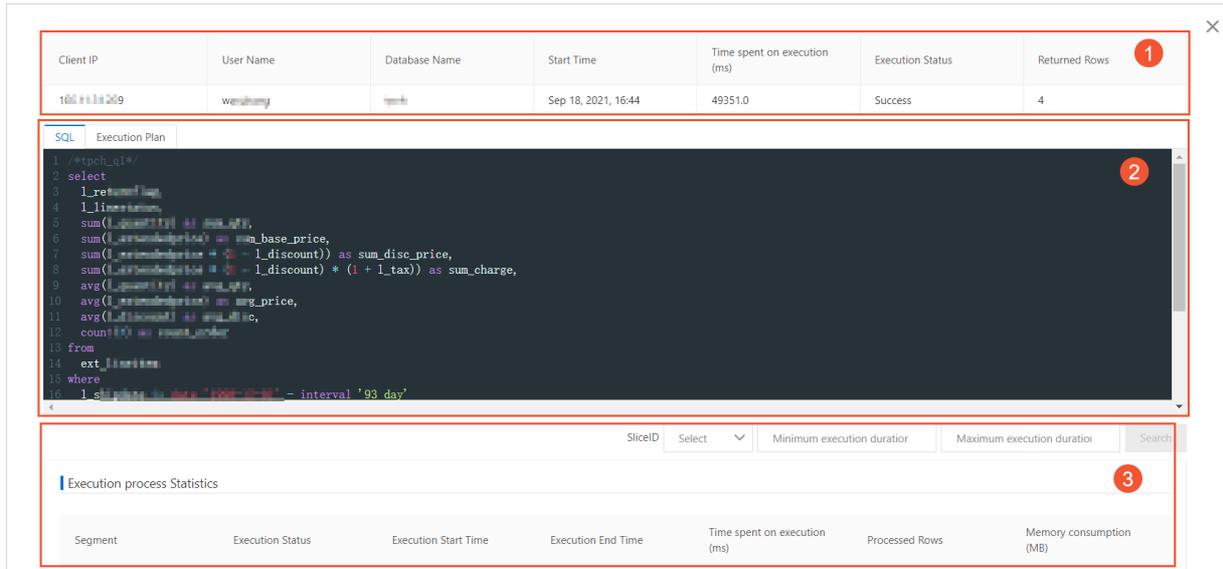
| No. | Parameter or section                | Description   |
|-----|-------------------------------------|---|
| ①   | <b>Database Name</b>                | Enter the name of a database. If you want to search for slow SQL queries related to all databases in the instance, leave this parameter empty.  |
| ②   | <b>Select an execution duration</b> | Select an SQL execution duration from the drop-down list. Valid values: <ul style="list-style-type: none"> <li>• Short (Duration &lt; 10 Seconds)</li> <li>• Medium (10 Seconds &lt;= Duration &lt; 10 Minutes)</li> <li>• Long (Duration &gt;= 10 Minutes)</li> <li>• Custom</li> </ul> If you select Custom, you can specify <b>Minimum Duration</b> and <b>Maximum Duration</b> to customize the duration range. Unit: <b>milliseconds</b> or <b>seconds</b> . |

| No. | Parameter or section                     | Description  |
|-----|--|--|
| ③   | <b>Time range</b>                        | <p>Select a time range to view slow SQL queries. By default, the last 1 hour is selected.</p> <div style="background-color: #e0f2f1; padding: 10px; border: 1px solid #ccc;"> <p><b>?</b> <b>Note</b></p> <ul style="list-style-type: none"> <li>You can search for slow SQL queries only within the last 14 days.</li> <li>The end date must be later than the start date. Their interval must be less than seven days.</li> </ul> </div> |
| ④   | <b>Search</b>                            | After you specify <b>Database Name</b> , <b>Select an execution duration</b> , and <b>time range</b> , click <b>Search</b> to view the search results.   |
| ⑤   | <b>Download SQL Queries</b>              | Click this button to download SQL query information displayed in the <b>SQL query details</b> section.   |
| ⑥   | <b>Line chart for SQL query quantity</b> | Allows you to view the change trend in the number of SQL queries executed within the specified time range.   |
| ⑦   | <b>SQL query details</b>                 | <p>Allows you to view <b>View Execution Plan</b>, <b>Query ID</b>, <b>Database Name</b>, <b>Start Time</b>, <b>Execution Duration (ms)</b>, and <b>SQL Statement</b> of SQL queries.</p> <div style="background-color: #e0f2f1; padding: 10px; border: 1px solid #ccc;"> <p><b>?</b> <b>Note</b> Query ID and <b>View Execution Plan</b> are displayed only for the SQL queries that are executed for longer than 1 second.</p> </div>     |

## View execution plan details

You can click the  icon in the **View Execution Plan** column to view execution plan details.

The following figure shows the execution plan details.



| No. | Section                          | Description  |
|-----|----------------------------------|--|
| ①   | Basic query information          | Allows you to view <b>Client IP</b> , <b>User Name</b> , <b>Database Name</b> , <b>Start Time</b> , <b>Execution Duration (ms)</b> , <b>Execution Status</b> , and <b>Returned Rows</b> of the current SQL query.  |
| ②   | SQL Statement and Execution Plan | <ul style="list-style-type: none"> <li>• <b>SQL Statement</b>: allows you to view the SQL statement information.</li> <li>• <b>Execution Plan</b>: allows you to view the execution plan tree that is displayed based on the PEV2 library. You can click the hidden nodes to view the execution plan details.</li> </ul> |
| ③   | Execution Process Statistics     | Allows you to view the execution process statistics of an execution plan, including <b>Compute Node</b> , <b>Execution Status</b> , <b>Execution Start Time</b> , <b>Execution End Time</b> , <b>Execution Duration (ms)</b> , <b>Processed Rows</b> , and <b>Memory Consumed (MB)</b> .                                 |

## 15.2. View SQL distribution statistics

You can view the proportions of SQL queries by using dimensions such as databases, SQL types, SQL execution durations, and users within a specific time range in the AnalyticDB for PostgreSQL console.

### Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. In the left-side navigation pane, choose **Diagnos**tics and Optimization > **SQL Distribution Statistics**.
5. On the **SQL Distribution Statistics** page, select a time range and click **Search** to view SQL

distribution statistics.

AnalyticDB for PostgreSQL allows you to view the proportions of SQL queries by using the following dimensions: databases, SQL types, SQL execution durations, and users.

# 16. CloudMonitor

## 16.1. View monitoring data and configure an alert rule

This topic describes how to configure an alert rule for an AnalyticDB for PostgreSQL instance to monitor resource usage. If the conditions specified in an alert rule are met, the system notifies all contacts in the specified contact groups.

### Background information

The monitoring and alerting feature of AnalyticDB for PostgreSQL is implemented by using [CloudMonitor](#). CloudMonitor allows you to set metrics and specify contact groups. If an alert is triggered, CloudMonitor notifies all contacts in the specified contact groups. You can maintain contact groups for metrics to ensure that the contacts receive alerts at the earliest opportunity.

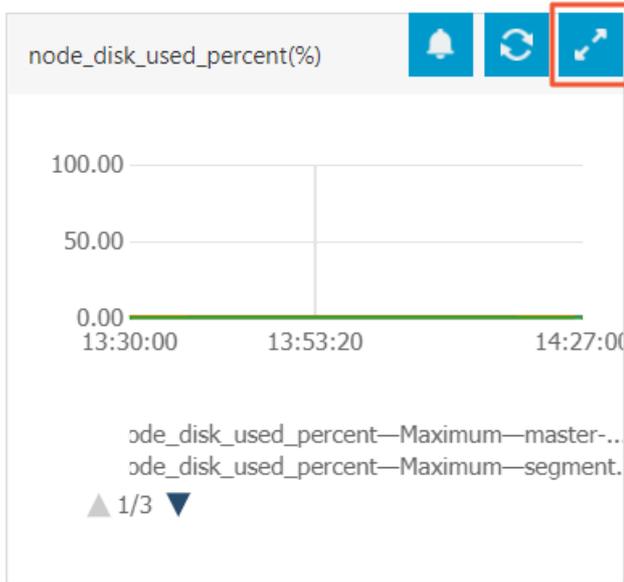
CloudMonitor supports threshold-triggered and event-triggered alert rules. The following table describes the supported alert rule types for different instance resource types.

| Instance resource type | Threshold-triggered alert | Event-triggered alert |
|------------------------|---------------------------|-----------------------|
| Elastic storage mode   | Supported                 | Not supported         |
| Reserved storage mode  | Supported                 | Supported             |

 **Note** To enable the monitoring and alerting feature, you must configure alert rules.

### View monitoring data

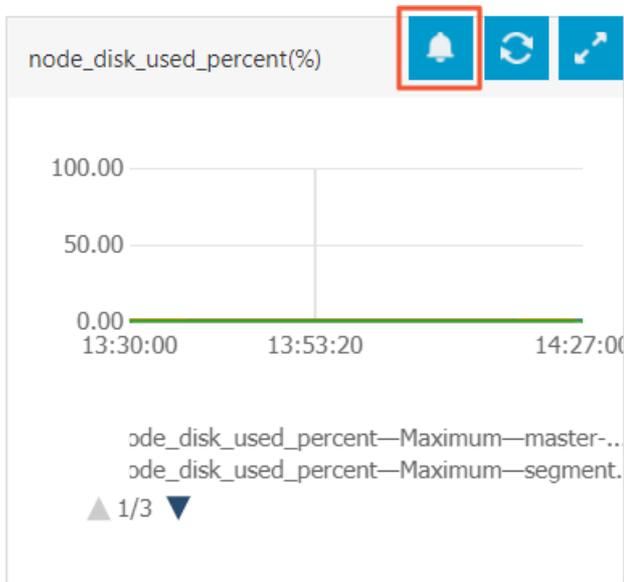
1. Log on to the [CloudMonitor console](#).
2. In the left-side navigation pane, click **Cloud products**.
3. On the **Cloud products** page, choose **Database > AnalyticDB for PostgreSQL**.
4. Click the ID of the AnalyticDB for PostgreSQL instance to go to the **Cloud product charts - AnalyticDB for PostgreSQL** page.
5. Move the pointer over a chart and click the enlarge icon to zoom in the chart.



**Note** The metric names of compute nodes are displayed in different manners in the CloudMonitor and AnalyticDB for PostgreSQL consoles. In the AnalyticDB for PostgreSQL console, the names of compute nodes are displayed. In the CloudMonitor console, the names of the hosts for compute nodes are displayed.

### Configure a threshold-triggered alert

1. In the CloudMonitor console, move the pointer over a chart and click the alert icon to configure a threshold-triggered alert for the corresponding metric.



2. On the Create Alert Rule page, configure an alert rule. To configure multiple alert rules, click **Add Alert Rule** in the Set Alert Rules section. You can create an individual alert rule for each metric. If the value of a metric exceeds the related threshold, the system sends you an alert notification.
  - o Metric: For information about the metrics and recommended settings, see [Metrics and system events](#).
  - o 1Minute cycle: The monitoring data within 1 minute is aggregated into one monitoring data

point for comparison with the specified threshold. CloudMonitor provides one data point per minute. If you select 1Minute cycle, one data point is generated and no aggregation is required. If you select 5Minute cycle, five data points are generated and the system aggregates the five data points into a single data point.

- Duration: If you select Continue for 3 periods for 1Minute cycle, the alert is triggered after the metric value exceeds the threshold for 3 minutes.
  - Average, Max. Value, and Min. Value: If you select 5Minute cycle, five data points must be aggregated. Assume that the five data points are 10, 20, 30, 40, and 50. The average value is 30, the maximum value is 50, and the minimum value is 10. You can specify a threshold for the average value, the maximum value, or the minimum value.
  - instance\_component: You can select a specific server or component for which you want to create the alert rule. You can also select All.
  - Mute for: This parameter specifies the interval at which an alert notification is sent if the issue is not rectified after the first notification.
3. Select an existing contact group or create a contact group.
  4. (Optional) Enter the email content.
  5. Click **Confirm** to go to the Alert Rules page. The created threshold-triggered alert rule is displayed.

On the Threshold Value Alert tab, you can view the alert state or alert logs. You can also disable the alert.

## Configure an event-triggered alert

1. In the [CloudMonitor](#) console, choose **Alerts > Alert Rules**.
2. On the **Event Alert** tab, click **Create Event Alert**.
  - Product Type: Select AnalyticDB for PostgreSQL.
  - Event Type: Select All types or a specific type.
  - Event Level: Select All Levels or a specific level.
  - Event Name: Select All Events or a specific event.

For more information about system events, see [Metrics and system events](#).

3. Set Contact Group and Notification Method, and then click **OK**.

On the Event Alert tab, you can view the alert state or alert logs. You can also disable the alert.

## 16.2. Configure an alert template

You can save the alert rules configured for an AnalyticDB for PostgreSQL instance into an alert template. Then, you can create or modify an alert rule directly by using the alert template instead of repeatedly configuring the alert rule.

### Precautions

- Alert templates must be used together with application groups. You can create an application group first. Then, you can create an alert template and apply the alert template to the application group to create alert rules for the application group. This simplifies the creation and maintenance of alert rules. For information about how to create an application group, see [创建应用分组](#).
- When you create an alert template, take note of the following limits:

- Up to 100 alert templates can be created for each Alibaba Cloud account.
- Up to 30 metrics can be specified in each alert template.

## Create an alert template

1. Go to the **Alert Templates** page. Perform the following steps:
  - i. Log on to the [CloudMonitor console](#).
  - ii. In the left-side navigation pane, choose **Alerts > Alert Templates**.
2. Create an alert template. Perform the following steps:
  - i. In the upper-right corner of the **Alert Templates** page, click **Create Alert Template**.
  - ii. In the **Create/modify alert templates** panel, enter a template name in the **Template Name** field, click **Products**, and then select **AnalyticDB for PostgreSQL** from the drop-down list.
  - iii. Click **Add Rules** in the **AnalyticDB for PostgreSQL** section to configure alert rules for the alert template.

The following types of alert templates are supported in CloudMonitor:

### ■ Metric Template

The following table describes the parameters that you can configure for the rules in a metric template.

| Parameter   | Description  |
|---|--|
| <b>Rule Name</b>  | The name of the threshold-triggered alert rule.  |
| <b>Metric Name</b>                                      | The name of the metric. For more information about the metrics in AnalyticDB for PostgreSQL, see <a href="#">Metrics and system events</a> .   |
| <b>role</b><br>or<br><b>instance_component,hostname</b> | The compute node or compute group to monitor.<br>You can select <b>All</b> to monitor all instances or select a specific compute node or compute group to monitor within an instance.  |
| <b>Threshold and Notification Methods</b>               | The alert condition, alert threshold, and alert level of the rule.<br>You can configure multiple alert levels. When the alert threshold falls within different ranges, the rule triggers alerts at different alert levels.<br>CloudMonitor sends alert notifications by using different methods. |

### ■ Event Template

Event-triggered alert rules can be configured only for system events. The following table describes the parameters that you can configure for the rules in an event template.

| Parameter               | Description   |
|-------------------------|---|
| <b>Rule Name</b>        | The name of the event-triggered alert rule.   |
| <b>Event Type</b>       | The type of the event that triggers alerts. For more information, see <a href="#">Metrics and system events</a> .   |
| <b>Event Level</b>      | The level of the event that triggers alerts. For more information, see <a href="#">Metrics and system events</a> .  |
| <b>Event Name</b>       | The name of the event that triggers alerts. For more information, see <a href="#">Metrics and system events</a> .   |
| <b>MNS queue</b>        | The specified queue in Message Service (MNS) to which the event alert is delivered.   |
| <b>Function service</b> | The specified function of Function Compute to which the event alert is delivered.   |
| <b>URL callback</b>     | The callback URL and request method. Enter a callback URL that can be accessed over the Internet. CloudMonitor sends a POST or GET request to push an alert to the specified callback URL. Only HTTP requests are supported. For more information about how to configure callbacks, see <a href="#">Configure callbacks for system event-triggered alerts</a> . |
| <b>Log Service</b>      | The specified Logstore in Log Service to which the event alert is delivered.  |

- iv. Click OK to save the alert rule.
  - v. Click OK to save the alert template.
3. In the **Create/modify alert template complete** message, click OK.

 **Note** If you click Cancel, the alert template is created but not applied to application groups. For more information about how to apply an alert template to an application group, see [应用报警模板到应用分组](#).

4. In the **Apply Template to Group** dialog box, select an application group and set the Muted, Effective Period, HTTP WebHook, and Option parameters.
5. Click OK.
6. In the **Apply Template to Group** message, click OK.

## References

- [View monitoring data and configure an alert rule](#)
- [Create an alert template](#)

## 16.3. Metrics and system events

This topic describes the CloudMonitor metrics and system events in AnalyticDB for PostgreSQL.

## Metrics

The following tables describe the metrics and recommended settings for AnalyticDB for PostgreSQL instances in elastic storage mode and reserved storage mode.

### Elastic storage mode

| Metric                 | Unit | Recommended cycle                       | Recommended threshold  |
|------------------------|------|---|--|
| adbpg_conn_count       | N/A  | 1Minute cycle<br>Continue for 5 periods | Average >= 90%.<br>Assume that you have an instance that has 2 cores and 16 GB memory, and the maximum number of connections allowed for the instance is 550. You can set the metric to 495, which is obtained by using the following formula: 550 × 90%. For more information about the maximum number of user connections of instances, see <a href="#">Limits</a> . |
| node_cpu_used_percent  | %    | 1Minute cycle<br>Continue for 5 periods | Average >= 90%.  |
| node_disk_iops         | N/A  | None                                    |  |
| node_disk_used_percent | %    | 1Minute cycle<br>Continue for 5 periods | Max. value >= 80%.   |
| node_mem_used_percent  | %    | 1Minute cycle<br>Continue for 5 periods | Average >= 90%.  |

### Reserved storage mode

| Metric          | Unit | Recommended cycle                       | Recommended threshold |
|-----------------|------|---|-----------------------|
| ConnectionUsage | %    | 1Minute cycle<br>Continue for 5 periods | Average >= 90%.       |
| CpuUsage        | %    | 1Minute cycle<br>Continue for 5 periods | Average >= 90%.       |

| Metric      | Unit | Recommended cycle                       | Recommended threshold |
|-------------|------|---|-----------------------|
| DiskUsage   | %    | 1Minute cycle<br>Continue for 5 periods | Average >= 80%.       |
| IOPSUsage   | %    | 1Minute cycle<br>Continue for 5 periods | Average >= 90%.       |
| MemoryUsage | %    | 1Minute cycle<br>Continue for 5 periods | Average >= 90%.       |

## System events

The following table describes the system events of AnalyticDB for PostgreSQL instances.

| Type      | Event                    | Description  | Status | Level    |
|-----------|--------------------------|--|--------|----------|
| Exception | ComputeGroupCpuHigh      | The CPU utilization of a compute group exceeds 90%.                    | Failed | Critical |
| Exception | ComputeGroupMemoryHigh   | The memory usage of a compute group exceeds 85%.                       | Failed | Critical |
| Exception | HighComputeGroupDiskUsed | The disk usage of a compute group exceeds 80%.                         | Failed | Critical |
| Exception | InstanceDown             | An instance is unavailable.  | Failed | Critical |
| Exception | LongTransaction          | The number of long-running transactions is greater than or equal to 5. | Failed | Critical |
| Exception | MasterCpuHigh            | The CPU utilization of a master node exceeds 90%.                      | Failed | Critical |
| Exception | MasterMemoryHigh         | The memory usage of a master node exceeds 85%.                         | Failed | Critical |

---

| Type      | Event              | Description   | Status | Level    |
|-----------|--------------------|---|--------|----------|
| Exception | NonSuperConnection | The proportion of connections to an instance exceeds 80%. | Failed | Critical |
| Exception | XidAgeHigh         | The age of a transaction ID (XID) exceeds 400 million.    | Failed | Critical |

## References

- [View monitoring data and configure an alert rule](#)
- [Configure an alert template](#)

# 17. Release an instance

This topic describes how to manually release pay-as-you-go AnalyticDB for PostgreSQL instances based on your business requirements.

## Note

- You cannot manually release subscription instances. Subscription instances are automatically released when they expire.
- You can manually release pay-as-you-go instances only when they are in the **Running** state.

## Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the upper-left corner of the console, select the region where the instance resides.
3. Find the instance that you want to manage and click its ID.
4. On the **Basic Information** page of the instance, click **Release** in the **Status** section.
5. In the **Release Instance** message, click **OK**.

 **Warning** After an instance is released, it cannot be restored. Proceed with caution.

## Related operations

| API                              | Description           |
|----------------------------------|-----------------------|
| <a href="#">DeleteDBInstance</a> | Releases an instance. |

# 18. Configure parameters

AnalyticDB for PostgreSQL allows you to configure some key kernel parameters to meet customized requirements in various scenarios. This topic describes how to configure these parameters in the AnalyticDB for PostgreSQL console.

## Procedure

1. Log on to the [AnalyticDB for PostgreSQL console](#).
2. In the top navigation bar, select the region where the instance for which you want to configure parameters resides.
3. Find the instance and click its ID to go to the Basic Information page.
4. In the left-side navigation pane, click **Parameter Configuration**.
5. Click the Edit icon next to the parameter that you want to modify, modify the value of the parameter, and then click **OK**.

| Parameter         | Description   |
|-------------------|---|
| statement_timeout | The maximum timeout period. The default value is 10800000. Unit: milliseconds. If the timeout period exceeds the value of the parameter, the statement is terminated.   |
| rds_master_mode   | The consistency model of the AnalyticDB for PostgreSQL instance. This parameter is valid only when the instance has multiple coordinator nodes. Default value: Single. Valid values: <ul style="list-style-type: none"><li>◦ Single: The instance has only a single coordinator node.</li><li>◦ multi_write_ec: the session consistency model. Monotonic reads, monotonic writes, reads after writes, and writes after reads are consistent in a session.</li><li>◦ multi_write_sc: the global consistency model. This mode provides compatibility of atomicity, consistency, isolation, durability (ACID) and linearizability.</li></ul> |