ALIBABA CLOUD

Alibaba Cloud

Application Real-time Monitoring Service

Product overview

Document Version: 20210308

Alibaba Cloud

Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

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

Table of Contents

1.What is ARMS?

Application Real-Time Monitoring Service (ARMS) is an application performance management (APM) service. It provides three sub-services: Browser Monitoring, Application Monitoring, and Prometheus Monitoring. ARMS provides APM features that cover distributed applications, containers, browsers, mini-programs, and mobile apps. You can use ARMS to implement full-stack performance monitoring and perform end-to-end tracking. ARMS makes your O&M simple and efficient.

Features

ARMS provides the following sub-services and function modules for various monitoring scenarios.

Application Monitoring

ARMS Application Monitoring is an application performance management (APM) service. To monitor an application, you only need to install the ARMS agent. You do not need to modify the code of the application. The ARMS agent helps you identify abnormal and slow API operations, view request parameters, and detect system bottlenecks. This improves the efficiency of online troubleshooting.

> **Note** The asterisk (*) indicates that Tracing Analysis must be activated.

For more information about ARMS Application Monitoring, see Overview.

Supports various programming languages

Introduction to Application Monitoring

Browser Monitoring

ARMS Browser Monitoring is applicable to scenarios such as web page monitoring, Weex monitoring, and mini-program monitoring. You can monitor web pages and mini-programs based on the following metrics: page loading speed (speed test), page stability (JavaScript errors), and success rate of external service calls (API).

For more information about ARMS Browser Monitoring, see What is ARMS Browser Monitoring?

Applicable to various scenarios

Introduction to Browser Monitoring

Prometheus Monitoring

Prometheus Monitoring is a managed monitoring service of ARMS. Prometheus Monitoring is compatible with the open source Prometheus ecosystem. Prometheus Monitoring monitors a wide variety of components and provides various ready-to-use dashboards.

For more information about ARMS Prometheus Monitoring, see Prometheus monitoring overview

Applicable to various scenarios

Introduction to Prometheus Monitoring

Customer stories

The following figures show the comments of some ARMS customers.

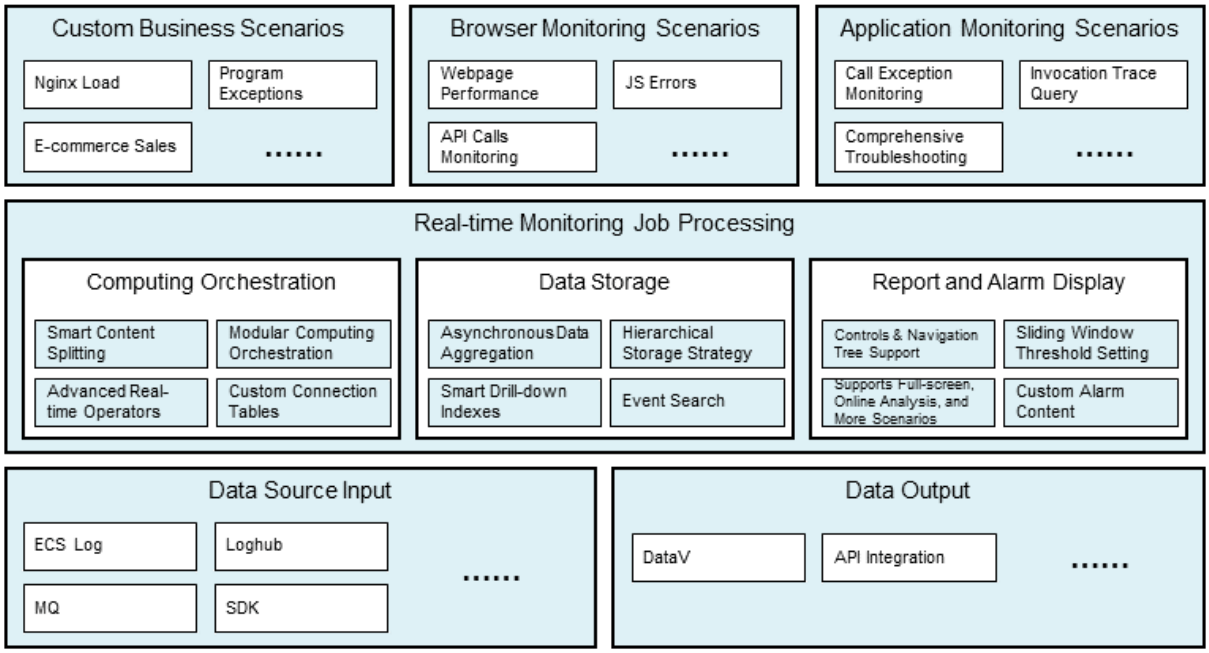
Activate ARMS

Click the button below to go to the buy page of ARMS.

2.Features

Application Real-Time Monitoring Service (ARMS) provides a series of custom monitoring features. These features include data import, computing, storage, visualization, and alerting. ARMS can call and be called by APIs of downstream services.

The following figure shows the ARMS features.



The following section describes these features.

- Multi-dimensional browser monitoring
  - High timeliness: detects the response time and error rate of accessed websites in real time.
  - Multi-dimensional monitoring and analysis: analyzes access speeds and errors by region, carrier, and browser.
  - Page exception monitoring: monitors and diagnoses the performance and success rate of a large number of asynchronous data calls of applications.
- Efficient and easy-to-use application monitoring
  - Application topology self-discovery: generates the call relationship map among distributed applications based on the dynamic analysis and intelligent computing of the remote procedure call (RPC) information.
  - Drill-down metric analysis in common diagnosis scenarios: analyzes metrics such as the application response time, number of requests, and error rate. Then, this feature displays the analysis data by application, transaction, and database.
  - Capturing of abnormal and slow transactions: analyzes timeouts and exceptions based on traces, and associates the transactions with API calls, for example, with specific SQL statements and message queues.
  - Transaction snapshot query: collects trace-based transaction errors and identifies the sources of exceptions or errors by checking detailed data.
- Powerful custom monitoring

- Rich data sources: supports various types of real-time data sources, such as logs, SDK, Message Queue (MQ), and LogHub.
- Flexible real-time computing and storage orchestration: allows you to orchestrate real-time computing and storage modes based on specified dimensions and computing modes.
- Flexible integration with alerts or dashboards: integrates monitoring datasets with ARMS alerts and dashboards. This way, ARMS provides monitoring capabilities for various scenarios.
- Reference scenario templates: provides a large number of reference scenario templates, such as NGINX monitoring, exception monitoring, and e-commerce monitoring.

- ARMS also provides the following features:
  - Flexible real-time computing jobs
    - Supports drag-and-drop modular programming of real-time computing and supports most logic expressions, such as general mathematical operations, regular expressions, and if and else statements.
    - Supports a variety of real-time computing and storage operators. These operators include SUM, COUNT, MAX, MIN, SAMPLE, TOPN, and COUNT DISTINCT.
  - Stable and efficient time series and event storage
    - Aggregates data online in a continuous manner to ensure a controllable data capacity.
    - Supports intelligent hierarchical storage policies.
    - Supports a maximum of three levels of drill-down indexes.
  - Custom alert settings
    - Supports alerts for moving average and maximum values within a time period.
    - Supports custom alert content.
    - Provides multiple alert notification methods, such as email, SMS message, and DingTalk notification.
  - Flexible customization of interactive dashboards
    - Provides rich presentation widgets, such as bar charts, heat maps, pie charts, and ticker boards.
    - Supports dashboard sharing and full screen display.
  - Flexible integration with downstream applications
    - Supports integration with APIs of Java, Python, Perl, and C# applications.
    - Supports integration with other display tools such as DataV.

3.Scenarios

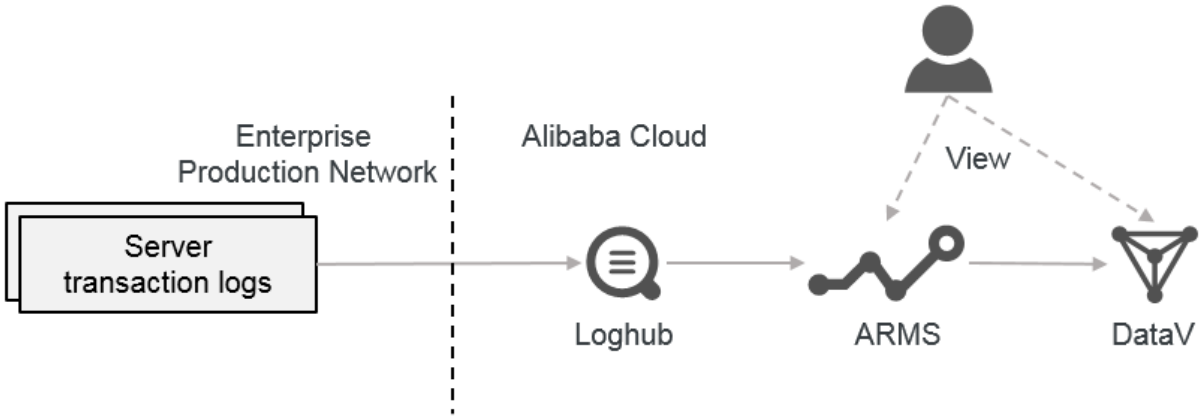
3.1. Real-time monitoring solution in the retail industry

In this use case, a leader in the apparel industry adopted a hybrid cloud solution based on Application Real-Time Monitoring Service (ARMS) to build a real-time monitoring system for its retail industry.

- The original monitoring platform for this customer adopted a traditional commercial Online Analytical Processing (OLAP) database, with considerable license expenditure.
- In addition, it cannot meet business needs in terms of scalability and real-time performance.

ARMS-based retail monitoring solution

The following figure shows the overall architecture.



- Transaction logs are uploaded in real time to LogHub of Alibaba Cloud Log Service by using Logtail.
- The ARMS real-time application monitoring function connects to LogHub for computing and storage, and uses its interactive dashboard to perform real-time analysis and display sales data. The following information is displayed:
  - Computing orchestration and storage: Extracts detailed data of each transaction from logs, including data on the total price and number of items, and then aggregates the data according to multiple dimensions, such as the transaction location, sales company name, and customer membership information.
  - Interactive presentation: Presents the sales status and analysis on various types of cases according to multiple dimensions, such as region, store, member, and product category.
- The data generated by ARMS is delivered to the downstream DataV component for dashboard presentation.

Business value of the ARMS-based monitoring system

- It decreases the total costs of operation (TCO) by hundreds of times, and achieves high-timeliness multidimensional analysis. It can not only help you grasp frontline sales details in real time, but also help you cope with challenges through sales and inventory configuration policies.
- It can also meet the needs in multiple scenarios. The DataV dashboard is used for overall presentation in the monitoring room, and the interactive ARMS dashboard is used for in-depth

troubleshooting.

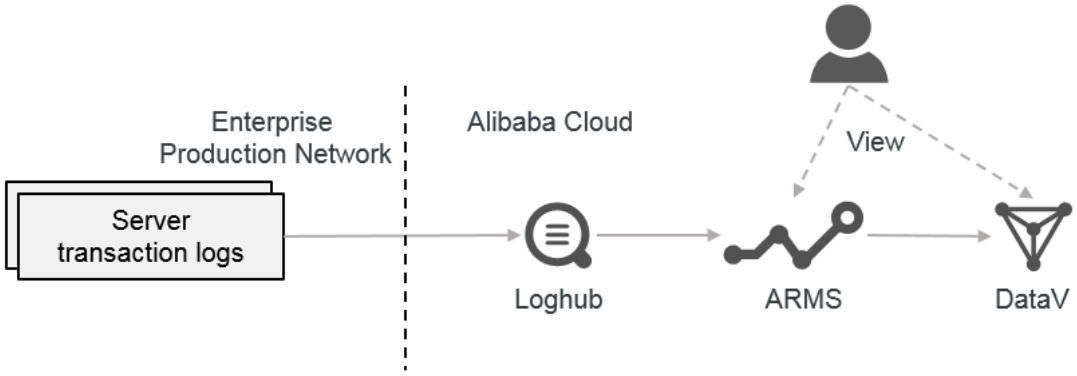
3.2. Real-time IoV monitoring solution

In this scenario, a solution provider in the Internet of Vehicles (IoV) industry from Shanghai adopted a solution based on Application Real-Time Monitoring Service (ARMS). This solution is used to collect statistics about online conditions of vehicles.

- Multidimensional statistics on raw data cannot be implemented based on databases because a large volume of data is involved (approximately 100,000 vehicle data records per second).

ARMS-based IoV monitoring solution

The following figure shows the overall architecture.



- The platform of the solution provider uploads real-time data of new-energy vehicles to Alibaba Cloud by using Message Queue (MQ).
- The application monitoring feature of ARMS works with MQ to obtain the online data of all vehicles and perform real-time statistics and storage.
  - Computing orchestration and storage: ARMS provides real-time statistics of the online rate and failure rate based on the reported vehicle information in multiple dimensions, such as the region, vehicle type, and enterprise. ARMS also stores the statistical results in columns by custom aggregation.
  - Data delivery: ARMS delivers data to downstream applications after the applications call the corresponding API operations.
- Downstream applications of Enterprise Distributed Application Service (EDAS) call API operations to retrieve data from ARMS. The retrieved data is then analyzed and displayed on the user applications.

Business value of the IoV monitoring solution

- The monitoring solution allows you to grasp real-time status of vehicles. The solution collects statistics on fault data in real time and reports statistical results based on different vehicle types. This improves the efficiency of quality enhancement.
- The monitoring solution monitors the status of new-energy vehicles to detect illegal behavior such as subsidy cheating at the earliest opportunity.

3.3. Java application monitoring and diagnosis solution

In this scenario, the application monitoring solution based on Application Real-Time Monitoring Service (ARMS) is used to resolve pain points in monitoring distributed Java applications.

The rapid growth of Internet businesses has produced increasing traffic pressure and more complicated business logic. As a result, traditional single-machine applications can no longer meet the business requirements of enterprises. An increasing number of websites has used the distributed deployment architecture. The basic development frameworks, such as Spring Cloud and Dubbo, is becoming mature. More enterprises have vertically split their website architectures by business module and used the microservice architecture. The microservice architecture is more suitable for collaborative development and fast iterations.

The distributed microservice architecture has advantages in terms of development efficiency. However, this architecture poses huge challenges on traditional monitoring, O&M, and diagnosis technologies. For example, when the distributed microservice architecture was implemented on Taobao (www.taobao.com), Alibaba encountered the following challenges:

- Difficulty in locating issues

  The customer service team submits the purchase issues of consumers to the technical support for troubleshooting. A website request in the distributed microservice architecture passes through multiple services and nodes before a result is returned. If an error occurs, the O&M engineers must check the logs on multiple servers to identify the error cause. In most cases, multiple teams must be work together to troubleshoot a simple issue.

- Difficulty in identifying bottlenecks

  When a website fails to respond to access requests, the O&M engineers cannot identify the bottleneck. This is because the bottleneck can be a network issue between the user terminal and the server, server overloading, or database overloading. Even if the cause is identified, the error in the code is still difficult to identify.

- Difficulty in identifying the architecture

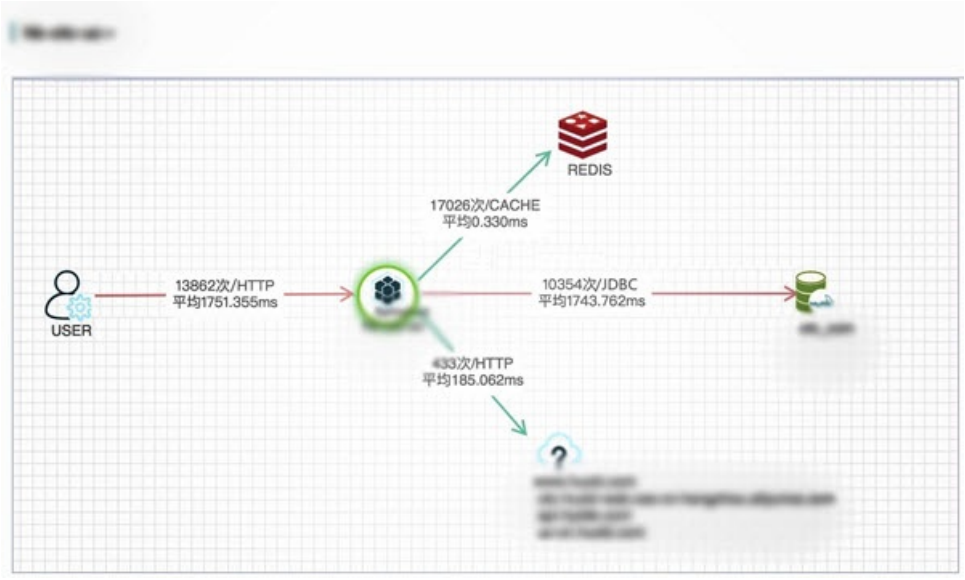
  The business logic has become more complicated. The downstream services (databases, HTTP APIs, or caches) on which an application depends are difficult to identify based on the code. The external calls that depend on the application are also difficult to identify based on the code. In addition, business logic identification, architecture governance, and capacity planning become more difficult for enterprises. For example, when enterprises prepare for Double 11 promotion campaigns, the number of servers required for each application is difficult to predict.

ARMS-based application monitoring solution

ARMS provides the **application monitoring** feature based on the distributed tracing and monitoring system named EagleEye. This feature helps website developers and operators resolve the preceding issues without the need to modify the existing code.

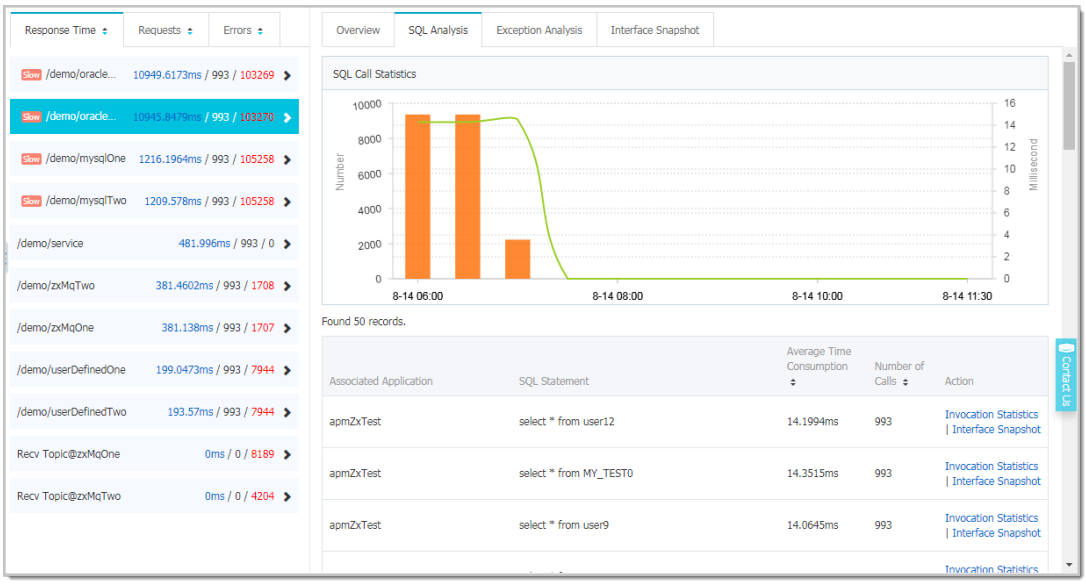
View the call topology

You can view the call topology of an application on ARMS, for example, services that depend on the application and downstream services on which the application depends. In the following figure, a bottleneck occurs when an unknown application calls the monitored application. The average consumed time is more than 3,000 ms.



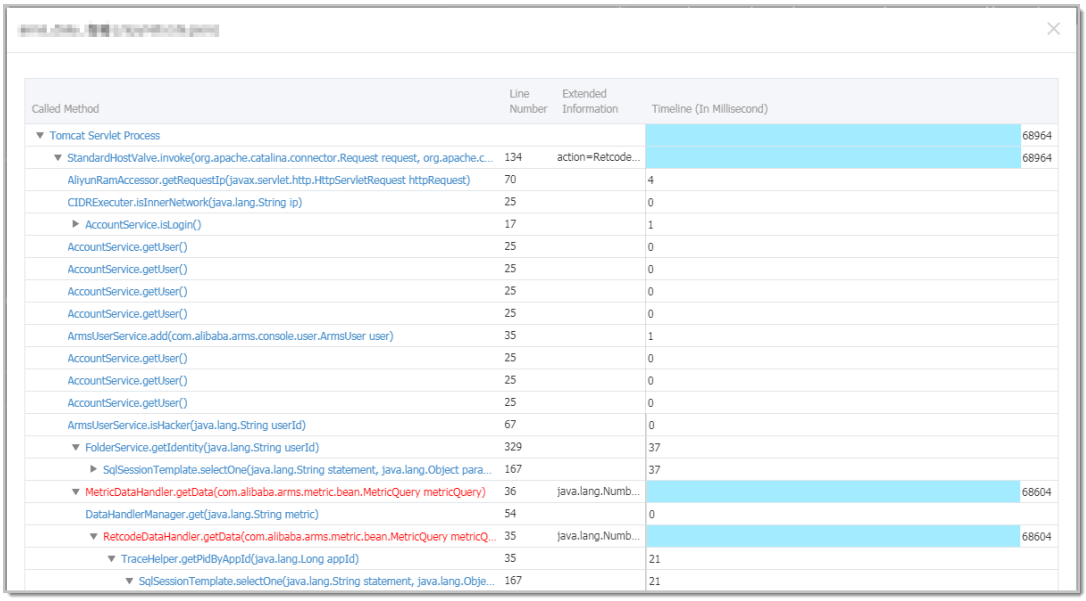
Reports of slow services and slow SQL statements

You can go to the SQL analysis report of an application to locate the slow SQL statements and slow services.



Query distributed trace

You can click the interface snapshot of a slow SQL statement. Then, you can find a request that includes the SQL call, view the call stack of the method, and then identify the code of the issue.



Either from the global perspective or from a single call, ARMS provides a comprehensive solution to address your pain points in distributed Java application monitoring. You can integrate application monitoring with browser monitoring and business transaction. ARMS provides all-around protection for your sites from critical business metrics and user experience to application performance.

3.4. Monitoring scenario of user experience

This topic describes the scenario where user experience is monitored.

When you access Application Real-Time Monitoring Service (ARMS), the whole process can be divided into three phases: page production phase (server status), page loading phase, and page running phase. To ensure stable online services, the server monitors the status of these services. The existing server monitoring system is quite mature, but the monitoring of the page load and run statuses is still unsatisfactory. This is because less emphasis is placed on browser monitoring. Monitoring on the server is mistakenly regarded as a replacement of browser monitoring. As a result, the details of users who access the system cannot be perceived. Therefore, frontend issues encountered by users are difficult to locate.

Challenges

- Difficulty in locating performance bottlenecks

  When the loading speed of a page is slow, the performance bottleneck is difficult to identify. Is the slow page loading speed caused by network issues, resource loading issues, or page Document Object Model (DOM) parsing issues? Are the issues related to the province and country where the user resides, or the user's browser and device? These issues cannot be reproduced and the causes cannot be identified.

- Failure to obtain the details of failed user access requests

  If a large number of JavaScript (JS) errors occur after a system goes online, users cannot use the system as expected. If you cannot obtain the error details at the earliest opportunity, you may lose a large number of users. When a user gives feedback on page usage, you cannot immediately reproduce the scenario of the user. Even if you obtain an error message from a user, you cannot quickly fix the error. These are difficulties that you may encounter.

- Failure to obtain the details of asynchronous API calls

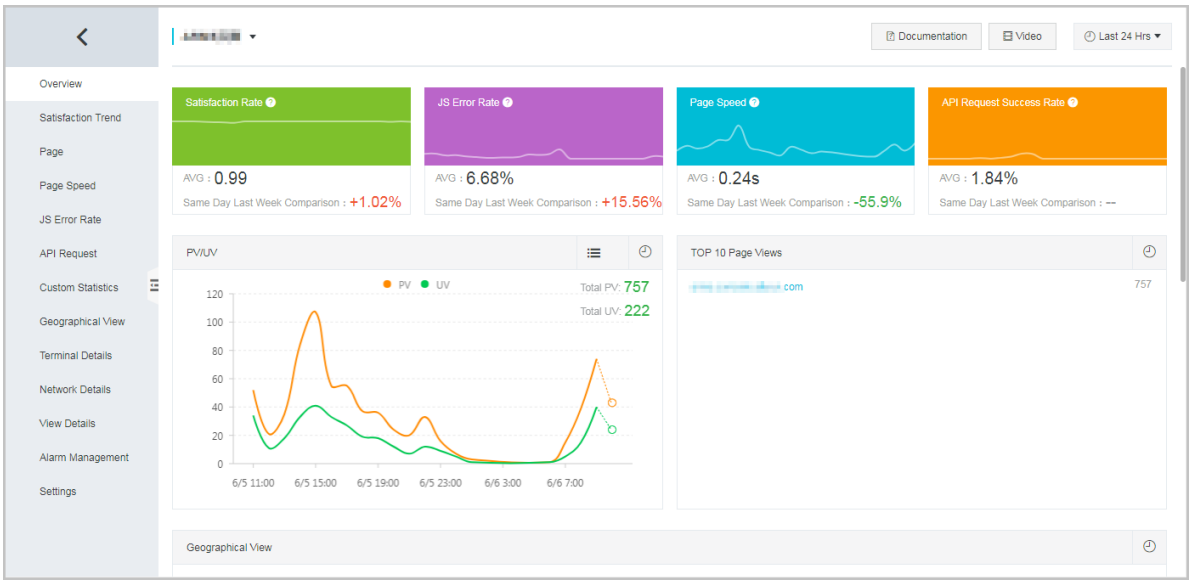
Even if the HTTP status code 200 is returned from an API call, the API may be abnormal. If a business logic exception occurs, can the exception be perceived at the earliest opportunity? If normal responses are returned for all API calls, but the whole process consumes a long time, how can we get the whole picture and reduce the response time? Before you cannot answer these questions, you cannot identify the issues or improve user experience.

ARMS-based browser monitoring solution

Based on large volumes of real-time log analysis and processing provided by ARMS, the browser monitoring feature monitors the access requests from all real online users and resolves the preceding issues.

- Check the application overview page to identify exceptions

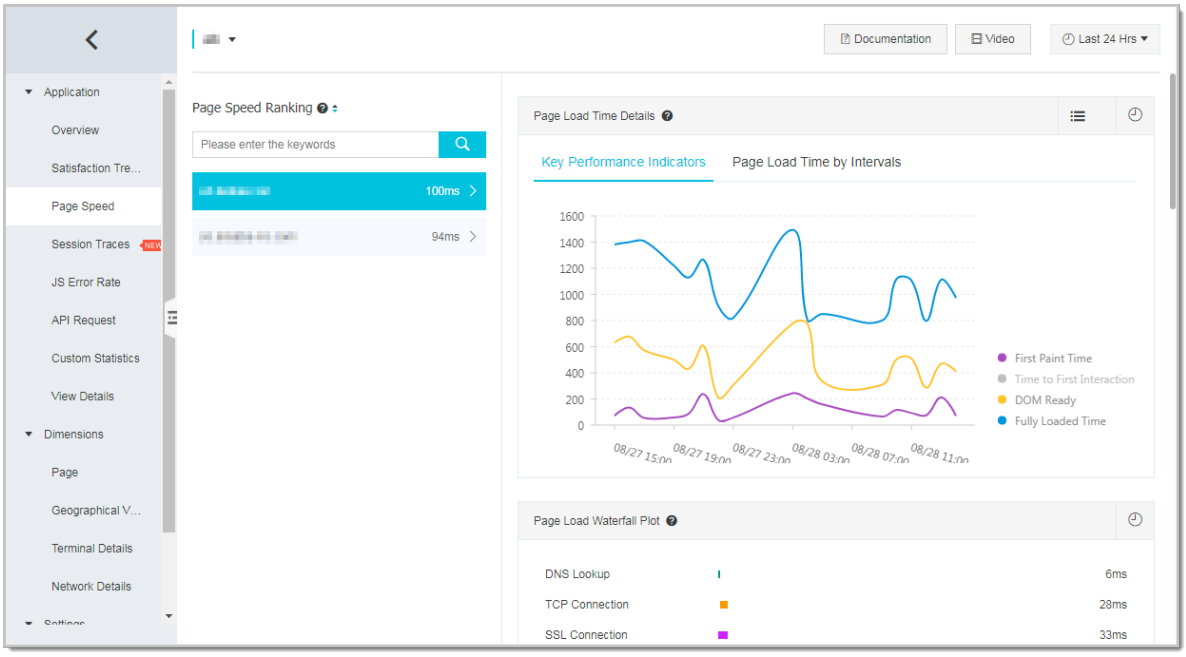
  You can view the information on the application overview page of the browser monitoring feature. The information includes application satisfaction rate, JS error rate, page speed, API success rate, and page view (PV) details. In the following figure, the average JS error rate is 6.68%. The average JS error rate increases 15.56% over the last week.



  Browser monitoring overview

- Performance data trend and waterfall chart

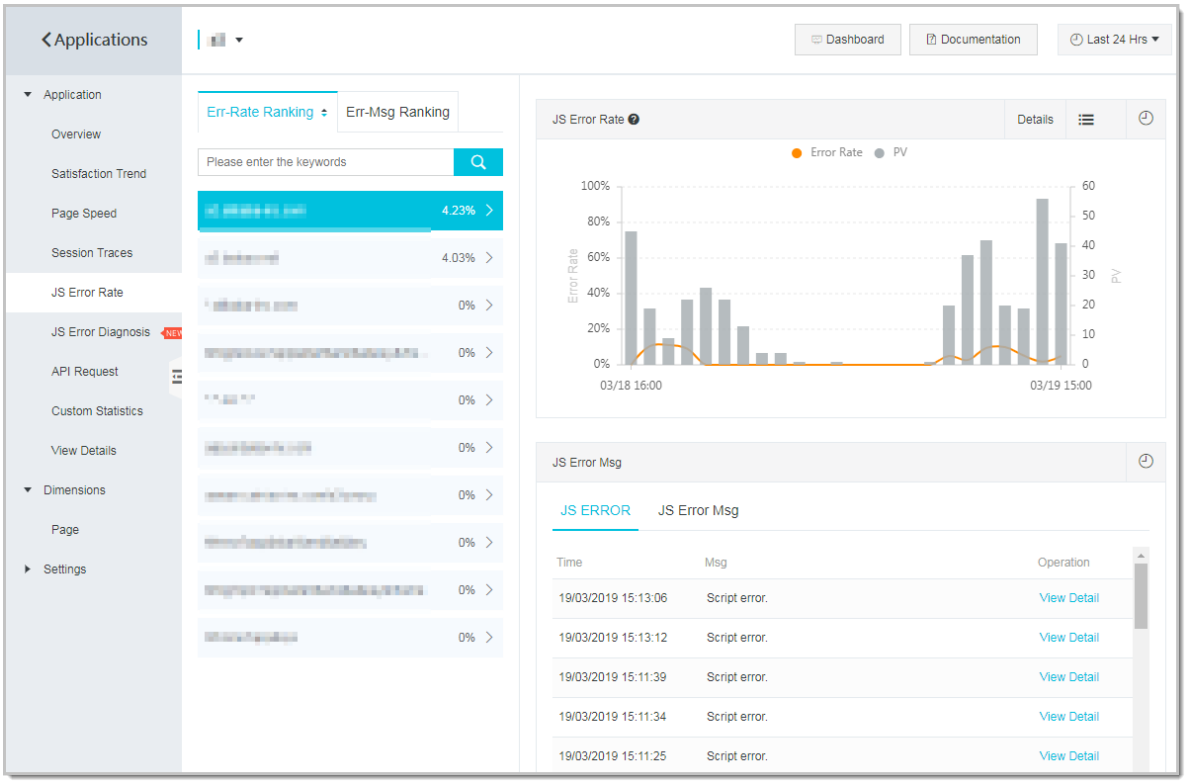
  On the **Page Speed** page, you can view specific metrics related to the page performance and the waterfall chart of page loading. You can then locate the performance bottleneck based on the detailed data.



Performance data trend and waterfall chart

- JS error rate and error aggregation

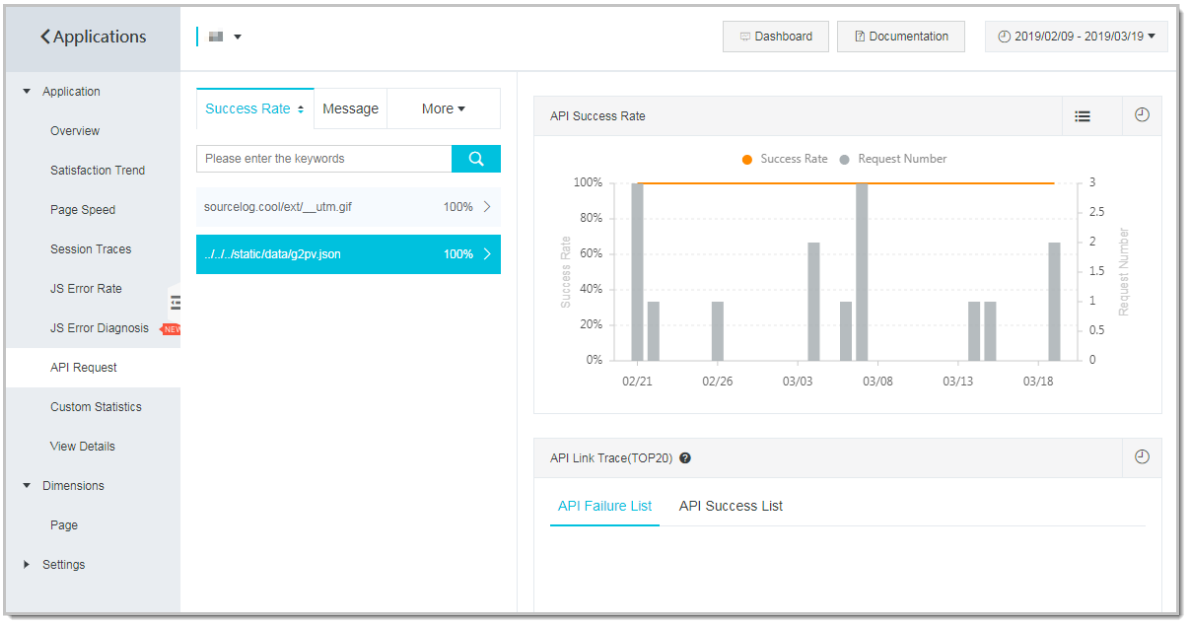
  On the **JS Error Rate** page, you can view a ranking on page error rate and a ranking on frequent errors. Therefore, you can get a straightforward impression about which pages have the highest JS error rate, and which errors are the most frequent.



JS error rate and error aggregation

- API request

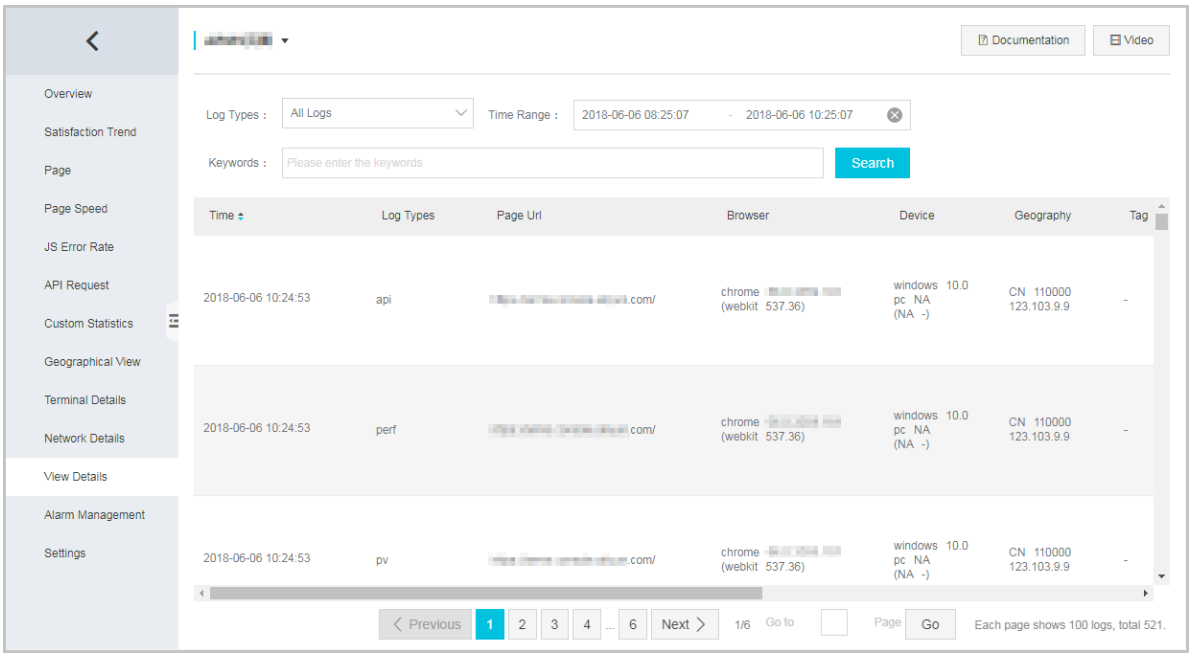
On the **API Request** page, you can view the success rate and response time of API requests. This allows you to obtain the details of API requests.



API request

- Access details

  On the **View Details** page, you can view the details of access requests. For example, you can locate an error based on the error information of stacks, files, lines, and columns.



View details

4.Terms

This topic lists the terms of Application Real-Time Monitoring Service (ARMS).

B | C | J | S | Y | Z

B

B

Alert rule
An alert rule defines how to trigger alerts based on datasets and send alert notifications through a specific channel. The severity of an alert can be Warning, Error, or Critical.

[Back to the top]

C

Collection rule
A collection rule defines how to collect data from data source instances in a custom monitoring job. You must define collection rules for custom monitoring jobs.

[Back to the top]

J

Dashboard
A dashboard is a set of interactive monitoring data reports customized based on datasets in ARMS. The dashboard can display datasets by using different types of charts. The query time range of the dashboard is user-defined.

[Back to the top]

S

Dataset
A dataset defines how to pre-aggregate and persistently store the logs that are collected in monitoring jobs. You can define a dataset directly or indirectly by using the dashboard and alert notification features.

Dataset dimension
A dataset dimension is the key used for aggregation when a dataset is created. It is similar to the GroupBy column name in a database, or an attribute in multidimensional Online Analytical Processing (OLAP). A dataset performs aggregation operations on the real-time data based on the configured dimensions.

Dataset metric
A dataset metric is a specific monitoring metric stored in a dataset, which is typically of the numerical type and similar to a value in multidimensional OLAP. ARMS metrics correspond to values of Count, Max, Sum, and Count Distinct after real-time computing.

Data cleansing
Data cleansing is a process during which operations such as splitting and static join are performed

on custom monitoring logs to convert them into standard key-value (KV) pairs.

Data screening
Data screening is used to filter the data in datasets for dataset calculation. Data that do not meet criteria are filtered out from the dataset.

Data source
A data source is the source of logs obtained by ARMS custom monitoring jobs, including Elastic Compute Service (ECS), Message Queue (MQ), and SDK data sources.

[Back to the top]

Y

Mapping table
A custom static mapping table is used for mapping query results to business attribute fields. For example, you can map the province, city, and district names in query results to postal codes.

Z

Custom monitoring job
A custom monitoring job is a process in which ARMS captures, processes, and stores data, and then presents and exports the results.

[Back to the top]

Links to other terms

- Terms of application monitoring
- Terms of browser monitoring

5.Release notes

This topic provides the release notes of Application Real-Time Monitoring Service (ARMS) and describes feature changes in each release.

V2.7.7.1

Release date: November 3, 2020

New features:

- Synthetic monitoring
  - The recorder V0.1 for synthetic monitoring is added.
  - The synthetic monitoring scripts can be recorded if the Google Chrome browser is used.

V2.7.7

Release date: October 15, 2020

New features:

- Synthetic monitoring
  - Synthetic monitoring V0.1 is added.
  - Scheduled synthetic monitoring is supported for HTTP or HTTPS and networks in multiple cities and for multiple carriers.

V2.7.6.2

Release date: September 15, 2020

New features:

- Dashboards and alerting
  - Dynamic baselines are supported for the thresholds of alerting metrics.

V2.7.5.3

Release date: August 20, 2020

New features:

- Application monitoring
  - Node.js applications can be monitored.

V2.7.4.1

Release date: August 3, 2020

New features:

- Application monitoring
  - AIOps automatic diagnostics supports evaluating business impacts.

V2.7.3

Release date: July 9, 2020

New features:

- Prometheus Monitoring
  - By default, Function Compute can be monitored.

V2.7.2

Release date: June 5, 2020

New features:

- Prometheus Monitoring
  - User-created Kubernetes clusters can be monitored.

V2.7.1

Release date: May 15, 2020

New features:

- Prometheus Monitoring
  - Internal networks can be monitored and inspected based on the blackbox.

V2.7.0

Release date: April 2, 2020

New features:

- Application monitoring
  - The performance issues of traces can be automatically analyzed.

V2.6.5

Release date: March 5, 2020

New features:

- Browser monitoring
  - Multi-dimensional queries of performance metrics are supported.

V2.6.2

Release date: February 14, 2020

New features:

- Application monitoring
  - Business monitoring is supported.
  - The components of Spring Cloud are supported, such as WebFlux and Gateway.

V2.6.1

Release date: February 11, 2020

New features:

- Application monitoring
  - Relevant features, such as obtaining metadata of microservices, are supported.

V2.6.0.2

Release date: January 2, 2020

New features:

- Application monitoring
  - A new version of exception analysis is supported.

Optimized features:

- Application monitoring
  - Bugs of the Thrift plug-in are fixed.

V2.6.0

Release date: December 17, 2019

New features:

- Application monitoring
  - Asynchronous traces are supported.
  - The parameters that are called by Dubbo or HSFProvider can be recorded.

V2.5.9.5

Release date: November 28, 2019

New features:

- Application monitoring
  - The jfinal-undertow plug-in is supported.

Optimized features:

- Application monitoring
  - Several bugs are fixed, including the bug of failing to obtain the thread profiling data of Dubbo.

V2.5.9.3

Release date: November 25, 2019

New features:

- Application monitoring
  - ARMS can be integrated with Tracing Analysis.

V2.5.9

Release date: September 6, 2019

Optimized features:

- Application monitoring
  - The denial-of-service (DoS) vulnerability of Fastjson is fixed.
  - The logic that is used to obtain the IP addresses of network interface controllers is modified.

V2.5.8

Release date: August 2, 2019

New features:

- Application monitoring
  - The dual-state alerting feature is supported. This allows you to configure alert rules for metrics that have only two states: yes and no or being and not being.
  - The plug-ins of Dameng Database, a Chinese company, are supported.

V2.5.7.2

Release date: July 30, 2019

New features:

- Application monitoring
  - Metrics of Java Virtual Machine (JVM) Metaspace are supported.
  - HTTP status codes to be ignored can be customized. By default, HTTP status codes greater than 400 are counted as errors. You can customize HTTP status codes that are greater than 400 but not counted as errors. For more information, see Configure advanced settings.

V2.5.7

Release date: July 11, 2019

Optimized features:

- Application monitoring
  - The Fastjson version that is being dependent on and has security vulnerabilities is upgraded.

V2.5.6.1

Release date: June 28, 2019

New features:

- Application monitoring
  - The Dubbo and MariaDB plug-ins are supported.
  - Custom configurations allow you to obtain the value that is bound by SQL. The variable value that is bound to the PrepareStatement parameter can be captured. The setting takes effect without the need to restart the application. For more information, see Configure advanced settings.

Optimized features:

- Application monitoring
  - Dependency on Log4j is removed to avoid conflicts.

V2.5.6

Release date: June 7, 2019

New features:

- Application monitoring
  - Quantile statistics are supported.

V2.5.5

Release date: June 3, 2019

New features:

- Application monitoring
  - HSF-HTTP calls are supported.

V2.5.3

Release date: March 15, 2019

New features:

- Application monitoring
  - Thread metrics can be reported when applications are running.
  - The Spring-Data-Redis plug-in is supported.
  - The plug-in of the Druid database connection pool is supported.

V2.5.1

Release date: February 1, 2019

New features:

- Application monitoring
  - Applications that are deployed on Container Service for Kubernetes (ACK) are supported. Application monitoring can be enabled in the console or open source environments for the applications that are deployed on ACK. For more information, see Install the ARMS agent for a Java application deployed in Container Service for Kubernetes and Install the ARMS agent for an application deployed in an open source Kubernetes environment.
- Browser monitoring
  - Mini programs can be monitored. Browser monitoring can be enabled for the mini programs on DingTalk, Alipay, and WeChat, and other types of mini programs. For more information, see Monitor DingTalk mini programs, Monitor Alipay mini programs, Monitor WeChat mini programs, and Monitor other mini programs.

V2.5.0

Release date: January 21, 2019

New features:

- Application monitoring
  - The tagging feature is added. Application sits can be classified by tag in the list of monitored applications.

Optimized features:

- Browser monitoring
  - The alerting feature is optimized.

V2.4.8

Release date: January 6, 2019

New features:

- Application monitoring

- The class conflict detection feature is added.
- URL convergence rules can be dynamically enabled and disabled.

V2.4.7

Release date: December 13, 2018

New features:

- Application monitoring
  - Application monitoring can be enabled within a few clicks for Java applications. For more information, see Install the ARMS agent for a Java application by using scripts.
  - Application monitoring can be enabled for PHP applications. For more information, see Install the ARMS agent for a PHP application.

Optimized features:

- Application monitoring
  - The traditional method to enable application monitoring for Java applications is optimized. For more information, see Manually install the ARMS agent for a Java application.
- Browser monitoring
  - The feature of locating JavaScript errors is optimized. In most cases, online JavaScript code is compressed into one line. An error cannot be located based on the error line number that is reported by the browser. Browser monitoring of ARMS can locate errors by using Source Map. For more information, see Diagnose JS errors by using ARMS browser monitoring.

V2.4.6

Release date: October 26, 2018

New features:

- Application monitoring
  - Monitoring can be enabled for NoSQL databases.
  - Brand-new Overview page: The page layout is optimized by adding the application monitoring details, browser monitoring details, and product newsletter modules.

V2.4.5

Release date: September 17, 2018

New features:

- Application monitoring
  - MongoDB is supported. The supported components and frameworks include MongoDB.

V2.4.4

Release date: August 17, 2018

New features:

- Application monitoring:

- Thread profiling is supported. For threads whose requests time out, you can quickly identify the time consumption of all the internal stacks in the threads. For more information, see Analyze errors in code by using ARMS thread profiling.

- Browser monitoring:
  - Resource loading details are supported. You can locate all the slowly loaded resources on the page, such as images, JavaScript, CSS, and APIs. For more information, see Slow session tracing.
- Custom monitoring:
  - Custom monitoring is put into commercial use.

V2.4.3.4

Release date: July 16, 2018

New features:

- Application monitoring:
  - Comprehensive troubleshooting is supported. You can query the distributed traces of application monitoring by business ID. This makes problem diagnostics much more efficient.

V2.4.3.3

Release date: June 16, 2018

Optimized features:

- General:
  - Resource Access Management (RAM) users can call OpenAPI. Security authentication for API calls is further improved. For more information, see Grant different permissions to RAM users.
- Application monitoring:
  - A brand-new revision of the Application Monitoring homepage is provided. The homepage displays more critical summary information.
  - The feature of capturing and analyzing memory snapshots is optimized in different network environments. This improves the capture efficiency by more than 50%.

V2.4.3

Release date: May 19, 2018

New features:

- Application monitoring:
  - The feature of analyzing memory snapshots is added. This provides an uninterrupted view of the distribution of memory objects and helps you quickly locate memory leaks. For more information, see Memory snapshot.
  - The feature of custom configuration of monitoring methods is added. This allows you to dynamically configure specific methods to monitor and capture exceptions, and enables more fine-grained monitoring. The configurations immediately take effect without the need to restart the machine.
  - The Overview page of application monitoring is added. This makes issue troubleshooting and location more convenient and exact.

- Trace monitoring of Message Queue (MQ) is supported. This helps you quickly locate delayed messages, error messages, and stacked messages.

V2.4.2

Release date: April 19, 2018

New features:

- Application monitoring:
  - The JVM monitoring feature is added. This allows you to monitor important metrics of JVM, such as heap memory, non-heap memory, and the number of threads. For more information, see JVM monitoring.
  - The host monitoring feature is added. This allows you to monitor a series of host performance metrics, such as CPU, memory, disks, and network traffic. For more information, see Host monitoring.
  - The custom configuration feature is added. This allows you to modify configurations on the user interface, such as trace sampling, agent switch, and threshold settings. For more information, see Custom configuration.
- Browser monitoring:
  - The sample reporting configuration is supported. You can use random sample reporting to reduce reports from users and workloads. For more information, see SDK reference.

V2.4.1

Release date: March 22, 2018

New features:

- General:
  - Alert rules can be imported and exported in batches. This makes alert management more convenient.
  - Application monitoring for Enterprise Distributed Application Service (EDAS) is put into commercial use, and 50% off is available for long-term use.
- Application monitoring:
  - A list of online Agents can be viewed. The versions and status of the Agents that are installed by a user are clear at a glance.
- Browser monitoring:
  - The response time can be viewed in the form of quantile values rather than average values. This helps you view the time consumption and distribution of slow requests more clearly.
- Application monitoring:
  - Quantile operators are supported. You can view more detailed metric statistics in quantile values, in addition to average, maximum, and minimum values.

V2.4.0

Release date: February 26, 2018

New features:

- The browser monitoring and application monitoring features of ARMS are put into commercial use.

V2.3.3.1

Release date: January 31, 2018

New features:

- Application monitoring of ARMS fully supports EDAS applications. EDAS users can connect EDAS applications to ARMS applications within a few clicks. For more information, see Enable ARMS to monitor an EDAS application.

V2.3.3

Release date: January 14, 2018

New features:

- A brand-new homepage and documentation are provided. Major revisions are made for the product page. The product is officially positioned as a combined Application Performance Management (APM) product that is oriented to **application monitoring**, **browser monitoring**, and **custom monitoring**.
- The base features that are specific to APM are basically reconstructed. This provides users with a unified alert platform and a unified interactive dashboard. The alert platform and the interactive dashboard are oriented to three subproducts: application monitoring, browser monitoring, and custom monitoring.
- Five regions are fully supported: China (Hangzhou), China (Beijing), China (Shanghai), China (Qingdao), and China (Shenzhen). The China (Shanghai), China (Qingdao), and China (Shenzhen) regions are newly supported.

Optimized features:

- The console is centralized. A centralized console is used across all the regions in China.
- The Alert Query interface is optimized. Multi-dimensional alert queries are provided.

V2.3.1

Release date: December 14, 2017

New features:

- The application monitoring feature supports most general APM features of Java application monitoring, such as call topology, link tracing, slow transaction reports, and slow SQL queries. More than 10 types of Java stack frameworks that are required by standard users on the cloud are supported, such as Spring, Redis, MySQL(RDS), and Dubbo. You can connect to applications by mounting javaagent and do not need to modify application code.

Optimized features:

- The method to configure datasets for dashboard widgets is optimized. A mixture of basic and compound indicators are displayed.
- The prompt logic for new users is optimized. By default, the prompt for new users appears only for the first logon.

V2.2.6.2

Release date: September 23, 2017

New features:

- Interactive dashboards support heat maps.

- An exception splitter is added, and supports data cleansing for Java Exception.
- A functional module is added for mappings between IP addresses and physical addresses in the process of data cleansing.
- The NULL type can be used as a filter condition of datasets.

Optimized features:

- Alert content is optimized. The alert content in emails includes log sampling content.
- The NGINX template is optimized. The NGINX monitoring feature that is more straightforward and easier to use is released.

V2.2.6

Release date: August 31, 2017

New features:

- The browser monitoring feature that is specific to quality and performance monitoring is released.
- MQ data sources can be used for business monitoring.

V2.2.5

Release date: July 26, 2017

New features:

- MQ data sources can be connected to ARMS data sources.
- Queries of millions of data records in a dataset are supported.
- Alerts of the same type can be displayed in an aggregated way. This improves the display effect.
- The share links of interactive dashboards can be created and connected to. This allows users to view the dashboards without logon.
- Light and dark themes are added to interactive dashboards. This makes the interactive dashboards more aesthetic.
- Interactive table properties are optimized. When you browse a table, you can pin the table header, and sort data in reverse chronological order.

V2.2.4

Release date: June 21, 2017

New features:

- General datasets are supported.
- Frontend visualization components are fully revised and changed to G2. A single widget supports multiple datasets.
- External links of ARMS interactive dashboards are supported.
- On-the-fly editing of dataset alerts is supported.
- The API supports queries of millions of data records.
- XML logs can be split by line breaks in data cleansing.

V2.2.3

Release date: March 30, 2017

New features:

- The Pop API of datasets is supported in Python.
- The widgets of interactive dashboards are improved. The stacking mode is available for area charts.
- The China (Beijing) region is available.

Optimized features:

- The alerting feature is overall optimized. The time to take effect, the alert severity, and the alerting method can be flexibly specified for a single alert.
- API query results can be associated with dimension tables for output.

V2.2.2

Release date: March 8, 2017

New features:

- The dimension table feature is supported. This allows you to customize the mapping relationships of attributes. For example, you can map the ZIP code of a city to the province, city, and district of a specific city.
- A standard task template is released. This allows you to quickly customize monitoring tasks based on the standard task template.
- The alert recovery feature is added. When an alert is recovered, a notification is sent by email.

Optimized features:

- Interactive dashboards continue to be optimized. TopN filtering and zero filling are added, and more time granularities are supported.

V2.2.1

Release date: February 17, 2017

New features:

- More rate operators are supported. Rate operators apply to scenarios, such as statistics of rate changes.
- RAM authorization rules are supported.

Optimized features:

- Response time for real-time computing of ARMS is significantly reduced. In some scenarios, response time is shortened from more than 10 seconds to less than 5 seconds.
- Interactive dashboards continue to be optimized. Finer granularity is supported for the zooming modes of various windows.

V2.2.0

Release date: January 23, 2017

New features:

- Interactive dashboards are released.
- Tasks can be copied, imported, and exported. This helps you quickly copy existing monitoring plans.
- Self-check is supported for tasks of users, including error statistics and error sampling.
- Compound operators are supported.

V2.1.0

Release date: October 8, 2016

New features:

- Log Service LogHub, SDKs, and APIs can be used as data sources.
- The alerting feature supports various general advanced operators. You can specify the past N minutes, days, and hours, and can set the MAX, AVG, and MIN thresholds of metrics.
- The advanced operators Count Distinct and Sample are supported.
- An interactive search interface is supported for datasets.

V2.0.0

Release date: August 4, 2016

New features:

- The Elastic Compute Service (ECS) data that is collected in the form of logs can be connected.
- Various types of cleansing and computing are supported: single, multiple, and sequential delimiters, KV cleansing, JSON cleansing, and cleansing based on other various custom logic, such as exception stacks.
- Multiple types of aggregation calculations are supported: all the general aggregation calculations based on various time granularities, such as SUM, COUNT, and MAX.
- Business alerting settings support the definitions of various content metrics, severity differentiation, and various notification delivery methods for contacts.
- A set of solutions based on time series or other similar dimensions are provided for you to customize display charts. General display forms, such as column charts, line charts, pie charts, ticker boards, and tables, and dashboard configurations are integrated. Data drilling up and down are supported.
- The defined alerts and charts can be dragged and dropped to customize dashboards.

6.Updates

6.1. ARMS Prometheus Service allows you to monitor databases by using web browsers

An update of Application Real-Time Monitoring Service (ARMS) Prometheus Monitoring is released on March 20, 2020. The new version allows you to connect Redis, MySQL, Elasticsearch, and MangoDB databases to ARMS Prometheus Monitoring and then monitor these databases with a few clicks. Out-of-the-box dashboards are also provided.

> **Note** You can use ARMS Prometheus Monitoring free of charge for 15 days. Click to start your free trail. You can join the DingTalk group for free trial users: 23155358.

Overview of ARMS Prometheus Monitoring

ARMS Prometheus Monitoring provides one-stop, out-of-the-box Prometheus monitoring services that have the following features:

- Simplified and stable Prometheus collectors: Prometheus collectors are deployed on your cluster by using Helm charts. Based on simplified and optimized Prometheus servers, Prometheus collectors focus on service detection and data collection. The resource consumption of a Prometheus collector is reduced to 5% of the resource consumption of a Prometheus server. Prometheus collectors support horizontal scaling.
- Stable and large-capacity cloud storage: Cloud storage is based on Alibaba Cloud Log Service and provides unlimited distributed storage space. This ensures high efficiency and stability.
- Efficient and powerful data query in the cloud: Prometheus servers in the cloud support metric query that is fully compatible with PromQL. In addition, Prometheus servers allows you to sample and speed up queries.
- Default alerting service: ARMS Prometheus Monitoring supports native alert rules and provides multiple alert notification methods by default, such as emails and DingTalk. Alert notifications can be sent to relevant contacts by using the cloud monitoring service.
- Out-of-the-box monitoring dashboards for Grafana Cloud: By default, ARMS Prometheus Monitoring provides monitoring dashboards for Kubernetes clusters. This allows you to monitor hosts, pods, deployments, API servers, Ingress, and coreDNS. ARMS Prometheus Monitoring also provides monitoring dashboards for various third-party components.

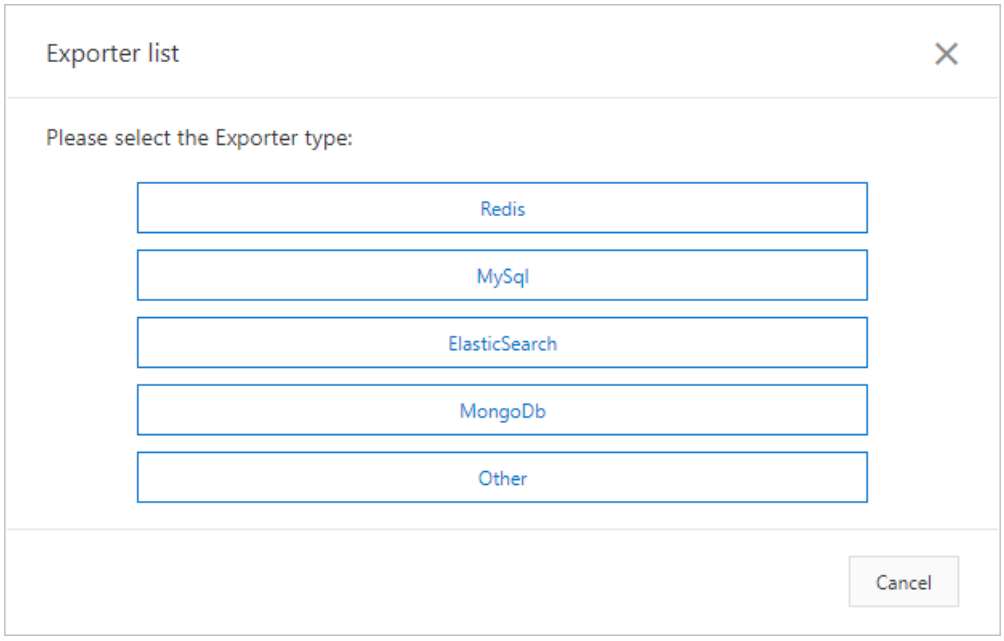
For more information about ARMS Prometheus Monitoring, see Prometheus monitoring overview.

One-click database monitoring

ARMS Prometheus Monitoring supports one-click access to monitor mainstream databases. To view your database monitoring dashboard, perform the following steps:

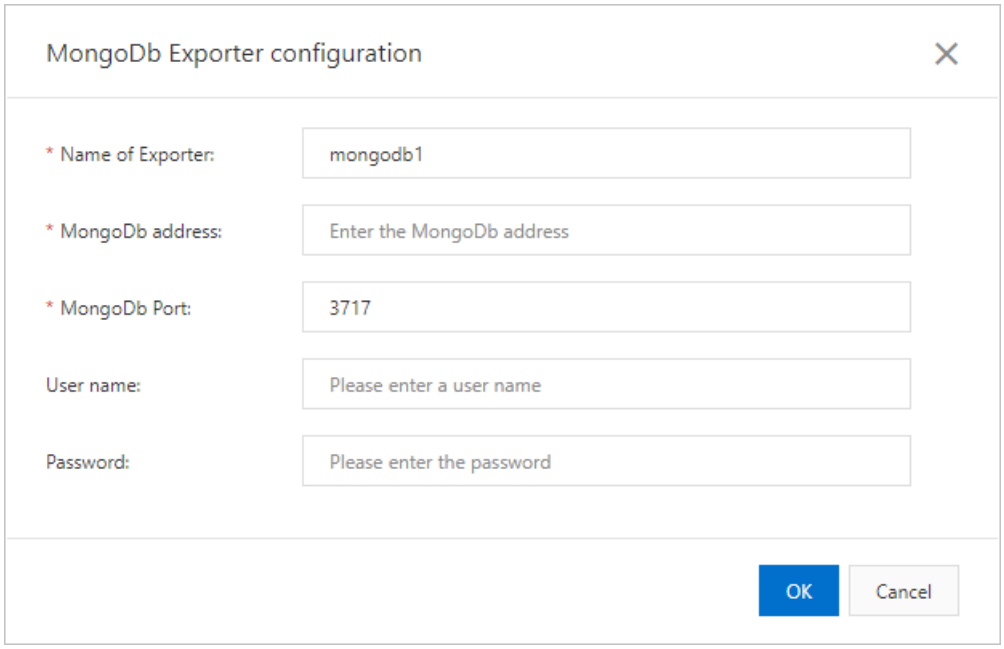
1. Select the type of the database that you want to monitor.

   Only Redis, MySQL, Elasticsearch, and MangoDB databases are supported.

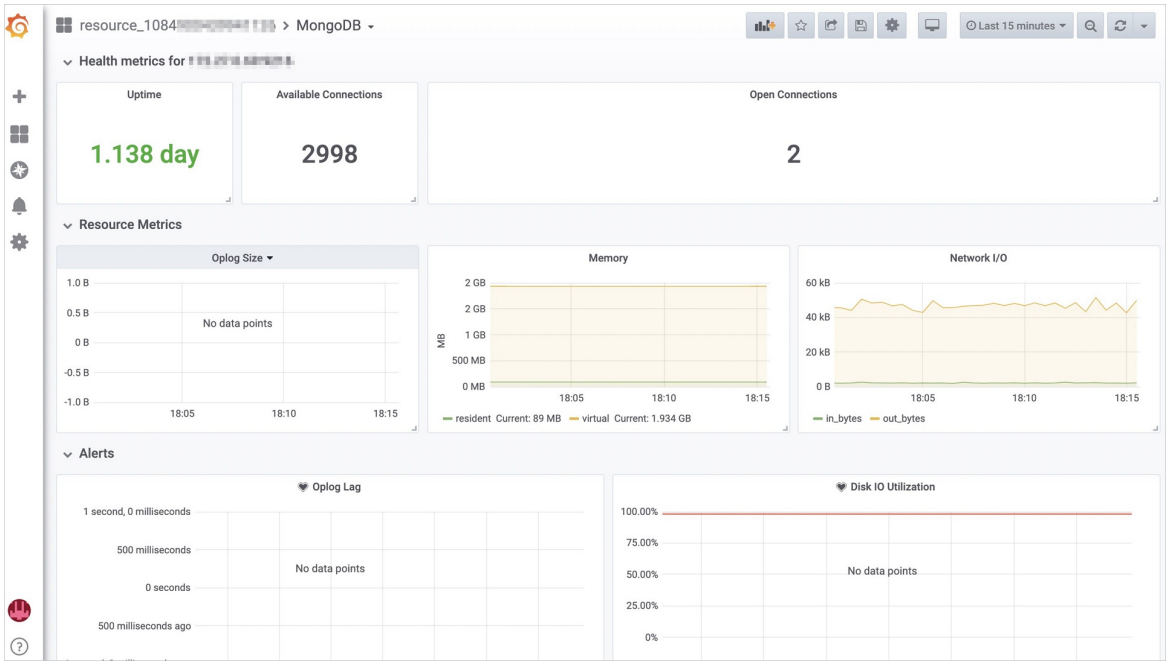


2. Set the parameters for the database that you want to monitor.

   Take MangoDB as an example. Enter the exporter name, connect string, port number, username, and password of the MangoDB database.



3. Find your new exporter and view the monitoring dashboard.



ARMS Prometheus Monitoring enables you to configure exporters through web browsers, and provides ready-to-use monitoring dashboards. ARMS Prometheus Monitoring supports most mainstream database components. More application components will be supported in later versions. Stay tuned for further updates.

Related information

- Prometheus monitoring overview

7.Differences between ARMS and Tracing Analysis

Both Application Real-Time Monitoring Service (ARMS) and Tracing Analysis can solve tracing problems in a distributed environment. This topic describes the differences between ARMS and Tracing Analysis.

Background information

ARMS is an application performance management (APM) service provided by Alibaba Cloud. ARMS allows you to build application monitoring capabilities that feature response time within several seconds for enterprises from custom dimensions such as the browser, application, and business.

Tracing Analysis provides developers with various features for distributed application development, including trace mapping, request counting, trace discovery, and application dependency analytics. Tracing Analysis helps you analyze and diagnose performance bottlenecks in a distributed application architecture. It also makes microservice development and diagnostics efficient, and reduces the costs of developing tracing applications such as Jaeger and Zipkin and related storage services such as HBase and Elasticsearch.

Comparison between ARMS and Tracing Analysis

Difference	ARMS	Tracing Analysis
Service positioning	ARMS is an APM service that supports application performance monitoring, user experience monitoring, tracing, and problem diagnostics.	Tracing Analysis is designed for distributed tracing.
Access method	Non-intrusive agent loading	Intrusive SDK programming
Billing method	Billed based on the number of agents, with competitive pricing	Billed based on the number of requests, with competitive pricing
Programming language	Java and PHP	Languages such as Java, PHP, Go, Python, JavaScript, .NET, and C++
Thread and memory diagnostics	Supported	Not supported
Local method stack	Supported	Not supported

The following sections compare ARMS and Tracing Analysis in terms of the service positioning, access method, and billing method.

Service positioning

ARMS is positioned as a heavyweight APM service that has rich features. You can install the ARMS agent for an application to access ARMS. The ARMS agent provides features such as performance monitoring, user experience monitoring, tracing, and fault diagnostics.

Tracing Analysis is designed to solve tracing problems in a distributed environment. You can implement distributed tracing by using Tracing Analysis SDK, which supports only trace monitoring.

Access method

Positioned as an APM service, ARMS can be accessed in non-intrusive mode, which is popular among commercial APM tools. By using this mode, you do not need to modify the application code. You need only to install an agent for the application that you want to monitor, and modify the application startup mode. For example, when you use ARMS to monitor a Java application, you must add the -javaagent startup parameter during the application startup.

Positioned as a distributed tracing service, Tracing Analysis is based on open source products such as Jaeger and Zipkin, and the OpenTracing open source standard. You can access Tracing Analysis by using an SDK based on Jaeger, Zipkin, or OpenTracing. This method brings the following benefits:

- You can seamlessly migrate applications that use SDKs based on Jaeger, Zipkin, or OpenTracing to Tracing Analysis, without code modification.
- You do not need to worry about vendor lock-in because Tracing Analysis SDK is based on an open source standard.
- The community allows you to support a large number of programming languages at a time. This significantly reduces the threshold for access to trace monitoring when you develop applications in a heterogeneous environment.

Billing method

Similar to other APM services, ARMS is billed based on the number of agents when you use features such as application monitoring and browser monitoring. The resulting fee may account for a large proportion of your budget. Based on a high-performance and efficient architecture, ARMS is billed at a rate that is only 10% to 20% of the industry average.

Designed for trace diagnostics in a distributed environment, Tracing Analysis is billed based on the number of requests, at a low rate.

More information

Though ARMS and Tracing Analysis are differently positioned, both of them are monitoring services for development in Alibaba Cloud and will be interconnected in the future.

- ARMS will support custom distributed tracing based on Tracing Analysis SDK.
- Tracing Analysis will support queries in the ARMS console to improve your diagnostic experience.

Related information

-
- Tracing Analysis