

# Alibaba Cloud Apsara Stack Enterprise

## **User Guide - Middleware and Enterprise Applications**

**Version: 1909, Internal: V3.8.1**

**Issue: 20200227**

## Legal disclaimer

---

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent









ial, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

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

.



## Document conventions

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

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



# Contents

---

<b>Legal disclaimer.....</b>	<b>I</b>
<b>Document conventions.....</b>	<b>I</b>
<b>1 Enterprise Distributed Application Service (EDAS).....</b>	<b>1</b>
1.1 What is EDAS?.....	1
1.2 Quick start.....	3
1.2.1 Log on to the EDAS console.....	3
1.2.2 Deploy Java applications in ECS clusters.....	4
1.2.3 Host Spring Cloud applications to EDAS.....	7
1.2.4 Host Dubbo applications to EDAS.....	16
1.3 Deploy applications.....	22
1.3.1 Deploy applications in the console.....	22
1.3.1.1 Deploy web applications in ECS clusters.....	22
1.3.1.2 Deploy applications in Container Service Kubernetes clusters by using images.....	26
1.3.2 Use CLI to deploy applications.....	33
1.3.2.1 Use toolkit-maven-plugin to automatically deploy applications.....	33
1.3.2.2 Use CLI to deploy applications in EDAS.....	41
1.3.2.3 Use Alibaba Cloud Toolkit for Eclipse to deploy applications....	43
1.3.2.4 Use Alibaba Cloud Toolkit for IntelliJ IDEA to deploy applications.....	46
1.3.3 Deploy applications in hybrid clouds.....	50
1.4 Console user guide.....	55
1.4.1 Overview page.....	55
1.4.2 Resource management.....	55
1.4.2.1 Import ECS instances.....	55
1.4.2.2 View SLB instance information.....	56
1.4.2.3 View a VPC.....	57
1.4.2.4 Manage clusters.....	58
1.4.2.4.1 Create an ECS cluster.....	58
1.4.2.4.2 Import a user-created Kubernetes cluster.....	59
1.4.2.5 Manage resource groups.....	60
1.4.3 Manage applications.....	62
1.4.3.1 Namespace.....	62
1.4.3.2 Lifecycle management for applications in ECS clusters.....	62
1.4.3.2.1 Publish an application.....	63
1.4.3.2.1.1 Create an application in the Apsara Stack console.....	63
1.4.3.2.1.2 Deploy an application (applicable to ECS clusters).....	64
1.4.3.2.1.3 Create an application (applicable to ECS clusters).....	69
1.4.3.2.2 Manage applications.....	71



1.4.3.2.2.1 Scaling (applicable to ECS clusters).....	71
1.4.3.2.2.2 Create an application branch version.....	72
1.4.3.2.2.3 Upgrade the container version.....	74
1.4.3.2.2.4 Roll back an application.....	74
1.4.3.2.2.5 Delete an application.....	75
1.4.3.2.3 Application settings.....	75
1.4.3.2.3.1 Set JVM parameters.....	75
1.4.3.2.3.2 Configure Tomcat.....	76
1.4.3.3 Lifecycle management for Container Service Kubernetes applications.....	78
1.4.3.3.1 Container Service Kubernetes clusters.....	78
1.4.3.3.2 Prepare an application image (a Container Service Kubernetes cluster).....	78
1.4.3.3.3 Deploy an application (applicable to Container Service Kubernetes clusters).....	83
1.4.3.3.4 Scaling (applicable to Container Service Kubernetes clusters).....	86
1.4.3.4 Lifecycle management for applications in a user-created Kubernetes cluster.....	87
1.4.3.4.1 Deploy an application (applicable to user-created Kubernetes clusters).....	87
1.4.3.4.2 Application management (applicable to user-created standard Kubernetes clusters).....	92
1.4.3.5 Log management.....	94
1.4.3.6 Application monitoring.....	95
1.4.3.6.1 Install a log collector.....	97
1.4.3.6.2 Dashboard.....	98
1.4.3.6.3 Infrastructure monitoring.....	99
1.4.3.6.4 Service monitoring.....	101
1.4.3.6.5 Advanced monitoring.....	103
1.4.3.7 Notifications and alarms (only applicable to HSF applications in ECS clusters).....	103
1.4.3.8 Auto scaling (only applicable to HSF applications in ECS clusters).....	106
1.4.3.9 Throttling and degradation (only applicable to HSF applications in ECS clusters).....	108
1.4.3.9.1 Throttling management.....	109
1.4.3.9.2 Degradation management.....	111
1.4.3.10 Application diagnosis (only applicable to HSF applications in ECS clusters).....	113
1.4.3.10.1 Common operations.....	114
1.4.3.10.2 Connector.....	116
1.4.3.10.3 Method tracing.....	117
1.4.3.10.4 Commons Pool monitoring.....	123
1.4.3.10.5 Druid database connection pool monitoring.....	125

1.4.3.10.6 Hot thread.....	128
1.4.3.11 Container version management (only applicable to HSF applications in ECS clusters).....	129
1.4.4 Microservice management.....	130
1.4.4.1 Query traces.....	131
1.4.4.2 Trace details.....	132
1.4.4.3 Service statistics.....	134
1.4.4.4 Service topology.....	134
1.4.4.5 Redis support.....	135
1.4.5 Batch operations.....	147
1.4.6 System management.....	149
1.4.6.1 Introduction to the EDAS account system.....	149
1.4.6.2 Manage RAM users.....	150
1.4.6.2.1 RAM user overview.....	152
1.4.6.2.2 Use a primary account for RAM user operations.....	152
1.4.6.3 Manage roles.....	153
1.4.6.4 View all permissions.....	154
1.5 FAQ.....	154
1.5.1 Development FAQ.....	154
1.5.1.1 Ali-Tomcat FAQ.....	154
1.5.1.2 Lightweight configuration center FAQ.....	157
1.5.1.3 HSF FAQ.....	159
1.5.1.4 HSF error codes.....	161
1.5.1.5 Other development problems.....	168
1.5.2 Usage FAQ.....	168
1.5.2.1 Account management.....	168
1.5.2.2 Resource management.....	169
1.5.2.3 Application lifecycle.....	171
1.5.2.4 Monitoring and alarms.....	174
<b>2 API Gateway.....</b>	<b>175</b>
2.1 What is API Gateway?.....	175
2.2 Log on to the API Gateway console.....	175
2.3 Quick start for consumers.....	176
2.3.1 Overview.....	176
2.3.2 Step 1: View API settings.....	177
2.3.3 Step 2: Create an application.....	177
2.3.4 Step 3: Obtain authorization.....	178
2.3.5 Step 4: Call an API.....	179
2.4 Quick start for providers.....	180
2.4.1 Overview.....	180
2.4.2 Create a group.....	181
2.4.3 Create an API.....	181
2.4.4 Publish an API.....	186
2.4.5 Authorize an application.....	187
2.5 Call an API.....	189

2.5.1 Manage applications.....	189
2.5.1.1 Create an application.....	189
2.5.1.2 View application details.....	190
2.5.1.3 Modify an application.....	190
2.5.1.4 Delete an application.....	190
2.5.2 View existing APIs.....	191
2.5.3 Authorize an application.....	191
2.5.4 Encrypt a signature.....	191
2.5.5 Request signatures.....	192
2.5.6 API call examples.....	195
2.6 APIs.....	197
2.6.1 Manage groups.....	197
2.6.1.1 Create a group.....	197
2.6.1.2 Manage domain names.....	197
2.6.1.3 Manage certificates.....	198
2.6.1.4 Delete a group.....	199
2.6.1.5 Manage environments.....	200
2.6.2 Create an API.....	201
2.6.2.1 Overview.....	201
2.6.2.2 Create an API.....	202
2.6.2.3 Security authentication.....	207
2.6.2.4 Network protocol.....	207
2.6.2.5 Request body configuration.....	208
2.6.2.6 VPC ID.....	208
2.6.2.7 Configure an API in mock mode.....	208
2.6.2.8 Return the Content-Type header.....	210
2.6.3 API management.....	210
2.6.3.1 View and modify an API.....	210
2.6.3.2 Publish an API.....	210
2.6.3.3 Authorize an application.....	212
2.6.3.4 Revoke authorization.....	213
2.6.3.5 Unpublish an API.....	213
2.6.3.6 View the version history of an API.....	214
2.6.3.7 Change the version of an API.....	214
2.6.4 Plugin management.....	215
2.6.4.1 Create a plugin.....	215
2.6.4.1.1 Create an IP-based access control plugin.....	215
2.6.4.1.2 Create a throttling plugin.....	216
2.6.4.1.3 Create a signature key plugin.....	217
2.6.4.2 Bind a plugin to an API.....	218
2.6.4.3 Delete a plugin.....	219
2.6.4.4 Unbind a plugin.....	220
2.7 Advanced usage.....	220
2.7.1 Business parameters of custom logs.....	220
2.7.2 Configure Log Service logs for API Gateway.....	221

2.7.2.1 Initialize the default Log Service configuration of API Gateway.....	221
2.7.2.2 Configure API Gateway to deliver logs to your Log Service project.....	225
2.7.3 Cross-user VPC authorization.....	229
2.7.3.1 User authorization across VPCs.....	229
2.7.3.2 API configurations.....	231

# 1 Enterprise Distributed Application Service (EDAS)

---

## 1.1 What is EDAS?

**Enterprise Distributed Application Service (EDAS) is a PaaS platform for application hosting and microservice management, providing full-stack solutions such as application development, deployment, monitoring, and O&M. It supports Dubbo, Spring Cloud, and other microservice runtime environments, helping you easily migrate applications to the cloud.**

Diverse application hosting environments

**You can select instance-exclusive ECS clusters, Container Service Kubernetes clusters, and user-created Kubernetes clusters based on your application systems and resource needs.**

Abundant microservice frameworks

**You can develop applications and services in the native Dubbo, native Spring Cloud, and HSF frameworks, and host the developed applications and services to EDAS.**

- **You can host Dubbo and Spring Cloud applications to EDAS by adding dependencies and modifying a few configurations. You have access to the functions of EDAS, such as enterprise-level application hosting, service governance, monitoring and alarms, and application diagnosis, without having to build ZooKeeper, Eureka, and Consul. This lowers the costs of deployment and O&M.**
- **HSF is the distributed RPC framework that is widely used within the Alibaba Group. It interconnects different service systems and decouples inter-system implementation dependencies. HSF unifies the service publishing and call methods for distributed applications to help you conveniently and quickly develop distributed applications. HSF provides or uses common function modules, and frees developers from various complex technical details involved in distributed architectures, such as remote communication, serialization, performance loss, and the implementation of synchronous and asynchronous calls.**

## Complete application management

**You can perform end-to-end management, service governance, and microservice management for your applications in the EDAS console.**

- **Application lifecycle management**

**EDAS provides end-to-end application management, allowing you to deploy, scale out, scale in, stop, and delete applications. Applications of all sizes can be managed in the EDAS console.**

- **Service governance**

**EDAS integrates a wide variety of service governance components, such as auto scaling, throttling and degradation, and health check, to deal with unexpected traffic spikes and crashes caused by dependencies. This greatly improves platform stability.**

- **Microservice management**

**EDAS provides the service topology, service statistics, and trace query functions to help you manage every component and service in a distributed system.**

## Comprehensive monitoring and diagnosis

**You can monitor the status of resources and services in applications in the EDAS console to promptly identify problems and quickly locate their causes through the logging and diagnosis components.**

- **Application monitoring**

**EDAS monitors the health status of application resources at the IaaS layer in real time, helping you quickly locate problems.**

- **Application diagnosis**

**EDAS provides the container-based application diagnosis function. Based on the provided data, this function allows you to identify application runtime errors , such as errors in Garbage Collection (GC), class loading, connectors, memory allocated for objects, thread hotspots, Druid database connection pools, and Commons Pool.**

## 1.2 Quick start

This topic describes how to use EDAS to publish a simple web application that only contains a welcome page in Alibaba Cloud Virtual Private Cloud (VPC).

### 1.2.1 Log on to the EDAS console

This topic describes how to log on to the EDAS console.

#### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

#### Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username `super`. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.
5. In the top navigation bar of the page, choose **Middleware > Enterprise Distributed Application Service**.

6. On the EDAS page, click Console, select a region and department, and click EDAS to go to the EDAS console.

**Note:**

If you are using EDAS for the first time, click Agree to Authorize in the Service Access Authorization dialog box.

## 1.2.2 Deploy Java applications in ECS clusters

To help you quickly get started, EDAS provides a Java web application demo that only contains a welcome page so that you can quickly learn how to publish the Java application on multiple ECS instances. To use these ECS instances, you must create them on Alibaba Cloud and then deploy them to Alibaba Cloud VPCs.

### Prerequisites

- You have created a VPC, VSwitch, and security group.
- You have created an ECS cluster and imported instances to the cluster.
- Before deploying an application, ensure that the RAM is authorized.

### Procedure

#### Create an application

1. [Log on to the EDAS console](#) Perform the following steps to create an application:
2. In the EDAS console, choose Application Management > ApplicationsApplication Management from the left-side navigation pane. On the Applications page, click Create Application in the upper-right corner.
3. On the Application Information page, set the parameters of the application. Then, click Next Step: Application Configurations.

Table 1-1: Basic parameters

Parameter	Description
Namespace	Select a namespace from the drop-down list.
Cluster Type	Select ECS Cluster from the drop-down list and select an ECS cluster.
Application Name	Enter a descriptive application name.
Application Runtime Environment	Select the latest EDAS Container version, such as EDAS Container 2 [support FatJar deployment].





Parameter	Description
Java Runtime Environment	Select JRE 8 or JRE 7.
Application Description	Enter remarks for the application.

4. On the Application Configuration page, add an instance and configure the instance as instructed. After configuring the instance, click Create.

Table 1-2:

Parameter	Description
Select Instances	<p>Click Add. On the Instances page, select instances and click &gt; to add the instances to the field on the right. Then, click OK.</p> <ul style="list-style-type: none"> <li>· If no instances are selected, click Create an Empty Application. Then, you can choose Scale Out, Add Instance, or Deploy Applications to publish the application.</li> <li>· If instances are selected, click Create to create an empty application that contains the instances. Then, you can click Deploy Application to publish the application.</li> </ul>
Deploy Now	Select this option after instances are added. Set the deployment parameters in the lower section.
Deployment Method	Select WAR or JAR. The configuration processes for WAR package deployment and JAR package deployment are similar. Here, WAR package deployment is used as an example.

Parameter	Description
File Uploading Method	<p>Select Upload WAR Package or WAR Package Location.</p> <ul style="list-style-type: none"> <li>• <b>Upload WAR Package:</b> Click <a href="#">Download Sample WAR Package</a>. After the demo is downloaded, click Select File and select the WAR package.</li> <li>• <b>WAR Package Location:</b> Right-click <a href="#">Download Sample WAR Package</a> and choose Copy Link Address from the shortcut menu. Copy and paste the address in the WAR package address bar.</li> </ul> <div>  <b>Note:</b>  The name of the application deployment package can only contain letters, numbers, hyphens (-), and underscores (_). A JAR package can be uploaded only when the JAR package deployment method is selected. Otherwise, you can only deploy the application by using a WAR package. </div>
Enter Version	<p>Enter the version of the deployment package for publishing the application.</p> <div>  <b>Note:</b>  You can add a version or textual description when deploying an application. We do not recommend using a timestamp as the version. </div>
Version	Specify a version (such as 1.1.0). We do not recommend using a timestamp as the version.
(Optional) Application Health Check	Set a URL for application health check. The system checks the health of the application after EDAS Container has started or is running. Then, it performs a service routing task based on the health check result. In this example, the health check URL is set to <code>http://127.0.0.1:8080/healthCheck.html</code> .

Parameter	Description
Batch	Specify a number of deployment batches. Select an option from the drop-down list. The options are automatically generated based on the number of instances for the application. If you select two or more batches, you must set the batch wait time .
Batch Mode	Select Automatic.

## Result

After creating the application, go to the Change Details page. Click the Basic Information and Instance Information tabs to view the application status.

### 1.2.3 Host Spring Cloud applications to EDAS

This topic uses a provider-consumer example to describe how to develop and test a Spring Cloud application locally and deploy it to EDAS by adding dependencies and required configurations. This implements service registration and discovery for the application and allows the consumer to call the provider.

## Prerequisites

- You have downloaded [Maven](#) and set the environment variables. If you have installed Maven in your local instance, skip this step.
- You have downloaded, started, and configured the lightweight configuration center.

To help you develop applications locally, EDAS provides the lightweight configuration center, which contains the basic functions of the EDAS service registry. Applications that are developed in the lightweight configuration center can be deployed to off-premises EDAS, without the need to modify any code or configuration.

## Context

ANS is a service discovery component provided by EDAS, which is a commercial version of open source Nacos.

Spring Cloud Alibaba ANS implements the standard interfaces and specifications of Spring Cloud Registry and can fully replace the service discovery functions of Spring Cloud Eureka and Spring Cloud Consul.

In addition, ANS offers the following advantages over Spring Cloud Eureka and Spring Cloud Consul:

- ANS is a shared component that saves you the costs of deploying, operating, or maintaining Spring Cloud Eureka or Spring Cloud Consul.
- ANS provides link encryption for both service registration and discovery calls, protecting your service from being detected by others.
- ANS is fully integrated with other EDAS components to provide you with a complete set of microservice solutions.

If you are familiar with Spring Cloud, after reading this topic, you will find that ANS implements service registration and discovery in the same way as Eureka or Consul. You can migrate from Eureka and Consul to ANS without code modification.



**Note:**

Currently, EDAS only supports the minor versions of Spring Cloud Finchley and Spring Cloud Edgware. For information about the corresponding Spring Boot versions, visit the Spring website. Spring Cloud Alibaba 0.2.1.RELEASE corresponds to Spring Cloud Finchley, and Spring Cloud Alibaba 0.1.1.RELEASE corresponds to Spring Cloud Edgware.

This topic describes key information for developing Spring Cloud applications locally. For more information about Spring Cloud, download [service-provider](#) and [service-consumer](#).

## Procedure

Create a service provider.

1. Create an application project named `service-provider`.
2. Add dependencies to the `pom.xml` file.

Spring Boot 2.0.6.RELEASE and Spring Cloud Finchley.SR1 are used as examples.

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.6.RELEASE</version>
  <relativePath/>
</parent>
<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-alicloud-ans</artifactId>
    <version>0.2.1.RELEASE</version>
  </dependency>
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
</dependencies>
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>Finchley.SR1</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

**Spring Boot 2.x is used in the example. If you want to use Spring Boot 1.x, use Spring Boot 1.5.x, Spring Cloud Edgware, and Spring Cloud Alibaba 0.1.0.RELEASE.**



**Note:**

**Spring Boot 1.x will expire in August 2019, so we recommend that you use a later version to develop applications.**

3. **Develop the startup class of the provider, in which the `@EnableDiscoveryClient` annotation indicates that the service registration and discovery functions must be enabled for the application.**

```
@SpringBootApplication
@EnableDiscoveryClient
public class ServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(ServerApplication.class, args);
    }
}
```

4. **Create a simple controller, set URL mapping to `/echo/{String}`, set the HTTP method to `GET`, retrieve method parameters from the URL path, and set the logic to echo the received parameters.**

```
@RestController
public class EchoController {
    @RequestMapping(value = "/echo/{string}", method = RequestMethod.GET)
    public String echo(@PathVariable String string) {
        return string;
    }
}
```

```
}
```

5. In the `application.properties` file, add the following configuration and specify the EDAS lightweight configuration center as the registry.

In the configuration, `127.0.0.1` is the address of the lightweight configuration center. If your lightweight configuration center is deployed on another instance, change the address to the IP address of the instance. The default port of the lightweight configuration center is 8080 and cannot be changed.

```
spring.application.name=service-provider
server.port=18081
spring.cloud.alicloud.ans.server-list=127.0.0.1
spring.cloud.alicloud.ans.server-port=8080
```

**Note:**

`spring.cloud.alicloud.ans.server-list=127.0.0.1` and `spring.cloud.alicloud.ans.server-port=8080`

are used only when the lightweight configuration center is used as the service registry during local development. When you deploy applications in EDAS, these two parameters can be removed or retained, without affecting service registration and usage.

You can add the following parameters as needed:

Table 1-3: Reference configuration items

Configuration item	Key	Default value	Description
Service Name	<code>spring.cloud.alicloud.ans.client-domains</code>	<code>spring.application.name</code>	When this parameter is not set, it is retrieved from <code>spring.application.name</code> by default. To publish multiple services, separate the service names with commas (,).

Configuration item	Key	Default value	Description
Register	spring.cloud.alicloud.ans.register-enabled	true	When you only need service discovery but do not need registration, set this parameter to false to disable registration.
IP Address to Be Registered	spring.cloud.alicloud.ans.client-ip	Not supported	Set this parameter to the registered IP address of the local instance. It has the highest priority.
Network Adapter for the IP Address to Be Registered	spring.cloud.alicloud.ans.client-interface-name	Not supported	Set this parameter to the name of the network adapter corresponding to the IP address to which the application will be published.
Port to Be Registered	spring.cloud.alicloud.ans.client-port	Not supported	Customize the port to be registered.

6. Execute the main function of `service-provider` to start the service.

7. Log on to the lightweight configuration center console (<http://<IP address of the instance where the lightweight configuration center is installed:8080>>). In the left-side navigation pane, click Services to view Providers.

`service-provider` is included in the provider list, and you can query the service group and address.

### Create a service consumer

This topic describes how to discover a service and how Application Naming Service (ANS) service discovery is used together with RestTemplate, AsyncRestTemplate, and FeignClient.

8. Create an application project named `service-consumer`.

## 9. Add required dependencies to the `pom.xml` file.

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.6.RELEASE</version>
  <relativePath/>
</parent>
<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-alicloud-ans</artifactId>
    <version>0.2.1.RELEASE</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-openfeign</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>Finchley.SR1</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

## 10. Configure RestTemplate, AsyncRestTemplate, and FeignClient.

- a) Create an interface, add the `@FeignClient` annotation, and set the corresponding HTTP URL and HTTP method.

**FeignClient** is the client that completes HTTP-to-RPC conversion to facilitate calls.

```
@FeignClient(name = "service-provider")
public interface EchoService{
    @RequestMapping(value = "/echo/{str}", method = RequestMethod.GET)
    String echo(@PathVariable("str") String str);
}
```



```
}
```

## b) Create a startup class and add the configuration.

- **Enable service registration and discovery by using the `@EnableDiscoveryClient` annotation.**
- **Activate FeignClient by using the `@EnableFeignClients` annotation.**
- **Add the `@LoadBalanced` annotation to combine `RestTemplate`, `AsyncRestTemplate`, and service discovery.**

```
@SpringBootApplication
@EnableDiscoveryClient
@EnableFeignClients
public class ConsumerApplication {
    @LoadBalanced
    @Bean
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }
    @LoadBalanced
    @Bean
    public AsyncRestTemplate asyncRestTemplate(){
        return new AsyncRestTemplate();
    }
    public static void main(String[] args) {
        SpringApplication.run(ConsumerApplication.class, args);
    }
}
```

## 11.Create a controller to demonstrate and verify the service discovery feature.

```
@RestController
public class TestController {
    @Autowired
    private RestTemplate restTemplate;
    @Autowired
    private AsyncRestTemplate asyncRestTemplate;
    @Autowired
    private EchoService echoService;
    @RequestMapping(value = "/echo-rest/{str}", method = RequestMethod.GET)
    public String rest(@PathVariable String str) {
        return restTemplate.getForObject("http://service-provider/echo/" + str, String.class);
    }
    @RequestMapping(value = "/echo-async-rest/{str}", method = RequestMethod.GET)
    public String asyncRest(@PathVariable String str) throws Exception{
        ListenableFuture<ResponseEntity<String>> future =
            asyncRestTemplate.
                getForEntity("http://service-provider/echo/"+str, String.
                    class);
        return future.get().getBody();
    }
    @RequestMapping(value = "/echo-feign/{str}", method = RequestMethod.GET)
    public String feign(@PathVariable String str) {
```

```
        return echoService.echo(str);  
    }  
}
```

**12**In the `application.properties` file, add the following configuration and specify the EDAS lightweight configuration center as the registry.

In the configuration, `127.0.0.1` is the address of the lightweight configuration center. If your lightweight configuration center is deployed on another instance, change the address to the IP address of the instance. The default port of the lightweight configuration center is 8080 and cannot be changed.

```
spring.application.name=service-consumer  
server.port=18081  
spring.cloud.alicloud.ans.server-list=127.0.0.1  
spring.cloud.alicloud.ans.server-port=8080
```

You can add the following parameters as needed:

Table 1-4: Reference configuration items

Configuration item	Key	Default value	Description
Service Name	<code>spring.cloud.alicloud.ans.client-domains</code>	<code>spring.application.name</code>	When this parameter is not set, it is retrieved from <code>spring.application.name</code> by default. To publish multiple services, separate the service names with commas (,).
Register	<code>spring.cloud.alicloud.ans.register-enabled</code>	<code>true</code>	When you only need service discovery but do not need registration, set this parameter to <code>false</code> to disable registration.

Configuration item	Key	Default value	Description
IP Address to Be Registered	spring.cloud.alicloud.ans.client-ip	Not supported	Set this parameter to the registered IP address of the local instance. It has the highest priority.
Network Adapter for the IP Address to Be Registered	spring.cloud.alicloud.ans.client-interface-name	Not supported	Set this parameter to the name of the network adapter corresponding to the IP address to which the application will be published.
Port to Be Registered	spring.cloud.alicloud.ans.client-port	Not supported	Customize the port to be registered.

13 Execute the main function of `service-consumer` to start the service.

14 Log on to the lightweight configuration center console (<http://<IP address of the instance where the lightweight configuration center is installed:8080>>). In the left-side navigation pane, click Services to view Providers.

`service-consumer` is included in the provider list, and you can query the service group and address.

## Deploy applications to EDAS

15 Add the following configuration to the `pom.xml` files of `service-provider` and `service-consumer`, and run `mvn clean package` to compress local applications into executable JAR packages.

### • Provider

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

### • Consumer

```
<build>
```

```
<plugins>
  <plugin>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-maven-plugin</artifactId>
  </plugin>
</plugins>
</build>
```

**16. Deploy applications according to the relevant documentation for the cluster type you want to deploy.**

**If you select an ECS cluster to deploy the JAR package of the Spring Cloud application, you must select Standard Java Application Runtime Environment for Application Runtime Environment.**

### Result

**After the deployment is complete, in the EDAS console, choose Microservice Management > Service Query from the left-side navigation pane. On the Service Query page, select a region and namespace. Then, search for `service-provider` and `service-consumer` to query the deployed applications.**

## 1.2.4 Host Dubbo applications to EDAS

**If you only have basic Java knowledge and Maven experience and are not familiar with Dubbo, this topic helps you develop Dubbo from scratch and register and discover Dubbo applications through the EDAS service registry.**

### Prerequisites

- You have downloaded, started, and configured the lightweight configuration center.

**To help you develop applications locally, EDAS provides the lightweight configuration center that contains the basic functions of the EDAS service registry. Applications that are developed in the lightweight configuration center can be deployed to off-premises EDAS, without the need to modify any code or configuration.**

- You have downloaded [Maven](#) and set the environment variables. If you have installed Maven in your local instance, skip this step.

### Context

The EDAS service registry implements SPI-based [registry extension](#) provided by Dubbo. It completely supports functions such as service registration, [routing rules](#), and [rule configuration](#) on Dubbo.

The EDAS service registry can completely replace ZooKeeper and Redis as the registry for your services on Dubbo. In contrast to ZooKeeper and Redis, the EDAS service registry has the following advantages:

- The EDAS service registry is a shared component, saving you the costs of operating, maintaining, and deploying components such as ZooKeeper.
- The EDAS service registry implements authentication and encryption in the communication process, improving the security of your service registration links.
- The EDAS service registry is fully integrated with other EDAS components to provide you with a complete set of microservice solutions.

**Note:** Currently, EDAS supports Dubbo 2.5.3 to 2.7.0.

- If you use Dubbo 2.5.3 to 2.6.x, use `edas-dubbo-extension 1.0.6`.
- If you use Dubbo 2.7.0, change the `edas-dubbo-extension` version to 2.0.2.

If you are familiar with Dubbo, you can selectively read the sections that might be helpful to you.

This topic uses a provider-consumer example to describe how to locally develop the provider and consumer through XML, deploy the provider and consumer in EDAS, register services in the EDAS service registry, and enable the consumer to call the provider. You can also develop Dubbo applications through Spring Boot. For more information, see "Application Developer Guide."

This topic mainly describes the important information about development. For the complete Dubbo program, download [edas-dubbo-demo](#).

## Procedure

### Create a service provider

#### 1. Create a Maven project and add required dependencies.

- a) Create a Maven project by using IDE, such as IntelliJ IDEA and Eclipse.
- b) In the Maven project, add the dubbo and edas-dubbo-extension dependencies to the `pom.xml` file. The versions are 2.6.2 and 1.0.6.

```
<dependencies>
```

```
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>dubbo</artifactId>
  <version>2.6.2</version>
</dependency>
<dependency>
  <groupId>com.alibaba.edas</groupId>
  <artifactId>edas-dubbo-extension</artifactId>
  <version>1.0.6</version>
</dependency>
</dependencies>
```

## 2. Develop a Dubbo service provider.

- a) Create a package named `com.alibaba.edas` in the `src/main/java` path.
- b) Create an interface named `IHelloService` that includes a `SayHello` method in `com.alibaba.edas`.

```
package com.alibaba.edas;
public interface IHelloService {
    String sayHello(String str);
}
```

- c) Create a class named `IHelloServiceImpl` in `com.alibaba.edas` to implement the interface.

```
package com.alibaba.edas;
public class IHelloServiceImpl implements IHelloService {
    public String sayHello(String str) {
        return "hello " + str;
    }
}
```

## 3. Configure the Dubbo service.

- a) Create a file named `provider.xml` in the `src/main/resources` path and open the file.
- b) In `provider.xml`, add Spring-related XML namespace (`xmlns`) and XML schema instance (`xmlns:xsi`), and Dubbo-related XML namespace (`xmlns:dubbo`) and XML schema instance (`xsi:schemaLocation`).

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.
xsd
http://code.alibabatech.com/schema/dubbo
```

```
http://code.alibabatech.com/schema/dubbo/dubbo.xsd">
```

4. In `provider.xml`, expose the interface and implementation class as a Dubbo service.

```
<dubbo:application name="demo-provider"/>
<dubbo:protocol name="dubbo" port="28082"/>
<dubbo:service interface="com.alibaba.edas.IHelloService" ref="
helloService"/>
<bean id="helloService" class="com.alibaba.edas.IHelloServiceImpl
"/>
```

5. Start the service.

- a) Create the class `Provider` in `com.alibaba.edas` and load Spring context to the main function of `Provider` based on the following code to expose the configured Dubbo service:

```
package com.alibaba.edas;
import org.springframework.context.support.ClassPathXmlApplication
ionContext;
public class Provider {
    public static void main(String[] args) throws Exception {
        ClassPathXmlApplicationContext context = new ClassPathX
mlApplicationContext(new String[] {"provider.xml"});
        context.start();
        System.in.read();
    }
}
```

- b) Execute the main function of `Provider` to start the Dubbo service.

6. Log on to the [lightweight configuration center console](#). In the left-side navigation pane, choose Services to view Providers.

`com.alibaba.edas.IHelloService` is included in the provider list, and you can query the service group and provider IP address.

Create a service consumer

7. Create a Maven project and add required dependencies.

The procedure is the same as that for creating a provider. For more information, see the procedure for creating a provider.

8. Develop the Dubbo service.

- a) Create a package named `com.alibaba.edas` in the `src/main/java` path.
- b) Create an interface named `IHelloService` that includes a `SayHello` method in `com.alibaba.edas`.



**Note:**

Generally, an interface is defined in an independent module. The provider and consumer reference the module through Maven dependencies. In this topic, two identical interfaces are created for the provider and consumer for ease of description. We do not recommend this procedure in actual use.

```
package com.alibaba.edas;  
public interface IHelloService {  
    String sayHello(String str);  
}
```

## 9. Configure the Dubbo service.

- a) Create a file named `consumer.xml` in the `src/main/resources` path and open the file.
- b) In `consumer.xml`, add Spring-related XML namespace (`xmlns`) and XML schema instance (`xmlns:xsi`), and Dubbo-related XML namespace (`xmlns:dubbo`) and XML schema instance (`xsi:schemaLocation`).

```
<beans xmlns="http://www.springframework.org/schema/beans"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  
    xsi:schemaLocation="http://www.springframework.org/schema/  
beans  
    http://www.springframework.org/schema/beans/spring-beans-4.0.  
xsd  
    http://code.alibabatech.com/schema/dubbo  
    http://code.alibabatech.com/schema/dubbo/dubbo.xsd">
```

- c) Add the following configuration to `consumer.xml` to subscribe to the Dubbo service:

```
<dubbo:application name="demo-consumer"/>  
<dubbo:registry id="edas" address="edas://127.0.0.1:8080"/>  
<dubbo:reference id="helloService" interface="com.alibaba.edas.  
IHelloService"/>
```

## 10. Start and verify the Dubbo service.

- a) Create the class `Consumer` in `com.alibaba.edas` and load Spring context to the main function of `Consumer` based on the following code to subscribe to and consume the Dubbo service:

```
package com.alibaba.edas;  
import org.springframework.context.support.ClassPathXmlApplicat  
ionContext;  
import java.util.concurrent.TimeUnit;  
public class Consumer {  
    public static void main(String[] args) throws Exception {  
        ClassPathXmlApplicationContext context = new ClassPathX  
mlApplicationContext(new String[] {"consumer.xml"});  
        context.start();  
        while (true) {
```



```
        try {
            TimeUnit.SECONDS.sleep(5);
            IHelloService demoService = (IHelloService)context.
getBean("helloService");
            String result = demoService.sayHello("world");
            System.out.println(result);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

b) Execute the main function of Consumer to start the Dubbo service.

## 11. Verify the creation result.

- a) After Dubbo is started, the console outputs "hello world" continuously, indicating successful service consumption.
- b) Log on to the [lightweight configuration center console](#). In the left-side navigation pane, choose Services. On the Services page, click the Callers tab to view com.alibaba.edas.IHelloService and query the service group and caller IP address.

## Result

### 1. Locally verify the result.

```
curl http://localhost:17080/sayHello/EDAS
Hello, EDAS (from Dubbo with Spring Boot)
```

### 2. Verify the result in EDAS.

```
curl http://localhost:8080/sayHello/EDAS
Hello, EDAS (from Dubbo with Spring Boot)
```

## What's next

### Deploy applications to EDAS

edas-dubbo-extension is designed for migrating applications from a local instance to EDAS, allowing you to directly deploy Dubbo applications in EDAS.

1. Add the following configuration to the pom.xml files of Provider and Consumer, and run mvn clean package to compress local applications into executable JAR packages.

- Provider

```
<build>
  <plugins>
```

```

        <plugin>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-maven-plugin</artifactId>
          <executions>
            <execution>
              <goals>
                <goal>repackage</goal>
              </goals>
              <configuration>
                <classifier>spring-boot</classifier>
                <mainClass>com.alibaba.edas.Provider</
mainClass>
              </configuration>
            </execution>
          </executions>
        </plugin>
      </plugins>
    </build>

```

- **Consumer**

```

    <build>
      <plugins>
        <plugin>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-maven-plugin</artifactId>
          <executions>
            <execution>
              <goals>
                <goal>repackage</goal>
              </goals>
              <configuration>
                <classifier>spring-boot</classifier>
                <mainClass>com.alibaba.edas.Consumer</
mainClass>
              </configuration>
            </execution>
          </executions>
        </plugin>
      </plugins>
    </build>

```

2. Deploy applications according to the relevant documentation for the cluster type you want to deploy.

## 1.3 Deploy applications

### 1.3.1 Deploy applications in the console

#### 1.3.1.1 Deploy web applications in ECS clusters

**In an ECS cluster, an ECS instance can only deploy one application. This topic describes how to create a Java web application that only contains a welcome page,**

and use a WAR package to deploy, update, view, and manage the application in the Enterprise Distributed Application Service (EDAS) console.

## **Prerequisites**

### **Prerequisites**

- You have activated EDAS.
- You have created a VPC.
- (Optional) You have created a namespace.
- You have created an ECS cluster.

## **Procedure**

1. Log on to the EDAS console.
2. In the left-side navigation pane, choose Application Management > Applications.
3. On the Applications page, select a region and a namespace (optional). Click Create Application.
4. On the Application Information page that appears, enter the application information. After setting the parameters, click Create an Empty Application.

This creates an application without any instance. Click Next. On the Application Configuration page, set the required parameters.

- **Namespace:** Select a region from the left-side drop-down list and select a namespace from the right-side drop-down list. If you do not select a namespace, the default one is used.
- **Cluster Type:** Select ECS Cluster from the left-side drop-down list, and select a specific ECS cluster from the right-side drop-down list.
- **Application Name:** The name of the application.
- **Deployment Method:** After selecting an ECS cluster, you can deploy the application through a WAR package or a JAR package.
- **Application Runtime Environment:**
  - Deploy applications by using a WAR package:
    - If you want to create a native Spring Cloud or Dubbo application, select `apache-tomcat`.
    - If you want to create an HSF application, select `EDAS-Container`.
  - Deploy applications by using a JAR package:
    - If you want to create a native Spring Cloud or Dubbo application, select `Default Environment`.
    - If you want to create an HSF application, select `EDAS-Container`.
- **Java Environment:** Select Open JDK 8 or Open JDK 7.
- **Application Description:** Enter the basic information of the application. The maximum length of the description is 128 characters.

5. On the Application Configuration page, add an instance and configure the instance as instructed. After configuring the instance, click Create.

- **Instance Source:** In ECS clusters, you can add an instance in any of the following three methods. If you do not select any instance, click Create an Empty Application to create an application that contains no instances. Then, add instances to the application through [application scale-out](#) and deploy the application.
  - Select an instance from the cluster: Click Add next to Available Instances. On the Instances page, select an idle instance from the cluster of the

application, click > to add the instance to the Selected Instances area, and then click OK.

- Create an instance based on the existing instance specifications:
  - a. Click Host Selection next to Template Host.
  - b. In the Template Host dialog box, select any instance in the cluster and use it as the template. Click Recycling Mode, and then click OK in the lower-right corner.
  - c. On the Application Configuration tab page, configure the password and purchase quantity. Then, select ECS Service Terms | Image Service Terms.
- Create an instance from a template:
  - a. Click Select Template next to Launch Template.
  - b. In the Select Template to Be Launched dialog box, select the template based on which the instance is created and the template version, select Recycling Mode, and then click OK in the lower-right corner.
  - c. On the Application Configuration tab page, configure the purchase quantity of the instance, and select ECS Service Terms | Image Service Terms.
- **Deploy Now:** This option is available only after you have selected an instance. Turn on Deploy Now and configure the instance as instructed.
- **File Uploading Method:** Select Upload WAR Package or WAR Package Location.
  - **Upload WAR Package:** Click Select File and select the target WAR package.
  - **WAR Package Location:** Copy the storage path of the WAR package and paste the path to the WAR package location bar.



**Note:**

The name of the application deployment package can only contain letters, numbers, hyphens (-), and underscores (\_). The JAR package can be uploaded only when the JAR package deployment method is selected. Otherwise, you can only deploy the application by using the WAR package.

- **Version:** Specify the version, for example, *1.1.0*. We recommend that you do not use the timestamp as the version number.
- **Application Health Check:** (Optional) Specify a URL for application health check. The system checks the health of the application after the container is

started or is running. Then, it performs a service routing task based on the health check result. A sample URL is [http://127.0.0.1:8080/\\_etc.html](http://127.0.0.1:8080/_etc.html).

- **Batch:** Specify the number of batches. You can specify the number of batches and publish the application to the selected instances in batches only when two or more instances are selected.
- **Batch Mode:** Select Automatic or Manual. When you select Automatic, you need to specify Batch Wait Time, which is the interval between different application deployment batches.

## Result

Wait several minutes until the application is created. After the application is created, you can view the application information on the Application Details page. On the Application Details page, click the Instance Information tab. On the Instance Information tab page, view the instance running status. If Running Status/Time is Running, the application is published.

### 1.3.1.2 Deploy applications in Container Service Kubernetes clusters by using images

You can deploy applications in Container Service Kubernetes clusters by using images. For this purpose, you must prepare images in advance, create a Container Service Kubernetes cluster in the Container Service for Kubernetes console, import the cluster to the EDAS console, and then create and deploy applications in the cluster.

## Prerequisites

### Prerequisites

- Your Alibaba Cloud account has activated EDAS and Container Service for Kubernetes.
- You have granted the required permissions for Container Service for Kubernetes.
- You have prepared the application images (the Container Service Kubernetes cluster).

## Context

Container Service for Kubernetes provides enterprise-level, high-performance scaling management for Kubernetes containerized applications throughout the application lifecycle. This service simplifies cluster creation and scale-out and integrates Alibaba Cloud capabilities in virtualization, storage, networking, and

security. It provides an ideal runtime environment for Kubernetes containerized applications.

## Procedure

**Step 1: Create a Container Service Kubernetes cluster.**

1. Log on to the Container Service for Kubernetes console.
2. In the left-side navigation pane, choose Clusters. On the Cluster List page, click Create Kubernetes Cluster.

Container Service allows you to create three types of clusters, namely, Kubernetes clusters, managed Kubernetes clusters, and multi-zone Kubernetes clusters.

- **Create Kubernetes clusters:** Three of the instances that you buy and add must be used as master nodes. Applications cannot be deployed on these three instances. You can only deploy applications on other instances (workers).
- **Create managed Kubernetes clusters:** All the instances that you buy and add are workers and can be used to deploy applications.
- **Create multi-zone Kubernetes clusters:** Unlike Kubernetes clusters, in multi-zone Kubernetes clusters, nodes are deployed in different zones. When one zone becomes unavailable, services fail over to nodes in other zones. Three of the instances that you buy and add must be used as master nodes. Applications cannot be deployed on these three instances. You can only deploy applications on other instances (workers).

**Step 2: Import the Container Service Kubernetes cluster to the EDAS console**

3. Log on to the EDAS console.
4. In the left-side navigation pane, choose Resource Management > Clusters.
5. On the Clusters page, click Container Service K8S Cluster. In the cluster list, locate the row that contains the Container Service Kubernetes cluster you created and click Import in the Actions column. In the Import Kubernetes Cluster dialog box, click Import.

When the button in the Actions column of the target cluster changes to Delete and the cluster status is Running, the cluster is imported to EDAS.

**Step 3: Create applications in the Container Service Kubernetes cluster.**

6. In the left-side navigation pane, choose Application Management.

7. On the Applications page, set Region and Namespace, and then click Create Application in the upper-right corner.
8. On the Application Information page, set the basic application information and parameters, and click Next Step: Application Configurations.
  - **Namespace:** Select a region from the left-side drop-down list. Select a namespace from the right-side drop-down list. If no namespace is set, Default is selected.
  - **Cluster Type:** Select Container Service K8S Cluster from the left-side drop-down list and select a specific cluster from the right-side drop-down list.
  - **K8S Namespace:** Internal system objects are allocated to different namespaces to form logically isolated projects, groups, or user groups. In this way, different groups can share resources of the whole cluster while being managed separately.
    - **default:** When the object is not set with a namespace, "default" is used.
    - **kube-system:** The namespace used by objects that are created by the system.
    - **kube-public:** The namespace that is automatically created by the system. It can be read by all users, including users that are not authenticated.
  - **Application Name:** The name of the application.
  - **Application Description:** The basic information of the application.
9. Go to the Application Configuration page and configure an image. By default, Image is selected for Deployment Method. Select an image in My Image.

#### 10 Set pods.

##### a) Set Total Pods.

When a pod fails to run or encounters a fault, it can automatically restart or services on the pod seamlessly fail over to other pods, ensuring a high availability for applications. For stateful applications that use persistent storage, instance data is retained when the applications are redeployed. For



stateless applications, instance data is not retained when the applications are redeployed. You can set Total Pods to a maximum value of 50.

**b) Set Single Pod Resource Quota.**

No quota is set by default. Therefore, both the CPU Cores and Memory values of a single pod are 0. To set the quota, enter a number.

**c) Optional: Set the startup command and parameters.**



**Note:**

If you do not know the [CMD](#) and [ENTRYPOINT](#) content of the original Dockerfile image, do not modify the custom startup command and parameters.

Otherwise, you cannot create applications due to an incorrect custom command.

- **Startup Command:** Enter the startup command. To run the CMD `["/usr/sbin/sshd","-D"]` command, enter `/usr/sbin/sshd -D` in the text box.
- **Startup Parameters:** Enter one parameter per line. For example, args `":["-c"; "while sleep 2"; "do echo date"; "done"]` contains four parameters. In this case, enter the parameters in four lines.

## **11 Set environment variables.**

When creating the application, inject the environment variables you have entered to the container to be generated. This saves you from repeatedly adding common environment variables.

If you are using a MySQL image, refer to the following environment variables:

- **MYSQL\_ROOT\_PASSWORD:** (required) allows you to set a root password for MySQL.
- **MYSQL\_USER** and **MYSQL\_PASSWORD:** (optional) allow you to add an account besides the root account and set a password.
- **MYSQL\_DATABASE** (optional): allows you to set the database that you want to create when the container is generated.

If you are using another type of image, configure the environment variables as needed.

## 12 Set persistent storage.

In the Container Service Kubernetes cluster of Alibaba Cloud, the physical storage of the native volume object is not persistent. That is to say, the volume object is a temporary storage object and has the same lifecycle as the Kubernetes pods. You can use *Network Attached Storage (NAS)*, a persistent storage service of Alibaba Cloud, to store instance data persistently. The instance data is retained when the instance is upgraded or migrated.

**Note:** Before enabling persistent storage, ensure that you have activated *NAS* for your EDAS account. The *billing method* of NAS is Pay-As-You-Go. Ensure that your account has a sufficient balance or is billed in Pay-As-You-Go mode.

- **Storage Type:** The default value is NAS, which cannot be changed.
- **Storage Service Type:** Currently, only SSD (Performance Type) is supported, which cannot be changed.
- **Select NAS:**
  - **Buy New NAS:** Select a NAS mount directory and a local mount directory. A single region supports up to 10 NAS instances. Once you have 10, you cannot create any more. If you must create more instances, open a ticket.
  - **Use Existing NAS:** Select an existing NAS instance. You can create up to two mount points. Only compliant NAS instances appear in the drop-down list.
- **Mount Directory:** Set the mount directory command.

## 13 Set local storage.

You can map part of the file system of the host to the container as needed. Before using this function, read *hostpath* and consider the rationality of the solution.

Table 1-5: File types

Parameter	Value	Description
Default	Null string	The file is directly mounted, without checking the file type.
(New) File directory	DirectoryOrCreate	The file directory. A new directory is created if it does not exist.

Parameter	Value	Description
File Directory	Directory	The file directory. Container startup fails if it does not exist.
(New) File	FileOrCreate	The file. A new file is created if it does not exist .
File	File	The file. Container startup fails if it does not exist.
Socket	Socket	The standard UNIX Socket file. Container startup fails if it does not exist.
CharDevice	CharDevice	The character device file. Container startup fails if it does not exist.
BlockDevice	BlockDevice	The block storage device file. Container startup fails if it does not exist.



**Note:**

You do not need to concern yourself with the Value column in this step.  
However, the Value column may be used by APIs after the application is created.

## 14. Configure the application lifecycle management.

The Container Service Kubernetes cluster supports stateless applications and stateful applications.

- **Stateless:** A stateless application supports multi-replica deployment. When a stateless application is redeployed, instance data is not retained. A stateless application can be either of the following applications:
  - A web application that does not retain instance data during upgrade or migration.
  - An application that can be scaled out to address changing service volumes.
- **Stateful:** A stateful application stores data that requires persistent storage and retains instance data during upgrade or migration. A stateful application can be either of the following applications:
  - An application that frequently operates on containers through SSH.
  - An application that requires persistent data storage (such as applications using MySQL) or that supports inter-cluster election and service discovery, such as ZooKeeper and etcd.

You can set lifecycle management for a stateful application as needed.

Lifecycle management scripts:

- **Poststart script:** This is a container hook, which is triggered immediately after a container is created to notify the container of its creation. The hook does not pass any parameters to the corresponding hook handler. If the corresponding hook handler fails to run, the container is killed and the system determines whether to restart the container according to the restart policy of the container. For more information, see [Container Lifecycle Hooks](#)
- **PreStop script:** This is a container hook, which is triggered before a container is deleted. The corresponding hook handler must be completed before the container deletion request is sent to Docker daemon. Docker daemon sends

an SGTERN semaphore to itself to delete the container, regardless of the hook handler execution result. For more information, see [Container Lifecycle Hooks](#)

- **Liveness script:** This is a container status probe, which monitors the health status of applications. If an application is unhealthy, the container is deleted and created again. For more information, see [Pod Lifecycle](#)
- **Readiness script:** This is a container status probe, which monitors whether applications have started successfully and are running properly. If an application is abnormal, the container status is updated. For more information, see [Pod Lifecycle](#)

15. Then, click Create.

## Result

It takes several minutes to create an application. You can trace the creation process based on the change record and change details. Kubernetes applications do not need to be deployed because they are immediately deployed upon creation. After the application is created, go to the Application Details page to check whether the pod status in the Instance Information section is Running. If yes, the application is published.

## 1.3.2 Use CLI to deploy applications

### 1.3.2.1 Use toolkit-maven-plugin to automatically deploy applications

Previously, EDAS applications had to be deployed according to the step-by-step instructions in the console. To improve the developer experience, toolkit-maven-plugin has been provided for auto application deployment. You can use toolkit-maven-plugin to automatically deploy applications that are developed based on the HSF, Dubbo, or Spring Cloud framework in ECS or Swarm clusters.

Automatically deploy applications

1. 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>1.0.2</version>
    </plugin>
  </plugins>
```

```
</plugins>  
</build>
```

2. Create a file named `.edas_config.yaml` 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-beijingapp:  app_id: eb20****-e6ee-4f6d-a36f-  
5f6a5455****  endpoint: xxxxx
```

In the preceding configuration, `region_id` indicates the ID of the region where the ECS instance that hosts the application is located. `app_id` indicates the ID of the application. `endpoint` indicates the point of presence (POP) of EDAS in Apsara Stack. The preceding parameter values are for reference only. Replace them with your actual application parameters. For example, to obtain an endpoint, contact EDAS Customer Services. For more information about configuration items, see [More configuration items](#).

To obtain the values of these configuration items, perform the following steps:

- a. Log on to the EDAS console.
  - b. In the left-side navigation pane, choose Application Management. On the Applications page, locate the row that contains the target application and click the name of the application. On the Application Management page, click Deploy Application.
  - c. On the Deploy Application page, click Generate Maven Plug-in Configuration to obtain the parameter values.
3. Create an account file and configure the AccessKey ID and AccessKey Secret in yaml format. Obtain the AccessKey ID and AccessKey Secret on the [User Info](#) page in the Alibaba Cloud console. We recommend that you use a [RAM user](#) that has been granted application management permissions to improve the application security. The following provides a configuration example:

```
access_key_id: abcaccess_key_secret: 1234567890
```



**Note:**

In the preceding configuration, `abc` and `1234567890` are for reference only. Replace them with your actual AccessKey ID and AccessKey Secret. In this configuration, the AccessKey ID and AccessKey Secret are only used to generate request signatures and not for any other purposes, such as network transfers.

4. Go to the root directory (or the submodule directory if multiple Maven modules exist) and run the following 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 it is packaged successfully. The application is deployed only when this parameter is configured.
- **access\_key\_file:** The file of the Alibaba Cloud account. For more information about how to specify a key pair, see [Account configuration](#).

5. After you run the preceding command, you have successfully deployed the application with toolkit-maven-plugin.

More configuration items

Configuration items for deploying applications are classified as follows:

- Basic environment variables (env)
- Application configuration items (app)
- Storage configuration items (oss)

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

Type	Parameter	Required	Note
env	region_id	Yes	The ID of the region where the application is located.
	endpoint	No	The POP of the application.
app	app_id	Yes	The ID of the application.

Type	Parameter	Required	Note
	package_version	No	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	The deployment description.
	group_id	No	The ID of the group to which the application is deployed. The default value is All Groups.
	batch	No	The number of deployment batches. The default value is 1 and the maximum value is 5.
	batch_wait_time	No	The waiting time (in minutes ) between deployment batches. The default value is 0.



Type	Parameter	Required	Note
	stage_timeout	No	The timeout period (in minutes) for each change stage . The default value is 5. If batch_wait_time is set, it is automatically counted with this parameter during calculation. During runtime , if a stage waits for a time longer than this threshold value, the plug-in automatically exits .
oss	region_id	No	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	The name of the target bucket. The default value is the free OSS bucket provided by EDAS . If OSS configuration items are specified, you must specify the bucket parameter . Otherwise, the instances use the free OSS bucket automatically allocated by EDAS.

Type	Parameter	Required	Note
	key	No	The custom path used to upload the application package to OSS. The instances use the free OSS bucket provided by EDAS by default. If you use a specified OSS bucket, specify the package storage path in this parameter and use the {region_id}, {app_id}, and {version} variables to set the path through parameters, for example, pkgs/petstore/{version}/store.war. The default value is {region_id}/{app_id}/{version}.
	access_key_id	No	The custom account ID that is used to upload the application package to OSS.
	access_key_secret	No	The custom account key that is used to upload the application package to OSS.

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

Assume that you want to deploy application eb20dc8a-e6ee-4f6d-a36f-5f6a545\*\*\*\* to group 06923bb9-8c5f-4508-94d8-517b692f\*\*\*\* in China (Beijing). The

version of the deployment package is 1.2. In this case, the configuration is as follows:

```
env: region_id: cn-beijingapp: app_id: eb20dc8a-e6ee-4f6d-a36f-5f6a5455**** package_version: 1.2 group_id: 06923bb9-8c5f-4508-94d8-517b692f****
```

### Configuration example 2: Specify an OSS bucket

Assume you want to deploy an application whose ID is eb20dc8a-e6ee-4f6d-a36f-5f6a5455\*\*\*\* and upload the deployment package to your own bucket named release-pkg in China (Beijing). The file object name is my.war, the ID of the OSS account is ABC, and the key of the OSS account is 1234567890. In this case, the configuration is as follows:

```
env: region_id: cn-beijingapp: app_id: eb20dc8a-e6ee-4f6d-a36f-5f6a5455****oss: region_id: cn-beijing bucket: release-pkg key: my.war access_key_id: ABC access_key_secret: 1234567890
```

### Configuration file

- When no configuration file is specified, the plug-in uses the .edas\_config.yaml file in the root directory of the packaged project as the configuration file by default. If the packaged project is a submodule of the Maven project, the configuration file is in 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 but another configuration file is specified using the parameter, the plug-in uses the latter.

### Account configurations and priorities

When using this plug-in to deploy applications, you must provide the AccessKey ID and AccessKey Secret of an Alibaba Cloud account for application deployment. Currently, the plug-in supports multiple configuration methods. If duplicate configurations exist, the configuration method with the higher priority overrides that with the lower priority. Configuration methods are listed as follows in descending order of priority:

- **Specify the AccessKey ID and AccessKey Secret in the CLI: You can specify the AccessKey ID and AccessKey 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 you configure this plug-in in the pom.xml file, configure both parameters as follows:**

```
<plugin>      <groupId>com.alibaba.cloud</groupId>      <artifactId>  
>toolkit-maven-plugin</artifactId>      <version>1.0.2</version><  
configuration>      <accessKeyId>abc</accessKeyId>      <accessKeyS  
ecret>1234567890</accessKeySecret></configuration></plugin>
```

- **Specify the account file in the CLI (recommended): When you package the project by running Maven commands, specify the account file in yaml format with `-Daccess_key_file={account file path}`. For example:**

```
access_key_id: abcaccess_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 plug-in uses the Alibaba Cloud account you set previously to deploy the application.**
- **aliyuncli: If you have used the latest Alibaba Cloud CLI and configured your Alibaba Cloud account, Alibaba Cloud generates the `.aliyuncli` directory in the current Home directory and creates the `credentials` file in the `.aliyuncli` directory to store your account information. Here, the MacOS system is used as an example. Assume that the system user is jack. Then, the following information is stored in the `/Users/jack/.aliyuncli/credentials` file:**

```
[default]aliyun_access_key_secret = 1234567890aliyun_access_key_id  
= abc
```

This plug-in uses this account file as the account for deploying the application

- **aliyun: If you have used a legacy Alibaba Cloud CLI and configured the Alibaba Cloud account, the Alibaba Cloud CLI generates the `.aliyun` directory in the current Home directory and creates the `config.json` file in the `.aliyun` directory. Here, the MacOS system is used as an example. Assume that the system**

**user is jack. Then, the following 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:** Then, the plug-in attempts to retrieve the values of `access_key_id` and `access_key_secret` from system environment variables. In other words, the plug-in retrieves the values from `System.getenv("access_key_id")` and `System.getenv("access_key_secret")`.

### 1.3.2.2 Use CLI to deploy applications in EDAS

The command line interface (CLI) was the most widely used type of user interface before graphical user interfaces (GUIs) become popular. CLIs usually do not support the use of a mouse. Instead, you enter instructions through a keyboard, and the computer receives and runs the instructions. By using the CLI, you can accurately control the system and efficiently and reliably perform complex operations.

#### Prerequisites

Before performing the steps in this tutorial, you must have done the following: [Import ECS instances](#)

#### Context

Alibaba Cloud CLI is an open source tool built on the Go SDK provided by Alibaba Cloud. Alibaba Cloud CLI can directly call the EDAS API. Make sure that you have activated EDAS and know how to use SDKs to call operations in EDAS. For more information about how to call operations, see *Developer Guide*. You can use Alibaba Cloud CLI to deploy all applications developed based on the HSF, Dubbo, or Spring Cloud framework in ECS or Swarm clusters in EDAS.

#### Procedure

## 1. Install CLI

Alibaba Cloud CLI is available after you download and decompress it. It is supported on MacOS, Linux, and Windows (64-bit) clients. Download the appropriate installation package:

- [MacOS](#)
- [Linux](#)
- [Windows \(64-bit\)](#)

After decompressing the installation package, move the *aliyun* file to the */usr/local/bin* directory or add it to the *\$PATH* environment variable.

## 2. Configure CLI

Before using Alibaba Cloud CLI, run the `aliyun configure` command to configure the AccessKey, region, and language for calling your Alibaba Cloud account.

You can create and view your AccessKey on the [Security Management](#) page, or obtain the AccessKey from your system administrator.

```
$ aliyun configureConfiguring profile 'default' ...Aliyun Access Key
ID [None]: <Your AccessKey ID>Aliyun Access Key Secret [None]: <
Your AccessKey Secret>Default Region Id [None]: cn-hangzhouDefault
output format [json]: jsonDefault Language [zh]: zh
```

## 3. Use CLI to create applications

Run the following script to create an application:

```
#!/bin/bash # Region for deployment REGION="cn-beijing" #
ID of the ECS instance ECS_ID="i-2z*****b6" # ID of the
VPC where the ECS instance is located VPC_ID="vpc-t*****c"
# Name of a namespace (which is automatically created if it does
not exists) NAMESPACE="myNamespace" # Name of a cluster (which
is automatically created) CLUSTER_NAME="myCluster" # Name of
an application APP_NAME="myApp" # Step 1: Create a namespace.
aliyun edas InsertOrUpdateRegion --RegionTag $REGION:$NAMESPACE --
RegionName $NAMESPACE --region $REGION --endpoint "edas.cn-beijing.
aliyuncs.com" >> /dev/null # Step 2: Create a cluster. CLUSTER_ID
=`aliyun edas InsertCluster --ClusterName $CLUSTER_NAME --ClusterTyp
e 2 --NetworkMode 2 --VpcId $VPC_ID --logicalRegionId $REGION:$
NAMESPACE --region $REGION --endpoint "edas.cn-beijing.aliyuncs.
com" | sed -E 's/. *"ClusterId":"([a-z0-9-]*)".*/\1/g'` # Step 3
: Convert the ECS instance (which takes some time). aliyun edas
TransformClusterMember --InstanceIds $ECS_ID --TargetClusterId $
CLUSTER_ID --Password Hello1234 >> /dev/nullfor i in `seq 300` do
    OUT=`aliyun edas ListClusterMembers --ClusterId $CLUSTER_ID
    | grep EcuId` && break    sleep 1 done ECU_ID=`echo $OUT |
sed -E 's/. *"EcuId":"([a-z0-9-]*)".*/\1/g'` # Step 4: Create an
application. APP_ID=`aliyun edas InsertApplication --Applicatio
```

```
nName $APP_NAME --BuildPackId 51 --EcuInfo $ECU_ID --ClusterId $
CLUSTER_ID --logicalRegionId $REGION:$NAMESPACE | sed -E 's/. *"
AppId":"([a-z0-9-]*)".*/\1/g'` printf "An application is created by
CLI, App ID:"$APP_ID"\n"
```

#### 4. Use CLI to deploy applications

Run the following code to use Alibaba Cloud CLI to deploy an application:

```
#!/bin/bash # ID of the application to be deployed (which must
be created in advance) APP_ID="87a6*****4d1"
# ID of the group to which the application belongs GROUP_ID="54b
*****f27" # Name of the OSS bucket for uploading
(the bucket must support public read) OSS_BUCKET="eda*****mo"
# Installation package file (created by your CI system) PACKAGE
="hello-edas.war" # Step 1: Upload the deployment package to OSS
. aliyun oss cp -f $PACKAGE oss://$OSS_BUCKET/$PACKAGE >> /dev/
null PKG_URL=`aliyun oss sign oss://$OSS_BUCKET/$PACKAGE|head -
1` # Step 2: Initiate a deployment request. CO_ID=`aliyun edas
DeployApplication --AppId $APP_ID --PackageVersion $VERSION --
DeployType url --WarUrl "${PKG_URL}" --GroupId $GROUP_ID | sed -E '
s/. *"ChangeOrderId":"([a-z0-9-]*)".*/\1/g'` # Step 3: Wait until
the application is deployed. for i in `seq 300` do STATUS=`
aliyun edas GetChangeOrderInfo --ChangeOrderId $CO_ID | sed -E 's/.
*"Status":(.).*/\1/g'` [ 2 = ${STATUS} ] && break sleep 1
done
```

In the preceding configuration items, APP\_ID and GROUP\_ID are two configuration parameters of the application. All parameters in the preceding code are for reference only. Replace them with the actual values.

To obtain the values of these configuration items, perform the following steps:

- a) Log on to the EDAS console.
- b) In the left-side navigation pane, choose Application Management. On the Applications page, locate the row that contains the target application and click the name of the application. On the Application Management page, click Deploy Application.
- c) On the Deploy Application page, click Generate Maven Plug-in Configuration to retrieve the parameter values.

#### 1.3.2.3 Use Alibaba Cloud Toolkit for Eclipse to deploy applications

Alibaba Cloud Toolkit (hereinafter referred to as "Cloud Toolkit") is a free IDE plug-in that helps users use Alibaba Cloud more efficiently. You only need to register or use an existing Alibaba Cloud account to download Cloud Toolkit for free. After the plug-in is downloaded, you can install it to Eclipse. You can use Cloud Toolkit to automatically deploy applications that are developed based on the HSF, Dubbo,

or Spring Cloud framework in ECS or Swarm clusters. This topic describes how to install Cloud Toolkit to Eclipse and use Cloud Toolkit to deploy an application in 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*. The program must be suitable for Java EE developers.

#### Install Cloud Toolkit

1. Start Eclipse.
2. In the top navigation bar, choose Help > Install New Software.
3. In the Available Software dialog box, set Work with to the URL `http://toolkit.aliyun.com/eclipse/` of Cloud Toolkit for Eclipse.
4. In the Name section, select Alibaba Cloud Toolkit Core and Alibaba Cloud Toolkit Deployment Tools. In the Details section, clear Connect all update sites during install to find required software. Then, click Next.
5. Perform the subsequent steps as instructed on the Install page of Eclipse.



#### Note:

During the installation process, a dialog box indicating no digital signature may appear. In this case, click Install anyway.

6. After Cloud Toolkit is installed, restart Eclipse. Then, the Alibaba Cloud Toolkit icon appears in the toolbar.

#### Configure Cloud Toolkit

1. Start Eclipse.
2. Set the AccessKey ID and AccessKey Secret.
  - a. In the toolbar, click the drop-down arrow of the Alibaba Cloud Toolkit icon. In the drop-down list, select Alibaba Cloud Preference....
  - b. In the Preference (Filtered) dialog box, choose Accounts from the left-side navigation pane.
  - c. On the Accounts page, set Access Key ID and Access Key Secret, and click OK.



#### Note:



If you use the AccessKey ID and AccessKey Secret of a RAM user, make sure that the RAM user has the permission to deploy applications. For more information about how to grant permissions to RAM users, see [RAM account authorization](#).

- If you already have an Alibaba Cloud account, on the Accounts page, click **Manage existing Account** to go to the logon page of Alibaba Cloud. After you log on to the system with an existing account, you are redirected to the Security Management page. On this page, obtain the AccessKeyId and AccessKeySecret of the account.
- If you do not have an Alibaba Cloud account, on the Accounts page, click **Sign up**. You are redirected to the Register account page of Alibaba Cloud. On this page, register an Alibaba Cloud account. Then, obtain the AccessKeyId and AccessKeySecret of the account.

### 3. Set an endpoint.

- a. In the Preference (Filtered) dialog box, choose **Appearance & Behavior > Endpoint** from the left-side navigation pane.
- b. On the Endpoint page, set an endpoint and click **Apply** and **Close**.



**Note:**

To obtain an endpoint, contact EDAS Customer Services.

## Deploy applications to EDAS

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

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

2. In the Deploy to EDAS dialog box, select Region, Namespace, Application, Group, and Deploy File as needed. Then, click Deploy.

Parameters for deploying an application to EDAS:

- **Region:** The region where the application is located.
- **Namespace:** The namespace where the application is located.
- **Application:** The name of the application.
- **Group:** The group of the application.



**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 and create an application. For more information about how to create an application, see [Deploy Java applications in ECS clusters](#).

3. When the deployment process starts, the deployment logs are printed on the Console tab of Eclipse. You can view the deployment result based on the logs.

Stop Cloud Toolkit

If you want to stop Cloud Toolkit, end the EDAS-deploy process on the Progress tab.

### 1.3.2.4 Use Alibaba Cloud Toolkit for IntelliJ IDEA to deploy applications

Alibaba Cloud Toolkit for IntelliJ IDEA (hereinafter referred to as "Cloud Toolkit") is a free IDE plug-in that helps users use Alibaba Cloud more efficiently. You only need to register or use an existing Alibaba Cloud account to download Cloud Toolkit for free. After the plug-in is downloaded, you can install it to IntelliJ IDEA. You can use Cloud Toolkit to automatically deploy applications that are developed based on the HSF, Dubbo, or Spring Cloud framework in ECS or Swarm clusters. This topic describes how to install Cloud Toolkit in IntelliJ IDEA and how to use Cloud Toolkit to deploy an application in EDAS.

Prerequisites

- You have downloaded and installed [JDK 1.8 or later](#).
- You have downloaded and installed [IntelliJ IDEA \(2018.3 or later\)](#).



**Note:**

**The official server of the JetBrains plug-in is deployed outside China. If you cannot download IntelliJ IDEA due to a slow network response, join the discussion group provided at the end of this topic to obtain the offline installation package for IntelliJ IDEA from Cloud Toolkit Customer Services.**

#### Install Cloud Toolkit

1. Start IntelliJ IDEA.
2. Install Cloud Toolkit to IntelliJ IDEA.
  - **MacOS system:** On the Preferences page, choose Plugins from the left-side navigation pane. Search for Alibaba Cloud Toolkit and then click Install.
  - **Windows system:** Go to the Plugins page. Search for Alibaba Cloud Toolkit and then click Install.
3. After Cloud Toolkit is installed to IntelliJ IDEA, restart IntelliJ IDEA. The Alibaba Cloud Toolkit icon appears in the toolbar.

#### Configure Cloud Toolkit

After Alibaba Cloud Toolkit is installed, use the AccessKey ID and AccessKey Secret to configure the Cloud Toolkit account.

1. Start IntelliJ IDEA.
2. Set the AccessKey ID and AccessKey Secret.
  - a. Click the Alibaba Cloud Toolkit icon and select Preferences from the drop-down list. On the Settings page, choose Alibaba Cloud Toolkit > Accounts from the left-side navigation pane.
  - b. On the Accounts page, set Access Key ID and Access Key Secret, and click OK.



#### Note:

If you use the AccessKey ID and AccessKey Secret of a RAM user, make sure that the RAM user has the permission to deploy applications. For more information about how to grant permissions to RAM users, see [RAM account authorization](#).

- If you already have an Alibaba Cloud account, on the Accounts page, click Get existing AK/SK to go to the logon page of Alibaba Cloud. After you log on to the system with an existing account, you are redirected to the

**Security Management page. On this page, obtain the AccessKeyId and AccessKeySecret of the account.**

- **If you do not have an Alibaba Cloud account, on the Accounts page, click Sign up. You are redirected to the Register account page of Alibaba Cloud. On this page, register an Alibaba Cloud account. Then, obtain the AccessKeyId and AccessKeySecret of the account.**

**3. Set an endpoint.**

- On IntelliJ IDEA, click the Cloud Toolkit icon and select Preferences from the drop-down list.**
- In the Preferences dialog box, choose Appearance & Behavior > Endpoint from the left-side navigation pane.**
- On the Endpoint page, set the endpoint of EDAS and click Apply.**



**Note:**

**To obtain an endpoint, contact EDAS Customer Services.**

**Deploy applications to EDAS**

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

- 1. On IntelliJ IDEA, click the Alibaba Cloud Toolkit icon and select EDAS on Alibaba Cloud from the drop-down list.**

2. In the Deploy to EDAS dialog box, configure the application deployment parameters. Then, click Apply to save the configurations.
  - a. In the Deploy to EDAS dialog box, select Region, Namespace, Application, and Group in the Application section as needed.
    - **Region:** The region where the application is located.
    - **Namespace:** The namespace where the application is located.
    - **Application:** The name of the application.
    - **Group:** The group of the application.
  - b. Set the build mode.
    - **Maven Build:** If this option is selected for building the application, the system adds a Maven task by default to build the deployment package.
    - **Upload File:** If this option is selected for building the application, upload the WAR package or JAR package, and then deploy the application.



**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 and create an application. For more information about how to create an application, see [Deploy Java applications in ECS clusters](#).

3. Click Run to run the configurations you made in the preceding step. The deployment logs are printed on the Console tab of IntelliJ IDEA. You can view the deployment result based on the logs.

#### Manage Maven tasks

In Cloud Toolkit installed in IntelliJ IDEA, you can deploy Maven tasks. In the Deploy to EDAS dialog box, you can also add, delete, modify, or move Maven tasks in the Before launch section.

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

## Deploy multi-module projects

**Most Maven projects involve multiple modules. These modules can be separately developed 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 section in the Deploy to EDAS dialog box is built for the submodule. For more information about how to manage Maven tasks, see [Manage Maven tasks](#).**

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

- carshop
  - itemcenter-api
  - itemcenter
  - detail

**Itemcenter and detail are submodules and depend on the itemcenter-api module. In this case, how is the itemcenter submodule deployed? In the Before launch section of the Deploy to EDAS dialog box, 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.**

### 1.3.3 Deploy applications in hybrid clouds

**EDAS provides complete solutions for scaling, networking, and central management in hybrid clouds, allowing you to deploy applications in hybrid cloud environments. You can connect instances from Alibaba Cloud, on-premises IDCs, and other cloud service providers (CSPs) through leased lines, and add the instances to hybrid cloud (non-Alibaba Cloud) ECS clusters in EDAS. Then, you can deploy and manage HSF, Dubbo, and Spring Cloud applications in the EDAS console in a unified manner. EDAS supports the auto scaling of ECS instances in Alibaba Cloud.**

#### Prerequisites

- You have created a VPC.

- You have activated Express Connect.
- You have applied for a physical connection to connect your on-premises IDC to Alibaba Cloud VPC.
- The instances in your on-premises IDC meet the following requirements:
  - Operating system: CentOS 7
  - Docker not supported
  - Hardware: no special requirements for CPU and memory

## Context

Your application system may have the following requirements or problems:

- The Alibaba Cloud traffic has a certain degree of volatility and you may face traffic spikes in special scenarios, such as flash sales. You can predict the traffic volumes in such scenarios, but deviations may exist. Since you need to buy ECS instances in advance, it is hard to control the number of needed ECS instances. Knowing when to add ECS instances is also a challenge.
- Some core business systems have high security requirements and you may want to deploy such applications in your own IDC. However, you cannot deploy and manage applications in different environments in a unified manner because instances from Alibaba Cloud, on-premises IDCs, and other CSPs cannot communicate with each other.
- Considering your business needs and availability requirements, you may want to deploy your applications on instances from multiple CSPs, that is, in multi-cloud mode. In this mode, manual processing is required because you cannot centrally manage these applications. This often leads to misoperations.
- Connect Alibaba Cloud to on-premises IDCs or to the clouds of other CSPs through Express Connect.
- Create a hybrid cloud cluster. Then, add ECS instances from Alibaba Cloud and instances from on-premises IDCs and other CSPs to the cluster.
- Deploy your applications to instances in this cluster.

In hybrid clouds, EDAS is used in the following scenarios:

- Manage applications deployed on instances in on-premises IDCs through Alibaba Cloud. After connecting your IDC to the Alibaba Cloud VPC through a leased line, you can manage your applications in the IDC by using Alibaba Cloud EDAS.

- **Scale applications deployed on instances from Alibaba Cloud in or out.** EDAS supports auto scaling and helps you automatically purchase and release instances in Alibaba Cloud. You only need to associate EDAS with your billing account and do not need to buy instances in advance.
- **Deploy and manage instances from other CSPs.** EDAS allows you to deploy applications to instances from CSPs other than Alibaba Cloud and manage these instances in a unified manner.

This topic describes how to use Alibaba Cloud to manage applications deployed on instances in on-premises IDCs. To deploy and manage instances from other CSPs, you only need to connect the target instances to the Alibaba Cloud VPC of EDAS through a leased line. Then, you can operate and manage these instances in the same way as instances in on-premises IDCs. For more information about how to scale out applications deployed on ECS instances from Alibaba Cloud, see [Auto scaling \(only applicable to HSF applications in ECS clusters\)](#).



**Note:**

Currently, only EDAS Professional Edition and EDAS Enterprise Platinum Edition allow you to deploy applications in hybrid cloud environments.

## Procedure



## 1. Create a cluster.

- a) Log on to the EDAS console. For more information, see [Log on to the EDAS console](#).
- b) In the left-side navigation pane, choose Resource Management > Clusters.
- c) On the Clusters page, select the region and namespace, and click Create Cluster.
- d) In the Create Cluster dialog box, enter the cluster information and click Create.

### Parameters for creating a cluster:

- **Cluster Name:** Enter a name for the cluster. The name can only contain letters, numbers, underscores (\_), and periods (.), with a length up to 64 characters.
- **Cluster:** Select Non-Alibaba Cloud.
- **Cluster Type:** The default value is ECS, which cannot be changed.
- **Network Type:** The default value is VPC, which cannot be changed.
- **VPC:** From the drop-down list, select the VPC where you want to create the cluster.
- **Namespace:** The namespace you selected for the hybrid cluster on the Clusters page, which cannot be edited.

After the cluster is created, Created successfully appears in the upper-right corner of the page, and the cluster appears in the cluster list.

## 2. Add instances to the cluster.

To add ECS instances from Alibaba Cloud and instances from on-premises IDCs and other CSPs, perform the following steps:

- a) On the Clusters page, click the name of the cluster you just created.
- b) On the Cluster Details page, click Add ECS Instance.
- c) In the Add ECS Instance dialog box, copy the command for installing EDAS Agent.
- d) Use the root account to log on to your Alibaba Cloud ECS instance or the instance in the on-premises IDC.
- e) Paste the EDAS Agent installation command and run it.

### 3. Open the required ports.

To ensure that your applications in the hybrid cloud cluster can use EDAS normally, you must open the following ports after adding the instances:

- 8182: This port is used to capture infrastructure monitoring and trace monitoring logs.
- 12200 to 12300: These ports are used for Remote Procedure Calls (RPCs).
- 65000 to 65535: These are web ports.

You must open the ports based on the instance type.

- ECS instances from Alibaba Cloud: Open the ports by referring to relevant documents.
- Instances from on-premises IDCs and other CSPs: Open the ports by referring to relevant solutions.

### 4. Check the cluster and instance statuses.

- a. Return to the Clusters page. In the cluster list, locate the cluster you just created and check the values of Status and Instances.

If the cluster status is Normal, the cluster is created. If the value of Instances is same as the number of instances you added, the instances are added successfully.

- b. Click the cluster name. On the Cluster Details page, check the values of Instance Name and Status in the cluster information section.

If the cluster status is Running, the instance is running properly.

### 5. Deploy an application.

Currently, the hybrid cloud cluster type can only be ECS cluster. Therefore, you can deploy applications only in hybrid cloud ECS clusters.

The method for deploying applications in hybrid cloud clusters is the same as that for deploying applications in ECS clusters. See relevant topics to deploy applications.

### Result

Wait several minutes until the application is created. After the application is created, you can view the application information on the Application Details page. On the Application Details page, click the Instance Information tab. On the Instance

Information tab, view the instance running status. If Running Status/Time is Running, the application is published.

## 1.4 Console user guide

### 1.4.1 Overview page

The Overview page of the EDAS console displays the subscription type, runtime status, and number of application instances under the current account, allowing you to intuitively know the resource status of the account.

- **Applications:** the number of applications that you publish in EDAS.
- **Application Instances:** the number of instances on which your applications are deployed.
- **Services:** the number of services included in your applications.
- **Deployments in the Last 7 Days:** the number of times applications were deployed during the past seven days.

### 1.4.2 Resource management

This topic describes EDAS resources and how to use and manage the resources.

In the EDAS console, you can view and use resources, such as ECS and Server Load Balancer (SLB) instances. The EDAS resource management function allows you to use the resources by application. EDAS also supports resource group management. When EDAS is used by multiple users or departments, the permissions to use resources can be controlled by using a primary Alibaba account and its RAM users.

#### 1.4.2.1 Import ECS instances

Before deploying applications by using EDAS, import ECS instances to the specified cluster and install EDAS Agent.

#### Prerequisites

EDAS Agent must be installed on each target ECS instance. Before installing EDAS Agent, ensure that the RAM user is authorized. For the authorization procedure, see *the Apsara Stack Console User Guide* and read the **RAM management** topic

#### Procedure

1. [Log on to the EDAS console](#).

2. In the EDAS console, choose Resource Management > ECS from the left-side navigation pane.
3. On the ECS page, click Import ECS in the upper-right corner.
4. On the Select Cluster and ECS page, select a namespace and click Select Cluster to Import. In the instance list, select ECS instances and click Next.
5. On the Ready to import page, select I agree to convert the above instances, and fully understand that the data in the original systems will be lost after conversion. Then, enter a new password for the root user and confirm the new password, and click Next.
6. On the Import page, view the import status.

On the Import page, the statuses of the imported instances become Converting. It may take 5 minutes. If you click "Click to return to the Cluster Details page" before the import is complete, the health check status shows Converting and the conversion progress is shown as a percentage. When the import is complete, the health check statuses become Running, indicating that the instances are successfully imported.

## Result

Click Click to return to the Cluster Details page to go to the Cluster Details page. In the ECS Instances and Applications section, view the import status and progress.

### 1.4.2.2 View SLB instance information

SLB instances are created in the SLB console. After synchronizing resources in the EDAS console, you can view SLB instance information.

## Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose Resource Management > SLB to go to the SLB page and view the SLB instance information and status.

Table 1-6: SLB instance status description

Name	Description
Instance ID/Name	The ID of an SLB instance, which is automatically generated by the system.
IP Addresses	The internal IP address of your SLB instance.

Name	Description
Backend Servers	The ECS instances that are added in the SLB console and used to receive the requests distributed by the SLB instance.
Status	The status of an SLB instance, which may be Running or Stopped. Expired SLB instances do not appear.

**Note:**

SLB instance synchronization may encounter latency. Click Synchronize SLB in the upper-right corner of the page to manually update SLB instance information.

### 1.4.2.3 View a VPC

VPCs are created in the VPC console. After synchronizing resources in the EDAS console, you can view VPC information.

#### Context

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

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose Resource Management > VPC to go to the VPC page and view the VPC information and status.

Table 1-7: Instance information

Name	Description
VPC ID	The ID that is automatically generated when a VPC is created.
Name	The name that you set when creating a VPC.
CIDR	VPC statuses include Running and Stopped. Expired VPCs do not appear.
Status	The status of an SLB instance, which may be Running or Stopped. Expired SLB instances do not appear.

Name	Description
ECS Instance	The number of ECS instances created in this VPC. Click the number to go to the ECS page, where you can view all the ECS instances in this VPC.

## What's next

In the VPC, ECS instances are isolated from the EDAS server. You need to install a log collector to collect ECS instance information. Locate the row that contains the target instance, and click **Install Log Collector** in the **Actions** column. For the installation procedure, see [Install a log collector](#).

### 1.4.2.4 Manage clusters

A cluster is a set of ECS instances necessary to deploy applications. Cluster management mainly includes creating clusters, viewing clusters, and managing cluster hosts.

#### 1.4.2.4.1 Create an ECS cluster

Create a cluster before publishing applications.

## Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose **Resource Management > Clusters**.
3. On the **Clusters** page, click **EDAS Cluster**. On the **EDAS Cluster** tab page, click **Create Cluster** in the upper-right corner.
4. In the **Create Cluster** dialog box, set the cluster parameters and click **Create**.

Table 1-8: Cluster parameters

Name	Description
Cluster Name	Enter a name for the cluster. The name can only contain letters, numbers, underscores (_), and periods (.), with a length up to 64 characters.
Cluster	The options are Alibaba Cloud and Non-Alibaba Cloud. Select Alibaba Cloud in this case. Select Non-Alibaba Cloud when creating a hybrid cloud cluster.

Name	Description
Cluster Type	Currently, only ECS clusters are supported.
Network Type	Only VPC is supported.
VPC	Select a specific VPC.
Namespace	A namespace has been selected on the Clusters page, so this parameter cannot be set here.

## Result

After the cluster is created, the message Cluster created successfully appears in the upper-right corner of the page, and the cluster appears in the cluster list and is in the Normal state.

## What's next

Add ECS instances after the cluster is created.

1. On the Cluster Details page, click Add ECS Instance in the upper-right corner.
2. On the Add ECS Instance page, click Import ECS or From Existing Cluster to add ECS instances.
  - **Import ECS:** See [Import ECS instances](#).
  - **From Existing Cluster:** In the current region, select a namespace and source cluster. In the ECS instance list, select ECS instances and click > to add them to the field on the right. Then, click Next. The subsequent procedure is the same as that for importing ECS instances.
3. After ECS instances are added, return to the Cluster Details page to view the health status of the ECS instances. The ECS instances are successfully added if the health status is Normal.

### 1.4.2.4.2 Import a user-created Kubernetes cluster

Create a user-created Kubernetes cluster before publishing applications in it.

## Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose Resource Management > Clusters.

3. On the Clusters page, select a region and namespace and click the Self-Built K8S Cluster tab. On the tab page that appears, click Import into Self-Built K8S Cluster on the right.
4. In the Import into Self-Built Kubernetes Cluster dialog box, set parameters.
  - a) Cluster Name: Enter a name for the cluster.
  - b) ApiServer: Enter the public network address of the API server of the Kubernetes cluster. https// is the default prefix and cannot be set manually.
  - c) Generate Kubeconfig File Script:
    - A. Log on to the master node of the Kubernetes cluster as the root user.
    - B. Delete the `/root/importk8scluster.sh` and `edas-admin.kubeconfig` files.
    - C. Copy the script of the Kubeconfig file. The following parameters are for reference only. Change the values of parameters based on the actual script.

```
wget http://xxxxxx.importk8scluster.sh && sh importk8scluster  
.sh -ca= the CA of your cluster -key = the CA key of your  
cluster
```
  - d) Kubeconfig file: View the content of the `/root/edas-admin.kubeconfig` file by running a command and copy the file content and paste it in the Kubeconfig file field.
5. Click OK. The created cluster appears in the cluster list.

## Result

Return to the Cluster List page. If Cluster Status is Normal, the cluster is created and runs properly.

### 1.4.2.5 Manage resource groups

Resource groups are groups of EDAS resources, which can be ECS instances and SLB instances, but not VPCs. You can control account permissions through resource groups. You can grant resource group access permissions to the RAM users, and each RAM user has the permission to operate on all the resources in the specified group.

## Typical scenarios

- Assume that your company publishes its application systems through EDAS. Department A is responsible for user-related applications and Department B for goods-related ones.



- The company registers an EDAS account (the primary account) to activate EDAS and creates one RAM user each for Departments A and B.
- Departments A and B have dedicated ECS and SLB instances for deploying user-related applications and goods-related applications, respectively.
- You have created two resource groups in EDAS and bound them to the ECS and SLB instances of Departments A and B, respectively. Then, you grant the RAM users of Departments A and B the permissions to access the two resource groups, respectively.
- Department A uses its RAM user only to operate the ECS and SLB instances in the authorized resource group. Department B does the same for its resource group. There is no conflict between Departments A and B during resource management.

Create a resource group

1. In the left-side navigation pane, choose **Resource Management > Resource Groups**.
2. On the **Resource Groups** page, click **Create Resource Group** in the upper-right corner.
3. In the **Create Resource Group** dialog box, enter **Resource Group Name** and **Resource Group Description**, and click **OK**.

After the resource group is created, you can edit or delete it as needed.

Bind resources to resource groups

You can bind ECS instances, SLB instances, and clusters to resource groups. The procedures for binding different types of resources are similar. This topic describes how to bind Elastic Compute Service (ECS) instances.

1. On the **Resource Groups** page, 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 one or more ECS instances and click **OK**.

Grant RAM users the permissions to access resource groups

You can grant RAM users the permissions to access specified resource groups.

1. Log on to the EDAS console with your primary account.
2. In the left-side navigation pane, choose **Account Management > Sub-accounts**.
3. Locate the row that contains the target user, and click **Resource Group Permission** in the **Actions** column.

4. In the Resource Group Permission dialog box, select a resource group and click OK.

## 1.4.3 Manage applications

In the EDAS console, you can perform application lifecycle management, O&M, monitoring, and service governance.

### 1.4.3.1 Namespace

With namespaces, you can completely isolate the resources in different environments and use the same account to centrally manage them.

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose Namespace.
3. On the Namespace page, select a region and click Create Namespace in the upper-right corner.
4. In the Create Namespace dialog box, enter Namespace Name, Namespace ID, and Description (optional). Then, click OK.

#### Result

On the Namespace page, view the created namespace.

### 1.4.3.2 Lifecycle management for applications in ECS clusters

Applications are the basic units for EDAS management. A single application contains a group of instances on which the same application is deployed. EDAS provides a comprehensive application lifecycle management mechanism, covering the entire process from application publishing to operation, including application creation, deployment, startup, rollback, scaling, stop, and deletion.

Application lifecycle management includes application publishing, management, and configuration.

- Application publishing includes application creation, deployment, start, and stop.
- Application management includes application rollback, scale-out, scale-in, and deletion and instance reset and deletion.
- Application configuration includes container, JVM parameter, SLB, and health check configuration.



**Note:**

- You can deploy, scale out, roll back, reset, and configure an application no matter if the application is running or stopped.
- After the parameters of the Tomcat container and JVM are set and saved, the related configuration files are modified. The changes take effect only after you restart the application.

### 1.4.3.2.1 Publish an application

This topic describes how to publish an application in the EDAS console, helping you quickly familiarize yourself with EDAS operations and application publishing.



**Note:**

- If your EDAS service is deployed in Sugon, you can create an application in the Apsara Stack console or EDAS console.
  - If you create an application in the Apsara Stack console, your RAM user is authorized by default.
  - If you create an application in the EDAS console, you need to authorize your RAM user manually. For more information, see [Use a primary account for RAM user operation](#).

If required authorization is not performed in the EDAS console, an exception may occur when you manage applications in the Apsara Stack console.

- Applications must be managed in the EDAS console.
- If you publish an HSF application, create a service group before starting the application. Otherwise, application publishing may fail due to failed authentication. In the EDAS console, choose Service Market > Service Groups from the left-side navigation pane. On the page that appears, click Create Service Group in the upper-right corner to create a service group. The service group name must be globally unique. After the service group is created, restart the application to allow the service group to take effect.

#### *1.4.3.2.1.1 Create an application in the Apsara Stack console*

This topic describes how to create an application in the Apsara Stack console.

#### Procedure

1. [Log on to the EDAS console](#).

2. On the Applications page, click Create Application.
3. On the Create Application page, enter Application Name and select a department, project, container version, and region. Then, click OK.

### What's next

You need to log on to the EDAS console by using the account for the same department (with the same permissions) to manage the application created in the Apsara Stack console.

#### *1.4.3.2.1.2 Deploy an application (applicable to ECS clusters)*

### Prerequisites

- An ECS cluster has been created. For more information, see [Create an ECS cluster](#).
- ECS instances have been created. For more information, see [ECS User Guide](#).
- The ECS instances have been imported to EDAS. For more information, see [Import ECS instances](#).
- An SLB instance has been created. For more information, see [SLB User Guide](#).

### Procedure

1. [Log on to the EDAS console](#).
2. In the EDAS console, choose Application Management from the left-side navigation pane. On the Applications page, click Create Application in the upper-right corner.
3. On the Application Information page, set the parameters of the application. Then, click Next Step: Application Configurations.

Table 1-9: Basic parameters

Name	Description
Namespace	Select a region and namespace from the drop-down list.
Cluster Type	Select ECS Cluster from the drop-down list and select an ECS cluster.
Application Name	Enter a descriptive application name.
Deployment Method	The options are WAR and JAR.

Name	Description
Application Runtime Environment	<ul style="list-style-type: none"> <li>For an HSF application, select the EDAS-Container version.</li> <li>For a native Spring Cloud or Dubbo application, select Apache-Tomcat (applicable to WAR package deployment) or Default Environment (applicable to JAR package deployment).</li> </ul>
Java Runtime Environment	Select JDK 8 or JDK 7
Application Description	Enter remarks for the application.




**Note:**


You can click **Create an Empty Application** in the lower part of the page to create an ECS instance-free application and then add ECS instances and deploy the application.

#### 4. Deploy applications.

- a) On the Application Configuration page, click Add to the right of Selected Instances.
- b) In the Instances dialog box, select an ECS instance and click > to add the instance to the field on the right. Then, click OK.
- c) Return to the Application Configuration page and select Deploy Now.
- d) Set the deployment parameters below Deploy Now, and click Create.

Table 1-10: Deployment parameters


Name	Description
File Uploading Method	<p>Select Upload WAR Package or WAR Package Location.</p> <ul style="list-style-type: none"> <li>• <b>Upload WAR Package:</b> Click <a href="#">Download Sample WAR Package</a>. After the sample is downloaded, click Select File and select the WAR package.</li> <li>• <b>WAR Package Location:</b> Right-click <a href="#">Download Sample WAR Package</a> and choose Copy Link Address from the shortcut menu. Copy and paste the address in the WAR package address bar.</li> </ul> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p> <b>Note:</b></p> <p>The name of the application deployment package can only contain letters, numbers, hyphens (-), and underscores (_). The JAR package can be uploaded only when the JAR package deployment method is selected. Otherwise, you can only deploy the application by using the WAR package.</p> </div>

Name	Description
Enter Version	<p>Enter the version of the deployment package for the application.</p> <div>  <b>Note:</b>            You can add a version or textual description when deploying an application. We do not recommend using a timestamp as the version.         </div>
Version	Specify a version (such as 1.1.0). We do not recommend using a timestamp as the version.
(Optional) Application Health Check	<p>Set a URL for application health check. The system checks the health of the application after the container is started or is running. Then, it performs a service routing task based on the health check result. In this example, the health check URL is set to <code>http://127.0.0.1:8080/healthCheck.html</code>.</p>
Batch	Specify a number of deployment batches. Select an option from the drop-down list. The options are automatically generated based on the number of instances for the application. If you select two or more batches, you must set the batch wait time.
Batch Mode	Select Automatic.

- In the Application Settings section of the Basic Information page, click Add on the right of SLB (Public-facing). In the Bind SLB to Application dialog box, set the SLB parameters and then click Save.

Table 1-11: SLB configuration description

Name	Description
SLB	Select the internal network or public-facing SLB address from the drop-down list.

Name	Description
Use VServer Group	<p>A virtual server group contains a group of ECS instances for processing frontend requests distributed by SLB instances. Listeners can be associated with different virtual server groups to monitor request forwarding. If you select Use VServer Group, you must set virtual server group parameters.</p> <ul style="list-style-type: none"> <li>• <b>VServer Group:</b> Select an existing virtual server group from the drop-down list. If no virtual server group is available, click Create VServer Group in the drop-down list.</li> <li>• <b>VServer Group Name:</b> Enter a name for the new virtual server group if you select Create VServer Group. The system creates a virtual server group with the specified name.</li> </ul>
Listener	<p>A listener defines how to forward inbound requests 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 is available, click Create Listener.</p> <ul style="list-style-type: none"> <li>• <b>SLB Frontend Protocol:</b> The default value is TCP, which cannot be configured manually.</li> <li>• <b>SLB Frontend Port:</b> Enter the frontend port of the SLB instance.</li> <li>• <b>Application Port:</b> The default value is 8080, which cannot be configured manually.</li> </ul> <div>  <b>Notice:</b> Do not delete the listener in the SLB console. Otherwise, the application cannot be accessed normally.         </div>

## Result

- On the Instance Information tab page of the Application Details page, check the runtime status of the ECS instance. If Status or Time is Normal, the application is successfully deployed.
- To expose your application on the Internet, you must configure SLB (Internet). Click the Basic Information tab on the Application Details page. In the Application Settings section, copy the SLB (Internet) IP Address and Port, for example, `118.31.159.169:81`. Then, paste it in the address bar of your web browser



and press Enter. If the welcome page of the application appears, the application is successfully deployed.

### ***1.4.3.2.1.3 Create an application (applicable to ECS clusters)***

You can create an undeployed application during the planning phase and then deploy the application independently.

#### **Prerequisites**

An ECS cluster has been created. For more information, see [Create an ECS cluster](#).

#### **Context**

You can create an undeployed application in either of the following two states:

- **Instance-free application:** an empty application that is configured only with basic information, including a region, namespace, cluster, deployment method, and runtime environment.
- **Instance-based application:** an application that is configured with basic information (including a region, namespace, cluster, deployment method, and runtime environment) and ECS instances.

#### **Procedure**

1. [Log on to the EDAS console](#).
2. In the EDAS console, choose Application Management from the left-side navigation pane. On the Applications page, click Create Application in the upper-right corner.
3. On the Application Information page, set the parameters of the application. Then, click Next Step: Application Configurations.

Table 1-12: Basic parameters

Name	Description
Namespace	Select a region and namespace from the drop-down list.
Cluster Type	Select ECS Cluster from the drop-down list and select an ECS cluster.
Application Name	Enter a descriptive application name.
Deployment Method	The options are WAR and JAR.

Name	Description
Application Runtime Environment	<ul style="list-style-type: none"> <li>For an HSF application, select the EDAS-Container version.</li> <li>For a native Spring Cloud or Dubbo application, select Apache-Tomcat (applicable to WAR package deployment) or Default Environment (applicable to JAR package deployment).</li> </ul>
Java Runtime Environment	Select JDK 8 or JDK 7
Application Description	Enter remarks for the application.



**Note:**

You can click **Create an Empty Application** in the lower part of the page to create an ECS instance-free application and then add ECS instances and deploy the application.

- On the Application Configuration page, click **Add** to the right of **Selected Instances**.



**Note:**

If no instances have been added on the Application Configuration page, you can click **Create an Empty Application** to create an instance-free application.

- In the Instances dialog box, select an ECS instance and click **>** to add the instance to the field on the right. Then, click **OK**.
- Return to the Application Configuration page and click **Create**.

## Result

Return to the Application Details page to view the statuses of the application and instances.

- An instance-free application is an application that contains basic information, including the application name, ID, namespace, and deployment package type, but it does not contain instance information.
- An instance-based application is an application that contains basic information (including the application name, ID, namespace, and deployment package type) and the instance information and status.

## What's next

You can deploy the application after it is created. For more information, see

### 1.4.3.2.2 Manage applications

You can manage a published application in the EDAS console. This includes viewing application information, upgrading, starting, stopping, scaling out, and scaling in the application, creating branch versions, upgrading container versions, and rolling back and deleting the application. If the application is deployed on ECS instances, you need to manage those instances.

This topic briefly describes some simple management operations.

#### *1.4.3.2.2.1 Scaling (applicable to ECS clusters)*

If an application is overloaded, you can use the application scale-out function to manually scale out the application and share its load.

## Procedure

### Scale-out

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the target application.
3. On the **Application Details** page, click **Application Scale Out** in the upper-right corner.
4. On the **Scale-Out Method** tab page of the **Purchase Instances** dialog box, select the target group, the ECS instance, and the target instance for scale-out, and click **Scale Out**.



#### Note:

The runtime status of the added ECS instance depends on the runtime status of the application on the instance.

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

### Scale-in

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

**6. Scale in the instance on the Instance Information tab page.**

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

***1.4.3.2.2 Create an application branch version***

When you create an application, EDAS automatically creates an application group named "Default Group" for the application and adds the ECS instances of the application to this group. You can create subgroups under the default group and add some instances to the subgroups. If you deploy different versions of the application on the instances in the subgroups, these versions of the application are the branch versions of the application.

**Context**

You can create branch versions if you have the following requirements for your application:

- To perform an online test before publishing a new version
- A/B testing
- Canary deployment

**Procedure**

**1. Create a subgroup.**

a) *Log on to the EDAS console.*

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

c) On the Application Details page, click the Instance Information tab. On the tab page that appears, click Create Group in the upper-right corner.

d) In the Create Group dialog box, enter a group name and click Create.

After the group is created, the message Group created appears in the upper-right corner of the page.

**2. Add instances to a new group.**

After a group is created, you can add instances to the new group in two ways: Scale Out and Change Group. For more information about application scale-

out methods, see [Scaling \(applicable to ECS clusters\)](#). This topic describes how to add instances from the default group to the new group by changing the group.

- a) On the Instance Information tab page of the Application Details page, select the instance whose group you want to change, and click Change Group on the right of the list.
- b) In the Change Group dialog box, select an option for Target Group.
- c) Click Change Group.



**Note:**

- If no application is deployed in the new group while an application deployment package has been deployed on the added instance, this deployment package is deployed in the group.
- If an instance is added to an existing group rather than a new group, the versions of the deployment package in the group and on the instance are different. When the system displays the following messages, select the appropriate option as needed:
  - Select Redeploy current instance for target group to redeploy the deployment package on the instance using that in the group.
  - Select Change group without redeployment to add the instance without changing its deployment package.

**3. Deploy the application in the new group.**

- a) On the Application Details page, click Deploy Application in the upper-right corner.
- b) Based on [Deploy an application \(applicable to ECS clusters\)](#), set the target publish group as the new group, set the deployment parameters, and click Deploy.

**Result**

On the Instance Information tab page of the Application Details page, you can view the deployment package version and runtime status of the new group to check that the new application version is successfully published.

#### ***1.4.3.2.2.3 Upgrade the container version***

WAR and JAR packages are used for application deployment. The deployment involves an application runtime environment and the EDAS container. You can upgrade the EDAS container to the specified version.

##### **Procedure**

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose Application Management. On the Applications page, click the name of the target application.
3. On the Application Details page, choose Container Version from the left-side navigation pane.
4. On the Container Version page, view the current container version for the application.

The current version is marked with a tick (✓) in the Actions column. The Actions column also displays the availability status of other versions.

5. Click the corresponding button in the Actions column to upgrade the container to the desired version.

#### ***1.4.3.2.2.4 Roll back an application***

To roll back a published application to an earlier version, you can use the application rollback function and select the target version.

##### **Procedure**

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose Application Management.
3. On the Applications page, click the name of the target application. On the Application Details page, click Roll Back in the upper-right corner.
4. Based on the name of the published WAR package and the publishing time that appear on the Roll Back page, select the target version and click Roll Back.



##### **Note:**

The rollback target option appears only when you have deployed a beta instance. Otherwise, all instances under the application are rolled back by default. A maximum of five rollback versions appear.

#### ***1.4.3.2.2.5 Delete an application***

After an application is deleted, all information related to the application is deleted, all instances under the application are released, and all deployment packages and container files on the instances are deleted.

##### **Prerequisites**

Before deleting an application, be sure to save the logs, WAR packages, and configurations of all instances in the application.

##### **Procedure**

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose Application Management. On the Applications page, click the name of the target application.
3. Click Instance Information. On the Instance Information tab page, locate the row that contains the instance for the application, and click Delete in the Actions column.
4. Click Delete Application.

After the application is deleted, the message Deleted successfully appears in the upper-right corner of the page.

#### **1.4.3.2.3 Application settings**

On the Application Settings page, you can set the JVM parameters, Tomcat, SLB, and health check of applications.

##### ***1.4.3.2.3.1 Set JVM parameters***

By setting JVM parameters, you can enable the container parameter setting when an application is started. Correctly setting JVM parameters helps reduce the overhead of GC and thus shorten the server response time and improve throughput. If container parameters are not set, JVM parameters are allocated by default.

##### **Procedure**

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose Application Management. On the Applications page, click the name of the target application.
3. On the Application Details page, click Settings on the right of the Application Settings section.

4. On the JVM Parameters tab page of the Application Settings dialog box, click Memory Configuration, Applications, GC Policy, Tool, and Custom to set relevant parameters. Then, click Save.



**Note:**

The JVM parameter settings are written in the `bin/setenv.sh` file in the container directory. To apply the settings, restart the application.

## Result

After setting, the message Setting JVM successfully appears in the upper-right corner.

### 1.4.3.2.3.2 Configure Tomcat

EDAS supports Tomcat container parameter settings. You can configure settings such as the port number, application access path, and the number of connections in the connection pool of the Tomcat container in the EDAS console.

## Prerequisites



**Note:**

- After setting Tomcat container parameters, restart the container to apply the parameter settings.
- Tomcat container configuration is supported by EDAS Agent 2.8.0 and later.

## Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose Application Management. On the Applications page, click the name of the target application.
3. On the Application Details page, click Settings on the right of the Application Settings section.



4. In the Application Setting dialog box, click the Tomcat tab and set Tomcat parameters. Then, click Save.

Table 1-13: Tomcat configuration description

Name	Description
Application Port	The port range is (1024, 65535). The admin authority is needed for container configuration and the root authority is required to operate on 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	<p>The access path of an application.</p> <ul style="list-style-type: none"> <li>• If you select Package Name, you do not need to set a custom path. The default value is the WAR package name.</li> <li>• If you select Root, you do not need to set a custom path. The default value is a slash (/).</li> <li>• If you select Custom, enter a custom path below, namely, the access path. If this parameter is not set, the default value is the WAR package name.</li> </ul>
Maximum Threads	The maximum number of connections in a connection pool . The parameter is maxThreads. The default value is 400. We recommend that this parameter be set under professional guidance.
Tomcat encoding	Select an encoding format for Tomcat: UTF-8, ISO-8859-1, GBK, or GB2312. The default format is ISO-8859-1. Select useBodyEncodingForURI as needed.



**Note:**

Click Advanced Settings to configure the full text of `server.xml`. The application groups use the application configuration after the advanced settings are enabled.

### 1.4.3.3 Lifecycle management for Container Service Kubernetes applications

#### 1.4.3.3.1 Container Service Kubernetes clusters

Kubernetes is a popular orchestration technology for open source containers.

Kubernetes-published applications have unique management advantages. For more information, see the [Kubernetes official documentation](#).

A *Container Service Kubernetes cluster* is a Kubernetes cluster that is provided by Alibaba Cloud and has passed the CNCF standardized test. It runs stably and integrates other Alibaba Cloud services, such as SLB and Network Attached Storage (NAS). After creating a Kubernetes cluster in Container Service and importing it to EDAS, you can deploy applications to the Container Service Kubernetes cluster in EDAS.

#### 1.4.3.3.2 Prepare an application image (a Container Service Kubernetes cluster)

EDAS allows you to deploy RPC applications (HSF) in Container Service Kubernetes clusters by using custom images (Dockerfile).

Observe the following specifications and limits when creating a custom image by using a Dockerfile:

Tenant and encryption information

The tenant and encryption information is used for user authentication and credential encryption of EDAS applications.

Table 1-14: Resources

Resource type	Resource name	Description
Secret	edas-certs	An encryption dictionary that stores EDAS tenant information.

Table 1-15: Environment variables

Environment variable key	Type	Description
tenantId	String	The ID of an EDAS tenant.
accessKey	String	The AccessKeyId for authentication .

Environment variable key	Type	Description
secretKey	String	The AccessKeySecret for authentication.

Table 1-16: Local files

Path	Type	Description
/home/admin/.spas_key/default	File	The authentication information of an EDAS tenant, including the preceding environment variable information.

## Service information

The service information includes the EDAS domain and port to be connected during runtime.

Table 1-17: Resources

Resource type	Resource name	Description
ConfigMap	edas-envs	EDAS service information

Table 1-18: Environment variables

Environment variable key	Type	Description
EDAS_ADDRESS_SERVER_DOMAIN	String	The service domain or IP address of the configuration center.
EDAS_ADDRESS_SERVER_PORT	String	The service port of the configuration center.
EDAS_CONFIGSERVER_CLIENT_PORT	String	The port of ConfigServer.

## (Mandatory) Environment variables during application runtime

The following environment variables are provided during EDAS deployment to ensure the proper running of applications. For this reason, do not overwrite the current configuration.

Table 1-19: Environment variables

Environment variable key	Type	Description
POD_IP	String	The IP address of a pod.
EDAS_APP_ID	String	The ID of an EDAS application.
EDAS_ECC_ID	String	EDAS ECC ID
EDAS_PROJECT_NAME	String	Same as EDAS_APP_ID and used for trace parsing.
EDAS_JM_CONTAINER_ID	String	Same as EDAS_ECC_ID and used for trace parsing .
EDAS_CATALINA_OPTS	String	The CATALINA_OPTS parameter required during middleware runtime.
CATALINA_OPTS	String	The default startup parameter of Tomcat , which is the same as EDAS_CATALINA_OPTS.

## Procedure

### 1. Define a standard Dockerfile.

A standard [Dockerfile](#) defines the EDAS application runtime environment, including the definitions of download, installation, JDK startup, Tomcat, and WAR and JAR packages.

By modifying the Dockerfile, you can replace the JDK version, modify the Tomcat configuration, change the runtime environment, and make other changes.

The following example shows how to define an EDAS application.



#### Note:

The example will be occasionally updated to incorporate the latest EDAS features.

- Sample Dockerfile that uses Tomcat and a WAR package

```
FROM centos:7
MAINTAINER EDAS development team <edas-dev@list.alibaba-inc.com>
```

```
# Install and package the required software.
RUN yum -y install wget unzip
# Prepare JDK and Tomcat system variables.
ENV JAVA_HOME /usr/java/latest
ENV CATALINA_HOME /home/admin/taobao-tomcat
ENV PATH $PATH:$JAVA_HOME/bin:$CATALINA_HOME/bin
# Set the EDAS-Container version.
ENV EDAS_CONTAINER_VERSION V3.5.0
LABEL pandora V3.5.0
# Download and install JDK 8.
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/agent/prod/
files/jdk-8u65-linux-x64.rpm -O /tmp/jdk-8u65-linux-x64.rpm && \
    yum -y install /tmp/jdk-8u65-linux-x64.rpm && \
    rm -rf /tmp/jdk-8u65-linux-x64.rpm
# Download and install Ali-Tomcat 7.0.85 to the /home/admin/taobao
-tomcat.
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-
container/7.0.85/taobao-tomcat-production-7.0.85.tar.gz -O /tmp/
taobao-tomcat.tar.gz && \
    mkdir -p ${CATALINA_HOME} && \
    tar -xvf /tmp/taobao-tomcat.tar.gz -C ${CATALINA_HOME} && \
    mv ${CATALINA_HOME}/taobao-tomcat-production-7.0.59.3/* ${
CATALINA_HOME}/ && \
    rm -rf /tmp/taobao-tomcat.tar.gz ${CATALINA_HOME}/taobao-
tomcat-production-7.0.59.3 && \
    chmod +x ${CATALINA_HOME}/bin/*sh
# Download and install an EDAS container based on environment
variables.
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-plugins/
edas.sar. ${EDAS_CONTAINER_VERSION}/taobao-hsf.tgz -O /tmp/taobao-
hsf.tgz && \
    tar -xvf /tmp/taobao-hsf.tgz -C ${CATALINA_HOME}/deploy/ && \
    rm -rf /tmp/taobao-hsf.tgz
# Downloads and deploys the EDAS demo WAR package.
RUN wget http://edas.oss-cn-hangzhou.aliyuncs.com/demo/hello-edas.
war -O /tmp/ROOT.war && \
    unzip /tmp/ROOT.war -d ${CATALINA_HOME}/deploy/ROOT/ && \
    rm -rf /tmp/ROOT.war
# Set the Tomcat installation directory as the container startup
directory, start Tomcat in run mode, and output the catalina log
in the standard CLI.
WORKDIR $CATALINA_HOME
CMD ["catalina.sh", "run"]
```

- **Sample Dockerfile that uses a JAR package**

```
FROM centos:7
MAINTAINER EDAS development team <edas-dev@list.alibaba-inc.com>
# Install and package the required software.
RUN yum -y install wget unzip
# Prepare JDK and Tomcat system variables.
ENV JAVA_HOME /usr/java/latest
ENV CATALINA_HOME /home/admin/taobao-tomcat
ENV PATH $PATH:$JAVA_HOME/bin
# Set the EDAS-Container version.
ENV EDAS_CONTAINER_VERSION V3.5.0
LABEL pandora V3.5.0
# Download and install JDK 8.
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/agent/prod/
files/jdk-8u65-linux-x64.rpm -O /tmp/jdk-8u65-linux-x64.rpm && \
    yum -y install /tmp/jdk-8u65-linux-x64.rpm && \
    rm -rf /tmp/jdk-8u65-linux-x64.rpm
```

```
# Download and install an EDAS container to /home/admin/taobao-  
tomcat based on environment variables.  
RUN mkdir -p ${CATALINA_HOME}/deploy/  
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-plugins/  
edas.sar. ${EDAS_CONTAINER_VERSION}/taobao-hsf.tgz -O /tmp/taobao-  
hsf.tgz && \  
    tar -xvf /tmp/taobao-hsf.tgz -C ${CATALINA_HOME}/deploy/ && \  
    rm -rf /tmp/taobao-hsf.tgz  
# Download and deploy the EDAS demo JAR package.  
RUN mkdir -p /home/admin/app/ && wget http://edas.oss-cn-hangzhou  
.aliyuncs.com/demoapp/fatjar-test-case-provider-0.0.1-SNAPSHOT.jar  
-O /home/admin/app/provider.jar  
# Include the startup command in the startup script start.sh.  
RUN echo '$JAVA_HOME/bin/java -jar $CATALINA_OPTS -Djava.security  
.egd=file:/dev/./urandom -Dcatalina.logs=$CATALINA_HOME/logs -  
Dpandora.location=$CATALINA_HOME/deploy/taobao-hsf.sar "/home/  
admin/app/provider.jar" --server.context-path=/ --server.port=  
8080 --server.tomcat.uri-encoding=ISO-8859-1 --server.tomcat.max-  
threads=400' > /home/admin/start.sh && chmod +x /home/admin/start.  
sh  
WORKDIR $CATALINA_HOME  
CMD ["/bin/bash", "/home/admin/start.sh"]
```

## 2. Customize settings in the Dockerfile.

The following describes how to customize settings in the standard Dockerfile prepared previously.

### a) Upgrade JDK.

Change the download and installation methods in the standard Dockerfile.

The following uses JDK 8 as an example.

```
# Download and install JDK 8.  
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/agent/prod/  
files/jdk-7u80-linux-x64.rpm -O /tmp/jdk-7u80-linux-x64.rpm && \  
    yum -y install /tmp/jdk-7u80-linux-x64.rpm && \  
    rm -rf /tmp/jdk-7u80-linux-x64.rpm
```

### b) Upgrade EDAS Java Container.

When using a WAR package and Tomcat, upgrade the EDAS container to use new middleware features or fix known bugs. The upgrade procedure is as follows:

- A. Locate the latest version (3.X.X) of the EDAS container.
- B. Replace the version in the Dockerfile, such as 3.5.0.
- C. Recreate and publish an application image.

```
# Prepare ENV
```

```
ENV EDAS_CONTAINER_VERSION V3.5.0
```

c) Add the EDAS runtime environment to Tomcat startup parameters.

See [\(Mandatory\) Environment variables during application runtime](#). EDAS provides the JVM environment variable `EDAS_CATALINA_OPTS`, which contains the minimum parameters required during runtime. Tomcat provides the custom JVM parameter configuration option `JAVA_OPTS` for setting `xmx`, `xms`, and other parameters.

```
# Set the JVM parameters of the EDAS application.
ENV CATALINA_OPTS ${EDAS_CATALINA_OPTS}
# Set the JVM parameters.
ENV JAVA_OPTS="\
    -Xmx3550m \
    -Xms3550m \
    -Xmn2g \
    -Xss128k"
```

### 1.4.3.3.3 Deploy an application (applicable to Container Service Kubernetes clusters)

You can deploy applications in a Container Service Kubernetes cluster.

#### Prerequisites

- [Prepare an application image \(a Container Service Kubernetes cluster\)](#) is complete, and the image has been pushed to the container image repository.
- The Container Service Kubernetes cluster has been imported to EDAS.

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click **Create Application** in the upper-right corner.
3. On the **Application Information** page, set the parameters of the application. Then, click **Next Step: Application Configurations**.

Table 1-20: Basic parameters

Name	Description
Namespace	Select a namespace from the drop-down list.
Deploy Cluster	Select a Container Service Kubernetes cluster from the drop-down list.

Name	Description
Application Type	The application type is determined by the cluster where the application is deployed. If you select a Container Service Kubernetes cluster, the application type is Kubernetes application. This parameter cannot be set manually.
Application Name	Enter a descriptive application name.
Application Description	Enter remarks for the application.

4. On the Application Configuration page, select Image for Deployment Method, select an image and a version, and click Select.
5. Set Total Pods and Single Pod Resource Quota (CPU cores and Memory).
6. Drag the slider on the right of Advanced Setting to the right to set advanced parameters. Then, click Next Step: Application Access Settings.
  - a) Optional: Set the startup command and parameters.



**Note:**

If you do not know the *CMD* and *ENTRYPOINT* content of the original Dockerfile image, do not modify the custom startup command and parameters.



Otherwise, you cannot create applications due to an incorrect custom command.

- **Startup Command:** Enter the content in [“”]. For example, set Startup Command to `/usr/sbin/sshd -D` for CMD [“/usr/sbin/sshd”, “-D”].
- **Startup Parameters:** Enter one parameter per line. For example, args : [“-c”; “while sleep 2”; “do echo date”; “done”] contains four parameters. In this case, enter the parameters in four lines.

**b) Optional: Set environment variables.**

When creating the application, inject the environment variables you have entered to the container to be generated. This saves you from repeatedly adding common environment variables.

**c) Optional: (Applicable to stateful applications) Set the application lifecycle management script.**

**Lifecycle management scripts:**

- **PreStop script:** This is a container hook, which is triggered before a container is deleted. The corresponding hook handler must be completed before the container deletion request is sent to Docker daemon. Docker daemon sends an SGTERN semaphore to itself to delete the container, regardless of the hook handler execution result. For more information, see [Container Lifecycle Hooks](#)
- **Liveness script:** This is a container status probe, which monitors the health status of applications. If an application is unhealthy, the container is deleted and created again. For more information, see [Pod Lifecycle](#)
- **Readiness script:** This is a container status probe, which monitors whether applications have started successfully and are running properly. If an application is abnormal, the container status is updated. For more information, see [Pod Lifecycle](#)
- **Poststart script:** This is a container hook, which is triggered immediately after a container is created to notify the container of its creation. The hook does not pass any parameters to the corresponding hook handler. If the corresponding hook handler fails to run, the container is killed and the system determines whether to restart the container according to the restart policy of the container. For more information, see [Container Lifecycle Hooks](#)

**7. On the Application Access Settings page, set SLB and click Create.**

SLB corresponds to TCP/UDP settings. You can configure multiple port mappings for multi-port listening.

- **Intranet SLB:** This option ensures that all the nodes in a VPC can access the application.
- **Public-facing SLB:** After you enable this option, the system buys a public-facing SLB instance for the application to ensure that the application is accessible from the Internet.

SLB parameters:

- **SLB Port:** This parameter indicates the frontend port of the internal network or public-facing SLB instance, which is used to access the application. For example, NGINX uses port 80 by default.
- **Container Port:** This is the port that listens to processes. It is generally defined by the program. For example, the web service uses port 80 or 8080 by default, while the MySQL service uses port 3306 by default. The container port can be the same as the port used by the SLB instance.
- **Network Protocol:** You can select TCP or UDP.

**Result**

Return to the Applications page and check whether the created application is running properly.

#### 1.4.3.3.4 Scaling (applicable to Container Service Kubernetes clusters)

Compared with common applications, Kubernetes applications feature much greater scalability due to the advantages of Kubernetes in container orchestration.

**Procedure**

1. Log on to the EDAS console and choose Application Management from the left-side navigation pane.
2. On the Application Management page, click the target Container Service Kubernetes application.
3. On the Application Details page, click Application Scaling in the upper-right corner.
4. In the Application Scaling dialog box, set Total Application Pods and click OK.

## Result

A message that indicates successful operation appears after scaling is complete. Return to the Application Details page and click Instance Information to view the instance information and runtime status after scaling.

### 1.4.3.4 Lifecycle management for applications in a user-created Kubernetes cluster

#### 1.4.3.4.1 Deploy an application (applicable to user-created Kubernetes clusters)

After you import a user-created Kubernetes cluster, you can deploy applications in the cluster by using images.

## Prerequisites

*A user-created Kubernetes cluster has been imported.*

## Procedure

1. *Log on to the EDAS console.*
2. In the left-side navigation pane, choose Application Management. On the Applications page, click Create Application in the upper-right corner.
3. On the Application Information page, set the basic application information and parameters, and click Next Step: Application Configurations.
  - **Namespace:** Select a region and namespace from the drop-down list. If you do not select any namespace, the default namespace is automatically selected.
  - **Cluster Type:** Select Self-Built K8S Cluster from the left-side drop-down list, and select a cluster from the right-side drop-down list.
  - **K8S Namespace:** Internal system objects are allocated to different namespaces to form logically isolated projects, groups, or user groups. In this way,

different groups can share resources of the whole cluster while being managed separately.

- **default:** When the object is not set with a namespace, "default" is used.
- **kube-system:** The namespace used by objects that are created by the system.
- **kube-public:** The namespace that is automatically created by the system. It can be read by all users, including users that are not authenticated.
- **Application Name:** The name of the application.
- **Application Description:** Enter the basic information of the application. The maximum length of the description is 128 characters.

4. Go to the Application Configuration page and configure an image. By default, Image is selected for Deployment Method.

a) In the Configure Image section, select an image repository and an image.

- **Alibaba Cloud Image:** Select Alibaba Cloud Image, select a project of the Alibaba Cloud image repository from the Project drop-down list, and select an image and a version in the lower part.
- **Third-Party Image:** It mainly refers to *Harbor*.

A. Select Third-Party Image.

B. In the Connect to Third-Party Image Warehouse dialog box, enter the third-party image address, logon account, and password, and click Test Connectivity. The images in the repository are displayed after the test is successful.

C. Select a Harbor project from the Project drop-down list, and select an image and a version in the lower part.

## 5. Set pods.

Pods are the smallest units for deploying an application. An application can contain multiple pods. In an SLB instance, a request is randomly allocated to a pod for processing.

### a) Set Total Pods:

When a pod fails to run or encounters a fault, it can automatically restart or services on the pod seamlessly fail over to other pods, ensuring a high availability for applications. For stateful applications that use persistent storage, instance data is retained when the applications are redeployed. For stateless applications, instance data is not retained when the applications are redeployed. You can set up to 50 pods.

### b) Set Single Pod Resource Quota:

No quota is set by default. Therefore, both the CPU Cores and Memory values of a single pod are 0. To set the quota, enter a number.

## 6. Set the startup command and parameters.



### Note:

If you do not know the **CMD** and **ENTRYPOINT** content of the original Dockerfile image, do not modify the custom startup command and parameters. Otherwise, you cannot create applications due to an incorrect custom command.

- **Startup Command:** Enter the startup command. To run the CMD `["/usr/sbin/sshd", "-D"]` command, enter `/usr/sbin/sshd -D` in the text box.
- **Startup Parameters:** Enter one parameter per line. For example, `args: ["-c"; "while sleep 2"; "do echo date"; "done"]` contains four parameters. In this case, enter the parameters in four lines.

## **7. Set environment variables.**

When creating the application, inject the environment variables you have entered to the container to be generated. This saves you from repeatedly adding common environment variables.

If you are using a MySQL image, refer to the following environment variables:

- **MYSQL\_ROOT\_PASSWORD (required):** allows you to set a root password for MySQL.
- **MYSQL\_USER and MYSQL\_PASSWORD (optional):** allow you to add an account besides the root account and set a password.
- **MYSQL\_DATABASE (optional):** allows you to set the database that you want to create when the container is generated.

If you are using another type of image, configure the environment variables as needed.

## **8. Local Storage: Set Host Mount Directory, Mount Directory in the Container, and Read Permission.**

- a. The host path is the path to an empty local disk volume and used to temporarily store and share runtime data. Volumes are deleted when application pods are deleted or migrated.
- b. The local disk volume with a host path specified is used to store data persistently on the host where the container is located. The volume is still on the original host after application pods are migrated.
- c. When the host node is abnormal and cannot be restored, the data in the local disk cannot be restored either.

## 9. Set the application lifecycle management script:

The Container Service Kubernetes cluster supports stateless applications and stateful applications.

- **Stateless:** A stateless application supports multi-replica deployment. When a stateless application is redeployed, instance data is not retained. A stateless application can be either of the following applications:
  - A web application that does not retain instance data during upgrade or migration.
  - An application that can be scaled out to address changing service volumes.
- **Stateful:** A stateful application stores data that requires persistent storage and retains instance data during upgrade or migration. A stateful application can be either of the following applications:
  - An application that frequently operates on containers through SSH.
  - An application that requires persistent data storage (such as applications using MySQL) or that supports inter-cluster election and service discovery, such as ZooKeeper and etcd.

You can set lifecycle management for a stateful application as needed.

### Lifecycle management scripts:

- **Poststart script:** This is a container hook, which is triggered immediately after a container is created to notify the container of its creation. The hook does not pass any parameters to the corresponding hook handler. If the corresponding hook handler fails to run, the container is killed and the system determines whether to restart the container according to the restart policy of the container. For more information, see [Container Lifecycle Hooks](#)
- **PreStop script:** This is a container hook, which is triggered before a container is deleted. The corresponding hook handler must be completed before the container deletion request is sent to Docker daemon. Docker daemon sends

an SGTERN semaphore to itself to delete the container, regardless of the hook handler execution result. For more information, see [Container Lifecycle Hooks](#)

- **Liveness script:** This is a container status probe, which monitors the health status of applications. If an application is unhealthy, the container is deleted and created again. For more information, see [Pod Lifecycle](#)
- **Readiness script:** This is a container status probe, which monitors whether applications have started successfully and are running properly. If an application is abnormal, the container status is updated. For more information, see [Pod Lifecycle](#)

10. Then, click Create.

## Result

Creating an application may take up to several minutes. During the creation process, you can use an application change order to track the creation progress. Kubernetes applications do not need to be deployed. The deployment is complete after creation. After an application is created, return to the Application Details page to view the pod status in the Instance Information section. If the pod status is Running, the application is successfully deployed.

### 1.4.3.4.2 Application management (applicable to user-created standard Kubernetes clusters)

After a user-created standard Kubernetes application is published, you can deploy the application to upgrade the application, scale the application to change the number of instances, and delete the application to delete the application information and release the instance.

## Update an application

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose Application Management > Applications. On the Applications page, click the name of the target application.
3. On the Application Details page, click Deploy Application in the upper-right corner.
4. On the Deploy Application page, you can change the settings of Configure Image and reset the number of instances, startup command, environment variables, local storage, and application lifecycle.



5. After you complete the parameter settings, click OK. After the application is deployed, you can track and view the deployment process on the Change Details page. After the change process ends successfully, return to the Application Details page to view the deployed image version and check whether the application is successfully deployed.

#### Scale out or scale in an application

Compared with common applications, Kubernetes applications feature much greater scalability due to the advantages of Kubernetes in container orchestration.

1. Log on to the EDAS console and choose Application Management from the left-side navigation pane.
2. On the Application Management page, click the target user-created Kubernetes application.
3. On the Application Details page, click Application Scaling in the upper-right corner.
4. In the Application Scaling dialog box, set Total Application Pods.
5. Click OK. A message that indicates successful operation appears after scaling is complete. Return to the Application Details page and click Instance Information to view the instance information and runtime status after scaling.

#### Delete an application

After an application is deleted, all information related to the application is deleted, all instances under the application are released, and all deployment packages and container files on the instances are deleted.

1. Log on to the EDAS console and choose Application Management from the left-side navigation pane.
2. On the Application Management page, click the target user-created Kubernetes application.
3. On the Application Details page, click Delete Application in the upper-right corner.
4. In the window that appears, click Delete. After the application is deleted, the message Application deletion triggered successfully appears in the upper part of the page.

### 1.4.3.5 Log management

The EDAS console provides the runtime log function, allowing you to view the runtime logs of applications without having to log on to the ECS instance. When an exception occurs in an application, you can check logs to troubleshoot the problem.

#### Procedure

1. *Log on to the EDAS console.*
2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the target application.
3. On the **Application Details** page, choose **Log Management > Log Directories** from the left-side navigation pane.

By default, the **Log Directories** page contains two log paths: the log path of the Tomcat container (such as `/home/admin/taobao-tomcat-production-2.0.59.4/logs`) and the log path of EDAS Agent (such as `/home/admin/edas-agent/logs`).  
Tomcat The path to container logs varies depending on the actual version.

4. Click the log folder or path to show all log files in the folder.



**Note:**

Only readable files but not folders are displayed.

5. Double-click a log file to view log details.
  - Select an instance from the ECS Instance ID/Name/IP drop-down list to view its real-time logs.
  - Click **Enable Real-time Additions** in the lower-right corner of the page to ensure that the latest additions to the file have been added (similar to the `tail -f` ).
6. Optional: Bookmark a log path.
  - a) On the **Log Directories** page, select a path or folder and click **Bookmark Log Directory** in the upper-right corner of the page.
  - b) In the **Add Application Log Path** dialog box, enter an application log path and click **Add**.



**Notice:**

- The path must be in the `/home/admin` directory and contain "log" or "logs".

- **The file name must end with a slash (/) to indicate that it is a folder.**

To cancel the bookmark status, click the name of a folder in the selected directory and click **Remove Directory from Bookmark** in the upper-right corner of the page. When a path is removed from favorites, it is no longer displayed on the logs page. This operation does not delete or change any files on the server.

#### 1.4.3.6 Application monitoring

Application monitoring can accurately reflect the real-time traffic and historical information of an application, allowing you to monitor application health and quickly locate faults.

##### Terms

- **TraceId:** the trace ID corresponds to a request. It is globally unique and transmitted between systems.
  - **IP address:** the IP address (in hexadecimal format) of the ECS instance that creates the TraceId.
  - **Created at:** the time when a trace is created.
  - **Sequence number:** the sequence number that is used for trace sampling.
  - **Flag bit:** (optional) the flag bit that is used for debugging and marking.
  - **Process ID:** (optional) the process ID that is used by single-instance multi-process applications.
- **RpcId:** the RPC ID that flags the call log tracking sequence and nesting relationship. It is transmitted between systems.
- **Service dimension:** Service monitoring monitors data 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, you have an application A that provides services a, b, and c.
- **Application drilling down:** a feature that shows the metrics of upstream and downstream applications associated with the target metric.

Table 1-21: Types of monitoring data

Name	Description
Summary	Contains the overall information about services that are provided and called by an application.
Entrance	Displays entrances provided by an application. (An entrance indicates a front-end request, which is based on HTTP in most cases.)
RPC Services Provided	Displays the RPC services (including the HSF and other custom services) provided by an application as the provider.
RPC Call Source	Displays the consumer call status of an application as the provider.
RPC Call Dependency	Displays the RPC services (including the HSF and other consumer services) that are called by an application as the consumer.
Database Accesses	Displays the database access by an application as the consumer.
Message Type	Displays the messages that are generated and consumed by an application.

#### Types of monitoring statistics

- **(Default) Block:** Statistics of the block type are displayed in a table and graph. Information contained includes the monitoring target, time, QPS, elapsed time, elapsed time of the provider, errors, and results. By default, the graph shows the data from the past hour, and the table shows the data from the past five minutes.
- **Multi-graph:** Statistics of the multi-graph type are displayed in graphs. The information contained includes the monitoring target, time, QPS, elapsed time, errors, and results. By default, the graphs show the data from the past hour, and the latest data is also listed.
- **Table:** Statistics of the table type are displayed in a table. Information contained includes the monitoring target, QPS, elapsed time, errors, and results. Data from the past minute is displayed.

## Metrics

- **Time:** displays data within a particular minute. For example, 08:00 indicates that the data from 8:00:00 to 8:00:59 is displayed.
- **QPS:** indicates the average value of the access volume per second within 1 minute. The formula for calculating this metric is as follows:  $\text{QPS} = \text{Total access volume within the minute} / 60$ .
- **Elapsed Time:** indicates the average access time per minute recorded on the consumer, in ms. The formula for calculating this metric is as follows:  $\text{Elapsed time} = \text{Total elapsed time for all accesses within the minute} / \text{Total access volume}$ .
- **Elapsed Time on Server:** indicates the average elapsed time per minute recorded on the provider, in milliseconds. The formula for calculating this metric is the same as that for calculating Elapsed Time.
- **Errors:** indicates the number of RPC errors per minute. The formula for calculating this metric is as follows:  $\text{Errors} = \text{Total number of errors within the minute} / 60$ .
- **Result:** indicates the returned result in the format of "Result: QPS", where "Result" indicates the RPC result. The HTTP result is consistent with the HTTP ErrorCode.

### 1.4.3.6.1 Install a log collector

EDAS provides a suite of functions that pull a lot of data from ECS instances. This requires that the EDAS cluster be connected to the relevant instances. In a VPC, a log collector connects EDAS clusters to ECS instances.

#### Prerequisites

The log collector must be installed on an ECS instance in the VPC. Ensure that the RAM user is authorized. For the authorization procedure, see *the Apsara Stack Console User Guide* and read the **RAM management** topic.

#### Context

You can connect a VPC to EDAS clusters by installing the log collector on an ECS instance in the VPC. The installation procedure is as follows:

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose Resource Management > VPC.

3. On the VPC Instances page, click Install Log Collector.
4. Copy the installation script in the Install Log Collector dialog box.
5. On the VPC Instances page, click the quantity of ECS instances on which log collectors are installed in the ECS Instance column. On the ECS Instances page, view and record the ID, name, or IP address of an ECS instance.
6. Log on to the ECS console and click the Instance tab. In the instance list, click the ID of the ECS instance on which the log collector is to be installed.
7. On the Instance Details page, click Log on to VNC in the upper-right corner. On the Change VNC Logon Password page, enter the VNC password and click Submit.
8. Log on to the ECS instance as the root user and use the logon password that is set when the ECS instance is bought.
9. After logging on to the console, copy and run the log collector installation script.

## Result

After the installation is completed, manually run the `netstat -antp|grep beam` command to check the installation result.

- If a connection is established, the log collector is successfully installed.
- If no connection is established, this indicates installation encountered a problem. In this case, open a ticket to seek help from Customer Services.

## 1.4.3.6.2 Dashboard

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

## Context

- **Service provisioning:** displays the metrics for the RPC and HTTP services.
- **Service consumption:** displays the metrics for database access.
- **Infrastructure monitoring:** displays the metrics for CPU, load, memory, disk, and network.

## Procedure

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose Application Management. On the Applications page, click the name of the target application.

3. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the application to be monitored.
4. On the **Application Details** page, choose **Application Monitoring > Dashboard** from the left-side navigation pane.

On the **Dashboard** page, you can view the monitoring charts for service provisioning, service consumption, and infrastructure monitoring.

- Place the pointer over a point on an abscissa of a monitoring chart to view the data and status at that time point.
- Click a project name, such as **RPC Service**, at the top of a monitoring chart to switch to the **Service Monitoring** tab and view details.

### 1.4.3.6.3 Infrastructure monitoring

EDAS collects data from the ECS instances that run applications and provides the CPU, memory, load, network, and disk metrics by instance or cluster based on the analysis results.

#### Prerequisites

Infrastructure monitoring involves ECS instances. Ensure that your RAM user is authorized. For the authorization procedure, see *the Apsara Stack Console User Guide and read the RAM management topic*.

#### Context

Due to the latency between data collection and data analysis, EDAS cannot provide real-time dashboards. The current latency is 2 minutes. All monitoring data is collected and processed by application.

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the target application.
3. On the **Application Details** page, choose **Application Monitoring > Infrastructure Monitoring** from the left-side navigation pane.

On the **Infrastructure Monitoring** page, cluster data from the past half an hour appears by default.

#### **4. Select a type of monitoring data.**

**Monitoring data includes group data and single-instance data.**

**These two data types correspond to the same set of monitoring metrics related to clusters (groups) and instances (single instance), including:**

- a. CPU Data indicates the CPU usage, which is the sum of the user utilization and system utilization. The group data graph shows the average value of this data for all instances in the cluster.**
- b. Memory Data indicates the total size and actual utilization of the physical memory. The group data graph shows the sum of this data for all instances in the cluster.**
- c. Load Data indicates the "1 min load" field in the system load. The group data graph shows the average value of this data for all instances in the cluster.**
- d. Network Speed Data indicates the read and write speeds of the NIC. If an ECS instance contains multiple NICs, this parameter indicates the total read and write speeds of all NICs whose names start with "eth". The group data graph shows the average value of this data for all instances in the cluster.**
- e. Disk Data indicates the total size and actual utilization of all disks attached to the instance. The group data graph shows the value of this data for all instances in the cluster.**
- f. Disk Reading and Writing Speed indicates the sum of the read and write speeds of all disks attached to the instance. The group data graph displays the average value of the data for all instances in the cluster.**
- g. Disk Reading and Writing Numbers indicates the sum of the input/output per second (IOPS) of all disks attached to the instance. The group data graph displays the average value of the data for all instances in the cluster.**

#### **5. Set Time Interval.**

**You can set Time Interval to Half an Hour, 6 Hours, One Day, or 1 Week.**

- Half an Hour: collects monitoring data from the past half an hour. Time Interval is set to Half an Hour by default for infrastructure monitoring. In**



this statistical cycle, data is collected every minute, which is the finest query granularity provided by EDAS.

- **6 Hours:** collects monitoring data from the past 6 hours. In this statistical cycle, data is collected every 5 minutes.
- **One Day:** collects monitoring data from the past 24 hours. In this statistical cycle, data is collected every 15 minutes.
- **1 Week:** collects monitoring data from the past seven days. In this statistical cycle, data is collected every hour, which is the longest statistical cycle provided by EDAS.



**Note:**

Start Time and End Time on the page indicate the time span of the current view. When you set one of the parameters, the corresponding parameter is automatically updated. For example, if you select Half an Hour and set End Time to 2016-05-20 12:00:00, then Start Time automatically changes to 2016-05-20 11:30:00.

After setting, monitoring data is automatically updated based on the selected interval.

**6. Optional: View the enlarged graph of a detailed metric.**

When viewing a dashboard, you can click **Zoom In** under a metric to view the enlarged graph of the metric, and adjust the interval in the enlarged graph.

### 1.4.3.6.4 Service monitoring

By collecting and analyzing tracked logs in different network call middleware products, you can obtain the traces of systems for a single request. This helps sort out application request portals and service call sources and dependencies, analyze system call bottlenecks, estimate the link capacity, and quickly locate exceptions.

#### Procedure

##### Monitoring service

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose **Application Management**.
3. On the **Applications** page, click the name of the application to be monitored.

4. On the Application Details page, choose Application Monitoring > Service Monitoring from the left-side navigation pane.

Monitoring metrics:

- **RPC Services Provided:** displays the call records on 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.

The Summary tab page of the default group appears by default.

5. (Optional) Set the monitoring conditions and click Update to update the monitoring data.

Name	Description
Latest	The data at the current time is displayed by default. Select a period from the drop-down list.
Sort By	Data is sorted by QPS by default. Select an option from the drop-down list to sort data by the elapsed time or errors/s (average QPS errors per minute).
Results	10 is selected by default. Select the number of results to be displayed from the drop-down list. The 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 multi-graph or table.

6. View monitoring data.

For information about monitoring metrics, see [Application monitoring](#).

7. Click a metric of a column in the monitoring graph. The custom query page appears. You can view the monitoring data of the metric.

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

View traces

9. In the monitoring graph, click View Trace next to a call service or called service and choose Trace Analysis > Trace Query.

10. On the Trace Query page, you can view the traces between the application and the call service or called service.

## Monitor drilled-down applications

**11.** On the **RPC Services Provided**, **RPC Call Source**, or **RPC Call Dependency** tab page, click **Source Application**, **Called Service**, or **Call Service** next to **Drill Down** at the top of the monitoring graph. The monitoring page of the drilled-down application appears.

**12.** Monitor data of the drilled-down application.

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

### 1.4.3.6.5 Advanced monitoring

**Application Real-Time Monitoring Service (ARMS)** is an application performance management (APM) monitoring product of Alibaba Cloud. EDAS can seamlessly interoperate with ARMS through advanced monitoring.

You can enable advanced monitoring to provide the APM function of ARMS to applications in EDAS for phased performance management of the applications. For more information, see *ARMS User Guide*.



#### Note:

By default, the native Spring Cloud and Dubbo applications in EDAS are monitored by ARMS.

### 1.4.3.7 Notifications and alarms (only applicable to HSF applications in ECS clusters)

The notification and alarm function analyzes and evaluates collected data based on your configured rules. When some resources are overused, alarms are triggered and notifications are sent to contacts by SMS and email.

## Context

The notification and alarm function allows you to configure alarm rules, set alarm contacts, and view alarm records.



#### Note:

Currently, EDAS only provides SMS and email notifications but does not support custom notifications.

## Procedure

**Configure alarm rules.**

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the target application.
3. On the **Application Details** page, choose **Notifications and Alarms > Alarm Rules** from the left-side navigation pane.
4. On the **Alarm Rules** page, click **Create Rule** in the upper-right corner.
5. On the **Create Rule** page, set rule parameters and click **OK**.

Table 1-22: Alarm rule parameters

Name	Description
Rule Name	Enter an easy-to-understand name for the rule, which can contain numbers, letters, or underscores (_).
Monitoring Target	Create comparable rules based on the monitoring metrics (infrastructure monitoring, HTTP, HSF, and application container) and set the threshold. Create at least one or multiple monitoring targets.
Trigger Conditions	<p>Select <b>Any One of the Indicators</b> or <b>All Indicators</b>.</p> <ul style="list-style-type: none"> <li>• <b>Any One of the Indicators:</b> An alarm is triggered when any of the metrics of the monitoring target satisfies the alarm rule.</li> <li>• <b>All Indicators:</b> An alarm is triggered when all of the metrics of the monitoring target satisfy the alarm rules.</li> </ul>
Statistical Cycle	Select 1 minute, 5 minutes, 15 minutes, 30 minutes, or 1 hour. A false alarm may be triggered when the system encounters transient jitter, for example, when CPU usage is high during service startup but returns to the normal range within two minutes. To prevent false alarms, you can select a statistical cycle to trigger alarms only when the alarm rules are continuously satisfied within this period. For example, if you select the 5-minute statistical cycle for the metric CPU usage above 30%, then EDAS determines that an exception occurs when the CPU usage of the system exceeds 30% for 5 consecutive minutes.

Name	Description
Retries Before Alarm	Set it to 1, 3, or 5. This parameter indicates the number of consecutive statistical cycles during which alarm rules are satisfied that are required to trigger an alarm.

Alarm rules take effect once created. To disable an alarm rule, select it on the Alarm Rules page and click Delete. The rule is disabled immediately.

#### Set alarm contacts



##### Notice:

- **Alarm contact source:** You can configure alarms to be sent to the contacts that have a primary and RAM user relationship with the current account.
- **Alarm contact information (email and mobile phone):** Add the contact information, including email addresses and mobile phone numbers, for alarm contacts.

6. Log on to the EDAS console. In the left-side navigation pane, choose Application Management. On the Applications page, click the name of the target application.
7. On the Application Details page, choose Notifications and Alarms > Alarm Contacts from the left-side navigation pane.
8. On the Alarm Contacts page, click Add Alarm Contacts in the upper-right corner.
9. Select the desired contacts from the contact list and click OK.

#### Set employees as alarm contacts



##### Note:

To add an employee that has never used EDAS as an alarm contact, follow these steps (assume that the current EDAS primary account is master@aliyun.com and the account to be added is employee@company.com):

- 10 Log on to EDAS as the newly created RAM user and modify information.
  - a) Log on to the EDAS console as the RAM user.
  - b) In the left-side navigation pane, choose Account Management > Personal Information and enter your mobile phone number and email address.
- 11 Add the master@company.com account as an alarm contact based on the procedure in Set alarm contacts.

## View alarm records



### Note:

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

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

13 On the **Application Details** page, choose **Notifications and Alarms > Alarm Records** from the left-side navigation pane.

Alarm records from the past 10 days are provided. After an alarm is cleared, a notification is generated and sent to contacts by SMS and email.

### 1.4.3.8 Auto scaling (only applicable to HSF applications in ECS clusters)

To ensure the service quality and availability of a distributed cluster, EDAS introduces crucial O&M capabilities that can detect the status of each instance in the cluster and can scale the cluster in or out in real time based on the system load.

#### Prerequisites

Auto scaling involves ECS instances. Ensure that your RAM user is authorized. For the authorization procedure, see *the Apsara Stack Console User Guide* and read the **RAM management** topic.

#### Context

EDAS provides the auto scaling function to automatically scale in or out a cluster based on the CPU, RT, and load of the ECS instances in the cluster. All these metrics are entered in positive integers without floating point numbers. If multiple ECS instances exist in the application, the average values of all ECS instances are used for all the preceding metrics. Auto scaling includes auto scale-in and scale-out, for which rules can be configured separately.

Metric	Description
CPU	The CPU usage of the ECS instance, expressed as a percentage . If multiple ECS instances exist in the application, the value of this metric is the average CPU usage of all ECS instances.
RT	The time for the system to respond to a request, in ms.
Load	The system load, which is a positive integer.

## Procedure

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose **Application Management**. On the Application Management page, click the name of the target application.
3. On the Application Details page, choose **Auto Scaling > Auto Scaling Rules** from the left-side navigation pane.
4. Select **Scale-Out Rule** and set scale-out rule parameters.

Table 1-23: Scale-out rule parameters

Name	Description
Trigger Indicators	Select CPU, RT, or Load.
Trigger Conditions	<ul style="list-style-type: none"><li>Any One of the Indicators: Auto scale-out is implemented when the threshold of any metric is exceeded.</li><li>All Indicators: Auto scale-out is implemented only when the thresholds of three metrics are exceeded.</li></ul>
Last for More Than	Auto scale-out is implemented when the threshold of a metric is exceeded for the specified period (in minutes). Set this parameter based on the sensitivity of the cluster service capability.
Number of Instances for Each Scale-Out	The number of ECS instances that are automatically added each time a scale-out is triggered. Set this parameter based on the single-instance service capability of the application.
Maximum Number of Instances	Scale-out stops when the number of ECS instances in the cluster reaches the maximum. Set this parameter based on your resource quota.



### Note:

When both the scale-in and scale-out rules are configured, the metric values of the scale-in rules cannot be greater than those of the scale-out rules. Otherwise, an error message appears when you click Save.

5. Select **Scale-In Rule** and set scale-in rule parameters.

Scale-in parameters are similar to scale-out parameters. For more information, see [Table 1-23: Scale-out rule parameters](#). Minimum Number of Instances indicates the minimum number of ECS instances reserved during scale-in to ensure service provisioning by the application.

## Result

After auto scaling rules are set, if auto scale-out or scale-in is implemented, use any of the following methods to view the auto scaling results:

- Choose Basic Information > Instance Information of the application to check whether the number of ECS instances is increased or reduced.
- Choose Auto Scaling > History Record to view the scale-out and scale-in history records.

### 1.4.3.9 Throttling and degradation (only applicable to HSF applications in ECS clusters)

Throttling and degradation are mainly used to solve slow system response or breakdown due to excessive burden on backend core services. These features are generally used in high-traffic scenarios, such as flash sales, shopping sprees, major promotions, and empty box scam protection.

#### Throttling

This function controls the traffic threshold or adjusts the traffic ratio. It controls traffic when front-end websites are dealing with heavy access traffic to prevent service unavailability that results from damage to backend core systems. By adjusting the traffic threshold, the throttling function controls the maximum traffic volume of the system to make sure secure and stable system operation.

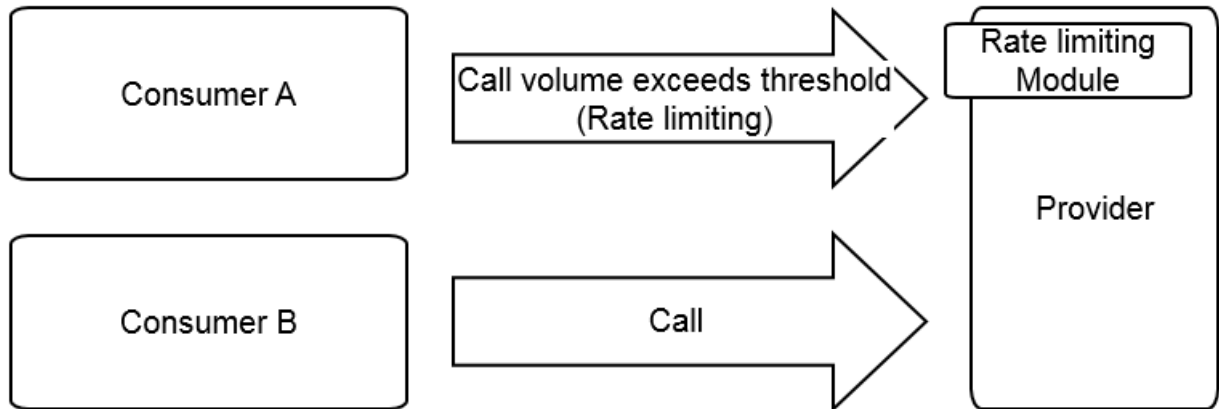
#### Principles

After the throttling code is configured for a provider and a throttling policy is configured in EDAS, the provider has the throttling function. When a consumer calls the provider, all access requests are calculated by the throttling module. If the



call volume of the consumer exceeds the preset threshold in a specific period, the throttling policy is triggered.

Figure 1-1: Throttling



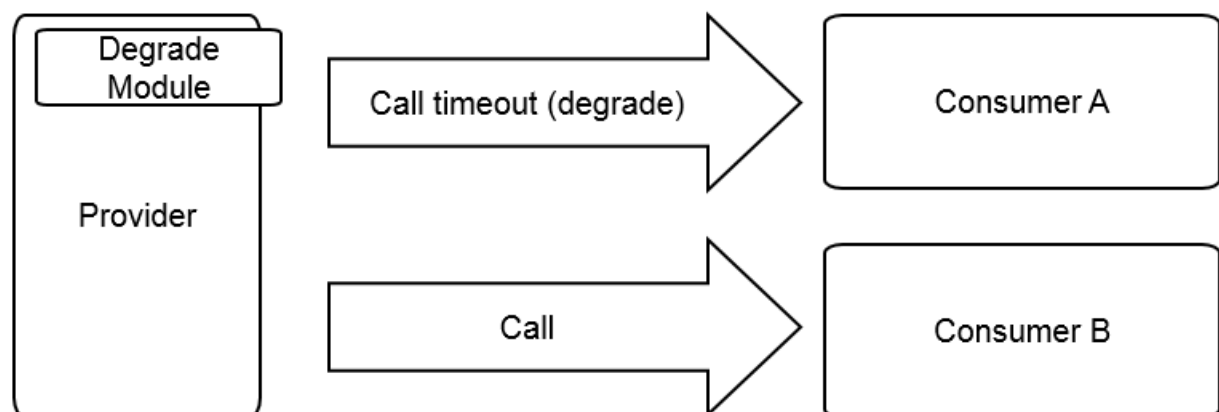
#### Degradation

In EDAS, degradation refers to the reduction of the call priority of downstream non-core providers that have timed out to make sure the availability of core consumers.

#### Principles

After degradation code is configured for a consumer and a degradation policy is configured in EDAS, the consumer has the degradation function. When the consumer calls a provider, if the response time of the provider exceeds the preset threshold, the degradation policy is triggered, as shown in the following figure.

Figure 1-2: Degradation



#### 1.4.3.9.1 Throttling management

One application provides multiple services. EDAS allows you to configure throttling rules for the services, ensuring service stability and rejecting traffic that exceeds

the service capabilities. EDAS allows you to configure throttling rules based on the QPS and threads to ensure the optimal operation stability of application systems during traffic peaks.

## Context

- **HSF rate limiting:** When the traffic during a traffic spike exceeds the upper threshold defined by the throttling rules, the `BlockException` error occurs for some consumers. Based on the set threshold, the same number of services as the set threshold are successfully called within 1s.
- **HTTP rate limiting:** When a traffic spike occurs, some consumers are redirected to an error page. During actual access, the Taobao homepage appears. Based on the set threshold, some requests can be successfully sent to the services.



### Notice:

Throttling rules apply only to providers but cannot be configured for consumers. Before configuration, make sure that the application serves as the provider.

## Procedure

### 1. Write the throttling rule code.

a) *Log on to the EDAS console.*

b) In the left-side navigation pane, choose Application Management. On the Applications page, click a deployed provider application.

c) On the Application Details page, choose Service Degradation > Rate Limiting Rules from the left-side navigation pane.

d) On the Rate Limiting Rules page, click Application Configuration Guide in the upper-right corner. Write throttling code based on the example.

### 2. Add the throttling rule code to the application and then compile the code and

*Publish an application.*

### 3. Return to the EDAS console. In the left-side navigation pane, choose Service Degradation > Rate Limiting Rules. On the Rate Limiting Rules page, click Add Rate Limiting Rules in the upper-right corner.

4. On the Add Rate Limiting Rules page, set the throttling rule parameters and then click OK.

Table 1-24: Throttling rule parameters

Name	Description
Rate Limiting Type	Select HSF Rate Limiting or HTTP Rate Limiting based on the access type of the application.
Interface	Select the interface to which the throttling rule applies from the listed interfaces as needed.
Method	Select a specific method or all methods to which the throttling rule applies after all methods of the selected interface are automatically loaded.
Application	Select the application to which the throttling rule applies from the application list as needed. The application list includes all applications that may access the current application, excluding the current application itself.
Rate Limiting Granularity	<p>Select QPS or Thread.</p> <ul style="list-style-type: none"> <li>· QPS indicates limiting the number of requests per second.</li> <li>· Thread indicates limiting the number of threads.</li> </ul> <p>The QPS value is typically proportional to the number of threads. However, the QPS of a thread is generally greater than 1 because a thread keeps sending requests and the response time is dozens of milliseconds.</p>
Rate Limiting Threshold	Throttling is triggered when the set threshold is exceeded.

## What's next

On the Rate Limiting Rules page, locate the row that contains the target rule, and click Edit, Stop, Enable, or Delete on the right.

### 1.4.3.9.2 Degradation management

Each application calls multiple external services. Service degradation can be configured to pinpoint and block poor services. This feature ensures the stable operation of your application and prevents the functionality of your application from being compromised by dependency on poor services.

## Context

EDAS allows you to configure degradation rules based on the response time, preventing your application from depending on poor services during traffic peaks. The consumer who triggers a degradation rule will not initiate an actual remote call within the specified time window and returns the `DegradeException` error. After the time window ends, the original remote service call is restored.



**Note:**

The degradation rules apply only to consumers and cannot be configured for providers. Before configuration, make sure that the application serves as a consumer.

## Procedure

1. Write the degradation rule code.
  - a) *Log on to the EDAS console.*
  - b) In the left-side navigation pane, choose Application Management. On the Applications page, select a deployed provider application.
  - c) On the Application Details page, choose Service Degradation > Degradation Rules from the left-side navigation pane. Click Application Configuration Guide in the upper-right corner. Write degradation rule code based on the example.
2. Add the degradation rule code to the application and then compile the code and *Publish an application.*
3. Return to the EDAS console. In the left-side navigation pane, choose Service Degradation > Degradation Rules. On the Degradation Rules page, click Add Degradation Rules in the upper-right corner.
4. On the Add Degradation Rules page, set degradation rule parameters and click OK.

Table 1-25: Degradation rule parameters

Name	Description
Degradation Type	Select HSF Degradation and HTTP Degradation as needed.
Interface	All interfaces that the consumer is consuming are listed. Select the interface to be degraded as needed.

Name	Description
Method	All methods are automatically loaded based on the selected interface. You can select whether to degrade all methods or a specific method as needed.
RT Threshold	The threshold of the service response time that triggers degradation, in ms. If this threshold is exceeded, the selected interface or method is degraded.
Time Window	The rule execution duration after degradation is triggered.

### What's next

On the Degradation Rules page, locate the row that contains the target rule, and click Edit, Stop, Enable, or Delete on the right.

#### 1.4.3.10 Application diagnosis (only applicable to HSF applications in ECS clusters)

Applications are deployed and run in Tomcat containers. EDAS provides application diagnosis for container monitoring. You can locate memory, class conflict, and other application runtime problems based on data.

EDAS provides a refined statistical function designed for application containers, which collects statistics on the application-running instance based on a range of statistical items, including JVM heap memory, non-heap memory, class loader, thread, and Tomcat connector. Similar to infrastructure monitoring, container monitoring (application diagnosis) lists application-specific data by instance.

The differences are as follows:

- The monitoring target of infrastructure monitoring is ECS instances, whereas that of container monitoring is application containers.
- Application diagnosis supports query of diagnostic information in single-instance mode rather than cluster mode.
- Infrastructure monitoring has latency, whereas container monitoring is near real-time because statistical computing is not performed on collected data (except memory monitoring data).

### 1.4.3.10.1 Common operations

This topic describes how to locate the ECS instance for an application and view the application diagnosis information.

#### Context

#### Procedure

1. *Log on to the EDAS console.*
2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the target application.
3. On the **Application Details** page, choose **Application Diagnosis** from the left-side navigation pane.
4. Select an ECS instance from the **ECS Instance (Instance ID/Name/IP)** drop-down list.

5. Click tabs to view the monitoring metrics of the container.

The Application Diagnosis page contains the following tabs:

Tab	Description
GC Diagnosis	<p>This tab is divided into GC Diagnosis and Memory.</p> <ul style="list-style-type: none"> <li>· <b>GC Diagnosis:</b> This function can monitor some performance metrics of the current instance when GC occurs according to the instances of the current application and analyze the GC of the current instance according to the selected time range. These metrics help you to judge the health status of an instance of the application, that is, check whether the application has memory leakage or large objects. <ul style="list-style-type: none"> <li>- The GC policy of the current instance is -XX:+UseParallelGC or -XX:+UseParallelOldGC</li> <li>- FGC is short for Full Garbage Collection. YGC is short for Young Garbage Collection.</li> <li>- When the total number of YGC times exceeds 6 or that of FGC times exceeds 10 within 1 minute and the number of YGC or FGC times within this 1 minute is the greatest in the selected time range, this 1 minute is called the most active YGC or FGC time.</li> <li>- When the total time consumed by all YGC events within 1 minute is greater than 100 ms or that by all FGC events within 1 minute is greater than 300 ms and the total time consumed by YGC or FGC within this 1 minute is the longest in the selected time range, this 1 minute is called the most time-consuming YGC or FGC period.</li> <li>- The memory difference before and after GC refers to the memory occupied by the application.</li> </ul> </li> <li>· <b>Memory:</b> EDAS provides heap and non-heap statistics for the JVM process where the Tomcat container of the application is located.</li> </ul>

Tab	Description
Class loading	EDAS provides real-time information about JAR package loading. When a JAR package of the application has a version conflict, you can use this function to easily monitor the path to which the JAR package is loaded, which lowers troubleshooting costs.
Connector	For more information, see <a href="#">Connector</a> .
Object memory distribution	Select System Classes, Java Primitive Object Classes, and Class Loading Related Classes. Based on the selected three classes, the number of objects, occupied space, and usage percentage of the total system memory are displayed in a pie chart and a list.
Method tracing	Method tracing is complex. For more information, see <a href="#">Method tracing</a> .
Hot thread	Two functions are provided: get thread snapshots and analyze call statistics. For more information, see <a href="#">Hot thread</a> .
Druid database connection pool monitoring	For more information, see <a href="#">Druid database connection pool monitoring</a> .
Commons Pool monitoring	For more information, see <a href="#">Commons Pool monitoring</a> .

### 1.4.3.10.2 Connector

A Tomcat connector is the `<Connector />` in the XML configuration of Tomcat. Each `<Connector />` configuration item can be considered as the information pulled from a line. This view displays the runtime status of the corresponding connector over the past 10 minutes.

#### Context

Each connector has a certain number of threads (which form a thread pool) to process incoming requests. When concurrency or throughput bottlenecks occur



, statistics must be collected on the processing status of the thread pool of the connector. For example, an HTTP connector has the following XML configuration:

```
<Connector port="8080" protocol="HTTP/1.1" maxThreads="250" .... />
```

## Procedure

1. Click Thread Pool Information in the Actions column to the right of Connector Statistics to view details, as shown in the following figure.

The preceding figure shows that the connector of the application is almost load-free. If the value of Busy Thread Count is close to that of Maximum Thread Pool Size, the system encounters serious concurrency problems. To solve the problem, scale up the application or optimize the service code.

2. Click Details in the Actions column to view the connector details.

### 1.4.3.10.3 Method tracing

EDAS method tracing helps you quickly troubleshoot application runtime problems in the following typical scenarios:

## Context

- It takes a long time to execute a service logic during the application runtime. In this case, you may want to identify the time consumed by each part of the code during the runtime to determine the most time-consuming part.
- Applications run properly and services are normal most of the time. However, the service response may be extremely slow when a specific parameter is entered. In this case, you may want to check the code execution related to a specific input parameter of the method.
- When a method with complex service logic is executed, you cannot determine the called logic and the call sequence. In this case, you may want to know the logic and time sequence of execution in detail.

EDAS method tracing is designed to meet the preceding requirements without interfering with code or stopping applications.

EDAS method tracing adopts the JVM bytecode enhancement technique to record the elapsed time and sequence during the entire call process of the selected method. This allows you to check the execution sequence while execution is in progress.

## Limits

If the following limits affect your services or troubleshooting, open a ticket so that we can improve some limits or configure a whitelist.

1. In principle, only tracing of service-type classes is supported. Therefore, packages are filtered by name before tracing starts.
2. Sampled output is adopted to prevent excessive logs due to frequent method calling. The default policy is to output logs for 10 calls per second.
3. When you exit and then log on to the EDAS console again or refresh the page, historical trace records are lost and previously pulled tracing information is no longer retained.
4. Automatic stop policy: If method tracing is in the inactive state for 10 minutes, EDAS automatically detaches the tracing module and restores the method to the initial state (state prior to enhancement).
5. Parameter printing: Currently, EDAS only supports printing for the data of basic Java types (string, char, int, float, double, short, and boolean).
6. If the selected string contains excessive characters, EDAS truncates the string to output the first 100 characters.
7. If the JVM restarts during the tracing process, you must disable the tracing function and then enable it again.
8. Currently, a maximum of 10,000 trace logs can be output. To output more logs, restart the tracing function.
9. The current version does not support Docker applications.

## Environment check

JVM bytecode enhancement is implemented during method tracing. The environment check tool is unavailable when some check items are not passed. This ensures the normal operation of applications.

Before the method tracing function starts, EDAS automatically checks that:

1. Ali-Tomcat is in the Running state.
2. The CPU usage is lower than 60%.
3. The idle system memory is larger than 100 MB.
4. The available size of the JVM permanent generation or metaspace is more than 20 MB.

If the environment check fails, we recommend that you clear the alarms and click **Retry**.

#### Procedure

1. *Log on to the EDAS console.*
2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the target application.
3. On the **Application Details** page, choose **Application Diagnosis** from the left-side navigation pane.
4. On the **Application Diagnosis** page, click the **Method Tracing** tab.



#### Note:

If the **Method Tracing** tab page does not appear, follow these steps:

- a. Make sure you are using Google Chrome and refresh the page. (We have only tested functionality in Google Chrome.)
- b. If you log on as a RAM user, check that the RAM user is assigned the required permissions.

Portal for permission check: **Application Management > Application Diagnosis > Method Tracing and Tool Authorization**.

5. Before the permission check starts, EDAS performs an environment check on the ECS instance where the application is located. When the environment check dialog box appears, select **I confirm that the above conditions are met, and I agree to start using this function**. Click **Confirm Use** to start the environment check.

The **Method Tracing** page appears after the environment check is successful.

6. Set tracing parameters.



#### Note:

**Class Name** and **Method Name** are required. Set the parameters to the class and method to be traced.

The configuration items are described as follows.

<p><b>Class Name</b></p>	<p>(Required) Set this parameter to a full path name that starts with the package path, such as <code>com.test.alibaba.demo.HelloWorldServlet</code>. EDAS does not support the tracing of classes whose names start with the following package paths:</p> <ul style="list-style-type: none"> <li>• <code>java.*</code></li> <li>• <code>javax.*</code></li> <li>• <code>com.google.*</code></li> <li>• <code>com.alibaba.*</code></li> <li>• <code>com.aliyun.*</code></li> <li>• <code>com.taobao.*</code></li> <li>• <code>org.apache.*</code></li> <li>• <code>org.dom4j.*</code></li> <li>• <code>org.springframework.*</code></li> <li>• <code>redis.clients.*</code></li> </ul> <p>After a complete package path is entered, EDAS checks whether the class exists on the selected ECS instance.</p> <ul style="list-style-type: none"> <li>• If the entered class exists, it appears in the drop-down list. Select the class to continue.</li> <li>• If the entered class does not exist, the message "This class does not exist" appears in the drop-down list.</li> </ul>
--------------------------	--

<p><b>Method Name</b></p>	<p>(Required) After a class is selected, the system automatically searches for all the methods under the class and shows the method list under this field, as shown in the following figure.</p> <p>The icon on the left of each method indicates the modifier of the method.</p> <ul style="list-style-type: none"> <li>• public: a green lock</li> <li>• protected: a yellow key</li> <li>• private: a red lock</li> <li>• package: a blue block</li> <li>• abstract: no icon</li> </ul> <p>Select the method to be traced from the drop-down list and continue.</p> <ul style="list-style-type: none"> <li>• <b>Exception Tracing Only:</b> The execution of a method has either of two results: return a response normally or end execution due to an exception. If you select Exception Tracing Only, the tracing results are printed and output only when the method is ended due to an exception.</li> <li>• <b>Print Returned Values:</b> If you select this option, the returned value of the method is printed on the result page. If the return type of the method is <code>void</code>, <code>null</code> is output.</li> </ul>
---------------------------	---

When you select a method, the Start Tracing button becomes available and turns blue.

7. Click Start Tracing to trace the method. Whenever the method is called, the call information appears in the result section.



**Notice:**

After method tracing starts, EDAS periodically checks whether the tracing is active. If the tracing is inactive within 10 minutes, EDAS automatically stops method tracing and restores the traced method to its original status.

8. Check the method call information.

After method tracing starts, EDAS displays the generated call logs in the EDAS console.

On the left of the display section, each record shows the log that is generated each time the method is called.

- On the left of the table, 44-62/150 appears, indicating that the web browser retrieves a total of 150 records and that the tracing records with row numbers from 44 to 62 are currently displayed.
- At the bottom of the table, the message "Press H for help information" appears. Press H on the keyboard to display the usage instructions for shortcut keys.
  - H: displays the help document.
  - Ctrl+G: displays the latest retrieved data in real time. As more and more calls are made, it is impossible to render and display all records. Ctrl+ G is similar to the tail function. The latest retrieved data appears upon input of new data.
  - G: jumps to a specific record. Search for the trace record in a specific row and select it to display details.
  - Ctrl+C or Esc: ends the command.
  - Ctrl+H: pages down to display the following 10 trace records.
  - Ctrl+I: pages up to display the preceding 10 trace records.
  - J or ↓: selects the next trace record.
  - K or ↑: selects the previous trace record.
  - Enter or Double-click: zooms in on or restores the selected trace record.

On the right of the display section, the details of the selected record appear. You can zoom in on the details page by pressing Enter or double-clicking. Press Esc

to restore the selected trace record. Only some basic information appears on the details page.

- **Tracing Details:** shows the elapsed time and execution sequence of each call of the selected method. The elapsed time in **blue** indicates the total elapsed time by method execution. The elapsed time in **red** indicates that the specific call consumes time more than 30% of the total elapsed time.
- **Output Details:** shows the exceptions, returned values, and input parameters (which are selected with the More option) during output.
- **Method Stack Details:** shows the stack information before the traced method is called.

#### 9. View the source code of the traced method.

- Click **Decompile** to obtain the decompiled source code of the current tracing method (the corresponding class).
- Click *Method Name* (such as `doPost`

You can select or deselect **Show Decompiled Source Code** to show or hide the source code window.

#### 10 Stop tracing.

After method tracing starts, the **Start Tracing** button changes to **Stop Tracing**. . After you click **Stop Tracing**, EDAS restores the traced method to its original status (before enhancement) and records tracing information by instance. The last tracing information is automatically entered on this page when it is opened next time.

If you modify tracing items (such as the method name) when tracing stops and then click **Start Tracing**, the changes that you made are submitted and the tracing starts with these changes applied.

### 1.4.3.10.4 Commons Pool monitoring

When Commons Pool2 (v2.0), such as the Jedis and Commons DBCP2 connection pools on a Redis client, is used by an application or application class library, the EDAS Commons Pool monitoring component monitors the configuration and usage of these pools. The monitoring data is recorded every 10 seconds.

#### Procedure

1. [Log on to the EDAS console](#).

2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the target application.
3. On the **Application Details** page, choose **Application Diagnosis** from the left-side navigation pane.
4. On the **Application Diagnosis** page, select an instance from the **ECS Instance (Instance ID/Name/IP)** drop-down list.
5. Click the **Commons Pool** tab.
6. Click **Start Monitoring**.

After monitoring starts, EDAS automatically traces the existing or newly created pools. The usage and configuration of each pool object appear on the page. The page is refreshed every 10s by default. The tracing automatically stops when the pools are closed.

**Note:**

If the application has no Commons Pool2 class library or no pool has been loaded, the **Start Monitoring** button is unavailable.

7. Click **Stop Monitoring** to stop monitoring.

**Result**

The pool monitoring information includes two parts:

- **Pool usage:** The information is presented in a line graph. This graph shows the numbers of active, idle, and blocked threads and the maximum numbers of objects and idle objects.
- **Pool configuration:** The information is presented in a table. This table includes the configuration items of the pool. For more information, see [GenericObjectPoolCon fig.](#)

**Commons Pool usage fields:**

Name	Field	Description
Maximum number of objects	Max Total	The maximum number of objects in a pool, including the active and idle objects.



Name	Field	Description
Maximum number of idle objects	Max Idle	The maximum number of available objects in the pool.
Active objects	Num Active	The number of active objects in the pool. If this number often exceeds the maximum number of idle objects, you need to increase the latter.
Idle objects	Num Idle	The number of available objects in the pool.
Blocked threads	Num Waiters	The number of blocked threads. If it is greater than 0, increase the maximum number.

#### 1.4.3.10.5 Druid database connection pool monitoring

When the data connection pool of an application uses a Druid database, the EDAS Druid database connection pool monitoring component monitors the data connection pool and SQL execution. The monitoring data is refreshed every 10s.

##### Context

To use the Druid database connection pool, complete the following steps:

##### Procedure

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose **Application Management**. On the **Applications** page, click the name of the target application.
3. On the **Application Details** page, choose **Application Diagnosis** from the left-side navigation pane.
4. On the **Application Diagnosis** page, select an instance from the **ECS Instance (Instance ID/Name/IP)** drop-down list.
5. Click the **Druid Database Connection Pool Monitor** tab.

## 6. Click Start Monitoring.

The page shows information about the database connection pool and SQL execution. The page is refreshed every 10s by default.



### Note:

When you click Start Monitoring, if StatFilter provided by the Druid database connection pool is not configured for the application, EDAS automatically adds StatFilter to the application. Given that this may slightly affect the performance, we strongly recommend that you manually add StatFilter to your application.

## 7. Click Close to exit monitoring.

### Result

The monitoring information about the database connection pool includes the following:

- Database connection pool monitoring metrics include the database type, driver class, initial connection pool size, and maximum connections.
- The runtime information about the database connection pool includes the size of available connections, peak size of available connections, and number of active connections.

The following table lists the Druid database connection pool monitoring metrics.

Name	Field	Description
Database type	DB Type	The database type of a data source connection, such as MySQL.
Driver class	Driver Class	The class name of a data driver.
Username	User Name	The user who connects to the database.
Initial connection pool size	Init Size	The initial size of the database connection pool.
Maximum connection count of the connection pool	Max Active	The maximum number of connections in the connection pool.

Name	Field	Description
Connection count of the connection pool	Pool Size	The number of available connections in the database connection pool.
Peak connection count of the connection pool	Maximum Pool Size	The maximum number of available connections in the database connection pool.
Active connection count	Active Count	The number of active connections in the data connection pool.

The SQL execution information consists of the information about SQL statements executed over the last 10 seconds and the information about SQL statements whose maximum execution time exceeds 100 milliseconds. The monitoring metrics for the two types of information are the same, as described in the following table.

Name	Field	Description
SQL	SQL	The SQL statement that is executed.
Number of executions	Executed Count	The number of SQL executions.
Total time consumed by execution	Total Executed Time	The total time consumed by SQL execution.
Maximum time consumed by execution	Maximum Executed Time	The maximum time consumed by SQL execution.
Maximum number of returned rows	Maximum Returned Rows	The maximum number of rows that are returned after SQL execution.
Record time	Monitor Time	The time when SQL information is recorded.

### 1.4.3.10.6 Hot thread

On the Hot Thread page, the Get Thread Snapshots and Analyze Call Statistics functions are available.

Retrieve thread snapshots

Similar to the `jstack` command, the hot thread retrieves the stack frames of all current threads. After retrieving the thread stacks, it filters out the identified idle threads, such as HSF, Tomcat, and GC threads. To avoid a high overhead, the system returns the data of only 30 threads by default.

Analyze call statistics

The hot thread collects statistics on and analyzes the method calls in applications over a time period, and presents the method calls and call relationship (call stack). The final results are displayed in a tree graph and a flame graph, with your service methods highlighted, allowing you to quickly locate the call sources of the most time-consuming service methods. Results are returned about 10 seconds after the call is initiated.

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose Application Management. On the Application Management page, click the name of the application to be diagnosed.
3. On the Application Details page, choose Application Diagnosis from the left-side navigation pane.
4. On the Application Diagnosis page, click Hot Thread.

**5. On the Hot Thread page, click Get Thread Snapshots or Start Call Statistics Analysis.**

- **Retrieve thread snapshots**

After you click Get Thread Snapshots, the system attempts to group and identify the resulting threads, locates service threads and service stack frames, and expands them.

- **Start call statistics analysis**

After you click Start Call Statistics Analysis, the system performs call analysis. The call analysis results can be displayed in two formats:

- **Tree graph (default)**

The system attempts to locate and expand the service logic stack frames.

- **Flame graph**

The call statistics and call relationships are displayed in a flame graph. By clicking a stack frame (method), you can view the call statistics for the call path.

### **1.4.3.11 Container version management (only applicable to HSF applications in ECS clusters)**

EDAS allows you to view container versions and historical publishing details and perform upgrade and downgrade.

#### **Context**

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

You must select a container version when creating an application in EDAS. EDAS containers are maintained and published by the EDAS development team. Choose **Application Management > Container Version** to view the container publishing

history and the description of each publishing operation. Generally, a container of a later version is superior to a container of an earlier version in terms of stability and function variety.

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

#### **Procedure**

1. In the left-side navigation pane, choose **Application Management** to go to the **Applications** page.
2. Click the name of the target application to go to the **Application Details** page.
3. In the left-side navigation pane, choose **Container Version** to go to the **Container Version** page.
4. Locate the row that contains the target container version and click **Upgrade to This Version** or **Downgrade to This Version** on the right to upgrade or downgrade the container in one click.

### **1.4.4 Microservice management**

Microservice management is an important function of EDAS. It allows you to view services in applications and inter-service traces.

Microservice management provides the following main functions:

- **Trace query**

By setting filter criteria, you can accurately locate services with poor performance or exceptions.

- **Trace details**

Based on the trace query results, you can view details of slow or abnormal services and reorganize their dependencies. This information allows you to identify frequent failures, performance bottlenecks, strong dependencies, and other problems. You can also evaluate service capacities based on trace call ratios and peak QPS.

- **Service topology**

The service topology intuitively presents the call between services and relevant performance data.

### 1.4.4.1 Query traces

The trace query function is used to view the trace status in the system, especially for slow or abnormal services.

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose **Microservice Management > Trace Query**.  
.
3. On the Trace Query page, click **Show Advanced Options** in the upper-right corner to display filter criteria.
4. Set the filter criteria and click **Search**.

#### Trace parameters:

- **Time Range:** Click the time selector, set the query start time, and select a time period. The time period options are **This Second**, **To 1 Minute Later**, and **To 10 Minutes Later**. These options correspond to the following maximum delays: current time (last second), 1 minute ago, and 10 minutes ago.
  - **Application Name:** Select an application from the drop-down list. You can enter a keyword to search for an application. Manual input of an application name is not supported.
  - **Call Type:** Select a call type from the drop-down list. The options are **HTTP**, **HSF Provider**, **HSF Consumer**, **MySQL**, **Redis Cache**, **Message Sending**, and **Message Receiving**.
  - Set the threshold of elapsed time, request, or response for querying slow services in the system.
  - In the query section, select **Error** in the upper-right corner to query abnormal services in the system.
  - Set the other parameters as needed.
5. In the query results, click a slow or abnormal service to view trace details.

For the procedure of viewing trace details, see [Trace details](#).

### 1.4.4.2 Trace details

On the Trace Details page, you can query the details about a trace based on the TraceId in the selected region.

#### Prerequisites

The Trace Details page shows traces for which remote methods are called. It does not display local methods that are called.

Trace details are used to locate the elapsed time and exceptions in each step during a distributed call. Local calls are not the focus of traces. We recommend that you view service logs to check the elapsed time and exceptions for local calls. For example, the Trace Details page does not display the process where the local logic methodA() calls localMethodB() and localMethodC(). Therefore, sometimes the elapsed time on a parent node is greater than the total elapsed time on all subnodes .

You can search trace details on the Trace Details page. A more typical scenario is checking the slow or abnormal services in trace query results. The following uses an example to describe how to view details of a trace through trace query.

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose **Microservice Management > Trace Details**.
3. On the Trace Details page, view trace details.
4. On the Trace Query Result page, locate the most time-consuming HSF method, database request, or other remote calls.
  - For database, Redis, MQ, or other simple calls, identify the cause of slow access to these nodes and check whether slow SQL or network congestion occurs.
  - For an HSF method, further analyze the reason why the method consumes so much time.
5. Confirm the elapsed time on a local method. Place the pointer over the timeline of the method. A pop-up window appears, showing the time it takes the



consumer to send the request, the time it takes the provider to process the request, and the time it takes the consumer to receive the response.

- If the time it takes the provider to process the request is long, analyze the service.
- Otherwise, analyze the cause by using the method for analyzing call timeout.

6. Check whether the total elapsed time on subnodes is close to the elapsed time on this method.

- If the time difference is small, most of the time is consumed by network calls. In this case, reduce network calls as much as possible to shorten the elapsed time on each method. The FOR statement cyclically calls the same method. The methods should be called in one batch to retrieve the response whenever possible.
- If the time difference is large (for example, the elapsed time on the parent node is 607 ms while the total elapsed time on the subnodes does not reach 100 ms), the time is consumed on the service logic of the provider, rather than the request of the remote call.

7. Locate the time-consuming call. Inspect time-consuming calls by viewing the timelines of nodes to first locate the call initiated before the excessive time consumption. This is the local logic, for which further troubleshooting is required.

- a. After locating the time-consuming logic, review the code or add a log method to the code to locate the specific error.

If the code does not consume so much time, perform the following step:

- b. Check whether GC occurred at that time. Therefore, the gc.log file is important.

8. Locate the timeout error. A timeout error occurred. Perform the following steps to evaluate the time.

The time is divided into three parts:

- Consumer sends request (0 ms): indicates the elapsed time from when the consumer sends a request to its receipt by the provider, including the time for serialization, network transfer, and deserialization. If this process takes a long time, check whether consumer GC is triggered. A lot of time is consumed if

the serialization or deserialization object is large, the network is under a high transmission load, or provider GC occurs.

- **Provider processes request (10,077 ms):** indicates the elapsed time from the receipt of the request by the provider to its response to the consumer. During this period, the provider processes the request, and the time consumed by other operations are not included.
- **Consumer receives the response (3,002 ms):** indicates the elapsed time from when the provider sends the response to the receipt of the response by the consumer. With the 3-second timeout period, the provider directly returns a timeout error if the operation times out, but the provider continues processing the request. If this process consumes a lot of time, perform troubleshooting by using the same method as that for the consumer sending the request.

#### 1.4.4.3 Service statistics

Service statistics show the runtime status of all the services of all the applications under the current tenant from the past 24 hours, including the service call volume, call time consumption, and number of call errors. These statistics allow you to easily compare all services in the system.

##### Procedure

1. Log on to the EDAS console.
2. In the left-side navigation pane, choose **Microservice Management > Service Statistics**.
3. On the Service Statistics page, select a region in the upper part to view the runtime status of services.

#### 1.4.4.4 Service topology

The service topology feature is used to view the real-time (last second) call topology between services in the system.

##### Context

##### Procedure

1. *Log on to the EDAS console.*
2. In the left-side navigation pane, choose **Microservice Management > Service Topology**.

### 3. View the service topology.

The service topology shows the real-time (last second) call topology between all services under the current account.

- Place the pointer over a service to view the call topology for this service.
- Click a service to view its call topology and traffic data.

Traffic data refers to the QPS of the current service, including:

- Incoming Traffic: indicates the QPS for calls from other services to this service.
- Outgoing Traffic: indicates the QPS for calls from this service to other services.

#### 1.4.4.5 Redis support

After Redis support is added, whenever applications access and perform operations on Redis, the process is recorded in EagleEye trace logs and EDAS collects, analyzes, and computes statistics on the logs. Then, information about Redis calls appears on the distributed tracing and call analysis page of the EDAS console.

##### Support range

Currently, only [Spring Data Redis 1.7.4.RELEASE](#) is compatible, in consideration of the large number of Redis databases and the usability of Spring Data repositories. If your project uses a database (such as Jedis) other than Spring Data Redis, then you cannot view related information in the EagleEye link (in the left-side navigation pane of the EDAS console, choose Operational Data > Trace Details).



##### Notice:

If your applications use a Spring Data Redis version later than 1.7.4.RELEASE and the provided functions are not supported in this version, open a ticket to seek help from Alibaba Cloud Customer Services.

##### Procedure

For applications in the EDAS console, use Spring Data Redis for replacement, which is used in the same way as Spring Data Redis. For instructions on using Spring Data Redis, see [its official documentation](#). For code development, EDAS is compatible with Spring Data Redis 1.7.4 RELEASE. To enable Redis support, complete these steps:

**1. Open the {user.home}/.m2/settings.xml file and configure the Maven local repository.**

```
<profile>
  <id>edas.oss.repo</id>
  <repositories>
    <repository>
      <id>edas-oss-central</id>
      <name>taobao mirror central</name>
      <url>http://edas-public.oss-cn-hangzhou.aliyuncs.
com/repository</url>
      <snapshots>
        <enabled>true</enabled>
      </snapshots>
      <releases>
        <enabled>true</enabled>
      </releases>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <id>edas-oss-plugin-central</id>
      <url>http://edas-public.oss-cn-hangzhou.aliyuncs.
com/repository</url>
      <snapshots>
        <enabled>true</enabled>
      </snapshots>
      <releases>
        <enabled>true</enabled>
      </releases>
    </pluginRepository>
  </pluginRepositories>
</profile>
</profiles>
```

**Activate the corresponding profile as follows:**

```
<activeProfiles>
  <activeProfile>edas.oss.repo</activeProfile>
</activeProfiles>
```

**2. Add the following dependency to the pom.xml file of Maven in the project.**

```
<dependency>
  <groupId>com.alibaba.middleware</groupId>
  <artifactId>spring-data-redis</artifactId>
  <version>1.7.4.RELEASE</version>
</dependency>
```

Redis command override

**The following describes the Redis command override by Spring Data Redis and the support for EagleEye trace logs.**

Table 1-26: Key-type operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
Key	DEL	RedisOperations.delete	Y	
	DUMP	RedisOperations.dump	Y	
	EXISTS	RedisOperations.hasKey	Y	
	EXPIRE	RedisOperations.expire	Y	
	EXPIREAT	RedisOperations.expireAt	Y	
	KEYS	RedisOperations.keys	Y	
	MIGRATE	Not supported		
	MOVE	RedisOperations.move	Y	
	OBJECT	Not supported		
	PERSIST	RedisOperations.persist	Y	
	PEXPIRE	RedisOperations.expire	Y	
	PEXPIREAT	RedisOperations.expireAt	Y	
	PTTL	RedisOperations.getExpire	Y	
	RANDOMKEY	RedisOperations.randomKey	Y	
	RENAME	RedisOperations.rename	Y	key: oldKey : \${oldKey} };newKey:\${newKey}

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	RENAMENX	RedisOperations.renameIfAbsent	Y	
	RESTORE	RedisOperations.restore	Y	
	SORT	RedisKeyCommands.sort	Y	key: query:\${SortQuery}
	TTL	RedisOperations.getExpire	Y	
	TYPE	RedisOperations.type	Y	
	SCAN	RedisKeyCommands.scan	N	

Table 1-27: String-type operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
String	APPEND	ValueOperations.append	Y	
	BITCOUNT	Not supported		
	BITOP	Not supported		
	BITFIELD	Not supported		
	DECR	ValueOperations.increment	Y	
	DECRBY	ValueOperations.increment	Y	
	GET	ValueOperations.get	Y	

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	GETBIT	ValueOperations.getBit	Y	
	GETRANGE	ValueOperations.get	Y	
	GETSET	ValueOperations.getAndSet	Y	
	INCR	ValueOperations.increment	Y	
	INCRBY	ValueOperations.increment	Y	
	INCRBYFLOAT	ValueOperations.increment	Y	
	MGET	ValueOperations.multiGet	Y	
	MSET	ValueOperations.multiSet	Y	
	MSETNX	ValueOperations.multiSetIfAbsent	Y	
	PSETEX	ValueOperations.set	Y	
	SET	ValueOperations.set	Y	
	SETBIT	ValueOperations.setBit	Y	
	SETEX	ValueOperations.set	Y	

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	SETNX	ValueOperations.setIfAbsent	Y	
	SETRANGE	ValueOperations.set	Y	
	STRLEN	ValueOperations.size	Y	

Table 1-28: Hash-type operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
Hash	HDEL	HashOperations.delete	Y	
	HEXISTS	HashOperations.hasKey	Y	
	HGET	HashOperations.get	Y	
	HGETALL	HashOperations.entries	Y	
	HINCRBY	HashOperations.increment	Y	
	HINCRBYFLOAT	HashOperations.increment	Y	
	HKEYS	HashOperations.keys	Y	
	HLEN	HashOperations.size	Y	
	HMGET	HashOperations.multiGet	Y	
	HMSET	HashOperations.putAll	Y	



Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	HSET	HashOperations.put	Y	
	HSETNX	HashOperations.putIfAbsent	Y	
	HVALS	HashOperations.values	Y	
	HSCAN	HashOperations.scan	Y	
	HSTRLEN	Not supported		

Table 1-29: List-type operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
List	BLPOP	ListOperations.leftPop	Y	
	BRPOP	ListOperations.rightPop	Y	
	BRPOPLPUSH	ListOperations.rightPopAndLeftPush	Y	key: sourceKey : \${sourceKey} ; destKey: \${destKey}
	LINDEX	ListOperations.index	Y	
	LINSERT	ListOperations.leftPush	Y	
	LLEN	ListOperations.size	Y	
	LPOP	ListOperations.leftPop	Y	
	LPUSH	ListOperations.leftPush	Y	

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	LPUSHX	ListOperations. .leftPushIf Present	Y	
	LRANGE	ListOperations. .range	Y	
	LREM	ListOperations. .remove	Y	
	LSET	ListOperations. .set	Y	
	LTRIM	ListOperations. .trim	Y	
	RPOP	ListOperations. .rightPop	Y	
	RPOPLPUSH	ListOperations. .rightPopAnd LeftPush	Y	key: sourceKey : \${sourceKey} }; destKey: \${ destKey}
	RPUSH	ListOperations. .rightPush	Y	
	RPUSHX	ListOperations. .rightPushIf Present	Y	

Table 1-30: Set-type operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
Set	SADD	SetOperations. add	Y	
	SCARD	SetOperations. size	Y	
	SDIFF	SetOperations. difference	Y	

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	<b>SDIFFSTORE</b>	<b>SetOperations.differenceAndStore</b>	<b>Y</b>	
	<b>SINTER</b>	<b>SetOperations.intersect</b>	<b>Y</b>	
	<b>SINTERSTORE</b>	<b>SetOperations.intersectAndStore</b>	<b>Y</b>	
	<b>SISMEMBER</b>	<b>SetOperations.isMember</b>	<b>Y</b>	
	<b>SMEMBERS</b>	<b>SetOperations.members</b>	<b>Y</b>	
	<b>SMOVE</b>	<b>SetOperations.move</b>	<b>Y</b>	
	<b>SPOP</b>	<b>SetOperations.pop</b>	<b>Y</b>	
	<b>SRANDMEMBER</b>	<b>SetOperations.randomMember randomMembers distinctRandomMembers</b>	<b>Y</b>	
	<b>SREM</b>	<b>SetOperations.remove</b>	<b>Y</b>	
	<b>SUNION</b>	<b>SetOperations.union</b>	<b>Y</b>	
	<b>SUNIONSTORE</b>	<b>SetOperations.unionAndStore</b>	<b>Y</b>	
	<b>SSCAN</b>	<b>SetOperations.scan</b>	<b>Y</b>	

Table 1-31: SortedSet-type operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
SortedSet	ZADD	ZSetOperations.add	Y	
	ZCARD	ZSetOperations.size/zCard	Y	
	ZCOUNT	ZSetOperations.count	Y	
	ZINCRBY	ZSetOperations.incrementScore	Y	
	ZRANGE	ZSetOperations.range rangeWithScores	Y	
	ZRANGEBYSCORE	ZSetOperations.rangeByScore rangeByScoreWithScores	Y	
	ZRANK	ZSetOperations.rank	Y	
	ZREM	ZSetOperations.remove	Y	
	ZREMRANGEBYRANK	ZSetOperations.removeRange	Y	
	ZREMRANGEBYSCORE	ZSetOperations.removeRangeByScore	Y	
	ZREVRANGE	ZSetOperations.reverseRange reverseRangeWithScores	Y	

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	ZREVRANGEBYSCORE	ZSetOperations.reverseRangeByScore reverseRangeByScoreWithScores	Y	
	ZREVRANK	ZSetOperations.reverseRank	Y	
	ZSCORE	ZSetOperations.score	Y	
	ZUNIONSTORE	ZSetOperations.unionAndStore	Y	
	ZINTERSTORE	ZSetOperations.intersectAndStore	Y	
	ZSCAN	ZSetOperations.scan	Y	
	ZRANGEBYLEX	ZSetOperations.rangeByLex	Y	
	ZLEXCOUNT	Not supported		
	ZREMRANGEBYLEX	Not supported		

Table 1-32: HyperLogLog operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
HyperLogLog	PFADD	HyperLogLogOperations.add	Y	
	PFCOUNT	HyperLogLogOperations.size	Y	

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	PFMERGE	HyperLogLogOperations.union	Y	key: dest:\${destination}

Table 1-33: Publish/Subscribe operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
Pub/Sub	PSUBSCRIBE	Not supported		
	PUBLISH	RedisOperations.convertAndSend	Y	key: msg:\${msg}
	PUBSUB	RedisMessageListenerContainer.setMessageListeners.addMessageListener	N	
	PUNSUBSCRIBE	Not supported		
	UNSUBSCRIBE	Not supported		

Table 1-34: Transaction operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
Transaction	DISCARD	RedisOperations.discard	Y	
	EXEC	RedisOperations.exec	Y	key: execRaw

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
	<b>MULTI</b>	RedisOperations.multi	Y	
	<b>UNWATCH</b>	RedisOperations.unwatch	Y	
	<b>WATCH</b>	RedisOperations.watch	Y	

Table 1-35: Script operations

Data structure or object	Operation	Method in Spring Data Redis	EagleEye supported by EDAS	Remarks
<b>Script</b>	<b>EVAL</b>	ScriptExecutor.execute	Y	key: null
	<b>EVALSHA</b>	ScriptExecutor.execute	Y	key: null
	<b>SCRIPT EXISTS</b>	RedisScriptingCommands.scriptExists	N	
	<b>SCRIPT FLUSH</b>	RedisScriptingCommands.scriptFlush	N	
	<b>SCRIPT KILL</b>	RedisScriptingCommands.scriptKill	N	
	<b>SCRIPT LOAD</b>	RedisScriptingCommands.scriptLoad	N	

## 1.4.5 Batch operations

In the EDAS console, you can run machine commands to perform batch operations on the ECS instances with EDAS Agent installed.

### Procedure

1. [Log on to the EDAS console.](#)

2. In the left-side navigation pane, choose **Batch Operations > Machine Commands**.
3. On the **Batch Operations** page, select a region and namespace.
4. In the **Machine Commands** section, click **By Clusters**, **By Applications**, or **By Instances** to determine the operation level.

5.



**Note:**

This topic describes operations at the cluster level. The procedures at the other two levels are similar.

Click **Add** next to **Select Cluster**. In the **Select Cluster** dialog box, select a cluster (or search for the target cluster by performing a keyword search for its name) in the field on the left. Click **>** to add the cluster to the **Selected** field on the right. Then click **OK**.

6. Enter a command in the **Command** field.
7. (Optional) Select an operation range.
  - Skip this step if all the selected items are ECS clusters, common applications, or common single-server instances. The system uses the admin account to log on to instances and run commands.
  - If the selected items include Swarm or Kubernetes clusters, Docker or Kubernetes applications, or Docker single-server instances, select **Execute in Host**, **Execute in Docker Container**, or **Execute in Host and Docker Container** (or select the three options). The system uses the admin account to log on to the host and run commands, and uses the root account to log on to the Docker container and run commands.
8. Click **Run**.

## Result



- **View operation results and details**

You are redirected to the View Details page after commands are executed. The View Details page includes the Overview, Basic Information, and Details tabs.

- The Overview tab page shows the comprehensive analysis results of the command execution for batch operations, the number of successful and failed execution instances, and the time consumption.
- The Basic Information tab page shows the batch operator, operating time, and executed commands.
- The Details tab page shows the IP addresses and statuses (successful or failed) of the ECS and Docker instances for batch operations, and the command execution details.

The Execution Details section shows the detailed command execution processes on instances. If command execution fails, an error message that indicates the cause is returned.

In this case, select the instance and click Retry. You can rerun the command on the selected instance.

- **View operation records**

On the Batch Operations page, view the batch operation record in the lower section. The record contains the operator name, creation time, end time, commands, and status (indicated by the execution results).

- If the current account is the primary account, you can view all the batch commands that are executed by the primary account and all its RAM users.
- If the current account is a RAM user, you can view only the batch commands that are executed by this RAM user.

The entries in the operation record are sorted in descending order by time. You can sort the entries by operator name, creation time, or end time.

Click View in the Details column to go to the Details page.

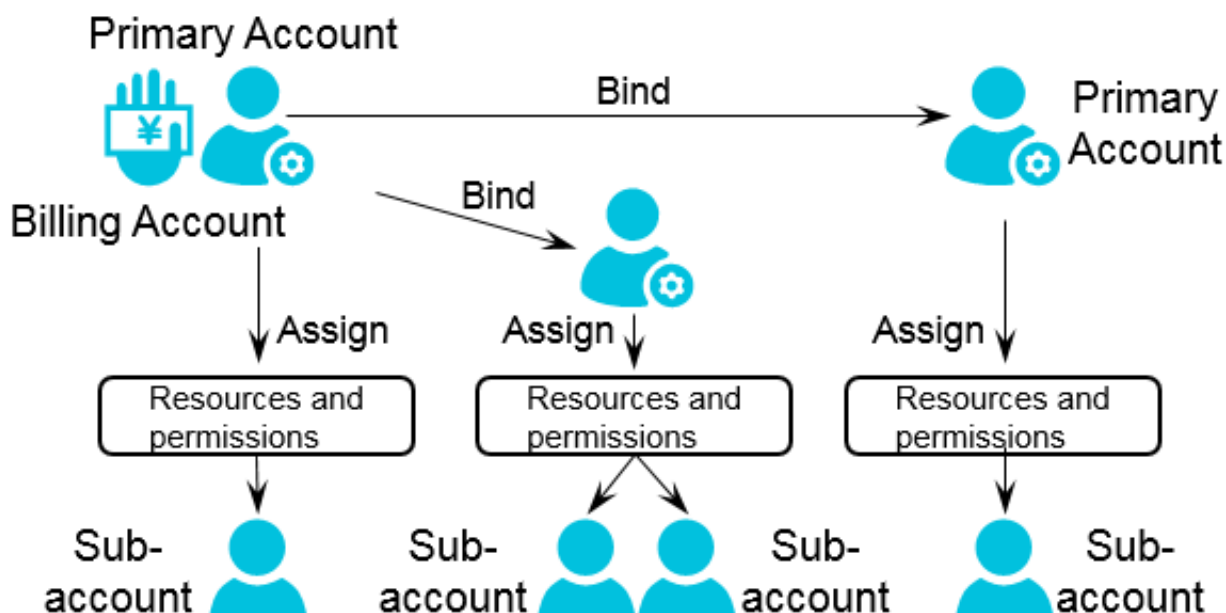
## 1.4.6 System management

### 1.4.6.1 Introduction to the EDAS account system

EDAS provides a comprehensive primary and RAM user management system. A primary account can assign permissions and resources to multiple RAM users

as needed in accordance with the minimum permission principle. This lowers the risks to enterprise information security and reduces the work burden on the primary account.

EDAS account system



### 1.4.6.2 Manage RAM users

EDAS supports the Alibaba Cloud Resource Access Management (RAM) account system. You can create RAM users under your primary account to avoid sharing your account key with other users, and assign minimum permissions to RAM users so staff can complete different types of jobs with different user identities. This allows for effective enterprise management. This topic is divided into the following parts:

#### Context

##### Introduction to RAM users

When you use your primary account to operate EDAS, you can assign different roles and resources to the RAM users under the primary account to complete different types of jobs with different user identities, such as application administrator (with the permissions to create, start, query, and delete applications) and O&M administrator (with the permissions to list resources, check application monitoring, and manage alarm rules, throttling rules, and degradation rules). This primary and RAM user permission model works in a similar way to the system and common

user model in a Linux operating system, where system users can grant or revoke permissions to or from common users.

#### Description of RAM users:

- RAM users are created by a primary account in the RAM system. Validity checks are not required provided that each RAM user under the same primary account has a unique name.
- A dedicated logon portal is available for RAM users. For more information about the logon portal, see the relevant description of the RAM console.

#### Procedure

##### Create a RAM user

1. *Log on to the EDAS console.*
2. In the left-side navigation pane, choose Account Management > Sub-accounts.
3. Click Bind Sub-Account in the upper-right corner. In the Bind Sub-Account dialog box, enter an EDAS account. Click Add Account to enter multiple EDAS accounts. Click OK to save the settings.
4. The User Management page shows a new username, indicating that an EDAS user is successfully created.

##### Account operations

5. Click the following options in the Actions column for account operations.

Operation	Description
Manage Roles	In the Manage Roles dialog box, select a role for the account and click OK to save the settings.
Application Permission	In the Application Permission dialog box, select the target application and click OK to save the settings.
Resource Group Permission	In the Resource Group Permission dialog box, select the target resource group and click OK to save the settings.
Unbind	Click Unbind to unbind the account. All the permissions and applications that are assigned to the account are deleted when the account is unbound. RAM users need to log on to the RAM system to delete accounts.

#### 1.4.6.2.1 RAM user overview

When you use your primary account to operate EDAS, you need to complete different types of jobs with different user identities, such as application administrator (with the permissions to create, start, stop, query, and delete applications) and operation administrator (with the permissions to list resources, check application monitoring data, and manage alarm rules, throttling rules, and degradation rules). You can allocate different roles and resources to the RAM users under the primary account to complete different types of jobs with different user identities. This primary account and RAM user permission model works in a similar way to the system and common user model in a Linux operating system, where system users can grant or revoke permissions to or from common users.

##### Primary account and RAM user relationship

- In the EDAS system, you can bind your primary account to a RAM user to avoid sharing your account key with other users, and assign minimum permissions to the RAM user to complete different types of jobs with different user identities for effective enterprise management.
- When a primary account is bound to a RAM user, their binding relationship is valid only within EDAS, and both are independent accounts in other environments.
- A primary account can be a primary account with RAM users or be a RAM user under another primary account.

#### 1.4.6.2.2 Use a primary account for RAM user operations

You can use a primary account for RAM user operations, such as Manage Role, Authorize Application, Authorize Resource Group, and Unbind. The procedures for these operations are similar. The following describes how to manage roles in detail and how to perform the other three operations briefly.

##### Context

A primary account can assign a role to a RAM user to grant the role-associated permissions to this sub-account.

##### Procedure

1. [Log on to the EDAS console.](#)
2. In the left-side navigation pane, choose System Management > Sub-Account.

3. Locate the row that contains the target RAM user, and click **Manage Roles** in the **Actions** column.
4. Select the target role and click **OK**.

After the preceding settings, the role name appears in the **Role** field for the RAM user on the **Sub-Accounts** page.



**Note:**

- **Authorize an application**

A primary account can assign an application to a RAM user to grant the application ownership to this RAM user.

Application authorization only grants the application ownership to the RAM user. To grant application operation permissions (to start or delete the application, for example) to the RAM user, assign a role to the RAM user. Therefore, application authorization is typically followed by role authorization.

- **Authorize a resource group**

A primary account can assign a resource group to a RAM user, allowing the RAM user to use resources in the resource group. For the definition of a resource group, see [Resource management](#).

- **Unbind**

Through the unbinding operation, you can release the binding relationship between a RAM user and the primary account. The relationships with the assigned role, application, and resource group are also released. If you have not bought the EDAS service for the RAM user, you cannot log on to the EDAS console by using this RAM user after unbinding.

### 1.4.6.3 Manage roles

A primary account can define different operation permissions for its RAM users by creating different roles.

#### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose **System Management > Roles**.
3. Click **Create Role** in the upper-right corner of the page.

4. Enter a role name, add the permissions in the left-side field to the right, and click OK.

After a role is added, you can perform actions on this role, such as View Permissions, Manage Permissions, and Delete.

#### 1.4.6.4 View all permissions

You can list all permissions of the EDAS system in the console.

##### Procedure

1. [Log on to the EDAS console](#).
2. In the left-side navigation pane, choose System Management > All Permissions.
3. Click a level to view the details of permissions at this level.

## 1.5 FAQ

This topic describes the common problems and solutions during product development and use.

### 1.5.1 Development FAQ

The development FAQ covers Ali-Tomcat, lightweight configuration center, HSF, HSF error codes, and other development problems.

#### 1.5.1.1 Ali-Tomcat FAQ


This topic describes the problems frequently encountered during the Ali-Tomcat development process and their solutions.

- Problem locating procedure

Ali-Tomcat may fail to start due to various errors. Check the catalina.out and localhost.log files to locate the error. If you use the Tomcat4E plug-in, you can view the detailed problem description in the Eclipse console.

- How do I distinguish an EDAS error from a code error when an exception occurs?

Check whether the last part of the error stack contains the code itself. Example:  
Caused by: com.yourcompany.yourpack.

Problem	Error message	Solution
Service authentication failure	<p>java.lang.Exception: Service authentication failed</p> <div>  <b>Note:</b>  This problem only occurs in the EDAS production environment. </div>	<ul style="list-style-type: none"> <li>The AccessKeyId and AccessKeySecret used for installing EDAS Agent are incorrect or they became incorrect due to web-based installation or other reasons. <ol style="list-style-type: none"> <li>Run <code>cat /home/admin/.spas_key/default</code>.</li> <li>Log on to the EDAS console. In the left-side navigation pane, choose Resource Management &gt; ECS. On the Instances page, click Install Agent.</li> <li>On the page that appears, check whether AccessKeyId and AccessKeySecret are set to the preceding values (case-sensitive). Web-based installation may cause case inconsistency.</li> </ol> </li> <li>The ECS instance has a delay of more than 30s. Adjust the time of the ECS instance. <ul style="list-style-type: none"> <li>Run the <code>date</code> command to check whether the date is accurate.</li> </ul> </li> </ul>
Unknown host exception	Caused by: java.net.UnknownHostException: iZ25ax7xuf5Z	iZ25ax7xuf5Z indicates the current hostname. Check whether <code>/etc/hosts</code> contains the IP address and name of the current host. If not, configure them, for example, <code>192.168.1.10 iZ25ax7xuf5Z</code> .
Port in use	Caused by: java.net.BindException: Address already in use: JVM_Bind	The port is in use. The troubleshooting method is the same as the method for troubleshooting port conflict in the lightweight configuration center.

Problem	Error message	Solution
com.ali.unit.rule.Router initialization failure	SEVERE: Context initialization failed java.lang.NoClassDefFoundError: Could not initialize class com.ali.unit.rule.Router	Address server connection failure jmenv.tbsite.net. Bind the domain. Add the following content to the hosts file to bind the domain name server address: 192.168.1.10 jmenv.tbsite.net. Change 192.168.1.10 to the IP address of your lightweight configuration center. The path to the hosts file is as follows: <ul style="list-style-type: none"><li>• <b>Windows:</b> C:\Windows\System32\drivers\etc\hosts</li><li>• <b>Linux:</b> /etc/hosts</li></ul>
QoS port binding exception, resulting in a Pandora startup failure	Cannot start pandora qos due to qos port bind exception	The QoS port is in use. The troubleshooting method is the same as the method for troubleshooting port conflict in the lightweight configuration center.
Insufficient JVM memory	java.lang.OutOfMemoryError	Set the memory size. For more information about the solution, search JVM memory settings on the Internet.
A null pointer exception during WAR package deployment	deployWAR NullPointerException	Check whether the WAR package is normal. Run jar xvf xxx.war to check whether the WAR package can be decompressed properly.
com.taobao.diamond.client.impl.DiamondEnvRepo initialization failure	Could not initialize class com.taobao.diamond.client.impl.DiamondEnvRepo	If DiamondServer data on the address server is empty, check whether the address server is correctly configured and is running stably. Access <a href="http://jmenv.tbsite.net:8080/diamond-server/diamond">http://jmenv.tbsite.net:8080/diamond-server/diamond</a> and check whether a response is properly returned.



## 1.5.1.2 Lightweight configuration center FAQ

This topic describes the common problems related to the lightweight configuration center and their solutions.

Problem	Error message	Solution
Startup fails when startup .bat and startup.sh are executed.	Java version not supported, must be 1.6 or 1.6+	Check whether Java is properly installed. If Java is not installed, install Java 1.6 or a later version.
	Unable to start embedded Tomcat servlet container	Check whether port 8080 is in use. If the port is used by another application, stop the application and run the startup script.
	Tomcat connector in failed state	Perform the following operations:  Windows:  1. Open the CMD window and run <code>netstat -aon findstr "8080"</code> . Record the last column of numbers in the queried data, that is, the process ID (PID), such as 2720. 2. Run <code>tasklist findstr "2720"</code> . The application that corresponds to the current PID, such as <code>javaw.exe</code> , appears. 3. Run <code>taskkill /PID 2720 /T /F</code> . 4. Start the lightweight configuration center again.  Linux:  1. Run <code>netstat -antp grep 8080</code> . The PID of the process that uses port 8080 appears, for example "2720". 2. Run <code>kill -9 2720</code> . 3. Start the lightweight configuration center again.
	Caused by: java.net.UnknownHostException: iZ25ax7xuf5Z	iZ25ax7xuf5Z indicates the current hostname. Check whether the IP address and name of the current host are configured in <code>/etc/hosts</code> . If not, configure them, for example, <code>192.168.1.10 iZ25ax7xuf5Z</code> .

- **How do I specify the startup IP address for instances with multiple NICs?**

In the startup script `startup.bat` or `startup.sh`, add the startup parameter –  
`Daddress.server.ip={accessible IP address}`.

- **How do I customize service publishing IP addresses?**

In some cases, a service must be published on a vNIC or a non-physical IP address (for example, the EIP of an ECS instance) associated with the local host. If the virtual IP address is specified by using `-Dhsf.server.ip`, an error may occur when the service is started and the service cannot be published. This is because the virtual IP address cannot be found on the NIC of the local host during publishing.

To solve this problem, EDAS provides the service IP address customization function for the provider that allows the provider to publish a service in the configuration center without specifying any IP address. After the service is successfully published, modify the IP address and then publish the service again. The consumer does not need to make any changes.

Perform the following operations:

1. After the service is published, find it in Configuration List and click Update on the right of the service.

You can also find the published service on the Services tab page.

2. On the Edit Configuration page, modify the IP address in the Content field.



**Notice:**

Do not modify the content after the IP address unless necessary. Otherwise, a service call error may occur.

3. Click OK to save the settings.
4. Restart the service. The service with the new IP address is registered again to enable the modification to take effect.

After modification, the consumer does not need to make any changes and can call the service in the normal way. You can query logs in `{user.home}\logs\configclient\config-client.log` to check the real IP address that is called by the consumer. Check the content next to the keyword [Data-received] in the logs to view the complete information about the called service.

### 1.5.1.3 HSF FAQ

- **Locate and solve HSF problems**

**HSF problems are logged in `/home/admin/logs/hsf/hsf.log`. If any HSF problem occurs, check this file to locate the error. HSF errors have corresponding error codes. You can use these error codes to find the appropriate solution.**

- **Set the timeout period for an HSF service**

**Use the HSF tags `methodSpecials` and `clientTimeout` to configure the timeout period.**

- **`methodSpecials`: sets the timeout period (unit: ms) for a single method.**
- **`clientTimeout`: sets the general timeout period (unit: ms) for all methods in the interface.**

**The timeout period settings are sorted in descending order of priority as follows:**

**Consumer `methodSpecials` > Consumer `clientTimeout` > Provider `methodSpecials` > Provider `clientTimeout`**

**An example of the Consumer tag settings is as follows:**

```
<hsf:consumer id="service" interface="com.taobao.edas.service.
SimpleService"
version="1.1.0" group="test1" clientTimeout="3000"
target="10.1.6.57:12200? _TIMEOUT=1000" maxWaitTimeForCsAddress="
5000">
  <hsf:methodSpecials>
    <hsf:methodSpecial name="sum" timeout="2000" ></hsf:methodSpec
ial>
  </hsf:methodSpecials>
</hsf:consumer>
```

- **HSF invalid call is removed**

**Error message:**

**invalid call is removed because of connection closed**

**Causes:**

- **Transient network disconnection: After the provider and consumer establish a connection, the consumer initiates a call request. An error is returned if the provider is still processing this request within the timeout period of**

the consumer and the consumer is disconnected due to network and other problems.

- **Provider restart:** After the consumer initiates a request, it waits for a response from the provider. If the consumer is restarted at this time, the socket is disconnected and the consumer receives an operating system connection closed callback. In this case, an error is returned.

#### **Solution**

If the service is idempotent, retry the service. Check the HSF provider network. This problem is often caused by network disconnection (transient disconnection).

- **Binding an IP address and port fails upon HSF startup**

**Problem:** An error is returned when HSF is started. The error message is as follows:

**Java.net.BindException:** Can't assign requested address

**Cause:** The current IP address and port cannot be obtained.

**Solution:** Set the following JVM parameter:

**-Dhsf.server.ip=IP address of your local network adapter -Dhsf.server.port=12200**

- **Keep user logs from being overwritten**

**Problem:** After EDAS is used, the log4j log cannot be generated.

**Cause:** The log4j log is overwritten and thus cannot be generated.

**Solution:** Set the JVM parameter `Dlog4j.defaultInitOverride` to false to generate user logs.

- **HSF Others**

**Error message:** The following error is reported during startup:

**java.lang.IllegalArgumentException: HSFApiConsumerBean.ServiceMetadata.  
ifClazz is null.**

**Solution:** The class for the interface cannot be loaded. Check that the interface class is loaded to class path.

**Error message:** failure to connect 10.10.1.1

**Solution:**

**Check whether the HSF services are in the same VPC and the same region. If not, they cannot be connected.**

**Check whether the HSF services are in the same security group. If not, enable port 12200.**

**Run `telnet 10.10.1.1 12200` to check whether the port can be connected. If the port cannot be connected, check the firewall settings of the ECS instance with the IP address 10.10.1.1.**

### 1.5.1.4 HSF error codes

Error code: HSF-0001

**Error message:**

**HSFServiceAddressNotFoundException:** This error message is returned when the address of the target service to be called is not found.

**Description:**

**The target service to be called is xxxx, which is in the xxxx group.**

**Solution:**

- 1. In the case of name mismatch, check whether the service name, version, and group (case-sensitive, without leading or trailing spaces) are set consistently for the provider and consumer.**
- 2. Check whether an error is reported when the Tomcat container is started. Go to the Tomcat installation directory and check whether `/logs/catalina.out` localhost.log. 2016-07-01 (current date) contains any errors. If yes, fix the errors.**

3. No service group is created. Log on to the EDAS console. In the left-side navigation pane, choose Service Marketplace > Service Groups to check whether a service group is created for the application. Example:

```
<hsf:provider
    id="sampleServiceProvider" interface="com.alibaba.edas.
SampleService" ref="target"
    version="for-test" group="your-namespace" ></hsf:provider>
```

The corresponding group named *your-namespace* must exist in the service group list.

4. In the case of failed authentication, go to the ECS instance that corresponds to the provider and check whether `/home/admin/configclient/logs/configclient.log` contains the `spas-authentication-failed` error. If this error exists:

- No service group is created.
- The AccessKeyId and AccessKeySecret used for installing EDAS Agent are incorrect or they became incorrect due to web-based installation or other reasons.
  - a. Run `cat /home/admin/.spas_key/default`.
  - b. Log on to the EDAS console. In the left-side navigation pane, choose Resource Management > ECS and click Install Agent.
  - c. On the page that appears, check whether AccessKeyId and AccessKeySecret are set to the preceding values (case-sensitive). Web-based installation may cause case inconsistency.
  - d. The IP address of the provider cannot be pinged. If multiple NICs exist, publish the IP address that is inaccessible from the consumer. Use `-Dhsf.server.ip` to specify the IP address of the provider.

5. The service call is initiated too early. A call is initiated before ConfigServer pushes the address, resulting in an error. Add `maxWaitTimeForCsAddress` to the consumer configuration file. For more information, see *Developer Guide*.

6. In the case of a data push error, contact a developer for troubleshooting.

Error code: HSF-0002

Error message:

Consumer error: HSFTimeOutException

Solution:

- Check whether the network of the ECS instance is healthy. Check whether the IP address of the provider can be pinged.
- If the processing time of the provider is greater than 3s, find the service execution timeout logs in hsf.log of the provider to locate the specific class and method:
  - A serialization error has occurred for the provider. Check the codes. The stream type, files, and oversized objects may cause a serialization error, and they cannot be transferred.
  - The code performance is inadequate. Optimize the code.
  - The logic of the provider is complex, and service processing requires more than 3s. Modify the timeout period. (See the *Developer Guide*.)
- Timeout occurs occasionally, and GC occurs for both the provider and consumer. Check the GC logs of the provider and consumer. GC that requires a long time may result in timeout. For more information about troubleshooting methods, search Java GC optimization on the Internet.
- The consumer is heavily loaded and fails to send the request, resulting in timeout. Add more instances for the consumer.

Error code: HSF-0003

**Error message:**

**Consumer error:** java.io.FileNotFoundException: /home/admin/logs/hsf.log (The specified path is not found.)

**Solution:** The default HSF log path cannot be found or is under access control. Load `-DHSF.LOG.PATH=xxx` during startup to modify the default path.

Error code: HSF-0005

**Error message:**

**Startup error:**

**java.lang.IllegalArgumentException:** This error message is returned when the object to be published as a service is not configured. The service name is com.taobao.hsf.jar.test.HelloWorldService:1.0.zhouli.

**Solution:**

**The target attribute is missing from the bean of the provider. Check the configuration file.**

**The implementation class of the service specified by target does not exist. Check the configuration file.**

Error code: HSF-0007

**Error message:**

**java.lang.IllegalArgumentException: This error message is returned during startup when the serialization type is not supported.**

**Solution: The serializeType or preferSerializeType attribute is incorrectly configured for the bean of the provider. Check the configuration file. We recommend that you use Hessian or Hessian 2.0.**

Error code: HSF-0008

**Error message: java.lang.IllegalArgumentException, which is returned when the service type specified by ProviderBean is not [com.taobao.hsf.jar.test.HelloWorldServiceImpl].**

**Solution: serviceInterface configured for the bean of the provider is not an interface. serviceInterface must be set to an interface name. Check the configuration file.**

Error code: HSF-0009

**Error message: java.lang.IllegalArgumentException, which is returned when the real service object [com.taobao.hsf.jar.test.HelloWorldServiceImpl@10f0a3e8] does not implement the specified interface [com.taobao.hsf.jar.test.HelloWorldService].**

**Solution: No interface is implemented for the bean specified by target of the provider. Check that the corresponding interface is implemented in the interface class.**

Error code: HSF-0014

**Error message: java.lang.IllegalArgumentException, which is returned when the interface class specified by ProviderBean does not contain [com.taobao.hsf.jar.test.HelloWorldService1].**

**Solution: The serviceInterface attribute of the provider is incorrectly configured, and the specified interface does not exist.**



Error code: HSF-0016

**Error message:**

**Startup error: Failed to start the HSF provider.**

**Solution:**

- Check whether port 12200 is already occupied. A server binding failure may cause a startup failure.
- If multiple NICs and an instance with a public network IP address exist, specify the local IP address by using `-Dhsf.server.ip`.

Error code: HSF-0017

**Error message:**

**Startup error: java.lang.RuntimeException: [ThreadPool Manager] Thread pool allocated failed for service [com.taobao.hsf.jar.test.HelloWorldService:1.0.zhouli]: balance [600] require [800]**

**Solution:** The allocated thread pool is insufficient. By default, the maximum thread pool size of HSF is 600. You can set the JVM parameter `-Dhsf.server.max.poolsize=xxx` to modify the default global maximum thread pool size.

Error code: HSF-0020

**Error message:**

**WARN taobao.hsf - HSF service: com.taobao.hsf.jar.test.HelloWorldService:1.0.zhouli, which is returned when initialization is repeated.**

**Solution:** In one HSF process, a service is uniquely identified by the service name and version. Services with the same name and version but of different groups cannot be published or subscribed to in a single process. Check the configuration file. For example, the service `com.taobao.hsf.jar.test.HelloWorldService` cannot be published or subscribed to in a single process if the following two configurations exist in the configuration file:

`com.taobao.hsf.jar.test.HelloWorldService 1.0 groupA`

`com.taobao.hsf.jar.test.HelloWorldService 1.0 groupB`

Error code: HSF-0021

**Error message:**

**Startup error:**

**java.lang.IllegalArgumentException**, which is returned when the interface class specified by **ProviderBean** does not contain **[com.taobao.hsf.jar.test.HelloWorldService1]**.

**java.lang.IllegalArgumentException**: This error message is returned when the interface class specified by **ConsumerBean** does not contain **[com.taobao.hsf.jar.test.HelloWorldService1]**.

**Solution:** The **serviceInterface** attribute of **HSFSpringProviderBean** is incorrectly configured, the specified interface does not exist (HSF-0014), or the interface specified by the **interfaceName** field in **HSFSpringConsumerBean** does not exist (HSF-0021).

Error code: HSF-0027

**Error message:** [HSF-Provider] HSF thread pool is full

**Solution:**

The processing speed of a service on the HSF provider is too slow, and requests from the client cannot be processed in time. As a result, the thread pool of the HSF provider for service execution reaches the maximum value. By default, HSF dumps the **/home/admin/logs/hsf/HSF\_JStack.log** file (default path). View the **HSFBizProcessor-xxx** thread stack information about the file and analyze the performance bottleneck.

The maximum number of threads of HSF is 600 by default. To increase the number, change the value of the **-Dhsf.server.max.poolsize=xxx** JVM parameter.

Error code: HSF-0030

**Error message:** [HSF-Provider] cannot find the method to be called.

**Solution:**

- The method is not provided by the provider. Log on to the EDAS console. In the left-side navigation pane, choose **Application Management** and click the name of the application that corresponds to the service provider to go to the **Application Details** page. In the left-side navigation pane, choose **Services** and check whether the corresponding service is successfully published.

- An earlier version and a later version coexist. The wrong version of a service is called. View the details of the service by using the preceding method.
- The interfaces of the provider and the consumer are inconsistent. For example , the provider provides `java.lang.Double`, whereas the consumer uses `double` to call the provider.
- Inconsistent interface classes are loaded for the provider and consumer. Check whether the MD5 values in the interface-contained JAR packages of the provider and consumer are consistent.

Error code: HSF-0031

**Error message:** [HSF-Provider] takes xxx ms to execute the xxx method of the xxx HSF service. The time approximates the timeout period.

**Solution:** The provider prints this log when the timeout period minus the actual time elapsed is less than 100 ms. The timeout period is 3s by default.

- If the timeout period is short, for example, less than 100 ms, this log is printed in each call, and you can ignore it.
- If this log is still printed for a long timeout period, it indicates service execution is slow. Analyze the performance bottleneck in service execution.

Error code: HSF-0032

**Error message:** please check log on server side that unknown server error happens.

**Solution:** An uncaptured error occurs when the provider processes a request. Check the `hsf.log` file of the provider.

Error code: HSF-0033

**Error message:** Serialization error during serialize response.

**Solution:**

An error occurs when the provider returns data during serialization. Check the `hsf.log` file of the provider.

If the log file contains "must implement `java.io.Serializable`", implement a serializable interface on the DO.

Error code: HSF-0038

**Error message:** Multiple NICs are configured for the HSF provider, and the HSF provider is bound to an incorrect IP address.

**Solution:** Add `-Dhsf.server.ip=xxx.xxx.xx.xx` to the JVM startup parameters to specify the desired IP address.

Error code: HSF-0035

**Error message:** RPCProtocolTemplateComponent invalid address.

**Solution:** A TCP connection cannot be established between the current instance and the corresponding address. Check whether the corresponding remote address and port can be connected.

### 1.5.1.5 Other development problems

- **Q:** How do I develop an HSF application by using a framework other than Spring?

**A:** We recommend that you use Spring to develop HSF applications. If you use another framework, you can develop applications by using LightAPI. For more information, see the *Developer Guide*.

- **Q:** Can I access the services in a production environment directly from a development environment?

**A:** No. The production environment is isolated for security.

- **Q:** Does EDAS provide APIs? What functions do they have?

**A:** EDAS provides APIs to implement resource query, application lifecycle management, and account management.

- **Q:** Does EDAS support other languages in addition to Java?

**A:** HSF is developed in Java by default. HSF clients are also available in C++ and PHP, allowing you to access the backend HSF services provided by Java.

## 1.5.2 Usage FAQ

Common problems during development are related to accounts, resources, application lifecycle, and monitoring and alarms.

### 1.5.2.1 Account management

- **Q:** Can I create multiple RAM users?

**A:** Yes.

- **Q:** Who can grant application operation permissions for RAM users?

EDAS allows you to grant application operation permissions to RAM users only by using the primary account.

### 1.5.2.2 Resource management

- **Q: Why doesn't the a prompt appear after EDAS Agent installation and EDAS Agent version is not displayed?**

**A: Perform the following steps to troubleshoot the problem:**

- 1. Log on to the ECS instance and check `/home/admin/edas-agent/logs/agent.log`. If `UnauthorizedException` exists, check whether:**
    - **The `AccessKeyId` and `AccessKeySecret` used for installing EDAS Agent are incorrect or they became incorrect due to web-based installation or other reasons.**
      - a. Run `cat /home/admin/.spas_key/default`.**
      - b. Log on to the EDAS console. In the left-side navigation pane, choose `Resource Management > ECS`. On the `Instances` page, click `Install Agent`.**
      - c. On the page that appears, check whether `AccessKeyId` and `AccessKeySecret` are set to the preceding values (case-sensitive). Web-based installation may cause case inconsistency.**
    - **The region script used for installation is incorrect.**
  - 2. Check `/home/admin/edas-agent/logs/std.log`. If `"Java not found"` or other error messages exist, run `java -version` to check whether the Java version is 1.7. If the version is Java 1.5, run `rpm -e` corresponding installed RPM name to remove it and reinstall EDAS Agent.**
- **Q: Why is the status `Unknown` or `Abnormal` after EDAS Agent is installed?**

**A: Check the `std.log` and `agent.log` files in the `/home/admin/edas-agent/logs` directory of the ECS instance.**

- **`std.log` is the log of EDAS Agent installation.**
- **`agent.log` is the runtime log of EDAS Agent.**

**The possible causes are as follows:**

- **If `"Permission denied"` or `"Not such file"` is found in those logs, the possible cause is the lack of required file and directory permissions. In this case, check whether**

the admin account has permissions for all files in the `/home/admin` directory, and reinstall EDAS Agent.

- Check whether the ECS hostname is the same as that in the `/etc/hosts` file. If not, modify the name and restart EDAS Agent.

```
/home/admin/edas-agent/bin/shutdown.sh  
/home/admin/edas-agent/bin/startup.sh
```

- Q: Which version of Java is EDAS using? Can I choose another version?

A: EDAS provides Java 7 and Java 8. Java 7 is used by default. You can select a Java version when installing EDAS Agent. Run the following command to select a Java version:

```
install.sh -ak -sk [-java <7(default)|8>]
```

- Q: What can happen if the heartbeat process of EDAS Agent stops?

A: If no application is installed on that ECS instance, no services are affected.

If an application is installed on that ECS instance, the *real-time status* of the ECS instance in the ECS instance list of the application (which appears in the lower part of the page after you select the application on the Application Management page and go to the Basic Information page) changes to Agent Abnormal. Any commands for the ECS instance, such as deploy, start, and stop, are ineffective.

Log on to the ECS instance and run `sudo -u admin /home/admin/edas-agent/bin/startup.sh` to start EDAS Agent. Troubleshoot the EDAS Agent crash as follows:

Check whether error messages are logged in `/home/admin/edas-agent/logs/agent.log`.

Check whether the system memory is sufficient. If the system memory is insufficient, the OOM Killer may be triggered. For more information, search for Linux OOM Killer on the Internet. If the OOM Killer is triggered, we recommend that you check the system memory usage and adjust memory allocation.

- **Q: What should I do if the Ali-Tomcat container suddenly exits?**

**A: Log on to the EDAS console to start the corresponding application. Troubleshoot the crash of Ali-Tomcat as follows:**

- Check whether error messages are logged in `/home/admin/tomcat (installation directory)/logs/catalina.out`.
- Check whether the system memory is sufficient. If the system memory is insufficient, the OOM Killer may be triggered. For more information, search for Linux OOM Killer on the Internet. If the OOM Killer is triggered, we recommend that you check the system memory usage and adjust the memory allocation policy.

- **Q: Why doesn't EDAS Agent start after the system is restarted?**

**A: Currently, EDAS Agent of the CentOS 6.5 version supports automatic startup. Testing is not performed in other systems for the moment. If EDAS Agent is not started, run the following program:**

```
sudo -u admin /home/admin/edas-agent/bin/startup.sh  
/usr/alisys/dragon/bin/DragonAgent
```

### 1.5.2.3 Application lifecycle

- **Q: Does EDAS Agent automatically restart after the target ECS instance is restarted?**

**A: Yes. EDAS Agent automatically restarts after you restart the target ECS instance, but your Tomcat does not.**

- **Q: Why cannot I start EDAS Agent?**

**A: Run the following command on the ECS instance where the EDAS console is deployed to check whether the instance is reachable:**

```
ping edas-internal.console.aliyun.com
```

**Then, check whether the security token file is correctly set:**

```
cat /home/admin/.spas_key/default
```

- **Q: Can I deploy multiple applications on the same ECS instance in EDAS?**

**A: EDAS allows you to deploy only one application on a single ECS instance.**

- **Q: Can I set the URL of an application deployment package to any address?**

**A: Ensure that the application deployment package can be downloaded from this URL.**

- **Q: Why does an application operation (such as starting, stopping, or deploying an application) fail?**

**A: Check whether EDAS Agent runs properly on the ECS instance with the failed operation. An application operation usually fails because EDAS Agent is not running properly.**

- **Q: Why don't the ECS instances under my account appear in the instance selection dialog box when I create an application?**

**A: Check whether EDAS Agent is correctly installed on your ECS instances. Install EDAS Agent based on the procedure described in Resource management > ECS instance list > Install EDAS Agent.**



**Notice:**

**Be sure to select the correct region when installing EDAS Agent.**

- **Q: Why is the ECS instance status in the EDAS console "Unknown"?**

**A: EDAS Agent reports heartbeat data periodically to the EDAS console. If EDAS Agent stops reporting heartbeat data, the ECS instance is set to the Unknown state after a certain time. This problem is typically caused by the stopping of EDAS Agent.**

- **Q: Why doesn't the service list appear while services can be called normally?**

**A: APIs have generics, but the generics do not have a specific type, which results in a failure to resolve the service list. In this case, modify the corresponding code**

**.**

- **Q: What should I do when a service appears as Normal in the service list but I cannot call it?**

**A:**

- 1. Check whether the group that corresponds to the service provider has been created. If not, authentication may fail.**
- 2. Check `/home/admin/logs/hsf/hsf.log` to determine the error code, and query [HSF FAQ](#) based on the error code.**



- **Q: Can I restore an application after I delete it?**

**A: No. Application deletion is irrevocable. All application data is cleared after the application is deleted.**

- **Q: How do I perform batch or beta publishing?**

**A:**

- **If an application has multiple ECS instances, select batch publishing and set the number of batches to a value greater than 1 to publish the application in batches.**
- **If an application has multiple ECS instances, set some of these instances to beta instances. You can separately publish the application to the beta instances. Only beta instances are updated during publishing. Other instances are not updated.**

- **Q: How do I share cluster sessions after deploying my application on multiple ECS instances?**

**A: Currently, EDAS does not support distributed session management. You can use a distributed cache system (such as OCS and Redis) to manage distributed sessions.**

- **Q: How do I set the health check URL?**

**A: When an application is published, the provided WAR file is automatically deployed in the Tomcat directory. Therefore, the WAR package name is added to the health check URL by default, and files in the WAR package must return the 200-400 HTTP codes. For example, assume a WAR package is named *order.war* and includes the file *index.jsp*. The health check URL can be set to <http://127.0.0.1:8080/order/index.jsp>.**

- **Q: Can I use SLB for load balancing after deploying my application on multiple ECS instances?**

**A:**

- 1. HTTP-based web applications in EDAS use SLB for load balancing. You can configure SLB on the application configuration page of EDAS.**
- 2. Load balancing does not need to be considered for applications that belong to RPC providers of EDAS. EDAS natively supports loading balancing for RPC providers.**

## 1.5.2.4 Monitoring and alarms

- **Q: Can infrastructure monitoring provide application-specific data?**

**A: Infrastructure monitoring provides data by application and aggregates the metrics of the ECS instances where the application is deployed, providing an aggregated result for you. Infrastructure monitoring also allows you to view the metrics of a single ECS instance.**

- **Q: What metrics does infrastructure monitoring provide?**

**A: The infrastructure monitoring function monitors CPU usage, memory usage, network, disk usage, disk I/O speed, and disk IOPS of ECS instances.**

## 2 API Gateway

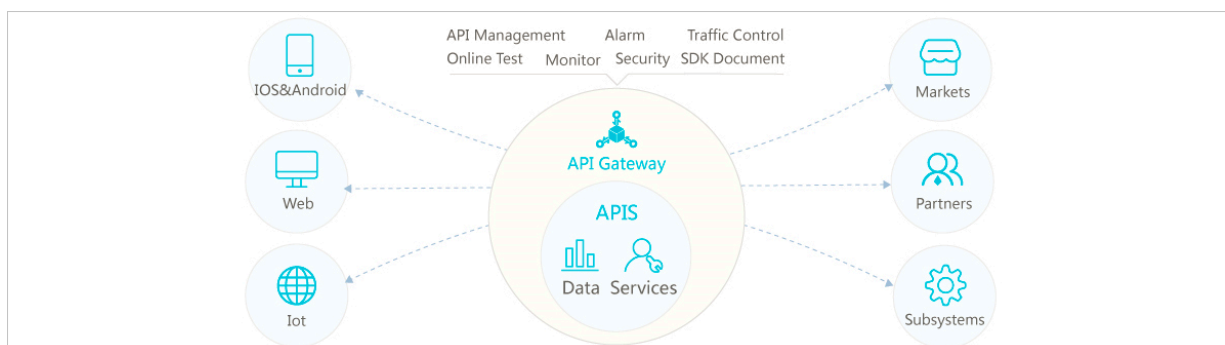
### 2.1 What is API Gateway?

API Gateway provides a comprehensive suite of API hosting services that help you share capabilities, services, and data with partners in the form of APIs. API Gateway also enables you to release APIs in the marketplace for other developers to purchase and use.

- API Gateway provides multiple security mechanisms to secure APIs and reduce the risks introduced by open APIs. These mechanisms include protection against replay attacks, request encryption, identity authentication, permission management, and throttling.
- API Gateway provides API lifecycle management that allows you to create, test, publish, and unpublish APIs. It also generates SDKs and API documentation to improve API management and iteration efficiency.

API Gateway allows enterprises to reuse and share their capabilities with each other so that they can focus on their core business.

Figure 2-1: API Gateway



### 2.2 Log on to the API Gateway console

This topic demonstrates how to log on to the API Gateway console from Google Chrome.

#### Prerequisites

- Before logging on to the Apsara Stack console, make sure that you obtain the IP address or domain name of the Apsara Stack console from the deployment personnel. The access address of the Apsara Stack console is `http://IP address or domain name of the Apsara Stack console/manage`.
- We recommend that you use the Chrome browser.

## Procedure

1. Open your browser.
2. In the address bar, enter the access address of the Apsara Stack console in the format of `http://IP address or domain name of the Apsara Stack console/manage`, and then press Enter.
3. Enter the correct username and password.
  - The system has a default super administrator with the username `super`. The super administrator can create system administrators who can create other system users and notify them of their default passwords by SMS or email.
  - You must modify the password of your username as instructed when you log on to the Apsara Stack console for the first time. To improve security, the password must meet the minimum complexity requirements, that is to be 8 to 20 characters in length and contain at least two types of the following characters: English uppercase/lowercase letters (A to Z or a to z), numbers (0 to 9), or special characters (such as exclamation marks (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)).
4. Click LOGIN to go to the Dashboard page.

## 2.3 Quick start for consumers

### 2.3.1 Overview

You can use API Gateway to call the API services enabled by other Alibaba Cloud users or third-party service providers. API Gateway provides a series of management and support services.

Call an API based on the following conditions:

- **API:** The API that you call is clearly defined by API parameters.
- **Application:** The application that you use to call the API has a key pair that uniquely identifies you.

- **Authorization relationship between the API and application:** An application can be used to call an API only when the application has been granted the permission to call that API. This permission is granted through authorization.

### 2.3.2 Step 1: View API settings

You must create an application and provide the application ID to the API provider before the application can be authorized in the API Gateway console. For more information about applications, see [Create an application](#). Assume that you have created an application and the API provider has authorized your application.

#### Procedure

1. [Log on to the API Gateway console](#).
2. Click the Applications tab to go to the Applications tab.  
Your created applications are displayed on the Applications tab.
3. Click the application ID to go to the application details page.  
Basic Information, AppKey, and Callable APIs are displayed.

On the details page,

- The AppKey section shows the AppKey and the AppSecret of an application. Your API request must contain the AppKey and the AppSecret. API Gateway verifies your identity based on this key pair.
- The Callable APIs section shows the APIs that applications have been authorized to call. If the API provider has authorized the applications, the corresponding APIs are displayed. Click the management icon in the Actions column corresponding to the API and choose View Details from the shortcut menu to view details of the API.

### 2.3.3 Step 2: Create an application

Applications are the identities that you use to call APIs. You can own multiple applications. Your applications can be authorized to call different APIs based on your business requirements. Applications instead of user accounts are authorized to call APIs. In the API Gateway console, you can create, modify, or delete applications, view details of applications, manage key pairs, and view callable APIs of applications.

Each application has an AppKey and an AppSecret. You can regard them as an account and a password. When you call an API, you must pass in the AppKey as

a parameter. AppSecret is used to calculate the signature string. API Gateway authenticates the key pair to verify your identity. An application must be authorized to call an API. Both authorization and authentication are intended for applications.

You can log on to the API Gateway console to create applications on the Applications tab.

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the Applications tab to go to the Applications tab.
3. Click Create Application.
4. Specify parameters and click Create.

The application name must be globally unique. It must be 4 to 26 characters in length and can contain letters, digits, and underscores (\_). It must start with a letter.

After an application is created, the system automatically assigns an AppKey and an AppSecret to it. You must use the AppSecret to calculate the signature string. When calling an API, you must include the signature string in the request. API Gateway verifies your identity based on the signature string.

On the Applications tab, click the application ID to go to the application details page. The AppKey and AppSecret are displayed on the application details page. You can reset the AppSecret as needed.

### 2.3.4 Step 3: Obtain authorization

Authorization is the process of authorizing an application to call an API. Your applications must be authorized before they can call APIs.

You must provide your application IDs to the API provider for authorization. After authorization, you can view the APIs that your applications have been authorized to call in the API Gateway console.

The APIs that your applications have been authorized to call are displayed in the Callable APIs section on the application details page.

After the API provider authorizes your applications to call APIs, you do not need to and cannot authorize your applications.

## 2.3.5 Step 4: Call an API

**You can edit an HTTP or HTTPS request to call an API. Before calling the API, you can use API call examples of multiple programming languages in the API Gateway console to test the call.**

### Part 1: Request

#### Request URL

```
http://e710888d3ccb4638a723ff8d03837095-cn-qingdao.aliapi.com/demo/  
post
```

#### Request method

```
POST
```

#### Request body

```
FormParam1=FormParamValue1&FormParam2=FormParamValue2  
//HTTP request body
```

#### Request header

```
Host: e710888d3ccb4638a723ff8d03837095-cn-qingdao.aliapi.com  
Date: Mon, 22 Aug 2016 11:21:04 GMT  
User-Agent: Apache-HttpClient/4.1.2 (java 1.6)  
Content-Type: application/x-www-form-urlencoded; charset=UTF-8  
// The request body type. Set the request body type based on the  
actual request you want to make.  
Accept: application/json  
// The response body type. Some APIs can return data in the correspond  
ing format based on the specified response type. We recommend that you  
manually set this request header. If the request header is not set,  
some HTTP clients will use the default value */*, causing a signature  
error.  
X-Ca-Request-Mode: debug  
// Specifies whether to enable the Debug mode. This parameter is not  
case-sensitive. If it is not specified, the Debug mode is disabled by  
default. This mode is typically enabled in the API debugging phase.  
X-Ca-Version: 1  
// The API version number. Set the value to 1. Default value: 1.  
X-Ca-Signature-Headers: X-Ca-Request-Mode,X-Ca-Version,X-Ca-Stage,X-Ca-  
-Key,X-Ca-Timestamp  
// The custom request headers involved in signature calculation. The  
server will read the request headers based on this configuration to  
sign the request. This configuration does not include the Content-Type  
, Accept, Content-MD5, and Date request headers, which are already  
included in the basic signature structure. For more information about  
the signature, see Request signatures.  
X-Ca-Stage: RELEASE  
// The stage of the API. Valid values: TEST, PRE, and RELEASE. This  
parameter is not case-sensitive. The API provider can select the stage  
to which the API is published. The API can be called only after it is  
published to the specified stage. Otherwise, the system will prompt  
that the API cannot be found or that the request URL is invalid.  
X-Ca-Key: 60022326
```

```
// The AppKey of the request. You must obtain the AppKey in the API
Gateway console. Applications can only call APIs after they have been
authorized.
X-Ca-Timestamp: 1471864864235
// The request timestamp. This value is a UNIX timestamp representing
the number of milliseconds that have elapsed since January 1, 1970 00:
00:00 UTC. The timestamp is valid for 15 minutes by default.
X-Ca-Nonce:b931bc77-645a-4299-b24b-f3669be577ac
// The unique ID of the request. AppKey, API, and Nonce must be unique
within the last 15 minutes. To prevent replay attacks, you must
specify both the X-Ca-Nonce header and the X-Ca-Timestamp header.
X-Ca-Signature: FJleSrCYPGCU7dMLLTG+UD3Bc5Elh3TV3CWHtSKh1Ys=
// The request signature.
CustomHeader: CustomHeaderValue
// The custom request headers. CustomHeaderValue is used as an example
. You can set multiple custom request headers in actual requests based
on the definition of the API being called.
```

## Part 2: Response

### Status code

```
400
// The status code of the response. If the value is greater than or
equal to 200 and less than 300, the call is successful. If the value
is greater than or equal to 400 and less than 500, a client-side error
has occurred. If the value is greater than 500, a server-side error
has occurred.
```

### Response header

```
X-Ca-Request-Id: 7AD052CB-EE8B-4DFD-BBAF-EFB340E0A5AF
// The unique ID of the request. When API Gateway receives a request,
it generates a request ID and returns the request ID to the client in
the X-Ca-Request-Id header. We recommend that you record the request
ID in both the client and backend server for troubleshooting and
tracking.
X-Ca-Error-Message: Invalid Url
// The error message returned by API Gateway. When a request fails,
API Gateway returns the error message to the client in the X-Ca-Error-
Message header.
X-Ca-Debug-Info: {"ServiceLatency":0,"TotalLatency":2}
// The message can be returned only when the Debug mode is enabled.
The message is used only for reference at the debugging stage.
```

**Regardless of whether you call an API by using HTTP or HTTPS, the request must include the signature information. For more information about how to calculate and pass the encrypted signature, see [Request signatures](#).**

## 2.4 Quick start for providers

### 2.4.1 Overview

**This topic is a quick start guide for you to create and publish an API.**

#### 1. Create an API group.



2. Create an API.
3. Publish an API.
4. Authorize applications to call the API.

## 2.4.2 Create a group

You can create an API Group in the API Gateway console.

### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the Groups tab.
3. On the Groups tab, click Create Group.
4. In the Create Group dialog box that appears, set parameters and click Create.

The name of a group must be globally unique. The name must be 4 to 50 characters in length and can contain letters, digits, and underscores (\_). It must start with a letter.

## 2.4.3 Create an API

### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the APIs tab.
3. Click Create API.
4. Specify the basic information of the API and click Next.

Parameter	Description
Groups	The basic management unit of an API. You must create a group before creating an API. When you select a group, you select a region for the API.
API Name	The name of the API to be created.

Parameter	Description
<b>Authentication Mode</b>	<p><b>The authentication mode of API requests. Valid values:</b> Alibaba Cloud Applications <b>and</b> None.</p> <ul style="list-style-type: none"> <li>• <b>Alibaba Cloud Applications:</b> This mode requires the requester to pass the application authentication to call the API.</li> <li>• <b>None:</b> This mode allows any user who knows the request definition of the API to initiate a request. API Gateway directly forwards the requests to your back-end service without verifying the identity of the requesters.</li> </ul>
<b>Signature Algorithm</b>	<p><b>The algorithm used to sign API requests. Valid values:</b></p> <ul style="list-style-type: none"> <li>• HmacSHA256</li> <li>• HmacSHA1 and HmacSHA256</li> </ul>
<b>Description</b>	<b>The description of the API.</b>

5. Define API requests. Define how users call your APIs, including the request type, network protocol, URL suffix, HTTP method, and request mode.

Parameter	Description
<b>Request Type</b>	<p><b>The request type. Four options are available. However, only the common request type is supported.</b></p> <ul style="list-style-type: none"> <li>• <b>Common Request:</b> indicates common HTTP or HTTPS requests.</li> <li>• <b>Logon Request (Bidirectional Communication):</b> indicates the bidirectional control signal to register devices. It is sent from the client to the server.</li> <li>• <b>Logoff Request (Bidirectional Communication):</b> indicates the bidirectional control signal to register devices. It is sent from the client to the server. After devices are deregistered, server-to-client notifications are no longer received.</li> <li>• <b>Server-to-Client Notification (Bidirectional Communication):</b> After receiving the registration signal sent from the client, the back-end service records the device ID and send a server-to-client notification to API Gateway. Then, API Gateway sends the notification to the device. As long as the device is online, API Gateway sends the server-to-client notification to the device.</li> </ul>
<b>Network protocol</b>	<b>HTTP, HTTPS, and WEBSOCKET are supported in API calls.</b>

Parameter	Description
URL Suffix	The API request path. It corresponds to the service host . The request path can be different from the actual back-end service path. You can specify any valid and semantically-correct path as the request path. You can configure dynamic parameters in the request path, which requires users to specify path parameters in the request. At the same time, the path parameters can be mapped to query and header parameters that are received by the back-end service.
HTTP Method	You can select PUT, GET, POST, PATCH, DELETE, and HEAD.
Request Mode	<p>You can select either Request Parameter Mapping or Request Parameter Passthrough.</p> <ul style="list-style-type: none"><li>• Request Parameter Mapping indicates that you must configure request and response data mappings for query , path, and body form parameters. API Gateway only passes the configured parameters through to the back end. Other parameters are filtered out.</li><li>• Request Parameter Passthrough indicates that you do not need to configure query and body form parameters, but still must configure path parameters in the Request Parameters section. All parameters sent from the client are passed through by API Gateway to the back-end service.</li></ul>

## 6. Define request parameters.

Define the request parameters of your APIs. You can specify different request parameters for different parameter paths. Head, Query, Body, and Path can be selected. When you configure a dynamic path parameter, you must provide a description of this dynamic parameter. Supported parameter types include String, Number, and Boolean.

- Note that the names of all parameters must be unique.
- You can use the shortcut key on the left to adjust the parameter order.
- To delete unwanted parameters, you can click the management icon in the Actions column and choose Delete from the shortcut menu.

## 7. Configure parameter validation rules.

To configure validation rules, you can click the management icon in the Actions column and choose Configure Advanced Settings from the shortcut menu.

For example, you can configure validation rules such as the length of a string and enumeration of numbers. API Gateway pre-verifies requests based on the validation rules. Requests with invalid parameters are not sent to your back-end service. This greatly reduces the workload on your back-end service.

8. Configure your back-end service and click Next.

This section defines mappings between request and response parameters, and specifies the API configurations of your back-end service. Back-end service configurations include the back-end service URL, URLsuffix, timeout period, parameter mappings, constants, and system parameters. After receiving requests, API Gateway converts the format of the requests to the format required by the back-end service based on the back-end service configuration. Then, API Gateway forwards the requests to your back-end service.



**Note:**

You can enter the following parameters: dynamic parameters in the URL suffix , header parameters, query parameters, body parameters (non-binary), constants, and system parameters. The names of these parameters must be globally unique. For example, you cannot specify a parameter of the same name in both the header and query path.

a. Configure the basic settings of the back-end service.

Parameter	Description
Backend Service Type	By default, HTTP or HTTPS service is supported. API Gateway only supports HTTP or HTTPS service. The Mock option indicates that you do not access the back-end service, but expect API Gateway to simulate responses based on the specified values. For more information, see the Mock option.
VPC ID	Do Not Authorize Access to VPCs is selected by default. The back-end service can be connected with API Gateway directly. If the back-end service is deployed in a VPC, and API Gateway needs to access the back-end service, select the corresponding VPC ID for the back-end service.

Parameter	Description
Backend Service URL	<p>The host name of the back-end service can be a domain name or <code>http(s)://host:port</code>.</p> <p>If the back-end service is deployed in a VPC, the back-end service URL can be in the <code>http(s)://{ip}.{vpcId}.gateway.vpc:{port}</code> format. Example: <code>http://172.10.1.3.vpc-12ssar3e123.gateway.vpc:8080</code>.</p>
URL Suffix	The actual request path of your API service on your back-end server. If you want to receive dynamic parameters in the back-end path, you must declare parameter mappings by specifying the locations and names of the corresponding request parameters.
HTTP Method	PUT, GET, POST, PATCH, DELETE, and HEAD can be selected.
Timeout	The response time for API Gateway to access the back-end service after API Gateway receives an API request. The response time is from the time when API Gateway sends an API request to the back-end service to the time when API Gateway receives responses returned by the back-end service. The response time cannot exceed 30 seconds. If API Gateway does not receive a response from the back-end service within 30 seconds, API Gateway stops accessing the back-end service and returns an error message.

**b. Configure back-end service parameters.**

API Gateway can set up mappings between request and response parameters, including name mappings and parameter path mappings. API Gateway can map a request parameter at any location such as path, header, query, or body to a response parameter at a different location. In this way, you can package your back-end services into standard APIs. This section describes the mappings between front-end APIs and back-end APIs.



**Note:**

The request and response parameters must be globally unique.

**c. Configure constant parameters.**

If you want API Gateway to attach the `apigateway` tag to each request that API Gateway forwards to your back-end service, you can configure this tag as a

constant. Constants are not visible to your users. After API Gateway receives requests, API Gateway automatically adds constants to the specified locations and then forwards the requests to your back-end service.

d. Configure system parameters.

By default, API Gateway does not send its system parameters to you. However, if you need the system parameters, you can configure their locations and names. The following table lists the system parameters.

Parameter	Description
CaClientIp	The IP address of the client sending a request.
CaDomain	The domain name from which a request originates.
CaRequestHandleTime	The request time. It must be in GMT.
CaAppId	The ID of the application sending a request.
CaRequestId	The unique ID of the request.
CaApiName	The name of the API.
CaHttpSchema	The protocol that is used to call the API. The protocol can be either HTTP or HTTPS.
CaProxy	The proxy (AliCloudApiGateway).

9. Define responses and click Create.

You can set Response ContentType, Success Result Example, Error Response Example, and Error Codes. API Gateway does not parse responses, but forwards them to API users.

## 2.4.4 Publish an API

After you create an API, you must publish the API to the test, pre-release, or release environment before it can be called.

- When you use a second-level or independent domain name to access an API that has been published to an environment, you must specify the environment in the request header.
- If you attempt to publish an API that already has a running version in the test or release environment, the previously running version will be overwritten by the new API. All historical versions and definitions are recorded, allowing you to roll back the API to previous versions as needed.

- **You can unpublish an API in the test or release environment. The binding or authorization of policies, keys, and applications are retained when you unpublish an API. These relationships will take effect again if the API is republished. To revoke these relationships, you must delete the API.**

#### Step 1: Test the API

**To simulate API requests, you can create an application and authorize the application to call your API.**

**You can compile code based on actual scenarios, or use the SDK samples provided by API Gateway to call the API.**

**You can publish the API to the test or release environment. If no independent domain name is bound to the group to which the API belongs, you can test or call the API by using the second-level domain name. When you make an API request, specify the environment of the API by setting the X-Ca-Stage header to TEST, PRE, or RELEASE. If you do not specify the header, the API will be published to the release environment.**

#### Step 2: Publish the API

**After testing the API, you can publish it.**

**API Gateway enables you to manage versions of APIs in the test or release environment. You can publish or unpublish an API, and switch the versions of an API. The switch of versions takes effect in real time.**

1. *Log on to the API Gateway console.*
2. **Click the API tab.**
3. **Select the API that you want to publish, click the management icon in the Actions column corresponding to the API, and choose Publish from the shortcut menu.**
4. **In the dialog box that appears, select the environment where the API is published, enter a description, and click OK.**

### 2.4.5 Authorize an application

**You must authorize an application before it can call an API. After publishing an API to the release environment, you must authorize the user applications to call the**

**API.** You can grant or revoke the authorization of an application to call APIs. API Gateway verifies the authorization relationship.



**Note:**

- You can authorize one or more applications to use one or more APIs.
- If an API is published to both the test and release environments and the test environment has been selected, applications are only authorized to call the API in the test environment.
- You can locate an application based on its ID.
- If you want to revoke the authorization of an application to call an API, go to the Authorization tab of the API. Then, select the application, click the management icon in the Actions column corresponding to the application, and choose Deauthorize from the shortcut menu, or click Deauthorize in the upper-right corner.

An application indicates the identity of a requester. Before testing or calling an API, you or your users must create an application that is used as the identity of the requester. Then, you need to authorize the application to call the API.



**Note:**

Authorizations are environment-specific. If you want to use an application to call an API in both the test and release environments, you must authorize the application in both environments. Otherwise, errors may occur due to the inconsistencies between the authorized environment and the requested environment.

## Procedure

1. [Log on to the API Gateway console.](#)
2. Click the APIs tab.
3. Select the API that an application is authorized to call, click the management icon in the Actions column corresponding to the API, and choose Authorize from the shortcut menu. The Authorize Applications dialog box appears.



4. Select the environment and the application to authorize.

The system automatically displays the applications that belong to your account.

If you want to authorize an application that belongs to a different account, search for the application by application ID.

5. Select the application to authorize, and click the > icon to add the selected application to the right-side list.

6. Click OK to complete the authorization process.

## 2.5 Call an API

### 2.5.1 Manage applications

#### 2.5.1.1 Create an application

Applications are the identities that you use to call APIs. You can own multiple applications. Your applications can be authorized to call different APIs based on your business requirements. Applications instead of user accounts are authorized to call APIs. In the API Gateway console, you can create, modify, or delete applications, view details of applications, manage key pairs, and view callable APIs of applications.

Each application has an AppKey and an AppSecret. You can regard them as an account and a password. When you call an API, you must pass in the AppKey as a parameter. AppSecret is used to calculate the signature string. API Gateway authenticates the key pair to verify your identity. An application must be authorized to call an API. Both authorization and authentication are intended for applications.

You can log on to the API Gateway console to create applications on the Applications tab.

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the Applications tab to go to the Applications tab.
3. Click Create Application.

#### 4. Specify parameters and click Create.

The application name must be globally unique. It must be 4 to 26 characters in length and can contain letters, digits, and underscores (\_). It must start with a letter.

After you create the application, the system automatically assigns an AppKey and an AppSecret to it. You must use the AppSecret to calculate a signature string. When calling an API, you must include the signature string in the request. API Gateway verifies your identity based on the signature string.

On the Applications tab page, click the application ID to go to the application details page. The AppKey and AppSecret are displayed on the application details page. You can reset the AppSecret as needed.

### 2.5.1.2 View application details

You can view the details of existing applications.

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click Applications tab to go to the Applications tab.
3. Click the application that you want to view.

You can view the basic information, AppKey, and callable APIs of the application.

### 2.5.1.3 Modify an application

You can modify existing applications.

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the Applications tab to go to the Applications tab.
3. Select the application you want to modify, click the management icon in the Actions column corresponding to the application, and choose Change from the shortcut menu.
4. Modify the application information and click OK.

### 2.5.1.4 Delete an application

You can delete existing applications.

#### Procedure

1. [Log on to the API Gateway console.](#)

2. Click the Applications tab to go to the Applications tab.
3. Select the application you want to delete, click the management icon in the Actions column corresponding to the application, and choose Delete from the shortcut menu.
4. In the dialog box that appears, click OK.

## 2.5.2 View existing APIs

You can view existing APIs in the API Gateway console.

### Procedure

1. [Log on to the API Gateway console](#).
2. Click the APIs tab.

## 2.5.3 Authorize an application

Authorization is the process of authorizing an application to call an API. Your applications must be authorized before they can call APIs.

You must provide your application IDs to the API provider for authorization. After authorization, you can view the APIs that your applications have been authorized to call in the API Gateway console.

The APIs that your applications have been authorized to call are displayed in the Callable APIs section on the application details page.

After the API provider authorizes your applications to call APIs, you do not need to and cannot authorize your applications.

## 2.5.4 Encrypt a signature

When you call an API, you must construct a signature string and add the calculated signature string to the request header. API Gateway uses symmetric encryption to verify the identity of the request sender.

- Add the calculated signature string to the request header.
- Organize the request parameters into a string-to-sign based on [Request signatures](#). Then, use the algorithm provided in the SDK sample to calculate the signature. The result is the calculated signature string.
- Both HTTP and HTTPS requests must be signed.

For more information about how to construct a string-to-sign, see [Request signatures](#).

Replace the AppKey and AppSecret in the SDK sample with your own AppKey and

**AppSecret. Then, construct a string-to-sign based on Request signatures. After creating the string-to-sign, you can use it to initiate a request.**

## 2.5.5 Request signatures

### Endpoint

- **Each API belongs to an API group, and each API group has a unique endpoint. An endpoint is an independent domain name that is bound to an API group by the API provider. API Gateway uses endpoints to locate API groups.**
- **An endpoint must be in the `www.[ Independent domain name].com/[Path]?[HTTPMethod]` format.**
- **API Gateway locates a unique API group by endpoint, and locates a unique API in the group through the combination of Path and HTTPMethod.**
- **After you purchase an API, you can obtain the API documentation from the Purchased APIs list in the API Gateway console. If you have not purchased an API, you must obtain authorization from the API provider for your applications to call the API. After authorization, you can obtain the API documentation from the Callable APIs list on the application details page.**

### System-level header parameters

- **(Required) X-Ca-Key: AppKey.**
- **(Required) X-Ca-Signature: the signature string.**
- **(Optional) X-Ca-Timestamp: the timestamp passed in by the API caller. This value is a UNIX timestamp representing the number of milliseconds that have elapsed since January 1, 1970 00:00:00 UTC. The timestamp is valid for 15 minutes by default.**
- **(Optional) X-Ca-Nonce: the UUID generated by the API caller. To prevent replay attacks, you must specify both the X-Ca-Nonce header and the X-Ca-Timestamp header.**
- **(Optional) Content-MD5: When the request body is not a form, you can calculate the MD5 value of the request body. Then, you can send the value to API Gateway for MD5 verification.**
- **(Optional) X-Ca-Stage: the stage of the API. Valid values: TEST, PRE, and RELEASE. Default value: RELEASE. If the API that you intend to call has not been published to the release environment, you must specify the value of this parameter. Otherwise, a URL error will be reported.**

## Signature validation

### Construct the signature calculation strings

```
String stringToSign=
HTTPMethod + "\n" +
Accept + "\n" +           // We recommend that you specify the
    Accept header in the request. If the request header is not set,
    some HTTP clients will use the default value */*, causing signature
    verification to fail.
Content-MD5 + "\n"
Content-Type + "\n" +
Date + "\n" +
Headers +
Url
```

An HTTP method must be uppercase, such as POST.

If Accept, Content-MD5, Content-Type, and Date are empty, add a line break `\n` after each of them . If Headers is empty, `\n` is not required.

### Content-MD5

Content-MD5 indicates the MD5 value of the request body. The value is calculated as follows:

```
String content-MD5 = Base64.encodeBase64(MD5(bodyStream.getBytes("UTF-8")));
```

`bodyStream` indicates a byte array.

### Headers

Headers indicates the string constructed by the keys and values of the header parameters that are used for Headers signature calculation. We recommend that you use the parameters starting with X-Ca and custom header parameters for signature calculation.



#### Notice:

The following parameters are not used for Headers signature calculation: X-Ca-Signature, X-Ca-Signature-Headers, Accept, Content-MD5, Content-Type, and Date.

### Headers construction method:

Sort the header keys used for Headers signature calculation in alphabetical order. Construct the string based on the following rules: If the value of a header

**parameter is empty, use `HeaderKey + ":" + "\n"` for signature calculation. The key and colon (:) must be retained.**

```
String headers =
HeaderKey1 + ":" + HeaderValue1 + "\n"+
HeaderKey2 + ":" + HeaderValue2 + "\n"+
...
HeaderKeyN + ":" + HeaderValueN + "\n"
```

**The keys of the header parameters used for Headers signature calculation must be separated with commas (,), and placed in the request headers. The key is X-Ca-Signature-Headers.**

## Url

**Url indicates the Form parameter in `Path + Query + Body`. For `Query + Form`, sort keys specified by Key in alphabetical order and construct the string based on the following rules: If `Query` or `Form` is empty, no question marks `?` are required for `Url` = `Path`. If `Value` of a parameter is empty, only `Key` is used for signature calculation and an equal sign (=) is not required.**

```
String url =
Path +
"?" +
Key1 + "=" + Value1 +
"&" + Key2 + "=" + Value2 +
...
"&" + KeyN + "=" + ValueN
```



### Notice:

**Note: The `Query` parameter or the `Form` parameter may have multiple values specified by `Value`. If both parameters have multiple values, only the first value of each parameter is used for signature calculation.**

## Signature calculation

```
Mac hmacSha256 = Mac.getInstance("HmacSHA256");
byte[] keyBytes = secret.getBytes("UTF-8");
hmacSha256.init(new SecretKeySpec(keyBytes, 0, keyBytes.length, "
HmacSHA256"));
String sign = new String(Base64.encodeBase64(Sha256.doFinal(stringToSi
gn.getBytes("UTF-8")), "UTF-8"));
```

**secret indicates the AppSecret.**

## Signature passing

**Add the calculated signature to the request header. The key is X-Ca-Signature.**

## Signature troubleshooting

If signature verification fails, API Gateway places the returned `stringToSign` value in the HTTP response header and sends the response to the client. The key is `X-Ca-Error-Message`. Compare the `stringToSign` value calculated by the client with the one returned by the server.

If the `stringToSign` values from the client and server are the same, check the `AppSecret` used for signature calculation.

HTTP headers do not support line breaks. Line breaks in `stringToSign` values are filtered out. Ignore the line breaks when you make a comparison.

## Signature demo

For more information about the Java demo of signature calculation, see <https://github.com/aliyun/api-gateway-demo-sign-java>.

## 2.5.6 API call examples

You can edit an HTTP or HTTPS request to call an API. Before calling the API, you can use API call examples of multiple programming languages in the API Gateway console to test the call.

### Part 1: Request

#### Request URL

```
http://e710888d3ccb4638a723ff8d03837095-cn-qingdao.aliapi.com/demo/  
post
```

#### Request method

```
POST
```

#### Request body

```
FormParam1=FormParamValue1&FormParam2=FormParamValue2  
//HTTP request body
```

#### Request header

```
Host: e710888d3ccb4638a723ff8d03837095-cn-qingdao.aliapi.com  
Date: Mon, 22 Aug 2016 11:21:04 GMT  
User-Agent: Apache-HttpClient/4.1.2 (java 1.6)  
Content-Type: application/x-www-form-urlencoded; charset=UTF-8  
// The request body type. Set the request body type based on the  
actual request you want to make.  
Accept: application/json
```

```
// The response body type. Some APIs can return data in the correspond-
ing format based on the specified response type. We recommend that you
manually set this request header. If the request header is not set,
some HTTP clients will use the default value */*, causing a signature
error.
X-Ca-Request-Mode: debug
// Specifies whether to enable the Debug mode. This parameter is not
case-sensitive. If it is not specified, the Debug mode is disabled by
default. This mode is typically enabled in the API debugging phase.
X-Ca-Version: 1
// The API version number. Set the value to 1. Default value: 1.
X-Ca-Signature-Headers: X-Ca-Request-Mode,X-Ca-Version,X-Ca-Stage,X-Ca-
-Key,X-Ca-Timestamp
// The custom request headers involved in signature calculation. The
server will read the request headers based on this configuration to
sign the request. This configuration does not include the Content-
Type, Accept, Content-MD5, and Date request headers which are already
included in the basic signature structure. For more information about
the signature, see Request signatures.
X-Ca-Stage: RELEASE
// The stage of the API. Valid values: TEST, PRE, and RELEASE. This
parameter is not case-sensitive. The API provider can select the stage
to which the API is published. The API can be called only after it is
published to the specified stage. Otherwise, the system will prompt
that the API cannot be found or that the request URL is invalid.
X-Ca-Key: 60022326
// The AppKey of the request. You must obtain the AppKey in the API
Gateway console. Applications can only call APIs after they have been
authorized.
X-Ca-Timestamp: 1471864864235
// The request timestamp. This value is a UNIX timestamp representing
the number of milliseconds that have elapsed since January 1, 1970 00:
00:00 UTC. By default, the timestamp is valid for 15 minutes.
X-Ca-Nonce:b931bc77-645a-4299-b24b-f3669be577ac
// The unique ID of the request. AppKey, API, and Nonce must be unique
within the last 15 minutes. To prevent replay attacks, you must
specify both the X-Ca-Nonce header and the X-Ca-Timestamp header.
X-Ca-Signature: FJleSrCYPGCU7dMLTG+UD3Bc5Elh3TV3CWHtSKh1Ys=
// The request signature.
CustomHeader: CustomHeaderValue
// The custom request headers. CustomHeaderValue is used as an example
. You can set multiple custom request headers in actual requests based
on the definition of the API being called.
```

## Part 2: Response

### Status code

```
400
// The status code of the response. If the value is greater than or
equal to 200 and less than 300, the call is successful. If the value
is greater than or equal to 400 and less than 500, a client-side error
has occurred. If the value is greater than 500, a server-side error
has occurred.
```

### Response header

```
X-Ca-Request-Id: 7AD052CB-EE8B-4DFD-BBAF-EFB340E0A5AF
// The unique ID of the request. When API Gateway receives a request,
it generates a request ID and returns the request ID to the client in
the X-Ca-Request-Id header. We recommend that you record the request
```



```
ID in both the client and backend server for troubleshooting and tracking.  
X-Ca-Error-Message: Invalid Url  
// The error message returned by API Gateway. When a request fails, API Gateway returns the error message to the client in the X-Ca-Error-Message header.  
X-Ca-Debug-Info: {"ServiceLatency":0,"TotalLatency":2}  
// The message can be returned only when the Debug mode is enabled. The message is used only for reference at the debugging stage.
```

Regardless of whether you call an API by using HTTP or HTTPS, the request must include the signature information. For more information about how to calculate and pass the encrypted signature, see [Request signatures](#).

## 2.6 APIs

### 2.6.1 Manage groups

#### 2.6.1.1 Create a group

You can create an API Group in the API Gateway console.

##### Procedure

1. [Log on to the API Gateway console](#).
2. Click the Groups tab.
3. On the Groups tab, click Create Group.
4. In the Create Group dialog box that appears, set parameters and click Create.

The name of a group must be globally unique. The name must be 4 to 50 characters in length and can contain letters, digits, and underscores (\_). It must start with a letter.

#### 2.6.1.2 Manage domain names

In Apsara Stack, you can use the second-level domain name of a group to directly call an API, or bind your domain name to the group so that you can use your domain name to access APIs within the group.

##### Context

If you want to use your domain name to directly call APIs within a group, you must not only bind the domain name to the group, but also add a DNS record to your domain name. The domain name must be resolved to the second-level domain name of the group or the IP address corresponding to the second-level domain name.

Bind an independent domain name

1. *Log on to the API Gateway console.*
2. **Click the Groups tab.**
3. **Click the management icon in the Actions column corresponding to a group and choose View Details from the shortcut menu.**
4. **Click the Basic Information tab.**
5. **Click Bind Domain Name. In the Bind Domain Name dialog box that appears, enter a domain name and click OK.**

Delete a domain name

1. *Log on to the API Gateway console.*
2. **Click the Groups tab.**
3. **Click the management icon in the Actions column corresponding to a group and choose View Details from the shortcut menu.**
4. **Click the Basic Information tab.**
5. **Click the management icon in the Actions column corresponding to the domain name to be deleted and choose Delete Domain Name from the shortcut menu.**

### 2.6.1.3 Manage certificates

To use HTTPS on an independent domain, you must upload an SSL certificate.

#### Context

To make HTTPS API calls, you must use a domain that supports HTTPS and set Network Protocol to HTTPS in the Request Basic Settings section when defining API requests.

Upload a certificate

1. *Log on to the API Gateway console.*
2. **Click the Groups tab.**
3. **Click the management icon in the Actions column corresponding to a group and choose View Details from the shortcut menu.**
4. **Click the Basic Information tab.**
5. **Click + Add in the SSL Certificate column corresponding to a domain name. In the Create Certificate dialog box that appears, set Certificate Name, Certificate Contents, and Private Key, and click OK.**

#### Edit a certificate

1. [Log on to the API Gateway console.](#)
2. Click the **Groups** tab.
3. Click the management icon in the **Actions** column corresponding to a group and choose **View Details** from the shortcut menu.
4. Click the **Basic Information** tab.
5. Click the management icon in the **Actions** column corresponding to a certificate and choose **Edit Certificate** from the shortcut menu. In the **Edit Certificate** dialog box that appears, set **Certificate Name**, **Certificate Contents**, and **Private Key**, and click **OK**.

#### Delete a certificate

1. [Log on to the API Gateway console.](#)
2. Click the **Groups** tab.
3. Click the management icon in the **Actions** column corresponding to a group and choose **View Details** from the shortcut menu.
4. Click the **Basic Information** tab.
5. Click the management icon in the **Actions** column corresponding to a certificate and choose **Delete Certificate** from the shortcut menu. In the message that appears, click **OK**.

### 2.6.1.4 Delete a group

You can delete an existing group.

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the **Groups** tab.
3. Select the group to be deleted, click the management icon in the **Actions** column corresponding to the group, and choose **Delete** from the shortcut menu.
4. In the dialog box that appears, click **OK**.



#### Note:

Before deleting a group, you must delete APIs that belong to the group.

### 2.6.1.5 Manage environments

To understand environment management, you must understand two concepts: environment and environment variable.

- An environment is an API group configuration. You can configure several environments for a group. APIs that have not been published are considered as defined APIs. An API can only provide external services after it has been published to an environment.
- Environment variables are environment-specific variables that you can create and manage. For example, you can create an environment variable named `Path` and valued `/stage/release` for the release environment.

On the API Settings tab, set the `Path` parameter to `#Path#`, which indicates a variable named `Path`.

When you publish the API to the release environment, the value of the `#Path#` variable on the API Settings tab is `/stage/release`.

When you publish the API to another environment that does not have the `#Path#` variable, the variable will be null and the API cannot be called.

Environment variables enable backend services to be in different runtime environments. You can access various backend services by configuring the same API definition but different backend service endpoints and paths across different runtime environments. When you use environment variables, note the following limits:

- Variable names are case-sensitive.
- If you configure a variable in the API definition, you must configure the name and value of the variable for the environment to which the API is published. Otherwise, the variable will be null and the API cannot be called.

Create an environment variable

1. [Log on to the API Gateway console.](#)
2. Click the Groups tab.
3. Click the management icon in the Actions column corresponding to a group and choose View Details from the shortcut menu.
4. Click the Environment Variables tab.

5. Click **Create Variable**. In the **Create Environment Variable** dialog box that appears, set **Variable Name** and **Variable Value**, and click **OK**.



**Notice:**

The variable names for the release, pre-release, and test environments must be the same. However, the variable values for the three environments can be different. When an API is published to the corresponding environment, the variable value will be replaced.

Delete an environment variable

1. *Log on to the API Gateway console.*
2. Click the **Groups** tab.
3. Click the management icon in the **Actions** column corresponding to a group and choose **View Details** from the shortcut menu.
4. Click the **Environment Variables** tab.
5. Select the runtime environment and the variable to be deleted, click the management icon in the **Actions** column corresponding to the variable, and choose **Delete Variable** from the shortcut menu.
6. In the message that appears, click **OK**.

## 2.6.2 Create an API

### 2.6.2.1 Overview

Creating an API is the process of defining the API in the API Gateway console.

When creating an API, you must define the basic information, back-end service information, API request information, and response information of the API.

- API Gateway enables you to configure verification rules for input parameters . API Gateway can be configured to pre-verify and forward API requests that contain valid parameters.
- API Gateway enables you to configure mappings between front-end and back-end parameters. API Gateway can map a front-end parameter at one location to a back-end parameter at a different location. For example, you can configure API Gateway to map a **Query** parameter in an API request to a **Header** parameter in a back-end service request. In this way, you can encapsulate your back-end services into standard API operations.

- API Gateway enables you to configure constant and system parameters. These parameters are not visible to your users. API Gateway can add these parameters to requests based on your business requirements before sending the requests to your back-end services. If you want API Gateway to attach the keyword `apigateway` to each request that API Gateway forwards to your back-end services, you can configure `aligateway` as a constant parameter and specify where it is received.

## 2.6.2.2 Create an API

### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the APIs tab.
3. Click Create API.
4. Specify the basic information of the API and click Next.

Parameter	Description
Groups	The basic management unit of an API. You must create a group before creating an API. When you select a group, you select a region for the API.
API Name	The name of the API to be created.
Authentication Mode	<p>The authentication mode of API requests. Valid values: Alibaba Cloud Applications and None.</p> <ul style="list-style-type: none"><li>• Alibaba Cloud Applications: This mode requires the requester to pass the application authentication to call the API.</li><li>• None: This mode allows any user who knows the request definition of the API to initiate a request. API Gateway directly forwards the requests to your back-end service without verifying the identity of the requesters.</li></ul>
Signature Algorithm	<p>The algorithm used to sign API requests. Valid values:</p> <ul style="list-style-type: none"><li>• HmacSHA256</li><li>• HmacSHA1 and HmacSHA256</li></ul>
Description	The description of the API.

5. Define API requests. Define how users call your APIs, including the request type, network protocol, URL suffix, HTTP method, and request mode.

Parameter	Description
Request Type	<p>The request type. Four options are available. However, only the common request type is supported.</p> <ul style="list-style-type: none"> <li>• Common Request: indicates common HTTP or HTTPS requests.</li> <li>• Logon Request (Bidirectional Communication): indicates the bidirectional control signal to register devices. It is sent from the client to the server.</li> <li>• Logoff Request (Bidirectional Communication): indicates the bidirectional control signal to register devices. It is sent from the client to the server. After devices are deregistered, server-to-client notifications are no longer received.</li> <li>• Server-to-Client Notification (Bidirectional Communication): After receiving the registration signal sent from the client, the back-end service records the device ID and send a server-to-client notification to API Gateway. Then, API Gateway sends the notification to the device. As long as the device is online, API Gateway sends the server-to-client notification to the device.</li> </ul>
Network protocol	HTTP, HTTPS, and WEBSOCKET are supported in API calls.
URL Suffix	The API request path. It corresponds to the service host . The request path can be different from the actual back-end service path. You can specify any valid and semantically-correct path as the request path. You can configure dynamic parameters in the request path, which requires users to specify path parameters in the request. At the same time, the path parameters can be mapped to query and header parameters that are received by the back-end service.
HTTP Method	You can select PUT, GET, POST, PATCH, DELETE, and HEAD.

Parameter	Description
<b>Request Mode</b>	<p>You can select either Request Parameter Mapping or Request Parameter Passthrough.</p> <ul style="list-style-type: none"><li>• Request Parameter Mapping indicates that you must configure request and response data mappings for query , path, and body form parameters. API Gateway only passes the configured parameters through to the back end. Other parameters are filtered out.</li><li>• Request Parameter Passthrough indicates that you do not need to configure query and body form parameters, but still must configure path parameters in the Request Parameters section. All parameters sent from the client are passed through by API Gateway to the back-end service.</li></ul>

#### 6. Define request parameters.

Define the request parameters of your APIs. You can specify different request parameters for different parameter paths. Head, Query, Body, and Path can be selected. When you configure a dynamic path parameter, you must provide a description of this dynamic parameter. Supported parameter types include String, Number, and Boolean.

- Note that the names of all parameters must be unique.
- You can use the shortcut key on the left to adjust the parameter order.
- To delete unwanted parameters, you can click the management icon in the Actions column and choose Delete from the shortcut menu.

#### 7. Configure parameter validation rules.

To configure validation rules, you can click the management icon in the Actions column and choose Configure Advanced Settings from the shortcut menu. For example, you can configure validation rules such as the length of a string and enumeration of numbers. API Gateway pre-verifies requests based on the validation rules. Requests with invalid parameters are not sent to your back-end service. This greatly reduces the workload on your back-end service.

#### 8. Configure your back-end service and click Next.

This section defines mappings between request and response parameters, and specifies the API configurations of your back-end service. Back-end service configurations include the back-end service URL, URLsuffix, timeout period,



parameter mappings, constants, and system parameters. After receiving requests, API Gateway converts the format of the requests to the format required by the back-end service based on the back-end service configuration. Then, API Gateway forwards the requests to your back-end service.

**Note:**

You can enter the following parameters: dynamic parameters in the URL suffix, header parameters, query parameters, body parameters (non-binary), constants, and system parameters. The names of these parameters must be globally unique. For example, you cannot specify a parameter of the same name in both the header and query path.

**a. Configure the basic settings of the back-end service.**

Parameter	Description
Backend Service Type	By default, HTTP or HTTPS service is supported. API Gateway only supports HTTP or HTTPS service. The Mock option indicates that you do not access the back-end service, but expect API Gateway to simulate responses based on the specified values. For more information, see the Mock option.
VPC ID	Do Not Authorize Access to VPCs is selected by default. The back-end service can be connected with API Gateway directly. If the back-end service is deployed in a VPC, and API Gateway needs to access the back-end service, select the corresponding VPC ID for the back-end service.
Backend Service URL	The host name of the back-end service can be a domain name or <code>http(s)://host:port</code> .  If the back-end service is deployed in a VPC, the back-end service URL can be in the <code>http(s)://{ip}.{vpcId}.gateway.vpc:{port}</code> format. Example: <code>http://172.10.1.3.vpc-12ssar3e123.gateway.vpc:8080</code> .
URL Suffix	The actual request path of your API service on your back-end server. If you want to receive dynamic parameters in the back-end path, you must declare parameter mappings by specifying the locations and names of the corresponding request parameters.
HTTP Method	PUT, GET, POST, PATCH, DELETE, and HEAD can be selected.

Parameter	Description
Timeout	The response time for API Gateway to access the back-end service after API Gateway receives an API request. The response time is from the time when API Gateway sends an API request to the back-end service to the time when API Gateway receives responses returned by the back-end service. The response time cannot exceed 30 seconds. If API Gateway does not receive a response from the back-end service within 30 seconds, API Gateway stops accessing the back-end service and returns an error message.

**b. Configure back-end service parameters.**

API Gateway can set up mappings between request and response parameters, including name mappings and parameter path mappings. API Gateway can map a request parameter at any location such as path, header, query, or body to a response parameter at a different location. In this way, you can package your back-end services into standard APIs. This section describes the mappings between front-end APIs and back-end APIs.



**Note:**

The request and response parameters must be globally unique.

**c. Configure constant parameters.**

If you want API Gateway to attach the `apigateway` tag to each request that API Gateway forwards to your back-end service, you can configure this tag as a constant. Constants are not visible to your users. After API Gateway receives requests, API Gateway automatically adds constants to the specified locations and then forwards the requests to your back-end service.

**d. Configure system parameters.**

By default, API Gateway does not send its system parameters to you. However, if you need the system parameters, you can configure their locations and names. The following table lists the system parameters.

Parameter	Description
CaClientIp	The IP address of the client sending a request.
CaDomain	The domain name from which a request originates.

Parameter	Description
CaRequestHandleTime	The request time. It must be in GMT.
CaAppId	The ID of the application sending a request.
CaRequestId	The unique ID of the request.
CaApiName	The name of the API.
CaHttpSchema	The protocol that is used to call the API. The protocol can be either HTTP or HTTPS.
CaProxy	The proxy (AliCloudApiGateway).

#### 9. Define responses and click Create.

You can set Response ContentType, Success Result Example, Error Response Example, and Error Codes. API Gateway does not parse responses, but forwards them to API users.

### 2.6.2.3 Security authentication

The security authentication methods that are supported by API Gateway include Alibaba Cloud applications and none.

- **Alibaba cloud applications:** An application must be authorized by the API provider to call an API. An API caller must provide an AppKey and encrypted signature. Otherwise, the API request validation will fail. For more information about the signature method, see [Encrypt a signature](#).
- **None:** The API can be called without authorization after it is published. The AppKey and encrypted signature are not required when you make an API request.

### 2.6.2.4 Network protocol

HTTPS domain names that are configured in the API Gateway console are invalid because the console does not support HTTPS. To use an HTTPS domain name, you can call the API operations of API Gateway.

Find the API on the API tab, click the management icon in the Actions column corresponding to the API, and choose Change from the shortcut menu. In the Request Basic Settings section of the API Settings page, change the network protocol.

You can send API requests by using any of the following network protocols:

- **HTTP:** You can send API requests over HTTP.
- **HTTPS:** You can send API requests over HTTPS. HTTPS cannot be configured in the API Gateway console and will be supported in later versions.
- **WEBSOCKET:** You can send API requests over WebSocket.

### 2.6.2.5 Request body configuration

You can configure the request body when the HTTP method is POST, PUT, or PATCH. You can use two methods to configure the request body. The following two methods are mutually exclusive.

- **Form-based request body:** Add a request parameter and select Body as the parameter path. The configured request body can only be used to transmit form data. Form data occupies space in the URL request. The URL request cannot exceed 258 KB in size.
- **Non-form-based request body:** If the body content to be transmitted is in JSON or XML format, select "The body content description should be a non-form data. For example, JSON string, binary files, and others". The size of the request body cannot exceed 8 MB.



**Note:**

API Gateway cannot upload files through HTTP multipart requests. To upload a file, you can convert the file into a Base64 string and add the string to the request body.

### 2.6.2.6 VPC ID

If the back-end service is deployed in a classic network, select Do Not Authorize Access to VPCs.

If the back-end service is deployed in a VPC, select a VPC ID. After selecting the VPC ID, you must enter the back-end service URL in the format of `http://ip:port`. `ip` indicates the IP address of the back-end service.

### 2.6.2.7 Configure an API in mock mode

Typically, business partners work together to develop a project. However, the interdependence among business partners restricts them in the project development process. Misunderstandings also can affect the development progress or even delay the project. Therefore, the mock mode is used to simulate the

**predetermined API responses in the project development process. This can help reduce misunderstanding and improve development efficiency.**

**API Gateway provides a simplified configuration process of an API in mock mode.**

Configure a mock API

**Click the APIs tab. Select an API and click Change in the Actions column to go to the Change API page.**

**On the Change API page, choose Define Backend Service to set the API to mock mode.**

- 1. Set the back-end service type to Mock.**
- 2. Enter a mock response.**

**Enter your responses as the mock response body. Responses can be in JSON, XML, and text formats. Example:**

```
{
  "result": {
    "title": " Mock test for API Gateway",
  }
}
```

**Save the mock settings. Publish the API to the test or release environment for testing.**

- 3. Enter an HTTP status code for the mock response. Enter 200 to indicate a successful API request.**

Take an API out of mock mode

**To take an API out of mock mode, you can change the back-end service type from Mock to HTTP or HTTPS service. This change takes effect only after the API is published.**

### 2.6.2.8 Return the Content-Type header

The value of the Content-Type header is only used to generate API documentation. It does not affect responses returned by the back-end service. The Content-Type header is returned by the back-end service.

## 2.6.3 API management

### 2.6.3.1 View and modify an API

You can view and modify an API as needed.



**Note:**

If you modify an API that has been published, modifications made to an API that has been published will not affect its operations. You must re-publish the modified API before synchronizing the changes to the release environment.

#### Procedure

1. [Log on to the API Gateway console](#).
2. Click the APIs tab.
3. Select the API that you want to view and click the Management icon in the Actions column corresponding to the API.
  - Choose View Details from the shortcut menu. On the API Settings page, you can view the information of the API.
  - Choose Change from the shortcut menu. Modify the API as needed and click Change in the lower right corner.

The procedure to create an API is similar to that of modifying an API. For more information about how to create an API, see [Create an API](#). If you want to cancel the modifications before they are submitted, click the Back icon in the upper-left corner of the Change API page.

### 2.6.3.2 Publish an API

After you create an API, you must publish the API to the test, pre-release, or release environment before it can be called.

- When you use a second-level or independent domain name to access an API that has been published to an environment, you must specify the environment in the request header.

- If you attempt to publish an API that already has a running version in the test or release environment, the previously running version will be overwritten by the new API. All historical versions and definitions are recorded, allowing you to roll back the API to previous versions as needed.
- You can unpublish an API in the test or release environment. The binding or authorization of policies, keys, and applications are retained when you unpublish an API. These relationships will take effect again if the API is republished. To revoke these relationships, you must delete the API.

#### Step 1: Test the API

To simulate API requests, you can create an application and authorize the application to call your API.

You can compile code based on actual scenarios, or use the SDK samples provided by API Gateway to call the API.

You can publish the API to the test or release environment. If no independent domain name is bound to the group to which the API belongs, you can test or call the API by using the second-level domain name. When you make an API request, specify the environment of the API by setting the X-Ca-Stage header to TEST, PRE, or RELEASE. If you do not specify the header, the API will be published to the release environment.

#### Step 2: Publish the API

After testing the API, you can publish it.

API Gateway enables you to manage versions of APIs in the test or release environment. You can publish or unpublish an API, and switch the versions of an API. The switch of versions takes effect in real time.

1. [Log on to the API Gateway console](#).
2. Click the API tab.
3. Select the API that you want to publish, click the management icon in the Actions column corresponding to the API, and choose Publish from the shortcut menu.
4. In the dialog box that appears, select the environment where the API is published, enter a description, and click OK.

### 2.6.3.3 Authorize an application

You must authorize an application before it can call an API. After publishing an API to the release environment, you must authorize the user applications to call the API. You can grant or revoke the authorization of an application to call APIs. API Gateway verifies the authorization relationship.

**Note:**

- You can authorize one or more applications to use one or more APIs.
- If an API is published to both the test and release environments and the test environment has been selected, applications are only authorized to call the API in the test environment.
- You can locate an application based on its ID.
- If you want to revoke the authorization of an application to call an API, go to the Authorization tab of the API. Then, select the application, click the management icon in the Actions column corresponding to the application, and choose Deauthorize from the shortcut menu, or click Deauthorize in the upper-right corner.

An application indicates the identity of a requester. Before testing or calling an API, you or your users must create an application that is used as the identity of the requester. Then, you need to authorize the application to call the API.

**Note:**

Authorizations are environment-specific. If you want to use an application to call an API in both the test and release environments, you must authorize the application in both environments. Otherwise, errors may occur due to the inconsistencies between the authorized environment and the requested environment.

#### Procedure

1. *Log on to the API Gateway console.*
2. Click the APIs tab.
3. Select the API that an application is authorized to call, click the management icon in the Actions column corresponding to the API, and choose Authorize from the shortcut menu. The Authorize Applications dialog box appears.



4. Select the environment and the application to authorize.

The system automatically displays the applications that belong to your account.

If you want to authorize an application that belongs to a different account, search for the application by application ID.

5. Select the application to authorize, and click the > icon to add the selected application to the right-side list.

6. Click OK to complete the authorization process.

### 2.6.3.4 Revoke authorization

You can revoke the authorization of an application to call an API.



**Note:**

Before deleting a published API, you must unpublish it. Otherwise, you cannot delete this API.

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the APIs tab.
3. Click the management icon in the Actions column corresponding to an API, and choose View Details from the shortcut menu to go to the API Settings page.
4. Click the Authorization tab to view all the applications that are authorized to call the API.
5. Select an authorized application, click the management icon in the Actions column corresponding to the application, and choose Deauthorize from the shortcut menu.
6. In the dialog box that appears, click OK.

### 2.6.3.5 Unpublish an API

You can unpublish APIs that have been published.

You can unpublish an API in the test or release environment. The binding or authorization of policies, keys, and applications are retained when you unpublish an API. These relationships will take effect again if the API is republished. To revoke these relationships, you must delete the API. For more information, see

[Delete API definitions.](#)

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the API tab.
3. Select the API that you want to unpublish, click the management icon in the Actions column corresponding to the API, and choose Take Offline from the shortcut menu.
4. In the dialog box that appears, select the environment that you want to unpublish the API from, and click OK.

### 2.6.3.6 View the version history of an API

You can view the version history of an API, including the version number, description, environment, release time, and specific definition of each version.

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the APIs tab.
3. Click the management icon in the Actions column corresponding to an API, and choose View Details from the shortcut menu to go to the API Settings page.
4. Click the Version History tab.
5. Select the version that you want to view the history of, click the management icon in the Actions column corresponding to the version, and choose View from the shortcut menu. You can see the details of this API version.

### 2.6.3.7 Change the version of an API

When viewing the version history of an API, you can select a different version and switch to this version. The selected version then replaces the previous version and takes effect in the specified environment.

#### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the APIs tab.
3. Click the management icon in the Actions column corresponding to an API, and choose View Details from the shortcut menu to go to the API Settings page.
4. Click the Version History tab.
5. Select the version, click the management icon in the Actions column corresponding to the version, and choose Switch to This Version from the shortcut menu.

6. In the dialog box that appears, enter a description and click OK.

## 2.6.4 Plugin management

### 2.6.4.1 Create a plugin

#### 2.6.4.1.1 Create an IP-based access control plugin

IP-based access control can help API providers configure an IP whitelist or blacklist for API access. The following steps describe how to create an IP-based access control plugin.

#### Procedure

1. [Log on to the API Gateway console](#).
2. Click the plugins tab.
3. Click Create Plugin. On the Create Plugin page, set Type to IP-based Access Control.

Parameter	Description
Action	<ul style="list-style-type: none"><li>• <b>Allow:</b> Only requests that comply with the IP-based access control policy are allowed. All other requests are rejected.</li><li>• <b>Reject:</b> Requests that comply with the IP-based access control policy are rejected. All other requests are allowed.</li></ul>

Parameter	Description
AppId	The ID of the application making requests. This parameter is optional. If this value is null, this access control policy is applied to all applications . If this value is not null, this access control policy is only applied to the specified application. When Action is set to Reject, requests from the specified application are ignored.
IP Address	Required. The IP address of the API request. Note : The IP address used to access API Gateway is an egress IP address. The format can be an IP address or an IP address segment. Example: 10.1.1.1 or 11.0.0.0/8.

4. After configuring the parameters, click OK.

#### 2.6.4.1.2 Create a throttling plugin

You can use a throttling plugin to limit the number of API requests. The throttling plugin helps prevent the back-end service from being overwhelmed by large numbers of API requests. The following steps describe how to create a throttling plugin.

##### Procedure

1. [Log on to the API Gateway console.](#)
2. Click the plugins tab.

### 3. Click Create Plugin. On the Create Plugin page, set Type to Request Throttling.

**Create Plugin**

**Basic Settings**

Department: All

Region: cn-qingdao-env8d-d01  
Products from different network regions are not interoperable. Changing regions is not supported after selecting a region.

Project:

Name: A name must be 4 to 50 characters in length, and can contain English letters, Chinese characters, numbers, and underscores (\_). It must start with an English letter or a Chinese character.

Type: IP-based Access **Request Throttling** Signing Key

**Advanced Settings**

Time Unit: Seconds Minutes Hours Days

Maximum Requests per API: 1  
The maximum requests per API cannot exceed 100,000,000.

Maximum Requests per User: 1  
The value cannot exceed the value of Maximum Requests per API.

Maximum Requests per Application: 1  
The value cannot exceed the value of Maximum Requests per User.

Description: A description must be 0 to 200 characters in length.

Submit OK Cancel

Parameter	Description
Time Unit	The time granularity for throttling operations. Valid values: seconds, minutes, hours, and days.
Maximum Requests per API	The maximum number of times each API can be called per unit time. This parameter is set based on the back-end service capability.
Maximum Requests per User	The maximum number of requests that each user can make per unit time. The value of this parameter cannot exceed the value of Maximum Requests per API.
Maximum Requests per Application	The maximum number of requests that each application can make per unit time. The value of this parameter cannot exceed the value of Maximum Requests per User.

#### 4. After configuring the parameters, click OK.

### 2.6.4.1.3 Create a signature key plugin

A signature key plugin is used for the signature verification between API Gateway and backend services. You can create a signature key that consists of a key and a secret. The key and secret are equivalent to the account and password issued to API Gateway. When API Gateway sends a request to your backend service, API Gateway uses the signature key to calculate a signature string and passes the signature string to your backend service. Your backend service obtains the signature string for

symmetric calculation to verify the identity of API Gateway. The following steps describe how to create a signature key plugin.

## Procedure

1. [Log on to the API Gateway console.](#)
2. Click the plugins tab.
3. Click Create Plugin. On the Create Plugin page, set Type to Signing Key.

Configure the plugin parameters as needed.

4. After configuring the parameters, click OK.

### 2.6.4.2 Bind a plugin to an API

After you create a plugin, you must bind the plugin to an API for the plugin to take effect.

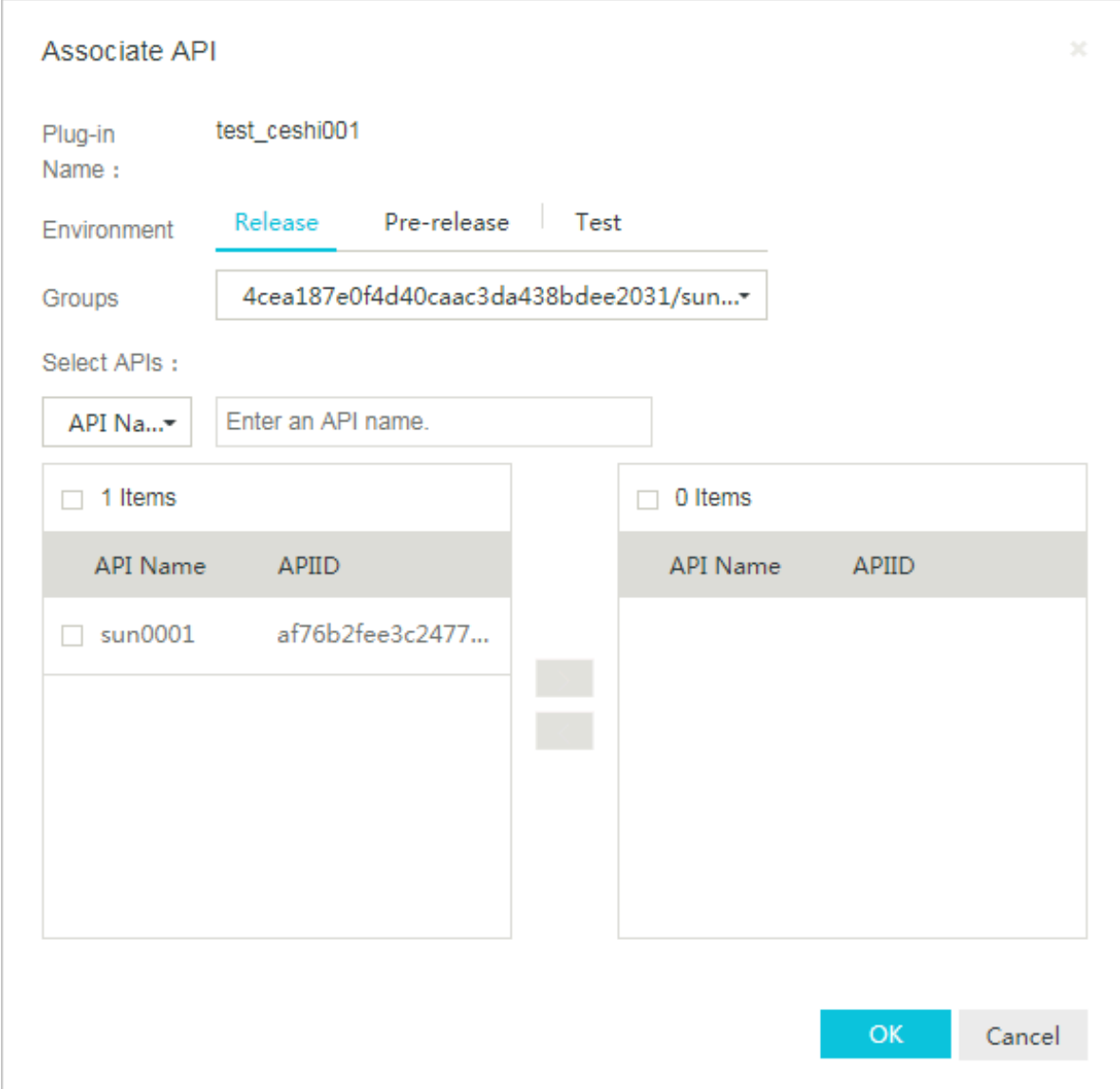
## Context

You can bind a plugin to multiple APIs. The plugin will take effect on each API individually. Only one plugin of each type can be bound to an API. If you bind a plugin to an API that already has a bound plugin of the same type, the new plugin will replace the previous one and take effect.

## Procedure

1. [Log on to the API Gateway console.](#)
2. Click the plugins tab.
3. Select the plugin to be bound, click the management icon in the Actions column corresponding to the plugin, and choose Associate API from the shortcut menu.
4. Select the environment and group of the API.

5. Select the API that you want to add, click the > icon to add the selected API to the right-side list, and click OK.



The dialog box is titled "Associate API" and contains the following fields and controls:

- Plug-in Name :** test\_ceshi001
- Environment:** Release (selected), Pre-release, Test
- Groups:** 4cea187e0f4d40caac3da438bdee2031/sun...▼
- Select APIs :**
  - API Na...▼** (dropdown menu)
  - Enter an API name.** (text input field)
- Left List (1 Items):**

API Name	APIID
<input type="checkbox"/> sun0001	af76b2fee3c2477...
- Right List (0 Items):**

API Name	APIID
----------	-------
- Buttons:** OK, Cancel

### 2.6.4.3 Delete a plugin

You can delete existing plugins.

#### Procedure

1. *Log on to the API Gateway console.*
2. Click the plugins tab.
3. Select the plugin that you want to delete, click the management icon in the Actions column corresponding to the plugin, and choose Delete plugin from the shortcut menu.
4. In the dialog box that appears, click OK.

## 2.6.4.4 Unbind a plugin

You can unbind plugins from the APIs that they are bound to.

### Procedure

1. *Log on to the API Gateway console.*
2. Click the **plugins** tab.
3. Select the plugin that you want to unbind, click the management icon in the **Actions** column corresponding to the plugin, and choose **View Details** from the shortcut menu.
4. On the plugin Management page that appears, click the management icon in the **Actions** column corresponding to the plugin, and choose **Unbind** from the shortcut menu.
5. In the dialog box that appears, click **OK**.

## 2.7 Advanced usage

### 2.7.1 Business parameters of custom logs

API Gateway provides the **Hack mode**, which allows you to record the request and response parameters of API calls in logs.

### Procedure

1. Log on to the Apsara Stack console.
2. In the left-side navigation pane, choose **Compute, Storage & Networking > API Gateway**.
3. Click the **Groups** tab.
4. Click the management icon in the **Actions** column corresponding to a group and choose **Change Group** from the shortcut menu.
5. In the **Description** field, add the following content:

```
logConf:reqBody=1024,reqHeaders,reqQuery,respHeaders,respBody=1024
```



#### Note:

- The content must be in a separate line. Otherwise, the configuration fails.



- You can also modify the content that follows `logConf:reqBody=1024`. For example, you can remove `respHeaders` to omit the response header and its content in the logs.
- `reqBody = 1024` indicates that the log only records up to 1,024 characters from the request body. Extra characters are truncated.

## 6. Log on to the Log Service console and check whether the description change has been recorded.

```
apiUid:73c226d7169148489a71c5d33813b3a5
appId:157555089414951
appName:integration_root
clientIp:172.31.224.26
clientNonce:c3de6af9-17be-49a4-b516-12d7d7535101
domain:ad41bfca1175433389bc6fa1669e2472.apigateway.inter.env11b.shuguang.com
errorCode:
errorMessage:
exception:
httpMethod:GET
initialRequestId:
instanceId:
path:/testIpControl
providerApiUid:1663875445751523
region:cn-qingdao-env11-d01
requestBody:
requestHandleTime:2020-01-03T07:52:14Z
requestHeaders:{"X-Ca-Key":"157555089415157","X-Ca-Stage":"PRE","X-Forwarded-Proto":"http","Host":"ad41bfca1175433389bc6fa1669e2472.apigateway.inter.env11b.shuguang.com","Date":"Fri, 03 Jan 2020 07:52:14 GMT","X-Ca-Signature-Headers":"X-Ca-Key,X-Ca-Nonce,X-Ca-Timestamp","X-Ca-Nonce":"c3de6af9-17be-49a4-b516-12d7d7535101","X-Ca-Timestamp":"1578037934480","X-Ca-Signature-Method":"HmacSHA256","X-Forwarded-For":"172.31.224.26","X-Ca-Signature":"CIBQR3ezw5GgZQGzZT8f5m3V4C2TncCKUVnn0se5c","eagleeye-poid":"0.1","X-Real-IP":"172.31.224.26","accept-encoding":"gzip","user-agent":"unirest-java/1.3.11"}
requestId:F88B9A2C-1191-4BC7-98AE-D5304BFC88DA
requestProtocol:http
requestQueryString:
requestSize:554
responseBody:{"Body":"","Headers":{"date":"Fri, 03 Jan 2020 07:52:14 GMT","host":"172.31.224.26:8080","x-ca-request-id":"F88B9A2C-1191-4BC7-98AE-D5304BFC88DA","connection":"Keep-Alive","accept-encoding":"gzip","x-ca-stage":"PRE","user-agent":"unirest-java/1.3.11","via":"73c226d7169148489a71c5d33813b3a5"},"Method":"GET","Params":{},"Path":"/web/cloudapi","RequestURL":"http://172.31.224.26:8080/web/cloudapi"}
responseHeaders:{"Transfer-Encoding":"chunked","Date":"Fri, 03 Jan 2020 07:52:14 GMT","Content-Type":"application/json"}
responseSize:220
serviceLatency:1
statusCode:200
```

## 2.7.2 Configure Log Service logs for API Gateway

### 2.7.2.1 Initialize the default Log Service configuration of API Gateway

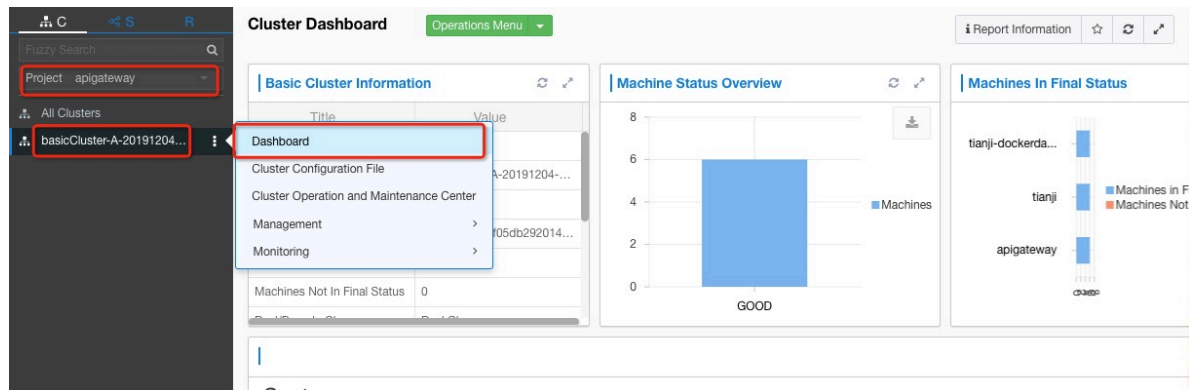
By default, API call logs in API Gateway are synchronized to Log Service for Apsara Stack services. However, your account can be activated only after you log on to the Log Service console from Apsara Infrastructure Management Framework.

#### Procedure

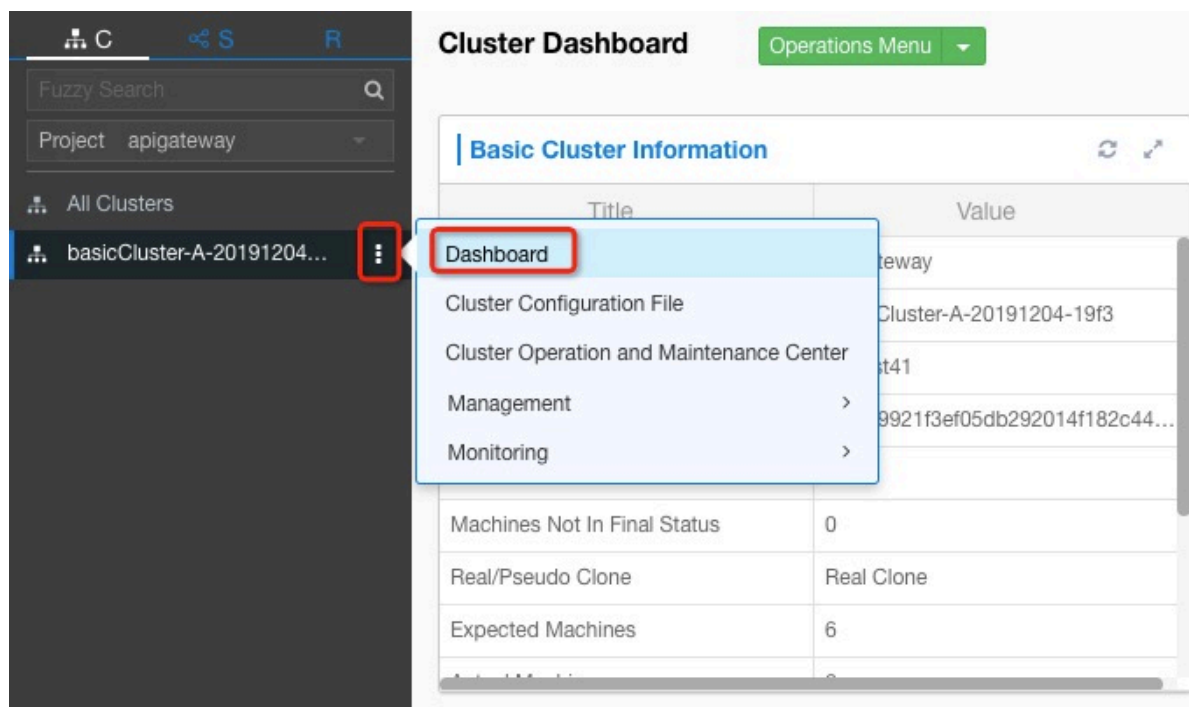
1. Log on to the Apsara Infrastructure Management Framework console.

2. In the left-side navigation pane, click the C tab and then select **apigateway** from the Project drop-down list.

Figure 2-2: Cluster O&M page



3. Place the pointer over the  icon next to one of the filtered clusters, and choose **Dashboard** from the shortcut menu.



4. In the Cluster Resources section, find the service with Name set to **apigateway-sls** and Type set to **accesskey**. Right-click the value in the Result column and

**choose Show More from the shortcut menu to view the values of accesskey-id and accesskey-secret.**

Cluster Resource	Serv...	App	Name	Type	Status	Error Msg	Param...	Result	Res	Repro...	Repro...	Repro...	Refer V...
apigate...	apigatew...	console-ba...	apigatew...console	dns	done		{ "domain": ...	{ "ip": "\[10...	60212c5a5...				[ "d4c24d5
apigate...	apigatew...	apigatew...	api_gateway	db	done		{ "minirds_p...	{ "passwd": "...	9fe0e0e859...				[ "d4c24d5
apigate...	apigatew...	cloudapi-ga...	apigatew...inner	tg-vip	done		{ "bid": "clo...	{ "domain": "...	0e2d2f654a...	done		{ "domain": ...	[ "d4c24d5
apigate...	apigatew...	cloudapi-ga...	apigatew-load	accesskey	done		{ "name": "...	{ "name": "a...	ecb6a27b6...				[ "d4c24d5
apigate...	apigatew...	cloudapi-ga...	apigatew	tg-vip	done		{ "bid": "clo...	{ "domain": "...	3aaf716f3...	done		{ "domain": ...	[ "d4c24d5
apigate...	apigatew...	cloudapi-op...	apigatew-sls	accesskey	done		{ "name": "...	{ "name": "a...	3c742c22f8...				[ "d4c24d5
apigate...	apigatew...	cloudapi-op...	apigatew	accesskey	done		{ "name": "...	{ "name": "a...	2a7da0edc...				[ "d4c24d5
apigate...	apigatew...	cloudapi-dns	apigatew-dns	dns	done		{ "domain": ...	{ "ip": "\[10...	153cf0ead8...				[ "d4c24d5
apigate...	apigatew...	cloudapi-op...	apigatew-api-vpc	dns	done		{ "domain": ...	{ "ip": "\[10...	57dc3662a...				[ "d4c24d5
apigate...	apigatew...	service_test	apigatew-test	accesskey	done		{ "name": "...	{ "name": "a...	652637bc76...				[ "d4c24d5

Details

Formatted Value

Original Value

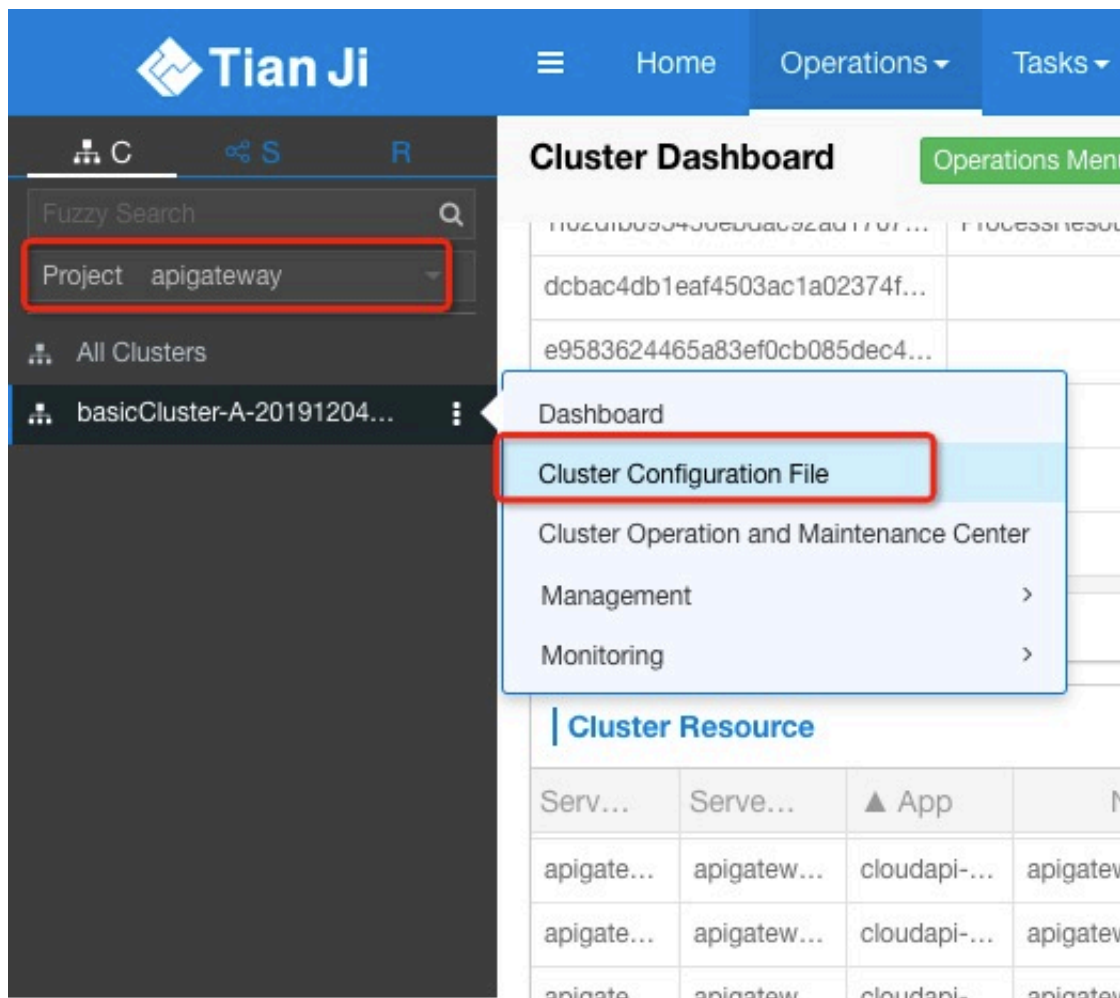
```
{
  "name": "apigateway-test",
  "accesskey-id": "AKIAQ0C3K377C6U",
  "password_encrypted": "Y1ne9FX9kFtNvYoLVGfsc5XDh8XlbaDwyDQidVVqMBIXtFeSVu0qtIUQJP+zYsbq5jYdZ8nyHhqiwm/lvAg6B8NgubghKqH1hn8w0QYiEuZXiAvkI4dIR50AoL",
  "accesskey-secret": "u0Ufz7vbmnc7hctjyQQzwwfms255um",
  "accesskey-secret_encrypted": "WzPcwCienDPEcNmZJIMzYE9qh5c6eKwPwtFzLL8shWN0+jMeaAOpY0zW8EThow0mRnnIBPzF6Kq4zGXJ2ASqt0HBFgvs7EUNl3ERkQv4lba00bvXN",
  "user": "apigateway-test@aliyun.com",
  "password": "sLrQisYtyz44b57nvg",
  "id": "1663875445751523"
}
```

5. Use the obtained `accesskey-id` and `accesskey-secret` values to log on to the Log Service console of Apsara Infrastructure Management Framework.

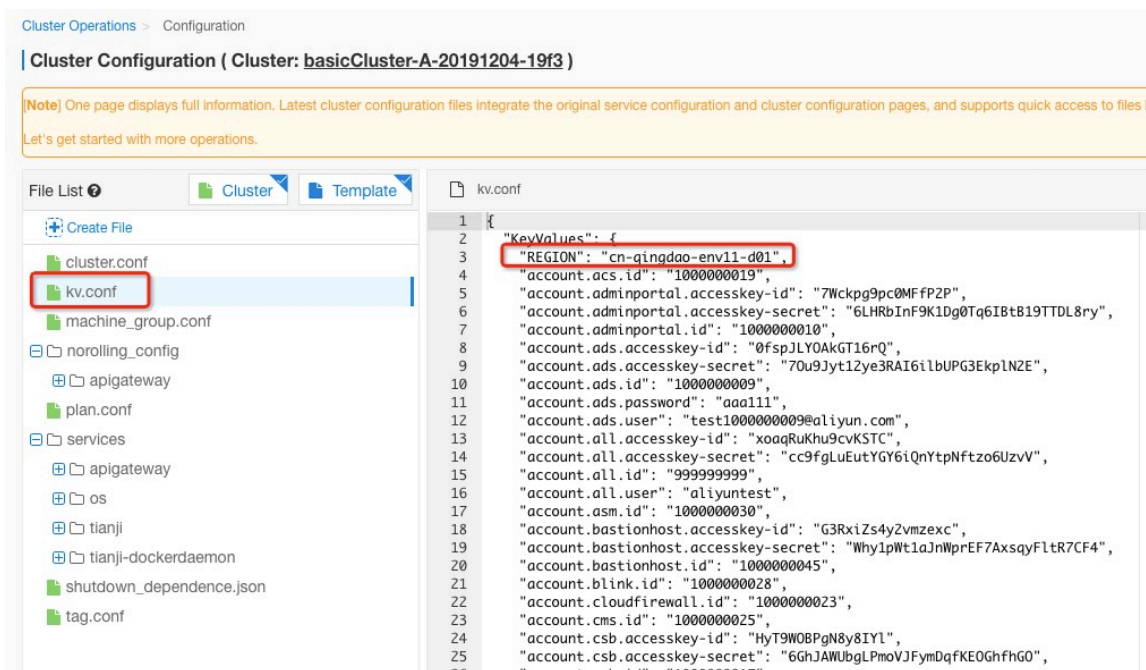
**The URL of Log Service in Apsara Infrastructure Management Framework is `http://portal.${region}.sls.${internet-domain}`. You can obtain the values of region**

and internet-domain from the kv.conf file in Apsara Infrastructure Management Framework.

- a. Place the pointer over the More icon next to the apigateway cluster and choose Cluster Configuration File from the shortcut menu.



- b. Click the kv.conf file to view the values of region and internet-domain.

**Note:**

After you log on to the system, Log Service is automatically configured for API Gateway. This operation takes several minutes.

## 2.7.2.2 Configure API Gateway to deliver logs to your Log Service project

If you do not want to use Log Service and a fixed account in Apsara Infrastructure Management Framework, you can create a Log Service project in the Apsara Stack console through Log Service. Then, configure API Gateway to deliver logs to your Log Service project.

### Context

**You must first modify the configurations on machines where API Gateway resides. Then, create a Log Service project and configure the Logstores and machine groups. The procedure is as follows:**

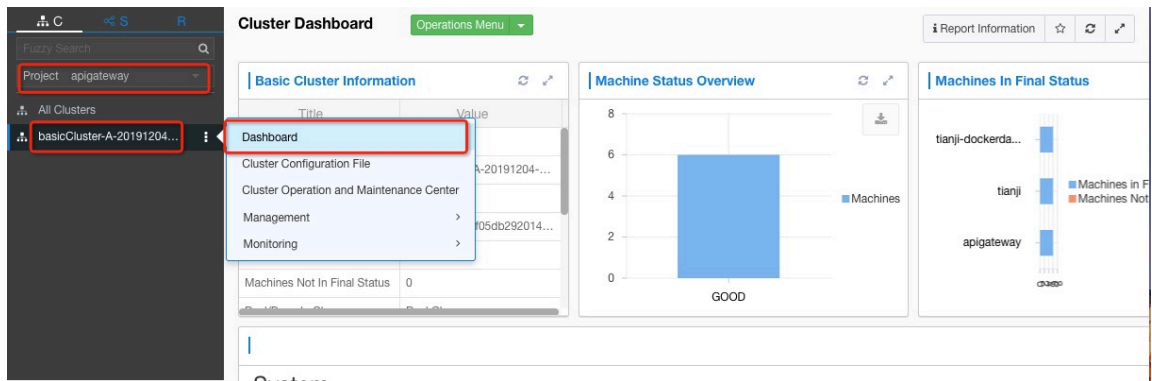
1. Modify the configurations on machines where API Gateway resides

**You can use the Hack method to manually modify the Logtail configurations of API Gateway and change the log delivery method.**

1. First, find the machines where API Gateway resides.

- a) Log on to the Apsara Infrastructure Management Framework console. In the left-side navigation pane, click the C tab and select apigateway from the

**Project drop-down list. Place the pointer over the More icon next to one of the filtered clusters and choose Dashboard from the shortcut menu.**



**b) On the dashboard page, go to Service Instance List, and find apigateway in the Service Instance column.**

Service Instances					
Service Instance	Final Status	Expected Server Roles	Server Roles In Final Status	Server Roles Going Offline	Actions
apigateway	True	5	5	0	Actions ▾ Details
os	True	--	--	--	Actions ▾ Details
tianji	True	1	1	0	Actions ▾ Details
tianji-dockerdaemon	True	1	1	0	Actions ▾ Details

**c) Click Details in the Actions column corresponding to the apigateway service instance. In the Server Role List section, find ApigatewayLite# in the Server Role column.**

Server Role List							
Server Role	Current Status	Expected Machines	Machines In Final ...	Machines Going ...	Rolling Task Status	Time Used	Actions
ApigatewayConsole#	In Final Status	2	2	0	no rolling		Details
ApigatewayDB#	In Final Status	1	1	0	no rolling		Details
ApigatewayLite#	In Final Status	3	3	0	no rolling		Details
ApigatewayOpenAPI#	In Final Status	2	2	0	no rolling		Details
ServiceTest#	In Final Status	1	1	0	no rolling		Details

**d) Click Details in the Actions column corresponding to the ApigatewayLite# server role. Then, find the Machine Information section.**

Machine Information									
Machi...	IP	Machi...	Machi...	Server...	Server...	Curren...	Target ...	Error ...	Actions
ecsapigate...	172.31.228.9	good		good   PR...		f52a09921...	f52a09921...		Terminal Restart Details Machine System View Machine Operation
ecsapigate...	172.31.22...	good		good   PR...		f52a09921...	f52a09921...		Terminal Restart Details Machine System View Machine Operation
ecsapigate...	172.31.22...	good		good   PR...		f52a09921...	f52a09921...		Terminal Restart Details Machine System View Machine Operation



#### Note:

**Perform steps 2, 3, and 4 on each machine.**

2. Log on to the gateway terminal and go to the `/alidata/settings` directory to create the `init_ilogtail.sh` script. Note that the name cannot be changed. The script content is as follows:

```
#!/bin/bash
wget data.cn-qingdao-env4b-d01.sls-pub.env4b.shuguang.com/logtail.sh
sudo sh logtail.sh install
```

**Note:**

In the preceding content, `data.cn-qingdao-env4b-d01.sls-pub.env4b.shuguang.com` needs to be replaced with the value of the `sls_data.endpoint` variable in the `sls-backend-server` service. You can find the value from the Registration Vars of Services report in Apsara Infrastructure Management Framework.

3. Add execution permissions to the `init_ilogtail.sh` file.

```
sudo chmod +x init_ilogtail.sh
```

4. Access the gateway container (the container in the server role starting with `ApigatewayLite`) from Apsara Infrastructure Management Framework, and then run the following commands:

```
cd /home/admin/bin
sudo sh config_ilogtail.sh MANUAL
```

If the following information is displayed, the installation is successful.

```
install logtail success
```

## 2. Configure logs in the Log Service console

1. Log on to the Apsara Stack console.
2. In the left-side navigation pane, choose **Compute, Storage & Networking > Log Service**.
3. Click **Create Project** and set the parameters.
4. Click **Go to Console**. Set **Region** and **Department**, and then click **SLS**.  
You will be directed to the Log Service console.
5. In the Log Service console, click the project you created.
6. On the **Logstores** page, create a Logstore.



7. In the left-side navigation pane, choose **LogHub - Collect > Logtail Config**. The **Logtail Configurations** page appears.

8. Click **Create**. The configuration page appears. select text

9. Configure the data source as follows, and then click **Next**.

Parameter	Value
Configuration Name	gateway_request_log
Log Path	/alidata/www/logs/java/cloudapi-gateway/logs/request_user.log
Mode	Delimiter Mode
Log Sample	<pre> 2019-08-15 14:25:05 CC7C526B-C915-44F1- 93A2-F44DBBE35177 b2909c9fa6 6146f19baf2bd8f4709ab8  integration_root e4032f87ac e14cc6965cf5352a9637a6 RELEASE  0619f7763b004fc3a16a41dd712ce8 d7 biz1_anonymous 10.4.21.241   b2909c9fa66146f19baf2bd8f4709ab8 .apigateway.env4b.shuguang.com  POST /biz1/anonymous 403 A403JT: Invalid JWT: deserialized JWT failed  1453964555641148 cn- qingdao-env4b-d01 2019-08-15T06 :25:05Z 614 0 0 A403JT http   cf802da1-54d0-49b8-b77c- 2b20b172d90a {"X-JWT-Token":" bad jwt token"} a=%21111      </pre>
Delimiter	Vertical Line

10. Click **Next** to go to the **Apply to Machine Group** page. If no machine groups are available, create a machine group. Enter a machine group name and enter all IP addresses of Docker containers.

11. Select the newly created machine group and click **Apply to Machine Group**.

Use the default settings until the Logstore is created.

12. After the configuration is complete, check whether the configuration takes effect. In the left-side navigation pane of the Log Service project, choose **LogHub - Collect > Logtail Machine**. On the **Machine Groups** page, find the created



machine group and click Status in the Actions column to check whether the Logtail heartbeat of the machine group is normal.



Note:

:

- The heartbeat check takes several minutes.
- If the heartbeat is normal, you can query logs.

## 2.7.3 Cross-user VPC authorization

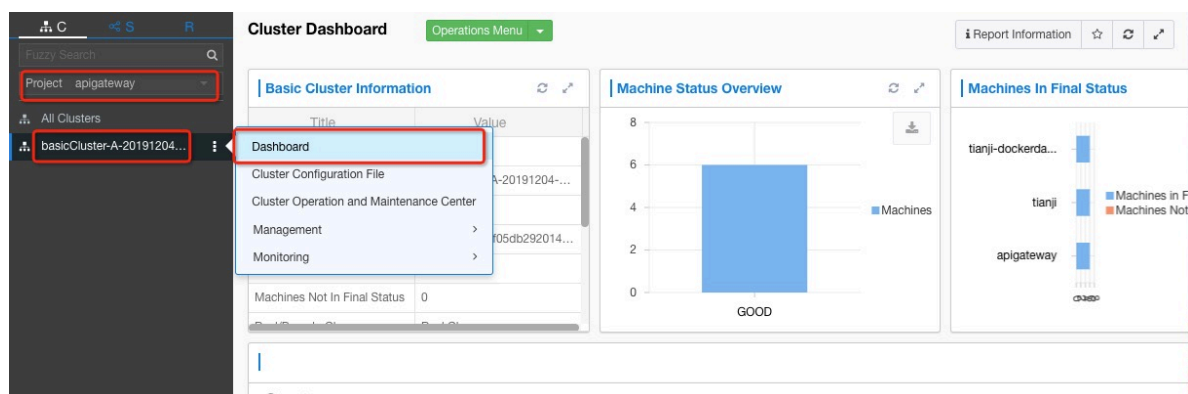
If your backend service resides in a VPC, you must configure the backend service address through VPC authorization. By default, a VPC owner must be the same user as an API owner. Starting from Apsara Stack V3.8.1, API Gateway provides two internal APIs that allow VPC owners to authorize their VPCs to other users.

### 2.7.3.1 User authorization across VPCs

APIs can be used across multiple VPCs. For security reasons, VPC owners must call APIs to explicitly authorize access to VPCs before API providers can use the VPCs. The OpenAPI component of API Gateway has a built-in Aliyun CLI tool. You can use this tool to authorize access to VPCs. The following steps describe how to call the API:

#### Procedure

1. Log on to the Apsara Infrastructure Management Framework console. In the left-side navigation pane, click the C tab and select apigateway from the Project drop-down list. Place the pointer over the More icon next to one of the filtered clusters and choose Dashboard from the shortcut menu.



2. On the dashboard page, go to Service Instance List, and find apigateway in the Service Instance column.

Service Instances					
Service Instance	Final Status	Expected Server Roles	Server Roles In Final Status	Server Roles Going Offline	Actions
apigateway	True	5	5	0	Actions ▾ Details
os	True	--	--	--	Actions ▾ Details
tianji	True	1	1	0	Actions ▾ Details
tianji-dockerdaemon	True	1	1	0	Actions ▾ Details

3. Click Details in the Actions column corresponding to the apigateway service instance. In the Server Role List section, find ApigatewayLite# in the Server Role column.

Server Role List							
Server Role	Current Status	Expected Machines	Machines In Final ...	Machines Going ...	Rolling Task Status	Time Used	Actions
ApigatewayConsole#	In Final Status	2	2	0	no rolling		Details
ApigatewayDB#	In Final Status	1	1	0	no rolling		Details
ApigatewayLite#	In Final Status	3	3	0	no rolling		Details
ApigatewayOpenAPI#	In Final Status	2	2	0	no rolling		Details
ServiceTest#	In Final Status	1	1	0	no rolling		Details

4. Click Details in the Actions column corresponding to the ApigatewayLite# server role. Then, find the Machine Information section.

Machine Information									
Machi...	IP	Machi...	Machi...	Server...	Server...	Curren...	Target ...	Error ...	Actions
ecsapigate...	172.31.228.9	good		good   PR...		f52a09921...	f52a09921...		Terminal Restart Details Machine System View Machine Operation
ecsapigate...	172.31.22...	good		good   PR...		f52a09921...	f52a09921...		Terminal Restart Details Machine System View Machine Operation
ecsapigate...	172.31.22...	good		good   PR...		f52a09921...	f52a09921...		Terminal Restart Details Machine System View Machine Operation

5. Click Terminal in the Actions column corresponding to a machine in the server role to access the container.

6. Configure the AccessKey pair used to call the CLI tool. Run the following commands:

```
aliyun configure -- profile vpctest //Add the AccessKey pair
configuration. vpctest is the profile name, which can be customized
. After you press Enter, configure AccessKeyId and AccessSecret as
prompted.
aliyun configure list //View the config configuration to check
whether the preceding profile has been added.
```

7. Run the following command to perform authorization:

```
aliyun cloudapi AuthorizeVpc --VpcId vpc-tb5mfcwx3s4zqctzw**** --
TargetUserId 147546214349****
--endpoint apigateway.cn-qingdao-env11-d01.inter.env11b.shuguang.com
--force
```

```
--profile vpctest
```

**Note:**

- Log on to the Apsara Infrastructure Management Framework console. In the top navigation bar, choose Reports > System Reports. On the System Reports page, click Registration Vars of Services. On the Registration Vars of Services report, right-click the value in the Service Registration column corresponding to the apigateway service and choose Show More from the shortcut menu. The value of the apigateway.openapi.endpoint variable must be used as the endpoint in the preceding command. The profile value also needs to be replaced based on actual needs.
- This command authorizes the user whose ID is 1475462143497330 to use the VPC with the ID vpc-tb5mfcwx3s4zqctzw19w2. You can replace the parameter values as needed.

**Success Operation Sample**

```
[root@docker010011102023 ~]#
#aliyun cloudapi AuthorizeVpc --VpcId vpc-tb5mfcwx3s4zqctzw19w2 --TargetUserId 1475462143497330 --endpoint apigateway.cn-qingdao-env
11-d01.inter.env11b.shuguang.com --force --profile vpctest
{"RequestId":"8E64BC51-60D7-4D85-B5F0-3E3F12C4B6D2"}
```

## 2.7.3.2 API configurations

After an application is authorized to call an API, the API owner must configure the API and define the API backend service. You cannot select the VPC ID of another user in the Apsara Stack console. Therefore, you need to configure the API backend service address.

**Context**

When you configure an API, take note of the following points:

- Choose Do Not Authorize Access to VPCs from the VPC ID drop-down list.
- The backend service address must be in the http(s)://{backend service IP address}.{vpcId}.gateway.vpc:{port} format. The content in {} can be substituted as needed. Example:

```
http://192.168.XX.XX.vpc-tb5mfcwx3s4zqctzw****.gateway.vpc:8080
```