

Alibaba Cloud Application Configuration Management

Product Introduction

Issue: 20191014

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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent

ial, exemplary, incidental, 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 contact Alibaba Cloud directly if you discover any errors in this document

.

Document conventions

Style	Description	Example
	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.
	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.
	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.
	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 <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
{} or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Contents

Legal disclaimer.....	I
Document conventions.....	I
1 What is ACM.....	1
2 Function overview.....	4
3 Technical architecture.....	6
4 Product comparison.....	8
5 Use cases.....	10
6 Terms.....	16
7 Release note.....	19

1 What is ACM

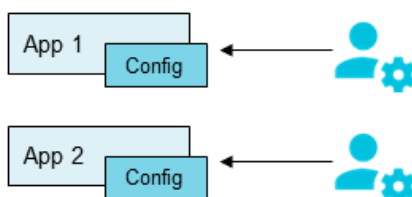
Application Configuration Management (ACM), formerly known as Taobao's internal configuration center Diamond, is an application configuration center that enables you to centralize the management of application configurations, and accomplish real-time configuration push in a distributed environment. With ACM, you can greatly reduce the workload of configuration management and enhance service capabilities in scenarios such as microservices, DevOps, and big data.

In application lifecycle management, developers usually extract some of the configuration items or metadata of the application from the code and manage them in a separate configuration file. These separately managed content is called application configuration, one of the common ways to manage application changes. After the application is published, the maintenance personnel or the end user can change the configuration to adjust application behaviors and adapt to environment changes.

ACM serves as the configuration center in a distributed system. It offers a series of functions such as configuration modification, configuration push, historical version management, gray release, and configuration modification audit. With these features, ACM helps you centralize the management of configurations in every application, reduce the cost of configuration management in distributed systems, and lower the risks of availability issues or even failures caused by incorrect configuration changes.

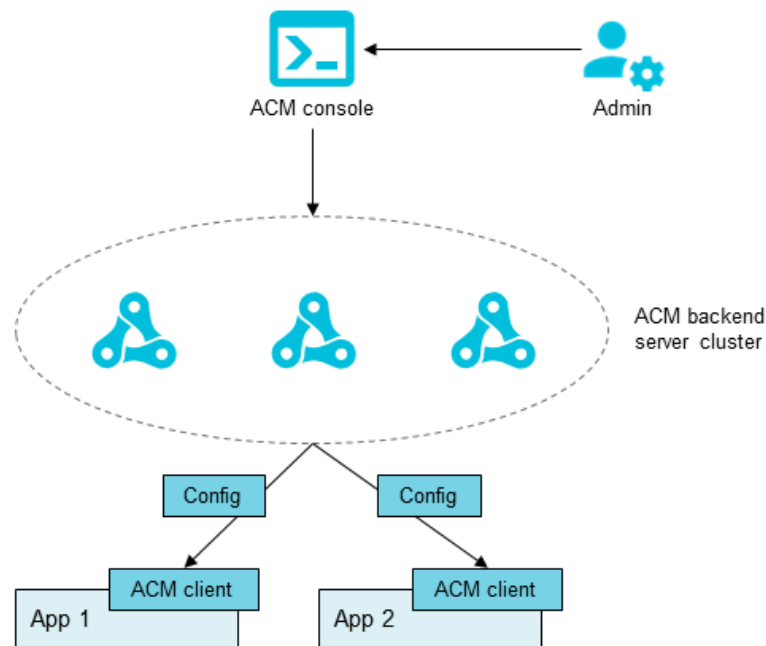
Configuration management in traditional architecture

In the traditional architecture, for any configuration changes, you often need to log on the specific server and manually modify the configurations for them to take effect, as shown in the following figure.



Configuration management with ACM

With ACM, you only need to change the configuration in ACM console and the configuration information is automatically pushed to each server and takes effect with a latency of mere seconds. ACM mainly consists of three components: the client, the server, and the console for configuration management.



Why ACM

Managing configurations using ACM brings the following benefits to IT operations and maintenance:

- Updated configurations are automatically delivered to each machine in seconds, which greatly reduces the workload of manual configuration distribution;
- By using the ACM configuration listening API, configurations on each application can take effect immediately without restarting the application;
- Information about configuration listening, changes, and versions is automatically recorded, enhancing the capabilities in terms of audit, version management, and diagnosis.

Learning Path

With the [ACM Learning Path](#), you can quickly get to know how to use the basic configuration management functions of ACM, and one-click rollback, push tracks,

namespaces, access control, and more advanced features. You can also understand how to meet your specific needs with a range of APIs and SDKs.

2 Function overview

This topic summarizes ACM' s functionalities.

- **Core functionality/feature**
 - **Creating, deleting, modifying, and querying configurations: the core functionalities**
 - **Importing and exporting configurations: makes it easier for you to synchronize your configurations across multiple environments**
 - **Batch operation: allows you to manage in batch**
 - **Configuration description: helps others understand what the configuration is for**
 - **Configuration tags: organizing configurations with tags helps you manage configurations more efficiently**
 - **Editing files of a variety of formats online: supports syntax highlighting and staging validation, which improves online editing efficiency and reduces editing mistakes**
 - **Multiple languages support: supports Java/CPP/Python/Shell/HTTP OpenAPI**
- **Advanced feature**
 - **View changes with diff tool: view all changes at a glance, which helps reducing the risk of misoperation**
 - **Gray release: controls the risk of changing configurations**
 - **Change history: makes everything auditable**
 - **One-click rollback: reduces the risk of changing configurations**
 - **Push tracks: improves the efficiency of troubleshooting**
 - **Namespaces: isolate the daily, staging, and production environments**
- **Stability**
 - **Multi-level cache: disaster recovery for host and backup database storage, full amount cache for server, and SDK cache**
 - **Throttling: prevents the impact on cluster stability from misuse**
 - **Capacity management: prevents the impact on cluster stability from misuse**
 - **City-wide disaster recovery: data center-level disaster recovery**

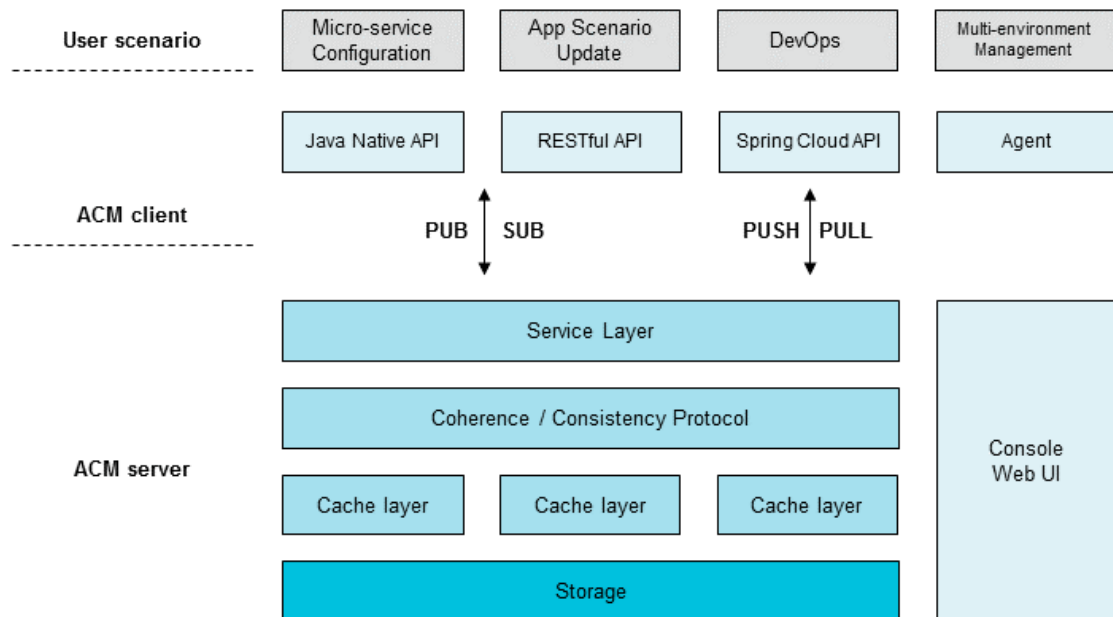
- **Performance/capacity**
 - **The capability of managing millions of configurations: effectively adapts to explosive growth of business**
 - **Supports millions of connections: effectively adapts to explosive growth of business**
 - **Push hundreds of thousands of configurations concurrently in seconds: effectively adapts to explosive growth of business**
- **Security**
 - **HTTPS: enhances security**
 - **Authentication: prevents the data from impact from other users**
 - **Data Encryption: enhances the security of sensitive data**

3 Technical architecture

This topic explains the technical architecture of ACM.

The architecture of ACM is as follows.

Figure 3-1: ACM technical architecture



ACM server

An ACM distributed service node contains the following:

- **Service protocol layer:** performs protocol conversion and authentication.
- **Consistency management layer:** performs configuration consistency management and configuration push.
- **Configuration cache layer:** enhances configuration query and push efficiency with distributed cache.
 - **Storage layer:** refers to backend distributed storage, which is used to store configurations and features high performance and high scalability.
 - **Console:** refers to the ACM console for configuration management.

ACM client

Based on RESTful API, ACM client enables cross-language access. ACM provides Java Native APIs and configuration read APIs based on Spring Cloud Config, so as to simplify your development process.

In some cases, ACM provides an agent for dynamic replacement of the configuration files on the host machine. In this case, you must manually specify the mapping relationship between ACM configuration items and configuration files.

4 Product comparison

This document compares several mainstream application configuration management products.

Similar Products

- ZooKeeper

ZooKeeper is an open-source implementation of Google Chubby, which provides distributed application coordination service. ZooKeeper provides consistency management services for distributed applications, including configuration maintenance, domain name services, distributed synchronization, group services, and so on. In scenarios such as a Hadoop cluster, ZooKeeper also performs application configuration management. However, as a CP (Consistency, Partition Tolerance) application, its availability and performance can be affected.

- etcd

Like ZooKeeper, *etcd* is a key value storage system with high availability designed for configuration sharing and service discovery. *etcd* is developed and maintained by CoreOS, inspired by ZooKeeper and Doozer. It is written in Go and processes log replication using the Raft consistency algorithm to ensure strong consistency. Similar to ZooKeeper, *etcd* can also be used for application configuration management. However, as a management application aimed to ensure strong consistency, its availability and performance can be compromised in some scenarios.

- Spring Cloud Config Server

Similar to ACM, *Spring Cloud Config Server* provides external configuration support for both the server and the client in distributed systems. The config server provides a centralized configuration service for every application in any environment. Yet unlike ACM, the Spring Cloud Config Server uses Git for configuration information storage by default, and its features like configuration storage, version management, and configuration publishing are all based on Git or other peripheral systems. In addition, ACM and Spring Cloud Config are quite different in terms of configuration management functionalities.

Product comparison

The following table details the differences among ACM, etcd, ZooKeeper, and Spring Cloud Config in terms of application configuration management.

Product	ACM	Spring Cloud Config Server	ZooKeeper	etcd
Configuration modification	Done on ACM console directly	Done on Git repository	Done by calling the ZK API	Done by calling the etcd API
Automatic configuration push	Modified configurations are automatically pushed to listening clients	The client can only load the configurations when it is started	Modified configurations are automatically pushed to the listening client	Modified configurations are automatically pushed to the listening client
API	Based on RESTful API, and supports Java Native API, Spring Cloud API, and API of other languages	Based on RESTful API and Spring Cloud specifications , and supports clients using other languages	Supports Java Native API	Based on the RESTful API
Version Management	Automatic version recording of each change	Indirect version management by Git	Without any version control	Without any version control
Configure push Tracing	Supports queries of configuration push status and tracks for all clients	Unable to query configuration push history	Unable to query configuration push history	Unable to query configuration push history



Note:

Unlike the CP applications such as etcd and ZK, ACM is not positioned for strict transactional configuration services such as distributed locks.

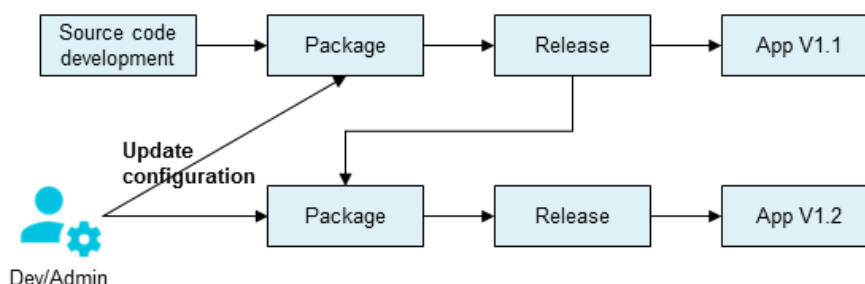
5 Use cases

As one of the most widely used middleware products, ACM has been adopted extensively to manage configurations of Alibaba's internal applications since 2008. ACM has a wide range of use cases in many core scenarios. This topic describes a few typical scenarios where ACM is used.

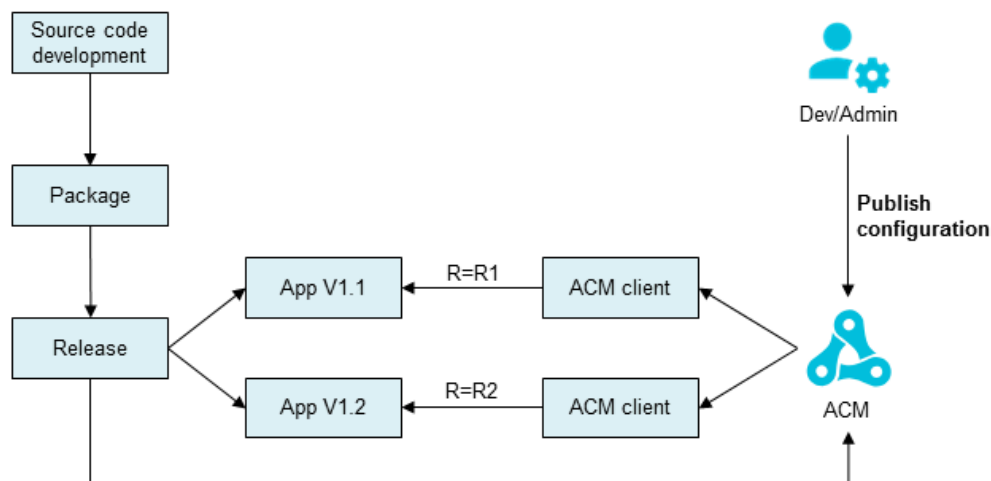
Configuration management under the microservice application architecture

Under the microservice architecture, configuration management (such as DB_URL access information, service connection pool, and internal cache size of services) becomes cumbersome as the number of applications and machines increases . In this case, configuration distribution across multiple machines in a single application, and application-to-application configuration dependency can be great challenges.

In traditional architecture, the entire application must be re-packaged and published again even though only one configuration item is modified. This process is complex and error-prone, as shown in the following figure.



In an ACM-based microservice scenario, important application configuration information is published to ACM. The release of new configurations does not require configuration packaging. The applications take effect immediately after the new configurations are released, as shown in the following figure.



The use of ACM as a configuration center brings microservices the following benefits:

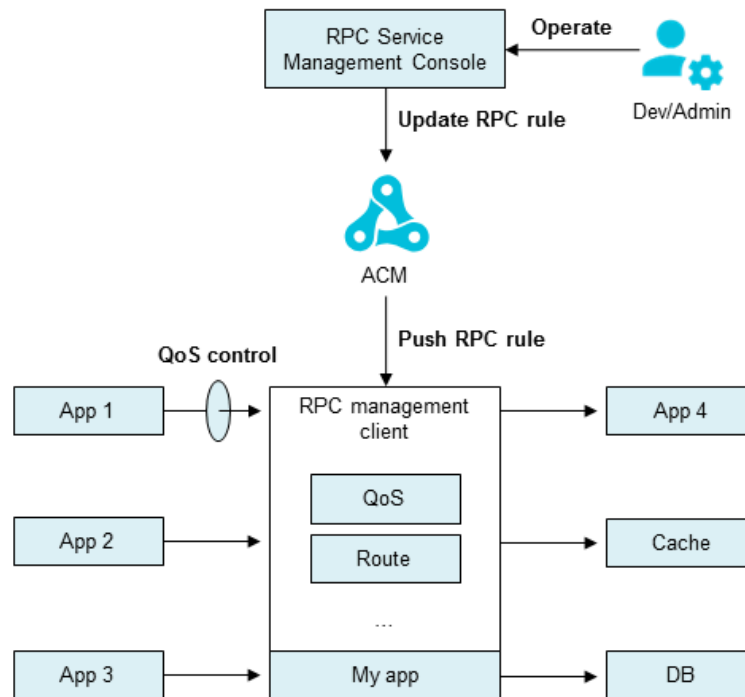
- All configurations are centralized, which makes it easier to manage the configurations of large amount of applications.
- Configuration publishing is not dependent on version updates, making it more flexible to modify configurations.
- ACM supports gray release and rollback, improving the security of configuration updates in a microservice architecture.

Service governance under the distributed architecture

Under various distributed architectures, it is critical to optimize service governance based on a certain RPC framework (such as RESTful, HSF, and Dubbo). Service administration functions including service routing, rate limiting, service downgrading, and service authentication can all be done using the configuration center.

Take request throttling and service degradation as an example. During Alibaba's Double 11 events, each operation related to request throttling or service degradation requires response within seconds. This can be achieved using ACM.

In this process, the server of each RPC listens for the service rate limiting information by registering listeners through ACM. When an application requires rate throttling, the administrator performs request control in the service governance console. Then, the service governance system pushes the throttling information through ACM to the target application server to enable the corresponding configuration to take appropriate throttling action.



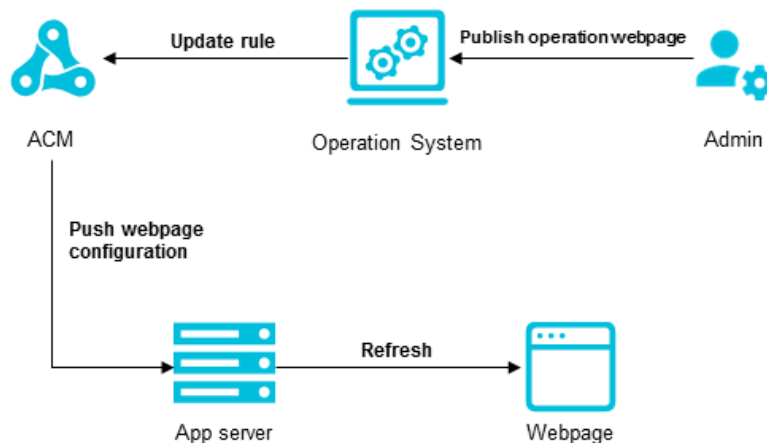
ACM brings the following benefits to service governance under distributed architectures:

- **Good performance.** Listens for service governance information through configuration push without compromising performance.
- **Quick response.** Service governance information can be pushed in seconds.
- **Quick error correction.** When service rate throttling or degradation information is pushed incorrectly, configuration can be rolled back in seconds.

Dynamic configuration push in business scenarios

Other typical use cases of ACM include speeding up the rollout of web pages in marketing activities, so as to reduce development costs and improve marketing efficiency.

Take e-commerce operations as an example. By embedding the ACM configuration (such as the third-party library version number and the static resource URL) in the frontend Javascript, operation staff can modify ACM configuration rules using operation tools to bring frontend Javascript presentation into effect when running promotional activities.



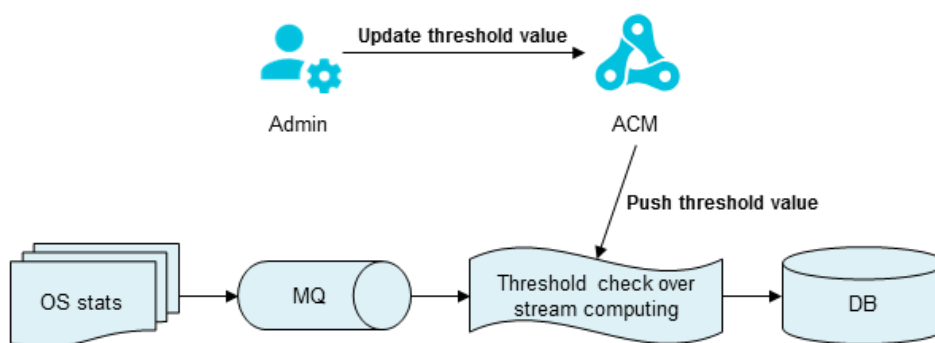
ACM brings the following benefit to configuration push in business scenarios:

- Decoupling static business code from business scenarios using configurations to greatly optimize the operation-related application publishing process.

Algorithm adjustment for real-time big data computing

In real-time big data computing, calculation parameters are adjusted dynamically to obtain the most accurate real-time calculation results.

Take an APM monitoring system in Alibaba as an example. The monitoring system dynamically adjusts the threshold values of businesses to control the real-time computing system and create business alarms. The threshold value modification must be completed in real time without application downtime. The calculated threshold values of the monitoring system are pushed under ACM rules.



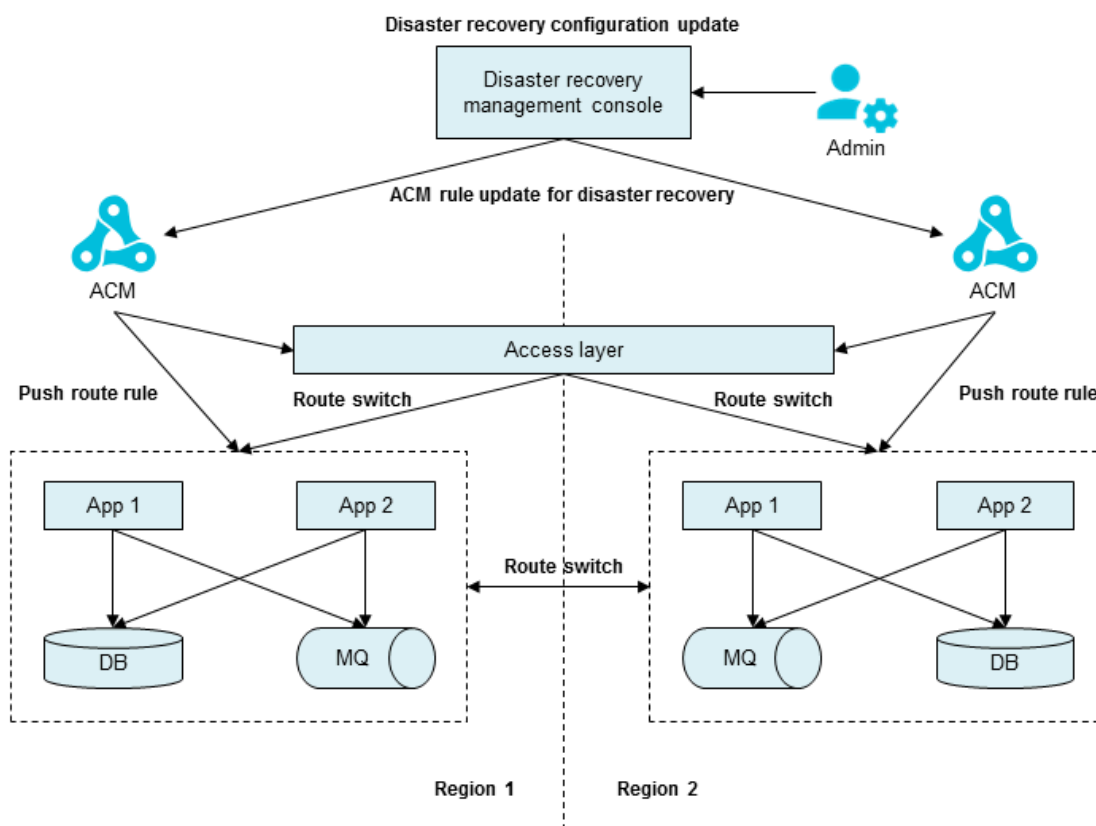
ACM brings the following benefit to real-time big data computing scenarios:

- Dynamic configuration of calculation parameters, which can take effect quickly with little impact on performance.

Multi-Site High Availability architecture in enterprise Internet architectures

Multi-Site High Availability solution is an advanced disaster recovery architecture in enterprise Internet architectures. Compared with the traditional disaster recovery architecture, it features quick business recovery, low capacity requirements, and effective and simple maintenance. Currently, the Multi-Site High Availability architecture has been widely used in companies like Alibaba and Ele.me.

At Alibaba, the core algorithms, ID shards and relevant routing rules of the Multi-Site High Availability architecture are all pushed by ACM dynamically. The related clients and servers, such as RPC, MQ, and DB are embedded with the routing path. During a disaster recovery drill test or when a real disaster occurs, the administrator only needs to dynamically push the rules, and these rules will have an effect on every architecture component. This is shown in the following figure.



The use of ACM brings the applications in the remote Multi-Site High Availability architecture the following benefits:

- **The infrastructure and the disaster recovery logic are decoupled. The specific routing logic is determined by the disaster recovery switching rules.**

- **Theoretically, disaster recovery switching rules can be pushed to hundreds of thousands of machines and take effect in seconds.**

6 Terms

Configuration

During system development, developers usually extract some parameters or variables that need to be changed from the code and manage them in a separate configuration file. This enables the static system artifacts or deliverables (such as WAR and JAR packages) to fit with the physical operating environment in a better way. Configuration management is generally a part of the system deployment process, which is completed by the system administrator or maintenance personnel. Configuration modification is one of the most effective methods to adjust the behavior of a running system.

Dynamic configuration and static configuration

System configuration can be static or dynamic. Static configurations means that configurations are defined when the system is built and deployed and cannot be changed unless there is a version upgrade of the system. Dynamic configurations, however, are those that can be changed from time to time while the system is running. For example, configuration of build-version: 1.0.0 is bounded with the software version and is a static configuration; the configuration of the thread pool size can be changed multiple times when the system is running, and thus is a dynamic configuration.

Configuration management

In the data center, all configuration-related activities such as editing, storage, distribution, change management, history version management, and change audit are collectively referred to as configuration management.

Configuration push

During configuration management, the configuration management system often needs to distribute the configuration changes to relevant systems, and the process from configuration distribution to configuration validation is called configuration push.

Push track

The entire track from configuration change, configuration push to configuration validation is called push track. By looking at the push track of a configuration, you

can see on which applications or machines, or at what time a configuration change is made and what impact it has.

Configuration listening

Configuration listening means that the ACM allows the system to register the listener by SDK to listen for and consume the configuration changes.

Configuration item

It is a specific configurable parameter with its value range, usually in the form of param-key=param-value. For example, the log output level (logLevel=INFO|WARN|ERROR) of a system is regarded as a configuration item.

Configuration set

A collection of related or unrelated configuration items is called a configuration set. Usually a configuration file in the system is a configuration set which contains configurations of all aspects of the system. For example, a configuration set may contain configuration items such as data sources, thread pools, and log levels.

Data ID

The ID of a configuration set in ACM. It is one of the dimensions according to which configurations are organized. Data ID is generally used to organize the system configuration sets. A system or application can contain multiple configuration sets, each of which can be identified by a meaningful name. The Data ID usually uses the naming rule similar to Java packages (for example, com.taobao.tc.refund.log.level) to ensure global uniqueness. This naming rule is not mandatory.

Group

The group of configuration sets in ACM. It is one of the dimensions according to which configurations are organized. The configuration sets are always grouped by a meaningful string such as Buy or Trade to differentiate the configuration sets with the same Data ID. When you create a configuration on ACM, the group name is replaced by DEFAULT_GROUP by default if not specified. A typical scenario of Group is when the same configuration type is used for different applications or components, such as database_url configuration and MQ_topic configuration.

Namespace

Namespace in ACM is used for the isolation of configurations by tenants. Different namespaces may have configurations with the same Group or Data ID. One of the

common scenarios of namespace is to differentiate and isolate configurations in different environments, such as development and test environment or production environment.

Configuration snapshot

The ACM client SDK can generate snapshots of configurations on local machines. Snapshots can be used to indicate the overall disaster recovery capabilities of the system when the client cannot connect to the ACM server. Configuration snapshot is similar to local commit in Git, or cache, which is updated at the appropriate time , but does not have the notion of expiration as in cache.

7 Release note

V4.5.0

Released date: 2018.06.22

New features:

- **Supports fine-granularity authorization with RAM: makes configuration management more secure by effectively avoiding configuration misoperations or data leakage.** [[Related documents](#)]

V4.4.0

Released date:2018.05.18

New features:

- **Support of ECS instance RAM roles: allows applications to access ACM configurations on ECS without setting any sensitive information (for example AK/SK), which solves cloud application security compliance issues.** [[Related documents](#)]

Optimization and improvements:

- **Fixed the compatibility issue with EDAS.**

V4.3.0

Released date: 2018.04.08

New features:

- **Supports encrypted configuration: by integrating with KMS (Key Management Service), you can encrypt your configurations on ACM to make sure they are secure.**
- **Supports RAM (Resource Access Management) policies: with RAM, you can authorize sub-accounts, service accounts, and other primary accounts to read your configurations.**
- **Supports capacity management policies: based on capacity management policies of fine granularity, you can now write your own configurations directly with API or SDK.**

- **Supports importing, exporting, and cloning of configurations: you can conveniently export your configurations from one region and namespace to a specified region and namespace.**

Optimization and improvements:

- **Fixed the compatibility issue with EDAS.**

V4.2.0

Released date: 2018.02.01

New features:

- **Supports adding tags and descriptions for configurations: makes it easier for you to manage configurations your own way and get more information about them at a glance.**
- **Enhanced configuration editing capability: supports online editing of YAML, HTML, and other formats, and supports saving configuration formats.**
- **Provides OpenAPI and supports Shell language: now you can implement multi-language extension with ease or retrieve configurations in DevOps scenarios.**
- **Supports retrieving the configuration list under a tenant.**
- **Supports both Chinese and English.**
- **Supports downloading standalone version.**

Optimization and improvements:

- **Optimized namespace components.**
- **Solved conflicts with EDAS.**
- **Reduced the size of SDK to be sent to the public warehouse, and reduced the JAR conflicts.**

V4.1.0

Released date:2017.12.10

New features:

- **Client supports Node.js and C++.**
- **Supports internationalization, including Chinese and English versions.**
- **Supports deleting namespaces and viewing their details.**
- **Supports Group suggestion.**
- **Shares tenant data with EDAS.**

Optimization and improvements:

- **Optimized environment components, and solved the problem of occasional failure to asynchronously load user namespaces.**
- **The configuration management page now displays the namespace IDs, allowing you to quickly identify the namespaces.**
- **The namespace management displays namespace ID column by default, allowing you to view the relationship between the namespaces and the IDs conveniently.**
- **Solved the problem that the updated configuration overwrites the ownership application field.**
- **Added sample code for Node.js and CPP.**

V4.0.0

Released date:2017.10.10**New features:**

- **Supports release, modification, deletion, and gray release of configurations.**
- **Supports query of the configuration's historical versions.**
- **Supports query of the configuration push track.**
- **Supports query of the configuration listening.**
- **Client supports Java Native SDK and Java Spring Cloud SDK.**