Enterprise Distributed Application Service (EDAS)

Console User Guide

MORE THAN JUST CLOUD | C-D Alibaba Cloud

Console User Guide

Resources

ECS

Create ECS instances

If you do not have any ECS instances, follow the instructions in this document to create an ECS instance.

Prerequisites

Before creating an ECS instance, determine the network type of your ECS instance:

Alibaba Cloud stopped providing ECS instances in classic networks to new ECS instance users at 17:00 (UTC +8:00) on June 14, 2017. For first-time purchasing of ECS instances, you can choose a VPC only. If you have already purchased ECS instances in classic networks, you can still select them.

To create an ECS instance in a VPC, you must select a specific VPC. The ECS instance created in the selected VPC cannot be moved to another VPC.Therefore, make sure you select the correct VPC before creating the ECS instance.If you do not have any VPCs, create a VPC first.

Procedure

Log on to the EDAS console.

In the left-side navigation pane, choose **Resource Management** > **ECS**.

On the **ECS** page, select **Region** and **Namespace** (optional) for the ECS instance to create. Then, click **Create Instance** in the upper-right corner of the page.

On the ECS purchase page, select ECS type and configuration and pay for it. See Create ECS instances for details.

Note: To use and manage an ECS instance in the EDAS console, you must install the EDAS Agent on the ECS instance. When purchasing an ECS instance, you can choose the EDAS base image to automatically install the EDAS Agent. The procedure is as follows:

In the **Image** section, select **Marketplace Images**. Then, click **Select from image market (including operating system)**.

Enter EDAS in the search box and click Search.

In the search results, choose **EDAS Java Environment (common ECS instance)** and the version (the latest version is selected by default). Then, click **Apply**.

Result verification

Click **Management Console** to return to the **ECS Console**. The ECS instance creation process usually takes one to five minutes.Click the Refresh button, and the ECS instance status will change to **Running**, indicating that the ECS instance has been created successfully.

Import ECS instances

After purchasing an ECS instance, you need to import it in order to install EDAS Agent and synchronize it to EDAS.

If you did not select any EDAS base image when purchasing an ECS instance, you can click **Import ECS** in the EDAS console to install EDAS Agent. The process of importing ECS is divided into direct import and import after conversion.

An ECS instance cannot be directly imported in any of the following conditions:

- The ECS instance was created before December 1, 2017.
- A classic network ECS instance is imported into a classic network cluster.
- The ECS instance is not running (it is stopped, starting, or stopping).
- The ECS instance is a Windows instance or does not support simple shell commands.
- The ECS instance is not imported from an ECS cluster.

Note: The ECS instance is formatted if image conversion is required. If you do not want the disks of the ECS instance to be formatted, click **Use command script for manual installation** on the conversion page.

Log on to the EDAS console. In the left-side navigation pane, choose **Resource** Management > ECS.

On the **ECS** page, select **Region** and **Namespace** (optional). Then, click **Import ECS** in the upper-right corner of the page.

On the **Select Cluster and ECS** page, select **Namespace** and **Select Cluster to Import**. Then, select an ECS instance and click **Next**.

Note:

- If the desired cluster is not in the list, click **Create Cluster** to the right of **Select Cluster to Import** to create a cluster.
- To manually install EDAS Agent, select **Switch to Manual Installation** in the upperright corner to go to the **Install EDAS Agent on Single Instance Manually** page, and use the script to manually install EDAS Agent.

On the Ready to Import Page, view the information of the selected instance.

- If the ECS instance can be imported directly, click **Confirm and Import**. If the ECS instance needs to be imported after conversion, select **I agree to convert the above instances, and fully understand that the data in the original systems will be lost after conversion** and enter the new password for root user logon after conversion. Click **Confirm and Import**.

On the **Import** tab page, view the import progress of the instance.

- If the ECS instance supports direct import, you can view its import progress on the **Import** tab page. If the message **Instance transfer succeeded** is displayed, the ECS instance has been imported successfully.Click **Click to return to the Cluster Details page**. When the ECS instance status changes to **Running**, the ECS instance is imported to the cluster successfully.
- For an ECS instance that needs to be converted before being imported, the import progress of the ECS instance displayed on the Import tab page is Converting now.
 This might take 5 minutes. If you click Click to return to the Cluster Details page

before the import is completed, the health check status **Converting** and the conversion progress in percentage are displayed. When the import is completed, the health check status **Running** is displayed, indicating that the ECS instance has been successfully imported.

Install EDAS Agent

EDAS Agent is the daemon program installed on ECS instances that communicates between the EDAS cluster and the applications deployed on the corresponding ECS instances. EDAS Agent provides the following functions:

- Application management: deploys, starts, and stops applications.
- Status reporting: reports application viability status, health check results, and Ali-Tomcat container status.
- Information retrieval: retrieves the monitoring information on ECS instances and containers.

In addition to this application-based management, EDAS Agent is also used to communicate between the EDAS console and your applications.For example, EDAS Agent must be used to determine whether an ECS instance has correctly and promptly published a service that an application publishes.

Note: You can transparently access these functions by installing EDAS Agent.



Install EDAS Agent

EDAS deploys applications (including first-time installation and later on expansion) on ECS instances that are installed with EDAS Agent only. The application instances in the EDAS billing system refer to the ECS instances which are installed with EDAS Agent and deployed with applications. To use EDAS, you must install EDAS Agent after you purchase an ECS instance.

There are three ways to install EDAS Agent:

- Use an EDAS base image
- Import an ECS instance
- Manually execute the script for installation

Note:

- JDK 8 is installed on EDAS Agent by default in these three scenarios. To use JDK 7 or another version, run the corresponding installation script.
- This script requires you to log on to your ECS instance as the root user. This script can be executed repeatedly, which overwrites the existing version of EDAS Agent installed on the ECS instance. Therefore, you can upgrade EDAS Agent by running the same script.
- The script for installation is region-specific. You must switch to the appropriate region before clicking **Install EDAS Agent**.
- Different installation methods or different images or clusters selected during installation result in different EDAS Agent statuses. This determines the type of applications that you can create on the ECS instance. When using an installation method, follow the related instructions.

Use an EDAS base image to automatically install EDAS Agent when purchasing an ECS instance

The easiest method of installation of EDAS Agent is to use the EDAS base image when purchasing an ECS instance.

Note: This method requires disk formatting. To avoid formatting disks, we recommend that you use the command script to manually install EDAS Agent.

Log on to the EDAS console. In the left-side navigation pane, choose **Resource** Management > ECS.

In the upper-right corner of the ECS page, click Create Instance.

On the purchase page, select **Image Market** in the **Image** section, and then click **Select from image market (including operating system)**.

Enter **EDAS** in the search box and click **Search**.

Select an image from the search results based on the requirements of your applications. By default, the latest version of image is selected (we recommend that you do not select an earlier version). Then, click **Use**.

To create a Common Application, select EDAS Java Environment (Common ECS).

To create a **Docker application**, select **EDAS**.

Follow the instructions on the page to purchase an ECS instance.

Import an ECS instance to automatically install EDAS Agent

If you did not select any EDAS base images when purchasing an ECS instance, you can use the **Import ECS** function in the EDAS console to install EDAS Agent. An ECS instance can be imported either directly or after image conversion.

Under any of the following conditions, direct import of an ECS instance is not possible:

- The ECS instance was created before December 1, 2017.
- A classic network ECS instance is imported into a classic network cluster.
- The ECS instance is not running (it is stopped, starting, or stopping).
- The ECS instance is a Windows instance or does not support simple shell commands.
- The ECS instance is not imported from an ECS cluster.

Note: If image conversion is required, disk formatting will be performed. To avoid formatting disks, we recommend that you manually execute the script to install EDAS Agent.

In the left-side navigation pane of the EDAS console, choose **Resource Management** > **ECS**.

On the **ECS** page, select **Region** and **Namespace**. Then, click **Import ECS** in the upper right corner of the page.

On the **Select Cluster and ECS** page, enter **Namespace** and **Select Cluster to Import**.In the ECS instance list, select the ECS instance that needs to be imported, and click **Next**.

Note:

- If the desired cluster is not in the list, click **Create Cluster** to the right of **Select Cluster to Import** to create a cluster.
- To manually install EDAS Agent, click **Switch to Manual Installation** in the upperright corner to go to the **Install EDAS Agent on Single Instance Manually** page and use the command script to manually install EDAS Agent.

On the **Ready to Import** Page, view the information of the selected instance.

- If the ECS instance can be imported directly, click **Confirm and Import**. If the ECS instance needs to be converted before import, select **I agree to convert the above instances**, and fully understand that the data in the original systems will be lost

after conversion and enter the new password for root user logon after conversion.Click **Confirm and Import**.

On the **Import** tab page, view the import progress of the instance.

- If the ECS instance supports direct import, you can view its import progress on the Import tab page. If the message Instance transfer succeeded is displayed, the ECS instance has been imported successfully.Click Click to return to the Cluster Details page. If the status of the ECS instance is Running, the ECS instance has been imported to the cluster and EDAS Agent has been successfully installed.
- For an ECS instance that needs to be converted before being imported, the import progress of the ECS instance displayed on the **Import** page is **Converting now. This may take 5 minutes**. If you click **Click to return to the Cluster Details page** before the import is completed, the health check status **Converting** and the conversion progress in percentage are displayed. When the import is completed, the health check status **Running** is displayed, indicating that the ECS instance has been imported to the cluster and EDAS Agent has been installed successfully.

Use the command script to manually install EDAS Agent

Note: This approach applies to ECS instances in the ECS cluster only.

1.In the left-side navigation pane, choose Resource Management > ECS.

On the **ECS** page, select **Region** and **Namespace**. Then, click **Import ECS** in the upper right corner of the page.

Click **Switch to Manual Installation** in the upper-right corner of the **Import ECS** page. On the **Install EDAS Agent on Single Instance Manually** page, click **Click to Copy**.

Note: To install EDAS Agent with an image, click **Switch to Image Installation** in the upperright corner of the page and then **import an ECS instance**.

Log on to the ECS instance where the EDAS Agent is to be installed as the root.

On the ECS instance, paste the copied command and execute it.

Result verification

After installing EDAS Agent, choose **Resource Management** > **ECS** from the left-side navigation pane of the EDAS console. On the **ECS** page, choose the appropriate region to view the **Agent Status**.

If the EDAS Agent installation is successful, the Agent status is **Online** (for ECS clusters) or

Docker Online (for Swarm or Kubernetes clusters).

If the EDAS Agent installation failed, the Agent status is **Unknown**.

Upgrade EDAS Agent

The procedure for upgrading EDAS Agent is similar to that for installing EDAS Agent with a command script. For details, see Use the command script to manually install EDAS Agent.

SLB

If you have purchased Alibaba Cloud Server Load Balancer, EDAS synchronizes the SLB instances to the EDAS console, allowing you to use and configure load balancing functions.

View SLB instances

Log on to the EDAS console.

In the left-side navigation pane, choose **Resource Management** > **SLB**.

On the **SLB** page, select **Region** to view information about SLB instances in the region.

Description of SLB instances:

- **SLB ID/Name**: The ID is automatically generated by the system while the instance name is user-defined.Click the SLB ID to go to the SLB console.
- Service Address: indicates the Internet or intranet IP address of the SLB instance.
- **Backend Server**: indicates the ECS instance added in the SLB console that is used to receive the requests distributed by the SLB instance.
- **Status**: indicates the status of the SLB server, which can be running or stopped.Expired SLB instances are not displayed.

Note: To create an SLB instance, click **Create Instances** in the upper-right corner of the page. Then, go to the SLB purchase page on the Alibaba Cloud official website to purchase and create an SLB instance.For more information, see the **SLB documentation**.

VPC

Alibaba Cloud provides two network types:

Classic networks

Cloud products on a classic network are all deployed in Alibaba Cloud's public infrastructure and planned and managed by Alibaba Cloud. These products are suitable for customers who have high ease-of-use requirements.

VPC networks

VPC networks are virtual private clouds that allow custom isolation settings. You can define the custom VPC topology and IP addresses. VPCs are suitable for customers with high cybersecurity requirements and network management capabilities.

After purchasing a VPC and synchronizing it to the EDAS console, you can view its information in the EDAS console.

Log on to the EDAS console.

In the left-side navigation pane, choose **Resource Management > VPC**.

VPC (*) Install log collector Synchroniz						
Tip: Latency exists in VPC sync	Tip: Latency exists in VPC synchronization. Click the Synchronize VPC button to update manually					
VPC ID	Name	CIDR	Status	ECS Instances	Actions	
sys. Hereitysan Koshida	janiergies/01/2	10.0510/0	 Available 	0	Install Log Collector	
spectoperately defined as	1925	00000000	Available	0	Install Log Collector	
sector contractions	ana ayo analishina adalahasing monus-bar	102 103 1010	Available	0	Install Log Collector	
100 - FRANK SALAN (1913)	nye laha ku na natili bishin kitarita Tanonye angali bi	10, 1010, 975	Available	0	Install Log Collector	

On the VPC page, select Region to view information about VPC instances in the region.

Description of VPC instances:

- **VPC ID**: automatically generated by the system when the VPC is created.Click the VPC ID to go to the VPC console.
- Name: indicates the VPC name that you specified during VPC creation.
- **CIDR Block**: indicates the VPC' s CIDR block you specified when the VPC is created.
- **Status**: indicates the status of the VPC instance: Running or Stopped. Expired VPC instances are no longer displayed.

- **ECS instances**: indicates the number of ECS instances created in this VPC network. Click the number to go to the ECS page, where you can see all the ECS instances in this VPC.

In a VPC network, ECS instances are isolated from EDAS instances. Therefore, you must install the log collector to collect the information on ECS instances. Click **Install Log Collector** in the Actions column on the Instances page. For log collector installation instructions, see **Install log collector**.

Resource groups

Overview

Resource groups are groups of EDAS resources, which can be ECS instances, SLB instances or clusters, but not VPCs.You can use a primary account to purchase and group resources and assign appropriate permissions for the resource groups to sub-accounts for application O&M. You can assign permissions to access resource groups to your sub-accounts. Each sub-account has the right to operate on all the resources in the specified group.For more information about primary accounts and sub-accounts, see EDAS account system.

Typical scenarios

- A company uses EDAS to create business applications.Department A is responsible for userrelated applications and Department B for goods-related ones.
- The company registers for an EDAS account (the primary account) to activate EDAS and creates two sub-accounts for Departments A and B. Department A has a several ECS instances and SLB instances for deploying user-related applications.
- The company creates a resource group in EDAS, adds Department A' s ECS instances and SLB instances to the resource group, and grants permissions for this resource group to Department A' s sub-account.
- Department A can use its sub-account to operate on the resources in this resource group only. The resources for Department A and Department B are not conflicted.

This is shown in the following figure:



Resource group operations

Resource group operations are performed in the EDAS console. Follow the procedure discussed below to go to the Resource Groups page.

Log on to the EDAS console.

In the left-side navigation pane, choose **Resource Management** > **Resource Groups**.

On the **Resource Groups** page, select a region to view the resource groups in the region and the ECS and SLB instances in each group.

View resource groups

On the **Resource Groups** page, you can view information about resource groups, including resource group names, descriptions, ECS instances (intranet/Internet IP addresses), and SLB instances, and clusters.

Create a resource group

- 1. In the upper-right corner of the **Resource Groups** page, click **Create Resource Group**.
- 2. In the **Create Resource Group** dialog box, enter **Resource Group Name** and **Resource Group Description** and click **OK**.

Add ECS instances to the resource group

- 1. In the resource group list, locate the row that contains the target resource group, and click **Bind ECS** in the Actions column.
- 2. In the **Bind ECS** dialog box, select the target ECS instance and click **OK**.

Add an SLB instance to the resource group

- 1. In the resource group list, locate the row that contains the target resource group, and click **Bind SLB** in the Actions column.
- 2. In the Bind SLB dialog box, select the target SLB instance and click OK.

Edit a resource group

- 1. In the resource group list, locate the row that contains the target resource group, and click **Edit** in the Actions column.
- 2. In the **Edit Resource Group** dialog box, edit the resource group name and description and click **OK**.

Grant resource group permissions to a sub-account

- 1. Log on to the EDAS console using the primary account.
- 2. In the left-side navigation pane, choose Account Management > Sub-Accounts.
- 3. Locate the row that contains the target account and click the **Resource Group Permission** button in the Actions column.
- 4. In the Resource Group Permission dialog box, select a resource group and click OK.

Delete a resource group

- 1. In the resource group list, locate the row that contains the target resource group, and click **Delete** in the Actions column.
- 2. In the **Delete Resource Group** dialog box, click **OK** to confirm the deletion.

Clusters

Create ECS clusters

To create an ECS cluster for an application, you need to complete the following steps:

- 1. Create an ECS cluster
- 2. Add ECS instances

When no application is deployed on an ECS instance in a cluster and the ECS instance is no longer needed, you can **remove** to release the ECS instance.

Create clusters

Log on to the EDAS console.

In the left-side navigation pane, choose **Resource Management** > **Clusters**.

On the **Clusters** page, select **Region** and **Namespace**. Then, click **Create Instance** on the right of the page.

In the Create Cluster dialog box, set the cluster parameters and click Create.

Create Cluster						×
* Cluster Name:	Enter a cluster na	me.				
Cluster:	Alibaba Cloud	Non-Alibaba Cloud				
* Cluster Type:	ECS		~			
* Network Type:	VPC		~			
VPC Network:	defaultvpc		~	G	Create VPC	
Namespace:	cn-hangzhou					
					Create	Cancel

Description of cluster parameters:

- **Cluster Name**: Enter the cluster name.The name can only contain letters, digits, underscores (_), and periods (.), with a length up to 64 characters.
- Cluster type: Select ECS.
- Network Type: Select Classic Network or VPC.Select a network type from the dropdown list as needed.If you select the VPC Network, ensure that you have created a VPC instance first.
- VPC Network: Select a VPC instance you created from the drop-down list.
- **Namespace**: indicates the namespace displayed on the Clusters page, which cannot be modified.If you do not make a selection, the name of the displayed region is

selected by default.

After creating the cluster, you will see the message **Created successfully** in the upper-right corner of the page, and the cluster is displayed in the cluster list.

Note: The created cluster is empty.Only after you add an ECS instance can you meet the requirements of applications.

Add ECS instances

There are currently two ways of adding ECS instances to a cluster in EDAS:

- Direct import: Image conversion is not required.
- Import after conversion: The EDAS official image is used to reinstall the system. After reinstallation, all data in the ECS instance is deleted and the ECS instance login password needs to be reset.

An ECS instance cannot be directly imported in any of the following conditions:

- The ECS instance was created before December 1, 2017.
- A classic network ECS instance is imported into a classic network cluster. The ECS instance is not running (it is stopped, starting, or stopping).
- The ECS instance is a Windows instance or does not support simple shell commands.

Direct import

On the **Clusters** page, click the name of the cluster you created.

On the Cluster Details page, click Add ECS Instance in the upper-right corner.

On the **Select Cluster and ECS** page, select an import method and ECS instance in the instance list, and then click **Next**.

- **Import ECS**: You cannot modify the namespace and the imported cluster. Additionally, ECS instances are imported from the default namespace and cluster in this region.
- **From Existing Cluster**: A namespace and a cluster are selected from the region.Then, the ECS instance is moved from the left pane to the right pane on the page.

If no instances meet the necessary conditions, click **Create ECS Instance** in the upper-right corner of the page. This takes you to the ECS purchase page on the Alibaba Cloud official website, where you can purchase and create a new ECS instance.For more information, see

Create ECS instances.

集群 / 集群详情 / 添加ECS实例				创建ECS实例	
选择集群和已有云服务器实例		准备导入			
包出前所在命名空间:W					
 ・身入ECS ・从已有集群选择 ・命名空间: ・数以命名空间	◇ *选择导入集群:				
温馨提示: ECS同步至EDAS存在一定的延时,如	如果搜索不到,您可以点击"同步ECS"按钮再	搜索,或从ECS控制台选中对应实例后,手病	协安装EDAS Agent。 同步ECS		
核期接票 和人系的系统、IDBDP 成素 刷新					
实例ID/名称	VPC	IP 地址	规格	状态	
international and the second	en e	(公) (私)	cpu:1核 内存:2048MB	运行中	
			共有1条 ,每页显示:10条	〈 1 〉 前往 1 页	

On the **Ready to Import** page, view the selected ECS instance information, and click **Confirm and Import**.

On the **Import** page, view the import progress of the ECS instance.

When **Direct import successful** is displayed, the ECS instance has been successfully added to the cluster.

Click **Click to return to the Cluster Details page**. After the ECS instance is imported, the **Health Check** status of the ECS instance is **Running**.

Import after conversion

You need to reset the password during import. An ECS instance may be reinstalled and the original data may be deleted in the process.

See steps 1 to 3 of [Direct import] to select an ECS instance.

On the **Ready to import** page, check the information about the selected ECS instance and select **I agree to convert the above instances, and fully understand that the data in the original systems will be lost after conversion**.Enter the new password of the root user for logging on to the system after conversion, and click **Confirm and Import**.

On the **Import** page, view the import progress of the ECS instance.

When you begin the import, the **Status** of the ECS instance is **Converting now. This might take 5 minutes.**

Click **Click to return to the Cluster Details page**. The **Health Check** status of the ECS instance is **Converting** and the progress percentage is displayed.

When the ECS instance is imported, the **Health Check** status is **Running**.

Remove an ECS instance

In the **ECS Instances and Applications** area of the **Cluster Details** page, locate the row that contains the target ECS instance and click **Remove** in the Actions column.

In the **Remove an ECS instance** dialog box, confirm the instance information and click **Remove**.

In the removal process, the **Health Check** status of the ECS instance is **Deleting** and the progress percentage is displayed.

After the ECS instance is deleted, it is no longer displayed in the instance list.

Manage clusters

View the cluster list

Log on to the EDAS console.

In the left-side navigation pane, choose **Resource Management** > **Clusters**.

On the **Clusters** page, select **Region** to view information about clusters in the region.

Note:

- There are two types of clusters: EDAS Cluster and Container Service K8S Cluster. EDAS Cluster includes ECS Cluster and Swarm Cluster.
- Two network types, namely, **VPC** and **Classic Network**, are available.

View cluster details

On the **Clusters** page, click the name of the target cluster.

On the Cluster Details page, view the cluster details.

The **Cluster Details** page mainly consists of Cluster Information and ECS Instances and Applications.

Cluster Information: displays basic information about the cluster.

ECS Instances and Applications: displays the list of ECS instances in the cluster, information about specific ECS instances, and applications deployed.

Deployed Applications: displays the applications deployed on the ECS instance.Click the name of the target application to go to the Application Details page.

Click a specific button in the **Actions** column to perform relevant operations on the ECS instance.

View Details: View details of the instance.

- Event (applies to Swarm clusters only): View the events that occurred to the ECS instance in the cluster. Event information helps you locate problem causes.
- **Remove**: Remove the ECS instance. If any application is deployed on the ECS instance, the Remove button is unavailable.

Transfer ECS instances (for ECS clusters)

In the **ECS Instances and Applications** area on the **Cluster Details** page, select an ECS instance and then click **Transfer ECS Instances** on the right side.

On the Select Target Cluster page, select Namespace and Target Cluster, and click Next.

On the Ready to Import page, click Confirm and Import.

- If the ECS instance cannot be imported directly, select I agree to convert the above instances, and fully understand that the data in the original systems will be lost after conversion. Reset the password of the root user for logging on to the ECS instance.

Deploy application

Application deployment overview

After developing applications, you can choose console or development tools to deploy the applications.

Note: Applications will be deployed in clusters. Packaging methods for applications are limited results of different type of cluster. For details, see the following table.

Application	Deployment Cluster	Package
Spring Cloud or Dubbo	ECS cluster	WAR or JAR
	ECS cluster	WAR or JAR
HSF	Container Service Kubernetes cluster	Image

Application runtime environment

You can deploy a WAR or JAR package of Spring Cloud, Dubbo or HSF application into ECS cluster, but you should select different application runtime environment for Spring Cloud, Dubbo and HSF.

when deploying Spring Cloud and Dubbo applications using WAR, you shoud selec **apache-tomcat** related versions of application runtime environment.

when deploying Spring Cloud and Dubbo applications using JAR, you shoud selec **Standard Java Application Runtime Environment**.

When deploying HSF applications using WAR or JAR, you should select **EDAS-Container** related versions of application runtime environment.

Deploy applications in the console

Note : The Chrome browser is recommended for operations in the EDAS console.

Use the following resources for the application deployment tutorials to start developing applications

in the EDAS console.

Note: The configuration procedures for WAR and JAR deployment methods are similar. The following describes how to deploy an application with a **WAR** package.

Deploy ECS applications using WAR packages



Deploy Container Service Kubernetes applications

Deploy applications with tools

In addition to deploying applications in the EDAS console, you can also deploy applications with the following tools.



Use the Maven plugin to automatically deploy



Use Eclipse to quickly deploy applications



Use CLI to quickly deploy applications in the EDAS



Use IntelliJ IDEA to quickly deploy applications

Application Deployments (Console)

Deploy Java Web applications in ECS clusters

An ECS instance can deploy only one ECS application.To publish an ECS application is to install an EDAS container on your ECS instance and then use a WAR or JAR package to deploy the application in the EDAS container.

This topic describes how to create a Java Web application with only a welcome page in the EDAS console, and how to deploy, update, view, and manage the application with a WAR package.

After reading this document, you will be able to:

- Create ECS applications in the EDAS console.
- Deploy an ECS application to the web.
- Update a deployed ECS application.
- Verify a deployed application
- Manage applications

Prerequisites

To perform the steps in this tutorial, you must have completed the following steps:

- 1. Activate EDAS.
- 2. [Create VPCs].(https://www.alibabacloud.com/help/doc-detail/87742.html#creatVPCInEDAS)
- [Create ECS instances].(https://www.alibabacloud.com/help/docdetail/87742.html#creatECSInEDAS)
- 4. Create namespaces.
- 5. Create ECS clusters.
- 6. Synchronize SLB instances to the EDAS console (only if you need to configure SLB).

Create ECS applications

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. At the top of the **Applications** page, select a namespace. All applications in the namespace are displayed.

Click **Create Application** in the upper-right corner of the Applications page.

In the **Create Application** dialog box, enter application information and click **Next Step: Application Configurations**.

Appli	ication Information	A	pplication Configuration		Application Creation Complete
* Namespace:	China North 2 (Beijing)	-	Default		0
* Cluster Type:	ECS Cluster V	-	CN-BEIJING_CLASSIC_DEFAULT	~	0
* Application Name:	Demotest				
* Deployment Method:	• WAR 🔾 JAR				
* Application Runtime Environment:	EDAS-Container 3.5.2 [Support FatJar Deployment] \vee				
Java Environment: 🕧	Open JDK 8				
Application Description:	Enter application description			G	
				0/128	
					Create an Empty Application Next

- **Namespace**: Select **Region** and **Namespace** from the drop-down list.If you do not select a namespace, the **default** namespace is automatically selected.
- Cluster Type: Select ECS Cluster from the drop-down list and then select a specific cluster.
- Application Name: Enter the application name.
- **Deployment Method**: Select **WAR** or **JAR**.(The configuration procedure for WAR and JAR deployment methods are similar. The following describes how to deploy an application with a **WAR** package.))

Application Runtime Environment: Select the latest EDAS Container version.For example, EDAS container V3.5.1 [supports FatJar deployment].

Java Environment: Select JRE 8 or JRE 7.

- **Application Description**: Enter up to 100 characters of basic application information.

On the **Application Configuration** page, add an instance and configure the instance as instructed. Then click **Create**.

Selected Instances: Click **Add**. On the displayed **Instances** page, select an instance and click > to add the instance to the right area, and then click **OK**.

For no instance selected: Click Create Empty Application to create an empty application. Then, click Scale Out, Add Instance, or Deploy Application to publish the application.

For an instance selected: Click **Create** to create an empty application containing instances. Then, click **Deploy Application** to publish the application.

Deploy Now: This option is available only after you select an instance.Enable **Deploy Now** and configure the parameters as instructed on the page.

Deploy Now:	
 File Uploading Method: 	Upload WAR Package
* Upload WAR Package:	Select File
	Download Sample WAR Package
* Version:	Version
Application Health Check:	http://127.0.0.1.8080/_ebc.html
* Batch:	1 Batches \checkmark
* Batch Mode:	Automatic \checkmark

File Uploading Method: Select Upload WAR Package or WAR Package Location.

• Upload WAR Package: Click Download Sample WAR Package. After the sample package is downloaded, click Select File and select a WAR package.

WAR Package Location: Right-click Download Sample WAR Package and copy the link address. Paste the link address to the WAR package address bar.

Note: The name of an application development package can contain letters, digits, hyphens (-), and underscores (_). A JAR package can be uploaded only if the selected version supports JAR deployment. Otherwise, the application must be deployed through a WAR package.

Version: Set a version (for example, 1.1.0) for the application. We recommend that you do not use a timestamp as the version number.

Application Health Check (optional): Set a URL for application health checking.The system checks the health of the application after the container is started or is running to see whether the application is working normally. Then, it performs a

service routing task based on the results of the health check. A sample URL is as follows: http://127.0.0.1:8080/healthCheck.html

Batch: Specify the number of batches. If you select two or more batches, you need to set the **Interval.Batch Mode**: Select Automatic.

Wait several minutes until the application is created. After creating an application, go to the Application Details page to view the application. On the **Instance Information** tab page of the Application Details page, view the running status of the application instance. If the running status/time is **Running**, the application is published successfully.

Deploy and update applications

If only an empty application is created, on the Application Details page, click **Deploy Application** to deploy the application. If the created application has been deployed, click **Deploy Application** to update the application.

Log on to the EDAS console. In the left-side navigation pane, choose **Application Management** > **Applications**. On the Applications page, click the name of the application to be deployed.

On the **Application Details** page, click the **Instance Information** tab. On the tab page that is displayed, check whether an instance is available for the application. If not, click **Scale Out** to add at least one instance for the application.

Click **Deploy Application**, follow the instructions on the interface to set the deployment parameters, and click **Publish**.

Deploy Application			×
* Deployment Method:	• WAR Historical Version		
* File Uploading Method:	Upload WAR Package	~	Download Sample Project
* Upload WAR Package:			Select File
* Version:	Enter a version number.		Use Timestamp as Version Number
* Group:	Default Group	~	
* Batch:	1 Batches	~	
* Batch Mode:	Automatic	\sim	
Java Environment:	選択してください	~	
Application Description:	For example: "This release has fixed the following bugs:". Maximum 128 characters.	li	
> Generate Maven	Plug-in Configuration		
Learn more about depl	loyment methods		
			Deploy Cancel

- Deployment Method

- For a new application: Select **WAR**, **JAR**, or **Historical Version** to deploy the application.
- For an upgraded application: Select the method used for the first deployment or **Historical Version** to deploy the application.
- File Uploading Method: Select Upload WAR Package or WAR Package Location (in the example, a WAR package is used).
 - Upload WAR Package: Click Select File and select the WAR package used to deploy the application.
 - WAR Package Location: Copy the WAR package storage address and paste it into the WAR package address bar.
- **Version**: Set a version (for example, 1.2.0) for the application. We recommend that you do not use a timestamp as the version number.
- Group: Select a group.
- **Batch**: Specify the number of batches. If you select two or more batches, you need to set the **Interval.Batch Mode**: Select Automatic.
- Java Environment (optional): Select JRE 8 or JRE 7.
- **Application Description** (optional): The description can be blank. The description content should describe the purpose of deploying this application. The maximum length of the content is 128 characters.
- Generate Maven Plugin Configuration: Generate the Maven plugin configuration and deploy an application based on the procedure in Use toolkit-maven-plugin to automatically deploy applications.

After you redeploy an application, the **Change Details** page is displayed and you can view the deployment procedure and operation logs on the page.After you complete the deployment, the execution status becomes **Execution Successful**.

Verify the results of application publishing

After publishing an application, you can view the instance running status or log on to the SLB instance for address configuration to verify that the application is published successfully.

View the running status of the application instance

On the **Instance Information** tab page of the Application Details page, view the running status of the ECS instance.If the running status/time is **Running**, the application is published successfully.

Configure Internet SLB and access the application

Because the application is created and published in the VPC, it has no public IP address unless configured specially. If you deploy your application on multiple ECS instances and want to allow external access, we recommend that you configure Internet SLB instances to distribute application access traffic to ECS instances based on a forwarding policy. This enhances the service capabilities and application availability.

In the Application Settings area on the Basic Information page, click **Add** to the right of **SLB** (Internet).

In the **Bind SLB to Application** dialog box that appears, set the SLB parameters and click **Configure SLB**.

After SLB port listening is en Do not delete the monitor on	abled, a port listener will be added to the newly-added SLB a the SLB console. Otherwise, application access will be affect	utomatically.	
SLB(Internet) :	47.91.4.228 (auto_named_slb)	•	
Use VServer Group: 🗹			
VServer Group		*	
(Internet) :			
Listener (Internet) :		Ŧ	
Create New Listener :	V		
SLB Frontend	ТСР		
Protocal:			
SLB Frontend Port:	Enter the frontend port number of SLB		
]	
Application Port:	8080		

- **SLB (Internet)**: Select the intranet or Internet address of the SLB instance as needed from the drop-down list.
- Use VServer Group: A VServer group is a group of frontend ECS instances used to process requests distributed by SLB instances.Different listeners can be associated with different VServer groups to distribute different requests to different backend servers.If you select Use VServer Group, you must configure the VServer Group parameter.
- VServer Group Name: Enter the name for the new VServer group if you select Create VServer Group. The system creates a VServer group with the specified name.
- Listener (Internet): SLB listeners regulate how requests are forwarded to backend servers.At least one listener must be created for each SLB instance.You can select a listening port from the Listener drop-down list.If no listener has been created, click Create New Listener.- Do not delete the listener in the SLB console; otherwise, the application cannot be accessed normally.
- **SLB Frontend Protocol**: The default setting is TCP, which cannot be configured manually.
- SLB Frontend Port: Enter the frontend port of the SLB instance.
- Application Port: The default setting is 8080, which cannot be configured manually.

Copy the URL consisting of the IP address and port of the SLB instance, for example, *118.31.159.169:81*. Then, paste it into the address box of the browser and press Enter. The

welcome page of the application appears.

Manage ECS applications

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the Applications page, click an application name.

On the Application Details page, click the **Basic Information** tab. On the tab page that is displayed, you can perform basic management tasks, for example, viewing, setting, stopping, starting, scaling up/down, rolling back, and deleting an application.

Deploy applications with tools

Use toolkit-maven-plugin to automatically deploy applications

Previously, EDAS applications had to be deployed by following the step-by-step instructions in the console. To improve the developer experience, toolkit-maven-plugin has been provided for automatic application deployment. Developers who deploy applications by using WAR or JAR packages can now use toolkit-maven-plugin to automate the deployment of ECS applications and Swarm applications.

Deploy applications automatically

Add the following plug-in dependencies to the pom.xml file in your packaged project.

<build> <plugins> <plugin> <groupId>com.alibaba.cloud</groupId> <artifactId>toolkit-maven-plugin</artifactId> <version>1.0.0</version> </plugin> </plugins> </build>

Create the .edas_config.yaml file in the root directory of the packaged project.If the packaged project is a Maven submodule, create the file in the submodule directory.

env: region_id: cn-beijing app: app_id: eb20dc8a-e6ee-4f6d-a36f-5f6a54550453

Among the preceding configuration items, region_id indicates the region ID of the application instance, while app_id indicates the application ID. The values of the preceding configuration items are for reference only. Replace them as needed.For information about other configuration items, see More configuration items.

Perform the following steps to obtain the values of these parameters:

Log on to the EDAS console.

In the left-side navigation pane, click **Application Management**. On the Applications page, click the application that you want to deploy. On the Application Management page, click **Deploy Application**.

On the **Deploy Application** page, click **Generate Maven Plugin Configuration** to retrieve the parameter values.

✓ Generate Maven Plug-in Configuration
<pre># This configuration is generated based on your input. Refer to the following document for details: # //help.aliyun.com/document_detail/92242.html</pre>
env:
region_id: cn-hangzhou
app:
app_id:
group_id:
batch: 1
<pre>batch_wait_time: 0</pre>

Create an account file and configure the Access Key ID and Access Key Secret in yaml format. To view the Access Key ID and Access Key Secret, see User information management. We recommend that you configure application permissions with a RAM subaccount to improve application security. The following provides an example of configuration: access_key_id: abc access_key_secret: 1234567890

Note: In this configuration, abc and 1234567890 are for reference only. Replace them as needed.In this configuration, the Access Key ID and Access Key Secret are only used to generate request signatures and not for any other purposes, such as network transfers.

Go to the root directory (or the submodule directory if multiple Maven modules are used) and run this packaging command:

mvn clean package toolkit:deploy -Daccess_key_file={Account file path}

The preceding parameters are described as follows:

toolkit:deploy: Use toolkit-maven-plugin to deploy the application after successful packaging. The application is deployed only when this parameter is added.

access_key_file: indicates the Alibaba Cloud account file.(For more information on how to specify an access key, see **Account configuration**).

After running the preceding command, the following message is displayed, indicating that you have successfully deployed the application with toolkit-maven-plugin.

[INFO]	Sending deploy request to EDAS
[INFO]	Deploy request sent, changeOrderId is: 71548877-5bb3-4503-88e5-04761c0ac688
[INFO]	Begin to trace change order: 71548877-5bb3-4503-88e5-04761c0ac688
[INFO]	PipelineName:Batch: 1, PipelineId:9d7b8038-534c-4891-8f70-e63db6606297
[INF0]	<pre>StageName:Workflow Start, StageId:cda21377-260d-43f0-9c82-98b55c3fd1ec</pre>
[INF0]	ServiceStageName:Workflow Start, ServiceStageId:cda21377-260d-43f0-9c82-98b55c3fd1ec
[INFO]	StageName:SLB Offline, StageId:1060b156-38b3-47aa-990d-5b5925256a57
[INFO]	ServiceStageName:SLB Offline, ServiceStageId:1060b156-38b3-47aa-990d-5b5925256a57
[INFO]	StageName:Deploy, StageId:035c7222-b84a-4c3a-bec2-3aeb3a5ac90e@co_virtual_stage
[INFO]	InstanceName:nahai-20180920, InstanceIp:47.93.157.199(Public) br>10.30.137.182(Inner)
[INFO]	<pre>InstanceStageName:RPC Offline, InstanceStageId:3a0fbf0b-1dda-4368-8b67-b2e8bf7c7de7</pre>
[INFO]	Waiting
[INFO]	Waiting
[INF0]	Waiting
[INF0]	Waiting
[INF0]	Waiting
[INFO]	Waiting
[INFO]	<pre>InstanceStageName:Stop Application, InstanceStageId:f8df6de7-adea-47c0-a2cd-6a18b49ebb2f</pre>
[INFO]	InstanceStageName:Environment initial, InstanceStageId:033ca419-7d0e-4996-ba91-0b4e67877548
[INFO]	InstanceStageName:Start Application, InstanceStageId:634e45bc-a411-429d-a549-b693794a41ac
[INF0]	Waiting
[INF0]	Waiting
[INFO]	<pre>InstanceStageName:Health Check, InstanceStageId:77f179ee-f8ac-40ad-85a5-86cad41b1e4a</pre>
[INFO]	Waiting
[INFO]	StageName:SLB Online, StageId:035c7222-b84a-4c3a-bec2-3aeb3a5ac90e
[INFO]	ServiceStageName:SLB Online, ServiceStageId:035c7222-b84a-4c3a-bec2-3aeb3a5ac90e
[INFO]	StageName:Workflow Complete, StageId:510d5110-d430-4508-93c4-864dc506b9ae
[INFO]	ServiceStageName:Workflow Complete, ServiceStageId:510d5110-d430-4508-93c4-864dc506b9ae
[INFO]	Deploy application successfully!
[INFO]	
[INFO]	BUILD SUCCESS

More configuration items

Configuration items for deploying applications are classified as follows:

- Basic environment (env)
- Application configuration (app)
- Storage configuration (OSS)

The currently supported configuration items are listed in the following table.

Туре	key	Required	Description
env	region_id	Yes	Indicates the ID of the region where the application is located.
	app_id	Yes	The ID of the application.
	package_version	No	Indicates the version of the deployment package.The default value is the string of the pom.xml file version plus the instance creation time, for example, 1.0 (2018- 09-27 19:00:00).
	desc	No	Indicates the deployment description.
app	group_id	No	Indicates the deployment group ID.The default value is "all groups" .
	batch	No	Indicates the number of deployment batches.The default value is one batch and the maximum is five batches.
	batch_wait_time	No	Indicates the waiting time (in minutes) between deployment batches.The default value is 0.
	buildPackId	No	If you created the application without specifying the runtime environment and deploying it:For a native Dubbo or Spring Cloud application, this

			field can be ignored the first time it is deployed.For a non- native Dubbo or Spring Cloud application, this field is required the first time it is deployed.(Retrieve the field value over the ListBuildPackRequest API.)
	componentIds	No	If you created the application without specifying the runtime environment and deploying it:For a native Dubbo or Spring Cloud application to be deployed for the first time using a WAR package, specify the Apache Tomcat version through this field.(Retrieve the field over the ListComponentsReques t API).For a non-native Dubbo or Spring Cloud application to be deployed for the first time using a WAR package, this field can be ignored the first time it is deployed.
	stage_timeout	No	Indicates the timeout period (in minutes) for each release stage. The default value is 5 minutes.If batch_wait_time is set, it is automatically added to this parameter during calculation.During runtime, if a stage waits for a longer time than this threshold value, the plugin automatically exits.
OSS	region_id	No	Indicates the ID of the region where the target bucket is located.The default value is the ID of the region where the application is located.

	bucket	No	Indicates the name of the target bucket.The default value is the free OSS storage space provided by EDAS.If OSS configuration items are specified, you must specify the bucket parameter. Otherwise, the instance uses the free OSS storage space automatically allocated
	key	No	Indicates the custom path used to upload the application package to OSS. The instance uses the free OSS storage space provided by EDAS by default.If you are using a specified OSS storage space, specify the package storage path in this parameter and use the {region_id}, {app_id}, and {version} variables to set the parameterized path, for example, pkgs/petstore/{version} /store.war.The default value is {region_id}/{app_id}/{ve rsion}.
	access_key_id	No	Indicates the custom account ID used to upload the application package to OSS.
	access_key_secret	No	Indicates the custom account key used to upload the application package to OSS.

Configuration example 1: Specify the group and the deployment package version

For example, take an application whose ID is eb20dc8a-e6ee-4f6d-a36f-5f6a54550453 is deployed in Beijing. The group ID is 06923bb9-8c5f-4508-94d8-517b692f30b9, and the deployment package version is 1.2.In this case. the configuration is as follows:

env: region_id: cn-beijing app: app_id: eb20dc8a-e6ee-4f6d-a36f-5f6a54550453 package_version: 1.2 group_id: 06923bb9-8c5f-4508-94d8-517b692f30b9

Configuration example 2: Specify an OSS storage space

For example, we want to deploy the application whose ID is eb20dc8a-e6ee-4f6d-a36f-5f6a54550453 and upload the deployment package to our own bucket named release-pkg in Beijing. The file object name is my.war, the OSS account ID is ABC, and the OSS account key is 1234567890.In this case, the configuration is as follows:

env: region_id: cn-beijing app: app_id: eb20dc8a-e6ee-4f6d-a36f-5f6a54550453 oss: region_id: cn-beijing bucket: release-pkg key: my.war access_key_id: ABC access_key_secret: 1234567890

Specify a configuration file

When no configuration file is specified, this plugin uses the .edas_config.yaml file in the root directory of the packaged project as the configuration file.If the packaged project is a submodule of the Maven project, the configuration file is under the root directory of the submodule by default but not the root directory of the entire Maven project.

You can also specify a configuration file by setting the -Dedas_config=xxx parameter.

If the default configuration file exists while another configuration file is specified, the plugin uses the latter.

Description of account configuration and priority

When deploying an application using this plugin, you must provide an Alibaba Cloud Access Key for application deployment.Currently, this plugin supports multiple configuration methods. If duplicate configuration methods are present, the configuration method with a higher priority overrides that with a lower priority.Configuration methods are as follows in descending order of priority:

ak/sk parameter specified in the command line: You can specify the access_key_id and access_key_secret in either of the following ways:

If you package the project by running Maven commands, specify both parameters with -Daccess_key_id=xx -Daccess_key_secret=xx.

When configuring this plugin in the pom file, configure both parameters as follows:

```
<plugin>
<groupId>com.alibaba.cloud</groupId>
<artifactId>toolkit-maven-plugin</artifactId>
<version>1.0.0</version>
<configuration>
<accessKeyId>abc</accessKeyId>
<accessKeySecret>1234567890</accessKeySecret>
</configuration>
</plugin>
```

Specify the account file in the command line (recommended): When you package the project by running Maven commands, specify the account file in yaml format with - Daccess_key_file={account file path.Example:

access_key_id: abc access_key_secret: 1234567890

Use the default Alibaba Cloud account file: If you choose not to specify an account in either of the preceding ways, the plugin uses the set Alibaba Cloud account to deploy the application.

aliyuncli: If you have used the latest aliyuncli tool and configured your Alibaba Cloud account, Alibaba Cloud generates the .aliyuncli sub-directory under the current Home directory and creates the credentials file in the .aliyuncli sub-directory to store your account information.Here, we use the Mac system as an example and assume the system user is "jack", so the following information is stored in the /Users/jack/.aliyuncli/credentials file:

[default] aliyun_access_key_secret = 1234567890 aliyun_access_key_id = abc

This plugin uses this account file as the account for deploying the application.

aliyun: If you have used a legacy Aliyun tool and configured the Alibaba Cloud account, the

Aliyun tool generates the .aliyun sub-directory under the current Home directory and creates the config.json file in the .aliyun sub-directory.Here, we use the Mac system as an example and assume the system user is "jack", so the account information is stored in the /Users/jack/.aliyun/config.json file:

{ "current": "", "profiles": [{ "name": "default", "mode": "AK", "access_key_id": "" "access_key_secret": "", "sts_token": "", "ram_role_name": "", "ram_role_arn": "", "ram_session_name": "", "private_key": "", "key_pair_name": "" "expired seconds": 0, "verified": "", "region_id": "", "output_format": "json", "language": "en", "site": "", "retry_timeout": 0, "retry_count": 0 }, { "name": "", "mode": "AK", "access_key_id": "abc", "access_key_secret": "xxx", "sts_token": "", "ram role name": "", "ram_role_arn": "", "ram_session_name": "", "private_key": "", "key_pair_name": "" "expired_seconds": 0, "verified": "", "region_id": "cn-hangzhou", "output_format": "json", "language": "en", "site": "", "retry_timeout": 0, "retry_count": 0 }], "meta_path": "" }

- **System environment variables**: Finally, the plugin attempts to retrieve the access_key_id and access_key_secret values from system environment variables.Namely, the plugin retrieves these values from System.getenv("access_key_id") and System.getenv("access_key_secret") in the Java code.
Use Eclipse to quickly deploy applications

Alibaba Cloud Toolkit for Eclipse (hereinafter referred to as Cloud Toolkit) is a free IDE plugin that helps users use Alibaba Cloud more efficiently.

You can register or use an existing Alibaba Cloud account to download Cloud Toolkit for free.After it is downloaded, you can install it to Eclipse.

After developing, debugging, and testing an application locally, you can use Cloud Toolkit to easily deploy the application to Alibaba Cloud.Currently, you can use Cloud Toolkit to deploy applications to ECS instances, EDAS, and Container Service for Kubernetes.

This topic describes how to install Cloud Toolkit in Eclipse and use it to quickly deploy an application to EDAS.

Prerequisites

- You have downloaded and installed JDK 1.8 or later.
- You have downloaded and installed Eclipse IDE, 4.5.0 (code: Mars) or later for Java EE developers.

Install Cloud Toolkit

Start Eclipse.

In the top navigation bar, choose **Help** > **Install New Software**.

In the **Available Software** dialog box that is displayed, set **Work with** to the URL http://toolkit.aliyun.com/eclipse/ of Cloud Toolkit for Eclipse.

In the Name column, select Alibaba Cloud Toolkit Core and Alibaba Cloud Toolkit Deployment Tools. In the Details area, clear Connect all update sites during install to find required software. Then, click Next.

Install				- 0 💌
Available Software				
Check the items that you wish to install.				
Work with: http://toolkit.aliyun.com/eclipse/				→ <u>A</u> dd
		Find more soft	ware by working with the <u>*Available Soft</u>	ware Sites [*] preferences.
type filter text				
Name	Version			
🔺 📝 💷 Alibaba Cloud Toolkit Core				
Alibaba Cloud Toolkit for Eclipse Core (Required)	0.0.2.v201809140302			
Alibaba Cloud Toolkit Deployment Tools Section 2018	0.0.2 v201809140302			
	0.0.2.1101000140502			
勾选				
Select All Deselect All 2 items selected				
Details				
				6
Show only the latest versions of available software		Hide items that are already installed		
Group items by category		What is already installed?		
Show only software applicable to target environment				
Contact all update sites during install to find required software	2			
	进			
e= 1*	A2			
3			< <u>Back</u> <u>N</u> ext > <u>Fir</u>	nish Cancel

Perform the subsequent installation steps as instructed on the Install page of Eclipse.

Note: In the installation process, a dialog box indicating no digital signature may appear. In such case, select **Install anyway**.

After installing Cloud Toolkit, restart Eclipse. The Alibaba Cloud Toolkit icon is displayed in the toolbar.



Configure a Cloud Toolkit account

To configure a Cloud Toolkit account, you must have an Access Key ID and an Access Key Secret.

Start Eclipse.

Click the drop-down arrow of the Alibaba Cloud Toolkit icon in the toolbar. In the dropdown list, click **Alibaba Cloud preference...**.

In the Preference (Filtered) dialog box, click Accounts in the left-side navigation pane.

Preferences (Filtered)		
type filter text 🖉	Accounts	
▲ Alibaba Cloud Tooll Accounts	AlibabaCloud Toolkit Preferences	
 Java Code Analys Regions 	Default Profile:	Add profile Remove profile
	Profile Details:	Sign up Manage existing Account
	<u>P</u> rofile Name:	
	(defined as a second seco	
	<u>A</u> ccess Key ID:	
	ground advertiges	
	<u>A</u> ccess Key Secret:	
	Show access key secret	
4 III •		Restore <u>D</u> efaults <u>Apply</u>
? (OK Cancel

On the Accounts page, set Access Key ID and Access Key Secret, and click OK.

If you have an Alibaba Cloud account, on the **Accounts** page, click **Manage existing Account** to go to the logon page of Alibaba Cloud.Log on to the Alibaba Cloud console. Go to the Security Management page to retrieve your **Access Key ID** and **Access Key Secret**.

If you do not have an Alibaba Cloud account, on the **Accounts** page, click **Sign up** to go to the registration page of Alibaba Cloud. Register an Alibaba Cloud account on this page.Then, retrieve your **Access Key ID** and **Access Key Secret** using the preceding method.

Deploy applications to EDAS

Currently, you can use Cloud Toolkit to deploy applications to EDAS with WAR or JAR packages.

In the **Package Explorer** left-side navigation pane of Eclipse, right-click the name of your application and choose **Alibaba Cloud** > **Deploy to EDAS...** from the shortcut menu.

In the **Deploy to EDAS** dialog box, select **Region**, **Namespace**, **Application**, and **Group** as needed, and then click **Deploy**.

Parameter description:

- Region: indicates the region where your application is located.
- Namespace: indicates the namespace where your application is located.
- Application: indicates the name of your application.
- Group: indicates the group to which your application belongs.

Note: If you have not created an application in EDAS, click **Create application on EDAS console...** in the upper-right corner of the dialog box to go to the EDAS console. Then, create an application.For more information about how to create applications, see **Deploy** Java applications in ECS clusters.

When the deployment process starts, the deployment logs are printed on the Eclipse Console tab page.You can check the deployment result based on the logs.

Stop Cloud Toolkit

If you want to stop the Cloud Toolkit when it is running, terminate the **EDAS-deploy** process on the **Progress** tab page.



Use IntelliJ IDEA to quickly deploy applications

Alibaba Cloud Toolkit for IntelliJ IDEA (hereinafter referred to as Cloud Toolkit) is a free IDE plugin that helps users use Alibaba Cloud more efficiently.

You can register or use an existing Alibaba Cloud account to download Cloud Toolkit for free.After downloading Cloud Toolkit, you can install it in IntelliJ IDEA.

After developing, debugging, and testing an application locally, you can use Cloud Toolkit to easily deploy the application to Alibaba Cloud.Currently, you can use Cloud Toolkit to deploy applications to ECS instances, EDAS, and Container Service for Kubernetes.

This topic describes how to install Cloud Toolkit to IntelliJ IDEA and use it to quickly deploy an application to EDAS.

Prerequisites

- You have downloaded and installed JDK 1.8 or later.
- You have downloaded and installed IntelliJ IDEA.

Install Cloud Toolkit

Run IntelliJ IDEA.

Install Cloud Toolkit to IntelliJ IDEA.

Mac system: On the **Preference** page, click **Plugins** in the left-side navigation pane. In the search box on the right, enter **Alibaba Cloud Toolkit**, and click **Install**.



Windows system: Click Plugins. In the search box, enter Alibaba Cloud Toolkit, and click Install.



After the Cloud Toolkit is successfully installed to IntelliJ IDEA, restart IntelliJ IDEA. The Alibaba Cloud Toolkit icon is displayed in the toolbar.

Configure a Cloud Toolkit account

To configure a Cloud Toolkit account, you must have an Access Key ID and an Access Key Secret.

Run IntelliJ IDEA.

Click the Alibaba Cloud Toolkit icon (C), and click **Preference...** in the drop-down list to go to the Settings page. In the left-side navigation pane, choose **Alibaba Cloud Toolkit** > **Accounts**.

E Settings		x
Qr	Alibaba Cloud Toolkit > Accounts	Reset
 Appearance & Behavior Alibaba Cloud Toolkit Accounts Regions Docker Keymap Editor Plugins Version Control Build, Execution, Deployment Languages & Frameworks Tools 	Accounts AlibabaCloud Toolkit Preferences DefaultProfile: Default Profile Details Profile Name: Sign up. Get existing AK/SK Default Access Key ID: Access Key Secret: Show access key secret	
		ply

On the Accounts page, set Access Key ID and Access Key Secret, and click OK.

If you have an Alibaba Cloud account, on the **Accounts** page, click **Get Existing AK/SK** to go to the logon page of Alibaba Cloud.Log on to the Alibaba Cloud console. Go to the Security Management page to retrieve your **Access Key ID** and **Access Key Secret**.

If you do not have an Alibaba Cloud account, on the **Accounts** page, click **Sign up** to go to the registration page of Alibaba Cloud. Register an Alibaba Cloud account on this page.Then, retrieve your **Access Key ID** and **Access Key Secret** using the preceding method.

Deploy applications to EDAS

Currently, you can use Cloud Toolkit to deploy applications to EDAS with WAR or JAR packages.

On IntelliJ IDEA, click the Alibaba Cloud Toolkit icon (C). In the drop-down list, click EDAS on Alibaba Cloud.

On the **Deploy to EDAS** page, configure the application deployment parameters. Then, click **Apply** to save the settings.

Deploy to EDAS	LABOSE	
<u>N</u> ame: edas		Share Allow running in parallel
EDAS Deployment Co	onfigurations	
		Create application on EDAS console
Application		
Region:	华北 2 (北京) ▼	
Namespace:	默认 🔻	
Application:	naitail 🔹	
Group:	All 🔻	
Build		
Deploy File:	 Maven Build Upload File 	(maven build added in Before launch automatically)
Advanced		
Description:		
▼ <u>B</u> efore launch: Ma	ven Goal, Activate tool window	
Show this page	Activate tool window	
?		Run Cancel Apply

- i. On the configuration page, select **Region**, **Namespace**, **Application**, and **Group** as needed.
 - i. **Region**: indicates the region where your application is located.
 - ii. Namespace: indicates the namespace where your application is located.
 - iii. **Application**: indicates the name of your application.
 - iv. Group: indicates the group to which your application belongs.
- ii. Sets the build mode.
 - i. **Maven Build**: When you select the Maven Build mode to build applications, a Maven task is added by default to build the deployment package.
 - ii. **Upload File**: When you select the Upload File mode to build applications, upload WAR or JAR packages to deploy the applications.

Note: If you have not created an application in EDAS, click **Create application on EDAS console...** in the upper-right corner of the dialog box to go to the EDAS console. Then, create an application.For more information about how to create applications, see **Deploy** Java applications in ECS clusters.

Click **Run** to run the configuration completed in the preceding step. Deployment logs are printed in the IntelliJ IDEA Console area.You can check the deployment result based on the

logs.

Manage Maven tasks

In Cloud Toolkit installed in IntelliJ IDEA, you can deploy Maven tasks.Additionally, you can add, delete, modify, and move Maven tasks in the **Before launch** area of the **Deploy to EDAS** page.

Deploy to EDAS	a difference in a	18 A. H. L. L.		0.4	L 9	×
<u>N</u> ame: edas				Share	Allow runnir	ng in <u>p</u> arallel
EDAS Deployment Con	ifigurations					
Application ———						
Region:	华北 2 (北京)					
Namespace:	默认					
Application:	nahaiC					
Group:	All	•				
Build						
Deploy File:	🖲 Maven Build 🛛 Up	oload File (mav	ven build a	dded in Befc	ore launch auto	matically)
Advanced						
Description:						
▼ <u>B</u> efore launch: Mave	en Goal (2), Activate tool w	indow				
+ - / 🔺 🔻						
M Run Maven Goal 'E	.das-Demo-Carshop: clea narket provider Maven V	n install' /ebapp: package'				
Show this page	Activate tool window					
				▶ Run	Cancel	Apply

In the **Select Maven Goal** dialog box, click the folder button on the right of Working directory and select all available modules for the current project. Enter the building command in **Command line**.



Deploy multi-module projects

Most of Maven projects involve multiple modules. These modules can be developed separately and some of them may use the functions of other modules. This type of project is a multi-module project.

If your project is a Maven multi-module project and you want to deploy a submodule in the project, make sure that the last Maven task in the **Before launch** area of the **EDAS Deployment Configurations**

page is built for the submodule.(For more information about how to manage Maven build tasks, see the Manage Maven build tasks section.)

For example, the current CarShop project has the following submodules:

- carshop

- itemcenter-api
- itemcenter
- detail

where itemcenter and detail are two submodules and depend on the itemcenter-api module. In this case, how is the itemcenter submodule deployed? In the **Before launch** area of the **EDAS Deployment Configurations** page, add the following two Maven tasks:

- 1. Add a Maven task to run the mvn clean install command in the carshop parent project.
- 2. Add a Maven task to run the mvn clean package command in the itemcenter submodule.

Note: Make sure that the Maven task for the itemcenter submodule is the last task in the **Before** launch area.

Application management

Namespaces

Overview

You can use namespace to isolate resources and services.

For example, you have three environments that are used separately used for development, testing and production. You can create three namespaces such as *Dev*, *Test* and *Prod* for the three environments. Then you can create clusters and deploy applications in these namespaces. The resources like ECS instances, applications and services are isolated with each other. The services cannot be invoked between different namespaces. Configurations can be pushed in only one namespacse.

Create namespaces

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Namespace**.

On the **Namespace** page, select **Region** and click **Create Namespace** in the upper-right corner.

In the **Create Namespace** dialog box, set **Namespace Name** (required), **Namespace ID** (required) and **Description**, and then click **Create**.

Note: The prefix of the namespace ID is determined by the specified region (target region) and cannot be modified.Rather, only the custom part can be modified.

Edit namespaces

On the Namespaces page, locate the row that contains the target namespace and click **Edit** in the Actions column.

In the **Edit Namespace** dialog box, edit the **Namespace Name** and description. Then, click **OK**.

Note: You cannot change the namespace ID and the namespace type.

Delete namespaces

The following conditions must be met before you can delete namespaces:

- No ECS instances exist in the namespace.
- No clusters exist in the namespace.

On the Namespaces page, locate the row that contains the target namespace and click **Delete** in the Actions column.

In the **Delete Namespace**dialog box, confirm the namespace to be deleted and click **Delete**.

Applications in ECS Cluster

Deploy applications in ECS clusters

In the ECS cluster, you can deploy applications through **WAR** packages or **JAR** packages, with similar parameters and operations.

You can deploy applications in the following scenarios:

- If you have deployed an application after creating it, you can upgrade its version through redeployment.
- If you just create an empty application, you can deploy it by following the instructions provided in this document.

After change process for deploying an application is completed, the application has been started without clicking **Start Application**. The **Start Application** button is unavailable untill the application is stopped.

You can use the EDAS console to deploy applications in an ECS cluster.

Configure ECS applications

This topic describes how to set the **JVM** and **Tomcat** parameters as well as **Basic Information** for applications in the ECS cluster.

Note: the JVM and Tomcat parameters are application parameters. That is, they are valid for the entire application. If JVM and Tomcat parameters are set by group on the **Instance Information** page, the priority of the parameters is higher than application-level settings.

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**.

On the **Applications** page, click the name of the target application.

On the **Basic Information** page, click **Settings** in the **Application Settings** area.

Set the JVM parameters

JVM parameters are container parameters that must be configured when an application is started.Correct configuration of JVM parameters helps reduce the overhead of garbage collection (GC) and thus shorten the server response time and improve throughput.If container parameters are not set, JVM parameters are allocated by default.

In the Application Settings dialog box, click the JVM tab.

Click **Memory Configuration**, **Application**, **GC Policy**, **Tool**, and **Custom** to set relevant parameters.Click **Save** to save the settings.

Note: Manually restart the application to make the JVM parameters take effect.

Set the Tomcat parameters

You can configure the following parameters for the Tomcat container: port number, application access path, and maximum number of threads.

In the Application Settings dialog box, click the **Tomcat** tab.

Set the Tomcat parameters and click Configure Tomcat.

Description of Tomcat parameters:

Configuration	Description
Application Port	The parameter value ranges from 1024 to 65535.The admin authority is used by container configuration and the root authority is required to operate ports with numbers less than 1024.Therefore, enter a port number greater than 1024.If this parameter is not set, the default value 8080 is used.
Tomcat Context	 Select an application access path: If you select the Package Name, you do not have to set the Custom Path parameter. The application access path is the name of the WAR package. If you select the Root Directory, you do not have to set the Custom Path parameter. The Path Path Path parameter. The Path Path Path Path Path Path Path Path

	application access path is /. - If you select Custom , you must set the Custom Path parameter.If the "Custom Path" parameter is not set, the default application access path is the same as the name of the WAR package.
Maximum Threads	Set the number of connections to the connection pool. The corresponding parameter is maxThreads, which is 400 by default.This parameter has significant implications for performance. We recommend that this parameter be set under professional guidance.
Tomcat Encoding	Select a Tomcat encoding format: UTF-8, ISO-8859-1, GBK, or GB2312.The default format is ISO-8859-1.

Set basic information

In the Application Settings dialog box, click the **Basic Information** tab, set the **Application Name** and **Application Description**, and then click **Modify**.

Scale out and scale in applications

Application scale-out or scale-in increases or decreases the computational capacity of an application by changing the number of ECS instances.You can add a new ECS instance when the ECS instance where an application resides is overloaded, or can remove the ECS instance when it is no longer required.You can also use **Auto Scaling** to dynamically adjust the number of ECS instances.

Scale out

When the ECS instance for hosting an application is overloaded, you can manually scale the application out.

Note: The running status of the ECS instance added depends on the running status of the application.

- If the application is running during scale-out, the ECS instance automatically deploys, starts, and runs the application.

- If the application is stopped during scale-out, the ECS instance automatically deploys but does not start or run the application.

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications** page, click the name of the target application.

On the Application Details page, click **Scale Out** in the upper-right corner.

In the Scale-Out Method dialog box, select Target Group for scale-out.

Select Scale-Out Method, and complete the subsequent scale-out steps.

Select from Current Cluster

Idle ECS instances in the cluster are added to the application for scale-out.

Select an ECS instance, and then click Scale-Out.

The message **Scale-out successful** is displayed on the page.

Go to the **Instance Information** page of the application to view the running status of the added ECS instance.

If **Normal** is displayed, the scale-out operation is successful.

Purchases based on current ECS specifications

This method allows you to select any ECS instance in the cluster as the specification template. Then, new purchases are configured according to the specifications of the template.

Select the ECS instance you want to use as the specification template, and click **Next**.

On the **Purchase Details** page, enter **Purchase Quantity** and **Login Key**, select **ECS Service Terms** | **Image Service Terms**, and then click **Next**.

On the **Confirm** page, view the information about the purchased ECS instance, and click **Confirm**.

On top of the page, the message **Automatic purchasing is triggered**. **Check the real-time information in the application change process.** is displayed after your submission.

Go to the **Instance Information** page of the application to view the running status of the added ECS instance.

If Normal is displayed, the scale-out operation is successful.

Note:

- The "Purchase based on current ECS specifications" method involves two change orders: the first is made by EDAS for your ECS instance purchase, while the other automatically adds the purchased ECS instance to the application.
- It takes about three minutes between the request submission and the start of application scale-out. The two change orders are executed at an interval of 10 seconds.
- The purchased ECS instance carries the same basic information as the selected one, including the instance specifications, disks, network, user data, and labels.
- All billing details are subject to normal ECS and EDAS charges. This operation itself does not generate any additional charges.
- The default logon information in the ECS instance is based on the key pair that you set. EDAS does not intrude on any of your private information.

Scale in

1. Log on to the Application scale-out or scale-in increases or decreases the computational capacity of an application by changing the number of ECS instances.You can add a new ECS instance when the ECS instance where an application resides is overloaded, or can remove the ECS instance when it is no longer required.You can also use **Auto Scaling** to dynamically adjust the number of ECS instances.

Scale out

When the ECS instance for hosting an application is overloaded, you can manually scale the application out.

Note: The running status of the ECS instance added depends on the running status of the application.

- If the application is running during scale-out, the ECS instance automatically deploys, starts, and runs the application.

- If the application is stopped during scale-out, the ECS instance automatically deploys but does not start or run the application.

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications** page, click the name of the target application.

On the Application Details page, click **Scale Out** in the upper-right corner.

In the Scale-Out Method dialog box, select Target Group for scale-out.

Select Scale-Out Method, and complete the subsequent scale-out steps.

Select from Current Cluster

Idle ECS instances in the cluster are added to the application for scale-out.

Select an ECS instance, and then click Scale-Out.

The message **Scale-out successful** is displayed on the page.

Go to the **Instance Information** page of the application to view the running status of the added ECS instance.

If **Normal** is displayed, the scale-out operation is successful.

Purchases based on current ECS specifications

This method allows you to select any ECS instance in the cluster as the specification template. Then, new purchases are configured according to the specifications of the template.

Select the ECS instance you want to use as the specification template, and click **Next**.

On the **Purchase Details** page, enter **Purchase Quantity** and **Login Key**, select **ECS Service Terms** | **Image Service Terms**, and then click **Next**.

On the **Confirm** page, view the information about the purchased ECS instance, and click **Confirm**.

On top of the page, the message **Automatic purchasing is triggered**. **Check the real-time information in the application change process.** is displayed after your submission.

Go to the **Instance Information** page of the application to view the running status of the added ECS instance.

If **Normal** is displayed, the scale-out operation is successful.

Note:

- The "Purchase based on current ECS specifications" method involves two change orders: the first is made by EDAS for your ECS instance purchase, while the other automatically adds the purchased ECS instance to the application.
- It takes about three minutes between the request submission and the start of application scale-out. The two change orders are executed at an interval of 10 seconds.
- The purchased ECS instance carries the same basic information as the selected one, including the instance specifications, disks, network, user data, and labels.
- All billing details are subject to normal ECS and EDAS charges. This operation itself does not generate any additional charges.
- The default logon information in the ECS instance is based on the key pair that you set. EDAS does not intrude on any of your private information.

Scale in

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications** page, click the name of the target application.

On the Application Details page, click the Instance Information tab.

On the Instance Information tab page, delete the ECS instance.

If the ECS instance is running, click Stop and then click Delete.

If the ECS instance is stopped, click **Delete**..

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications** page, click the name of the target application.

On the Application Details page, click the **Instance Information** tab.

On the **Instance Information** tab page, delete the ECS instance.

If the ECS instance is running, click **Stop** and then click **Delete**.

If the ECS instance is stopped, click **Delete**.

Manage instance groups

Overview

This function groups all ECS instances for an application so that you can deploy different application package versions for the ECS instances in different groups.

For example, there are 10 instances in the "itemcenter" application, which are divided into two groups: "Default Group" and "Beta Group". The default group contains six instances and the Beta group contains four.Now there are two groups of instances in the application to which you can deploy different versions of the application package.

Note:

- 1. When an application is created in EDAS, a new group "Default Group" will be created by default for the application, and cannot be deleted.
- 2. If no multi-version deployment is required, the default group is generally adequate.

Create groups

Gated release is often used for the launch of new application versions, so that the new version can be test run without affecting the traffic in the production environment. In this case, you need to create a new group for the application.

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications** page, click the name of the target application.

On the Application Details page, click the **Instance Deployment Information** tab, and click **Create Group** in the upper-right corner of the page.

In the Create Group dialog box, enter the Group Name and then click Create.

After you successfully create a group, the message **Group created successfully** is displayed in the upper-right corner of the page.

Add instances

After creating a group, you can add instances to it through **Scale Out** and **Change Group**. Perform the following operations:

Add instances to the group through **Scale Out**. For details, see **Scale out and scale in** applications (ECS cluster).

Add instances to the group through Change Group

Click the **Instance Information** tab on the **Application Details** page, select the instance whose group you want to change and click **Change Group** on the right of the list.

In the Change Group dialog box, select Target Group, and then click OK.

If the application package version for the instance is different from that of the target group, use the version for the target group or keep the existing version for the instance.

Notes about changing instance groups

- If no package version is available for the target group and a package is already deployed to the instance whose group you want to change, the package version for the instance will be used as the package version for the group.
- Choose **Redeploy current instance for target group** to redeploy the deployment package on the instance using the version in the group.
- If you select **Change Group Only Without Re-Deployment**, the deployment status of the instance remains unchanged.
- If the package version for the instance is different from that of its group, a

prompt is displayed.

View groups

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications page**, click the name of the target application.

On the Application Details page, click the **Instance Information** tab to check the application group information and deployment package versions for different groups.

Notes about groups:

- Each group corresponds to one deployment package version, which is displayed following the group name.
- No package version is available for a new group. The package version for a group is always the version of the package last deployed to the group.
- Application instances are displayed by instance group.

Set group parameters

On the **Application Details** page, click the **Instance Information** tab and click the **Group Settings** button on the right of the group.

On the **Group Settings** page, click **JVM** or **Tomcat**, set JVM and Tomcat parameters, and then click **Save** or **Configure Tomcat Parameters**.

JVM and Tomcat parameters are described in Application settings (ECS cluster).

Delete groups

A group with no instances in it can be deleted. The delete operation cannot be undone. Exercise caution when performing this operation.

On the Application Details page, click the **Instance Information** tab and click the **Delete Group** button.

In the **Delete Group** dialog box, click **Delete**.

Container version management

An EDAS container consists of AliTomcat, Pandora, and custom Pandora plugins. In addition to the support for existing Apache Tomcat core functions, a class isolation mechanism, QoS, and Tomcat-Monitor are provided.Besides, highly customized plug-ins are added to EDAS containers to implement complex and advanced functions, such as container monitoring, service monitoring, and tracing.Applications deployed using EDAS must run in EDAS containers.

Basic concepts

AliTomcat

AliTomcat is a version of Apache Tomcat for internal use, which is developed by the Alibaba middleware team after performance optimization, bug fixing, and new feature development based on Apache Tomcat.AliTomcat is widely deployed and used in Alibaba Group, and offers enhanced performance, security, and stability compared with the community version.

Pandora and Pandora plugins

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

Container version

You must select the container version when creating an application on EDAS.EDAS containers are maintained and published by the EDAS development team.You can choose **Application Management** > **Software Version** to check the history and description of container publishing. Alternatively, you can view **Container version description**.Generally, a container of a higher version is superior to a container of a lower version in stability and functions variety.

The publishing of an EDAS container does not affect deployed applications. Once a new container is available, you can immediately upgrade the container to the latest version.

Upgrade or downgrade containers

In the left-side navigation pane of the EDAS console, choose **Application Management** > **Applications**. On the **Applications** page, click the name of the target application.

On the Application Details page, click Container Version in the left-side navigation pane.

On the **Container Version** page, click **Upgrade to This Version** or **Downgrade to This Version** on the right side of the row corresponding to the container to upgrade or downgrade the container with a single click.

Stop and start applications

After an application is released, you can stop and start the application on the Enterprise Distributed Application Service (EDAS) console.

Stop the application

Log on to the EDAS console.

Click **Applications** in the left-side navigation pane, and click an application on the Applications page to go to the **Basic Information** page.

When you click **Stop**, the application is stopped.

Start the application

Common applications and Docker applications are released upon deployment. You do not need to start them.Therefore, you only need to restart it after it is stopped.

Log on to the EDAS console.

Click **Applications** in the left-side navigation pane, and click an application on the Applications page to go to the **Basic Information** page.

When you click Start, the application is restarted.

Roll back ECS application

After updating an application, you can roll it back to an earlier version if any problem is detected.

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the Applications page, click the name of the target application.

Click Roll back application in the upper-right corner of the Application Details page.

In the **Roll Back Application** dialog box, select a **deployment package version** and **group**, set the **batch** and **batch mode**, and click **Roll Back**.

Delete applications

After an application is deleted, all information relating to the application is deleted, all instances under the application are released, and all WAR packages and container files are deleted from the ECS instance.

Note: Before deleting an application, disable the application and ensure that the logs, WAR packages, and configurations of all instances under the application are backed up.

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the Applications page, click the name of the target application.

On the Application Details page, click **Delete Application**.

Delete the application as instructed on the page.

After you delete the application, the Application deletion triggered successfully message is

displayed in the upper-right corner of the page.

View application changes

After lifecycle operations in the EDAS console, such as application deployment, starting, scale-out or scale-in, you can go to the **Application Details** page to check the change status or go to the **Change Logs** page to check historical change logs.

View application changes

The following uses an example of application deployment to describe how to view application changes.

Return to the Application Details page.

At the top of the **Application Details** page, the message **A change order is being executed** is displayed.

Click **Details**. On the **Change Details** page, check change information and real-time status of the application.

The page contains two areas: change summary and change process execution information.

- Change summary: includes the change order ID, execution time, batch number, and batch mode (manual or automatic trigger).

Change process execution information: includes each stage of the entire process.For example, *Process Start, SLB Offline, Deploy, SLB Online*, and *Process Complete* are included.The execution results of each stage are identified by icons.

- \checkmark : indicates that the step is completed successfully.
- Running: indicates that the step is being executed.
- X : indicates that the step failed.Click View Details to check specific information and locate failure causes.

Deploy is a virtual stage.

Click **View Details** under **Deploy** to check the execution process and stages of an application.

Note: Host indicates application instance information.

Click **View Log** under a specific stage of the application instance to check task execution logs.

Click a task, such as *Start Application Instance*. The task execution log is displayed in the right-side area.

View application change logs

On the **Application Details** page, click **Change Logs** in the left-side navigation pane to check the logs of all application changes.

You can click **View** in the **Actions** column to check change orders and details about actions.

Health check

In the health check process, EDAS Agent periodically checks and reports the status of containers and applications and then sends the results to the console.Health checking can help you understand the overall condition of service operation in the cluster environment, and facilitates auditing and troubleshooting.You can configure a health check URL on the EDAS console to check whether the deployed applications are running properly.

If your business is highly sensitive to the traffic load, frequent health checking may impact normal service.You can reduce this impact by reducing the frequency of health checking or increasing the health check interval.To guarantee the service availability, we recommend that you do not disable health checking.

Health check process

You can configure health checks to monitor end nodes specified by IP address or domain name.EDAS health check automatically submits requests to your applications, servers, or other resources at a fixed interval you specify to verify whether it is accessible, available, and functional.You can also make a request similar to a user' s request by configuring a health check URL to verify the running status of the application features.

A health check is triggered every 10 seconds. The details of Steps 1 and 2 in the figure are as follows:

EDAS Agent checks whether the Ali-Tomcat process for running your application is alive.

If the process is alive, EDAS Agent proceeds to Step 2.

If the process is not alive, the health check ends and the check result is "Fail" .

EDAS Agent checks whether status code 200 is returned for the set URL.

If no URL is configured, health checking stops. If you set a URL, EDAS Agent checks whether the HTTP code 200 is returned for the set URL.

Create, update, and delete health checks (console)

In the EDAS console, you can view the running status of applications ("Normal" or "Runtime Error") and health checking process reports. For all health checks, except calculated health checks, you can also view the reasons for failure of the last health check.

You can create a health check URL when creating an application. You can also add or modify health checks on the **Application Settings** page after an application has been created and deployed.For ECS applications and Swarm applications, you can also click the **Settings** button on the **Application Settings** page, and set the Health Check parameters.

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications** page, click **Create Application** in the upper-right corner.

In the **Create Application** dialog box, enter application information and click **Next Step: Application Configurations**.On the **Application Configuration** page, click **Deployment Method** and follow the instructions on the page.On the **Application Configuration** page, you can create *Application Health Check**.

Note: The port range is 1 to 65535.

应用健康检查: http://127.0.0.1:8080/healthCheck.html

Example: When deploying a WAR package, you can set the health check URL to http://127.0.0.1:8080/order/healthCheck.html if no other container parameters are configured. Also, you can set the health check URL to

http://127.0.0.1:8081/healthcheck.html if the container path is set to the root path, the port is set to "8081", and the WAR package contains the "healthcheck.html" file used for health status identification.

After creating a health check, you can view the settings of the health check in the **Application Settings** area on the Application Details page.You can also click **Modify** to modify or delete the health check settings.

On the Application Details page, click the **Settings** button to the right of the **Application Settings** area for ECS applications and Swarm applications. In the displayed Application Settings dialog box, click the **Health Check** tab. On the tab page that is displayed, follow the instructions in Step 3 to add, modify, or delete health check settings.

Description of health check status:

After you configure the application health check, the application displays different statuses.

Container Exited: displayed if EDAS Agent detects that the Ali-Tomcat process is not alive in Step 1 of the health check flow chart.

Application Exception: displayed when any code other than 200 is returned for the URL set in step 2 of the health check flow chart.

Normal: displayed if no exception occurs in steps 1 and 2 of the health check flow chart.

If EDAS Agent detects that no URL is set, the Normal state followed by an exclamation mark is displayed. When you move the cursor over the status, the prompt "Please configure the application's health check URL so that its running status can be reflected more accurately" is displayed."

Agent Exception: displayed if EDAS Agent does not report status information to the EDAS server within 30 seconds.

Log directories

The EDAS console allows you to view application runtime logs without having to log on to the server. When an exception occurs, you can check logs to troubleshoot the problem.

Follow these steps to view a runtime log:

Log on to the EDAS console. Choose Applications Management > Applications in the left-

side navigation pane.

On the **Applications** page, click the name of the application you want to check to go to the application details page.

In the left-side navigation pane, select **Logs** > **Log Directories**. Alternatively, you can also go to the **Instance Information** tab of the application details page, and click **Logs** in the **Actions** column.

On the Log Directories page, 3 log directories are displayed by default:

Tomcat container logs directory: The specific paths for Tomcat container logs depend on its actual version.

EDAS Agent logs directory.

Log files for log framework configurations.

Runtime Logs
Application Log File:
/home/admin/taobao-tomcat-production-7.0.59.3/logs/
+ catalina.out(0 bytes)
+ catalina.out.2018-01-11(15 KB)
+ catalina.log.2018-01-10(9 KB)
+ hsf.lock(0 bytes)
+ localhost.log.2018-01-10(125 bytes)
+ localhost.log.2015-10-27(0 bytes)
📁 /home/admin/edas-agent/logs/
+ agent.log(10 MB)
+ std.log(10 KB)
+ servicemetadata.log(0 bytes)
+ agent0.log(0 bytes)
Log files configured inside log framework (open to extract)

Note: Only readable files are displayed in the file directory. No folders are displayed.

Double-click a log file to view the details of the log.

Time Log & Back to Log List ECS Instance ID/Name/IP:	Number of Contract	Ŧ		
Name: /home/admin/taobao-tomcat-production-7.0.59.3/logs/catalina.log.2018-01	1-10			
Please enter the search text.	Matching Mode: Case Insensitive			
Search Towards File Header Search Towards File End Q Search				
H4 File Manuface File Location:		0.8 / 0.09 //8	Mar a l	
R File Header File Cocadon		0 D7 5.00 KD	PP File End	
		0 b / 5.00 kb	Hie End	
message		0 D / 9.00 KD	MM File End	
message 2018-01-10 21:58-03,078 com.taobao.tomcat.digester.ServerListenerCreateRule s	shouldIgnore	0 D / 5.00 KD	199 File End	
message 2018-01-10 2158-03,078 com.taobao.tomcat.digester.ServerListenerCreateRule s WARNING: found <listener <="" dassname="com.taobao.tomcat.monitor.MonitorServer" td=""><td>shouldfgnore rviceListener"/> in server.xml, ignore</td><td>0.07 5.00 KD</td><td>PM Hile End</td><td></td></listener>	shouldfgnore rviceListener"/> in server.xml, ignore	0.07 5.00 KD	PM Hile End	
message 2018-01-10 21:58:03,078 com.taobao.tomcat.digester: ServerListenerCreateRule S WARNING: fond -Clustener className="com.taobao.tomcat.digester: ModuleServiceCreateRule b 2018-01-10 21:58:03,097 com.taobao.tomcat.digester: ModuleServiceCreateRule b	shouldIgnore viceListener"/> in server.xml, ignore begin	00, 300 KD	M Hie End	
message 2018-01-10 21158-03,078 com.tabdao.tomcat.digester.ServerListenerCreateRule s WARNING: Fond <listener classname="com.tabdao.tomcat.digester.ServerListenerCreateRule s
WARNING: fond <Listener className=" com.tabdao.tomcat.digester.moduleservercreaterule="" s<br="">WARNING: round <abduleserverf form="" ill,="" pore<="" server-ani,="" td=""><td>shouldIgnore viceListener"/> in server.xml, ignore begin</td><td>0 D / 330 KD</td><td>M Hie End</td><td></td></abduleserverf></listener>	shouldIgnore viceListener"/> in server.xml, ignore begin	0 D / 330 KD	M Hie End	
message 2018-01-10 21:58:03,078 com.taobao.tomcat.digester: ServerListenerCreateRule & WARNING: fond <1.istener dassName="com.taobao.tomcat.digester: ModuleServiceCreateRule & 2018-01-10 21:58:03,227 com.taobao.tomcat.digester: ModuleServiceCreateRule & WARNING: fond <1boduleService/> from server.xml, grant 2018-01-10 21:58:03,222 com.taobao.tablan.statuu_ht/ststConfigRule mapping	shouldignore viceListener"/> in server.xml, ignore begin	0 D / 3300 KD	79 Hee End	ĺ
Martine indust Martine in the account in the account of t	shouldIgnore viceLastener"/> in server.xml, ignore begin _AllHostConfig"> in server.xml, ignore	0 D / 3300 KB	M He End	
message 2018-01-10 21:58:03,078 com.taobao.tomcat.digester.ServerListeneCreateRule 6 WARNING: fond <listener 6<br="" classname="com.taobao.tomcat.digester.ServerListeneCreateRule 6
WARNING: fond <Listener className=" com.taobao.tomcat.digester.moduleservicecreaterule="">WARNING: fond <listener< td=""><td>shouldIgnore viceListene"/> in server.xml, ignore begin -AllHostConfig"> in server.xml, ignore</td><td>0 0 / 300 KD</td><td>M He End</td><td>ĺ</td></listener<></listener>	shouldIgnore viceListene"/> in server.xml, ignore begin -AllHostConfig"> in server.xml, ignore	0 0 / 300 KD	M He End	ĺ
menual	shouldIgnore viceUsterer'/> in server.xml, ignore Degin -AliHostConfig"> in server.xml, ignore	0 0 / 300 KD	M ne End	
THE Product Proceedings Proc	shouldIgnore viceListener'/> in server.xml, ignore begin AliHostConfig"> in server.xml, ignore	0 0 7 3 00 KD	W ne End	
Martine indust Processing 2018-01-10 21:58-03,078 com.taobao.tomcat.digester.ServerListenerCreateRule is WARNING: found Standard Com.taobao.tomcat.digester.ServerListenerCreateRule is WARNING: found Standard Standard	shouldIgnore viceLatener'/> in server.xml, ignore Degin -AilHostConfig"> in server.xml, ignore	0 0 / 300 KD	W ne End	ĺ
The Preside The Second	shouldIgnore viceListener'/> in server.xml, ignore begin .AliYostConfig"> in server.xml, ignore	0 0 7 3 00 KB	W ne End	ĺ
Martin Peaked	shouldIgnore viceLatener'/> in server.xml, ignore begin -AilHostConfig"> in server.xml, ignore		W ne End	ĺ

At the top of the page, select an instance ID or name in the drop-down list next to **ECS Instance ID/Name/IP Address** to view the details.

At the bottom of the page, click **Enable Real-Time Additions** to keep loading the latest additions to the log file (similar to the tailf command).

In addition to checking the logs in the default path, you can also add log paths to favorites for later viewing, or remove a path from your favorites.

Bookmark a directory

On the Log Directories page, click Bookmark Log Directory to add a log directory.

Note: This path must be under the /home/admin directory, and must contain wordings "log" or "logs" in the complete path. The file must end with a slash "/ " to indicate that it is a folder.

Remove from bookmark

On the **Log Directories** page, click to select a folder name. Then click **Remove Directory from Bookmark** at the top right corner of the page. The path will no longer be displayed on the page. This operation does not delete or change any files on the server.

Alarm and notification

EDAS provides the alarm function to notify users of online problems when resource usage exceeds

the limit. Based on policies configured by users and data collected in the background, EDAS checks whether the resource usage limit is exceeded. If the limit is exceeded, a text message or an email is sent to specified contacts.

Note: Currently, EDAS only provides SMS and email notification and does not support custom notification.

Configure alarm policies

Follow these steps:

Log on to the EDAS console, select **Applications Management** > **Applications** in the left-side navigation pane, and click the name of the application from the application list.

Select **Alarm and Notification** > **Alarm Rules** in the left-side navigation pane and click **Create Rule** in the upper-right corner.

Enter relevant information in the Create Rule page.

*Rule Name:	Rule names must only contain numbers, letters and underscores.
*Monitoring Target:	Monitoring Metrics Compare Threshold Actions
	CPU Usage v > v %
	+ Add Monitor Items
*Trigger Conditions:	Any One o 💌
*Statistical Cycle:	5 Minutes 💌
*Retries Before Alarm:	1 •
	OK Cancel and Return to Rule List

Field description:

Rule Name: Name of the alarm rule, which can contain numbers, letters, and underscores (_). Use an understandable name.

Monitoring Target: Create comparison policies based on metrics (basic monitoring, HTTP, HSF, and application container) and the configured threshold. You can add more than one target as needed.

Trigger Conditions: Select Any One of the Indicators or All Indicators.

• Any One of the Indicators: An alarm is triggered when any of the indicators of the monitored object meets the alarm rules.

• All Indicators: An alarm is triggered when all of the indicators of the monitored object meet the alarm rules.

Statistical Cycle: It can be set to 1 minute, 5 minutes, 15 minutes, 30 minutes, or 1 hour. A false alarm might be generated when the system encounters transient jitter, for example, when CPU usage is high during service startup but recovers to the normal range within 2 minutes. To avoid false alarms, you can select a statistical period to allow alarm trigger only when **alarm rules are continuously satisfied** within this period. For example, if you select the 5-minute statistical period for the metric "CPU usage above 30%", then EDAS determines an exception occurs when the CPU usage of the system exceeds 30% for 5 **consecutive** minutes.

- **Retries Before Alarm**: The number of consecutive statistical cycles when alarm policies are satisfied that are required to trigger an alarm. Optional values include 1, 3, and 5.

Click OK.

Alarm rules take affect once created. To disable an a rule, select it from the rule list and click "Delete" . The rule is obsolete immediately.

Add alarm contacts

Follow these steps:

- 1. Log on to the EDAS console, select **Applications Management** > **Applications** in the left-side menu bar, and click the name of the application in the application list.
- 2. Select Notification Alert > Alarm Contacts on the left-side menu bar and click Add Alarm Contacts in the upper-right corner.
- 3. Select the contact from the contact list and click OK.

Note:

- Alarm Contact Source: You can configure to send alarms to the contacts that have a primary and sub-account relationship with the current account. Details are as follows:
 - Other Alibaba Cloud accounts that are bound as sub-accounts to the primary account
 - RAM sub-accounts with EDAS logon history
- **Contact Info (Email Address and Mobile Number)**: By default, emails and contacts are obtained from Alibaba Cloud. For privacy protection, the contact information is only available after logon. If you want to receive notifications using a mobile number other than the one registered at Alibaba Cloud, make modifications on **Personal Information** page.

Add employees as alarm contacts

To add an employee that has never used EDAS as an alarm contact, following these steps. Assume that the current EDAS primary account is master@alibabacloud.com and the account to be added is employee@company.com:

Add a RAM sub-account:

Log on to the Alibaba Cloud console with the account master@alibabacloud.com and select **Products & Services** > **RAM** to go to the **RAM Console**.

Click **Users** in the left-side navigation pane to go to the RAM sub-account page, click **Create User** in the upper-right corner, and fill in employee information to create a sub-account (assume that the employee name is "employee").

Log on to EDAS with the sub-account and modify information.

Log on to Alibaba Cloud with the employee RAM sub-account by clicking the link provided in RAM, and select **EDAS** to go to the EDAS console.

Select **Accounts** > **Personal Information** in the left-side navigation pane and enter your mobile number and email address.

After relevant information is modified, follow the steps in the **Add alarm contacts** section to add the employee to the alarm contact list.

View alarm records

After an alert is generated, the system sends the alert to contacts while recording the alert.

- 1. Log on to the EDAS console, select **Applications Management** > **Applications** in the left-side navigation pane, and select the expected application from the application list.
- 2. Select Alarm and Notification > Alarm Records on the left-side menu bar.

Alarm records from the past 10 days are displayed. After an alarm is cleared, a notification is generated and sent to contacts by means of text message and email.

Rate limiting and degradation

Overview of rate limiting and degradation

The rate limiting and degradation feature of EDAS resolves slow system responses or crashes caused by the high pressure of the backend core services. This feature is generally used in high-traffic scenarios, for example, flash sales, shopping sprees, great promotions, and anti-empty box scams.

Rate limiting

This feature is used to control the traffic threshold or adjust the ratio. When a frontend website encounters high-traffic access, the traffic is controlled to prevent service unavailability and damage to the backend core system.By adjusting the traffic threshold, the maximum traffic volume of the system is controlled to ensure secure and stable system running.

Basic principles

After a rate limiting module code is configured for a service provider and a rate limiting policy is configured on EDAS, the service provider can use the rate limiting function. When a service consumer calls the service provider, all access requests are calculated by the rate limiting module. If the call volume of the service consumer exceeds the preset threshold in a specific period, the rate limiting policy is triggered.



Degradation

Degradation is to lower the called priority of non-core service providers that are timed out to ensure the availability of core service consumers.

Basic principles

After a degradation module code is configured for a service consumer and a degradation policy is

configured on EDAS, the degradation function is enabled for service consumers. When the service consumer calls a service provider, if the response time of the service provider exceeds the preset threshold, the degradation policy is triggered.



Configuration management

Application Configuration Management (ACM) has been integrated in EDAS as a component. You can use ACM in EDAS console to centrally manage and push configurations for applications, including Create configuration, View push status, Push track query and Query and rollback versions.

Auto scaling

To ensure the service quality and availability of an EDAS distributed cluster, it is crucial to introduce O&M capabilities that can detect the status of each server in the cluster and can scale the cluster in or out in real time based on the system load.

If the ECS instances in your cluster are insufficient for scale-out, use the Alibaba Cloud Auto Scaling feature to create hosts (you must use EDAS to activate Alibaba Cloud Auto Scaling in advance). You are charged in pay-as-you-go mode for hosts created through elastic resources.

EDAS provides the auto scaling function to automatically scale up or down a cluster based on the CPU, RT, and load metrics of the cluster servers.

Metric descriptions:

- **CPU**: CPU usage of the server in percentage. If multiple servers exist in the application, the average value of all servers is used.
- RT: Time for the system to respond to a request in ms.
- Load: System load, which is a positive integer.

These metrics must be positive integers without floating-point numbers. If multiple servers exist in the application, the average values of all servers are used for all the preceding metrics. Auto scaling includes automatic scale-in and scale-out, for which the rules can be configured separately.

Automatic scale-out

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications** page, click the name of the target application.

On the Application Details page, click **Auto Scaling** in the left-side navigation pane.

Click the switch to the right of **Scale-out Rule** to enable scale-out rules.

Configure the scale-out rule parameters, and then click Save.

Instance Source (for applications other than Docker)

Existing Resources: uses only the idle hosts in the cluster for automatic scale-out.

Elastic Resources: uses only the hosts created by Auto Scaling for automatic scale-out.

Existing Resources First: uses the idle hosts in the cluster first for automatic scale-out. If the cluster does not have sufficient idle hosts, the hosts created with elastic resources are used.

Note: If you select Elastic Resources or Existing Resources First, the hosts created by Alibaba Cloud Auto Scaling may be used, and in this case you are charged in pay-as-you-go mode.

In addition, you need to set the following parameters for the hosts created with elastic resources:

- **Specifications Template**: uses one of the existing hosts in the cluster as a template for automatic scale-out. Based on the template, new hosts inherit the CPU, memory, network, disk, and security group settings.
- Network Type and Multi-Zone Scaling Policy: Network Type indicates the network
where the current application to be scaled out is located and cannot be changed. If the current network is Virtual Private Cloud (VPC), you must specify one or more virtual switches for the new host. If you specify two or more virtual switches, EDAS automatically allocates these switches through the **Multi-Zone Scaling Policy**.

Login Password: This password is used as the administrator (root user) password for a new host.

Trigger Indicators: includes the thresholds of CPU, RT, and Load indicators.Scaleout is triggered when a threshold is exceeded.

Trigger Condition

Any Metric: Automatic scale-up is implemented if any of the set metrics is triggered.

All Metrics: Automatic scale-up is implemented only when all of the set metrics are triggered.

Lasts for More Than: indicates the duration for which the indicators are triggered continuously, in minutes.Within the duration, if the average value of an indicator per minute continuously reaches the set threshold, scale-out is triggered. You can configure the duration based on the sensitivity of the cluster service capabilities.

Number of Instances for Each Scale-Out: indicates the number of servers automatically added upon each scale-out. You can set this parameter based on the service capabilities of a single server of the application.

Maximum Number of Instances: When the number of ECS instances in the cluster reaches the maximum, no more scale-out can be performed. You can set this parameter based on the resource quota.

Automatic scale-in

The **Automatic Scale-In** configuration process is similar to the **Automatic Scale-Out** configuration. For the definitions and configuration methods of the metrics, see **Automatic scale-out**.

Note:

- When you configure both scale-out and scale-in rules, the metrics of the scale-in rules cannot be greater than those of scale-out rules. Otherwise, an error message will be displayed when you click **Save**.

- If elastic resources are used, the hosts created with elastic resources are released first during scale-in.

View auto scaling results

After auto scaling rules have been set, when an automatic scale-in or scale-out is performed, you can check whether the number of ECS instances has changed in **Instance Information** on the **Basic Information** page.

Application monitoring

Application monitoring overview

Application monitoring can accurately reflect the real-time traffic and history information of an application, allowing you to monitor the application health and quickly pinpoint issues.

- You can use EagleEye provided by EDAS for the infrastructure monitoring and service monitoring of applications, and view the dashboard.
- If you have enabled the advanced monitoring service, you can use ARMS to monitor applications.
- Meanwhile, for applications deployed in the native Spring Cloud or Dubbo framework later than x x, xxxx and applications deployed in Container Service Kubernetes clusters, ARMS is used for monitoring by default.

Glossary

TraceId: corresponds to a request. It is globally unique and transmitted between systems.

IP Address: indicates the IP address (hexadecimal) of the ECS instance that creates the TraceId.

Creation Time: indicates the link creation time.

Order: used for link sampling.

Flag Bit (optional): used for debugging and identification.

Process ID (optional): used for single-instance multi-process applications.

RpcId: calls and identifies the log track order and nesting relationships. This item is transmitted between systems.

Service Dimension: Service data is monitored in the application and service dimensions. Data in the application dimension is aggregated by application, while data in the service dimension is aggregated by custom service. For example, application A provides services a, b, and c.

Drill Down: views the metrics of upstream (downstream) applications associated with the target metric.

Types of monitoring data

The Service Monitoring page provides the tabs for different data types, enabling pertinent monitoring.

- **RPC Call Overview**: displays the RPC services (including the HSF and other customer services) that are provided by an application as the server.
- **RPC Call Source**: displays the records of the following applications that call the RPC services provided by the current application.
- **RPC Call Dependency**: displays the records of the current application that calls the RPC services (including HSF and other custom services) provided by other applications.

Types of monitoring reports

- **Mix of Graph and Table (Default)**: displays data in the "table + graph" combination, including the monitored target, time, QPS, response time, server response time, errors, and results. By default, the graph shows the data for the last hour, and the table lists the data for the last five minutes.
- **Multi-graph**: displays data in a graph, including the monitored target, time, QPS, response time, errors, and results. By default, this graph shows the data for the last hour and separately lists the latest data.
- **Table**: displays data in a table, including the monitored target, QPS, response time, errors, and results. The table lists the data for the last minute.

Metric descriptions

- **Errors per Second**: records the RPC error rate per minute, which is the total number of errors that occurred within the minute divided by 60.

- **Results per Second**: records the returned result in "Result: QPS" format, where the "Result" indicates the RPC result. The HTTP result is consistent with the HTTP ErrorCode.

Dashboard

Based on different groups, the monitoring dashboard displays the overall metrics related to provided services, service consumption, and infrastructure monitoring using charts.

Service provided: Displays information about RPC services and HTTP services provided by the application.

Service consumption: Displays the database access metrics.

Infrastrucure monitoring: Displays the metrics about CPU, load, memory, disk, and network.

Follow these steps to view the monitoring dashboard.

Log on to the EDAS console.

Click **Applications** on the left-side menu bar.

In the application list, click the application name which you want to view information about.

On the application details page, select **Application Monitoring** > **Dashboard** from the leftside menu bar.

The page shows information about the service provided, service consumption and infrastructure monitoring.

Hover the mouse over a point on an abscissa of a monitoring chart to view the information and status data at a specific time point.

Click a project name, "RPC Service" for example, at the top of a monitoring chart to switch to the service monitoring page and view details. For details about the monitoring parameters, see Application monitoring overview.

Infrastructure monitoring

EDAS collects data from the ECS instance that runs applications and provides the single-instance and cluster views of the CPU, memory, load, network, and disk metrics based on the analysis results.Data in all monitoring views is collected and processed in the units of applications.

Note:

- Due to the latency between data collection and data analysis, EDAS cannot provide real-time monitoring views. It currently has a latency of two minutes.
- For Kubernetes applications, if pods are changed upon upgrades or scaling, breakpoints occur and the monitoring data becomes discontinuous.

Perform the following steps to view a cluster or single-instance statistical view:

Log on to the EDAS console.

In the left-side navigation pane, choose **Application Management** > **Applications**. On the **Applications** page, click the name of the target application.

On the **Application Details** page, choose **Application Monitoring** > **Infrastructure Monitoring** in the left-side navigation pane.

On the Infrastructure Monitoring page, group data in the latest half of an hour is monitored by default.

You can select an interval to monitor group data at another interval, or click the **Single Instance Data** tab to monitor single-instance data.

On the **Infrastructure Monitoring** page, select a monitoring data type.

Data to be monitored includes group data and single-instance data.

The following column metrics of the two data types are monitored from different dimensions:

CPU Data: indicates the CPU usage, which is the sum of the user usage and system usage. The group data graph displays the average usage of all ECS instances in the application group.

Memory Data: indicates the total and actually used physical memory. The group data graph displays the total memory size and the total memory usage of all ECS instances in the

application group.

Load Data: indicates the "One-Minute Load" field for the system workload. The group data graph displays the average "One-Minute Load" of all ECS instances in the application group.

Network Speed Data: indicates the read and write speeds of the NIC. If an ECS instance contains multiple NICs, this parameter indicates the summed read and write speeds of all NICs whose names start with "eth". The group data graph displays the average summed read and writes speeds of all ECS instances in the application group.

Disk Data: indicates the total and actually used size of all disks mounted to the ECS instance. The group data graph displays the total disk size and the total disk usage of all ECS instances in the application group.

Disk Reading and Writing Speeds: indicates the summed read and write speeds of all disks mounted to the ECS instance. The group data graph displays the average summed read and write speeds of the disks on all ECS instances in the application group.

Disk Reading and Writing Numbers: indicates the summed IOPS values of all disks mounted to the ECS instance. The group data graph displays the average summed IOPS values of the disks on all ECS instances in the application group.

Set the interval.

You can set the interval to "Half an Hour," "Six Hours," "One Day," or "One Week."

Half an Hour: collects the monitoring data generated in the last 30 minutes. This statistical period is applied by default when you are on the Infrastructure Monitoring page.In this statistical period, data is collected every minute, which is the finest query granularity by EDAS.

Six Hours: collects the monitoring data in the last six hours. During this statistical period, data is collected every five minutes.

One Day: collects the monitoring data in the last 24 hours. During this statistical period, data is collected every 15 minutes.

One Week: collects the monitoring data in the last seven days. During this statistical period, data is collected every hour. This period is also the longest statistical cycle supported by EDAS.

Note: The period between "Start Time" and "End Time" indicates the time span

displayed currently.When you set one of the parameters, the corresponding parameter is automatically updated. For example, if you select "30 minutes" and set "End Time" to "2016-05-20 12:00:00," "Start Time" automatically changes to "2016-05-20 11:30:00."

After configuration, monitored data is automatically updated based on the selected interval.

(Optional) View the enlarged graph of a specific metric.

When viewing a monitoring view, click "Zoom In" under a metric to view an enlarged graph of the metric, and adjust the interval in the enlarged graph.

Service monitoring

By collecting and analyzing tracked logs of the various middleware services in the network calls, you can obtain the call traces of a specific request across systems. This helps sort out application request entrances, service call initiators and dependencies, and helps you to analyze system call bottlenecks, estimate capacity, and quickly locate exceptions.

Monitor a service

Log on to the EDAS console.

Select Applications Management > Applications in the left-side navigation pane.

Click the name of the application in the application list.

On the application page, select **Application Monitoring** > **Service Monitoring** from the leftside navigation pane.

The service monitoring page contains the following tabs:

- **RPC Call Overview**: Displays the call records of the RPC service provided by the current application.
- **RPC Call Source**: Displays the applications that call the RPC service provided by the current application.
- **RPC Call Dependency**: Displays the applications whose services are called by the current application.

(Optional) Set the monitoring conditions, and click Update to refresh monitor data.

Latest: Displays data at the current time by default. Select a period from the dropdown list.

Sort by: Sorts data by QPS by default. Select an option from the drop-down menu to sort data by the elapsed time or errors/s (average QPS errors per minute).

Results: 10 records are displayed by default. Select the number of results to be displayed from the drop-down menu. Options are 1, 5, 30, 50, 100, and unlimited.

Display: Results are displayed in blocks by default. You can also set the display mode to chart or table.

View monitor data.

For details about the metrics, see Application monitoring overview.

Click a metric of a column in the monitoring graph. The custom query page is displayed.

In the Metrics area, select metrics to view data of different groups.

View traces

When monitoring a service, you can monitor the service call trace between the application and other applications. You can also view detailed call traces.

In the monitoring graph, click **View Trace** next to a calling or called service to go to the **Trace Query** page.

On the trace query page, you can view the call trace between the application and the calling/called service.

For details about how to query traces, see Trace query.

Monitor a drilled-down application

On the service monitoring page, besides querying call traces related to an application, you can drill down to view monitor data about interdependent applications.

On the **RPC Call Overview**, **RPC Call Source**, or **RPC Call Dependency** tab, click **Source Application**, **Called Service**, or **Calling Service** next to **Drill Down** at the top of the monitoring graph. The monitoring page of the drilled down application is displayed.

Monitor data of the drilled down application.

The method for monitoring data of a drilled-down application is the same as that for monitoring the current application.

Advanced monitoring

Application Real-Time Monitoring Service (ARMS) is an application performance management (APM) monitoring product provided by Alibaba Cloud.ARMS can work with EDAS seamlessly.For applications deployed in EDAS, you can enable advanced monitoring to activate APM provided by ARMS so that you can manage the performance of applications in advanced mode.

To enable advanced monitoring, see One-click EDAS application access.

Install log collectors

Install the log collector before using the EDAS application monitoring function.

EDAS provides a suite of functions, where a lot of data is pulled from application instances. This requires that the server can be connected to the relevant instances.

Alibaba Cloud's network environment consists of classic networks and VPCs.

- In a classic network, if the firewall and security groups have no port (8182) restrictions, the server can be connected to instances directly.
- In a VPC, instances are intrinsically isolated from servers. EDAS provides a special utility for VPCs: the log collector.

The log collector is divided into two types, Server and Client.SProxy is the log collector client installed on the instance, as shown in the following figure:



Some instances in EDAS now support the automatic installation of a log collector, while others only support its manual installation.

Under any of the following conditions, an ECS instance does not support the automatic installation of a log collector:

- The ECS instance was created before December 1, 2017.
- A classic network ECS instance is imported into a classic network cluster.
- The ECS instance is not running (it is stopped, starting, or stopping).
- The ECS instance is a Windows instance or does not support simple shell commands.
- The ECS instance is not imported from an ECS cluster.

Install the log collector automatically

Log on to the EDAS console.

In the left-side navigation pane, choose **Resource Management** > **VPC**.

In the VPC ID list, locate the row that contains the VPC where you want to install the log collector and click **Install Log Collector** in the Actions column.

In the displayed **Install Log Collector** dialog box, locate the row that contains the target ECS instance, if it supports automatic log collector installation, click **Automatic Installation** in the Actions column .

After a short time, the **Installation complete** status is displayed.

Note: If the installation fails, you can **manually install the log collector**. If both automatic and manual installation fail, submit a ticket.

Manually install the log collector

Locate the row that contains the target ECS instance according to steps 1 to 3 in Automatically install the log collector.

If the ECS instance does not support the automatic installation of the log collector, click **Manual Installation** in the Actions column.

On the **Install Log Collector** page, click **Copy** to paste the script for the ECS instance.

Log on to the ECS instance as the root user. Paste the copied script for installation and press **Enter**.

After the installation is complete, manually execute the netstat -ant|grep 8002 command.

If a connection can be detected, the log collector has been installed successfully.

If no connection can be detected, the installation encountered a problem. In this case, submit a ticket.