Enterprise Distributed Application Service (EDAS)

Product Introduction

Product Introduction

What is EDAS

Enterprise Distributed Application Service (EDAS) is a PaaS platform involving applications and microservices. It provides a variety of application release capabilities and lightweight microservice solutions that help you solve monitoring, diagnosis, and high-availability O&M issues in your application and service management processes.

Fully utilizing the existing resource and service management systems of Alibaba Cloud, EDAS provides functions such as distributed service frameworks, service governance, centralized configuration management, distributed link tracing, high availability, and data-driven operations.The features have been rigorously tested by core E-commerce platforms at Alibaba Group for many years.

EDAS allows you to easily build the microservice architecture, publish and manage applications, digitally analyze the application monitoring status, and construct distributed systems.EDAS assists you in IT system transformation to meet increasing business needs.

Application publishing and management

Application release and management can become complicated in cloud environments. You need to log on to each server to deploy and publish locally-developed applications. You may also need to add, delete, modify, query, scale up, and scale down the applications in the whole lifecycle. More and more O&M personnel must be allocated for a growing number of servers, bringing huge costs for and many obstacles to business development of your company.

For this scenario, EDAS provides a visual console that allows you to easily publish applications and perform full-lifecycle management on the console regardless of the size of the cluster.

Digital analysis of application monitoring status

After developing and deploying an application in the production environment, you often need to monitor the running status of the application, including CPU usage, instance workload, memory usage, and network traffic.However, this type of basic monitoring cannot meet all business needs. For example, you can neither identify the bottleneck when the system runs slow nor troubleshoot when an error occurs upon opening a page.

To address these challenges, EDAS provides a series of digital operation components, which allow you to precisely monitor and trace every single component or service in the distributed system and pinpoint the bottleneck quickly.

Distributed system

The transformation from a centralized system to a distributed system brings many challenges to distributed systems, for example, consistency dependency, middleware O&M, and troubleshooting.

EDAS supports multiple mainstream RPC frameworks and their extension, such as Dubbo, Spring Cloud, and HSF. It has been proven to have outstanding performance and stability by the internal systems and key events of Alibaba, such as Double 11 Shopping Festival. EDAS allows you to focus on business development by considering various technical details, such as distributed service discovery, service routing, service calling, and service security.In addition, EDAS provides high-availability and low-cost middleware services, freeing you from the tedious work of middleware O&M, and saving your time and labor costs.EDAS also provides a sound full-link tracing and monitoring solution that allows you to quickly locate the problem of your application, reducing the loss therefrom.

Closed-loop R&D and O&M system

With the rising of Internet+, the integrated innovation of "technology + industry" has become the core strategy for industrial transformation. The finance + technology mode makes Zhong An Insurance, Tianhong Asset Management, MYbank, and other innovative financial enterprises become the benchmark for industrial transformation. To better support business innovation, how to construct the shared business middle offices of enterprises and how to support the efficiency improvement of R&D projects in the Internet architecture become two core challenges of enterprises.

EDAS for DevOps is a leading all-in-one R&D efficiency platform in the industry, which enables 24h integration and delivery with project management processes and automated efficiency improvement tools.

Learn more

As a key product of the Alibaba distributed service architecture, EDAS provides a great variety of features ranging from application lifecycle management to service maintenance management. For feature descriptions of EDAS, see Features.

For EDAS-related terminology, see Terminology.

Features

As a key product of the Alibaba distributed service architecture, EDAS provides a great variety of features ranging from application lifecycle management to service maintenance management.

Rich distributed services

RPC framework runtime environment

Since 2007, in response to the demand for massive transformation of the Alibaba e-commerce platform to a distributed architecture, Alibaba has developed the distributed service frameworks HSF (High Speed Service Framework) and Dubbo.

Dubbo is a distributed service framework that provides high-performance and transparent RPC remote service calls. Dubbo is the core framework of Alibaba's SOA service governance solution. Each day, over 2,000 services support over 3 billion access requests.

Built on a high-performance network communication framework, HSF is a distributed service framework for enterprise-level Internet architectures, providing various proven features such as service publishing, registration, calling, routing, authentication, service throttling, service degradation, and distributed tracing.

Spring Cloud running environment

EDAS provides a complete runtime environment for Dubbo and Spring Cloud.

Application-centric middleware PaaS platform

Basic management and maintenance of applications

As the basic management unit in EDAS, an application usually involves multiple instances. On the EDAS console, you can perform full-lifecycle application management, including releasing, starting, stopping, scaling up, scaling down, and deleting applications. Leveraging Alibaba's rich experience in operating and managing massive clusters, you can easily operate and maintain applications on thousands of instances. Also, you can group ECS instances and publish applications in batches based on the specified group. You can set the wait time between different batches of applications to determine whether or not to proceed with releasing of the remaining batches. Also, you can view publishing logs in real time.

Diverse application publishing modes

On EDAS, applications can be published using WAR or JAR packages, images, and other

modes.

Auto scaling

The auto scaling capability detects the status (including CPU, memory, and load status) of each server in the cluster and scales up or scales down in real time based on the detected status. This helps ensure the service quality and improve the availability of the cluster system.

Primary account and subaccount

EDAS provides a unique primary and subaccount system for enterprise-level users.This allows you to build primary and sub-account relationships on the EDAS platform based on your enterprise’ s setup at department, team, and project levels. At the same time, ECS resources are organized according to these primary and sub-account relationships, so that you can easily allocate resources.

Namespace

Namespace is a resource isolation service provided by EDAS. Different namespaces are logically isolated from each other by nature.Namespaces help you completely isolate resources in different environments from each other, and you can use one account to manage them in unified mode.

Roles and permissions

Application lifecycle management generally involves R&D, O&M, instance resources, and other roles. Different roles are permitted to perform different application management operations.EDAS provides a role and permission control mechanism that allows you to define roles and assign permissions for different accounts.

Support for deployment of applications in container service Kubernetes cluster

Kubernetes

Kubernetes is an open source system for automatically deploying, scaling up, scaling down, and managing containerized applications.

Kubernetes clusters

Kubernetes clusters are used to run Kubernetes applications. They are private clusters

invisible to other users, which ensures more secure isolation between your container applications and other users' applications.

Kubernetes applications

Kubernetes applications are containerized applications deployed by Kubernetes. They are divided into stateful applications and stateless applications. The current version only supports stateless applications. A stateless application can have multiple pods to ensure high availability.

O&M management and service governance

Service authentication

The high-speed service framework (HSF) is designed to ensure the reliability and security of each distributed service call.Strict authentication is implemented in every phase, from service registration and subscription to service call.

Service throttling

EDAS allows you to configure throttling rules for the services provided by each application, ensuring service stability.Throttling rules can be configured based on QPS and threads to ensure the system's best operation stability during traffic peaks.

Service downgrade

Contrary to service throttling, service degradation pinpoints and blocks poor services that your application calls. This feature ensures the stable operation of your application and prevents the functionality of your application from being compromised by dependency on poor services.EDAS allows you to configure downgrade rules based on the response time, preventing your application from depending on poor services during traffic peaks.

Automated stress testing

Our original automated stress testing tool makes performance stress testing part of your daily routine.All the traffic for automated stress testing is the actual data in the product environment. By controlling the weights of each service, you can perform online stress testing without degrading the stability.

Comprehensive monitoring and digital operations

Distributed link tracing

EDAS EagleEye provides analysis of every service call, message sending, and database access occurred within the distributed system to help you identify system bottlenecks and risks precisely.

Service call monitoring

EDAS can fully monitor the service calls made by your application in terms of the QPS, response time, and error rate of your services.

Tenant-level service reports

Tenant-level service reports sort all services within the current tenant based on various metrics in tenant dimension, including service call volume in 24 hours, average time for calling services, and call error rate. They give a clear comparison among all services in the system.

IaaS basic monitoring

EDAS can thoroughly monitor the running status of your application in terms of basic metrics, such as CPU, memory, workload, network, and disk.

Fully compatible with Apache Tomcat containers

Ali-Tomcat

Ali-Tomcat is a container that EDAS relies on to run services. It integrates service publishing, subscription, service call tracing, and other core functions. You can publish applications in this container in both development and runtime environments.

Pandora

Pandora is a lightweight isolation container, that is, taobao-hsf.sar. It is used to isolate dependency between web applications and middleware products and between middleware products so that they do not affect each other. Plug-ins implementing service discovery, configuration push, call link tracing, and other functions are integrated in EDAS Pandora.By using these plug-ins, you can monitor, process, track, analyze, maintain, and manage services of EDAS applications in all dimensions.

EDAS Container

EDAS Container is the basic container for running applications on EDAS. EDAS Container includes Ali-Tomcat and Pandora.

EDAS Container integrates the Alibaba middleware technology stack to greatly enhance the startup, monitoring, stability, and performance of containers.Also, EDAS Container is fully compatible with Apache Tomcat.

Terminology

This document defines and explains the special names and terms related to EDAS, so that you can better understand EDAS and how to use it.

EDAS Container

EDAS Container is the basic container for running applications on EDAS.EDAS Container includes Ali-Tomcat and Pandora.

Ali-Tomcat

Ali-Tomcat is a container that EDAS relies on to run services. It integrates service publishing, subscription, service call tracing, and other core functions. You can publish applications in this container in both development and runtime environments.

Pandora

Pandora is a lightweight isolation container, that is, taobao-hsf.sar.It is used to isolate dependency between web applications and middleware products and between middleware products so that they do not affect each other.Plug-ins implementing service discovery, configuration push, call link tracing, and other functions are integrated in EDAS Pandora.By using these plug-ins, you can monitor, process, track, analyze, maintain, and manage services of EDAS applications in all dimensions.

Pandora Boot

Derived from Pandora, the more lightweight Pandora Boot can directly start a Pandora environment in IDE based on Pandora and FatJar, greatly improving the development and commissioning efficiency. It deeply integrates with Spring Boot AutoConfigure to provide your the convenience of the Spring Boot framework.

Dubbo

Dubbo is a distributed service framework that provides high-performance and transparent RPC remote service calls. Dubbo is the core framework of Alibaba’s SOA service governance solution. Each day, over 2,000 services support over 3 billion access requests. This solution is widely used in various member sites of Alibaba Group.

EDAS

EDAS is the core product in Alibaba enterprise-level Internet architecture solutions. With the many proven distributed products developed by Alibaba’s middleware team as its core basic components, this product provides a high-availability distributed solution for the enterprise-level cloud computing market.EDAS makes full use of Alibaba Cloud’s resource management and service systems and introduces the entire sophisticated distributed computing product of the middleware team. It helps enterprise-level customers easily build large-scale distributed application service systems.

EDAS Agent

EDAS Agent is the Daemon program installed on ECS instances to implement communication between the EDAS cluster and the applications deployed on the corresponding ECS instances. It is used for application management, status reporting, and information retrieval. It also serves as the communication channel between the EDAS console and your applications.

EDAS RPC service

The EDAS RPC service provides support for the Dubbo framework.This application was developed using the Dubbo framework and deployed using the WAR method. It can seamlessly perform application deployment and management, and uses all the service governance and data operations functions provided by EDAS.

EDAS application life cycle

As the basic management unit in EDAS, an application usually involves multiple instances.EDAS provides a comprehensive application life cycle management mechanism, covering the entire process from application publishing to operation, including application creation, deployment, startup, rollback, resizing, shutdown, and removal.

FatJar

FatJar (also known as executable JARs) is an archive of compiled classes and dependency JARs for running the code. The application can be run with the Java -jar command.

Auto scaling

The auto scaling capability detects the status (including CPU, RT, and load status) of each server in the cluster and scales up or scales down in real time based on the deleted status. This helps ensure the service quality and improve the availability of the cluster system.

Billing account

The billing account is a primary account used to buy the EDAS product. One billing account can be bound to a maximum of five primary accounts.

HSF

Built on a high-performance network communication framework, HSF is a distributed service framework for enterprise-level Internet architectures, providing various features such as service publishing, registration, calling, routing, authentication, service throttling, degradation, and call link

tracing.

Basic monitoring

EDAS collects data from the ECS instance on which the application is running, to monitor the instance based on basic metrics, such as CPU, memory, load, network, and disk size.Data in all monitoring views is collected and processed in the unit of application.

Configuration push

Configuration push is a function provided by EDAS to centrally manage distributed system configuration information on the EDAS Console. It can add, modify, and delete configurations in real time and can push configuration updates globally or within a specific application.

Application diagnosis

EDAS provides detailed troubleshooting and performance analysis of applications, which collects statistics and method tracing of the single instance on which the application is running from a range of dimensions, including JVM heap memory, non-heap memory, class loader, thread, and Tomcat connector.

EDAS EagleEye Monitoring System

The EDAS EagleEye monitoring system tracks and analyzes the service calls, sent messages, and database accesses of the distributed system to help you precisely identify the bottlenecks and risks of the system.

Instance grouping

This function puts all ECS instances for an application in a group so that you can deploy different application package versions for the ECS instances in different groups.The instance grouping is a feature on EDAS designed to manage instances for an application by group. By grouping the instances for the application, you can maintain it through beta release, A/B testing, and gated release. You can quickly improve your maintenance efficiency by performing application lifecycle management, and resource monitoring and alerting by group.

Application monitoring

Application monitoring accurately reflects the real-time traffic and history information of an application, allowing you to monitor the application health status and quickly discover and locate problems.

Kubernetes

Kubernetes is an open source system for automatically deploying, scaling up, scaling down, and managing containerized applications.

Kubernetes clusters

Kubernetes clusters are used to run Kubernetes applications.They are private clusters invisible to other users, which ensures more secure isolation between your container applications and other

users' applications.

Kubernetes applications

Kubernetes applications are containerized applications deployed by Kubernetes.They are divided into stateful applications and stateless applications. The current version only supports stateless applications.A stateless application can have multiple pods to ensure high availability.

Pod

Pods are the smallest deployment units and billing units in Kubernetes, which comprise one or more containers based on the application scenario. When one pod comprises multiple containers, these containers share the compute resources, storage space, IP addresses, and ports of the pod. You can limit the proportion of computing resources allocated for each container.The pod can remove and create a stateless application at any time. When a pod has an exception, the system deletes the pod and creates a new one to keep the total number of pods for the service unchanged. This allows to clear an exception flexibly without affecting service availability.

Lightweight configuration center

EDAS lightweight configuration center can be run locally and includes features of service discovery and configuration management.

Collated logs

Collated logs are logs that are merged and sorted in a similar format in a single application.

Method tracing

EDAS method tracing adopts the JVM bytecode enhancement technique when recording the time consumption and sequence during the entire call process of the selected method, allowing you to check the execution sequence while execution is in progress.

Application runtime environment

It is the environment where the application runs, namely, EDAS Container. It includes HSF and Pandora.

EDAS application instance quota

This quota sets the maximum number of instances for all applications held by the primary account and its sub-accounts.When the payment account allocates a quota to each primary account, the sum of the quotas of all primary accounts bound to the payment account cannot be greater than the total application instance quota of the payment account.

IaaS basic monitoring

This is one of EDAS's monitoring functions. It monitors application running statuses and performs detailed monitoring of basic metrics, such as CPU, memory, load, network, and disk metrics.

Distributed link tracing

EDAS EagleEye provides analysis of every service call, message sending, and database access

occurred within the distributed system to help you identify system bottlenecks and risks precisely.

Service authentication

HSF is designed to ensure the reliability and security of each distributed service call.Strict authentication is implemented in every phase, from service registration and subscription to service call.

Service downgrade

Contrary to service throttling, service degradation pinpoints and blocks poor services that your application calls. This feature ensures the stable operation of your application and prevents the functionality of your application from being compromised by dependency on poor services. EDAS allows you to configure downgrade rules based on the response time, preventing your application from depending on poor services during traffic peaks.

Service call monitoring

This is another EDAS monitoring function. It can fully monitor the service calls made by your application in terms of the service QPS, response time, and error rate.

Service throttling

EDAS allows you to configure throttling rules for the services provided by each application to ensure service stability. Throttling rules can be configured based on QPS and threads to ensure the system’s best operation stability during traffic peaks.

Automated stress testing

The original automated stress testing tool helps you make performance stress testing a part of your daily routines.All the traffic for automated stress testing is the actual data in the product environment. By controlling the weights of each service, you can perform online stress testing without degrading the stability.

Containner version notes

Version	Release date	Basic edition	Change
3.5.0	2018-9-10	3.4.7	1. Upgraded eagleeye-core to V1.7.4.8. Fixed the problem of

			returning garbled Chinese parameter values for an URL request of the Web application. 2. Upgraded HSF to V2.2.6.7-edas. Fixed the problem of failing to obtain an HSF service list by running the Pandora QoS command. 3. Removed the ons-client plugin because of the possible conflict between the JAR Package for the ons-client plugin and that for the application.

3.4.7	2018-8-1	3.4.6	Upgraded ONS to V1.7.8-EagleEye. Eliminated the class conflicts resulted from the MQ Trace function.
3.4.6	2018-7-5	3.4.5	<Ol> - Upgraded HSF to V2.2.6.1. - <Ol> - Support for the CSB function. - Fixed serialization errors in some scenarios. - Fixed the problem of strong VIP dependency. - Support for health check on Dubbo in the Spring Boot runtime environment. - Upgraded config-client to V1.9.6. Support for dynamic adjustment of the maximum number of

			registrations. - Upgraded Sentinel to V2.12.12-edas. Support for Spring Boot 2. - </Ol>
3.4.5	2018-6-14	3.4.4	Upgraded ACM to V3.8.10. Fixed the problem of ineffective multi-tenant monitoring over a native interface.
3.4.4	2018-5-18	3.4.3	1. The time-out value is 0 during asynchronous processing on HSF provider or a local call, resulting in a time-out error. 2. Some peer IP attributes are missing in RpcContext for using Dubbo on EDAS. 3. Support for use of the service tags on Dubbo in

			the AOP scenario. 4. Unsupported if the Bool value is included in Map during a generalized call of HSF.
3.4.3	2018-4-24	3.4.1	1. Upgraded the Diamond to V3.8.8. 2. Fixed the problem of failing to locate the print certificate and improved the security level. 3. Upgraded EDAS-Assist to V2.0. 4. Optimized the port availability detection logic and upgraded fastjson to V1.2.48.
3.4.1	2018-3-15	3.4.0	1. Upgraded hsf-plugin.

			Support for dubboX. 2. Upgraded diamond-client and configcenter-client. 3. Upgraded edas-assit and canceled the port check after the specified port value is displayed.
3.4.0	2018-3-7	3.3.9	1. Upgraded edas-assist and improved the speed to check the available ports. Support for dynamic setting of a CSP interface. 2. Provided the ConfigCenter version for tenants. 3. Upgraded configclien

			t to ensure unified internal and external consumers. Support for CS2.0 and CS3.0 providers.
3.3.9	2018-1-17	3.3.8	1. Upgraded HSF and solved the problem of ZooKeeper blocking. 2. Upgraded Sentinel to provide additional system protection (effective only after you introduce the console push rules). 3. Modified the links of default errors for internationalization.
3.3.8	2018-1-3	3.3.6	1. Upgraded Sentinel to provide an interface

			that dynamically generates metadata. 2. Upgraded HSF to provide the tid transmission feature. 3. Upgraded ons and fixed the problem of ineffective link tracing.
3.3.6	2017-12-20	3.3.4	1. When the same exception is thrown repeatedly, the header information increases constantly, resulting in a ultra-long prompt message. 2. EagleEye parses an exception into UNKnown, affecting call trace parsing and topologica

			l display of an application .
3.3.4	2017-11-30	3.3.3	1. Upgraded Diamond to the latest version to be compatible with ACM. 2. Fixed a series of problems, including HSF generalization, Unit dependency, resolution exception of multiple ZK addresses, and InetAddress serialization. Support for setting whitelist rules. 3. Fixed the problem that the settings on the custom throttling and

			downgrade page have to wait for about 30 seconds to take effect. 4. EagleEye supports health check, with alimetric and tomcat monitor features. 5. Upgraded ons-client to configure message cache size in MQ on the consumer.
3.3.3	2017-10-18	3.3.2	1. Support for automatic registration of applications (disabled by default). 2. Fixed the problem of HSF file handle occupation. 3. Sentinel

			supports HSF2.2,4. Enhanced the Pandora QoS command.
3.3.2	2017-10-18	3.3.1	1. Fixed the problem of hsf.lock handle occupation on HSF. 2. Added Redis tracing. 3. Upgraded tddl-driver for online comprehensive stress testing. 4. Enhanced the Pandora QoS command.
3.3.1	2017-07-13	3.3.0	Upgraded tddl-driver separately for online comprehensive stress testing.