

Application Performance Management 2.0

User Guide

Issue	01
Date	2026-09-18



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2026. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 Before You Start.....	1
2 Application List.....	4
3 CMDB Management.....	7
3.1 Introduction.....	7
3.2 Creating an Application.....	8
3.3 Creating a Sub-application.....	10
3.4 Configuring an Application and Sub-application.....	11
3.5 Checking the Basic Information About an Application, Sub-Application, Component, and Environment	13
4 Application Metric Monitoring.....	16
4.1 Introduction.....	16
4.2 Application Monitoring Details.....	17
4.2.1 Topology.....	18
4.2.2 URL.....	21
4.2.3 JVM.....	25
4.2.4 Exception.....	31
4.2.5 Call.....	32
4.2.6 SQL.....	37
4.2.7 Web Container.....	46
4.2.8 Cache.....	47
4.2.9 Profiler Performance Analysis.....	48
4.2.9.1 Overview.....	49
4.2.9.2 Profiler.....	50
4.2.9.3 Performance Analysis.....	51
4.2.9.4 Comparison.....	54
4.2.9.5 Snapshots.....	57
4.2.9.6 Suggestions.....	59
4.2.10 Debugging Diagnosis.....	60
4.2.10.1 Overview.....	60
4.2.10.2 Thread Analysis.....	61
4.2.10.3 Method Analysis.....	63
4.2.10.4 Object Analysis.....	65

4.3 Application Monitoring Configuration.....	66
4.3.1 Configuration Details.....	66
4.3.2 Configuring the MySQL Monitoring Item.....	68
4.3.3 Configuring the HttpClient Monitoring Item.....	69
4.3.4 Configuring the URL Monitoring Item.....	71
4.3.5 Configuring the JavaMethod Monitoring Item.....	74
4.3.6 Configuring the Druid Monitoring Item.....	74
4.3.7 Configuring the ApacheHttpAsyncClient Monitoring Item.....	74
4.3.8 Configuring the Redis Monitoring Item.....	74
4.3.9 Configuring the Jedis Monitoring Item.....	75
4.3.10 Configuring the HBase Monitoring Item.....	75
4.3.11 Configuring the ApacheHttpClient Monitoring Item.....	75
4.3.12 Configuring the Tomcat Monitoring Item.....	75
4.3.13 Configuring the EsRestClient Monitoring Item.....	75
4.3.14 Configuring the WebSocket Monitoring Item.....	75
4.3.15 Configuring the KafkaProducer Monitoring Item.....	76
4.3.16 Configuring the Hikari Monitoring Item.....	76
4.3.17 Configuring the Exception Monitoring Item.....	76
4.3.18 Configuring the Thread Monitoring Item.....	76
4.3.19 Configuring the GC Monitoring Item.....	76
4.3.20 Configuring the JVMInfo Monitoring Item.....	77
4.3.21 Configuring the JVMMonitor Monitoring Item.....	77
4.3.22 Configuring ProbeInfo Monitoring Item.....	77
4.4 Monitoring Item Views.....	77
4.5 Instance.....	77
4.6 Collection Status.....	79
4.7 Component Settings.....	80
5 Tracing.....	88
6 Application Topology.....	96
6.1 Viewing a Component Topology.....	96
6.2 Viewing a Global Application Topology.....	96
7 URL Tracing.....	98
8 Resource Tag Management.....	102
9 Managing Tags.....	105
10 Alarm Management.....	108
10.1 Alarm List.....	108
10.2 Alarm Policies.....	109
10.2.1 Configuring an Alarm Template.....	109
10.2.2 Creating a Custom Alarm Policy.....	114
10.2.3 Recommended Alarm Templates.....	115

10.3 Alarm Notification.....	116
11 Agent Management.....	118
11.1 Overview.....	118
11.2 Agent Download Addresses.....	118
11.3 Agent Access Addresses.....	121
11.4 Performing Operations on Agents.....	121
11.5 Upgrading Agents.....	123
12 Web Monitoring.....	124
12.1 Overview.....	124
12.2 Websites.....	124
12.3 Overview.....	133
12.4 Load Performance.....	138
12.5 API Requests.....	150
12.6 JS Errors.....	155
12.7 Access Analysis.....	157
12.8 Custom Statistics.....	163
12.9 Session Tracing.....	165
13 Open Tracing.....	172
13.1 Overview.....	172
13.2 Application Management.....	172
13.3 Metrics.....	176
13.3.1 Overview.....	176
13.3.2 Topology.....	177
13.3.3 URL.....	179
13.3.4 SQL.....	180
13.3.5 Exception.....	181
13.3.6 JVM.....	182
13.3.7 Alarm Policies.....	183
13.3.8 Component Settings.....	186
13.4 Tracing.....	188
13.5 Agent Access.....	191
14 App Monitoring.....	205
14.1 Overview.....	205
14.2 App List.....	205
14.3 Overview.....	210
14.4 Crashes.....	213
14.5 Freezes.....	218
14.6 Errors.....	222
14.7 Startup Performance.....	226
14.8 Network Requests.....	229
14.9 Devices.....	232

14.10 Custom Statistics.....	241
15 System Management.....	244
15.1 Collection Center.....	244
15.2 Data Masking.....	246
15.3 Usage Statistics.....	248
15.4 Access Keys.....	248
15.5 General Configuration.....	250
16 Performance Monitoring.....	252
16.1 Performance Monitoring Overview.....	252
16.2 Monitoring Component Performance Metrics.....	255
16.2.1 Configuring Component Settings for Performance Monitoring.....	258
16.2.2 Viewing the Component Overview.....	266
16.2.3 Viewing the Call Relationships Between Components in the Topology.....	268
16.2.4 Viewing the URL Monitoring Item of a Component.....	270
16.2.5 Viewing the Info Monitoring Item of a Component.....	271
16.2.6 Viewing the Exception Monitoring Item of a Component.....	271
16.2.7 Viewing the External Call Monitoring Item of a Component.....	273
16.2.8 Viewing the Database Monitoring Item of a Component.....	274
16.2.9 Viewing the Cache Monitoring Item of a Component.....	275
16.2.10 Viewing the Web Container Monitoring Item of a Component.....	276
16.2.11 Viewing the Message Queue Monitoring Item of a Component.....	277
16.2.12 Viewing the Log Monitoring Item of a Component.....	277
16.2.13 Viewing the Infrastructure of a Component.....	278
16.2.14 Viewing the Traces of a Component.....	278
16.2.15 Viewing the Communication Protocol Monitoring Item of a Component.....	279
16.2.16 Viewing the RPC Monitoring Item of a Component.....	280
16.3 Viewing the Call Relationships Between Microservices Through Tracing.....	281
16.4 Viewing the Call Relationships Between Applications in the Global Topology.....	286
16.5 Managing Performance Monitoring Agents.....	287
17 SDK Reference.....	289
17.1 Privacy and Sensitive Information Protection Statement.....	289
17.1.1 Privacy Statement of APM Data Collection SDKs.....	289
17.1.2 Developer Compliance Guide for APM Data Collection SDKs.....	292
17.2 iOS SDK Access.....	295
17.3 Android SDK Access.....	301
17.4 Web & HTML5 SDK Access.....	310
17.5 WeChat SDK Access.....	312
17.6 Baidu SDK Access.....	317
17.7 Alipay SDK Access.....	320
17.8 DingTalk SDK Access.....	324
17.9 HarmonyOS SDK Access.....	327

17.10 Flutter SDK Access..... 331

17.11 Error Codes..... 337

18 Using IAM to Grant Access to APM..... 340

18.1 Using IAM Roles or Policies to Grant Access to APM..... 340

18.2 Using IAM Identity Policies to Grant Access to APM..... 343

19 CTS Auditing..... 346

19.1 APM Operations Supported by CTS..... 346

19.2 Viewing CTS Traces in the Trace List..... 352

1 Before You Start

This document describes how to use Application Performance Management (APM).

Application List	The Applications page displays information such as components, environments, Agent status, and supported operations.
CMDB Management	APM has built-in CMDB for managing the application structure and related configurations.
Application Metric Monitoring	APM can manage tags and monitor the metric data of JVM, GC, service calls, exceptions, external calls, database access, Profiler performance analysis, debugging diagnosis, and middleware, helping you comprehensively monitor application running.
Tracing	Information such as the call status, duration, and API is displayed, helping you further locate fault causes.
Application Topology	The call and dependency relationships between applications are displayed, and abnormal instances can be automatically discovered. Application topologies are classified into single-component topologies, cross-application global topologies, and global application topologies. Single-component topology: topology of a single component under an environment. You can also view the call relationships of direct and indirect upstream and downstream components. Global application topology: topology of some or all components under an application. Cross-application global topology: topology of all components under one or more applications.
URL Tracing	Through URL tracing, you can monitor the call relationships between important APIs and downstream services, and then detect problems more precisely.

Resource Tag Management	You can tag resources under your account for classification.
Managing Tags	You can add tags for different environments and applications for easy management.
Alarm Management	When an application connected to APM meets a preset alarm condition, an alarm is triggered and reported in a timely manner. In this way, you can quickly learn about service exceptions and rectify faults to prevent loss.
Agent Management	Agent Management allows you to check the deployment and running statuses of the Agents that are connected to APM, and to stop, start, or delete them.
Web Monitoring	APM Agents regularly collect performance metric data to measure the overall health of websites, HTML5 pages, and applets. They collect the data about loading performance, API requests, JS errors, access analysis, custom statistics, session tracing, scenario analysis, and network analysis, enabling you to monitor your websites.
Open Tracing	Open tracing collects complete traces of distributed applications and provides data about topologies, URLs, databases, and exceptions. Alarm policies can be customized as required. OpenTelemetry and SkyWalking support association with logs. Therefore, developers can quickly analyze and diagnose performance bottlenecks in the distributed application architecture and improve development and diagnosis efficiency of microservices.
App Monitoring	APM Agents regularly collect performance metric data to measure the overall health status of Android, iOS, and HarmonyOS apps. The collected data includes crashes, application not responding (ANR), errors, startup performance, network requests, devices, and custom statistics, helping you better monitor your app running.

System Management	System Management includes Collection Center, Data Masking, Usage Statistics, Access Keys, and General Configuration. Collection Center: displays collectors in a centralized manner. You can view and manage various collectors, metrics, and collection parameters supported by APM. Data Masking: You can set policies to mask the data reported using APM APIs. Usage Statistics: After Agents are connected, you can check Agent Statistics on the Usage Statistics page. Access Keys: long-term identity credentials. They ensure that the requests are secret, complete, and correct. General Configuration: You can determine whether to collect data through bytecode instrumentation, and specify the slow request threshold and maximum number of rows to collect. You can also specify the slow SQL request threshold and set web monitoring aggregation.
Performance Monitoring	To improve user experience, APM provides performance monitoring, which addresses issues such as a lack of O&M view, difficult-to-use topology, and insufficient association analysis.
SDK Reference	APM provides different SDKs for connecting iOS, Android, HarmonyOS, Flutter, browser/HTML5, and applets.
Using IAM to Grant Access to APM	Enterprise Project Management Service (EPS) is used to control user access to APM resources.
CTS Auditing	Cloud Trace Service (CTS) auditing, which involves: Recording APM operations: With CTS, you can record operations associated with cloud servers for future query, audit, and backtracking. Querying APM operation records: After you enable CTS and create the management tracker, CTS starts recording operations on cloud resources.
Learn more	Getting Started Learn how to connect applications to APM in different scenarios.

2 Application List

Application List

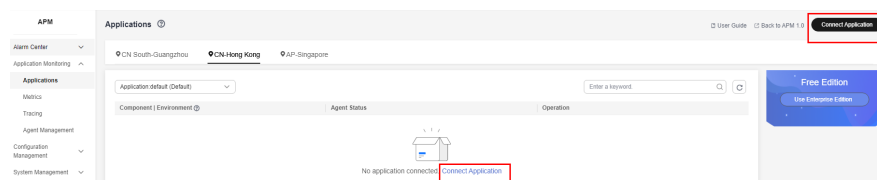
The **Applications** page displays information such as components, environments, Agent status, and supported operations.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **Application Monitoring > Applications**.

Figure 2-1 Application list



Component|Environment: name of a component or environment. You can click the text in blue to go to the corresponding to component or environment page.

Agent Status: number of Agents in different statuses.

The following table describes the Agent statuses.

Status	Description
Enabled	The Agent is running properly.
Offline	- The Agent is offline due to network problems. Check and restore the network. - The Agent is offline if your process does not exist. - The Agent is disabled and offline if the trial period expires.

Status	Description
Disabled	The Agent is manually or globally disabled. Contact technical support.

----End

More Operations

Perform the operations listed in [Table 2-1](#) if needed.

Table 2-1 Related operations

Operation	Description
Selecting an application	Select an application from the Application drop-down list on the left of the page.
Viewing the topology of an environment	Click Topology in the Operation column of an environment.
Setting a component or environment	Click Configure in the Operation column. On the displayed Instance tab page, set the component or environment as required.
Deleting an environment	Click Delete in the Operation column of an environment.
Searching for a component or environment	Enter a component or environment keyword or name on the right.
Back to APM 1.0	In the upper right corner of the page, click Back to APM 1.0 .

Subscribing to Enterprise-Edition APM

After you subscribe to the enterprise-edition APM, preferential package information will be displayed. You can purchase preferential packages online.

- Step 1** In the navigation pane on the left, choose **Application Monitoring > Applications**.
- Step 2** Click **Buy Package** on the right. The **Buy Package** page is displayed.
- Step 3** Select a region.
- Step 4** Click a certain type of Agent package and click + or - to increase or decrease the number of packages as required. If needed, select Agent packages of multiple levels.

Step 5 Click **Buy Now**. The list of selected Agent packages is displayed.

Step 6 Read the [APM Service Agreement](#) and [Privacy Statement](#) and then select the checkbox to confirm that you have read them.

Step 7 Click **Submit**. The APM purchase information is displayed.

Step 8 Select a payment method and pay money.

- The current edition is the enterprise edition. The pay-per-use price is \$0.96 USD Agent × day. Select required preferential packages to save money.
- The Agent quota of your preferential package will be used first. If the quota is used up or your package expires, APM will charge by your usage.
- After you purchase a preferential package, it cannot be unsubscribed. After the package expires, you can continue using APM and your data is still secure.
- Please complete the payment within three days. Otherwise, the order will be automatically canceled.

----End

3 CMDB Management

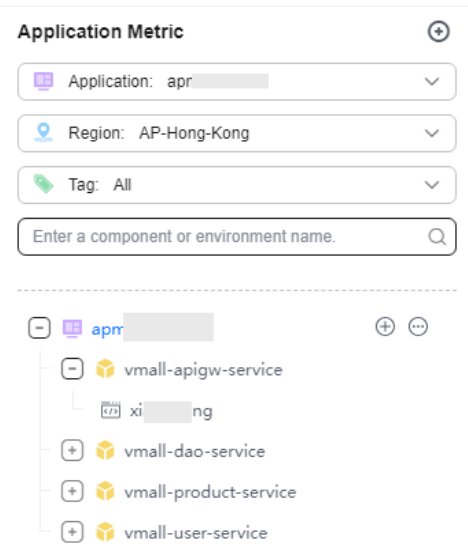
3.1 Introduction

APM has a built-in CMDB for managing application structure information and related configurations. It involves the following concepts:

- **Enterprise project:** An enterprise project can contain one or more applications.
- **Application** (global concept): a logical unit. An application can be an independent functional module. The same application information can be viewed in all regions. It is optional to associate an application with an enterprise project. The application associated with an enterprise project is managed based on enterprise project permissions. The application not associated with any enterprise project is managed based on the Identity and Access Management (IAM) permissions.
- **Sub-application** (global concept): similar to a folder. There can be up to three layers of sub-applications under an application.
- **Component** (global concept): a program or microservice. It is generally used together with environments. It may contain one or more environments. For example, an order app can be deployed in the function test environment, pressure test environment, pre-release environment, or live network environment.
- **Environment:** Components or programs with different configurations are deployed in different environments. Each environment has its own region attribute. You can filter environments by region. You can also add one or more tags to an environment and filter environments by tag.
- **Instance:** a process in an environment. It is named in the format of "host name+IP address+instance name". An environment is usually deployed on different hosts or containers. If an environment is deployed on one host, differentiation by instance is supported.
- **Environment tag:** an attribute for filtering environments. Different environments may have the same tag. Tags carry public configuration capabilities. For example, the configuration set on a tag can be shared by the environments with the same tag. Tags defined for environments of one application cannot be applied to other applications.

The following shows an example of the CMDB structure.

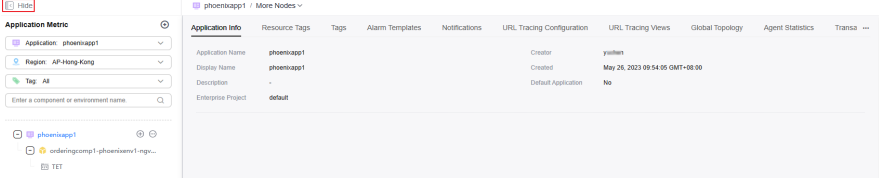
Figure 3-1 CMDB structure



The CMDB structure tree can be hidden.

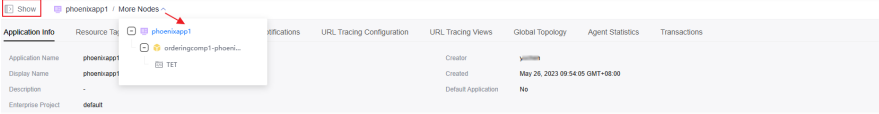
Step 1 Click **Hide** to hide the CMDB structure tree.

Figure 3-2 Hiding the CMDB structure tree



Step 2 Go to the path above in the upper part of the page and select your target node.

Figure 3-3 Selecting a node



Step 3 Click **Expand** to display the CMDB structure tree.

----End

3.2 Creating an Application

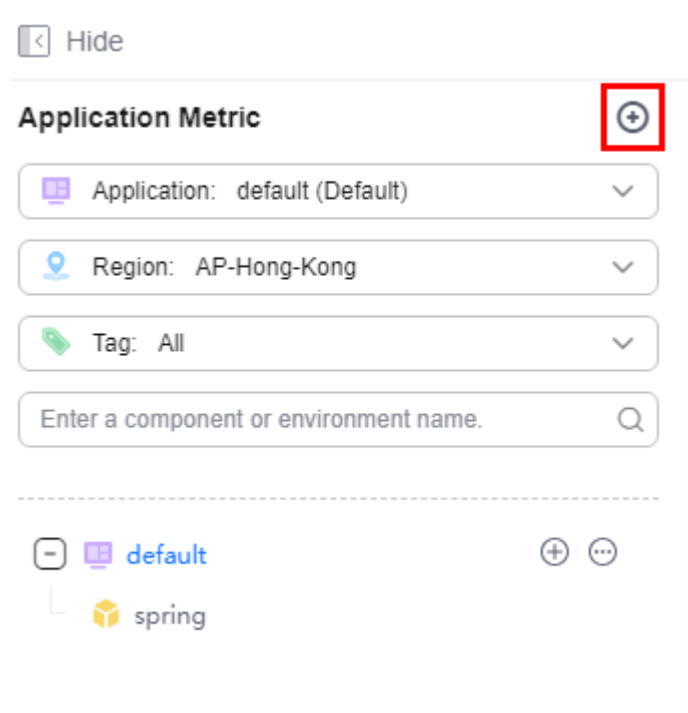
Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 Click  on the right of **Application Metric** to create an application.

Figure 3-4 Creating an application



Step 5 In the displayed dialog box, set application parameters.

Table 3-1 Parameters for creating an application

Parameter	Description
Application Name	Name of an application, which cannot be empty. Enter 1 to 128 characters and start with a letter. Only digits, letters, underscores (_), and hyphens (-) are allowed.
Display Name	Application alias. The alias takes precedence over the application name to display. Enter 1 to 128 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Enterprise Project	Select an enterprise project from the drop-down list. This parameter is displayed only when you use the enterprise edition.
Description	Description of the application. Enter up to 1,000 characters.

Step 6 Click **Confirm**.

After an application is created, connect it to APM for monitoring. For details, see [Getting Started](#).

----End

3.3 Creating a Sub-application

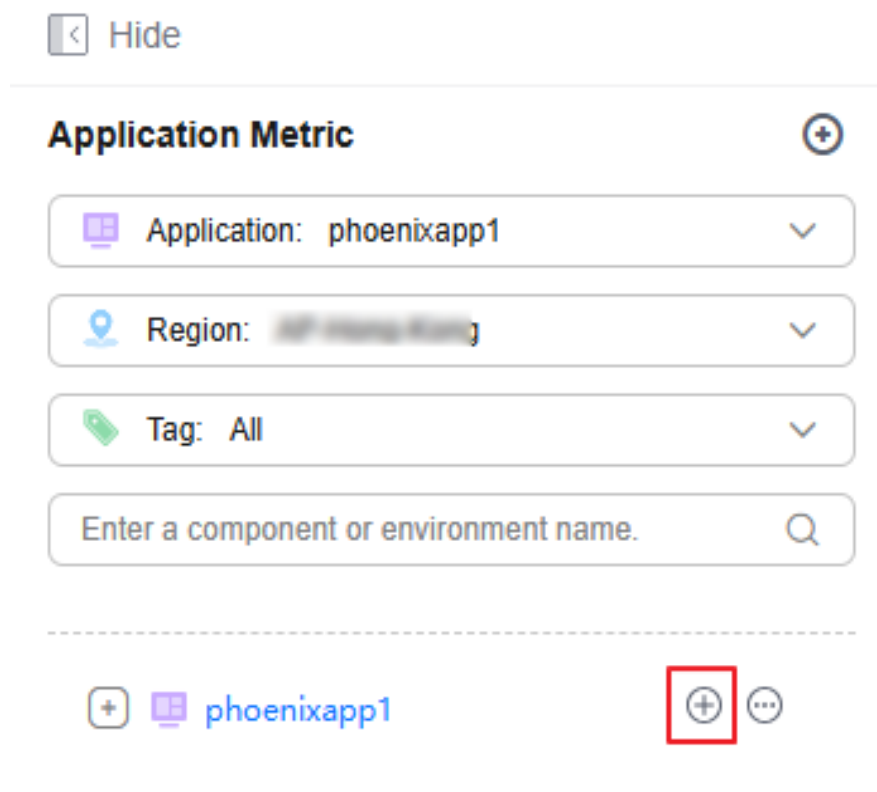
Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 Click  next to your target application in the tree.

Figure 3-5 Creating a sub-application



Step 5 In the displayed dialog box, set sub-application parameters.

Table 3-2 Parameters for creating a sub-application

Parameter	Description
Sub-application Name	Name of a sub-application, which cannot be empty. Enter 1 to 128 characters and start with a letter. Only digits, letters, underscores (_), and hyphens (-) are allowed.
Display Name	Display name of a sub-application, which cannot be empty. Enter 1 to 128 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Description	Description of the sub-application. Enter up to 1,000 characters.

Step 6 Click **Yes**. A maximum of three layers of sub-applications can be created.

----End

3.4 Configuring an Application and Sub-application

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 Click  next to the application or sub-application name in the tree.

Figure 3-6 Configuring an application and sub-application

< Hide

Application Metric +

Application: phoenixapp1 ▾

Region: AP-Hong Kong ▾

Tag: All ▾

Enter a component or environment name. 🔍

+ phoenixapp1 + ⋮

Step 5 Configure the application and sub-application according to [Table 3-3](#).

Table 3-3 Parameters for configuring the application and sub-application

Operation	Description
Modify	Click Modify . In the displayed dialog box, modify the information about the application or sub-application.
Set as Default	If you select Set as Default for an application, it will become the default application. When you log in to the system, the default application will be selected. This option is not available for sub-applications.
Delete	Click Delete .

Step 6 Click **Yes**.

----End

3.5 Checking the Basic Information About an Application, Sub-Application, Component, and Environment

Checking the Basic Information About an Application

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 Click an application in the tree on the left.

Step 5 In the right pane, click **Application Info** to check the basic information about the application.

Table 3-4 Basic information of the application

Parameter	Description
Application Name	Name of the application.
Display Name	Alias of the application.
Description	Description of the application. Enter up to 1,000 characters.
Enterprise Project	Enterprise project name. It is displayed only when you have enabled the enterprise edition.
Creator	User who creates the application.
Created	Time when the application is created.
Default Application	Whether the application is the default one. If it is a default application, (Default) will be displayed next to its name in the Application column in the tree on the left.

----End

Checking the Basic Information About a Sub-Application and Component

Step 1 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 2 Click a sub-application or component in the tree on the left. **Sub-application Info** or **Component Info** is displayed on the right.

Table 3-5 Basic information about the sub-application or component

Parameter	Description
Sub-Application/Component Name	Name of the sub-application or component.
Display Name	Alias of the sub-application or component.
Description	Description of the sub-application or component. Enter up to 1,000 characters.
Creator	User who creates the sub-application or component.
Created	Time when the sub-application or component is created.

----End

Environment and components are created when you connect an Agent. To monitor Java applications, see [Connecting Agents](#).

Checking the Basic Information About an Environment

- Step 1** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 2** In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.
- Step 3** Click the **Basic Info** tab to check the basic information about the environment.

Table 3-6 Basic information about the environment

Basic Attributes	Parameter	Description
Component	Component Name	Name of a component.
	Created	Time when the component is created.
	Creator	User who creates the component.
Environment	Environment Name	Name of an environment.
	Environment Type	Type of the environment.
	Tag Name	Environment tag. For details about how to manage tags, see Managing Tags .
	Region	Region where the environment is located.
	Creator	User who creates the environment.
	Created	Time when the environment is created.

Basic Attributes	Parameter	Description
	UUID	Component ID.

----End

4 Application Metric Monitoring

4.1 Introduction

APM Agents periodically collect performance metric data to measure the overall health status of applications. They can collect the metric data of JVM, GC, service calls, exceptions, external calls, database access, profiler performance analysis, debugging diagnosis, and middleware, helping you comprehensively monitor application running.

APM has strict definitions on metric data collection. Each type of data to be collected corresponds to a collector. For example, for JVM data of Java applications, a JVM collector is provided. A collector collects data of multiple metric sets. For details about collectors and metric sets, see [Collection Center](#).

After collectors are deployed in the environment, monitoring items are generated. During data collection, the monitoring items determine data structures and collection behaviors.

- **Collection period:** A monitoring item has the same period attribute as a data collector. The default data collection period is 1 minute and cannot be changed.
- **Monitoring item status:** A monitoring item is enabled by default. You can disable it so that an Agent does not intercept or report the metric data. For details, see [Enabling or Disabling a Monitoring Item](#).
- **Collection status:** Each collection instance or monitoring item has a collection status. If a collection error occurs, you can view it on the **Collection Status** tab page. A common error is that there are too many primary keys. As a result, data aggregation on the client is abnormal.

Monitoring Item Types

Agents automatically discover collection plug-ins and instantiate collectors to form monitoring items. Monitoring items are instantiated in an environment.

There are many types of collectors, which are hard to distinguish. The system backend groups collectors for easy data query.

Application monitoring enables you to monitor the overall health status of applications. Basic monitoring items include the topology, URL, JVM, and

exceptions. Other monitoring items, such as calls, communication protocols, databases, web containers, caches, and message queues, are automatically discovered and displayed.

The **Metrics** page displays only the involved monitoring item metrics of connected applications.

Based on collector functions, monitoring items can be classified into:

- **Topology**: displays the call relationships between services within a period. The statistics can be collected from the caller or the callee. You can also check the trend.
- **URL**: monitors the external services that call the current application.
- **JVM**: monitors basic system performance metrics.
- **Exception**: monitors application exceptions.
- **Call**: monitors the external services called by the current application.
- **SQL**: monitors database access.
- **Web Container**: monitors web containers such as Tomcat. Generally, the total number of threads, number of busy threads, and number of connections are collected to measure the overall system capacity.
- **Profiler Performance Analysis**: automatically obtains the during application running, and displays the calling relationship and execution efficiency of each method in real time through flame graphs, helping you optimize code performance.
- **Debugging Diagnosis**: You can analyze the thread status and obtain real-time method and object parameters. Developers can obtain real-time debugging information from remote applications without restarting the applications or causing service interruption.
- **RPC**: monitors remote procedure calls such as GRPCServer calls.

Monitoring Item Configuration

Collectors corresponding to monitoring items define collection parameters. You can modify collection parameters on the page as required. These parameters will be delivered to Agents with heartbeat parameters to change collection behaviors. By default, Redis instruction content is not collected for security purposes. If necessary, modify collection parameters to collect specific instruction data. Collection parameters can also be defined on environment tags. Collectors automatically inherit collection parameter attributes of corresponding environment tags. In this way, configuration is automated.

Monitoring Item Views

On the metric monitoring details page, a monitoring item corresponds to one or more tab views, and each view corresponds to a metric set. APM provides summary tables, trend graphs, latest data tables, and original tables. For details, see [Monitoring Item Views](#).

4.2 Application Monitoring Details

4.2.1 Topology

The topology displays the call relationships between services within a period. The statistics can be collected from the caller or the callee. You can also view the trend. On the topology, you can view the call relationships between services and check whether the calls between services are normal to quickly locate faults. The application relationships, call data (service and instance metrics), and health status are clearly displayed.

Viewing the Topology

Step 1 Log in to the [APM console](#).

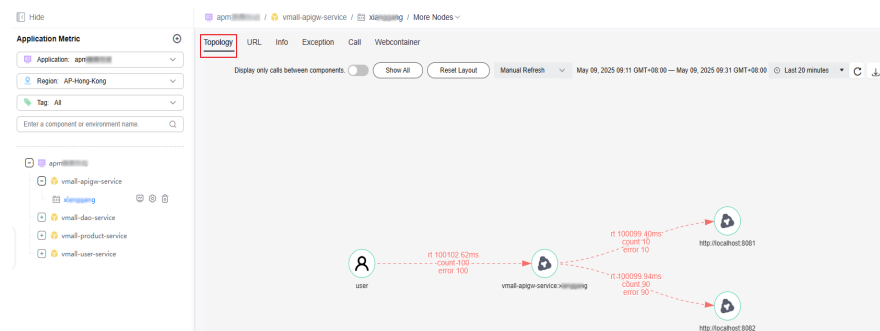
Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

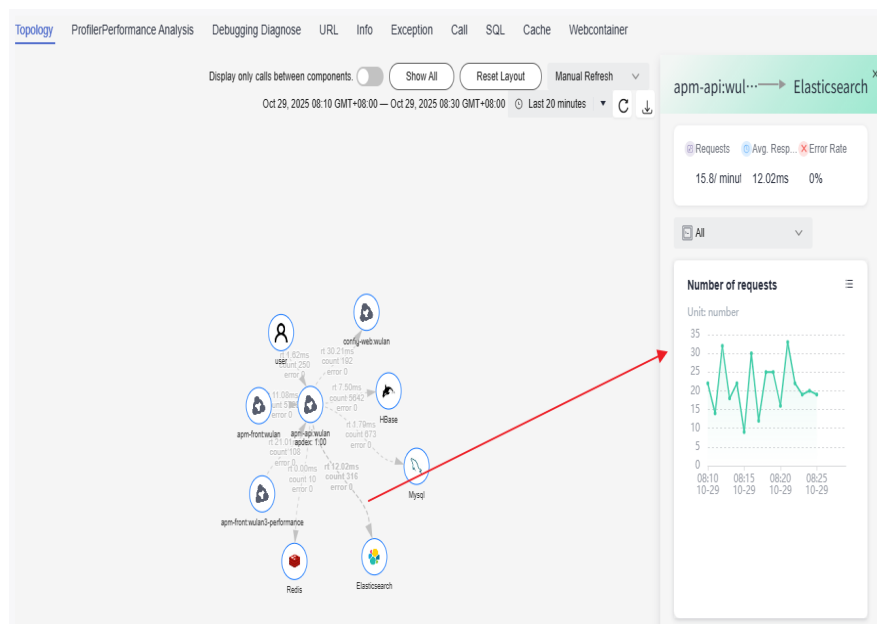
Step 5 Switch to the **Topology** tab page. The call trend of the selected instance is displayed.

Figure 4-1 Viewing the topology



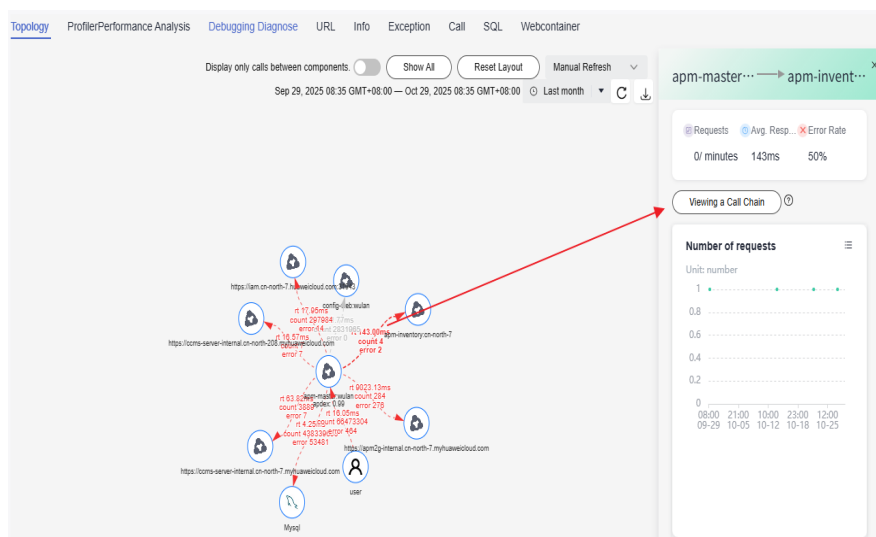
Step 6 Click a line in the diagram. The relationships between components or between components and instances are displayed.

Figure 4-2 Relationships between instances



- The diagram includes the number of requests, average RT, error rate, and corresponding line graphs.
 - Click the drop-down list in the upper left corner of the line graph and select an instance.
 - Click ≡ in the upper right corner of the line graph. All instance views corresponding to a parameter are displayed.
- The diagram shows the call relationships between components and enables you to check the traces between components.

Figure 4-3 Checking the traces between components



- The function of redirecting to the trace page is supported only by Agent 2.5.3 or later.

- Click **Viewing a Call Chain**. The **trace** page is displayed, showing the trace information between components. On the trace page, the value of **Custom Parameter** is the value of **lubanops-nenv-id** by default. For example, **lubanops-nenv-id=13**.

Step 7 Right-click a component and then click **Viewing a Call Chain** or **Viewing Indicators**.

Figure 4-4 Viewing the call chain or indicators of the component



- Viewing traces
Click **Viewing a Call Chain** to go to the trace page of the component. For details, see [Tracing](#).
- Viewing indicators
Click **Viewing Indicators** to go to the URL page of the component. For details, see [URL](#).

Step 8 Click  next to **Display only calls between components**.

When the button turns blue, only the calls between components are displayed.

Step 9 Click **Show All** to display all call relationships of the selected instance in a specified time range.

Step 10 Click **Reset Layout** to restore to the initial topology.

Step 11 Select the refresh mode and time. Default: **Manual Refresh**. In addition, **Automatic refresh in 1 minute**, **Automatic refresh in 5 minutes**, and **Automatic refresh in 15 minutes** are supported.

Step 12 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

- Click  to refresh the topology.
- Click  to download the current topology to the default path.

-----End

4.2.2 URL

This function monitors the calls of the current application by external services. It includes URL, Dubbo server, CSE server, CSEProvider cluster, and FunctionGraph monitoring. This type of monitoring item demonstrates the actual external status of the entire service. For example, if the average response time of a URL is long, it means that external users take a long time to query the corresponding data.

This section focuses on URL monitoring.

Setting the Maximum Number of Rows for Monitoring Metric Data Collection

When the number of collected metric data records of **URL**, **Call**, or **SQL** exceeds the default or configured value, a threshold-crossing alarm will be displayed.

You can configure **Maximum number of rows for monitoring metric data collection** on the **URL**, **Call**, and **SQL** tab pages. Take the **URL** page as an example. The procedure is as follows:

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment. The **URL** tab page is displayed by default.

Step 5 Click **Config** in the alarm and change the value of **Maximum number of rows for monitoring metric data collection**.

Step 6 Click **Confirm**.

- Only when the number of metric data records exceeds the default or configured value will a threshold-crossing alarm be displayed.
- If the number of collected metric data records does not exceed the default or configured value, the value of **Maximum number of rows for monitoring metric data collection** cannot be configured.
- For **Maximum number of rows for monitoring metric data collection**, the default value is **500**. You can enter 10 to 2000.
- If the number of collected metric data records exceeds the configured value, the system will stop metric collection to prevent excessive memory usage. The settings may take up to 5 minutes to take effect.

----End

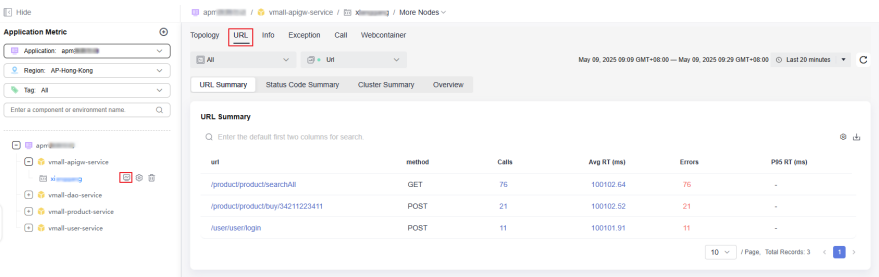
Going to the URL Page

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

- Step 3
- In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4
- In the tree on the left, click  next to the target environment. On the **URL** tab page that is displayed, check URL monitoring information of all instances.

Figure 4-5 Going to the URL page



- Step 5
- On the displayed **URL** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.
- Step 6
- Select a time range. Default: **Last 20 minutes**.
Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.
- Step 7
- Click  in the upper right corner of the list and select the target metric data.
- Step 8
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

-----End

Viewing URL Monitoring Data

URL summary

For common URL calls, the system collects the metrics of each URL. For details about the metrics, see [Table 4-1](#).

Figure 4-6 URL summary under URL monitoring

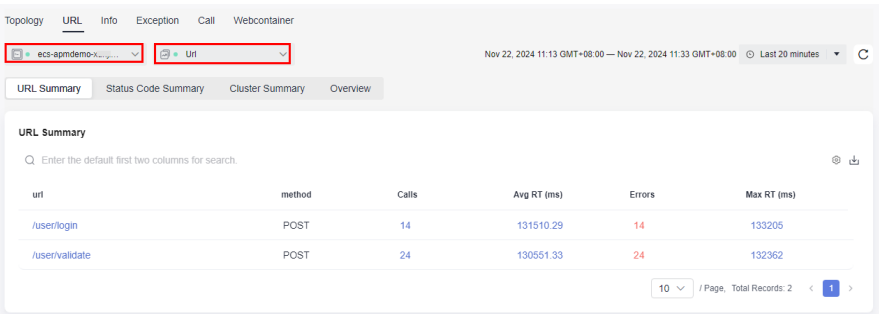


Table 4-1 Parameters of the URL summary

Metric Set	Metric	Description
URL summary	url	URL.
	method	Request HTTP method.
	Calls	Number of times that the URL is called.
	Avg RT (ms)	Average response time of the URL in a collection period.
	Errors	Number of call errors of the URL.
	Max Concurrency	Maximum concurrency of the URL.
	Max RT (ms)	Maximum response time of the URL in a collection period.
	Apdex	Application performance index (Apdex), which indicates users' satisfaction. The value ranges from 0 to 1. The closer the value is to 1, the higher the satisfaction is. For details, see Basic Concepts - Apdex .
	Exceptions	Number of exceptions of the URL.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

- URL invocation is the starting point of tracing. When you click a URL, the tracing page is displayed, showing the URL invocation condition in a certain period (default: 20 minutes).
- You can add a URL for tracing by referring to [Configuring URL Tracing](#).
- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

Status code summary

APM supports status code-based summary. The system collects the metrics of each URL. For details about the metrics, see [Table 4-2](#).

Table 4-2 Parameters of status code summary

Metric Set	Metric	Description
Status code summary	code	Status code.
	Count	Number of times that the status code occurred.
	Latest URL	Sample URL which returns the status code in a collection period.

- Click a status code in the **code** column. The tracing page is displayed, showing the invocation condition of the status code of the selected instance in the environment in last 20 minutes (default).
- Click a number in the **Count** column to view the trend of the status code in a specified period.
- Click the latest URL to view the invocation details of the corresponding status code.

Cluster summary

APM can summarize metrics by cluster. For details about the metrics, see [Table 4-3](#).

Table 4-3 Parameters of the cluster summary

Metric Set	Parameter	Description
Cluster summary	Cluster ID	Cluster ID of the caller.
	Calls	Number of times the cluster is called.
	Avg RT (ms)	Average response time of the cluster in a collection period.
	Errors	Number of times that the cluster fails to be called.
	Max Concurrency	Maximum concurrency of the cluster.
	Max RT (ms)	Slowest call time of the cluster in a collection period.

Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

Overview

View the metric trend of the selected instance on the **Overview** tab page. For details about the metrics, see [Table 4-4](#). When you hover over the graph, you can check the total requests, average RT (ms), errors, and Apdex in a specific time.

Table 4-4 Overview metrics

Metric Set	Metric	Description
Overview	Total Requests	Total number of URL requests.
	Avg RT (ms)	Average response time of the URL.
	Errors	Total number of URL errors.
	Apdex	Users' satisfaction with the URL.

4.2.3 JVM

This function monitors JVMInfo, JVMMonitor, GC, thread, and JavaMethod.

Going to the JVM Page

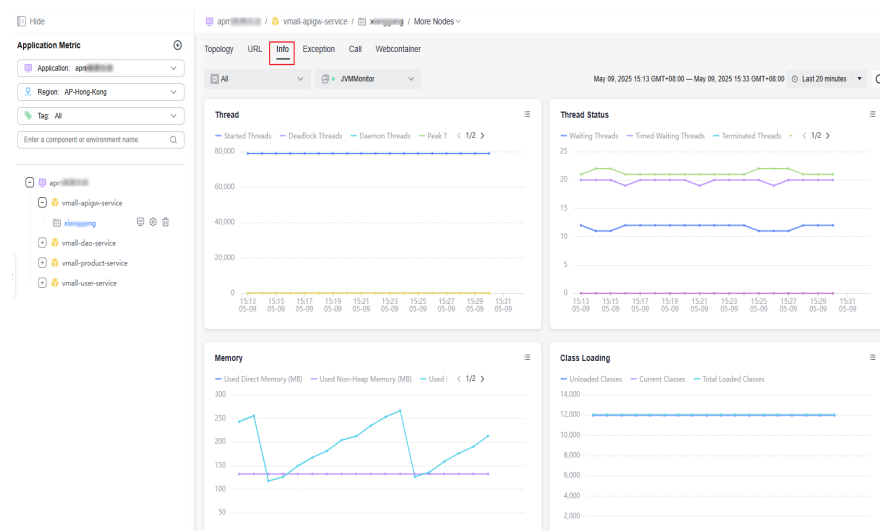
Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **JVM** tab. By default, the JVMMonitor information of all instances is displayed. When you hover over a graph, you can check the actual value of each metric in a specific time.

Figure 4-7 Going to the JVM page

Step 6 On the displayed **JVM** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

----End

Viewing JVM Information

On the **JVM** tab page, view the JVMInfo metrics of the corresponding instance. For details about the metrics, see [Table 4-5](#).

Figure 4-8 Viewing JVM information

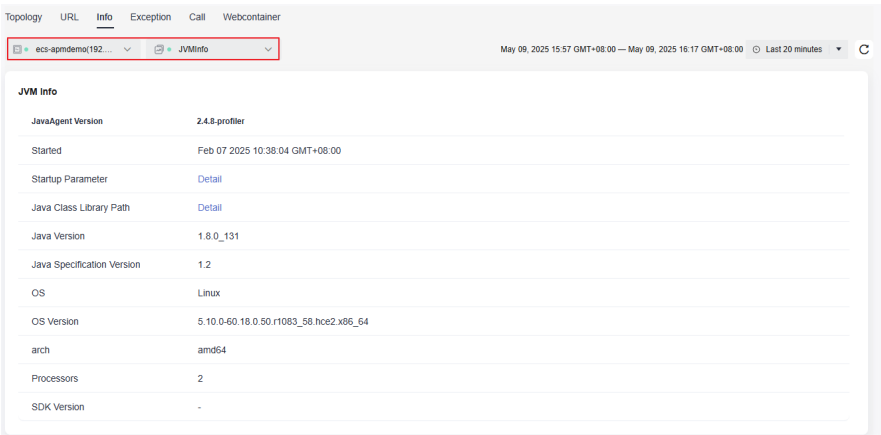


Table 4-5 JVMInfo metrics

Metric Set	Metric	Description
JVMInfo	JavaAgent Version	Java Agent version.
	Started	JVM startup time.
	Startup Parameter	JVM startup parameter. Click Detail to view JVM startup parameter details.
	Java Class Library Path	Java class library path. Click Detail to view Java class library path details.
	Java Version	Java version.
	Java Specification Version	Java specification version.
	OS	OS name.
	OS Version	OS version.

Metric Set	Metric	Description
	arch	CPU architecture.
	Processors	Number of processors.
	SDK Version	SDK version.

Viewing JVM Monitoring Data

APM monitors JVM metrics. For details about the metrics, see [Table 4-6](#). JVM monitoring metrics are displayed in graphs, so that you can view and analyze JVM monitoring data more easily.

- When you hover over a graph, you can check the actual value of each metric in a specific time.
- Click  in the upper right corner of the **Memory Pool** list and select the target metric data.
- Click  in the upper right corner of the **Memory Pool** list to export information. A maximum of 100 records can be exported.

Table 4-6 JVM monitoring metrics

Metric Set	Metric	Description
Thread	Current Threads	Number of current threads.
	Deadlock Threads	Number of deadlock threads.
	Daemon Threads	Number of daemon threads.
	Started Threads	Number of started threads.
	Peak Threads	Peak number of threads.
Thread Status	Waiting Threads	Number of waiting threads.
	Terminated Threads	Number of threads in the terminated state.
	Runnable Threads	Number of runnable threads.
	Blocked Threads	Number of blocked threads.
	New Threads	Number of new threads.
	Timed Waiting Threads	Number of threads that timed out.
Memory	Used Non-Heap Memory	Size of the used non-heap memory.

Metric Set	Metric	Description
	Used Heap Memory	Size of the used heap memory.
	Used Direct Memory	Size of the used direct memory.
Class loading	Current Classes	Number of current classes.
	Total Loaded Classes	Total number of loaded classes.
	Unloaded Classes	Number of unloaded classes.
Memory pool	committed(M)	Size of available memory.
	init(M)	Size of the initialized memory.
	max(M)	Size of the maximum memory.
	name	Memory pool name.
	used(M)	Size of the used memory.
CPU	CPU Usage	CPU usage of the Java process.

Viewing GC Information

APM monitors GC metrics. For details about the metrics, see [Table 4-6](#).

Table 4-7 GC metrics

Metric Set	Metric	Description
GC statistics	Full GC (times)	Number of full GC times in a collection period.
	Full GC Duration (ms)	Full GC duration in a collection period.
	Young GC (times)	Number of young GC times in a collection period.
	Young GC Duration (ms)	Young GC duration in a collection period.
GC Details	GC Type	GC type, which can be major or minor .
	GC Cause	GC cause.
	Count	Number of times that GC has occurred.
	Total GC Duration (ms)	GC duration.
	Max GC Duration (ms)	Time consumed by the slowest GC.

Metric Set	Metric	Description
	GC Recycler	GC recycler name.
	Slowest GC Details	Details about the slowest GC.

- Click the numbers in blue (such as those in the **Count**, **Total GC Duration (ms)**, or **Max GC Duration (ms)** column) to view the corresponding GC trend graph in a certain period (default: 20 minutes).
- When you hover over a graph, you can check the actual value of each metric in a specific time.
- On the GC details area, you can view the GC type, GC cause, count, total GC duration (ms), maximum GC duration (ms), GC recycler, and slowest GC details (details and history).
- Click  in the upper right corner of the **GC Details** list and select the target metric data.
- Click  in the upper right corner of the **GC Details** list to export information. A maximum of 100 records can be exported.

Viewing Threads

You can view the thread details of the corresponding instance on APM. For details, see [Table 4-8](#).

Table 4-8 Thread metrics

Metric Set	Metric	Description
Thread details	Thread Name	Thread name.
	Threads	Number of threads.
	CPU Time (ms)	Thread CPU time.
	Memory (MB)	Memory size in MB.
	Thread Stack	Thread stack.

- Click a number in the **Threads** column to view the trend of the thread in a specified period.
- Click **Detail** in the **Thread Stack** column to view the thread details.
- Click **History** in the **Thread Stack** column to view the historical thread stack list.
- Click  in the upper right corner of the **Thread Details** list and select the target metric data.

- Click  in the upper right corner of the **Thread Details** list to export information. A maximum of 100 records can be exported.

Viewing Java Methods

- By default, APM does not monitor Java methods. To monitor them, [configure the JavaMethod monitoring item](#) first.
- After the configuration is complete, the system monitors the methods and classes of JavaMethod.
- On the **JVM** page, select a target instance and **JavaMethod** to view details. For details, see [Table 4-9](#).

Table 4-9 JavaMethod metrics

Metric Set	Metric	Description
JavaMethod	Class	Class of a Java method.
	Method	Method.
	Calls	Number of times that the method is called.
	Avg RT (ms)	Average response time.
	Errors	Number of times that the method fails to be called.
	Max Concurrency	Maximum concurrency of the method.
	Max RT (ms)	Maximum response time of the method.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

- Click a number (such as those in the **Calls** or **Errors** column) to view the trend of the method in a specified period.
- Click  in the upper right corner of the **Method** list and select the target metric data.

- Click  in the upper right corner of the **Method** list to export information. A maximum of 100 records can be exported.

4.2.4 Exception

This function monitors application exception logs. Take the monitoring of Java exception logs as an example. Once you use the log system to print logs, they will be collected by APM. The exception collection type varies according to the collector type.

Viewing Exception Logs

- Step 1
- Log in to the [APM console](#).
- Step 2
- Click  on the left and choose **Application > Application Performance Management**.
- Step 3
- In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4
- In the tree on the left, click  next to the target environment.
- Step 5
- Click the **Exception** tab. By default, exception logs of all instances are displayed. For details about the metrics, see [Table 4-10](#).

Figure 4-9 Exception monitoring data

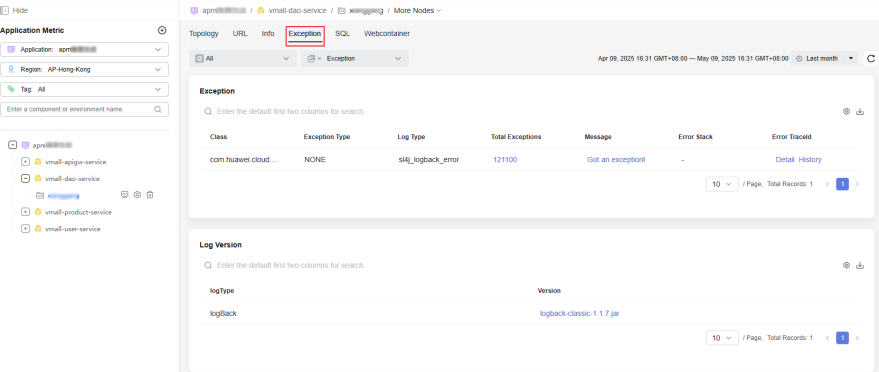


Table 4-10 Exception and log parameters

Metric Set	Parameter	Description
Exception	Class	Exception class
	Exception Type	Exception type
	Log Type	Exception log type
	Total Exceptions	Number of times that an exception has occurred
	Message	Message returned when the exception occurred

Metric Set	Parameter	Description
	Error Stack	Error stack
	Error Trace ID	Information about the error trace
Log Version	Log Type	Log type
	Version	Version

- Click a number in blue in the **Total Exceptions** column to view the trend of the total exceptions in a specified period.
- Click the blue text in the **Message** column to view the message time and content.
- Click **Detail** in the **Error Stack** column to view exception details.
- Click **History** in the **Error Stack** column to view the historical error stack list.
- Click **Detail** in the **Trace** column to view trace details.
- Click **History** in the **Trace** column to view the historical trace list.
- Click the blue text in the **Version** column to view details.

Step 6 On the **Exception** tab page, select a target instance and then select **Exception** to view the exception monitoring data.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Step 8 Click  in the upper right corner of the list and select the target metric data.

Step 9 Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

----End

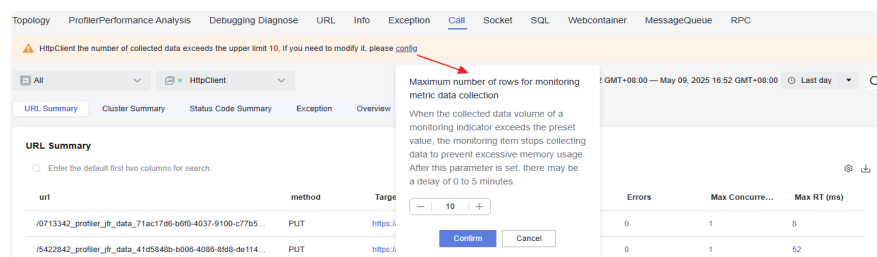
4.2.5 Call

This function monitors the calls of external services by the current application. It covers CSEConsumer, ApacheHttpClient, ApacheHttpAsyncClient, DubboConsumer, and HttpClient monitoring.

This section focuses on HttpClient monitoring.

Setting the Maximum Number of Rows for Monitoring Metric Data Collection

When the number of collected metric data records of **URL**, **Call**, or **SQL** exceeds the default or configured value, a threshold-crossing alarm will be displayed.

Figure 4-10 HttpClient monitoring metric collection

You can configure **Maximum number of rows for monitoring metric data collection** on the **URL**, **Call**, and **SQL** tab pages. Take the **Call** page as an example. The procedure is as follows:

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment. The **URL** tab page is displayed by default.

Step 5 Click the **Call** tab.

Step 6 Click **Config** in the alarm and change the value of **Maximum number of rows for monitoring metric data collection**.

Step 7 Click **Confirm**.

- Only when the number of metric data records exceeds the default or configured value will a threshold-crossing alarm be displayed.
- If the number of collected metric data records does not exceed the default or configured value, the value of **Maximum number of rows for monitoring metric data collection** cannot be configured.
- For **Maximum number of rows for monitoring metric data collection**, the default value is **500**. You can enter 10 to 2000.
- If the number of collected metric data records exceeds the configured value, the system will stop metric collection to prevent excessive memory usage. The settings may take up to 5 minutes to take effect.

----End

Going to the Call Page

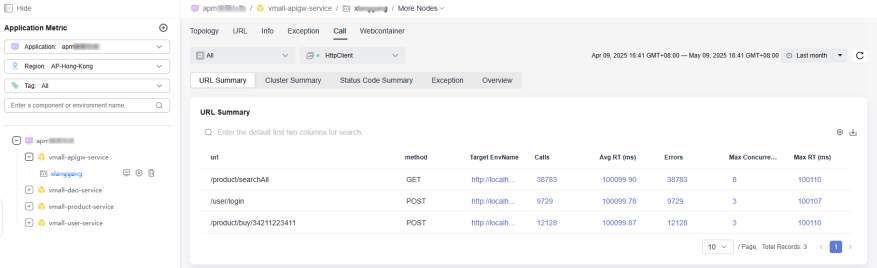
Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

- Step 4
- In the tree on the left, click  next to the target environment.
- Step 5
- Click the **Call** tab. By default, the HttpClient monitoring information of all instances is displayed.

Figure 4-11 External call data



- Step 6
- On the displayed **Call** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.
- Step 7
- Select a time range. Default: **Last 20 minutes**.
Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.
- Step 8
- Click  in the upper right corner of the list and select the target metric data.
- Step 9
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.
- End

Viewing HttpClient Monitoring Data

URL summary

The HttpClient monitoring system collects the metrics of each URL. For details about the metrics, see [Table 4-11](#). Click  in the upper right corner of the list and select the target metric data.

Table 4-11 Parameters of URL summary under HttpClient monitoring

Metric Set	Metric	Description
URL summary	url	Called URL.
	method	HTTP method of the URL.
	TargetEnvName	Name of the component whose URL is called.
	Calls	Number of times that the URL is called.
	Avg RT (ms)	Average response time of the called URL.
	Errors	Number of call errors of the URL.

Metric Set	Metric	Description
	Max Concurrency	Maximum concurrency of the URL.
	Max RT (ms)	Maximum response time of the called URL.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.
	Error Trace	ID of the trace that encounters an error in a collection period.
	Slowest Trace	ID of the slowest trace in a collection period.

- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.
- Click the address in the **TargetEnvName** column. The details about the called component are displayed in a list.
- Click the text in blue (such as those in the **Slowest Trace** or **Error Trace** column) to view more details.

Cluster summary

APM can summarize external call metrics by cluster. For details about the metrics, see [Table 4-12](#).

Table 4-12 Parameters of cluster summary under HttpClient monitoring

Metric Set	Metric	Description
Cluster summary	envId	Cluster ID of the called party.
	hostUri	URL of the called party.
	Calls	Number of times that the cluster URL is called.

Metric Set	Metric	Description
	Avg RT (ms)	Average response time for calling the cluster URL.
	Errors	Number of call errors of the URL.
	Max RT (ms)	Maximum response time for calling the cluster URL.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.
- Click the address in the **hostUri** column. The details of the called party are displayed in a list.

Status code summary

APM can summarize external call metrics by status code. For details about the metrics, see [Table 4-13](#).

Table 4-13 Parameters of status code summary under HttpClient monitoring

Metric Set	Metric	Description
Status code summary	code	Status code.
	Count	Number of times that the status code occurred.
	Latest URL	URL that returns the status code.

- Click a status code in the **code** column. The tracing page is displayed, showing the invocation condition of the status code of the selected instance in the environment in last 20 minutes (default).

- Click a number in the **Count** column to view the trend of the status code in a specified period.
- Click the latest URL to view the invocation details of the corresponding status code.

Exception

On the **Exception** tab page, view the exception statistics about HttpClient calls. For details about the metrics, see [Table 4-14](#).

Table 4-14 Parameters of HttpClient monitoring exceptions

Metric Set	Metric	Description
Exception	causeType	Exception class.
	exceptionType	Exception type.
	Count	Number of times the exception occurred.
	Error Message	Message returned when the exception occurred.
	Error Stack	Exception stack information.

- Click a number in blue in the **Count** column to view the trend of the thread in a specified period.
- Click the text in blue in the **Error Message** column to view message details.
- Click **Detail** in the **Error Stack** column to view exception details.
- Click **History** in the **Error Stack** column to view the historical error stack list.

Overview

On the **Overview** tab page, view the metrics of the selected instance. For details about the metrics, see [Table 4-15](#).

When you hover over a graph, you can check the actual value of each metric in a specific time.

Table 4-15 Overview parameters of HttpClient monitoring

Metric Set	Metric	Description
Overview	Calls	Total number of calls.
	Avg RT (ms)	Average response time
	Errors	Total number of errors.

4.2.6 SQL

This function monitors database access. The databases that can be monitored include the C3P0, Cassandra, ClickHouse, DBCP, Druid, EsRestClient, GaussDB,

Hikari, Jetcd, ObsClient, MySQL, PostgreSQL, Oracle, HBase, and MongoDB. APM collects details about executed statements to help you locate performance problems in code.

This section focuses on MySQL database monitoring.

Setting the Maximum Number of Rows for Monitoring Metric Data Collection

When the number of collected metric data records of **URL**, **Call**, or **SQL** exceeds the default or configured value, a threshold-crossing alarm will be displayed.

Step 1 Click **Config** in the alarm and change the value of **Maximum number of rows for monitoring metric data collection**.

Step 2 Click **Confirm**.

- Only when the number of metric data records exceeds the default or configured value will a threshold-crossing alarm be displayed.
- If the number of collected metric data records does not exceed the default or configured value, the value of **Maximum number of rows for monitoring metric data collection** cannot be configured.
- For **Maximum number of rows for monitoring metric data collection**, the default value is **500**. You can enter 10 to 2000.
- If the number of collected metric data records exceeds the configured value, the system will stop metric collection to prevent excessive memory usage. The settings may take up to 5 minutes to take effect.

----End

Going to the SQL Page

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **SQL** tab. By default, the MySQL database information of all instances is displayed.

Step 6 On the displayed **SQL** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Step 8 Click  in the upper right corner of the list and select the target metric data.

----End

Viewing MySQL Database Monitoring Data

SQL summary

APM can monitor MySQL databases by SQL. For details about the metrics, see

Table 4-16. Click  in the upper right corner of the list and select the target metric data.

Table 4-16 SQL summary parameters

Metric Set	Metric	Description
SQL monitoring	sql	Unique ID of the SQL statement, which is used for alarm configuration.
	SQL Statement	SQL statement.
	Calls	Number of times that the SQL statement is called.
	Avg RT (ms)	Average response time (ms).
	Errors	Number of errors that the SQL statement encounters.
	Rows Read	Number of read rows of the SQL statement.
	Rows Updated	Number of updated rows of the SQL statement.
	Max Concurrency	Maximum concurrency of the SQL statement.
	Max RT (ms)	Maximum response time of the SQL statement.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–200 ms	Number of requests with 100 ms–200 ms response time.
	200 ms–1s	Number of requests with 200 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

Metric Set	Metric	Description
	Slowest Trace	ID of the slowest trace in a collection period.
	Error Trace	ID of the trace that encounters an error in a collection period.

- Click an SQL statement to view details.
- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.
- Click a slow or an error trace to view its details.

Database summary

APM can summarize MySQL database metrics by database. For details about the metrics, see [Table 4-17](#).

Table 4-17 Database summary parameters

Metric Set	Metric	Description
Database connections	db	Database name.
	Connections Created	Number of connections created by the database.
	Connections Destroyed	Number of the database's connections that have been destroyed.
	Avg RT (ms)	Average response time (ms).
	Calls	Number of times that the database is called.
	Errors	Number of errors that the database encounters.
	Rows Read	Number of rows read from the database.
	Rows Updated	Number of rows updated in the database.
	Max RT (ms)	Maximum response time of the database.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.

Metric Set	Metric	Description
	100 ms–200 ms	Number of requests with 100 ms–200 ms response time.
	200 ms–1s	Number of requests with 200 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

Exception

On the **Exception** tab page, view exception statistics about SQL calls. For details about the metrics, see [Table 4-18](#).

Table 4-18 Exception parameters

Metric Set	Metric	Description
Exception	causeType	Exception class.
	exceptionType	Exception type.
	Count	Number of exceptions.
	SQL	SQL statement that encounters an exception.
	Error Stack	Exception stack information.
	Message	Error message.

Overview

On the **Overview** tab page, view the call trend of the selected instance. For details about the metrics, see [Table 4-19](#). When you hover over a graph, you can check the actual value of each metric in a specific time.

Table 4-19 Overview parameters

Metric Set	Metric	Description
Overview	Calls	Total number of calls.
	Rows Read	Total number of read rows.

Metric Set	Metric	Description
	Avg RT (ms)	Average response time (ms).
	Errors	Total number of errors.
	Rows Updated	Number of rows updated in the database.

Info

On the **Info** tab page, view the driver version information. Click the text in blue to view more details.

Viewing Druid Connection Pool Monitoring Data

The Druid connection pool monitoring system collects data sources, connection

details, additional configurations, and exception information. You can click  in the upper right corner of the list to customize the columns you want to view. For details about the metrics, see [Table 4-20](#).

Table 4-20 Druid connection pool parameters

Metric Set	Metric	Description
Data source	Connection Address	Connection address.
	Driver	Driver name.
	Initialized Connections	Number of initialized connections.
	Min Idle Connections in Pool	Minimum of idle connections in a pool.
	Max Idle Connections in Pool	Maximum number of idle connections in a pool.
	Max Connections in Pool	Maximum number of connections in a pool.
	Idle Connections	Number of idle connections.
	Max Idle Connections	Maximum number of idle connections.
	Active Connections	Number of active connections.
	Max Active Connections	Maximum number of active connections.
	Waiting Threads	Number of waiting threads.
	Max Waiting Threads	Maximum number of waiting threads.

Metric Set	Metric	Description
	Upper Limit for Waiting Threads	Upper limit for waiting threads.
	Total Connections	Total number of connections.
Connection details	Connection Address	Connection address.
	Calls	Number of calls.
	Total RT (ms)	Total response time (ms).
	Avg RT (ms)	Average response time (ms).
	Errors	Number of errors.
	Max Concurrency	Maximum number of concurrent connections.
	Max RT (ms)	Maximum response time.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.
Additional configuration	Connection Address	Connection address.
	Max Wait (ms)	Maximum waiting time.
	Test on Borrow	Whether to verify the validity of a connection before obtaining it from the connection pool.
	Test on Return	Whether to verify the validity of a connection when it is returned.
	Test While Idle	Whether to verify the validity of an idle connection when an application applies for it from the pool
	Remove Abandoned	Whether to automatically reclaim timeout connections.

Metric Set	Metric	Description
	Remove Abandoned TimeoutMillis (ms)	If a connection in the pool is not returned within the specified duration, the connection will be reclaimed.
	Remove Abandoned Count	Number of timeout connection reclaims.
	Min Evictable Idle TimeMillis (ms)	Minimum idle time of a connection in the pool.
	Time Between EvictionRunsMillis (ms)	Interval for checking the validity of idle connections.
Exception	causeType	Exception class.
	Exception Type	Exception type.
	Count	Number of times the exception occurred.
	Error Message	Message returned when the exception occurred.
	Error Stack	Exception stack information.
Version	Driver Version	Driver version.

- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.
- Click the text in blue (such as those in the **Driver** or **Driver Version** column) to view more details.

Viewing Hikari Connection Pool Monitoring Data

The Hikari connection pool monitoring system collects the pool details, connection details, and exception information. You can click  in the upper right corner of the list to customize the columns you want to view. For details about the metrics, see [Table 4-21](#).

Table 4-21 Hikari connection pool parameters

Metric Set	Metric	Description
Connection pool details	Pool Address	Connection address.
	Driver Name	Driver name.
	Max Pool Size	Maximum number of connections that are allowed.
	Total Connections	Total number of current connections.

Metric Set	Metric	Description
	Active Connections	Number of active connections.
	Idle Connections	Number of idle connections.
	Threads Awaiting Connection	Number of waiting connections.
Connection details	Connection Address	Connection address.
	Calls	Number of calls.
	Total RT (ms)	Total response time (ms).
	Avg RT (ms)	Average response time (ms).
	Errors	Number of errors.
	Max Concurrency	Maximum number of concurrent connections.
	Max RT (ms)	Maximum response time.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.
Exception	causeType	Exception class.
	Exception Type	Exception type.
	Count	Number of times the exception occurred.
	Error Message	Message returned when the exception occurred.
	Error Stack	Exception stack information.

- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

4.2.7 Web Container

This function monitors web containers, including Tomcat. This section focuses on Tomcat monitoring.

Going to the Web Container Page

- Step 1
- Log in to the [APM console](#).
- Step 2
- Click  on the left and choose **Application > Application Performance Management**.
- Step 3
- In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4
- In the tree on the left, click  next to the target environment.
- Step 5
- Click the **Web Container** tab. By default, the Tomcat monitoring information of all instances is displayed. For details about the metrics, see [Table 4-22](#).

Figure 4-12 Going to the web container page

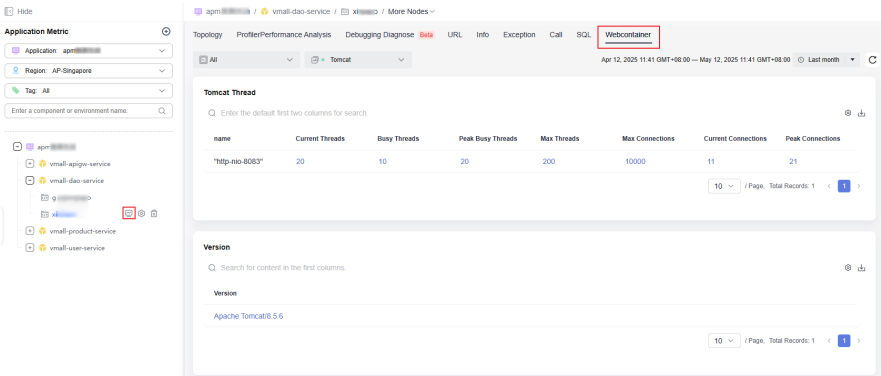


Table 4-22 Tomcat monitoring parameters

Metric Set	Metric	Description
Tomcat port monitoring	name	Port name.
	Current Threads	Number of current threads on the port.
	Busy Threads	Number of busy threads on the port at the time of collection.
	Peak Busy Threads	Maximum number of busy threads on the port in a collection period.
	Max Threads	Maximum number of threads on the port.
	Max Connections	Maximum number of connections on the port.

Metric Set	Metric	Description
	Current Connections	Number of current connections of the port at the time of collection.
	Peak Connections	Maximum number of connections on the port in a collection period.
Version	Version	Tomcat version.

- Click a number in blue (such as those in the **Current Threads**, **Busy Threads**, or **Peak Busy Threads** column) to view the trend graph of the target web container in a specified period.
- Click a version in the **Version** column to view details.

Step 6 On the displayed **Web Container** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Step 8 Click  in the upper right corner of the list and select the target metric data.

Step 9 Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

----End

4.2.8 Cache

This function monitors caches, including Redis, Jedis, and Lettuce clients.

Going to the Cache Page

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Cache** tab.

Step 6 On the displayed **Cache** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

----End

Viewing Jedis Metrics

Table 4-23 Jedis metrics

Metric Set	Metric	Description
Connection Pool [Details]	pool	Unique identifier of a connection pool (host name + port number)
	active	Number of active connections
	waiters	Number of waiting connections
	idle	Number of idle connections
	created	Number of connections that have been created
	destroyed	Number of connections that have been destroyed
	borrowed	Number of borrowed connections.
	total	Maximum number of connections
Switchovers [Summary]	From [Latest]	Source host
	To [Latest]	Destination host
	Total Switchovers	Total number of switchovers between two hosts
Switchovers	-	Click  to display all instance views.
Connection Pool Configuration	maxTotal	Maximum number of connections
	maxIdle	Maximum number of idle connections
	maxWaitMillis	Maximum waiting time (ms)
	maxBorrowWaitTimeMillis	Maximum waiting time of borrowed connections (ms)
	meanActiveTimeMillis	Average activation time of connections (ms)
	meanBorrowWaitTimeMillis	Average waiting time of borrowed connections (ms)
Client Info	Version	Client version
	Running Mode	Redis running mode. Options: standalone and cluster .
	Redis Node	Number of master Redis nodes

4.2.9 Profiler Performance Analysis

4.2.9.1 Overview

Constraints

- Profiler performance analysis is not generally available.
- Alpine Linux does not support Profiler. Alpine Linux is a lightweight Linux distribution and does not have the dependencies for running Profiler. To use Profiler, you are advised to use the standard Linux OS.
- To ensure stable application running, you are advised to upgrade the JDK version as required. If you use Profiler on the JDK of an earlier version, the application may break down.

Table 4-24 JDK version description

JDK Type	Version
OpenJDK	• OpenJDK 8u352+ • OpenJDK 17.0.5+
Oracle JDK	• Oracle JDK 17.0.9+

Table 4-25 JavaAgent updates

Agent Version	Description
2.4.17-profiler 2.4.17-profiler-jdk17	Updates of Agent 2.4.17-profiler/2.4.17-profiler-jdk17 and later versions: 1. Supported trace generation based on the OpenTelemetry protocol. 2. Fixed the issue that the gateway service returns multiple trace IDs. 3. Fixed the issue that HttpClient collection is incomplete. 4. Fixed the issue that the regular expression for external calls is invalid.
2.4.16-profiler 2.4.16-profiler-jdk17	Updates of Agent 2.4.16-profiler/2.4.16-profiler-jdk17 and later versions: 1. (SQL collector) Supported the setting and collection of slow SQL queries and database names. 2. (URL collector) Supported exception counting by URL and exception type.

Table 4-26 Constraints

N o.	Constraint
1	Profiler is not generally available. A maximum of 500 Profilers can be enabled in each environment.
2	The Profiler guide page is displayed only when you enable Profiler for the first time.
3	Profiler supports only Java.
4	If the value of -XX:MaxJavaStackTraceDepth is 0 or smaller, no Java stack will be collected.
5	• CPU Profiler depends on the perf_event_open system call. However, due to the Syscall security policy (seccomp) control of the Linux kernel, processes may be prohibited from calling specific Syscalls. • Docker environment: Run the following command to run the container. For details about refined system call control, see https://docs.docker.com/engine/security/seccomp/. <pre>docker run --security-opt seccomp=unconfined XXX</pre> • Kubernetes environment: Set privileged to true for a privileged container. The privileged container is always in the Unconfined state. For more refined system call control, see https://kubernetes.io/docs/tutorials/security/seccomp/.
6	Profiler allocates 8 KB perf_event buffer to each thread of the target process. However, the Linux kernel limits the perf_event related MMap memory. For a non-privileged user, run the following command to ensure that the value of /proc/sys/kernel/perf_event_mlock_kb is large enough: echo 8*threads+ 4 > /proc/sys/kernel/perf_event_mlock_kb

Impact on Performance

According to performance tests, when all diagnosis functions are enabled for Spring Web applications, the CPU overhead increases by about 5%, off-heap memory overhead increases by about 50 MB, and GC and request latency do not increase significantly.

4.2.9.2 Profiler

Enabling Profiler

Step 1 Log in to the [APM console](#).

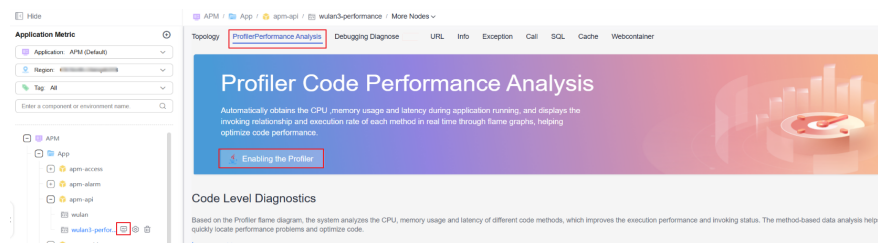
Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Profiler Performance Analysis** tab.

Figure 4-13 Profiler performance analysis

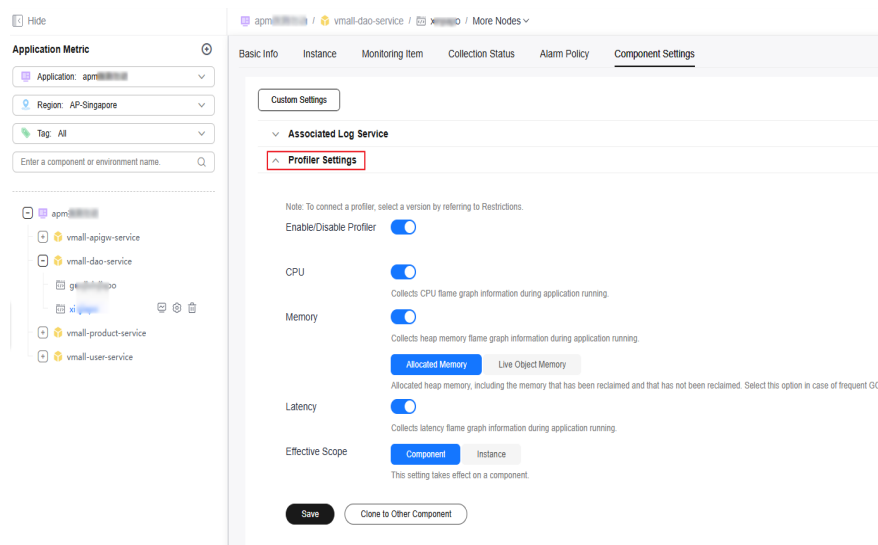


Step 6 Click **Enable Profiler**.

Step 7 In the tree on the left, click  next to the target environment.

Step 8 Click **Component Settings**. Under **Profiler Settings**, enable or disable Profiler. For details, see [Profiler Setting](#).

Figure 4-14 Profiler settings



----End

4.2.9.3 Performance Analysis

Profiler automatically obtains the CPU usage, memory usage, and latency during application running, and displays the calling relationship and execution efficiency of each method, class, thread, and package in real time through flame graphs, helping you optimize code.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

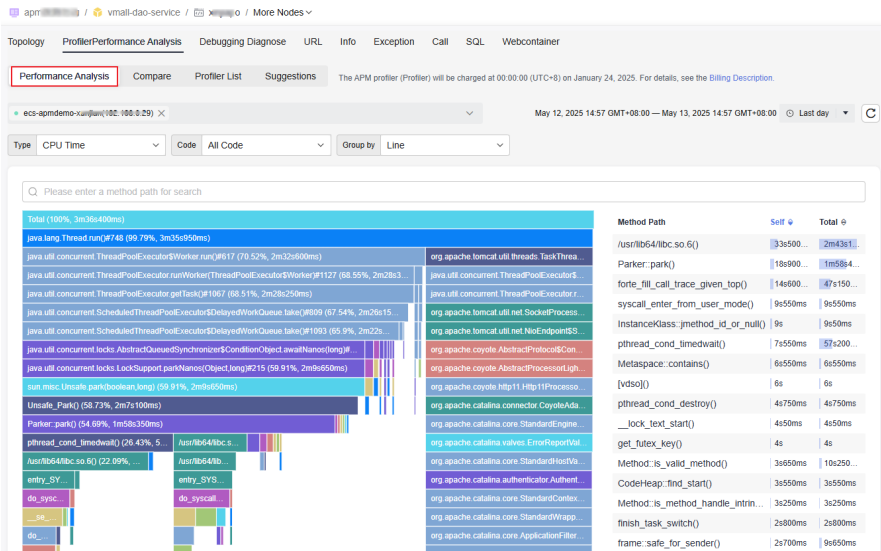
Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Profiler Performance Analysis** tab.

Step 6 Click **Performance Analysis**.

Step 7 On the displayed page, select an instance, performance analysis type, code, and aggregation dimension.

Figure 4-15 Performance analysis



- **Instance:** Select up to two instances from the drop-down list.
- **Type:** Select a type from the drop-down list. Options: **CPU Time**, **Allocated Memory**, **Latency**, and **Live Object Memory**.
- **Code:** Select a range from the drop-down list. Options: **My Code** and **All Code**.
- **Group By:** Select an aggregation dimension from the drop-down list. Options: **Line**, **Method**, **Class**, and **Package**.

Table 4-27 Performance analysis parameters

Item	Parameter	Description
Type	CPU Time	CPU time for each method.
	Allocated Memory	Allocated heap memory (including both reclaimed and unreclaimed memory). It can be used in the frequent GC scenario.

Item	Parameter	Description
	Live Object Memory	Heap memory that has been allocated but has not been reclaimed. It is used in the memory leakage scenario. This feature requires the JDK version to be 17 or later.
	Latency	Total time consumed for executing the methods of trace-related threads. It includes the CPU time and I/O wait time.
Code	My Code	User code.
	All Code	All code.
Group By	Line	Code line.
	Method	Method.
	Class	Class.
	Package	Java package name.

- Search by method path: Set search criteria in the search box and click  to view the flame graph that meets the search criteria.
- Select a time range. Default: **Last 20 minutes**.
Options: **Last 5 minutes**, **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, or **Custom**.
- The calling relationships and execution efficiency of each thread, method, class, and package are displayed in real time in the lower part of the page.

Table 4-28 Profiler parameters

Parameter	Description
Flame Graph	– The Y axis (vertical axis) of the flame graph indicates the call stack. Each layer is a function (package). The deeper a call, the higher a flame layer. The bottom (flame end) is the function that is being executed, and the top is the parent function. – The X axis (horizontal axis) of the flame graph indicates the CPU time, total time for executing trace-related thread methods, allocated heap memory, or heap memory that has been allocated but has not been reclaimed. Take the CPU time as an example. The longer the CPU time, the longer the distance on the X axis of the flame graph. – In the flame graph, the same methods or packages have the same color. – Move the cursor to the flame graph to check the details about a method, class, or thread. – Click a method bar in the flame graph. It is zoomed in horizontally. To restore the flame graph, click Total in the first column.
Method Path	– Move the cursor to a method of the Location class. The detailed information about the method is displayed. – Click a method of the Location class. This method is highlighted on the flame graph, and other methods are grayed. To restore the flame graph, click the same method again.
Self	Performance consumption of the method itself only. The value is the sum of the execution results of all threads and may be greater than the selected time range.
Total	Total performance consumption of both the method itself and called methods. The value is the sum of the execution results of all threads and may be greater than the selected time range.

----End

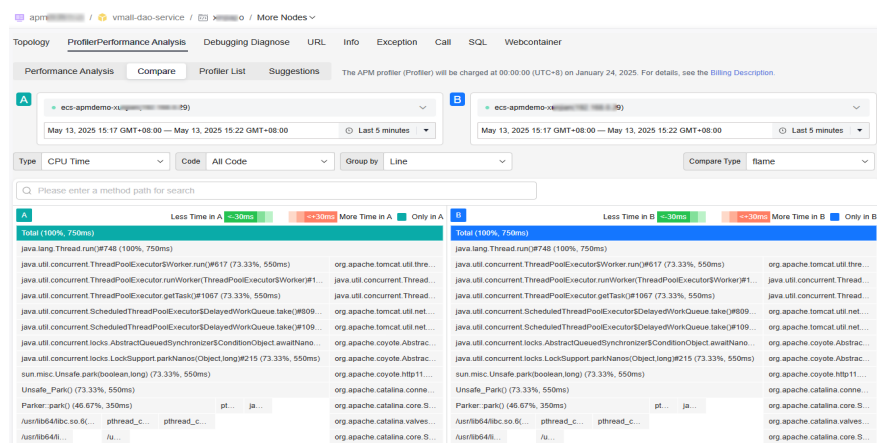
4.2.9.4 Comparison

Profiler compares the method, class, and thread calling relationships and execution efficiency of two instances or the same instance in different time ranges through flame graphs, helping you understand code performance differences.

Procedure

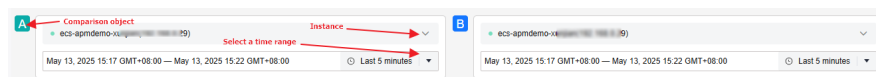
Step 1 Log in to the [APM console](#).

- Step 2** Click  on the left and choose **Application > Application Performance Management**.
- Step 3** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4** In the tree on the left, click  next to the target environment.
- Step 5** Click the **Profiler Performance Analysis** tab.
- Step 6** Click **Compare**.
- Step 7** On the displayed page, select an instance, performance analysis type, code, aggregation dimension, and comparison type.

Figure 4-16 Comparison

- Comparison object: **A** and **B** can be different instances or the same instance in different time ranges.
- Instance: Only one instance can be selected.
- Select a time range. Default: **Last 20 minutes**.

Options: **Last 5 minutes, Last 20 minutes, Last hour, Last 3 hours, Last 6 hours, Last day, Today, Yesterday, or Custom**.

Figure 4-17 Instance comparison

- **Type:** Select a type from the drop-down list. Options: **CPU Time, Allocated Memory, Latency, and Live Object Memory**.
- **Code:** Select a range from the drop-down list. Options: **My Code** and **All Code**.
- **Group By:** Select an aggregation dimension from the drop-down list. Options: **Line, Method, Class, and Package**.

Table 4-29 Type description

Item	Parameter	Description
Type	CPU Time	CPU time for each method.
	Allocated Memory	Allocated heap memory (including both reclaimed and unreclaimed memory). It can be used in the frequent GC scenario.
	Latency	Total time consumed for executing the methods of trace-related threads. It includes the CPU time and I/O wait time.
	Live Object Memory	Heap memory that has been allocated but has not been reclaimed. It is used in the memory leakage scenario. This feature requires the JDK version to be 17.
Code	My Code	User code.
	All Code	All code.
Group By	Line	Code line.
	Method	Method.
	Class	Class.
	Package	Java package name.

- **Comparison Type:** Select a type from the drop-down list. Options: **Table** and **Flame**.
- Search by method path: Set search criteria in the search box and click  to view the flame graph that meets the search criteria.
- The calling relationships and execution efficiency of each method, class, and thread are displayed in real time in the lower part of the page.

Table 4-30 Comparison parameters

Parameter	Description
flame	– The Y axis (vertical axis) of the flame graph indicates the call stack. Each layer is a function (package). The deeper a call, the higher a flame layer. The bottom (flame end) is the function that is being executed, and the top is the parent function. – The X axis (horizontal axis) of the flame graph indicates the CPU time, total time for executing trace-related thread methods, allocated heap memory, or heap memory that has been allocated but has not been reclaimed. Take the CPU time as an example. The longer the CPU time, the longer the distance on the X axis of the flame graph. – In the flame graph, red indicates that the current instance has a larger value, green indicates that the current instance has a smaller value, blue indicates that the method, class, or thread exists only on the current instance, and white indicates that the two instances have the same value. – Move the cursor to the flame graph to check the details about a method, class, or thread. – Click a method bar in the flame graph. It is zoomed in horizontally. To restore the flame graph, click the Total column.
table	– Displays Method Path, Baseline, Comparison, and Diff. – Move the cursor to a target method to check comparison details.

----End

4.2.9.5 Snapshots

APM Profiler supports snapshot query. The snapshot list allows you to obtain the application debugging information by minute, including the time, number of CPU cores, memory allocation rate, and instance name.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

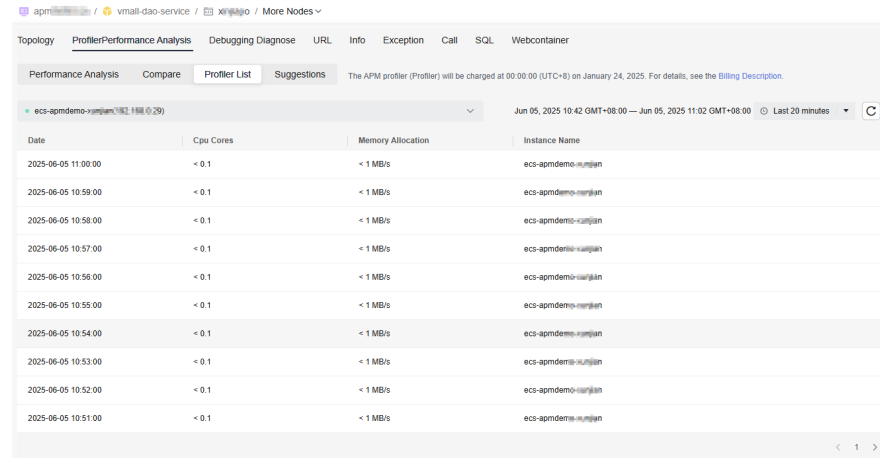
Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Profiler Performance Analysis** tab.

Step 7 On the displayed page, select the target instance to check data.

Figure 4-18 Snapshots



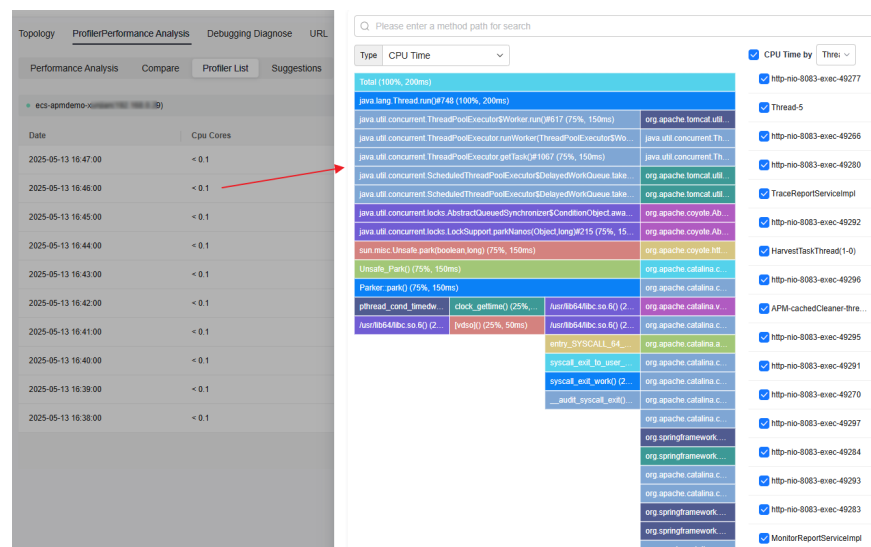
Instance: Select up to one instance from the drop-down list.

Table 4-31 Snapshot parameters

Parameter	Description
Date	Time when the debugging information is obtained
CPU Cores	Number of CPU cores
Memory Allocation	The amount of memory allocated per time unit
Instance	Instance name

Step 8 Locate a row in the list and click it. The corresponding flame graph is displayed.

Figure 4-19 Snapshot flame graph



Search by method name: Set search criteria in the search box and click  to view the flame graph that meets the search criteria.

Type: Select a type from the drop-down list. Options: **CPU Time**, **Allocated Memory**, **Latency**, and **Live Object Memory**.

CPU Time by: Select a value from the drop-down list. Options: **Line**, **Method**, **Class**, **Package**, **Thread**, and **API**. You can select one or more options.

Table 4-32 Flame graph description

Parameter	Description
Flame Graph	• The Y axis (vertical axis) of the flame graph indicates the call stack. Each layer is a function (package). The deeper a call, the higher a flame layer. The bottom (flame end) is the function that is being executed, and the top is the parent function. • The X axis (horizontal axis) of the flame graph indicates the CPU time, total time for executing trace-related thread methods, allocated heap memory, or heap memory that has been allocated but has not been reclaimed. Take the CPU time as an example. The longer the CPU time, the longer the distance on the X axis of the flame graph. • In the flame graph, the same methods or packages have the same color. • Move the cursor to the flame graph to check the details about a method, class, or thread. • Click a method bar in the flame graph. It is zoomed in horizontally. To restore the flame graph, click Total in the first column.

----End

4.2.9.6 Suggestions

APM Profiler performance analysis provides suggestions. Based on the performance data, APM identifies the methods with excessively high CPU usage and provides suggestions.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Profiler Performance Analysis** tab.

Step 6 Click **Suggestions**. The **Suggestions** page is displayed.

Instance: Select up to one instance from the drop-down list.

Table 4-33 Suggestion parameters

Parameter	Description
Type	Performance issue type. Only the CPU type is available. The memory and latency types will be available later.
Performance Issue	Description of a performance issue.
Call Method	Name of the method where the issue occurs.
Number of Issues	Number of times that the performance issue occurred within a specified time range.
CPU Usage	CPU usage when the issue occurred last time.
Operation	Operations that are supported. Option: View Details .

Step 7 Click **View Details** in the **Operation** column to view suggestions. The information includes the start time, end time, issue description, status, handling procedure, and issue details. You can search for data by method path.

----End

4.2.10 Debugging Diagnosis

4.2.10.1 Overview

Debugging diagnosis helps you analyze the thread status and obtain real-time method and object parameters. Developers can obtain real-time debugging information from remote applications without restarting the applications or causing service interruption.

Constraints

1. To ensure stable application running, you are advised to upgrade the JDK version as required. If you use debugging diagnosis on the JDK of an earlier version, the application may break down.
2. Debugging diagnosis is not generally available.

Table 4-34 JDK version description

JDK Type	Version
OpenJDK	• OpenJDK 8u352+ • OpenJDK 17.0.5+

JDK Type	Version
Oracle JDK	• Oracle JDK 17.0.9+

Table 4-35 Agent version description

Agent Version	Description
2.4.10-profiler	Agent 2.4.10-profiler and later versions support debugging diagnosis.

Table 4-36 Constraints

No.	Constraint
1	Debugging diagnosis requires JDK 8 or JDK 17. This function cannot be used when there is only JRE.
2	Debugging diagnosis supports only Java.

Impact on Performance

Based on performance tests on the applications with all common functions enabled, debugging diagnosis increases the CPU usage by less than 5%, memory by less than 100 MB, and latency by less than 5%.

4.2.10.2 Thread Analysis

Thread analysis displays all threads of an application and thread stack information, helping you quickly locate time-consuming threads.

Constraints

Because method and object parameters are obtained for thread analysis in real time, information such as the thread status may change.

Procedure

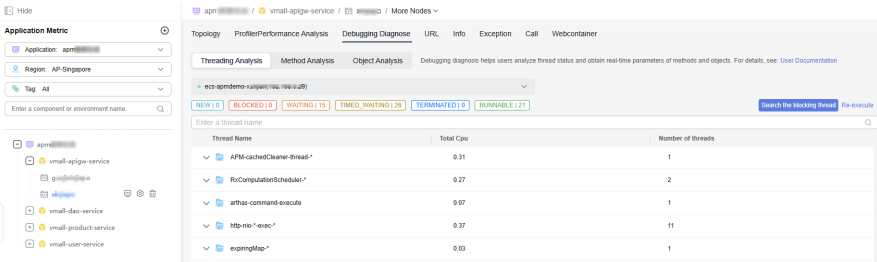
Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Figure 4-20 Debugging diagnosis



- Step 5** Click the **Debugging Diagnosis** tab.
- Step 6** Check the **Thread Analysis** page, which is displayed by default.
1. Select an instance from the drop-down list.
 2. In the upper right corner of the thread analysis page, click **Searching the blocking thread** to view the details about the blocked threads.
 3. Click **Re-execute** to search the threads again.
 4. Enter the filter criteria in the search box. Click  to view the threads that meet the filter criteria.
 5. Thread analysis: displays the thread name, CPU usage, and number of threads.
 6. Threads can be classified by status (**NEW**, **BLOCKED**, **WAITING**, **TIMED_WAITING**, **TERMINATED**, or **RUNNABLE**). You can click one type of threads to view thread analysis details.
- Step 7** Click  next to a thread to view the thread analysis information, including the thread name, thread ID, CPU usage, state, and operations.

Figure 4-21 Checking thread analysis information

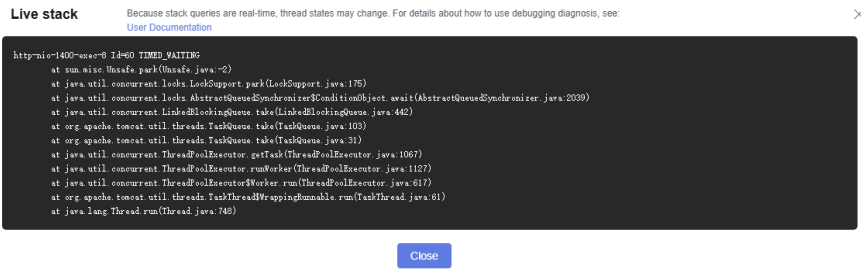
Thread Name	Total Cpu	Number of threads		
APM-cachedCleaner-thread-*	0.31	1		
Thread Name	Thread ID	Total Cpu	State	Actions
APM-cachedCleaner-thread-1	17	0.31	TIMED_WAITING	View the live stack

Table 4-37 Thread analysis parameters

Parameter	Description
Thread Name	Name of a thread.
Thread ID	ID of a thread.
Total CPU	Total CPU
State	Thread state. Options: NEW , BLOCKED , WAITING , TIMED_WAITING , TERMINATED , and RUNNABLE .

- Step 8** Click **View the live stack** in the **Operation** column to obtain the stack information.

Figure 4-22 Live stack



If the value of **Thread ID** is **-1**, the **View the live stack** option is grayed, showing that internal threads do not support stack query.

Figure 4-23 Stack query not supported



Step 9 Click **Close**. The **Thread Analysis** page is displayed.

----End

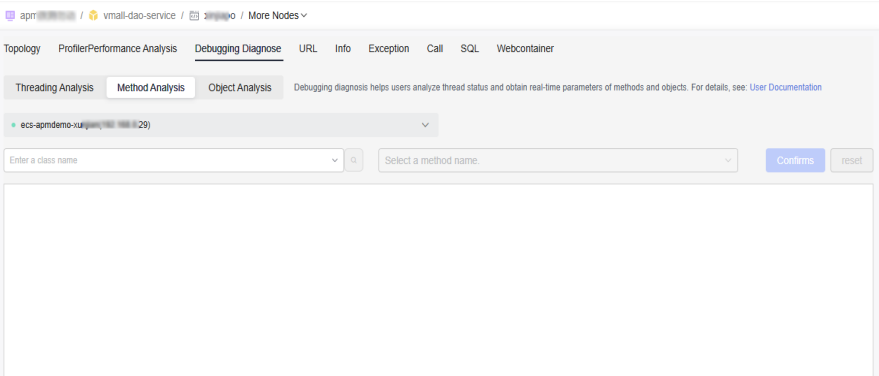
4.2.10.3 Method Analysis

Method analysis shows the time consumption details, invocation source, and source code of running methods. This helps you quickly locate root causes in scenarios where faults cannot be reproduced offline or logs are missing.

Procedure

- Step 1 Log in to the [APM console](#).
- Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.
- Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4 In the tree on the left, click  next to the target environment.
- Step 5 Click the **Debugging Diagnosis** tab.
- Step 6 Click **Method Analysis**. The **Method Analysis** page is displayed.

Figure 4-24 Method analysis



Step 7 In the **Enter a class name** text box, enter a class name. Click  and select a class name from the drop-down list.

In the **Select a method name** text box, select a method name. Click **Confirm**. The execution information of the method is displayed.

Figure 4-25 Class name and method name

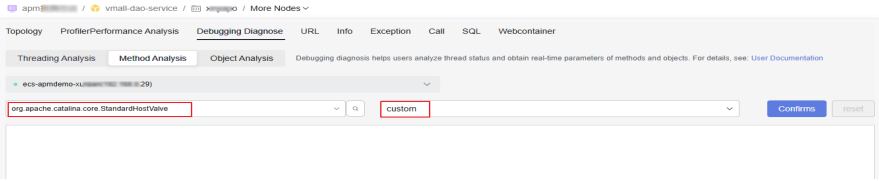
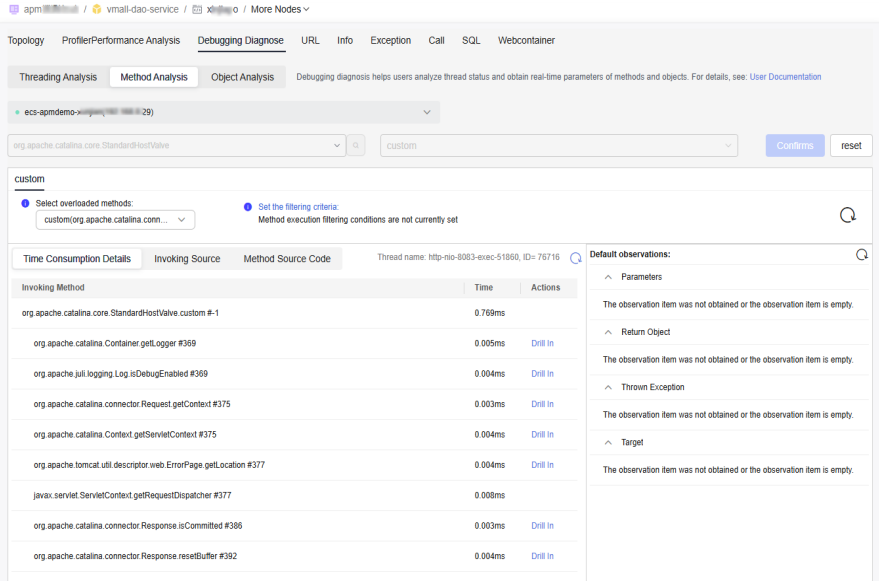


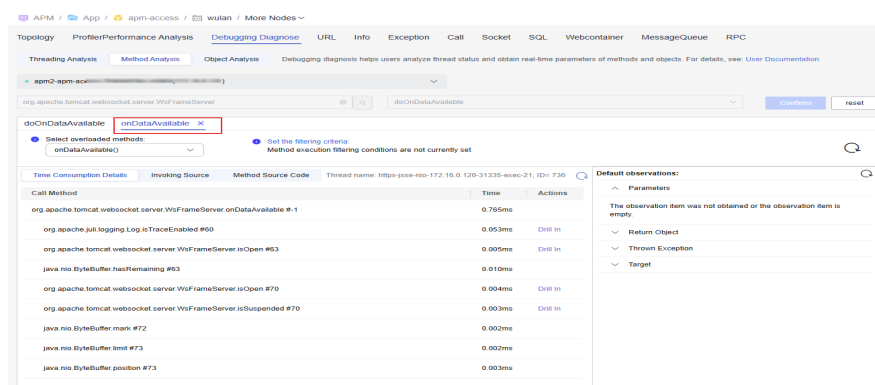
Figure 4-26 Method analysis details



- The execution records of the method are displayed on the left of the page, including **Time Consumption Details**, **Invoking Source**, and **Method Source Code**.
 - a. **Time Consumption Details**: include the call method, time consumed, and drill-in operation.

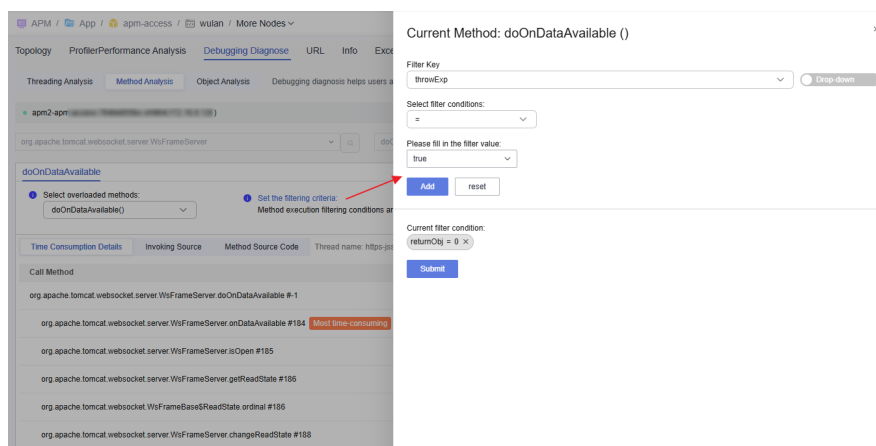
- b. Click **Drill In** to view details about the called method.

Figure 4-27 Drill-down



- Default observation items are displayed on the right of the page, including **Parameters**, **Return Object**, **Thrown Exception**, and **Target**.
- **Select overloaded methods:** In the drop-down list, select an overloaded method.
- **Set the filtering criteria:** Filter method records that meet specified criteria.

Figure 4-28 Setting the filtering criteria



- Select a filter key from the drop-down list or enter one.
- Select a filter criterion from the drop-down list.
- Enter a filter value.
- Click **Add**. The filter criteria will be displayed in the current dialog box.
- Click **Submit** to complete the configuration.

----End

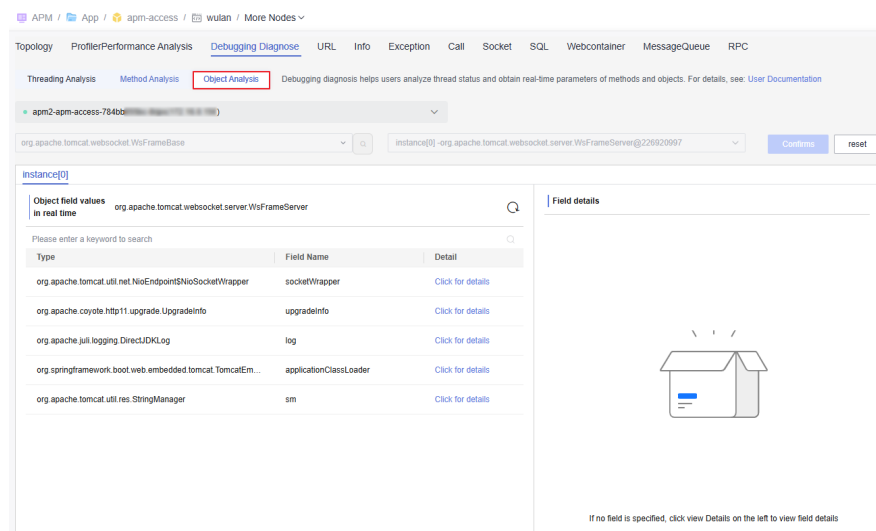
4.2.10.4 Object Analysis

Object analysis enables you to analyze the internal status of an object and view the parameter values and details of the object.

Procedure

- Step 1** Log in to the [APM console](#).
- Step 2** Click  on the left and choose **Management & Governance > Application Performance Management**.
- Step 3** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4** In the tree on the left, click  next to the target environment.
- Step 5** Click the **Debugging Diagnosis** tab.
- Step 6** Click **Object Analysis**. The **Object Analysis** page is displayed.

Figure 4-29 Object analysis



- Step 7** Enter a class name, select an instance, and click **Confirm**. The real-time information of the instance's field is displayed. The details include the type, field name, and details.
- Step 8** Click **View Details** in the **Details** column. The detailed information about the field is displayed in the right pane.

----End

4.3 Application Monitoring Configuration

4.3.1 Configuration Details

You can define collection parameters for some collectors corresponding to monitoring items.

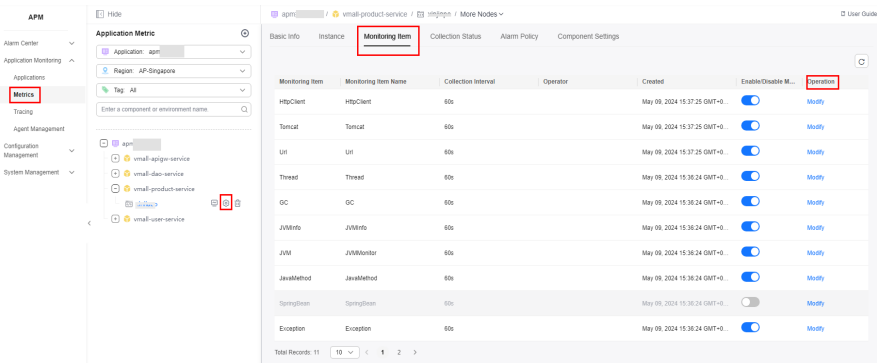
Constraints

On the **Monitoring Item** tab page, only monitoring items related to the connected application are displayed.

Configuring a Monitoring Item

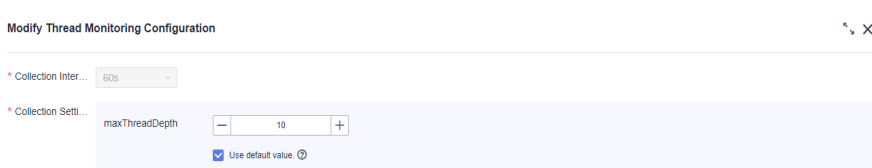
- Step 1
- Log in to the [APM console](#).
- Step 2
- Click  on the left and choose **Application > Application Performance Management**.
- Step 3
- In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4
- In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.
- Step 5
- Click the **Monitoring Item** tab.
- Step 6
- Locate the row that contains the target monitoring item and click **Modify** in the **Operation** column.

Figure 4-30 Configuring a monitoring item



- Step 7
- On the displayed page, edit the monitoring configuration. For details, see the corresponding section.

Figure 4-31 Editing the thread monitoring configuration



- Step 8
- Click **Yes**.
- End

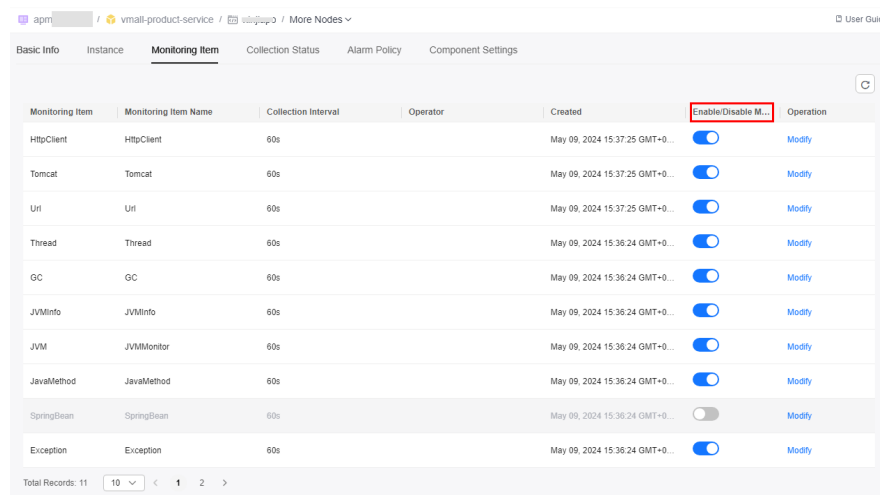
Enabling or Disabling a Monitoring Item

- Step 1
- Log in to the [APM console](#).
- Step 2
- Click  on the left and choose **Application > Application Performance Management**.
- Step 3
- In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.

Step 5 Click the **Monitoring Item** tab.

Figure 4-32 Enabling or disabling a monitoring item



Monitoring Item	Monitoring Item Name	Collection Interval	Operator	Created	Enable/Disable M...	Operation
HttpClient	HttpClient	60s		May 09, 2024 15:37:25 GMT+0...	☒	Modify
Tomcat	Tomcat	60s		May 09, 2024 15:37:25 GMT+0...	☒	Modify
Uri	Uri	60s		May 09, 2024 15:37:25 GMT+0...	☒	Modify
Thread	Thread	60s		May 09, 2024 15:36:24 GMT+0...	☒	Modify
GC	GC	60s		May 09, 2024 15:36:24 GMT+0...	☒	Modify
JVMInfo	JVMInfo	60s		May 09, 2024 15:36:24 GMT+0...	☒	Modify
JVM	JVMMonitor	60s		May 09, 2024 15:36:24 GMT+0...	☒	Modify
JavaMethod	JavaMethod	60s		May 09, 2024 15:36:24 GMT+0...	☒	Modify
SpringBean	SpringBean	60s		May 09, 2024 15:36:24 GMT+0...	☐	Modify
Exception	Exception	60s		May 09, 2024 15:36:24 GMT+0...	☒	Modify

Total Records: 11 10 < 1 2 >

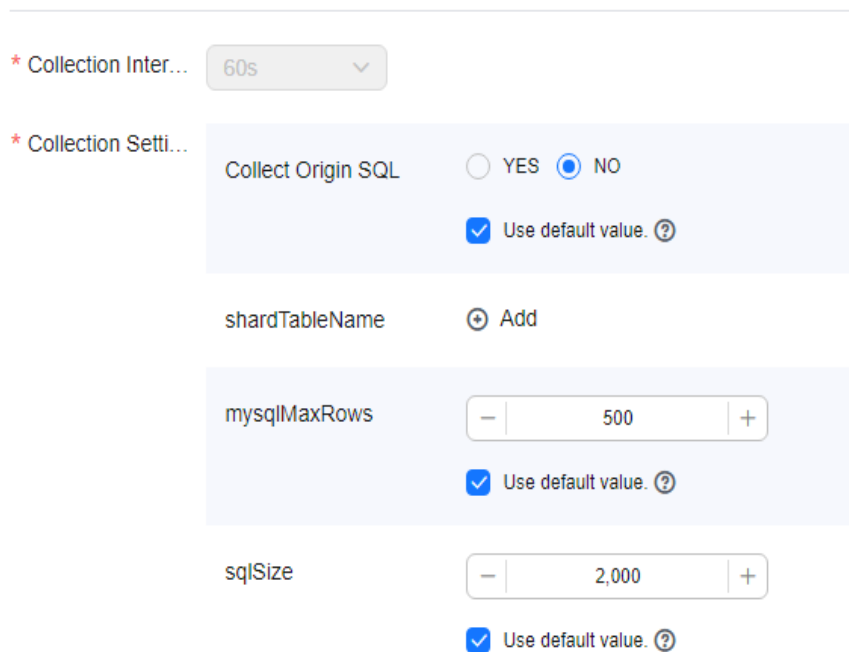
Step 6 Locate the row that contains the target monitoring item and enable or disable it.

----End

4.3.2 Configuring the MySQL Monitoring Item

On the **Modify MySQL Monitoring Configuration** page, set the following parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **Collect Original SQL:** This function is disabled by default. In that case, only SQL statements without values are collected, for example, **select name from user where id=?**. When this function is enabled, SQL statements with values are collected, for example, **select name from user where id=1**.
- **shardTableName:** specified when you need to aggregate multiple tables into one table. For example, there are two tables: **UserTable_1** and **UserTable_2**. By default, two SQL statements (**select name from UserTable_1** and **select name from UserTable_2**) are displayed on the SQL monitoring page. If you set **shardTableName** to **UserTable**, tables starting with **UserTable** are aggregated into the same table. Only one SQL statement (**select name from UserTable**) is displayed on the SQL monitoring page.
- **mysqlMaxRows:** the maximum number of MySQL rows that can be collected by the Agent. If this value has been reached, the Agent stops collecting data immediately. The default value is **500**. The value ranges from 10 to 2000.
- **sqlSize:** the maximum number of SQL statement bytes that can be collected by the Agent. Only when the number of bytes is within **sqlSize**, can SQL statements be collected. The default value is **2000**. The value ranges from 10 to 4096.

Figure 4-33 Configuring the MySQL monitoring item**Modify Mysql Monitoring Configuration**

* Collection Inter... 60s

* Collection Setti...

Collect Origin SQL ☐ YES ☒ NO

☒ Use default value. ?

shardTableName

mysqlMaxRows

☒ Use default value. ?

sqlSize

☒ Use default value. ?

4.3.3 Configuring the HttpClient Monitoring Item

On the **Modify HttpClient Monitoring Configuration** page, set the following URL normalization parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- URL normalization is used to aggregate URLs that meet the conditions you set. For example, **http://localhost/rest/v1/test/123** and **http://localhost/rest/v1/test/234** can be aggregated into **http://localhost/rest/v1/test/{id}**. For details, see [URL Normalization Configuration](#).
- **httpClientMaxRows:** the maximum number of HttpClient rows that can be collected by the Agent. If this value has been reached, the Agent stops collecting data immediately. The default value is **500**. The value ranges from 10 to 2000.
- **Blocklist configuration:** The data of URLs that meet the conditions you set will not be collected. For details, see [Blocklist Configuration](#).

Figure 4-34 Configuring the HttpClient monitoring item

Modify HttpClient Monitoring Configuration

* Collection Inter... 60s

* Collection Sett...

URL Normalization C...

Match Mode (Startwith, Endwith...	Match Expression	Normalized URL	Operation
startwith	/v1/test/	/v1/test/{id}	Delete

Add

httpClientMaxRows 500

☒ Use default value

Actual Value Global Tag 10

Blocklist Configuration

Match Mode (Startwith, Endwith, Include, or Re...	Match Expression	Operation
startwith	/v1/test*	Delete
endwith	/healthcheck	Delete
include	test	Delete
regex	http://.*test*	Delete

Add

Confirm Cancel

URL Normalization Configuration

There are four modes: **Startwith**, **Endwith**, **Include**, and **Regex**.

- **Startwith**: The URLs starting with a certain expression are counted as normalized URLs. For example, URLs starting with **http://127.0.0.1/v1** are aggregated into **/v1/test/{id}**, as shown in [Figure 4-34](#).
- **Endwith**: The URLs ending with a certain expression are counted as normalized URLs. For example, URLs ending with **/test** are aggregated into **/ {id}/test**, as shown in [Figure 4-34](#).
- **Include**: The URLs containing a certain expression are counted as normalized URLs. For example, URLs containing **test** are aggregated into **/test/{id}**, as shown in [Figure 4-34](#).
- **Regex**: The URLs that meet the wildcard expression are counted as normalized URLs. For details about the wildcard rules, see [Table 4-38](#).

Table 4-38 Wildcard description

Wildcard	Description
?	Matches any character.
*	Matches zero, one, or more characters.
**	Matches zero, one, or more directories.

Blocklist Configuration

There are four modes: **Startwith**, **Endwith**, **Include**, and **Regex**.

- **Startwith**: The URLs starting with a certain expression are blocked, as shown in [Figure 4-34](#).

- **Endwith:** The URLs ending with a certain expression are blocked, as shown in [Figure 4-34](#).
- **Include:** The URLs containing a certain expression are blocked, as shown in [Figure 4-34](#).
- **Regex:** The URLs that meet the wildcard expression are blocked. For details about the wildcard rules, see [Table 4-38](#).

Usage Example

The following is an example:

URL Path	Description
/app/p?ttern	Matches files such as /app/pattern and /app/pAttern , excluding /app/pttern .
/app/*.x	Matches all .x files in the app directory.
/**/*.example	Matches /app/example , /app/foo/example , and /example .
/app/**/*.dir/file.*	Matches /app/dir/file.jsp , /app/foo/dir/file.htm , /app/foo/bar/dir/file.pdf , and /app/dir/file.c .
/**/*.jsp	Matches all .jsp files.

4.3.4 Configuring the URL Monitoring Item

On the **Modify URL Monitoring Configuration** page, set the following parameters:

Constraints

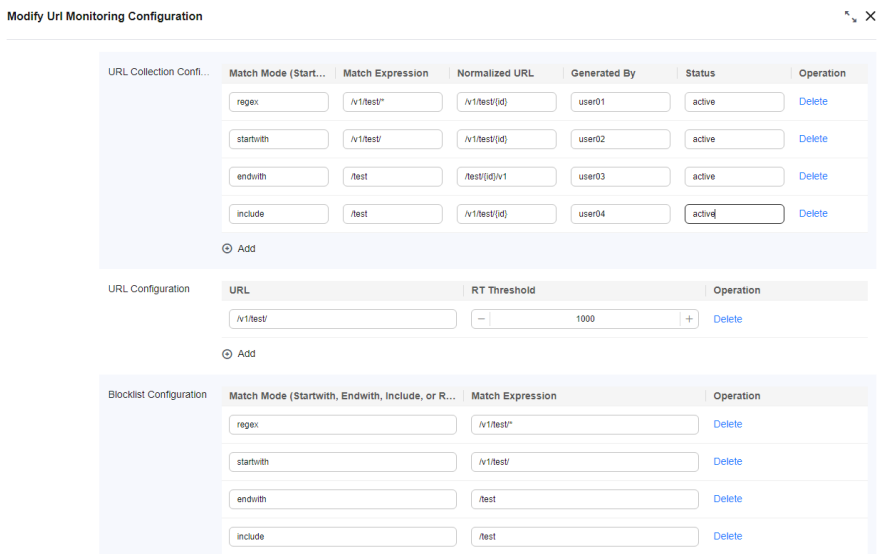
For security purposes, do not contain sensitive data in headers, URL parameters, cookies, or other parameters.

Parameter	Description	Example
Collection Interval	The default value is 60s and cannot be changed.	60s
Key for Header Value Interception	Key specified for collecting values in headers. The collected information can be seen in the trace parameters.	Host

Parameter	Description	Example
Key for Parameter Value Interception	Key specified for collecting values in URLs. The collected information can be seen in the trace parameters. Take http://127.0.0.1/test?param=123 as an example. If the key is set to param , value 123 can be seen in the trace parameters.	param
Key for Cookie Value Interception	Key specified for collecting values in cookies. The collected information can be seen in the trace parameters.	testKey
URL Collection Configuration	URLs that meet the conditions you set are aggregated. For example, /rest/v1/test/123 and /rest/v1/test/234 can be aggregated into /rest/v1/test/{id}. The configuration method is the same as that described in HttpClient URL Normalization. Generated By: user or automatic. If Generated By is set to automatic, the normalization mode can only be regex. Status: active or inactive. If Status is set to inactive, the current URL collection configuration does not take effect.	Figure 4-35
Blocklist Configuration	Data of URLs that meet the conditions you set will not be collected. The configuration method is the same as that described in HttpClient URL Normalization .	Figure 4-35
Service Code Length	Maximum length of the response body to be parsed to prevent the performance from being affected. Content that beyond this limit will not be parsed, but corresponding service status codes are regarded as normal by default.	-
Key for Service Code Interception	Key specified for collecting service status codes. If the custom API returned content is {"errorCode":500,"errorMsg":"error msg"} , set this parameter to errorCode .	errorCode
Normal Service Code	If this status code is returned, traces are regarded as normal. If other codes are returned, traces are regarded as abnormal.	-
Slow Request Threshold	Global response time threshold. The default value is 800 . Requests with the response time longer than 800 ms are regarded as slow requests. The sampling ratio of slow requests will be increased.	-

Parameter	Description	Example
URL Configuration	Response time threshold separately set for a URL. If the response time of this URL exceeds the threshold, the sampling rate of this URL will be increased. If this parameter is not set, the global slow request threshold is used by default.	Figure 4-35
Error Code	Options: 400 or greater and 500 or greater (default). By default, if status code 500 or greater occurs, the system regards that there is an error.	-
URL Automatic Normalization	Example: There are three URL invocations: /get/xxx/a /get/xxx/b /get/xxx/b - If this parameter is set to Yes, URL automatic normalization is enabled. After normalization: /get/xxx/a 1 /get/xxx/b 2 - If this parameter is set to No, URL automatic normalization is disabled. /get/xxx/{p} 3 Use default value: The inherited tag value is preferentially used.	-

Figure 4-35 Example



4.3.5 Configuring the JavaMethod Monitoring Item

On the **Modify JavaMethod Monitoring Configuration** page, set method interception parameters.

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **Method Interception Configuration:** is used to collect specified service methods. The method data is displayed on the JavaMethod metric page and in traces.
- **Intercepted Class:** name of the fully-qualified class to be collected. Both the package name and class name need to be specified.
- **Intercepted Method:** name of the method to be collected. If multiple methods exist, separate them by commas (,), for example, **testMethod1,testMethod2**.

Figure 4-36 Configuring the JavaMethod monitoring item

Modify JavaMethod Monitoring Configuration

* Collection Inter... 60s

* Collection Setti...

Method Interception ...	Intercepted Class	Intercepted Method (Multi...	Operation
	com.test.TestClass	testMethod	Delete

+ Add

4.3.6 Configuring the Druid Monitoring Item

On the **Modify Druid Monitoring Configuration** page, set the following parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **TraceReportTimeSpanThreshold(ms):** threshold for reporting getConnection method traces. If the threshold is not exceeded, such traces will not be reported. The default value is **1**. If you select **Use default value**, the value of the inherited tag is preferentially used.
- **Get pool info when calling getConnection:** specifies whether to obtain the pool information when calling the **getConnection** method. The default value is **No**. If you select **Use default value**, the value of the inherited tag is preferentially used.

4.3.7 Configuring the ApacheHttpAsyncClient Monitoring Item

On the **Modify ApacheHttpAsyncClient Monitoring Configuration** page, set the following parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.

4.3.8 Configuring the Redis Monitoring Item

On the **Modify Redis Monitoring Configuration** page, set the following parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **Parameter Parsing:** The default value is **No**. If you select **Use default value**, the value of the inherited tag is preferentially used.
- **Parameter Length:** The default value is **1000**. If you select **Use default value**, the value of the inherited tag is preferentially used.
- **Distinguish Redis Ports:** The default value is **No**. If you select **Use default value**, the value of the inherited tag is preferentially used.

4.3.9 Configuring the Jedis Monitoring Item

On the **Modify Jedis Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.3.10 Configuring the HBase Monitoring Item

On the **Modify HBase Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.3.11 Configuring the ApacheHttpClient Monitoring Item

On the **Modify ApacheHttpClient Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.3.12 Configuring the Tomcat Monitoring Item

On the **Modify Tomcat Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.3.13 Configuring the EsRestClient Monitoring Item

On the **Modify EsRestClient Monitoring Configuration** page, set the following parameter:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **Index Normalization Configuration:** The system matches indexes based on a regular expression and then normalizes them.

4.3.14 Configuring the WebSocket Monitoring Item

On the **Modify WebSocket Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.3.15 Configuring the KafkaProducer Monitoring Item

On the **Modify KafkaProducer Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.3.16 Configuring the Hikari Monitoring Item

On the **Modify Hikari Monitoring Configuration** page, set the following parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **TraceReportTimeSpanThreshold(ms):** The default value is **1**. If **Use default value** is selected, the value of the inherited tag is preferentially used.
- **Get pool info when calling getConnection:** The default value is **No**. If **Use default value** is selected, the value of the inherited tag is preferentially used.

4.3.17 Configuring the Exception Monitoring Item

Constraints

To record thread names in logs, you need to add **%thread** to the **logback.xml** file.

```
<pattern>%d{yyyy-MM-dd HH:mm:ss.SSS}|%X{apm-traceid}| %X{apm-gtraceid}| %thread %-5level | %logger{35} | %msg %n</pattern>
```

On the **Modify Exception Monitoring Configuration** page, set the following parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **Determine Trace Exception upon Log Error Detection:** The default value is **Yes**. If **Use default value** is selected, the value of the inherited tag is preferentially used.
- **Associating Service Logs with Traceld:** The default value is **No**. If **Use default value** is selected, the value of the inherited tag is preferentially used.

4.3.18 Configuring the Thread Monitoring Item

On the **Modify Thread Monitoring Configuration** page, set the following parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **maxThreadDepth:** The default value is **10** and the maximum number is **50**. If you select **Use default value**, the value of the inherited tag is preferentially used.

4.3.19 Configuring the GC Monitoring Item

On the **Modify GC Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.3.20 Configuring the JVMInfo Monitoring Item

On the **Modify JVMInfo Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.3.21 Configuring the JVMMonitor Monitoring Item

On the **Modify JVMMonitor Monitoring Configuration** page, set the following parameters:

- **Collection Interval:** The default value is **60s** and cannot be changed.
- **Call Chain Stack Collection Threshold:** When the request latency exceeds the threshold, the stack is automatically printed. The default value is **0** and the maximum value is **10000**.

4.3.22 Configuring ProbeInfo Monitoring Item

On the **Modify ProbeInfo Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

4.4 Monitoring Item Views

APM supports summary tables, trend graphs, latest data tables, and original data tables. The view of each monitoring item is configured in the background and has not been opened. You can check views together with corresponding background metric sets. For details, see [Metric Sets](#).

- **Summary table:** records the summary calculation results based on the primary key metric within a period. You can click a number or string in the summary table to view the trend graph of the primary key metric.
- **Trend graph:** displays the trend of a primary key metric in a period. A trend graph may have breakpoints, indicating that no data is collected in this period. There are multiple reasons why data is not collected. For example, collectors do not collect the metrics with zero calls or the data may be lost.
- **Original data table:** For character strings, no trend graphs can be generated. Therefore, original data tables are used. Each row indicates the mapping between a time and a value.
- **Latest data table:** displays the latest data. You can click a data record to view its trend graph.

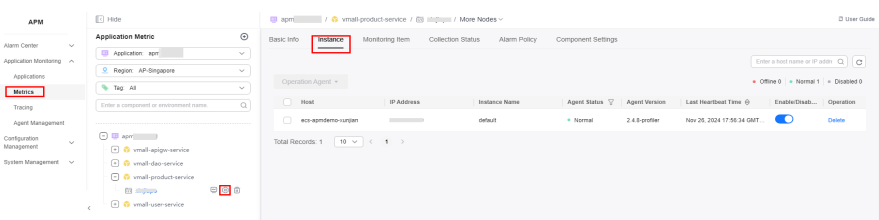
4.5 Instance

On the **Instance** page, you can view instance information, and stop, start, or delete Agents on instances.

Viewing Instances

- Step 1
- Log in to the [APM console](#).
- Step 2
- Click  on the left and choose **Application > Application Performance Management**.
- Step 3
- In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4
- In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.
- Step 5
- On the displayed page, click the **Instance** tab to view the instance list.

Figure 4-37 Instance



1.
- Set the search criteria and click  in the search box in the upper right corner of the page to filter instances.
2.
- In the upper right corner of the page, view the number of offline, normal, or disabled instances.

Table 4-39 Instance parameters

Parameter	Description
Host	Host name.
IP Address	IP address of an instance.
Instance Name	Instance name.
Agent Status	Agent status. Options: Offline , Normal , and Disabled . - By default, normal Agents are displayed first, then disabled ones, and offline ones. - You can click  in the Agent Status column to filter data by Agent status.
Agent Version	Agent version.
Last Heartbeat Time	Latest time when instance data was collected. You can click  in the Last Heartbeat Time column to sort data.
Enable/Disable Agent	Enable or disable the Agent.

Parameter	Description
Operation	Operation performed on the instance. You can click Delete to delete the instance.

----End

Enabling or Disabling Agents

When the Agent is enabled,  is displayed. When the Agent is disabled,  is displayed.

Disabling an Agent

Step 1 In the instance list, locate the instance for which you want to disable an Agent and click .

Step 2 In the displayed dialog box, click **Yes**.

----End

Enabling an Agent

Step 1 In the instance list, locate the instance for which you want to enable an Agent and click .

Step 2 In the displayed dialog box, click **Yes**.

----End

Operating Agents in Batches

Step 1 In the instance list, select desired instances.

Step 2 Click **Operate Agent**.

Step 3 In the drop-down list, select **Enable Agent**, **Disable Agent**, or **Delete Agent**.

Step 4 In the displayed dialog box, click **Yes** to enable, disable, or delete the Agent.

----End

4.6 Collection Status

On the **Collection Status** page, you can view the collection status of hosts.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

- Step 3
- In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4
- In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.
- Step 5
- On the displayed page, click the **Collection Status** tab to view the collection status of hosts.

Figure 4-38 Viewing the collection status

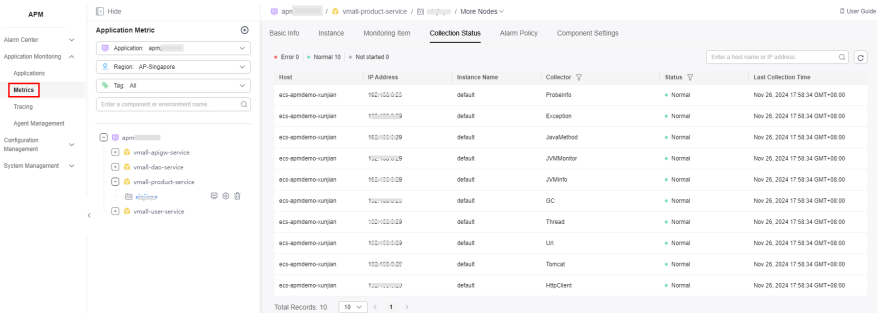


Table 4-40 Collection status description

Parameter	Description
Host	Host name.
IP Address	IP address of an instance.
Instance Name	Instance name.
Collector	Collector name. You can click  in the Collector column to filter data by collector name.
Status	Collection status. Options: Normal , Error , and Not started . You can click  in the Status column to filter data by status.
Last Collection Time	Latest time when the collector collected data.

----End

4.7 Component Settings

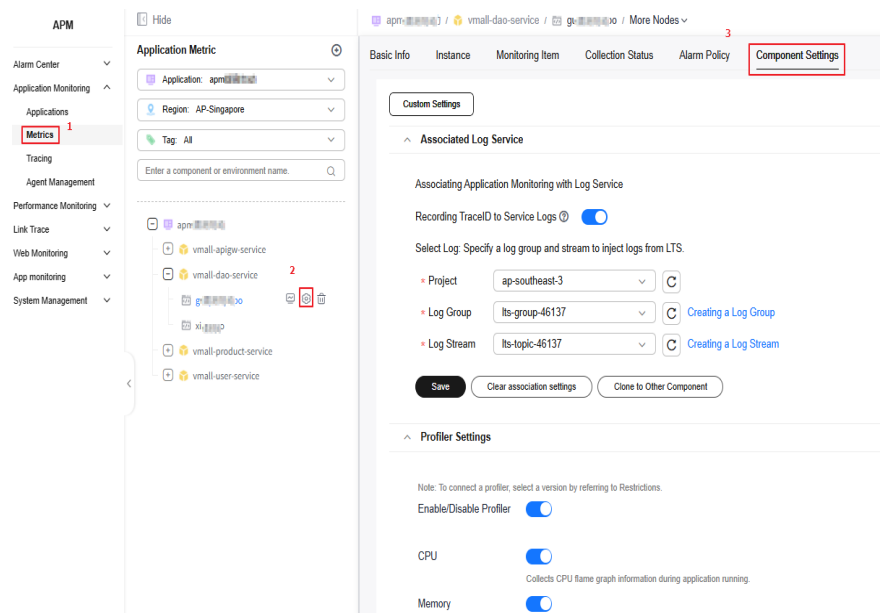
Component settings include log association setting, Profiler setting, and database monitoring setting.

Procedure

- Step 1
- Log in to the **APM console**.

- Step 2** Click  on the left and choose **Application > Application Performance Management**.
- Step 3** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4** In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.
- Step 5** Click the **Component Settings** tab.

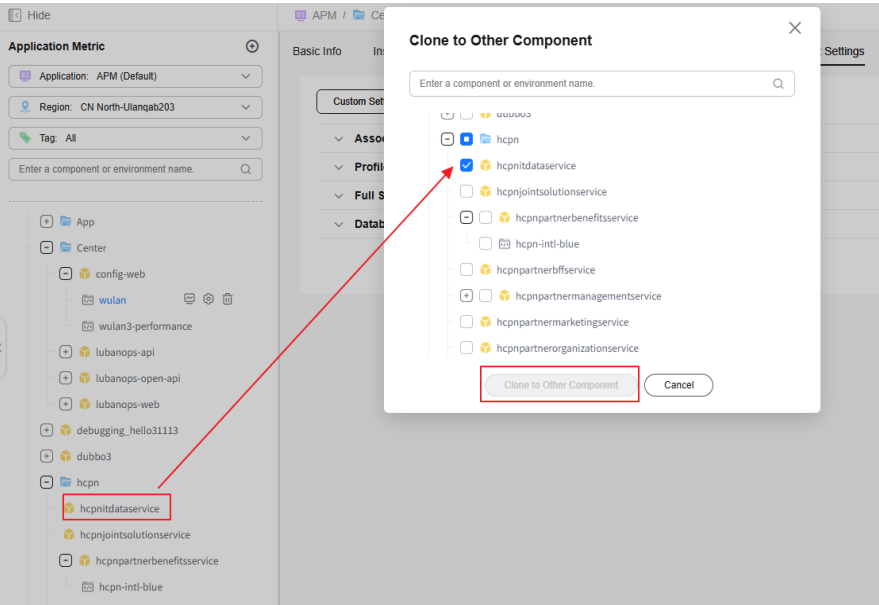
Figure 4-39 Component settings



- Step 6** Click **Multiplexing to Other Component**.
- Step 7** On the displayed page, select one or more components and click **Clone to Other Component**. The log association setting, Profiler setting, and database monitoring setting will then be copied to the selected components.

Before you click **Clone to Other Component** or **Multiplexing to Other Component**, ensure that the components you selected contain environments. Otherwise, the button will be grayed and your settings cannot be cloned.

Figure 4-40 Settings that cannot be cloned to other components



----End

Associating Traces with Logs

Trace IDs can be associated with logs in Log Tank Service (LTS). Then you can find out the logs based on the associated trace IDs for fast troubleshooting.

- Printing trace IDs to service logs
 - a. If you need to print trace IDs to service logs, thread names must be printed to logs first. The following figure shows how to print thread names to logs. That is, add the **%thread** field to the **logback.xml** file.

```
<pattern>%d{yyyy-MM-dd HH:mm:ss.SSS}|%X{apm-traceid}| %X{apm-gtraceid}| %thread | %-5level | %logger{35} | %msg %n
</pattern>
```
 - b. If this function is enabled, trace IDs are automatically generated in service logs.
 - c. If this function is disabled, the log association setting becomes ineffective.
- The Log4j, Log4j2, and Logback log components are supported for associating service logs.
- Only the Java type is supported.

Step 1 Under log association settings, enter related information.

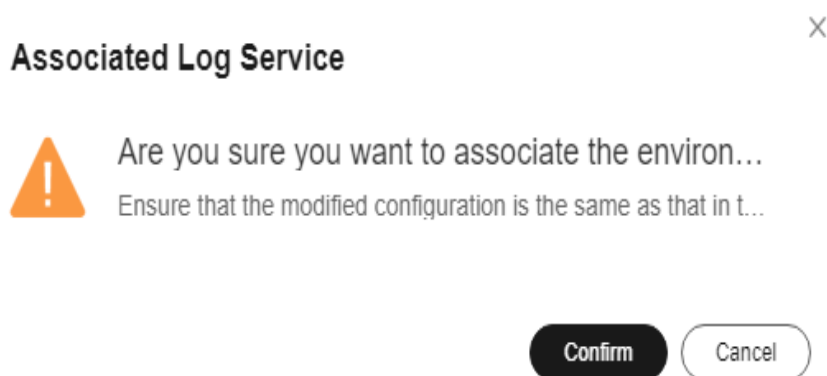
Table 4-41 Parameters for log association settings

Parameter	Description
Project	Select a project from the drop-down list.
Log Group	A log group is a group of log streams which share the same log retention settings. Up to 100 log groups can be created for a single account. For how to create a log group, see Creating a Log Group .

Parameter	Description
Log Stream	A log stream is the basic unit for reading and writing logs. You can put different types of logs into different streams to ease management. For how to create a log stream, see Creating a Log Stream .

Step 2 Click **Save**.

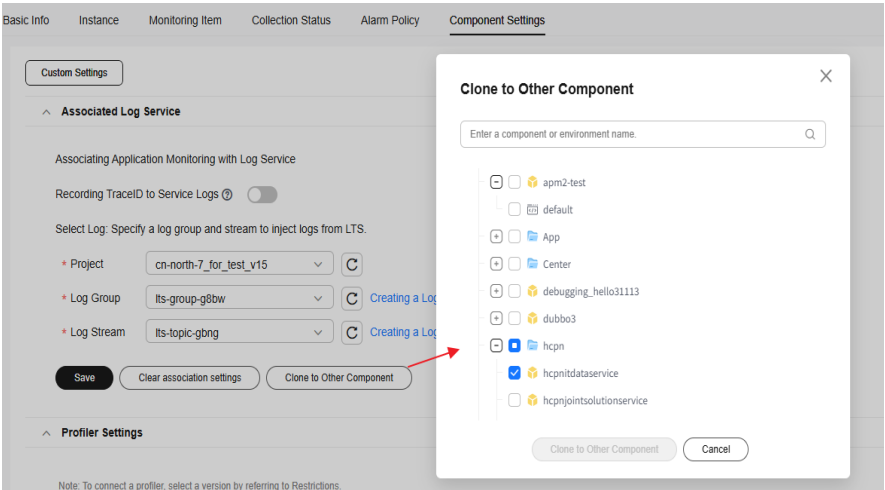
Figure 4-41 Saving the log association setting



Step 3 In the displayed dialog box, click **Confirm**.

Step 4 Perform other operations if needed.

- Cloning the setting to other components
 - a. Click **Clone to Other Component**.
 - b. On the displayed page, select one or more components and click **Clone to Other Component**. The current log association setting will then be copied to the components you have selected.
- Clearing the association setting
 - a. Click **Clear Association Settings**. The **Clear Association Settings** dialog box is displayed.
 - b. On the displayed page, select one or more components and click **Clear Association Settings**. The current log association setting will then be cleared for the components you have selected.
- Before you click **Clone to Other Component** or **Clear Association Settings**, ensure that the components you selected contain environments. Otherwise, the button will be grayed and your settings cannot be cloned.



----End

Profiler Setting

Step 1 Under **Profiler Settings**, enter related information.

Figure 4-42 Profiler setting

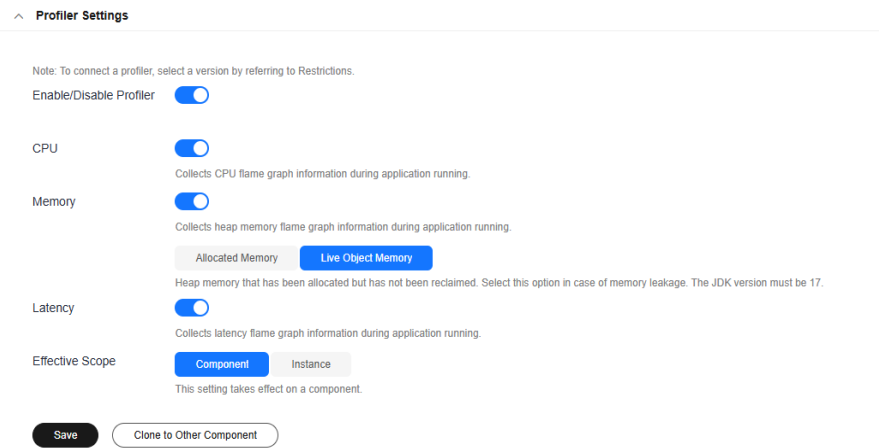
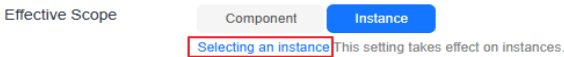


Table 4-42 Profiler setting

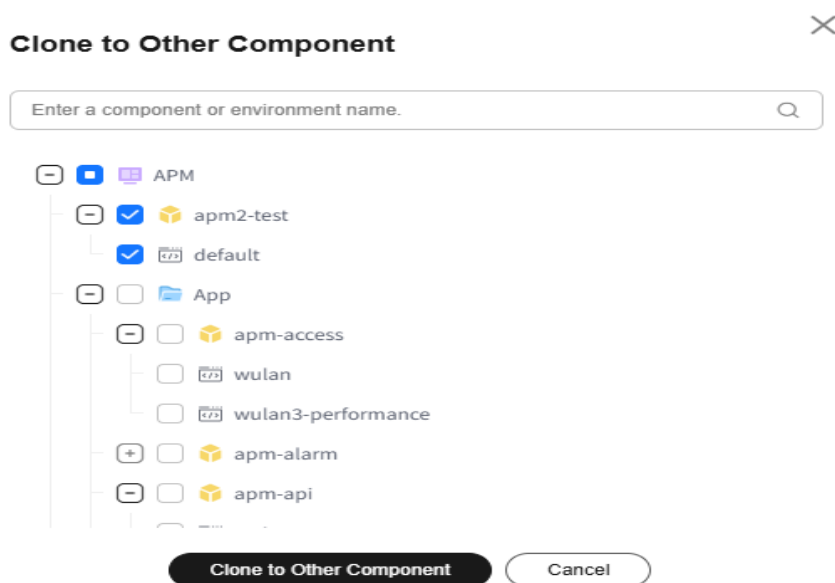
Parameter	Description
Enable/Disable Profiler	If this option is enabled, APM will diagnose applications to solve performance bottlenecks caused by high CPU/memory usage or latency. This option is disabled () by default. If this option is disabled, other options will also be disabled.

Parameter	Description
CPU	Collects CPU flame graph information during application running. This option is disabled () by default.
Memory	Collects heap memory flame graph information during application running. This option is disabled () by default.
Latency	Collects latency flame graph information during application running. This option is disabled () by default.
Effective Scope	- This setting takes effect in a component. Profiler data is generated for all instances associated with the component. This option is enabled (blue toggle) by default. - Instance: This setting takes effect in instances. Profiler data is generated only for the selected instance. By default, this option is disabled (). - Click Instance. The instance button is in blue. - Click Select Instance to select required instances. Figure 4-43 Selecting instances  - Select one or more instances and click Confirm. If Effective Scope is set to Instance, the settings cannot be cloned to other components or environments. Figure 4-44 Selecting required instances  Component and Instance cannot be selected at the same time.

Step 2 Click **Save**.

Step 3 In the displayed dialog box, click **Confirm**.

Step 4 Click **Clone to Other Component**.

Figure 4-45 Cloning the setting to other components

Step 5 On the displayed page, select one or more components and click **Clone to Other Component**. The current Profiler setting will then be copied to the components you have selected.

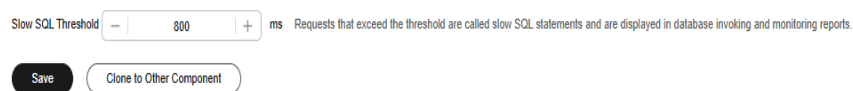
-----End

Database Monitoring Setting

You can define a slow SQL threshold. Requests that exceed this threshold are regarded as slow and will be displayed on the URL page and in monitoring reports.

Step 1 On the **Component Settings** page, set **Slow SQL Threshold** under **Database Monitoring Setting**.

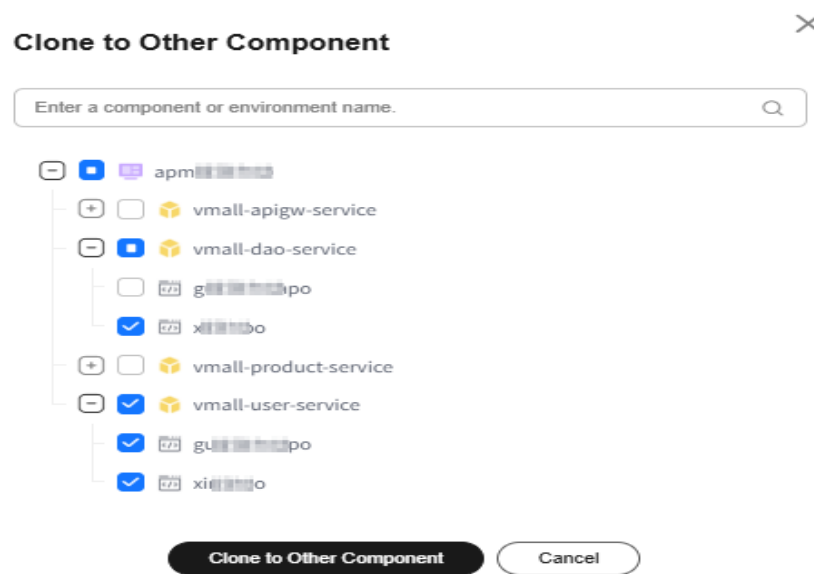
- Value range: [100, 20000]. The default value is the same as the SQL threshold configured in the system.

Figure 4-46 Database monitoring setting

Step 2 Click **Save**.

Step 3 Click **Clone to Other Component**.

Figure 4-47 Cloning the setting to other components



Step 4 On the displayed page, select one or more components and click **Clone to Other Component**. The current database monitoring setting will then be copied to the components you have selected.

----End

5 Tracing

When the calls between enterprise microservices are complex, APM Agents sample some requests, and intercept corresponding requests and subsequent call information. For example, in the scenario where service A calls service B and then calls service C, after service A receives a request, APM determines whether to trace the request based on the intelligent sampling algorithm.

Intelligent Sampling Algorithm

APM uses the intelligent sampling algorithm to determine whether to trace requests.

- If a request needs to be traced, a trace ID is generated and details (events) about some important methods (generally the tree structure with the parent-child relationship) under service A are intercepted. At the same time, the trace ID is transparently transmitted to service B. The important methods under service B are also intercepted. The trace ID is also transparently transmitted to service C. Some methods under service C are intercepted in a similar way as those under services B and A. Each node respectively reports event information and an association relationship can be formed based on the trace ID. In this way, you can view the call details of the entire request based on the trace ID.
- If a request does not need to be traced, no trace ID is generated. Service B does not receive the trace ID and uses the same algorithm as service A to determine whether to perform tracing.
- After data is reported, APM stores not only all event details, but also the root event (called span) information of each service for subsequent trace search. Generally, you search for the span information and then obtain the overall trace details based on the trace ID in the span information.
- By default, the intelligent sampling policy is used. There are three types of URLs: error URLs, slow URLs (use the default 800 ms or customize a threshold), and normal URLs. The sampling ratio of each type of URL is calculated separately. For APM, statistics are collected and reported every minute. In the first collection period, all URLs are regarded as normal for sampling. In the second collection period, URLs are classified into error, slow, and normal URLs based on the statistics collected in the previous period.
 - Sampling rate of error URLs: If the CPU usage is less than 30%, 100 records are collected per minute. If the CPU usage is greater than or

equal to 30% but less than 60%, 50 records are collected per minute. If the CPU usage is greater than or equal to 60%, 10 records are collected per minute. At least two records are collected for each URL.

- Sampling rate of slow URLs: If the CPU usage is less than 30%, 100 records are collected per minute. If the CPU usage is greater than or equal to 30% but less than 60%, 50 records are collected per minute. If the CPU usage is greater than or equal to 60%, 10 records are collected per minute. At least two records are collected for each URL.
- Sampling rate of normal URLs: If the CPU usage is less than 30%, 20 records are collected per minute. If the CPU usage is greater than or equal to 30% but less than 60%, 10 records are collected per minute. If the CPU usage is greater than or equal to 60%, 5 records are collected per minute. At least one record is collected for each URL.

The advantage of the preceding algorithm is that once the trace information is generated, the link is complete, helping you make correct decisions. If a large number of URLs are called, abnormal requests may fail to be collected. In this case, you can collect metrics to locate system exceptions.

Trace Search

This function is used to search for span information, that is, the root event of a node. A trace can be found in multiple environments. For example, in the scenario where service A calls service B and then calls service C, the same trace may be found from services A, B, and C.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Tracing**.

Step 4 Select a time range in the upper right corner of the page. Default: **Last 20 minutes**. Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, or **Custom**.

Step 5 Specify the following search criteria or add custom criteria to query traces.

Table 5-1 Search criteria of traces

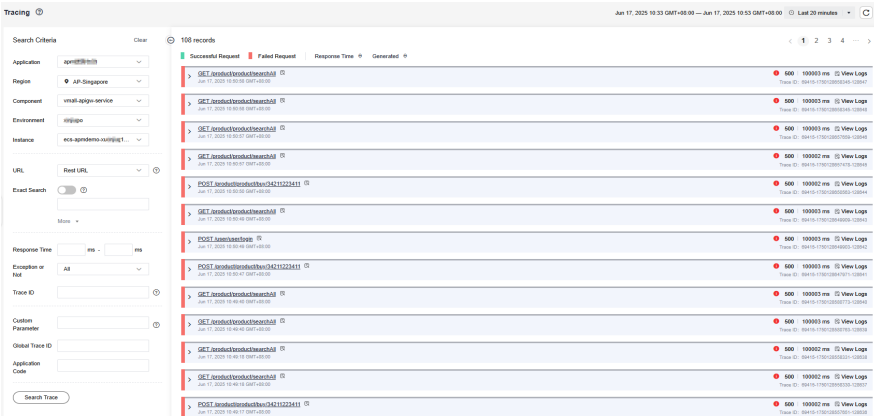
Search Criterion	Description	Mandatory
Application	Application to which the trace belongs.	Yes
Region	Region where the trace is located.	Yes
Component	Component to which the trace belongs.	No
Environment	Environment to which the trace belongs.	No
Instance	Instance to which the trace belongs.	No

Search Criterion	Description	Mandatory
URL	Trace URL, which can be a REST URL or real URL. A REST URL contains a variable name, for example, <code>/apm/get/{id}</code> . A real URL indicates an actual URL.	No
Exact Search	Whether to perform exact match on URLs. If this option is selected, exact match is performed. If this option is not selected, fuzzy match is performed.	No
Call Method	HTTP method of the trace.	No
Status Code	HTTP status code returned by the trace.	No
Response Time	Response time range of the trace. You can specify the minimum and maximum response time to search for the trace or leave them empty.	No
Exception or Not	Whether to filter the traces that are regarded as exceptions.	No
Trace ID	If you specify this parameter, other search criteria become invalid and the search will be performed based on the trace ID you specify.	No
Custom Parameter	- Custom search is supported for otel.scope.name and thread.id only. For more, contact O&M personnel. - Search for traces by call parameter. The format is key=value. Example: exceptionMsg=failed. - Configure required parameters before you search for traces by custom parameter. For example, if you have configured Key for Header Value Interception, Key for Parameter Value Interception, and Key for Cookie Value Interception for URL monitoring, you can set key=value to search, for example, httpMethod=POST. For details about how to configure URL monitoring, see Configuring the URL Monitoring Item.	No
Global Trace ID	Global ID of a trace. If you specify this parameter, other search criteria become invalid and the search will be performed based on the trace ID you specify.	No

Search Criterion	Description	Mandatory
Application Code	If you have configured Service Code Length, Key for Service Code Interception, and Normal Service Code , corresponding application codes will be collected. You can search information based on application codes. Generally, the value of Application Code is the same as the value of Normal Service Code . For details about how to configure URL monitoring, see Configuring the URL Monitoring Item .	No

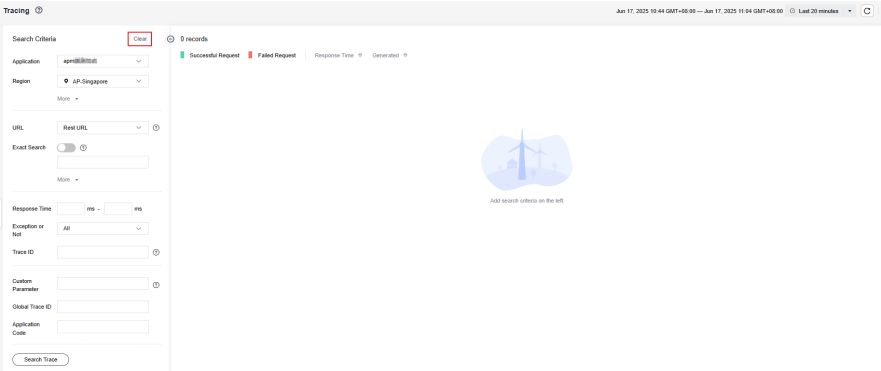
Step 6 Click **Search Trace** to filter the traces that meet search criteria.

Figure 5-1 Viewing traces



- Only the traces of successful or failed requests can be displayed. The following uses **Successful Request** as an example.
 - Click the green button next to **Successful Request**, only the traces of successful requests are displayed. The red button next to **Failed Request** becomes grayed.
 - If you click the green button again, both the traces of successful and failed requests are displayed. The red button is no longer grayed.
 - The green and red buttons cannot be grayed at the same time.
- Response Time**: response time of a trace. Click **Response Time**. Traces are sorted by response time in descending order, and **Response Time** is displayed. Click **Response Time** again. Traces are sorted by response time in ascending order, and **Response Time** is displayed. Click **Response Time** for the third time. The trace list is restored to the initial state and **Response Time** is displayed.
- Generated**: start time of a trace. You can sort traces by generation time. The sorting method is similar to that of **Response Time**.
- You can click **Clear** to clear the search criteria and results.

Figure 5-2 Performing clearance



You can also click a specific URL on the monitoring item view page, for example, the table view of the URL monitoring item. In this way, you can quickly search for required trace information based on preset search criteria.

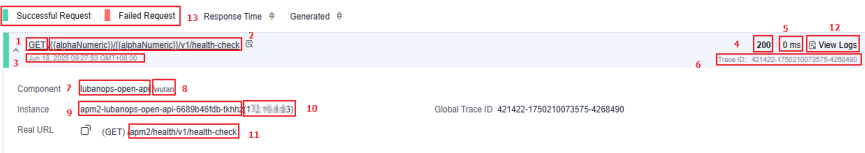
----End

Checking Trace Details

Viewing Basic Information About the Trace Filtered Based on the Search Criteria

In the displayed trace list, click  next to the target trace to view its basic information, as shown in the following figure.

Figure 5-3 Basic information about a trace



Parameter description:

- 1. HTTP method of the trace.
- 2. REST URL of the trace. A REST URL contains a variable name, for example, `/apm/get/{id}`. You can click the URL to go to the trace details page.
- 3. Start time of the trace.
- 4. HTTP status code returned by the trace.
- 5. Response time of the trace.
- 6. Trace ID.
- 7. Component to which the trace belongs.
- 8. Environment to which the trace belongs.
- 9. Host of the instance to which the trace belongs.
- 10. IP address of the instance to which the trace belongs.
- 11. Actual URL of the trace.

12. View logs.

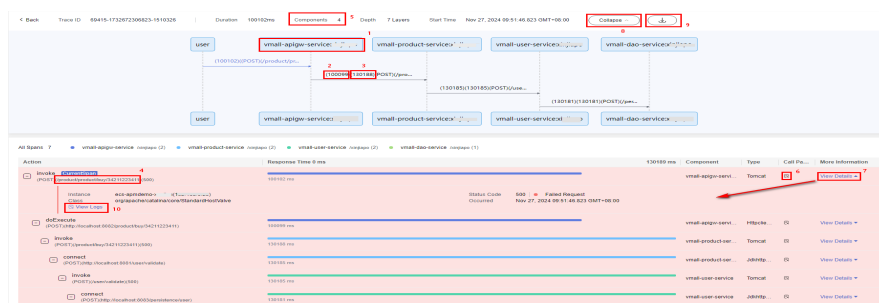
- On the LTS console, configure collection. For details, see [section "Ingesting ECS Text Logs to LTS" in the Log Tank Service \(LTS\) User Guide](#).
- Click **View Logs** to go to the LTS page based on the trace ID. For details about how to search for logs on the LTS console, see [Searching for Logs](#).
- If the function of associating trace IDs with logs is not enabled, a warning dialog box is displayed. Click **Associate Now** to go to the log association settings page. For details about how to associate logs with trace IDs, see [Component Settings](#).

Viewing the Complete Information About the Trace, Including Local Method Stacks and Remote Call Relationships

Click the name of a trace to view its details, as shown in the following figure.

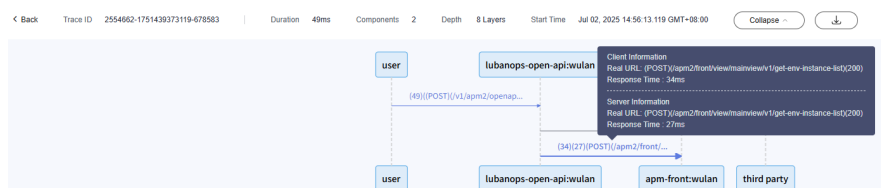
- The upper part is the sequence diagram of the trace, which shows complete call relationships between components. This diagram contains the information about the client and server corresponding to each call. The lower the line is, the later a call occurs.
- The lower part lists the method stack details of the trace. Each line indicates a method call. You can view the detailed method call relationships of the trace. By default, only component methods supported by JavaAgents are displayed. To display application methods, configure the application methods to be intercepted during JavaMethod configuration.

Figure 5-4 Call relationship



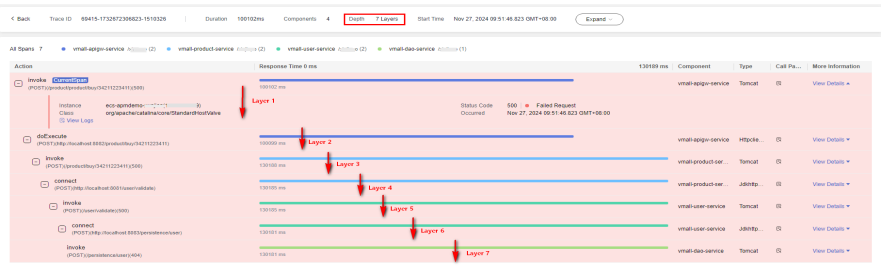
Parameter description:

- Component and environment to which the called API belongs
- Response time (unit: ms) of the client. You can hover the pointer over the line between components to view the client and server information.
 - Real URL: REST URL of the trace. A REST URL contains a variable name, for example, `/apm/get/{id}`.
 - Response time: response time of the trace.



3. Response time (unit: ms) of the server.
4. Key parameter of the method in the trace method stack. For example, for a Tomcat entry method, a real URL is displayed. For a MySQL call method, an executed SQL statement is displayed.
5. Number of components that are involved.
6. Call parameters. Click  to view the call parameters.
7. Click **View Details**. The span details are displayed.
8. Click  to download the sequence diagram.
9. Click **View Logs** to go to the LTS page based on the trace ID. For details, see [Viewing Logs](#).
10. Traces are displayed in a tree. **Depth** indicates the number of layers in the tree.

Figure 5-5 Depth

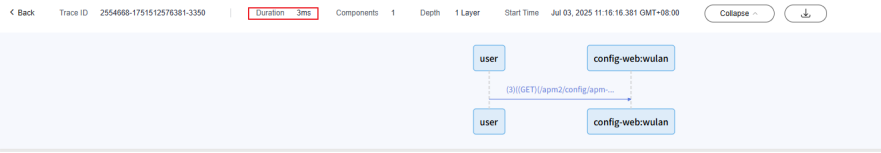


11. The content displayed in positions 1 and 2 in the sequence diagram of a trace varies depending on the Agent type or connected website/app.

Agent Type or Connected Website/App	Display Content
Web	WEB
WeChat	WE_CHAT
Baidu	BAIDU
Alipay	ALIPAY
DingTalk	DING_TALK
Mobile app (Android)	ANDROID-APP
Mobile app (iOS)	IOS-APP
Mobile app (HarmonyOS)	HARMONY-APP
Others	user

12. Duration, that is, the total time of a trace request.

Figure 5-6 Duration



6 Application Topology

On the tracing page, you can view the topology of a single call, as well as the overall topology between different services based on collected metric data.

Each line in the topology indicates the call relationship between services within a period. The statistics can be collected from the calling or called party. You can click a line to view the call trend on the right. The topology can also display the call relationships between middleware. On the topology, you can view the call relationships between services and check whether the calls between services are normal to quickly locate faults.

6.1 Viewing a Component Topology

A component topology means the topology of a single component under an environment. You can also view the call relationships of direct and indirect upstream and downstream components.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Topology** tab to view the call and dependency relationships of the component. For details, see [Topology](#).

----End

6.2 Viewing a Global Application Topology

A global application topology means the topology of some or all components under an application.

Step 1 Log in to the [APM console](#).

- Step 2** Click  on the left and choose **Application > Application Performance Management**.
- Step 3** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4** In the tree on the left, click an application. The application details page is displayed.
- Step 5** Click the **Global Topology** tab to view the call and dependency relationships of all components under the application.
- Click a line between components. The detailed data such as the number of requests, average RT, and error rate is displayed on the right.
 - Enable **Third-party invocation** to view non-application internal call relationships.
 - **Alarm Latency**: can be 0.5s, 1s, 3s, 5s, or 10s.
 - **Error Latency**: can be 1s, 3s, 5s, 10s, or 15s.
 - Use tags to filter calls.
 - Click  to obtain the latest information in the statistical period.
 - Click  to download the topology.
- Step 6** Right-click a component and then click **Viewing a Call Chain** or **Viewing Indicators**. For details, see [Step 7](#).
- Step 7** Click a line in the diagram. The detailed data such as the number of requests, average RT, and error rate, and line graphs are displayed on the right.
- Click the drop-down list in the upper left corner of the line graph and select an instance.
 - Click  in the upper right corner of the line graph. All instance views corresponding to a parameter are displayed.
 - The diagram shows the call relationships between components and enables you to check the traces between components.
 - The function of redirecting to the trace page is supported only by Agent 2.5.3 or later.
 - Click **Viewing a Call Chain**. The [trace](#) page is displayed, showing the trace information between components.

----End

7 URL Tracing

You can view the topology of a single call, as well as the overall topology between different services. In some scenarios, the call relationships of an important business need to be traced. This process is called URL tracing. For example, to trace the API for creating online shopping orders. In APM, URL tracing consumes a large number of resources. Therefore, an entry URL will not be added for tracing by default. However, you can set that if necessary. APM has a limit on the total number of URLs added for tracing. It focuses on tracing the downstream calls for the APIs that are added for tracing. Through URL tracing, you can monitor the call relationships between important APIs and downstream services, and then detect problems more precisely.

Configuring URL Tracing

Step 1 Log in to the [APM console](#).

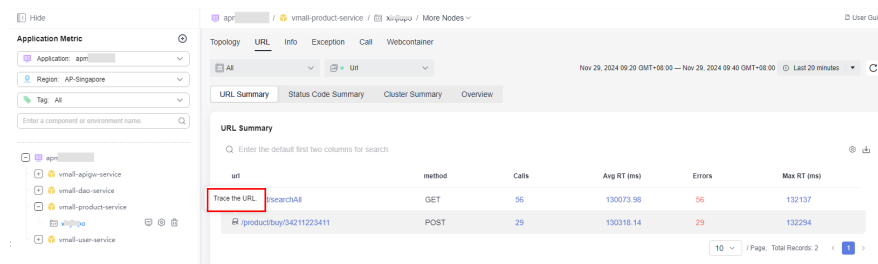
Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the tree on the left, click the environment that needs URL tracing. The environment details page is displayed. By default, the **URL** tab page is displayed.

Step 5 Move the mouse pointer to the target URL, click , and add it for URL tracing.

Figure 7-1 Configuring URL tracing



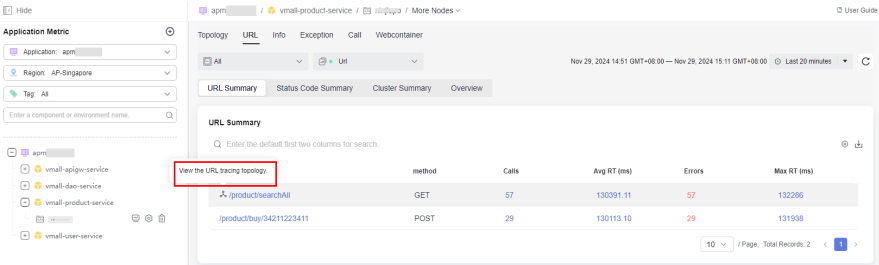
----End

Checking the URL Tracing View

- On the **URL** tab page:

For the URL added for tracing, click  next to it to view its topology.

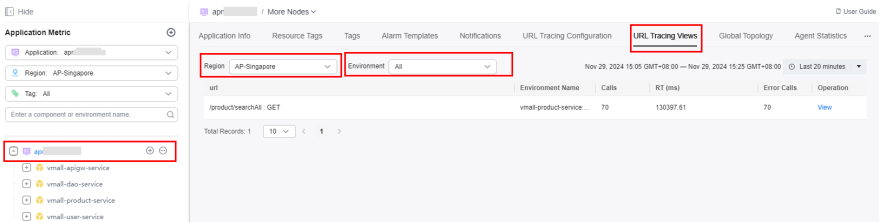
Figure 7-2 Viewing URL tracing details



- On the **URL Tracing Views** tab page:

- Step 1** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 2** In the tree on the left, click an application. The application details page is displayed.
- Step 3** Click the **URL Tracing Views** tab to check all URL tracing views of the application.
- Step 4** Filter transaction views by region and environment.
- Step 5** Click **View** in the **Operation** column of the row that contains the URL you want to view.

Figure 7-3 Checking the URL tracing view



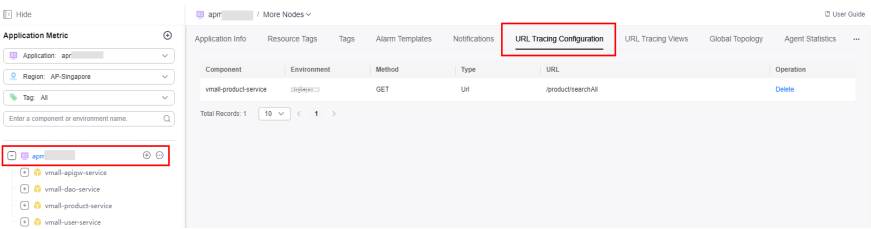
----End

Viewing the URL Tracing Configuration

The URL which has been added for tracing will be displayed in the URL tracing configuration list.

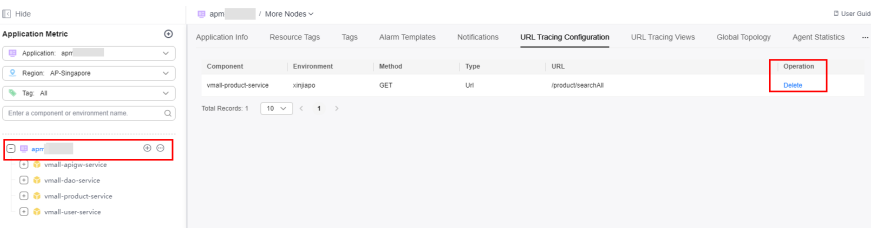
- Step 1** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 2** In the tree on the left, click an application. The application details page is displayed.
- Step 3** Click the **URL Tracing Configuration** tab to check all URL tracing configurations of the application.

Figure 7-4 Viewing the URL tracing configuration list



Step 4 To delete a URL tracing configuration, click **Delete** in the **Operation** column.

Figure 7-5 Deleting a URL tracing configuration



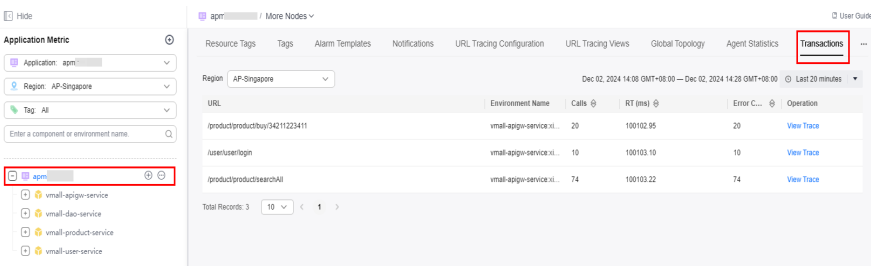
----End

Viewing Transactions

Transaction URLs are displayed in a list. By default, the system displays the invocation of all entries.

- Step 1**
- In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 2**
- In the tree on the left, click an application. The application details page is displayed.
- Step 3**
- Click the **Transactions** tab to view all transactions of the application.

Figure 7-6 Viewing transactions



Step 4 Click **View the call chain** in the **Operation** column of the target transaction. For operations related to call chains, see [Tracing](#).

Figure 7-7 Viewing the call chain

apn / More Nodes

User Guide

Resource TagsTagsAlarm TemplatesNotificationsURL Tracing ConfigurationURL Tracing ViewsGlobal TopologyAgent StatisticsTransactions

Region: AP-SingaporeDec 02, 2024 14:11 GMT+08:00 — Dec 02, 2024 14:31 GMT+08:00Last 20 minutes

URL	Environment Name	Calls	RT (ms)	Error C...	Operation
/product/productbuy/34211223411	vmali-apigw-service.xi...	20	100102.99	20	View Trace
/user/userlogin	vmali-apigw-service.xi...	10	100103.00	10	View Trace
/product/productsearchAll	vmali-apigw-service.xi...	68	100103.15	68	View Trace

Total Records: 310<1>

-----End

8 Resource Tag Management

You can tag resources under your account for classification. This section describes how to use tags to query resources and how to modify and delete tags.

Resource tag management is related to [Tag Management Service](#), [Cost Center](#), and [Billing Center](#).

Constraints

1. Each application supports up to 20 tags.
2. It is recommended that you use the TMS predefined tag function to add the same tag to different cloud resources. For details, see [Creating Predefined Tags](#).

Checking Resource Tags

Step 1 Log in to the [APM console](#).

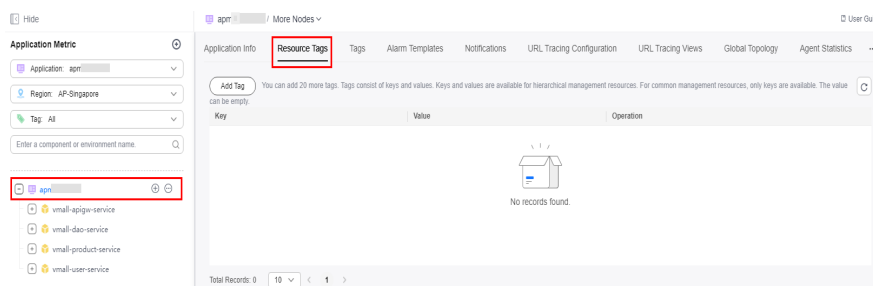
Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the navigation tree on the left, click a target application and click the **Resource Tag Management** tab.

Step 5 Checking the tag list of the current application, as shown in the following figure.

Figure 8-1 Checking the tag list



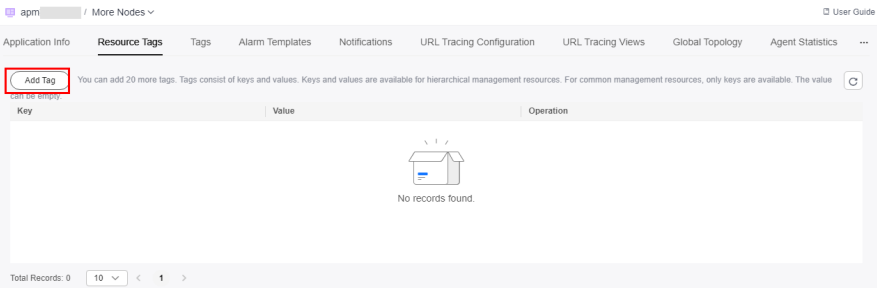
-----End

Adding Resource Tags

To add a tag with the same key to all resources in the search result list, click **Add Tag**.

- Step 1
- In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 2
- In the navigation tree on the left, click the application to which you want to add a tag and choose **Resource Tag Management > Add Tag**.

Figure 8-2 Adding a tag



- Step 3
- Set tag parameters.

Figure 8-3 Setting tag parameters

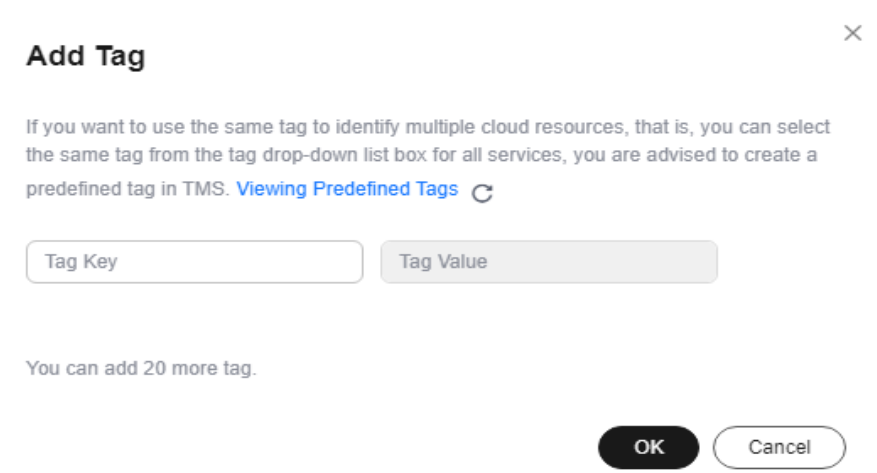


Table 8-1 Tag parameters

Parameter	Description
Tag Key	- The tag key cannot be empty or start or end with a space. - Enter 1 to 128 characters. Only letters, digits, spaces, and special characters (_ . : = + - @) are allowed. - Each tag key must be unique.

Parameter	Description
Tag Value	- Enter up to 255 characters. Only letters, digits, spaces, and special characters (_.:=-@) are allowed. - The resource tag value can be empty, but the predefined tag value cannot be empty.

Step 4 Click **OK**.

----End

Editing Resource Tags

When you modify a tag, the modification applies only to the cloud resources that contain this tag. To modify a tag, perform the following steps:

Step 1 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 2 In the navigation tree on the left, click a target application and click the **Resource Tag Management** tab.

Step 3 Click **Edit** in the **Operation** column to modify the tag content, as shown in the following figure.

Step 4 Click **OK**.

----End

Deleting Resource Tags

Step 1 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 2 In the navigation tree on the left, click a target application and click the **Resource Tag Management** tab.

Step 3 Click **Remove** in the **Operation** column to delete the target tag, as shown in the following figure.

Step 4 Click **Yes**.

----End

9 Managing Tags

You can add tags for different environments and applications for easy management.

Tag management covers tags and global tags.

- A tag is used to set a collector corresponding to the monitoring item under one or more environments of an application.
- A global tag is used to set a collector corresponding to the monitoring item under all environments of an application.
- Priority: Global tag collector configuration > Tag collector configuration > Collector configuration of a monitoring item under an environment

Adding a Tag

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

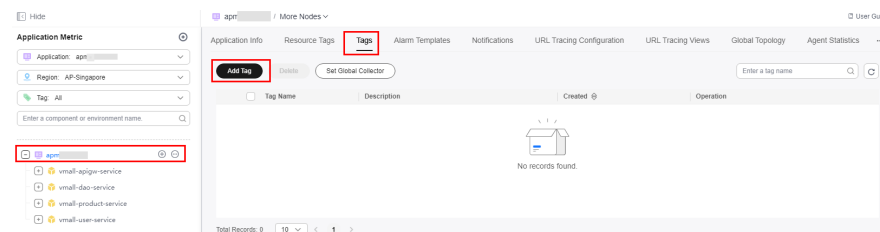
Step 3 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 4 In the navigation tree, select a target application.

Step 5 Click the **Tags** tab.

Step 6 Click **Add Tag**.

Figure 9-1 Adding a tag



Step 7 On the page that is displayed, set **Tag** and **Description**, and select the environment to be associated.

Table 9-1 Tag parameters

Parameter	Description
Tag	Enter 1 to 128 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Description	Enter up to 1,000 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Bind Environment	• Search by component, environment, or application name is supported. • You can select one or more environments.

Step 8 Click **Yes**.

----End

Modifying a Tag

Step 1 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 2 In the navigation tree, select a target application.

Step 3 Click the **Tags** tab.

Step 4 Locate the row that contains the tag to be modified and click **Collector Configuration** in the **Operation** column. In the dialog box that is displayed, select your desired collector from the drop-down list and click **Yes**.

Locate the row that contains the tag to be modified and click **Change Environment** in the **Operation** column. In the dialog box that is displayed, select your desired environment and click **Yes**.

Locate the row that contains the tag to be modified and click **Modify Tag** in the **Operation** column. In the dialog box that is displayed, modify the tag and description.

----End

Deleting a Tag

Step 1 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 2 In the navigation tree, select a target application.

Step 3 Click the **Tags** tab.

Step 4 Locate the row that contains the target tag and click **Delete** in the **Operation** column. Alternatively, select the tags to delete and click **Delete**.

Step 5 In the dialog box that is displayed, click **Yes**.

----End

Global Tag Collector Configuration

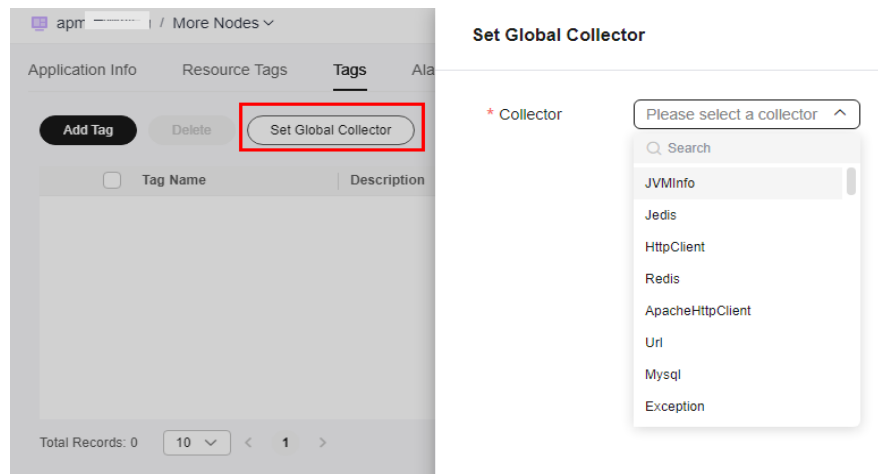
Step 1 In the navigation pane, choose **Application Monitoring > Metrics**.

Step 2 In the navigation tree, select a target application.

Step 3 Click the **Tags** tab.

Step 4 Click **Global tag collector configuration**.

Figure 9-2 Global tag collector configuration



Step 5 Select a collector from the drop-down list and click **Yes**. For details about how to configure monitoring items, see [Application Monitoring Configuration](#).

----End

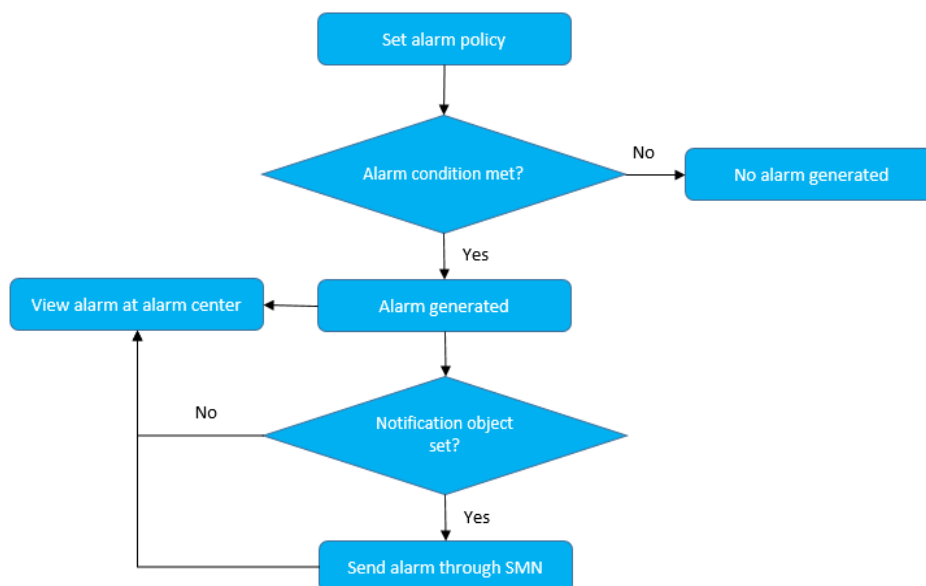
10 Alarm Management

10.1 Alarm List

Alarms are reported by services connected to APM Agents when specified conditions are met. You can learn about service exceptions in a timely manner and quickly rectify faults to prevent service loss.

Alarm process

Figure 10-1 Alarm process



Viewing Alarms

Step 1 Log in to the [APM console](#).

- Step 2** Click  on the left and choose **Application > Application Performance Management**.
- Step 3** In the navigation pane, choose **Alarm Center > Alarm List**.
- Step 4** View alarms on the **Alarm List** page.
1. Select an application from the application drop-down list to view its alarms.
 2. In the search text box, set search criteria, and click  to view the alarms that meet the criteria.
 3. Click  next to **Alarm Status** to filter alarms by alarm status.
 4. Click  to export the alarm list on the current page. A maximum of 100 alarms can be exported.
 5. When necessary, enable **Scheduled Refresh**. In that case, the alarm list is refreshed every 5s.
 6. Click  in the **Operation** column to view the alarm details and notification.
- End

10.2 Alarm Policies

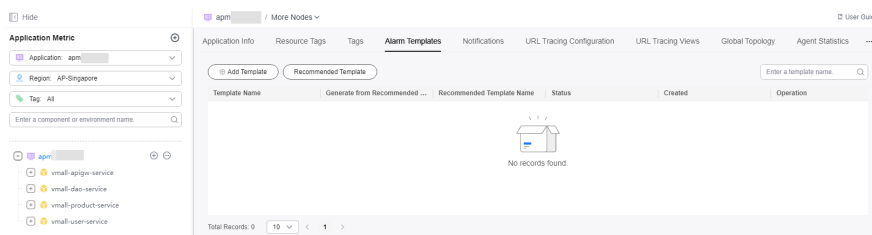
When you configure alarm rules for applications, you may have to perform the same operations multiple times. To solve this problem, APM supports the function of copying alarm rules across applications. You can copy the alarm rule of one application to other applications, which improves the configuration efficiency and consistency. The following will focus on using this function to reduce workload.

10.2.1 Configuring an Alarm Template

APM allows you to configure alarm templates. You can create multiple alarm policies under a template and bind them to nodes.

Procedure

- Step 1** Log in to the [APM console](#).
- Step 2** Click  on the left and choose **Application > Application Performance Management**.
- Step 3** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4** In the tree on the left, click an application. The metric details page of the application is displayed.
- Step 5** Click the **Alarm Templates** tab.

Figure 10-2 Adding an alarm template**Step 6** Click **Add Template** to add an alarm template as prompted.

1. Enter basic information and then click **Next**.

Template Name: Enter up to 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

Remarks: Enter up to 512 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

2. Click **Add Alarm Policy** to add an alarm policy.
 - a. Basic information

Table 10-1 Basic information about an alarm policy

Parameter	Description
Policy Name	Custom name, which cannot be left blank. Only letters, digits, underscores (_), and hyphens (-) are allowed. Enter up to 512 characters.
Alarm Severity	Severity of an alarm. Options: COMMON and CRITICAL .
Alarm Policy Type	Options: Single-node and Aggregate . Single-node indicates single-instance metric alarms, and Aggregate indicates aggregated metric alarms of all instances under an environment.
Monitoring Item	Select a target monitoring item. The information about the selected item is displayed on the right.

Parameter	Description																														
Metric Set	Select a target metric set. The information about the selected metric set is displayed on the right. Basic Info Policy Name test_dbc Alarm Severity COMMON Alarm Policy Type Single-node Aggregate Monitoring Item ApacheHttpClient Metric Set Connection Pool Alarm Rule Dimension Metric <table><tr><th>Condition</th><th>Metric</th><th>Operator</th><th>Threshold</th><th>Operation</th></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> Variable Preview <table><tr><th>Name</th><th>Description</th></tr><tr><td>hostInfo</td><td>Host info</td></tr><tr><td>envPath</td><td>Environmen...</td></tr><tr><td>region</td><td>Region</td></tr><tr><td>ruleName</td><td>Alarm Polic...</td></tr><tr><td>poolId</td><td>poolId</td></tr><tr><td>available</td><td>Number of ...</td></tr><tr><td>leased</td><td>Number of ...</td></tr><tr><td>max</td><td>Maximum n...</td></tr><tr><td>pending</td><td>Number of ...</td></tr></table>	Condition	Metric	Operator	Threshold	Operation						Name	Description	hostInfo	Host info	envPath	Environmen...	region	Region	ruleName	Alarm Polic...	poolId	poolId	available	Number of ...	leased	Number of ...	max	Maximum n...	pending	Number of ...
Condition	Metric	Operator	Threshold	Operation																											
Name	Description																														
hostInfo	Host info																														
envPath	Environmen...																														
region	Region																														
ruleName	Alarm Polic...																														
poolId	poolId																														
available	Number of ...																														
leased	Number of ...																														
max	Maximum n...																														
pending	Number of ...																														

b. Alarm rule

Figure 10-3 Alarm rule

Dimension ?

+

* Metric

Condition	Metric	Operator	Threshold	Operation
	Select a metric.	Select an opera...	Enter a threshold.	+

* Alarm Condition

In

A

 collection periods, if alarms are triggered

B

 times, alarms will not be repeated for

C

 minutes.

* Recovery Policy

☐ No alarm is generated during the period.

* Notification upon Recovery

☒ Yes ☐ No

Multi-Line Matching ?

☐

* Notification Content

Variable

0/500

Table 10-2 Alarm rule parameters

Parameter	Description
Dimension	(Optional) A category of metrics.
Metric	Metric for which you want to define one or more alarm rules. Metric: a metric in the metric set. For example, if Monitoring Item is set to Url and Metric Set is set to total, you can select the errorCount metric. Operator: operation to be performed. Threshold: threshold of the metric.
Alarm Condition	Condition for triggering an alarm. A: the number of collection periods. Range: 1–10. B: the number of times the alarm is triggered. Range: 1–10. This value cannot be greater than that of A. C: period (in minutes) during which identical alarms will not be sent. This period cannot be shorter than 10 minutes.
Recovery Policy	Condition for clearing an alarm. For example, if this parameter is set to 3 , the alarm status will change to "Cleared" if no alarm is generated within 3 minutes.
Notification upon Recovery	Whether to notify recipients of alarm clearance.
Multi-Line Matching	(Optional) Whether to define data in the alarm notification content line by line.
Notification Content	- If Multi-Line Matching is enabled, the alarm notification content supports both Variable and Loop. If Multi-Line Matching is disabled, only Variable can be selected. - Alarm notification content. You can customize the content or select required metrics. - Alarm details, which contain up to 500 characters. - Select required metrics. Specifically, on the right of the page, click  next to the target metric. The metric will then be displayed in the notification content. 

c. Notification object

Figure 10-4 Notification object

Notification Object

Notification Object Use default settings ☐

AP-Singapore 

CN South-Guangzhou 

AP-Hong-Kong 

Table 10-3 Alarm notification parameters

Parameter	Description
Use default settings	– If this option is enabled, alarms will be sent to one or more default notification objects. For details about how to set notification objects, see Alarm Notification. – After this option is enabled, you can no longer select notification objects from the Notification Object drop-down list. – If all the values in the Default column of the Notifications page are No, no default notification objects have been set. In this case, the Use default settings option on the alarm policy creation page is grayed and cannot be enabled.
Notification Object	– Select a notification object from the drop-down list. – Alarms will then be sent to the selected notification object. Unselected notification objects will not receive the alarm notifications. – Click  to refresh notification objects.

3. Click **Yes**.
4. Click **Next**. The **Bind Node** page is displayed.
5. Click **Bind Node** to bind nodes by environment, environment tag, or region.

Table 10-4 Node parameters

Parameter	Description
All	All nodes (including those added subsequently) in all regions will be bound.
Environment	All nodes in the selected environment will be bound.

Parameter	Description
Environment Tag	All nodes with the same tag will be bound.
Region	All nodes in the selected region will be bound.

Step 7 Click **Yes**. The information about the bound nodes is displayed in the lower part of the page, including the environment, tag, and region names.

If there are multiple types of bound nodes, only information about the nodes with the same environment, tag, and region is displayed.

Step 8 Click **Complete**.

-----End

More Operations

After the alarm template is created, perform the operations listed in [Table 10-5](#) if needed.

Table 10-5 Related operations

Operation	Description
Copying a template	Click Copy in the Operation column in the row that contains the template you want to copy.
Modifying a template	Click Modify in the Operation column in the row that contains the template you want to modify.
Deleting a template	Click Delete in the Operation column in the row that contains the template you want to delete.
Starting and stopping a template	Turn on or off the button () in the Operation column in the row that contains the template you want to start or stop.

10.2.2 Creating a Custom Alarm Policy

You can create a custom alarm policy for a single component.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.

Step 5 Click the **Alarm Policy** tab.

Step 6 Click **Add Custom Alarm Policy** and set the alarm condition in the same way as that when you create an alarm template. For details, see [6.2](#).

----End

Create an Alarm Policy Based on a Template

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.

Step 3 Click the **Alarm Policy** tab.

Step 4 In the template list, click **Copy** in the **Operation** column in the row that contains the template you want to copy.

----End

More Operations

After the alarm policy is created, perform the operations listed in [Table 10-6](#) if needed.

Table 10-6 Related operations

Operation	Description
Starting or stopping a policy	In the custom alarm policy list, start ( in the Operation column) or stop the target policy.
Modifying a policy	Click Edit in the Operation column in the row that contains the policy you want to modify.
Deleting a policy	Click Delete in the Operation column in the row that contains the policy you want to delete.

10.2.3 Recommended Alarm Templates

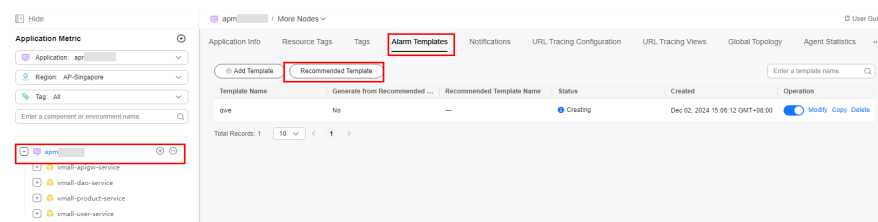
APM provides recommended alarm templates.

Using Recommended Alarm Templates

Step 1 Log in to the [APM console](#).

- Step 2** Click  on the left and choose **Application > Application Performance Management**.
- Step 3** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4** In the tree on the left, click an application. The metric details page of the application is displayed.
- Step 5** Choose **Alarm Templates > Recommendation Template** to view the configured alarm templates.

Figure 10-5 Viewing recommended alarm templates



- Step 6** Click **View Details** in the **Operation** column in the row that contains the target alarm template.
- Step 7** Click **Copy** to copy the recommended template to the template list. You can customize the template name as required.
- Step 8** Click **Yes**. The copied alarm template is displayed on the template list.
- Step 9** Click **Modify** in the **Operation** column and **bind nodes** to make the copied template to take effect.

----End

10.3 Alarm Notification

Alarm notifications can be sent to specified personnel through emails, SMS, WeCom, DingTalk, Lark, HTTP, or HTTPS so that they can detect and handle faults in a timely manner, preventing loss. Ensure that you have the SMN permission. For details, see [Permissions Management](#).

- If you do not create any notification object, no alarm notifications will be received. To view alarms, log in to the APM console and choose **Alarm Center > Alarm List** in the navigation pane.

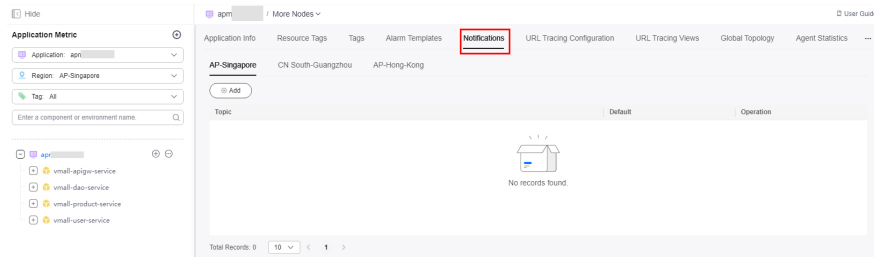
Creating a Notification Object

- Step 1** Log in to the [APM console](#).
- Step 2** Click  on the left and choose **Application > Application Performance Management**.
- Step 3** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 4** In the tree on the left, click an application. The metric details page of the application is displayed.

Step 5 Click the **Notifications** tab.

Step 6 Click **Add**.

Figure 10-6 Creating a notification object



Step 7 On the displayed page, specify **Region** and **Topic**, and determine whether to enable default notification. If it is enabled, alarm notifications will be sent based on the topic and region you specify.

- If no topic is available, **create one**.
- If default notification is enabled, alarms will be sent to the specified region when you create an alarm policy.

Step 8 Click **Yes**.

-----End

11 Agent Management

11.1 Overview

APM Agents use the bytecode enhancement technology to collect application performance data in real time. They run on the server where applications are deployed.

Install Agents before using APM. For details, see section "Manually Installing Agents for Java Applications" in *APM Getting Started*.

APM can count the number of Agents used by tenants. For details, see [Usage Statistics](#).

11.2 Agent Download Addresses

Constraints

- Only JDK 8, JDK 11, and JDK 17 are supported.
- In the JDK 8 environment, select a version not ended with `-jdkxx`. In other JDK environments, select a version ended with `-jdkxx`. If profiling is required, select a version ended with `-profiler`.
- For versions ended with `-jdk17` or `-jdk11`, add `-jdk17` or `-jdk11` to the end of the Agent download path.

Example: If the download address for Agent 2.4.12 is `https://apm2-javaagent***/apm2_javaagent/apm-javaagent-2.4.12.tar`, the download address for Agent 2.4.12-jdk17 will be `https://apm2-javaagent***/apm2_javaagent/apm-javaagent-2.4.12-jdk17.tar`.

Region	Latest Version	Earlier Versions					
CN-Hong Kong	2.5.3 sha256:f5b59a4a443dd179df2c24e92f4662660fe0405d62dba4d02121e5f4cf37c200	2.5.2 sha256:8c28c24120951c04a937356e4a5610a18a1f8105199b71c0910c2a79b4b19a1b	2.5.1 sha256:6:3d9a24e42b75c8d184df31c93e9129a00472631a5fccf1f7bf9126732acdeae4	2.4.14 sha256:6:fdb38728342b69f9ccdf e824975d82c1d03d51e33037ce3107c55f0a1cb50c7e	2.4.13 sha256:12a9d2b9aec31e41edce6c27391677f565805a8cc4efe9e0f3c89e9bfcd16eae	2.4.12 sha256:6:d9481030d4481f4c50fa86cee7d6918ea2577ff9fdae9f8457f8b8e48be2e625	2.4.11 sha256:c752cc96fe09f62cb76daa9094eb4dfc936b0f865e7b35ca89a0d6f4b5aec d7e
AP-Singapore	2.5.3 sha256:f5b59a4a443dd179df2c24e92f4662660fe0405d62dba4d02121e5f4cf37c200	2.5.2 sha256:8c28c24120951c04a937356e4a5610a18a1f8105199b71c0910c2a79b4b19a1b	2.5.1 sha256:6:3d9a24e42b75c8d184df31c93e9129a00472631a5fccf1f7bf9126732acdeae4	2.4.14 sha256:6:fdb38728342b69f9ccdf e824975d82c1d03d51e33037ce3107c55f0a1cb50c7e	2.4.13 sha256:12a9d2b9aec31e41edce6c27391677f565805a8cc4efe9e0f3c89e9bfcd16eae	2.4.12 sha256:6:d9481030d4481f4c50fa86cee7d6918ea2577ff9fdae9f8457f8b8e48be2e625	2.4.11 sha256:c752cc96fe09f62cb76daa9094eb4dfc936b0f865e7b35ca89a0d6f4b5aec d7e
CN North-Beijing4	2.5.5 sha256:05ef1049965d99472996f0579d4928fc7e872f0d07c90a5b88ad02802374991a	2.5.4 sha256:8a502bdbb75660bcfff144e4622b753b247a515541ec39a1cb114ad5f7d24cca	2.5.3 sha256:6:f5b59a4a443dd179df2c24e92f24e92f4662660fe0405d62dba4d02121e5f4cf37c200	2.5.2 sha256:6:8c28c24120951c04a937356e4a5610a18a1f8105199b71c0910c2a79b4b19a1b	2.5.1 sha256:3d9a24e42b75c8d10951c04a937356e4a5610a18a1f8105199b71c0910c2a79b4b19a1b	2.4.14 sha256:6:fdb38728342b69f9ccdf e824975d82c1d03d51e33037ce3107c55f0a1cb50c7e	2.4.13 sha256:12a9d2b9aec31e41edce6c27391677f565805a8cc4efe9e0f3c89e9bfcd16eae

Region	Latest Version	Earlier Versions					
LA-Sao Paulo1	2.4.3 sha256:73d e7b49e148 b102b74a6 075c3d8b25 e76f43d614 4ec5aafe6a c190ab3c1e f0b	-	-	-	-	-	-
TR-Istanbul	2.4.8 sha256:889 3e89cd3174 879232704 828027fd63 6471525d6 e5772ff77e 3313f79a4b 6b5	-	-	-	-	-	-
AF-Johannesburg	2.4.8 sha256:029 c57990339f 2a9de418b 4d19e6f734 9106a630b 7eae559abc 9d6d9ea87 76d7	-	-	-	-	-	-
AF-Cairo	2.4.5 sha256:37b 155626f46f 3a8b19772 e3fd6597ec e92bb255fdf fa21e3e9e1 467cc00939 2	-	-	-	-	-	-
AP-Manila	2.4.5 sha256:37b 155626f46f 3a8b19772 e3fd6597ec e92bb255fdf fa21e3e9e1 467cc00939 2	-	-	-	-	-	-

Region	Latest Version	Earlier Versions					
ME-Riyadh	2.4.5 sha256:37b155626f46f3a8b19772e3fd6597ec e92bb255fdf fa21e3e9e1 467cc00939 2	-	-	-	-	-	-

11.3 Agent Access Addresses

Table 11-1 Access addresses of Enhanced Agents

Region	Access Address
LA-Sao Paulo1	https://100.125.11.27:41333
CN-Hong Kong	https://100.125.6.106:41333
AP-Singapore	https://100.125.4.25:41333
TR-Istanbul	https://100.125.8.18:41333
ME-Riyadh	https://100.125.250.45:41333
AF-Johannesburg	https://100.125.255.21:41333
AF-Cairo	https://100.125.251.10:41333
AP-Manila	https://100.125.241.30:41333

11.4 Performing Operations on Agents

Agent Management allows you to check the deployment and running statuses of the Agents that are connected to APM, and to stop, start, or delete them.

Viewing Agents

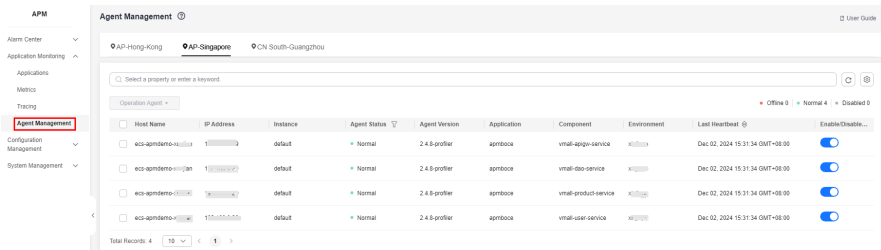
Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring > Agent Management**.

- Step 4** On the Agent management page, view the Agent list.
1. In the upper left corner of the page, select a target region and application.
 2. Set the search criteria and click  in the search box in the upper right corner of the page to filter Agents.

Figure 11-1 Viewing Agents



----End

The following table describes the Agent statuses.

Status	Description
Enabled	The Agent is running properly.
Offline	The Agent is abnormal due to a network error. Check and restore the network.
Disabled	The Agent is manually or globally disabled. Contact technical support.

Batch Operations

- Step 1** In the navigation pane, choose **Application Monitoring > Agent Management**.
- Step 2** Select target objects and click **Operation**.
- Step 3** Select **Disable Agent**, **Enable Agent**, or **Delete Agent**.
- Step 4** In the dialog box that is displayed, click **Yes** to disable, enable, or delete the Agents for the selected hosts.

----End

Checking the Number of Agents Used by an Application

- Step 1** In the navigation pane, choose **Application Monitoring > Metrics**.
- Step 2** On the displayed page, select an application to view. The **Application Info** tab page is displayed by default.
- Step 3** Switch to the **Agent Count** tab page to view the number of Agents used by the current application.
- **Current Agent:** number of Agents used by the current application.

- **Historical Agent:** number of Agents used in each hour of today, yesterday, or a custom day.

----End

11.5 Upgrading Agents

Update Agent versions according to the following procedure.

Upgrading the Manually Installed Agents

To upgrade the manually installed Agent, download the new Agent by referring to [Agent Download Addresses](#). For details, see section "Manually Installing Agents for Java Applications" in *APM Getting Started*.

Upgrading the Agents for Java Applications Deployed in CCE Containers

To upgrade the Agents for the Java applications deployed in CCE containers, select the new version for installation. For details, see section "Installing Agents for the Java Applications Deployed in CCE Containers" in *APM Getting Started*.

Upgrading Agents of Other Types

Install new Agents. For details, see section [Manually Installing Agents for Java Applications](#).

12 Web Monitoring

12.1 Overview

APM Agents regularly collect performance metric data to measure the overall health of browsers, HTML5 pages, and applets.

They collect the data about loading performance, API requests, JS errors, access analysis, custom statistics, session tracing, scenario analysis, and network analysis, enabling you to monitor your websites.

Web monitoring only supports pay-per-use billing. For details, see [Pay-per-Use Billing](#).

12.2 Websites

Checking Websites

- Step 1
- Log in to the [APM console](#).
- Step 2
- Click  on the left and choose **Management & Governance > Application Performance Management**.
- Step 3
- In the navigation pane, choose **Web Monitoring > Websites**. The connected website list is displayed.

Figure 12-1 Checking websites

APM

Alarm Center

Application Monitoring

Performance Monitoring

Link Trace

Web Monitoring

Websites

Overview

Load Performance

API Requests

JS Errors

Access analysis

Custom Statistics

Session Trace

Scenario Analysis

Network Access

App monitoring

System Management

Website

Usage Statistics

Connect Website

Step 02, 2025-12-05 GMT+08:00 — Step 02, 2025-12-05 GMT+08:00

Last 3 hours

Do not show again

Enter a website name

Website	Application	UV	PV	Avg. Load Time (ms)	JS Error Rate (%)	API Success Rate (%)	Collection Status	Operation
 1	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 num0029	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 2	APItest	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 0	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 2	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 test	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 00012	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 test01	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 num1535-100	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete
 num00701	L902-WEB	0	0	0	0	0	Running	Disable View Alarm Policy Edit Delete

Total Records: 122

10 < 1 2 3 4 5 6 > 12 >

Table 12-1 Website parameters

Parameter	Description
Website	Name of a website.
Application	Application to which the website belongs.
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
Avg. Load Time	Average page load time. Avg. Load Time is displayed only for web/HTML5 pages. For other applets, Avg. Load Time is displayed as 0 .
JS Error Rate	JS error rate = Number of errors triggered during JS running/Total number of Java scripts
API Success Rate	API success rate = Number of APIs that are successfully called/Total number of APIs
Collection Status	Collection status of the website. Options: Running and Disabled .
Operation	Operations that can be performed on the website. For details, see Table 12-7 .

Step 4 In the upper right corner of the **Websites** page, select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Enter a website name in the search box in the upper left corner of the list and click . Only the information about this website is displayed.

Click  in the upper right corner of the list to export information. A maximum of 200 records can be exported.

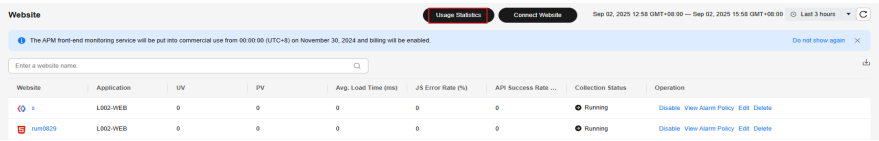
----End

Viewing Usage Statistics

APM provides usage statistics for web monitoring.

Step 1 In the navigation pane, choose **Web Monitoring** > **Websites**. On the displayed page, click **Usage Statistics** to view the usage information of a website.

Figure 12-2 Viewing usage statistics



By default, the trend graphs of **Total Reported Records**, **Reported PVs**, **Reported APIs**, and **Reported Custom Logs**, and the usage overview of all sites are displayed.

Figure 12-3 Usage statistics - metric trends

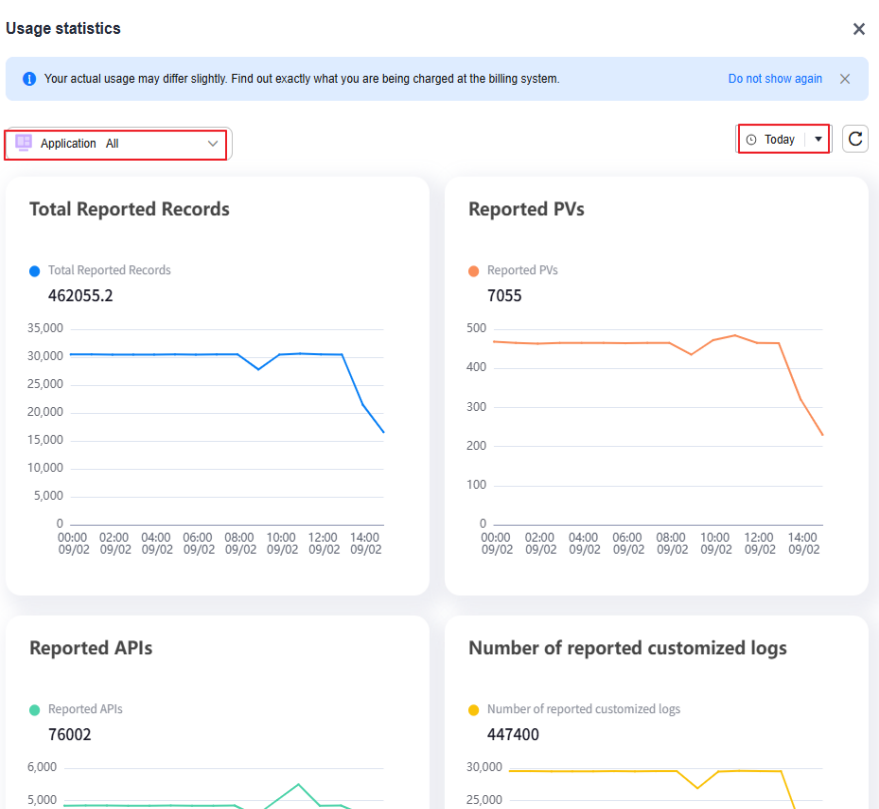


Table 12-2 Usage statistics parameters

Parameter	Description
Total Reported Records	Total number of reported records = Number of reported PVs + Number of reported APIs/10 + Number of reported custom logs
Reported PVs	Number of reported PVs.
Reported APIs	Number of reported APIs.
Reported Custom Logs	Number of reported custom logs.

- In the upper right corner of the **Usage Statistics** page, select a time range. Default: **Today**. Options: **Today**, **Yesterday**, and **Custom**.
- Click  on the right of **PV**, **API**, **Custom**, and **Total** to sort sites.

Step 2 Click the site drop-down list and select a site name to view the usage statistics of the site.

----End

Connecting Websites

Step 1 In the navigation pane, choose **Web Monitoring > Websites**, and click **Connect Website**. On the displayed page, set related parameters.

- After a website is created, **appld** will be automatically generated.
- Before you enter a website name, the **Create Website** button is grayed.
- After a website is created, you can click **Edit** in the **Operation** column to check the SDK extended configuration items.
- Web monitoring data is reported through the automatically generated **appld**. Due to the lack of authentication, dirty data may be generated.
- For details about how to connect applet SDKs, see [SDK Reference](#).
- Web & HTML5 access
 - a. Select **Web & HTML5**.
 - b. Select an application from the drop-down list box and enter a website name.
 - c. Click **Create Website**.
- WeChat mini program access
 - a. Select **WeChat**.
 - b. Select an application from the drop-down list box and enter a website name.
 - c. Click **Create Website**.
- Baidu mini program access
 - a. Select **Baidu**.
 - b. Select an application from the drop-down list box and enter a website name.
 - c. Click **Create Website**.
- Alipay mini program access
 - a. Select **Alipay**.
 - b. Select an application from the drop-down list box and enter a website name.
 - c. Click **Create Website**.
- DingTalk mini program access
 - a. Select **DingTalk**.
 - b. Select an application from the drop-down list box and enter a website name.

c. Click **Create Website**.

Step 2 Track the service.

- Web&H5

Copy the following code to the first line of the **<body>** of **Index.html**.

- Fixed version access

```
<script>
!function(x, n) {
  window[n] = window[n] || {};
  window[n].config = { domain: "https://rum.rnd.****.com",appId: "",apiRepo: true,thirdApi:
true,hashMode: true,JsErrClean: true,smartJsErr: true,webResource: true,traceType: "apm"};
  var o = document.createElement('script');
  o.src = x, o.async = !0;
  var d = document.body.firstChild;
  document.body.insertBefore(o, d);
}('https://test-static-resource.obs.****.ulanqab.****.com/js-agent/jsagent.min.js', '__fr');
</script>
```

- Latest version access

```
<script>
!function(x, n) {
  window[n] = window[n] || {};
  window[n].config = { domain: "https://rum.rnd.****.com",appId: "",apiRepo: true,thirdApi:
true,hashMode: true,JsErrClean: true,smartJsErr: true,webResource: true,traceType: "apm"};
  var o = document.createElement('script');
  o.src = x, o.async = !0;
  var d = document.body.firstChild;
  document.body.insertBefore(o, d);
}('https://test-static-resource.obs.****.ulanqab.****.com/js-agent/jsagent.min.js', '__fr');
</script>
```

- WeChat

 NOTE

1. Add the report address **https://apm-web.region.myhuaweicloud.com** and register it as a valid domain name.

Currently, this function is supported in the AP-Singapore region only. The report address is **https://apm-web.ap-southeast-3.myhuaweicloud.com**.

To add an Agent to the WeChat mini program, perform the following steps:

- a. Run the following command on the terminal to install the SDK:

```
npm i apm-mini-sdk
```

Use the developer tool to build the npm library file of the current project. If the **apm-mini-sdk** file exists in the **miniprogram_npm** folder, the import is successful.

- b. Add the following code to the header of **app.js**:

```
import apmAgent from 'apm-mini-sdk';
apmAgent.startSDK({
  domain: "https://rum.rnd.****.com",appId: "",apiRepo: true,thirdApi: true,hashMode:
true,JsErrClean: true,smartJsErr: true,webResource: true,traceType: "apm"
platform: 'wx'
});
```

- Baidu

 NOTE

- Add the report address **https://apm-web.region.myhuaweicloud.com** and register it as a valid domain name.

Currently, this function is supported in the AP-Singapore region only. The report address is **https://apm-web.ap-southeast-3.myhuaweicloud.com**.

To add an Agent to the Baidu mini program, perform the following steps:

- a. Run the following command on the terminal to install the SDK:

```
npm i apm-mini-sdk
```

- b. Add the following code to the header of **app.js**:

```
import apmAgent from 'apm-mini-sdk';
apmAgent.startSDK({
  domain: "https://rum.rnd.***.com", appId: "cb868c9076de4324a22c6496ab55aadf", apiRepo:
true, thirdApi: true, hashMode: true, JsErrClean: true, smartJsErr: true, webResource: true, traceType:
"apm",
  platform: 'baidu'
});
```

- Alipay

NOTE

- Add the report address **https://apm-web.region.myhuaweicloud.com** and register it as a valid domain name.

Currently, this function is supported in the AP-Singapore region only. The report address is **https://apm-web.ap-southeast-3.myhuaweicloud.com**.

To add an Agent to the Alipay mini program, perform the following steps:

- a. Run the following command on the terminal to install the SDK:

```
npm i apm-mini-sdk
```

- b. Add the following code to the header of **app.js**:

```
import apmAgent from 'apm-mini-sdk';
apmAgent.startSDK({
  domain: "https://rum.rnd.***.com", appId: "cb868c9076de4324a22c6496ab55aadf", apiRepo:
true, thirdApi: true, hashMode: true, JsErrClean: true, smartJsErr: true, webResource: true, traceType:
"apm",
  platform: 'alipay'
});
```

- DingTalk

NOTE

- Add the report address **https://apm-web.region.myhuaweicloud.com** and register it as a valid domain name.

Currently, this function is supported in the AP-Singapore region only. The report address is **https://apm-web.ap-southeast-3.myhuaweicloud.com**.

To add an Agent to the DingTalk mini program, perform the following steps:

- a. Run the following command on the terminal to install the SDK:

```
npm i apm-mini-sdk
```

- b. Add the following code to the header of **app.js**:

```
import apmAgent from 'apm-mini-sdk';
apmAgent.startSDK({
  domain: "https://rum.rnd.***.com", appId: "cb868c9076de4324a22c6496ab55aadf", apiRepo:
true, thirdApi: true, hashMode: true, JsErrClean: true, smartJsErr: true, webResource: true, traceType:
"apm",
  platform: 'ding_talk'
});
```

----End

Setting an Alarm Policy

- Step 1** In the navigation pane, choose **Web Monitoring > Websites**. Locate a target website and click **View Alarm Policy** in the **Operation** column to view the alarm policy list.

Table 12-3 Parameters in the alarm policy list

Parameter	Description
Policy Name	Policy name of the current site.
Monitoring Type	Select a monitoring type. Options: API collection, JS error collection, Page performance collection, and Traffic (PV/UV) collection. The information about the selected type is displayed on the right.
Alarm Severity	Severity of an alarm. Options: COMMON and CRITICAL .
Monitored Object	Website to be monitored.
Alarm Rule	Rule based on which an alarm is triggered.
Updated	Time when the alarm policy was last changed.
Status	Status of the alarm policy. Options: Running and Disabled .
Operation	Edit: Click Edit to edit the alarm policy. Disable/Enable: Disable or enable the alarm policy. Delete: Delete the alarm policy.

You can search for alarm policies by policy name.

1. Enter a policy name in the search box.
2. Click . The information about the policy is displayed.

Step 2 Click **Add Alarm Policy**.

1. Basic information

Table 12-4 Basic information parameters

Parameter	Description
Policy Name	Custom name, which cannot be left blank. Only letters, digits, underscores (_), and hyphens (-) are allowed. Enter up to 512 characters.
Alarm Severity	Severity of an alarm. Options: COMMON and CRITICAL .
Monitored Object	Website to be monitored.

Parameter	Description
Monitoring Type	Select a monitoring type. – Options: API collection , JS error collection , Page performance collection , and Traffic (PV/UV) collection . The information about the selected type is displayed on the right.
Metric Set	Select a target metric set. The information about the selected metric set is displayed on the right.

2. Alarm rule

Figure 12-4 Alarm rule

Alarm Rule

Dimension ⓘ

⊕

* Metric

Condition	Metric	Operator	Threshold	Operation
	Total number of API reques	>	0	⊕

* Alarm Condition

In

10

 collection periods, if alarms are triggered

2

 times, alarms will not be repeated for

10

 minutes.

A

B

C

* Recovery Policy

10

 minutes since last alarm.

* Notification upon Recovery

☒

* Strict Mode ⓘ

☐

Table 12-5 Alarm rule parameters

Parameter	Description
Dimension	(Optional) A fine-grained category of metrics.
Metric	Metric for which you want to define one or more alarm rules.
Alarm Condition	Condition for triggering an alarm. A: 1–10 B: 1–10; not greater than A C: ≥ 10

Parameter	Description
Recovery Policy	Condition for clearing an alarm.
Notification upon Recovery	Whether to notify recipients of alarm clearance.
Strict Mode	Checks all returned values against alarm rules to detect exceptions.

3. Alarm notification

Table 12-6 Alarm notification parameters

Parameter	Description
Notification Content	Alarm details, which contain up to 500 characters. • If Strict Mode is enabled, the alarm notification content supports both Variable and Loop. If Strict Mode is disabled, only Variable is supported. • Alarm notification content. You can customize the content or select required metrics. • Select the metrics to be collected. On the right of the page, click  next to a metric. The metric will then be referenced to the notification content box.
Notification Object	Select a notification object from the drop-down list. Alarms will then be sent to the selected notification object.

Step 3 Click **OK**.

----End

More Operations

Perform the operations listed in [Table 12-7](#) if needed.

Table 12-7 Related operations

Operation	Description
Enabling or disabling a website	Click Disable or Enable in the Operation column to disable or enable a website, respectively.

Operation	Description
Checking the alarm policy	Click View Alarm Policy in the Operation column. For details, see Setting an Alarm Policy .
Editing a website	Click Edit in the Operation column.
Deleting a website	Click Delete in the Operation column.

12.3 Overview

APM Agents periodically collect performance data of websites, including load performance, JS stability, API requests, access traffic, top 5 slowest pages, top 5 URLs with the most JS errors, top 5 slowest APIs, and top 5 URLs with the most visits. The **Overview** page displays metrics such as **UV**, **PV**, **Avg. Load Time**, **JS Errors**, **Slow APIs**, and **API Success Rate**.

Viewing Overall Metrics

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **Web Monitoring > Overview**.

Figure 12-5 Overview

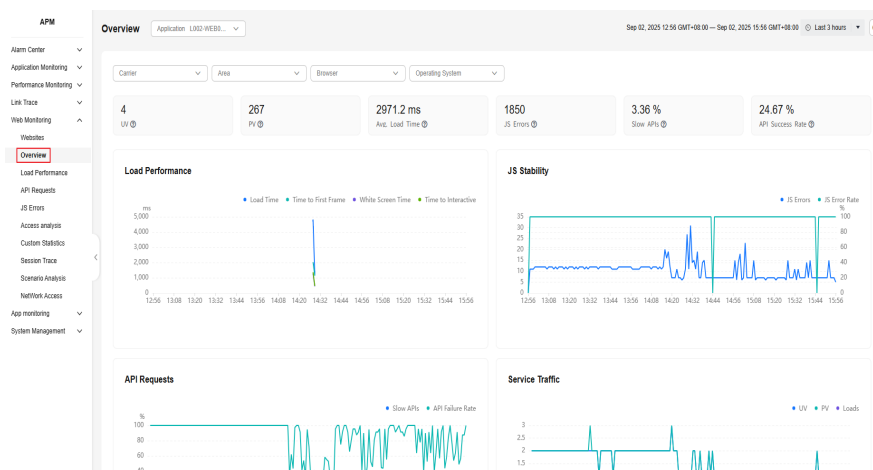


Table 12-8 Overview metrics

Parameter	Description
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
Avg. Load Time	Average page load time.
JS Errors	JS error rate = Number of errors triggered during JS running/ Total number of Java scripts
Slow APIs	Slow APIs refer to APIs with load time longer than 1,000 ms. Slow API proportion = Number of slow APIs/Total number of APIs
API Success Rate	API success rate = Number of APIs that are successfully called/ Total number of APIs

Step 4 Perform other operations if needed.

- Select a site, carrier, region, browser, and OS from the drop-down lists to view required information.
- Select a time range. Default: **Last 20 minutes**.
Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

----End

Load Performance

Step 1 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 2 In the navigation pane, choose **Web Monitoring > Overview**.

- Move the cursor to the load performance graph. The detailed load performance data is displayed.

Figure 12-6 Load performance

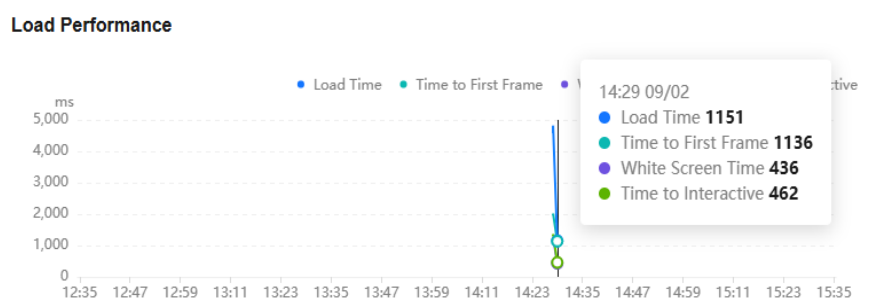


Table 12-9 Load performance parameters

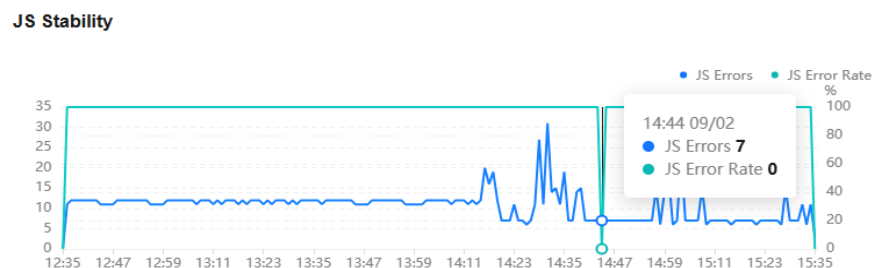
Parameter	Description
Load Time	Page load time.
First Contentful Paint (FCP)	Duration from the time when you open a website to the time when the content on the home page of the browser has been rendered.
First Paint (FP)	Duration from the time when you click a link or enter a URL to the time when the browser starts to display content.
Time to Interactive	Duration from the time when a page starts to be loaded to the time when main resources are rendered and a response is made.

----End

JS Stability

Step 1 In the navigation pane, choose **Web Monitoring > Overview**.

- Move the cursor to the JS stability trend graph. The detailed JS stability data is displayed.

Figure 12-7 JS stability**Table 12-10** JS stability parameters

Parameter	Description
JS Errors	Number of JS errors.
JS Error Rate	JS error rate = Number of errors triggered during JS running/Total number of Java scripts

----End

API Requests

Step 1 In the navigation pane, choose **Web Monitoring > Overview**.

- Move the cursor to the API request trend graph. The detailed API request data is displayed.

Figure 12-8 API requests

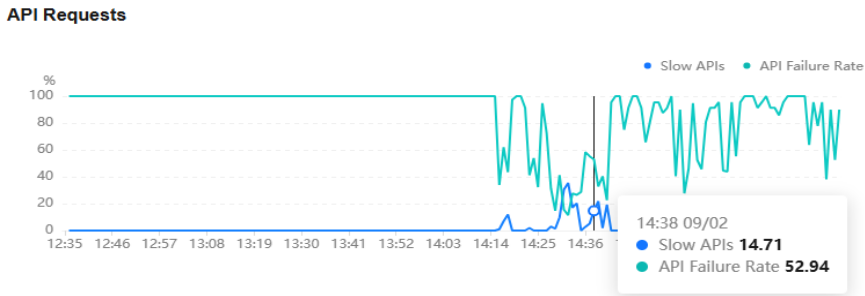


Table 12-11 API request parameters

Parameter	Description
Slow APIs	Slow APIs refer to APIs with load time longer than 1,000 ms. Slow API proportion = Number of slow APIs/Total number of APIs
API Failure Rate	Probability that APIs fail to be called.

----End

Service Traffic

APM Agents regularly collect access traffic data of websites and HTML5 clients. The data includes **UV**, **PV**, and **Loads**.

- Step 1** In the navigation pane, choose **Web Monitoring > Overview**. Move the cursor to the service traffic trend graph. The traffic data is displayed.

Figure 12-9 Service traffic

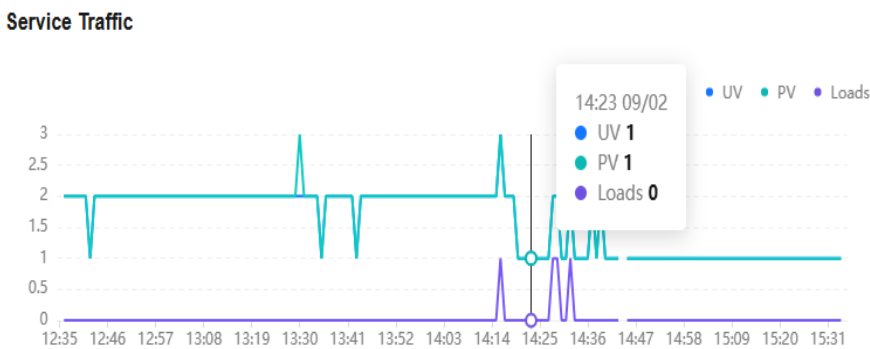


Table 12-12 Service traffic parameters

Parameter	Description
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
Loads	Total number of times pages were loaded.

----End

Top 5 Slowest Pages

This card displays the five URLs with the longest load time. The metrics include **URL**, **Loads**, and **Avg. Load Time**.

Step 1 In the navigation pane, choose **Web Monitoring > Overview**. The **Top 5 Slowest Pages** card is displayed.

Step 2 Click a URL in the **URL** column. The load performance details of the URL are displayed. For details, see [Step 1.a](#) to [Step 1.b](#).

----End

Top 5 JS Errors

This card displays the five URLs with the most JS errors. The metrics include **URL**, **Error**, and **Number of errors**.

Step 1 In the navigation pane, choose **Web Monitoring > Overview**. The **Top 5 JS Errors** card is displayed.

Step 2 Click a URL in the **URL** column. The clustering analysis details of the URL are displayed. For details, see [Step 4](#) to [Step 6](#).

----End

Top 5 Slowest APIs

This card displays the top 5 slow APIs. The metrics include **URL**, **Request Method**, **Number of Slow Requests**, and **Avg. Load Time**.

Step 1 In the navigation pane, choose **Web Monitoring > Overview**. The **Top 5 Slow APIs** card is displayed.

Step 2 Click a URL in the **URL** column. The request statistics details of the URL are displayed. For details, see [Step 4](#).

----End

Top 5 Visits

This card displays five URLs with the most visits. The metrics include **URL**, **Page View (PV)**, and **Number of visitors (UV)**.

- Step 1** In the navigation pane, choose **Web Monitoring > Overview**. The **Top 5 Visits** card is displayed.
- Step 2** Click a URL in the **URL** column. The clustering analysis details of the URL are displayed. For details, see [Step 2](#) to [Step 3](#).

----End

12.4 Load Performance

The **Load Performance** page displays page performance metrics. It includes **Page Performance** and **Feature Statistics**.

Page Performance: displays page load performance trends from multiple dimensions.

Detailed Analysis: displays page performance and resource loading metrics from multiple dimensions, and provides detailed analysis on the metrics or samples.

For mini programs such as WeChat, Baidu, Alipay, and DingTalk, some resource loading metrics (including **Resource Loading Times**, **Resource Loading Success Rate**, **Average Resource Loading Time**, **Static Resource Loading Times**, **Avg. Time for Static Resource Loading**, **Dynamic Resource Loading Times**, and **Avg. Time for Dynamic Resource Loading**) are not supported.

Procedure

- Step 1** Log in to the [APM console](#).
- Step 2** Click  on the left and choose **Management & Governance > Application Performance Management**.
- Step 3** In the navigation pane on the left, choose **Web Monitoring > Load Performance**. The **Load Performance** page is displayed by default.
- Select a site, carrier, region, browser, and OS from the drop-down lists to view required information.
 - Displays the average loading time, P75 Largest Contentful Paint (LCP), number of JS errors, percentage of slow APIs, API success rate, number of resource loading times, resource loading success rate, average resource loading time, number of static resource loading times, average time for static resource loading, number of dynamic resource loading times, and average time for dynamic resource loading.
 - **Page Performance:** displays the access speed metric trend, performance trend, performance sample distribution, loading times, loading time, and loading status.
 - **Detailed Analysis:** displays the URLs for page performance analysis and resources that have been loaded.

- **Table 12-13** Load performance parameters

Parameter	Description
Avg. Load Time	Average page load time.
P75 Largest Contentful Paint (LCP)	LCP reports the render time of the largest image or text block visible in the viewport, relative to when the user first navigated to the page. The 75th percentile LCP is used here.
JS Errors	Number of JS errors.
Slow APIs	Slow APIs refer to APIs with load time longer than 1,000 ms. Slow API proportion = Number of slow APIs/Total number of APIs
API Success Rate	API success rate = Number of APIs that are successfully called/Total number of APIs
Resource Loading Times	Total number of static and dynamic resource loading times.
Resource Loading Success Rate	Success rate of dynamic and static resource loading.
Average Resource Loading Time	Average loading duration of static and dynamic resources.
Static Resource Loading Times	Number of times static resources are loaded.
Avg. Time for Static Resource Loading	Average time for static resource loading.
Dynamic Resource Loading Times	Number of times dynamic resources are loaded.
Avg. Time for Dynamic Resource Loading	Average time for dynamic resource loading.

Step 4 In the upper right corner of the **Load Performance** page, select a time range.
Default: **Last 20 minutes**.

Options: **Last 20 minutes, Last hour, Last 3 hours, Last 6 hours, Last day, Today, Yesterday, Last week, Last month, or Custom.**

----End

Page Performance

The **Page Performance** tab page displays the access speed metric trend, performance trend, performance sample distribution, loading times, loading time, and loading status.

- Access speed metrics

Displays the number of load times, average load time, First Contentful Paint (FCP), First Paint (FP), Time to Interactive (TTI), Cumulative Layout Shift (CLS), Interaction to Next Paint (INP), and average Largest Contentful Paint (LCP).

Table 12-14 Access speed metrics

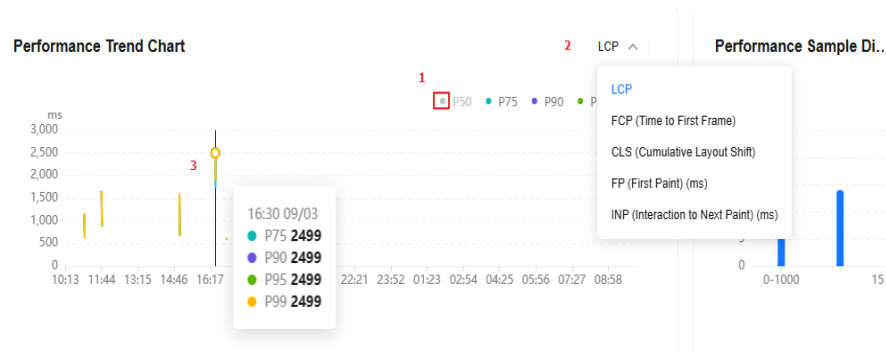
Parameter	Description
Load Times	Number of times the page was loaded.
Avg. Load Time	Average page load time. Duration from the time when you open a website to the time when the content on the home page of the browser has been rendered.
First Contentful Paint (FCP)	Duration for the first content (such as the text, image, or canvas) to be rendered on the page.
First Paint (FP)	Duration for the first pixel to be rendered on the page. This occurs when the HTML parsing is completed or partially completed.
Time to Interactive (TTI)	Duration from the time when a page starts to be loaded to the time when main resources are rendered and a response is made.
Cumulative Layout Shift (CLS)	CLS measures the scores for every unexpected layout shift that occurs during the entire lifecycle of a page and add them together.
Interaction to Next Paint (INP)	INP measures how responsive a page is to visitor interactions. It measures the elapsed time between a click or a keypress. (This option is disabled by default. You can configure it by setting enableInp in initialization parameters.)
Avg. Largest Contentful Paint (LCP)	LCP reports the render time of the largest image or text block visible in the viewport, relative to when the user first navigated to the page. The average LCP is used here.

- Click the colored circle next to each metric. When the cursor is hovered over a circle, the corresponding metric will be highlighted. If a circle is

gray, the corresponding metric will not be displayed in the trend graph. If a circle is colored, the corresponding metric will be displayed in the trend graph.

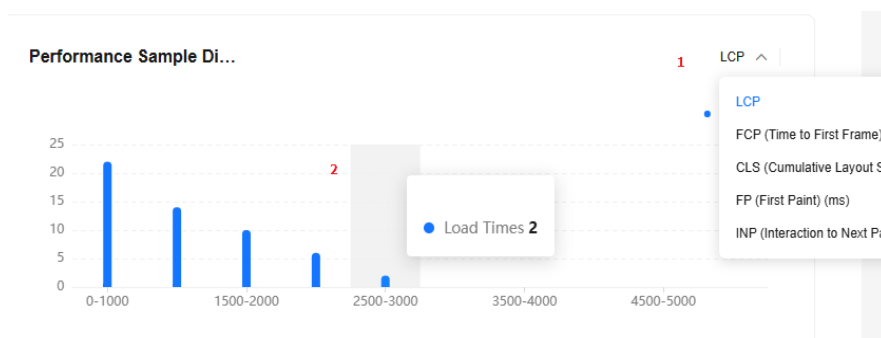
- Move the cursor to the graph. The detailed metric information is displayed.
- Performance trend
Displays the P50, P75, P90, P95, and P99 values of a metric at different time.

Figure 12-10 Performance trend



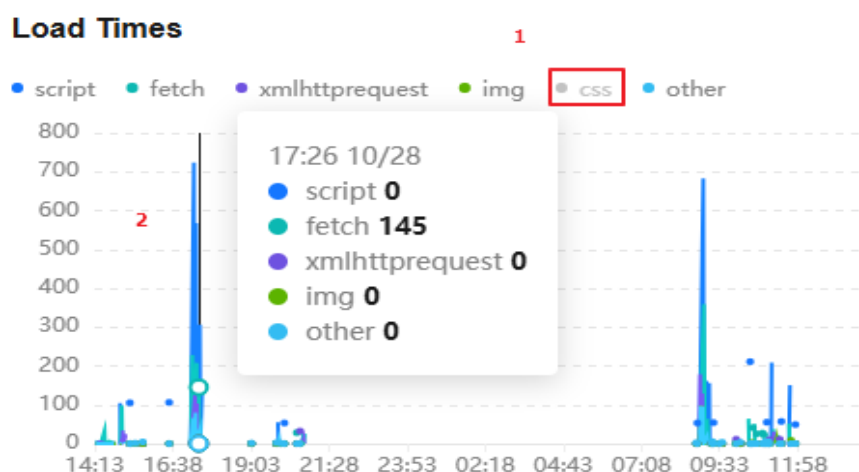
- **1** in **Figure 12-10**: Click the colored circle next to each metric. When the cursor is hovered over a circle, the corresponding metric will be highlighted. If a circle is gray, the corresponding metric will not be displayed in the trend graph. If a circle is colored, the corresponding metric will be displayed in the trend graph.
- **2** in **Figure 12-10**: Select a metric from the drop-down list. Options: **Largest Contentful Paint (LCP)**, **First Contentful Paint (FCP)**, **Cumulative Layout Shift (CLS)**, **First Paint (FP)**, and **Interaction to Next Paint (INP)**.
- **3** in **Figure 12-10**: Move the cursor to the graph. The detailed metric data is displayed.
- Performance sample distribution
Displays the number of load times at different time. Move the cursor to the graph. The detailed metric information is displayed.
- **1** in **Figure 12-11**: Select a metric from the drop-down list. Options: **Largest Contentful Paint (LCP)**, **First Contentful Paint (FCP)**, **Cumulative Layout Shift (CLS)**, **First Paint (FP)**, and **Interaction to Next Paint (INP)**.
As shown in **Figure 12-11**, there are two loads when LCP is in the range of 2500–3000 ms.
- **2** in **Figure 12-11**: Move the cursor to the graph. The detailed metric data is displayed.

Figure 12-11 Performance sample distribution



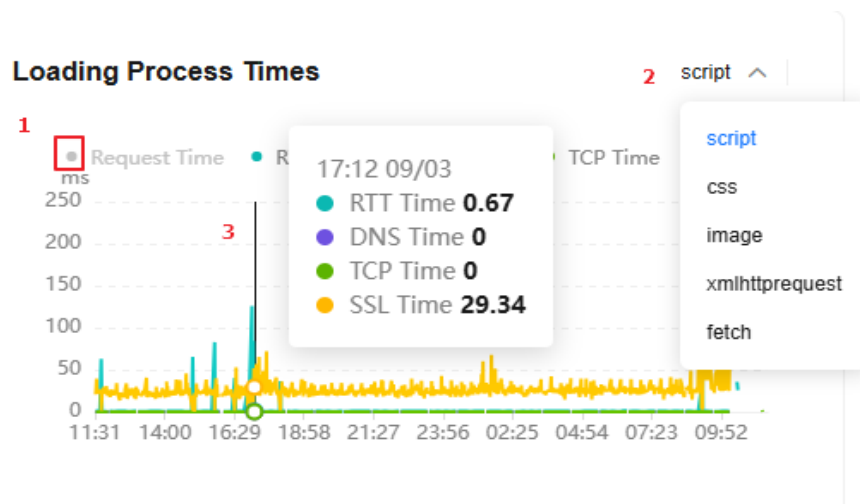
- Load times
Displays the trends of **script**, **fetch**, **xmlhttprequest**, **img**, and **other** at a certain time point.

Figure 12-12 Load times



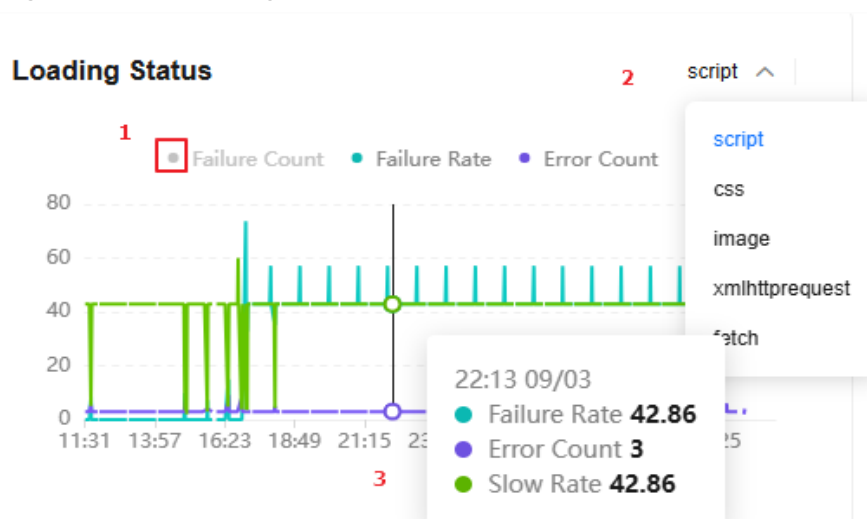
- **1** in **Figure 12-12**: Click the colored circle next to each metric. When the cursor is hovered over a circle, the corresponding metric will be highlighted. If a circle is gray, the corresponding metric will not be displayed in the trend graph. If a circle is colored, the corresponding metric will be displayed in the trend graph.
- **2** in **Figure 12-12**: Move the cursor to the graph. The detailed metric data is displayed.
- Loading time
Displays the RTT, DNS, TCP, and SSL time at a certain time point.

Figure 12-13 Loading time



- 1 in Figure 12-13: Click the colored circle next to each metric. When the cursor is hovered over a circle, the corresponding metric will be highlighted. If a circle is gray, the corresponding metric will not be displayed in the trend graph. If a circle is colored, the corresponding metric will be displayed in the trend graph.
 - 2 in Figure 12-13: Select a metric from the drop-down list, such as **script**, **fetch**, **xmlhttprequest**, **img**, or **other**.
 - 3 in Figure 12-13: Move the cursor to the graph. The detailed metric data is displayed.
- Loading status
Displays the failure rate, error count, and slow rate of a metric a certain time point.

Figure 12-14 Loading status



- 1 in Figure 12-14: Click the colored circle next to each metric. When the cursor is hovered over a circle, the corresponding metric will be highlighted. If a circle is gray, the corresponding metric will not be

displayed in the trend graph. If a circle is colored, the corresponding metric will be displayed in the trend graph.

- **2** in **Figure 12-14**: Select a metric from the drop-down list, such as **script**, **fetch**, **xmlhttprequest**, **img**, or **other**.
- **3** in **Figure 12-14**: Move the cursor to the graph. The detailed metric data is displayed.

Detailed Analysis

The **Detailed Analysis** page displays the URLs for page performance analysis and resources that have been loaded.

Step 1 In the navigation pane, choose **Web Monitoring > Load Performance**. Scroll down the page. The **Detailed Analysis** tab page is displayed by default.

- Page performance analysis

Displays performance analysis details of pages. The metrics include **Page URL**, **Load Times**, **P75 Largest Contentful Paint (LCP)**, **Avg. Largest Contentful Paint (LCP)**, **P75 Largest Contentful Paint (LCP) (Pre-rendering Included)**, **Avg. Largest Contentful Paint (LCP) (Pre-rendering Included)**, **Avg. Load Time**, **First Contentful Paint (FCP)**, **First Paint (FP)**, **Time to Interactive (TTI)**, **Cumulative Layout Shift (CLS)**, and **Interaction to Next Paint (INP)**.

- Enter the search criteria in the search box and click  to view the page performance analysis list that meets the search criteria.
- Click  in the upper right corner of the page performance analysis list and select the target metric data.
- By default, the page performance analysis list is sorted by **Load Times** in descending order.
- a. Click a URL in the **Page URL** column under **Page Performance Analysis** to view the load performance metrics of the URL.

Table 12-15 Load performance parameters

Parameter	Description
Load Times	Number of times the page was loaded.
Avg. Load Time	Average page load time. Duration from the time when you open a website to the time when the content on the home page of the browser has been rendered.
First Contentful Paint (FCP)	Duration for the first content (such as the text, image, or canvas) to be rendered on the page.
First Paint (FP)	Duration for the first pixel to be rendered on the page. This occurs when the HTML parsing is completed or partially completed.
Time to Interactive (TTI)	Duration from the time when a page starts to be loaded to the time when main resources are rendered and a response is made.

Parameter	Description
First Meaningful Paint (FMP)	Duration from the time when the browser responds to the website address you entered to the time when the browser renders meaningful content.
P75 Largest Contentful Paint (LCP)	LCP reports the render time of the largest image or text block visible in the viewport, relative to when the user first navigated to the page. The 75th percentile LCP is used here.
Images	Number of images displayed on the page.
First Load APIs	Number of first load APIs.
APIs	Number of APIs for accessing a page.
Static Resources	Number of resources that are directly displayed on a fixed web page without database queries or program processing.
Size of First-Loaded Resources	Size of the resource loaded for the first time.
JS Error Rate	JS error rate = Number of errors triggered during JS running/Total number of Java scripts
Resource Load Time	Time consumed by loading resources.
Network Time	Time consumed in network communication.

- Page access speed metrics

Displays the number of load times, average load time, First Contentful Paint (FCP), First Paint (FP), Time to Interactive (TTI), Cumulative Layout Shift (CLS), Interaction to Next Paint (INP), and average Largest Contentful Paint (LCP) of a URL. For details about the parameters, see [Table 12-14](#).

- 1) Click the colored circle next to each metric. When the cursor is hovered over a circle, the corresponding metric will be highlighted. If a circle is gray, the corresponding metric will not be displayed in the trend graph. If a circle is colored, the corresponding metric will be displayed in the trend graph.
- 2) Move the cursor to the graph. The detailed metric information is displayed.

- Performance trend

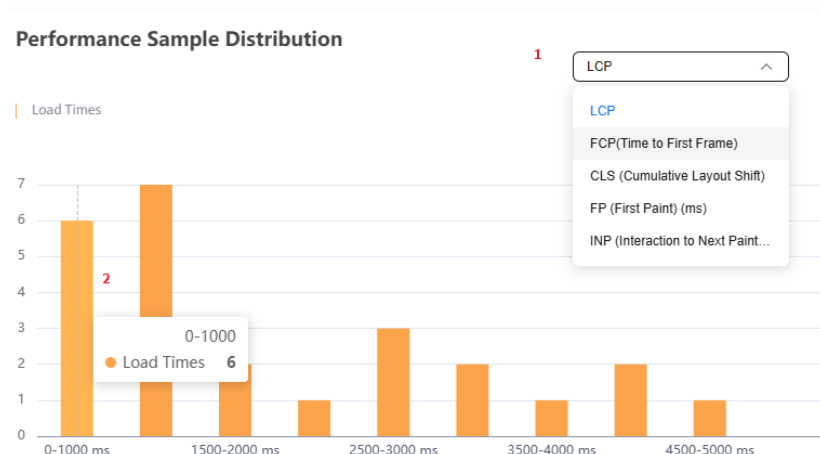
Displays the P50, P75, P90, P95, and P99 values of a metric at different time.

- 1) **1** in the preceding figure: Click the colored circle next to each metric. When the cursor is hovered over a circle, the

corresponding metric will be highlighted. If a circle is gray, the corresponding metric will not be displayed in the trend graph. If a circle is colored, the corresponding metric will be displayed in the trend graph.

- 2) **2** in the preceding figure: Select a metric from the drop-down list. Options: **Largest Contentful Paint (LCP)**, **First Contentful Paint (FCP)**, **Cumulative Layout Shift (CLS)**, **First Paint (FP)**, and **Interaction to Next Paint (INP)**.
 - 3) **3** in the preceding figure: Move the cursor to the graph. The detailed metric data is displayed.
- **Performance sample distribution**
Displays the number of load times at different time. Move the cursor to the graph. The detailed metric information is displayed.
 - 1) **1** in [Figure 12-15](#): Select a metric from the drop-down list. Options: **Largest Contentful Paint (LCP)**, **First Contentful Paint (FCP)**, **Cumulative Layout Shift (CLS)**, **First Paint (FP)**, and **Interaction to Next Paint (INP)**.
As shown in [Figure 12-15](#), there are six loads when LCP is in the range of 0–1,000s.
 - 2) **2** in [Figure 12-15](#): Move the cursor to the graph. The detailed metric data is displayed.

Figure 12-15 Performance sample distribution



- **Page resource load time**
Displays **Resource Load Time**, **Static Resources**, **APIs**, and **Static Resource Size** of a URL.

Table 12-16 Parameters related to the page resource load time

Parameter	Description
Resource Load Time	Time consumed by loading resources.

Parameter	Description
Static Resources	Number of resources that are directly displayed on a fixed web page without database queries or program processing.
APIs	Number of APIs for accessing a page.
Static Resource Size	Size of resources that are directly displayed on a fixed web page without database queries or program processing.

- Loading performance distribution

Displays the number of URL loads in different load time ranges. Move the cursor to the graph. The detailed metric information is displayed.

- Page loading

The page loading time consists of network time, DOM parsing time, and resource loading time. For the network connection details, see the waterfall graph.

Table 12-17 Parameters related to the page load time

Parameter	Description
DNS Lookup	Time spent on DNS lookup for the page.
TCP Connection	Time required for connecting to the server, including the SSL handshake, SSL connection, and TCP three-way handshake time.
SSL Connection	Time required for establishing an SSL connection. HTTPS requests involve such time. It is included in the TCP connection time.
Time to First Byte	Time from when the request is sent to when the first byte is received.
Content Transmission	Time taken by the browser to receive the server's response.
Network Time	Time taken from the time when a user sends a page request to the time when the browser receives a server response, involving DNS lookup, TCP connection, request response, and network transmission events.
DOM Parsing	Time taken to draw the DOM structure tree during DOM parsing.

Parameter	Description
Resource Loading	Time required for document, HTML, and JS parsing, and resource (such as CSS and image) loading.

- Single page access performance

- 1) Go to the **Single Page Access Performance** area at the bottom of the page and click **Details**.

By default, the single page access performance analysis list is sorted by access time in descending order.

- 2) On the displayed details page, view the page performance details. There are three modules on this page: performance metrics, page load exploded view, and resource loading analysis.

- 1) **Performance Metrics**

- 2) **Page Load Exploded View**: displays the total time required for page loading. For details, see [Table 12-17](#).

- 3) **Resource Loading Analysis**: displays resource loading information. Different resource types are in different colors. The length of a bar indicates the time taken to load resources. You can hover the cursor over a bar to view the loading time details or click the tab in the upper right corner to filter data by type.

- b. Click **View** under **APIs. Total Requests, Errors, Error Rate, Slow Requests, Slow Request Proportion, Avg. Success Call Time, Specified API Traces, API Summary, and Status Code Summary** are displayed.

Table 12-18 URL parameters

Parameter	Description
Total Requests	Total number of API requests.
Errors	Number of API request errors.
Error Rate	Error rate of API requests.
Slow Requests	Number of slow API requests.
Slow Request Proportion	Proportion of slow API requests.
Avg. Success Call Time	Average time for calling an API.

- Specified API traces: Displays metrics such as the reporting time, URL, API, device ID, session ID, UID, tag, additional information, request time, wait time, connection time, time to first byte,

download time, API request status, status code, and backend information. For details, see [step 4](#).

- API summary: Displays the API, total number of requests, and proportion of requests. For details, see the [API summary list](#).
- Status code summary: Displays the status code, total number of requests, and proportion of requests. For details, see [Request Status Code Distribution Details](#).
- Resource loading

Displays resource loading details, including the resource name, resource type, number of requests, average duration (ms), failure rate (%), slow requests, and slow request rate (%).

 - Enter the search criteria in the search box above the loaded resource list and click  to view the resources that meet the search criteria.
 - Click  in the upper right corner of the resource loading list and select the target metric data.
 - By default, the resources are sorted by **Solicited** in descending order.
 - The resource loading list supports detailed information drilldown.
 - i. Click a resource name in the list. The number of loading times, average loading time, resource request failure rate, data volume, TCP time, SSL time, and DNS time of the resource are displayed.

Loading Status: When you move the cursor to the loading status graph, the slow request latency, slow request rate, error request latency, and error request rate at the current time are displayed.

Request Latency: When you move the cursor to the request latency graph, the average request time, average RTT, average DNS time, average TCP time, and average SSL time at the current time are displayed.

Loading Status Code: When you move the cursor to the loading status code graph, the proportion of each status code at the current time is displayed.

Geographical Distribution: When you move the cursor to the geographical distribution graph, the proportion of each region at the current time is displayed.

Browser Distribution: When you move the cursor to the browser distribution graph, the proportion of each browser at the current time is displayed.

System Distribution: When you move the cursor to the system distribution graph, the proportion of each system at the current time is displayed.

ISP Distribution: When you move the cursor to the ISP distribution graph, the proportion of each ISP at the current time is displayed.

List: displays network resource details of pages, including the visit time, status code, average request time, average RTT, average DNS time, average TCP time, average SSL time, and session ID.
 - ii. Click a visit time record in the **Visit Time** column. The access data corresponding to the visit time is displayed.

- **Loading Information:** displays the loading information at the current time, including the load time, size of static resources, status code, request response (TTFB), content transfer time, TCP time, SSL time, and DNS time.
- **Access Information:** displays the access information at the current time, including the accessed page and access time.
- **Access User:** displays the information about the access user at the current time, including the user ID, region, browser, resolution, IP address, OS, carrier, and session ID.
Click **Session ID**. The session tracing page is displayed, showing the detailed information about the user track. For details about session tracing, see [Session Tracing](#).
- **Page Load Breakdown Chart:** displays the page loading information at the current time. It shows the DNS query, TCP connection, SSL connection establishment, content transfer time, and content transfer at a certain time point. Content transfer includes the start time, end time, and duration.

----End

12.5 API Requests

The **API Requests** page displays the API failure rate trend, average call time trend, API request list, and status code distribution.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **Web Monitoring > API Requests**. The API request performance data is displayed.

- **API Failure Rate Trend:** displays the number of API calls, proportion of slow requests, and API failure rate. Move the cursor to the graph. The detailed metric data is displayed.
- **Avg. Call Time Trend:** displays the API call queuing time, connection time, time to first byte (TTFB), and download time. Move the cursor to the graph. The detailed metric data is displayed.
- **API Request List:** displays the API request links, total number of requests, average request duration, number of slow requests, proportion of slow requests, number of errors, and error rate.
 - Enter **API** in the search box and click  to view the API requests that meet the search criteria.
 - Click  in the upper right corner of the list and select the target metric data.

- By default, API requests are sorted by **Total Requests** in descending order.
- **Status Code Distribution:** displays API request status codes, number of requests with different status codes returned, and proportion of those requests. Move the cursor to the ring. The number of requests with a certain status code returned is displayed.

Click  in the upper right corner of the list and select the target metric data.

Step 4 Drill down to the details of the API request list and status code distribution.

- Drill down to the details of the API request list.
 - Click an API in the **API Request List** column. The total number of requests, number of errors, error rate, number of slow requests, proportion of slow requests, and average success call time are displayed.
 - Specified API Traces:** displays specified API traces. The displayed metrics include the report time, URL, API, device ID, session ID, UID, tag, additional information, request time, wait time, connection time, time to first byte (TTFB), download time, API request status, status code, and backend information.
 - Click  in the upper right corner of the list and select the target metric data.
 - Enter the search criteria in the search box and click  to view the traces that meet the search criteria.
 - By default, specified API traces are sorted by report time in descending order.
 - In the specified API trace list, click a session ID in the **Session ID** column to go to the session tracing page.
 - In the specified API trace list, click **Analyze** in the **Operation** column to go to the tracing page.

If the **Analyze** button is unavailable, move the cursor to it to check the possible causes.

Figure 12-16 Reason why no data is collected

Specified API Traces 

Enter a keyword. 

Reported	URL	API	Session ID	uid	tag	Duration	Wait Ti	Connect	Time to	Download	API Req	Operation
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	No	1. The backend is not connected to the APM probe. 2. The backend is connected to the APM probe, but no data is collected based on the sampling rate. 3. Suban-rttrace-id is not allowed to return for cross-origin requests. 
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	No	
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	Normal	
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	Normal	
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	Normal	
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	439	11	--	425	3	Normal	
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	439	11	--	425	3	Normal	
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	260	12	--	246	2	Normal	
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	260	12	--	246	2	Normal	

- API Failure Rate Trend:** displays the number of API calls and the API failure rate.

Table 12-19 API request parameters

Parameter	Description
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
Avg. Load Time	Average page load time.
JS Errors	Number of JS errors.
Slow APIs	Slow APIs refer to APIs with load time longer than 1,000 ms. Slow API proportion = Number of slow APIs/Total number of APIs
API Success Rate	API success rate = Number of APIs that are successfully called/Total number of APIs
Queuing Time	API call queuing time, connection time, TTFB, and download time.
Connection Time	Connection time of an API call.
Time to First Byte	Duration from the time when a request is sent to the time when the first response byte is received.
Download Time	Download time of an API call.
API	API request link.
Total Requests	Total number of API requests.
Avg. Duration (ms)	Average API request duration.
Slow Requests	Number of slow API requests.
Slow Request Proportion	Proportion of slow API requests.
Errors	Number of API request errors.
Error Rate	Error rate of API requests.
Status Code	Status code that is returned when an API is called.
Requests with This Status Code	Proportion of API requests with a certain status code returned.
Avg. Success Call Time	Average time for calling an API.
Reported	Time when an API trace was reported.
URL	URL of an API trace.

Parameter	Description
Device ID	Device ID of an API trace.
Session ID	Session ID of an API trace.
Wait Time	Wait time of an API trace.
Connection Time	Connection time of an API trace.
API Request Status	Options: Normal , Slow , and Error .
UID	User ID of an API trace.
Tag	User tag of an API trace.
Duration (ms)	Request duration of an API trace.
Additional Information	Additional information of an API request.
Status Code	Status code corresponding to an API request.
Backend Information	Backend information of an API request.

- Drill down to the details about request status code distribution.
 - a. Click a status code in the **Status Code** column. The metrics such as **Total Requests**, **Errors**, **Error Rate**, **Slow Requests**, **Slow Request Proportion**, and **Avg. Success Call Time** of the status code are displayed.

Table 12-20 Status code parameters

Parameter	Description
Total Requests	Total number of API requests.
Errors	Number of API request errors.
Error Rate	Error rate of API requests.
Slow Requests	Number of slow API requests.
Slow Request Proportion	Proportion of slow API requests.
Avg. Success Call Time	Average time for calling an API.

- i. **Specified API Traces:** displays specified API traces. The displayed metrics include the report time, URL, API, device ID, session ID, UID, tag, additional information, request time, wait time, connection time, time to first byte (TTFB), download time, API request status, status code, and backend information.

- ii.

Click  in the upper right corner of the list and select the target metric data.
- iii.

Enter the search criteria in the search box and click  to view the traces that meet the search criteria.
- iv.

By default, specified API traces are sorted by report time in descending order.
- v.

In the specified API trace list, click a session ID in the **Session ID** column to go to the session tracing page.
- vi.

In the specified API trace list, click **Analyze** in the **Operation** column to go to the tracing page.
- If the **Analyze** button is unavailable, move the cursor to it to check the possible causes.

Figure 12-17 Reason why no data is collected

Specified API Traces ⓘ

Enter a keyword.

Reported	URL	API	Session ID	uid	tag	Duration	Wait T	Connec	Time to	Downlo	API Res	Operation
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	NO	1. The backend is not connected to the APM probe. 2. The backend is connected to the APM probe, but no data is collected based on the sampling rate. 3. Suben-trace-id is not allowed to return for cross-origin requests.
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	NO	
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	Normal	Analyze
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	Normal	Analyze
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	Normal	Analyze
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	339	8	--	307	24	Normal	Analyze
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	439	11	--	425	3	Normal	Analyze
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	439	11	--	425	3	Normal	Analyze
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	260	12	--	246	2	Normal	Analyze
Aug 11, 2...	https://co...	/apm2/w...	bff1d475c...	585cc5c1...	--	260	12	--	246	2	Normal	Analyze

- b.

API summary list
- i.

Click an API in the **API** column. The metrics such as the report time, URL, API, device ID, session ID, UID, tag, additional information, duration, wait time, connection time, time to first byte (TTFB), download time, API request status, status code, and backend information are displayed.

Table 12-21 Parameters about a specified API trace

Parameter	Description
Reported	Time when an API trace was reported.
URL	API trace URL.
API	API request link.
Device ID	Device ID of an API trace.
Session ID	Session ID of an API trace.
UID	User ID of an API trace.
Tag	User tag of an API trace.

Parameter	Description
Additional Information	Additional information of an API request.
Duration (ms)	Request duration of an API trace.
Wait Time	Wait time of an API trace.
Connection Time	Connection time of an API trace.
Time to First Byte	Duration from the time when a request is sent to the time when the first response byte is received.
Download Time	Download time of an API call.
API Request Status	Options: Normal , Slow , and Error .
Status Code	Status code corresponding to an API request.
Backend Information	Backend information of an API request.

- ii. On the **Request Statistics xxx; Status Code xxx** page, click **Analyze** in the **Operation** column to go to the tracing page.
- If the **Analyze** button is unavailable, move the cursor to it to check the possible causes.

----End

12.6 JS Errors

The **JS Errors** page displays site error statistics, including JS stability, JS error list, JS error types, and JS error rate ranking.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **Web Monitoring > JS Errors**. The JS error data is displayed.

- **JS Stability**: displays the number of errors, error rate, and proportion of affected users. Move the cursor to the graph. The detailed metric information is displayed.
- Click  in the upper right corner of **JS Error List**, **JS Error Type List**, and **JS Error Rate Ranking** and select the target metric data.

- **JS Error List:** displays the page URL, error information, error stack, latest error time, number of errors, and number of affected users. By default, the JS error list is sorted by **Errors** in descending order.
- **JS Error Type List:** displays the error type, error rate, number of errors, number of affected users, and PV encountering errors. By default, the JS error type list is sorted by **Error Rate (%)** (preferred) or **Errors** in descending order.
- **JS Error Rate Ranking:** displays the page URL, page error rate, number of errors, and PVs.

Step 4 Click a URL in the **Page URL** column. The detailed performance data of this URL is displayed.

- **Page JS Error Rates:** displays the page JS error rate, PV, and proportion of affected users. Move the cursor to the graph. The detailed metric data is displayed.
- **Page JS Error Statistics:** displays the time when JS errors occurred, UID, tag, error info, error stack, and allowed operations. By default, the page JS error statistic list is sorted by **Time** in descending order.

Table 12-22 JS errors parameters

Parameter	Description
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
Avg. Load Time	Average page load time.
JS Errors	Number of JS errors.
Slow APIs	Slow APIs refer to APIs with load time longer than 1,000 ms. Slow API proportion = Number of slow APIs/Total number of APIs
API Success Rate	API success rate = Number of APIs that are successfully called/Total number of APIs
Errors	Number of errors reported during page access.
Error Rate	Error rate = Number of errors reported during page access/Total number of page views
Affected Users	Proportion of affected users = Number of users who fail to access a page/Total number of users who access the page. For example, if 100 users access a page and 10 of them encounter errors, the proportion of affected users is 10%.
Page URL	URL of the page.
Error Info	Error information.
Error Stack	Stack where an error occurred.

Parameter	Description
Latest Error Time	Time when the last error was reported.
Affected Users	Number of users who are affected.
Page Error Rate	Page error rate = Number of errors reported during page browsing/Total number of browsed pages

Step 5 Click **Analyze** in the **Operation** column to go to the summary page.

Step 6 Click  on the right of the page, select a map file, and upload it.

Based on the uploaded map file, the backend parses the error stack to locate JS errors in source code.

- Only **.map** files can be uploaded. The file size cannot exceed 20 MB.
- To use the stack information function, upload a map file on the right first. After the file is uploaded, the **Stack Info** option will be automatically enabled. If the **Stack Info** button is enabled, you can locate errors in the source code.
- Parsing errors include vue, angular, and JavaScript errors.
- To obtain the specified js.map file, ensure that the packaged code and dependencies are the same version as those in use.

In the **JS Error Type List** area, click the blue text in the **Error Type** column to view the details of the error type. For details, see [Step 4](#), [Step 5](#), and [Step 6](#).

In the **JS Error Rate Ranking** area, click a URL in the **Page URL** column. The detailed performance data of this URL is displayed. For details, see [Step 4](#), [Step 5](#), and [Step 6](#).

----End

12.7 Access Analysis

APM supports access analysis. This function collects statistics on the number of active users, page views, average loading time, JS errors, slow API rate, and API success rate. It includes page performance and feature statistics.

Page Performance: displays access traffic trend and statistics for analysis.

Feature Statistics: displays region, device, or system access features for analysis.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Web Monitoring > Access Analysis**. Select a site from the **Site** drop-down list. By default, the access analysis details of the site are displayed.

- Displays the number of active users, page views, average loading time, JS errors, slow API rate, and API success rate.
- **Page Performance**: displays the access traffic trend and statistics.
- **Feature Statistics**: displays the region distribution, region visit distribution, browser distribution, carrier distribution, and OS distribution.

Table 12-23 Parameters related to access analysis

Parameter	Description
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
Avg. Load Time	Average page load time.
JS Errors	Number of JS errors.
Slow APIs	Slow APIs refer to APIs with load time longer than 1,000 ms. Slow API proportion = Number of slow APIs/Total number of APIs
API Success Rate	API success rate = Number of APIs that are successfully called/Total number of APIs

Step 4 In the upper right corner of the **Access Analysis** page, select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

----End

Page Performance

Step 1 In the navigation pane, choose **Web Monitoring > Access Analysis**. By default, the page performance details are displayed.

- **Access Traffic Trend**: displays **UV**, **PV**, and **Number of Loads**. Move the cursor to the graph. The detailed metric information is displayed.
- **Access Statistics**: displays **Page URL**, **Page Name**, **PV**, **UV**, **New Users**, **Page Clicks**, and **Time on Page**. By default, the access statistics list is sorted by **PV** in descending order

Table 12-24 Parameters related to page performance

-	Parameter	Description
Access Traffic Trend	UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
	PV	Page view (PV), which is the number of times a page is opened or refreshed.
	Number of Loads	Total number of times pages were loaded.
Access Statistics	Page URL	URL of the accessed page.
	Page Name	Name of the accessed page.
	PV	Number of page views.
	UV	Number of users who access the site within a specified period. If a user accesses the site multiple times within a period, only one user will be counted.
	New Users	Number of users who access a page.
	Page Clicks	Number of clicks on a page.
	Time on Page	Duration for staying on a page.

Step 2 Click a URL in the **Page URL** column under **Access Statistics** to view the access statistics details of the URL.

- In the search box, enter a page address and click  to view the access statistics that meet the criteria.
- Click  in the upper right corner of the list to export information. A maximum of 200 records can be exported.
- Click  in the upper right corner of the list and select the target metric data.
 - Displays **PV**, **UV**, **New Users**, **Page Clicks**, and **Time on Page**.
 - **Access Traffic Trend**: displays **UV**, **PV**, and **Number of Loads**. Move the cursor to the graph. The detailed metric information is displayed.
 - **Access Statistics Details**: displays **Started**, **URL**, **UID**, **Client IP Address**, **Load Type**, **Browser**, **OS**, **Carrier**, **Region**, and **Operation**. By default, the access statistics details list is sorted by start time in descending order.
 - Click  in the upper right corner of the access statistics details list and select the target metric data.

Table 12-25 Parameters related to access statistics

Parameter	Description
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
New Users	Number of users who access a page for the first time.
Page Clicks	Number of clicks on a page.
Time on Page	Duration for staying on a page.
Number of Loads	Total number of times pages were loaded.
Started	Time when a user accesses a page.
URL	URL of the accessed page.
UID	User ID.
Client IP Address	IP address of the client.
Load Type	Page load type.
Browser	Type of the browser used by a user.
OS	Type of the OS used by a user.
Carrier	Carrier.
Region	Region where a user is located.

Step 3 Click **Trace Session** in the **Operation** column of the row that contains the target UID. The user track details page is displayed. For details, see [Step 2](#).

----End

Feature Statistics

Step 1 In the navigation pane, choose **Web Monitoring > Access Analysis**. Select a site from the **Site** drop-down list and click the **Feature Statistics** tab. The feature statistics page is displayed.

- If no province name is displayed in the **Region** column, the region distribution graph is not displayed.



Region	Access User (UV)	Views (PV)
Unknown	28	37515

- When you hover the cursor over a region, the bar chart on the right displays the load times of the region.
- **Region Distribution:** displays the region distribution graph, region name, UV, and PV. When you hover the cursor over a region, the PV of the region is displayed in the bar chart on the right. You can scroll the mouse wheel to zoom in or out the region distribution graph.
- **Region Access Distribution:** displays the line graph of a region's access traffic at different time points.
- **Browser Distribution:** displays the browser distribution pie chart, browser name, UV, and PV.
- **Carrier Distribution:** displays the carrier distribution pie chart, carrier name, UV, and PV.
- **OS Distribution:** displays the OS distribution pie chart, OS name, UV, and PV.

Table 12-26 Parameters related to feature statistics

-	Parameter	Description
Region Distribution	Region	Region name.
	Access User (UV)	Number of users in a region.
	Views (PV)	Number of page views or clicks in a region.
Browser Distribution	Browser	Browser name.
	Access User (UV)	Number of users who accesses the site using a specific browser.
	Views (PV)	Number of page views or clicks related to a specific browser.
Carrier Distribution	Carrier	Carrier name.
	Access User (UV)	Number of users served by a specific carrier.
	Views (PV)	Number of page views or clicks related to a specific carrier.
OS Distribution	OS	OS name.
	Access User (UV)	Number of users using a specific OS.
	Views (PV)	Number of page views or clicks related to a specific OS.

Step 2 You can click the blue text in the **Region Distribution**, **Browser Distribution**, **Carrier Distribution**, and **OS Distribution** areas to drill down to details.

- Region distribution
 - a. Click a region name in the **Region** column. The access statistics details of the region are displayed.

Displays **PV**, **UV**, **New Users**, **Page Clicks**, and **Time on Page (s)**.

Access Traffic Trend: displays **UV**, **PV**, and **Number of Loads**. Move the cursor to the graph. The detailed metric information is displayed.

Access Statistics Details: displays **Started**, **URL**, **UID**, **Client IP Address**, **Load Type**, **Browser**, **OS**, **Carrier**, **Region**, and **Operation**.

Table 12-27 Parameters related to access statistics

Parameter	Description
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
New Users	Number of users who access a page for the first time.
Page Clicks	Number of clicks on a page.
Time on Page (s)	Duration for staying on a page.
Number of Loads	Total number of times pages were loaded.
Started	Time when a user accesses a page.
URL	URL of the accessed page.
UID	User ID.
Client IP Address	IP address of the client.
Load Type	Page load type.
Browser	Type of the browser used by a user.
OS	Type of the OS used by a user.
Carrier	Carrier.
Region	Region where a user is located.

- b. Click **Trace Session** in the **Operation** column of the row that contains the target UID. The user track details page is displayed. For details, see [User track details](#).
- Browser distribution
 - a. Click a browser name in the **Browser** column. The access statistics of the browser are displayed. For details, see [Step 2.a](#) to [Step 2.b](#).
- Carrier distribution
 - a. Click a carrier name in the **Carrier** column. The access statistics of the carrier are displayed. For details, see [Step 2.a](#) to [Step 2.b](#).

- OS distribution
 - a. Click an OS name in the **OS** column. The access statistics of the OS are displayed. For details, see [Step 2.a](#) to [Step 2.b](#).

----End

12.8 Custom Statistics

APM web monitoring supports custom statistics.

Checking Custom Statistics List

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **Web Monitoring > Custom Statistics**. Then select a site from the **Site** drop-down list. All custom statistics of the site are displayed.

1. Custom statistics parameters include **Event Name**, **Event Address**, **Trigger Times**, **Users**, and **Last Reported**. By default, custom statistics are sorted by **Last Reported** in descending order.

Table 12-28 Custom statistics parameters

Parameter	Description
Event Name	Name of an event.
Event Address	Address of an event.
Trigger Times	Number of times that an event is triggered.
Users	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
Last Reported	Time when the last event was reported.

2. In the upper right corner of the **Custom Statistics** page, select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

3. In the search box, enter an event name and click . The custom statistics that meet the search criterion are displayed.
4. Click **Tracking Examples**. Event, log, and numerical statistics tracking examples are displayed.
 - The tracking format is "`__rum.log(key, value)`". **value** can be of the string or numeric type.

- When the value is of the string type, the value is displayed when the custom statistics are queried. Length: [0, 30 × 1,024].
- When the value is of the numeric type, "--" is displayed when custom statistics are queried. Value range: [-2147483648, 2147483647].

----End

Checking Custom Statistics

- Step 1** In the navigation pane on the left, choose **Web Monitoring > Custom Statistics**. Then select a site from the **Site** drop-down list. All custom statistics of the site are displayed.
- Step 2** Click an event name in the **Event Name** column. **Event Trigger Trend** and **Event Trigger Records** are displayed.
- **Event Trigger Trend:** displays the number of requests, PV, UV, maximum event value, minimum event value, and average event value. Move the cursor to the event trigger trend graph. The event trigger details are displayed.
 - **Event Trigger Records:** displays the event value, event address, number of event trigger times, maximum event value, minimum event value, average event value, UV, and average number of trigger times per user.
- Step 3** Click  next to an event name in the **Event Name** column. **Event Value, Event Address, Trigger Times, Users, and Last Reported** are displayed. By default, events are sorted by **Last Reported** in descending order.
- Step 4** Click an event value in the **Event Value** column. **Event Trigger Trend** and **Event Trigger Records** are displayed.
- **Event Trigger Trend:** displays the number of requests, PV, and UV of the event. Move the cursor to the event trigger trend graph. The event trigger details are displayed.
 - **Event Trigger Records:** displays the event value, event address, and last report time.

Table 12-29 Custom statistics parameters

Parameter	Description
Requests	Number of requests.
PV	Page view (PV), which is the number of times a page is opened or refreshed.
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.
Max. Event Value	Maximum event value.
Min. Event Value	Minimum event value.

Parameter	Description
Avg. Event Value	Average event value.
Event Value	Event value. For details, see Custom Statistics .
Event Address	Event address.
Trigger Times	Number of times that a custom event was triggered.
Trigger Times Per User	Average number of times that users trigger the custom event.
Last Reported	Time when the last custom event was reported.

----End

12.9 Session Tracing

APM supports session tracing. This function enables full-link tracing based on the user name or ID. The behavior of a user can be reproduced to locate faults. Session tracing includes access, performance, API, JS error, behavior, and custom tracing.

Viewing Session Tracing

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Web Monitoring > Session Tracing**. Select a site from the **Site** drop-down list. By default, the access tracing details of the site are displayed.

1. In the upper right corner of the **Session Tracing** page, select a time range.
Default: **Last 20 minutes**.
Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.
2. Click the search box and select criteria from the drop-down list to search.
Click  to view the session tracing information that meets the search criteria at the site.
3. Click  in the upper right corner of the list to export information. A maximum of 200 records can be exported.

----End

Access Tracing

Step 1 In the navigation pane, choose **Web Monitoring > Session Tracing**. Select a site from the **Site** drop-down list. By default, the access tracing details of the site are displayed.

- Click  in the upper right corner of the access tracing list and select the target metric data.
- By default, the access tracing list is sorted by **Start Time** in descending order.

Table 12-30 Access tracing parameters

Parameter	Description
Session ID	Unique ID for a user to access a server or web page.
Page URL	URL of the accessed page.
Start Time	Time when the session starts.
Client IP Address	IP address of the client.
UID	User ID.
Tag	User-defined tag, which can be used as a keyword to search.
Views (PV)	Page view (PV), which is the number of times a page is opened or refreshed.

Step 2 Click a session ID in the **Session ID** column to view the user track details.

- Metrics include **Page Loads**, **API Requests**, **JS Errors**, **Visits**, **Clicks**, and **Custom Statistics**.
- User track details include **All**, **Page Loads**, **API Requests**, **JS Errors**, **Clicks**, **Visits**, and **Custom Statistics**.

Click **All** to display all user track details of the session ID. A maximum of 200 tracks can be displayed.

Click other tabs such as **Page Loads** to display the user track details of the session ID on the corresponding tab. For example, if you click the **Page Loads** tab, the user track details of all page loads under the session ID are displayed.

You can collapse and expand user track details.

Table 12-31 Parameters of user track details

Type	Parameter	Description
Page Loads	Page URL	URL of the accessed page.
	Request Source	Source of the request.
	Load Time	Page load time.

Type	Parameter	Description
	First Contentful Paint (FCP)	Duration from the time when you open a website to the time when the content on the home page of the browser has been rendered.
	First Paint (FP)	Duration from the time when you click a link or enter a URL to the time when the browser starts to display content.
	Time to Interactive (TTI)	Duration from the time when a page starts to be loaded to the time when main resources are rendered and a response is made.
API Requests	Page URL	URL of the accessed page.
	Device ID	ID of the device.
	API Address	API address.
	Request Method	The request method can be GET , POST , PUT , DELETE , or OPTION .
	Status Code	Status code.
JS Errors	Error Occurred	Address where the JS error occurred.
	Device ID	ID of the device.
	Error Cause	Cause of the JS error.
Clicks	Page URL	URL of the accessed page.
	Device ID	ID of the device.
	X-axis Position	X-axis position of the browser coordinate.
	Y-axis Position	Y-axis position of the browser coordinate.
	Operation	Options: click and dbclick .
Visits	Access Address	URL of the accessed page.
	Client IP Address	Client IP address.
	Client System	Client OS type.
	Browser	Browser type, for example, Chrome, Internet Explorer, or Firefox.

Type	Parameter	Description
Custom Statistics	Page URL	URL of the page to be reported.
	Event Name	Name of an event.
	Event Value	Event value. For details, see Custom Statistics .

----End

Performance Tracing

Step 1 In the navigation pane, choose **Web Monitoring > Session Tracing**. Select a site from the **Site** drop-down list. Click **Performance** to switch to the performance tracing details of the site.

Click  in the upper right corner of the performance tracing list and select the target metric data.

By default, the performance tracing list is sorted by **Load Times** in descending order.

Table 12-32 Performance tracing parameters

Parameter	Description
Session ID	Unique ID for a user to access a server or web page.
Page URL	URL of the accessed page.
Client IP Address	IP address of the client.
UID	User ID.
Tag	User-defined tag, which can be used as a keyword to search.
Load Times	Total number of times pages were loaded.
Avg. Load Time	Average page load time.
Avg. First Contentful Paint (FCP)	Average first contentful paint time.
Avg. First Paint (FP)	Average first paint time.
Avg. Time to Interactive	Average interaction time.

Step 2 Click a session ID in the **Session ID** column to view the user track details. For details, see [Step 2](#).

----End

API Tracing

Step 1 In the navigation pane, choose **Web Monitoring > Session Tracing**. Select a site from the **Site** drop-down list. Click **APIs** to switch to the API tracing details of the site.

Click  in the upper right corner of the API tracing list and select the target metric data.

By default, the API tracing list is sorted by **Load Times** in descending order.

Table 12-33 API tracing parameters

Parameter	Description
Session ID	Unique ID for a user to access a server or web page.
Page URL	URL of the accessed page.
Client IP Address	IP address of the client.
UID	User ID.
Tag	User-defined tag, which can be used as a keyword to search.
Load Times	Total number of times pages were loaded.
Avg. Load Time	Average page load time.
Slow Requests	Number of slow API requests.
Slow Request Rate	Slow API request rate.
Failed Requests	Number of failed requests.
Failed Request Rate	Failed API request rate.

Step 2 Click a session ID in the **Session ID** column to view the user track details. For details, see [Step 2](#).

----End

JS Error Tracing

Step 1 In the navigation pane, choose **Web Monitoring > Session Tracing**. Select a site from the **Site** drop-down list. Click **JS Error Tracing** to switch to the JS error tracing details of the site.

Click  in the upper right corner of the JS error tracing list and select the target metric data.

By default, the JS error tracing list is sorted by **Load Times** in descending order.

Table 12-34 JS error tracing parameters

Parameter	Description
Session ID	Unique ID for a user to access a server or web page.
Page URL	URL of the accessed page.
End Time	Time when the last error was reported.
Client IP Address	IP address of the client.
UID	User ID.
Tag	User-defined tag, which can be used as a keyword to search.
Load Times	Total number of times pages were loaded.
Views (PV)	Page view (PV), which is the number of times a page is opened or refreshed.
UV	Number of visitors who access the website. If a user accesses the site multiple times in a day, only one UV will be counted.

Step 2 Click a session ID in the **Session ID** column to view the user track details. For details, see [Step 2](#).

----End

Behavior Tracing

Step 1 In the navigation pane, choose **Web Monitoring > Session Tracing**. Select a site from the **Site** drop-down list. Click **Behavior** to switch to the behavior tracing details of the site.

Click  in the upper right corner of the behavior tracing list and select the target metric data.

By default, the behavior tracing list is sorted by **Start Time** in descending order.

Table 12-35 Behavior tracing parameters

Parameter	Description
Session ID	Unique ID for a user to access a server or web page.
Page URL	URL of the accessed page.
Start Time	Time when the first operation is performed.
Client IP Address	IP address of the client.
UID	User ID.

Parameter	Description
Tag	User-defined tag, which can be used as a keyword to search.
Load Times	Total number of times pages were loaded.
Views (PV)	Page view (PV), which is the number of times a page is opened or refreshed.

Step 2 Click a session ID in the **Session ID** column to view the user track details. For details, see [Step 2](#).

----End

Custom Tracing

Step 1 In the navigation pane, choose **Web Monitoring** > **Session Tracing**. Select a site from the **Site** drop-down list. Click **Custom** to switch to the custom tracing details of the site.

Click  in the upper right corner of the custom tracing list and select the target metric data.

By default, the custom tracing list is sorted by **Load Times** in descending order.

Table 12-36 Custom tracing parameters

Parameter	Description
Session ID	Unique ID for a user to access a server or web page.
Page URL	URL of the accessed page.
End Time	Last report time.
Client IP Address	IP address of the client.
UID	User ID.
Tag	User-defined tag, which can be used as a keyword to search.
Load Times	Total number of times pages were loaded.
Views (PV)	Page view (PV), which is the number of times a page is opened or refreshed.

Step 2 Click a session ID in the **Session ID** column to view the user track details. For details, see [Step 2](#).

----End

13 Open Tracing

13.1 Overview

This function collects the complete traces of distributed applications and displays URL, database, and exception data. It helps developers quickly analyze and diagnose performance bottlenecks in the distributed application architecture, improving development and diagnosis efficiency in the microservice era.

Constraints

- Only whitelisted users can use the open tracing function.
- Installing Agents is the prerequisite for using open tracing. For details, see [Agent Access](#).

13.2 Application Management

On the **Application Management** page, you can create and delete SkyWalking and OpenTelemetry applications, check basic application information, set sampling rates, define slow SQL statements and calls, and check component environments and Agents.

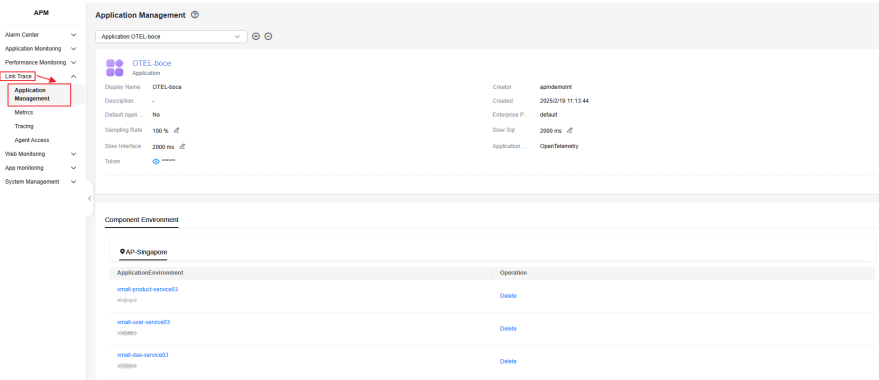
Checking Application Management

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

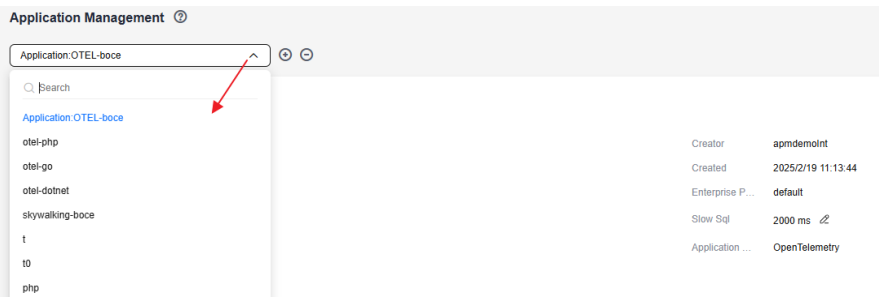
Step 3 In the navigation pane, choose **Link Trace > Application Management**.

Figure 13-1 Application management



Step 4 Select an application from the **Application** drop-down list on the right of the page.

Figure 13-2 Selecting an application



1. Application information

Table 13-1 describes the application information.

Table 13-1 Application information parameters

Parameter	Description
Display Name	Display name of the application.
Creator	Account for creating the application.
Description	Description of the application.
Created	Time when the application is created.
Default Application	Whether the application is a default one. Default: No .
Enterprise Project	Enterprise project name.
Sampling Rate	Sampling rate of trace data. Value range: 0 to 100.
Slow SQL	SQL threshold. When the collected value exceeds this threshold, the SQL statement is considered slow.
Slow Call	URL threshold. When the collected value exceeds this threshold, the URL is considered slow.

Parameter	Description
Application Type	Type of the application.
APPKEY	Unique identifier of an application.

2. Component information

Table 13-2 Component information

Parameter	Description
Component/ Environment	Name of a component or environment. You can click the text in blue to go to the corresponding component or environment page.
Agent Status	Displays the Agent statuses and number of Agents in different statuses. Options: Normal and Abnormal .
Operation	Delete

3. Agent list

Table 13-3 Agent parameters

Parameter	Description
Instance Name	Instance name.
Agent Type	Options: JAVA , Python , GO , Nodejs , PHP , and .Net .
Agent Status	Options: Normal and Offline .
Component/ Environment	Name of a component or environment.
Last Heartbeat	Latest data collection time.
Operation	Delete the Agent. - If the Agent status is Normal, the Agent cannot be deleted. - If the Agent status is Offline, the Agent can be deleted.

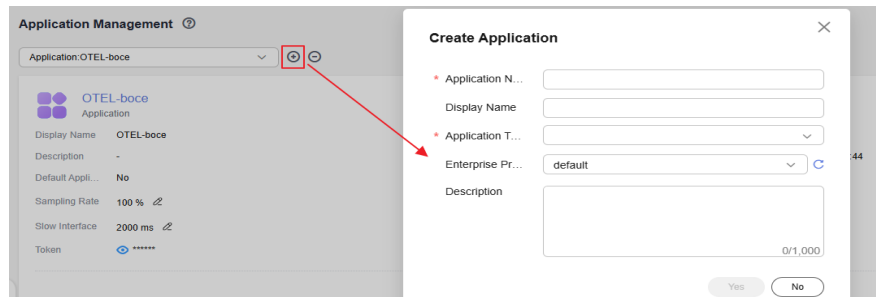
----End

Creating an Application

Step 1 In the navigation pane, choose **Link Trace > Application Management**.

Step 2 Click  next to the **Application** drop-down list. The **Create Application** dialog box is displayed.

Figure 13-3 Creating an application



Step 3 Set application parameters.

Table 13-4 Parameters for creating an application

Parameter	Description
Application Name	Name of an application, which cannot be empty. Enter 1 to 128 characters and start with a letter. Only digits, letters, underscores (_), and hyphens (-) are allowed.
Display Name	Application alias. Enter 1 to 128 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Application Type	Options: SkyWalking or OpenTelemetry .
Enterprise Project	Select an enterprise project from the drop-down list. This parameter is displayed only when you use the enterprise edition.
Description	Description of the application. Enter up to 1,000 characters.

Step 4 Click **Yes**.

----End

Deleting an Application

Step 1 In the navigation pane, choose **Link Trace > Application Management**.

Step 2 Click  next to the **Application** drop-down list. The **Delete Application** dialog box is displayed.

Step 3 Click **Yes** to delete the application.

----End

13.3 Metrics

13.3.1 Overview

The **Overview** page displays the summary data of all component instances. The data includes the total number of requests, average response time, API error rate, slow calls, slow SQL statements, and number of exceptions. **Slow Invoking** displays the five invoking records with the longest duration, **Error Invoking** displays the five invoking records with the most errors, **Slow SQL** displays the five SQL records with the longest duration, and **Exception** displays the five records with the most exceptions.

- Step 1 Log in to the [APM console](#).
- Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.
- Step 3 In the navigation pane, choose **Link Trace > Metrics**.
- Step 4 In the tree on the left, click the target environment and click the **Overview** tab. On the **Overview** tab page, view the application monitoring data of the instance.
- Step 5 Select a time range. Default: **Last 20 minutes**.
Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, or **Custom**.

Figure 13-4 Overview

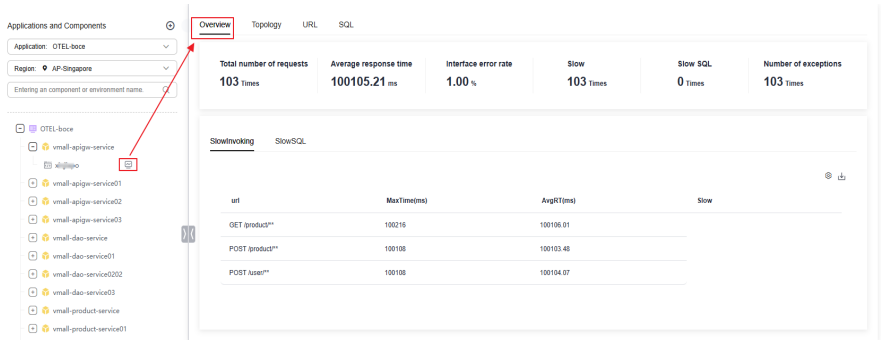


Table 13-5 Overview parameters

Parameter	Description
Total Requests	Total number of trace requests in an environment.
Avg. Response Time	Average response time of traces in an environment.
API Error Rate	API error rate of traces in an environment.
Slow Calls	Total number of slow calls in an environment.

Parameter	Description
Slow SQL	Total number of slow SQL statements in an environment.
Exceptions	Total number of trace exceptions in an environment.

- Click  in the upper right corner of the list and select the target metric data.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.
- **Slow Invoking**: displays the five URLs with the longest duration, maximum response time, average response time, and number of slow calls in an environment.
- **Error Invoking**: displays the five URLs with the most errors, maximum response time, average response time, and number of errors in an environment.
- **Slow SQL**: displays the five SQL statements with the longest duration, maximum response time, average response time, and number of slow SQL calls in an environment.
- **Exception**: displays the five URLs with the most exceptions, maximum response time, average response time, and number of exceptions in an environment.

----End

13.3.2 Topology

The topology displays the call relationships between services within a period. The statistics can be collected from the caller or the callee. You can also view the trend. On the topology, you can view the call relationships between services and check whether the calls between services are normal to quickly locate faults. The application relationships, call data (service and instance metrics), and health status are clearly displayed.

Viewing the Topology

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Link Trace > Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Switch to the **Topology** tab page. The call trend of the selected instance is displayed.

Figure 13-5 Viewing the topology

Step 6 Click  next to **Display only calls between components**.

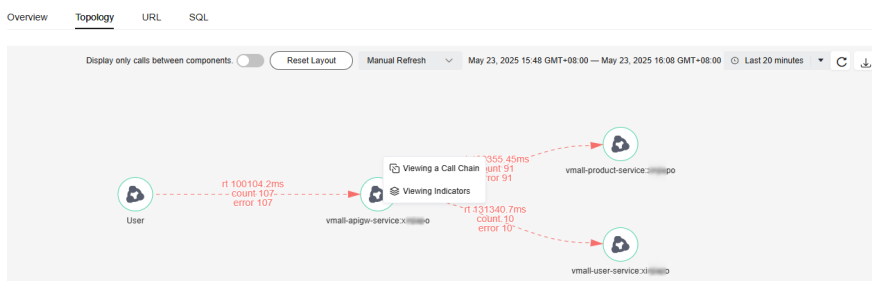
When the button turns blue, only the calls between components are displayed.

Step 7 Select the refresh setting. Default: **Manual Refresh**. In addition, **Automatic refresh in 1 minute**, **Automatic refresh in 5 minutes**, and **Automatic refresh in 15 minutes** are supported.

Step 8 Select a time range. Default: **Last 20 minutes**.

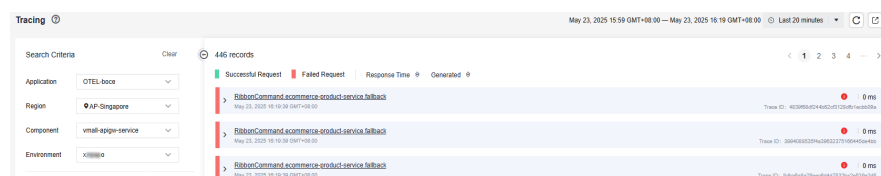
Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, or **Custom**.

Step 9 Right-click a component and then click **Viewing a Call Chain** or **Viewing Indicators**.

Figure 13-6 Viewing the call chain or indicators of the component

- Viewing the call chain

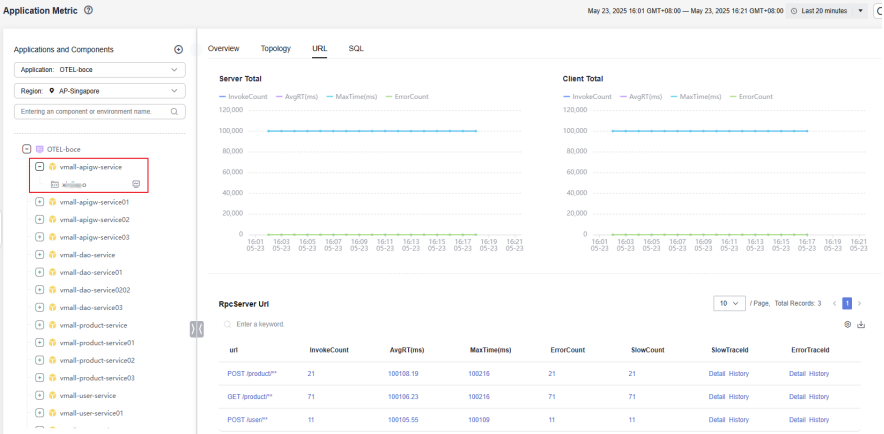
Click **Viewing a Call Chain** to go to the trace page of the component. For details, see [Tracing](#).

Figure 13-7 Viewing the call chain

- Viewing indicators

Click **Viewing Indicators** to go to the URL page of the component. For details, see [URL](#).

Figure 13-8 Viewing indicators



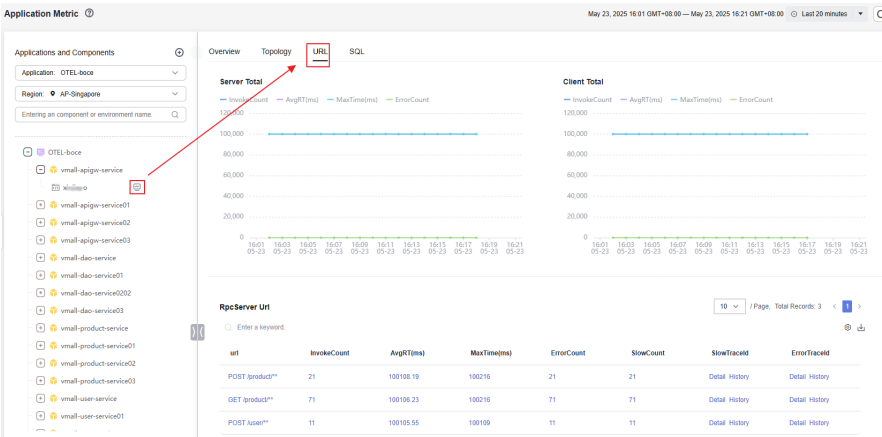
----End

13.3.3 URL

This function monitors the URL of the current application when being called by an external service and URL of an external application when being called by the current service. Based on this monitoring item, you can obtain the actual external status and call conditions of your service.

- Step 1
- Log in to the [APM console](#).
- Step 2
- Click  on the left and choose **Management & Governance > Application Performance Management**.
- Step 3
- In the navigation pane, choose **Link Trace > Metrics**.
- Step 4
- In the tree on the left, click an environment. Then click the **URL** tab.

Figure 13-9 URL



- Step 5
- On the displayed page, select an instance to view its monitoring data.
- Step 6
- Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, or **Custom**.

----End

Server

Monitors the URLs called by external services and displays the trends of metrics such as the number of calls, maximum duration, average duration, and number of errors.

Client

Monitors the URLs of external services called by the current application and displays the trends of metrics such as the number of calls, maximum duration, average duration, and number of errors.

Server API statistics

Monitors called server APIs. The monitored metrics include the URL, number of calls, average duration, maximum duration, number of errors, number of slow calls, slow trace, and error trace.

- Set the search criteria and click  in the search box to filter instances.
- Click  in the upper right corner of the list and select the target metric data.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

Client API statistics

Monitors called client APIs. The monitored metrics include the URL, number of calls, average duration, maximum duration, number of errors, number of slow calls, slow trace, and error trace.

- Set the search criteria and click  in the search box to filter instances.
- Click  in the upper right corner of the list and select the target metric data.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

13.3.4 SQL

This function monitors database access, including MySQL, Elasticsearch, and HBase. Collectors collect details about executed statements to help you locate performance problems in code.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Link Trace > Metrics**.

Step 4 In the tree on the left, click an environment. Then click the **SQL** tab.

Step 5 On the displayed page, select an instance to view its monitoring data.

----End

Overview

Displays the number of database call errors, average duration, number of calls, and maximum duration of the selected instance.

SQL statistics

Monitors databases by SQL statement. The monitored metrics include the statement, number of calls, average duration, maximum duration, number of errors, number of slow SQL statements, slow SQL trace, and error SQL trace.

- Set the search criteria and click  in the search box to filter instances.
- Click  in the upper right corner of the list and select the target metric data.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

Database statistics

Displays monitoring metrics by database, including the database, number of calls, average duration, maximum duration, and number of errors.

- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.
- Set the search criteria and click  in the search box to filter instances.
- Click  in the upper right corner of the list and select the target metric data.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

13.3.5 Exception

This function monitors application exception logs. Take the monitoring of Java exception logs as an example. Once you use the log system to print logs, they will be collected by APM.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Link Trace > Metrics**.

Step 4 In the tree on the left, click the target environment and click the **Exception** tab.

Step 5 On the displayed page, select an instance to view its exception data.

----End

Overview

Displays the total number of exceptions for the selected instance.

Exception

Monitors the exception logs of applications. The monitored metrics include the exception type, count, message, exception stack, and exception trace.

- Set the search criteria and click  in the search box to filter instances.
- Click  in the upper right corner of the list and select the target metric data.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

13.3.6 JVM

The **JVM** page displays the GC information, CPU usage, threads, class load, memory usage, and memory pools of applications.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Link Trace > Metrics**.

Step 4 In the tree on the left, click an environment. Then click the **JVM** tab.

Step 5 On the displayed page, select an instance to view its monitoring data.

----End

GC

Monitors GC and displays the GC duration and GC times in a graph.

CPU

Monitors CPU and displays the CPU usage in a graph.

Threads

Monitors threads and displays the number of blocked threads, peak threads, terminated threads, runnable threads, active threads, Daemon threads, and waiting threads in graphs.

Class loading

Monitors loaded classes and displays the total number of loaded classes, number of current classes, and total number of unloaded classes in a graph.

Memory

Monitors the memory of applications. The monitored metrics are **isHeap**, **max**, **used**, **init**, and **committed**.

- Set the search criteria and click  in the search box to filter instances.
- Click  in the upper right corner of the list and select the target metric data.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

Memory pool

Monitors the memory pool of applications. The monitored metrics are **type**, **max**, **used**, **init**, and **committed**.

- Set the search criteria and click  in the search box to filter instances.
- Click  in the upper right corner of the list and select the target metric data.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

13.3.7 Alarm Policies

Only applications whose access mode is **SkyWalking** support alarm policy settings for a single component.

Creating an APM Alarm Policy

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Link Trace > Metrics**.

Step 4 In the tree on the left, click  next to the target environment. The **Component Settings** tab page is displayed by default. Switch to the **Alarm Policies** tab page.

Step 5 Click **Add Alarm Policy**.

1. Basic information

Table 13-6 Basic information about an alarm policy

Parameter	Description
Policy Name	Custom name, which cannot be left blank. Only letters, digits, underscores (_), and hyphens (-) are allowed. Enter up to 512 characters.
Alarm Severity	Severity of an alarm. Options: COMMON and CRITICAL .
Monitored Object	Name of the object to be monitored. The default value is the name of the current environment and cannot be changed.
Alarm Policy Type	Options: Single-node and Aggregate . Single-node indicates single-instance metric alarms, and Aggregate indicates aggregated metric alarms of all instances under a component.
Monitoring Type	Select a monitoring type. The information about the selected type is displayed on the right.

Parameter	Description
Metric Set	Select a target metric set. The information about the selected metric set is displayed on the right.

2. Alarm rule

Figure 13-10 Alarm rule

Alarm Rule

Dimension ⓘ

⊕

* Metric

Condition	Metric	Operator	Threshold	Operation
	Total number of API reques	> ▾	0	⊕

* Alarm Condition

In collection periods, if alarms are triggered times, alarms will not be repeated for minutes.

A B C

* Recovery Policy

minutes since last alarm.

* Notification upon Recovery

☒

* Strict Mode ⓘ

☐

Table 13-7 Parameters for setting an alarm rule

Parameter	Description
Dimension	(Optional) A fine-grained category of metrics.
Metric	Metric for which you want to define one or more alarm rules. Metric: a metric in the metric set. For example, if Monitoring Item is set to Url and Metric Set is set to total , you can select the errorCount metric. Operator: operation to be performed. Threshold: threshold of the metric.

Parameter	Description
Alarm Condition	Condition for triggering an alarm. A: the number of collection periods. Range: 1–10. B: the number of times the alarm is triggered. Range: 1–10. This value cannot be greater than that of A. C: period (in minutes) during which identical alarms will not be sent. This period cannot be shorter than 10 minutes.
Recovery Policy	Condition for clearing an alarm.
Notification upon Recovery	Whether to notify recipients of alarm clearance.
Strict Mode	Checks all returned values against alarm rules to detect exceptions. For example, if a host has multiple CPU cores, there are multiple single-core CPU usages (such as cpu_no=0 and usage=98; cpu_no=1 and usage=99).

3. Alarm Notification

Table 13-8 Alarm notification parameters

Parameter	Description
Notification Content	Alarm details, which contain up to 500 characters. • If Strict Mode is enabled, the alarm notification content supports both Variable and Loop. If Strict Mode is disabled, only Variable is supported. • Alarm notification content. You can customize the content or select required metrics. • Select the metrics to be collected. On the right of the page, click  next to a metric. The metric will then be referenced to the notification content box.
Notification Object	• Select a notification object from the drop-down list. • Alarms will then be sent to the selected notification object. Unselected notification objects will not receive the alarm notifications. • Click  to refresh notification objects.

Step 6 Click **OK**.

----End

More Operations

After the alarm template is created, perform the operations listed in [Table 13-9](#) if needed.

Table 13-9 Related operations

Operation	Description
Editing an alarm policy	Click  in the Operation column.
Deleting an alarm policy	Click  in the Operation column.
Enabling or disabling an alarm policy	Click  or  in the Operation column to disable or enable an alarm policy, respectively.

13.3.8 Component Settings

Component settings include log association only.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **Link Trace > Metrics**.

Step 4 In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.

Step 5 Click the **Component Settings** tab.

----End

Associating Traces with Logs

Trace IDs can be associated with logs in Log Tank Service (LTS). Then you can find out the logs based on the associated trace IDs for fast troubleshooting.

- To associate traces with logs, you need to write trace IDs to logs first.
- If you need to print trace IDs to service logs, thread names must be printed to logs first. The following figure shows how to print thread names to logs. That is, add the **%thread** field to the **logback.xml** file.

- The Log4j, Log4j2, and Logback log components are supported for associating service logs.

Step 1 Under log association settings, enter related information.

Table 13-10 Parameters for log association settings

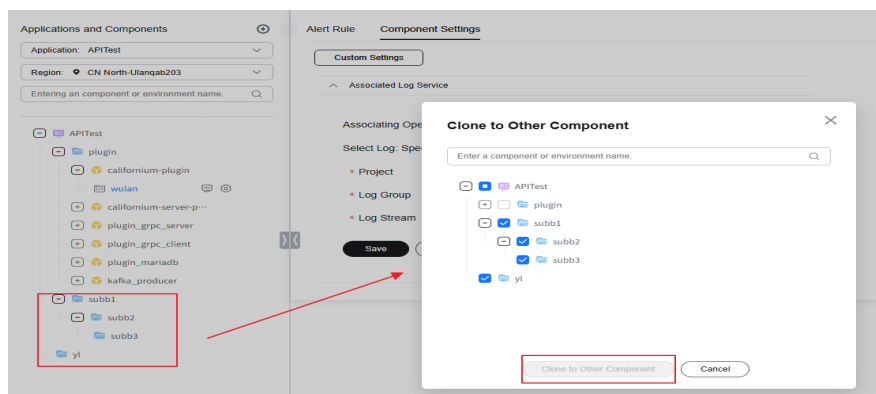
Parameter	Description
Project	Select a project from the drop-down list.
Log Group	A log group is a group of log streams which share the same log retention settings. Up to 100 log groups can be created for a single account. For how to create a log group, see Creating a Log Group .
Log Stream	A log stream is the basic unit for reading and writing logs. You can put different types of logs into different streams to ease management. For how to create a log stream, see Creating a Log Stream .

Step 2 Click **Save**.

Step 3 In the displayed dialog box, click **Confirm**.

Step 4 Perform other operations.

- Cloning the settings to other components
 - a. Click **Clone to Other Component**.
 - b. On the displayed page, select one or more components and click **Clone to Other Component**. The current log association setting will then be copied to the components you have selected.
- Clearing the association setting
 - a. Click **Clear Association Settings**. The **Clear Association Settings** dialog box is displayed.
 - b. On the displayed page, select one or more components and click **Clone to Other Component**. The current log association setting will then be cleared for the components you have selected.
- Before you clone settings to other components or clear association settings, ensure that the components you selected contain environments. Otherwise, the **Clone to Other Component** button will be grayed and your settings cannot be cloned.



----End

13.4 Tracing

When the calling between enterprise microservices is complex, the Agent samples some requests based on the preset sampling rate (see [Application Management](#)) and intercepts the corresponding requests and subsequent calling information. For example, in the scenario where service A calls service B and then calls service C, after service A receives a request, APM determines whether to trace the request based on the intelligent sampling algorithm.

Traces in this section refer to those collected using open tracing Agents.

Searching for traces

This function is used to search for span information, that is, the root span of a node. A trace can be found in multiple environments. For example, in the scenario where service A calls service B and then calls service C, the same trace may be found from services A, B, and C.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Link Trace > Tracing**.

Step 4 Specify criteria to filter traces.

----End

Table 13-11 Search criteria of traces

Search Criterion	Description	Mandatory
Application	Application to which the trace belongs.	Yes
Region	Region where the trace is located.	Yes
Component	Component to which the trace belongs.	No
Environment	Environment to which the trace belongs.	No
Instance	Instance to which the trace belongs.	No
URL	URL of the trace.	No
Call Method	HTTP method of the trace.	No

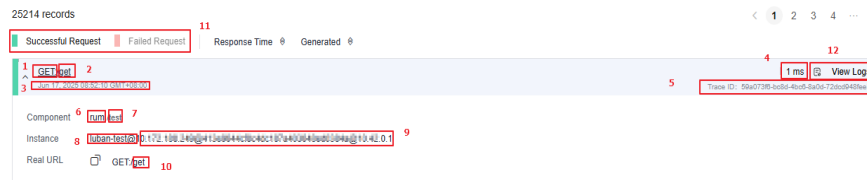
Search Criterion	Description	Mandatory
Response Time	Response time range of the trace. You can specify the minimum and maximum response time to search for the trace or leave them empty.	No
Exception or Not	Whether to filter the traces that are regarded as exceptions.	No
Trace ID	If you specify this parameter, other search criteria become invalid and the search will be performed based on the trace ID you specify.	No
Custom Parameter	- Search for traces by call parameter. The format is key=value. Example: thread.id=23. - Configure required parameters before you search for traces by custom parameter. If you have configured Key for Header Value Interception, Key for Parameter Value Interception, and Key for Cookie Value Interception for URL monitoring, you can set key=value to search. - On the span details page, the fixed display items do not support search by custom parameter. - If the value is too long, only the first line is used. Example: exception.stacktrace=java.lang.IndexOutOfBoundsException: Index: 1, Size: 0.	No

Checking trace details

Check basic information about the trace filtered based on the search criteria.

In the trace list, click the down arrow next to the target trace to check the basic information about the trace, as shown in the following figure.

Figure 13-11 Basic information about a trace



Parameter description:

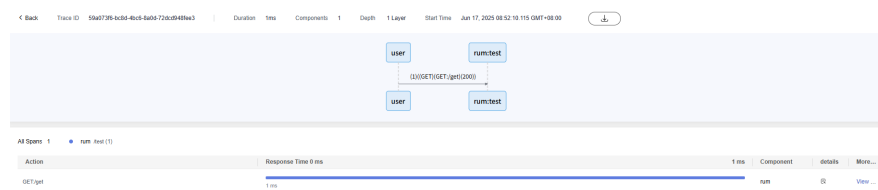
1. HTTP method of the trace
2. URL of the trace
3. Start time of the trace
4. Response time of the trace
5. Trace ID
6. Component to which the trace belongs
7. Environment to which the trace belongs
8. Instance to which the trace belongs
9. IP address of the instance to which the trace belongs
10. URL of the trace
11. Only the traces of successful or failed requests can be displayed. The following uses **Successful Request** as an example.
 - Click the green button next to **Successful Request**, only the traces of successful requests are displayed. The red button next to **Failed Request** becomes grayed.
 - If you click the green button again, both the traces of successful and failed requests are displayed. The red button is no longer grayed.
 - The green and red buttons cannot be grayed at the same time.
12. View logs.
 - a. On the LTS console, configure the collection. For details, see [Ingesting ECS Text Logs to LTS](#).
 - b. Click **View Logs** to go to the LTS page based on the trace ID. For details, see [Searching for Logs](#).
 - c. If the function of associating trace IDs with logs is not enabled, a warning dialog box is displayed. Click **Associate Now** to go to the log association settings page.

You can also click a specific URL on the monitoring item view page, for example, the table view of the URL monitoring item. In this way, you can quickly search for required trace information based on preset search criteria.

Checking the complete trace information on the trace details page

Click the name of a trace to view its details, as shown in the following figure.

Figure 13-12 Call details



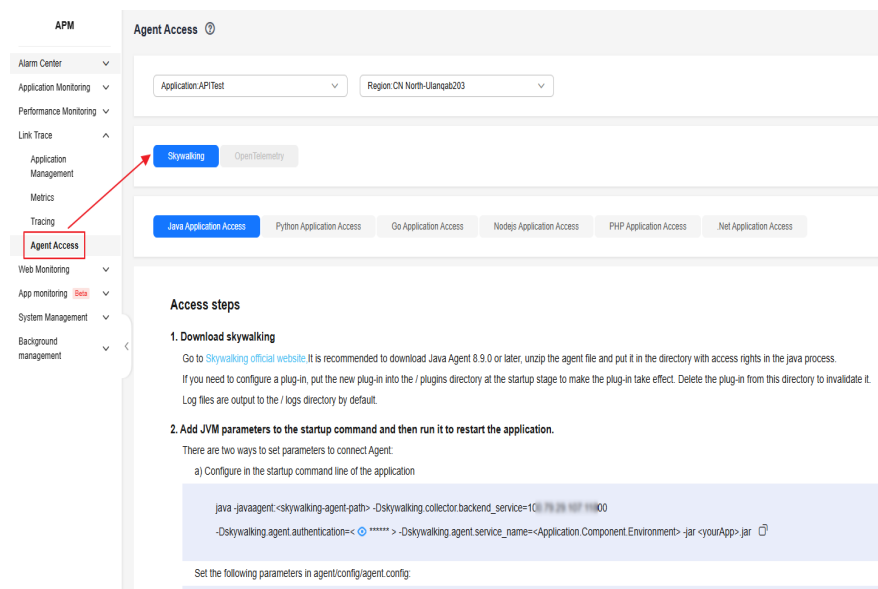
13.5 Agent Access

Currently, SkyWalking and OpenTelemetry Agents are supported.

SkyWalking Access

- Step 1** Log in to the [APM console](#).
- Step 2** Click  on the left and choose **Management & Governance > Application Performance Management**.
- Step 3** In the navigation pane, choose **Link Trace > Agent Access**.
- Step 4** Select a region and application.
- Step 5** Select **SkyWalking**. If the current application supports **SkyWalking**, **OpenTelemetry** will be grayed and cannot be selected.

Figure 13-13 Selecting SkyWalking



- Step 6** Select an access mode based on the application type and connect the Agent to the application by following the instructions. Note that each component under an application must have a unique name.

- Java application access

- a. Download SkyWalking.

Go to the [SkyWalking official website](#) to download Java Agent 8.9.0 or later (recommended). Decompress the package and save the Agent file to a directory that the Java process has the access permission.

To enable a plug-in, place the new plug-in to the **/plugins** directory during startup. To disable the plug-in, delete the plug-in from the directory.

By default, log files are output to the **/logs** directory.

- b. Add the startup command and restart the application.

There are two ways to configure the Agent:

- Configure the application startup command.

```
java -javaagent:skywalking-agent-path -
Dskywalking.collector.backend_service=***.***.***.***.***.*** -
Dskywalking.agent.authentication=***** -Dskywalking.agent.service_name=Application
name.Component name.Environment name -jar <User application>.jar
```

- Configure the **agent/config/agent.config** file.

```
collector.backend_service=100.79.29.107:11800
agent.authentication=*****
agent.service_name=Application name.Component name.Environment name
//Configure the Java Agent path and restart the application.
java -javaagent:skywalking-agent-path -jar User Application.jar
```

If you configure the Agent in both ways, only the configuration in the startup command takes effect.

- Python application access

- a. [Download SkyWalking](#).

- b. Install the Python Agent through PyPI.

```
pip install "apache-skywalking"
pip install "apache-skywalking[http]"
pip install "apache-skywalking[kafka]"
pip install apache-skywalking==0.7.0
```

After the SkyWalking Python Agent is installed using pip, the **sw-python** script will be installed in your environment.

- c. Connect the Agent.

To connect the Agent, you need to configure the access point, token, and application name in your startup file:

```
from skywalking import agent, config
config.init(
    collector_address='***.***.***.***.***.***',
    authentication='*****',
    service_name='apply name',
)
agent.start()
```

- d. Restart the application.

sw-python CLI provides the run option. You can use this option to run your application.

Original command:

```
gunicorn app.wsgi
```

New command:

```
sw-python run gunicorn app.wsgi
```

The SkyWalking Python Agent will then start with your application.

- Go application access

- a. [Download SkyWalking](#).

- b. Run the following command to install the SkyWalking Go SDK:

```
go get -u github.com/SkyAPM/go2sky-plugins/sql
```

- c. Connect the SDK to APM.

Import the SDK package to the **main.go** file.

```
import (  
    "github.com/SkyAPM/go2sky"  
    "github.com/SkyAPM/go2sky/reporter"  
)
```

Add the SDK to the main method.

```
r, err := reporter.NewGRPCReporter(***.***.***.***.***.***,  
    reporter.WithCDS(0),  
    reporter.WithAuthentication(*****)  
    if err != nil {  
        log.Fatalf("new reporter error %v \n", err)  
    }  
    defer r.Close()  
    tracer, err := go2sky.NewTracer(<Application name.Component name.Environment name>,  
    go2sky.WithReporter(r))  
    if err != nil {  
        log.Fatalf("create tracer error %v \n", err)  
    }  
    go2sky.SetGlobalTracer(tracer)
```

- Node.js application access

- a. [Download SkyWalking](#).

- b. Use the newly recommended library. The original module is **SkyAPM-nodejs**. Ensure that Node.js 10 or later is used. Run the following command on the console:

```
npm i skywalking-backend-js
```

- c. Connect the Agent.

To connect the Agent, you need to configure the access point, token, and application name in your startup file:

```
const { default: agent } = require('skywalking-backend-js');  
agent.start({  
    serviceName: 'Application name.Component name.Environment name',  
    collectorAddress: '***.***.***.***.***.***',  
    authorization: '*****'  
});
```

- d. Restart the application. Start the project using the common command. If no error is reported and data is reported normally, the startup is successful. If an error is reported, check the configuration and version. If no data is reported, check whether the token and application name are correct.

- PHP application access

- a. [Download SkyWalking](#).

- Use **pecl.net** (recommended) to install the Agent. You can run the following command on the console:

```
pecl install skywalking_agent
```

- Alternatively, install the Agent using the source code:

```
git clone --recursive https://github.com/apache/skywalking-php.git  
cd skywalking-php
```

```
phpize
./configure
make
make install
```

b. Connect the Agent.

Modify the following configuration items in the **php.ini** file:

```
[skywalking]
extension=skywalking.so
skywalking.app_code = Application name.Component name.Environment name
skywalking.enable = 1
skywalking.version = 8
skywalking.grpc = ***.***.***.***.***.***.***.***
skywalking.authentication = *****
skywalking.error_handler_enable = 0
```

c. Restart the application.

- Method 1: Set the value of **daemonize** to **no** in the **php-fpm.conf** file.

- Method 2: Run the **nohup** command to restart **php-fpm**:
`nohup /usr/local/services/php7/sbin/php-fpm > /usr/local/services/php7/log/php-fpm-output.log 2>&1 &`

• .NET application access

a. [Download SkyWalking](#).

Add the **SkyAPM .NET Core Agent** library.

```
dotnet add package SkyAPM.Agent.AspNetCore
```

b. Connect the Agent.

Download the configuration file generation tool **SkyAPM.DotNet.CLI**.

```
dotnet tool install -g SkyAPM.DotNet.CLI
```

Use the tool to generate the **skyapm.json** file.

```
dotnet skyapm config Application name.Component name.Environment name ***.***.***.***.***.***.***.***
```

Modify **Authentication** in the **skyapm.json** file.

```
Authentication: "*****"
```

Add the environment variables

ASPNETCORE_HOSTINGSTARTUPASSEMBLIES and

SKYWALKING__SERVICENAME to the **launchSettings.json** file under **Properties**.

```
environmentVariables: {
  "SKYWALKING__SERVICENAME": "Application name.Component name.Environment name",
  "ASPNETCORE_HOSTINGSTARTUPASSEMBLIES": "SkyAPM.Agent.AspNetCore"
}
```

----End

OpenTelemetry Access

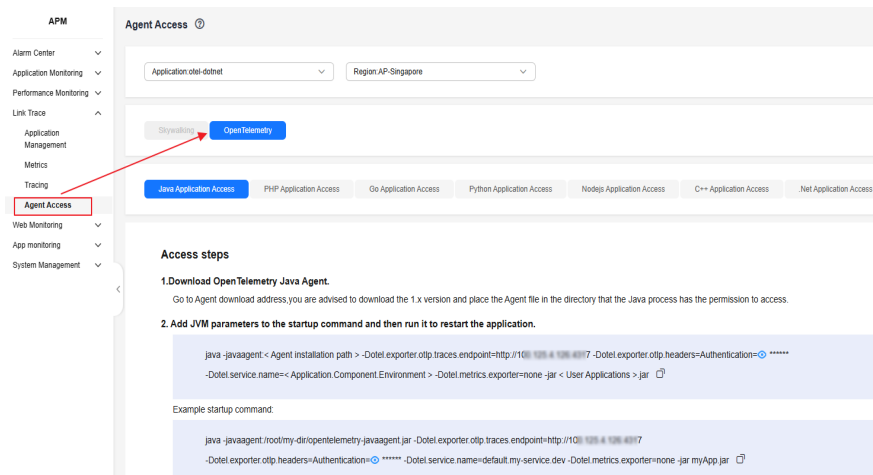
Step 1 In the navigation pane, choose **Link Trace > Agent Access**.

Step 2 Select a region and application.

Step 3 Select **OpenTelemetry**.

If the current application supports **OpenTelemetry**, **SkyWalking** will be grayed and cannot be selected.

Figure 13-14 Selecting OpenTelemetry



Step 4 Select an access mode based on the application type and connect the Agent to the application by following the instructions. Note that each component under an application must have a unique name.

- Java application access

- a. Download the OpenTelemetry Java Agent. [Download the Agent](#) (1.x is recommended). After the download, place the Agent file in a directory that the Java process can access.
- b. Add the startup command and restart the application.

```
java -javaagent:opentelemetry-agent-path -Dotel.exporter.otlp.traces.endpoint=http://  
*** ** *** -Dotel.exporter.otlp.headers=Authentication=***** -Dotel.service.name=Application  
name.Component name.Environment name -Dotel.metrics.exporter=none -jar User  
application.jar
```

Example of a startup command:

```
java -javaagent:/root/my-dir/opentelemetry-javaagent.jar -  
Dotel.exporter.otlp.traces.endpoint=http://100.79.29.106:4317 -  
Dotel.exporter.otlp.headers=Authentication=QxHHU69weF2c66b5DaCSNiGJAmqnRrH55DUikIjaf3  
ghVygIMMzgmUzH6hyE8GEOGVbXzjIXCPUjoYuUYEdjdvB6rm00nOZTXeADOfjSR84hKPNQui  
Qg39egulwtTzORW3pLHgi03D2mynBncuna1afpx0tCxEFrETXex2o7SPJRijJK9Z3k7PHTqgk7WiMV0  
hcMid7TLjMRPTqj7ziFOZk2EljkQi2DSpkbOKEOBE1iHEUeYO5dB167ic5 -  
Dotel.service.name=default.my-service.dev -Dotel.metrics.exporter=none -jar myApp.jar
```

- PHP application access

- Method 1: Non-intrusive access

- i. Prepare the environment.

PHP 8.0+, PECL, and composer. After the installation is complete, verify the PHP and composer versions in the shell script.

- ii. Install OpenTelemetry extension components.

- 1) Install the required build tool.

```
sudo apt-get install gcc make autoconf
```

- 2) Install the extension component using the build tool.

```
pecl install opentelemetry-beta  
pecl install grpc  
pecl install protobuf-3.8.0
```

- 3) Add the extension name to your **php.ini** file.

```
[opentelemetry]  
extension=opentelemetry.so  
[PHP]
```

```
extension=grpc.so
extension=protobuf.so
```

- 4) Check whether the extension has been installed and enabled.

```
php --ri opentelemetry
php --ri grpc
php --ri protobuf
```

- iii. Install the OpenTelemetry dependency packages required by the service framework.

- 1) Install the OpenTelemetry automatic plug-in package on which the service framework depends. For example, the Slim framework requires **open-telemetry/opentelemetry-auto-slim**.

```
composer config allow-plugins.php-http/discovery false
composer require open-telemetry/sdk
composer require open-telemetry/exporter-otlp
composer require open-telemetry/transport-grpc
composer require open-telemetry/opentelemetry-auto-slim
composer update
```

- 2) To connect the Agent, add the following content to the environment variables or **php.ini** file:

```
OTEL_EXPORTER_OTLP_ENDPOINT=http://***.***.***.***
OTEL_EXPORTER_OTLP_HEADERS=Authentication=*****
OTEL_EXPORTER_OTLP_PROTOCOL=grpc
OTEL_PHP_AUTOLOAD_ENABLED=true
OTEL_SERVICE_NAME=Application name.Component name.Environment name
```

- Intrusive access: See [official website](#).

- Go application access

Use the gRPC protocol to report data.

- a. Add OpenTelemetry dependencies.

```
import (
    "go.opentelemetry.io/contrib/instrumentation/net/http/otelhttp"
    "go.opentelemetry.io/otel"
    "go.opentelemetry.io/otel/baggage"
    "go.opentelemetry.io/otel/exporters/otlp/otlptrace"
    "go.opentelemetry.io/otel/exporters/otlp/otlptrace/otlptracegrpc"
    "google.golang.org/grpc"
    "go.opentelemetry.io/otel/propagation"
    "go.opentelemetry.io/otel/sdk/resource"
    sdktrace "go.opentelemetry.io/otel/sdk/trace"
    semconv "go.opentelemetry.io/otel/semconv/v1.4.0"
)
```

- b. Interconnect with OpenTelemetry.

```
const (
    SERVICE_NAME = "Application name.Component name.Environment name"
    GRPC_ENDPOINT="***.***.***.***"
    GRPC_TOKEN="*****"
)

func main() {
    init := initialization()
    defer init()
    grpc := otel.Tracer("Service name")
    method, _ := baggage.NewMember("method", "repl")
    client, _ := baggage.NewMember("client", "cli")
    bag, _ := baggage.New(method, client)
    context := baggage.ContextWithBaggage(context.Background(), bag)
    for {
        ctx, span := grpc.Start(context, "CreateSpan")
        createSpan(ctx)
        span.End()
        time.Sleep(time.Duration(1) * time.Second)
    }
}
```

```

func initialization() func() {
    ctx := context.Background()
    headers := map[string]string{"Authentication": APM_TOKEN}
    traceClientGrpc := oteltracegrpc.NewClient(
        oteltracegrpc.WithInsecure(),
        oteltracegrpc.WithEndpoint(APM_ENDPOINT),
        oteltracegrpc.WithHeaders(headers),
        oteltracegrpc.WithDialOption(grpc.WithBlock()))
    traceExp, err := oteltrace.New(ctx, traceClientGrpc)
    if err != nil {
        log.Fatalf("%s: %v", "can not create exporter", err)
    }
    res, err := resource.New(ctx,
        resource.WithFromEnv(),
        resource.WithProcess(),
        resource.WithTelemetrySDK(),
        resource.WithHost(),
        resource.WithAttributes(
            semconv.ServiceNameKey.String(SERVICE_NAME),
            semconv.HostNameKey.String(APM_ENDPOINT),
        ),
    )
    if err != nil {
        log.Fatalf("%s: %v", "can not create exporter", err)
    }
    bsp := sdktrace.NewBatchSpanProcessor(traceExp)
    tracerProvider := sdktrace.NewTracerProvider(
        sdktrace.WithSampler(sdktrace.AlwaysSample()),
        sdktrace.WithResource(res),
        sdktrace.WithSpanProcessor(bsp),
    )
    otel.SetTextMapPropagator(propagation.TraceContext{})
    otel.SetTracerProvider(tracerProvider)
    return func() {
        cxt, cancel := context.WithTimeout(ctx, time.Second)
        defer cancel()
        if err := traceExp.Shutdown(cxt); err != nil {
            otel.Handle(err)
        }
    }
}

func createSpan(ctx context.Context) {
    address := "Remote server address"
    client := http.Client {
        Transport: otelhttp.NewTransport(http.DefaultTransport),
    }
    req, err := http.NewRequestWithContext(ctx, "GET", address, nil)
    res, err := client.Do(req)
    if err != nil {
        log.Println(err)
    } else {
        res.Body.Close()
    }
}

```

- Python application access

- Non-intrusive access

- i. Download related packages.

```

pip install opentelemetry-distro
pip install opentelemetry-exporter-otlp
opentelemetry-bootstrap -a install

```

- ii. Run the following command to start the Python application.

To connect the Agent, you need to configure the token, endpoint, service name, and host name to your startup file:

```

opentelemetry-instrument --traces_exporter otlp --service_name Application
name.Component name.Environment name --resource_attributes

```

```
host.name=host.name --exporter_otlp_endpoint http://***.***.***.***:*** --  
exporter_otlp_headers Authentication=*****  
python myapp.py
```

- Node.js application access

- Intrusive access

- i. Add dependencies.

```
npm install --save @grpc/grpc-js @opentelemetry/exporter-trace-otlp-grpc  
@opentelemetry/instrumentation-express @opentelemetry/instrumentation-http  
@opentelemetry/resources @opentelemetry/sdk-node @opentelemetry/sdk-trace-base  
@opentelemetry/semantic-conventions
```

- ii. Interconnect with OpenTelemetry.

Add the following content to the top of the express entry file:

```
const { NodeSDK } = require("@opentelemetry/sdk-node");  
const { Resource } = require("@opentelemetry/resources");  
const { OTLPTraceExporter } = require("@opentelemetry/exporter-trace-otlp-grpc");  
const { ExpressInstrumentation } = require("@opentelemetry/instrumentation-express");  
const { HttpInstrumentation } = require("@opentelemetry/instrumentation-http");  
const { SemanticResourceAttributes } = require("@opentelemetry/semantic-  
conventions");  
const { BatchSpanProcessor } = require("@opentelemetry/sdk-trace-base");  
const grpc = require("@grpc/grpc-js");  
  
const service_name = "Application name.Component name.Environment name";  
const endpoint = "http://***.***.***.***:***";  
const authentication = "*****";  
  
const metadata = new grpc.Metadata();  
metadata.set("Authentication", authentication);  
const exporter = new OTLPTraceExporter({url: endpoint, metadata});  
const sdk = new NodeSDK({  
  spanProcessor: new BatchSpanProcessor(exporter),  
  resource: new Resource({  
    [SemanticResourceAttributes.HOST_NAME]: require("os").hostname(), // your host  
    name  
    [SemanticResourceAttributes.SERVICE_NAME]: service_name,  
  }},  
  instrumentations: [HttpInstrumentation, ExpressInstrumentation],  
});  
sdk.start();
```

- Non-intrusive access

For details, see the [official website](#).

- C++ application access

- a. [Use CMake to connect to OpenTelemetry](#).

- b. Check the environment requirements.

- i. C++ version ≥ 14 (earlier OpenTelemetry-cpp: C++ version ≥ 11)

- ii. Check [supported C++ versions and development platforms](#).

- c. Prepare the environment.

- i. Install the build tool Bazel. For details, see [Bazel Installation Guide](#).

- ii. Build the OpenTelemetry C++ library to set up the environment.

```
git clone https://github.com/open-telemetry/opentelemetry-cpp.git  
cd opentelemetry-cpp  
bazel build //...
```

- d. Use the OpenTelemetry C++ library in the project.

- Example **WORKSPACE** file:

```
load("@bazel_tools//tools/build_defs/repo:utils.bzl", "maybe")
load("@bazel_tools//tools/build_defs/repo:http.bzl", "http_archive")
load("@bazel_tools//tools/build_defs/repo:git.bzl", "git_repository")

http_archive(
    name = "io_opentelemetry_cpp",
    sha256 = "<sha256>", # specify sha256 based on the URL specified in the urls
    parameter.
    strip_prefix = "opentelemetry-cpp-1.10.0",
    urls = [
        "https://github.com/open-telemetry/opentelemetry-cpp/archive/refs/tags/v1.10.0.tar.gz"
    ],
)

# Load OpenTelemetry dependencies after load.
load("@io_opentelemetry_cpp//bazel:repository.bzl", "opentelemetry_cpp_deps")

opentelemetry_cpp_deps()

# (required after v1.8.0) Load extra dependencies required for OpenTelemetry
load("@io_opentelemetry_cpp//bazel:extra_deps.bzl", "opentelemetry_extra_deps")

opentelemetry_extra_deps()

# Load gRPC dependencies after load.
load("@com_github_grpc_grpc//bazel:grpc_deps.bzl", "grpc_deps")

grpc_deps()

# Load extra gRPC dependencies due to https://github.com/grpc/grpc/issues/20511
load("@com_github_grpc_grpc//bazel:grpc_extra_deps.bzl", "grpc_extra_deps")

grpc_extra_deps()
```

- **BUILD file:**

```
# Copyright The OpenTelemetry Authors
# SPDX-License-Identifier: Apache-2.0

cc_binary(
    name = "example_http_server",
    srcs = [
        "server.cc",
        "server.h",
    ],
    tags = ["ostream"],
    deps = [
        "@io_opentelemetry_cpp//api",
        "@io_opentelemetry_cpp//exporters/ostream:ostream_span_exporter",
        "@io_opentelemetry_cpp//ext:headers",
        "@io_opentelemetry_cpp//sdk/src/trace",
        "@io_opentelemetry_cpp//exporters/otlp:otlp_grpc_exporter",
    ],
)
```

- The test code contains the **server.cc** and **server.h** files.

server.cc file:

```
#include "server.h"
#include "opentelemetry/trace/context.h"
#include "opentelemetry/trace/semantic_conventions.h"

#include "opentelemetry/exporters/ostream/span_exporter_factory.h"
#include "opentelemetry/sdk/trace/exporter.h"
#include "opentelemetry/sdk/trace/processor.h"
#include "opentelemetry/sdk/trace/simple_processor_factory.h"
#include "opentelemetry/sdk/trace/tracer_context.h"
#include "opentelemetry/sdk/trace/tracer_context_factory.h"
#include "opentelemetry/sdk/trace/tracer_provider_factory.h"
#include "opentelemetry/trace/provider.h"
```

```
#include "opentelemetry/context/propagation/global_propagator.h"
#include "opentelemetry/context/propagation/text_map_propagator.h"
#include "opentelemetry/trace/propagation/http_trace_context.h"

#include <cstring>
#include <string>
#include <unistd.h>
#include <iostream>
#include <vector>
#include "opentelemetry/ext/http/client/http_client.h"
#include "opentelemetry/nostd/shared_ptr.h"

#include "opentelemetry/sdk/trace/tracer_provider.h"
#include "opentelemetry/exporters/otlp/otlp_grpc_exporter_factory.h"

#include <iostream>
#include <thread>

namespace trace = opentelemetry::trace;
namespace trace_sdk = opentelemetry::sdk::trace;
namespace otlp = opentelemetry::exporter::otlp;

namespace {

    using namespace opentelemetry::trace;
    namespace context = opentelemetry::context;
    opentelemetry::exporter::otlp::OtlpGrpcExporterOptions opts;

    uint16_t server_port = 8800;
    constexpr const char *server_name = "localhost";

    std::string getHostName() {
        char hostname[256];
        if (gethostname(hostname, sizeof(hostname)) == 0) {
            return hostname;
        } else {
            return "default_hostname";
        }
    }

    template<typename T>
    class HttpTextMapCarrier : public opentelemetry::context::propagation::TextMapCarrier {
    public:
        HttpTextMapCarrier(T &headers) : headers_(headers) {}

        HttpTextMapCarrier() = default;

        virtual opentelemetry::nostd::string_view Get(opentelemetry::nostd::string_view key)
        const noexcept override {
            std::string key_to_compare = key.data();
            // Header's first letter seems to be automatically capitalized by our test http-
            server, so
            // compare accordingly.
            if (key == opentelemetry::trace::propagation::kTraceParent) {
                key_to_compare = "Traceparent";
            } else if (key == opentelemetry::trace::propagation::kTraceState) {
                key_to_compare = "Tracestate";
            }
            auto it = headers_.find(key_to_compare);
            if (it != headers_.end()) {
                return it->second;
            }
            return "";
        }

        virtual void Set(opentelemetry::nostd::string_view key,
            opentelemetry::nostd::string_view value) noexcept override {
            headers_.insert(std::pair<std::string, std::string>(std::string(key), std::string(value)));
        }
    };
}
```

```
}

T headers_;
};

void InitTracer() {
    // Change this parameter based on the configuration of the access site.

    std::string endpoint = "http://**.*.*.***.***";
    std::string token = "*****";
    std::string serviceName = "Application name.Component name.Environment name";

    opts.endpoint = endpoint;
    opts.metadata.insert(std::pair<std::string, std::string>("authentication", token));
    // Create OTLP exporter instance
    auto exporter = otlp::OtlpGrpcExporterFactory::Create(opts);
    auto processor = trace_sdk::SimpleSpanProcessorFactory::Create(std::move(exporter));

    opentelemetry::sdk::resource::ResourceAttributes attributes = {
        {"service.name", serviceName},
        {"host.name", getHostName()}
    };
    auto resource = opentelemetry::sdk::resource::Resource::Create(attributes);

    std::shared_ptr<opentelemetry::trace::TracerProvider> provider =
    trace_sdk::TracerProviderFactory::Create(std::move(processor), std::move(resource));
    // Set the global trace provider
    trace::Provider::SetTracerProvider(provider);
}

void CleanupTracer() {
    // We call ForceFlush to prevent to cancel running exportings, It's optional.
    opentelemetry::nostd::shared_ptr<opentelemetry::trace::TracerProvider> provider =
    trace::Provider::GetTracerProvider();
    if (provider) {
        static_cast<trace_sdk::TracerProvider*>(provider.get())->ForceFlush();
    }

    std::shared_ptr<opentelemetry::trace::TracerProvider> none;
    trace::Provider::SetTracerProvider(none);
}

opentelemetry::nostd::shared_ptr<opentelemetry::trace::Tracer> get_tracer(std::string
tracer_name) {
    auto provider = opentelemetry::trace::Provider::GetTracerProvider();
    return provider->GetTracer(tracer_name);
}

class RequestHandler : public HTTP_SERVER_NS::HttpRequestCallback {
public:
    virtual int onHttpRequest(HTTP_SERVER_NS::HttpRequest const &request,
HTTP_SERVER_NS::HttpResponse &response) override {
        StartSpanOptions options;
        options.kind = SpanKind::kServer; // server
        std::string span_name = request.uri;

        // extract context from http header
        std::map<std::string, std::string> &request_headers =
            const_cast<std::map<std::string, std::string> &>(request.headers);
        const HttpTextMapCarrier<std::map<std::string, std::string>>
        carrier(request_headers);
        auto prop =
        context::propagation::GlobalTextMapPropagator::GetGlobalPropagator();
        auto current_ctx = context::RuntimeContext::GetCurrent();
        auto new_context = prop->Extract(carrier, current_ctx);
        options.parent = GetSpan(new_context)->GetContext();
    }
};
```

```
// start span with parent context extracted from http header
auto span = get_tracer("http-server")
->StartSpan(span_name,
    {{"server.address", server_name},
    {"server.port", server_port},
    {"http.method", request.method},
    {"url.scheme", "http"},
    {"http.request.body.size",
    static_cast<uint64_t>(request.content.length())},
    {"client.address", request.client}},
    options);

auto scope = get_tracer("http_server")->WithActiveSpan(span);

for (auto &kv : request.headers) {
    span->SetAttribute("http.header." + std::string(kv.first.data()), kv.second);
}
if (request.uri == "/helloworld") {
    span->AddEvent("Processing request");
    response.headers[HTTP_SERVER_NS::CONTENT_TYPE] =
HTTP_SERVER_NS::CONTENT_TYPE_TEXT;
    response.body = "Welcome to the APM demo!";
    span->End();
    return 200;
}
span->End();
return 404;
}
};
} // namespace

int main(int argc, char *argv[]) {
    InitTracer();

    // The port the validation service listens to can be specified via the command line.
    if (argc > 1) {
        server_port = (uint16_t) atoi(argv[1]);
    }

    HttpServer http_server(server_name, server_port);
    RequestHandler req_handler;
    http_server.AddHandler("/helloworld", &req_handler);
    auto root_span = get_tracer("http_server")->StartSpan(__func__);
    Scope scope(root_span);
    http_server.Start();
    std::cout << "Server is running..Press ctrl-c to exit...";
    while (1) {
        std::this_thread::sleep_for(std::chrono::seconds(100));
    }
    http_server.Stop();
    root_span->End();
    CleanupTracer();
    return 0;
}
```

server.h file:

```
// Copyright The OpenTelemetry Authors
// SPDX-License-Identifier: Apache-2.0

#pragma once

#include <atomic>
#include <string>
#include "opentelemetry/ext/http/server/http_server.h"

namespace {

    class HttpServer : public HTTP_SERVER_NS::HttpRequestCallback {
```

```
protected:
    HTTP_SERVER_NS::HttpServer server_;
    std::string server_url_;
    uint16_t port_;
    std::atomic<bool> is_running_{false};

public:
    HttpServer(std::string server_name = "test_server", uint16_t port = 8800) :
    port_(port) {
        server_.setServerName(server_name);
        server_.setKeepalive(false);
    }

    void AddHandler(std::string path, HTTP_SERVER_NS::HttpRequestCallback
    *request_handler) {
        server_.addHandler(path, *request_handler);
    }

    void Start() {
        if (!is_running_.exchange(true)) {
            server_.addListeningPort(port_);
            server_.start();
        }
    }

    void Stop() {
        if (is_running_.exchange(false)) {
            server_.stop();
        }
    }

    ~HttpServer() { Stop(); }
};

} // namespace
```

- .NET application access

- a. Add OpenTelemetry dependencies.

```
dotnet add package OpenTelemetry
dotnet add package OpenTelemetry.Exporter.OpenTelemetryProtocol
dotnet add package OpenTelemetry.Extensions.Hosting
```

- b. Interconnect with OpenTelemetry.

Add the following code to the top of the entry file:

```
using OpenTelemetry;
using OpenTelemetry.Resources;
using OpenTelemetry.Trace;
using OpenTelemetry.Exporter;
```

Add the following content between **WebApplication.CreateBuilder()** and **builder.Build()** in the entry file:

```
builder.Services.AddOpenTelemetry()
    .ConfigureResource(resource => resource
        .AddService(Application name.Component name.Environment name)
        .AddAttributes(new Dictionary<string, object> {
            ["host.name"] = System.Net.Dns.GetHostName()
        })
    )
    .WithTracing(tracing => tracing
        .AddAspNetCoreInstrumentation()
        .AddOtlpExporter(opt => {
            opt.Endpoint = new Uri(*** ** ***.***.***);
            opt.Headers = "Authentication=" + *****,
            opt.Protocol = OtlpExportProtocol.Grpc;
        })
    );
```

For details about non-intrusive access, see the [official website](#).

----End

14 App Monitoring

14.1 Overview

APM Agents regularly collect performance metric data to measure the overall health status of Android, iOS, and HarmonyOS apps. The collected data includes crashes, freezes, errors, startup performance, network requests, devices, and custom statistics, helping you better monitor your app running.

Constraints

- App monitoring is available only to whitelisted users.
- HarmonyOS startup performance data cannot be reported.

14.2 App List

Viewing the App List

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **App Monitoring > App List**. The list of all connected apps is displayed.

Table 14-1 App parameters

Parameter	Description
Mobile App	Name of a mobile app.
Application	Application to which the mobile app belongs.
Collection Status	Collection status of the app. Options: Normal and Disabled .

Parameter	Description
Created	Time when the mobile app is connected.
Operation	Operations that can be performed on the app. For details, see Table 14-9 .

----End

Setting an Alarm Policy

Step 1 In the navigation pane, choose **App Monitoring > App List**. Locate a target app and click **View Alarm Policy** in the **Operation** column to view the alarm policy list.

Step 2 Click **Add Alarm Policy**.

1. Basic information

Table 14-2 Basic information parameters

Parameter	Description
Policy Name	Custom name, which cannot be left blank. Only letters, digits, underscores (_), and hyphens (-) are allowed. Enter up to 512 characters.
Alarm Severity	Severity of an alarm. Options: COMMON and CRITICAL .
Monitored Object	Name of the mobile app to be monitored.
Monitoring Type	Select a monitoring type. – Options: Frame freezing monitoring collection , Crash monitoring collection , Device monitoring collection , Error monitoring collection , Customized statistics collection , and Launch performance collection . The information about the selected type is displayed on the right.
Metric Set	Select a target metric set. The information about the selected metric set is displayed on the right.

2. Alarm rule

Figure 14-1 Alarm rule

Alarm Rule

Dimension ⓘ

⊕

* Metric

Condition	Metric	Operator	Threshold	Operation
	Total number of API reqes	> ▾	0	⊕

* Alarm Condition

In collection periods, if alarms are triggered times, alarms will not be repeated for minutes.

A B C

* Recovery Policy

minutes since last alarm.

* Notification upon Recovery

☒

* Strict Mode ⓘ

☐

Table 14-3 Alarm rule parameters

Parameter	Description
Dimension	(Optional) A fine-grained category of metrics.
Metric	Metric for which you want to define one or more alarm rules.
Alarm Condition	Condition for triggering an alarm. A: 1-10 B: 1-10; not greater than A C: ≥ 10
Recovery Policy	Condition for clearing an alarm.
Notification upon Recovery	Whether to notify recipients of alarm clearance.
Strict Mode	Checks all returned values against alarm rules to detect exceptions.

3. Alarm Notification

Table 14-4 Alarm notification parameters

Parameter	Description
Notification Content	Alarm details, which contain up to 500 characters. - If Strict Mode is enabled, the alarm notification content supports both Variable and Loop. If Strict Mode is disabled, only Variable is supported. - Alarm notification content. You can customize the content or select required metrics. - Select the metrics to be collected. On the right of the page, click  next to a metric. The metric will then be referenced to the notification content box.
Notification Object	Select a notification object from the drop-down list. Alarms will then be sent to the selected notification object.

Step 3 Click **OK**.

Step 4 After the alarm template is created, perform the following operations if needed.

Table 14-5 Alarm policy-related operations

Operation	Description
Enabling or disabling an alarm policy	Click Disable or Enable in the Operation column to disable or enable an alarm policy, respectively.
Editing an alarm policy	Click Edit in the Operation column.
Deleting an alarm policy	Click Delete in the Operation column.

----End

Connecting a Mobile App

1. Connect an Android app.

Table 14-6 Connecting an Android app

Parameter	Description
Create Mobile App	• Application: Select an application from the drop-down list. • App Name: name of the mobile app to be created, which is used to distinguish it from other mobile apps.
Add Extended Configuration Item	• Select one or more SDK extended configuration items, or do not select any item. • Data of unselected items will not be displayed.
Track Service	Add an Agent to track the Android app. 1. Download a configuration file. 2. Place the file to the root directory of the project. 3. Add SDK dependencies.

2. Connect an iOS app.

Table 14-7 Connecting an iOS app

Parameter	Description
Create Mobile App	• Application: Select an application from the drop-down list. • App Name: name of the mobile app to be created, which is used to distinguish it from other mobile apps.
Add Extended Configuration Item	• Select one or more SDK extended configuration items, or do not select any item. • Data of unselected items will not be displayed.
Track Service	Add an Agent to track the iOS app. 1. Download a configuration file. 2. Place the file to the root directory of the project. 3. Add SDK dependencies.

3. Connect a HarmonyOS app.

Table 14-8 Connecting a HarmonyOS app

Parameter	Description
Create Mobile App	• Application: Select an application from the drop-down list. • App Name: name of the mobile app to be created, which is used to distinguish it from other mobile apps.
Add Extended Configuration Item	• Select one or more SDK extended configuration items, or do not select any item. • Data of unselected items will not be displayed.
Track Service	Add an Agent to track the HarmonyOS app. 1. Download a configuration file. 2. Place the file to the rawfile folder in the resource directory of the project. 3. Add SDK dependencies.

More Operations

After the mobile app is connected, perform the operations listed in [Table 14-9](#) if needed.

Table 14-9 Related operations

Operation	Description
Enabling or disabling an app	In the Operation column, click Disable to disable an app or click Enable to enable an app.
Checking the alarm policy	Click View Alarm Policy in the Operation column. For details, see Setting an Alarm Policy .
Editing an app	Click Edit in the Operation column.
Deleting an app	Click Delete in the Operation column.

14.3 Overview

APM Agents regularly collect performance metrics about apps, including the number of active devices, network error rate, startup time, and the number of crashes, ANRs, and errors. The **Overview** page displays the number of active devices, network error rate, startup response time, crash rate, ANR rate, and error rate.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **App Monitoring > Overview**.

Step 4 Select an application from the drop-down list in the upper left corner of the page. In the upper right corner of the page, select a time range. Default: **Last 20 minutes**.

The metrics displayed on the **Overview** page include the number of active devices, network error rate, startup time, and number of crashes, freezes, and errors.

Table 14-10 Overview parameters

Parameter	Description
Active Devices	Number of active devices.
Network Error Rate	Network error rate = Number of error packets/Total number of packets
Avg. Startup Time	Average response time for app startup.
Crash Rate	Crash rate = Number of devices that encounter crashes/Total number of devices
Freeze Rate	Freeze rate = Number of devices that encounter freezes/Total number of devices
Error Percentage	Error percentage = Number of devices that encounter errors/Total number of devices

----End

Active Devices

Move the cursor to the corresponding graph. The total number of devices at the specified time point will be displayed.

Table 14-11 Parameter related to active devices

Parameter	Description
Total Devices	Total number of active devices.

Network Error Rate

Move the cursor to the corresponding graph. The API failure rate at the specified time point will be displayed.

Table 14-12 Parameter related to the network error rate

Parameter	Description
Error Rate	Error rate = Number of failed APIs/Total number of APIs × 100%

Avg. Startup Time

Move the cursor to the corresponding graph. The average startup time at the specified time point will be displayed.

Table 14-13 Parameter related to the startup time

Parameter	Description
Avg. Response Time	Average response time for app startup.

Crashes

Move the cursor to the corresponding graph. The total number of crashes at the specified time point will be displayed.

Table 14-14 Parameter related to crashes

Parameter	Description
Total Crashes	Total number of app crashes.

ANRs

Move the cursor to the corresponding graph. The total number of ANRs at the specified time point will be displayed.

Table 14-15 Parameter related to ANRs

Parameter	Description
Total ANRs	Total number of app ANRs.

Errors

Move the cursor to the corresponding graph. The total number of errors at the specified time point will be displayed.

Table 14-16 Parameter related to errors

Parameter	Description
Total Errors	Total number of app errors.

14.4 Crashes

The **Crash** page displays the crash information, including the total number of crashes, device crash rate, number of affected devices, number of affected versions, crash trend, the five versions with the most crashes, the five devices with the most crashes, and crash list.

Checking Crashes

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **App Monitoring > Crashes**. The crash data is displayed.

Table 14-17 Crash parameters

Parameter	Description
Total Crashes	Total number of app crashes.
Device Crash Rate	Device crash rate = Number of devices that encounter crashes/Total number of devices
Startup Crash Rate	Startup crash rate = Total number of crashes/Total number of startups
Affected Devices	Number of affected devices.
Affected Versions	Number of affected versions.

- Select an application from the drop-down list in the upper left corner of the page.
- Select a carrier, region, device, version, and tag from the drop-down lists to view required information.

- In the upper right corner of the page, select a time range. Default: **Last 20 minutes**.

----End

Crash Trend

The crash trend graph displays the total number of crashes and the number of affected devices. Move the cursor to the graph. The total number of crashes and the number of affected devices will be displayed.

Table 14-18 Parameters of the crash trend graph

Parameter	Description
Total Crashes	Total number of app crashes.
Affected Devices	Number of affected devices.

The 5 Versions with the Most Crashes

A bar graph shows the 5 versions with the most crashes. Move the cursor to a version in the graph. The total number of crashes of the current version will be displayed.

Table 14-19 Related parameters

Parameter	Description
Total Crashes	Total number of crashes of a version.

The 5 Devices with the Most Crashes

A bar graph shows the 5 devices with the most crashes. Move the cursor to a device in the graph. The total number of crashes of the current device will be displayed.

Table 14-20 Related parameters

Parameter	Description
Total Crashes	Total number of crashes of a device.

Crash List

The crash list displays the crash summary, crash type, number of crashes, number of affected devices, number of affected versions, first occurrence time, last occurrence time, and operation.

Table 14-21 Crash list parameters

Parameter	Description
Crash Summary	Crash information summary.
Type	Crash type.
Crashes	Number of crashes.
Affected Devices	Number of devices affected by crashes.
Affected Versions	Number of versions affected by crashes.
First Occurred	Time when the first crash occurred.
Last Occurred	Time when the last crash occurred.

- In the search box in the upper left corner of the list, enter a keyword about the crash summary or type and click  to view the crashes that meet the search criteria.
- Click  in the upper right corner of the crash list and select the target metric data.
- By default, the crash list is sorted by **Crashes** in descending order.

Step 1 Click **Analyze** in the **Operation** column in the row that contains the target crash to view details. The details include the crash trend graph, 5 versions with the most crashes, 5 devices with the most crashes, and crash list.

- In the search box in the upper left corner of the crash list, enter a keyword about the device model/ID or crash type and click  to view the crashes that meet the search criteria.
- Click  in the upper right corner of the crash list and select the target metric data.
- By default, the crash list is sorted by occurrence time in descending order.

Step 2 Click **View Details** in the **Operation** column of the target device model/ID. The crash information and stack are displayed.

Stack analysis is supported only for iOS and Android, but not for HarmonyOS.

Figure 14-2 Crash - Stack analysis for iOS apps

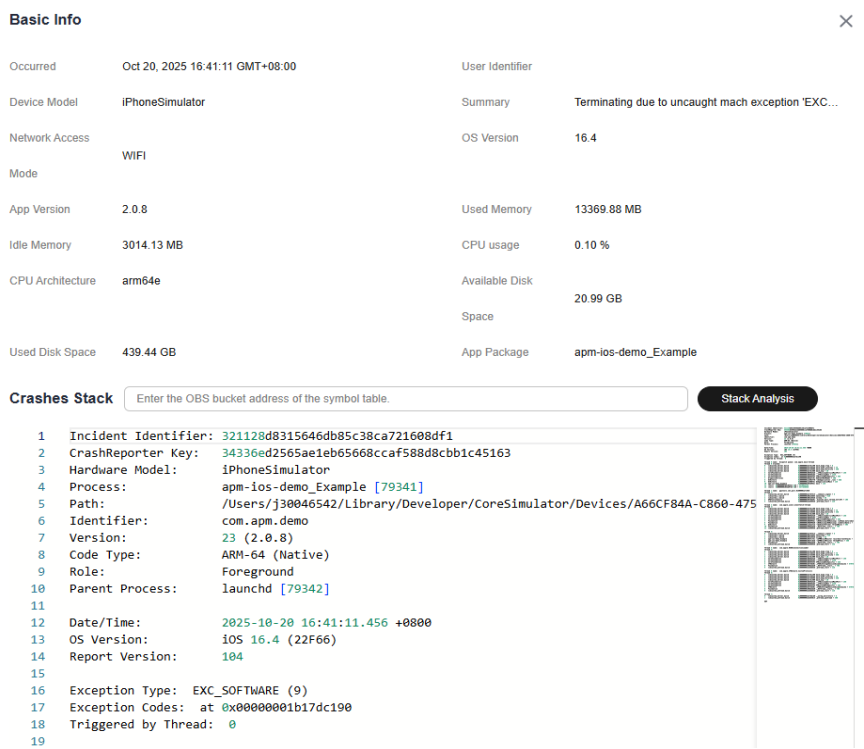


Figure 14-3 Crash - Stack analysis for Android apps

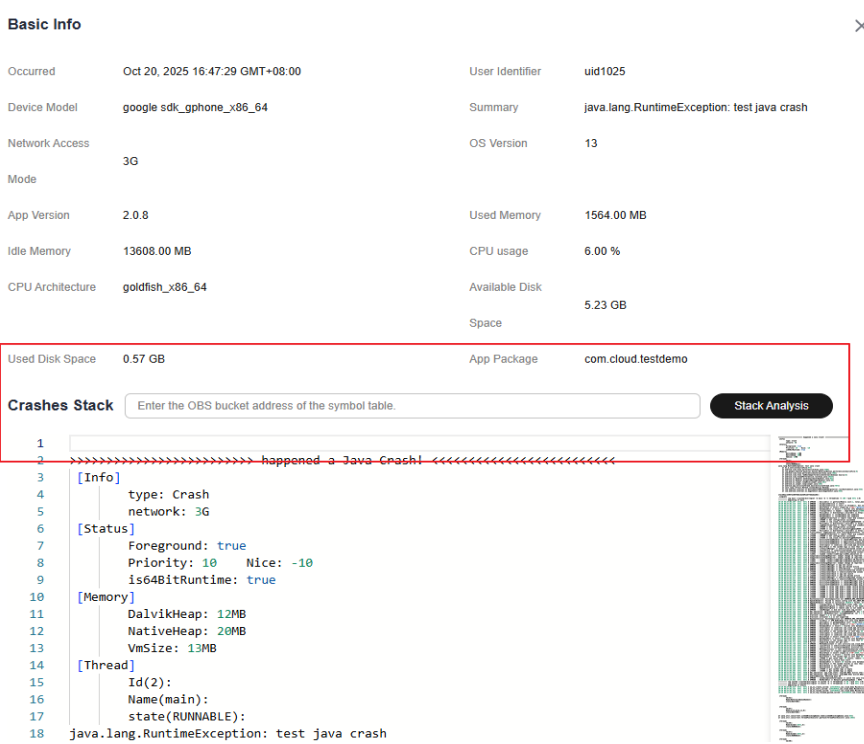


Table 14-22 Related parameters

Parameter	Description
Occurred	Time when the crash occurred.
User Identifier	Unique identifier of the user.
Device Model	Device model.
Summary	Crash information summary.
Network Access Mode	Network access mode of the app.
OS Version	OS version.
App Version	App version.
Used Memory	Used memory.
Idle Memory	Idle memory.
CPU Usage	CPU usage.
CPU Architecture	CPU architecture.
Available Disk Space	Available disk space.
Used Disk Space	Used disk space.
App Package	Name of the app package.
Crash Stack	Crash stack.

Step 3 Enter the OBS bucket address of the symbol table in the crash stack box. For details about how to upload the symbol table file, see [Using OBS Console](#).

Step 4 Click **Stack Analysis**. The analysis result is displayed in the box.

----End

14.5 Freezes

The **Freezes** page displays the freeze information, including the total number of freezes, device freeze rate, number of affected devices, number of affected versions, freeze trend, the five versions with the most freezes, the five devices with the most freezes, and freeze list.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **App Monitoring > Freezes**. The freeze data is displayed.

Table 14-23 Freeze parameters

Parameter	Description
Total Freezes	Total number of freezes.
Device Freeze Rate	Device freeze rate = Number of the devices that encounter freezes/Total number of devices
Affected Devices	Number of devices affected by freezes.
Affected Versions	Number of versions affected by freezes.

- Select an application from the drop-down list in the upper left corner of the page.
- Select a carrier, region, device, version, and tag from the drop-down lists to view required information.
- In the upper right corner of the page, select a time range. Default: **Last 20 minutes**.

----End

Freeze Trend

The freeze trend graph displays the total number of freezes and the number of affected devices. Move the cursor to the graph. The total number of freezes and the number of affected devices will be displayed.

Table 14-24 Parameters of the freeze trend graph

Parameter	Description
Total Freezes	Total number of freezes.
Affected Devices	Number of devices affected by freezes.

The 5 Versions with the Most Freezes

A bar graph shows the 5 versions with the most freezes. Move the cursor to a version in the graph. The total number of freezes of the current version will be displayed.

Table 14-25 Related parameters

Parameter	Description
Total Freezes	Total number of freezes that a version encounters.

The 5 Devices with the Most Freezes

A bar graph shows the 5 devices with the most freezes. Move the cursor to a device in the graph. The total number of freezes of the current device will be displayed.

Table 14-26 Related parameters

Parameter	Description
Total Freezes	Total number of freezes that a device encounters.

Freeze List

The freeze list displays the freeze summary, freeze type, number of freezes, number of affected devices, number of affected versions, first occurrence time, last occurrence time, and operation.

Table 14-27 Parameters related to the freeze list

Parameter	Description
Freeze Summary	Freeze information summary.
Type	Freeze type.
Freezes	Number of freezes.
Affected Devices	Number of devices affected by freezes.
Affected Versions	Number of versions affected by freezes.

Parameter	Description
First Occurred	Time when the first freeze occurred.
Last Occurred	Time when the last freeze occurred.

- In the search box in the upper left corner of the freeze list, enter a keyword about the freeze summary or type and click  to view the freezes that meet the search criteria.
- Click  in the upper right corner of the freeze list and select the target metric data.
- By default, the freeze list is sorted by **Freezes** in descending order.

Step 1 Click **Analyze** in the **Operation** column in the row that contains the target freeze to view details. The details include the freeze trend, 5 versions with the most freezes, 5 devices with the most freezes, and freeze list.

- In the search box in the upper left corner of the freeze list, enter a keyword about the device model/ID or freeze type and click  to view the freezes that meet the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the freeze list and select the target metric data.
- By default, the freeze list is sorted by occurrence time in descending order.

Step 2 Click **View Details** in the **Operation** column of the target device model/ID. The freeze information and stack are displayed.

Stack analysis is supported only for iOS and Android, but not for HarmonyOS.

Figure 14-4 Freeze - Stack analysis for Android apps

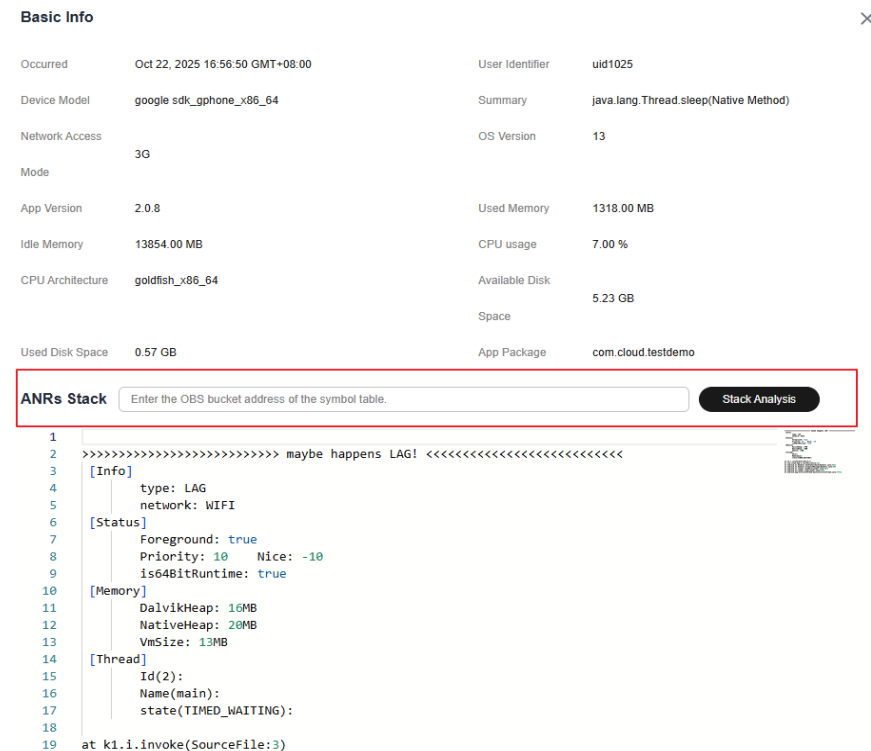


Table 14-28 Related parameters

Parameter	Description
Occurred	Time when the freeze occurred.
User Identifier	Unique identifier of the user.
Device Model	Device model.
Summary	Freeze information summary.
Network Access Mode	Network access mode of the app.
OS Version	OS version.
App Version	App version.
Used Memory	Used memory.
Idle Memory	Idle memory.
CPU Usage	CPU usage.

Parameter	Description
CPU Architecture	CPU architecture.
Available Disk Space	Available disk space.
Used Disk Space	Used disk space.
App Package	Name of the app package.
Freeze Stack	Freeze stack.

Step 3 Enter the OBS bucket address of the symbol table in the crash stack box. For details about how to upload the symbol table file, see [Using OBS Console](#).

Step 4 Click **Stack Analysis**. The analysis result is displayed in the box.

----End

14.6 Errors

The **Error** page displays the error information, including the total number of errors, device error rate, number of affected devices, number of affected versions, error trend, the five versions with the most errors, the five devices with the most errors, and error list.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **App Monitoring > Error**. The error data is displayed.

Table 14-29 Error parameters

Parameter	Description
Total Errors	Total number of errors.
Device Error Rate	Device error rate = Number of devices that encounter errors/Total number of devices
Affected Devices	Number of devices affected by errors.
Affected Versions	Number of versions affected by errors.

- Select an application from the drop-down list in the upper left corner of the page.
- Select a carrier, region, device, version, and tag from the drop-down lists to view required information.
- In the upper right corner of the page, select a time range. Default: **Last 20 minutes**.

----End

Error Trend

The error trend graph displays the total number of errors and the number of affected devices. Move the cursor to the graph. The total number of errors and the number of affected devices will be displayed.

Table 14-30 Parameters of the error trend graph

Parameter	Description
Total Errors	Total number of errors.
Affected Devices	Total number of devices affected by errors.

The 5 Versions with the Most Errors

A bar graph shows the 5 versions with the most errors. Move the cursor to a version in the graph. The total number of errors of the current version will be displayed.

Table 14-31 Related parameters

Parameter	Description
Total Errors	Total number of errors that a version encounters.

The 5 Devices with the Most Errors

A bar graph shows the 5 devices with the most errors. Move the cursor to a device in the graph. The total number of errors of the current device will be displayed.

Table 14-32 Related parameters

Parameter	Description
Total Errors	Total number of errors that a device encounters.

Error List

The error list displays the error summary, error type, number of errors, number of affected devices, number of affected versions, first occurrence time, last occurrence time, and operation.

- In the search box in the upper left corner of the error list, enter a keyword about the error summary or type and click  to view the errors that meet the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the error list and select the target metric data.
- By default, the error list is sorted by **Errors** in descending order.

Table 14-33 Parameters related to the error list

Parameter	Description
Error Summary	Error information summary.
Error Type	Error type.
Errors	Number of errors.
Affected Devices	Number of devices affected by errors.
Affected Versions	Number of versions affected by errors.
First Occurred	Time when the first error occurred.
Last Occurred	Time when the last error occurred.

Step 1 Click **Analyze** in the **Operation** column in the row that contains the target error to view details. The details include the error trend, 5 versions with the most errors, 5 devices with the most errors, and error list.

- In the search box in the upper left corner of the error list, enter a keyword about the device model/ID or error type and click  to view the errors that meet the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the error list and select the target metric data.
- By default, the error list is sorted by occurrence time in descending order.

Step 2 Click **View Details** in the **Operation** column of the target device model/ID. The error information and stack are displayed.

Stack analysis is supported only for iOS and Android, but not for HarmonyOS.

Figure 14-5 Error - Stack analysis for iOS apps

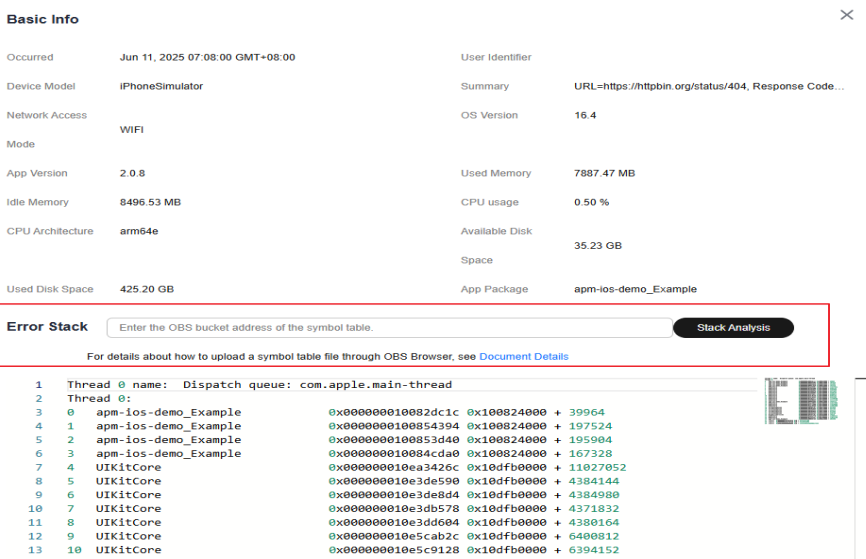


Table 14-34 Related parameters

Parameter	Description
Occurred	Time when an error occurred.
User Identifier	Unique identifier of the user.
Device Model	Device model.
Summary	Error information summary.
Network Access Mode	Network access mode of the app.
OS Version	OS version.
App Version	App version.
Used Memory	Used memory.
Idle Memory	Idle memory.
CPU Usage	CPU usage.
CPU Architecture	CPU architecture.
Available Disk Space	Available disk space.

Parameter	Description
Used Disk Space	Used disk space.
App Package	Name of the app package.
Error Stack	Error stack.

Step 3 Enter the OBS bucket address of the symbol table in the crash stack box. For details about how to upload the symbol table file, see [Using OBS Console](#).

Step 4 Click **Stack Analysis**. The analysis result is displayed in the box.

----End

14.7 Startup Performance

The **Start Analysis** page displays the startup information, including the number of failed startups, total number of startups, number of slow startups, number of normal startups, average response time, device startup trend, call time trend, and startup performance list.

Constraints

HarmonyOS startup performance data cannot be reported.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **App Monitoring > Start Analysis**. Select an application from the drop-down list in the upper left corner of the page. In the upper right corner of the page, select a time range. Default: **Last 20 minutes**.

Table 14-35 Startup performance parameters

Parameter	Description
Failed Startups	Number of startups that failed.
Total Startups	Total number of startups.
Slow Startups	Number of slow startups.
Normal Startups	Number of normal startups.
Avg. Response Time	Average response time for startup.

----End

Device Startup Trend

A bar graph shows the device startup trend. Move the cursor to the graph. The total number of startups, number of normal startups, number of slow startups, and number of failed startups at the specified time are displayed.

Table 14-36 Parameters of the startup trend graph

Parameter	Description
Total Startups	Total number of startups.
Normal Startups	Number of normal startups.
Slow Startups	Number of slow startups.
Failed Startups	Number of failed startups.

Call Time Trend

A bar graph shows the call time trend. Move the cursor to the graph. The average response time at the specified time point is displayed.

Table 14-37 Call time parameters

Parameter	Description
Avg. Response Time	Average response time of calls.

Startup Performance

Startup Performance displays information about the app startup performance on the device, including the device model/ID, total number of startups, number of normal startups, number of slow startups, number of failed startups, average response time, and operations.

Table 14-38 Startup performance parameters

Parameter	Description
Device Model/ID	Model or ID of the device.
Total Startups	Total number of startups.
Normal Startups	Number of normal startups.
Slow Startups	Number of slow startups.

Parameter	Description
Failed Startups	Number of failed startups.
Avg. Response Time	Average response time for startup.

- In the search box in the upper left corner of the startup performance list, enter a keyword about the device model/ID and click  to view the startup performance list that meets the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the startup performance list and select the target metric data.
- By default, the startup performance list is sorted by **Total Startups** in descending order.

Step 1 Click **Analyze** in the **Operation** column in the row that contains the target device to view details. The details include the device startup trend, call time trend, and startup performance list.

- In the search box in the upper left corner of the startup performance list, enter a keyword about the device model/ID and click  to view the startup performance list that meets the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the startup performance list and select the target metric data.
- By default, the startup performance list is sorted by occurrence time in descending order.

Step 2 Click **View Details** in the **Operation** column of a device model/ID. The device details are displayed.

Table 14-39 Parameters for startup performance details

Type	Parameter	Description
-	Device ID	Device ID.
-	User ID	User ID.
-	Occurred	Time when the device starts up.
Startup Analysis	App Pre-load Time	Time required for app pre-loading during startup.
	App Load Time	Time required for app loading during startup.

Type	Parameter	Description
	View Load Time	Time required for loading views during startup.
	Other Time	Other time consumed during startup.
Device Info	Device Model	Device model.
	Jail Break	Whether a device is jailbroken. Yes for jailbroken iOS devices and rooted Android devices, and No for other devices.
	Area	Area where the device is located.
	UUID	Unique ID reported each time.
	Network	Network access method.
OS	System	OS name.
	OS Version	OS version.
	App Package	App package name.
	App Version	App version.
	Received on Server	Time when the server received the data.
	SDK Version	SDK version.

----End

14.8 Network Requests

The **Network Request** page displays the network request information, including the number of app calls, request duration, API failure rate, slow request proportion, API failure rate trend, average call duration trend, network request list, and status code distribution.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **App Monitoring > Network Request**. Select an application from the drop-down list in the upper left corner of the page. In the upper right corner of the page, select a time range. Default: **Last 20 minutes**.

Table 14-40 Network request parameters

Parameter	Description
Calls	Number of calls.
Duration	Network request duration.
Error Rate	Error rate = Number of failed API calls/Total number of API calls
Slow Request Proportion	Slow request proportion = Number of slow requests/Total number of requests

----End

API Failure Rate Trend

The API failure rate trend graph displays the number of calls, proportion of slow requests, and API failure rate. Move the cursor to the graph. The number of calls, proportion of slow requests, and API failure rate at the specified time point are displayed.

Table 14-41 Parameters of the API failure rate trend

Parameter	Description
Calls	Number of API calls.
Slow Request Proportion	Slow request proportion = Number of slow API requests/Total number of API requests
Error Rate	Error rate = Number of failed API calls/Total number of API calls

Average Call Time Trend

A bar graph shows the average call time trend. Move the cursor to the graph. The average call time at the specified time point is displayed.

Table 14-42 Parameters related to average call time

Parameter	Description
Duration	Network request duration.

Network Requests

Network Requests display the network request information, including the API, request mode, total number of requests, request duration, number of slow requests, proportion of slow requests, number of errors, error rate, and operations.

Table 14-43 Network request parameters

Parameter	Description
API	API information.
Request Method	API request method.
Total Requests	Total number of API requests.
Duration	API request duration.
Slow Requests	Number of slow requests.
Slow Request Proportion	Slow request proportion = Number of slow API requests/Total number of API requests
Errors	Number of call errors.
Error Rate	Error rate = Number of API calls that encounter errors/Total number of API calls

- In the search box in the upper left corner of the network request list, enter a keyword about the API and click  to view the network requests that meet the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the network request list and select the target metric data.
- By default, the network request list is sorted by **Total Requests** in descending order.

Step 1 Click **Analyze** in the **Operation** column in the row that contains the target API to view details. The details include the API failure rate trend, average call duration trend, network request list, and status code distribution.

If a network request does not have a correct trace ID, no traces can be found. In that case, **Analyze** in the **Operation** column is grayed.

By default, the network request list is sorted by occurrence time in descending order.

Step 2 Click **View Details** in the **Operation** column in the row that contains the target device model/ID. The trace details page corresponding to the trace ID is displayed. For details, see [Checking Trace Details](#).

----End

Status Code Distribution

The distribution of network request status codes is displayed.

Table 14-44 Parameters related to status codes

Parameter	Description
Status Code	Status code.
Requests	Number of requests with a certain status code returned.
Requests with This Status Code	Proportion of requests with a status code = Number of requests with a certain status code returned/Total number of requests

14.9 Devices

The **Device** page displays the device information, including the app version, system version, device model, area, network access mode, and carrier statistics.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **App Monitoring > Device**. Select an application from the drop-down list in the upper left corner of the page. In the upper right corner of the page, select a time range. Default: **Last 20 minutes**.

- Enter a keyword in the search box in the upper left corner of each list and click  to view the items that meet the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the list and select the target metric data.

----End

App Version Statistics

The app version statistics table displays the app version information, including the app version, usage, number of devices, and operations. By default, the app version statistics table is sorted by usage in descending order.

Table 14-45 Parameters about app version statistics

Parameter	Description
App Version	App version.
Usage Count	Number of times that an app version is used.
Devices	Number of devices that the app version is used on.

- Step 1** Click **Analyze** in the **Operation** column of an app version. The app version details are displayed. The details include the device model/ID, app, version, system, system version, user Agent, area, user ID, occurrence time, and operations.
- In the search box in the upper left corner of the app version statistics table, enter a keyword about the device model/ID and click  to view the items that meet the search criteria. Note that each keyword is case sensitive.
 - Click  in the upper right corner of the app version statistics list and select the target metric data.
 - By default, the app version statistics table is sorted by occurrence time in descending order.
- Step 2** Click **View Details** in the **Operation** column of a device model/ID. The basic information of the app version is displayed.

Table 14-46 Basic information of the app version

Parameter	Description
Device Model	Device model.
User Identifier	Unique identifier of the user.
App	App name.
App Version	App version.
App ID	App ID.
Device Identifier	Unique identifier of the device.
System	System type.
OS Version	OS version.
CPU Architecture	CPU architecture.
User Agent	User Agent.
IP Address	IP address of the device.
Carrier	Name of the carrier.
Platform	Platform type.
Jail Break	Whether a device is jailbroken. Yes for jailbroken iOS devices and rooted Android devices, and No for other devices.
SDK	SDK version.
Network Access Mode	Network access mode.
Max. Memory	Maximum memory capacity.
Max. Disk Space	Maximum disk space.

Parameter	Description
Area	Area where the device is located.
UUID	Unique ID reported each time.
Occurred	Time when a device is connected.

----End

System Version Statistics

The system version statistics table displays the system version information of a device, including the system version, usage, number of devices, and operations.

Table 14-47 Parameters about system version statistics

Parameter	Description
OS Version	OS version.
Usage Count	Number of times that an OS version is used.
Devices	Number of devices that the OS version is used on.

Step 1 Click **Analyze** in the **Operation** column of a system version. The system version details are displayed. The details include the device model/ID, app, version, system, system version, user Agent, area, user ID, occurrence time, and operations.

- In the search box in the upper left corner of the list, enter a keyword about the device model/ID and click  to view the items that meet the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the list and select the target metric data.

Step 2 Click **View Details** in the **Operation** column of a device model/ID. The basic information of the system version is displayed.

Table 14-48 Basic information of the system version

Parameter	Description
Device Model	Device model.
User Identifier	Unique identifier of the user.
App	App name.
App Version	App version.
App ID	App ID.

Parameter	Description
Device Identifier	Unique identifier of the device.
System	System type.
OS Version	OS version.
CPU Architecture	CPU architecture.
User Agent	User Agent.
IP Address	IP address of the device.
Carrier	Name of the carrier.
Platform	Platform type.
Jail Break	Whether a device is jailbroken. Yes for jailbroken iOS devices and rooted Android devices, and No for other devices.
SDK	SDK version.
Network Access Mode	Network access mode.
Max. Memory	Maximum memory capacity.
Max. Disk Space	Maximum disk space.
Area	Area where the device is located.
UUID	Unique ID reported each time.
Occurred	Time when a device is connected.

----End

Device Model Statistics

The device model statistics table displays the device model information, including the device name, usage, number of devices, and operations.

Table 14-49 Parameters of device model statistics

Parameter	Description
Device Name	Name of a device.
Usage Count	Number of used devices.
Total Devices	Total number of devices.

- Step 1** Click **Analyze** in the **Operation** column of a device. The device model details are displayed. The details include the device model/ID, app, version, system, system version, user Agent, area, user ID, occurrence time, and operations.
- In the search box in the upper left corner of the list, enter a keyword about the device model/ID and click  to view the items that meet the search criteria. Note that each keyword is case sensitive.
 - Click  in the upper right corner of the list and select the target metric data.
- Step 2** Click **View Details** in the **Operation** column of a device model/ID. The basic information of the device model is displayed.

Table 14-50 Basic information of the device model

Parameter	Description
Device Model	Device model.
User Identifier	Unique identifier of the user.
App	App name.
App Version	App version.
App ID	App ID.
Device Identifier	Unique identifier of the device.
System	System type.
OS Version	OS version.
CPU Architecture	CPU architecture.
User Agent	User Agent.
IP Address	IP address of the device.
Carrier	Name of the carrier.
Platform	Platform type.
Jail Break	Whether a device is jailbroken. Yes for jailbroken iOS devices and rooted Android devices, and No for other devices.
SDK	SDK version.
Network Access Mode	Network access mode.
Max. Memory	Maximum memory capacity.
Max. Disk Space	Maximum disk space.
Area	Area where the device is located.
UUID	Unique ID reported each time.

Parameter	Description
Occurred	Time when a device is connected.

----End

Area Statistics

The area statistics table displays the area information of a device, including the area, usage, number of devices, and operations.

Table 14-51 Parameters of area statistics

Parameter	Description
Area	Area name.
Usage Count	Number of used devices in a region.
Devices	Total number of devices in a region.

Step 1 Click **Analyze** in the **Operation** column of an area. The area details are displayed. The details include the device model/ID, app, version, system, system version, user Agent, area, user ID, occurrence time, and operations.

- In the search box in the upper left corner of the list, enter a keyword about the device model/ID and click  to view the items that meet the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the list and select the target metric data.

Step 2 Click **View Details** in the **Operation** column of a device model/ID. The basic information is displayed.

Table 14-52 Basic information

Parameter	Description
Device Model	Device model.
User Identifier	Unique identifier of the user.
App	App name.
App Version	App version.
App ID	App ID.
Device Identifier	Unique identifier of the device.
System	System type.

Parameter	Description
OS Version	OS version.
CPU Architecture	CPU architecture.
User Agent	User Agent.
IP Address	IP address of the device.
Carrier	Name of the carrier.
Platform	Platform type.
Jail Break	Whether a device is jailbroken. Yes for jailbroken iOS devices and rooted Android devices, and No for other devices.
SDK	SDK version.
Network Access Mode	Network access mode.
Max. Memory	Maximum memory capacity.
Max. Disk Space	Maximum disk space.
Area	Area where the device is located.
UUID	Unique ID reported each time.
Occurred	Time when a device is connected.

----End

Network Access Statistics

The network access statistics table displays the network access information, including the network access mode, usage, number of devices, and operations.

Table 14-53 Parameters of network access statistics

Parameter	Description
Network Access Mode	Network access mode.
Usage Count	Number of devices that access Internet using this mode.
Devices	Number of devices.

Step 1 Click **Analyze** in the **Operation** column of a network access mode. The network access mode details are displayed. The details include the device model/ID, app,

version, system, system version, user Agent, area, user ID, occurrence time, and operations.

- In the search box in the upper left corner of the list, enter a keyword about the device model/ID and click  to view the items that meet the search criteria. Note that each keyword is case sensitive.
- Click  in the upper right corner of the list and select the target metric data.

Step 2 Click **View Details** in the **Operation** column of a device model/ID. The basic information is displayed.

Table 14-54 Basic information

Parameter	Description
Device Model	Device model.
User Identifier	Unique identifier of the user.
App	App name.
App Version	App version.
App ID	App ID.
Device Identifier	Unique identifier of the device.
System	System type.
OS Version	OS version.
CPU Architecture	CPU architecture.
User Agent	User Agent.
IP Address	IP address of the device.
Platform	Platform type.
Carrier	Name of the carrier.
Jail Break	Whether a device is jailbroken. Yes for jailbroken iOS devices and rooted Android devices, and No for other devices.
SDK	SDK version.
Network Access Mode	Network access mode.
Max. Memory	Maximum memory capacity.
Max. Disk Space	Maximum disk space.
Area	Area where the device is located.
UUID	Unique ID reported each time.

Parameter	Description
Occurred	Time when a device is connected.

----End

Carrier Statistics

The carrier statistics table displays the carrier information, including the carrier, usage, number of devices, and operations.

Table 14-55 Parameters of carrier statistics

Parameter	Description
Carrier	Carrier name.
Usage Count	Number of this carrier's devices that are used.
Devices	Number of devices.

- Step 1** Click **Analyze** in the **Operation** column of a carrier. The carrier details are displayed. The details include the device model/ID, app, version, system, system version, user Agent, area, user ID, occurrence time, and operations.
- In the search box in the upper left corner of the list, enter a keyword about the device model/ID and click  to view the items that meet the search criteria. Note that each keyword is case sensitive.
 - Click  in the upper right corner of the list and select the target metric data.
- Step 2** Click **View Details** in the **Operation** column of a device model/ID. The basic information is displayed.

Table 14-56 Basic information

Parameter	Description
Device Model	Device model.
User Identifier	Unique identifier of the user.
App	App name.
App Version	App version.
App ID	App ID.
Device Identifier	Unique identifier of the device.
System	System type.

Parameter	Description
OS Version	OS version.
CPU Architecture	CPU architecture.
User Agent	User Agent.
IP Address	IP address of the device.
Platform	Platform type.
Carrier	Name of the carrier.
Jail Break	Whether a device is jailbroken. Yes for jailbroken iOS devices and rooted Android devices, and No for other devices.
SDK	SDK version.
Network Access Mode	Network access mode.
Max. Memory	Maximum memory capacity.
Max. Disk Space	Maximum disk space.
Area	Area where the device is located.
UUID	Unique ID reported each time.
Occurred	Time when a device is connected.

----End

14.10 Custom Statistics

APM app monitoring supports custom statistics reporting.

Checking Custom Statistics List

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **App Monitoring > Custom Statistics**. Then select an application from the application drop-down list. The custom statistics of the application are displayed. By default, custom statistics are sorted by **Last Reported** in descending order.

1. Custom statistics parameters include **Event Name**, **Trigger Times**, **Devices**, and **Last Reported**.

Table 14-57 Custom statistics parameters

Parameter	Description
Event Name	Name of an event.
Trigger Times	Number of times that an event is triggered.
Devices	Number of devices that trigger the event.
Last Reported	Time when the last event was reported.

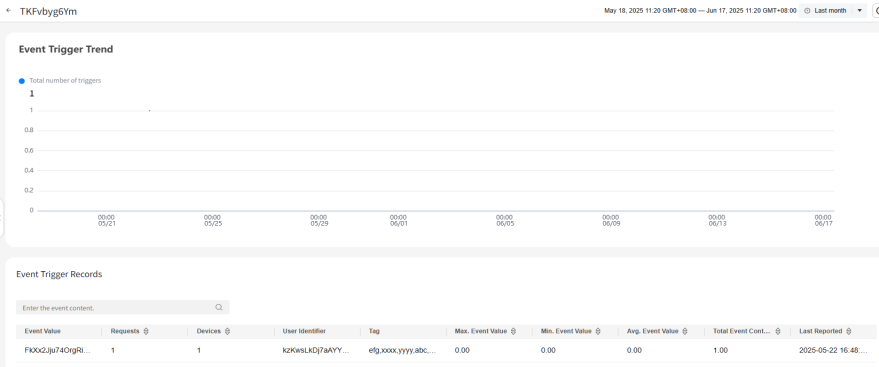
2. In the upper right corner of the **Custom Statistics** page, select a time range.
Default: **Last 20 minutes**.
Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.
3. In the search box, enter an event name and click . The custom statistics that meet the search criterion are displayed.
4. Click **Tracking Examples**. Event and time statistics tracking examples are displayed.
 - The format for Android or HarmonyOS tracking is "APMSDK(key, value)". **value** can be of the string or numeric type.
 - The format for iOS tracking is "[APMSDK(key, value)]". **value** can be of the string or numeric type.
 - If **value** is of the string type, the content of **value** will be displayed when custom statistics are queried.
If **value** is of the numeric type, -- will be displayed when custom statistics are queried.

----End

Checking Custom Statistics

- Step 1** In the navigation pane on the left, choose **App Monitoring > Custom Statistics**. Then select an application from the application drop-down list. The custom statistics of the application are displayed.
- Step 2** Click an event name in the **Event Name** column. **Event Trigger Trend** and **Event Trigger Records** are displayed.
- **Event Trigger Trend**: displays the total number of times that an event is triggered. Move the cursor to the event trigger trend graph. The event trigger details are displayed.
 - **Event Trigger Records**: displays the event content, number of requests, number of devices, user identifier, tag, maximum event value, minimum event value, average event value, event value sum, and last report time.
 - By default, event trigger records are sorted by **Last Reported** in descending order.

Figure 14-6 Event details



- Step 3 Click  next to an event name in the **Event Name** column. **Event Value**, **Trigger Times**, **Devices**, and **Last Reported** are displayed.
- Step 4 Click an event value in the **Event Name** column. **Event Trigger Trend** and **Event Trigger Records** are displayed.
 - Event Trigger Trend:** displays the total number of times that an event is triggered. Move the cursor to the event trigger trend graph. The event trigger details are displayed.
 - Event Trigger Records:** displays the event value, user identifier, tag, and last report time.

Table 14-58 Custom statistics parameters

Parameter	Description
Event Value	Event value. For details, see Custom Statistics .
Requests	Number of requests.
Devices	Number of devices.
User Identifier	User identifier.
Tag	Event tag.
Max. Event Value	Maximum event value.
Min. Event Value	Minimum event value.
Avg. Event Value	Average event value.
Event Value Sum	Sum of event values.
Last Reported	Time when the last custom event was reported.

----End

15 System Management

15.1 Collection Center

Collection Center displays collectors in a centralized manner. You can view and manage various collectors, metrics, and collection parameters supported by APM.

Checking Collector Details

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **System Management > Collection Center**.
All the supported collectors are displayed.

Step 4 In the collector list, click **View Details** in the **Operation** column in the row that contains the target collector. The collector details page is displayed.

Step 5 The collector details page consists of three modules: basic information, collection parameters, and metric sets.

- Basic information
This module displays collector information such as collector name and type.
- Collection parameters
This module displays the custom parameter settings supported by the collector. The settings take effect after being delivered to JavaAgents and are used for custom collection.
- Metric sets
This module displays information about the metrics collected by the collector.

----End

Collector

A collector is a plug-in for collecting metric data. It consists of the collector description, metric set, and collection parameters. Collector description describes

the data collected by a collector. Metric set is the data collected according to specifications. Collection parameters are the custom data to be collected.

- Data is collected by APM Agents. For example, Java performance data is collected by JavaAgents. The data collected by APM Agents must correspond to the data models of collectors' metric sets so that servers can process the data.
- The Agent of each language and framework defines its own collector.
- After a collector is added to an environment, it is instantiated as a monitoring item. This process is generally automated. APM Agents automatically discover collection plug-ins used by applications and add collectors to the environment to form monitoring items. For example, if a Java application connects to a database through the JDBC driver for MySQL, the MySQL collector is automatically added to the environment to form a monitoring item.

Collection Parameters

Collectors corresponding to monitoring items define collection parameters. You can modify collection parameters on the page as required. These parameters will be delivered to Agents with heartbeat parameters to change collection behaviors. By default, Redis instruction content is not collected for security purposes. If necessary, modify collection parameters to collect specific instruction data. Collection parameters can also be defined on environment tags. Collectors automatically inherit collection parameter attributes of corresponding environment tags. In this way, configuration is automated. For details about how to set collection parameters, see [Application Monitoring Configuration](#).

Metric Sets

A collector collects data of multiple metric sets. For example, the URL collector collects URL details, overall call condition, and status statistics. Each type of statistics corresponds to a metric set. Each metric set contains multiple metrics. For example, the metric set of URL details contains metrics such as the URL, method, number of calls, number of errors, and slowest call. Each metric corresponds to a data type.

APM supports the following types of metric data:

Table 15-1 APM metric data types

Data Type	Description	Description
ENUM	Enumeration	Primary key type. In the example of URL monitoring, the URL and method metrics are primary keys, and other metrics such as the number of calls correspond to the URL and method.
INT	Integer	Maximum size: 8 bytes

Data Type	Description	Description
DOUBLE	Floating-point number	8-byte floating-point number
STRING	String	Maximum length: 1,024 characters
CLOB	Large string	Maximum size: 1 MB
DATETIME	Time	Time is automatically displayed on the page.

15.2 Data Masking

You can set policies to mask the data reported using APM APIs.

Constraints

APM will collect and store masked data. Do not upload privacy or sensitive data to APM. If you need to upload such data, encrypt them.

Viewing a Data Masking Configuration

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the tree on the left, choose **System Management > Data Masking** and select your target node. The configuration information is displayed.

Step 4 In the search box, enter a configuration name keyword and click the search icon or press **Enter**.

----End

Adding a Data Masking Configuration

Step 1 In the navigation pane, choose **System Management > Data Masking** and select your target node.

Step 2 Click **Add Configuration** and set configuration parameters.

Table 15-2 Configuration parameters

Parameter	Description
Configuration Name	Used to identify a data masking configuration. This parameter cannot be empty. Enter up to 30 characters. Only letters, digits, and special characters are allowed.
Description	Used to describe the data masking configuration. This parameter cannot be empty. Enter up to 1,000 characters. Only letters, digits, and special characters are allowed.
Configuration Item	• Enter up to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. • The configuration item cannot be empty. By default, an empty configuration item is displayed. If you select Hash code, content will be replaced with a globally unique random character string. If you select Mask, content will be replaced with a fixed number of asterisks (*). By default, Mask is selected. • Click the plus sign (+) to add a configuration item, or click the minus sign (-) to delete one. • Each configuration can contain up to 20 configuration items. • The httpMethod, remoteAddr, exceptionType, contentType, charset, api_address, url, method, requestBody, responseBody, exceptionMsg, cookie, and Cookie fields have special functions in APM traces and do not support masking. • If you use one of these fields as a key, the system will display a message indicating that an invalid name exists.

Step 3 Click **Yes**.

Step 4 In the navigation pane, choose **Application Monitoring > Tracing**.

Step 5 Specify criteria to query traces and click  to view call parameters.

Step 6 On the **View Call Parameter** page, check the value of **transactionNames**. It is displayed as *********.

----End

Modifying a Data Masking Configuration

Step 1 In the navigation pane, choose **System Management > Data Masking** and select your target node.

Step 2 Click **Modify** in the **Operation** column to modify the configuration.

Step 3 Click **Yes**.

----End

Deleting a Data Masking Configuration

- Step 1** In the navigation pane, choose **System Management** > **Data Masking** and select your target node.
- Step 2** Click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes** to delete the configuration.
- Step 3** Select multiple data masking configurations and click **Batch Delete** above the list. In the displayed dialog box, click **Yes** to delete multiple data masking configurations at a time.

----End

15.3 Usage Statistics

After Agents are connected, you can check **Agent Statistics** on the **Usage Statistics** page.

Checking Agent Statistics

- Step 1** Log in to the [APM console](#).
- Step 2** Click  on the left and choose **Application** > **Application Performance Management**.
- Step 3** In the navigation pane, choose **System Management** > **Usage Statistics**. The **Application Monitoring Usage Statistics** page is displayed. By default, the **Agent Statistics** tab page is displayed. There are statistics about self-developed Agents only.

Agent Statistics: displays the total number of Agents, total number of Java Agents, total number of Node.js Agents, and historical Agent statistics.



- Click  and select a time range to display historical Agent statistics. Max.: 24 hours. Options: **Yesterday**, **Today**, and **Custom**.
- Click  to refresh historical Agent statistics.

----End

15.4 Access Keys

Access Key ID (AK) and Secret Access Key (SK) are your long-term identity credentials. JavaAgents report data with an AK. An AK is used together with an SK to sign requests cryptographically, ensuring that the requests are secret, complete, and correct.

Precautions

A user can create a maximum of two access keys with identical permissions and unlimited validity. Keep your access keys secure and change them periodically for security purposes. To change an access key, delete it and create a new one.

APM allows you to encrypt and decrypt the SK in the **apm.config** file.

The encryption and decryption process is as follows:

1. Compile a Java class, for example, **com.demo.DecryptDemo**, and add a decryption method, for example, decrypt both the input and output to character strings.
2. Compile the decryption method to decrypt the SK and return the decrypted value.
3. Pack the **com.demo.DecryptDemo** class into a JAR package and place this JAR package and its dependent packages in the **apm-javaagent/ext** folder of JavaAgent.
4. Add the following content to the **apm.config** file:
decrypt.className=com.demo.DecryptDemo
decrypt.methodName=decrypt
secret.key={Character string encrypted by users}

Adding an Access Key

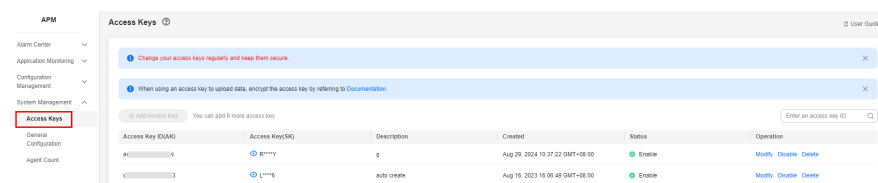
Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application > Application Performance Management**.

Step 3 In the navigation pane, choose **System Management > Access Keys**.

Step 4 On the page that is displayed, click **Add Access Key**.

Figure 15-1 Adding an AK/SK



Step 5 Enter an access key description and click **Yes** to generate an access key.

To modify the description, click **Modify** in the **Operation** column in the row that contains the target access key.

----End

Deleting an Access Key

Step 1 In the navigation pane, choose **System Management > Access Keys**.

Step 2 On the **Access Keys** page, locate the row that contains the target access key and click **Delete** in the **Operation** column.

Step 3 On the page that is displayed, click **Yes** to delete the access key.

----End

Enabling or Disabling an Access Key

Each access key is enabled by default. To disable it, do as follows:

Step 1 In the navigation pane, choose **System Management** > **Access Keys**.

Step 2 On the **Access Keys** page, locate the row that contains the target access key and click **Disable** in the **Operation** column.

Step 3 On the page that is displayed, click **Yes** to disable the access key.

To enable it again, click **Enable** in the row that contains the access key. On the page that is displayed, click **Yes**.

----End

15.5 General Configuration

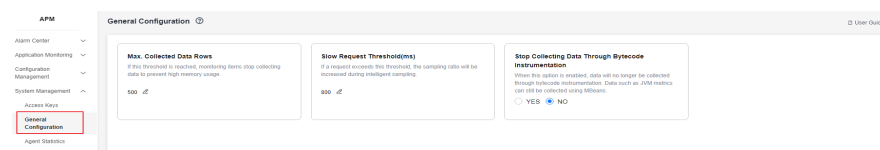
In **General Configuration**, you can determine whether to collect data through bytecode instrumentation, specify the slow request threshold and maximum number of rows to collect, and set web monitoring aggregation.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Application** > **Application Performance Management**.

Step 3 In the navigation pane, choose **System Management** > **General Configuration**.

Figure 15-2 Modifying general configuration



- **Stop Collecting Data Through Bytecode Instrumentation**

Enable or disable this function as required. Data such as JVM metrics will always be collected using MBeans. The default value is **No**.

Note: When the **Stop Collecting Data Through Bytecode Instrumentation** option is enabled, data will no longer be collected through bytecode instrumentation. Data such as JVM, GC, and Tomcat thread metrics can still be collected using MBeans.

- **Slow Request Threshold**

If this threshold is reached, more samples will be collected during intelligent sampling. The default value is **800**.

- **Max. Collected Data Rows**
If this value is reached, data will not be collected to prevent excessive memory usage. The default value is **500**.
- **Slow Request Threshold**
If a request exceeds this threshold, it is recorded as a slow SQL request.
- **Web Monitoring Aggregation**
If this function is enabled, aggregation will be implemented by URL or API based on the rules you have configured.
 - a. Click **Setting**.
 - b. On the displayed page, click the **URL Aggregation** or **API Aggregation** tab.
 - c. Click **Add Rule**.
 - d. In the displayed dialog box, set parameters.

Table 15-3 Parameters for adding an aggregation rule

Parameter	Description
Priority	Priority of an aggregation rule. Only digits are allowed. Range: 0–99. A larger value indicates a higher priority.
Aggregation Rule	(Mandatory) Example: furion/groups/*/detail .
Aggregation Target	(Mandatory) Example: furion/groups/UUID/detail .

- e. Click **Yes**.

----End

16 Performance Monitoring

16.1 Performance Monitoring Overview

To improve user experience, APM provides performance monitoring, which addresses issues such as a lack of O&M view, difficult-to-use topology, and insufficient association analysis.

Constraints

- Only whitelisted users can use performance monitoring.
- There are two editions for APM Agents: free and enterprise. After you switch from the free edition to the enterprise edition, the pay-per-use billing is used by default. You can purchase enterprise-edition [preferential packages](#) online. You can get discounts by buying these packages.

Functions

Table 16-1 Functions

Function	Description
Monitoring Component Performance Metrics	The component details page displays the monitoring items of components.
Viewing the Call Relationships Between Microservices Through Tracing	When the calls between enterprise microservices are complex, Agents sample some requests, and intercept corresponding requests and subsequent call information. You can view the call relationships between microservices through tracing.

Function	Description
Viewing the Call Relationships Between Applications in the Global Topology	On the topology, you can view the call relationships between services and check whether the calls between services are normal to quickly locate faults.
Managing Performance Monitoring Agents	You can enable or disable APM Agents.

Viewing Application Overview

The **Overview** page displays metric data by application, including **Overview**, **Response Time**, and **Error Distribution**. You can also click **Buy Package** and **Connect Application**.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **Performance Monitoring > Overview**. Metric data is displayed by application.

Figure 16-1 Viewing application overview

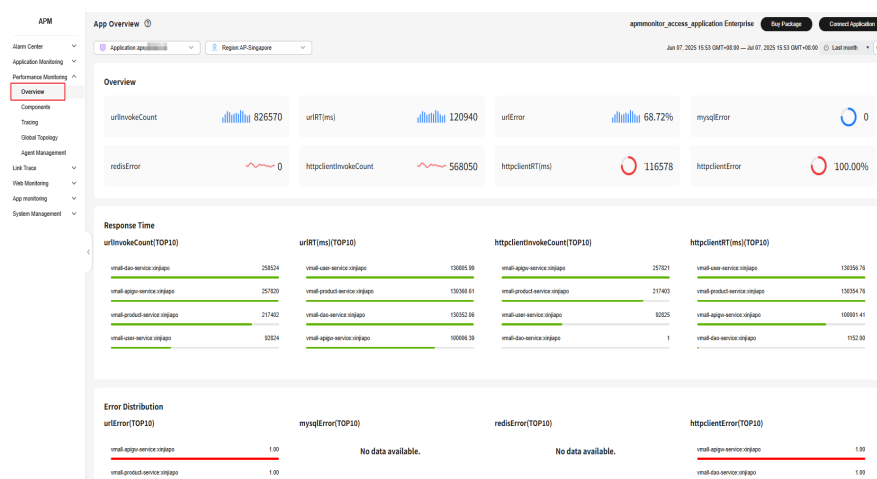


Table 16-2 Application overview metrics

Type	Metric	Description
Overview	URL Calls	Number of times that the URL is called.
	URL Avg. RT (ms)	Average response time of the URL in a collection period.
	URL Error Rate (%)	Error rate of the URL.
	MySQL Error Rate (%)	Error rate of MySQL.
	Redis Error Rate (%)	Error rate of Redis.
	HttpClient Calls	Number of HttpClient calls.
	HttpClient Avg. RT	Average response time of HttpClient.
	HttpClient Error Rate (%)	Error rate of HttpClient.
Response Time	URL Calls (Top 10)	Top 10 component environments sorted by the number of URL calls in descending order.
	URL Avg. RT (ms) (Top 10)	Top 10 component environments sorted by average URL response time in descending order.
	HttpClient Calls (Top 10)	Top 10 component environments sorted by the number of HttpClient calls in descending order.
	HttpClient Avg. RT (ms) (Top 10)	Top 10 component environments sorted by average HttpClient response time in descending order.
Error Distribution	URL Error Rate (%) (Top 10)	Top 10 component environments sorted by URL error rate in descending order.
	MySQL Error Rate (%) (Top 10)	Top 10 component environments sorted by MySQL error rate in descending order.
	Redis Error Rate (%) (Top 10)	Top 10 component environments sorted by Redis error rate in descending order.
	HttpClient Error Rate (%) (Top 10)	Top 10 component environments sorted by HttpClient error rate in descending order.

If the metrics displayed on the **Overview** page are not listed in [Table 16-2](#), see [Metric Overview](#).

Step 4 Select an application and a region to view metric data.

Step 5 Select a time range. Default: **Last 20 minutes**.

Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.

- Step 6** Click a green data bar in the **Response Time** area or a red data bar in the **Error Distribution** area to go to the monitoring item tab page of the current component. For details about the monitoring items, see [Monitoring Component Performance Metrics](#).

----End

Other Functions

Table 16-3 Other functions

Name	Description
Buying a preferential package	See Subscribing to Enterprise-Edition APM .
Connecting an application	See Monitoring Java Applications .

16.2 Monitoring Component Performance Metrics

The component details page displays components' monitoring items, including **Overview**, **Topology**, **Profiler Performance Analysis**, **Debugging Diagnosis**, **URL**, **Info**, **Exceptions**, **External Calls**, **Communication Protocols**, **Databases**, **Cache**, **Web Containers**, **Message Queues**, **RPCs**, **Logs**, **Infrastructure**, **Tracing**, and **Component Settings**.

Procedure

- Step 1** Log in to the [APM console](#).

- Step 2** Click  on the left and choose **Management & Governance > Application Performance Management**.

- Step 3** In the navigation pane, choose **Performance Monitoring > Components**.

- Step 4** In the filter bar on the left, select a region and an application.

- In the search box, specify an application and component to filter data.
- Click  in the upper right corner of the list and select the columns to display.

Table 16-4 Component metrics

Metric	Description
Component	Component name.

Metric	Description
Language	Programming language of the component.
Environment	Environment where the component is located.
Requests/s	Number of requests per second of the component.
Avg. RT	Average response time in a collection period.
Error Rate	Error rate of the component = Number of abnormal URL calls of the component/Total number of URL calls of the component
Alarms	Severity of the alarms generated by the component and the number of alarms. The alarm severity can be major or minor.
Application	Application where the component is located.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the name of a monitoring item. The corresponding monitoring item page is displayed. By default, the **Overview** page is displayed. For details about each monitoring item of the component, see [Table 16-5](#).

Table 16-5 Monitoring items of a component

Module	Description
Overview	The Overview page summarizes the performance monitoring data of component instances and displays Metric Overview , Performance Analysis , Resource Usage , and Top 5 APIs of the component in a specified period (max.: 30 days). For details, see Viewing the Component Overview .
Topology	The topology displays the call relationships between services within a period. The statistics can be collected from the caller or the callee. You can also view the trend. For details, see Viewing the Call Relationships Between Components in the Topology .
URL	This type of monitoring item monitors the calls of the current application by external services. It includes URL, Dubbo server, CSE server, CSEProvider cluster, and FunctionGraph monitoring. This type of monitoring item reflects the actual external status of the entire service. For example, if the average response time of a URL is long, it means that external users take a long time to query data. For details, see Viewing the URL Monitoring Item of a Component .

Module	Description
Info	This type of monitoring item monitors JVMInfo, JVMMonitor, GC, threads, and JavaMethod. For details, see Viewing the Info Monitoring Item of a Component.
Exceptions	This type of monitoring item monitors application exception logs. Take the monitoring of Java exception logs as an example. Once you use the log system to print logs, they will be collected by APM. The types of exceptions to be collected vary depending on the collector type. For details, see Viewing the Exception Monitoring Item of a Component.
External Calls	This type of monitoring item monitors the calls of external services by the current application. It includes CSEConsumer, ApacheHttpClient, ApacheHttpAsyncClient, DubboConsumer, and HttpClient. For details, see Viewing the External Call Monitoring Item of a Component.
Communication Protocols	This type of monitoring item monitors communication protocols, including the WebSocket protocol. For details, see Viewing the Communication Protocol Monitoring Item of a Component.
Databases	This type of monitoring item monitors database access. The databases that can be monitored include the C3PO, Cassandra, ClickHouse, DBCP, Druid, EsRestClient, GaussDB, Hikari, Jetcd, ObsClient, MySQL, PostgreSQL, Oracle, HBase, and MongoDB. APM collects statement execution details to help you locate performance problems in code. For details, see Viewing the Database Monitoring Item of a Component.
Web Containers	This type of monitoring item monitors web containers, including Tomcat. For details, see Viewing the Web Container Monitoring Item of a Component.
Message Queues	This type of monitoring item monitors message queues, including KafkaProducer. For details, see Viewing the Message Queue Monitoring Item of a Component.
RPCs	This type of monitoring item monitors remote procedure calls, including GRPCServer. For details, see Viewing the RPC Monitoring Item of a Component.

Module	Description
Logs	You can enter a keyword and specify a time range to search for logs. For details, see Viewing the Log Monitoring Item of a Component .
Infrastructure	Different graphs (such as line graphs and digit graphs) can be displayed on one dashboard, so you can view monitoring data comprehensively. For details, see Viewing the Infrastructure of a Component .
Tracing	When the calls between enterprise microservices are complex, APM Agents sample some requests, and intercept corresponding requests and subsequent call information. For example, in the scenario where service A calls service B and then calls service C, after service A receives a request, APM determines whether to trace the request based on the intelligent sampling algorithm. For details, see Viewing the Traces of a Component .
Component Settings	Component settings include log association, Profiler, component/infrastructure association analysis, full sampling, and database monitoring settings. For details, see Configuring Component Settings for Performance Monitoring .

----End

16.2.1 Configuring Component Settings for Performance Monitoring

APM component settings:

- **Custom Settings:** You can configure log association, component/infrastructure association analysis, Profiler, full sampling, and database monitoring settings.
- **Configuring an Agent:** You can enable or disable APM Agents.
- **Configuring a Monitoring Item:** You can view monitoring items and enable or disable them.

Custom Settings

You can configure [log association](#), [Profiler](#), [component/infrastructure association analysis](#), [full sampling](#), and [database monitoring](#) settings.

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and application to filter components.

You can also specify an application and component in the search box to filter the target component.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Component Settings** tab. On the displayed page, perform the following operations:

- Log Association Settings:

You can associate trace IDs with logs in LTS. If a fault occurs, you can quickly find out the logs based on the associated trace IDs for troubleshooting.

- a. In the **Log Association Settings** area on the **Custom Settings** page, enable or disable **Associate Log with Trace ID**.
 - If this option is enabled, trace IDs will be automatically generated in service logs.
 - If this option is disabled, the log association settings below this option will not take effect.
 - Only Log4j, Log4j2, and Logback log components support the association.
 - Log association settings can be configured for Java components only.
- b. In the **Log Association Settings** area, set parameters by referring to the following table. After the configuration is complete, the **Save** and **Clone to Other Component** buttons turn black.

Table 16-6 Parameters for log association settings

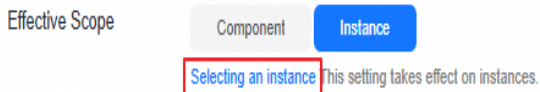
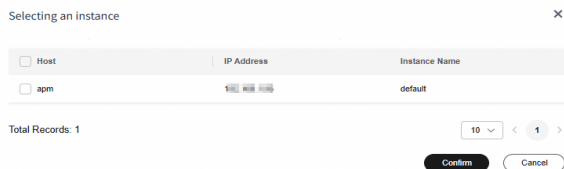
Parameter	Description
Project	Select a project from the drop-down list.
Log Group	Select a log group. A log group is a group of log streams which share the same log retention settings. Up to 100 log groups can be created for a single account. For details about how to create a log group, see Creating a Log Group .
Log Stream	Select a log stream. A log stream is the basic unit for reading and writing logs. You can put different types of logs into different streams to ease management. For details about how to create a log stream, see Creating a Log Stream .

- c. Click **Save**.
- d. In the displayed dialog box, click **Confirm**.

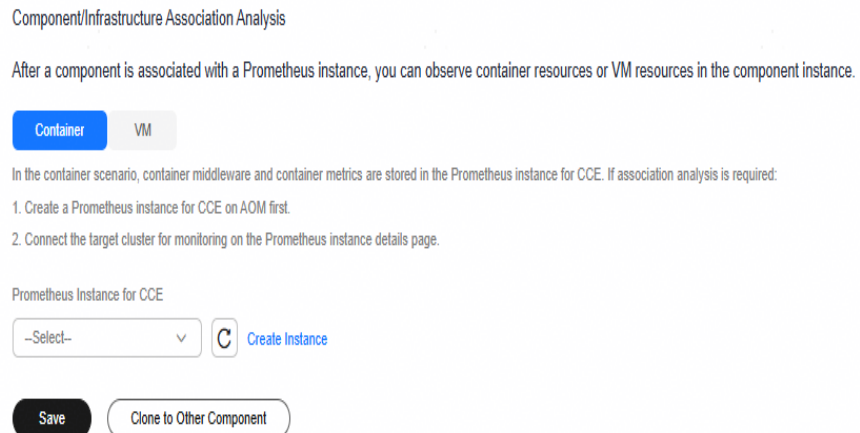
- Profiler Settings
 - a. In the **Profiler Settings** area on the **Custom Settings** page, set parameters by referring to the following table.
 - Profiler settings can be configured for Java components only.
 - To use Profiler for performance analysis, install [an Agent version that supports Profiler](#).

Table 16-7 Parameters for Profiler settings

Parameter	Description
Enable/Disable Profiler	Enable or disable Profiler: ▪ Enable: APM performs Profiler analysis. APM will then diagnose applications to solve performance bottlenecks caused by high CPU/memory usage or latency. ▪ Disable: APM does not perform Profiler analysis. This option is disabled () by default. When Enable/Disable Profiler is disabled, all the options below are disabled.
CPU	Enable or disable CPU: ▪ Enable: APM collects CPU flame graph information during application running. ▪ Disable: APM does not collect CPU flame graph information during application running. This option is disabled () by default.
Memory	Enable or disable Memory. ▪ Enable: APM collects heap memory flame graph information during application running. ○ Live Object Memory: heap memory that has been allocated but has not been reclaimed. It is used in the memory leakage scenario. Ensure that JDK17 is used. ○ Allocated Memory: allocated heap memory (including the memory that has been reclaimed and that has not been reclaimed). It can be used in the frequent Garbage Collection (GC) scenario. ▪ Disable: APM does not collect heap memory flame graph information during application running. This option is disabled () by default.

Parameter	Description
Latency	Enable or disable Latency. ▪ Enable: APM collects latency flame graph information during application running. ▪ Disable: APM does not collect latency flame graph information during application running. This option is disabled () by default.
Effective Scope	Set the effective scope: ▪ Component: This setting takes effect in a component. Profiler data is generated for all instances associated with the component. The default effective scope is Component. ▪ Instance: This setting takes effect in instances. Profiler data is generated for the selected instances only. If the effective scope is set to Instance, you need to select instances. 1. Click Selecting an instance under Instance.  2. Select one or more instances and click Confirm. 

- Click **Save**.
 - In the displayed dialog box, click **Confirm**.
- Component/Infrastructure Association Analysis
 - In the **Component/Infrastructure Association Analysis** area on the **Custom Settings** page, select a scenario and a Prometheus instance. After components are associated with Prometheus instances, you can monitor containers or VMs on the component details page.

Figure 16-2 Component/Infrastructure association analysis

- i. **Container:** Container middleware and container metrics are stored in the Prometheus instance for CCE.
If association analysis is required, [create a Prometheus instance for CCE](#) in AOM and then connect the target cluster for monitoring on the Prometheus instance details page.
 - ii. **VM:** Metrics of resources such as VMs and self-built middleware are stored in the Prometheus instance for ECS.
To perform association analysis, [create a Prometheus instance for ECS](#) in AOM and then install the Node Exporter or self-built middleware Exporter in AOM's access center.
- b. Click **Save**.
- **Full Sampling Settings**
 - To enable full sampling, go to [JavaAgent Updates](#) and select a proper version. A higher sampling ratio means more resources consumed. Ensure that your resources are sufficient before setting a sampling ratio.
 - When the sampling ratio is set to 100%, the CPU usage is about 5%, 10%, or 20% higher than that of intelligent sampling in the case of 500 TPS, 1,000 TPS, or 2,000 TPS.
 - Full sampling can be configured for Java components only.
 - a. In the **Full Sampling Settings** area on the **Custom Settings** page, select a sampling policy.
 - **Intelligent:** Intelligent algorithms are used for sampling. For details, see [Intelligent Sampling Algorithm](#).
 - **Global:** Set a sampling ratio for all traces.

Table 16-8 Global sampling parameters

Sampling Ratio Name	Default Value	Range	Description
Maximum sampling threshold per second	100	0–200	- Default value: 100 This parameter is set to protect services from being affected by sampling in large traffic scenarios. If you set a value greater than the default value, additional system resources will be consumed. Exercise caution. - Ensure that Agent 2.5.5 or later is installed.
Global Sampling Ratio	100%	0%–100%	Set a sampling ratio for all traces.

- **Custom:** Set a custom sampling ratio as required.

Table 16-9 Parameters of custom sampling

Sampling Ratio Name	Default Value	Range	Description
Maximum sampling threshold per second	100	0–200	- This parameter is set to protect services from being affected by sampling in large traffic scenarios. If you set a value greater than the default value, additional system resources will be consumed. Exercise caution. - Ensure that Agent 2.5.5 or later is installed.
Successful Request Sampling Ratio	100%	0%–100%	Set a success request sampling ratio. Successful requests do not return the defined error codes.

Sampling Ratio Name	Default Value	Range	Description
URL Sampling Ratio	100%	0%–100%	The URL sampling ratio takes effect only for successful requests. Click Add and set the following parameters: ○ URL: URL path. Example: <code>/apm2/health/v1/health-check</code>. ○ Sampling Ratio: 0%–100%. To delete the configured URL sampling ratio, click Delete. The URL refers to the URL with RT Threshold defined in URL Configuration of the URL monitoring item. For details about the configuration, see Configuring the URL Monitoring Item.
Error Request Sampling Ratio	100%	0%–100%	Set an error request sampling ratio. An error request refers to the request that returns the Error Code defined in the URL monitoring item. For details about the configuration, see Configuring the URL Monitoring Item.
Slow Request Sampling Ratio	100%	0%–100%	Set a slow request sampling ratio. A slow request refers to the request whose response time is greater than Slow Request Threshold defined in the URL monitoring item. For details about the configuration, see Configuring the URL Monitoring Item.

- b. Click **Save**. The full sampling settings take about 10 minutes to be effective.
- Database Monitoring Settings
 - You can define a slow SQL threshold. Requests that exceed this threshold are regarded as slow and will be displayed on the URL page and in monitoring reports.
 - i. On the **Custom Settings** page, set **Slow SQL Threshold** in the **Database Monitoring Settings** area.
 - Value range: [100, 20000]. The default value is the same as the SQL threshold configured in the system.

Step 7 Click **Save**. You can click **Clone to Other Component** in each configuration area to clone the configuration to other components. You can also click **Clone to Other Component** in the lower right corner of the page to clone entire configurations to other components in batches.

----End

Configuring an Agent

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and application to filter components.

You can also specify an application and component in the search box to filter the target component.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Component Settings** tab and go to the **Agent Settings** page. Then you can view host information and perform operations such as enabling or disabling Agents. For details, see [Managing Performance Monitoring Agents](#).

----End

Configuring a Monitoring Item

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

----End

Step 1 In the navigation pane, choose **Performance Monitoring > Components**.

Step 2 In the filter bar on the left, select a region and application to filter components.

You can also specify an application and component in the search box to filter the target component.

Step 3 Click a component name in the **Component** column. The component details page is displayed.

Step 4 Click the **Component Settings** tab and go to the **Monitoring Item** page. Then you can view parameters such as **Monitoring Item**, **Monitoring Item Name**, and **Collection Interval**, and perform the following operations:

Table 16-10 Operations on monitoring items

Operation	Description
Enabling or disabling a monitoring item	Click the button in the Enable/Disable Monitoring Item column. A monitoring item is enabled () by default.
Editing a monitoring item	Click Modify in the Operation column to modify the monitoring item configuration. For details, see Configuring a Monitoring Item .
Refreshing monitoring items	Click  in the upper right corner of the monitoring item list to obtain the latest information about all monitoring items in real time.

----End

16.2.2 Viewing the Component Overview

The **Overview** page summarizes the performance monitoring data of component instances and displays **Metric Overview**, **Performance Analysis**, **Resource Usage**, and **Top 5 APIs** of the component in a specified period (max.: 30 days).

- **Metric Overview:** displays the total number of requests, average response time, P99/P95/P50 response time, total number of errors, total number of slow calls, and total number of slow SQL statements of the component server. In addition, the day-on-day and week-on-week performance statistics are also displayed. Day-on-day comparison is to compare the data in the current specified time range with the data in the same range of the previous day. Week-on-week comparison is to compare the data in the current specified time range with the data in the same range of the previous week.
- **Performance Analysis:** displays the average number of requests, average response time, P99/P95/P50 response time, and error rate of the component server in graphs. In addition, the day-on-day data, that is, the data in the same range of the previous day, is also displayed.
- **Resource Usage:** displays the CPU usage and used memory of the component in graphs. In addition, the day-on-day data, that is, the data in the same range of the previous day, is also displayed.
- **Top 5 APIs:** ranked by the total number of requests, error rate, average response time, and P99/P95/P50 response time.

Viewing the Component Overview

Step 1 Log in to the [APM console](#).

- Step 2** Click  on the left and choose **Management & Governance > Application Performance Management**.
- Step 3** In the navigation pane on the left, choose **Performance Monitoring > Components**. All components are displayed.
- Select a region and an application to filter components.
 - **Region:** Select a region to filter components. The current region is selected by default.
 - **Application:** Select the application to which the components belong. You can select up to 50 applications. If you do not select any application, no component data will be displayed.
 - In the component list, select **Period-on-Period** to view data. The period-on-period data varies according to the selected time range. < 1 day: range in current day vs. same range in previous day. 1–7 days: range in current week vs. same range in previous week. 7–15 days: range in current time window vs. same range in previous time window. > 15 days: No comparative data displayed.
 - Set a time range to view the overview data. There are two methods to set a time range:
 - Method 1: Use a predefined time label, such as **Last hour** or **Last 6 hours**. You can select a time range as required.
 - Method 2: Specify the start time and end time (max. 30 days).
 - Click the search box and select an application and component from the drop-down list to view data.
 - Click  in the upper right corner of the component list and select the columns to display. Only the **Component** column is mandatory. You can select other columns as required.
- Step 4** Click a component name in the **Component** column. The component details page is displayed.
- Step 5** Click the **Overview** tab. On the displayed page, view the overview data of the selected component.

You can also perform the following operations if needed:

- Set a time range to view the overview data. There are two methods to set a time range:
 - Method 1: Use a predefined time label, such as **Last hour** or **Last 6 hours**. You can select a time range as required.
 - Method 2: Specify the start time and end time (max. 30 days).
- In the **Metric Overview** area, you can perform the following operations:
 - View the total number of requests, average response time, P99/P95/P50 response time, total number of errors, total number of slow calls, and total number of slow SQL statements of the component server in a specified time range.
 - View the day-on-day and week-on-week performance data. No comparative data is displayed if you select a time range longer than the maximum allowed limit.

- Click the link next to  on the metric card to go to the corresponding tab page.
- In the **Performance Analysis** area, you can perform the following operations:
 - View the average number of requests, average response time, P99/P95/P50 response time, and error rate of the component server in a specified time range.
 - In a metric graph, you can select **Day-on-Day** to view the performance metric data of the previous day.
The solid line indicates the data within the current specified time range, and the dotted line indicates the data in the same range of the previous day.
 - Move the pointer to a time point in the metric graph and click **View Request** to go to the **Tracing** tab page. For details, see [Viewing the Call Relationships Between Microservices Through Tracing](#).
- In the **Resource Usage** area, you can perform the following operations:
 - View the CPU usage and used memory of the component in a specified time range.
 - In a metric graph, you can select **Day-on-Day** to view the performance metric data of the previous day.
The solid line indicates the data within the current specified time range, and the dotted line indicates the data in the same range of the previous day.
 - Move the pointer to a time point in the metric graph and click **View CPU** or **View Memory** to go to the **Info** tab page. For details, see [Viewing the Info Monitoring Item of a Component](#).
- In the **Top 5 APIs** area, you can perform the following operations:
 - View the component's top 5 APIs ranked by the total number of requests, error rate, average response time, and P99/P95/P50 response time in the specified time range.
 - Click an API to go to the **Tracing** tab page to view trace details. For details, see [Viewing the Call Relationships Between Microservices Through Tracing](#).

----End

16.2.3 Viewing the Call Relationships Between Components in the Topology

The topology displays the call relationships between services within a period. The statistics can be collected from the caller or the callee. You can also view the trend.

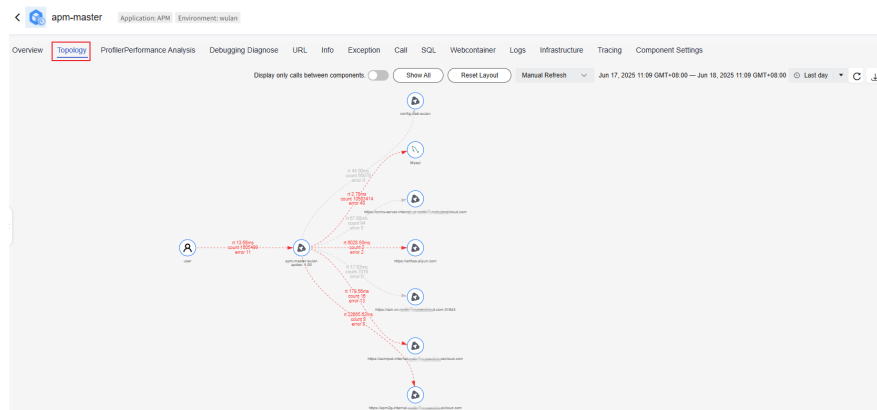
Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

- Step 3** In the navigation pane, choose **Performance Monitoring > Components**.
- Step 4** In the filter bar on the left, select a region and an application.
- Step 5** Click a component name in the **Component** column. The component details page is displayed.
- Step 6** Switch to the **Topology** tab page. The call trend of the selected instance is displayed.

Figure 16-3 Component topology



- Step 7** Click a line in the diagram. The relationships between components or between components and instances are displayed.
- The diagram includes the number of requests, average RT, error rate, and corresponding line graphs.
 - Click the drop-down list in the upper left corner of the line graph and select an instance.
 - Click  in the upper right corner of the line graph. All instance views corresponding to a parameter are displayed.
 - The diagram shows the call relationships between components and enables you to check the traces between components.
 - The function of redirecting to the trace page is supported only by Agent 2.5.3 or later.
 - Click **View Trace**. The **trace** page is displayed, showing the trace information between components.

----End

Other Operations

- Click  next to **Display only calls between components**. The button will then turn blue, showing only the calls between components.
- Click  next to **Display only calls between components**. The button will then turn gray, showing the component call details.
- Click **Show All** to display all call relationships of the selected instance in a specified period. Click **Reset Layout** to restore the initial topology.

- Select the refresh mode. Default: **Manual Refresh**. In addition, **Automatic refresh in 1 minute**, **Automatic refresh in 5 minutes**, and **Automatic refresh in 15 minutes** are supported.
- Select a time range. Default: **Last 20 minutes**.
Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- For more information about the topology, see [Topology](#).

16.2.4 Viewing the URL Monitoring Item of a Component

This type of monitoring item monitors the calls of the current application by external services. It includes URL, Dubbo server, CSE server, CSEProvider cluster, and FunctionGraph monitoring.

This type of monitoring item reflects the actual external status of the entire service. For example, if the average response time of a URL is long, it means that external users take a long time to query data.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **URL** tab.

Step 7 On the displayed **URL** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 8 Perform the following operations if needed:

- Select a time range. Default: **Last 20 minutes**.
Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- Fuzzy search is supported. Enter the URL or method content in the search box and press **Enter**. Matched data will then be displayed.
- Click  in the upper right corner of the list and select the columns to display.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

- View URL monitoring information. For details, see [Viewing URL Monitoring Data](#).

----End

For more information about URLs, see [URL](#).

16.2.5 Viewing the Info Monitoring Item of a Component

This type of monitoring item monitors JVMInfo, JVMMonitor, GC, threads, and JavaMethod.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Info** tab. By default, the **JVMMonitor** information of all instances is displayed.

Step 7 On the displayed **Info** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets. For details about the metrics, see [JVM](#).

You can also perform the following operations if needed:

- Select a time range. Default: **Last 20 minutes**.
Options: **Last 3 minutes, Last 5 minutes, Last 10 minutes, Last 20 minutes, Last 1 hour, Last 3 hours, Last 6 hours, Last day, Today, Yesterday, Last week, Last month, and Custom**.
- Click  in the upper right corner of the memory pool list and select the columns to display.
- Click  in the upper right corner of the memory pool list to export information. A maximum of 100 records can be exported.

----End

For more information about basic monitoring, see [JVM](#).

16.2.6 Viewing the Exception Monitoring Item of a Component

This type of monitoring item monitors application exception logs. Take the monitoring of Java exception logs as an example. Once you use the log system to

print logs, they will be collected by APM. The types of exceptions to be collected vary depending on the collector type.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Exceptions** tab. By default, the exception logs of all instances are displayed. For details about the metrics, see [Table 16-11](#).

Table 16-11 Exception log parameters

Metric Set	Parameter	Description
Exception	Class	Exception class.
	Exception Type	Exception type.
	Log Type	Exception log type.
	Total Exceptions	Number of times that an exception has occurred.
	Message	Message returned when the exception occurred.
	Error Stack	Error stack.
	Error Trace	Information about the error trace.
Log Version	logType	Log type.
	Version	Log version.

Step 7 On the **Exceptions** tab page, select a target instance and then select **Exception** to view the exception monitoring data.

You can also perform the following operations if needed:

- Set a time range to view data. Select a time range. Default: **Last 20 minutes**.
Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.

- Click  in the upper right corner of the list. In the **Table Settings** dialog box, select or deselect columns to show or hide them.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.
- Click the blue text in the **Message** column to view the message time and content.
- Click **Detail** in the **Error Stack** column to view exception details.
- Click **History** in the **Error Stack** column to view the historical error stack list.
- Click **Detail** in the **Error Trace** column to view trace details.
- Click **History** in the **Error Trace** column to view trace history.
- Click the blue text in the **Version** column to view details.

----End

For more information about exceptions, see [Exception](#).

16.2.7 Viewing the External Call Monitoring Item of a Component

This type of monitoring item monitors the calls of external services by the current application. It covers CSEConsumer, ApacheHttpClient, ApacheHttpAsyncClient, DubboConsumer, and HttpClient monitoring.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **External Calls** tab. By default, the HttpClient monitoring information of all instances is displayed.

Step 7 On the displayed **External Calls** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

You can also perform the following operations if needed:

- Set a time range to view data. Select a time range. Default: **Last 20 minutes**. Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- Click  in the upper right corner of the list. In the **Table Settings** dialog box, select or deselect columns to show or hide them.

- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.
- View HttpClient monitoring information. For details, see [Viewing URL Monitoring Data](#).

----End

For more information about external calls, see [Call](#).

16.2.8 Viewing the Database Monitoring Item of a Component

This type of monitoring item monitors database access. The databases that can be monitored include the C3P0, Cassandra, ClickHouse, DBCP, Druid, EsRestClient, GaussDB, Hikari, Jetcd, ObsClient, MySQL, PostgreSQL, Oracle, HBase, and MongoDB. APM collects statement execution details to help you locate performance problems in code.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Databases** tab. By default, the MySQL database information of all instances is displayed.

Step 7 On the displayed **Databases** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

You can also perform the following operations if needed:

- Set a time range to view data. Select a time range. Default: **Last 20 minutes**. Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- Click  in the upper right corner of the list. In the **Table Settings** dialog box, select or deselect columns to show or hide them.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.
- View the MySQL database monitoring information. For details, see [Viewing MySQL Database Monitoring Data](#).

----End

Setting the Maximum Number of Rows for Monitoring Metric Data Collection

When the number of metric data records exceeds the default or configured value, a threshold-crossing alarm will be displayed.

Step 1 Click **Config** in the alarm and change the value of **Maximum number of rows for monitoring metric data collection**.

Step 2 Click **Confirm**.

- Only when the number of metric data records exceeds the default or configured value will a threshold-crossing alarm be displayed.
- If the number of collected metric data records does not exceed the default or configured value, the value of **Maximum number of rows for monitoring metric data collection** cannot be configured.
- For **Maximum number of rows for monitoring metric data collection**, the default value is **500**. You can enter 10 to 2000.
- If the number of collected metric data records exceeds the configured value, the collection of the corresponding metric will be stopped immediately.
- Your settings take five minutes to become effective.

----End

For more information about databases, see [SQL](#).

16.2.9 Viewing the Cache Monitoring Item of a Component

This type of monitoring item monitors caches, including Redis, Jedis, and Lettuce clients.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Cache** tab. By default, the Redis information of all instances is displayed.

Step 7 On the displayed **Cache** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

You can also perform the following operations if needed:

- Set a time range to view data. Select a time range. Default: **Last 20 minutes**. Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.

- Click  in the upper right corner of the list. In the **Table Settings** dialog box, select or deselect columns to show or hide them.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

----End

For more information about the cache, see [Cache](#).

16.2.10 Viewing the Web Container Monitoring Item of a Component

This type of monitoring item monitors web containers, including Tomcat. This section focuses on Tomcat monitoring.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Web Containers** tab. By default, the Tomcat monitoring information of all instances is displayed.

Step 7 On the displayed **Web Containers** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

You can also perform the following operations if needed:

- Set a time range to view data. Select a time range. Default: **Last 20 minutes**. Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- Click  in the upper right corner of the list. In the **Table Settings** dialog box, select or deselect columns to show or hide them.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.
- Click a number in blue (such as those in the **Current Threads**, **Busy Threads**, or **Peak Busy Threads** column) to view the trend graph of the target web container in the specified period.
- Click a version in the **Version** column to view details.

----End

For more information about the web containers, see [Web Container](#).

16.2.11 Viewing the Message Queue Monitoring Item of a Component

This type of monitoring item monitors message queues, including KafkaProducer. This section focuses on KafkaProducer monitoring.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Message Queues** tab. By default, the KafkaProducer monitoring information of all instances is displayed.

Step 7 On the displayed **Message Queues** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

You can also perform the following operations if needed:

- Set a time range to view data. Select a time range. Default: **Last 20 minutes**. Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- Click  in the upper right corner of the list. In the **Table Settings** dialog box, select or deselect columns to show or hide them.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

----End

16.2.12 Viewing the Log Monitoring Item of a Component

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Logs** tab. For details, see [Searching for Logs](#).

----End

16.2.13 Viewing the Infrastructure of a Component

When the calls between enterprise microservices are complex, Agents sample some requests, and intercept corresponding requests and subsequent call information. This section describes how to view the infrastructure of a component.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Infrastructure** tab. The infrastructure metrics of the component are displayed.

Step 7 In the upper left corner of the page, select an instance from the instance drop-down list to view its monitoring data.

You can also perform the following operations if needed:

- Set a time range to view data. You can use a predefined time label, such as **Last hour** and **Last 6 hours**, or set a custom time range. Max.: 30 days.
- Move the pointer over a time point in the graph. The metric values at this time point will then be displayed.
- Click  to refresh the data. There are two refresh modes: manual and auto.
 - Manual refresh: Click **Refresh Now** to refresh dashboard data.
 - Auto refresh: Click an auto refresh mode to automatically refresh dashboard data.

----End

16.2.14 Viewing the Traces of a Component

When the calls between enterprise microservices are complex, Agents sample some requests, and intercept corresponding requests and subsequent call information. This section describes how to view the traces of a component.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Tracing** tab. The trace details of the component are displayed.

Step 7 Filter traces by instance, URL, or other criteria. You can also add custom criteria in the lower part of the page to filter traces. For details, see [Viewing the Call Relationships Between Microservices Through Tracing](#).

----End

For more information about traces, see [Tracing](#).

16.2.15 Viewing the Communication Protocol Monitoring Item of a Component

This type of monitoring item monitors communication protocols, including the WebSocket protocol. This section focuses on WebSocket monitoring.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **Communication Protocols** tab. By default, the WebSocket monitoring information of all instances is displayed.

Step 7 On the displayed **Communication Protocols** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

You can also perform the following operations if needed:

- Click a number in blue (such as those in the **Calls**, **Avg RT (ms)**, or **Errors** column). The trend in the last 20 minutes is displayed by default.
- Click a URL in the **URL** column to view the call details of the URL.

- On the **Overview** tab page, move the pointer over a time point. The number of errors, number of closed connections, traffic, average response time (ms), number of calls, and number of created connections in this time point will then be displayed.
- Set a time range to view data. Select a time range. Default: **Last 20 minutes**. Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- Click  in the upper right corner of the list. In the **Table Settings** dialog box, select or deselect columns to show or hide them.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

----End

16.2.16 Viewing the RPC Monitoring Item of a Component

This type of monitoring item monitors remote procedure calls, including GRPCServer.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Components**.

Step 4 In the filter bar on the left, select a region and an application.

Step 5 Click a component name in the **Component** column. The component details page is displayed.

Step 6 Click the **RPCs** tab. By default, the GRPCServer monitoring information of all instances is displayed.

Step 7 On the displayed **RPCs** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

You can also perform the following operations if needed:

- Set a time range to view data. Select a time range. Default: **Last 20 minutes**. Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- Click  in the upper right corner of the list. In the **Table Settings** dialog box, select or deselect columns to show or hide them.
- Click  in the upper right corner of the list to export information. A maximum of 100 records can be exported.

----End

16.3 Viewing the Call Relationships Between Microservices Through Tracing

When the calls between enterprise microservices are complex, APM Agents sample some requests, and intercept corresponding requests and subsequent call information.

Viewing Basic Information About Traces

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Tracing**.

Step 4 Select a time range in the upper right corner of the page. Default: **Last 20 minutes**.

Options: **Last 3 minutes, Last 5 minutes, Last 10 minutes, Last 20 minutes, Last 1 hour, Last 3 hours, Last 6 hours, Last day, Today, Yesterday, and Custom**.

Step 5 Specify the following search criteria or add custom criteria to query traces.

Table 16-12 Search criteria of traces

Search Criterion	Description	Mandatory
Application	Application to which the trace belongs.	Yes
Region	Region where the trace is located.	Yes
Component	Component to which the trace belongs.	No
Environment	Environment to which the trace belongs.	No
Instance	Instance to which the trace belongs.	No
URL	Trace URL, which can be a REST URL or real URL. A REST URL contains a variable name, for example, <code>/apm/get/{id}</code> . A real URL indicates an actual URL.	No
Exact Search	Whether to perform exact match on URLs. If this option is selected, exact match is performed. If this option is not selected, fuzzy match is performed.	No
Call Method	HTTP method of the trace.	No
Status Code	HTTP status code returned by the trace.	No

Search Criterion	Description	Mandatory
Response Time	Response time range of the trace. You can specify the minimum and maximum response time to search for the trace or leave them empty.	No
Exception or Not	Whether to filter the traces that are regarded as exceptions.	No
Trace ID	If you specify this parameter, other search criteria become invalid and the search will be performed based on the trace ID you specify.	No
Custom Parameter	If you have configured Key for Header Value Interception , Key for Parameter Value Interception , and Key for Cookie Value Interception for URL monitoring, you can set key=value to search, for example, httpMethod=POST . For details about how to configure the URL monitoring item, see Configuring the URL Monitoring Item .	No
Application Code	If you have configured Service Code Length , Key for Service Code Interception , and Normal Service Code , corresponding application codes will be collected. You can search information based on application codes. Generally, the value of Application Code is the same as the value of Normal Service Code . For details about how to configure the URL monitoring item, see Configuring the URL Monitoring Item .	No

Step 6 Click **Search Trace** to filter the traces that meet search criteria.

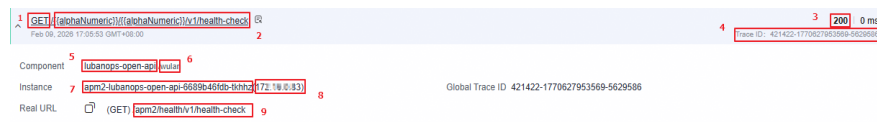
- Only the traces of successful or failed requests can be displayed. The following uses **Successful Request** as an example.
 - Click the green button next to **Successful Request**, only the traces of successful requests are displayed. The red button next to **Failed Request** becomes grayed.
 - If you click the green button again, both the traces of successful and failed requests are displayed. The red button is no longer grayed.
 - The green and red buttons cannot be grayed at the same time.
- Response Time**: response time of a trace. Click **Response Time**. Traces are sorted by response time in descending order, and **Response Time**  is displayed. Click **Response Time** again. Traces are sorted by response time in ascending order, and **Response Time**  is displayed. Click **Response Time** for the third

time. The trace list is restored to the initial state and **Response Time** is displayed.

- **Generated**: start time of a trace. Traces can be sorted by generation time. For details, see **Response Time**.
- You can click **Clear** to clear the search criteria and results.

Step 7 In the displayed trace list, click  next to the target trace to view its basic information.

Figure 16-4 Basic information about a trace

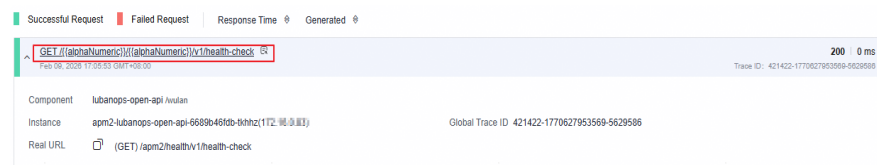


Parameter description:

1. HTTP method of the trace.
2. REST URL of the trace. A REST URL contains a variable name, for example, `/apm/get/{id}`. You can click the URL to go to the trace details page.
3. HTTP status code returned by the trace.
4. Trace ID.
5. Component to which the trace belongs.
6. Environment to which the trace belongs.
7. Host of the instance to which the trace belongs.
8. IP address of the instance to which the trace belongs.
9. Actual URL of the trace.

Step 8 Hover the pointer over a trace name. The trace name turns blue. Click the trace name to view the trace details.

Figure 16-5 Trace name

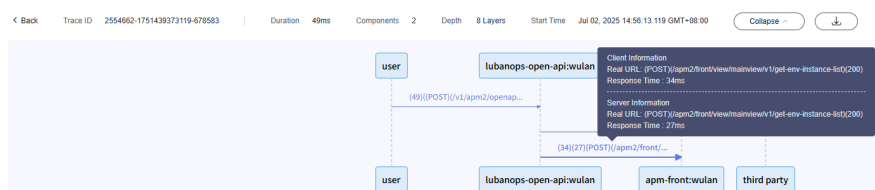


On the trace details page, you can view the trace's complete information, including the local method stack and remote call relationships.

Figure 16-6 Call relationship

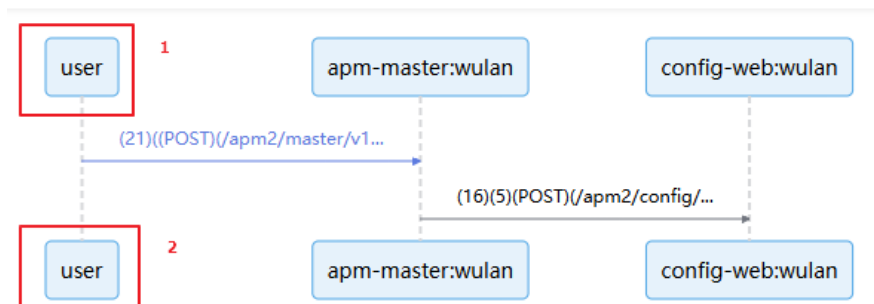
Parameter description:

1. Component and environment to which the called API belongs
2. Response time (unit: ms) of the client. You can hover the pointer over the line between components to view the client and server information.
 - Real URL: REST URL of the trace. A REST URL contains a variable name, for example, `/apm/get/{id}`.
 - Response time: response time of the trace.



3. Response time (unit: ms) of the server.
4. Key parameter of the method in the trace method stack. For example, for a Tomcat entry method, a real URL is displayed. For a MySQL call method, an executed SQL statement is displayed.
5. Number of components that are involved.
6. Traces are displayed in a tree. **Depth** indicates the number of layers in the tree.
7. Click **Collapse** if required.

The content displayed in positions 1 and 2 in the sequence diagram of a trace varies depending on the Agent type or connected website/app.

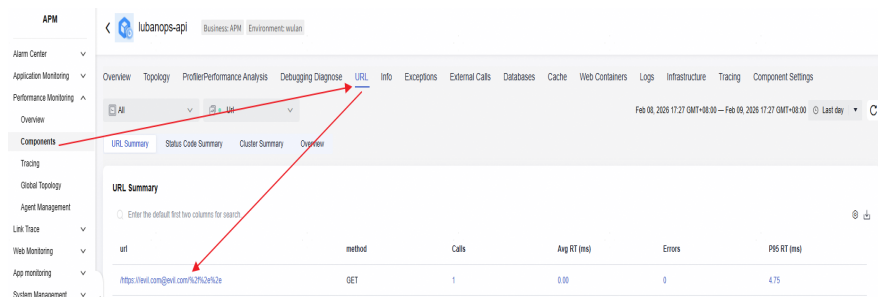
Figure 16-7 Sequence diagram

Agent Type or Connected Website/App	Display Content
Web	WEB
WeChat	WE_CHAT
Baidu	BAIDU
Alipay	ALIPAY
DingTalk	DING_TALK
Mobile app (Android)	ANDROID-APP
Mobile app (iOS)	IOS-APP
Mobile app (HarmonyOS)	HARMONY-APP
Others	user

8. Click  to download the sequence diagram.

You can also click a specific URL on the monitoring item view page, for example, the table view of the URL monitoring item. In this way, you can quickly search for required trace information based on preset search criteria.

Figure 16-8 Redirecting from the URL page to the trace page



9. Duration, that is, the total time of a trace request.

Step 9 Click a resource in the resource list. The trace information, exceptions, logs, infrastructure, and Profiler analysis of the resource will then be displayed.

- The **Trace Info** tab page displays the instance, class, and status code of the resource.
- The **Exceptions** tab page displays the exception information of the resource. If the resource has no exception, the **Exceptions** tab page will not be displayed.
- For details about the **Logs** tab page, see [Viewing the Log Monitoring Item of a Component](#).
- For details about the **Infrastructure** tab page, see [Viewing the Infrastructure of a Component](#).

- If the Agent version is 2.4.9-profiler or later, you can view the Profiler performance analysis data on the trace page.

----End

16.4 Viewing the Call Relationships Between Applications in the Global Topology

Each line in the topology indicates the call relationship between services within a period. The statistics can be collected from the calling or called party. You can click a line to view the call trend on the right. The topology can also display the call relationships between middleware. On the topology, you can view the call relationships between services and check whether the calls between services are normal to quickly locate faults.

You can view the global topology of some or all components of the current application, and the global topology of components of one or more applications.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane on the left, choose **Performance Monitoring > Global Topology**. On the displayed page, view the call and dependency relationships of all components of the application.

Step 4 View the call and dependency relationships of all components of one or more applications.

You can also perform the following operations if needed:

- Select one or more applications from the **Application** drop-down list.
- Click the button next to **Third-party invocation**. If the button is , only the calls between components are displayed. If the button is , the calls of external components are displayed.
- Toggle on **Application Topology**. The call relationships of the selected application will then be displayed.
- Set a time range to view data. Click the time dimension selection button and set a time range. Default: **Last 20 minutes**. Options: **Last 3 minutes**, **Last 5 minutes**, **Last 10 minutes**, **Last 20 minutes**, **Last 1 hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, and **Custom**.
- **Alarm Latency**: can be 0.5s, 1s, 3s, 5s, or 10s.
- **Error Latency**: can be 1s, 3s, 5s, 10s, or 15s.
- Right-click an application and choose **View Topology Details** from the shortcut menu. The component topology of the application is displayed.
- Click a line in the diagram. The detailed data such as the number of requests, average RT, and error rate, and line graphs are displayed on the right.

- Click the drop-down list in the upper left corner of the line graph and select an instance.
 - Click  in the upper right corner of the line graph. All instance views corresponding to a parameter are displayed.
 - The diagram shows the call relationships between components and enables you to check the traces between components.
 - The function of redirecting to the trace page is supported only by Agent 2.5.3 or later.
 - Click  to obtain the latest information in the statistical period.
 - Click  to download the topology.
- End

16.5 Managing Performance Monitoring Agents

APM Agents use bytecode enhancement technology to collect application performance data in real time. They run on the server where applications are deployed.

Procedure

Step 1 Log in to the [APM console](#).

Step 2 Click  on the left and choose **Management & Governance > Application Performance Management**.

Step 3 In the navigation pane, choose **Performance Monitoring > Agent Management**.

Step 4 View the Agent list.

1. Filter Agents by application, component, environment, IP address, or host.
2. Click  in the upper right corner of the list and select the columns to display.

----End

The following table describes the Agent statuses.

Status	Description
Normal	The Agent is running properly.
Offline	The Agent is abnormal due to a network error. Check and restore the network.
Disabled	The Agent is manually or globally disabled. Contact technical support.

Other Operations

Step 1 In the navigation pane, choose **Performance Monitoring > Agent Management**.

Step 2 Select target objects and click **Operation**.

Step 3 In the drop-down list, select **Disable Agent**, **Enable Agent**, or **Delete Agent**.

Step 4 In the dialog box that is displayed, click **Yes** to disable, enable, or delete the Agents for the selected hosts.

----**End**

17 SDK Reference

17.1 Privacy and Sensitive Information Protection Statement

17.1.1 Privacy Statement of APM Data Collection SDKs

APM data collection SDKs are open capabilities and services provided by Huawei Cloud Computing Technologies Co., Ltd. (hereinafter referred to as "Huawei") for application developers (hereinafter referred to as "developers") to report data to the Huawei Cloud APM backend using code.

After developers integrate APM data collection SDKs into their apps according to the development documentation and guide, Huawei will provide related services for end users (hereinafter referred to as "you" or "users") through the integrated SDKs, and process the developers' application data (which may contain your personal information). Huawei takes your personal information and privacy seriously, and is committed to taking appropriate security measures to protect your personal information in accordance with relevant laws and established industry security standards. This Statement will describe how Huawei collects, uses, discloses, protects, stores, and transmits your personal information.

Note: Huawei requires all developers who use APM data collection SDKs to strictly comply with applicable laws and regulations, developer agreements, and *Developer Compliance Guide for APM Data Collection SDKs*. Before accessing and using open capabilities, developers are required to inform you of the situation about personal information processing using the SDK in their privacy policies, and obtain your consent or other legal bases. However, Huawei is unable to control how developers and their apps process your personal information, and will not be liable for their behaviors. Huawei recommends that you carefully read user agreements and privacy policies related to developers' apps and use them only after you fully understand and agree to how the developers will process your personal information.

How Does Huawei Collect and Use Your Personal Information

Huawei will only collect and use your personal information in accordance with the purposes and methods described in this Statement. If Huawei wants to use your personal information for purposes not stated herein, Huawei will explicitly inform you about this by themselves or through developers, and obtain your consent again or other legal bases. If the SDK has extended functions or collects and uses optional personal information, Huawei will specify it in the document.

- **APM data collection SDKs:**

You can call the SDKs to report app logs to APM. The SDKs can collect information about the device model/name/disk/memory/brand/CPU, OS name/version, app version/name/process/thread, Wi-Fi status, and IP address.

Required Permissions

When you use relevant functions and services, Huawei will ask you for necessary device permissions through developer apps. You can check the permissions on your device settings or privacy settings page, and enable or disable some or all permissions as you want.

When integrating and using Huawei's open capabilities, developers will determine the required permissions by themselves. Therefore, developers should notify you of the requested permissions and purposes. By granting any permission as requested by a developer's application, you are authorizing Huawei to collect and use relevant personal information to provide you with corresponding services. Once you cancel the authorization, Huawei will no longer process relevant personal information based on the corresponding permission. The services that require the permission may not be provided for you. However, your decision to cancel the authorization does not affect previous personal information processing that you authorized.

Permission	Description	Purpose
Obtaining the network status	Obtains the network status.	To check whether the current network connection is valid.
Obtaining the Wi-Fi status	Obtains the Wi-Fi status.	To obtain the Wi-Fi access status.

Protection of Minors

Huawei attaches great importance to the protection of minors' personal information. Huawei will provide services and protection for minors in strict accordance with national laws and regulations. If you are a minor, your parents or other guardians must agree to your use of a developer application, and agree to the applicable terms and conditions. Parents or other guardians should also take appropriate precautions to protect minors, including monitoring their use of the developer application.

In particular, if you are a minor under the age of 14, please carefully read the privacy statements of developers' apps and this Statement with your parents or

guardians, and obtain content their consent or guidance before you use the developers' apps and Huawei services or provide information for Huawei. If you are a parent or guardian, please ensure that the child under your guardianship uses the developers' apps and Huawei services and provides information for Huawei with your consent or under your guidance.

Managing Your Personal Information

Huawei prioritizes your concern for personal information. Huawei will comply with applicable laws and regulations to ensure that you can exercise personal data subject rights, such as accessing, copying, modifying, and deleting personal information.

As you are using APM SDKs and their services through a developer application, if you need to access, copy, or modify personal information related to the SDKs and their services, you should exercise your rights as a personal data subject through the channel provided by the developer application.

To protect your rights of accessing, copying, modifying, and deleting your personal information, Huawei explicitly requires developers in agreements to provide convenient methods for you to exercise your rights based on applicable laws and regulations. In addition, Huawei open capabilities also provide related APIs for developers to execute your requests of accessing, copying, modifying, and deleting personal information. You can contact Huawei by referring to [How to Contact Huawei](#) and Huawei will try the best to coordinate, support, and promote your exercise of the preceding rights.

If you directly request personal data subject rights from Huawei, to ensure your data security and other legitimate rights, Huawei may verify your identity and ask you to provide the required personal information. Huawei may also disclose collected information to developers to verify your identity. After verifying your identity, Huawei will respond to your request promptly, in accordance with applicable laws and regulations.

If you have any further requests, questions, comments, or suggestions regarding your rights as a data subject, please contact Huawei by referring to [How to Contact Huawei](#).

Information Storage Location and Period

1. Storage Location

The above information will be stored on the server in the current region.

2. Storage Period

Huawei will retain your personal information for the period necessary to fulfill the purposes described in this Statement, and delete or anonymize your personal information beyond the time set out below, unless otherwise specified in laws and regulations.

For the storage duration and processing upon expiration of personal information collected by each APM function, see [Data Collection](#).

How to Contact Huawei

Address: Huawei Cloud Data Center, Jiaoxinggong Road, Qianzhong Avenue, Gui'an New District, Guizhou Province

If you are not satisfied with the response you receive, especially if you feel that Huawei's processing of your personal information has violated or infringed on your legal rights and/or interests, you may seek external resolution, whether by filing a lawsuit with the people's court in the relevant jurisdiction, submitting a complaint to the industry self-regulation association or relevant government regulatory agencies, or by other means. You may also consult Huawei regarding the available channels for voicing your complaint.

Huawei Cloud's collection and use of your personal information is governed by [Privacy Statement](#).

17.1.2 Developer Compliance Guide for APM Data Collection SDKs

After the *Personal Information Protection Law of the People's Republic of China* came into force on November 1, 2021, regulators, industry stakeholders, and end consumers are increasingly concerned about user privacy protection. To crack down on the unauthorized collection and use of personal information by apps and SDKs, regulators have released related standards and specifications.

When providing services for end users, developers (hereinafter referred to as "you") must comply with applicable laws and regulations (including but are not limited to the *Personal Information Protection Law of the People's Republic of China*, *Cybersecurity Law of the People's Republic of China*, and *Data Security Law of the People's Republic of China*) as well as relevant standards and specifications, fulfill personal information protection obligations, and process users' personal information in compliance with the principles of lawfulness, fairness, necessity, and integrity.

This document helps you better understand APM data collection SDKs and use them in compliance with applicable laws and regulations. It applies only to developers in Chinese mainland.

Basic Requirements

Your products and services shall respect user privacy and comply with data protection laws and regulations in countries or regions where your products and services are sold. Do not participate in any activities that interfere with, damage, or access any terminal device, server, or network without authorization.

• (1) Privacy Policy Requirements

You need to release a privacy policy in your own name according to laws, and obtain user consent or other legal bases for personal information processing. The requirements of the privacy policy include but are not limited to:

- a. There is a document independent from the user agreement.
- b. Before apps run for the first time to collect and process personal information, the apps need to clearly prompt users to read the privacy policy. The privacy policy must be easy to view. For example, a user can read the privacy policy with fewer than four touches or slides on the main function screen of an app.
- c. The description must be clear and plain, comply with the general language habits, and avoid ambiguity.

- d. The privacy policy must include the purposes, methods, and scope of personal information collection for products and services, and the names and contact information of the personal information processors.
- e. If your products and services need to share personal information with third parties or integrate third-party SDKs, disclose and explain the situation to users in the privacy policy to obtain users' authorization or consent.
- **(2) Requirements for processing personal information**

When processing users' personal information, your products and services must comply with the following requirements:

 - a. Process personal information necessary for stated purposes and in compliance with the data minimization principle.
 - b. The scope and purposes of personal information collected and processed must be the same as those specified in the privacy policy.
 - c. The frequency of personal information collection must be the same as that specified in the privacy policy. Personal information collection beyond the stated frequency is not allowed.
 - d. There is a clear mechanism for deleting personal information upon expiration. The retention period of personal information is the same as that in the privacy policy. Personal information is deleted or anonymized upon expiration.
 - e. Before processing personal information about minors under the age of 14, consent shall be obtained from the minors' parents or other guardians.
 - f. If personal information is processed for personalized recommendation or big data analysis, end users shall be notified and authorization from end users shall be obtained before related service functions are implemented.
 - g. If sensitive personal information needs to be processed, obtain the consent from end users.
 - h. If cross-border transfer of personal information is required, security assessment must be conducted according to the methods and standards formulated by the Cyberspace Administration of China and relevant departments under the State Council, and the transfer must comply with relevant requirements. In addition, the consent from end users shall be obtained.
 - i. Users can smoothly exercise rights as a personal data subject, such as checking, copying, modifying, and deleting personal information.

Personal Information Entrusted for Processing

Before connecting to or using an APM data collection SDK, you are required to inform users of the SDK name, SDK provider name, categories of personal information to be collected, purposes for using the information, and privacy policy link, and obtain the users' consent or other legal bases. You can provide the terms in the following ways:

- In text

Third-party SDK: APM data collection SDK (iOS/Android/HarmonyOS/web/WeChat/Baidu/DingTalk/Alipay/quick apps)

Third-party company name: Huawei Cloud Computing Technologies Co., Ltd.

Personal information to be collected: device model/name/disk/memory/brand/CPU, OS name/version, app version/name/process/thread, and Wi-Fi status

Purpose: To collect app performance data and report the data to APM.

Privacy policy link: [Privacy Statement of APM Data Collection SDKs](#)

In a table

Third-Party SDK	Third-Party Company	Personal Data to Be Collected	Purpose	Link to Privacy Statement
APM data collection SDK (iOS/Android/HarmonyOS/web/WeChat/Baidu/DingTalk/Alipay/quick apps)	Huawei Cloud Computing Technologies Co., Ltd.	Device model/name/disk/memory/brand/CPU, OS name/version, app version/name/process/thread, and Wi-Fi status	To collect app performance data and report the data to APM.	Privacy Statement of APM Data Collection SDKs

Permission Requirements

When providing services, the SDK complies with the principle of least permission. You need to apply for system permissions as required and obtain users' consent.

Permission	Description	Purpose
Obtaining the network status	Obtains the network status.	To check whether the current network connection is valid.
Obtaining the Wi-Fi status	Obtains the Wi-Fi status.	To obtain the Wi-Fi access status.

Requirements of Delayed Initialization

To prevent apps from processing users' personal information before obtaining users' consent, Huawei provides the **APMSDK.start()** API for SDK initialization. Your app can call this API to initialize the SDK only when users agree that.

Requirements of Minimum Functions

Huawei SDK provides configuration capabilities for extended functions and optional personal information processing. You can enable or disable related functions based on your service requirements by compiling files.

Protecting Personal Information Subjects' Rights

To ensure that users can easily access, copy, modify, and delete their personal information, Huawei provides related APIs in the SDK. You can then call these APIs to execute users' requests for accessing, copying, modifying, and deleting personal information.

17.2 iOS SDK Access

The APM iOS SDK can collect and report app startup performance, crashes, ANRs, errors, network requests, devices, and custom statistics.

Prerequisites

- Before using the SDK, you need to register a Huawei Cloud Account and create an iOS app on the **App Monitoring** page.

Transport Protocol

HTTPS

Version Updates

Table 17-1 Version updates

Version	SDK Download Address	Verification File Download Address	Update Description	System
2.1.0	Link	Link	1. Compatible with the APM Flutter SDK.	iOS 10 and Xcode 11 and later
2.0.12	Link	Link	1. Resolved network request monitoring issues.	iOS 10 and Xcode 11 and later
2.0.11	Link	Link	1. Resolved class name conflicts.	iOS 10 and Xcode 11 and later
2.0.10	Link	Link	1. Resolved the issue of abnormal app loading duration during cold start.	iOS 10 and Xcode 11 and later
2.0.9	Link	Link	1. Resolved the issue related to custom event reporting.	iOS 10 and Xcode 11 and later

Version	SDK Download Address	Verification File Download Address	Update Description	System
2.0.8	Link	Link	1. Supported sampling rate configuration.	iOS 10 and Xcode 11 and later
2.0.7	Link	Link	1. Supported remote configuration.	iOS 10 and Xcode 11 and later
2.0.6	Link	Link	1. Resolved the SDK startup failure.	iOS 10 and Xcode 11 and later
2.0.5	Link	Link	1. Resolved the issues related to the compatibility between network requests and error monitoring systems.	iOS 10 and Xcode 11 and later
2.0.1	Link	Link	1. Resolved the related to system compatibility.	iOS 10 and Xcode 11 and later
2.0.0	Link	Link	1. Supported collection and reporting of app startup performance, crashes, ANRs, errors, network requests, devices, and custom statistics.	iOS 10 and Xcode 11 and later

Procedure

Step 1 Integrate the access SDK.

Method 1: Using CocoaPods

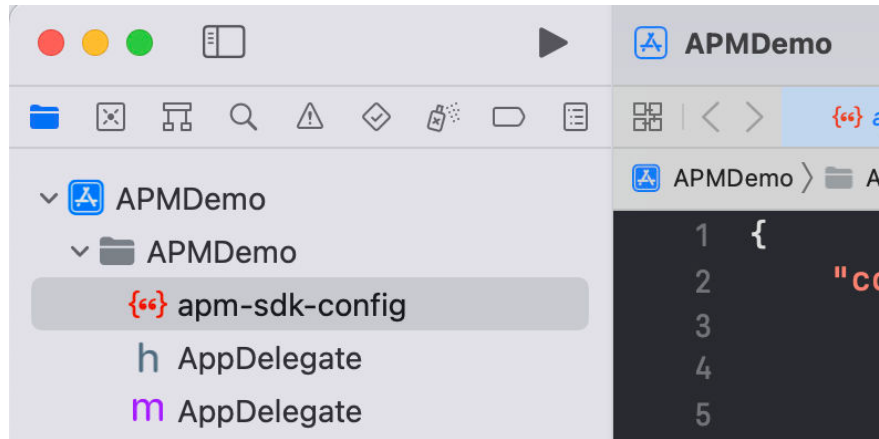
- Add the official CocoaPods repository to the Podfile.
source 'https://github.com/CocoaPods/Specs.git'
- Add dependencies to the Podfile.
pod 'APMSDK', '2.1.0'
- Run the following command on the device:
pod install --repo-update

Method 2: Manual integration

- Decompress the downloaded SDK to a specified directory. Note: Decompress the package only.
- Add the decompressed **xcframework** static library to your project. Ensure that the **xcframework** static library to be imported is in the same disk space as the workspace. If not, select **Copy items if needed** and **Create groups** to copy the **xcframework** library file to the workspace.

Step 2 Add a configuration file.

During the creation of the iOS app, a configuration file named **apm-sdk-config.json** is generated. Download and drag the file to the **root** directory of the Xcode project, as shown in the following figure.

**Table 17-2** Configuration parameters

Parameter	Mandatory	Default Value	Description
appld	Yes	-	Mobile app ID.
authorization	Yes	-	Used for app authentication.
region	Yes	-	Region where APM is deployed. Currently, cn-north-4 and ap-southeast-3 are supported.
UID	No	-	Custom ID.
tag	No	-	Custom tag. Use commas (,) to separate multiple tags.
url	No	-	Public domain name for reporting data to APM.
networkWhiteList	No	-	Network monitoring whitelist.
cacheThreshold	No	200	When the number of records stored in the local database reaches this threshold, an alarm is reported. Value range: 30 to 1,000.
timeInterval	No	60s	Interval for triggering reporting. Value range: 60 to 1,800, in seconds.
reportBackground	No	true	Whether to report logs upon app backend switching.

Parameter	Mandatory	Default Value	Description
reportLaunch	No	false	Whether to report logs upon app startups. To enable this function, start the SDK before sending UIApplicationDidFinishLaunchingNotification .
enableNetwork	No	false	Whether to enable network collection.
enableCrash	No	false	Whether to enable crash collection.
enableLaunch	No	false	Whether to enable startup collection.
enableANR	No	false	Whether to enable freeze collection.
enableError	No	false	Whether to enable error collection.
enableDevice	No	false	Whether to enable device collection.
enableEvent	No	false	Whether to enable custom statistics reporting.
traceType	No	apm	Whether to enable open tracing. If this function is enabled, OpenTelemetry will be used. If this function is disabled, APM tracing will be used.
logLevel	No	off	Debug log level. Options: debug , info , warn , error , and off .
samplingRate	No	1.0	Sampling rate. Range: 0–1.0.

Step 3 Start the SDK.

- Objective-C code example

```
@import APMSDK;

// Start the SDK based on the apm-sdk-config.json configuration file.
[APMSDK start];

// Set a custom ID. You can also add an ID to the configuration file.
[APMSDK setUid:@"Custom ID"];

// Set a custom tag. You can also add a tag to the configuration file.
[APMSDK addTag:@"Custom tag 1,Custom tag 2,Custom tag 3"];
```

- Swift code example

```
import APMSDK

// Start the SDK based on the apm-sdk-config.json configuration file.
APMSDK.start()
```

```
// Set a custom ID. You can also add an ID to the configuration file.  
APMSDK.setUid ("Custom ID")  
  
// Set a custom tag. You can also add a tag to the configuration file.  
APMSDK.addTag ("Custom tag 1, Custom tag 2, Custom tag 3")
```

Step 4 Configure custom statistics reporting.

Method	Description
+ (BOOL)event:(NSString *)key value:(id)value;	key: event name. Max.: 2,048 characters. value: event content. Valid JSON objects such as NSString , NSArray , NSDictionary , and NSNumber are supported. Max.: 30,720 characters.

Objective-C code example

```
// Event statistics  
[APMSDK event:@"User Information" value:@{@"name":@"XXX"}];  
[APMSDK event:@"Error Description" value:@"The request timed out."];  
[APMSDK event:@"Call Stack" value:@{@"0x0000000101ee9c6c", @"0x0000600000e61d80"}];  
  
// Numeric statistics  
[APMSDK event:@"Custom Load Time" value:@(0.238)];  
[APMSDK event:@"Purchase Statistics" value:@(1)];
```

Swift code example

```
// Event statistics  
APMSDK.event("User Information", value: ["name": "XXX"])  
APMSDK.event("Error Description", value: "The request timed out.")  
APMSDK.event("Call Stack", value: ["0x0000000101ee9c6c", "0x0000600000e61d80"])  
  
// Numeric statistics  
APMSDK.event("Custom Load Time", value: 0.238)  
APMSDK.event("Purchase Statistics", value: 1)
```

Step 5 Configure identity authentication for network requests (applicable only when network request monitoring is enabled): When a request is sent to protected resources on the server, the process of challenging a user to prove their identity is triggered. The app needs to implement the **APMURLProtocolDelegate** protocol to complete this process.

Objective-C code example

```
@interface AppDelegate : UIResponder <UIApplicationDelegate, APMURLProtocolDelegate>  
  
@end  
  
@implementation AppDelegate  
  
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {  
    //Set a proxy for identity authentication challenge.  
    [APMURLProtocol setDelegate:self];  
    // Start the SDK.  
    [APMSDK start];  
}  
  
#pragma mark -- APMURLProtocolDelegate  
  
- (BOOL)APMURLProtocol:(APMURLProtocol *)protocol canAuthenticateAgainstProtectionSpace:
```

```
(NSURLProtectionSpace *)protectionSpace {
    // Authenticate the server identity: The following shows an example. Perform operations based on the
    original authentication mode of the app.
    return [[protectionSpace authenticationMethod] isEqual:NSURLAuthenticationMethodServerTrust];
}

- (void)APMURLProtocol:(APMURLProtocol *)protocol didReceiveAuthenticationChallenge:
(NSURLAuthenticationChallenge *)challenge {
    //Trust the server identity: The following shows an example. Perform operations based on the original
    authentication mode of the app.
    NSURLCredential *credential = [NSURLCredential
    credentialForTrust:challenge.protectionSpace.serverTrust];
    [protocol resolveAuthenticationChallenge:challenge withCredential:credential];
}

- (void)APMURLProtocol:(APMURLProtocol *)protocol didCancelAuthenticationChallenge:
(NSURLAuthenticationChallenge *)challenge {
}

@end
```

Swift code example:

```
class AppDelegate: UIResponder, UIApplicationDelegate, APMURLProtocolDelegate {

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        //Set a proxy for identity authentication challenge.
        APMURLProtocol.setDelegate(self)
        // Start the SDK.
        APMSDK.start()
    }

    func apmURLProtocol(_ `protocol`: APMURLProtocol, canAuthenticateAgainstProtectionSpace
protectionSpace: URLProtectionSpace) -> Bool {
        // Authenticate the server identity: The following shows an example. Perform operations based on the
        original authentication mode of the app.
        return protectionSpace.authenticationMethod == NSURLAuthenticationMethodServerTrust
    }

    func apmURLProtocol(_ `protocol`: APMURLProtocol, didReceive challenge: URLAuthenticationChallenge)
{
        //Trust the server identity: The following shows an example. Perform operations based on the original
        authentication mode of the app.
        if let serverTrust = challenge.protectionSpace.serverTrust {
            let credential = URLCredential(trust: serverTrust)
            `protocol`.resolve(challenge, with: credential)
        }
        else {
            `protocol`.resolve(challenge, with: nil)
        }
    }

    func apmURLProtocol(_ `protocol`: APMURLProtocol, didCancel challenge: URLAuthenticationChallenge) {
    }
}
```

----End

Data Reporting Policy

The SDK supports the following reporting policies: threshold-based reporting, scheduled reporting, reporting when the app is switched to the backend, and reporting upon startup. Set parameters such as **cacheThreshold**, **timeInterval**, **reportBackground**, and **reportLaunch** based on the scenario.

Enabling the Debug Log Function

Enable the debug log function if needed. Then you can monitor real-time SDK running through the console, check logs, and make adjustments. The log level can be **debug**, **info**, **warn**, **error**, or **off** (disabled).

Disable the debug log function before bringing an app online. That is, set the log level to **off**.

Example: Set the log level in the **apm-sdk-config.json** configuration file.

```
{
  "logLevel": "debug"
}
```

17.3 Android SDK Access

The APM Android SDK can collect and report app startup performance, crashes, ANRs, errors, network requests, devices, and custom statistics.

Prerequisites

- Before using the SDK, you need to register a Huawei Cloud Account and create an Android app on the **App Monitoring** page.

Transport Protocol

HTTPS

Version Updates

- For details about how the SDK processes personal information, see [Privacy Statement of APM Data Collection SDKs](#).
- You must meet personal information protection requirements when integrating and using the SDK. For details, see [Developer Compliance Guide for APM Data Collection SDKs](#).

Table 17-3 Version updates

Version	SDK Download Address	Verification File Download Address	Update Description	System
2.1.0	SDK link Plug-in link	SDK link Plug-in link	1. Compatible with the APM Flutter SDK.	Android 5 or later
2.0.24	SDK link Plug-in link	SDK link Plug-in link	1. Resolved open tracing issues.	Android 5 or later

Version	SDK Download Address	Verification File Download Address	Update Description	System
2.0.23	SDK link Plug-in link	SDK link Plug-in link	1. Resolved HttpURLConnection network monitoring issues.	Android 5 or later
2.0.22	SDK link Plug-in link	SDK link Plug-in link	1. Resolved the issue of abnormal startup duration.	Android 5 or later
2.0.21	SDK link Plug-in link	SDK link Plug-in link	1. Resolved the issue of instrumentation failure in specific environments.	Android 5 or later
2.0.20	SDK link Plug-in link	SDK link Plug-in link	1. Fixed the issue that the OkHttp2 custom interceptor does not take effect.	Android 5 or later
2.0.19	SDK link Plug-in link	SDK link Plug-in link	1. Resolved OkHttp3 network monitoring issues.	Android 5 or later
2.0.17	SDK link Plug-in link	SDK link Plug-in link	1. Resolved HttpURLConnection network monitoring issues.	Android 5 or later
2.0.16	SDK link Plug-in link	SDK link Plug-in link	1. Resolved HttpURLConnection network monitoring issues.	Android 5 or later
2.0.15	SDK link Plug-in link	SDK link Plug-in link	1. Fixed the issue that the network request whitelist does not take effect.	Android 5 or later
2.0.14	SDK link Plug-in link	SDK link Plug-in link	1. Supported sampling rate configuration. 2. Optimized the collection of crash information.	Android 5 or later

Version	SDK Download Address	Verification File Download Address	Update Description	System
2.0.13	SDK link Plug-in link	SDK link Plug-in link	1. Resolved AGP8.0+ compatibility issues. 2. Supported minSDK 21.	Android 5 or later
2.0.12	SDK link Plug-in link	SDK link Plug-in link	1. Supported compatibility with Gradle 6.7.1, AGP 4.2.0, JDK 11, and Android 30 (compileSdk).	Android 7 or later
2.0.11	SDK link Plug-in link	SDK link Plug-in link	1. Optimized the data upload logic.	Android 7 or later
2.0.10	SDK link Plug-in link	SDK link Plug-in link	1. Fixed the issue that null is reported when a network error occurs. 2. Fixed the issue that the configuration is not complete when the plug-in is loaded. 3. Fixed the issue that files fail to be copied after a plug-in instrumentation exception occurs.	Android 7 or later
2.0.9	SDK link Plug-in link	SDK link Plug-in link	1. Allowed users to enable/disable the plug-in function for fault locating. 2. Fixed the issue that open tracing in the remote configuration does not take effect at the correct time. 3. Fixed the issue that repetitive ANR error stack logs are displayed.	Android 7 or later
2.0.8	SDK link Plug-in link	SDK link Plug-in link	1. Fixed the conflict caused by project configuration.	Android 7 or later

Version	SDK Download Address	Verification File Download Address	Update Description	System
2.0.7	SDK link Plug-in link	SDK link Plug-in link	1. Supported remote configuration.	Android 7 or later
2.0.6	SDK link Plug-in link	SDK link Plug-in link	1. Fixed the issue that the SDK fails to be started due to improper configuration.	Android 7 or later
2.0.5	SDK link Plug-in link	SDK link Plug-in link	1. Adapted to the transform API of Gradle 8.0 or later.	Android 7 or later
2.0.4	SDK link Plug-in link	SDK link Plug-in link	1. Adapted to the fields in the configuration file and fixed the issue that the plug-in cannot respond to log fields. 2. Upgraded network interception. Fixed the issue that reporting cannot be implemented in the case of special conditions when URLConnection is used.	Android 7 or later
2.0.1	SDK link Plug-in link	SDK link Plug-in link	1. Supported collection and reporting of app startup performance, crashes, ANRs, errors, network requests, devices, and custom statistics.	Android 7 or later

Procedure

Step 1 Integrate the access SDK.

1. Method 1: Integrate the SDK from the Maven repository.

- Add dependencies to **build.gradle**.

```
buildscript {  
    dependencies {  
        ...  
        classpath 'io.github.apm-sdk:apm-sdk-plugin:2.1.0'  
        ...  
    }  
}  
plugins {  
    ...  
    id 'org.jetbrains.kotlin.android' version 'KOTLIN_VERSION' apply false // Add kotlin 1.6.20 or later.  
}
```

- ```
...
}
```
- Add dependencies to **app/build.gradle**.

```
plugins {
 ...
 id 'org.jetbrains.kotlin.android'
 id 'kotlin-kapt'
 id 'com.cloud.apm.plugin' // Add dependency plug-ins.
 ...
}
dependencies {
 ...
 implementation 'io.github.apm-sdk:apm-sdk-android:2.1.0' // Add APM SDK dependencies.
 ...
}
```
  - Add the Maven repository to **settings.gradle**.

```
pluginManagement{
 ...
 repositories {
 ...
 // Add the following content:
 mavenCentral()
 }
}
dependencyResolutionManagement {
 ...
 repositories {
 ...
 // Add the following content:
 mavenCentral()
 }
}
```

## 2. Method 2: Manual integration

- Download the SDK and plug-in packages.
- Add the **.jar** plug-in package to your project.
- Add dependencies to **build.gradle**.

```
buildscript {
 dependencies {
 classpath fileTree(dir: $dir, include: '*.jar') // Enter the folder where the .jar package is
 located, for example, libs.
 }
}
```
- Add the **.aar** SDK package to your project.
- Add dependencies to **app/build.gradle**.

```
dependencies {
 ...
 implementation fileTree(dir: $dir, include: ['*.aar']) // Enter the folder where the .aar package
 is located, for example, libs.
 implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.4.0'
 implementation 'androidx.core:core-ktx:1.6.0'
 implementation 'com.google.code.gson:gson:2.8.9'
 kapt 'androidx.room:room-compiler:2.4.0'
 api 'androidx.room:room-runtime:2.4.0'
 api 'androidx.room:room-common:2.4.0'
 ...
}
```
- Add the Maven repository to **settings.gradle**.

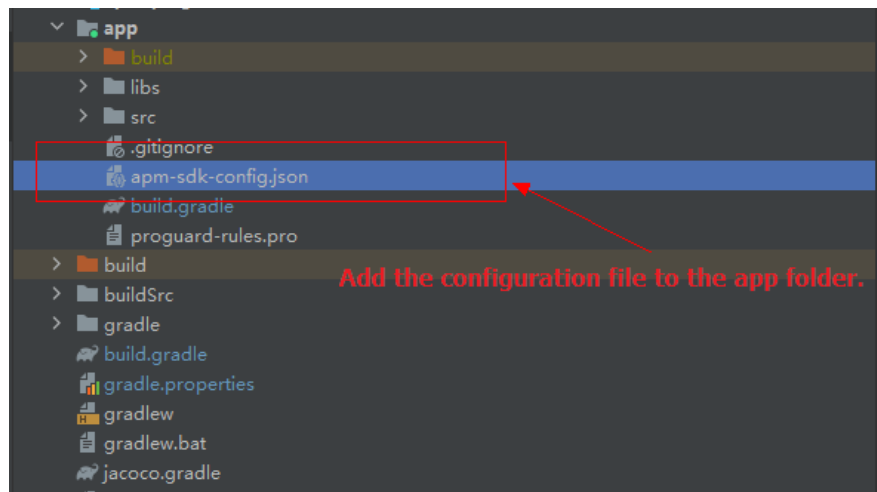
```
pluginManagement{
 ...
 repositories {
 ...
 // Add the following content:
 google()
 }
}
```

```
 mavenCentral()
 }
}
dependencyResolutionManagement {
 ...
 repositories {
 ...
 // Add the following content:
 google()
 mavenCentral()
 }
}
```

If your Gradle version is earlier than 7.0, add the Maven repository to the **build.gradle** file.

**Step 2** Add a configuration file.

Add the Android app on the console, download the **apm-sdk-config.json** configuration file, and drag it to the **root** directory of the app, as shown in the following figure.



**Table 17-4** Configuration parameters

| Parameter     | Mandato<br>ry | Default<br>Value | Description                                                                                         |
|---------------|---------------|------------------|-----------------------------------------------------------------------------------------------------|
| appld         | Yes           | -                | Mobile app ID.                                                                                      |
| authorization | Yes           | -                | Used for app authentication.                                                                        |
| region        | Yes           | -                | Region where APM is deployed. Currently, <b>cn-north-4</b> and <b>ap-southeast-3</b> are supported. |
| UID           | No            | -                | Custom ID.                                                                                          |
| tag           | No            | -                | Custom tag. Use commas (,) to separate multiple tags.                                               |
| url           | No            | -                | Public domain name for reporting data to APM.                                                       |

| Parameter        | Mandatory | Default Value | Description                                                                                                                                      |
|------------------|-----------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| networkWhiteList | No        | -             | Network whitelist, which is used to exclude network requests that do not need to be monitored.                                                   |
| cacheThreshold   | No        | 200           | When the number of records stored in the local database reaches this threshold, an alarm is reported. Value range: 30 to 1,000.                  |
| timeInterval     | No        | 60s           | Interval for triggering reporting. Value range: 60 to 1,800, in seconds.                                                                         |
| reportBackground | No        | true          | Whether to report logs upon app backend switching.                                                                                               |
| reportLaunch     | No        | false         | Whether to report logs upon app startups. To enable this function, start the SDK in the <b>onCreate</b> method of the application.               |
| enableNetwork    | No        | false         | Whether to enable network collection.                                                                                                            |
| enableCrash      | No        | false         | Whether to enable crash collection.                                                                                                              |
| enableLaunch     | No        | false         | Whether to enable startup collection.                                                                                                            |
| enableANR        | No        | false         | Whether to enable freeze collection.                                                                                                             |
| enableError      | No        | false         | Whether to enable error collection.                                                                                                              |
| enableDevice     | No        | false         | Whether to enable device collection.                                                                                                             |
| enableEvent      | No        | false         | Whether to enable custom statistics reporting.                                                                                                   |
| traceType        | No        | apm           | Whether to enable open tracing. If this function is enabled, OpenTelemetry will be used. If this function is disabled, APM tracing will be used. |
| logLevel         | No        | off           | Debug log level. Options: <b>debug</b> , <b>info</b> , <b>warn</b> , <b>error</b> , and <b>off</b> .                                             |
| samplingRate     | No        | 1.0           | Sampling rate. Range: 0–1.0.                                                                                                                     |

**Step 3** Start the SDK.

## • Kotlin code example

```
import com.cloud.apm.APMSDK

// Start the SDK based on the apm-sdk-config.json configuration file.
APMSDK.start(this)

// (Optional) Set a custom ID.
APMSDK.setUid ("Custom ID")

// (Optional) Set custom tags.
APMSDK.addTag ("Custom tag 1, Custom tag 2, Custom tag 3")
```

## • Java code example:

```
import com.cloud.apm.APMSDK

// Start the SDK based on the apm-sdk-config.json configuration file.
APMSDK.start();

// (Optional) Set a custom ID.
APMSDK.setUid ("Custom ID");

// (Optional) Set custom tags.
APMSDK.addTag ("Custom tag 1, Custom tag 2, Custom tag 3");
```

**Step 4** (Optional) Configure custom statistics reporting.

| Method                                                                                                                                                                                                                                                              | Description                                                                                                                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>fun &lt;T&gt; event(key: String, event: HashMap&lt;String, T&gt;);</code><br><code>fun &lt;T&gt; event(key: String, event: Array&lt;T&gt;)</code><br><code>fun event(key: String, event: Number)</code><br><code>fun event(key: String, event: String)</code> | <b>key</b> : event name (string). Max.: 2,048 characters. <b>value</b> : event content. Valid JSON objects such as <b>String</b> , <b>Array</b> , <b>HashMap</b> , and <b>Number</b> are supported. Max.: 30,720 characters. |

## Kotlin code example

```
// Example of event statistics
APMSDK.event("User Information", hashMapOf(Pair("name","zhang san")))
APMSDK.event("Call Stack", listOf("0x0000000101ee9c6c","0x0000600000e61d80").toTypedArray())
APMSDK.event("Error Description", "The request timed out.")

// Example of numeric statistics
APMSDK.event("Purchase Statistic", 1653)
APMSDK.event("Custom Load Time", 1.653)
```

## Java code example:

```
// Example of event statistics
HashMap<String, String> user = new HashMap();
user.put("name","zhang san");
APMSDK.event("User Information", user);
String[] stacks = {"0x0000000101ee9c6c", "0x0000600000e61d80"};
APMSDK.event("Call Stack", stacks);
APMSDK.event("Error Description", "The request timed out.");

// Example of numeric statistics
APMSDK.event("Purchase Statistic", 1653);
APMSDK.event("Custom Load Time", 1.653);
```

----End

## Obfuscation Configuration

Add the following code to the obfuscation configuration file **proguard-rules.pro**:

```
Retain the source file name and source code line number as required.
-keepattributes LineNumberTable, SourceFile

Prevent the APM SDK from being obfuscated.
-keep class com.cloud.apm.**{*;}

If Gradle 8.x is used, use the following configuration:
-keepclasseswithmembers class okhttp3.**{*;} # Use okhttp3&okhttp4.
-dontwarn okhttp3.** # okhttp3&okhttp4 is not used.
-keepclasseswithmembers class com.squareup.okhttp.**{*;} # Use okhttp2.
-dontwarn com.squareup.okhttp.** # okhttp2 is not used.
```

## ASM Version Configuration

If the following compilation errors occur:

```
java.lang.UnsupportedOperationException: NestMember requires ASM7
 at org.objectweb.asm.ClassVisitor.visitNestMember(ClassVisitor.java:265)
 at org.objectweb.asm.ClassReader.accept(ClassReader.java:697)
 at org.objectweb.asm.ClassReader.accept(ClassReader.java:425)
 at
com.cloud.apm.plugin.tasks.trace.methodTrace.MethodTracer.innerTraceMethodFromJar(MethodTracer.kt:184
)
 at com.cloud.apm.plugin.tasks.trace.methodTrace.MethodTracer.traceMethodFromJar
$lambda-4$lambda-3(MethodTracer.kt:89)
```

Add the following code to **gradle.properties**:

```
com.cloud.apm.asmApi=ASM7
```

## Setting Permissions

The APMSDK requires the network and network status permissions. The permissions have been configured in the **Manifest.xml** file of the SDK.

## Data Reporting Policy

The SDK supports the following reporting policies: threshold-based reporting, scheduled reporting, reporting when the app is switched to the backend, and reporting upon startup. Set parameters such as **cacheThreshold**, **timeInterval**, **reportBackground**, and **reportLaunch** based on the scenario.

## Enabling the Debug Log Function

Enable the debug log function if needed. Then you can monitor real-time SDK running through the console, check logs, and make adjustments. The log level can be **debug**, **info**, **warn**, **error**, or **off** (disabled).

Example: Set the debug level in the **apm-sdk-config.json** configuration file.

```
{
 "logLevel": "debug"
}
```

## 17.4 Web & HTML5 SDK Access

### Prerequisites

Before using the web & HTML5 SDK, you need to register a Huawei Cloud Account and enable APM.

### Transport Protocol

HTTPS

### Version Updates

Table 17-5 Version updates

| Version | Update Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0.62  | <ol style="list-style-type: none"><li>Supported the collection of LCP (excluding the preloading time) and the cancellation of the LCP listening event if the <b>click</b>, <b>keydown</b>, or <b>visibilitychange</b> event occurs.</li><li>Supported the function of listening on the XHR background (100 fields are intercepted). For fetch requests, <b>statusCode</b> will be listened and displayed as the background information.</li><li>Added the <b>hideHost</b> parameter, so that the domain name can be hidden on the performance loading page.</li></ol> |
| 1.0.61  | Filtered out performance loading data whose LCP value is <b>0</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 1.0.59  | Supported sampling rate settings.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 1.0.58  | Supported the Furion Agent.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 1.0.57  | Added the Cumulative Layout Shift (CLS) and Largest Contentful Paint (LCP) metrics.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 1.0.48  | Supported the collection of Interaction to Next Paint (INP). By default, this function is disabled. You can configure this function by setting the <b>enableInp</b> parameter.                                                                                                                                                                                                                                                                                                                                                                                        |
| 1.0.47  | Supported the reporting of the <b>pvid</b> and <b>ssid</b> fields for session tracing and traffic analysis.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 1.0.46  | Added the custom tags <b>uid</b> and <b>tag</b> , and added the <b>__rum.setUid</b> and <b>__rum.setTag</b> methods.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

### Procedure

**Step 1** On the **Web Monitoring** page, click **Connect Website**.

**Step 2** Select an application, enter a site name, and click **Create Website**.

**Step 3** Copy the code in **Track Service** to the entry HTML file. When the code is executed, the SDK is automatically introduced for tracking. Ensure that the app ID in the code is not empty.

**Table 17-6** Configuration parameters

| Parameter    | Mandatory | Default Value | Description                                                                                                                                                                                                                                                         |
|--------------|-----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| appId        | Yes       | -             | Website ID. Each site has its own app ID.                                                                                                                                                                                                                           |
| apiRepo      | No        | true          | Auto API reporting.<br>Reports API call details and success rate.                                                                                                                                                                                                   |
| thirdApi     | No        | true          | Third-party API reporting.<br>Reports third-party API information.                                                                                                                                                                                                  |
| hashMode     | No        | true          | Auto hash mode parsing.<br>Processes the content following # as a part of a URL. If this option is disabled, # is regarded as an anchor.                                                                                                                            |
| JsErrorClear | No        | true          | JS error tracing.<br>Listens on and reports all JS errors. This will change the service paths displayed on your browser.                                                                                                                                            |
| smartJsError | No        | true          | JS errors mitigation.<br>Stops reporting a JS error after 100 times.                                                                                                                                                                                                |
| webResource  | No        | true          | Page resource reporting.<br>Collects loading performance.                                                                                                                                                                                                           |
| traceType    | No        | apm           | Open tracing.<br><b>Tracing:</b> Traces the frontend and backend using OpenTelemetry. If this option is disabled, APM tracing will be used instead. To view the tracing effect, choose <b>API Requests</b> , go to the URL details page, and click <b>Analyze</b> . |
| UID          | No        | -             | Custom ID (string), which is added in version 1.0.46.                                                                                                                                                                                                               |
| tag          | No        | -             | Custom tag (string), which is added in version 1.0.46.                                                                                                                                                                                                              |
| enableInp    | No        | false         | Whether to collect INP.                                                                                                                                                                                                                                             |

| Parameter  | Mandatory | Default Value | Description                                                                                                                                                                     |
|------------|-----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sampleRate | No        | 1             | Sampling rate. Range: 0–1. For example, if this parameter is set to <b>0.5</b> , the sampling rate is 50%. This parameter can be configured in web & HTML5 SDK 1.0.59 or later. |

**Step 4** (Optional) Configure custom statistics reporting.

| Method                                             | Description                                                                                                                                                                                                                     |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>__rum.log(key: String, value: String)</code> | <b>key</b> : event name (string). Max.: 2,048 characters.<br><b>value</b> : event content. Valid JSON objects such as <b>String</b> , <b>Array</b> , <b>HashMap</b> , and <b>Number</b> are supported. Max.: 30,720 characters. |

**Step 5** (Optional) Dynamically set the UID.

| Method                                 | Description                                 |
|----------------------------------------|---------------------------------------------|
| <code>__rum.setUid(uid: String)</code> | <b>uid</b> (string): max. 2,048 characters. |

**Step 6** (Optional) Dynamically set the tag.

| Method                                 | Description                                 |
|----------------------------------------|---------------------------------------------|
| <code>__rum.setTag(tag: String)</code> | <b>tag</b> (string): max. 2,048 characters. |

----End

## Data Reporting Policy

1. Basic user data and custom statistics are reported immediately when a user accesses the page.
2. Page performance data is triggered to report after the page is loaded.
3. Other types of data (such as API data and JS error data) are reported every minute.

## 17.5 WeChat SDK Access

### Prerequisites

- Before using the WeChat SDK, you need to register a Huawei Cloud Account and enable APM.

- To ensure that WeChat logs can be reported, add the report address to the domain name list of the WeChat developer platform as a valid domain name.

## Transport Protocol

HTTPS

## Version Updates

**Table 17-7** Version updates

| Version | Update Description                                                                          |
|---------|---------------------------------------------------------------------------------------------|
| 2.0.7   | Supported sampling rate settings.                                                           |
| 2.0.5   | Added the <b>uid</b> and <b>tag</b> fields and the <b>setUid</b> and <b>setTag</b> methods. |
| 2.0.4   | Fixed the issue that the SDK overwrites the function scope of the applet.                   |
| 2.0.1   | Added the <b>README</b> help document.                                                      |
| 2.0.0   | This is the first release of the applet SDK.                                                |

## Configuring the Report Address

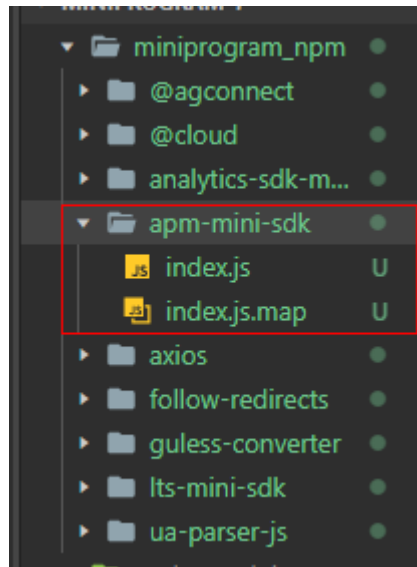
Add the APM report address **<https://apm-web.region.myhuaweicloud.com>** to the domain name list of the applet developer platform.

Currently, this function is supported in the AP-Singapore region only. The report address is **<https://apm-web.ap-southeast-3.myhuaweicloud.com>**.

## Procedure

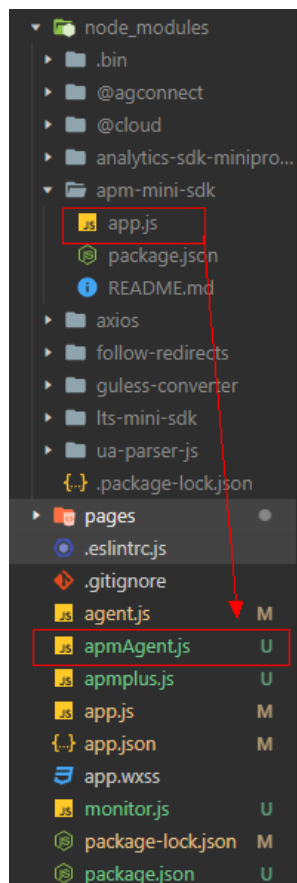
- Step 1** On the **Web Monitoring** page, click **Connect Website**.
- Step 2** Select **WeChat**, select an application, enter a site name, and click **Create Website**.
- Step 3** Install the SDK. Run the **npm i apm-mini-sdk** command in the **root** directory of the project to pull the SDK software package.
- Step 4** Enable the SDK. The code will be automatically embedded in the applet. You can execute the code anywhere in the project (**app.js** is recommended). The SDK can be imported using a file or npm.
  - Integrating the SDK using npm
    - Ensure that the project has the **package.json** file. If it does not exist, run the **npm init** command in the **root** directory of the project to create one:
    - Run the SDK installation command to install the SDK software package:

- c. Use the developer tool to build the npm library file of the current project. If the **apm-mini-sdk** file exists in the **miniprogram\_npm** folder, the import is successful.



4. Copy the code for tracking from the page to the **app.js** file.

- Integrating the SDK by importing files
  - a. Find the **app.js** file (**node\_modules** > **apm-mini-sdk** > **app.js**) in the SDK folder, copy it from **node\_module** to the **root** directory, and rename it.



- b. Copy the code for tracking from the page to the **app.js** file.

**Table 17-8** Configuration parameters

| Parameter    | Mandatory | Default Value | Description                                                                                                                                                                                                                                                         |
|--------------|-----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| appId        | Yes       | -             | Website ID. Each site has its own app ID.                                                                                                                                                                                                                           |
| apiRepo      | No        | true          | Auto API reporting.<br>Reports API call details and success rate.                                                                                                                                                                                                   |
| thirdApi     | No        | true          | Third-party API reporting.<br>Reports third-party API information.                                                                                                                                                                                                  |
| hashMode     | No        | true          | Auto hash mode parsing.<br>Processes the content following # as a part of a URL. If this option is disabled, # is regarded as an anchor.                                                                                                                            |
| JsErrorClear | No        | true          | JS error tracing.<br>Listens on and reports all JS errors. This will change the service paths displayed on your browser.                                                                                                                                            |
| smartJsError | No        | true          | JS errors mitigation.<br>Stops reporting a JS error after 100 times.                                                                                                                                                                                                |
| webResource  | No        | false         | Page resource reporting.<br>Collects loading performance. Only WeChat can collect First Paint (FP) information. Other mini programs do not support page resource reporting. If required, use the custom reporting function.                                         |
| traceType    | No        | apm           | Open tracing.<br><b>Tracing:</b> Traces the frontend and backend using OpenTelemetry. If this option is disabled, APM tracing will be used instead. To view the tracing effect, choose <b>API Requests</b> , go to the URL details page, and click <b>Analyze</b> . |
| UID          | No        | -             | Custom ID (string), which is added in version 1.0.46.                                                                                                                                                                                                               |
| tag          | No        | -             | Custom tag (string), which is added in version 1.0.46.                                                                                                                                                                                                              |

| Parameter  | Mandatory | Default Value | Description                                                                                                                                                        |
|------------|-----------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sampleRate | No        | 1             | Sampling rate. Range: 0–1. For example, if this parameter is set to <b>0.5</b> , the sampling rate is 50%. This parameter can be configured in SDK 2.0.7 or later. |

**Step 5** (Optional) Configure custom statistics reporting.

| Method                                | Description                                                                                                                                                                                                                     |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| agent.log(key: String, value: String) | <b>key</b> : event name (string). Max.: 2,048 characters.<br><b>value</b> : event content. Valid JSON objects such as <b>String</b> , <b>Array</b> , <b>HashMap</b> , and <b>Number</b> are supported. Max.: 30,720 characters. |

**Step 6** (Optional) Dynamically set the UID.

| Method                    | Description                                 |
|---------------------------|---------------------------------------------|
| agent.setUid(uid: String) | <b>uid</b> (string): max. 2,048 characters. |

**Step 7** (Optional) Dynamically set the tag.

| Method                    | Description                                 |
|---------------------------|---------------------------------------------|
| agent.setTag(tag: String) | <b>tag</b> (string): max. 2,048 characters. |

**Step 8** Customize the reporting of loading performance data. If the parameters under **PerformanceData** have values and the sum of these values is greater than 0, data will be reported.

```
agent.addPerformanceData(data: PerformanceData)
```

**Table 17-9** PerformanceData parameters

| Parameter | Type   | Mandatory | Default Value | Description                  |
|-----------|--------|-----------|---------------|------------------------------|
| loadTime  | Number | No        | 0             | Load time                    |
| fcp       | Number | No        | 0             | First Contentful Paint (FCP) |
| fp        | Number | No        | 0             | First Paint (FP)             |

| Parameter | Type   | Mandatory | Default Value | Description                     |
|-----------|--------|-----------|---------------|---------------------------------|
| domtti    | Number | No        | 0             | Time to Interactive (TTI)       |
| cls       | Number | No        | 0             | Cumulative Layout Shift (CLS)   |
| inp       | Number | No        | 0             | Interaction to Next Paint (INP) |
| lcp       | Number | No        | 0             | Largest Contentful Paint (LCP)  |

**Step 9** Customize the error data to be reported.

```
agent.addErrorData(data: ErrorData)
```

**Table 17-10** ErrorData parameters

| Parameter | Type   | Mandatory | Default Value | Description    |
|-----------|--------|-----------|---------------|----------------|
| type      | String | Yes       | None          | Error type.    |
| message   | String | Yes       | None          | Error message. |
| stack     | String | No        | None          | Error stack.   |

----End

## Data Reporting Policy

1. Basic user data and custom statistics are reported immediately when a user accesses the page.
2. Page performance data is triggered to report after the page is loaded.
3. Other types of data (such as API data and JS error data) are reported every minute.

## 17.6 Baidu SDK Access

### Prerequisites

- Before using the Baidu SDK, you need to register a Huawei Cloud Account and enable APM.

- To ensure that Baidu logs can be reported, add the report address to the domain name list of the Baidu developer platform as a valid domain name.

## Transport Protocol

HTTPS

## Version Updates

**Table 17-11** Version updates

| Version | Update Description                                                                          |
|---------|---------------------------------------------------------------------------------------------|
| 2.0.7   | Supported sampling rate settings.                                                           |
| 2.0.5   | Added the <b>uid</b> and <b>tag</b> fields and the <b>setUid</b> and <b>setTag</b> methods. |
| 2.0.4   | Fixed the issue that the SDK overwrites the function scope of the applet.                   |
| 2.0.1   | Added the <b>README</b> help document.                                                      |
| 2.0.0   | This is the first release of the applet SDK.                                                |

## Configuring the Report Address

Add the APM report address **`https://apm-web.region.myhuaweicloud.com`** to the domain name list of the applet developer platform.

Currently, this function is supported in the AP-Singapore region only. The report address is **`https://apm-web.ap-southeast-3.myhuaweicloud.com`**.

## Procedure

- Step 1** On the **Web Monitoring** page, click **Connect Website**.
- Step 2** Select **Baidu**, select an application, enter a site name, and click **Create Website**.
- Step 3** Install the SDK. Run the **`npm i apm-mini-sdk`** command in the **root** directory of the project to pull the SDK software package.
- Step 4** Enable the SDK. The code will be automatically embedded in the applet. You can execute the code anywhere in the project (**`app.js`** is recommended). The SDK can be imported using a file or npm.
  - Integrating the SDK using npm
    - Run the SDK installation command to install the SDK software package:
    - Copy the code for tracking from the page to the **`app.js`** file.

**Table 17-12** Configuration parameters

| Parameter    | Mandatory | Default Value | Description                                                                                                                                                                                                                                                         |
|--------------|-----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| appId        | Yes       | -             | Website ID. Each site has its own app ID.                                                                                                                                                                                                                           |
| apiRepo      | No        | true          | Auto API reporting.<br>Reports API call details and success rate.                                                                                                                                                                                                   |
| thirdApi     | No        | true          | Third-party API reporting.<br>Reports third-party API information.                                                                                                                                                                                                  |
| hashMode     | No        | true          | Auto hash mode parsing.<br>Processes the content following # as a part of a URL. If this option is disabled, # is regarded as an anchor.                                                                                                                            |
| JsErrorClear | No        | true          | JS error tracing.<br>Listens on and reports all JS errors. This will change the service paths displayed on your browser.                                                                                                                                            |
| smartJsError | No        | true          | JS errors mitigation.<br>Stops reporting a JS error after 100 times.                                                                                                                                                                                                |
| webResource  | No        | false         | Page resource reporting.<br>Collects loading performance. Only WeChat can collect First Paint (FP) information. Other mini programs do not support page resource reporting. If required, use the custom reporting function.                                         |
| traceType    | No        | apm           | Open tracing.<br><b>Tracing:</b> Traces the frontend and backend using OpenTelemetry. If this option is disabled, APM tracing will be used instead. To view the tracing effect, choose <b>API Requests</b> , go to the URL details page, and click <b>Analyze</b> . |
| UID          | No        | -             | Custom ID (string), which is added in version 1.0.46.                                                                                                                                                                                                               |
| tag          | No        | -             | Custom tag (string), which is added in version 1.0.46.                                                                                                                                                                                                              |
| sampleRate   | No        | 1             | Sampling rate. Range: 0–1. For example, if this parameter is set to <b>0.5</b> , the sampling rate is 50%. This parameter can be configured in SDK 2.0.7 or later.                                                                                                  |

**Step 5** (Optional) Configure custom statistics reporting.

| Method                                | Description                                                                                                                                                                                                                     |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| agent.log(key: String, value: String) | <b>key</b> : event name (string). Max.: 2,048 characters.<br><b>value</b> : event content. Valid JSON objects such as <b>String</b> , <b>Array</b> , <b>HashMap</b> , and <b>Number</b> are supported. Max.: 30,720 characters. |

**Step 6** (Optional) Dynamically set the UID.

| Method                    | Description                                 |
|---------------------------|---------------------------------------------|
| agent.setUid(uid: String) | <b>uid</b> (string): max. 2,048 characters. |

**Step 7** (Optional) Dynamically set the tag.

| Method                    | Description                                 |
|---------------------------|---------------------------------------------|
| agent.setTag(tag: String) | <b>tag</b> (string): max. 2,048 characters. |

----End

## Data Reporting Policy

1. Basic user data and custom statistics are reported immediately when a user accesses the page.
2. Page performance data is triggered to report after the page is loaded.
3. Other types of data (such as API data and JS error data) are reported every minute.

## 17.7 Alipay SDK Access

### Prerequisites

- Before using the Alipay SDK, you need to register a Huawei Cloud Account and enable APM.
- To ensure that Alipay logs can be reported, add the report address to the domain name list of the Alipay developer platform as a valid domain name.

### Transport Protocol

HTTPS

## Version Updates

Table 17-13 Version updates

| Version | Update Description                                                                          |
|---------|---------------------------------------------------------------------------------------------|
| 2.0.7   | Supported sampling rate settings.                                                           |
| 2.0.5   | Added the <b>uid</b> and <b>tag</b> fields and the <b>setUid</b> and <b>setTag</b> methods. |
| 2.0.4   | Fixed the issue that the SDK overwrites the function scope of the applet.                   |
| 2.0.1   | Added the <b>README</b> help document.                                                      |
| 2.0.0   | This is the first release of the applet SDK.                                                |

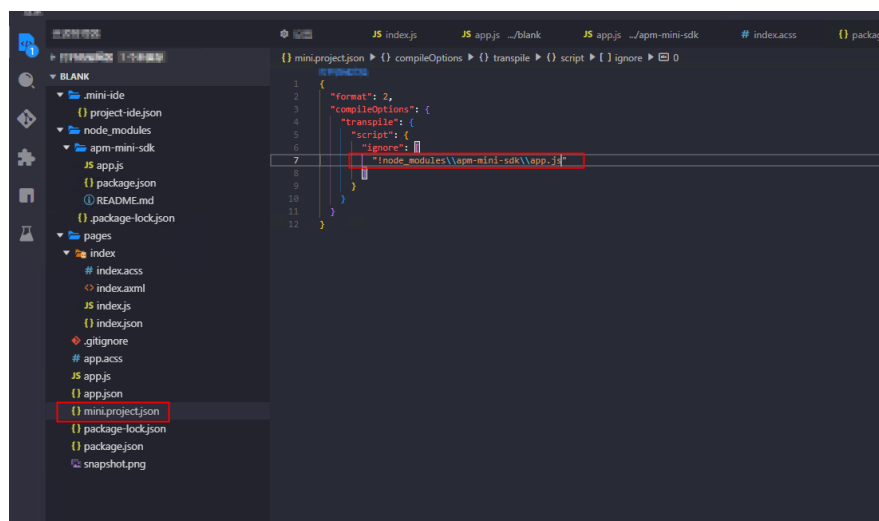
## Configuring the Report Address

Add the APM report address **<https://apm-web.region.myhuaweicloud.com>** to the domain name list of the applet developer platform.

Currently, this function is supported in the AP-Singapore region only. The report address is **<https://apm-web.ap-southeast-3.myhuaweicloud.com>**.

In the new version of the Alipay applet, ES2023 syntax compilation is enabled. It conflicts with the applet SDK 2.0.6 and 2.0.7. You need to configure the **mini.project.json** file to ignore secondary compilation of the applet SDK. Example:

```
{
 "format": 2,
 "compileOptions": {
 "transpile": {
 "script": {
 "ignore": [
 "!node_modules\\apm-mini-sdk\\app.js"
]
 }
 }
 }
}
```



## Procedure

- Step 1** On the **Web Monitoring** page, click **Connect Website**.
- Step 2** Select **Alipay**, select an application, enter a site name, and click **Create Website**.
- Step 3** Install the SDK. Run the **npm i apm-mini-sdk** command in the **root** directory of the project to pull the SDK software package.
- Step 4** Enable the SDK. The code will be automatically embedded in the applet. You can execute the code anywhere in the project (**app.js** is recommended). The SDK can be imported using a file or npm.
- Integrating the SDK using npm
    - a. Run the SDK installation command to install the SDK software package:
    - b. Copy the code for tracking from the page to the **app.js** file.

**Table 17-14** Configuration parameters

| Parameter    | Mandatory | Default Value | Description                                                                                                                                                                                                                 |
|--------------|-----------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| appId        | Yes       | -             | Website ID. Each site has its own app ID.                                                                                                                                                                                   |
| apiRepo      | No        | true          | Auto API reporting.<br>Reports API call details and success rate.                                                                                                                                                           |
| thirdApi     | No        | true          | Third-party API reporting.<br>Reports third-party API information.                                                                                                                                                          |
| hashMode     | No        | true          | Auto hash mode parsing.<br>Processes the content following # as a part of a URL. If this option is disabled, # is regarded as an anchor.                                                                                    |
| JsErrorClear | No        | true          | JS error tracing.<br>Listens on and reports all JS errors. This will change the service paths displayed on your browser.                                                                                                    |
| smartJsError | No        | true          | JS errors mitigation.<br>Stops reporting a JS error after 100 times.                                                                                                                                                        |
| webResource  | No        | false         | Page resource reporting.<br>Collects loading performance. Only WeChat can collect First Paint (FP) information. Other mini programs do not support page resource reporting. If required, use the custom reporting function. |

| Parameter  | Mandatory | Default Value | Description                                                                                                                                                                                                                                                         |
|------------|-----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| traceType  | No        | apm           | Open tracing.<br><b>Tracing:</b> Traces the frontend and backend using OpenTelemetry. If this option is disabled, APM tracing will be used instead. To view the tracing effect, choose <b>API Requests</b> , go to the URL details page, and click <b>Analyze</b> . |
| UID        | No        | -             | Custom ID (string), which is added in version 1.0.46.                                                                                                                                                                                                               |
| tag        | No        | -             | Custom tag (string), which is added in version 1.0.46.                                                                                                                                                                                                              |
| sampleRate | No        | 1             | Sampling rate. Range: 0–1. For example, if this parameter is set to <b>0.5</b> , the sampling rate is 50%. This parameter can be configured in SDK 2.0.7 or later.                                                                                                  |

**Step 5** (Optional) Configure custom statistics reporting.

| Method                                | Description                                                                                                                                                                                                                   |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| agent.log(key: String, value: String) | <b>key:</b> event name (string). Max.: 2,048 characters.<br><b>value:</b> event content. Valid JSON objects such as <b>String</b> , <b>Array</b> , <b>HashMap</b> , and <b>Number</b> are supported. Max.: 30,720 characters. |

**Step 6** (Optional) Dynamically set the UID.

| Method                    | Description                                 |
|---------------------------|---------------------------------------------|
| agent.setUid(uid: String) | <b>uid</b> (string): max. 2,048 characters. |

**Step 7** (Optional) Dynamically set the tag.

| Method                    | Description                                 |
|---------------------------|---------------------------------------------|
| agent.setTag(tag: String) | <b>tag</b> (string): max. 2,048 characters. |

----End

## Data Reporting Policy

1. Basic user data and custom statistics are reported immediately when a user accesses the page.

2. Page performance data is triggered to report after the page is loaded.
3. Other types of data (such as API data and JS error data) are reported every minute.

## 17.8 DingTalk SDK Access

### Prerequisites

- Before using the DingTalk SDK, you need to register a Huawei Cloud Account and enable APM.
- To ensure that DingTalk logs can be reported, add the report address to the domain name list of the DingTalk developer platform as a valid domain name.

### Transport Protocol

HTTPS

### Version Updates

Table 17-15 Version updates

| Version | Update Description                                                                          |
|---------|---------------------------------------------------------------------------------------------|
| 2.0.7   | Supported sampling rate settings.                                                           |
| 2.0.5   | Added the <b>uid</b> and <b>tag</b> fields and the <b>setUid</b> and <b>setTag</b> methods. |
| 2.0.4   | Fixed the issue that the SDK overwrites the function scope of the applet.                   |
| 2.0.1   | Added the <b>README</b> help document.                                                      |
| 2.0.0   | This is the first release of the applet SDK.                                                |

### Configuring the Report Address

Add the APM report address **<https://apm-web.region.myhuaweicloud.com>** to the domain name list of the applet developer platform.

Currently, this function is supported in the AP-Singapore region only. The report address is **<https://apm-web.ap-southeast-3.myhuaweicloud.com>**.

### Procedure

**Step 1** On the **Web Monitoring** page, click **Connect Website**.

**Step 2** Select **DingTalk**, select an application, enter a site name, and click **Create Website**.

**Step 3** Install the SDK. Run the **npm i apm-mini-sdk** command in the **root** directory of the project to pull the SDK software package.

**Step 4** Enable the SDK. The code will be automatically embedded in the applet. You can execute the code anywhere in the project (**app.js** is recommended). The SDK can be imported using a file or npm.

- Integrating the SDK using npm
  - a. Run the SDK installation command to install the SDK software package:
  - b. Copy the code for tracking from the page to the **app.js** file.

**Table 17-16** Configuration parameters

| Parameter    | Mandatory | Default Value | Description                                                                                                                                                                                                                 |
|--------------|-----------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| appId        | Yes       | -             | Website ID. Each site has its own app ID.                                                                                                                                                                                   |
| apiRepo      | No        | true          | Auto API reporting.<br>Reports API call details and success rate.                                                                                                                                                           |
| thirdApi     | No        | true          | Third-party API reporting.<br>Reports third-party API information.                                                                                                                                                          |
| hashMode     | No        | true          | Auto hash mode parsing.<br>Processes the content following # as a part of a URL. If this option is disabled, # is regarded as an anchor.                                                                                    |
| JsErrorClear | No        | true          | JS error tracing.<br>Listens on and reports all JS errors. This will change the service paths displayed on your browser.                                                                                                    |
| smartJsError | No        | true          | JS errors mitigation.<br>Stops reporting a JS error after 100 times.                                                                                                                                                        |
| webResource  | No        | false         | Page resource reporting.<br>Collects loading performance. Only WeChat can collect First Paint (FP) information. Other mini programs do not support page resource reporting. If required, use the custom reporting function. |

| Parameter  | Mandatory | Default Value | Description                                                                                                                                                                                                                                                         |
|------------|-----------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| traceType  | No        | apm           | Open tracing.<br><b>Tracing:</b> Traces the frontend and backend using OpenTelemetry. If this option is disabled, APM tracing will be used instead. To view the tracing effect, choose <b>API Requests</b> , go to the URL details page, and click <b>Analyze</b> . |
| UID        | No        | -             | Custom ID (string), which is added in version 1.0.46.                                                                                                                                                                                                               |
| tag        | No        | -             | Custom tag (string), which is added in version 1.0.46.                                                                                                                                                                                                              |
| sampleRate | No        | 1             | Sampling rate. Range: 0–1. For example, if this parameter is set to <b>0.5</b> , the sampling rate is 50%. This parameter can be configured in SDK 2.0.7 or later.                                                                                                  |

**Step 5** (Optional) Configure custom statistics reporting.

| Method                                | Description                                                                                                                                                                                                                   |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| agent.log(key: String, value: String) | <b>key:</b> event name (string). Max.: 2,048 characters.<br><b>value:</b> event content. Valid JSON objects such as <b>String</b> , <b>Array</b> , <b>HashMap</b> , and <b>Number</b> are supported. Max.: 30,720 characters. |

**Step 6** (Optional) Dynamically set the UID.

| Method                    | Description                                 |
|---------------------------|---------------------------------------------|
| agent.setUid(uid: String) | <b>uid</b> (string): max. 2,048 characters. |

**Step 7** (Optional) Dynamically set the tag.

| Method                    | Description                                 |
|---------------------------|---------------------------------------------|
| agent.setTag(tag: String) | <b>tag</b> (string): max. 2,048 characters. |

----End

## Data Reporting Policy

1. Basic user data and custom statistics are reported immediately when a user accesses the page.

2. Page performance data is triggered to report after the page is loaded.
3. Other types of data (such as API data and JS error data) are reported every minute.

## 17.9 HarmonyOS SDK Access

The APM HarmonyOS SDK collects and reports information about app crashes, ANRs, errors, network requests, devices, and custom events.

### Supported Platform

API 12 or later.

### Transport Protocol

HTTPS

### Integrating the SDK

**Step 1** Introduce the HarmonyOS SDK to the project.

Method 1: Pull the SDK from the HarmonyOS repository.

- Run the following command to pull the SDK from the HarmonyOS repository.

```
ohpm install apm_harmony_sdk
```

- Add dependencies to the configuration file and import the SDK.

Add dependencies to the app/oh-package.json5 file.

```
dependencies {
 ...
 "apm_harmony_sdk": "2.0.6"
 ...
}
```

../Run the following command to install the SDK:

```
ohpm install
```

Method 2: Manually import the SDK.

1. Download the [APM SDK package](#).
2. Add the downloaded HAR static library file to your project. The following uses the **plugin** folder in the **root** directory as an example.
3. Add dependencies to the **app/oh-package.json5** file.

```
dependencies {
 ...
 "apm_harmony_sdk": "file:../plugin/apm_harmony_sdk.har" //HAR package storage address
 ...
}
```

4. Run the following command to enable the SDK.

```
ohpm install
```

**Step 2** Get required permissions to report data and obtain the network status by setting the **module.json5** file. The file path is **application > src > main > module.json5**.

```
"requestPermissions": [
 {
 "name": "ohos.permission.INTERNET"
 },
 {
```

```
"name": "ohos.permission.GET_NETWORK_INFO"
}
],
```

----End

## Setting Parameters

- Step 1** Log in to the APM console and create a HarmonyOS app by referring to .
- Step 2** Download the configuration file **apm-sdk-config.json** and drag it into the project's resource directory (**src > main > resources > rawfile**).

**Table 17-17** HarmonyOS parameters

| Parameter        | Description                                                                                         | Mandatory | Default Value |
|------------------|-----------------------------------------------------------------------------------------------------|-----------|---------------|
| appld            | APM mobile app ID                                                                                   | Yes       | -             |
| authorization    | Used for app authentication.                                                                        | Yes       | -             |
| region           | Region where APM is deployed. Currently, <b>cn-north-4</b> and <b>ap-southeast-3</b> are supported. | Yes       | -             |
| UID              | Custom ID.                                                                                          | No        | -             |
| tag              | Custom tag. Use commas (,) to separate multiple tags.                                               | No        | -             |
| url              | Public domain name for reporting data to APM.                                                       | No        | -             |
| networkWhiteList | Network monitoring whitelist.                                                                       | No        | -             |
| cacheThreshold   | Cache threshold.                                                                                    | No        | 200           |
| timeInterval     | Scheduled reporting interval.                                                                       | No        | 60            |
| reportBackground | Whether to report a message when an app is switched to the background.                              | No        | true          |

| Parameter     | Description                                                                                                   | Mandatory | Default Value |
|---------------|---------------------------------------------------------------------------------------------------------------|-----------|---------------|
| logLevel      | Debug log level.<br>Options: <b>debug</b> ,<br><b>info</b> , <b>warn</b> , <b>error</b> ,<br>and <b>off</b> . | No        | off           |
| enableNetwork | Whether to<br>enable network<br>collection.                                                                   | No        | false         |
| enableCrash   | Whether to<br>enable crash<br>collection.                                                                     | No        | false         |
| enableANR     | Whether to<br>enable ANR<br>collection.                                                                       | No        | false         |
| enableError   | Whether to<br>enable error<br>collection.                                                                     | No        | false         |
| enableDevice  | Whether to<br>enable device<br>information<br>collection.                                                     | No        | false         |
| enableEvent   | Whether to<br>enable custom<br>statistics<br>collection.                                                      | No        | false         |
| traceType     | Trace type.<br>Options: <b>otel</b> and<br><b>apm</b> .                                                       | No        | apm           |

----End

## SDK Enablement

### Step 1 Import the dependency module.

```
import { APMSDK } from "apm_harmony_sdk";
```

### Step 2 Set startup and custom parameters.

```
//Start the SDK and add the following code to the onCreate function:
APMSDK.start(this.context.getApplicationContext());

// Set a custom ID.
APMSDK.setUid (Custom ID)

// Set a custom tag.
APMSDK.addTag (Custom tag);
```

----End

## Other Methods

### Step 1 Customize the error data to be reported.

```
// Specify the errors to capture and configure the function of reporting data to APM.
// 1. Capture "Error".
import { APMSDK } from "apm_harmony_sdk";
try {
 ...
} catch(e) {
 APMSDK.reportError(e);
}
// 2. Capture "BusinessError".
const response = await session.fetch(request).then((rep: rcp.Response) => {
 ...
}).catch((err: BusinessError) => {
 ...
 APMSDK.reportBusinessError(err);
});
```

### Step 2 Use the RCP interceptor for reporting API data to APM.

```
//Code example:
import { APMRcpInterceptor, ResponseCache } from "apm_harmony_sdk";
async function testInterceptor() {
 const cache: ResponseCache = new ResponseCache();
 const session = rcp.createSession({
 interceptors: [new APMRcpInterceptor(cache)]
 });
 const response = await session.get('report address');
}
```

### Step 3 Customize tracking for proactive data reporting.

```
// APMCallbackHttp: Listens to requests that contain the url, OPTIONS, and CALLBACK parameters.
import { APMCallbackHttp } from "apm_harmony_sdk";
try {
 APMCallbackHttp(
 // Customize EXAMPLE_URL in extraData on your own. You can determine whether to add parameters to the URL.
 "EXAMPLE_URL",
 {
 method: http.RequestMethod.GET, // Optional. The default value is http.RequestMethod.GET.
 // You can add header fields based on service requirements.
 header: {
 'Content-Type': 'application/json'
 }, (err: BusinessError, data: http.HttpResponse) => {
 if (!err) {
 // data.result carries the HTTP response. Parse the response based on service requirements.
 console.info('Result:' + JSON.stringify(data.result));
 console.info('code:' + JSON.stringify(data.responseCode));
 // data.header is the HTTP response header. Parse the response based on service requirements.
 console.info('header:' + JSON.stringify(data.header));
 console.info('cookies:' + JSON.stringify(data.cookies)); // 8+
 } else {
 console.error('error:' + JSON.stringify(err));
 }
 });
 } catch (e) {
 console.log(e);
 }

 // APMPromiseHttp: Listens to the requests that contain the url and OPTIONS parameters and return PROMISE.
 import { APMPromiseHttp } from "apm_harmony_sdk";
 try {
 let httpRequest = http.createHttp();
 const res = await APMPromiseHttp(
 // Customize EXAMPLE_URL in extraData on your own. You can determine whether to add parameters to the URL.
 "http://EXAMPLE_URL/apm2/web/cmd/b/business/v1/test2",
```

```
{
 method: http.RequestMethod.GET, // Optional. The default value is http.RequestMethod.GET.
 // You can add header fields based on service requirements.
 header: {
 'Content-Type': 'application/json'
 });
console.log('res:' + JSON.stringify(res));
} catch (e) {
 console.log(e);
}
```

----End

## 17.10 Flutter SDK Access

The APM Flutter SDK is based on the native APM SDK. It can collect and report app startup performance, crashes, freezes, errors, network requests, device information, and custom statistics.

### Prerequisites

Before using the SDK, you need to register a Huawei Cloud Account and create an iOS app and an Android app on the **App Monitoring** page.

### Transport Protocol

HTTPS.

### Version Updates

| Version | Description                                                                                                                                              | System                                                                                                                                      |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0.0   | 1. Supported collection and reporting of app startup performance, crashes, freezes, errors, network requests, device information, and custom statistics. | Flutter 3.35.6+, Dart 3.9.2+, iOS 15+, Android 7+, Gradle 8.11.1+, AGP: 8.9.1+, KGP 2.1.0+, JDK 17+, APM iOS SDK 2.1.0+, and APM SDK 2.1.0+ |

### Procedure

#### Step 1 Connect the SDK.

1. Add the APM Flutter SDK to **flutter\_app/pubspec.yaml**. The code is as follows:

```
dependencies:
...
 apm_flutter_plugin: 1.0.0
```

2. Add the APM Android SDK to **flutter\_app/android/build.gradle.kts**. The code is as follows:

```
buildscript {
 repositories {
 google()
 mavenCentral()
```

```
 }
 ...
 dependencies {
 ...
 classpath("io.github.apm-sdk:apm-sdk-plugin:2.1.0")
 }
}
```

3. Add the APM Android SDK to **flutter\_app/android/app/build.gradle.kts**. The code is as follows:

```
plugins {
 ...
 id("com.cloud.apm.plugin")
}
android {
 ...
 dependencies {
 ...
 implementation("io.github.apm-sdk:apm-sdk-android:2.1.0")
 }
}
```

4. Add the APM iOS SDK to **flutter\_app/ios/Podfile**. The code is as follows:
- ```
pod 'APMSDK', '2.1.0'
```

Step 2 Add a configuration file.

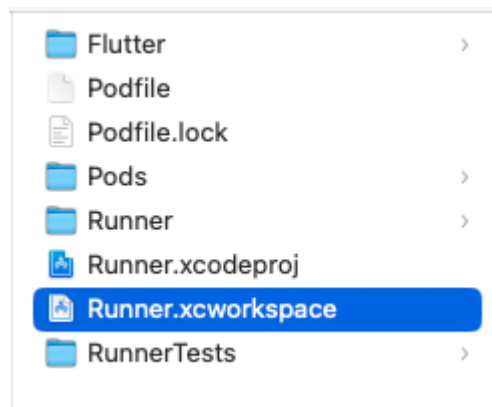
During the creation of a mobile app, a configuration file named **apm-sdk-config.json** will be generated. The following table lists the parameters.

Parameter	Mandatory	Default Value	Description
appId	Yes	-	Mobile app ID.
authorization	Yes	-	Used for app authentication.
region	Yes	-	Region where APM is deployed. Currently, ap-southeast-3 is supported.
uid	No	-	Custom ID.
tag	No	-	Custom tag. Use commas (,) to separate multiple tags.
url	No	-	Public domain name for reporting data to APM.
networkWhiteList	No	-	Network monitoring whitelist.

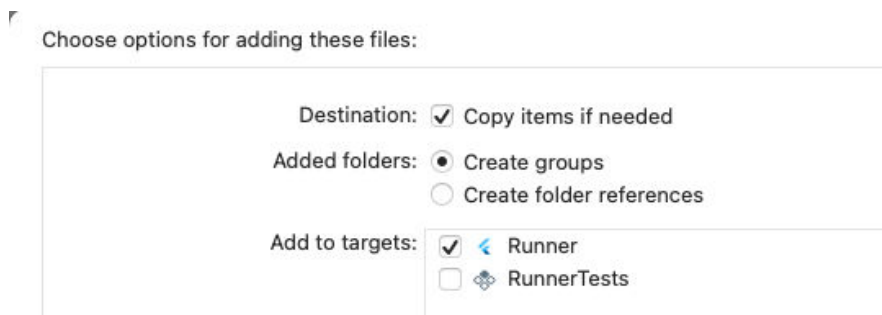
Parameter	Mandatory	Default Value	Description
cacheThreshold	No	200	When the number of records stored in the local database reaches this threshold, an alarm is reported. Value range: 30 to 1,000.
timeInterval	No	60s	Interval for triggering reporting. Value range: 60 to 1,800, in seconds.
reportBackground	No	true	Whether to report logs upon app backend switching.
reportLaunch	No	false	Whether to report logs upon app startups.
enableNetwork	No	false	Whether to enable network collection.
enableCrash	No	false	Whether to enable crash collection.
enableLaunch	No	false	Whether to enable startup collection.
enableANR	No	false	Whether to enable freeze collection.
enableError	No	false	Whether to enable error collection.
enableDevice	No	false	Whether to enable device collection.
enableEvent	No	false	Whether to enable custom statistics reporting.

Parameter	Mandatory	Default Value	Description
traceType	No	apm	Whether to enable open tracing. If this function is enabled, OpenTelemetry will be used. If this function is disabled, APM tracing will be used.
logLevel	No	off	Debug log level. Options: debug , info , warn , error , and off .
samplingRate	No	1.0	Sampling rate. Range: 0–1.0.

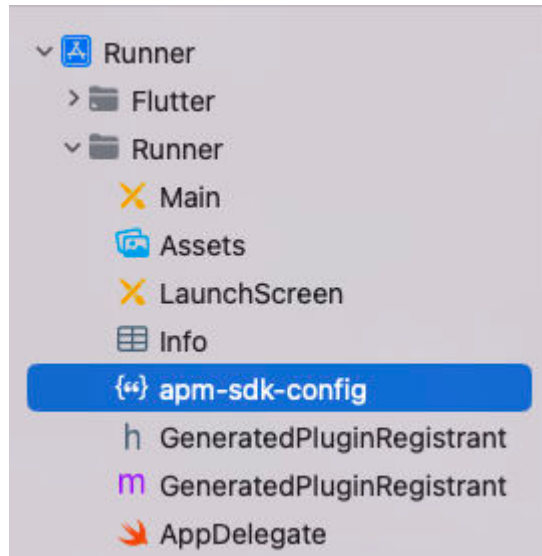
- Add a configuration file to the iOS app.
 - a. Use Xcode to open **flutter_app/ios/Runner.xcworkspace**.



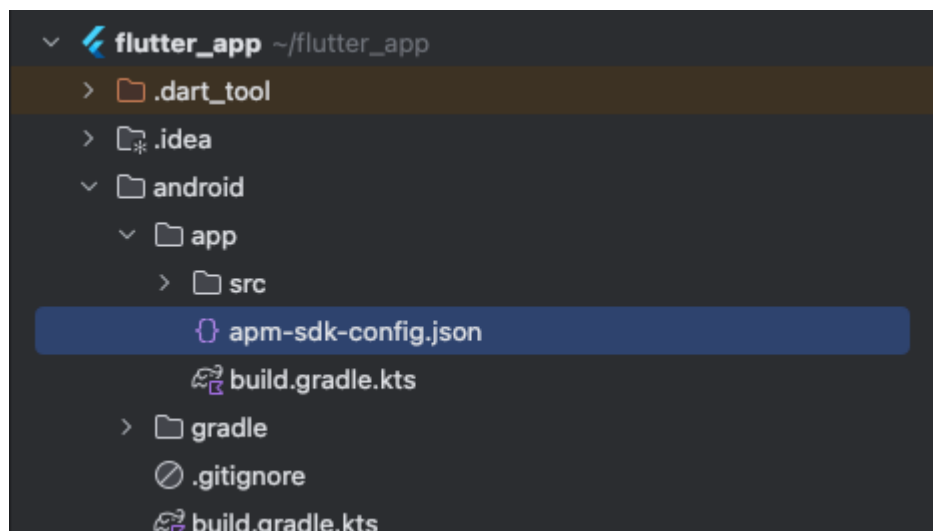
- b. Download the configuration file, drag it to the **root** directory of Runner, and configure the file as shown in the following figure.



- c. Check whether the configuration file is successfully added to the iOS app.



- Add a configuration file to the Android app.
 - a. Download the configuration file and drag it to the **root** directory of the app module.



Step 3 Start the SDK.

- Starting the APM Flutter SDK

```
import 'package:apm_flutter_plugin/apm_flutter_plugin.dart';

void main() {
  APMFlutterSDK.runApp(() async {
    .....// (Optional) Set a custom ID.
    APMFlutterSDK.setUid("custom ID");
    // (Optional) Set custom tags.
    APMFlutterSDK.addTag ("Custom tag 1,Custom tag 2,Custom tag 3");
    // Run the app.....
    return runApp(const MyApp());
  });
}
```

- Starting the APM Android SDK

```
import com.cloud.apm.APMSDK;
import com.cloud.apm_flutter_plugin.APMFlutterPlugin;

public class MyApplication extends Application {
```

```
@Override
public void onCreate() {
    super.onCreate();
    // Start the APM SDK.
    var extraConfig = new HashMap<String, Object>();
    extraConfig.put("crossPlatformUA", APMFlutterPlugin.crossPlatformUA());
    APMSDK.start(this, extraConfig);
}
}
```

Step 4 Configure custom statistics reporting.

Method	Description
static Future<void> event(String key, Object value)	key: event name. Max.: 2,048 characters. value: event content, which can be a string, list, map, or number. Max.: 30,720 characters.

Sample code:

```
// Event statistics
APMFlutterSDK.event("User Information", {"name": "XXX"});
APMFlutterSDK.event("Error Description", "The request timed out.");
APMFlutterSDK.event("Call Stack", ["0x0000000101ee9c6c", "0x0000600000e61d80"]);
// Numeric statistics
APMFlutterSDK.event("Custom Load Time", 0.238);
APMFlutterSDK.event("Purchase Statistics", 1);
```

----End

Obfuscation Configuration

Add the following code to the obfuscation configuration file **proguard-rules.pro**:

```
# Retain the source file name and source code line number as required.
-keepattributes LineNumberTable, SourceFile

# Prevent the APM SDK from being obfuscated.
-keep class com.cloud.apm_flutter_plugin.**{*,*}
-keep class com.cloud.apm.**{*,*}

# Use the following configuration as required.
-keepclasseswithmembers class okhttp3.**{*,*} # Use okhttp3&okhttp.
-dontwarn okhttp3.** # okhttp3&okhttp4 is not used.
-keepclasseswithmembers class com.squareup.okhttp.**{*,*} # Use okhttp2.
-dontwarn com.squareup.okhttp.** # okhttp2 is not used.
```

ASM Version Configuration

If the following compilation errors occur:

```
java.lang.UnsupportedOperationException: NestMember requires ASM7
    at org.objectweb.asm.ClassVisitor.visitNestMember(ClassVisitor.java:265)
    at org.objectweb.asm.ClassReader.accept(ClassReader.java:697)
    at org.objectweb.asm.ClassReader.accept(ClassReader.java:425)
    at
com.cloud.apm.plugin.tasks.trace.methodTrace.MethodTracer.innerTraceMethodFromJar(MethodTracer.kt:184)
    at com.cloud.apm.plugin.tasks.trace.methodTrace.MethodTracer.traceMethodFromJar
    $lambda-4$lambda-3(MethodTracer.kt:89)
```

Add the following code to **flutter_app/android/gradle.properties**:

```
com.cloud.apm.asmApi=ASM7
```

Data Reporting Policy

The SDK supports four reporting policies: threshold-based reporting, scheduled reporting, reporting upon app backend switching, and reporting upon app startups. Configure **cacheThreshold**, **timeInterval**, **reportBackground**, and **reportLaunch** based on scenarios.

Enabling the Debug Log Function

Enable the debug log function if needed. Then you can monitor real-time SDK running through the console, check logs, and make adjustments. The log level can be **debug**, **info**, **warn**, **error**, or **off** (disabled).

Disable the debug log function before bringing an app online. That is, set the log level to **off**.

Example: Set the log level in the **apm-sdk-config.json** configuration file.

```
{
  "logLevel": "debug"
}
```

17.11 Error Codes

Table 17-18 SDK error codes

Error Type	Error Code	Description	Possible Cause	Solution
Parameter error (APM.00XX)	APM.0001	%@ is null.	%@ is null.	Set the parameter and try again.
	APM.0002	%@ is invalid.	The type or format of %@ is invalid.	Set valid parameters and try again.
	APM.0003	The length of %@ exceeds the maximum value of %@.	Too many characters.	Modify the parameter and try again.
	APM.0004	The value of %@ must be between %@ and %@.	The value is out of the permitted range.	Modify the parameter and try again.
	APM.0006	%@ doesn't match pattern.	The parameter does not match the regular expression.	Set valid parameters and try again.
	APM.0007	Invalid configuration parameters.	Invalid parameters.	Set valid parameters and try again.

Error Type	Error Code	Description	Possible Cause	Solution
	APM.0008	This feature needs to be enabled in the configuration file.	The feature is disabled.	Enable this feature in the configuration file.
	APM.0009	Unsupported region.	The region is not supported.	Switch to the cn-north-4 region.
Integration error (APM.01XX)	APM.0100	The APMSDK is not initialized.	The SDK has not been initialized.	Set valid parameters to initialize the SDK.
	APM.0102	The APMSDK requires %@ or higher.	The OS version must be %@ or higher.	Ensure that the OS version meets requirements.
	APM.0103	Open database failed, code= %@.	Failed to open the database.	Handle the problem based on the SQLite 3 error code.
	APM.0104	Create database table failed, code= %@, error= %@.	Failed to create the data table.	Handle the problem based on the SQLite 3 error code.
	APM.0105	Insert into database failed, code= %@, error= %@.	Failed to insert data.	Handle the problem based on the SQLite 3 error code.
	APM.0106	Delete from database failed, code= %@, error= %@.	Failed to delete the data.	Handle the problem based on the SQLite 3 error code.
	APM.0107	Select from database failed, code= %@, error= %@.	Failed to query the data.	Handle the problem based on the SQLite 3 error code.
	APM.0108	Close database failed, code= %@.	Failed to close the database.	Handle the problem based on the SQLite 3 error code.
	APM.0109	Reduce database size failed, code= %@, error= %@.	Failed to reduce the database size.	Handle the problem based on the SQLite 3 error code.

Error Type	Error Code	Description	Possible Cause	Solution
	APM.0 110	Begin transaction failed, code=%@, error=%@.	Failed to enable the transaction.	Handle the problem based on the SQLite 3 error code.
	APM.0 111	Commit transaction failed, code=%@, error=%@.	Failed to submit the transaction.	Handle the problem based on the SQLite 3 error code.
	APM.0 112	The count of database records exceeds the maximum value of %@, will delete %@ records before inserting.	There are too many logs.	Automatically delete old logs before inserting new logs.
	APM.0 113	Update column(%@) to database failed, code=%d, error=%s.	Failed to update data.	Handle the problem based on the SQLite 3 error code.
System error (APM.02 XX)	APM.0 202	Write to file failed.	Failed to write the file.	Check the permissions on the file.
	APM.0 203	No file read and write permissions.	No file read and write permissions.	Grant the read and write permissions on the file or directory.
	APM.0 204	Call the system API failed, API=%s, error=%s.	An error occurred when calling the system API.	Handle the problem based on the error description.
Network error (APM.03 XX)	APM.0 300	Request failed, response code=%@, error=%@.	Failed to send the request.	For details, see Status Codes .

18 Using IAM to Grant Access to APM

18.1 Using IAM Roles or Policies to Grant Access to APM

System-defined permissions in "Role/Policy-based Authorization" provided by [Identity and Access Management \(IAM\)](#) let you control access to APM. With IAM, you can:

- Create IAM users for personnel based on your enterprise's organizational structure. Each IAM user has their own identity credentials for accessing APM resources.
- Grant users only the permissions required to perform a given task based on their job responsibilities.
- Entrust a Huawei Cloud account or a cloud service to perform efficient O&M on your APM resources.

If your Huawei Cloud account meets your permissions requirements, you can skip this section.

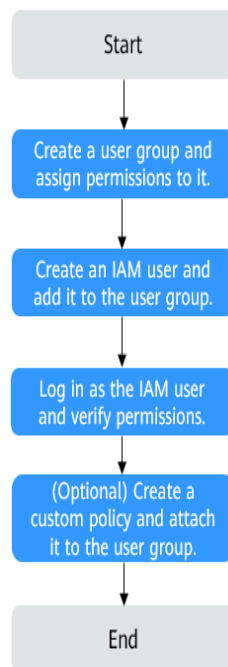
[Figure 18-1](#) shows the process flow of role/policy-based authorization.

Prerequisites

Before granting permissions to user groups, learn about system-defined permissions in [Role/Policy-based Permissions Management](#) for APM. To grant permissions for other services, learn about all [system-defined permissions](#) supported by IAM.

Process Flow

Figure 18-1 Process of granting APM permissions using role/policy-based authorization



1. On the IAM console, **create a user group and grant it permissions**.
Create a user group on the IAM console and assign the **APM ReadOnlyAccess** permissions to the group.
2. **Create an IAM user and add it to the created user group**.
On the IAM console, create a user and add it to the user group created in 1.
3. **Log in as the IAM user** and verify permissions.
In the authorized region, perform the following operations:
 - Choose **Service List > Application Performance Management**. If you can only check APM data but cannot connect an Agent, the **APM ReadOnlyAccess** policy is in effect.
 - Choose another service from **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **APM ReadOnlyAccess** policy is in effect.

Example Custom Policies

You can create custom policies to supplement the system-defined policies of APM. For details about actions supported in custom policies, see *Application Performance Management API Reference* > "Permissions and Supported Actions" > "Actions Supported by Policy-based Authorization."

To create a custom policy, choose either visual editor or JSON.

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy grammar.
- JSON: Create a JSON policy or edit an existing one.

For details, see [Creating a Custom Policy](#). The following lists examples of common APM custom policies.

- Example 1: Enable and disable collection for a specified instance.

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apm:apm2BusinessInstance:update"
      ]
    }
  ]
}
```

- Example 2: Delete an Agent.

A policy with only "Deny" permissions must be used together with other policies. If the permissions granted to an IAM user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

Assume that you want to grant the permissions of the **APM Admin** policy to a user but want to prevent them from deleting Agents. You can create a custom policy for denying Agent deletion, and attach this policy together with the **APM Admin** policy to the user. As an explicit deny in any policy overrides any allows, the user can perform all operations on APM excepting deleting Agents.

Example policy denying Agent deletion:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "apm:apm2BusinessInstance:delete"
      ],
      "Effect": "Deny"
    }
  ]
}
```

- Example 3: Create a custom policy containing multiple actions.

A custom policy can contain the actions of one or multiple services that are of the same type (global or project-level).

Example policy containing multiple actions:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apm:apm2BusinessInstance:get",
        "apm:apm2BusinessInstance:create"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:instanceScheduledEvents:list"
      ]
    }
  ]
}
```

```
} ]
```

18.2 Using IAM Identity Policies to Grant Access to APM

System-defined permissions in "Identity Policy-based Authorization" provided by [Identity and Access Management \(IAM\)](#) let you control access to APM. With IAM, you can:

- Create IAM users or user groups for personnel based on your enterprise's organizational structure. Each IAM user has their own identity credentials for accessing APM resources.
- Grant users only the permissions required to perform a given task based on their job responsibilities.
- Entrust a Huawei Cloud account or a cloud service to perform efficient O&M on your APM resources.

If your Huawei Cloud account meets your permissions requirements, you can skip this section.

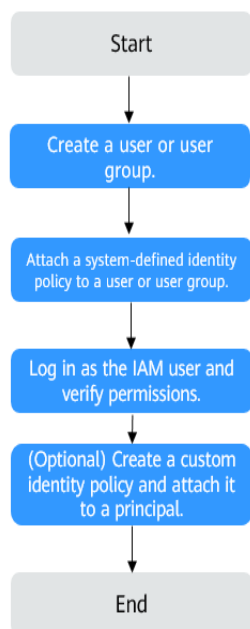
[Figure 18-2](#) shows the process flow of identity policy-based authorization.

Prerequisites

Before granting permissions, learn about system-defined permissions for APM. For details, see [Identity Policy-based Authorization](#). To grant permissions for other services, learn about all [system-defined permissions](#) supported by IAM.

Process Flow

Figure 18-2 Process of granting APM permissions using identity policy-based authorization



1. On the IAM console, **create an IAM user** or **create a user group**.
2. **Attach a system-defined identity policy** to the user or user group.
Assign the permissions defined in the system-defined identity policy **APM ReadOnlyAccess** to the user or group, or attach the system-defined identity policy to it.
3. **Log in as the IAM user** and verify permissions.
In the authorized region, perform the following operations:
 - Choose **Service List > Application Performance Management**. If you can only check APM data but cannot connect an Agent, the **APM ReadOnlyAccess** policy is in effect.
 - Choose another service from **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **APM ReadOnlyAccess** policy is in effect.

Example Custom Identity Policies

You can create custom identity policies to supplement the system-defined identity policies of APM. For details about actions supported in custom identity policies, see *Application Performance Management API Reference* > Permissions and Supported Actions > Actions Supported by Identity Policy-based Authorization.

To create a custom identity policy, choose either visual editor or JSON.

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy grammar.

- JSON: Create a JSON policy or edit an existing one.

For details, see [Creating a Custom Identity Policy and Attaching It to a Principal](#).

When creating a custom identity policy, use the Resource element to specify the resources the identity policy applies to and use the Condition element (service-specific condition keys) to control when the identity policy is in effect. For details about the supported resource types and condition keys, see [Actions Supported by Identity Policy-based Authorization](#). The following lists examples of common APM custom identity policies.

- Example 1: Grant permission to create and delete APM applications.

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apm:application:create",
        "apm:application:delete"
      ]
    }
  ]
}
```

- Example 2: Create a custom identity policy containing multiple actions.
A custom identity policy can contain the actions of one or more services.
Example identity policy containing multiple actions:

```
{
  "Version": "5.0",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apm:application:create",
        "apm:application:delete"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:cloudServers:listServersDetails"
      ]
    }
  ]
}
```

19 CTS Auditing

19.1 APM Operations Supported by CTS

Scenario

Cloud Trace Service (CTS) is available on the cloud platform. With CTS, you can record operations for later query, auditing, and backtracking.

Prerequisite

You have enabled CTS.

Key Operations Recorded by CTS

Table 19-1 Operations that can be recorded by CTS

Operation	Resource Type	Trace
Creating an AK/SK	AKSK	create-AKSK
Deleting an AK/SK	AKSK	delete-AKSK
Modifying an AK/SK	AKSK	update-AKSK
Changing the Agent collection status	AGENT	change-agent-status
Deleting an Agent	AGENT	delete-agent
Changing the profiler status	AGENT	profiler-status
Creating an account	account	create-account
Deleting an account	account	delete-account
Querying accounts by page	account	get-accounts

Operation	Resource Type	Trace
Updating an account	account	update-account
Creating an application	application	create-application
Deleting an application	application	delete-application
Updating an application	application	update-application
Modifying default attributes of an application	application	update-default-application
Deleting an environment	environment	delete-env
Disassociating a tag from an environment	envTag	delete-envTag
Modifying the environment associated with a tag	envTag	update-envs-of-tag
Modifying the tag associated with an environment	envTag	update-tags-of-env
Adding a sub-application	subApplication	create-subApplication
Deleting a sub-application	subApplication	delete-subApplication
Updating a sub-application	subApplication	update-subApplication
Creating a tag	tag	create-tag
Deleting a tag	tag	delete-tag
Obtaining the tag list	tag	get-envTag-List
Updating a tag	tag	update-tag
Saving the general configuration of a tenant	domain-common-config	save-domain-common-config
Changing the monitoring item status	monitor-item	change-monitor-item-status
Saving monitoring item information	monitor-item	save-monitor-item-config
Modifying specifications	spec	change-spec
Enabling the enterprise edition of APM	open-normal-service	open-APM-service

Operation	Resource Type	Trace
Enabling the free edition of APM	open-trail-service	open-trail-version
Querying the billing	billing	inquiry-price
Generating a Cloud Business Center (CBC) order ID	billing	create-order
Registering a region	region	register-region
Registering an Agent	agent	register-agent
Synchronizing Agent information	agent	sync
Registering a processing unit	process-unit	register-process-unit
Adding an alarm rule	alarm	add-an-alarm-rule
Deleting an alarm rule	alarm	delete-an-alarm-rule
Updating an alarm rule	alarm	update-an-alarm-rule
Updating the alarm rule status	alarm	update-an-alarm-rule-status
Copying an alarm template	alarm	copy-alarm-templates
Adding an alarm template rule	alarm	add-an-alarm-template-rule
Adding the basic information about an alarm template	alarm	add-an-alarm-template-base
Deleting an alarm template rule	alarm	delete-an-alarm-template-rule
Deleting the binding relationship between a template and alarm rule	alarm	delete-an-alarm-template-relation
Deleting an alarm template	alarm	delete-an-alarm-template
Updating an alarm template rule	alarm	update-an-alarm-template-rule
Updating the alarm template rule and environment status	alarm	update-an-alarm-template-rule-with-env-status

Operation	Resource Type	Trace
Updating the alarm template rule status	alarm	update-template-alarm-rule-status
Updating the basic information about an alarm template	alarm	update-template-base
Adding or updating the binding relationship between a template and alarm rule	alarm	update-template-relation
Updating debugging alarm information	alarm	update-debug-config
Adding SMN topic configuration	alarm	add-smn-topic-info
Deleting SMN topic configuration	alarm	delete-smn-topic-info
Updating SMN topic configuration	alarm	update-smn-topic-info
Adding user information	alarm	add-user-information
Deleting user information	alarm	delete-user-information
Updating user information	alarm	update-user-information
Adding transaction configuration	transaction	add-transaction-config
Deleting transaction configuration	transaction	delete-transaction-config
Saving masking configuration	desensitization	create-env-config
Modifying masking configuration	desensitization	update-env-config
Deleting configuration information	desensitization	delete-env-config
Creating a resource tag	tms-tag	batch-create-tms-tag
Deleting a resource tag	tms-tag	delete-tms-tag
Modifying a resource tag	tms-tag	update-tms-tag

Operation	Resource Type	Trace
Creating a resource tag on Tag Management Service (TMS)	tms-tag	create-tms-tags
Deleting a resource tag on TMS	tms-tag	delete-tms-tags
Creating a web monitoring site	rum	create-rum-instance
Updating a web monitoring site	rum	update-rum-instance
Deleting a web monitoring site	rum	delete-rum-instance
Updating the status of a web monitoring site	rum	update-rum-instance-status
Creating web monitoring rule configuration	rum	create-rum-config
Updating web monitoring rule configuration	rum	update-rum-config
Deleting web monitoring rule configuration	rum	delete-rum-config
Uploading a sourcemap file	rum	sourcemap-upLoad
Creating an application for tracing	skywalking	create-sw-business
Updating a sampling rate	skywalking	update-samplingrate
Defining a slow API	skywalking	update-slow-interface
Defining a slow SQL statement	skywalking	update-slow-sql
Generating a token	skywalking	create-token
Deleting a business	skywalking	delete-business
Deleting an Agent	skywalking	delete-agent
Deleting an environment	skywalking	delete-sw-env
Saving log association settings	lts-mapping	create-lts-mapping

Operation	Resource Type	Trace
Saving profiler performance analysis configuration	profiler-mapping	create-profiler-mapping
Querying profiler performance analysis configuration	profiler-mapping	list-profiler-mapping
Saving the association analysis settings of microservices and infrastructure	aom-mapping	create-aom-mapping
Querying the association analysis settings of microservices and infrastructure	aom-mapping	list-aom-mapping
Deleting the association analysis settings of microservices and infrastructure	aom-mapping	remove-aom-mapping
Copying custom settings in batches	aom-mapping	create-custom-setting-mapping
Creating an app	rum-app	create-rum-app-instance
Updating an app	rum-app	update-rum-app-instance
Deleting an app	rum-app	delete-rum-app-instance
Updating the app status	rum-app	update-rum-app-instance-status
Creating a report push rule	report	create-report-rule
Deleting a report push rule	report	delete-report-rule
Editing a report push rule	report	update-report-rule
Creating or editing component settings	component-setting	create-or-update-component-setting

19.2 Viewing CTS Traces in the Trace List

Scenarios

Cloud Trace Service (CTS) records operations performed on cloud service resources. A record contains information such as the user who performed the operation, IP address, operation content, and returned response message. These records facilitate security auditing, issue tracking, and resource locating. They also help you plan and use resources, and identify high-risk or non-compliant operations.

This section describes how to query or export operation records of the last seven days on the CTS console.

What Is a Trace?

A trace is an operation log for a cloud service resource, tracked and stored by CTS. Traces record operations such as adding, modifying, or deleting cloud service resources. You can view them to identify who performed operations and when for detailed tracking.

What Is a Management Tracker and Data Tracker?

A management tracker identifies and associates with all your cloud services, recording all user operations. It records management traces, which are operations performed by users on cloud service resources, such as their creation, modification, and deletion.

A data tracker records details of user operations on data in OBS buckets. It records data traces reported by OBS, detailing user operations on data in OBS buckets, including uploads and downloads.

Constraints

- Before the organization function is enabled, you can query the traces of a single account on the CTS console. After the organization function is enabled, you can only view multi-account traces on the **Trace List** page of each account, or in the OBS bucket or the **CTS/system** log stream configured for the management tracker with the organization function enabled. For details about organization trackers, see [Organization Trackers](#).
- You can only query operation records of the last seven days on the CTS console. They are automatically deleted upon expiration and cannot be manually deleted. To store them for longer than seven days, configure transfer to Object Storage Service (OBS) or Log Tank Service (LTS) so that you can view them in the OBS buckets or LTS log streams.
- After creating, modifying, or deleting a cloud service resource, you can query management traces on the CTS console one minute later.
- Data traces cannot be viewed on the CTS console. To view data traces, perform the following steps:

When configuring a data tracker, enable **Transfer to LTS**. This will automatically transfer data traces recorded by CTS to LTS. You can then view

audit logs of data traces in an LTS log stream. For details, see [Transferring CTS Traces to LTS and Viewing Them](#).

Prerequisites

1. **Register with Huawei Cloud and complete real-name authentication.**

If you already have a Huawei Cloud account, skip this step. If you do not have one, do as follows:

- a. Log in to the [Huawei Cloud official website](#), and click **Sign Up**.
- b. Sign up for a HUAWEI ID as prompted. For details, see [Signing Up for a Huawei Cloud Account](#).

Your personal information page is displayed after the registration completes.

- c. Complete [real-name authentication](#) for your individual or enterprise account.

2. **Grant permissions for users.**

If you log in to the console using a Huawei Cloud account, skip this step.

If you log in to the console as an IAM user, first contact your CTS administrator (account owner or a user in the **admin** user group) to obtain the **CTS FullAccess** permissions. For details, see [Assigning Permissions to an IAM User](#).

Viewing Traces in the Trace List

Step 1 Log in to the [CTS console](#).

Step 2 In the navigation pane, choose **Trace List**.

Step 3 In the time range drop-down list above the trace list, select a desired query time range: **Last 1 hour**, **Last 1 day**, or **Last 1 week**. You can also specify a custom time range within the last seven days.

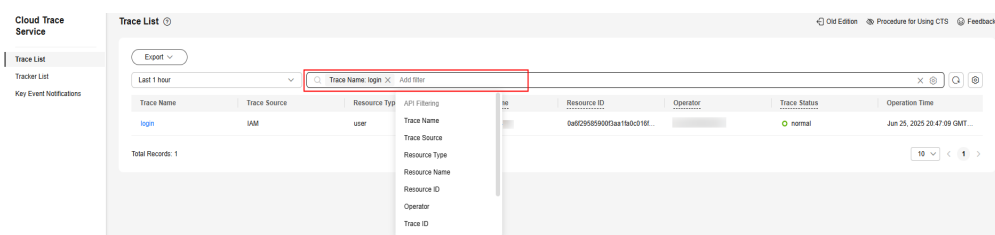
Step 4 The search box above the trace list supports advanced queries. Combine one or more filters to refine your search.

Table 19-2 Trace filtering parameters

Parameter	Description
Read-Only	After selecting the Read-Only filter, you can select either Yes or No from the drop-down list. • Yes: filters read-only operation traces, for example, resource query operations. This option is available after Read-Only Trace Reporting has been enabled in the Configuration Center and at least one read-only trace has been triggered. • No: filters non-read-only operation traces, such as creating, modifying, and deleting resources.

Parameter	Description
Trace Name	Name of a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. For details about the operations that can be audited for each cloud service, see Supported Services and Operations. Example: updateAlarm
Trace Source	Cloud service name abbreviation. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. Example: IAM
Resource Name	Name of a cloud resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty. Example: ecs-name
Resource ID	ID of a cloud resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. Leave this field empty if the resource has no resource ID or if resource creation failed. Example: <i>{VM ID}</i>
Trace ID	Value of the trace_id parameter for a trace reported to CTS. The entered value requires an exact match. Fuzzy matching is not supported. Example: 01d18a1b-56ee-11f0-ac81-*****1e229
Resource Type	Type of a resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. For details about the resource types of each cloud service, see Supported Services and Operations. Example: user

Parameter	Description
Operator	User who triggers a trace. Select one or more operators from the drop-down list. If the value of trace_type in a trace is SystemAction , the operation is triggered by the service and the trace's operator may be empty. For details about the relationship between IAM identities and operators and the operator username format, see Relationship Between IAM Identities and Operators .
Trace Status	Select one of the following options from the drop-down list: ● normal: The operation succeeded. ● warning: The operation failed. ● incident: The operation caused a fault that is more serious than a normal failure, for example, causing other faults.
Enterprise Project ID	ID of the enterprise project to which a resource belongs. To check enterprise project IDs, go to the Enterprise Project Management Service (EPS) console and choose Project Management in the navigation pane. Example: b305ea24-c930-4922-b4b9-*****1eb2
Access Key	Temporary or permanent access key ID. To check access key IDs, hover over your username in the upper right corner of the console and select My Credentials from the pop-up list. On the displayed page, choose Access Keys in the navigation pane. Example: HSTAB47V9V*****TLN9



Step 5 On the **Trace List** page, you can also export and refresh the trace list, and customize columns to display.

- Enter any keyword in the search box and press **Enter** to filter desired traces.
- Click **Export** to export all traces in the query result as a .csv file. The file can contain up to 5,000 records.
- Click  to view the latest information about traces.

- Click  to customize the information to be displayed in the trace list. If **Auto wrapping** is enabled (), excess text will move down to the next line; otherwise, the text will be truncated. By default, this function is disabled.

----End

Helpful Links

- For details about the key fields in the trace structure, see [Trace Structure](#) and [Example Traces](#).
- You can use the following examples to learn how to query a specific trace:
 - Use CTS to audit Elastic Volume Service (EVS) creation and deletion operations from the last two weeks. For details, see [Security Auditing](#).
 - Use CTS to locate a fault or creation failure for an Elastic Cloud Server (ECS). For details, see [Fault Locating](#).
 - Use CTS to check all operation records for an ECS. For details, see [Resource Tracking](#).