

# 阿里云

## 企业级分布式应用服务 EDAS 应用托管（ECS）

文档版本：20220708

## 法律声明

阿里云提醒您,在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

# 通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.应用托管概述 (ECS)	09
2.资源管理	11
2.1. ECS资源管理概述	11
2.2. 管理微服务空间	11
2.2.1. 使用控制台管理微服务空间	11
2.2.2. 使用API管理	13
2.3. ECS集群管理	16
2.3.1. 使用控制台创建ECS集群	16
2.3.2. 使用控制台扩容ECS集群	18
2.3.3. 使用API创建ECS集群	22
2.3.4. 使用API扩容ECS集群	24
2.3.5. 管理ECS集群	28
2.3.6. 使用控制台管理ECS集群中的应用实例分组	30
2.3.7. 使用API管理ECS集群中的应用实例分组	32
2.4. ECS实例管理	39
2.4.1. 创建ECS实例	39
2.4.2. 为ECS实例安装EDAS Agent	40
2.5. 创建混合云ECS集群	44
2.6. 管理资源组	47
3.创建和部署应用 (ECS)	50
3.1. 应用创建和部署概述 (ECS)	50
3.2. 在ECS集群中创建并部署应用	51
3.3. 在混合云中部署应用	55
3.4. 使用API创建应用 (ECS)	55
4.升级和回滚应用 (ECS)	59
4.1. 升级和回滚应用概述 (ECS)	59

---

4.2. 单批发布 (ECS)	59
4.2.1. 使用控制台升级应用 (ECS集群)	59
4.2.2. 使用API在ECS集群中单批发布应用	61
4.2.3. 在IntelliJ IDEA中使用Cloud Toolkit部署应用 (ECS集群)	64
4.2.4. 在Eclipse中使用Cloud Toolkit部署应用至ECS集群	69
4.2.5. 使用toolkit-maven-plugin插件升级应用 (ECS集群)	74
4.2.6. 使用CLI快速部署应用至ECS集群	76
4.3. 分批发布 (ECS)	80
4.3.1. 使用控制台分批发布应用	80
4.3.2. 使用API在ECS集群中分批发布应用	82
4.3.3. 使用toolkit-maven-plugin插件分批发布应用 (ECS集群)	85
4.3.4. 使用Cloud Toolkit分批发布应用	88
4.4. 金丝雀发布 (ECS)	92
4.4.1. 使用控制台金丝雀发布 (ECS集群)	92
4.4.2. 使用API金丝雀发布 (ECS集群)	95
4.5. 回滚应用 (ECS集群)	98
4.5.1. 使用控制台回滚应用 (ECS集群)	98
4.5.2. 使用API在ECS集群中回滚应用到目标历史版本	101
4.5.3. 使用API在ECS集群中终止应用升级变更流程	103
5. CI/CD	106
5.1. 使用云效2020部署Java应用至ECS集群	106
5.2. 使用Jenkins创建持续集成	113
6. 应用设置	120
6.1. 设置部署在ECS集群中的应用	120
6.2. 设置应用生命周期挂载脚本	121
6.3. 健康检查	124
6.4. 同可用区优先路由	126
7. 告警管理	129

---

---

7.1. 告警管理升级说明	129
7.2. 名词解释	131
7.3. 联系人管理	133
7.3.1. 联系人	133
7.3.2. 联系人组	135
7.3.3. 钉钉机器人	136
7.3.4. 通过Webhook自定义告警通知人	137
7.3.5. 获取钉钉机器人Webhook地址	141
7.4. 升级策略	143
7.5. 应用监报告警规则 (新版)	144
7.6. 查看告警发送历史	147
7.7. 查看告警事件历史	152
7.8. 创建应用监报告警并发送告警通知	154
8.应用运维	160
8.1. ECS 集群中的应用生命周期管理	160
8.2. 扩容部署在ECS集群中的应用	161
8.3. 负载均衡	164
8.3.1. 使用SLB访问ECS上的EDAS应用	164
8.3.2. 应用独享负载均衡实例	172
8.3.3. 使用URL实现应用实例分组的流量分发	174
8.3.4. 使用端口实现流量分发 (共享负载均衡实例)	177
8.3.5. 使用域名或URL实现流量分发 (共享负载均衡实例)	179
8.4. 对ECS集群中的应用进行弹性伸缩	181
8.5. 限流降级	184
8.6. 升级或降级运行时环境	186
8.7. 变更记录	188
8.7.1. 查看应用变更	188
8.7.2. 变更流程错误码说明	189

---

---

8.8. 日志	192
8.8.1. 日志简介	193
8.8.2. 日志管理	193
9. ECS集群使用常见问题	199
9.1. 资源管理	199
9.1.1. Agent 安装之后没有上报, 界面未显示 Agent 的版本号	199
9.1.2. Agent安装后控制状态“未知”或“异常”	199
9.1.3. 什么是 Docker 实例? 和 ECS 独占实例有何区别?	199
9.1.4. 购买了 EDAS 之后, 是否需要另行购买 ECS?	200
9.1.5. EDAS的Java版本是哪个版本, 我能够选择吗?	200
9.1.6. 如果 Agent 的心跳进程停了, 会有什么样的后果?	200
9.1.7. 如果 Ali-Tomcat 突然容器退出, 该如何处理	200
9.1.8. 资源未及时同步怎么办?	201
9.1.9. 安装 Agent 的过程中发现卸载 Java	201
9.1.10. 日志采集器相关问题	201
9.2. 应用管理	202
9.2.1. 重启服务器后 EDAS Agent 是否能自动重启?	202
9.2.2. EDAS Agent 为什么无法启动?	202
9.2.3. EDAS 是否支持在同一个实例上安装多个应用?	202
9.2.4. 为什么应用运行时进程突然消失了?	202
9.2.5. 应用部署时提供的发布包 URL 是不是可以随意设置?	204
9.2.6. 启停和部署应用等管理操作失败了怎么办?	204
9.2.7. 已买的 ECS 实例没有出现在实例列表中怎么办?	204
9.2.8. 为什么在 EDAS 控制台上实例的状态是“未知”?	205
9.2.9. 服务可以正常调用, 但是服务列表不显示	205
9.2.10. 服务列表显示正常, 但是调用失败怎么办?	205
9.2.11. 删除应用后是否可以恢复?	205
9.2.12. 如何进行分批发布或者分组发布	205

---

---

9.2.13. 应用部署在多个实例后, 集群中会话如何共享? .....	206
9.2.14. 应用部署在多个实例后, 如何实现负载均衡? .....	206
9.2.15. 如何分批升降级应用的运行环境 .....	206
9.2.16. 在 EDAS 控制台上对应用进行操作无响应 .....	206
9.2.17. 创建应用时的安全组规则 .....	206
9.2.18. ECS集群应用日志清理机制是什么 .....	206
9.2.19. 变更流程问题排查指南 .....	207
9.2.20. EDAS应用无法访问页面及RPC服务无法调用 .....	209
9.2.21. 快速入门常见问题 .....	210
9.2.22. 应用部署时的默认启动命令参数 .....	213
9.3. 监控报警 .....	216
9.3.1. 本地使用jconsole或jvisualvm监控ECS实例上的JVM进程 .....	216
9.4. 日志诊断常见问题 .....	219
9.4.1. Java 应用 CPU 使用率高 .....	219

# 1.应用托管概述 (ECS)

您可以将Java应用托管到EDAS的ECS集群或混合云ECS集群。EDAS为托管的应用提供了生命周期管理、CI/CD、监控、运维等一系列功能。

## 概览



## ECS环境类型

EDAS支持阿里云ECS集群和混合云ECS集群。

- 阿里云ECS集群：在专有网络VPC内创建的集群，由阿里云ECS实例组成。更多信息，请参见[使用控制台创建ECS集群](#)。
- 混合云ECS集群：在专有网络VPC内创建的集群，包含阿里云ECS实例、IDC或其它云服务提供商的服务器。更多信息，请参见[创建混合云ECS集群](#)。

## ECS环境支持的应用类型

您可以使用EDAS的ECS环境托管Spring Cloud和Dubbo等主流Java微服务框架应用，也可以托管HSF框架应用。

## ECS环境提供的应用托管功能

ECS环境为应用提供完整的生命周期管理、服务治理、监控、运维，以及托管应用所需的资源管理能力。

使用场景	功能
资源管理	EDAS是面向应用的PaaS平台，为应用提供ECS和K8s环境及相关资源。EDAS的ECS环境中主要包含命名空间、VPC、集群和ECS。更多信息，请参见 <a href="#">ECS资源管理概述</a> 。
部署应用	<ul style="list-style-type: none"> <li>• 创建和部署（首次部署）：在应用开发、测试完成后，可以在ECS集群中创建并部署。更多信息，请参见<a href="#">应用创建和部署概述 (ECS)</a>。</li> <li>• 升级和回滚：应用在EDAS中创建并部署后，还会不断迭代，需要升级。如果升级的应用版本发现问题，需要将应用回滚到历史版本。更多信息，请参见<a href="#">升级和回滚应用概述 (ECS)</a>。</li> </ul>

使用场景	功能
CI/CD	<p>应用在不断迭代过程中，需要持续集成（CI）和持续部署（CD）。EDAS支持通过<a href="#">Jenkins</a>和<a href="#">云效</a>对您部署的应用进行CI/CD。具体操作，请参见：</p> <ul style="list-style-type: none"><li>• <a href="#">使用云效2020部署Java应用至ECS集群</a></li><li>• <a href="#">使用Jenkins创建持续集成</a></li></ul>
应用监控	<p>EDAS集成了应用实时监控服务ARMS，为部署在EDAS中的应用提供关键指标的监控能力和报警能力。具体操作，请参见<a href="#">应用总览</a>。</p>
应用运维	<p>EDAS为应用提供了一系列运维功能，包括：</p> <ul style="list-style-type: none"><li>• <a href="#">ECS 集群中的应用生命周期管理</a></li><li>• <a href="#">负载均衡概述</a></li><li>• <a href="#">对ECS集群中的应用进行弹性伸缩</a></li><li>• <a href="#">限流降级</a></li><li>• <a href="#">查看应用变更</a></li><li>• <a href="#">日志简介</a></li></ul>

## 2. 资源管理

### 2.1. ECS资源管理概述

EDAS是面向应用的PaaS平台，为应用提供ECS和K8s环境及相关资源。本文介绍ECS环境的相关资源以及如何EDAS中管理。

EDAS的ECS环境中主要包含微服务空间、VPC、集群和ECS。

资源	作用	EDAS管理操作
微服务空间	微服务空间提供互相隔离的运行环境，如开发、测试和生产环境等。您可以使用微服务空间实现资源和服务的隔离，并可以使用一个账号进行统一管理。	创建、编辑和删除微服务空间。具体操作，请参见 <a href="#">使用控制台管理微服务空间</a> 。
VPC	阿里云提供的专有网络服务，VPC之间逻辑上完全隔离。如果您有较高的网络安全要求，可以选择在VPC中创建集群、部署应用。	VPC的创建和管理等操作在VPC控制台进行，且会自动同步到EDAS。
集群	应用运行所需的云资源组合。在EDAS环境提供了两类ECS集群：阿里云ECS集群和混合云ECS集群。	创建、扩容（添加实例）、管理和删除阿里云ECS集群和混合云ECS集群。 <ul style="list-style-type: none"> <li>• 阿里云ECS集群                             <ul style="list-style-type: none"> <li>◦ <a href="#">使用控制台创建ECS集群</a></li> <li>◦ <a href="#">使用控制台扩容ECS集群</a></li> <li>◦ <a href="#">管理ECS集群</a></li> </ul> </li> <li>• 混合云ECS集群                             <ul style="list-style-type: none"> <li>◦ <a href="#">创建混合云ECS集群</a></li> </ul> </li> </ul>
ECS	阿里云提供的基础云计算服务。ECS实例用于部署应用。	EDAS仅提供了创建（购买）ECS的跳转入口，实际创建是在ECS购买页完成。在EDAS中，对ECS实例最主要的管理操作为安装EDAS Agent。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <span style="font-size: 1.2em; color: #00aaff;">?</span> <b>说明</b> 创建ECS实例时，也可以通过选择EDAS基础镜像安装EDAS Agent。                     </div> 具体操作，请参见 <a href="#">为ECS实例安装EDAS Agent</a> 。

### 2.2. 管理微服务空间

#### 2.2.1. 使用控制台管理微服务空间

微服务空间为应用提供互相隔离的运行环境，如开发、测试和生产环境等，您可以使用微服务空间实现服务的隔离。在同一个地域内不允许创建两个同名的微服务空间。

## 应用场景

例如，您在同一个VPC内有三套环境，分别用于应用开发、测试和生产，可以为这三套环境分别创建一个微服务空间（如Dev、Test和Prod）。不同微服务空间下的应用和服务是相互隔离的，不能进行服务调用和配置推送。

## 默认微服务空间

在EDAS创建应用时，可以选择默认微服务空间，默认微服务空间即为没有微服务空间的状态，表示该应用不使用任何微服务空间来实现服务的隔离。

在应用列表内选择某地域下的默认微服务空间时，即表示筛选出来的应用都没有归属于任何微服务空间。

## 创建微服务空间

1. 登录EDAS控制台。
2. 在左侧导航栏中选择资源管理 > 微服务空间。
3. 在微服务空间页面右上角单击创建微服务空间。
4. 在创建微服务空间对话框配置微服务空间参数，然后单击创建。

创建微服务空间
✕

\* 微服务空间

\* 微服务空间ID

注册配置中心  MSE微服务注册中心  EDAS注册配置中心 i

\* MSE Nacos实例  ↕ + 创建MSE Nacos实例

请确保MSE Nacos实例与ECS/K8s集群在同一个VPC下，否则服务将无法注册

归属地域 华东1

允许远程调试

描述  0/64

创建
取消

参数	描述
微服务空间	请输入您创建的微服务空间的名称。
微服务空间ID	请输入自定义的字符来形成微服务空间的ID，仅允许输入英文字母或数字。

参数	描述
注册配置中心	<ul style="list-style-type: none"> <li>MSE Nacos: 您购买的阿里云MSE Nacos引擎, 可与EDAS无缝结合, 实现服务注册及配置管理。</li> <li>EDAS注册配置中心: EDAS提供的免费服务注册配置中心, 若应用对性能及稳定性有比较高的要求, 建议使用MSE Nacos作为服务注册配置中心。</li> </ul>
MSE Nacos实例	MSE构建的Nacos实例。具体操作, 请参见 <a href="#">创建MSE Nacos实例</a> 。
归属地域	当前微服务空间所归属的地域, 不可更改。
允许远程调试	当您想对应用进行端云互联时, 您在该应用所在的微服务空间的编辑页面手动开启 <a href="#">允许远程调试</a> 。端云调试的相关操作, 请参见 <a href="#">端云互联简介</a> 。
描述	请输入一段文字来描述微服务空间。

## 编辑微服务空间

- 在微服务空间列表的操作列单击**编辑**。
- 在**编辑微服务空间**对话框, 您可修改**微服务空间**名称及**描述**, 也可开启或关闭远程调试功能。在编辑完成后单击**确定**。

## 删除微服务空间

删除微服务空间需要满足以下前提条件:

- 微服务空间下无任何集群。
- 微服务空间下无任何微服务应用。
- 微服务空间下无任何ECS实例。

- 在微服务空间列表的操作列中单击**删除**。
- 在**确认对话框**中单击**删除**。

## 2.2.2. 使用API管理

本文介绍如何使用EDAS提供的Java SDK调用API创建、编辑和删除自定义微服务空间。

### 前提条件

- 安装EDAS的Java SDK。更多信息, 请参见[Java SDK接入指南](#)。
- 确定微服务空间创建的地域, 假设为 `cn-hangzhou`。

### 创建或编辑微服务空间

以下代码用于快速创建或编辑自定义微服务空间。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.InsertOrUpdateRegionRequest;
import com.aliyuncs.edas.model.v20170801.InsertOrUpdateRegionResponse;
public class InsertOrUpdateRegion {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //需要创建或编辑的微服务空间所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求, 并设置参数。
        InsertOrUpdateRegionRequest request = new InsertOrUpdateRegionRequest();
        //微服务空间ID, 格式为“地域ID:微服务空间标识符”, 例如: cn-hangzhou:doc。
        request.setRegionTag("cn-hangzhou:doc");
        //微服务空间名称, 例如: APIdoc。
        request.setRegionName("APIdoc");
        //微服务空间的描述信息。
        request.setDescription("使用API创建的微服务空间");
        //此参数为空或者为0表示创建微服务空间, 其他取值表示编辑微服务空间。
        request.setId((long) 0);
        //是否允许远程调试, true表示允许, false表示不允许。
        request.setDebugEnable(true);
        try {
            InsertOrUpdateRegionResponse response = client.getAcsResponse(request);
            System.out.println("RegionName=" + response.getUserDefineRegionEntity().getRegionName() + "\nRegionId=" + response.getUserDefineRegionEntity().getRegionId()+ "\nId=" + response.getUserDefineRegionEntity().getId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序, 返回的执行结果如下:

```
RegionName=APIdoc
RegionId=cn-hangzhou:doc
Id=17926
```

 **说明** 此处执行结果中的 `Id=17926`, 是系统中定义微服务空间的唯一标识序号。如果您需要删除微服务空间, 则需要通过这个标识进行删除。

## 删除微服务空间

如果您在创建或编辑微服务空间时，已记录微服务空间的唯一标识序号（如Id=17926），则可以直接进行删除操作。

如果您没有记录微服务空间的唯一标识，则需要调用ListUserDefineRegion接口，获取微服务空间的唯一标识序号。

以下代码用于快速删除目标微服务空间。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.DeleteUserDefineRegionRequest;
import com.aliyuncs.edas.model.v20170801.DeleteUserDefineRegionResponse;
public class DeleteUserDefineRegion {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //需要删除的微服务空间所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求，并设置参数。
        DeleteUserDefineRegionRequest request = new DeleteUserDefineRegionRequest();
        //待删除的微服务空间的唯一标识序号。
        request.setId((long) 17926);
        try {
            DeleteUserDefineRegionResponse response = client.getAcsResponse(request);
            System.out.println("RegionName=" + response.getRegionDefine().getRegionName() +
"\nRegionId=" + response.getRegionDefine().getRegionId() + "\nMessage=" + response.getMessag
e());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序，返回的执行结果如下所示：

```
RegionName=APIIdoc
RegionId=cn-hangzhou:doc
Message=success
```

## 结果验证

您在进行创建、编辑或删除微服务空间操作后，都可以调用ListUserDefineRegion接口查询您的自定义微服务空间列表，根据返回结果中的RegionName、RegionId和Description等参数进行判断您是否操作成功。

调用ListUserDefineRegion接口返回的结果如下：

```
{
  "Message": "success",
  "RequestId": "849E908D-E2DE-4798-937B-7210E00FFDD8",
  "UserDefineRegionList": {
    "UserDefineRegionEntity": [
      {
        "Description": "使用API创建的微服务空间",
        "RegionName": "APIdoc2",
        "UserId": "native_****",
        "DebugEnable": true,
        "Id": 17925,
        "RegionId": "cn-hangzhou:doc2",
        "BelongRegion": "cn-hangzhou"
      }
    ]
  },
  "Code": 200
}
```

## 2.3. ECS集群管理

### 2.3.1. 使用控制台创建ECS集群

ECS集群是EDAS为应用提供的ECS部署环境。如果想要将应用托管到EDAS的ECS环境，需要先创建ECS集群。

#### 前提条件

- 如果您要在专有网络VPC中创建ECS集群，请确保已创建了VPC。具体操作，请参见[搭建IPv4专有网络](#)。
- 如果您对资源和服务有较高的安全要求，可以创建微服务空间用于隔离。具体操作，请参见[使用控制台管理微服务空间](#)。

#### 背景信息

EDAS提供两类ECS集群：阿里云ECS集群和非阿里云ECS集群（混合云ECS集群）。

- 阿里云ECS集群：由阿里云的ECS组成的集群。本文介绍如何创建阿里云ECS集群。
- 非阿里云ECS集群：即混合云ECS集群，可以包含阿里云的ECS、本地IDC内的服务器以及其他云服务提供商的服务器。这些服务器通过专线连通，并添加到EDAS的非阿里云ECS集群中进行管理。关于非阿里云ECS集群的更多信息，请参见[创建混合云ECS集群](#)。

#### 操作步骤

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏选择资源管理 > EDAS ECS集群。
3. 在EDAS ECS集群页面顶部菜单栏选择地域，在页面中选择微服务空间，然后单击创建集群。  
微服务空间可以在该页面选择，也可以在创建集群对话框中选择。
  - 如果有环境隔离的需求，请选择您创建的微服务空间。
  - 如果没有环境隔离的需求，可以选择默认微服务空间。
4. 在创建集群对话框中设置集群参数，然后单击创建。



集群创建完成后，并未包含ECS实例，为空集群，需要为集群添加ECS实例。具体操作，请参见[使用控制台扩容ECS集群](#)。

## 2.3.2. 使用控制台扩容ECS集群

### 扩容ECS集群

创建ECS集群后，该集群为空集群，未包含ECS实例，需要添加ECS实例；或者由于业务发展，需要对ECS集群进行扩容。这两种情况下，都需要在ECS集群中导入ECS实例。扩容ECS集群有两种方式：购买ECS扩容和添加已有ECS。

### 前提条件

[使用控制台创建ECS集群](#)

### 背景信息

在不同的网络环境下，为ECS集群添加不同操作系统的实例时，适用的导入方式将不一样。

 **注意** 在EDAS中不支持使用Windows系统的ECS实例。

购买ECS扩容，指基于当前集群中的已有实例规格或基于已购买的ECS实例模板，由EDAS为您当前的ECS集群代购ECS实例。代购实例的计费方式包含包年包月和按量付费，可以根据实际需求选择。

#### 说明

- 经典网络的ECS集群将无法使用ECS启动模板扩容。
- 代购ECS实例时，不能删除ESS标签。
- 集群扩容时代购的按量付费的ECS实例，在缩容时不会关机，也不会释放。但如果手动关机，则对应的ECS实例会被释放。

如果您已经创建了ECS实例，可以将已有实例添加到ECS集群中。添加已有ECS实例包含3种方式：

- 直接导入：无需镜像转化。
- 转化后导入：使用EDAS官方镜像重装系统。重装后，实例中的所有数据都会被删除，并且需要设置新的实例登录密码。ECS实例满足以下任一情况，则不能直接导入。
  - 2017年12月01日之前创建的实例。
  - 向经典网络的集群中导入的经典网络实例。
  - 实例状态不是运行中，例如已停止、启动中或停止中等状态的实例。

 **说明** 上述需要转化导入的实例，如果您不想重装系统，可以选择手动执行脚本后导入。

- 手动执行脚本后导入：在经典网络创建的Ubuntu系统的实例导入至ECS集群时，您需要登录ECS实例并手动执行脚本来完成导入，该方式不会重装ECS实例的系统。

### 购买ECS扩容

### 添加已有ECS

### 购买ECS扩容

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏选择资源管理 > EDAS ECS集群。

3. 在EDAS ECS集群页面的顶部菜单栏选择地域，在页面上方选择微服务空间，然后单击目标ECS集群的集群ID。
4. 在集群详情页面的ECS实例区域的右上角单击购买ECS扩容。
5. 在集群扩容配置向导中选择扩容方式，并完成后续扩容步骤。
  - o 基于现有实例规格购买



- a. 在扩容方式页签选择基于现有实例规格购买，然后单击下一步。
- b. 在购买信息页签设置以下信息，然后单击下一步。

参数	描述
计费方式	包含包年包月和按量付费。
购买时长 (适用于包年包月)	选择包年包月后，在列表中选择ECS实例的购买时长。
购买数量	在复合框中设置要购买的ECS实例数量。
登录密钥	在列表中选择ECS实例的登录密钥。如果没有可选的登录密钥，请前往ECS控制台创建密钥对，相关操作，请参见 <a href="#">创建SSH密钥对</a> 。
服务协议	阅读并勾选《云服务器ECS服务条款》 《镜像商品使用条款》。

- c. 在确认扩容页签，确认扩容信息，然后单击确认扩容。
- o 基于启动实例模板购买

- 选择启动模板和版本，然后单击下一步。
- 在扩容方式页签选择基于实例启动模板购买，然后单击下一步。
- 在购买信息页签设置以下信息，然后单击下一步。

参数	描述
计费方式	包含包年包月和按量付费。
购买时长（适用于包年包月）	选择包年包月后，在列表中选择ECS实例的购买时长。
购买数量	在复合框中设置要购买的ECS实例数量。
服务协议	阅读并勾选《云服务器ECS服务条款》 《镜像商品使用条款》。

- 在确认扩容页签，确认扩容信息，然后单击确认扩容。

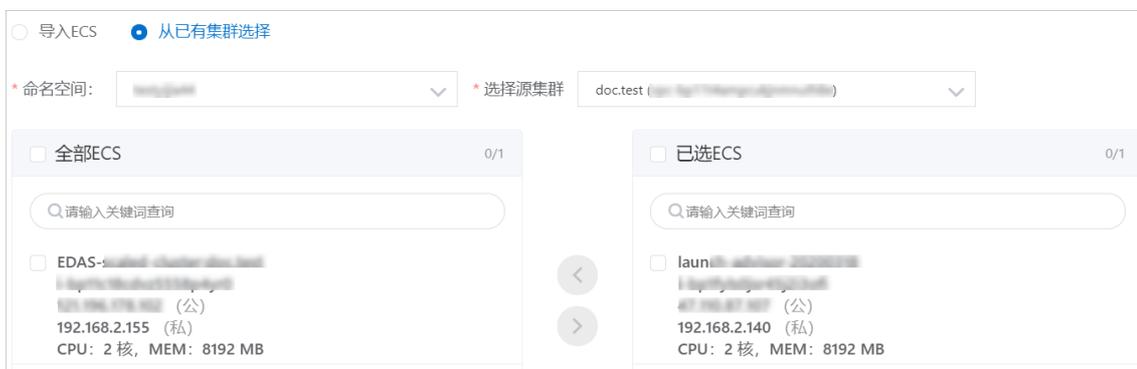
代购ECS实例后，页面上方会提示 已触发自动购买的流程，请查看变更流程获取实时信息 。当实例导入完成后，返回集群详情页面，实例的健康检查显示为运行中，则表示导入成功。

## 添加已有ECS

- 登录EDAS控制台。
- 在左侧导航栏选择资源管理 > EDAS ECS集群。
- 在EDAS ECS集群页面的顶部菜单栏选择地域，在页面上方选择微服务空间，然后单击目标ECS集群的集群ID。
- 在集群详情页面的ECS实例区域的右上角单击，然后选择添加已有ECS。▼
- 在添加ECS实例页面的实例列表中，选择导入方式和ECS实例，然后单击下一步。
  - 导入ECS：微服务空间和导入集群不可配置，您可选择所在微服务空间内的空闲ECS实例导入至集群。



- **从已有集群选择**：选择所在地域内的微服务空间以及源集群，然后从全部ECS列表中选择实例添加到已选ECS列表中。



**说明** 如果没有符合条件的实例，在页面右上角单击创建ECS实例，跳转到云服务器ECS的购买页面购买新的ECS实例。具体操作，请参见[创建ECS实例](#)。

6. 在**准备导入**页面，查看实例对应的导入方式。
  - **直接导入**：在**准备导入**页签，选择实例，单击**确认并导入**。
  - **转化后导入**：在**准备导入**页签，查看选择的实例信息，选中**同意对以上需要导入的实例进行转化**，并**已知转化后原有系统中的数据将会丢失**，重新设置ECS实例的登录密码，然后单击**确认并导入**。
  - **手动执行脚本后导入**：
    - a. 在**准备导入**页签复制脚本内容。
    - b. 以root身份登录需导入的ECS实例。
    - c. 在ECS实例中粘贴并执行脚本。
7. 在**进行导入**页签查看实例的导入状态。  
 当实例导入完成后，返回集群详情页，实例的**健康检查**显示为**运行中**则表示实例导入成功。

## 移除ECS实例

1. 在**集群详情**页面ECS实例区域，单击要移除实例操作列的**移除**。
2. 在**移除ECS实例**对话框中确认要移除的实例信息，单击**移除**。  
 当实例移除成功后，该实例将会从ECS列表中移除。

## 常见问题

在导入实例至集群时无法找到在ECS控制台创建的实例怎么办？

1. 请确认在ECS控制台创建ECS实例的账号和登录EDAS控制台的账号是一致的。
2. 请确认创建的ECS实例和集群属于同一个地域。
3. 登录EDAS控制台，在左侧导航栏选择资源管理 > ECS，在ECS页面右上角单击导入ECS，然后在选择集群和已有云服务器实例页签中单击同步ECS。

 **注意** 请确认您创建的ECS实例的操作系统版本为选择集群和已有云服务器实例页面提示支持的操作系统版本。

4. 如果经过以上排查之后还是无法在EDAS控制台显示您创建的ECS实例，请提工单。

## 相关文档

- [使用控制台创建ECS集群](#)
- [管理ECS集群](#)

## 2.3.3. 使用API创建ECS集群

本文介绍如何使用EDAS提供的Java SDK调用API创建ECS集群。

### 前提条件

在创建ECS集群前，您必须提前完成以下准备工作：

- 安装EDAS的Java SDK。更多信息，请参见[Java SDK接入指南](#)。
- 确定创建ECS集群的地域，假设为 `cn-hangzhou`。
- 创建微服务空间。具体操作，请参见[创建命名空间](#)。  
如果已经存在命名空间，您可以调用 `ListUserDefineRegion` 接口查询用户自定义命名空间列表，获取目标命名空间的 `RegionId`。
- 如果您要在专有网络VPC中创建ECS集群，请先创建VPC。具体操作，请参见[搭建IPv4专有网络](#)。  
如果您已经创建了专有网络VPC，则可以调用 `ListVpc` 接口查询VPC列表，获取VPC实例ID (`VpcId`)，假设为 `vpc-bp1hcg467ekqsv****`。

### 背景信息

EDAS提供两类ECS集群：阿里云ECS集群和非阿里云ECS集群（混合云ECS集群）。

- 阿里云ECS集群：由阿里云的ECS组成的集群。本文介绍如何创建阿里云ECS集群。
- 非阿里云ECS集群：即混合云ECS集群，可以包含阿里云的ECS、本地IDC内的服务器以及其他云服务提供商的服务器。这些服务器通过专线连通，并添加到EDAS的非阿里云ECS集群中进行管理。关于非阿里云ECS集群的更多信息，请参见[创建混合云ECS集群](#)。

### 创建ECS集群

以下代码用于创建ECS集群。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.InsertClusterRequest;
import com.aliyuncs.edas.model.v20170801.InsertClusterResponse;
public class InsertCluster {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //要创建ECS集群的目标地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求, 并设置参数。
        InsertClusterRequest request = new InsertClusterRequest();
        //需要创建ECS集群的目标微服务空间的RegionId, 例如: cn-hangzhou:doc。
        request.setLogicalRegionId("cn-hangzhou:doc");
        //自定义ECS集群名称, 例如: ECSCluster_doc。
        request.setClusterName("ECSCluster_doc");
        //设置集群类型, 设置为2, 即代表ECS集群。
        request.setClusterType(2);
        //设置集群网络类型, 2代表专有网络VPC, 1代表经典网络。
        request.setNetworkMode(2);
        //VPC实例ID, 当选择专有网络VPC时, 此参数必填。
        request.setVpcId("vpc-bplhcg467ekqsv****");
        try {
            InsertClusterResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage() + "\nClusterName=" + response.getCluster().getClusterName() + "\nClusterId=" + response.getCluster().getClusterId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序, 返回的执行结果如下:

```
Message=success
ClusterName=ECSCluster_doc
ClusterId=369d06d7-450b-4f3d-bf75-9536fcd9****
```

## 结果验证

在创建ECS集群后, 您可以调用 `ListCluster` 接口查询集群列表, 根据返回结果中的集群名称 (ClusterName) 参数值判断您是否操作成功。

调用 `ListCluster` 接口返回的结果如下:

```
{
  "ClusterList": {
    "Cluster": [
      {
        "OversoldFactor": 1,
        "NodeNum": 0,
        "ResourceGroupId": "rg-acfm3umruf2****",
        "ClusterId": "369d06d7-450b-4f3d-bf75-9536fcd9****",
        "CreateTime": 1618558401855,
        .....
        "VpcId": "vpc-bp1hcg467ekqsv0zr****",
        "UpdateTime": 1618558401855,
        "ClusterName": "ECSCluster_doc",
        "RegionId": "cn-hangzhou:doc",
        "IaasProvider": "ALIYUN",
        "CpuUsed": 0
      }
    ]
  },
  "Message": "success",
  "RequestId": "65C65D19-36BF-40C1-A3D0-23902D2AF49C",
  "Code": 200
}
```

## 2.3.4. 使用API扩容ECS集群

本文介绍如何使用EDAS提供的Java SDK调用API扩容ECS集群。

### 前提条件

在扩容ECS集群前，您必须提前完成以下准备工作：

- 安装EDAS的Java SDK。更多信息，请参见[Java SDK接入指南](#)。
- 确定创建ECS集群的地域。
- 创建ECS实例。假设为*i-bp13o01lzmbvhs1\*\*\*\**和*i-bp13o01lzmbvhs1\*\*\*\**。
- 创建ECS集群，具体操作，请参见[使用API创建ECS集群](#)。  
如果您已经创建了ECS集群，则可以调用ListCluster接口查询集群列表，获取目标ECS集群的集群ID (ClusterId)，假设为*369d06d7-450b-4f3d-bf75-9536fcd9\*\*\*\**。

### 背景信息

支持调用InstallAgent接口或InsertClusterMember接口向ECS集群导入ECS实例，推荐使用InstallAgent接口，不推荐使用InsertClusterMember接口。

- 调用InsertClusterMember接口导入ECS实例时，会重装ECS实例的操作系统，重装后，实例中所有数据都会被删除并且需要重新设置实例登录密码。因此不推荐使用该接口导入ECS实例。
- 调用InstallAgent接口导入ECS实例时，不会重装ECS实例的操作系统。因此推荐使用该接口导入ECS实例。

### 购买ECS实例扩容

购买ECS实例扩容，支持以下两种方式：

- 基于现有实例购买：需要指定现有实例的实例ID。

```
//基于现有实例规则购买，ECS实例的实例ID。  
request.setTemplateInstanceId("i-bp13o01lzmbvhs1****");
```

- 基于实例启动模板购买：需要设置ECS启动模板ID和模板版本。

```
//ECS启动模板ID。  
request.setTemplateId("lt-****hy9s2");  
//ECS启动模板版本，-1代表默认的启动模板版本。  
request.setTemplateVersion("-1");
```

#### 🔍 说明

实例启动模板用于快速创建实例的模板。模板中包含了您自主选择的用于创建实例的配置信息，所有配置为可选，能够满足不同场景的个性化需求。

实例启动模板不支持修改，但可以创建多个版本，每个版本可以配置不同的参数，通过版本管理体现实例配置的演进过程。您可以使用模板任意一个版本创建实例。

更多关于实例启动模板和模板版本的信息，请参见[创建实例启动模板](#)和[创建实例启动模板的新版本](#)。

以下代码用于在ECS集群购买ECS实例。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.ScaleoutApplicationWithNewInstancesRequest;
import com.aliyuncs.edas.model.v20170801.ScaleoutApplicationWithNewInstancesResponse;
public class ScaleoutApplicationWithNewInstances {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //集群所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求，并设置参数。
        ScaleoutApplicationWithNewInstancesRequest request = new ScaleoutApplicationWithNew
InstancesRequest();
        //需要扩容的ECS集群的集群ID。
        request.setClusterId("369d06d7-450b-4f3d-bf75-9536fcd9****");
        //基于现有实例规则购买，ECS实例的实例ID。如果是基于实例模板购买，则无需设置此参数，需要设置ECS
启动模板ID和模板版本。
        request.setTemplateInstanceId("i-bp13o01lzmbvhs1****");
        //购买的ECS实例数量。
        request.setScalingNum(2);
        //购买ECS实例的计费方式，PrePaid表示包年包月，PostPaid表示按量计费。
        request.setInstanceChargeType("PrePaid");
        //包年包月计费方式的时长单位，Month表示月，Week表示周。
        request.setInstanceChargePeriodUnit("Month");
        //购买资源的时长，时长单位为Month时可设置为：1~12和12的整数倍；时长单位为Week时可设置为：1~4
。
        request.setInstanceChargePeriod(12);
        //是否开启自动续费，true表示自动续费，false表示不自动续费。
        request.setAutoRenew(true);
        //单次自动续费的续费时长，时长单位为Month时可设置为：1~12和12的整数倍；时长单位为Week时可设置
为：1~4。
        request.setAutoRenewPeriod(12);
        try {
            ScaleoutApplicationWithNewInstancesResponse response = client.getAcsResponse(re
quest);
            System.out.println("Message=" + response.getMessage() + "\nInstanceIds=" + resp
onse.getInstanceIds());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序，返回的执行结果如下：

```
Message=success
InstanceIds=[i-bp1ilg5114r054eo****, i-bp1ilg5114r054eo****]
```

## 添加已有ECS实例扩容

以下代码用于向ECS集群导入已有ECS实例。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.InstallAgentRequest;
import com.aliyuncs.edas.model.v20170801.InstallAgentResponse;
public class InstallAgent {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //集群所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求，并设置参数。
        InstallAgentRequest request = new InstallAgentRequest();
        //目标ECS集群的集群ID。
        request.setClusterId("369d06d7-450b-4f3d-bf75-9536fcd9****");
        //需要导入的ECS实例ID，多个实例间以半角逗号(,)分隔。
        request.setInstanceIds("i-bp13o011zmbvsh1****,i-bp13o011zmbvsh1****");
        try {
            InstallAgentResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序，返回的执行结果如下：

```
Message=success
```

## 结果验证

在为ECS集群购买ECS实例或添加已有ECS实例后，都可以调用[InsertClusterMember](#)接口查询集群中ECS实例列表，根据返回结果中的ECS实例ID (EcsId) 判断扩容操作是否成功。

```
{
  "Message": "success",
  "RequestId": "C326EF8E-048E-45C8-BE87-18D512000573",
  "ClusterMemberPage": {
    "ClusterMemberList": {
      "ClusterMember": [
        {
          "Status": 1,
          "ClusterId": "369d06d7-450b-4f3d-bf75-9536fcd9****",
          "CreateTime": 1618827574965,
          "UpdateTime": 1618827655646,
          "EcuId": "08da39f3-257c-44c4-abce-9ce31ec1****",
          "ClusterMemberId": "49eeb7fa-7c97-436f-b87f-a875a8b8****",
          "EcsId": "i-bp1ilg5114r054eo****"
        },
        {
          "Status": 1,
          "ClusterId": "369d06d7-450b-4f3d-bf75-9536fcd9****",
          "CreateTime": 1618827574957,
          "UpdateTime": 1618827655645,
          "EcuId": "8287bcf6-cde3-4377-8906-086d2c32****",
          "ClusterMemberId": "f5b30a72-7137-4b5c-8d21-a2eac2f0****",
          "EcsId": "i-bp1ilg5114r054eo****"
        }
      ]
    },
    "PageSize": 20000,
    "CurrentPage": 1,
    "TotalSize": 2
  },
  "Code": 200
}
```

## 2.3.5. 管理ECS集群

### 管理ECS集群

本文介绍了如何查看、管理和删除ECS集群，以及对集群中的ECS实例和应用的一些管理操作。

### 查看集群列表

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏选择资源管理 > EDAS ECS集群。
3. 在EDAS ECS集群页面顶部菜单栏选择地域，在页面上方选择微服务空间，在集群类型列表中选择ECS集群，查看该微服务空间中的ECS集群列表。

集群列表中包含当前阿里云账号，或当前RAM用户、EDAS子账号所属阿里云账号下的所有ECS集群及其基本信息。

- 您可以在页面左上角单击创建集群来创建ECS集群，具体操作，请参见[使用控制台创建ECS集群](#)。
- 您可以在集群列表的操作列单击删除来删除集群。

 **注意** 当集群中存在实例时，该集群将无法被删除。

- 您可以单击集群ID进入集群详情页面，在该页面您可查看集群信息、管理ECS实例和查看部署在该集群中的应用列表。

## 查看集群信息

1. 在EDAS ECS集群页面，单击具体集群ID。
2. 在集群详情页面查看该集群的**集群信息**、**ECS实例**和**应用列表**。
  - **集群信息**区域展示该集群的基本信息，例如集群归属、集群ID、集群名称、所属微服务空间、集群类型、所属的VPC ID、网络类型和集群状态等。  
您还可以在**集群详情**对集群进行管理。
    - 在**集群详情**页面右上角单击**删除**可删除该集群。

 **注意** 当集群中存在ECS实例时，无法删除。

- 在**集群信息**区域单击**VPC ID**可跳转至集群所归属的**专有网络详情**页面。
- 在**集群信息**区域右上角单击**查看详情**可查看该集群的详细信息。
- 在**集群信息**区域右上角单击**编辑**可修改集群的描述信息。
- **ECS实例**区域展示该集群中的ECS实例及其基本信息。您可以对当前集群中的ECS实例进行管理，具体操作，请参见**管理集群中的ECS实例**。
- **应用列表**区域展示了集群内包含的应用列表，您可以查看应用名称、JDK版本、应用运行环境、实例总数、运行中的实例数和负责人等信息。单击应用名称，可以进入该应用的基本信息页面。

## 管理集群中的ECS实例

在**集群详情**页面的**ECS实例**区域可以查看该集群中包含的ECS实例列表及ECS实例的概要信息。您还可以在**ECS实例**区域管理ECS实例。

- **购买ECS扩容**  
在**ECS实例**区域右上角单击**购买ECS扩容**或在目标ECS实例的操作列单击**购买相同配置**，来为该集群代购ECS实例。相关操作，请参见**购买ECS扩容**。
- **添加已有ECS**  
在**ECS实例**区域右上角选择**添加已有ECS**，将已有ECS实例添加到当前集群。具体操作，请参见**添加已有ECS**。
- **查看ECS实例详情**  
在**ECS实例**区域单击ECS实例ID，跳转至ECS控制台的**实例详情**页面，查看ECS实例的详细信息。
- **查看ECS实例部署的应用详情**  
在**ECS实例**区域已部署了应用的ECS实例的操作列单击**查看详情**，查看部署应用的基本信息。在已部署应用列单击具体应用名称，可以跳转到部署在本ECS实例上的应用的基本信息页面。
- **移除ECS实例**  
当ECS实例内没有已部署应用后，您可在集群内移除ECS实例。
  - 在没有已部署应用的ECS实例的操作列单击**移除**，然后在**移除ECS实例**对话框单击**移除**。
  - 在ECS实例列表左侧批量选中没有已部署应用并且在运行中的ECS实例，在ECS实例区域下方单击**批量移除**，然后在**确认**对话框单击**确定**。
- **将按量付费实例转为包年包月**  
集群中包含按量付费的ECS实例，可批量转付费方式为包年包月。
  - i. 在ECS实例列表左侧批量选中按量付费的ECS实例，在ECS实例区域下方单击**按量付费转包年包月**。
  - ii. 在**按量付费转包年包月**页面单击**批量更改**。

- iii. 在**按量付费转包年包月**对话框选择包年包月的时长；如果要将数据盘转为包年包月磁盘，请选中**转为包年包月磁盘**。然后单击**确定**。
- **转移ECS实例至其他集群**  
集群中的ECS实例内未包含部署应用后，可批量转移至其他集群内。
    - i. 在ECS实例列表左侧批量选中未部署应用的ECS实例，在ECS实例区域下方单击**转移ECS实例**。
    - ii. 在**选择目标集群**页签，选择微服务空间和目标集群，单击**下一步**。
    - iii. 在**准备导入**页签，单击**确认并导入**。

 **说明** 如果该实例不能直接导入，则需要选中同意对以上需要导入的实例进行转化，并已知转化后原有系统中的数据将会丢失，然后设置ECS实例的新的登录密码。

## 删除集群

1. 查看集群列表，在EDAS ECS 集群页面的集群列表中查看目标集群的**实例数**是否为0。查看集群列表的具体操作，请参见[查看集群列表](#)。
  - 如果实例数为0，执行下一步。
  - 如果实例数不为0，需要先移除集群中的ECS实例。具体操作，请参见[管理集群中的ECS实例](#)中的移除ECS实例。
2. 在EDAS ECS 集群页面的集群列表中目标集群的操作列单击**删除**。
3. 在**确认对话框**中单击**确定**。

## 相关文档

- [使用控制台创建ECS集群](#)
- [使用控制台扩容ECS集群](#)

## 2.3.6. 使用控制台管理ECS集群中的应用实例分组

实例分组，指的是将应用中ECS实例进行分组，以便在不同分组中部署不同版本的应用。您可以通过实例分组进行流量管理和灰度发布。

### 实例分组简介

例如：应用 *itemcenter* 一共部署了10个ECS实例，将这些实例分为两个组，分别为**默认分组**和**Beta分组**。其中**默认分组**包含6个实例，**Beta分组**包含4个实例。此时应用就拥有了两个实例分组，可以分别部署不同的应用版本。

#### 说明

- 每个应用在创建的时候，EDAS默认会为应用创建“默认分组”，该分组不可删除。
- 如果没有流量管理或灰度发布等多版本部署需求，通常使用这个“默认分组”就足够了，不需要新建其他分组。

## 创建分组

如果某个应用发布了一个新版本，为了不影响生产环境的流量，又希望对新版本进行验证，通常会采用灰度发布的方式，这时就需要为该应用创建新的分组。

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏中单击**应用列表**，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在**应用列**

表页面单击具体的应用名称。

3. 在应用详情页面单击**实例部署信息**页签，然后单击页面右上方的**创建新分组**。
4. 在**创建分组**对话框输入**分组名**，选择**部署包版本**，然后单击**创建**。  
分组成功创建后，页面上方会出现**创建分组成功**的提示。

## 添加实例

创建分组后，可以通过**应用扩容**和**更换分组**两种方式为新分组添加实例。具体操作如下：

- 通过**应用扩容**为分组添加实例，请参见[扩容部署在ECS集群中的应用](#)。
- 通过**更换分组**将实例添加到新分组。
  - i. 在**实例部署信息**页签中实例的操作列单击**更换分组**。
  - ii. 在**更换分组**对话框内，选择**目标分组**，确认信息和选择部署策略，然后单击**确定**。  
分组、实例和应用版本之间存在一定的关联关系，且比较复杂，所以下面分别说明。
    - 如果新分组中**没有部署应用版本**，当前实例的应用版本（V1.0）将作为新分组的默认部署版本。
    - 如果新分组中**已经部署了应用版本（V2.0）**，与当前实例的应用版本（V1.0）不一致，则**更换分组**对话框中会显示这两个版本，您可以根据实际需求选择不同的部署策略。
      - 如果想将新分组中的应用版本（V2.0）部署到该实例，则选择采用**目标分组重新部署当前实例**。
      - 如果想继续使用该实例中已经部署的应用版本（V1.0），则选择**仅更换分组，不重新部署**。

 **说明** 当实例的部署包版本与该实例所属分组的部署包版本不一致时，会产生提示信息。

## 查看分组

1. 登录**EDAS控制台**。
2. 在左侧导航栏中单击**应用列表**，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在**应用列表**页面单击具体的应用名称。
3. 在应用详情页面单击**实例部署信息**页签，查看该应用的分组信息及各分组的部署包版本。  
分组信息说明：
  - 如果该分组已经部署过应用，则会在分组名之后显示最近一次部署的部署包版本。
  - 应用的实例会根据实例所属分组展示。

## 设置分组的JVM和Tomcat参数

您可以针对分组设置JVM和Tomcat参数，也可以设置应用级别的JVM和Tomcat参数。具体操作，请参见[设置部署在ECS集群中的应用](#)。

1. 在应用详情页面的**实例部署信息**页签中单击分组右侧的**分组设置**。
2. 在下拉列表中单击**JVM**或**Tomcat**。
3. 在**分组设置（JVM或Tomcat）**对话框中设置JVM或Tomcat参数，然后单击**设置JVM参数**或**配置Tomcat**。

JVM和Tomcat参数说明，请参见[设置部署在ECS集群中的应用](#)。

## 绑定实例启动模板

实例启动模板用于快速创建ECS实例，您可以在应用分组中绑定实例启动模板，这样可以为应用分组快速扩容ECS实例。

1. 在应用详情页面的实例部署信息页签中单击分组右侧的分组设置。
2. 在下拉列表中单击实例启动模板。
3. 在分组设置（实例启动模板）对话框中选中绑定启动模板并选择启动模板和版本，单击修改。

## 为分组绑定SLB

为分组绑定SLB的使用场景及详细操作步骤，请参见[负载均衡概述](#)。

## 删除分组

如果当前分组中已经没有实例，且不需要再使用该分组时，可以删除分组。删除操作不可恢复，请谨慎操作。

1. 在应用详情页面的实例部署信息页签中，单击分组对应的删除分组按钮。
2. 在删除分组对话框中，单击删除。

## 2.3.7. 使用API管理ECS集群中的应用实例分组

本文介绍如何使用EDAS提供的Java SDK调用API管理ECS集群中的应用实例分组。

### 前提条件

在管理ECS集群中的应用实例分组前，您必须提前完成以下准备工作：

- 安装EDAS的Java SDK。更多信息，请参见[Java SDK接入指南](#)。
- 确定应用所在地域，假设为 `cn-hangzhou`。
- 创建应用实例分组前，请先调用 `ListHistoryDeployVersion` 接口查询应用部署的历史版本列表，获取目标版本的唯一标识（`PackageVersion.Id`）。
- 更换分组前，请先调用 `ListApplicationEcc` 接口查询应用的ECC信息，获取目标ECS实例的ECC ID（`EccId`）。
- 创建ECS实例。假设为 `i-bp13o01lzmbvhs1****` 和 `i-bp13o01lzmbvhs1****`。
- 创建ECS集群，具体操作，请参见[使用API创建ECS集群](#)。  
如果您已经创建了ECS集群，则可以调用 `ListCluster` 接口查询集群列表，获取目标ECS集群的集群ID（`ClusterId`），假设为 `369d06d7-450b-4f3d-bf75-9536fcd9****`。

### 背景信息

实例分组，指的是将应用中ECS实例进行分组，以便在不同分组中部署不同版本的应用。您可以通过实例分组进行流量管理和灰度发布。

例如：应用 `itemcenter` 一共部署了10个ECS实例，将这些实例分为两个组，分别为默认分组和Beta分组。其中默认分组包含6个实例，Beta分组包含4个实例。此时应用就拥有了两个实例分组，可以分别部署不同的应用版本。

#### 说明

- 每个应用在创建的时候，EDAS默认会为应用创建“默认分组”，该分组不可删除。
- 如果没有流量管理或灰度发布等多版本部署需求，通常使用这个“默认分组”就足够了，不需要新建其他分组。

## 创建应用实例分组

以下代码适用于在ECS环境为指定应用创建应用实例分组。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.InsertDeployGroupRequest;
import com.aliyuncs.edas.model.v20170801.InsertDeployGroupResponse;
public class InsertDeployGroup {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //应用所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求, 并设置参数。
        InsertDeployGroupRequest request = new InsertDeployGroupRequest();
        //指定应用, 在该应用中创建应用实例分组。
        request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");
        //自定义新建的应用实例分组名称, 支持设置为中文、字母、数字、下划线 ( _ ) 和半角句号 ( . ), 最长64
个字符。
        request.setGroupName("Beta");
        //应用分组关联的初始化部署包版本ID, 可以调用接口ListHistoryDeployVersion接口获取。
        request.setInitPackageVersionId("3fc52328-8746-4422-a742-94e0cfc7****");
        try {
            InsertDeployGroupResponse response = client.getAcsResponse(request);
            System.out.println("GroupName=" + response.getDeployGroupEntity().getGroupName(
) + "\nId=" + response.getDeployGroupEntity().getId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序, 返回的执行结果如下所示:

```
GroupName=Beta
Id=941be68c-4aac-48a1-88fe-c9ad1502****
```

## 更换ECS实例的实例分组

 **说明** 更换应用中ECS实例到其他应用实例分组时, 只能一次更换一个ECS实例到其他分组。

以下代码适用于在ECS环境将指定应用的ECS实例更换到其他应用实例分组。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.ChangeDeployGroupRequest;
import com.aliyuncs.edas.model.v20170801.ChangeDeployGroupResponse;
public class ChangeDeployGroup {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //应用所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求, 并设置参数。
        ChangeDeployGroupRequest request = new ChangeDeployGroupRequest();
        //指定应用, 将应用中的ECS实例更换应用实例分组。
        request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");
        //指定要切换应用分组的ECS实例的ECC ID。可以调用ListApplicationEcc接口查询应用的ECC ID。
        request.setEccInfo("4009a824-ce33-4ba0-9ca2-346249a9****");
        //目标应用分组名称, 如果需要更换到默认分组, 则设置为_DEFAULT_GROUP。
        request.setGroupName("Beta");
        //ECC的部署包版本和应用分组的部署包版本不一致时是否强制更换。true表示强制, false表示不强制。
        request.setForceStatus(true);
        try {
            ChangeDeployGroupResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage() + "\nRequestId=" + respon
se.getRequestId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序, 返回的执行结果如下所示:

```
Message=success
ChangeOrderId=369d06d7-450b-4f3d-bf75-9536fcd9****
```

## 为应用实例分组设置JVM

以下代码适用于在ECS环境为应用实例分组设置JVM。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.UpdateJvmConfigurationRequest;
import com.aliyuncs.edas.model.v20170801.UpdateJvmConfigurationResponse;
public class UpdateJvmConfiguration {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //应用所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求, 并设置参数。
        UpdateJvmConfigurationRequest request = new UpdateJvmConfigurationRequest();
        //指定应用, 为应用中的应用实例分组设置JVM参数。
        request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");
        //指定要设置JVM参数的应用实例分组的ID。可以调用ListDeployGroup接口查询应用实例分组ID。
        request.setGroupId("941be68c-4aac-48a1-88fe-c9ad1502****");
        //自定义参数。
        request.setOptions("-Dproperty=value");
        //初始化堆内存大小, 单位MB。
        request.setMinHeapSize(500);
        //持久代内存大小, 单位MB。
        request.setMaxPermSize(500);
        //最大堆内存大小, 单位MB。
        request.setMaxHeapSize(500);
        try {
            UpdateJvmConfigurationResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序, 返回的执行结果如下所示:

```
Message=success
```

## 为应用实例分组设置Tomcat

以下代码适用于在ECS环境为应用实例分组设置Tomcat。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.UpdateContainerConfigurationRequest;
import com.aliyuncs.edas.model.v20170801.UpdateContainerConfigurationResponse;
public class UpdateContainerConfiguration {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //应用所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求, 并设置参数。
        UpdateContainerConfigurationRequest request = new UpdateContainerConfiguratio
st();
        //指定应用, 为应用中的应用实例分组设置Tomcat。
        request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");
        //指定要设置Tomcat参数的应用实例分组的ID。可以调用ListDeployGroup接口查询应用实例分组ID。
        request.setGroupId("941be68c-4aac-48a1-88fe-c9ad1502****");
        //Tomcat的ContextPath, 支持空字符串、null-war包名称、ROOT-根目录或其他非空自定义字符串。
        request.setContextPath("");
        //应用端口。
        request.setHttpPort(8080);
        //最大线程数。
        request.setMaxThreads(20);
        //URI编码方式, 支持ISO-8859-1、GBK、GB2312和UTF-8。
        request.setURIEncoding("ISO-8859-1");
        //useBodyEncodingForURI是否启用, true表示启用, false表示不启用。
        request.setUseBodyEncoding(true);
        try {
            UpdateContainerConfigurationResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序, 返回的执行结果如下所示:

```
Message=success
```

## 删除应用实例分组

**说明** 如果应用实例分组中还存在ECS实例，则无法删除分组。如果您确认要删除分组，请先将ECS实例全部更换分组，具体操作，请参见[更换ECS实例的实例分组](#)。

以下代码适用于在ECS环境为应用删除指定应用实例分组。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.DeleteDeployGroupRequest;
import com.aliyuncs.edas.model.v20170801.DeleteDeployGroupResponse;
public class DeleteDeployGroup {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //应用所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求，并设置参数。
        DeleteDeployGroupRequest request = new DeleteDeployGroupRequest();
        //指定应用。
        request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");
        //应用中需要删除的应用实例分组的名称。可以调用ListDeployGroup接口查询应用实例分组ID。
        request.setGroupName("Beta");
        try {
            DeleteDeployGroupResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage() + "\nData=" + response.ge
tData());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序，返回的执行结果如下所示：

```
Message=success
Data=1
```

## 结果验证

请根据您的操作选择对应的验证方式。

- 创建应用实例分组。  
您可以调用[ListDeployGroup](#)接口查询应用的实例分组列表，根据返回结果中的 *GroupName* 参数判断创建应用实例分组是否成功。  
调用 *ListDeployGroup* 接口返回的结果如下：

```

{
  "DeployGroupList": {
    "DeployGroup": [
      {
        "GroupName": "Beta",
        "AppId": "6bbc57a2-a017-4bec-b521-49a15bd3****",
        .....
        "GroupId": "941be68c-4aac-48a1-88fe-c9ad1502****"
      }
    ]
  },
  "Message": "success",
  "RequestId": "E8DC196C-F446-4465-A2D1-503728601680",
  "Code": 200
}

```

- 更换ECS实例的实例分组。

您可以调用 `ListApplicationEcc` 接口查询应用的ECC信息，根据返回结果中的 `GroupId` 和 `EccId` 参数判断将ECS实例更换分组是否成功。

调用 `ListApplicationEcc` 接口返回的结果如下：

```

{
  "EccInfoList": {
    "EccInfo": [
      {
        "EcuId": "8287bcf6-cde3-4377-8906-086d2c32****",
        "GroupId": "941be68c-4aac-48a1-88fe-c9ad1502****",
        "AppId": "6bbc57a2-a017-4bec-b521-49a15bd3****",
        "EccId": "02cdcdfa-f22b-45e6-8a0f-a738c766****",
        .....
      }
    ]
  },
  "Message": "success",
  "RequestId": "763D5FDF-3EC0-4F73-9FB5-A34E22BA9410",
  "Code": 200
}

```

- 为应用实例分组设置JVM。

您可以调用 `GetJvmConfiguration` 接口查询应用实例分组的JVM设置，根据返回结果中的JVM信息判断是否操作成功。

调用 `GetJvmConfiguration` 接口返回结果如下：

```

{
  "Message": "success",
  "RequestId": "8EEB65B1-FD4E-4279-A221-09BFB34B9BBF",
  "JvmConfiguration": {
    "Options": "-Dproperty=value",
    "MaxPermSize": 500,
    "MaxHeapSize": 500,
    "MinHeapSize": 500
  },
  "Code": 200
}

```

- 为应用实例分组设置Tomcat。

调用 `GetContainerConfiguration` 接口查询应用实例分组的Tomcat设置，根据返回结果中的Tomcat信息判断是否操作成功。

调用 `GetContainerConfiguration` 接口返回的结果如下：

```
{
  "Message": "success",
  "RequestId": "B04B77C9-3978-4A3D-A628-72D385CB7DD3",
  "Code": 200,
  "ContainerConfiguration": {
    "HttpPort": 8080,
    "ContextPath": "ROOT",
    "UseBodyEncoding": true,
    "URIEncoding": "ISO-8859-1",
    "MaxThreads": 20
  }
}
```

- 删除应用实例分组。

您可以调用 `ListDeployGroup` 接口查询应用的实例分组列表，根据返回结果中的 `GroupName` 参数判断删除应用实例分组是否成功。

调用 `ListDeployGroup` 接口返回的结果如下：

```
{
  "DeployGroupList": {
    "DeployGroup": [
      {
        "GroupName": "_DEFAULT_GROUP",
        "AppId": "6bbc57a2-a017-4bec-b521-49a15bd3****",
        .....
        "GroupId": "941be68c-4aac-48a1-88fe-c9ad1502****"
      }
    ]
  },
  "Message": "success",
  "RequestId": "E8DC196C-F446-4465-A2D1-503728601680",
  "Code": 200
}
```

## 2.4. ECS实例管理

### 2.4.1. 创建ECS实例

在ECS环境中，应用需要部署到ECS实例上。如果您需要将应用部署到EDAS的ECS环境（集群），需要先创建ECS实例。

#### 前提条件

在创建ECS实例前，请先确认您要创建实例的网络。

- 如果您在2017年6月14日17:00（UTC+8）之后，第一次购买ECS实例，不能再选择经典网络，只能选择专有网络（VPC）。在此之前创建的经典网络的ECS实例可以继续使用。
- 在VPC创建ECS实例，需要选择VPC，且选择VPC之后，该ECS实例不能再转移到其它VPC中。所以请确认您要创建ECS实例的VPC。如果没有VPC，请先创建VPC，具体操作，请参见[搭建IPv4专有网络](#)。

## 操作步骤

1. 登录EDAS控制台。
2. 在左侧导航栏中选择资源管理 > ECS。
3. 在ECS页面顶部菜单栏选择地域，在页面上方选择为服务空间（可选），然后在页面右上角单击创建实例。
4. 在ECS购买向导页面购买您需要的ECS实例。具体操作，请参见[使用向导创建实例](#)。

 **注意** 在EDAS中使用ECS实例需要在ECS实例上安装EDAS Agent。在购买ECS实例时，通过选择EDAS基础镜像自动安装EDAS Agent。

- i. 在镜像一栏中，选择镜像市场，然后单击从镜像市场获取更多选择（含操作系统）。
- ii. 在镜像市场对话框的搜索框中输入EDAS，单击搜索。
- iii. 在搜索结果中选择EDAS JAVA环境（普通ECS）和版本（默认选择当前最新版本）和单击使用。

创建实例一般需要1~5分钟，请您耐心等待。

## 结果验证

单击管理控制台返回云服务器管理控制台，单击刷新按钮，新建的ECS实例状态变为运行中，表示实例创建成功。

## 2.4.2. 为ECS实例安装EDAS Agent

EDAS Agent（简称Agent）是安装在ECS上，用于EDAS服务集群与部署在ECS上的应用程序之间进行通信的Daemon程序。所以，在EDAS中使用ECS实例部署应用前，需要为ECS实例安装EDAS Agent。

### 背景信息

EDAS只会选择安装了EDAS Agent的ECS进行应用的部署（包括第一次安装、扩容）。EDAS的应用实例，也是指安装了Agent并且部署了应用的ECS。所以在购买ECS之后，如果想在EDAS中使用，需要先为ECS实例安装Agent。

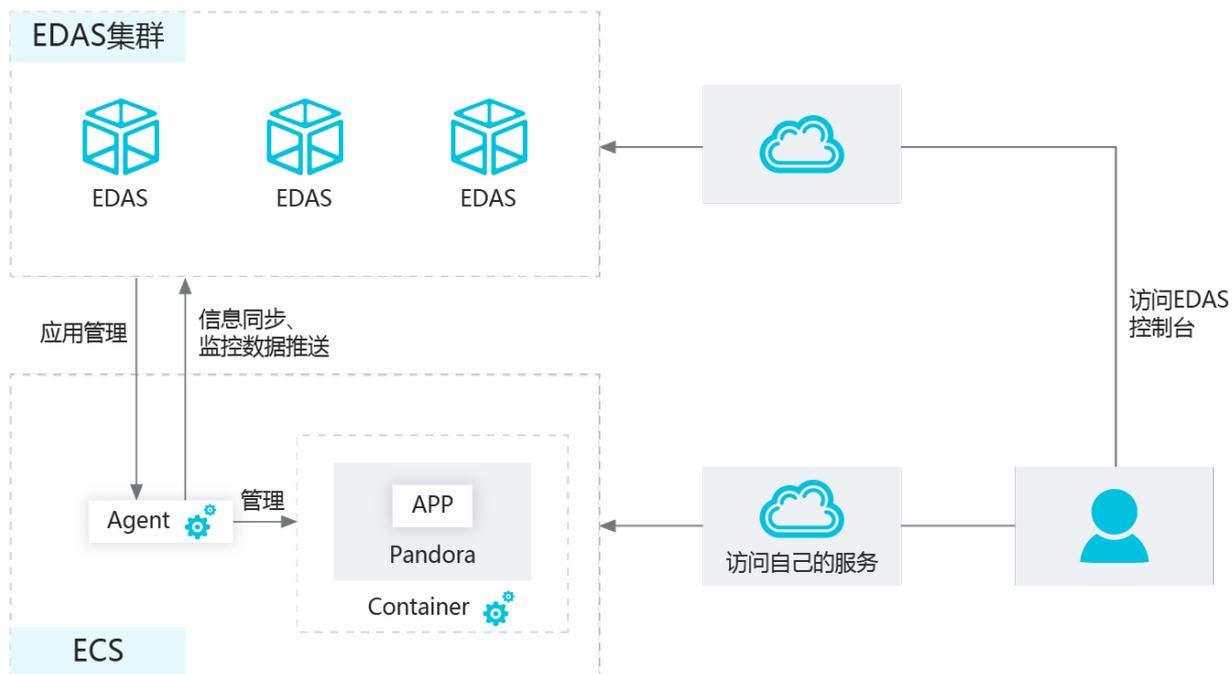
EDAS Agent主要承担以下的角色：

- 应用生命周期管理：包括应用部署、启动、停止等。
- 状态回报：包括应用存活状态、健康检查状态、Ali-Tomcat容器状态等。
- 信息获取：如获取ECS和容器的监控信息。

Agent除了完成以上基于应用的管控功能，还负责EDAS控制台与您的应用程序之间通信。简单的说，一个服务是否在某台ECS上正确及时的发布，获取这个信息就需要Agent的参与和协调。

EDAS Agent目前支持CentOS 7.0以上64位系统。

 **说明** 上述Agent所涉及的功能对您都是透明的，您只需要安装Agent即可。



EDAS提供了3种安装EDAS Agent的方式：

- 购买ECS实例时选择EDAS基础镜像自动安装Agent
- 通过导入ECS自动安装Agent
- 使用命令脚本手动安装Agent

**注意**

- Agent中默认安装JDK 8，如果需要使用JDK 7或者其它版本，请选择通过命令脚本安装。
  - 该脚本需要先以root身份登录您的ECS。
  - 该脚本可以重复执行，重复执行的结果是覆盖安装，所以Agent没有提供单独的升级脚本，即当需要升级Agent时，需要执行和安装时同样的脚本。
  - 目前不同地域的ECS服务器所使用的安装脚本是不同的，在选择安装脚本之前一定要切换到ECS所对应的地域，然后再单击安装Agent按钮。
- 不同的安装方式或安装过程中对于镜像、集群的选择，会形成不同的Agent状态，从而决定在该ECS能够创建的应用类型。请您关注不同安装方式中的相关说明。

### 购买ECS实例时选择EDAS基础镜像自动安装Agent

最便捷的安装Agent的方式，是在购买ECS的时候直接使用EDAS的基础镜像，即可一步完成Agent安装。

**注意** 该方法会导致磁盘格式化，如果不希望磁盘格式化，建议使用命令脚本手动安装Agent。具体操作，请参见[使用命令脚本手动安装Agent](#)。

1. 登录EDAS控制台。
2. 左侧的导航栏中选择资源管理 > ECS。
3. 在ECS页面右上角单击创建实例。
4. 在ECS购买向导页面的镜像区域中，选择镜像市场，然后单击从镜像市场获取更多选择（含操作系统）。

5. 在镜像市场对话框的搜索框中输入EDAS，单击搜索。
6. 在搜索结果中选择EDAS JAVA环境（普通ECS），版本默认选择为当前最新（不建议选择其他旧版本），单击使用。
7. 根据购买向导提示，完成ECS购买。

## 通过导入ECS自动安装Agent

如果在购买ECS实例时，没有选择安装EDAS的基础镜像EDAS JAVA环境（普通ECS），可以在EDAS控制台通过导入ECS安装EDAS Agent。导入ECS的过程分为直接导入和转化后导入。

ECS实例满足以下任一情况，均不能直接导入。

- 2017年12月01日之前创建的实例
- 向经典网络的集群中导入的经典网络的实例
- 实例处于已停止、启动中或停止中
- Windows系统实例或不支持简单Shell命令的实例
- 非ECS集群间导入的实例

**注意** 如果需要镜像转化，则该ECS会被格式化，如果不希望磁盘格式化，建议使用命令脚本手动安装Agent。

1. 登录EDAS控制台。
2. 在左侧的导航栏中选择资源管理 > ECS。
3. 在ECS页面顶部菜单栏选择地域，在页面上方选择微服务空间（可选），然后在页面右上角单击导入ECS。
4. 在导入ECS向导页面的选择集群和已有云服务器实例页签中，选择微服务空间和选择导入集群，并在实例列表中选中要导入的ECS实例，单击下一步。



### 说明

- 如果还未创建目标导入集群，在选择导入集群列表右侧单击创建集群来创建新的集群。
- 如果此时想要使用手动安装，可以在页面右上角单击切换成手动安装，跳转到单机手动安装EDAS Agent页面，使用命令脚本手动安装Agent。

5. 在准备导入页签查看选择的实例信息。
  - 能够直接导入的实例在以下实例可无需转化直接导入（预计转化时间10秒）区域，单击确认并导入。

- 需要转化后导入的实例在以下实例需要转化后导入，选中同意对以上需要转化的实例进行转化，并已知转化后原有系统中的数据将会丢失，并输入转化后系统root用户登录的新密码。完成设置后单击**确认并导入**。



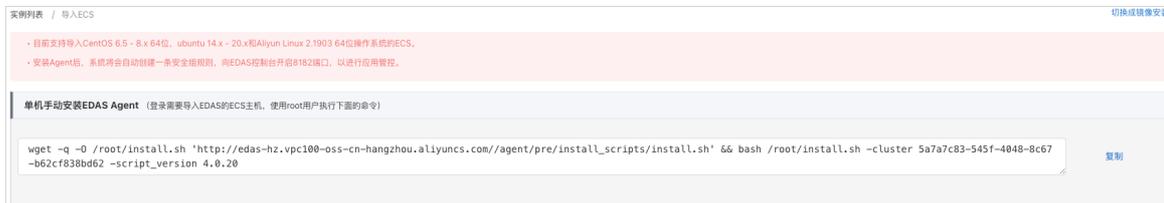
微服务空间：

- 在**进行导入**页签上可以看到实例的导入状态。
  - 能够直接导入的实例：在**进行导入**页面可以快速显示实例的导入状态，当显示**实例转移成功**时说明实例已成功导入。单击**返回集群详情页**，当实例状态显示为**运行中**时也说明实例导入集群成功。
  - 需要转化后导入的实例：在**进行导入**页面会显示实例的导入状态为**正在转化中**，该转化操作预计耗时**五分钟**。在导入完成前单击**返回集群详情页**，健康检查状态会显示为**转化中**并且会显示转化进度百分比，当导入完成时，健康检查的状态会显示为**运行中**。则表示实例导入成功。

### 使用命令脚本手动安装Agent

该方式仅适用于ECS集群中的ECS实例。

- 登录**EDAS控制台**。
- 左侧的导航栏中选择**资源管理 > ECS**。
- 在ECS页面顶部菜单栏选择地域，在页面上方选择**微服务空间**，然后在页面右上角单击**导入ECS**。
- 在导入ECS页面右上角单击**切换到手动安装**，跳转到**单机手动安装EDAS Agent**页面，单击**复制**。



**说明** 如果此时想要通过镜像安装，在页面右上角单击**切换到镜像安装**切换配置页面，完成导入。具体操作，请参见**通过导入ECS自动安装Agent**。

- 以**root**身份登录需要安装Agent的ECS。
- 在ECS实例中，粘贴复制的命令并执行。

## 结果验证

Agent 安装完成后，在EDAS控制台左侧导航栏中选择资源管理 > ECS，在ECS页面顶部菜单栏选择目标地域，在页面上方选择微服务空间，查看Agent状态。

- 如果Agent安装成功，其状态为在线或在线Docker。
- 如果Agent安装失败，其状态为未知。

## 升级Agent

Agent的升级步骤和使用命令脚本安装Agent方法一致，具体步骤，请参见[使用命令脚本手动安装Agent](#)。

# 2.5. 创建混合云ECS集群

EDAS支持混合云ECS集群，且可以对混合云集群的扩容、网络和统一管理等问题提供完整的解决方案。将阿里云ECS、本地IDC或其它云服务提供商的服务器通过专线连通，并添加到EDAS混合云ECS集群中，即可将应用托管到混合云ECS集群，使用EDAS提供的一系列应用托管能力。

## 前提条件

请确保您使用的是专业版或铂金版 EDAS。更多信息，请参见[产品系列](#)和[价格说明](#)。

## 背景信息

您的应用系统可能同时有以下需求及问题：

- 云端（阿里云）流量有一定的波动性，尤其在某些时间段内会迎来流量高峰。流量可以预估，但难免有偏差，对于扩容的时间及ECS实例数量不好控制，且需要提前购买。
- 某些核心业务系统安全性要求较高，需要将这部分应用部署在自己的IDC（Internet Data Center，互联网数据中心）内。这样会面临阿里云、IDC及其它云服务提供商的服务器间网络不通的问题，无法统一管理部署在不同环境下的应用。
- 根据实际的业务需求和可用性需求，希望将该应用系统同时部署在混合云环境中。但将无法实现统一管控，需要人工处理，容易导致误操作。

上述这些需求及问题都可以通过混合云解决。

混合云ECS的架构如下图所示。



- 混合云（非阿里云）集群包含阿里云的ECS实例、IDC和其它云提供商的服务器。
- 阿里云的ECS实例、IDC和其它云提供商的服务器通过高速通道连通。
- 应用可以部署在混合云ECS集群中。

混合云主要适用于以下场景：

- 阿里云管理IDC：在您本地IDC通过专线接入阿里云的VPC之后，您可以使用阿里云EDAS管理IDC内的应用。
- 部署和管理其它云服务提供商的服务器：使用EDAS，您能够将应用部署到除阿里云之外的云服务提供商的服务器上，并进行统一管理。

本文档主要介绍阿里云管理IDC场景。部署和管理其它云服务提供商的服务器只需要您将所需服务器通过专线接入EDAS阿里云的VPC，后续操作和公共云管理IDC的场景一致。

## 准备工作

- 创建VPC。具体操作，请参见[搭建IPv4专有网络](#)。
- 开通[高速通道](#)。
- 申请[创建和管理独享专线连接](#)，将您本地的IDC接入阿里云VPC。
- 请确保本地IDC中的服务器满足如下要求：
  - 操作系统：CentOS 7
  - 物理机或虚拟机（不支持Docker）
  - 硬件：CPU和内存等无特殊要求

## 步骤一：创建混合云集群

1. 登录EDAS控制台。
2. 在左侧导航栏选择资源管理 > EDAS ECS集群。
3. 在EDAS ECS集群页面顶部菜单栏选择地域，在页面上方选择微服务空间，然后单击创建集群。  
微服务空间可以在该页面选择，也可以在创建集群对话框中选择。
  - 如果有环境隔离的需求，请选择您创建的微服务空间。
  - 如果没有环境隔离的需求，可以选择默认微服务空间。
4. 在创建集群对话框中设置集群参数，然后单击创建。

创建集群
✕

\* 集群名称:

集群归属: 阿里云 非阿里云

\* 集群类型: ECS ▾

\* 集群网络类型: 专有网络 ▾

VPC网络: doc-test (vpc-47912421-12345678901234567890) ▾ 创建VPC

命名空间: docDemo ▾ 刷新

资源组 📌: 资源A ▾ 创建资源组

创建
取消

参数	描述
集群名称	集群名称仅支持字母、数字、下划线（_）和英文句号（.），且长度不能超过64个字符。
集群归属	选择非阿里云。
集群类型	仅支持ECS，不可设置。
集群网络类型	非阿里云ECS集群的网络类型只能选择专有网络。
VPC网络	选择专有网络后，需要选择创建的具体VPC。
微服务空间	集群所属的微服务空间。默认显示为在EDAS ECS 集群页面选择的微服务空间。如果未选择，列表中 will 显示默认微服务空间。请根据实际需求选择。
资源组	集群所属的资源组。该资源组为当前账号在阿里云资源管理中创建的资源组，而非EDAS的资源组。如果没有资源组，可以单击创建资源组，跳转到阿里云资源管理控制台进行创建。具体操作，请参见创建资源组。

集群创建成功后会在当前页面上方出现创建成功的提示，同时新创建的集群会在集群列表中显示。

 **说明** 集群创建成功后，仅仅是一个空集群，需要添加阿里云ECS实例和IDC中的服务器。

## 步骤二：为混合云集群添加实例

您需要为刚刚创建的空的混合云集群添加阿里云ECS实例、IDC和其它云提供商的服务器。

1. 在**集群列表**页面单击刚刚创建的集群ID。
2. 在**集群详情**页面的**ECS实例**区域右上角单击**添加已有ECS**。
3. 在**添加ECS实例**对话框中复制EDAS Agent的安装命令。

 **说明** 在混合云集群中仅支持通过命令行手动安装EDAS Agent的方式添加实例。

4. 使用root账号登录您的阿里云ECS实例或本地IDC中的服务器，并复制、执行EDAS Agent安装命令。安装命令执行后，该实例或服务器就添加到集群中了。

## 步骤三：开通所需端口

为保证您在混合云中的应用能正常使用EDAS提供的相关能力，在添加实例后您需要为实例开通以下端口：

- 8182：用于抓取基础监控、调用链监控日志
- 12200~12300：用于完成RPC调用
- 65000~65535：Web端口

集群中的实例类型不同，开通端口的方式也不同。

- 阿里云ECS实例：开通端口的具体操作，请参见[添加安全组规则](#)。
- IDC或其它云提供商的服务器：请参考对应方案开通相关端口。

## 结果验证

1. 返回EDAS ECS集群页面，在**集群列表**中查看刚创建的集群的**集群类型**、**集群状态**和**实例数**。
  - **集群类型**为ECS集群（非阿里云）、**集群状态**为正常，则说明集群创建成功。
  - **实例数**和实际添加的实例数量一致，则说明实例添加成功。
2. 单击目标集群的集群ID，进入**集群详情**页面。在**ECS实例**区域查看添加的实例名称及**健康检查**状态。各实例的**健康检查**为**运行中**说明实例状态正常。

## 后续步骤

混合云ECS集群创建成功后，即可在集群中部署应用。部署应用的方式与在阿里云ECS集群的部署步骤一致，具体操作，请参见[应用创建和部署概述 \(ECS\)](#)。

# 2.6. 管理资源组

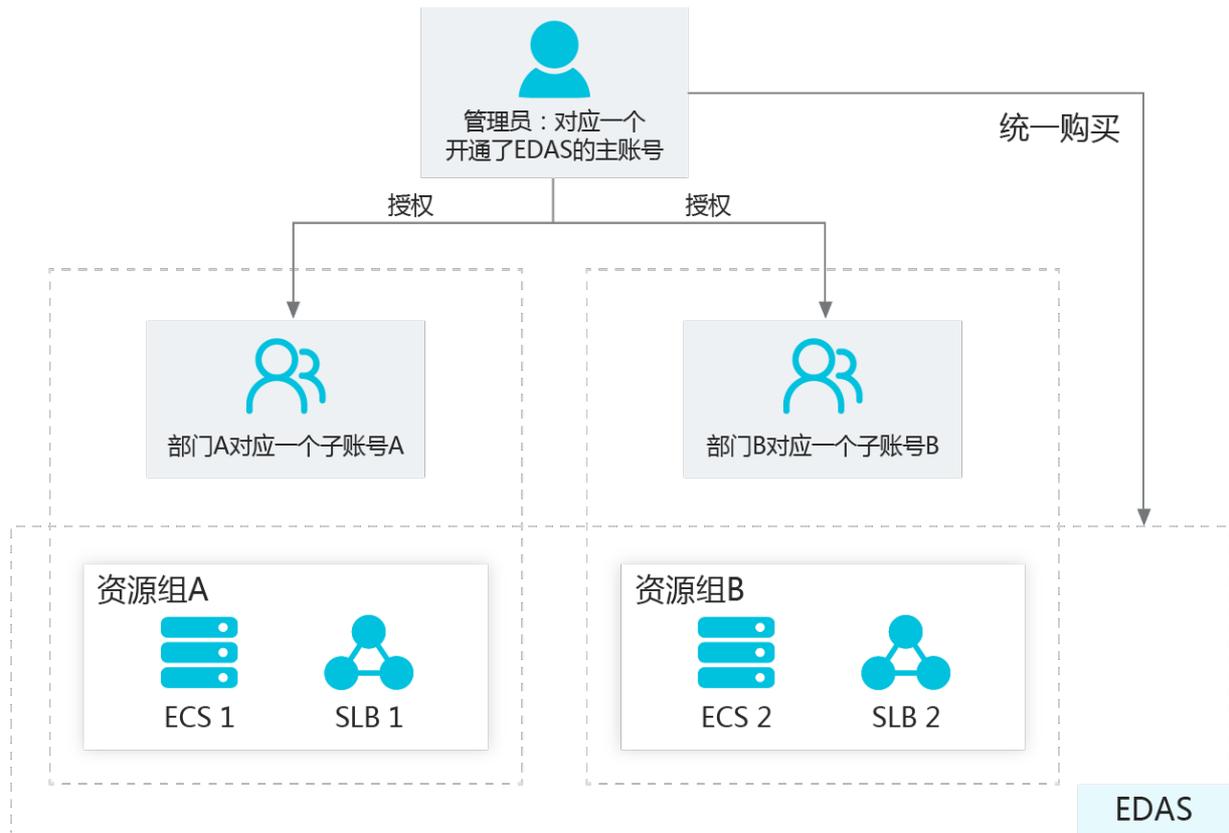
当您使用云账号负责购买资源，子账号或RAM用户负责应用运维时，可以使用资源组对账号的权限进行控制。

EDAS可以对子账号或RAM用户进行资源组授权，被授权子账号或RAM用户拥有操作这个资源组中所有资源的权限。更多信息，请参见[权限管理概述](#)。

 **说明** 资源组是EDAS对资源的一种划分，资源组只能绑定ECS、SLB或者集群，不能绑定VPC。

## 典型应用场景

- 某公司使用了EDAS搭建自己的业务应用。其中部门A负责用户中心的应用，部门B负责商品中心的应用。
- 两个部门共同使用了一个EDAS主账号开通的EDAS服务，并分别有一个子账号。部门A有几台专用ECS和SLB实例，用于部署用户中心的应用。
- 该公司在使用EDAS的时候创建了一个资源组，向这个资源组中添加部门A的ECS和SLB实例，然后将这个资源组授权给部门A使用的子账号。
- 部门A在使用这个子账号时只能操作这个资源组中的资源。部门A和部门B之间不会产生冲突。



## 进入资源组管理页面

1. 登录EDAS控制台。
2. 在左侧导航栏中选择资源管理 > 资源组。
3. 在资源组页面选择地域 (Region)，可以查看该地域内的资源组以及各资源组中的ECS和SLB实例。

## 查看资源组

在资源组页面中可以查看资源组的信息，包括资源组名称、描述、ECS实例（内外网IP地址）、SLB实例信息和集群。

## 创建资源组

1. 在资源组页面右上角单击创建资源组。
2. 在创建资源组对话框中输入资源组名称和资源组描述，单击确定。

## 添加ECS到资源组

1. 在资源组列表的操作列中，单击绑定ECS。
2. 在绑定ECS对话框中，选择要绑定的ECS实例，单击确定。

## 同步SLB并添加SLB到资源组

如果您购买了阿里云的负载均衡 (Server Load Balancer) 服务, EDAS会将SLB实例同步到EDAS控制台, 以便使用、配置负载均衡功能。

1. 在资源组列表的**操作列**中, 单击**绑定SLB**。
2. 在**绑定SLB**对话框中单击**同步SLB**, 以手动同步SLB到EDAS控制台。
3. 在**绑定SLB**对话框中, 选择要绑定的SLB实例, 单击**确定**。

## 编辑资源组

1. 在资源组列表的**操作列**中, 单击**编辑**。
2. 在**编辑资源组**对话框中, 对资源组名称和描述信息进行编辑, 然后单击**确定**。

## 删除资源组

1. 在资源组列表的**操作列**中, 单击**删除**。
2. 在**删除资源组**确认框中单击**确定**, 即可删除一个资源组。

## 对子账号或RAM用户授权资源组

1. 以云账号身份登录EDAS控制台。
2. 在控制台左侧菜单栏选择**系统管理 > 子账号**。
3. 在要授权的用户的**操作列**, 单击**授权资源组**。
4. 在**授权资源组**对话框中选择资源组, 单击**确定**。

## 3. 创建和部署应用 (ECS)

### 3.1. 应用创建和部署概述 (ECS)

在EDAS的ECS集群中，可以创建和部署基于Java微服务框架开发的应用。

支持的微服务框架	支持的版本	支持的部署包	应用运行环境	应用部署方式	
 Spring Cloud	Edgware 及以上版本		 Java	 控制台	
			 Apache-Tomcat		
 DUBBO	2.5.x 及以上版本		 Java		 API
			 Apache-Tomcat		
 HSF	最新版本		 EDAS-Container		

#### 应用

EDAS的ECS集群支持主流的Java微服务框架Spring Cloud、Dubbo，也支持阿里巴巴自研的HSF。

 说明 为帮助您快速体验应用创建和部署，EDAS提供了各种微服务框架的应用Demo。

#### ECS环境

EDAS提供的ECS环境分为阿里云ECS环境和混合云ECS环境。

- 阿里云ECS环境指相关资源均有阿里云提供，包括微服务空间、VPC、ECS集群和ECS实例。
  - 为了帮助您快速体验应用创建和部署，EDAS在每个地域内均提供了一个默认ECS环境。默认ECS环境即在默认微服务空间中创建一个默认集群，该集群使用该地域的默认VPC。
  - 您还可以根据实际需求，创建自定义ECS环境。自定义环境指您自行创建的微服务空间、VPC、ECS集群及ECS实例。
- 混合云ECS环境主要指部署应用的实例来自阿里云ECS、本地IDC和其他云服务提供商的服务器。但微服务空间、VPC和ECS集群还是在阿里云中创建的。

#### 创建和部署

您可以根据实际需求，在相应的ECS环境中创建和部署应用。

不同框架的应用支持的部署包类型不同，不同部署到类型需要选择的应用运行环境也有所不同。

- 部署包类型
  - Spring Cloud和Dubbo应用支持JAR、WAR两种部署包类型，HSF应用仅支持WAR包。

- 应用运行环境
  - Spring Cloud和Dubbo应用通过JAR包部署时，选择标准Java环境即可。
  - Spring Cloud和Dubbo应用通过WAR包部署时，应用运行环境需要选择Apache Tomcat。
  - HSF应用部署（只能选择WAR包部署）时，应用运行环境必须选择EDAS-Container。EDAS-Container包含Ali-Tomcat和Pandora。

ECS集群为Java应用提供了控制台和API两种创建和部署方式。

## 场景

在首次部署时，您可以根据实际场景选择不同的部署方案，并根据文档完成创建和部署。

- 快速体验应用创建和部署
  - 将应用Demo部署到默认ECS环境中。具体操作，请参见[在默认ECS环境中部署微服务应用](#)。
  - 将应用Demo部署到自定义ECS环境中，自定义ECS环境包括微服务空间、VPC和集群等。具体操作，请参见[在自定义ECS环境中部署微服务应用](#)。
- 阿里云部署
  - 将已有应用部署到阿里云ECS自定义环境中。具体操作，请参见[在ECS集群中创建并部署应用](#)。
- 混合云部署
  - 将已有应用部署到混合云ECS环境中。具体操作，请参见[在混合云中部署应用](#)。

## 3.2. 在ECS集群中创建并部署应用

在ECS集群中创建并部署应用

您可以在ECS集群中创建并部署已有应用。

### 前提条件

- 开通并授权弹性伸缩服务，请参见[授予弹性伸缩服务权限](#)。
- [使用控制台创建ECS集群](#)。

### 在ECS集群中创建并部署应用

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏单击[应用列表](#)。
3. 在[应用列表](#)页面的顶部菜单栏选择地域，并在页面上方选择微服务空间，然后在页面左上角单击[创建应用](#)。
4. 在创建应用向导页面的[应用基本信息](#)页签中，输入应用基本信息，然后单击下一步进入应用部署配置页面。

参数	描述
集群类型	选择ECS集群。

参数	描述
应用运行环境	<ul style="list-style-type: none"> <li>○ Dubbo或Spring Cloud应用           <ul style="list-style-type: none"> <li>■ 通过JAR包部署               <ul style="list-style-type: none"> <li>■ 选择Java。</li> <li>■ 在Java环境列表中选择JDK版本。如果您想使用自定义OpenJDK, JAVA环境请选择Custom OpenJDK, 然后输入OpenJDK安装命令。详细信息, 请参见<a href="#">自定义OpenJDK</a>。</li> </ul> </li> <li>■ 通过WAR包部署               <ul style="list-style-type: none"> <li>■ 选择Tomcat。</li> <li>■ 在Java环境列表中选择JDK版本。</li> <li>■ 在容器版本列表中选择Apache Tomcat版本。</li> </ul> </li> </ul> </li> <li>○ HSF应用           <ul style="list-style-type: none"> <li>■ 选择EDAS-Container (HSF)。</li> <li>■ 在Java环境列表中选择JDK版本。</li> <li>■ 在Pandora列表中选择Pandora版本。</li> <li>■ 在Ali-Tomcat列表中选择Ali-Tomcat版本。</li> </ul> </li> </ul>
应用名称	输入应用名称, 应用名称必须以字母开头, 允许数字、字母、下划线 ( _ ) 以及短划线 ( - ) 组合, 最多支持36个字符。
(可选) 应用描述	填写应用的基本情况, 最多支持128个字符。

5. 在应用配置页签选择部署包来源和实例来源, 完成设置后在页面最下方单击下一步。

参数	描述
部署包来源	<ul style="list-style-type: none"> <li>○ 自定义程序           <p>将已有应用部署到EDAS中, 需选择文件上传方式:</p> <ul style="list-style-type: none"> <li>■ 上传JAR包或上传WAR包或上传部署包: 选择本地的应用部署包并上传。</li> <li>■ JAR包地址或WAR包地址或部署包地址: 输入您在公网中的部署包地址。</li> </ul> </li> <li>○ 官方Demo           <p>EDAS提供了Spring Cloud、Dubbo和HSF框架的一对应用Demo, 仅为帮助您体验应用的创建、部署。更多信息, 请参见<a href="#">开始使用EDAS</a>。</p> </li> <li>○ 不部署           <p>仅创建应用, 生成应用的基本信息, 但不部署。</p> </li> </ul>

参数	描述
实例来源	<ul style="list-style-type: none"> <li>购买新实例：由EDAS为您代购ECS实例。</li> <li>使用已有实例：使用您集群内已有的闲置实例。</li> </ul>
环境	<ul style="list-style-type: none"> <li>默认环境：在默认微服务空间下新建一个默认集群，该集群使用当前地域内的默认VPC。</li> <li>自定义环境：选择您创建的微服务空间和集群。如果未创建微服务空间和集群，可以单击创建微服务空间和创建集群进行创建。</li> </ul>
购买方式	<ul style="list-style-type: none"> <li>基于推荐规格购买：根据您的业务需求来选择实例规格大小，设置购买数量和登录密码，并选择《云服务器ECS服务条款》 《镜像商品使用条款》。</li> <li>基于现有实例规格购买：在模板实例列表选择任一实例作为模板，选择回收模式，设置登录密钥和购买数量，并选择《云服务器ECS服务条款》 《镜像商品使用条款》。</li> <li>基于实例启动模板购买：选择启动模板和回收模式，设置购买数量，并选择《云服务器ECS服务条款》 《镜像商品使用条款》。</li> </ul>

- 在应用高级设置页签设置版本和（可选）应用健康检查，然后单击下一步。
- （可选）在分组设置（选填）页签中选择资源组，然后单击创建应用。
- 在应用创建完成页签确认应用基本信息、应用配置和应用高级设置等信息，确认完毕后单击确定创建应用。  
在应用基本信息页面顶部提示信息中单击查看详情，可在变更详情页面查看变更流程。应用部署需要几分钟，请您耐心等待。变更完成后在应用基本信息页面的实例部署信息页签查看实例的运行状态，如果运行状态为运行正常，说明应用部署成功。

（可选）

## 配置公网负载均衡

应用一般是在VPC内创建、部署，如果没有特别配置，该应用没有公网IP地址。如果您希望将应用对外开放，建议配置公网负载均衡SLB。

 说明 EDAS会为托管的HTTP监听设置为会话保持模式，不可更改。

- 登录EDAS控制台。
- 在左侧导航栏单击应用列表。
- 在应用列表页面的顶部菜单栏选择地域，在页面上方选择微服务空间和集群类型（ECS集群），然后单击目标应用名称。
- 在基本信息页面中基本信息页签的应用设置区域，单击负载均衡（公网）右侧的添加。

5. 在添加SLB与应用的绑定对话框中，根据页面提示设置负载均衡参数，具体操作，请参见[应用独享负载均衡实例](#)。

复制配置的负载均衡（公网）的公网IP和端口，如 `39.XX.XX.65:81`，粘贴至浏览器地址栏并按Enter键，即可访问该应用的Web页面。

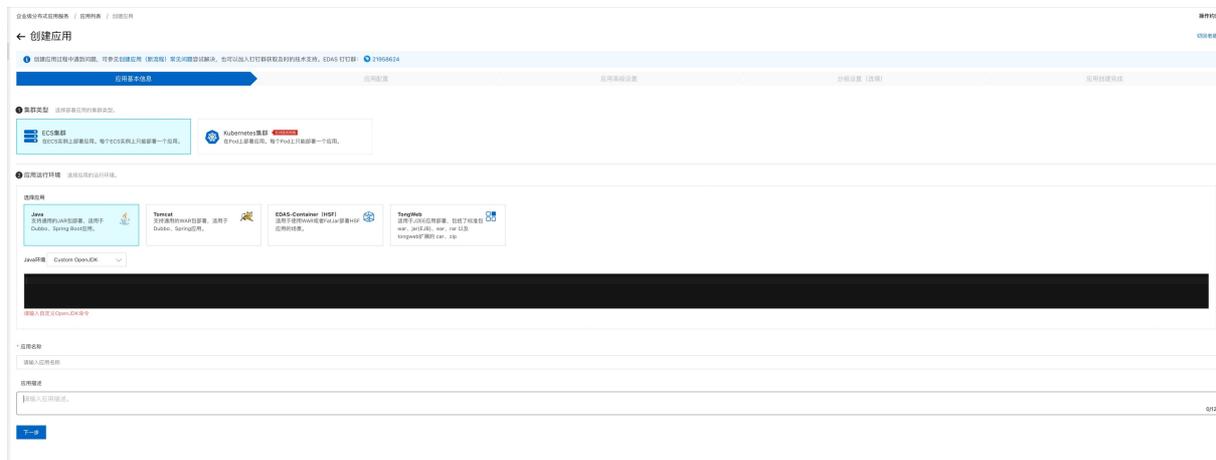
## ECS系统配置约束

应用创建后，可能需要登录ECS实例进行系统配置，请遵循以下约束：

- 不能删除admin用户。
- 不能删除 `/home/admin` 配置。
- 不能停止主机上的以下进程：
  - `/home/staragent/bin/staragentd`
  - `com.alibaba.edas.agent.AgentDaemon`
- 不能删除主机上的以下crontab：
  - root用户下的 `bash /home/admin/edas-agent/bin/monitor.sh` 任务
  - admin用户下的 `bash /home/admin/edas-agent/bin/rotator.sh` 任务
- 必须保证磁盘根分区有可用空间。
- 如果使用CentOS操作系统，需要保证yum源配置正确，功能正常。
- ECS如果使用多网卡，比如安装了Docker，且应用为HSF应用时，需要为应用指定 `-Dhsf.server.ip` 参数来确保注册的IP符合预期。更多信息，请参见[JVM-D启动配置参数](#)。

## 自定义OpenJDK

ECS应用部署提供自定义JDK安装命令输入，在创建ECS环境时，会执行您输入的命令。



ECS上自定义安装的JDK版本与您的操作系统相关联，此处以 `aliyun_3_x64_20G_alibase_20210910.vhd` 系统镜像为例，介绍如何自定义OpenJDK版本。

**说明** ECS应用需要安装JDK（Java Development Environment），并给Java和Jar命令配置了环境变量，即命令行输入 `java` 或 `jar` 命令后显示正常结果。

- 查看系统自带的JDK 8版本。  
在ECS上执行如下命令，搜索系统自带JDK 8版本：

```
# 搜索jre
yum search --showduplicates java-1.8.0-openjdk
# 搜索jdk
yum search --showduplicates java-1.8.0-openjdk | grep devel
```

示例结果如下：

```
java-1.8.0-openjdk-1:1.8.0.292.b10-0.1.a18.x86_64 : OpenJDK 8 Runtime Environment
java-1.8.0-openjdk-1:1.8.0.292.b10-0.1.a18.x86_64 : OpenJDK 8 Runtime Environment
java-1.8.0-openjdk-1:1.8.0.302.b08-0.1.a18.x86_64 : OpenJDK 8 Runtime Environment
java-1.8.0-openjdk-1:1.8.0.312.b07-2.a18.x86_64 : OpenJDK 8 Runtime Environment
java-1.8.0-openjdk-1:1.8.0.322.b06-2.a18.x86_64 : OpenJDK 8 Runtime Environment
java-1.8.0-openjdk-1:1.8.0.272.b10-1.1.a18.x86_64 : OpenJDK Runtime Environment 8
```

可以看到该系统镜像的Yum仓库提供了如上JDK 8小版本

- 安装JDK 8版本。

在ECS上执行如下命令测试安装自定义OpenJDK，并且执行 `java` 命令和 `jar` 命令查看结果：

```
yum install -y java-1.8.0-openjdk-1:1.8.0.302.b08-0.1.a18.x86_64 java-1.8.0-openjdk-devel-1:1.8.0.302.b08-0.1.a18.x86_64
```

## 相关文章

- [ECS 集群中的应用生命周期管理](#)

## 3.3. 在混合云中部署应用

EDAS支持混合云ECS集群，且可以对混合云集群的扩容、网络和统一管理等问题提供完整的解决方案。您可以在混合云ECS集群中创建并部署应用。

### 前提条件

- 请确保您使用的是专业版或铂金版EDAS。
- [创建混合云ECS集群](#)。

### 背景信息

EDAS支持混合云ECS集群，且可以对混合云集群的扩容、网络和统一管理等问题提供完整的解决方案。将公共云ECS、本地IDC或其它云服务提供商的服务器通过专线连通，并添加到EDAS混合云ECS集群中，即可将应用托管到混合云ECS集群。

### 部署应用

在混合云集群中部署应用的方式与在ECS集群的部署步骤一致，具体操作，请参见[在ECS集群中创建并部署应用](#)。

## 3.4. 使用API创建应用 (ECS)

本文介绍如何使用EDAS提供的Java SDK调用API在EDAS ECS环境中创建应用。

### 前提条件

在EDAS ECS环境中创建应用前，您必须提前完成以下准备工作：

- 安装EDAS的Java SDK。更多信息，请参见[Java SDK接入指南](#)。
- 确定创建应用的地域，假设为 `cn-hangzhou`。
- 创建微服务空间。具体操作，请参见[创建命名空间](#)。

如果已经存在命名空间，您可以调用 `ListUserDefineRegion` 接口查询用户自定义命名空间列表，获取目标命名空间的 `RegionId`。

- 创建ECS集群，具体操作，请参见[使用API创建ECS集群](#)。  
如果您已经创建了ECS集群，则可以调用 `ListCluster` 接口查询集群列表，获取目标ECS集群的集群ID (ClusterId)，假设为 `369d06d7-450b-4f3d-bf75-9536fcd9****`。
- 在ECS集群中添加ECS实例，具体操作，请参见[使用API扩容ECS集群](#)。
- 调用 `ListScaleOutEcu` 接口从指定ECS集群中查询ECU列表，获取ECU的唯一标识ID (EcuId)，假设为 `82a0ac57-433d-4ad1-ae89-0c6fcb70****` 和 `e2674482-1214-4c93-b767-9b6405e2****`。
- 如果需要创建HSF应用，则需要调用 `ListBuildPack` 接口查询容器版本列表，获取容器版本 (ConfigId) 参数，假设为57。

## 创建应用

以下代码适用于在EDAS ECS集群中快速创建应用。代码中未设置挂载脚本信息和JVM参数信息，如需了解更多API参数信息，请参见[InsertApplication](#)。

 **说明** 以下代码适用于创建Spring Cloud和Dubbo应用，如需创建HSF应用，则需要在代码中增加EDAS-Container信息：

```
//EDAS-Container构建包号，当创建HSF应用时，该参数必须指定。可以调用ListBuildPack接口获取，对应ConfigId参数。  
request.setBuildPackId(57);
```

```
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.exceptions.ServerException;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.edas.model.v20170801.InsertApplicationRequest;  
import com.aliyuncs.edas.model.v20170801.InsertApplicationResponse;  
public class InsertApplication {  
    public static void main(String[] args) {  
        //阿里云账号或RAM用户的AccessKey ID。  
        String aliyun_user_ak = "<yourAccessKeyId>";  
        //阿里云账号或RAM用户的AccessKey Secret。  
        String aliyun_user_sk = "<yourAccessKeySecret>";  
        //要执行API调用的应用所在地域ID。  
        String region_id = "cn-hangzhou";  
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak  
, aliyun_user_sk);  
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);  
        //创建API请求，并设置参数。  
        InsertApplicationRequest request = new InsertApplicationRequest();  
        //目标ECS集群的集群ID。  
        request.setClusterId("369d06d7-450b-4f3d-bf75-9536fcd993dc");  
        //应用描述信息。  
        request.setDescription("create by edas pop api");  
        //自定义的应用名称。应用名称必须以字母开头，允许数字、字母、下划线 ( _ ) 以及短划线 ( - ) 组合，最多支持36个字符。  
        request.setApplicationName("ECSAPP_API");  
        //微服务空间的RegionId。  
        request.setLogicalRegionId("cn-hangzhou:doc");  
        //ECS集群中的ECS实例的ecu_id (是入EDAS的ECS实例在EDAS中的唯一身份)。多个ecu_id用半角逗号 (
```

```

, ) 分开。
    request.setEcuInfo("82a0ac57-433d-4ad1-ae89-0c6fcb70****,e2674482-1214-4c93-b767-9b6405e2****");
    //应用部署包格式,可设置为war或者jar。
    request.setPackageType("jar");
    //应用组件ID。您创建Spring Cloud、Spring Boot或Dubbo应用应用时需要指定组件ID,创建HSF应用时无需设置。
    //4代表Apache Tomcat 7.0.91,7代表Apache Tomcat 8.5.42,5代表OpenJDK 1.8.x,6代表OpenJDK 1.7.x。
    request.setComponentIds("5");
    //是否启用健康检查URL,true代表启用,false代表不启用。
    request.setEnableUrlCheck(true);
    //应用健康检查URL。
    request.setHealthCheckUrl("http://127.0.0.1:8080/_ehc.html");
    try {
        InsertApplicationResponse response = client.getAcsResponse(request);
        System.out.println("Message=" + response.getMessage() + "\nAppName=" + response.getApplicationInfo().getAppName() + "\nAppId=" + response.getApplicationInfo().getAppId());
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
}
}

```

运行程序,返回的执行结果如下所示:

```

Message=success
AppName=ECSAPP_API
AppId=6275ffd0-d577-4b5a-894f-e92991b8****

```

## 结果验证

在EDAS ECS环境中创建应用后,您可以调用[ListApplication](#)接口获取应用列表。根据AppId和Appname等参数,判断应用是否已经创建成功。

调用ListApplication接口,返回的结果如下:

```
{
  "Message": "success",
  "RequestId": "9363A94A-96C5-4B41-93A9-001A4E9DDEE1",
  "ApplicationList": {
    "Application": [
      {
        "BuildPackageId": 0,
        "ApplicationType": "FatJar",
        "AppId": "6275ffd0-d577-4b5a-894f-e92991b8****",
        "ResourceGroupId": "rg-acfm3umruf2****",
        "ClusterId": "369d06d7-450b-4f3d-bf75-9536fcd9****",
        "RegionId": "cn-hangzhou:doc",
        "ClusterType": 2,
        "RunningInstanceCount": 0,
        "Name": "ECSAPP_API"
      }
    ]
  },
  "Code": 200
}
```

# 4. 升级和回滚应用 (ECS)

## 4.1. 升级和回滚应用概述 (ECS)

应用在EDAS中创建并部署后，还会不断迭代，需要升级。如果升级的应用版本发现问题，需要将应用回滚到历史版本。本文介绍EDAS为您提供的应用升级策略、方式，以及在需要时如何回滚应用。

### 升级应用

由于新版本应用的状态不同，升级的场景也不同，您需要使用的升级策略、方式也不同。

升级策略	适用场景	升级方式	相关文档
单批发布 (直接升级)	应用规模较小或服务架构比较简单，在开发、测试完成后，可以直接升级，即直接部署新版本。	<ul style="list-style-type: none"> <li>控制台</li> <li>API</li> <li>插件或工具</li> </ul>	<ul style="list-style-type: none"> <li><a href="#">在ECS集群中创建并部署应用</a></li> <li><a href="#">使用CLI快速部署应用至ECS集群</a></li> <li><a href="#">使用toolkit-maven-plugin插件升级应用 (ECS集群)</a></li> <li><a href="#">在IntelliJ IDEA中使用Cloud Toolkit部署应用 (ECS集群)</a></li> <li><a href="#">在Eclipse中使用Cloud Toolkit部署应用至ECS集群</a></li> </ul>
分批发布	应用规模较大或服务架构比较复杂，可以按批次进行应用部署。每次仅对应用的一部分实例进行升级，直至全量升级完成。分批发布过程中如果出现故障，可以及时终止并回滚，待问题修复后重新发布。	<ul style="list-style-type: none"> <li>控制台</li> <li>API</li> <li>插件或工具</li> </ul>	<ul style="list-style-type: none"> <li><a href="#">使用控制台分批发布应用</a></li> <li><a href="#">使用toolkit-maven-plugin插件分批发布应用 (ECS集群)</a></li> <li><a href="#">使用Cloud Toolkit分批发布应用</a></li> </ul>
金丝雀发布	应用的新版本需要先进行小规模验证，验证后，再将应用全量升级到新版本。	控制台	<a href="#">使用控制台金丝雀发布 (ECS集群)</a>

### 回滚应用

当应用升级过程中或升级完成后，发现新版本应用有问题或异常，可以回滚应用，待问题解决后再升级。

- 升级中回滚：升级过程中，如果发现异常，可以终止并回滚应用。
- 升级后回滚：如果升级已经完成，发现新版本应用有问题，可以通过部署历史版本回滚应用。

回滚应用的具体操作，请参见[使用控制台回滚应用 \(ECS集群\)](#)。

## 4.2. 单批发布 (ECS)

### 4.2.1. 使用控制台升级应用 (ECS集群)

使用控制台升级应用 (ECS集群)

您可以使用EDAS控制台升级 (单批发布) ECS集群中的应用。

#### 背景信息

除使用控制台单批发布应用外，还可以使用Cloud Toolkit和toolkit-maven-plugin插件单批发布应用。具体操作，请参见[在IntelliJ IDEA中使用Cloud Toolkit部署应用 \(ECS集群\)](#)、[在Eclipse中使用Cloud Toolkit部署](#)

应用至ECS集群和使用toolkit-maven-plugin插件升级应用 (ECS集群)。

## 操作步骤

1. 登录EDAS控制台。
2. 在左侧导航栏单击应用列表。
3. 在应用列表页面的顶部菜单栏选择地域，并在页面上方选择微服务空间，在集群类型列表中选择ECS集群，然后在应用列表中单击目标应用名称。
4. 在应用信息页面右上角单击部署应用。
5. 在部署模式选择页面的常规发布 (单批/多批) 区域右上角单击开始部署。
6. 在常规发布 (单批/多批) 页面设置新版本应用的部署包和相关信息，设置发布策略，然后单击确定。
  - i. 设置部署包和相关信息。

参数	说明
应用部署方式	<p>应用部署方式由当前应用首次部署的方式决定，不可修改。例如，首次部署应用时选择了JAR包部署，升级应用时只能选择JAR包部署。</p> <p> 说明 历史版本用于回滚应用。更多信息，请参见<a href="#">使用控制台回滚应用 (ECS集群)</a>。</p>
文件上传方式	<p>新版本应用部署包的上传方式，包含上传部署包 (JAR包或WAR包) 和部署包 (JAR包或WAR包) 地址。部署包类型同样由当前应用首次部署的方式决定。</p> <ul style="list-style-type: none"> <li>▪ 上传JAR包或上传WAR包：单击选择文件，选择本地的新版本应用部署包。</li> <li>▪ JAR包地址或WAR包地址：如果新版本应用已经上传到云端 (例如OSS)，在文本框中输入部署包的云端地址。</li> </ul>
版本	新版本应用的版本。可以输入新版本的版本号，也可以单击用时间戳作为版本号。
(可选) 描述	新版本应用的描述信息。

## ii. 设置发布策略。

参数	说明
分组	分批发布的目标分组。 <ul style="list-style-type: none"> <li>如果当前应用仅有一个分组，在列表中选择<i>所有分组</i>，分组后的括号中显示当前应用的最大实例数。列表中还包含<i>默认分组</i>，和<i>所有分组</i>实际上是一样的。</li> <li>如果当前应用包含多个分组，在列表中选择要分批发布的目标分组。</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><span style="color: #00aaff;">?</span> 说明 如果选择<i>所有分组</i>，则会将应用所有分组中的全部实例按下面设置的<i>每组批次</i>进行升级。</p> </div>
每组批次	选定分组中分批发布的批次数量。单批发布时，选择 <i>1批</i> 。
分批方式	单批发布时，分批方式默认为自动，不可设置。
Java环境	包含 <i>Open JDK 8</i> 、 <i>Open JDK 7</i> 和 <i>Dragonwell 8</i> 。请根据实际需求选择。

? 说明 发布策略设置完成后，在生成Maven插件配置区域会生成当前发布策略的脚本。使用toolkit-maven-plugin插件进行分批发布时，可以直接在使用脚本中使用。具体操作，请参见[使用toolkit-maven-plugin插件分批发布应用 \(ECS集群\)](#)。

## iii. 单击确定。

单批发布启动后，会自动跳转到**变更详情**页面，在**变更详情**中可以跟踪单批发布的进度和状态。具体操作，请参见[查看应用变更](#)。

## 结果验证

## 1. 查看变更记录。

- i. 在应用详情页左侧导航栏单击**变更记录**。
- ii. 在**变更记录**页面的**变更状态**列中查看此次发布的结果是否为**执行成功**。
  - 如果为**执行成功**，说明发布成功。
  - 如果为**执行失败**，说明发布失败。在**操作**列单击**查看**，通过变更详情排查失败原因。

## 2. 查看应用详情。

- i. 在应用详情页左侧导航栏单击**基本信息**。
- ii. 在**基本信息**页面单击**基本信息**页签，在**基本信息**页签的**应用信息**区域中查看应用的**部署包**是否为升级的部署包。
- iii. 在**基本信息**页面单击**实例部署信息**页签，在**实例部署信息**页签查看目标分组中各实例的运行状态是否为运行正常。

## 4.2.2. 使用API在ECS集群中单批发布应用

本文介绍如何使用EDAS提供的Java SDK调用API在EDAS ECS集群中单批发布应用。

### 前提条件

在EDAS ECS环境中单批发布应用前，您必须提前完成以下准备工作：

- 安装EDAS的Java SDK。更多信息，请参见[Java SDK接入指南](#)。

- 确定创建应用的地域，假设为 *cn-hangzhou*。
- 创建微服务空间。具体操作，请参见[创建命名空间](#)。  
如果已经存在命名空间，您可以调用 `ListUserDefineRegion` 接口查询用户自定义命名空间列表，获取目标命名空间的 *RegionId*。
- 在 ECS 集群中创建应用，具体操作，请参见[使用API创建应用 \(ECS\)](#)。  
如果已经创建应用，您可以调用 `ListApplication` 接口查询应用列表，获取目标应用的 *AppId*，假设为 *6bbc57a2-a017-4bec-b521-49a15bd3\*\*\*\**。
- 已将应用部署包 (JAR包或WAR包) 上传至目标地址。本示例以OSS存储路径为例，假设为 *https://doc\*\*\*.oss-cn-hangzhou.aliyuncs.com/sc-\*\*\*\*-D-0.0.1-SNAPSHOT.jar*。
- 如果需要发布HSF应用，则需要调用 `ListBuildPack` 接口查询容器版本列表，获取容器版本 (ConfigId) 参数，假设为 *57*。

## 单批发布应用

以下代码适用于在ECS环境单批发布应用。代码中未设置环境变量等高级参数，如您需了解更多API参数信息，请参见[DeployApplication](#)。

 **说明** 以下代码适用于单批发布Spring Cloud和Dubbo应用，如需发布HSF应用，则需要在代码中增加EDAS-Container信息：

```
//EDAS-Container构建包号，当发布HSF应用时，该参数必须指定。可以调用ListBuildPack接口获取，对应ConfigId参数。  
request.setBuildPackId(57);
```

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.DeployApplicationRequest;
import com.aliyuncs.edas.model.v20170801.DeployApplicationResponse;
public class DeployApplication {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //需要部署应用的所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求, 并设置参数。
        DeployApplicationRequest request = new DeployApplicationRequest();
        //目标应用的AppId。
        request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");
        //应用描述信息。
        request.setDesc("create by edas pop api");
        //部署类型, 仅支持设置为url。
        request.setDeployType("url");
        //应用部署包 (WAR或JAR) 的URL地址, 建议使用OSS存储的应用部署包路径。
        request.setWarUrl("https://doc***.oss-cn-hangzhou.aliyuncs.com/sc-****-D-0.0.1-SNAPSH
OT.jar");
        //部署的应用部署包版本, 最长64个字符, 建议使用时间戳。
        request.setPackageVersion("1619094147");
        //部署分组ID。如果要部署到所有分组, 参数设置为all。
        //如果需要部署到具体应用实例分组, 可以调用ListDeployGroup接口获取GroupId。
        request.setGroupId("all");
        //每组发布批次, 设置为1, 代表单批发布。
        request.setBatch(1);
        //分批方式, 0表示自动, 1表示批次间需要手动确认。
        request.setReleaseType((long) 0);
        //应用组件ID。您单批发布Spring Cloud或Dubbo时, JAR包部署需要选择JDK版本, WAR包部署需要选择A
pache Tomcat版本; 单批发布HSF应用时无需设置该参数。
        //4代表Apache Tomcat 7.0.91, 7代表Apache Tomcat 8.5.42, 5代表OpenJDK 1.8.x, 6代表OpenJ
DK 1.7.x。
        request.setComponentIds("5");
        try {
            DeployApplicationResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage() + "\nChangeOrderId=" + re
sponse.getChangeOrderId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序，返回的执行结果如下所示：

```
Message=success
ChangeOrderId=f32db5e3-a105-412c-9168-6f7c5295****
```

## 结果验证

您在ECS环境中部署应用后，可以调用GetChangeOrderInfo接口查看应用变更流程详情，获取部署应用的变更状态。具体API参数详情，请参见[GetChangeOrderInfo](#)。

调用GetChangeOrderInfo接口查看应用变更流程详情，返回的执行结果如下：

```
{
  "Message": "success",
  "RequestId": "9C487FE4-F188-45EC-87A5-8BD3D655568F",
  "Code": 200,
  "changeOrderInfo": {
    "Status": 2,
    "Desc": "create by edas pop api",
    "PipelineInfoList": {
      .....
      //此处未罗列全部执行结果，仅供参考。
    }
  }
}
```

请查看上述执行结果中的changeOrderInfo.Status参数值，通过该值判断单批发布应用的变更是否成功。changeOrderInfo.Status的取值如下：

- 0：准备
- 1：执行中
- 2：执行成功
- 3：执行失败
- 6：终止
- 8：手动分批发布模式下，等待手工确认执行下一批
- 9：自动分批发布模式下，等待下一批执行中
- 10：系统异常执行失败

## 4.2.3. 在IntelliJ IDEA中使用Cloud Toolkit部署应用 (ECS集群)

您在本地完成新版本应用的开发和测试后，可以在IntelliJ IDEA中使用Cloud Toolkit插件将新版本应用部署到EDAS的ECS集群中。

### 前提条件

- 下载并安装[JDK 1.8或更高版本](#)。
- 下载并安装[IntelliJ IDEA \(2018.3或更高版本\)](#)。

 **说明** 因JetBrains插件市场官方服务器在海外，如遇访问缓慢无法下载安装的，请加入文末交流群，向Cloud Toolkit产品运营获取离线包安装。

- 已在EDAS控制台创建了应用并完成了首次部署。具体操作，请参见[在ECS集群中创建并部署应用](#)。

- 已经在IDE中安装了Cloud Toolkit，并在Cloud Toolkit中配置了阿里云账号或RAM用户的AccessKey。具体操作，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)和[在Eclipse中安装和配置Cloud Toolkit](#)。

## 背景信息

Alibaba Cloud Toolkit (IDE插件，简称Cloud Toolkit) 是阿里云针对IDE平台为开发者提供的一款插件，用于帮助开发者高效开发并部署适合在云端运行的应用。更多信息，请参见[什么是Alibaba Cloud Toolkit](#)。

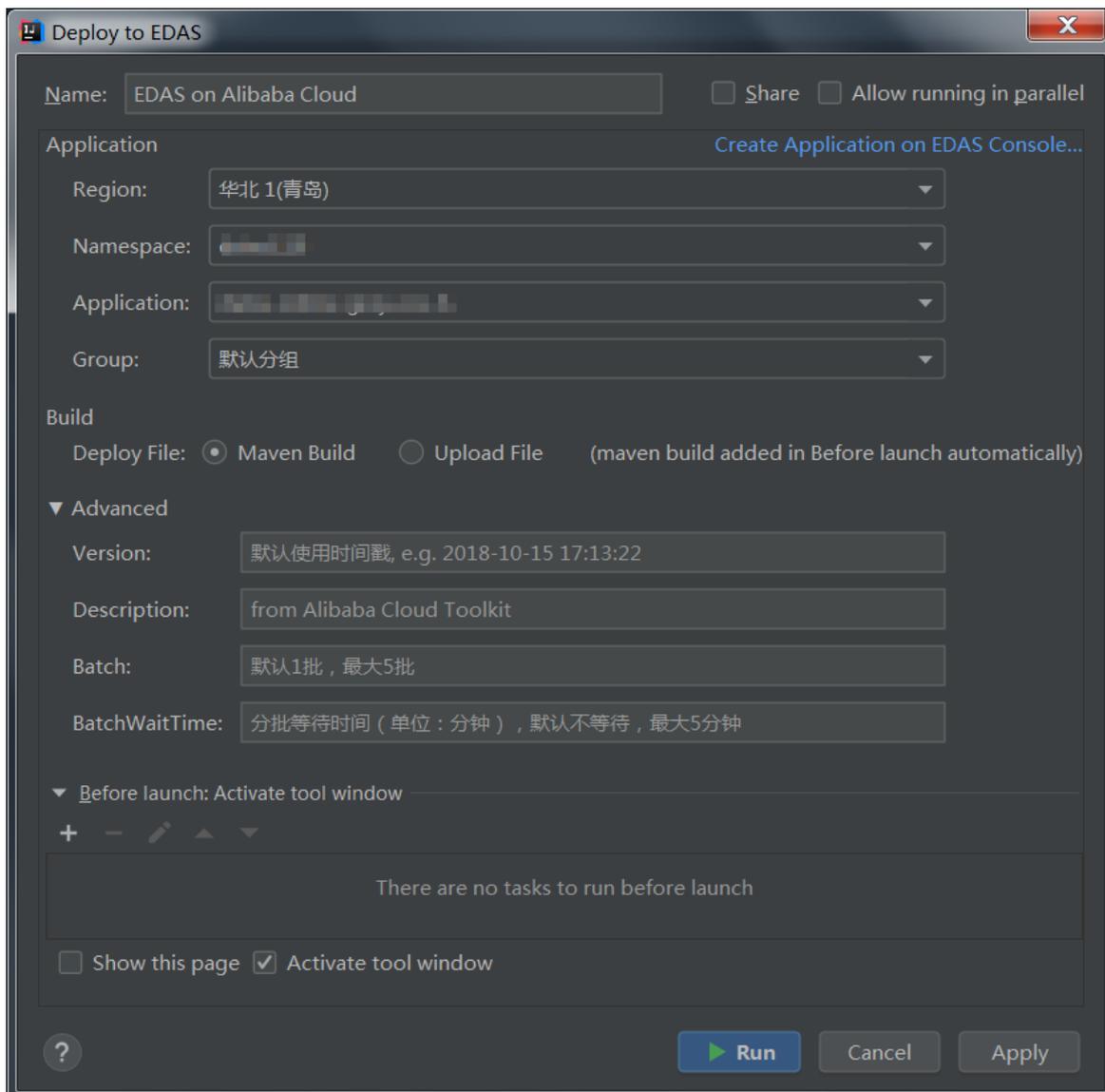
本文介绍在IntelliJ IDEA中如何使用Cloud Toolkit部署应用。您还可以：

- 使用Cloud Toolkit分批发布应用，具体操作，请参见[使用Cloud Toolkit分批发布应用](#)。
- 在Eclipse中使用Cloud Toolkit部署（单批发布）、升级（单批发布或分批发布）应用，具体操作，请参见[使用Eclipse部署应用到EDAS](#)。

## 操作步骤

1. 在IntelliJ IDEA的工具栏中单击Cloud Toolkit的图标，在下拉列表中选择Deploy to EDAS > EDAS for ECS Application。
2. 在Deploy to EDAS ECS Cluster对话框配置应用部署参数，然后单击Run。

 **说明** 如果您还没有在EDAS上创建应用，在对话框右上角单击Create application on EDAS console...，跳转到EDAS控制台创建应用。具体操作，请参见[在ECS集群中创建并部署应用](#)。



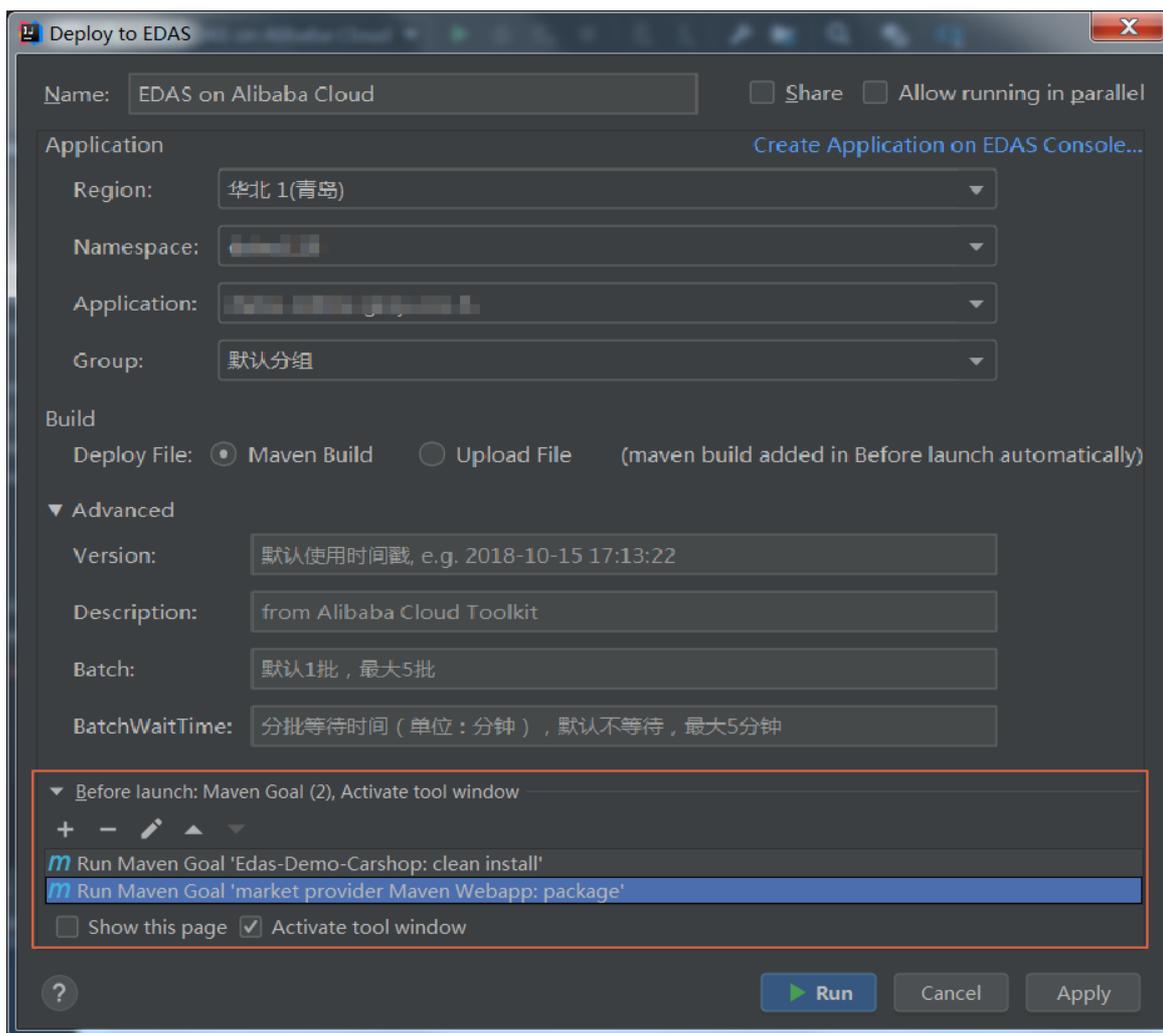
参数	说明
<b>ECS Cluster</b>	
<b>Region</b>	目标应用所属地域。
<b>Namespace</b>	目标应用所属微服务空间。
<b>Application</b>	目标应用。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 5px;"> <span style="font-size: 1.2em;">🔊</span> <b>注意</b> 如果在应用列表中获取不到应用, 请参见<a href="#">常见问题: 应用列表获取不到应用</a>进行操作排查。         </div>
<b>Group</b>	目标应用分组。
<b>Build</b>	

参数	说明
Deploy File	<p>新版本应用部署包文件形式，包含Maven Build和Upload File。请根据实际需求选择。</p> <ul style="list-style-type: none"> <li>◦ <b>Maven Build</b>：在部署应用前，通过Maven来构建新版本应用的部署包。</li> <li>◦ <b>Upload File</b>：在本地选择并上传新版本应用的部署包（WAR包或者JAR包）进行部署。</li> </ul>
Version	新版本应用的版本号。默认使用时间戳作为版本号，例如2020-11-17 17:17:17。
Description	此次分批发布的描述信息。
Batch	<p>应用分批发布的批次。默认为1批，最大为5批。</p> <p> <b>说明</b> 如果目标应用有多个分组，并且需要部署到全部分组，那么将会自动按照分组粒度来分批，Batch不用设置。</p>
BatchWaitTime	每批次间的发布时间间隔，单位为分钟。默认为不等待，最大等待时间为5分钟。

 **说明** Batch和BatchWaitTime为分批发布应用需要设置的参数。更多信息，请参见[使用Cloud Toolkit分批发布应用](#)。

实际工作中大部分Maven工程都是多模块的，各个项目模块可以独立开发，其中某些模块又可能会使用到其他的一些模块的功能，这样的项目工程就是多模块工程。如果需要部署多模块工程应用，请参见[使用IntelliJ IDEA部署多模块工程中的子模块](#)。

在IntelliJ IDEA中安装的Cloud Toolkit内可以管理Maven的构建任务。您也可以在Deploy to EDAS ECS Cluster对话框的Before launch区域来添加、删除、修改和移动Maven构建任务。



分批发布启动后，IntelliJ IDEA的Terminal区域会打印部署日志。您可以根据日志信息查看部署进度和结果。

## 常见问题：应用列表获取不到应用

通常出现这种情况为使用EDAS子账号或RAM用户来部署应用，且EDAS子账号或RAM用户没有同步到EDAS系统或者没有进行正确授权，从而导致在应用列表下拉框中看不到应用。您可以通过RAM授权或EDAS子账号授权来确保EDAS子账号或RAM用户已经同步到EDAS并且得到授权。

### • RAM授权

该授权方式可使RAM用户访问EDAS的所有资源。

- 在**RAM控制台**左侧导航栏中选择**人员管理 > 用户**。
- 在**用户**页面上找到需要授权的RAM用户，单击操作列中的**添加权限**。
- 在**添加权限**面板的**选择权限**区域中，搜索**AliyunEDASFullAccess**权限，单击权限策略将其添加至右侧的**已选择**列表中，然后单击**确定**。
- 在**添加权限**的授权结果页面上，查看授权信息摘要，并单击**完成**。
- 使用阿里云账号登录**EDAS控制台**，在左侧导航栏选择**系统管理 > 子账号**，单击子账号页面右上角的**同步子账号**。

### • EDAS子账号授权

该授权方式可使子账号细粒度授权访问EDAS的资源。

- 使用阿里云账号登录**EDAS控制台**。

- ii. 在左侧导航栏选择**系统管理 > 角色**，单击角色页面右上角的**创建角色**。
- iii. 输入一个角色名称，在可选权限区域框中，选择**应用管理 > 应用列表 > 基本信息 > 部署应用**，单击**添加**将部署应用角色添加到已选权限，然后单击**确定**。
- iv. 在左侧导航栏选择**系统管理 > 子账号**，单击子账号页面右上角的**同步子账号**。
- v. 选择需进行授权的子账号，在操作列单击**管理角色**，在左侧穿梭框中搜索并选择上面创建的角色，将该角色添加到右侧已选角色列表中，然后单击**确定**。
- vi. 选择需进行授权的子账号，在操作列单击**授权应用**，选择应用添加到右侧列表进行授权，然后单击**确定**。

## 4.2.4. 在Eclipse中使用Cloud Toolkit部署应用至ECS集群

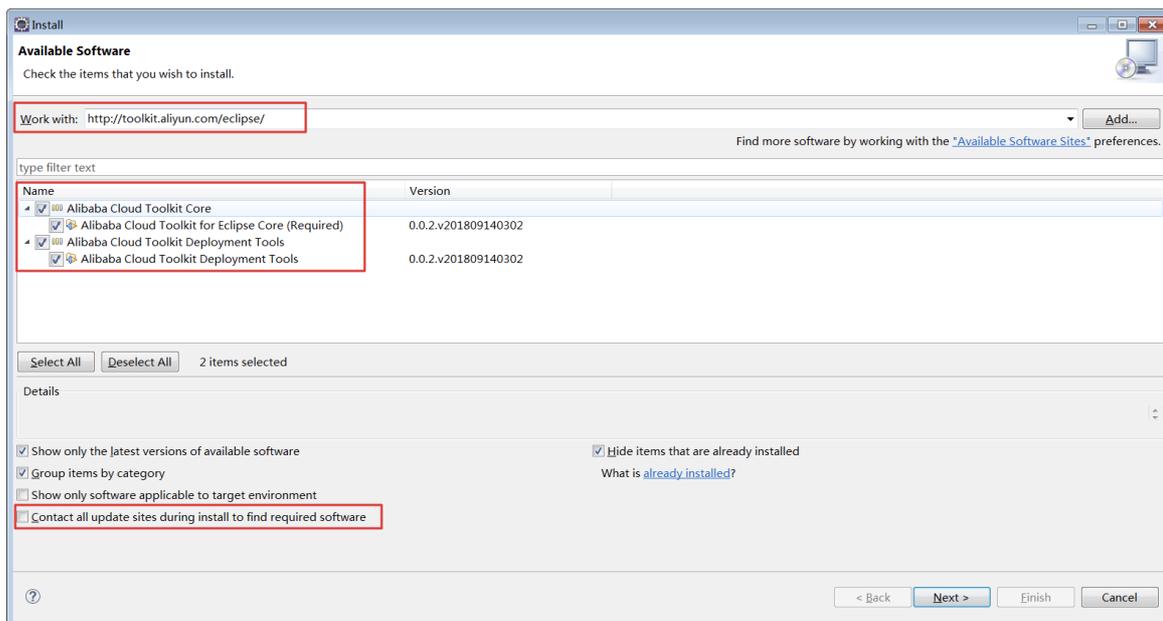
您在本地完成应用的开发、调试和测试后，可以使用在Eclipse中安装的Cloud Toolkit插件连接到云端部署环境并将应用部署到EDAS的ECS集群中。本文将向您介绍如何在Eclipse中安装Cloud Toolkit，并使用Cloud Toolkit部署应用到EDAS。

### 前提条件

- 下载并安装**JDK 1.8或更高版本**。
- 下载并安装适用于Java EE开发人员的**Eclipse IDE、4.5.0（代号：Mars）或更高版本**。
- 已在EDAS控制台创建了应用并完成了首次部署，相关操作请参见**在ECS集群中创建并部署应用**。

### 步骤一：安装Cloud Toolkit

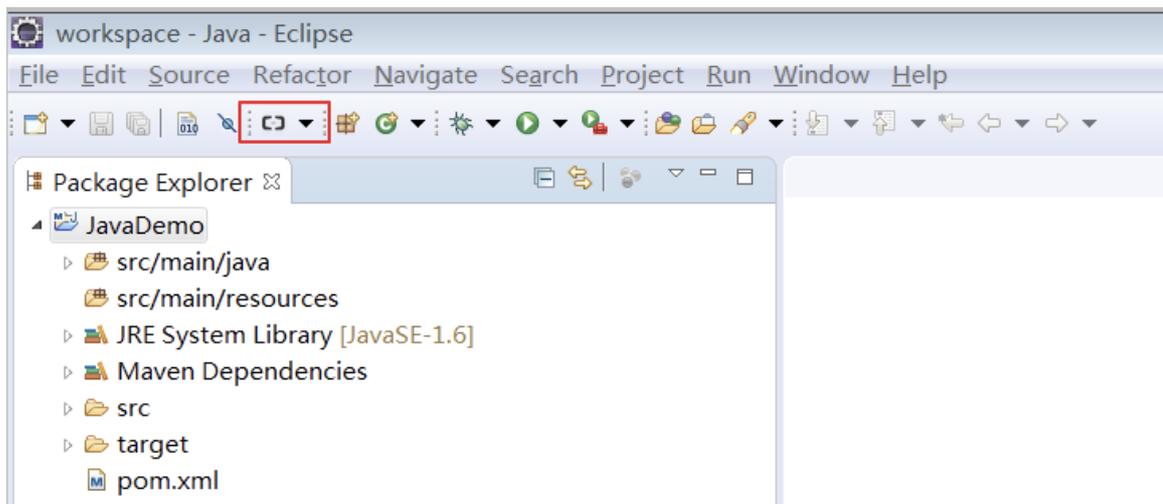
1. 启动Eclipse。
2. 在菜单栏中选择**Help > Install New Software**。
3. 在**Available Software**对话框的**Work with**文本框中输入Cloud Toolkit for Eclipse的URL `http://toolkit.aliyun.com/eclipse/`。
4. 在下面的列表区域中勾选需要的组件**Alibaba Cloud Toolkit Core**和**Alibaba Cloud Toolkit Deployment Tools**，并在下方**Details**区域中取消选中**Contact all update sites during install to find required software**。完成组件选择之后，单击**Next**。



5. 按照Eclipse安装页面的提示，完成后续安装步骤。

**注意** 安装过程中可能会提示没有数字签名，单击Inst all anyway即可。

6. Cloud Toolkit插件安装完成后，重启Eclipse，您可以在工具栏看到Alibaba Cloud Toolkit的图标。

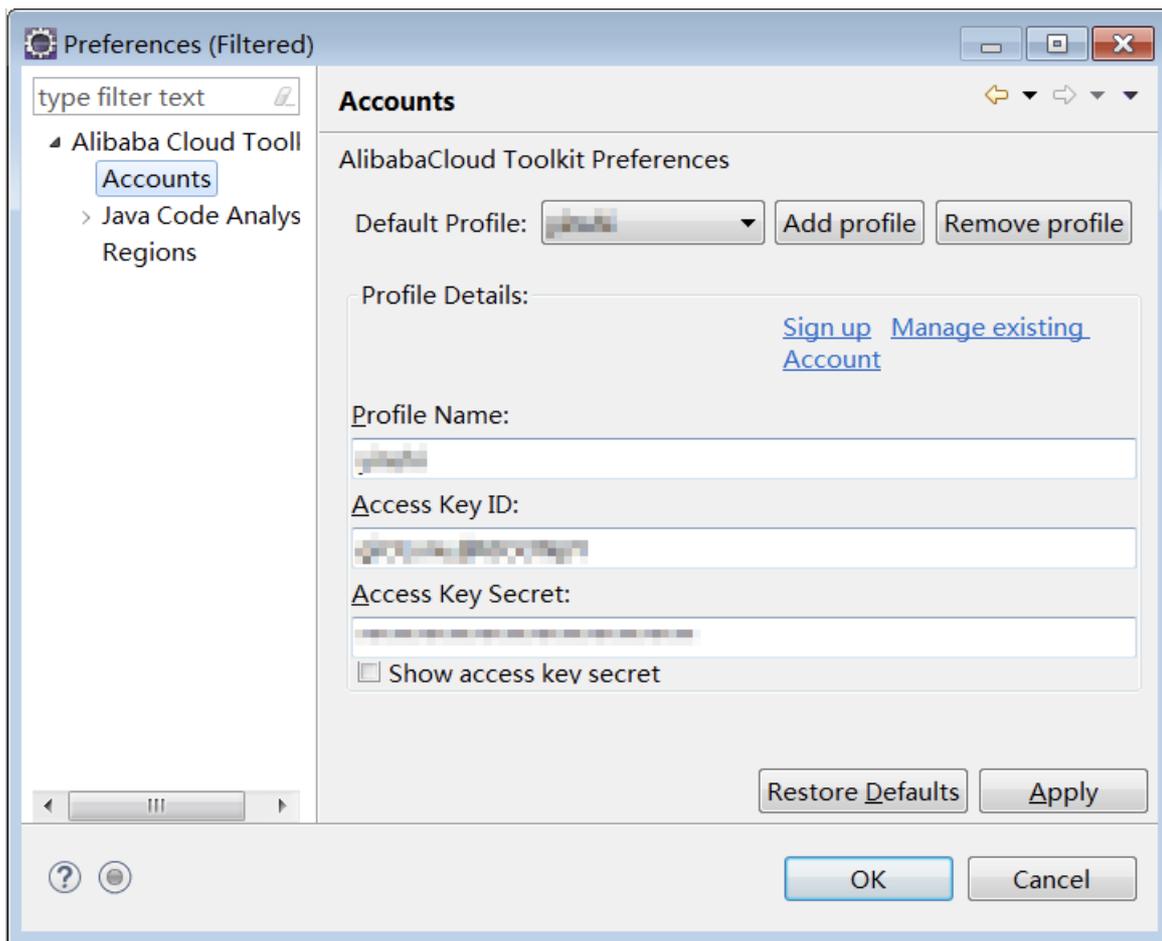


## 步骤二：配置Cloud Toolkit账号

您需使用AccessKey ID和AccessKey Secret来配置Cloud Toolkit的账号。

1. 启动Eclipse。
2. 在工具栏单击Alibaba Cloud Toolkit图标右侧的下拉按钮，在下拉菜单中单击Alibaba Cloud Preference。
3. 在Preference (Filtered) 对话框的左侧导航栏中选择Alibaba Cloud Toolkit > Accounts。
4. 在Accounts界面中设置AccessKey ID和AccessKey Secret，然后单击OK。

**注意** 如果您使用RAM用户的AccessKey ID和AccessKey Secret，请确保该RAM用户至少拥有部署应用的权限。具体操作，请参见[管理RAM用户](#)。



- 如果您已经注册过阿里云账号，在Accounts界面中单击Manage existing Account，进入阿里云登录页面。用已有账号登录后，将光标放置在右上角用户头像上，在列表中单击AccessKey管理，在AccessKey管理页面获取AccessKey ID和AccessKey Secret。
- 如果您还没有阿里云账号，在Accounts界面中单击Sign up，进入阿里云账号注册页面，注册账号。注册完成后按照上述方式获取AccessKey ID和AccessKey Secret。

? **说明** 如果使用EDAS专有云企业版，还需要按以下步骤在Cloud Toolkit中配置Endpoint。其中，Endpoint请联系EDAS技术支持获取。

- i. 在Preference (Filtered) 对话框的左侧导航栏中单击Appearance & BehaviorEndpoint。
- ii. 在Endpoint界面中设置Endpoint，配置完成后，单击Apply and Close。

### 步骤三：将应用部署到EDAS

目前支持使用Cloud Toolkit插件将应用通过WAR包或JAR包部署到EDAS。

1. 在Eclipse界面左侧的Package Explorer中右键单击您的应用工程名，在弹出的下拉菜单中选择Alibaba Cloud > Deploy to EDAS。
2. 在Deploy to EDAS的运行配置页面，配置应用部署参数，然后单击Deploy。

? **说明** 如果您还没有在EDAS上创建应用，在对话框右上角单击Create application on EDAS console...，跳转到EDAS控制台创建应用。

在配置页面中根据您的实际需求选择应用的Region、Namespace、Application和Group。

参数	描述
Region	应用所在地域。
Namespace	应用所在微服务空间。
Application	应用名称。
Group	应用分组。

**注意** 如果在应用列表中获取不到应用，请参见。

i. 设置构建方式。

- **Maven Build**：选择Maven Build方式来构建应用时，系统会默认添加一个Maven任务来构建部署包。
- **Upload File**：选择Upload File方式来构建应用时，选择上传您的WAR包或者JAR包，然后进行部署。

ii. 设置应用的版本描述信息和分批部署信息。

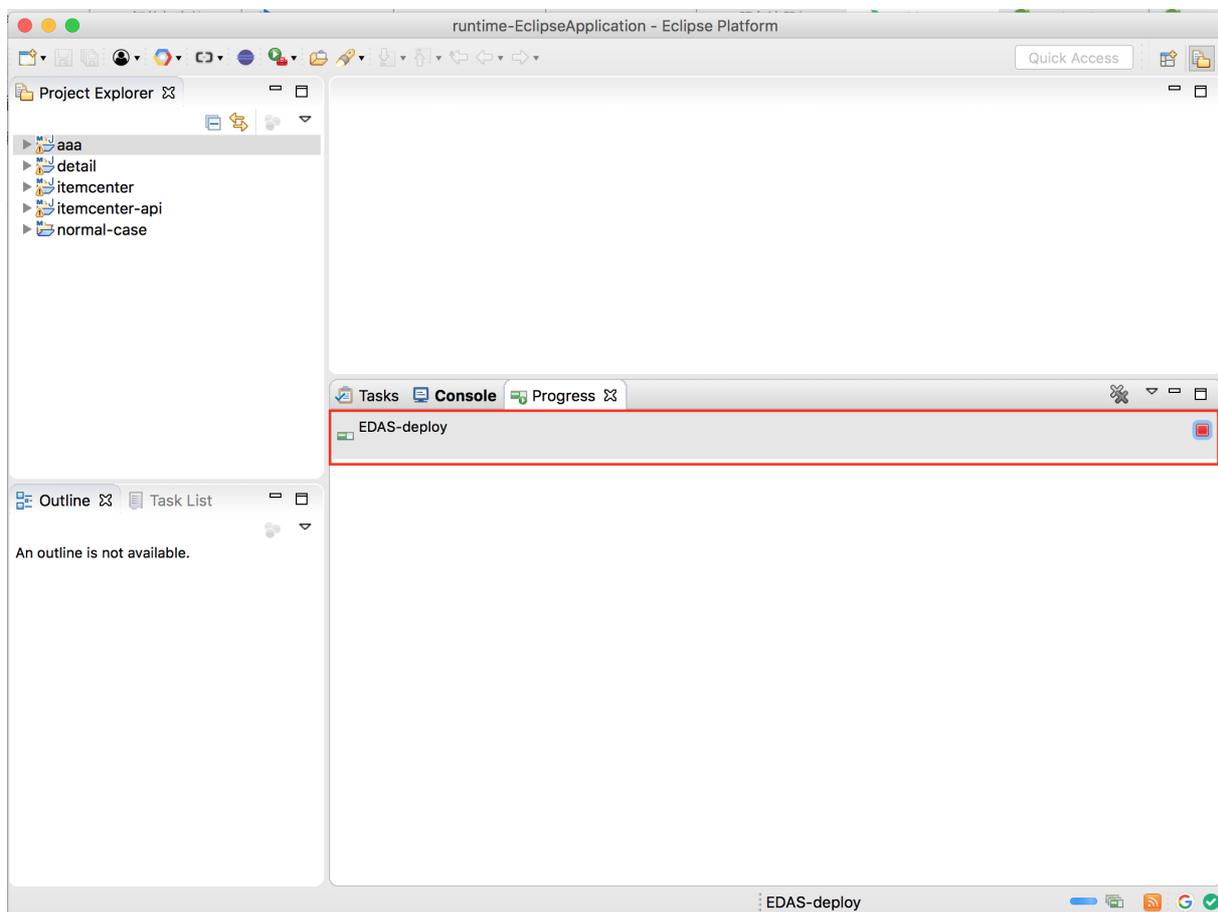
- **Version**: 部署版本。
- **Description**: 部署信息描述。
- **Batch**: 分批数。如果您的应用有多个分组，并且在部署时选择部署全部分组，那么将会自动按照分组粒度来分批，**Batch**值不用设置。
- **BatchWaitTime**: 分批部署等待时间，单位为分钟。

 **注意** 如果您的插件界面没有分批部署设置模块，请将您的插件升级至最新版本。

部署开始后，Eclipse的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

## 步骤四：终止Cloud Toolkit插件运行

在插件运行过程中，如果想停止插件运行，在Progress页面右上角单击终止按钮，终止EDAS-deploy进程。



## 常见问题：应用列表获取不到应用

通常出现这种情况为使用EDAS子账号或RAM用户来部署应用，且EDAS子账号或RAM用户没有同步到EDAS系统或者没有进行正确授权，从而导致在应用列表下拉框中看不到应用。您可以通过RAM授权或EDAS子账号授权来确保EDAS子账号或RAM用户已经同步到EDAS并且得到授权。

- RAM授权  
该授权方式可使RAM用户访问EDAS的所有资源。
  - i. 在RAM控制台左侧导航栏中选择人员管理 > 用户。

- ii. 在用户页面上找到需要授权的RAM用户，单击操作列中的添加权限。
  - iii. 在添加权限面板的选择权限区域中，搜索AliyunEDASFullAccess权限，单击权限策略将其添加至右侧的已选择列表中，然后单击确定。
  - iv. 在添加权限的授权结果页面上，查看授权信息摘要，并单击完成。
  - v. 使用阿里云账号登录EDAS控制台，在左侧导航栏选择系统管理 > 子账号，单击子账号页面右上角的同步子账号。
- EDAS子账号授权  
该授权方式可使子账号细粒度授权访问EDAS的资源。
    - i. 使用阿里云账号登录EDAS控制台。
    - ii. 在左侧导航栏选择系统管理 > 角色，单击角色页面右上角的创建角色。
    - iii. 输入一个角色名称，在可选权限区域框中，选择应用管理 > 应用列表 > 基本信息 > 部署应用，单击添加将部署应用角色添加到已选权限，然后单击确定。
    - iv. 在左侧导航栏选择系统管理 > 子账号，单击子账号页面右上角的同步子账号。
    - v. 选择需进行授权的子账号，在操作列单击管理角色，在左侧穿梭框中搜索并选择上面创建的角色，将该角色添加到右侧已选角色列表中，然后单击确定。
    - vi. 选择需进行授权的子账号，在操作列单击授权应用，选择应用添加到右侧列表进行授权，然后单击确定。

## 4.2.5. 使用toolkit-maven-plugin插件升级应用（ECS集群）

您可以使用toolkit-maven-plugin插件升级已经部署在EDAS的Spring Cloud、Dubbo和HSF应用。

### 背景信息

toolkit-maven-plugin是一个开源工具，帮助您构建一个完整的生命周期框架，完成项目的基础工具建设。更多信息，请参见[toolkit-maven-plugin概述](#)。

toolkit-maven-plugin也支持升级EDAS K8s集群中的应用。具体操作，请参见[使用Cloud Toolkit插件单批发布应用（K8s）](#)。

### 操作步骤

1. 在本地应用工程的pom.xml文件中添加toolkit-maven-plugin插件的依赖。

```
<build>
  <plugins>
    <plugin>
      <groupId>com.alibaba.cloud</groupId>
      <artifactId>toolkit-maven-plugin</artifactId>
      <version>1.1.5</version>
    </plugin>
  </plugins>
</build>
```

2. 在工程的根目录下创建配置文件 `.edas_config.yaml`，在文件中配置部署参数。
  - 以 `.` 开头的文件为IDE的默认配置文件。在未指定其它配置文件时，插件会默认使用根目录下的 `.edas_config.yaml` 文件。
  - 如果创建了其它非以 `.` 开头的配置文件，也可以在打包工程时，通过设置参数 `-Dedas_config=xx`

x 来指定该配置文件。

- 如果既存在默认配置文件，也通过参数指定了其它配置文件，那么插件会使用参数指定的配置文件。

**说明** 如果当前工程为一个Maven工程的子模块，那么默认配置文件应该存放在当前工程目录下，而不是整个Maven工程的根目录下。关于多模块工程部署更多方式，请参见[部署多模块工程](#)。

典型升级场景的配置示例如下：

- 升级应用所有分组

例如在北京地域有一个ID为 eb20\*\*\*\*-e6ee-4f6d-a36f-5f6a5455\*\*\*\* 的应用，需要升级该应用的所有分组。配置示例如下：

```
env:
  region_id: cn-beijing
app:
  app_id: eb20****-e6ee-4f6d-a36f-5f6a5455****
```

配置参数的值请以应用相关的实际信息替换。

- 升级应用指定分组并标识部署包版本

例如在北京地域有一个ID为 eb20dc8a-e6ee-4f6d-a36f-5f6a5455\*\*\*\* 的应用，需要升级的应用分组ID为 06923bb9-8c5f-4508-94d8-517b692f\*\*\*\* ，部署包版本为1.2。配置示例如下：

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

实际使用时，可能会有更复杂的场景，需要配置更多参数，请参见[部署应用配置参数说明](#)。

3. 在本地新建一个YAML账号文件，在文件中配置AccessKey ID和AccessKey Secret。

推荐使用RAM用户（需已授予应用管理权限）的AccessKey ID和AccessKey Secret，以便控制应用权限来提高应用的安全性。RAM用户的AccessKey ID和AccessKey Secret可以在阿里云[AccessKey管理页面](#)查看。

账号文件示例如下：

```
access_key_id: abc
access_key_secret: 1234567890
```

**说明** 示例中的参数值只是示例，请替换成您实际的AccessKey ID和AccessKey Secret。该配置中AccessKey ID和AccessKey Secret只会用来生成请求签名，不会用于网络传输等其他用途。

4. 在本地工程目录（如果是Maven多模块则进入子模块目录）中，执行以下命令，将应用打包并部署到EDAS。

```
mvn clean package toolkit:deploy -Daccess_key_file={账号文件路径}
```

命令参数说明：

- toolkit:deploy：使用toolkit-maven-plugin插件部署应用的参数。只有添加此参数，才会在应用打包成功后进行部署。
- access\_key\_file：账号文件路径参数。

 **说明** 更多指定密钥对的方式, 请参见[账号配置及账号优先级说明](#)。

- 还可以通过设置 `-Djar_path` 参数指定部署包文件路径, 跳过打包步骤而直接使用已有的部署包进行部署。部署包路径支持绝对路径和相对路径。

```
mvn toolkit:deploy -Daccess_key_file={账号文件路径} -Djar_path=target/demo.jar
```

 **注意** 您还可以在env中配置账号文件路径, 请参见[部署应用配置参数说明](#), 上述-D方式优先级大于env中的配置。

## 结果验证

1. 执行应用打包、部署命令后, 在IDE的Terminal中查看输出日志。
  - 显示以下信息, 并提示 `BUILD SUCCESS`, 表示应用部署成功。

```
[INFO] Deploy application successfully!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 55.939 s
[INFO] Finished at: 2021-01-20T14:28:26+08:00
[INFO] -----
```

- 如果输出日志中包含 `ERROR` 相关信息, 请根据日志排查失败原因。解决后再重新部署。
2. 登录EDAS控制台, 查看变更记录和应用详情, 检查应用是否升级(部署)成功。更多操作, 请参见[结果验证](#)。

## 专有云支持

toolkit-maven-plugin插件可以在专有云企业版V3.8.0及以上版本使用, 在使用前需要在插件中配置EDAS Endpoint。相关配置, 请参见[部署应用配置参数说明](#)。请联系EDAS技术支持获取Endpoint。

## 问题反馈

如果您在使用工具部署应用的过程中有任何疑问, 欢迎您扫描下面的二维码或搜索群号34556175, 加入钉钉群进行反馈。

## 4.2.6. 使用CLI快速部署应用至ECS集群

使用CLI快速部署应用至ECS集群

阿里云命令行工具 (Alibaba Cloud Command Line Interface) 可直接访问EDAS的OpenAPI, 您可使用SDK调用EDAS的API来部署Spring Cloud、Dubbo和HSF框架的应用。本文介绍了如何使用阿里云CLI来部署Demo应用至EDAS的ECS集群。

### 前提条件

[创建ECS实例](#)

### 安装阿里云CLI

阿里云CLI支持macOS、Linux和Windows。

各操作系统的安装步骤请参见:

- [在macOS上安装阿里云CLI](#)



```
#!/bin/bash
# 希望部署的地域REGION="cn-beijing"
# ECS实例ID
ECS_ID="i-2z*****b6"
# ECS所在的VPC ID
VPC_ID="vpc-t*****c"
# 微服务空间名 (若不存在会自动创建)
NAMESPACE="myNamespace"
# 集群名 (会自动创建)
CLUSTER_NAME="myCluster"
# 应用名APP_NAME="myApp"
# Step1: 创建微服务空间aliyun edas InsertOrUpdateRegion --RegionTag $REGION:$NAMESPACE --RegionName $NAMESPACE --region $REGION --endpoint "edas.cn-beijing.aliyuncs.com" >> /dev/null
# Step2: 创建集群CLUSTER_ID=`aliyun edas InsertCluster --ClusterName $CLUSTER_NAME --ClusterType 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'`
# Step3: 导入ECS实例aliyun edas TransformClusterMember --InstanceIds $ECS_ID --TargetClusterId $CLUSTER_ID --Password Hello1234 >> /dev/null
for 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'`
# Step4: 创建应用APP_ID=`aliyun edas InsertApplication --ApplicationName $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"
```

## 使用CLI快速部署应用

在阿里云CLI中执行以下代码快速部署应用。

 **注意** 以下代码内的参数均为示例参数，APP\_ID和GROUP\_ID为应用配置参数，请替换成您的应用参数。

```
#!/bin/bash
# 待部署应用ID (详情请参见步骤5。)
APP_ID="87a6*****4d1"
# 应用分组ID
GROUP_ID="54b*****f27"
# 上传的OSS Bucket名 (该Bucket需要公共可读)
OSS_BUCKET="eda*****mo"
# 安装包文件 (由您的CI系统构建生成)
PACKAGE="hello-edas.war"
# Step1: 上传部署包到OSS
aliyun oss cp -f $PACKAGE oss://$OSS_BUCKET/$PACKAGE >> /dev/null
PKG_URL=`aliyun oss sign oss://$OSS_BUCKET/$PACKAGE|head -1`
# Step2: 发起部署请求CO_ID=`aliyun edas DeployApplication --AppId $APP_ID --PackageVersion $VERSION --DeployType url --WarUrl "${PKG_URL}" --GroupId $GROUP_ID | grep '.*"ChangeOrderId":' | sed -E 's/.*"ChangeOrderId":\s"([a-z0-9-]*)".*/\1/g`
# Step3: 等待部署完成for i in `seq 300`
do
    STATUS=`aliyun edas GetChangeOrderInfo --ChangeOrderId $CO_ID | grep '.*"PipelineStatus":' | sed -E 's/.*"PipelineStatus":\s(.).*/\1/g`
    [[ 2 = ${STATUS} ]] && break
    sleep 1
done
```

若您不知如何设置以上参数值，请根据以下步骤获取。

1. 登录EDAS控制台。
2. 在左侧导航栏选择应用管理 > 应用列表，在应用列表页面单击需要部署的应用名称。
3. 在应用基本信息页面右上角单击部署应用。
4. 在部署模式选择页面的常规发布（单批/多批）区域的右上角单击开始部署。
5. 在常规发布（单批/多批）页面下方展开生成Maven插件配置，即可获取部署应用的参数信息。

```
生成Maven插件配置

# 此配置根据您的输入自动生成，关于配置使用详情参考如下文档：
# //help.aliyun.com/document_detail/.....html
---
env:
  region_id: cn-beijing
app:
  app_id: €.....
  group_id: all
  batch: 1
  batch_wait_time: 0
```

## 执行结果

应用部署完成后您可参照以下步骤查看应用的部署情况。

1. 登录EDAS控制台。
2. 在左侧导航栏选择应用管理 > 应用列表。
3. 单击创建的应用名称，在应用基本信息页面单击实例部署信息页签。  
如果应用实例的运行状态为运行正常则说明部署成功。

❓ 说明 如果应用部署失败，您可以参考变更流程错误码说明进行问题排查，详情请参见[变更流程错误码说明](#)。

## 相关链接

- [在ECS集群中创建并部署应用](#)
- [使用toolkit-maven-plugin插件升级应用（ECS集群）](#)
- [使用云效2020部署Java应用至ECS集群](#)
- [使用IntelliJ IDEA部署应用到EDAS](#)

## EDAS开发者工具交流群

如果您在EDAS中使用开发者工具的过程中有任何疑问或建议，请提交[工单](#)，或使用钉钉搜索钉钉群号 [34556175](#) 加入钉钉群进行反馈。

# 4.3. 分批发布（ECS）

## 4.3.1. 使用控制台分批发布应用

应用规模较大或服务架构比较复杂，可以按批次进行部署。每次仅对应用的一部分实例进行升级，直至全量升级完成。本文介绍如何使用EDAS控制台分批发布应用。

### 背景信息

如果一个应用有多个应用实例，可以分批发布。ECS集群中还可能包含实例分组，分组中有多个应用实例，也可以基于实例分组进行分批发布。

### 操作步骤

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏单击[应用列表](#)。
3. 在[应用列表](#)页面的顶部菜单栏选择地域，并在页面上方选择微服务空间，在集群类型列表中选择[ECS集群](#)，然后在应用列表中单击目标应用名称。
4. 在[应用信息](#)页面右上角单击[部署应用](#)。
5. 在[部署模式选择](#)页面的[常规发布（单批/多批）](#)区域右上角单击[开始部署](#)。
6. 在[常规发布（单批/多批）](#)页面设置新版本应用的部署包和相关信息，设置[发布策略](#)，然后单击[确定](#)。

## i. 设置部署包和相关信息。

参数	说明
应用部署方式	<p>应用部署方式由当前应用首次部署的方式决定，不可修改。例如，首次部署应用时选择了JAR包部署，升级应用时只能选择JAR包部署。</p> <p> <b>说明</b> 历史版本用于回滚应用。更多信息，请参见<a href="#">使用控制台回滚应用 (ECS集群)</a>。</p>
文件上传方式	<p>新版本应用部署包的上传方式，包含上传部署包 (JAR包或WAR包) 和部署包 (JAR包或WAR包) 地址。部署包类型同样由当前应用首次部署的方式决定。</p> <ul style="list-style-type: none"> <li>■ <b>上传JAR包或上传WAR包</b>：单击<b>选择文件</b>，选择本地的新版本应用部署包。</li> <li>■ <b>JAR包地址或WAR包地址</b>：如果新版本应用部署到已经上传到云端（例如OSS），在文本框中输入部署包的云端地址。</li> </ul>
版本	新版本应用的版本。可以输入新版本的版本号，也可以单击用 <b>时间戳</b> 作为版本号。
(可选) 描述	新版本应用的描述信息。

## ii. 设置发布策略。

参数	说明
分组	<p>分批发布的目标分组。</p> <ul style="list-style-type: none"> <li>■ 如果当前应用仅有一个分组，在列表中选择<b>所有分组</b>，分组后的括号中显示当前应用的最大实例数。列表中还包含<b>默认分组</b>，和<b>所有分组</b>实际上是一样的。</li> <li>■ 如果当前应用包含多个分组，在列表中选择要分批发布的目标分组。</li> </ul> <p> <b>说明</b> 如果选择<b>所有分组</b>，则会对应用所有分组按设置的<b>每组批次</b>进行分批发布。</p>
每组批次	指定分组中分批发布批次数，单批发布的最小实例数为1。每组批次默认为2，请根据实际需求选择。
分批方式	<p>每批发布间的处理方式，包含<b>自动</b>和<b>手动</b>。</p> <ul style="list-style-type: none"> <li>■ <b>自动</b>：按照<b>分批等待时间</b>自动发布剩余批次的发布。</li> <li>■ <b>手动</b>：需要手动触发下一个批次的发布。</li> </ul>
分批等待时间	每批次间的发布时间间隔，单位为分钟。根据实际需求，在列表中选择。
Java环境	新版本应用的Java环境，包含 <b>OpenJDK 8</b> 、 <b>OpenJDK 7</b> 和 <b>Dragonwell 8</b> 。请根据实际需求选择。

 **说明** 发布策略设置完成后，在生成Maven插件配置区域会生成当前发布策略的脚本。使用toolkit-maven-plugin插件进行分批发布时，可以直接在使用脚本中使用。具体操作，请参见[使用toolkit-maven-plugin插件分批发布应用 \(ECS集群\)](#)。

- iii. 单击确定。  
分批发布启动后，会自动跳转到变更详情页面，在变更详情中可以跟踪分批发布的进度和状态。具体操作，请参见[查看应用变更](#)。

## 结果验证

1. 查看变更记录。
  - i. 在应用详情页左侧导航栏单击变更记录。
  - ii. 在变更记录页面的变更状态列中查看此次分批发布的结果是否为执行成功。
    - 如果为执行成功，说明分批发布成功。
    - 如果为执行失败，说明分批发布失败。在操作列单击查看，通过变更详情排查失败原因。
2. 查看应用详情。
  - i. 在应用详情页左侧导航栏单击基本信息。
  - ii. 在基本信息页面单击基本信息页签，在基本信息页签的应用信息区域中查看应用的部署包是否为升级的部署包。
  - iii. 在基本信息页面单击实例部署信息页签，在实例部署信息页签查看目标分组中各实例的运行状态是否为运行正常。

## 4.3.2. 使用API在ECS集群中分批发布应用

本文介绍如何使用EDAS提供的Java SDK调用API在EDAS ECS集群中分批发布应用。

### 前提条件

在EDAS ECS环境中分批发布应用前，您必须提前完成以下准备工作：

- 安装EDAS的Java SDK。更多信息，请参见[Java SDK接入指南](#)。
- 确定创建应用的地域，假设为 `cn-hangzhou`。
- 创建微服务空间。具体操作，请参见[创建命名空间](#)。  
如果已经存在命名空间，您可以调用 `ListUserDefineRegion` 接口查询用户自定义命名空间列表，获取目标命名空间的 `RegionId`。
- 在ECS集群中创建应用，具体操作，请参见[使用API创建应用 \(ECS\)](#)。  
如果已经创建应用，您可以调用 `ListApplication` 接口查询应用列表，获取目标应用的 `AppId`，假设为 `6bbc57a2-a017-4bec-b521-49a15bd3****`。
- 已将应用部署包 (JAR包或WAR包) 上传至目标地址。本示例以OSS存储路径为例，假设为 `https://doc***.oss-cn-hangzhou.aliyuncs.com/sc-****-D-0.0.1-SNAPSHOT.jar`。
- 如果需要发布HSF应用，则需要调用 `ListBuildPack` 接口查询容器版本列表，获取容器版本 (ConfigId) 参数，假设为 `57`。

### 分批发布应用

以下代码适用于在ECS环境分批发布应用。代码中未设置环境变量等高级参数，如您需了解更多API参数信息，请参见[DeployApplication](#)。

② 说明 以下代码适用于分批发布Spring Cloud和Dubbo应用，如需发布HSF应用，则需要在代码中增加EDAS-Container信息：

```
//EDAS-Container构建包号，当发布HSF应用时，该参数必须指定。可以调用ListBuildPack接口获取，对应ConfigId参数。  
request.setBuildPackId(57);
```

```
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.exceptions.ServerException;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.edas.model.v20170801.DeployApplicationRequest;  
import com.aliyuncs.edas.model.v20170801.DeployApplicationResponse;  
public class DeployApplication {  
    public static void main(String[] args) {  
        //阿里云账号或RAM用户的AccessKey ID。  
        String aliyun_user_ak = "<yourAccessKeyId>";  
        //阿里云账号或RAM用户的AccessKey Secret。  
        String aliyun_user_sk = "<yourAccessKeySecret>";  
        //需要部署应用的所在地域ID。  
        String region_id = "cn-hangzhou";  
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak  
, aliyun_user_sk);  
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);  
        //创建API请求，并设置参数。  
        DeployApplicationRequest request = new DeployApplicationRequest();  
        //目标应用的AppId。  
        request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");  
        //应用描述信息。  
        request.setDesc("分批发布应用");  
        //部署类型，仅支持设置为url。  
        request.setDeployType("url");  
        //应用部署包 (WAR或JAR) 的URL地址，建议使用OSS存储的应用部署包路径。  
        request.setWarUrl("https://doc***.oss-cn-hangzhou.aliyuncs.com/sc-***-D-0.0.1-SNAPSHOT.jar");  
        //部署的应用部署包版本，最长64个字符，建议使用时间戳。  
        request.setPackageVersion("1619094147");  
        //部署分组ID。如果要部署到所有分组，参数设置为all。  
        //如果需要部署到具体应用实例分组，可以调用ListDeployGroup接口获取GroupId。  
        request.setGroupId("all");  
        //每组发布批次，设置为大于1，代表分批发布。  
        request.setBatch(2);  
        //分批方式，0表示自动，1表示批次间需要手动确认。建议使用自动发布方式。  
        request.setReleaseType((long) 0);  
        //分批等待时间，单位分钟。默认为0，表示不等待；最大值为5。  
        request.setBatchWaitTime(1);  
        //应用组件ID。您单批发布Spring Cloud或Dubbo时，JAR包部署需要选择JDK版本，WAR包部署需要选择Apache Tomcat版本；单批发布HSF应用时无需设置该参数。  
        //4代表Apache Tomcat 7.0.91，7代表Apache Tomcat 8.5.42，5代表OpenJDK 1.8.x，6代表OpenJDK 1.7.x。  
        request.setComponentIds("5");  
        try {
```

```
DeployApplicationResponse response = client.getAcsResponse(request);
System.out.println("Message=" + response.getMessage() + "\nChangeOrderId=" + response.getChangeOrderId());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

运行程序，返回的执行结果如下所示：

```
Message=success
ChangeOrderId=04c5638f-405e-41de-99b2-e0dccb2****
```

## 结果验证

您在分批发布应用后，可以调用GetChangeOrderInfo接口查看变更流程详情，获取分批发布应用的变更状态。具体API参数详情，请参见[GetChangeOrderInfo](#)。

调用GetChangeOrderInfo接口查看变更流程详情，返回的执行结果如下：

```
{
  "Message": "success",
  "RequestId": "01FAA54A-43C0-4681-B985-33BA44D64340",
  "Code": 200,
  "changeOrderInfo": {
    "Status": 2,
    "Desc": "分批发布",
    "PipelineInfoList": {
      "PipelineInfo": [
        {
          .....
          "PipelineId": "4ff02b63-e14d-4124-86ff-22239f80****"
          .....
          "PipelineId": "6dbbfff8-bcf7-4300-aa59-ccb06909****"
          .....
          //此处未罗列全部执行结果，仅供参考。
        }
      ]
    }
  }
}
```

请查看上述执行结果中的changeOrderInfo.Status参数值，通过该值判断分批发布应用的变更是否成功。changeOrderInfo.Status的取值如下：

- 0: 准备
- 1: 执行中
- 2: 执行成功
- 3: 执行失败
- 6: 终止
- 8: 手动分批发布模式下，等待手工确认执行下一批。
- 9: 自动分批发布模式下，等待下一批执行中。

- 10: 系统异常执行失败

## 手动确认下一批发布

如果您使用的是手动确认分批发布方式，则需要调用ContinuePipeline接口手动确认下一批发布。

1. 调用GetChangeOrderInfo接口，获取分批发布应用变更的相关批次的流程ID (PipelineId)。

调用GetChangeOrderInfo接口，返回结果如下：

```
{
  "Message": "success",
  "RequestId": "01FAA54A-43C0-4681-B985-33BA44D64340",
  "Code": 200,
  "changeOrderInfo": {
    "Status": 8,
    "Desc": "分批发布应用",
    "PipelineInfoList": {
      "PipelineInfo": [
        {
          .....
          "PipelineId": "4ff02b63-e14d-4124-86ff-22239f80****"
          .....
          "PipelineId": "6dbbfff8-bcf7-4300-aa59-ccb06909****"
          .....
          //此处未罗列全部执行结果，仅供参考。
        }
      ]
    }
  }
}
```

**说明** 如果您设置的是应用的ECS实例分2批发布+手动分批发布方式，在调用GetChangeOrderInfo接口返回结果中会出现2个流程ID (PipelineId)，第二次出现的流程ID (PipelineId) 代表需要手动确认发布的批次。按此规律，如果是应用的ECS实例分3批发布+手动分批发布方式，则会出现3个流程ID (PipelineId)，第二次和第三次出现的流程ID (PipelineId) 代表需要手动确认发布的批次。

2. 调用ContinuePipeline接口，手动确认下一批发布。

返回结果如下：

```
{
  "Message": "success",
  "RequestId": "B8ECF6F7-672D-40E9-91DD-1C33F06D4FD8",
  "Code": 200
}
```

**说明** 如果是需要手动确认多批的场景，请选择正确批次的流程ID (PipelineId) 并重复执行此步骤。

## 4.3.3. 使用toolkit-maven-plugin插件分批发布应用 (ECS集群)

您可以使用toolkit-maven-plugin插件分批发布已经部署在EDAS的Spring Cloud、Dubbo和HSF应用。

### 背景信息

toolkit-maven-plugin是一个开源工具，帮助您构建一个完整的生命周期框架，完成项目的基础工具建设。更多信息，请参见[toolkit-maven-plugin概述](#)。

toolkit-maven-plugin也支持升级EDAS K8s集群中的应用。具体操作，请参见[使用Cloud Toolkit插件单批发布应用 \(K8s\)](#)。

## 操作步骤

1. 在本地应用工程的pom.xml文件中添加toolkit-maven-plugin插件的依赖。

```
<build>
  <plugins>
    <plugin>
      <groupId>com.alibaba.cloud</groupId>
      <artifactId>toolkit-maven-plugin</artifactId>
      <version>1.1.5</version>
    </plugin>
  </plugins>
</build>
```

2. 在工程的根目录下创建配置文件 `.edas_config.yaml`，在文件中配置部署参数。
  - 以 `.` 开头的文件为IDE的默认配置文件。在未指定其它配置文件时，插件会默认使用根目录下的 `.edas_config.yaml` 文件。
  - 如果创建了其它非以 `.` 开头的配置文件，也可以在打包工程时，通过设置参数 `-Dedas_config=xx` 来指定该配置文件。
  - 如果既存在默认配置文件，也通过参数指定了其它配置文件，那么插件会使用参数指定的配置文件。

 **说明** 如果当前工程为一个Maven工程的子模块，那么默认配置文件应该存放在当前工程目录下，而不是整个Maven工程的根目录下。关于多模块工程部署更多方式，请参见[部署多模块工程](#)。

典型升级场景的配置示例如下：

例如在杭州地域有一个ID为 `eb20****-e6ee-4f6d-a36f-5f6a5455****` 的应用，包含多个应用分组。需要升级其中ID为 `62fe****-d931-4446-a30a-f396a6da****` 的分组，分为 `2` 批发布，每批间的等待时间间隔为 `2分钟`，配置示例如下：

```
env:
  region_id: cn-hangzhou
app:
  app_id: e130****-75d7-471e-9e31-95145701****
  group_id: 62fe****-d931-4446-a30a-f396a6da****
  batch: 2
  batch_wait_time: 3
```

实际使用时，可能会有更复杂的场景，需要配置更多参数，请参见[部署应用配置参数说明](#)。

3. 在本地新建一个YAML账号文件，在文件中配置AccessKey ID和AccessKey Secret。

推荐使用RAM用户（需已授予应用管理权限）的AccessKey ID和AccessKey Secret，以便控制应用权限来提高应用的安全性。RAM用户的AccessKey ID和AccessKey Secret可以在阿里云[AccessKey管理页面](#)查看。

账号文件示例如下：

```
access_key_id: abc
access_key_secret: 1234567890
```

**说明** 示例中的参数值只是示例，请替换成您实际的AccessKey ID和AccessKey Secret。该配置中AccessKey ID和AccessKey Secret只会用来生成请求签名，不会用于网络传输等其他用途。

4. 在本地工程目录（如果是Maven多模块则进入子模块目录）中，执行以下命令，将应用打包并部署到EDAS。

```
mvn clean package toolkit:deploy -Daccess_key_file={账号文件路径}
```

命令参数说明：

- `toolkit:deploy`：使用toolkit-maven-plugin插件部署应用的参数。只有添加此参数，才会在应用打包成功后进行部署。
- `access_key_file`：账号文件路径参数。

**说明** 更多指定密钥对的方式，请参见[账号配置及账号优先级说明](#)。

- 还可以通过设置 `-Djar_path` 参数指定部署包文件路径，跳过打包步骤而直接使用已有的部署包进行部署。部署包路径支持绝对路径和相对路径。

```
mvn toolkit:deploy -Daccess_key_file={账号文件路径} -Djar_path=target/demo.jar
```

**注意** 您还可以在env中配置账号文件路径，请参见[部署应用配置参数说明](#)，上述-D方式优先级大于env中的配置。

## 结果验证

1. 执行应用打包、部署命令后，在IDE的Terminal中查看输出日志。
  - 显示以下信息，并提示 `BUILD SUCCESS`，表示应用部署成功。

```
[INFO] Deploy application successfully!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 55.939 s
[INFO] Finished at: 2021-01-20T14:28:26+08:00
[INFO] -----
```

- 如果输出日志中包含 `ERROR` 相关信息，请根据日志排查失败原因。解决后再重新部署。
2. 登录EDAS控制台，查看变更记录和应用详情，检查应用是否升级（部署）成功。更多操作，请参见[结果验证](#)。

## 专有云支持

toolkit-maven-plugin插件可以在专有云曙光V3.8.0及以上版本使用，在使用前需要按照以下步骤在插件中配置EDAS Endpoint。相关配置，请参见[部署应用配置参数说明](#)。请联系EDAS技术支持获取Endpoint。

## 问题反馈

如果您在使用工具部署应用的过程中有任何疑问，欢迎您扫描下面的二维码或搜索群号34556175，加入钉钉群进行反馈。



## 4.3.4. 使用Cloud Toolkit分批发布应用

您可以使用Cloud Toolkit分批发布已经部署在EDAS的Spring Cloud、Dubbo和HSF应用。本文介绍如何使用Cloud Toolkit分批发布应用。

### 前提条件

- 下载并安装[JDK 1.8或更高版本](#)。
- 下载并安装[IntelliJ IDEA \(2018.3或更高版本\)](#)。

 说明 因JetBrains插件市场官方服务器在海外，如遇访问缓慢无法下载安装的，请加入文末交流群，向Cloud Toolkit产品运营获取离线包安装。

- 已在EDAS控制台创建了应用并完成了首次部署。具体操作，请参见[在ECS集群中创建并部署应用](#)。
- 已经在IDE中安装了Cloud Toolkit，并在Cloud Toolkit中配置了阿里云账号或RAM用户的AccessKey。具体操作，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)和[在Eclipse中安装和配置Cloud Toolkit](#)。

### 背景信息

Alibaba Cloud Toolkit (IDE插件，简称Cloud Toolkit) 是阿里云针对IDE平台为开发者提供的一款插件，用于帮助开发者高效开发并部署适合在云端运行的应用。更多信息，请参见[什么是Alibaba Cloud Toolkit](#)。

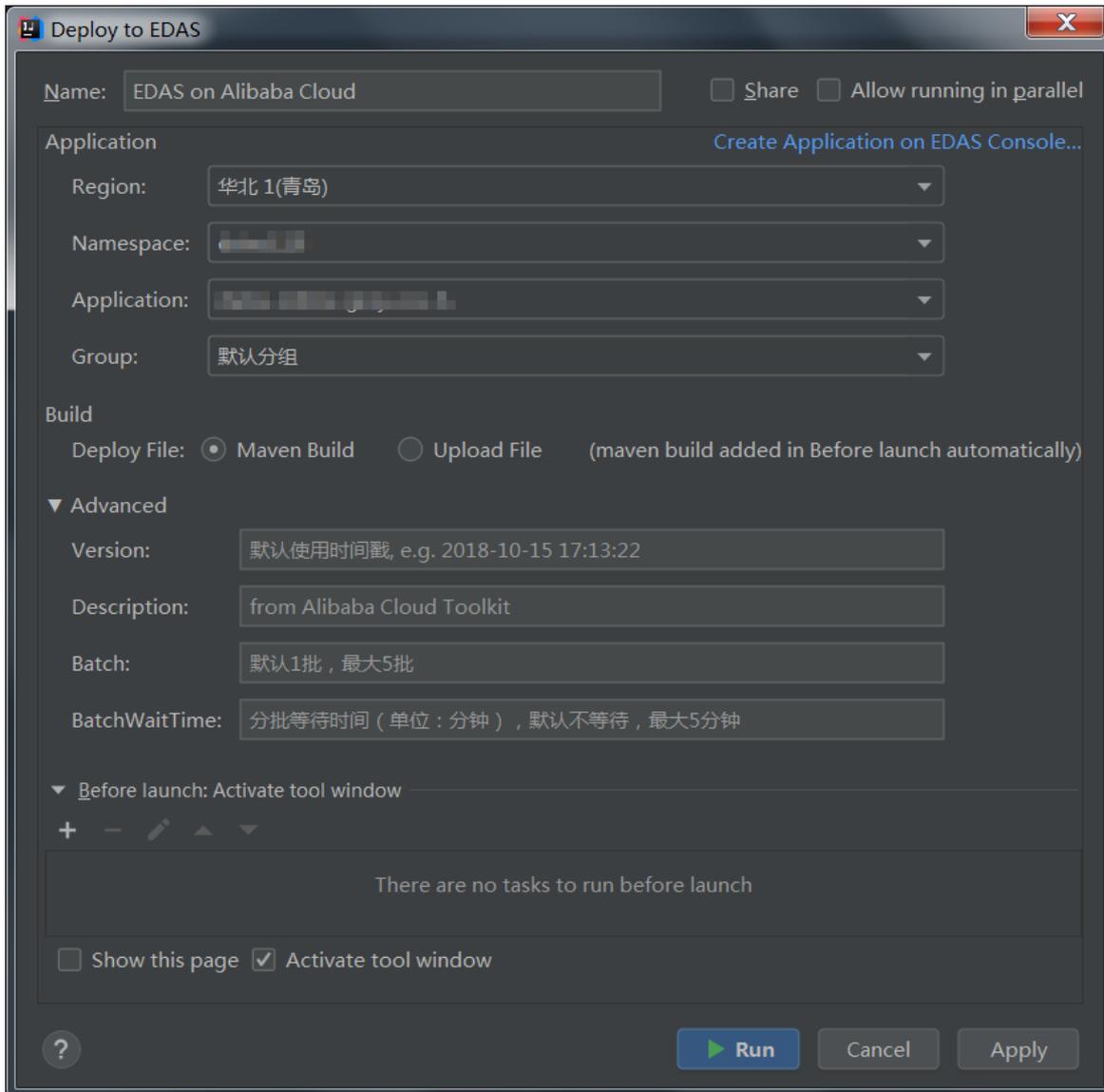
本文介绍在IntelliJ IDEA中如何使用Cloud Toolkit分批发布应用。您还可以：

- 使用Cloud Toolkit部署（单批发布）应用，具体操作，请参见[在IntelliJ IDEA中使用Cloud Toolkit部署应用 \(ECS集群\)](#)。
- 在Eclipse中使用Cloud Toolkit部署（单批发布）、升级（单批发布或分批发布）应用，具体操作，请参见[使用Eclipse部署应用到EDAS](#)。

### 操作步骤

1. 在IntelliJ IDEA的工具栏中单击Cloud Toolkit的图标，在下拉列表中选择Deploy to EDAS > EDAS for ECS Application。
2. 在Deploy to EDAS ECS Cluster对话框配置应用部署参数，然后单击Run。

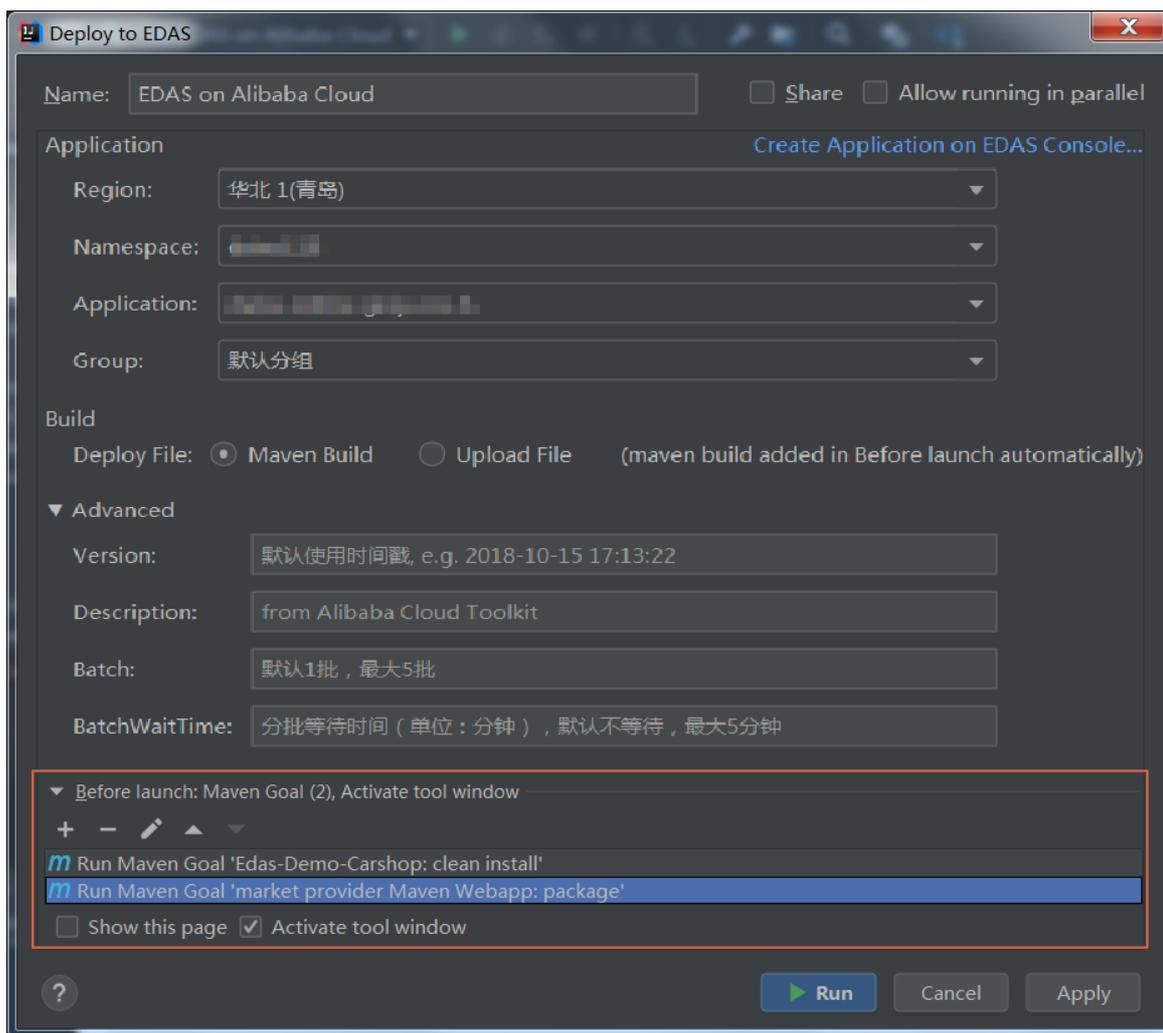
 说明 如果您还没有在EDAS上创建应用，在对话框右上角单击Create application on EDAS console...，跳转到EDAS控制台创建应用。具体操作，请参见[在ECS集群中创建并部署应用](#)。



参数	说明
<b>ECS Cluster</b>	
<b>Region</b>	目标应用所属地域。
<b>Namespace</b>	目标应用所属微服务空间。
<b>Application</b>	目标应用。 <div style="border: 1px solid #ccc; background-color: #e0f2f1; padding: 5px; margin-top: 10px;"> <span style="font-size: 1.2em;">🔊</span> <b>注意</b> 如果在应用列表中获取不到应用, 请参见<b>常见问题: 应用列表获取不到应用</b>进行操作排查。                 </div>
<b>Group</b>	目标应用分组。
<b>Build</b>	

参数	说明
Deploy File	新版本应用部署包文件形式，包含Maven Build和Upload File。请根据实际需求选择。 <ul style="list-style-type: none"><li>○ <b>Maven Build</b>：在部署应用前，通过Maven来构建新版本应用的部署包。</li><li>○ <b>Upload File</b>：在本地选择并上传新版本应用的部署包（WAR包或者JAR包）进行部署。</li></ul>
Version	新版本应用的版本号。默认使用时间戳作为版本号，例如2020-11-17 17:17:17。
Description	此次分批发布的描述信息。
Batch	应用分批发布的批次。默认为1批，最大为5批。 <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"><p> <b>说明</b> 如果目标应用有多个分组，并且需要部署到全部分组，那么将会自动按照分组粒度来分批，Batch不用设置。</p></div>
BatchWaitTime	每批次间的发布时间间隔，单位为分钟。默认为不等待，最大等待时间为5分钟。

在IntelliJ IDEA中安装的Cloud Toolkit内可以管理Maven的构建任务。您也可以在Deploy to EDAS ECS Cluster对话框的Before launch区域来添加、删除、修改和移动Maven构建任务。



分批发布启动后，IntelliJ IDEA的Terminal区域会打印部署日志。您可以根据日志信息查看部署进度和结果。

## 常见问题：应用列表获取不到应用

通常出现这种情况为使用EDAS子账号或RAM用户来部署应用，且EDAS子账号或RAM用户没有同步到EDAS系统或者没有进行正确授权，从而导致在应用列表下拉框中看不到应用。您可以通过RAM授权或EDAS子账号授权来确保EDAS子账号或RAM用户已经同步到EDAS并且得到授权。

- RAM授权

该授权方式可使RAM用户访问EDAS的所有资源。

- 在RAM控制台左侧导航栏中选择人员管理 > 用户。
- 在用户页面上找到需要授权的子用户，单击操作列中的添加权限。
- 在添加权限面板的选择权限区域中，搜索AliyunEDASFullAccess权限，单击权限策略将其添加至右侧的已选择列表中，然后单击确定。
- 在添加权限的授权结果页面上，查看授权信息摘要，并单击完成。
- 使用阿里云账号登录EDAS控制台，在左侧导航栏选择系统管理 > 子账号，单击子账号页面右上角的同步子账号。

- EDAS子账号授权

该授权方式可使子账号细粒度授权访问EDAS的资源。

- 使用阿里云账号登录EDAS控制台。

- ii. 在左侧导航栏选择系统管理 > 角色，单击角色页面右上角的创建角色。
- iii. 输入一个角色名称，在可选权限区域框中，选择应用管理 > 应用列表 > 基本信息 > 部署应用，单击添加将部署应用角色添加到已选权限，然后单击确定。
- iv. 在左侧导航栏选择系统管理 > 子账号，单击子账号页面右上角的同步子账号。
- v. 选择需进行授权的子账号，在操作列单击管理角色，在左侧穿梭框中搜索并选择上面创建的角色，将该角色添加到右侧已选角色列表中，然后单击确定。
- vi. 选择需进行授权的子账号，在操作列单击授权应用，选择应用添加到右侧列表进行授权，然后单击确定。

## 4.4. 金丝雀发布 (ECS)

### 4.4.1. 使用控制台金丝雀发布 (ECS集群)

如需升级部署在ECS集群中的Spring Cloud或Dubbo微服务应用，可以使用金丝雀发布进行小规模验证，验证通过后再全量升级。

#### 前提条件

请确保应用中至少包含两个实例分组，且至少两个分组下包含实例，才能进行金丝雀发布。创建实例分组及为分组添加ECS实例的操作步骤，请参见[使用控制台管理ECS集群中的应用实例分组](#)。

#### 使用限制

- HSF应用：不支持金丝雀发布。
- Dubbo应用：没有使用限制。
- Spring Cloud应用：如果应用依赖Deployment.Metadadata.Name或Deployment.Metadadata.Uid的功能与配置，请勿使用金丝雀发布，否则发布后该原生功能将出现异常。

#### 操作步骤

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏单击应用列表。
3. 在应用列表页面选择地域和微服务空间，然后单击具体应用名称。
4. 在应用详情页面右上角单击部署应用。
5. 在部署模式选择页面金丝雀发布（灰度）区域右上角单击开始部署。
6. 在灰度发布页面上传新版本应用的部署包、设置发布策略和灰度规则，然后单击确定。
  - i. 上传新版本应用的部署包。

\* 文件上传方式： 上传JAR包 下载样例工程

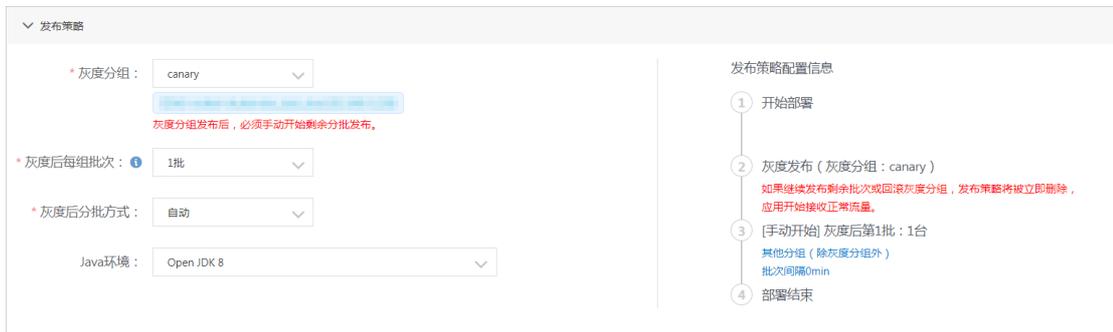
\* 上传JAR包：  选择文件

\* 版本： 1.1.0 用时间戳作为版本号

描述： 例如：“本次发布修复漏洞：”。长度限制在128字符以内。

ii. 在发布策略区域配置发布策略参数。

右侧的发布策略配置信息会根据配置显示灰度发布流程。



发布策略参数说明：

参数	描述
灰度分组	进行灰度发布的实例分组。
灰度后每组批次	<p>灰度分组发布后, 其它分组中的应用实例按照设置的批次完成分批发布。</p> <ul style="list-style-type: none"> <li>当所选分组为所有分组时, 会对应用下每一个分组都进行所选批次数的分批, 若分组实例数小于所选批次数, 则以实例数为准进行分批。</li> <li>当所选分组为指定分组时, 会对应用下指定分组进行所选批次数的分批。</li> </ul>
灰度后分批方式	<p>灰度后每组批次设置为2批或2批以上时需要设置, 包括 <i>自动</i> 和 <i>手动</i> 两种方式。</p> <ul style="list-style-type: none"> <li><i>自动</i>: 根据分批时间间隔自动分批发布。自动分批方式还需要设置 <b>分批等待时间</b>。 <b>分批等待时间</b> 包含 <b>不等待</b> 和 1~5 分钟。</li> <li><i>手动</i>: 手动触发下一个批次的发布。</li> </ul>
Java环境	该应用的运行环境, 根据实际需求选择。

iii. 设置灰度规则。

目前支持按内容灰度和按比例灰度两种方式设置灰度规则。

- 按内容灰度：单击创建新的入口流量规则，设置入口流量规则。

**说明** 您可以创建多个入口流量规则。

按内容灰度参数说明：

参数	描述
协议类型	包含 <i>Spring Cloud</i> 和 <i>Dubbo</i> ，按应用实际情况选择。 <ul style="list-style-type: none"> <li>■ <i>Spring Cloud</i>: 需要设置 <code>path</code>。</li> <li>■ <i>Dubbo</i>: 需要选择服务和方法。</li> </ul>
条件模式	包含同时满足下列条件和满足下列任一条件。
条件列表	<i>Spring Cloud</i> 和 <i>Dubbo</i> 协议的条件有所不同。包含 <i>Cookie</i> 、 <i>Header</i> 和 <i>Parameter</i> 3种方式，根据实际需求选择。 <ul style="list-style-type: none"> <li>■ <i>Spring Cloud</i>: 包含 <i>Cookie</i>、<i>Header</i> 和 <i>Parameter</i> 3种方式，根据实际需求设置。</li> <li>■ <i>Dubbo</i>: 根据应用实际的参数、参数值获取表达式设置。</li> </ul>

- 按比例灰度：设置流量比例，流量会按配置的比例被转发到当前的灰度分组。

灰度发布启动后，先在指定的灰度分组中部署新版本应用。基本信息页面提示 应用有变更流程正在执行，处于执行中状态。单击查看详情，进入变更详情页面查看部署进度和状态。

终止变更：应用处于金丝雀发布状态，并且该变更已经被终止。请先回滚应用，再进行其他操作。

7. 监控灰度流量是否符合预期，相关内容，请参见[监控灰度流量](#)。
8. 灰度流量验证完成后，在变更详情页面单击开始下一批，完成后续分批发布。

如果在验证过程中，发现新版本应用有问题，可以在变更详情页面右上角单击终止变更。终止变更后，基本信息页面提示 应用处于金丝雀发布状态，并且该变更已经被终止。请先回滚应用，再进行其他操作。

## 结果验证

灰度发布后，在基本信息页面查看部署包是否为新部署的应用版本。在实例部署信息页面查看应用实例的运行状态是否为运行正常。

## 4.4.2. 使用API金丝雀发布 (ECS集群)

本文介绍如何使用EDAS提供的Java SDK调用API在EDAS ECS环境中金丝雀发布应用。

### 前提条件

在EDAS ECS环境中金丝雀发布应用前，您必须提前完成以下准备工作：

- 安装EDAS的Java SDK。更多信息，请参见[Java SDK接入指南](#)。
- 确定创建ECS集群的地域。
- 创建微服务空间。具体操作，请参见[创建命名空间](#)。  
如果已经存在命名空间，您可以调用[ListUserDefineRegion](#)接口查询用户自定义命名空间列表，获取目标命名空间的RegionId。
- 在ECS集群中创建应用，具体操作，请参见[使用API创建应用 \(ECS\)](#)。  
如果已经创建应用，您可以调用[ListApplication](#)接口查询应用列表，获取目标应用的AppId，假设为6bbc57a2-a017-4bec-b521-49a15bd3\*\*\*\*。
- 已将应用部署包 (JAR包或WAR包) 上传至目标地址。本示例以OSS存储路径为例，假设为https://doc\*\*\*.oss-cn-hangzhou.aliyuncs.com/sc-\*\*\*\*-D-0.0.1-SNAPSHOT.jar。
- 请调用[ListDeployGroup](#)接口查询应用实例分组列表，获取应用实例分组的GroupId，假设为941be68c-4aac-48a1-88fe-c9ad1502\*\*\*\*。
- 如果需要创建HSF应用，则需要调用[ListBuildPack](#)接口查询容器版本列表，获取容器版本 (ConfigId) 参数，假设为57。

### 背景信息

使用API金丝雀发布应用，在首批灰度发布后，您必须手动确认继续剩余批次的发布。在确认剩余批次发布前，您可以进行小规模验证，验证后，再继续剩余批次的发布，即将应用全量升级到新版本。

使用API金丝雀发布应用，设置剩余批次 (除灰度发布分组外) 的发布方式时推荐选用自动分批发布。如果您选用手动分批发布方式，则还需要调用ContinuePipeline接口手动确认执行下一批发布。

### 金丝雀发布应用

#### 1. 运行程序，金丝雀发布应用。

以下代码适用于在ECS环境金丝雀发布应用。代码中未设置环境变量等高级参数，如您需了解更多API参数信息，请参见[DeployApplication EDAS开发指南](#)中的DeployApplication。

**说明** 以下代码适用于分批发布Spring Cloud和Dubbo应用，如需发布HSF应用，则需要在代码中增加EDAS-Container信息：

```
//EDAS-Container构建包号，当发布HSF应用时，该参数必须指定。可以调用ListBuildPack接口获取，  
对应ConfigId参数。  
request.setBuildPackId(57);
```

```
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.exceptions.ServerException;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.edas.model.v20170801.DeployApplicationRequest;  
import com.aliyuncs.edas.model.v20170801.DeployApplicationResponse;  
public class DeployApplication {  
    public static void main(String[] args) {  
        //阿里云账号或RAM用户的AccessKey ID
```

```
//阿里云账号或RAM用户的AccessKey ID
String aliyun_user_ak = "<yourAccessKeyId>";
//阿里云账号或RAM用户的AccessKey Secret。
String aliyun_user_sk = "<yourAccessKeySecret>";
//需要部署应用的所在地域ID。
String region_id = "cn-hangzhou";
DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak, aliyun_user_sk);
DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
//创建API请求，并设置参数。
DeployApplicationRequest request = new DeployApplicationRequest();
//目标应用的AppId。
request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");
//应用描述信息。
request.setDesc("金丝雀发布");
//部署类型，仅支持设置为url。
request.setDeployType("url");
//应用部署包 (WAR或JAR) 的URL地址，建议使用OSS存储的应用部署包路径。
request.setWarUrl("https://doc****.oss-cn-hangzhou.aliyuncs.com/sc-****-D-0.0.1-SNAPSHOT.jar");
//部署的应用部署包版本，最长64个字符，建议使用时间戳。
request.setPackageVersion("20210417.173831");
//是否为金丝雀发布，true表示金丝雀，false表示非金丝雀。金丝雀发布时必须设置分组ID，且后续分组批次由Batch控制。
request.setGray(true);
//金丝雀发布流量控制策略。示例策略表示：50%的流量会被转发到灰度分组。
request.setTrafficControlStrategy("{\"http\":{\"rules\":[{\"conditionType\":\"percent\",\"percent\":50}]}");
//部署分组ID。如果要部署到所有分组，参数设置为all。金丝雀发布时必须指定灰度发布分组的GroupId。
//如果需要部署到具体应用实例分组，可以调用ListDeployGroup接口获取GroupId。
request.setGroupId("941be68c-4aac-48a1-88fe-c9ad1502****");
//灰度发布后的每组发布批次，设置为大于1，代表分批发布。
request.setBatch(2);
//分批方式，0表示自动，1表示批次间需要手动确认。建议使用自动发布方式。
request.setReleaseType((long) 1);
//分批等待时间，单位分钟。默认为0，表示不等待；最大值为5。
request.setBatchWaitTime(1);
//应用组件ID。当创建的应用运行环境使用Apache Tomcat (适用于WAR包部署的Dubbo应用) 或者标准Java应用运行环境 (适用于JAR包部署的Spring Boot或Spring Cloud应用) 时需要指定。
//4代表Apache Tomcat 7.0.91, 7代表Apache Tomcat 8.5.42, 5代表OpenJDK 1.8.x, 6代表OpenJDK 1.7.x。
request.setComponentIds("5");
try {
    DeployApplicationResponse response = client.getAcsResponse(request);
    System.out.println("Message=" + response.getMessage() + "\nChangeOrderId=" + response.getChangeOrderId());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

运行程序，返回的执行结果如下所示：

```
Message=success
ChangeOrderId=4a3329de-2495-4c51-be33-f69ff264****
```

2. 调用 `GetChangeOrderInfo` 接口，获取金丝雀发布应用变更的相关批次的流程ID (PipelineId)。

调用 `GetChangeOrderInfo` 接口，返回结果如下：

```
{
  "Message": "success",
  "RequestId": "0EB98A7D-50CE-4F5F-855D-70BAF61DFB38",
  "Code": 200,
  "changeOrderInfo": {
    "Status": 8,
    "Desc": "金丝雀发布",
    "PipelineInfoList": {
      "PipelineInfo": [
        {
          .....
          "PipelineId": "09e7d9aa-e79d-49ef-99b9-5405532d****"
          .....
          "PipelineId": "1ea53844-2c60-46f1-b46d-df3b34d0****"
          .....
          "PipelineId": "d99da075-b87d-4fe7-bbb6-ae1c6c26****"
          .....
        }
      ]
    }
  }
} //此处未罗列全部执行结果，仅供参考。
```

**说明** 示例代码的场景是在灰度分组发布后，剩余分组批次的策略是剩余分2批发布+手动分批。当灰度分组发布完成以后，需要手动确认继续剩余分组批次的发布。在调用 `GetChangeOrderInfo` 接口返回结果中出现3个流程ID (PipelineId)，第二次出现的流程ID (PipelineId) 代表需要手动确认继续剩余批次的发布，第三次出现的流程ID (PipelineId) 代表剩余分组批次中第一批完成后需要手动确认继续下一批次的发布。

3. 调用 `ContinuePipeline` 接口，手动确认继续剩余分组批次发布。

返回结果如下：

```
{
  "Message": "success",
  "RequestId": "B8ECF6F7-672D-40E9-91DD-1C33F06D4FD8",
  "Code": 200
}
```

4. 调用 `ContinuePipeline` 接口，手动确认下一批发布。

返回结果如下：

```
{
  "Message": "success",
  "RequestId": "7BB4F043-7C28-4A0E-B6A5-D4023EB24388",
  "Code": 200
}
```

**说明** 示例代码的场景是在灰度分组发布后，剩余分组批次的策略是剩余分2批发布+手动分批。运行到剩余分组批次过程，只需要手动确认一批。如果您是剩余批次需要手动确认多批的场景，请选择正确批次的流程ID (PipelineId) 并重复执行此步骤。

## 结果验证

您在金丝雀发布应用后，可以调用GetChangeOrderInfo接口查看变更流程详情，获取金丝雀发布应用的变更状态。具体API参数详情，请参见[GetChangeOrderInfo EDAS开发指南](#)中的GetChangeOrderInfo。

调用GetChangeOrderInfo接口查看变更流程详情，返回的执行结果如下：

```
{
  "Message": "success",
  "RequestId": "0EB98A7D-50CE-4F5F-855D-70BAF61DFB38",
  "Code": 200,
  "changeOrderInfo": {
    "Status": 2,
    "Desc": "金丝雀发布",
    "PipelineInfoList": {
      "PipelineInfo": [
        {
          .....
          //此处未罗列全部执行结果，仅供参考。
        }
      ]
    }
  }
}
```

请查看上述执行结果中的changeOrderInfo.Status参数值，通过该值判断金丝雀发布应用的变更是否成功。changeOrderInfo.Status的取值如下：

- 0：准备
- 1：执行中
- 2：执行成功
- 3：执行失败
- 6：终止
- 8：手动分批发布模式下，等待手工确认执行下一批。
- 9：自动分批发布模式下，等待下一批执行中。
- 10：系统异常执行失败

## 4.5. 回滚应用（ECS集群）

### 4.5.1. 使用控制台回滚应用（ECS集群）

当应用升级过程中或升级完成后，发现新版本应用有问题或异常，可以回滚应用，待问题解决后再升级。

#### 背景信息

- 升级中回滚：单批发布、分批发布或金丝雀发布过程中，发现异常，可以终止并回滚应用。
- 升级后回滚：单批发布、分批发布或金丝雀发布已经完成，发现新版本应用有问题，可以通过部署历史版本回滚应用。

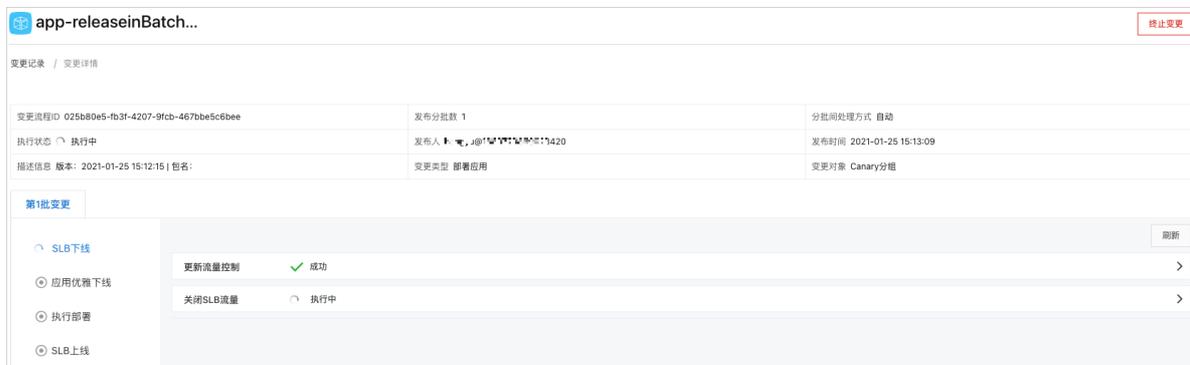
#### 升级中回滚

在升级应用过程中，发现异常，可以终止并回滚应用。升级应用包含单批发布、分批发布和金丝雀发布三种方式，下面分别介绍在各种方式升级过程中如何回滚应用。

● 单批发布中回滚

单批发布的具体操作，请参见[使用控制台升级应用 \(ECS集群\)](#)。

在单批发布启动后，会跳转到变更详情页面。在变更详情页面右上角单击终止变更。



● 分批发布中回滚

分批发布的具体操作，请参见[使用控制台分批发布应用](#)。

在分批发布启动后，会跳转到变更详情页面。根据分批方式设置不同，有两种回滚方式：

○ 自动分批发布中回滚

无需手动触发下一批次发布，在设置的分批等待时间后，自动发布下一批次，您可以在等待时间内完成新版本的验证。如果发现问题或异常，想要终止升级并回滚，在变更详情页面右上角单击终止变更。



○ 手动分批发布中回滚

需要手动开始下一批次发布。开始下一批次发布前，可以对新版本应用进行充分验证。如果发现问题或异常，在变更详情页面右上角单击终止变更。



● 金丝雀发布中回滚

金丝雀发布的具体操作，请参见[使用控制台金丝雀发布 \(ECS集群\)](#)。

在金丝雀发布启动后，会跳转到变更详情页面。金丝雀发布中，必须手动触发后续分批发布，所以在对灰度版本进行充分验证后，如果发现问题或异常，在变更详情页面右上角单击终止变更。



## 升级后回滚

应用升级已经完成，发现新版本应用有问题，可以通过部署历史版本回滚应用。

1. 登录EDAS控制台。
2. 在左侧导航栏单击应用列表。
3. 在应用列表页面的顶部菜单栏选择地域，并在页面上方选择微服务空间，在集群类型列表中选择ECS集群，然后在应用列表中单击目标应用名称。
4. 在应用信息页面右上角单击回滚应用。
5. 在回滚应用对话框中选择需要部署的历史部署包版本，并设置部署参数，然后单击回滚应用。



参数	说明
分组	<p>回滚应用的目标分组。</p> <ul style="list-style-type: none"> <li>如果当前应用仅有一个分组，在列表中选择<b>所有分组</b>，分组后的括号中显示当前应用的最大实例数。列表中还包含<b>默认分组</b>，和<b>所有分组</b>实际上是一样的。</li> <li>如果当前应用包含多个分组，在列表中选择要回滚应用的目标分组。</li> </ul> <p><b>说明</b> 如果选择<b>所有分组</b>，则会对应用所有分组按设置的<b>每组批次</b>进行回滚。</p>

参数	说明
每组批次	指定分组中分批发布批次数，单批发布的最小实例数为1。每组批次默认为2，请根据实际需求选择。
分批方式	每批发布间的处理方式，包含 <i>自动</i> 和 <i>手动</i> 。 <ul style="list-style-type: none"> <li><i>自动</i>: 按照分批等待时间自动发布剩余批次的发布。</li> <li><i>手动</i>: 需要手动触发下一个批次的发布。</li> </ul>
分批等待时间	每批次间的发布时间间隔，单位为分钟。根据实际需求，在列表中选择。

## 结果验证

- 查看变更记录。
  - 在应用详情页左侧导航栏单击**变更记录**。
  - 在**变更记录**页面的**变更状态**列中查看此次回滚应用的结果是否为**执行成功**。
    - 如果为**执行成功**，说明回滚应用成功。
    - 如果为**执行失败**，说明回滚应用失败。在操作列单击**查看**，通过变更详情排查失败原因。
- 查看应用详情。
  - 在应用详情页左侧导航栏单击**基本信息**。
  - 在**基本信息**页面单击**基本信息**页签，在**基本信息**页签的**应用信息**区域中查看应用的**部署包**是否为历史的部署包。
  - 在**基本信息**页面单击**实例部署信息**页签，在**实例部署信息**页签查看目标分组中各实例的运行状态是否为**运行正常**。

## 4.5.2. 使用API在ECS集群中回滚应用到目标历史版本

本文介绍如何使用EDAS提供的Java SDK调用API在ECS集群中回滚应用到目标历史版本。

### 前提条件

回滚应用到目标历史版本前，您必须提前完成以下准备工作：

- 安装EDAS的Java SDK。更多信息，请参见[Java SDK接入指南](#)。
- 确定创建ECS集群的地域。
- 您的应用已经升级版本。目前支持多种升级应用的方式，请根据实际需求选择：
  - [使用API在ECS集群中单批发布应用](#)
  - [使用API在ECS集群中分批发布应用](#)
  - [使用API金丝雀发布（ECS集群）](#)
- 调用 `ListHistoryDeployVersion` 接口查询应用历史版本列表，获取目标历史版本的PackageVersion，假设为 `20210417.154931`。
- 请调用 `ListDeployGroup` 接口查询应用实例分组列表，获取应用实例分组的GroupId，假设为 `941be68c-4aac-48a1-88fe-c9ad1502****`。

### 回滚应用到目标历史版本

以下代码适用于在EDAS ECS集群中回滚应用到目标历史版本。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.RollbackApplicationRequest;
import com.aliyuncs.edas.model.v20170801.RollbackApplicationResponse;
public class RollbackApplication {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //应用所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求, 并设置参数。
        RollbackApplicationRequest request = new RollbackApplicationRequest();
        //需要回滚历史版本的目标应用的AppId。
        request.setAppId("6bbc57a2-a017-4bec-b521-49a15bd3****");
        //应用要回滚到的历史版本。可以调用ListHistoryDeployVersion接口获取。
        request.setHistoryVersion("20210417.154931");
        //部署分组ID。如果要部署到所有分组, 参数设置为all。
        //如果需要部署到具体应用实例分组, 可以调用ListDeployGroup接口获取GroupId。
        request.setGroupId("all");
        //每组发布批次, 可设置范围为1~5, 1代表单批发布, 2~4代表分批发布。
        request.setBatch(1);
        //分批等待时间, 单位分钟。默认为0, 表示不等待; 最大值为5。如果是单批发布, 无需设置此参数。
        request.setBatchWaitTime(0);
        try {
            RollbackApplicationResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage() + "\nChangeOrderId=" + re
sponse.getChangeOrderId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序, 返回的执行结果如下所示:

```
Message=success
ChangeOrderId=b3496bf0-0b6b-42c3-921c-f5250af****
```

## 结果验证

您在回滚应用到目标历史版本后, 可以调用GetChangeOrderInfo接口查看变更流程详情, 获取回滚应用的变更状态。具体API参数详情, 请参见[GetChangeOrderInfo EDAS开发指南](#)中的GetChangeOrderInfo。

调用GetChangeOrderInfo接口查看变更流程详情, 返回的执行结果如下:

```
{
  "Message": "success",
  "RequestId": "EF09BCD3-3D37-4D76-9CF3-12B71A9EDE7C",
  "Code": 200,
  "changeOrderInfo": {
    "Status": 2,
    "Desc": "Version: 20210417.154931 | Package Name: sc-consumer-D-0.0.1-SNAPSHOT.jar?Expires=161864****&OSSAccessKeyId=TMP.3KgjU****fSsJ&Signature=LrUEz***X9c...",
    "PipelineInfoList": {
      "PipelineInfo": [
        {
          "PipelineStatus": 2,
          "PipelineName": "Batch 1 Change",
          .....
          //此处未罗列全部执行结果, 仅供参考。
        }
      ]
    }
  }
}
```

请查看上述执行结果中的changeOrderInfo.Status参数值, 通过该值判断回滚应用到历史版本的变更是否成功。changeOrderInfo.Status的取值如下:

- 0: 准备
- 1: 执行中
- 2: 执行成功
- 3: 执行失败
- 6: 终止
- 8: 手动分批发布模式下, 等待手工确认执行下一批。
- 9: 自动分批发布模式下, 等待下一批执行中。
- 10: 系统异常执行失败

### 4.5.3. 使用API在ECS集群中终止应用升级变更流程

本文介绍如何使用EDAS提供的Java SDK调用API在ECS集群中终止应用升级变更流程。

#### 前提条件

在终止应用升级变更流程前, 您必须提前完成以下准备工作:

- 安装EDAS的Java SDK。更多信息, 请参见[Java SDK接入指南](#)。
- 确定应用部署地域, 假设为cn-hangzhou。
- 确保目标应用处于升级变更流程中, 请根据实际需求选择应用升级方式:
  - [使用API在ECS集群中单批发布应用](#)
  - [使用API在ECS集群中分批发布应用](#)
  - [使用API金丝雀发布 \(ECS集群\)](#)
- (可选) 调用ListRecentChangeOrder接口, 查询应用变更记录, 获取应用正在升级变更的变更单流程ID (ChangeOrderId)。假设为47e2b863-adc8-4a0e-ac23-8361b95a\*\*\*\*。如果您在升级应用时已经记录变更单流程ID (ChangeOrderId), 则无需调用该接口。

#### 终止应用变更

以下代码适用于在EDAS ECS环境终止应用变更。

**说明** 以下示例代码只能终止应用变更，无法自动回滚到应用变更前的版本。您需要回滚应用到变更前版本，解决新版本应用问题后，再重新部署应用。回滚应用到变更前版本的操作，请参见[使用API在ECS集群中回滚应用到目标历史版本](#)。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.edas.model.v20170801.AbortChangeOrderRequest;
import com.aliyuncs.edas.model.v20170801.AbortChangeOrderResponse;
public class AbortChangeOrder {
    public static void main(String[] args) {
        //阿里云账号或RAM用户的AccessKey ID。
        String aliyun_user_ak = "<yourAccessKeyId>";
        //阿里云账号或RAM用户的AccessKey Secret。
        String aliyun_user_sk = "<yourAccessKeySecret>";
        //应用所在地域ID。
        String region_id = "cn-hangzhou";
        DefaultProfile defaultProfile = DefaultProfile.getProfile(region_id, aliyun_user_ak
, aliyun_user_sk);
        DefaultAcsClient client = new DefaultAcsClient(defaultProfile);
        //创建API请求，并设置参数。
        AbortChangeOrderRequest request = new AbortChangeOrderRequest();
        //应用升级变更流程ID。
        request.setChangeOrderId("47e2b863-adc8-4a0e-ac23-8361b95a****");
        try {
            AbortChangeOrderResponse response = client.getAcsResponse(request);
            System.out.println("Message=" + response.getMessage() + "\nRequestId=" + respon
se.getRequestId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

运行程序，返回的执行结果如下所示：

```
Message=success
RequestId=19C54DC3-C66C-40D0-AEED-8D5A1F00F3BA
```

## 结果验证

您在执行终止应用变更流程后，可以调用GetChangeOrderInfo接口查看变更流程详情，获取应用升级的变更状态。具体API参数详情，请参见[GetChangeOrderInfo](#)。

调用GetChangeOrderInfo接口查看变更流程详情，返回的执行结果如下：

```
{
  "Message": "success",
  "RequestId": "CCC48EB3-E628-45B4-8DF3-ED8551119742",
  "Code": 200,
  "changeOrderInfo": {
    "Status": 6,
    "Desc": "create by edas pop api",
    "PipelineInfoList": {
      "PipelineInfo": [
        {
          .....
          //此处未罗列全部执行结果，仅供参考。
        }
      ]
    }
  }
}
```

请查看上述执行结果中的changeOrderInfo.Status参数值，通过该值判断终止应用变更流程是否成功。changeOrderInfo.Status的取值如下：

- 0: 准备
- 1: 执行中
- 2: 执行成功
- 3: 执行失败
- 6: 终止
- 8: 手动分批发布模式下，等待手工确认执行下一批。
- 9: 自动分批发布模式下，等待下一批执行中。
- 10: 系统异常执行失败

## 5. CI/CD

### 5.1. 使用云效2020部署Java应用至ECS集群

使用云效2020部署应用至EDAS ECS集群

本文以云效的示例代码源为例，介绍如何使用云效以JAR或WAR包的方式部署应用至EDAS的ECS集群。

#### 前提条件

- [开通云效2020](#)。
- 将业务代码上传至代码库。关于云效支持的代码库，请参见[代码源](#)。

#### 部署流程



1. 如果您的应用已经部署在EDAS上，则无需创建新的应用。如果您是第一次部署应用到EDAS，则需要在EDAS控制台创建应用，以便云效将业务代码推送到该应用中。
2. 在EDAS创建应用完成后，在云效上创建企业并进入流水线FLOW。
3. 根据EDAS应用类型配置应用集成发布的流水线信息。
4. 流水线配置完成后，运行流水线完成EDAS应用部署。

#### 步骤一：在EDAS控制台创建应用

如果您未使用过EDAS创建应用，则需要预先在EDAS上创建应用，本文以在ECS集群中创建一个使用JAR包部署的Spring Cloud应用为例。相关操作，请参见[在ECS集群中创建并部署应用](#)。

 **注意** 在EDAS控制台创建应用时使用的部署方式（JAR或WAR），必须与在云效的流水线设置保持一致。例如：在EDAS上使用JAR方式部署应用，那么在云效设置流水线时，构建环节必须是构建和上传JAR包相关的设置。

#### 步骤二：在云效创建企业

如果您是第一次使用[云效Flow](#)，则需要在云效上创建您的企业。

1. 登录[云效Flow](#)。
2. 设置企业、组织或团队名称并选择[使用规模](#)，单击完成创建。



### 步骤三：在云效创建流水线

1. 登录云效Flow。
2. 在左侧导航栏，单击我的流水线，然后单击右上角的新建流水线。
3. 在选择流水线模板对话框，选择其他·空模板，单击创建。



创建完成后，会跳转到添加流水线源的页面。

### 步骤四：配置流水线

1. 在创建的流水线的阶段1区域单击空任务，在编辑面板，配置任务名称、构建集群、下载流水线源并添加任务步骤。



添加任务说明如下：

- i. 在任务步骤区域下，单击添加步骤，在下拉列表中选择构建 > Java构建，并配置该步骤参数。

**Java 构建**

步骤名称

JDK 版本

Maven 版本 ?

构建命令 ? ↕ ↗

**i** 上传您的依赖包文件至packages，通过mvn clean install 完成依赖包安装，[查看帮助](#)

```

1 # maven build default command
2 mvn -B clean package -Dmaven.test.skip=true
  -Dautoconfig.skip
3
4 # gradle build default command
5 # ./gradlew build
6
7 # ant build default command
            
```

参数	描述
步骤名称	输入您自定义的步骤名称。
JDK版本	在下拉列表中选择需要的Java版本。
Maven版本	在下拉列表中选择需要的Maven版本。
构建命令	自定义的构建命令。

ii. 单击添加步骤，在下拉列表中选择上传 > 构建物上传 (EDAS/SAE使用)，并配置该步骤参数。

▼ 构建物上传 (EDAS/SAE使用) -

步骤名称

制品名称 ?

上传文件 ?

参数	描述
步骤名称	输入您自定义的步骤名称，不修改则显示为默认名称。
制品名称	自定义产出物名称，例如 <i>default</i> ，用于区分多次构建产生的不同产出物，定义后在部署任务中使用。
上传文件	设置项目文件所在的路径。

2. 在新阶段区域单击新的任务，在选择任务组面板的左侧导航栏单击部署，然后选择EDAS ECS 应用发布。

**选择任务组** ×

代码扫描

测试

构建

测试构建

镜像构建

部署

工具

代码

执行命令

空模板

请输入任务名称



**ACK Nginx Ingr...**  
基于ACK Nginx Ingr...



**Kubernetes 发布**  
用于发布 Kubernetes...



**Kubernetes 镜...**  
升级 Kubernetes 下...



**Kubernetes 蓝...**  
基于Istio进行应用...



**Kubernetes 分...**  
Kubernetes 分批发布



**Helm Release ...**  
使用代码库中的 He...



**EDAS ECS应用...**  
EDAS ECS应用发布



**EDAS Kubernet...**  
EDAS Kubernetes ...



**ASM 蓝绿发布**  
基于阿里云Service ...



**Serverless(SAE)...**  
Serverless(SAE) 应...



**资源编排服务(R...**  
资源编排服务(ROS)...



**函数计算 (FC) ...**  
更新函数计算 (FC...

> 文档版本：20220708

109

3. 在编辑面板，配置任务信息。

编辑  ×

---

任务名称 

EDAS ECS 应用发布

构建集群 

云效北京构建集群 ▼

下载流水线源 

下载全部流水线源 ▼

任务步骤

选择服务连接  添加服务连接

[blurred] ▼

地域

[blurred] ▼

命名空间

默认命名空间 ▼

应用  

[blurred] ▼

软件包

制品名称.default ▼

发布批次

[blurred] ▼

分批等待时间

1分钟

分批方式

手动确认

---

任务插件

+ [添加插件](#)

配置的任务信息说明如下表所示。

参数名	说明
任务名称	自定义的任务名称。
构建集群	可为任务选择不同的 <b>构建集群</b> 。
下载流水线源	可为任务选择是否下载流水线源。开启下载流水线源后，您配置流水线源的源文件将会被下载至对应的工作目录下。
选择服务连接	<p>选择任务连接的服务。如果没有服务连接，单击<b>添加服务连接</b>，根据界面提示，添加您需要的连接。选择任务的服务授权，使云效能在EDAS上部署应用。</p> <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 5px;"> <p><span style="color: #007bff;">?</span> <b>说明</b> 如果您从未授权过，新建服务连接时，请先在<b>服务授权/证书</b>输入框后，单击<b>新建</b>，根据跳转完成阿里云RAM授权后再进行相应配置。</p> </div>
地域	请参见 <b>步骤一：在EDAS控制台创建应用</b> ，选择该步骤所创建应用所在的地域。
命名空间	选择所创建应用所在的命名空间。
应用	请参见 <b>步骤一：在EDAS控制台创建应用</b> ，选择该步骤所创建的应用。
软件包	选择 <b>步骤四：配置流水线</b> 中 <b>步骤1</b> 中构建物上传（EDAS/SAE使用）步骤所创建的产出物名称。
发布批次	选择批次数量来表示发布分批的数量。
分批等待时间	相邻发布批次之间的等待时间。

参数名	说明
分批方式	可选择 <b>手动确认</b> 或 <b>自动确认</b> 。例如，如果您需在完成第一批发布时先观察发布结果再决定后续操作，则可选择 <b>手动确认</b> 。
(可选) 任务插件	<p>您可以根据需要配置任务插件来发送流水线通知。</p> <ul style="list-style-type: none"> <li>钉钉机器人通知插件 <ul style="list-style-type: none"> <li><b>webhook地址</b>：配置钉钉机器人的webhook地址，钉钉机器人的创建步骤请参见<a href="#">钉钉机器人配置</a>。</li> <li><b>运行时机</b>：您可以选择任务插件发送通知的时机。</li> </ul> </li> <li>邮件通知 <ul style="list-style-type: none"> <li><b>邮件地址</b>：配置需要发送通知的邮件地址。</li> <li><b>运行时机</b>：您可以选择任务插件发送通知的时机。</li> </ul> </li> <li>Webhook通知插件 <ul style="list-style-type: none"> <li><b>webhook地址</b>：配置需要发送通知的webhook地址，地址必须公网可访问。</li> <li><b>运行时机</b>：您可以选择任务插件发送通知的时机。</li> </ul> </li> </ul>

4. 配置完成后在页面右上角单击**仅保存**或**保存并运行**。

## 步骤五：在云效部署应用至EDAS

1. 配置完流水线后，在流水线页面右上角单击**保存并运行**，然后在**运行配置**对话框中选择代码分支后单击**运行**。

本操作示例中创建的应用拥有2个实例，并在流水线中配置了分批发布和手动确认。故在**构建**任务运行成功后，在新阶段的发布任务中您可执行以下操作：



- 单击**发布单详情**：可跳转到EDAS控制台查看变更详情。
- 单击**继续发布**：可继续下一批的发布。

- 选择更多 > 终止变更：可终止本次流水线的运行。
  - 选择更多 > 日志：可查看流水线的运行日志。
  - 选择更多 > 取消：可取消本次任务运行。
2. 单击继续发布，等待流水线执行完毕。



### 结果验证

云效显示部署成功后，登录EDAS控制台查看应用的变更记录，查看是否有与云效相关的部署应用的变更记录，如果有并显示执行成功则代表部署成功。

### 常见问题

- 如何使用云效以镜像方式部署应用至EDAS？  
您可使用云效以镜像方式部署应用至EDAS的K8s集群，详情请参见[使用云效2020进行持续集成和部署 \(K8s\)](#)。
- 除Java应用外，其他编程语言能否通过云效部署至EDAS？可以使用哪些部署方式部署？  
除Java外的其他编程语言可以使用镜像部署方式，通过云效部署至EDAS。使用镜像部署时，请确保您应用代码中有包含应用的Dockerfile文件，操作部署请参见[使用云效2020进行持续集成和部署 \(K8s\)](#)。

### EDAS容器服务K8s和Serverless K8s交流群

如果您在EDAS中使用容器服务K8s集群和Serverless K8s集群过程中有任何疑问或建议，请提交工单，或使用钉钉搜索钉钉群号 23197114 加入钉钉群进行反馈。

## 5.2. 使用Jenkins创建持续集成

使用Jenkins可以构建EDAS应用的持续集成方案。该方案涉及下面的计算机语言或开发工具，阅读本文需要对下述的语言或工具有一定的理解。

### 前提条件

在开始持续集成之前，需要完成下述的准备工作。

1. 获取阿里云的AccessKey ID和AccessKey Secret。
  - i. 使用已经开通了EDAS服务的主账号登录阿里云官网。

ii. 进入Access Key控制台，创建AccessKey ID和AccessKey Secret。



2. 在使用Jenkins自动部署应用之前，需要先在EDAS控制台中创建并部署应用。

- i. 登录EDAS控制台。
- ii. 参考应用创建和部署概述 (ECS) 和创建和部署应用概述 (K8s)，部署应用。如果已经创建了应用，请忽略此步。
- iii. 在左侧导航栏中单击应用管理。找到您在上一步中创建的应用并单击进入详情页面，获取应用ID的字段内容。



- 3. 使用Git Lab托管您的代码。您可以自行搭建Git lab或者使用阿里云Code。本文使用通过自行搭建的Git Lab做演示，关于Git lab的更多信息，请参见Git Lab。
- 4. 了解并使用Jenkins。关于Jenkins的更多详细信息，请参见Jenkins官网。

### 背景信息

**说明** ECS集群和容器服务Kubernetes集群都支持使用Jenkins创建持续集成。

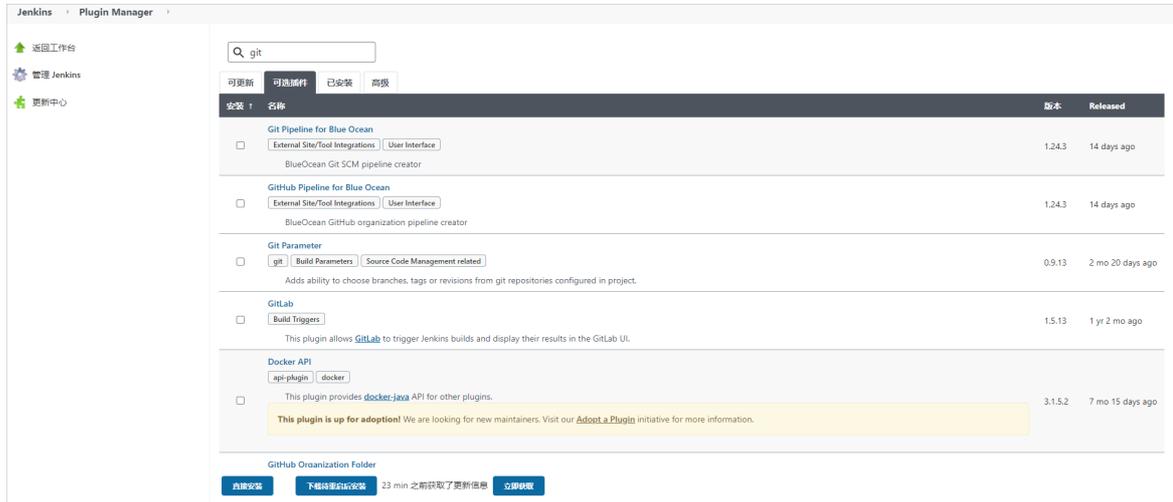
工具	说明
Maven	Maven是一个项目管理和构建的自动化工具。
Jenkins	Jenkins 是一个可扩展的持续集成引擎。
GitLab	GitLab 是一个利用Ruby on Rails开发的开源应用程序，实现一个自托管的Git项目仓库，可通过Web界面进行访问公开的或者私人项目。它拥有与GitHub类似的功能，能够浏览源代码，管理缺陷和注释。

### 配置项目

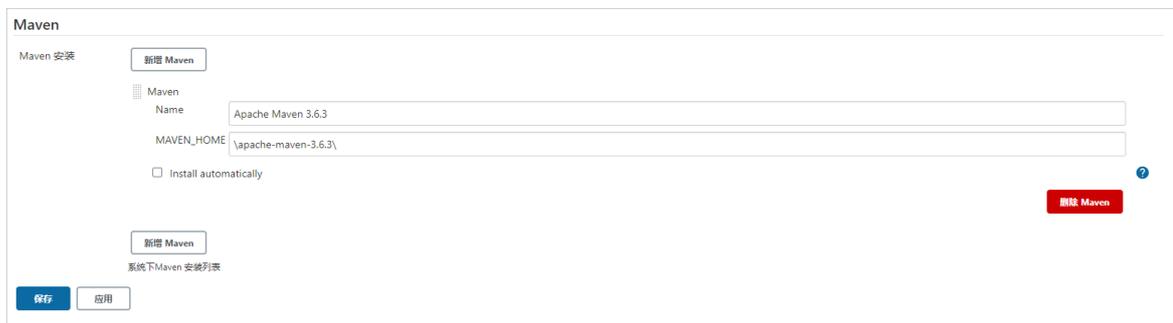
修改项目配置，添加toolkit-maven-plugin及部署信息，具体操作，请参见使用toolkit-maven-plugin插件升级应用 (ECS集群)。您在修改完项目配置后，建议在本地使用Maven构建验证配置是否正确。

## 安装和配置Jenkins

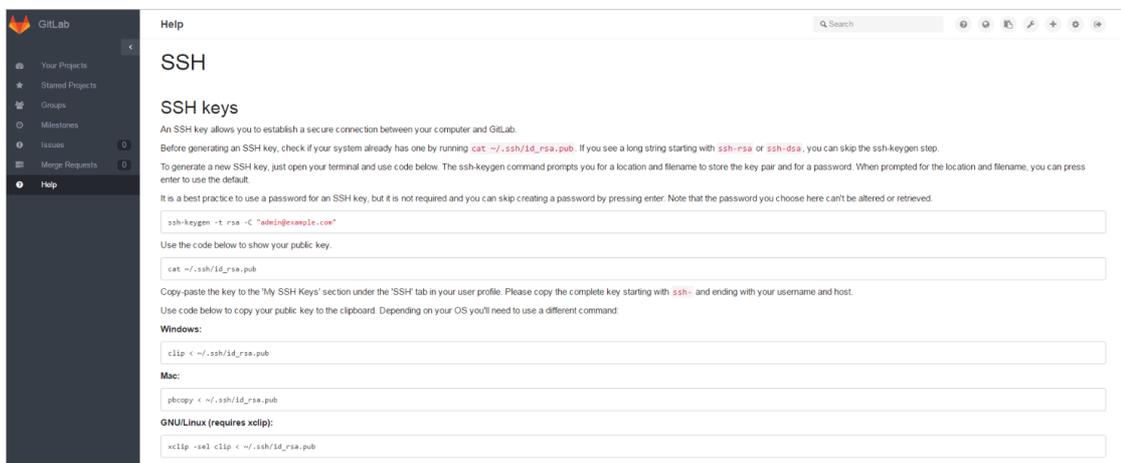
1. 进入[Jenkins官网](#)下载安装Jenkins。
2. 在Jenkins控制台的菜单栏中选择系统管理 > 插件管理，安装Git和Git Lab插件。
  - o 安装GIT Client Plugin和GIT Plugin插件可以帮助Jenkins拉取Git仓库中的代码。
  - o 安装Git lab Hook Plugin插件可以帮助Jenkins在收到Git lab发来的Hook后触发一次构建。



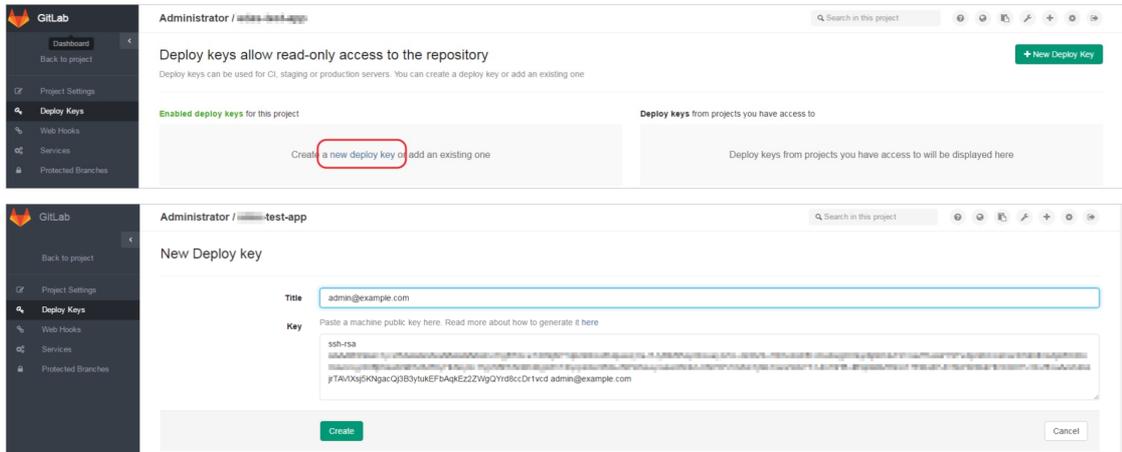
3. 安装Maven构建工具，请参见[Maven官网](#)。  
在Jenkins控制台的菜单栏中选择系统管理 > 全局工具配置，选择Maven版本名称并配置路径。



4. 在Jenkins服务器上生成SSH RSA密钥对，并将公匙导入Git Lab，实现Jenkins拉取Git Lab代码时自动认证。
  - i. 参考[Git Lab文档](#)，在Jenkins服务器运行Jenkins软件的用户下，生成SSH RSA密钥对。

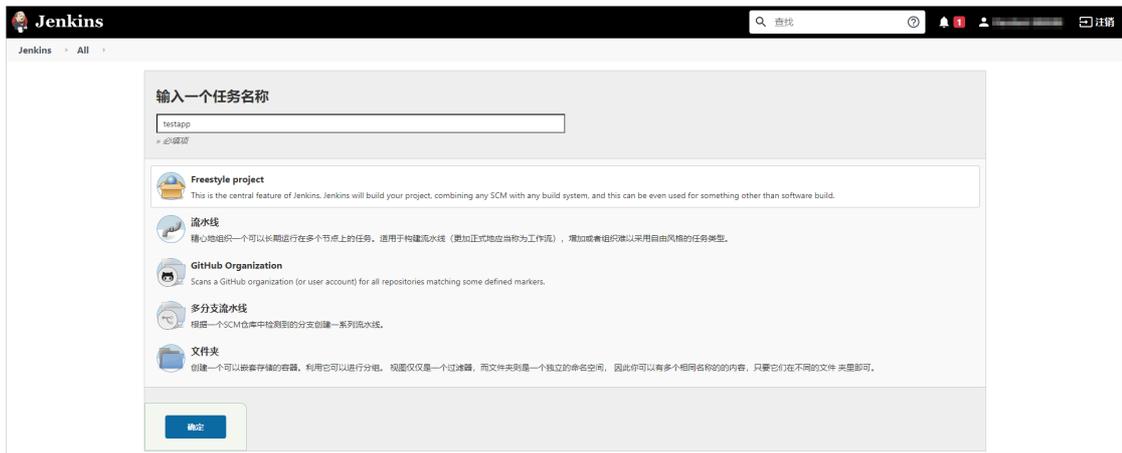


- ii. 进入Git Lab首页，在菜单栏选择Settings > Deploy Keys，并单击new deploy key添加key，导入在Jenkins服务器上创建的SSH RSA公匙。

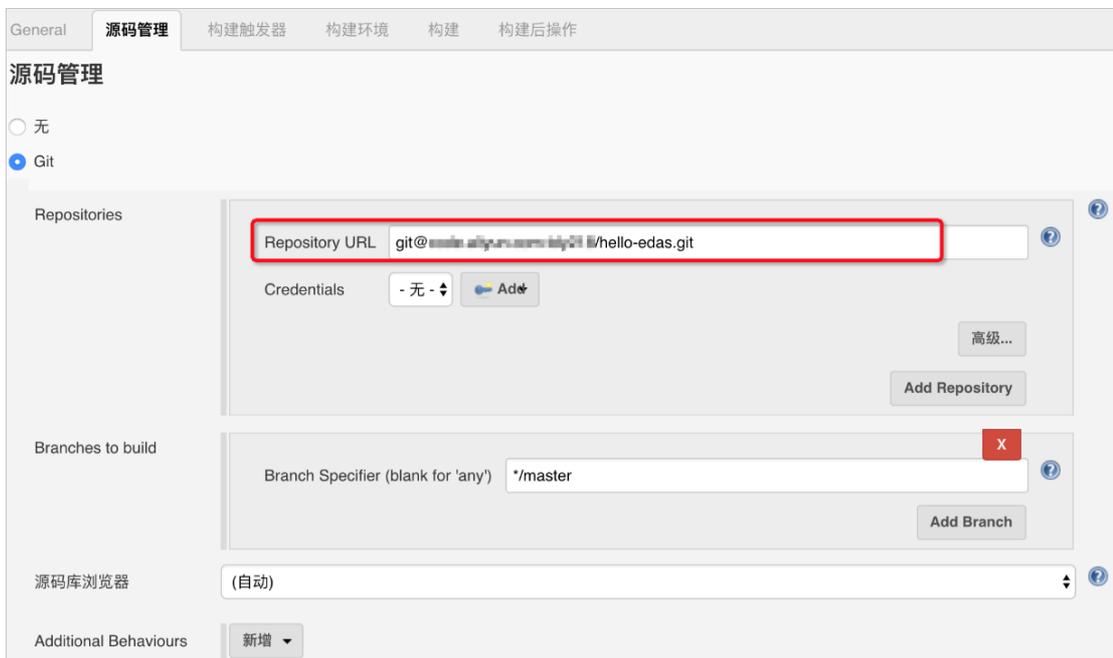


5. 创建Jenkins任务。

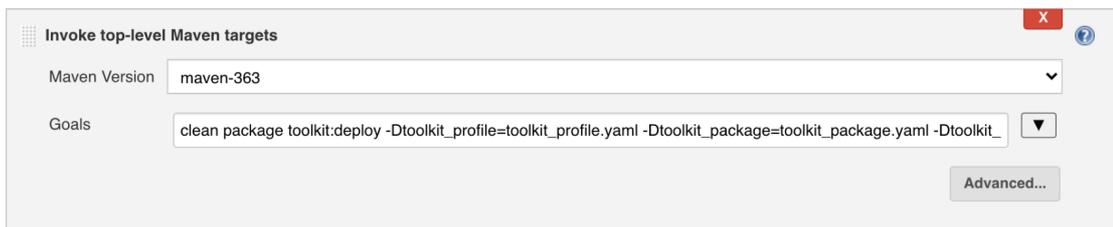
- i. 在Jenkins首页左侧导航栏中单击新建，创建Jenkins任务，并选择构建一个自由风格的软件项目。



- ii. 在源码管理页面中选择Git，并设置相关参数。
  - **Repository URL**：您的项目的Git协议地址。
  - **Credentials**：安全凭证，选择无即可（前提是运行Jenkins软件的用户SSH RSA公匙已添加到该Git项目所在的Git Lab中，否则这里会报错）。



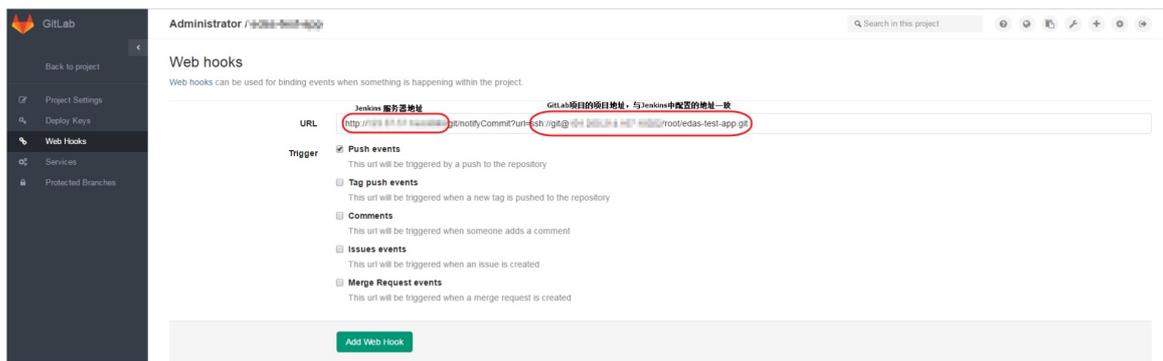
- iii. 单击构建触发器页签，勾选轮询SCM。
- iv. 单击构建环境页签，勾选Add timestamps to the Console Output（为控制台输出的信息添加时间戳）。
- v. 单击构建页签，然后单击增加构建步骤。
- vi. 在调用顶层Maven目标区域设置Maven版本和目标。如果您想部署多模块工程，请参见[创建多模块工程的Jenkins任务](#)。
  - **Maven Version**：单击该选项后面的下拉框，选择在全局工具配置里配置的Maven版本名称。
  - **Goals**：填入clean package toolkit:deploy（如有其它参数，请根据实际情况填入）



### 配置Gitlab的Web Hook，实现自动构建

1. 右键单击Git Lab工程，然后选择Setting > Web Hooks。
2. 在Web Hooks页面的在URL文本框中输入 `http://jenkins服务器地址:jenkins服务器监听端口/git/notifyCommit?url=本项目的git协议地址`。

例如：`http://123.57.xx.xxx:8080/git/notifyCommit?url=git@code.aliyun.com:tdy218/hello-edas.git`



图中表示的Jenkins服务器地址为您的Jenkins服务器的Web访问地址如 `http://123.57.xx.xxx:8080`。

3. 配置完成后，单击Test Hook，进行测试。



### 配置正确后，提交变更到GitLab

如果上述步骤配置正确，这次提交会触发一次GitLab Hook。Jenkins在接受到这个Hook后会构建您的Maven项目，并在构建结束时调用EDAS POP API脚本触发部署。

提交部署成功输出的日志信息（Build Number > 控制台输出）。

```
15:58:51 [INFO] Deploy application successfully!
15:58:51 [INFO] -----
15:58:51 [INFO] BUILD SUCCESS
15:58:51 [INFO] -----
15:58:51 [INFO] Total time: 24.330 s
15:58:51 [INFO] Finished at: 2018-12-25T15:58:51+08:00
15:58:51 [INFO] Final Memory: 23M/443M
15:58:51 [INFO] -----
15:58:51 Finished: SUCCESS
```

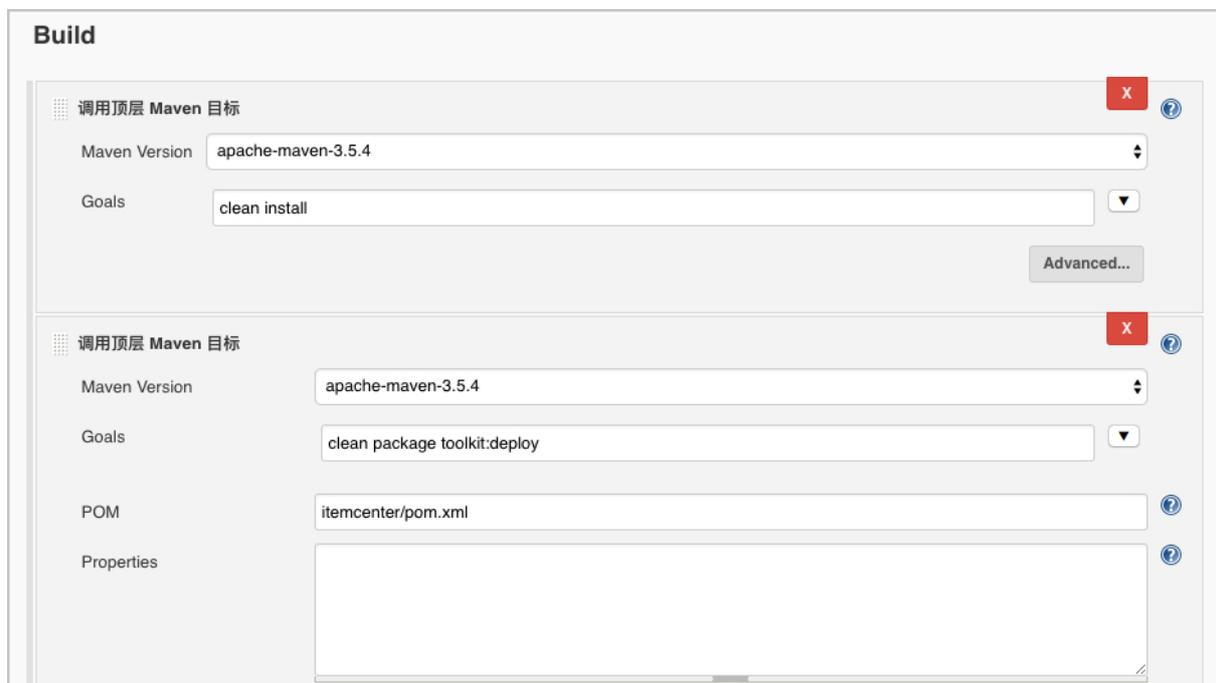
如果部署失败，可以登录EDAS控制台，在左侧导航栏中单击应用管理 > 应用列表，在应用列表页面单击具体应用名称，进入应用详情页面。在左侧导航栏单击变更记录来查看此次部署任务的执行过程。

### 创建多模块工程的Jenkins任务

创建多模块工程的Jenkins任务和安装和配置Jenkins第5步基本相同，只需要调整下调用顶层Maven目标。如果工程为多模块工程，想在Jenkins中部署子模块的话，那么需要在父模块中调用 `mvn clean install` 命令，然后在子模块中调用 `mvn clean package toolkit:deploy` 命令。以Demo工程为例，工程结构如下：

```
sh-3.2# tree -L 1 carshop
carshop
├── detail
├── itemcenter
├── itemcenter-api
└── pom.xml
```

其中，detail、itemcenter、itemcenter-api为子模块，现在想部署itemcenter模块的话，那么需要在父工程中设置一个clean install构建目标，然后在itemcenter模块中设置 `clean package toolkit:deploy` 构建目标。



## 6. 应用设置

### 6.1. 设置部署在ECS集群中的应用

在ECS集群中创建应用后，在应用基本信息页面的应用设置区域可以分别设置应用的JVM、Tomcat、基本信息（应用名称）和挂载脚本。

#### 访问应用设置页面

应用的JVM、Tomcat和挂载脚本均可以在应用和应用分组内进行设置，应用分组的设置优先于应用的设置，若应用分组没有相应的设置，则应用分组使用应用的设置。

1. 登录EDAS控制台。
2. 在左侧导航栏中单击应用列表，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在应用列表页面单击具体的应用名称。
3. 进入应用或应用分组设置页面。
  - 应用：  
在基本信息页面的应用设置区域，设置JVM参数、Tomcat、基本信息或挂载脚本。
  - 应用分组：
    - a. 选择实例部署信息页签，单击部署分组右侧的分组设置。
    - b. 在分组设置的下拉列表中选择JVM、Tomcat或挂载脚本。

#### 设置JVM参数

JVM参数用于在应用启动时配置容器的参数。正确配置这些参数有助于降低垃圾回收（GC）开销，从而缩短服务器响应时间并提高吞吐量。如果没有配置容器参数，默认由JVM分配。

1. 在基本信息页面的应用设置区域，选择JVM参数并单击后面的编辑。
2. 单击展开内存配置、应用、GC策略和工具，设置相关参数。

 说明 当上述参数类型和具体参数无法满足您的需求时，可以单击自定义，在自定义参数右侧的文本框中配置自定义参数。参数的详细说明，请参见[JVM -D启动配置参数](#)。

3. 单击配置JVM参数完成配置。
4. 手动重启应用使配置的JVM参数生效。

#### 设置Tomcat参数

您可以设置容器（Tomcat）的端口、应用访问路径以及最大线程数等参数信息。

1. 在设置的下拉列表中单击Tomcat。
2. 在应用设置区域，单击应用设置后的编辑，然后设置Tomcat参数。
  - 应用端口：端口范围为1024~65535。1024以下的端口需要Root权限才能操作，而容器配置的权限为Admin。所以请填写1024以上的端口。如果不配置，则默认为8080。
  - Tomcat Context：选择应用的访问路径。
    - 选择程序包名字，则自定义路径不需填写，应用的访问路径是WAR包名称。
    - 选择根目录，自定义路径也不需填写，应用的访问路径是/。
    - 选择自定义，则需要下面的自定义路径中填写自定义的路径。如果不配置此项，则应用的默认访问路径是WAR包名称。

- **最大线程数**：配置连接池的连接数大小，对应参数maxThreads，默认是400。此项配置对应用性能有很大影响，建议由专业人士配置。
- **Tomcat 编码**：Tomcat的编码格式，包括UTF-8、ISO-8859-1、GBK和GB2312。默认为ISO-8859-1。

3. 单击**配置Tomcat**完成配置。

### 修改应用名称

在应用设置区域单击**基本信息**页签，设置应用名称及应用描述，并单击**修改**。

### 设置运行状态报警

对于应用运行环境是EDAS-Container的ECS集群的应用，您可在**应用设置**区域设置**健康检查失败报警**、**OOM Kill报警**和**异常退出分析**。

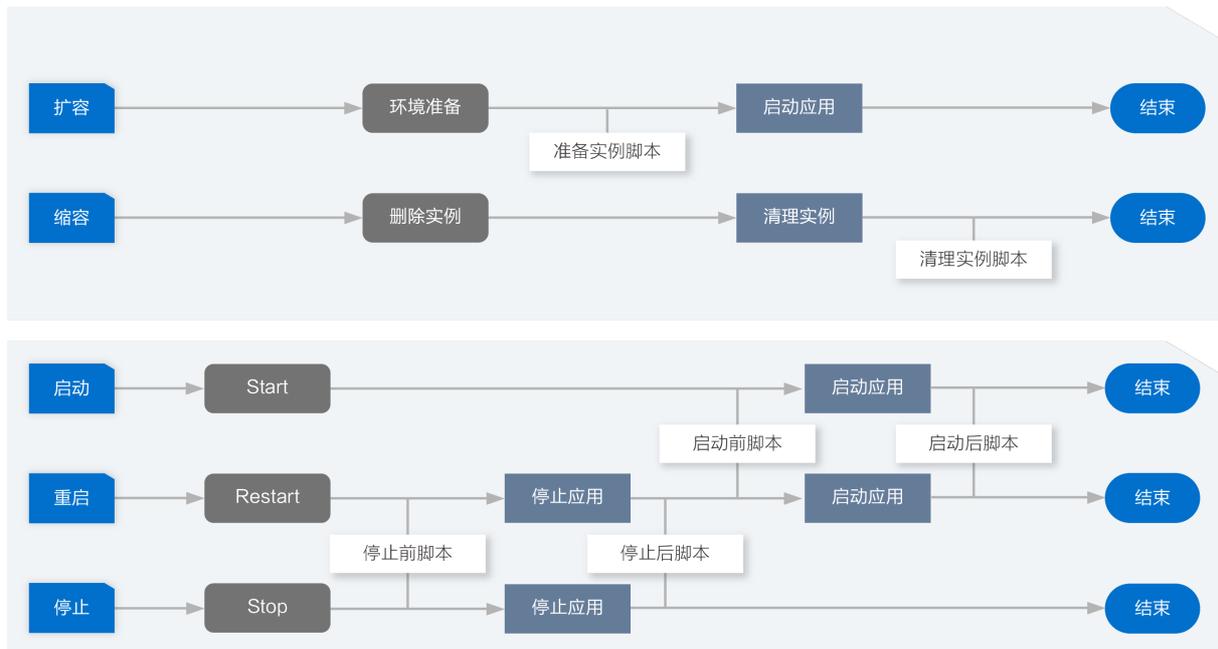
- **健康检查失败报警**  
设置了应用的“健康检查URL”并且选中**健康检查失败报警**后，EDAS会定时检查应用的该URL的HTTP状态码是否正常（即为2XX或3XX）。如果状态码异常则发出报警。
- **OOM Kill报警**  
选中**OOM Kill报警**后，EDAS会监控应用宿主机（或容器）的内存使用率。当发现内存使用率在一段时间内持续高于80% 并且应用存在被系统OOM Kill的风险时，EDAS会发出预警信息。当系统将应用OOM Kill之后，EDAS会发出报警信息。
- **异常退出分析**  
选中**异常退出分析**后，EDAS会监控应用的存活情况。当应用因系统或因人为因素异常退出时，EDAS会向用户发出报警信息。

## 6.2. 设置应用生命周期挂载脚本

在应用生命周期中，会发生各种事件，如启动、停止等。EDAS提供了当发生事件时，在ECS实例上执行用户自定义脚本的能力。

### 挂载脚本定义说明

生命周期挂载脚本的运行过程图示意如下：



脚本执行须知：

- 脚本采用 `root` 用户执行。
- 脚本执行时间最长为60s，当执行时间超过60s，将认为超时导致脚本执行失败。
- 脚本可能被重复执行，需要脚本自身保证重复执行的正确性，例如：脚本应避免将相同的配置项追加到同一配置文件。
- 脚本在独立的进程中执行，与应用进程无关。
- 如需设置环境变量在应用启动后使用，建议将环境变量写入到 `/home/admin/.bashrc`中。
- 脚本通过ECS的发送远程命令执行，命令名称为 `CoCustomHookScript`。
- 执行结果状态：
  - 成功：按照Linux、Unix的标准，执行退出码为 0 时为成功。如命令 `echo xyz` 退出码为 0。
  - 失败：执行退出码非 0 时为失败。如命令 `ls 不存在的目录` ，退出码非 0。
- 忽略失败：开启时，若 执行结果状态 为 失败 ，则不中断应用变更单的执行。
- 不忽略失败：当脚本执行失败并且健康检查URL通过时，挂载脚本将被重复执行。您可通过以下方法继续执行应用变更：
  - 修复相应的脚本问题，然后重试变更任务。
  - 若是启动后挂载脚本，可在变更任务单击略过，在执行挂载脚本这个变更单任务时，将采用忽略失败模式，变更记录将继续执行。
- 允许输入的脚本最大 `8KB` ，若超过，可将脚本放置于外部存储，如 OSS 存储服务中，通过网络下载后执行。

## 设置挂载脚本

您可参照以下步骤进入到挂载脚本的设置页面。

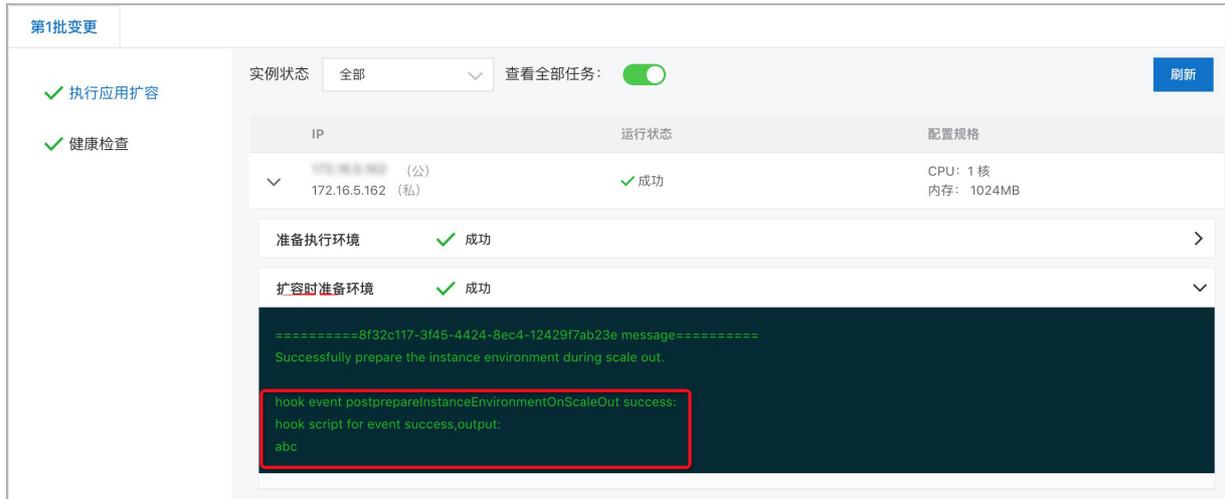
1. 登录EDAS控制台。
2. 在左侧导航栏中单击应用列表，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在应用列表页面单击具体的应用名称。
3. 进入应用或应用分组设置页面。
  - 应用：在基本信息页面的应用设置区域的左上角单击挂载脚本。
  - 应用分组：选择实例部署信息页签，在分组设置的下拉列表中选择挂载脚本。
4. 在挂载脚本对话框单击展开需设置的应用生命周期的脚本，在文本框中输入脚本内容，然后单击修改。



参数	描述
准备实例脚本	在ECS实例扩容到应用中时发生一次。通常可用于设置应用相关的环境变量、hosts等。相比于cloud-init在虚拟机初始化阶段触发，进一步提供了跟应用相关的触发点。
销毁实例脚本	ECS实例从应用移除时发生一次。通常可用于移除应用相关的配置等。
启动前脚本	通过变更触发的应用启动，在应用容器每次启动前，如调用Tomcat的启动脚本start.sh之前。
启动后脚本	通过变更触发的应用启动，在应用容器每次启动后，如调用Tomcat的启动脚本start.sh之后。具体触发点为：当应用配置了健康检查URL时，则通过健康检查后触发；若没有配置健康检查URL，则在端口存在后触发，不建议只使用端口检查方式，通常在应用未完全启动前，应用容器早已经打开网络端口。
停止前脚本	通过变更触发的应用停止，在应用容器每次停止前。
停止后脚本	通过变更触发的应用停止，在应用容器每次停止后。

### 查看挂载脚本执行结果

在修改挂载脚本后，您可[查看应用变更](#)。下图以执行准备实例脚本为例展示执行结果。



## 常见问题

怎么查找发送远程命令的执行状态？

在变更记录日志中的InvocationId可查找到相应的发送远程命令的命令执行状态。

## 6.3. 健康检查

健康检查是指由EDAS Agent（以下简称为Agent）针对容器和应用进行定时检查和汇报，然后将结果反馈在控制台上的过程。健康检查能帮助您了解集群环境下整个服务的运行状态，从而为审查与定位问题提供帮助。

### 健康检查简介

健康检查分为两部分：端口健康检查和URL健康检查。

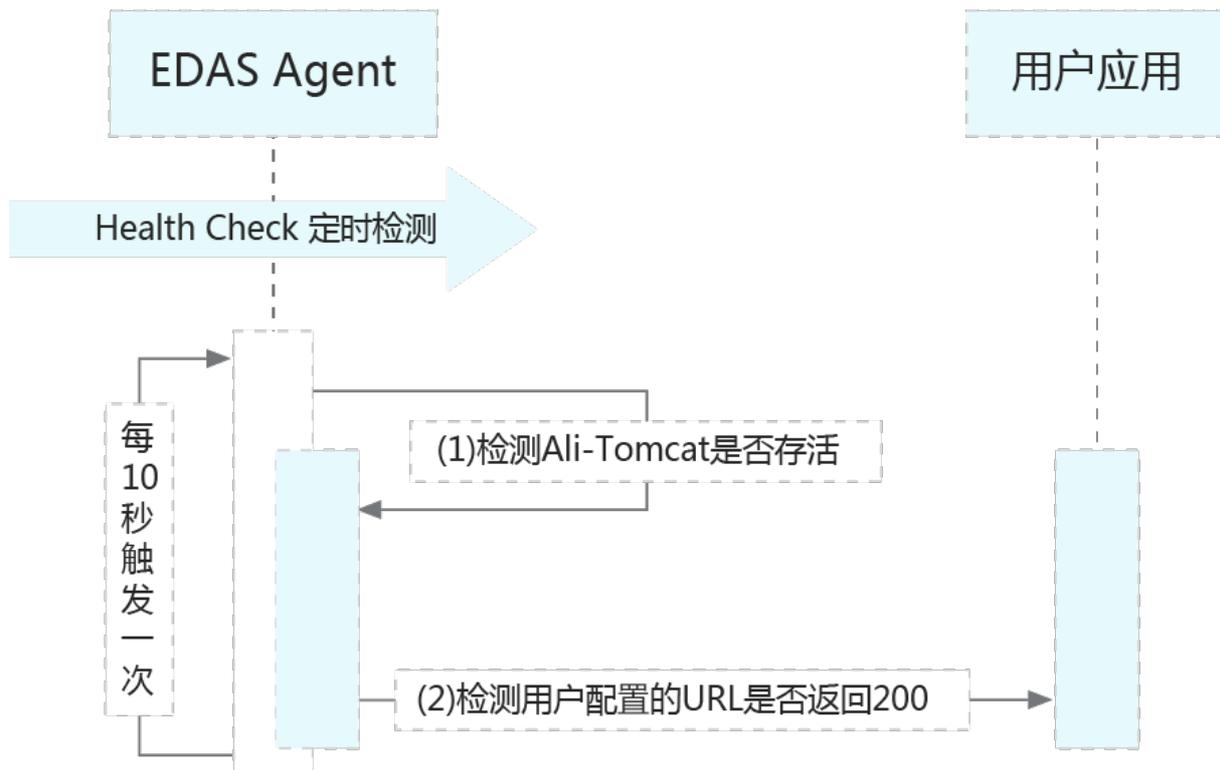
- 端口健康检查就是检查应用进程的相关端口是否成功打开，如果端口可以连接上，则该健康检查就会成功。因此如果应用正常运行，则端口健康检查一定是正常，但反之不一定正确，此时就需要进行URL健康检查。
- URL健康检查会按照固定的时间间隔（间隔为10秒），向应用程序、服务器或其他资源自动提交请求，以验证其是否可到达、是否可用及功能是否正常，如果该地址HTTP状态码返回200则认为健康检查通过。

 **说明** 如果您没有配置URL，将不会进行健康检查。

### 健康检查流程

您可以配置健康检查来监控通过IP地址或域名指定的终端节点。EDAS健康检查按照固定的时间间隔，向应用程序、服务器或其他资源自动提交请求，以验证其是否可到达、是否可用及功能是否正常。您也可以通过配置健康检查URL来发出与用户发出的请求类似的请求，以验证应用功能的运行状态。

下图为健康检查机制中的检查流程图，该图以每10秒触发一次健康检查作为示例。



图中的 (1) 和 (2) 流程的具体说明如下：

1. 检测应用所在的Ali-Tomcat是否存活。
  - 如果存活，进行第2步检测。
  - 如果没有存活，检测结束，说明健康检查失败。
2. 检测配置的URL是否返回 200 。

🔗 说明 如果您没有配置URL，将不会进行健康检查。

### 健康检查设置

在ECS集群中，您可以在创建应用过程中设置健康检查URL，也可以在应用部署完成后在应用设置页面添加或者修改健康检查。

1. 登录EDAS控制台。
2. 在左侧导航栏中单击应用列表，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在应用列表页面单击具体的应用名称。
3. 在基本信息页面的应用设置区域，单击健康检查右侧的修改。

🔗 说明 只有ECS集群中的应用会显示健康检查的设置项。

4. 在修改健康检查URL对话框设置健康检查。



- **端口健康检查**：开启来检查应用端口是否能正常访问。
- **健康检查URL**：开启并配置健康检查URL，监测配置的URL是否返回200，若返回200则说明服务可用。配置示例如下：
  - 一个WAR包应用的默认运行容器是Tomcat，则应用端口默认为 8080，则健康检查的URL可以设置为 `http://127.0.0.1:8080/order/healthCheck.html`。
  - 若该WAR包应用配置了容器路径为根路径，端口设置为 8081，WAR包中包含一个可以标识健康状态的文件 `healthcheck.html`，则健康检查URL为 `http://127.0.0.1:8081/healthcheck.html`。

**注意** 设置URL健康检查时选取的URL地址需要注意以下几点：

- 该地址在应用正常的情况下一定要返回200，应用不正常的时候返回非200。
- 该地址不要具有影响业务的操作。
- 该地址响应应该尽量迅速。
- 该地址最好不要调用其他内部接口，例如请求OSS资源，否则会造成对OSS资源非常大流量的请求。
- 该地址返回的数据量应当尽量小。

## 健康检查状态说明

在设置完应用健康检查之后，应用会有不同的显示状态。

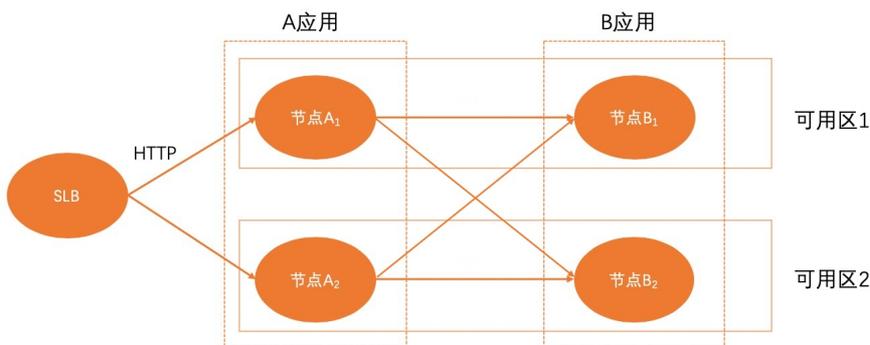
- **容器退出**：在健康检查的流程图中，当第1步发现Ali-Tomcat没有存活时，显示容器退出状态。
- **应用异常**：在健康检查的流程图中，当第2步发现在3s内您配置的URL返回非200时，显示应用异常状态。
- **运行正常**：在健康检查的流程图中，如果第1步与第2步均通过，则显示运行正常状态。  
检测过程中，如果发现您没有配置健康检查URL，此时应用的运行状态同样显示为运行正常，但运行状态右侧会显示感叹号，鼠标在叹号上悬停时会提示：请启用应用的健康检查URL，以便更精准的反映应用运行状态。
- **Agent异常**：在健康检查的流程图中，在EDAS Server端，如发现Agent在30秒钟之内没有任何状态的汇报，则显示Agent异常状态。
- **ECS已过期**：表示ECS在ECS控制台已被释放。

## 6.4. 同可用区优先路由

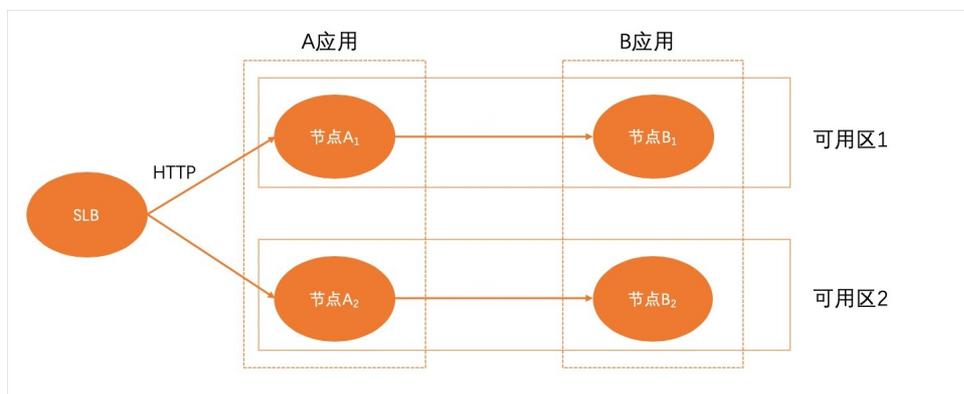
可用区优先是指在应用调用服务时，优先调用同可用区的服务提供者。Spring Cloud、Dubbo和HSF协议应用都支持配置可用区优先调用。本文介绍如何为ECS应用开启可用区优先调用。

### 使用场景

当应用部署在多个可用区的时候，应用之间互相调用会出现跨可用区的情况。调用链路如下图所示：如果可用区1的A应用调用可用区2的B应用，跨可用区调用导致网络延时增加，从而增加了HTTP的响应时间。



当应用服务开启同可用区优先，消费者应用会优先调用同可用区的提供者，调用链路如下图所示：避免了跨可用区带来的网络延时，从而减少了HTTP的响应时间。



如果当前可用区没有服务提供者时，服务调用者会调用到其他可用区的的服务提供者。

### 开启同可用区优先

要使用同可用区优先功能，您可以在服务提供者的应用详情页面上开启同可用区优先。

1. 登录EDAS控制台。
2. 在左侧导航栏中单击应用列表，在顶部菜单栏选择地域并在页面上方选择微服务空间，在应用列表页面单击部署在ECS集群中的应用名称。
3. 在应用基本信息页面的应用设置区域开启同可用区优先并设置阈值。

基本信息 实例部署信息

应用信息

ID	2886bb	微服务空间	cn-shanghai
集群类型	ECS集群	集群名称	默认集群
应用运行环境	标准Java应用运行环境	状态	运行中 1 / 共 1
部署包类型	FATJAR	部署包	默认分组: SPRING_CLOUD_PROVIDER.jar
负责人	test	应用描述	N/A
应用创建时间	2021-04-27 20:36:38	最后变更时间	2021-04-28 20:01:19

应用设置 基本信息 挂载脚本

应用设置	应用端口: 8080 Tomcat Context: 根目录 最大线程数: 400 Tomcat编码: ISO-8859-1 Use Body Encoding for URL	负载均衡 (私网)	添加
健康检查	端口健康检查被禁用 健康检查URL被禁用	负载均衡 (公网)	1.7.8888 修改   解绑
JVM 参数	编辑	同可用区优先	阈值: 0% 已关闭 开启

**说明** 只有在当前可用区实例数量占实例总数比例大于所设置的阈值时，同可用区优先调用策略才会生效。阈值填写范围为0~100。

# 7. 告警管理

## 7.1. 告警管理升级说明

EDAS告警管理全面升级，本次升级包括页面与功能的新增和变更。本文介绍新版告警管理基于旧版所做的一系列变更，帮助您了解新版与旧版的区别，快速上手新版告警管理。

 **说明** 升级后，原有报警规则均可以正常运行，告警也可以正常发送。

### 新增

#### 通知策略

通知策略功能可以对产生的告警事件进行分派、处理和发送。当规则被触发时，系统会以您指定的通知方式向处理人发送告警信息，以提醒处理人采取必要的解决措施。

#### 升级策略

对于一段时间内未解决的告警，您可以选择升级通知来提醒联系人及时解决。在通知策略中添加升级策略后，系统会以您指定的通知方式向处理人发送告警信息，以提醒处理人采取必要的解决措施。

#### 钉群通知

新版告警管理支持在钉钉群里通过卡片方式发送告警。您还可以在钉钉群里直接进行告警屏蔽、告警认领等操作。

### 变更

- 报警规则页面

移除原有报警规则页面。在应用监控中新增告警管理界面，您可在新增页面创建、查看、编辑和管理特定类型的告警规则。

- 通知方式

原有告警规则联系人、通知模板、恢复通知、静默时间和生效时间等功能替换为通知策略功能，现已无法编辑。若需编辑，您需在编辑页面一键迁移为通知策略后通过通知策略进行变更。

- 下线报警模板管理功能。

- 默认告警规则创建

新导入的应用监控应用不再自动创建默认告警规则。

升级后，您可以在告警管理 > 告警规则中创建、查看、编辑和管理应用监控告警规则。

告警规则							创建应用监控告警规则
请输入报警名称							Q
<input type="checkbox"/>	报警名称	类型	所属应用	告警规则	更新时间	状态	操作
<input type="checkbox"/>	事件报警 -应用紧急	默认应用紧急报警	● srs	默认应用紧急报警, 当应用出现: 内存泄漏异常、死锁异常、进程退出情况时, 发送紧急报警	2021-07-28 15:25:38	● 运行中	🔍   报警历史
<input type="checkbox"/>	事件报警 -应用紧急事件报警	默认应用紧急报警	● te	默认应用紧急报警, 当应用出现: 内存泄漏异常、死锁异常、进程退出情况时, 发送紧急报警	2021-07-27 10:42:18	● 运行中	🔍   报警历史
<input type="checkbox"/>	事件报警 -应用紧急事件报警	默认应用紧急报警	● te	默认应用紧急报警, 当应用出现: 内存泄漏异常、死锁异常、进程退出情况时, 发送紧急报警	2021-07-27 10:29:10	● 运行中	🔍   报警历史
<input type="checkbox"/>	em	应用监控报警	● em	最近1分钟节点机用户使用cpu_百分比最大值大于等于15	2021-07-23 21:22:31	● 运行中	编辑   停止   删除   报警历史
<input type="checkbox"/>	em -应用监控默认报警-数据库异常报警	默认应用监控报警	● em	最近5分钟数据库调用响应时间_ms平均值大于等于2000   最近5分钟数据库调用错误次数平均值大于等于1	2021-07-21 5:54:06	● 已停止	编辑   启动   删除   报警历史
<input type="checkbox"/>	em -应用监控默认报警-异常调用报警	默认应用监控报警	● em	最近5分钟调用响应时间_ms平均值大于等于2000   最近5分钟调用错误次数平均值大于等于1	2021-07-21 5:54:06	● 已停止	编辑   启动   删除   报警历史
<input type="checkbox"/>	em -应用监控默认报警-主机监控报警	默认应用监控报警	● em	最近5分钟节点机用户使用cpu_百分比平均值大于等于90   最近5分钟节点机空闲磁盘_byte平均值小于等于1048576	2021-07-21 5:54:06	● 已停止	编辑   启动   删除   报警历史

升级前，旧版和新版告警管理功能均支持在告警规则中选择已创建的联系人和联系人组进行通知，并且可以设置告警的通知模板、静默时间和生效时间等。当告警触发时，会按照告警规则的相关设置发送告警。

升级后，新版告警管理新增通知策略功能，用于告警通知。在创建通知策略时，您可以指定匹配告警规则的分派规则、对告警事件的处理和分组方式、通知人、通知方式、通知模板和静默时间或升级策略等。在创建告警规则时，您只能新建通知策略或者选择已有的通知策略，无法像升级之前直接指定通知相关的内容。

建议事先按需创建特定的通知策略，然后在创建告警规则时选择已有的通知策略来对告警事件进行处理和发送。或者您还可以选择在创建告警规则时均不指定通知策略，然后在创建通知策略时通过创建模糊匹配的分派规则来对满足分派条件的告警规则进行处理和发送。

#### 注意

- 已创建的旧版告警规则中，未使用通知策略的告警规则对应的告警事件和告警发送记录在新版无法查看。您只能在告警管理 > 告警发送历史中，单击返回旧版进行查看。
- 在编辑通过旧版创建的告警规则时，通知方式相关的内容均已经无法编辑。若需编辑，则需单击编辑页面上的一键迁移，将当前告警规则的通知方式自动转化为对应的通知策略，然后编辑该通知策略并保存。但需注意：
  - 请详细核对自动创建的通知策略各项是否满足您的告警需求。通知策略部分新增功能均为默认值，您可以参考文档，按需进行调整。
  - 使用通知策略后，告警通知的格式和内容会发生改变。请注意将钉群设置的关键词包含在新的告警内容中，否则钉群将无法接收告警。
  - 告警规则和通知策略是松耦合，一个告警规则产生的告警事件可以被多个通知策略匹配到，一个通知策略也可以匹配多个告警规则产生的告警事件。

新版告警管理移除了报警模板功能。新版中告警规则支持同时选择多个应用进行监控，建议使用此方式代替原有的报警模板功能。

升级前，在创建或导入应用监控应用集群时，EDAS会自动根据模板分别创建默认的告警规则。

升级后，在创建或导入应用监控应用集群时，将不再自动创建默认告警规则，但已创建的默认告警规则仍可正常运行。

## 默认告警规则创建变更

## 7.2. 名词解释

本文列出了EDAS告警管理的主要术语。

B | C | G | H | L | R | S | T | Y

### B

#### 补零、补一、补Null

报警数据修订策略可选择为补零、补一或补Null（默认）。此功能一般用于无数据、复合指标和环比同比等异常的数据修复。

- 补零：将被判断的数值修复为0。
- 补一：将被判断的数值修复为1。
- 补Null：不会触发报警。

应用场景：

- 异常情况一：无数据  
用户A想利用报警功能监控页面访问量。创建报警时，选择前端监控报警，设置报警规则为N=5时页面访问量的总和小于等于10则报警。若该页面一直没有被访问，则没有数据上报，不会发送报警。为解决此类问题，可将报警数据修订策略勾选为补零，将没有收到数据视为收到零条数据，符合报警规则，即可发送报警。
- 异常情况二：复合指标异常  
用户B想利用报警功能监控商品的实时单价。创建报警时，选择自定义监控报警，设置变量a的数据集为当前总价，变量b的数据集为当前商品总数，报警规则为N=3时（当前总价）/（当前商品总数）的最小值小于等于10则报警。若当前商品总数为0时，复合指标（当前总价）/（当前商品总数）的值不存在，则不会发送报警。为解决此类问题，可将报警数据修订策略勾选为补零，将复合指标（当前总价）/（当前商品总数）的值视为0，符合报警规则，即可发送报警。
- 异常情况三：指标环比、同比异常  
用户C想利用报警功能监控节点机用户使用CPU百分比。创建报警时，选择应用监控，设置报警规则为N=3时节点机用户使用CPU百分比的平均值环比下降100%则报警。若最近N分钟用户的CPU故障无法工作，即α无法获取，导致环比结果不存在，则不会发送报警。为解决此类问题，可将报警数据修订策略勾选为补一，将环比结果视为下降100%，符合报警规则，即可发送报警。

### C

#### 重复升级规则

如果升级策略判断过所有升级规则后，仍未解决告警，则将重新开始升级判断。默认为重复1次，您也可以将其调整为9以内的其他次数。

[\[回到顶部\]](#)

### G

#### 告警

指需要通知联系人去解决的告警。只有通过通知策略触发的需要解决的事件，才会创建告警。

#### 告警管理

指整个告警管理功能。

#### 告警卡片

钉钉群的机器人通过卡片的形式向钉钉群发送告警。联系人可以通过卡片查看、处理、解决告警。您可

以在控制台的钉群中配置钉钉群的机器人。

### 告警自动解决

当告警的事件都已恢复时，告警会自动设置为已解决。

### 故障

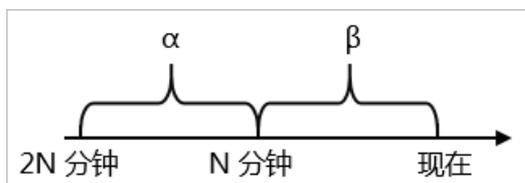
业务的规则会引起监控数据异常，监控工具检测到异常后会生成事件。

[\[回到顶部\]](#)

## H

### 环比上升 / 下降%

若 $\beta$ 为最近N分钟的数据（可选择为平均值、总和、最大值和最小值）， $\alpha$ 为前2N分钟到前N分钟的数据，环比为 $\beta$ 与 $\alpha$ 做比较。



## L

### 联系人

处理告警的运维人员，联系人可以通过钉钉查看、处理和解决告警。

[\[回到顶部\]](#)

## R

### 认领告警

如果联系人想将自己设置为尚未解决的告警的处理人，则可以认领该告警。

[\[回到顶部\]](#)

## S

### 升级超时

升级超时后，告警管理会将未确认或未解决的告警通知升级策略的联系人。默认升级超时设置为10分钟，您也可以将其设置在90分钟内的任一时间。

### 升级策略

在告警升级超时，告警管理会依次通知升级规则定义的不同批次的联系人。升级策略可以添加到特定的通知策略中。

### 事件分组

事件管理将多个事件汇总到单一告警中，用来减少需要联系人处理的告警。同时汇总关键信息，降低通知疲劳。

### 事件静默

不重要的事件可以将其静默，被静默的事件没有对应的告警。

### 事件自动恢复

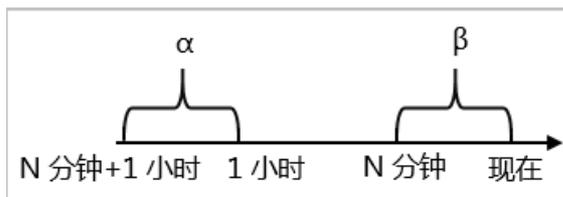
您可以设置在指定的时间后自动恢复事件。此时间过后，事件将自动被认定为已解决。事件自动恢复时间默认为5分钟。

[\[回到顶部\]](#)

## T

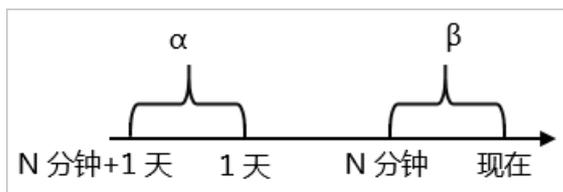
### 同比上升 / 下降% (与上小时)

若 $\beta$ 为最近N分钟的数据（可选择为平均值、总和、最大值和最小值）， $\alpha$ 为上小时最近N分钟的数据，与上小时同比为 $\beta$ 与 $\alpha$ 做比较。



### 同比上升 / 下降% (与昨日)

若 $\beta$ 为最近N分钟的数据（可选择为平均值、总和、最大值和最小值）， $\alpha$ 为昨日同一时刻最近N分钟的数据，与昨日同比为 $\beta$ 与 $\alpha$ 做比较。



## 通知

触发告警后，告警管理会立即发送通知给联系人。通知消息中包含触发的告警对象和告警标题。通知将通过短信、邮件、微信、钉群、WebHook等方式发送。

## 通知策略

通知策略确定了如何将事件汇总到单一告警中，以及如何将分配的告警通知发送给联系人。您可以通过设置电话，短信，邮件或钉群等方式，将需要解决或需要联系人知晓的告警1分钟内通知到他。

[\[回到顶部\]](#)

## Y

### 用户

指阿里云用户和RAM用户，用户可以通过EDAS控制台实现以下操作：

- 创建、编辑、修改通知策略。
- 创建、编辑、修改升级策略。
- 查看、处理、解决告警。

[\[回到顶部\]](#)

## 7.3. 联系人管理

### 7.3.1. 联系人

通知策略的分派规则被触发时可以向您指定的联系人发送通知。联系人支持通过电话、短信、邮件和钉钉等方式接收告警通知。

## 创建联系人

1. 登录[EDAS控制台](#)。
2. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
3. 在左侧导航栏中选择告警管理 > 联系人。
4. 在联系人页签上，单击右上角的新建联系人。
5. 在新建联系人对话框中输入联系人姓名，根据实际需求输入联系人手机号码、邮箱或钉钉机器人地址，设置是否接收系统通知，然后单击确认。

### 注意

- 手机号码、邮箱和钉钉机器人至少填写一项。每个手机号码或邮箱只能用于一个联系人。
- 最多可添加100个联系人。
- 获取钉钉机器人地址的方法，请参见[获取钉钉机器人Webhook地址](#)。
- 仅验证过的手机号码可以在通知策略中使用电话的通知方式，验证手机号的操作，请参见[验证手机号](#)。

## 验证手机号

仅验证过的手机号码可以在通知策略中使用电话的通知方式。

1. 登录[EDAS控制台](#)。
2. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
3. 在左侧导航栏中选择告警管理 > 联系人。
4. 在联系人页签上，选择为一个或多个联系人验证手机号。
  - 如需为单个联系人验证手机号：单击未验证手机号右侧操作列的更多图标，然后单击验证手机号。
  - 如需为多个联系人批量验证手机号：选择需要验证手机号的联系人，然后单击批量验证。  
系统将会给各联系人发送验证手机号短信。
5. 使用浏览器打开短信中的链接。



6. 在验证页面确认手机号信息，然后单击**验证**。

## 管理联系人

创建联系人后，您可以在**联系人**页签查询、编辑或删除联系人：

- 如需搜索联系人，请在**联系人**页签上，从搜索下拉框中选择姓名、手机号或Email，然后在搜索框中输入联系人姓名、手机号码或邮箱的全部或部分字符，并单击图标。🔍
- 如需编辑联系人，请单击联系人右侧**操作**列中的**编辑**，在**编辑联系人**对话框中编辑信息，然后单击**确认**。
- 如需删除单个联系人，请单击联系人右侧**操作**列中的**删除**，然后在提示对话框中单击**确认**。
- 如需删除多个联系人，请选择目标联系人，单击**批量删除**，然后在提示对话框中单击**确认**。

### 7.3.2. 联系人组

创建通知策略时，您可以将联系人组指定为通知对象，当通知策略的分派规则被触发时，EDAS告警管理会向该联系人组中的联系人通过电话、短信、邮件和钉钉等方式发送告警通知。

#### 前提条件

已创建联系人，具体操作，请参见**创建联系人**。

#### 创建联系人组

1. 登录**EDAS控制台**。

2. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
3. 在左侧导航栏中选择告警管理 > 联系人。
4. 在联系人页面单击联系人组页签，并单击右上角的新建联系人组。
5. 在新建联系人组对话框中输入组名，选择告警联系人，并单击确认。

## 管理联系人组

创建联系人组后，您可以在联系人组页签查询、编辑或删除联系人：

- 如需搜索联系人组，请在联系人组页签的搜索框中输入联系人组名称的全部或部分字符，并单击图标。🔍
- 如需编辑联系人组，请单击联系人组右侧的✎图标，并在编辑联系人对话框中编辑相关信息，然后单击确认。

🔍 说明 仅支持编辑当前用户创建的联系人组。

- 如需查看联系人组中的联系人信息，请单击联系人组左侧的▼图标展开联系人组。

🔍 说明 您可以在展开模式下移除联系人组中当前用户创建的联系人。如需移除，请单击目标联系人操作列中的移除。

- 如需删除联系人组，请单击联系人组右侧的✕图标，然后在弹出的提示对话框中单击确认。

🔊 注意

- 删除联系人组之前，请确保目标联系人组没有添加至通知策略中，否则可能导致告警通知无法发送。
- 仅支持删除当前用户创建的联系人组。

## 7.3.3. 钉钉机器人

在EDAS告警管理中创建钉钉机器人后，您可以在通知策略中指定对应的钉钉群用于接收告警。当通知策略的分派规则被触发时，系统会自动向您指定的钉钉群发送告警通知。钉钉群收到通知后，您可以在钉钉中对告警进行管理。

### 前提条件

已在钉钉客户端创建用于接收告警通知的钉钉群。

### 在EDAS控制台中创建告警通知钉群

1. 在钉钉群中创建自定义机器人并获取机器人Webhook地址，具体操作，请参见[获取钉钉机器人Webhook地址](#)。
2. 登录EDAS控制台。
3. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
4. 在左侧导航栏中选择告警管理 > 联系人。
5. 在联系人页面单击IM机器人页签。

6. 在IM机器人页签上，单击右上角的**新建机器人**。
7. 在**新建机器人**对话框中设置以下参数，然后单击**确认**。

 **注意** 在通知策略中需要选择通知方式为钉钉才能在钉钉群中接收告警，具体操作，请参见[通知策略](#)。

参数	说明
名称	自定义钉钉机器人的名称。
类型	选择类型为钉钉。
机器人地址	输入钉钉机器人的Webhook地址。
机器人是否发送每日统计	选中后，需要输入每日统计信息发送的时间点，使用英文逗号(,)分隔多个发送时间点，时间点格式为 HH:SS。EDAS告警管理将在设置的时间点发送今日产生告警的总数、解决数和待解决数。

## 在钉钉群中管理告警

在钉钉群中收到告警通知后，您可以在钉钉群里查看并管理告警。更多信息，请参见[在告警通知群中处理告警](#)。

## 相关操作

创建机器人后，您可以在IM机器人页签查询、编辑或删除机器人：

- 如需搜索指定的机器人，在搜索框中输入机器人名称的关键词，然后单击图标。
- 如需编辑机器人，单击机器人右侧操作列中的**编辑**，在**编辑机器人**对话框中修改信息，然后单击**确认**。
- 如需删除机器人，单击机器人右侧操作列中的**删除**，并在提示对话框中单击**确认**。

## 7.3.4. 通过Webhook自定义告警通知人

创建通知策略时，您可以将告警通知发送到自定义的Webhook地址中。ARMS告警管理支持对飞书、微信、钉钉等群组发送Webhook告警，本文以飞书为例，介绍如何创建Webhook告警。

### 步骤一：获取Webhook地址

1. 打开并登录飞书。
2. 单击图标，然后单击**创建群组**，新建一个用于发送告警的群组。
3. 单击群组设置图标，然后单击**群机器人**页签。
4. 在**群机器人**页签单击**添加机器人**。



5. 在添加机器人面板选择Custom Bot。



6. 在配置页设置显示名称和描述，然后单击添加。

< ×



**第一步：添加自定义机器人进群**

自定义机器人可以通过 webhook 向群聊推送来自外部服务的消息。请填写以下信息完成添加。[查看说明](#)

机器人名称\*

描述\*

19/256

7. 在添加情况区域单击复制链接，然后单击完成。

< ×

复制以下地址，在外部系统中 [配置 webhook 设置](#)

webhook 地址

请保管好此 webhook 地址。不要公布在 Github、博客等可公开查阅的网站上。地址泄露后可能被恶意调用发送垃圾信息

安全设置

- 自定义关键词 ?
- IP 白名单 ?
- 签名校验 ?

设置

- 仅群主和添加者可编辑、移除此机器人

## 步骤二：创建Webhook联系人

1. 登录EDAS控制台。
2. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
3. 在左侧导航栏中选择告警管理 > 联系人。
4. 在联系人页签上，单击右上角的新建webhook。
5. 在创建Webhook对话框中输入配置信息。

基本参数描述如下所示。

参数	说明
Webhook名称	必填，自定义Webhook名称。
Post和Get	必填，设置请求方法。URL不可超过100个字符。 此例中选择Post，并将 <a href="#">步骤一：获取Webhook地址</a> 中保存的Webhook地址粘贴至右侧文本框。
Header和Param	非必填，设置请求头，不可超过200个字符。单击+添加，可以添加其他Header信息或Param信息。默认请求头为Content-Type: text/plain; charset=UTF-8，Header和Param个数总数不能超过6个。 此例中设置以下两个Header： <ul style="list-style-type: none"> <li>Arms-Content-Type : json</li> <li>Content-Type : application/json</li> </ul>
通知模板	告警触发时发送的通知模板，非必填，在Post方法下出现，可使用\$content占位符输出通知内容，不可超过500个字符。通知模板如下： <pre> {   "告警名称": "{{ .commonLabels.alertname }}{{if .commonLabels.clustertype }}{{end}}",   "集群名称": "{{ .commonLabels.clustertype }} {{end}}{{if eq "app" .commonLabels._aliyun_arms_involvedObject_kind }}",   "应用名称": "{{ .commonLabels._aliyun_arms_involvedObject_name }} {{end}}",   "通知策略": "{{ .dispatchRuleName }}",   "告警时间": "{{ .startTime }}",   "告警内容": "{{ for .alerts }} {{ .annotations.message }} {{ end }}" } </pre> <p>此处以飞书为例可以设置如下文本格式：</p> <pre> {   "msg_type": "text",   "content": {     "text": "报警名称: {{ .commonLabels.alertname }}\n报警时间: {{ .startTime }}\n报警内容: {{ for .alerts }} {{ .annotations.message }}\n{{if .generatorURL }} 详情链接: {{.generatorURL}} {{end}} \n {{end}}\n"   } } </pre>

参数	说明
恢复模板	<p>告警恢复时发送的通知模板，非必填，在Post方法下出现，可使用\$content占位符输出通知内容，不可超过500个字符。恢复模板如下：</p> <pre> {   "告警名称": "{{ .commonLabels.alertname }}{{if   .commonLabels.clustername }}",   "集群名称": "{{ .commonLabels.clustername }} {{ end }}{{if eq "app"   .commonLabels._aliyun_arms_involvedObject_kind }}",   "应用名称": "{{ .commonLabels._aliyun_arms_involvedObject_name }} {{   end }}",   "通知策略": "{{ .dispatchRuleName }}",   "恢复时间": "{{ .endTime }}",   "告警内容": "{{ for .alerts }} {{ .annotations.message }} {{ end }}" }                     </pre> <p>此处以飞书为例可以设置如下文本格式：</p> <pre> {   "msg_type": "text",   "content": {     "text": "报警名称: {{ .commonLabels.alertname }}\n恢复时间: {{     .endTime }}\n报警内容: {{ for .alerts }} {{ .annotations.message }}     {{if .generatorURL }} 详情链接: {{.generatorURL}} {{ end }} \n {{ end     }}"   } }                     </pre>

6. (可选) 单击**测试**，验证配置是否成功。
7. 单击**创建**。

### 步骤三：设置通知策略

在通知策略中需要选择通知方式为WebHook才能在自定义的Webhook中接收告警。

1. 新建或编辑通知策略，选择通知人为Webhook联系人，在通知方式区域选中WebHook。具体操作，请参见[通知策略](#)。

 **说明** Webhook告警的超时时间为5秒，如果发出请求后5秒内没有返回，即没有收到告警信息，则表示发送失败。

## 7.3.5. 获取钉钉机器人Webhook地址

设置钉群或在联系人中设置钉钉机器人时需要先在钉钉群中获取自定义机器人Webhook地址。本文介绍如何获取钉钉机器人Webhook地址。

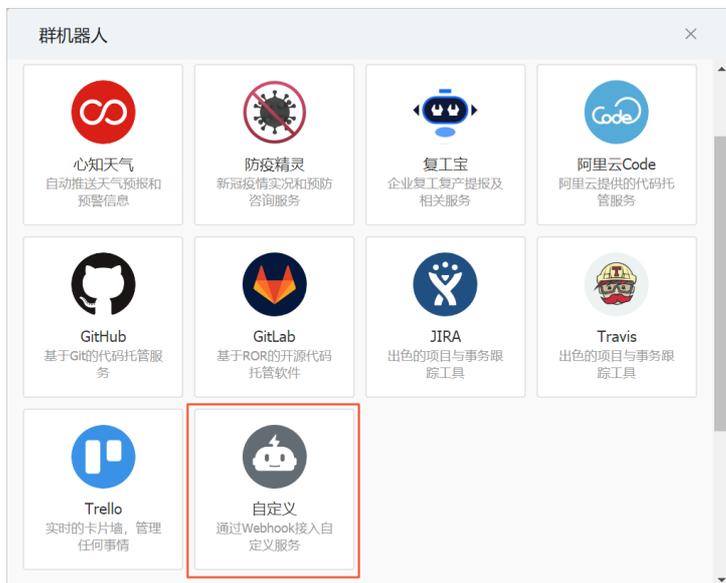
### 前提条件

已创建用于接收告警通知的钉钉群。

### 操作步骤

请按照以下步骤在钉钉群中添加自定义钉钉机器人并获取Webhook地址。

1. 在PC版钉钉上打开您想要添加报警机器人的钉钉群，并单击右上角的群设置图标。
2. 在群设置面板中单击智能群助手。
3. 在智能群助手面板单击添加机器人。
4. 在群机器人对话框单击添加机器人区域的+图标，然后选择添加自定义。



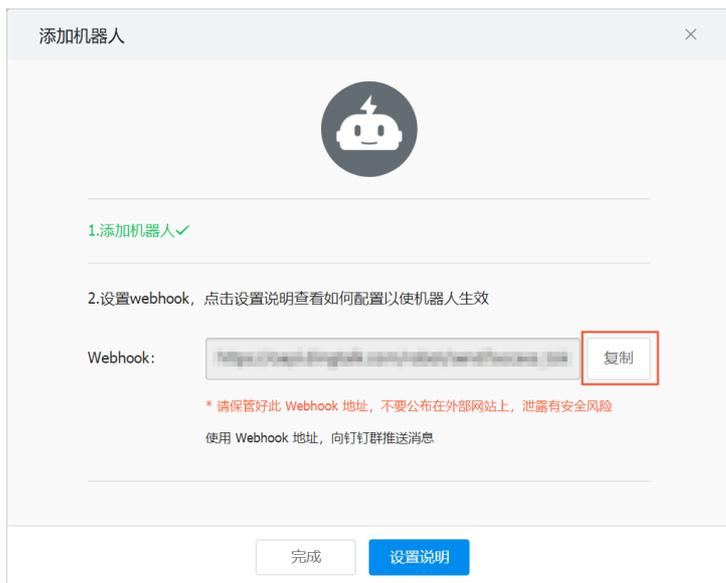
5. 在机器人详情对话框单击添加。
6. 在添加机器人对话框中执行以下操作。



- i. 设置机器人头像和名字。
- ii. 安全设置选中自定义关键词，设置关键词为告警。
- iii. 选中我已阅读并同意《自定义机器人服务及免责条款》。
- iv. 单击完成。

 说明 更多关于钉钉机器人的操作，请参见[自定义机器人接入](#)。

7. 在添加机器人对话框中复制生成的机器人Webhook地址，然后单击完成。



## 7.4. 升级策略

对于长期未解决的告警，可以选择升级通知来提醒联系人及时解决。在通知策略中添加升级策略后，系统会以您指定的通知方式向处理人发送告警信息，以提醒处理人采取必要的问题解决措施。

### 前提条件

创建联系人，具体操作，请参见[创建联系人](#)。

### 新建升级策略

1. 登录EDAS控制台。
2. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
3. 在左侧导航栏中选择告警管理 > 通知策略。
4. 在通知策略列表区域选择已创建的策略，单击  图标，并单击编辑按钮。
5. 在当告警生成时区域的升级通知方式选择升级策略，并单击添加。
6. 在弹出的对话框顶部文本框中输入升级策略名称。
7. 在升级规则区域，设置通知条件，即当告警在一段时间内未被认领或未被解决时发送告警通知。例如：当告警10分钟未认领时发送升级通知。

 说明 单击+添加规则，可以新增一条升级规则。

参数	说明
通知人	通知人支持设置联系人、联系人组和钉群。创建方法，请参见 <a href="#">创建联系人</a> 。

参数	说明
通知方式	通知方式支持钉钉、邮件、WebHook、短信和电话，可以同时选择多种方式。  <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p><span>?</span> <b>说明</b> 未验证手机号的联系人无法使用电话通知方式。验证手机号的操作，请参见<a href="#">验证手机号</a>。</p> </div>
通知时段	满足通知条件的告警会在设置的通知时段内重复发送告警通知。
重复次数	重复发送告警通知的次数。当告警不满足通知条件后，告警通知将不再发送。  <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px;"> <p><span>?</span> <b>说明</b> 至少发送一次告警，即设置为0时也会发送一次告警。</p> </div>

8. 设置完成后，单击右上角的  图标。

## 7.5. 应用监控告警规则（新版）

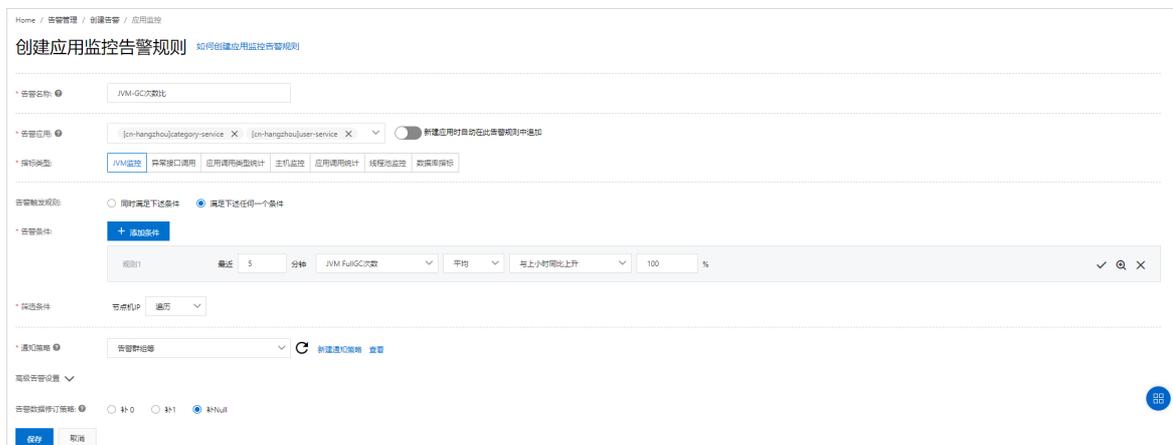
通过创建应用监控告警规则，您可以制定针对特定应用监控的告警规则。当告警规则被触发时，系统会以您指定的通知方式向告警联系人或钉群发送告警信息，以提醒您采取必要的解决措施。

### 前提条件

已接入应用监控，请参见[应用监控接入概述](#)。

### 操作步骤

1. 登录EDAS控制台。
2. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
3. 在左侧导航栏中选择告警管理 > 告警规则。
4. 在告警规则页面的右上角单击创建应用监控告警规则。
5. 在创建应用监控告警规则页面输入所有必填信息，完成后单击保存。



参数	说明
告警名称	告警的名称。例如：JVM-GC次数同比告警。
告警应用	选择需要设置告警的应用。可以选择多个应用。
新建应用时自动在此告警规则中追加	是否将之后接入的应用自动接入当前告警。
指标类型	<p>选择监控指标的类型：</p> <ul style="list-style-type: none"> <li>◦ JVM监控</li> <li>◦ 异常接口调用</li> <li>◦ 应用调用类型统计</li> <li>◦ 主机监控</li> <li>◦ 应用调用统计</li> <li>◦ 线程池监控</li> <li>◦ 数据库指标</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> <b>说明</b> 不同的指标类型，告警规则的条件字段和筛选条件不同。</p> </div>
告警触发规则	<ul style="list-style-type: none"> <li>◦ 同时满足下述规则：需满足所有告警条件才会触发告警。</li> <li>◦ 满足下述一条规则：满足任意一条告警条件就会触发告警。</li> </ul>
告警条件	<p>单击+添加条件，设置告警规则表达式。例如：最近5分钟JVM FullGC次数平均与上小时同比上升100%。然后单击右侧图标。</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> <b>说明</b></p> <ul style="list-style-type: none"> <li>◦ 单击规则右侧图标，可以修改告警规则表达式。</li> <li>◦ 单击规则右侧图标，可以预览应用在当前规则下的指标走势图。</li> <li>◦ 单击规则右侧图标，可以删除该告警规则表达式。</li> <li>◦ 若需设置多条告警规则，单击+添加条件，即可编辑第二条告警规则。</li> </ul> </div>
筛选条件	<p>告警指标的维度：</p> <ul style="list-style-type: none"> <li>◦ 无：告警内容中透出这个维度所有数值的和。</li> <li>◦ =：告警中只透出当前设置维度的内容。</li> <li>◦ 遍历：会在告警内容中透出实际触发告警的维度内容。</li> </ul>

参数	说明
通知策略	<ul style="list-style-type: none"> <li>不指定通知规则：告警被触发时不会发送告警，仅当通知策略的分派规则被触发时才会发送告警。</li> <li>指定通知规则发送告警：告警被触发时，ARMS通过指定通知策略的通知方式发送告警信息。您可以选择已有的通知策略，也可以新建一个通知策略。更多信息，请参见<a href="#">通知策略</a>。</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><span style="color: #00aaff;">?</span> <b>说明</b> 单击查看，可以查看选中的通知策略详情。</p> </div>
高级告警设置	
告警数据修订策略	<p>用于无数据、复合指标和环比同比等异常数据的修复。当告警指标没有达到设置的条件时，告警数据补0、补1或不补充数据。</p> <ul style="list-style-type: none"> <li>补零：将被判断的数值修复为0。</li> <li>补一：将被判断的数值修复为1。</li> <li>补Null：不会触发报警。</li> </ul> <p>更多详细信息，请参见<a href="#">告警管理名词解释</a>。</p>

## 管理告警

创建的应用监控告警规则在告警规则页面上，您可以对告警规则执行启动、停止、编辑、删除、查看告警详情等操作。

1. 登录[EDAS控制台](#)。
2. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
3. 在左侧导航栏中选择告警管理 > 告警规则。
4. （可选）在告警规则页面的搜索框中输入告警名称，并单击搜索图标。

? **说明** 您可以输入告警名称的一部分内容进行模糊搜索。

5. 在搜索结果列表的操作列中，按需对目标告警规则采取以下操作：

告警规则							创建应用监控告警规则
请输入报警名称							Q
<input type="checkbox"/>	报警名称	类型	所属应用	报警规则	更新时间	状态	操作
<input type="checkbox"/>	gateway-应用监控默认报警-数据库异常报警	默认应用监控报警	gateway	最近5分钟数据库调用成功_平均大于等于2000   最近5分钟数据库调用失败_平均大于等于1	2021-04-26 14:24:46	运行中	编辑   停止   删除   报警历史
<input type="checkbox"/>	gateway-应用监控默认报警-异常调用报警	默认应用监控报警	gateway	最近5分钟调用成功_平均大于等于2000   最近5分钟调用失败_平均大于等于1	2021-04-21 23:33:42	运行中	编辑   停止   删除   报警历史
<input type="checkbox"/>	gateway-应用监控默认报警-主机监控报警	默认应用监控报警	gateway	最近5分钟节点用户使用cpu_百分比平均大于等于90   最近5分钟节点应用磁盘使用_平均最小	2021-03-26 15:51:16	已停止	编辑   启动   删除   报警历史
<input type="checkbox"/>	gateway-应用监控默认报警-进程异常报警	默认应用监控报警	gateway	最近1分钟vm_线程总数平均量与上小时同比下降%10	2021-03-26 15:51:16	已停止	编辑   启动   删除   报警历史

- 如需编辑告警规则，请单击编辑，在编辑告警页面中编辑告警规则，并单击保存。
- 如需删除告警规则，请单击删除，并在提示对话框中单击确认。
- 如需启动已停止的告警规则，请单击启动，并在提示对话框中单击确认。
- 如需停止已启动的告警规则，请单击停止，并在提示对话框中单击确认。
- 如需查看告警事件历史和告警发送历史，请单击告警历史，在告警事件历史和告警发送历史页面上查看相关记录。

### 相关文档

- [查看告警发送历史](#)
- [查看告警事件历史](#)

## 7.6. 查看告警发送历史

通过告警发送历史页面，您可以筛选并查看根据通知策略分派的告警内容，并管理告警。

### 前提条件

EDAS告警管理功能仅对2021年04月30日0点之后开通EDAS的阿里云账户支持。

### 功能入口

1. 登录[EDAS控制台](#)。
2. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
3. 在左侧导航栏选择告警管理 > 告警发送历史。

### 告警

告警页面显示了告警的等级、名称、处理人、创建时间和状态。



在告警页面，您可以执行以下操作：

- 设置筛选字段，然后单击搜索，查看对应的告警发送历史。

说明 打开更多开关，可以设置更多的过滤筛选字段。

字段	说明
告警名称	创建的告警规则的名称。
告警状态	告警目前的处理状态，共有以下3种状态： <ul style="list-style-type: none"> <li>○ 待认领</li> <li>○ 处理中</li> <li>○ 已解决</li> </ul>

字段	说明
告警等级	告警的严重级别。 告警级别有以下几种： <ul style="list-style-type: none"> <li>○ 默认</li> <li>○ P1</li> <li>○ P2</li> <li>○ P3</li> <li>○ P4</li> <li>○ P5</li> </ul>
处理人	告警的处理人。
更多	
通知策略	告警对应的通知策略。
创建时间	告警产生的时间段。

- 单击告警名称，可以查看目标告警的详细信息。更多信息，请参见[告警详情](#)。
- 对于未解决的告警，可以认领、解决、指定告警处理人或修改告警等级。具体操作，请参见[处理告警](#)。

## 告警详情

告警详情页面显示了告警等级、发送信息、状态，以及告警基本信息、关联事件和活动记录。



在告警详情页面，您可以执行以下操作：

对于未解决的告警，可以认领、解决、指定告警处理人或修改告警等级。具体操作，请参见[处理告警](#)。

在详情、事件和活动页签可以分别查看以下信息：

- 详情页签显示了告警创建时间、告警对象、处理人和通知人。
- 事件页签显示了告警关联事件、事件创建时间和事件状态。单击事件名称，可以查看目标事件的详细信息。更多信息，请参见[事件详情](#)。



- 活动页签显示了告警的活动记录。



## 事件详情

事件详情面板显示了事件的基本信息、监控数据和扩展字段。

事件详情
✕

---

事件名称 容器CPU使用率大于2%

创建时间 2021-04-25 14:47:52

事件等级 warning

事件描述 命名空间: arms-prom / Pod: arms-prometheus-ack-arms-prometheus- / 容器: arms-prometheus-operator cpu使用率超过2%, 当前值3.09%

对象类型 ManagedKubernetes

事件对象 cc-test-temp

事件数量 4985

事件状态 未恢复

开始时间 2021-04-23 20:36:00

结束时间 2021-04-25 14:52:45

事件地址 无

集成名称 ARMS-DEFAULT

集成类型 ARMS-Prometheus监控

告警类型 ARMS-Prometheus监控

---

监控数据 告警发生时间前后

● 容器CPU使用率

---

扩展字段

```

container: arms-prometheus-operator
severity: warning
_aliyun_arms_alert_level: ERROR
clustername: cc-test-temp
_aliyun_arms_alert_type: 101
_aliyun_arms_integration_name: ARMS-DEFAULT
alertname: 容器CPU使用率大于2%

```

在事件详情面板的**监控数据**区域，您可以执行以下操作：

- 在**监控数据**区域右上角，设置数据显示的时间段为告警发生时间前后的6小时、12小时或1天。
- 使用光标选中一段时间，可以查看指定时间段的监控数据。单击**重置**，可以将曲线图恢复。

## 处理告警

在控制台中，对于未解决的告警，可以认领、解决、指定告警处理人或修改告警等级。

1. 绑定手机号。

i. 在钉钉群中单击告警卡片中的查看详情。



ii. 在钉钉群中首次查看告警信息时会弹出绑定手机号面板，输入手机号，并单击发送验证码。



iii. 收到验证码后输入验证码，然后单击确定。

2. 登录EDAS控制台。
3. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
4. 在左侧导航栏选择告警管理 > 告警发送历史。
5. 在告警页面右上角的选择操作人下拉框选择操作人。

### 说明

- 操作人的名称为钉钉昵称。
- 由于一个阿里云账号可能存在多人同时使用的情况，所以需要通过选择操作人来区分实际每次操作告警的人员。

6. 在目标告警右侧，或者单击目标告警，在告警详情页面：

- 单击认领，可以将自己设置为当前告警的处理人。
- 单击解决，可以关闭目标告警。
- 单击更多图标，然后单击指定告警处理人，可以指定钉钉群里的联系人为告警处理人。
- 单击更多图标，然后单击修改告警等级，可以修改告警等级。

## 7.7. 查看告警事件历史

通过事件列表页面，您可以筛选并查看所有告警事件。

### 前提条件

EDAS告警管理功能仅对2021年04月30日0点之后开通EDAS的阿里云账户支持。

### 功能入口

- 登录EDAS控制台。
- 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
- 在左侧导航栏选择告警管理 > 告警事件历史。

### 事件列表

事件列表页面显示了未恢复告警和已恢复告警的事件名称、通知策略、创建时间、事件数量、事件状态、事件对象和对象类型。

事件名称	噪音	通知策略	创建时间	事件数量	事件状态	事件对象	对象类型	操作
ccr_mysql er	否	通知策略2022-01-20 22:38:27	2022-01-20 19:28:49	1	已恢复	--	--	新建通知策略
ccr_mysql er	否	通知策略2022-01-20 22:38:27	2022-01-20 19:28:49	1	已恢复	--	--	新建通知策略
15证书过期告警	否	cloudops_	2022-01-19 11:42:38	8644	已恢复	--	--	新建通知策略
15证书过期告警	否	cloudops_	2022-01-12 22:52:53	9410	已恢复	--	--	新建通知策略

在事件列表页面，您可以执行以下操作：

- 设置筛选字段，然后单击搜索，可以查看对应的告警事件。

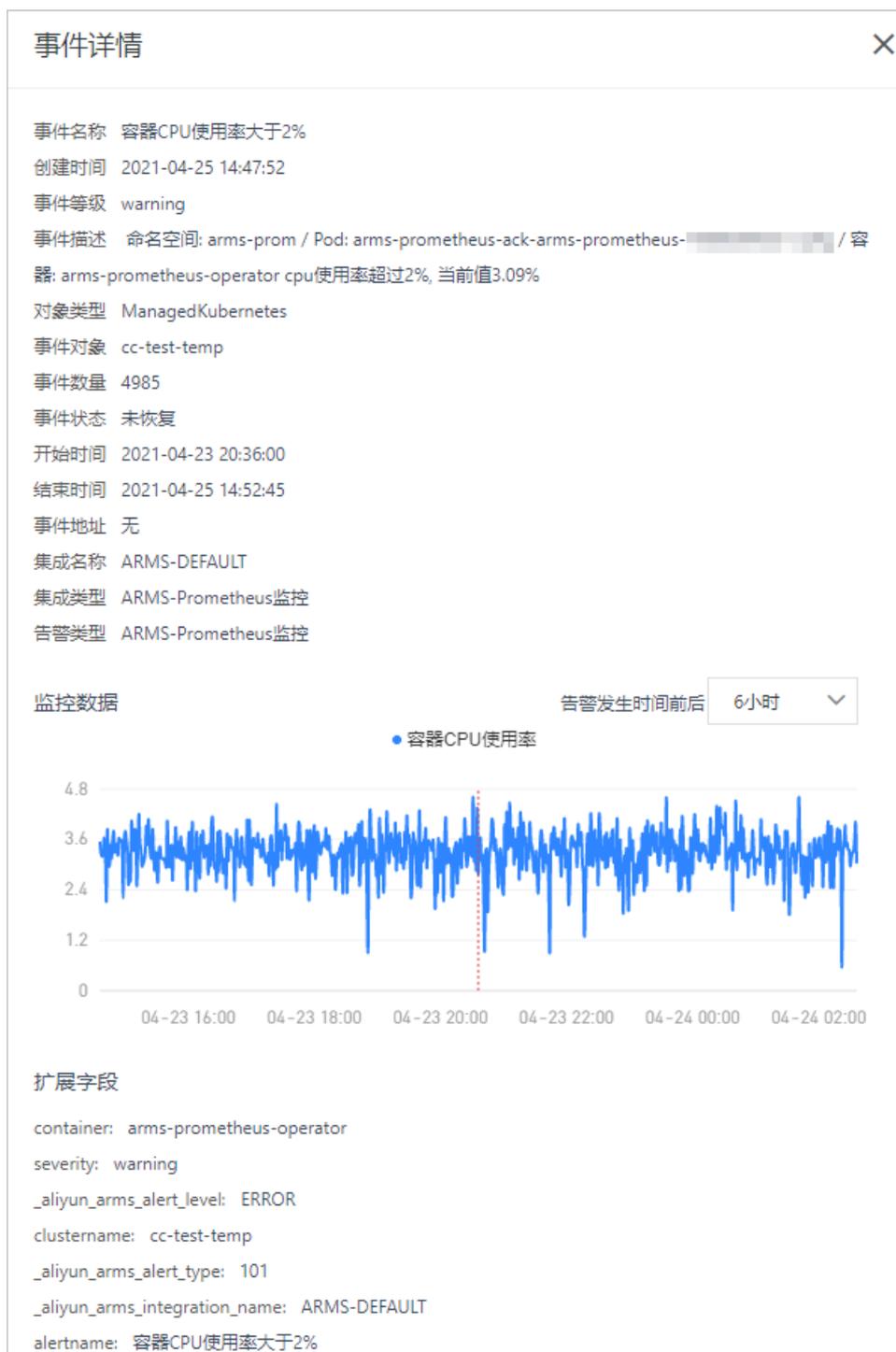
说明 打开更多开关，可以设置更多的过滤筛选字段。

字段	说明
事件名称	创建的告警规则的名称。
事件状态	告警事件的状态，共有以下3种状态： <ul style="list-style-type: none"><li>未恢复：告警事件持续被触发。</li><li>静默：在设置的自动恢复告警事件的时间内，和第一条告警事件名称和等级相同的告警事件的状态为静默。</li><li>已恢复：在设置的时间内，告警事件不再触发。</li></ul>
更多	
事件对象	监控任务名称或集群名称。
对象类型	前端站点、应用或Prometheus告警。

- 单击事件所在行，查看目标事件的详细信息。更多信息，请参见[事件详情](#)。
- 单击页面右上角的**发送测试事件**，在弹出的对话框中设置集成名称和事件内容，可以发送一次告警事件测试到指定集成中。

## 事件详情

事件详情面板显示了事件的基本信息、监控数据和扩展字段。



在事件详情面板的**监控数据**区域，您可以执行以下操作：

- 在**监控数据**区域右上角，设置数据显示的时间段为告警发生时间前后的6小时、12小时或1天。
- 使用光标选中一段时间，可以查看指定时间段的监控数据。单击**重置**，可以将曲线图恢复。

## 7.8. 创建应用监控告警并发送告警通知

本文通过示例说明如何创建应用监控告警并处理告警通知。

### 前提条件

已接入应用监控，具体操作，请参见[接入应用监控](#)。

## 背景信息

EDAS应用监控告警支持对采集到的各种应用指标配置告警规则。您可以根据需求和业务场景配置合理的告警规则，即可在应用不符合预期时收到通知，以便及时做出响应。本文以下述场景为例说明如何使用EDAS应用监控告警规则。

示例场景：核心业务请求延时过高或成功率下降。

示例应用名称：user-service。user-service应用是部署在阿里云容器服务Kubernetes版集群上面的一个Java应用，且已经成功接入了EDAS应用监控。

user-service应用是公司的核心服务，如果其接口可用性下降将导致严重的后果，因此，当此应用提供的接口调用延时增加或者成功率下降时，运维人员需及时收到通知并且迅速响应。

在此背景下，运维人员需要关注的指标有：

- 应用提供服务的平均响应时间
- 应用提供服务的请求成功率

## 步骤一：验证应用是否成功接入应用监控

执行以下步骤验证user-service应用是否成功接入应用监控。

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏选择应用列表，在展开的列表中单击user-service应用名称。
  - i. 在左侧导航栏单击应用总览。  
进入应用总览页面，查看应用总览页面是否有数据。
  - ii. 在左侧导航栏单击应用详情。  
进入应用详情的概览页签，查看概览页签是否有数据。

如果上面两个页面有数据，说明应用已经成功接入应用监控；如果上面两个页面都没有任何数据，说明应用没有成功接入，请将应用重新接入应用监控。

 **说明** 示例应用是容器服务Kubernetes版集群的Java应用，如果您的应用是其它类型，接入应用监控成功之后的页面显示可能与上图不同，但是只要页面有指标数据就说明已经接入成功。

## 步骤二：进入创建应用监控告警规则页面

1. 登录[EDAS控制台](#)。
2. 在顶部菜单栏，选择地域。
3. 在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
4. 在左侧导航栏中选择告警管理 > 告警规则。
5. 在告警规则页面右上角，单击创建应用监控告警规则。
6. 在创建应用监控告警规则页面，设置告警应用为user-service、指标类型为应用调用类型统计。

## 方法一

## 方法二

## 步骤三：创建告警规则

本示例中需要创建以下两条告警规则，并在告警触发时通过短信，邮件和钉钉群通知联系人：

- user-service应用提供请求延时过高的告警规则：

最近3分钟应用提供服务调用响应时间大于等于500 ms。

- user-service应用提供请求成功率下降的告警规则：

最近3分钟应用提供服务调用请求成功率小于等于90%。

**说明** 一般认为接口在短时间内有所波动为正常现象，持续性的异常才需要关注并且响应。所以在添加告警规则时，可以设置当最近3分钟的平均延时都比较高或者请求成功率低于90%的时候才需要告警。您在创建应用监控告警规则时，可以根据需要指定不同的时间和指标计算方法。

1. 在创建应用监控告警规则页面输入告警名称，添加告警条件。

此处先创建user-service应用提供请求延时过高的告警条件。

2. 设置通知策略。

- 如果您已经创建过通知策略，则在通知策略区域选择已有的通知策略。
- 如果您之前没有创建过通知策略，则执行步骤3新建通知策略。新建完通知策略后，在创建应用监控告警规则页面刷新并选择通知策略。

3. (可选) 单击通知策略右侧的新建通知策略。

在右侧弹出的新建通知策略面板中，选择通知人，并设置其他参数，然后单击确定。

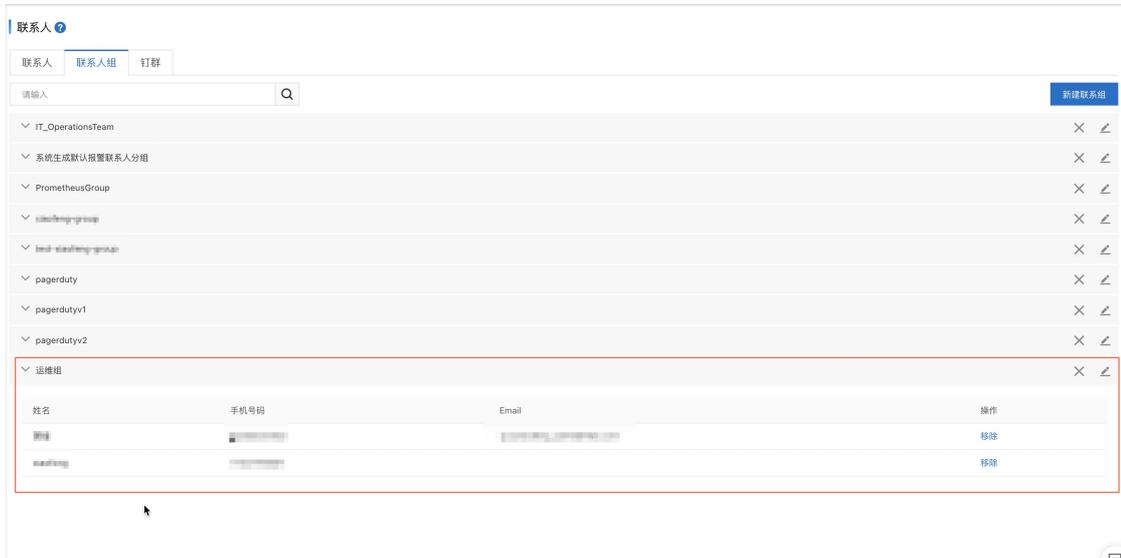
- 如果您已经创建过联系人，则在通知人区域选择已有的联系人。
- 如果您之前没有创建过联系人，则执行步骤4新建联系人。新建完联系人后，在新建通知策略面板刷新并选择通知人。

4. (可选) 单击通知人右侧的新建联系人。

在联系人页面创建联系人、联系人组和钉群。

- i. 在联系人页签，单击右上角的新建联系人。
- ii. 在新建联系人对话框中输入联系人参数，然后单击确认。
- iii. 在联系人组页签，单击右上角的新建联系组。

- iv. 在**新建联系组**对话框设置组名，并将上面创建的联系人添加至联系人组，然后单击**确认**。  
创建完成后，在**联系人组**页签中可以查看已创建的联系组。



- v. 在**钉群**页签，单击右上角的**新建钉群**。
- vi. 在**新建钉群**对话框设置钉群参数，然后单击**确认**。  
创建完成后，在**钉群**页签中可以查看已创建的钉群。



- 5. 告警规则创建完成后，单击**保存**。
- 6. 重复以上步骤创建user-service应用提供请求成功率下降的告警规则。  
创建完成后，在控制台的应用监控 > 应用监控告警规则页面可以查看创建的两条告警规则。

报警名称	类型	所属应用	告警规则	更新时间	状态	操作
user-service提供请求延时过高	应用监控报警	user-service	最近3分钟应用提供服务调用响应时间_ms大于等于500	2021-07-18 15:51:01	运行中	编辑   停止   删除   报警历史
user-service提供请求成功率下跌	应用监控报警	user-service	最近3分钟应用提供服务调用错误率小于等于90	2021-07-18 15:49:49	运行中	编辑   停止   删除   报警历史

### 查看告警通知

当满足告警条件时，收到的告警通知如下所示：

- 短信：



● 邮件:



● 钉钉群:



在钉钉群中收到告警通知后，您可以直接在钉钉群中处理告警。具体操作，请参见[在钉钉群中处理告警](#)。

## 处理告警

收到通知后，您可以在ARMS告警管理中查看告警的发送历史和详细告警时间信息。

1. 登录EDAS控制台。

- 在左侧导航栏，选择告警管理 > 告警发送历史。  
在告警页面可以查看告警发送历史详细告警时间信息。更多信息，请参见[查看告警发送历史](#)。



## 8. 应用运维

### 8.1. ECS 集群中的应用生命周期管理

本文档将介绍如何使用 EDAS 创建、部署、更新、扩缩容、启停、重置、重启和删除应用，以及升降级运行环境等管理操作。

#### 创建应用

您可基于 Dubbo、Spring Cloud 或 HSF 的框架来开发应用，并将应用程序构建为 WAR 包或 JAR 包，然后将应用部署至 EDAS 的 ECS 集群中。

- 在默认环境中将微服务应用部署至代购 ECS 实例
- 在自定义环境中代购的 ECS 实例上部署微服务应用
- 使用 toolkit-maven-plugin 插件升级应用 (ECS 集群)
- 使用 CLI 快速部署 EDAS 应用
- 在 Eclipse 中使用 Cloud Toolkit 快速部署应用至 ECS 集群
- 在 IntelliJ IDEA 中使用 Cloud Toolkit 快速部署应用至 ECS 集群
- 使用 Jenkins 创建持续集成

#### 部署应用

在 EDAS 创建一个未部署的应用后，您可以在应用基本信息页面右上角单击部署应用进行部署发布。

已部署过的应用，可在应用基本信息页面右上角单击部署应用来升级应用版本。

 **注意** 升级应用时选择的部署包类型需跟第一次部署时选择的部署包类型保持一致。

#### 回滚应用

如果应用在升级之后发现有异常，需将应用回滚之前的部署版本时，需在应用基本信息页面右上角单击回滚应用来回滚应用版本。

在实时跟踪升级时，如果应用实例升级突发异常，或部署包不可用、健康检查失败等 [变更流程错误码说明](#)，您可在变更详情页面右上角单击终止变更来回滚应用。

#### 扩容应用

应用扩容即通过增加应用实例的数量来增加应用的计算容量。您可以在应用实例负载过高时，在应用基本信息页面右上角单击应用扩容来增加应用实例。相关操作请参见 [扩容部署在 ECS 集群中的应用](#)。

 **说明** 增加实例时，如果所增应用实例加上应用原有实例所占的容量，超过应用资源的最大支持值，则扩容请求将失败；如果需要继续增加实例，请提 [工单](#) 进行增额申请，申请完成后进行扩容操作。

#### 缩容应用

应用缩容即通过减少应用实例的数量来减少应用的计算容量。在应用闲置时，可进入 [实例部署信息](#) 页签，单击 [批量操作实例](#)，在 [批量操作实例](#) 对话框选择要删除的应用实例，先批量停止，然后删除应用实例。

#### 启停应用

当应用处于运行状态时，可以根据业务需求停止应用，应用停止后将暂停应用实例的计费 and 计量。应用内必须至少包含 1 个应用实例时，才能进行应用启停操作。启停应用会使所有应用实例都随之一起启停。当应用实例大于 1 个时，启停应用实例只会影响本应用实例的运行状态；当应用实例只有 1 个时，启停应用实例会使应用随之启停。

 **注意** 应用停止时，如果其运行依赖其他产品，例如 SLB 和 VPC，所依赖的产品或者服务依然处于计费状态。

#### ● 启停应用

在应用基本信息页面右上角单击**停止应用/启动应用**，来启停应用。

#### ● 启停应用实例

您可以在**实例部署信息**页签内，单击单个应用实例操作的**启动**或**停止**来启停应用实例。

您可以在**实例部署信息**页签内，单击**批量操作实例**，在**批量操作实例**对话框批量选择应用实例来完成启停操作。

### 删除应用

您可以在应用基本信息页面的右上角单击**删除应用**，对应用进行删除。

### 重置应用

在应用的**实例部署信息**页签的应用实例的操作列单击**重置**，将重新配置对该应用实例的运行环境，重置后的运行时环境的版本不会变化。

### 重启应用

对于正在运行的应用实例，在应用的**实例部署信息**页签的应用实例的操作列单击**重启**，将会停止并重新启动应用实例。

### 运行环境升降级

ECS 集群中部署应用后，在后续运行、维护过程中可以升级、降级应用的运行环境，以满足不同的功能使用需求。相关操作请参见[升级或降级运行时环境](#)。

## 8.2. 扩容部署在ECS集群中的应用

当部署应用的ECS实例负载过高或者应用内没有实例时，可以通过添加实例来扩容。现EDAS支持三种扩容方式：从该应用部署的集群选择、基于现有实例规格购买和基于实例启动模板购买。

### 扩容实例的运行状态说明

扩容的实例的运行状态是根据所在应用的运行状态而定的。

- 如果扩容的时候应用已经部署并处于运行状态，那么扩容的实例会自动部署应用、启动、运行应用。
- 如果扩容的时候应用已经部署并处于停止状态，那么扩容的实例仍然会自动部署应用、启动、运行应用。
- 如果扩容的时候应用实例未部署，那么扩容的实例不会自动部署，也不会启动、运行应用，扩容的实例会处于停止状态。

### 进入应用扩容功能

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏中单击**应用列表**，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在**应用列表**页面单击具体的应用名称。
3. 在应用详情页面右上角单击**应用扩容**。在**购买实例**对话框的**扩容方式**页签内选择扩容的目标分组。
4. 选择**扩容方式**，并分别参照以下扩容方式完成后续扩容步骤。

## 从该应用部署的集群选择

该扩容方式是从应用所在集群内选择空闲的实例添加到应用来实现扩容。

1. 扩容方式选择从该应用部署的集群选择。
2. 在实例列表中选择用于扩容的空闲实例，然后单击扩容。

### 添加实例

扩容方式 购买信息 确认扩容

目标分组: 默认分组

扩容方式:  从该应用部署的集群选择  基于现有实例规格购买  基于实例自动模板购买

实例列表: 从该应用部署的集群选择ECS, 如果这里看不到添加到集群的ECS主机, 请在EDAS Console资源管理->ECS列表查看ECS所在的命名空间、集群名称与该应用所在的命名空间和集群是否一致。

已选择的实例: 暂无数据

实例ID/名称	VPC	IP 地址	规格
<input checked="" type="checkbox"/> i- EDAS scaled-clusterdoc-test-clus	vpc- doc-t	(公) 192.168.0.52 (私)	cpu : 1核 内存 : 4096MB

扩容 取消

页面上出现**扩容成功**的提示后，进入**应用实例部署信息**页面，查看扩容实例的运行状态，如果显示运行正常则说明扩容成功。

## 基于现有实例规格购买

该扩容方式是以所在集群内的任意一个实例作为规格模板，来配置购买实例实现扩容。

1. 扩容方式选择基于现有实例规格购买，该页面内会罗列出应用所在集群内的所有实例。
2. 选择您想作为规格模板的实例，选择**回收模式**，然后单击下一步。

? 说明 此次ECS实例代购默认会复制所选择ECS实例的规格、磁盘、网络、userdata、标签等基本信息。



3. 在购买信息页面，配置购买数量和登录密钥，并选中《云服务器ECS服务条款》|《镜像商品使用条款》，然后单击下一步。
4. 在确认扩容页面查看代购的实例信息，然后单击**确认扩容**。  
页面上方会出现已触发自动购买的流程，请查看**变更流程**获取实时信息的提示。
5. 进入应用的实例部署信息页面，查看扩容实例的运行状态，如果显示运行正常则说明扩容成功。

**说明**

- 基于现有ECS规格购买机器分为两个变更流程进行，第一个变更流程是EDAS从ECS为您代购ECS实例；第二个变更流程是将代购后的ECS实例自动扩容进应用中。
- 从您提交请求到应用开始扩容所需实例，前后约三分钟；两个变更流程执行相隔约10秒。
- 所有计费信息均按照ECS与EDAS的正常计费进行，此次操作不会产生额外的费用。
- ECS中默认的登录信息均以您自行设定的KeyPair进行，EDAS不会触碰您的任何私密信息。

### 基于实例启动模板购买

使用实例启动模板购买需您提前在ECS控制台创建实例启动模板，然后在EDAS控制台上基于您创建的模板来购买实例。创建启动模板的具体操作请参见[创建实例启动模板](#)。

**注意** 您所创建的启动模板必须和您的应用在同一VPC内，否则所创建的启动模板无法被有效选择。

1. 扩容方式选择**基于实例启动模板购买**。
2. 选择绑定模板和模板版本，选择回收模式，然后单击**下一步**。
  - **使用绑定模板**：在实例分组中绑定的启动模板，具体文档请参见[绑定实例启动模板](#)。
  - **使用指定模板**：在ECS控制台创建的模板。如果您创建过多个模板，需要选择**具体模板及版本**。
3. 在购买信息页面选择购买数量并选中《云服务器ECS服务条款》|《镜像商品使用条款》，然后单击**下一步**。

4. 在**确认扩容**页面，检查需要购买的ECS数量和启动模板信息。确认无误后，单击**确认扩容**。页面上方会出现已触发自动购买的流程，请查看**变更流程获取实时信息**的提示。
5. 进入应用的**实例部署信息**页面，查看扩容实例的运行状态，如果显示运行正常则说明扩容成功。

## 更多信息

- 您也可以使用弹性伸缩来动态调整ECS实例的数量，具体操作请参见[对ECS集群中的应用进行弹性伸缩](#)。
- 您也可以在集群中代购ECS实例导入到应用中实现扩容，具体操作请参见[在ECS集群中代购ECS实例](#)。

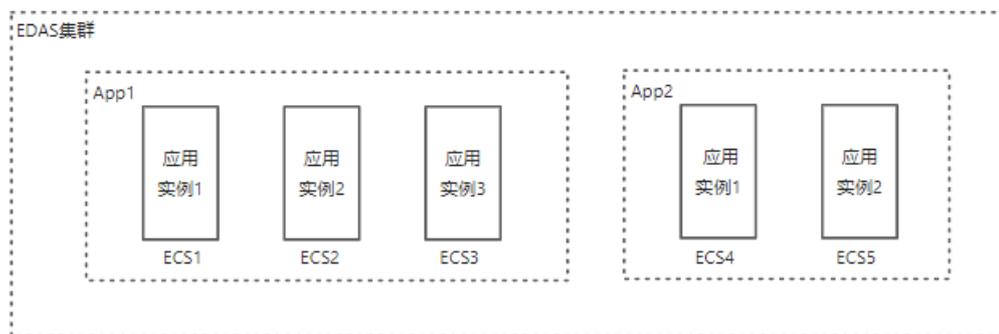
# 8.3. 负载均衡

## 8.3.1. 使用SLB访问ECS上的EDAS应用

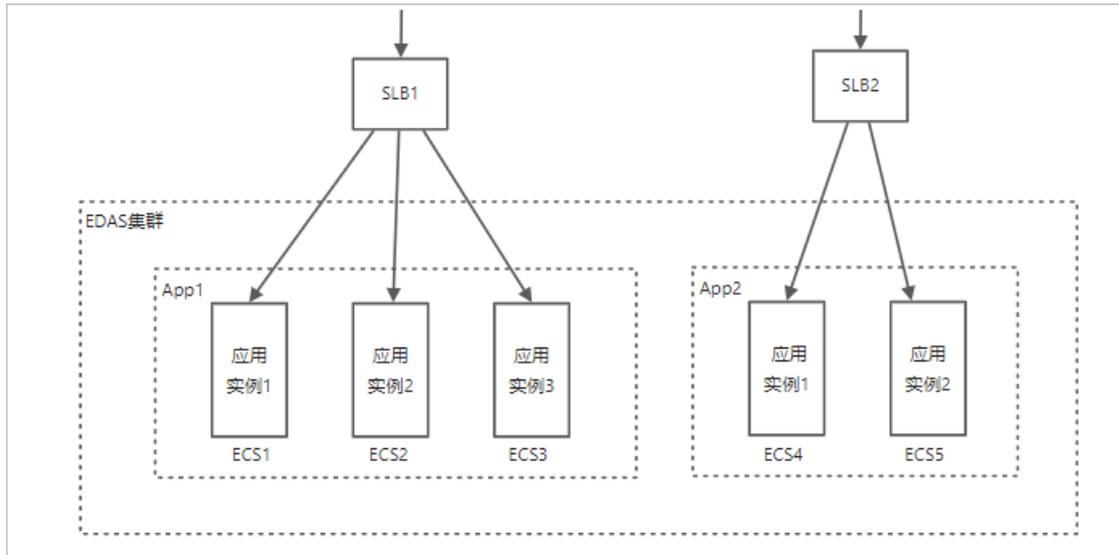
SLB是阿里云首选的流量分发控制服务，具有高可用、可扩展、低成本、安全和高并发等特点。EDAS应用可以绑定SLB，以SLB作为流量入口，将应用实例作为SLB的后端，由SLB根据转发策略，将流量分发到不同到应用实例。

### 背景信息

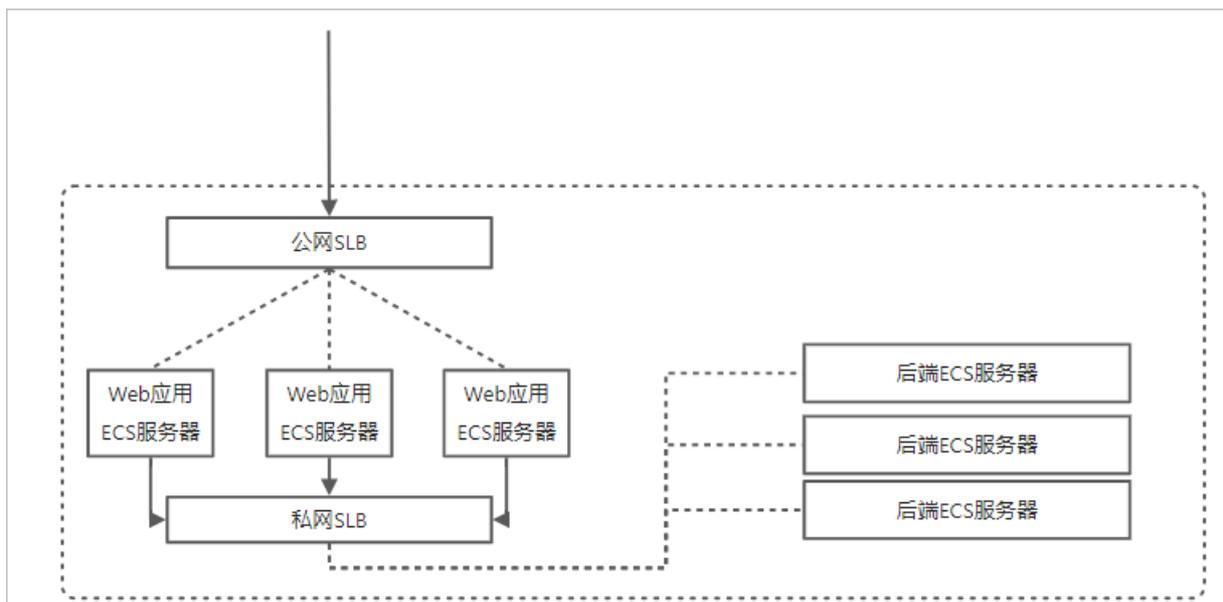
EDAS支持在阿里云ECS集群和混合云ECS集群上托管应用，您可以使用EDAS的ECS环境托管Spring Cloud和Dubbo等主流Java微服务框架应用，也可以托管HSF框架应用。应用托管到EDAS平台后，应用的多个实例需要作为一个整体对外提供服务，因此应用的流量要有一个统一的入口，需要一个能够将访问流量根据转发策略分发到后端应用不同实例的流量分发控制服务。



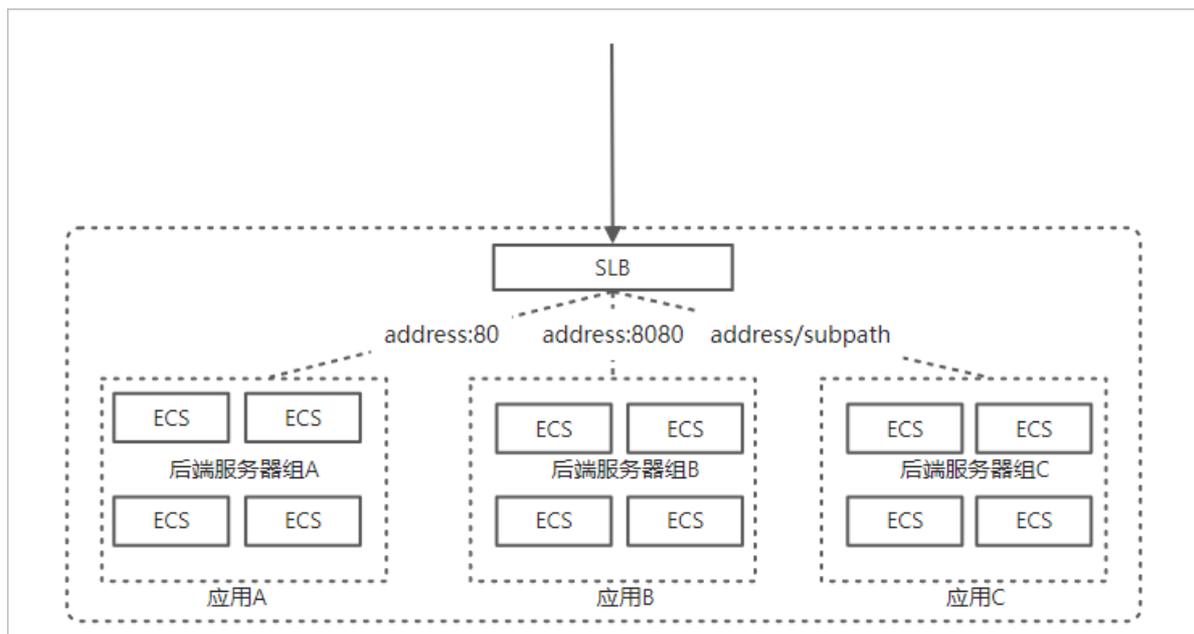
EDAS应用以SLB作为流量入口，将应用实例作为SLB的后端，由SLB根据转发策略，将流量分发到不同到应用实例。EDAS应用绑定SLB如下图所示：



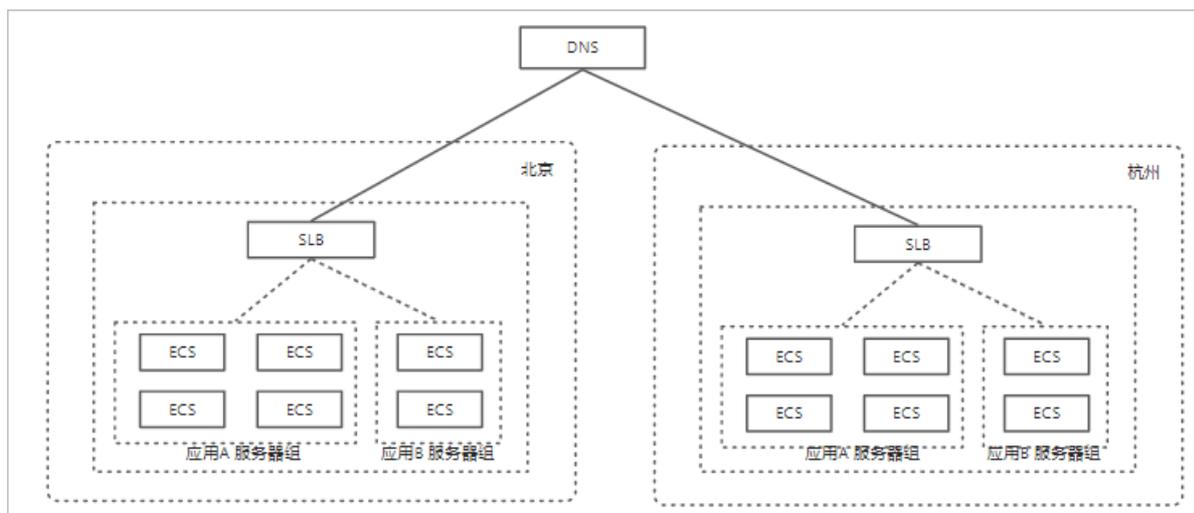
一个多机部署的应用能够结合SLB以保证应用的高可用，降低单台服务器的负载，实现流量的分发。SLB能够配置调度后端服务器的算法，如轮询、加权轮询。若开启会话保持功能，来自同一台客户端请求将转发到同一台应用服务器后端，以提高访问性能。



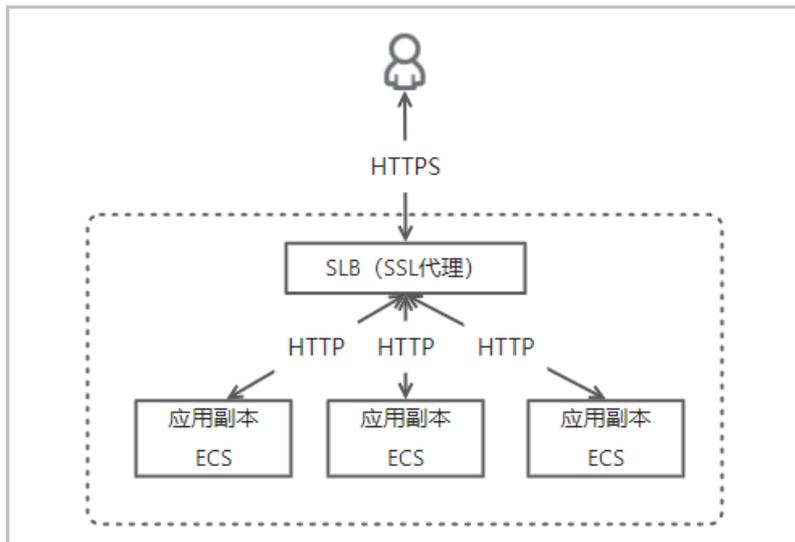
公网SLB和私网SLB的结合使用，能够提供Web应用的外部公网访问能力。后端服务器通过私网SLB进行通信，能够实现前端Web应用和后端应用分离，根据前后端服务性能需求的不同，各自进行后端服务器扩展。



SLB基于端口创建监听，通过规划应用后端服务器组，支持多个应用复用同一个SLB。此外，七层负载均衡支持基于域名和URL配置转发规则。



应用跨区域部署并结合SLB能够支持应用跨区域冗灾。在不同地域下部署负载均衡实例，并分别挂载相应地域内不同可用区的ECS服务器。上层利用云解析做智能DNS，将域名解析到不同地域的负载均衡实例服务地址下，可实现全局负载均衡。当某个地域出现不可用时，暂停对应解析即可实现所有用户访问不受影响。



SLB能够为用户提供SSL卸载功能，充当SSL代理服务器置于应用前端，卸除了后台应用ECS服务器对SSL加密数据处理的性能损耗，能够更加灵活地在云上进行应用流量转发，同时保证用户数据在对外公网传输的安全性。此外，SLB还能支持对DDoS、SYN Flood、UDP Flood、ACK Flood、ICMP Flood、DNS Flood等攻击防护，七层负载均衡还能额外支持一键集成WAF（Web Application Firewall），让应用层更加可靠。

SLB支持对DDoS、SYN Flood、UDP Flood、ACK Flood、ICMP Flood、DNS Flood等攻击防护，七层负载均衡还能额外支持一键集成WAF（Web Application Firewall），让应用层更加可靠。

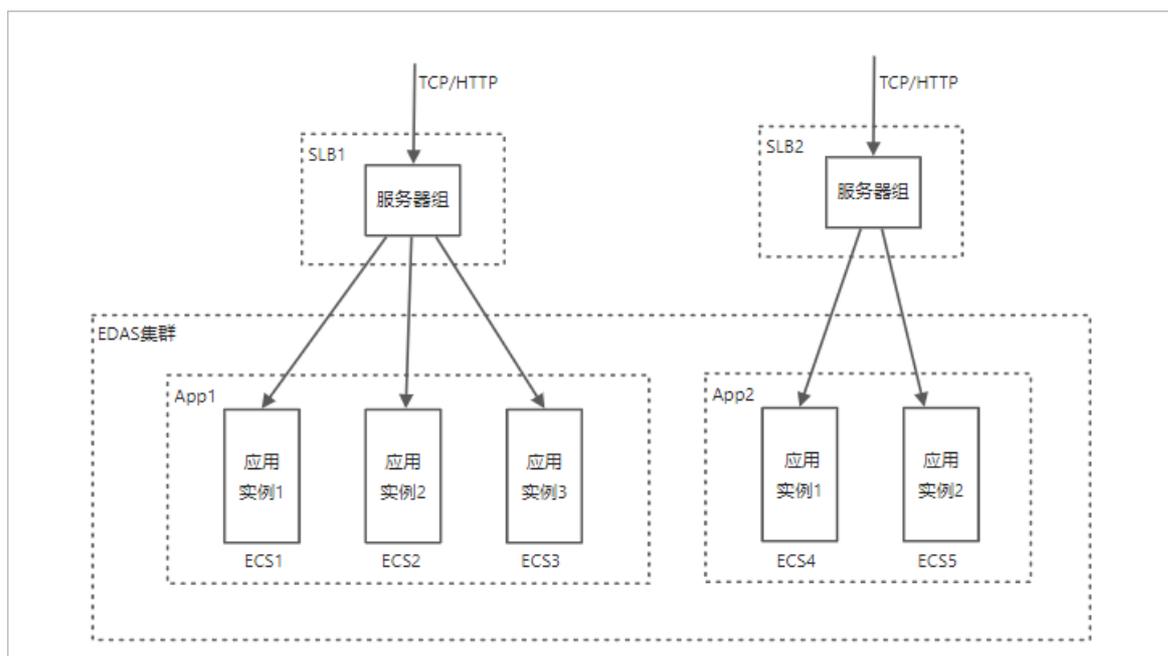
EDAS应用在使用负载均衡时，根据场景不同，分为以下使用方式：

- 单应用独享负载均衡实例。
- 多应用共享负载均衡实例。

本文以公网负载均衡实例为例，介绍上述两种使用场景，私网负载均衡实例的操作方式与公网一致，需要注意的是验证的时候需要选择相对应的网络环境。

### 单应用独享负载均衡实例

在EDAS控制台中，允许为单个应用配置独享的负载均衡实例。如下图所示，在一个EDAS集群中部署了2个应用，分别是App1和App2，App1包含3个实例，App2包含2个实例，可以分别为每个应用绑定单独的SLB。在EDAS控制台中绑定SLB时，EDAS会为应用创建SLB服务器组，然后将应用实例所在的ECS添加到服务器组中。

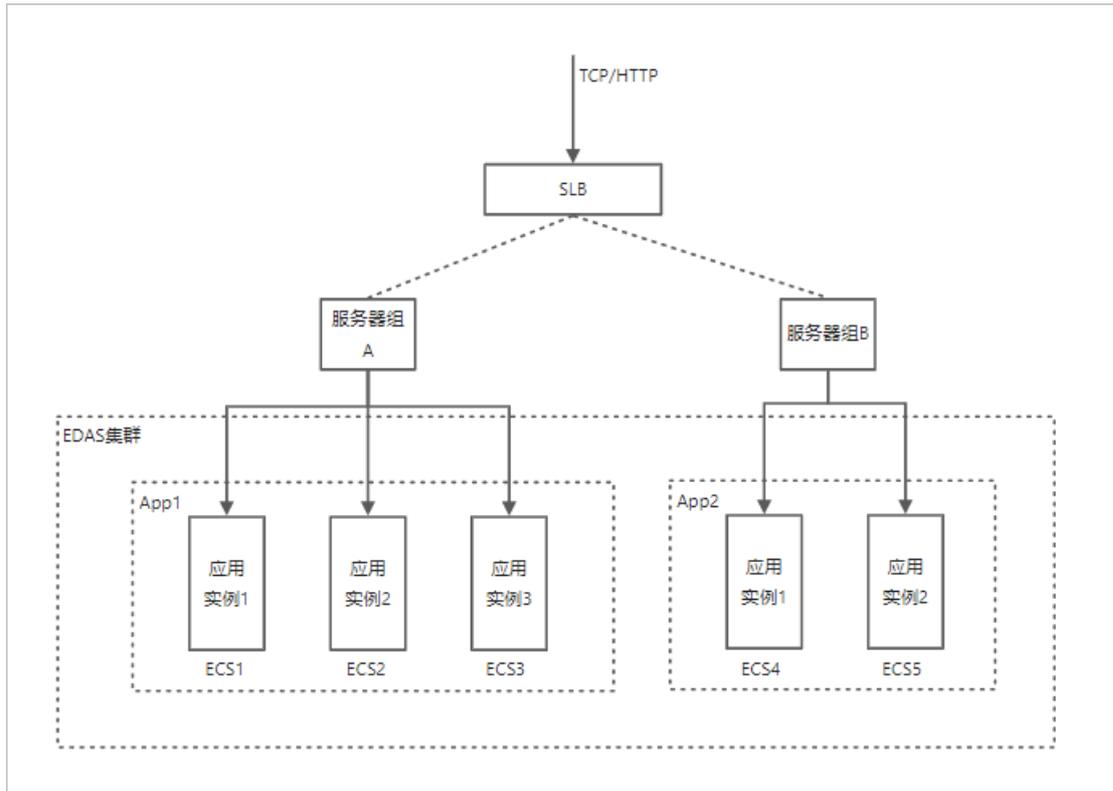


由于每个应用独占一个SLB实例，因此应用可以独占SLB的资源 and 带宽，每个应用的QPS取决于单个SLB的规格。

1. 创建SLB实例后，登录EDAS控制台，在应用列表页面顶部菜单栏选择地域，在页面中选择微服务空间，在集群类型下拉列表中选择ECS集群，然后单击目标应用名称。
2. 在应用总览页面的访问方式配置区域，单击负载均衡（公网）右侧的+图标。
3. 在弹出的页面中，完成选择负载均衡、选择监听配置和配置服务器组页面的配置，然后单击**确认变更**。详细操作流程请参见[应用独享负载均衡实例](#)。

### 多应用共享负载均衡实例

为了降低成本，减少SLB的实例数，EDAS允许为多个应用共享一个SLB。如下图所示，在一个EDAS集群中部署了2个应用，分别是App1和App2，App1包含3个实例，App2包含2个实例，两个应用共享同一个SLB实例。在EDAS控制台为应用绑定SLB时，EDAS会分别为每个应用在共享SLB实例中创建服务器组，然后将应用实例所在的ECS添加到服务器组中。由于多个应用共享同一个SLB实例，因此应用会共享SLB的资源 and 带宽。



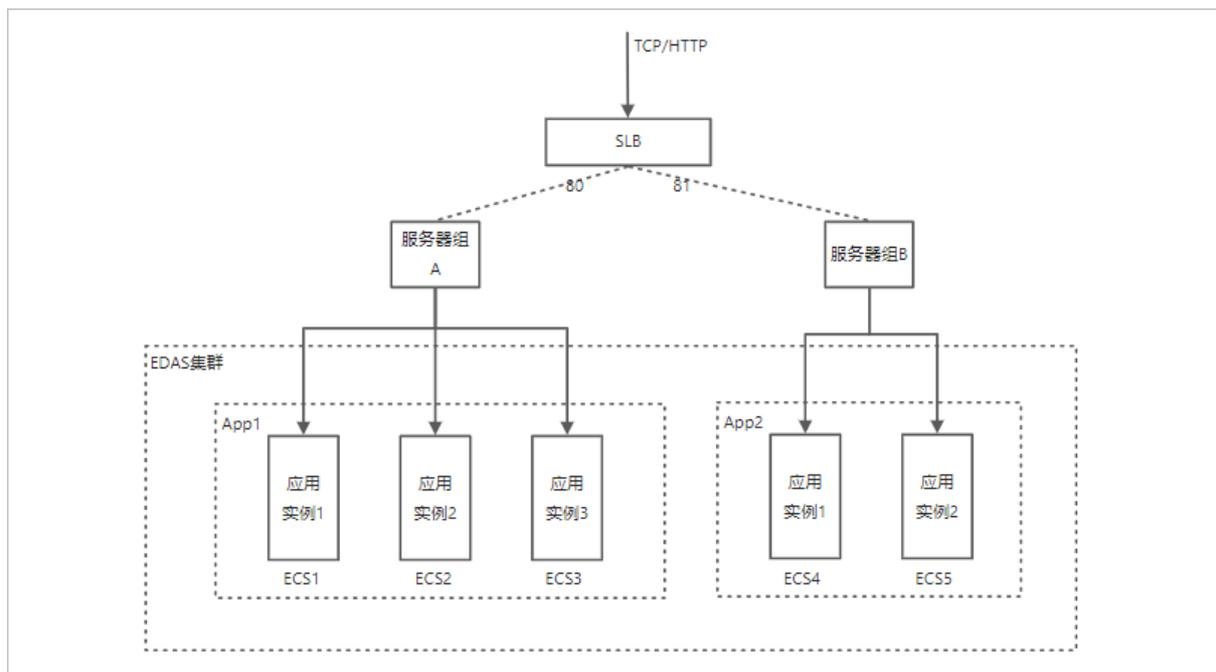
如图所示，SLB的入口流量可以根据流量转发规则将流量转发到不同的服务器组，从而将流量分发到不同的应用中。

EDAS支持两种流量分发规则：

- 使用端口实现流量分发。
- 使用域名或URL实现流量分发。

#### 使用端口实现流量分发

多应用共享负载均衡实例最简单的方式是为不同的服务器组配置不同的监听端口。如下图所示，两个应用所属的服务器组分别监听80和81端口，SLB接收到流量后，会根据目的端口进行流量分发，如果目的端口是80端口，会转发到服务器组A所属的应用App1，如果目的端口是81端口，则会转发到服务器组B所数的应用App2。



下面介绍为2个不同的应用绑定同一个SLB，通过配置不同的端口号实现流量分发的流程。

1. 创建SLB实例后，在EDAS控制台创建ECS类型的应用spring-cloud-provider，参考单应用独享负载均衡实例中的同名应用的配置过程为该应用配置公网负载均衡，监听协议依然选择TCP，端口为80。
2. 为spring-cloud-consumer应用配置公网负载均衡，配置过程与spring-cloud-provider应用类似，只是在配置端口时将端口配置为81。

配置完成后，假如SLB的IP地址为 `123.1.XX.XX`，可以使用 `http://123.1.XX.XX:80` 访问spring-cloud-provider应用，使用 `http://123.1.XX.XX:81` 访问spring-cloud-consumer应用。

在EDAS中多应用共享负载均衡实例使用端口进行流量分发的详细操作流程请参见[使用端口实现流量分发（共享负载均衡实例）](#)。

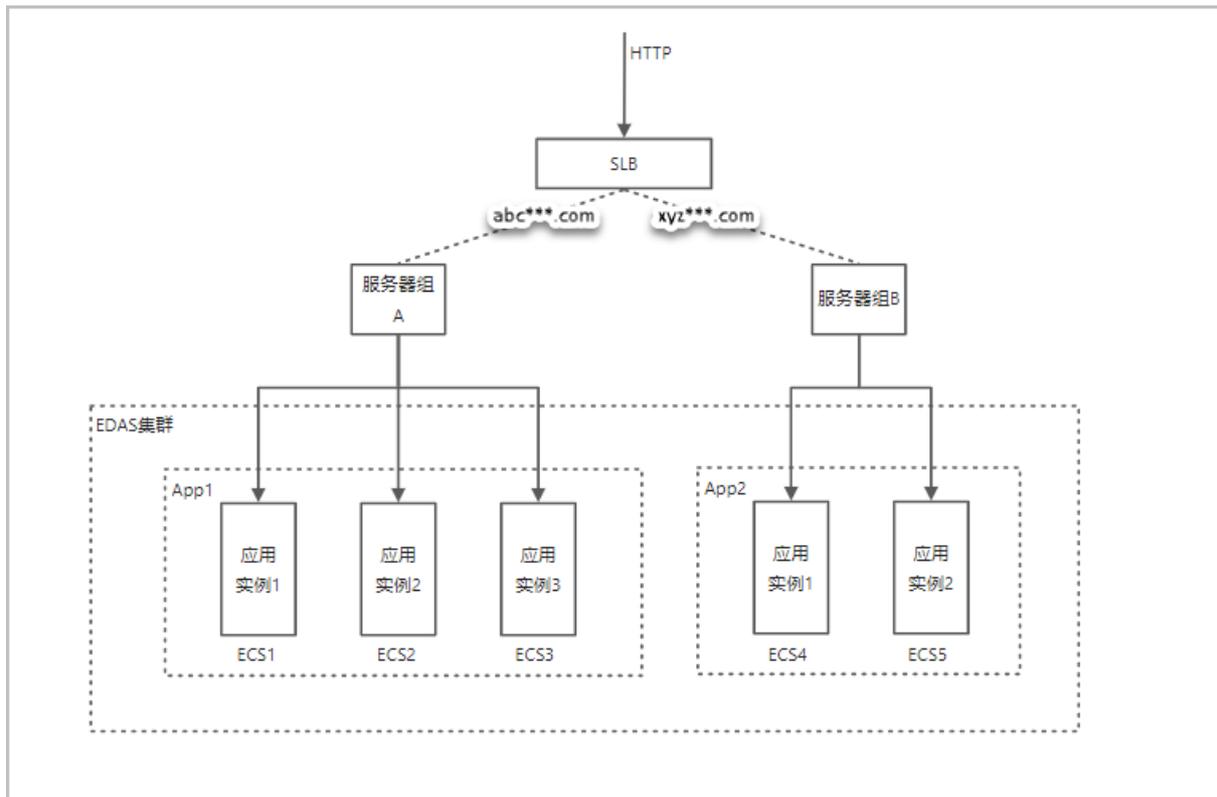
### 使用域名和URL实现流量分发

SLB支持HTTP协议，支持通过HTTP请求的域名和URL进行流量分发，如下图所示，SLB的IP地址绑定

了 `abc***.com` 和 `xyz***.com`，两个应用所属的服务器组分别绑定

到 `abc***.com` 和 `xyz***.com`，SLB接收到流量后，会根据HTTP请求的域名进行流量分发，如果域名

是 `abc***.com`，会转发到服务器组A所属的应用App1，如果域名是 `xyz***.com`，则会转发到服务器组B所数的应用App2。



下面介绍一下为2个不同的应用绑定同一个SLB，通过配置不同的域名实现流量分发的流程。

1. 创建SLB实例后，在EDAS控制台创建ECS类型的应用spring-cloud-provider，为该应用配置公网负载均衡，监听协议要选择HTTP，端口为80，将转发策略配置为 `provider.test.com`。
2. 为spring-cloud-consumer应用配置公网负载均衡，配置过程与spring-cloud-provider应用类似，监听协议要选择HTTP，端口同样为80，只是将转发策略配置为 `consumer.test.com`。

配置完成后，假如SLB的IP地址为 `123.1.XX.XX`，在Linux和macOS下可以通过修改`/etc/hosts`，添加一下条目进行Host绑定，方便测试。

```
123.1.XX.XX provider.test.com
123.1.XX.XX consumer.test.com
```

然后可以使用 `http://provider.test.com` 访问spring-cloud-provider应用，使用 `http://consumer.test.com` 访问spring-cloud-consumer应用。

在EDAS中多应用共享负载均衡实例使用域名和URL进行流量分发的详细操作流程请参见[使用域名或URL实现流量分发（共享负载均衡实例）](#)。

### 如何选择监听协议

负载均衡服务监听规定了如何将请求转发给后端服务器。EDAS中部署的应用在使用负载均衡时，需要选择监听协议。关于监听的更多信息，请参见[监听概述](#)。

在ECS集群的应用支持四层（TCP协议）和七层（HTTP协议）监听，您可根据应用场景选择监听协议。

- TCP：适用于注重可靠性，对数据准确性要求高，速度可以相对较慢的场景，如文件传输、发送或接收邮件、远程登录。
- HTTP：需要对数据内容进行识别的应用，如Web应用、小型的手机游戏等。

### 如何为应用选择负载均衡使用方式

应用在使用负载均衡时，根据场景不同，分为以下使用方式。

场景	相关文档
单应用独享负载均衡实例	<a href="#">应用独享负载均衡实例</a>
单应用中的不同实例分组通过URL实现流量分发	<a href="#">使用URL实现应用实例分组的流量分发</a>
多应用共享负载均衡实例时，使用端口实现流量分发	<a href="#">使用端口实现流量分发（共享负载均衡实例）</a>
多应用共享负载均衡实例时，使用域名或URL实现流量分发	<a href="#">使用域名或URL实现流量分发（共享负载均衡实例）</a>

## 应用场景

EDAS支持使用ALB和CLB添加负载均衡，使用复杂路由规则时，建议您使用ALB添加负载均衡。

- ALB：应用型负载均衡ALB，提供强大的应用层处理能力和丰富的高级路由功能，聚焦HTTP、HTTPS和QUIC应用层协议，是阿里云官方云原生Ingress网关。更多信息，请参见[ALB产品简介](#)。
- CLB：CLB通过设置虚拟服务地址，将添加的同一地域的多台ECS实例虚拟成一个高性能和高可用的后端服务池，并根据转发规则，将来自客户端的请求分发给后端服务器池中的ECS实例。CLB默认检查云服务器池中的ECS实例的健康状态，自动隔离异常状态的ECS实例，消除了单台ECS实例的单点故障，提高了应用的整体服务能力。此外，CLB还具备抗DDoS攻击的能力，增强了应用服务的防护能力。更多信息，请参见[CLB产品简介](#)。

### 8.3.2. 应用独享负载均衡实例

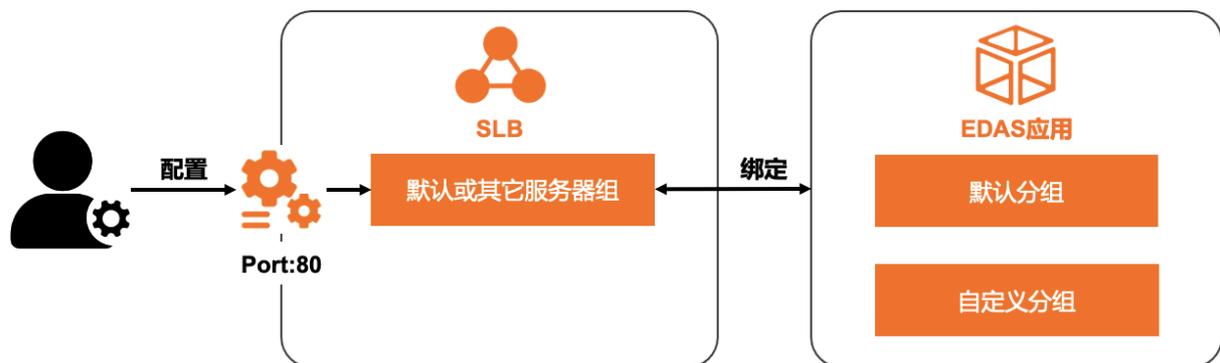
在ECS集群中，您为指定应用绑定一个负载均衡SLB实例，可以实现流量从SLB实例IP转发到该应用下的所有实例。

#### 前提条件

创建SLB实例。具体操作，请参见[创建实例](#)。

#### 背景信息

您有一个应用，包含多个ECS实例。为了保证应用实例的安全，您希望该应用对外提供一个公网IP地址供其他应用访问，那么您可以为该应用绑定一个SLB。



本文以公网负载均衡ALB实例为例，私网负载均衡实例的操作方式与公网一致，唯一需要注意的是验证的时候需要选择相应的网络环境。

#### 为应用绑定负载均衡实例

1. 登录[EDAS控制台](#)，在左侧导航栏单击应用列表。
2. 在应用列表页面的顶部菜单栏选择地域，并在页面上方选择微服务空间，在集群类型列表中选择ECS集

群，然后在应用列表中单击目标应用名称。

3. 在**基本信息**页签的**应用设置**区域，单击**负载均衡（公网）**右侧的**添加**。

**说明** 如果您已经配置过负载均衡实例，则在此处会显示负载均衡实例的IP和端口信息，您可单击**修改**进入配置页面修改负载均衡实例信息，或单击**解绑解除**与当前负载均衡实例的绑定。

4. 在**添加SLB与应用的绑定配置向导**中配置负载均衡实例。

- i. 在**选择负载均衡**页签中选择**负载均衡类型**为ALB或CLB，然后在**负载均衡（公网）**的下拉列表中选择您创建的SLB实例，然后单击**下一步**。
- ii. 在**选择配置监听**页签中配置监听端口，然后单击**下一步**。

当您**负载均衡类型**选择为ALB时，只支持选中已有的监听端口。**负载均衡类型**为CLB时，有以下两种情况可选：

- 如果想使用为选定SLB实例配置过的监听端口，选择**选择已有监听端口**，并在下方选择已有的监听端口。
- 如果没有为选定的SLB实例配置过监听端口，或想为当前应用配置一个新的监听端口，选择**添加新监听端口**，并为该SLB实例新建一个监听。

监听参数说明：

参数	说明
协议	支持TCP和HTTP协议，请根据实际需求选择。更多信息，请参见 <a href="#">监听概述</a> 。
前端端口号	负载均衡实例对外提供服务的端口。
(可选) 健康检查路径 (URI)	负载均衡检查后端应用存活状态的URI。默认为 <code>/_ehc.html</code> 。请根据应用输入正确的健康检查路径，具体操作，请参见 <a href="#">配置健康检查</a> 。健康检查路径 (URI) 参数在协议选择HTTP时出现。
后端端口号	应用的服务端口，不可设置。

iii. 在**配置服务器组**页签中配置SLB实例的后端服务器组，然后单击下一步。

 **说明** 应用级别独享SLB实例时，无需设置转发策略。

当您**负载均衡类型**选择为ALB时，只支持选择已有的虚拟服务器组。负载均衡类型为CLB时，有以下三种情况：

- 应用独享负载均衡实例时，推荐使用默认服务器组即可。

 **说明** 默认服务器组不支持配置转发策略。

- 如果想使用为选定的SLB实例创建过的虚拟服务器组，请执行以下操作：

a. 选择**现有虚拟服务器组**，并在下方选择已有的虚拟服务器组。

 **注意** 同一个虚拟服务器组只允许与一个应用或应用分组进行绑定。

b. 设置转发策略。

- 如果想使用该虚拟服务器组配置过的转发策略，选择**修改当前虚拟服务器组的转发策略**，并选中已有转发策略。
- 如果该虚拟服务器组没有配置过转发策略，或想要为其新增转发策略，选择**新增转发策略**，并在**转发策略**右侧的文本框中输入域名和路径规则。
- 如果没有为选定的SLB实例配置过虚拟服务器组，或想为当前应用配置一个新的虚拟服务器组，请执行以下操作：
  - a. 选择**新建虚拟服务器组**，并输入**虚拟服务器组名称**。
  - b. 选择**新增转发策略**，并在**转发策略**右侧的文本框中输入域名和路径规则。

iv. 在**确定变更**页签中查看设置的SLB实例信息，无误后，单击**确认变更**。

## 结果验证

1. 返回**基本信息**页签，在**应用设置**区域中**负载均衡（公网）**右侧复制负载均衡实例的 **IP:端口**，例如 `115.28.XX.XX:80`。  
如果**负载均衡（公网）**右侧未出现 **IP:端口**，则表示负载均衡实例绑定失败。请查看变更记录和变更详情，排查并修复失败原因。具体操作，请参见[查看应用变更](#)。
2. 在浏览器中访问复制的 **IP:端口**，查看是否能够成功访问应用。

## 8.3.3. 使用URL实现应用实例分组的流量分发

在ECS集群中，为应用绑定负载均衡实例后，可以为不同的应用实例分组配置不同的URL转发策略。负载均衡实例可以根据配置的转发策略将不同路径的请求转发到应用的不同实例分组中。本文介绍使用URL转发策略，实现单应用内不同实例分组的流量分发。

### 前提条件

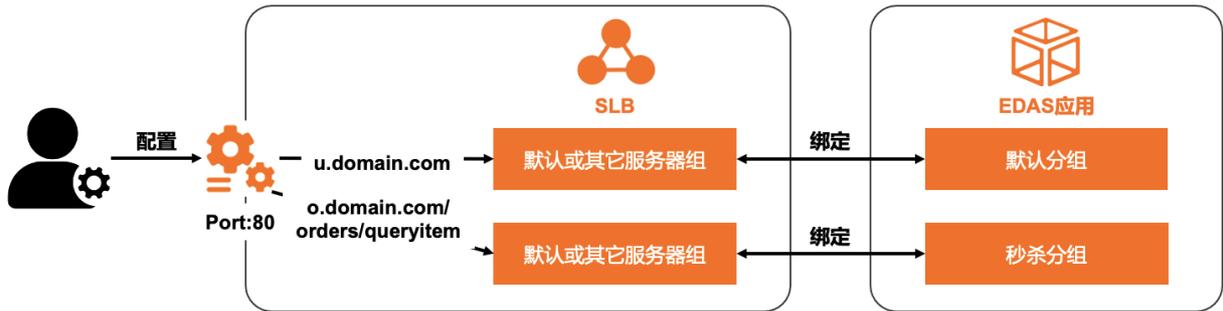
- 创建SLB实例。具体操作，请参见[创建实例](#)。
- 为应用创建不同实例分组。具体操作，请参见[创建分组](#)。
- 为应用绑定负载均衡实例。具体操作，请参见[应用独享负载均衡实例](#)。

### 背景信息

在ECS集群中，可以创建多个应用实例分组，在不同的应用实例分组上部署不同的应用部署包版本。应用绑定负载均衡实例后，会有以下情况：

- 如果没有为不同应用实例分组配置转发策略，那么应用的访问流量将会随机转发到各组的应用实例中。
- 如果为不同应用实例分组配置了不同的转发策略，那么可以实现将不同路径的请求转发到不同的应用实例分组的实例中。

例如秒杀场景下，查询订单系统中商品信息的URL (o.domain.com/orders/queryitem) 的访问量会明显高于其他服务的URL，可以将商品信息查询的服务部署到秒杀分组中，其他服务部署到默认分组中。在负载均衡实例中配置基于URL的转发策略，将商品信息查询的请求流量单独转发到秒杀分组，其他请求流量转发到默认分组。



### 为应用分组设置虚拟服务器组和转发策略

分别为场景示例中的两个分组在绑定的负载均衡实例中设置虚拟服务器组和转发策略。

**说明** 操作步骤几乎一致，唯一需要注意的是两个分组不能设置相同的虚拟服务器组。

1. 登录EDAS控制台。
2. 在左侧导航栏单击应用列表。
3. 在应用列表页面的顶部菜单栏选择地域，并在页面上方选择微服务空间，在集群类型列表中选择ECS集群，然后在应用列表中单击目标应用名称。
4. 在基本信息页面单击实例部署信息页签。
5. 在实例部署信息页签第一个分组（默认分组）右侧单击分组设置，在下列表中选择公网SLB信息。



6. 在分组设置对话框中选择虚拟服务器组，并设置转发策略，然后单击下一步。  
为应用分组配置虚拟服务器组（虚拟分组）和基于URL的转发策略包含以下两种方式：
  - o 如果想使用为选定的SLB实例创建过的虚拟服务器组，请执行以下操作：
    - a. 选择现有虚拟服务器组，并在下方选择已有的虚拟服务器组。

**注意** 同一个虚拟服务器组只允许与一个应用或应用分组进行绑定。

b. 设置转发策略。

- 如果想使用为该虚拟服务器组配置过的转发策略，选择修改当前虚拟服务器组的转发策略，并选中已有转发策略。
- 如果该虚拟服务器组没有配置过转发策略，或想要为其新增转发策略，选择新增转发策略，并在转发策略右侧的文本框中输入域名和路径规则。

The screenshot shows the 'Group Settings' dialog box. On the left, there are two sections: 'Current application bound SLB information' and 'Current group selected SLB information'. On the right, there are radio buttons for 'Existing virtual server group' (selected) and 'New virtual server group'. Below these is a table of existing virtual server groups:

Group Name/ID	Forwarding Strategy Bound to the Group	Reason for Not Being Selected
slb-vGroup1 rsp-xxxxxx	Will forward rules to this group: ID: rule-xxxxxx Rule Match: u.domain.com	N/A
slb-vCanary rsp-xxxxxx	Will forward rules to this group: ID: rule-xxxxxx Rule Match: /u	N/A
slb-vDefault rsp-xxxxxx	Will forward rules to this group: ID: rule-xxxxxx Rule Match: /o	N/A

Below the table, there are radio buttons for 'Modify current virtual server group forwarding strategy' (selected) and 'New forwarding strategy'. Underneath, there are checkboxes for 'Forwarding Strategy ID' and 'rule-xxxxxx', both checked, with a 'Forwarding Strategy' field containing 'URI Match: /u'. A 'Next Step' button is at the bottom right.

- 如果没有为选定的SLB实例配置过虚拟服务器组，或想为当前应用配置一个新的虚拟服务器组，请执行以下操作：
  - 选择新建虚拟服务器组，并输入虚拟服务器组名称。
  - 选择新增转发策略，并在转发策略右侧的文本框中输入域名和路径规则。

The screenshot shows the 'Group Settings' dialog box with 'New virtual server group' selected. The 'Virtual server group name' field contains 'slb-CanryGroup'. The 'New forwarding strategy' radio button is selected. The 'Forwarding strategy' field contains '/c'. The 'Next Step' button is at the bottom right.

7. 在应用SLB变更单页面单击确认变更。

结果验证

分别访问两个应用分组的URL，查看是否能够成功访问应用。

### 8.3.4. 使用端口实现流量分发（共享负载均衡实例）

为了节省成本，有可能将多个应用共享（复用）一个负载均衡实例，使用同一个公网IP地址和域名对外暴露，但仍希望通过转发策略将请求转发到对应的应用。本文介绍多应用共享负载均衡实例时，如何使用端口实现流量的转发。

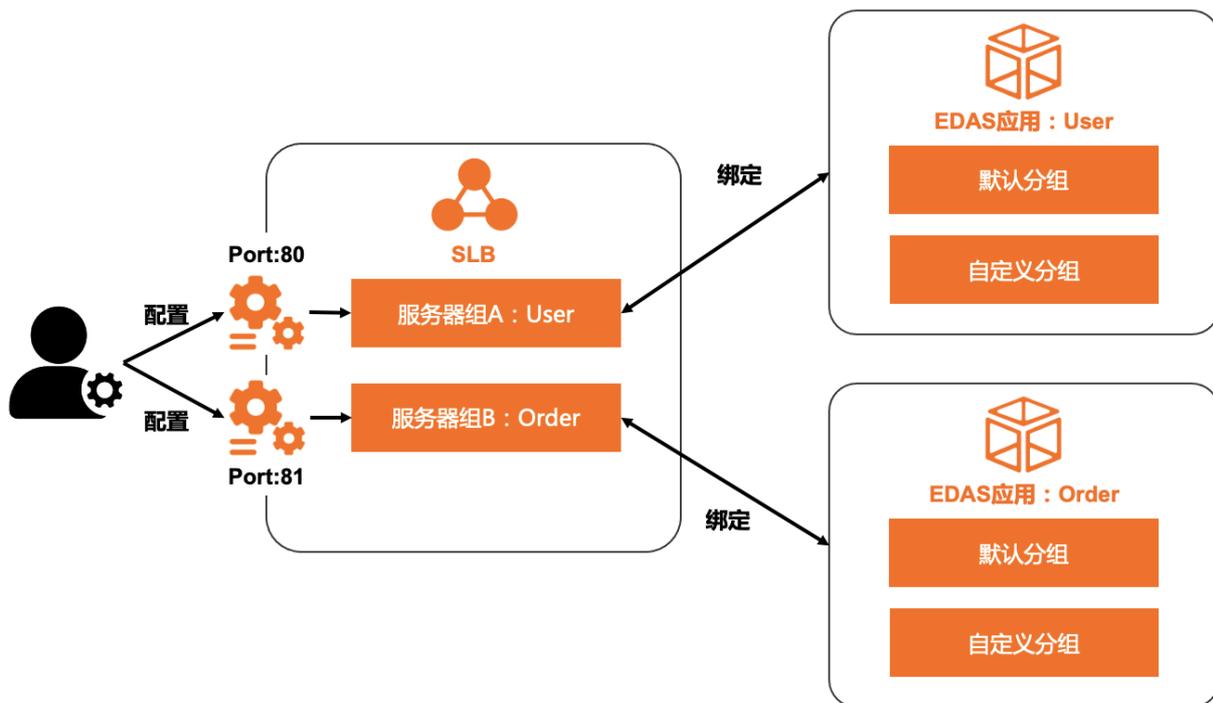
#### 前提条件

创建CLB实例。具体操作，请参见[创建实例](#)。

#### 背景信息

**说明** 在ECS集群中，仅HTTP监听协议支持流量分发。

例如有两个应用：一个提供订单查询的功能的应用User，一个提供用户登录的应用Order，这两个应用对外提供同一个公网IP地址，并且都绑定在一个域名上。您可以通过为两个应用绑定不同的监听端口来实现流量分发。



#### 为第一个应用绑定CLB并配置监听端口

在使用一个SLB实例的不同端口将请求转发至不同的应用的场景下，您需要先为一个应用绑定CLB实例，并设置监听端口。

1. 登录EDAS控制台，在左侧导航栏单击应用列表。
2. 在应用列表页面的顶部菜单栏选择地域，并在页面上方选择微服务空间，在集群类型列表中选择ECS集群，然后在应用列表中单击目标应用名称。
3. 在基本信息页签的应用设置区域，单击负载均衡（公网）右侧的添加。

**说明** 如果您已经配置过负载均衡实例，则在此处会显示负载均衡实例的IP和端口信息，您可单击修改进入配置页面修改负载均衡实例信息，或单击解绑解除与当前负载均衡实例的绑定。

4. 在添加SLB与应用的绑定配置向导中配置负载均衡实例。

- i. 在选择负载均衡页签中选择负载均衡类型为CLB，然后在负载均衡（公网）的下拉列表中选择您创建的SLB实例，然后单击下一步。
- ii. 在选择配置监听页签中配置监听端口，然后单击下一步。
  - 如果想使用为选定SLB实例配置过的监听端口，选择选择已有监听端口，并在下方选择已有的监听端口。
  - 如果没有为选定的SLB实例配置过监听端口，或想为当前应用配置一个新的监听端口，选择添加新监听端口，并为该SLB实例新建一个监听。

监听参数说明：

参数	说明
协议	<p>选择TCP。更多信息，请参见<a href="#">监听概述</a>。</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p> <b>说明</b> 您选择的 SLB 监听端口所使用的协议为 TCP，无法绑定转发策略。</p> </div>
前端端口号	<p>负载均衡实例对外提供服务的端口。</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p> <b>注意</b> 如果想添加的前端端口号已经被其他监听协议占用，请另外设置一个端口号，或者需登录<a href="#">SLB控制台</a>删掉该监听端口后，再在EDAS控制台上重新创建。</p> </div>
(可选) 健康检查路径 (URI)	<p>负载均衡检查后端应用存活状态的URI。默认为 <code>/_ehc.html</code>。请根据应用输入正确的健康检查路径，具体操作，请参见<a href="#">配置健康检查</a>。</p>
后端端口号	<p>应用的服务端口，不可设置。</p>

- iii. 在配置服务器组页签中配置SLB实例的虚拟服务器组，然后单击下一步。

参数	描述
默认服务器组	<p>默认服务器组不支持配置转发策略，所以在使用端口实现流量分发时，不可以选择默认服务器组，只能选择虚拟服务器组。</p>
现有虚拟服务器组	<p>如果想使用为选定的SLB实例创建过的虚拟服务器组，选择现有虚拟服务器组，并在下方选择已有的虚拟服务器组。</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p> <b>说明</b> 同一个虚拟服务器组只允许与一个应用或应用分组进行绑定。</p> </div>
新建虚拟服务器组	<p>如果没有为选定的SLB实例配置过虚拟服务器组，或想为当前应用配置一个新的虚拟服务器组，选择新建虚拟服务器组，并输入虚拟服务器组名称。</p>

- iv. 在确定变更SLB页签中查看设置的SLB实例信息，无误后，单击确认变更。
- 5. 验证当前应用的绑定的负载均衡实例IP和端口能否正常访问第一个应用，具体操作，请参见[结果验证](#)。

### 为第二个应用绑定CLB并配置监听端口

- 1. 登录EDAS控制台，在ECS集群应用的基本信息页签的应用设置区域，重复为第一个应用绑定CLB并配置监听端口的操作，为第二个应用绑定同一个SLB实例，并设置不同于第一个应用的监听端口。

## 8.3.5. 使用域名或URL实现流量分发（共享负载均衡实例）

为了节省成本，有可能将多个应用共享（复用）一个负载均衡实例，使用同一个公网IP地址和端口对外暴露，但仍希望通过转发策略将请求转发到对应的应用。本文介绍多应用共享负载均衡实例和端口时，如何使用域名实现流量的转发。

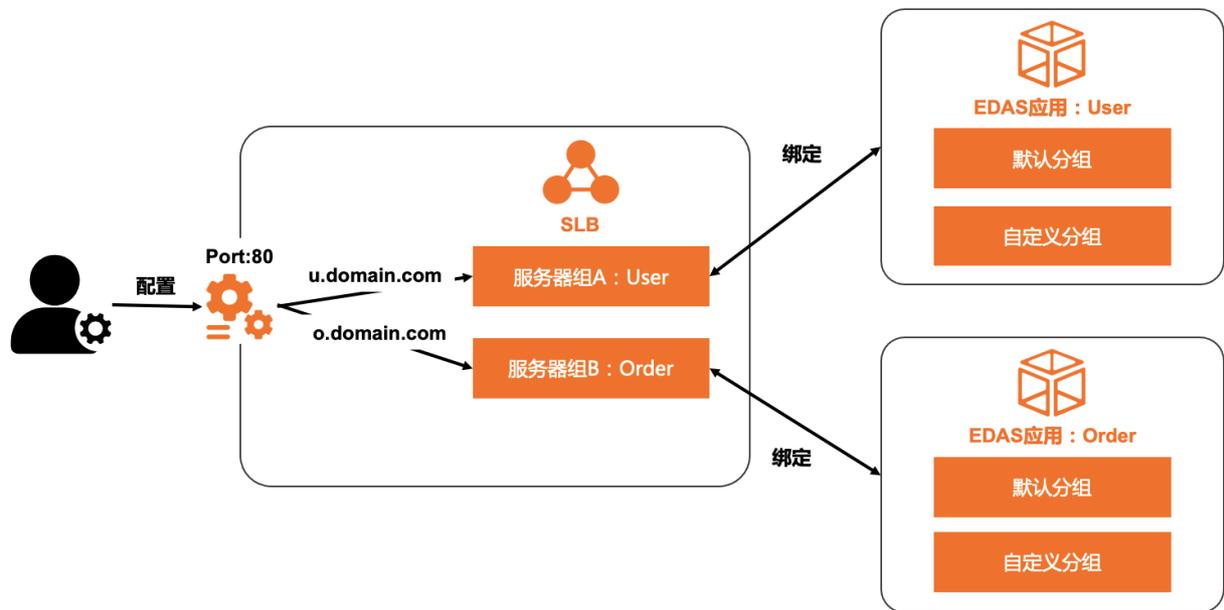
### 前提条件

创建SLB实例。具体操作，请参见[创建实例](#)。

### 背景信息

 **说明** 在ECS集群中，仅HTTP监听协议支持流量分发。

例如有两个应用：一个提供订单查询的功能的应用User，一个提供用户登录的应用Order，这两个应用对外提供同一个公网IP地址，且端口默认都是 80。您可以通过为两个应用设置不同的域名（*u.domain.com*为应用User的域名，*o.domain.com*为应用Order的域名），来实现流量分发。



### 为第一个应用绑定SLB并配置域名转发策略

为第一个应用绑定负载均衡实例，并配置监听端口、虚拟服务器组及域名转发策略。

- 1. 登录EDAS控制台，在左侧导航栏单击应用列表。
- 2. 在应用列表页面的顶部菜单栏选择地域，并在页面上方选择微服务空间，在集群类型列表中选择ECS集群，然后在应用列表中单击目标应用名称。

### 3. 在基本信息页签的应用设置区域，单击负载均衡（公网）右侧的添加。

 **说明** 如果您已经配置过负载均衡实例，则在此处会显示负载均衡实例的IP和端口信息，您可单击修改进入配置页面修改负载均衡实例信息，或单击解绑解除与当前负载均衡实例的绑定。

### 4. 在添加SLB与应用的绑定配置向导中配置负载均衡实例。

- i. 在选择负载均衡页签中选择负载均衡类型为ALB或CLB，然后在负载均衡（公网）的下拉列表中选择您创建的SLB实例，然后单击下一步。
- ii. 在选择配置监听页签中配置监听端口，然后单击下一步。

当您负载均衡类型选择为ALB时，只支持选中已有的监听端口。负载均衡类型为CLB时，有以下两种情况可选：

- 如果想使用为选定SLB实例配置过的监听端口，选择选择已有监听端口，并在下方选择已有的监听端口。
- 如果没有为选定的SLB实例配置过监听端口，或想为当前应用配置一个新的监听端口，选择添加新监听端口，并为该SLB实例新建一个监听。

监听参数说明：

参数	说明
协议	支持TCP和HTTP协议，请根据实际需求选择。更多信息，请参见 <a href="#">监听概述</a> 。
前端端口号	负载均衡实例对外提供服务的端口。   <b>注意</b> 如果想添加的前端端口号已经被其他监听协议占用，请另外设置一个端口号，或者需登录SLB控制台删掉该监听端口后，再在EDAS控制台上重新创建。
(可选) 健康检查路径 (URI)	负载均衡检查后端应用存活状态的URI。默认为 <code>/_e_hc.html</code> 。请根据应用输入正确的健康检查路径，具体操作，请参见 <a href="#">配置健康检查</a> 。
后端端口号	应用的服务端口，不可设置。

iii. 在配置服务器组页签中配置SLB实例的虚拟服务器组和基于域名的转发策略，然后单击下一步。

 **说明** 默认服务器组不支持配置转发策略，所以在使用域名实现流量分发时，不可以选择默认服务器组，只能选择虚拟服务器组。

为应用配置虚拟服务器组（虚拟分组）和基于域名的转发策略包含以下两种方式：

- 如果想使用为选定的SLB实例创建过的虚拟服务器组，请执行以下操作：
  - a. 选择**现有虚拟服务器组**，并在下方选择已有的虚拟服务器组。

 **注意** 同一个虚拟服务器组只允许与一个应用或应用分组进行绑定。

- b. 设置转发策略。
  - 如果想使用为该虚拟服务器组配置过的转发策略，选择**修改当前虚拟服务器组的转发策略**，并选中已有转发策略。
  - 如果该虚拟服务器组没有配置过转发策略，或想要为其新增转发策略，选择**新增转发策略**，并在**转发策略**右侧的文本框中输入域名和路径规则。
- 如果没有为选定的SLB实例配置过虚拟服务器组，或想为当前应用配置一个新的虚拟服务器组，请执行以下操作：
  - a. 选择**新建虚拟服务器组**，并输入虚拟服务器组名称。
  - b. 选择**新增转发策略**，并在**转发策略**右侧的文本框中输入域名和路径规则。

iv. 在**确定变更SLB**页签中查看设置的SLB实例信息，无误后，单击**确认变更**。

5. 验证当前应用通过设置的域名转发策略能否正常访问第一个应用，具体操作，请参见[结果验证](#)。

## 为第二个应用绑定SLB并配置域名转发策略

登录EDAS控制台，在ECS集群应用的基本信息页签的应用设置区域，重复为第一个应用绑定SLB并配置域名转发策略的操作，为第二个应用和第一个应用绑定相同的负载均衡实例和监听端口，设置不同于第一个应用的虚拟服务器组及域名转发策略。

## 8.4. 对ECS集群中的应用进行弹性伸缩

在分布式应用管理中，弹性伸缩是很重要的一个运维能力。弹性伸缩能够感知应用内各个实例的状态，并根据状态动态实现应用扩容、缩容。在保证服务质量的同时，提升应用的可用率。

### 弹性伸缩简介

互联网、游戏类等应用在促销活动期间容易出现突发性流量洪流，SLA和资源成本不易平衡，极易造成系统响应延迟、系统瘫痪等问题。EDAS继承阿里巴巴应对双11的流量洪流技术，提供秒级自动弹性功能，保证SLA的同时也节省机器保有成本。多适用于互联网、游戏以及社交平台等行业。

弹性伸缩适用于在ECS集群中创建的应用。在单实例环境中，弹性伸缩可确保始终有一个正在运行的实例。在流量变化很快的环境中，通过弹性伸缩配置要运行的实例数范围，EDAS将根据设置的负载规则按需添加或删除实例。

弹性伸缩功能根据应用实例的以下指标来判断并实现自动扩容或者缩容：

- CPU：CPU使用率，以百分比表示。
- RT：对请求作出的响应时间，单位为ms。
- Load：应用实例的负载大小，以正整数表示。

### 自动弹性伸缩

弹性伸缩包括自动扩容和自动缩容，可以分别配置扩容规则或者缩容规则。下面以配置扩容规则为例来展示操作步骤。

#### 说明

- 同时配置扩容和缩容规则的时候，缩容规则的指标不能大于扩容规则的指标，否则，在单击保存的时候会弹出错误提示。
- 若使用了弹性资源，缩容时会优先释放弹性资源提供的实例。

1. 登录EDAS控制台。
2. 在左侧导航栏中单击**应用列表**，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在应用列表页面单击具体的应用名称。
3. 在应用详情页面左侧的导航栏中，单击**弹性伸缩**。
4. 在**扩容规则**区域右上角打开开关启用扩容规则。
5. 配置扩容规则参数，然后在**弹性伸缩**页面左下角单击**保存**。
  - i. **配置触发指标**：设置CPU、RT和Load指标的阈值。当超过阈值时，触发扩容。
  - ii. **选择触发条件**：
    - **任一指标**：表示设定的指标中任意一个指标被触发都会引起扩容。
    - **所有指标**：表示设定所有指标必须全部被触发才能引发自动扩容操作。
  - iii. **持续时间超过**：指标持续被触发的时间，单位为分钟。表示在持续时间内，指标每分钟的平均值持续达到设置的阈值，就会引起扩容操作，您可根据集群服务能力的灵敏度酌情配置。
  - iv. **配置应用来源**：
    - **已有资源**：自动扩容时会从当前应用所在集群选择指定数量的闲置ECS实例扩容到该应用。

**说明** 当所在集群中现有的ECS实例数量不够，无法满足扩容需求时，EDAS会根据已有实例数量来进行扩容。

- **弹性资源**：基于现有实例规格或实例启动模版来代购实例，然后自动将代购的实例导入所在集群并用于应用扩容。

创建方式:  基于现有实例规格购买  基于实例启动模板购买

\* 模板主机: [Hqpt1r0k1vch1t0c0f0w0](#) 选择主机

2C4096MB / launch-ubuntu-20060818 / 释放模式

\* 登录密钥:  创建ECS密钥对 查看文档

\* 服务协议:  《云服务器 ECS 服务条款》 | 《镜像商品使用条款》

高级选项:

网络类型:  选择网络

多可用区扩缩容策略:  优先级策略  均衡分布策略

登录密码:

参数	描述
创建方式	<ul style="list-style-type: none"> <li>■ 基于现有实例规格购买：从所在集群内已有的实例规格中选择一个作为模板来代购实例。</li> <li>■ 基于实例启动模版购买：会基于您在ECS控制台创建的实例启动模版来代购实例。</li> </ul>
模板主机/启动模板	选择一个现有实例规格或者选择一个启动模板作为模板来代购实例。
登录密钥	当选择基于现有实例规格购买时，需要选择登录密钥。
服务协议	勾选《云服务器ECS服务条款》   《镜像商品使用条款》。
高级选项（可选）	<ul style="list-style-type: none"> <li>■ 基于现有实例规格购买：输入登录密码后，新创建的ECS实例将以此密钥对作为SSH登录认证方式。</li> <li>■ 基于实例启动模版购买：网络类型为您需要扩容的当前应用所在的网络，不可更改。如果当前网络为VPC网络，需要指定新创建实例连接的虚拟交换机；若指定多个虚拟交换机，EDAS将通过多可用区扩缩容策略来进行自动分配。</li> </ul>

- 已有资源优先：自动扩容时优先使用集群内空闲实例，如果集群内的空闲实例不足，则使用弹性伸缩功能为您代购实例。

v. 设置每次扩容的实例数：每次触发扩容操作后，自动增加的实例个数。

- vi. 设置分组最大实例数：当集群内服务器数量达到配置的最大实例数后，不再继续扩容，请根据您的资源限额配置。

## 查看弹性伸缩结果

设置了弹性伸缩规则后，如果发生了自动扩容或者自动缩容操作后，您可以通过以下方式来看伸缩结果：

- 在应用的基本信息页面中查看实例数量是否增加或者减少。
- 在应用详情页面的左侧导航栏单击变更记录。对于变更类型为应用扩容或应用缩容，且来源是auto\_scale的变更记录，在操作列单击查看进入变更详情页面查看变更明细。

变更类型	变更状态	模糊搜索	模糊搜索
应用扩容	执行成功	变更类型: 应用扩容	变更状态: 执行成功
集群扩容	执行成功	变更类型: 集群扩容	变更状态: 执行成功
应用扩容	执行成功	变更类型: 应用扩容	变更状态: 执行成功
集群扩容	执行成功	变更类型: 集群扩容	变更状态: 执行成功
启动应用	执行成功	变更类型: 启动应用	变更状态: 执行成功

## 更多信息

在ECS集群中的更多扩容操作，可参见以下文档：

- [扩容部署在ECS集群中的应用](#)
- [在ECS集群中代购ECS实例](#)
- [使用启动模板创建ECS实例](#)

## 8.5. 限流降级

EDAS已支持使用AHAS实现应用Spring Cloud应用、Dubbo应用和HSF应用的限流降级，支持实时查看限流降级详情和动态变更规则，全面保障您的应用的可用性。

### 前提条件

- 请确保您已开通AHAS，详情请参见[开通AHAS](#)。

说明 在使用AHAS时需要单独付费。

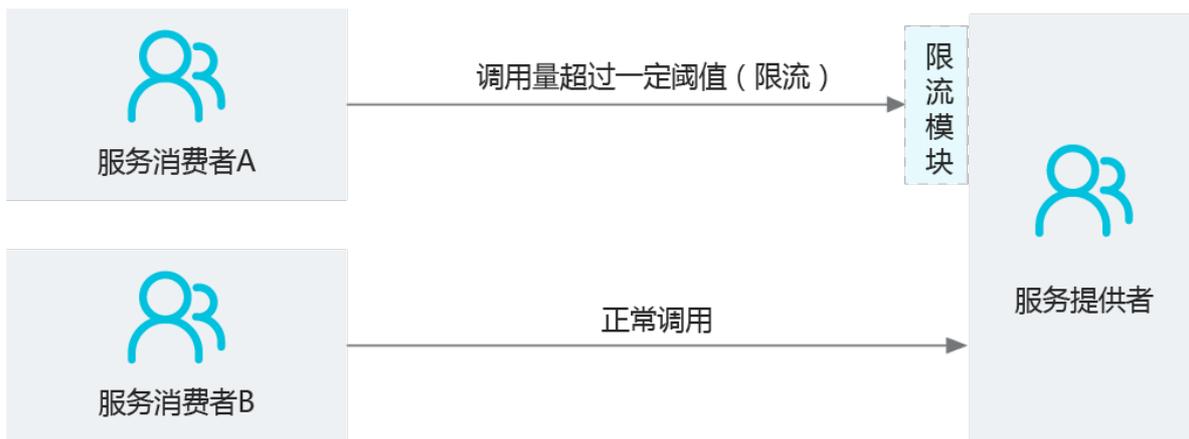
- 当前ECS应用需要您自行接入到AHAS。相关操作，请参见[AHAS应用防护支持的接入方式](#)。

### 限流降级简介

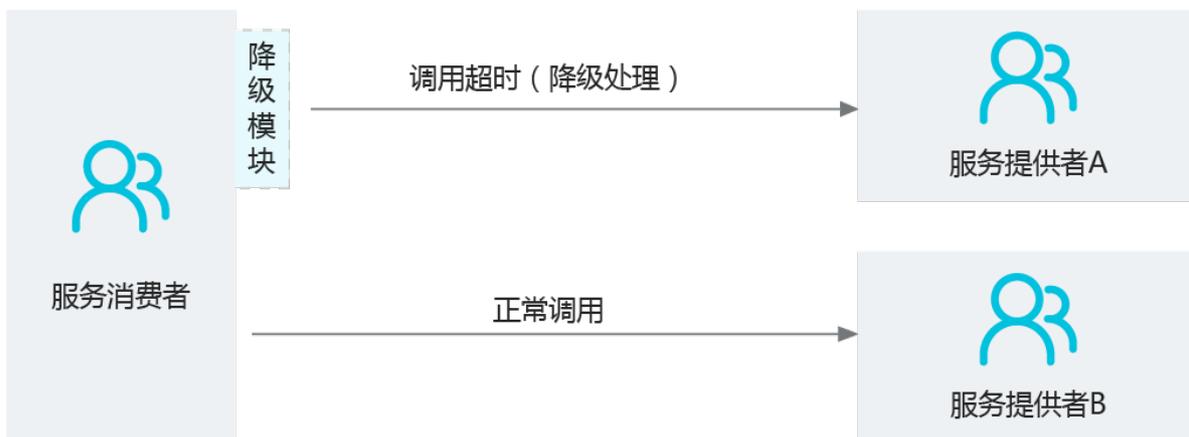
- 限流

限流可以理解为一个控制流量阈值或调节比例的功能。在前端网站面对大流量访问的时候，可以通过调节流量阈值来控制通过系统的最大流量值，防止大流量对后端核心系统造成破坏，导致服务不可用，保证系统安全可靠运行。

通过在服务提供者端配置限流模块代码，并在EDAS中配置限流策略后，使服务提供者具备限流功能。此时服务消费者去调用服务提供者时，所有的访问请求都会通过限流模块进行计算，若服务消费者调用量在一定时间内超过了预设阈值，则会触发限流策略，进行限流处理。



- 降级**  
 在EDAS中，降级通常用于对下游出现超时的非核心服务提供者进行低优先级调用，确保上游核心应用（服务消费者）不被影响。  
 通过在服务消费者端配置降级模块代码，并在EDAS中配置降级策略，使服务消费者具备降级功能。此时服务消费者去调用服务提供者时，若服务提供者服务响应时间超过了预设阈值，则会触发降级策略进行降级处理。



### 操作步骤

1. 登录EDAS控制台。
2. 在左侧导航栏中单击应用列表，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在应用列表页面单击具体的应用名称。
3. 在左侧导航栏中单击限流降级，然后在展开的菜单中单击具体的功能菜单。  
 AHAS提供的限流降级功能具体包括：
  - **接口详情**：展示该应用所有接口的通过QPS、限流QPS、异常QPS指标、RT、并发数据等，还可以在此页面为接口管理流控规则。更多信息，请参见[接口详情](#)。
  - **机器监控**：展示了所有节点的通过QPS、限流QPS、异常QPS、RT、并发等指标，还可以在此页面为接口管理流控规则。更多信息，请参见[机器监控](#)。
  - **规则设置**：展示了流控、隔离、降级等规则下包含的接口信息，还可以通过此页面管理各个接口的限流规则。更多信息，请参见[规则管理](#)。
  - **操作日志**：记录当前云账号及当前云账号创建的RAM用户对应用进行的操作，包括具体资源名、操作的内容、操作时间和操作人ID等，方便您追踪应用和资源的变更。
  - **权限管理**：如RAM用户需配置和推送AHAS中应用的流控降级规则，您需要对RAM用户单独授予读写

权限，从而严格管控权限，降低由于错误推送配置引发线上问题的可能性。

## 8.6. 升级或降级运行时环境

在ECS集群中部署应用后，在后续运行、维护过程中可以升级、降级应用的运行时环境（EDAS-Container），以满足不同的功能使用需求。本文介绍在EDAS中如何升、降级应用的运行时环境。

### 应用运行时环境

Spring Cloud、Dubbo和HSF应用都可以通过WAR或JAR包部署在ECS集群中，但所选择的应用运行时环境会有所不同。

- **使用WAR包部署的原生应用**：Spring Cloud、Dubbo、普通Java Web或Spring MVC应用，未使用过Pandora或Pandora Boot，通过WAR包部署时，需要选择Apache Tomcat相关版本的运行时环境。
- **使用JAR包部署的原生应用**：Spring Cloud、Dubbo、普通Java Web或Spring MVC应用，未使用过Pandora或Pandora Boot，通过JAR包部署时，需要选择标准Java应用运行时环境。
- **HSF应用**：包括使用Pandora Boot开发的Spring Cloud应用和由Dubbo转换为HSF的应用，通过WAR或JAR包部署时，需要选择EDAS-Container相关版本的运行时环境。

### 升级或降级运行时环境

1. 登录[EDAS控制台](#)。
2. 在左侧导航栏中单击**应用列表**，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在**应用列表**页面单击具体的应用名称。
3. 在**应用基本信息**页面右上角单击**运行时环境升降级**。
4. 在**运行时环境升降级**对话框中选择要升级或降级的运行时环境版本，并选择**Java环境**，然后单击**确定**。
  - 升、降级HSF应用的运行时环境。

### 运行时环境升降级

在进行此操作前，请妥善保存您登录ECS实例后手工修改的Tomcat配置文件，否则将被覆盖。通过EDAS控制台更新的Tomcat容器配置，将不受影响。  
此操作将重新部署实例上的Tomcat容器和应用程序。  
重新部署的应用程序，会和所在应用分组的应用版本保持一致。

\* Pandora版本: 4.0.0

\* Ali-Tomcat版本: 8.5.63

应用运行环境: EDAS-Container 4.0.0 [支持FatJar部署]

标签: 支持 TENGINE 支持 FATJAR 支持 RESTFUL 支持 EIP\_NODEPORT 支持 DPATH

发布日期: 2021-02-24 01:12:06

描述: 1、基于Dubbo 3.0重构的HSF插件的第一个Pandora Sar包版本(不建议现有生产HSF应用未经测试直接使用使用该Pandora Sar包版本); 2、如果您的应用是基于Dubbo 2.7之前(Alibaba Dubbo SDK)开发并且是使用EDAS Container方式部署，可能存在不兼容场景，如需升级至该Pandora Sar包版本请提工单。

分组: 所有分组

\* 每组批次: 1批

\* 分批方式: 自动

分批等待时间: 不等待

Java环境: Open JDK 8

确定 取消

- 升、降级Spring Cloud和Dubbo应用的运行时环境（此处以JAR包部署的应用为例）。

### 运行时环境升降级 ✕

在进行此操作前，请妥善保存您登录ECS实例后手工修改的Tomcat配置文件，否则将被覆盖。通过EDAS控制台更新的Tomcat容器配置，将不受影响。

**!** 此操作将重新部署实例上的Tomcat容器和应用程序。  
重新部署的应用程序，会和所在应用分组的应用版本保持一致。

\* 应用运行环境：

分组：

\* 每组批次：

\* 分批方式：

分批等待时间：

Java环境：

\* 版本： 用时间戳作为版本号

#### ? 说明

- 运行时环境升级或降级后无需重启应用。
- 运行时环境升级或降级后，可能由于功能兼容性影响应用的可用性，所以建议选择凌晨等业务量较小的时候升级。

5. 返回基本信息页面，在应用信息区域查看应用运行时环境版本。

- 如果升级成功，则会显示为目标版本。
- 如果升级失败，会有对应的变更记录，可参考[变更流程问题排查指南](#)定位、修复。如果需要，请联系EDAS技术支持人员。必要时，请将应用运行时环境变更到原有版本。

## 8.7. 变更记录

### 8.7.1. 查看应用变更

当您进行应用部署、启动、扩容、缩容等生命周期操作后，可以查看应用当前的变更状态，也可以通过变更记录页面查看该应用的历史变更记录。

#### 查看变更详情

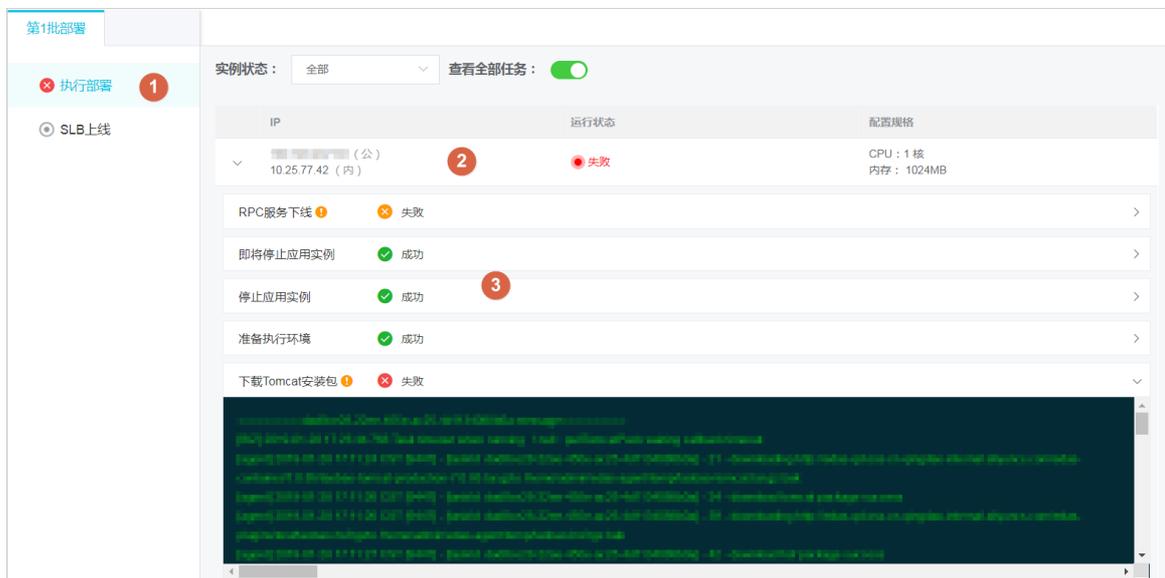
下面以一次部署应用为例说明如何查看应用变更。

1. 在执行完应用变更操作后，返回基本信息页。  
在基本信息页上方，会提示应用有变更流程正在执行，处于执行中状态。
2. 单击查看详情，进入变更详情页面，查看该应用的变更信息及实时状态。



- 变更概要信息：包括变更流程ID、执行状态、变更类型等信息。
- 变更流程执行信息：包含整个流程 (Process) 的每个阶段 (Stage) ，每个阶段 (Stage) 包含若干具体任务 (Task) 。
  - 如果是单实例任务，则在各阶段通过图标标识执行任务及结果。
  - 如果是多实例任务，则在每个阶段中基于实例分别展示执行任务及结果。

3. 在变更流程执行区域左侧的**第x批部署**下方单击某个阶段，查看该阶段中该实例的任务执行情况。



- 4. 单击具体任务，可以展开该任务执行的日志。  
对于执行失败的任务，系统会自动展开失败日志。

### 查看应用变更记录

1. 在基本信息页面左侧的导航栏中单击变更记录，查看该应用的所有变更操作的记录。
2. 在操作列单击查看，即可查看具体变更详情，及每一步操作的详细信息。

**说明** 变更操作的记录当前仅支持保留三个月。

### 8.7.2. 变更流程错误码说明

在EDAS控制台上进行ECS集群中的应用变更操作时，会生成变更记录。文本介绍了变更流程中可能出现的报错信息和错误码说明。

#### EDAS-10000

- 报错信息  
EDAS-10000 NO\_SPACE\_LEFT\_ON\_DISK

- **可能的原因**  
变更流程中如遇到此错误则代表磁盘空间已满。
- **处理办法**  
请登录ECS后删除日志。

## EDAS-10001

- **报错信息**  
EDAS-10001 STAR\_AGENT\_OFFLINE
- **可能的原因**  
StarAgent 不在线。
- **处理办法**  
请登录ECS执行命令 `sh /home/staragent/bin/staragentctl restart` 来重启相应的Agent。

## EDAS-10002

- **报错信息**  
EDAS-10002 Task timeout when running
- **可能的原因**  
任务执行超时。

## EDAS-10012

- **报错信息**  
EDAS-10012 UNZIP\_WAR\_FAILED
- **可能的原因**  
使用 `unzip` 命令解压WAR包失败，请检查您的WAR包格式是否可以正常解压成功。

## EDAS-10013

- **报错信息**  
EDAS-10013 JAR\_WAR\_FAILED
- **可能的原因**  
使用JAR命令解压WAR包失败，JDK未能正确安装或手工操作系统存在的是不包含JAR等命令的JDK版本。
- **处理办法**  
使用root身份登录报错应用的ECS实例，执行 `yum install -y java-1.8.0-openjdk-devel` 或者 `yum install -y java-1.7.0-openjdk-devel`（根据已有的JDK版本而定，同时需要保持联网）。

## EDAS-10014

- **报错信息**  
EDAS-10014 WAR\_URL\_403
- **可能的原因**  
下载WAR包失败，没有相应的访问权限。

## EDAS-10015

- **报错信息**  
EDAS-10015 WAR\_URL\_404
- **可能的原因**  
下载WAR包失败，没有找到对应的WAR包。

## EDAS-10016

- **报错信息**  
EDAS-10016 JAVA\_PROCESS\_ROOT
- **可能的原因**  
停止应用失败，应用进程是root身份启动的，EDAS无权操作。
- **处理办法**  
请使用root身份手动停止应用进程后重试。

## EDAS-10017

- **报错信息**  
EDAS-10017 HOME\_ADMIN\_FILE\_ROOT
- **可能的原因**  
启动应用失败，`tomcat`目录或者应用目录下存在ROOT权限文件或者目录。
- **处理办法**  
请检查 `/home/admin/app` 目录和 `/home/admin/${tomcat-path}` 目录下是否有root权限的文件或者目录，手动恢复成admin权限后重试。

## EDAS-10020

- **报错信息**  
EDAS-10020 SLB\_NOT\_ENOUGH\_BALANCE
- **可能的原因**  
绑定SLB失败，您的账户需要有100元余额才能正常购买SLB实例。

## EDAS-10021

- **报错信息**  
EDAS-10021 SLB\_BIND\_ALREADY\_HAVE\_SERVICE
- **可能的原因**  
绑定SLB失败，此应用已经绑定了一个SLB。
- **处理办法**  
请在控制台上先解绑原有的SLB，再重新绑定目标SLB。

## EDAS-10022

- **报错信息**  
EDAS-10022 K8s\_READINESS\_ERROR
- **可能的原因**  
应用启动失败，规定时间内READINESS检查失败。
- **处理办法**  
请检查相关配置是否正确。

## EDAS-10023

- **报错信息**  
EDAS-10023 K8s\_BACKOFF\_RESTART。
- **可能的原因**  
应用启动失败。
- **处理办法**  
请检查是否镜像有否有问题。

## EDAS-10024

- 报错信息

EDAS-10024 K8s\_LIVENESS\_ERROR

- 可能的原因

应用启动失败，规定时间内Liveness检查失败。

- 处理办法

请检查相关配置是否正确。

## EDAS-10025

- 报错信息

EDAS-10025 PORT\_CHECK\_TIMEOUT

- 可能的原因

应用启动时端口检查超时失败，超时时间通常是5分钟。即5分钟内探测端口没有正常打开。

- 处理办法

登录ECS检查应用端口是否正常打开，如未正常打开请查看应用进程的启动日志是否报错，并根据报错进行处理。

## EDAS-10026

- 报错信息

EDAS-10026 URL\_CHECK\_TIMEOUT

- 可能的原因

开启了URL健康检查的应用，应用启动时检查URL是否可以访问超时失败，超时时间通常是30分钟。当且尽当 `curl $URL` 返回HTTP状态码为200时才会认为URL健康检查通过。

- 处理办法

登录ECS执行命令 `curl -v $URL` 检查应用配置的URL是否可以正常访问。如不能正常访问，通常需要查看应用日志来确定应用进程是否启动失败。

## EDAS-10027

- 报错信息

EDAS-10027 STARAGENT\_CMD\_EXECUTE\_TIMEOUT

- 可能的原因

发布流程中的命令执行超时，通常情况下超时时间是60秒。即一个发布命令在60秒内不能执行结束就会导致这个错误发生。

- 处理办法

- 在EDAS控制台重试出错的任务。
- 如果重试还无法解决，请提交[工单](#)寻求EDAS支持帮助。

## EDAS-10028

- 报错信息

EDAS-10028 PULL\_WAR\_TIMEOUT

- 可能的原因

采用自定义WAR/JAR URL地址部署时，如果拉取WAR/JAR包超时，则会出现这个错误。通常超时时间为5分钟。

- 处理办法

确认URL地址正确无误，登录ECS执行 `wget` 命令手动下载URL检查是否能在5分钟内完成下载。

## 8.8. 日志

## 8.8.1. 日志简介

在应用的运维过程中可能会遇到各种问题，您可以通过日志定位、诊断。EDAS 提供各种类型的日志，您可以在线查看，也可以下载。

### 日志类型

EDAS 提供以下 4 种日志：

- EDAS 系统日志（通用）：您可以通过该日志定位、诊断 EDAS 系统问题。EDAS 系统日志为默认收藏目录，包含两个目录：
  - `/home/admin/edas-agent/logs/`：当遇到 EDAS Agent 异常时，可以查看该目录下的日志进行定位、诊断。
  - `/home/admin/edas-container/logs/`：当遇到容器退出或者应用异常时，可以查看该目录下的日志进行定位、诊断。
- 业务日志（通用）：您可以根据您的实际业务需求，添加相关的日志目录及该目录下的日志文件，以便定位、诊断业务问题。详情请参见[收藏日志目录](#)。
- 日志框架配置的文件（通用）：在应用运行时，从进程中自动提取出来的、符合日志框架（如 JDKLog、Log4j 等）的 EDAS 系统和业务日志文件。
- 实时日志（适用于 Kubernetes 集群）：提供 Docker 启动、运行时的标准输出日志，帮助您定位 Pod 问题。

### 查看日志

当您的应用部署到 EDAS 的 ECS 集群或容器服务 Kubernetes 集群后，您可以在 EDAS 控制台查看日志，而无需登录到具体 ECS，更加方便地定位、诊断应用运行时遇到的问题。

EDAS 提供应用和 ECS 实例两种级别的日志查看方式，其中，应用级日志还支持分布式搜索，详情请参见：

- [查看实例日志（通用）](#)
- [查看应用日志并进行分布式搜索（通用）](#)

### 下载日志

除了在线查看日志外，您还可以将日志下载到本地进行查看或者将日志文件作为应用分析、诊断的素材使用。

## 8.8.2. 日志管理

当您的应用出现异常情况的时候，可以通过查看实例和应用级别的日志来排查问题。EDAS 提供了日志目录收藏、日志查看、日志搜索和日志下线的功能。

### 收藏日志目录

日志目录页面包含 EDAS 相关的默认日志目录，您可以收藏（添加）应用日志目录。收藏日志之后即可查看该目录下的实例日志，详情请参见[查看实例日志](#)。

在收藏日志目录时还可以将该目录添加到[日志服务](#)，以便在[日志搜索](#)页面查看和搜索该目录下的应用日志。

 **说明** 收藏目录和取消收藏仅对日志目录可用。

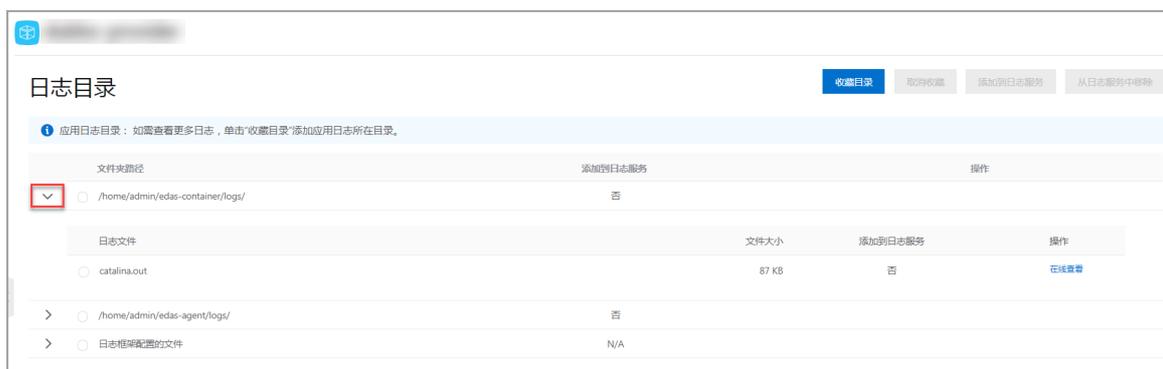
1. 登录[EDAS控制台](#)。
2. 在左侧导航栏中单击[应用列表](#)，在顶部菜单栏选择地域并在页面上方选择微服务空间，然后在[应用列表](#)页面单击具体的应用名称。

3. 在应用详情页面左侧的导航栏中选择日志管理 > 日志目录，然后在日志目录页面单击收藏目录。
4. 在收藏目录对话框中输入应用日志目录，选择是否将该目录添加到日志服务，然后单击添加。输入应用日志目录时，请遵循以下要求：
  - 此目录必须在 /home/admin 目录下。
  - 完整目录中必须包含 log 或者 logs。
  - 目录最后必须以斜杠 / 结尾，表示添加的是一个文件夹。

## 查看实例日志

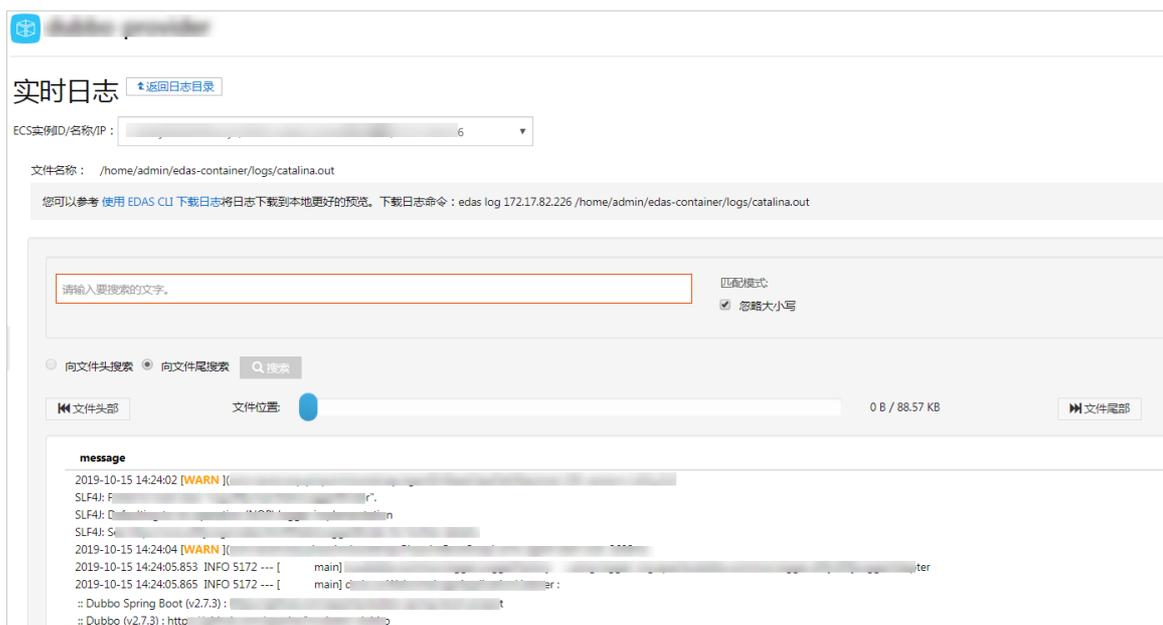
在收藏了应用日志后，可以查看实例和应用级别的日志。

1. 在应用详情页面左侧导航栏中选择日志管理 > 日志目录。
2. 在日志目录页面单击日志目录左侧的展开按钮 >，然后在展开的日志文件列表的操作列单击在线查看。



**说明** 日志框架配置的文件下的日志文件除了可以查看日志详细信息，还可以修改该日志文件的级别。日志级别由低到高依次为：TRACE、DEBUG、INFO、WARN和ERROR。修改后，系统会将所设置的级别及更高级别的日志显示在页面下方。

3. 在实时日志页面顶部单击ECS实例ID/名称/IP右侧的下拉箭头，选择实例，查看该实例中的日志详细信息。



在页面右下角单击开启实时追加，可以一直加载文件的最新追加内容（类似于 `tailf` 命令的效

果)。

### 将日志目录或文件添加到日志服务

日志服务提供了查看和搜索应用日志的功能。将日志目录或文件添加到日志服务后，即可查看和搜索应用日志。

在将日志目录或文件添加到日志服务时，还可以指定项目 (Project) 和日志库 (Logstore)。

- 如果您已经在日志服务中创建了Project和Logstore，可以使用已有的Project和Logstore。
- 如果您未在日志服务中创建Project和Logstore，可以新建Project和Logstore。EDAS将按 `edas-logstore-  
-<日志文件的绝对路径>` 格式为您创建Logstore，如 `a-homeadminlogs`。

**说明** 如果您为RAM用户，请先由云账号为RAM用户授权，详情请参见[为RAM用户授予日志服务权限](#)。

1. 在应用详情页面左侧导航栏中选择日志管理 > 日志目录。
2. 在日志目录页面选择日志目录或文件，然后单击添加到日志服务。  
如果您将日志目录添加到日志服务，则该目录下的日志文件默认都添加到日志服务中。您只能基于该日志目录进行分布式搜索，而不能再对该目录下的具体日志文件进行分布式搜索。如果您想对具体日志文件进行分布式搜索，建议不要将整个目录添加到日志服务中。
3. 在添加到日志服务对话框中确认日志目录并设置参数，然后单击确定。

添加到日志服务
✕

\* 应用日志目录

来源方式  新建  使用已有

Project

\* Logstore

\* 日志路径 ⓘ  /\*\*/

- 应用日志目录：选择的日志目录，不可修改。
- 来源方式选择新建：
  - Project：输入Project名称，推荐格式为 `edas-project-<region-id-timestamp>`，如 `edas-pr  
oject-cn-qingdao-`。
  - Logstore：EDAS将自动按 `edas-logstore-<日志文件的绝对路径>` 命名方式为您创建Logstore，如 `a-homeadminlogs`。
  - 日志路径：如果您需要当前目录下的部分日志文件，可以输入文件名的关键字或完整文件名。文件名称可以是完整名，也支持通配符模式匹配。
- 来源方式选择使用已有：

- **Project**：在下拉列表中选择已创建的Project。
- **Logstore**：在下拉列表中选择已创建的Logstore。
- **日志路径**：如果您需要当前目录下部分日志文件，那么可以输入文件名的关键字或完整文件名。文件名称可以是完整名，也支持通配符模式匹配。

## 查看应用日志并进行分布式搜索

已收藏过该应用日志目录，并将该目录或文件添加到日志服务后，可以查看应用日志并进行分布式搜索。

**说明** 如果您为RAM用户，请先由云账号为RAM用户授权，详情请参见[为RAM用户授予日志服务权限](#)。

1. 在应用详情页面左侧导航栏中选择**日志管理 > 日志搜索**。  
如果已经将某个日志文件添加到日志服务，也可以在**日志目录**页面日志文件列表的操作列单击**搜索**。
2. 在**日志搜索**页面日志文件右侧单击下拉箭头，在下拉列表中选择日志目录或日志文件。
3. 查看应用日志。  
查看应用日志包括[日志聚类](#)、[LiveTail](#)和[查询和分析](#)。
4. 输入关键字，对应用日志内容进行分布式搜索。

## 为RAM用户授予日志服务权限

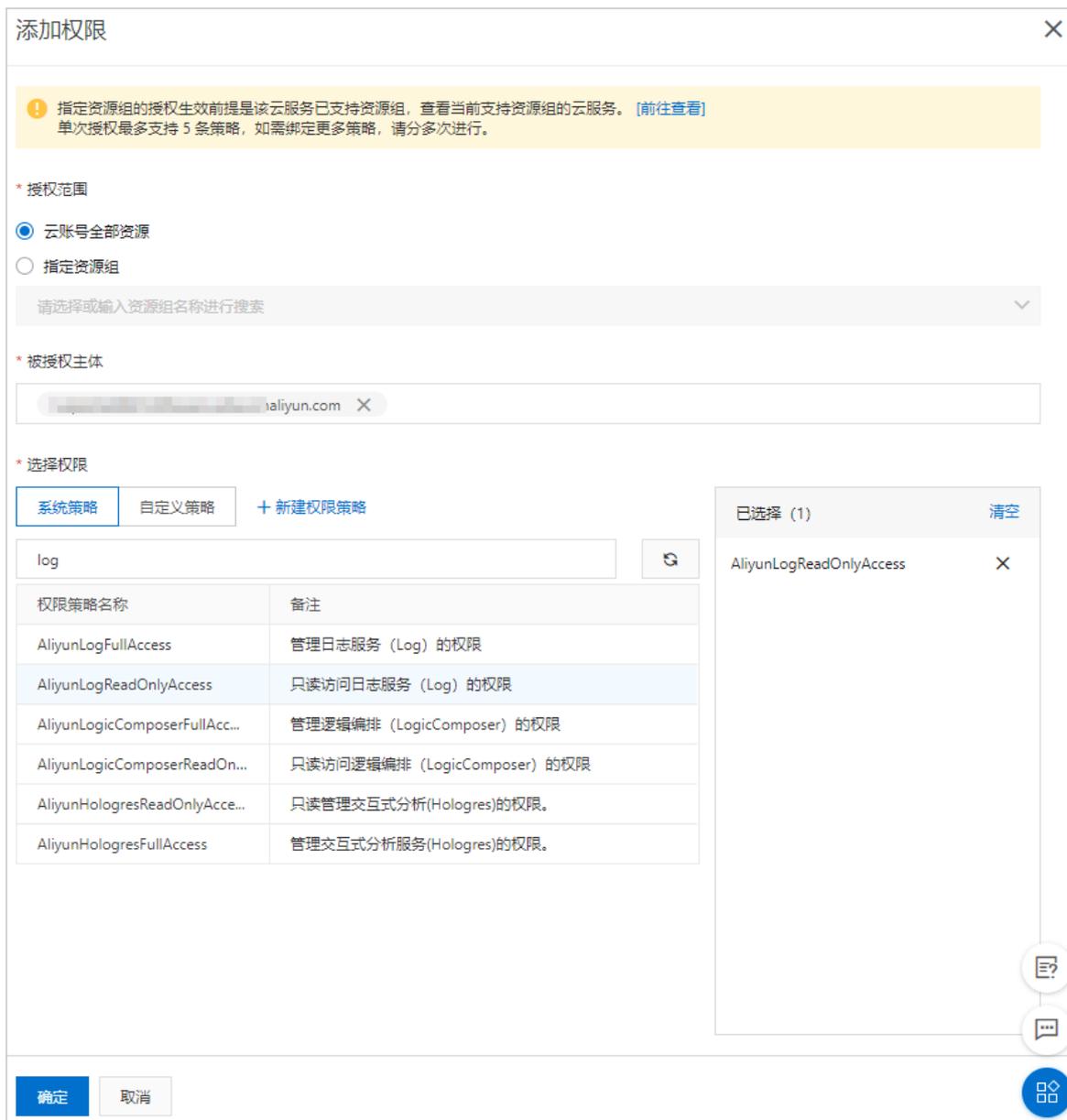
如果您是RAM用户，需要通过子账号使用日志服务（包括将日志目录或文件添加到日志服务和查看应用日志和分布式搜索），还需要由云账号（主账号）在访问控制RAM中为RAM用户授权。操作步骤如下：

1. 使用阿里云账号登录[RAM访问控制控制台](#)。
2. 在左侧导航栏中单击**用户**，然后在**用户**页面的用户列表中找到您要授权的RAM用户，如 *doctest*，在操作列单击**添加权限**。



3. 在**添加权限**页面选择权限下面的系统权限策略右侧的文本框输入 *log*，单击 **AliyunLogReadOnlyAccess** 将该权限添加到右侧列表中，单击**确定**。

**说明** 被授权主体会默认加载，无需设置。如果您需要同时给多个RAM用户授权，在被授权主体下面的文本框通过关键字搜索并添加。



4. 在授权结果页面查看RAM用户和被授权的权限，确认无误后，单击完成。

### 将指定日志目录或文件从日志服务中移除

将指定日志目录或文件从日志服务中移除。移除仅仅是从日志服务中移除，实际的日志目录和文件仍然存在，并未删除。包含两个动作：

- 将日志目录或文件从日志服务的采集规则中移除：移除后，日志服务将不再采集新的数据。
- 将日志目录或文件从日志服务的采集规则中移除，并删除日志服务中的关联对象：移除后，日志服务将不再采集新的数据。并且当关联对象（Project和Logstore）不再被其它应用使用时，将会被从日志服务中移除。该操作将会导致Logstore中的数据被删除且无法回滚。

**说明** 默认日志目录不可以从日志服务中移除。

1. 在日志目录页面选中某个日志目录或文件，然后单击从日志服务中移除。
2. 在从日志服务中移除对话框中确认要移除的日志目录或文件，并根据实际需求选择是否选中删除日志服务中的关联对象，然后单击确定。

删除日志服务中的关联对象将删除关联对象 (Project和Logstore) , 即Logstore中的数据被删除。

## 取消收藏日志目录

取消收藏日志目录即移除指定的应用日志目录。移除后, 该日志目录将不再显示在日志目录页面上, 也不能再查看该目录下的实例日志, 但并不会删除实际的日志目录及文件。

 说明 默认目录也可以取消收藏。

在取消收藏日志目录时, 还可以选择**并从日志服务中移除**。

- 如果未从日志服务中移除, 则可以查看该目录下的原有应用日志。
- 如果从日志服务中删除, 则不能再查看该目录下的应用日志。
- 在日志目录页面选中某个日志目录, 然后单击**取消收藏**。
- 在**取消收藏日志目录**对话框中, 确认要取消收藏的日志目录, 选择是否**并从日志服务中移除**, 单击**确定**。

## 9. ECS集群使用常见问题

### 9.1. 资源管理

#### 9.1.1. Agent 安装之后没有上报，界面未显示 Agent 的版本号

##### Condition

Agent 安装之后没有上报，界面未显示 Agent 的版本号。

##### Remedy

##### 操作步骤

1. 查看 `/home/admin/edas-agent/logs/std.log`，如果出现 `java not found` 或其他异常，请执行 `java --version` 确认是否是 1.7 版本。如果是 Java1.5，请先使用 `rpm -e` 对应安装 rpm 名称命令移除之后，再重新安装 Agent。

#### 9.1.2. Agent安装后控制状态“未知”或“异常”

##### Condition

Agent安装后控制状态“未知”或“异常”。

##### Remedy

##### 操作步骤

1. 请查看ECS上 `/home/admin/edas-agent/logs` 目录下的 `std.log` 和 `agent.log` 日志文件。
  - `std.log` 是 Agent 安装过程中的日志。
  - `agent.log` 是 Agent 的运行日志。
2. 根据错误排查问题。
  - 一般有如下原因：
    - 如果日志中有 `Permission denied` 或 `Not such file` 字样，这可能是没有相关文件目录权限导致，需要确认 admin 账号是否具有 `/home/admin` 目录下所有文件的权限，然后重新安装 Agent。
    - 检查 ECS 实例名和 `/etc/hosts` 文件中的是否一致，如果不一致，修改后重启 Agent 即可。

```
/home/admin/edas-agent/bin/shutdown.sh
/home/admin/edas-agent/bin/startup.sh
```

#### 9.1.3. 什么是 Docker 实例？和 ECS 独占实例有何区别？

##### Remedy

##### 操作步骤

1. EDAS 的应用部署有两种类型：

- EDAS 的应用部署有两种类型：  
在一台独立的 ECS 机器上，仅允许部署单独一个应用。
- Docker 实例  
使用 Docker 技术，EDAS 实现了在单个 ECS 实例上部署多个应用。在一台 ECS 机器上创建多个 Docker 实例，每一个 Docker 实例中可部署一个应用。

 **注意** 单个应用在同一 ECS 上只能部署一个实例。

## 9.1.4. 购买了 EDAS 之后，是否需要另行购买 ECS？

EDAS 是分布式应用的构建和管理平台，本身不包含 ECS 资源，您需要另行购买 ECS。购买之后，EDAS 能够自动同步您账号下的 ECS 资源并进行使用和管理。

## 9.1.5. EDAS的Java版本是哪个版本，我能够选择吗？

EDAS提供7, 8两个版本的Java，默认使用的是Java 7，您也可以在安装EDAS Agent过程中，选择其他Java版本，例如Java 8。命令语法如下：

```
install.sh -ak -sk [-java <7(default)|8>]
```

- 指定使用JDK 7

```
wget -q -O /root/install.sh http://edas-sh.oss-cn-shanghai-internal.aliyuncs.com/install.sh "云账号或RAM用户的AccessKey ID" -sk"云账号或RAM用户的AccessKey Secret" -t install.nonce.at.1499069083770 -Java 7
```

- 指定使用JDK 8

```
wget -q -O /root/install.sh http://edas-sh.oss-cn-shanghai-internal.aliyuncs.com/install.shak "云账号或RAM用户的AccessKey ID" -sk"云账号或RAM用户的AccessKey Secret" -t install.nonce.at.1499069083770 -Java 8
```

## 9.1.6. 如果 Agent 的心跳进程停了，会有什么样的后果？

如果相应的 ECS 上没有安装应用程序，则不会有任何的影响。如果已经安装了应用，则在应用的机器列表中（在应用管理中选择对应的应用进入基本信息之后，屏幕下方会列出该应用下的所部署的 ECS 列表），相应的 ECS 的实时状态会变成 Agent 异常；且此时，任何针对该 ECS 的部署、启动、停止等命令均是无效的。

登录到该机器执行，`sudo -u admin /home/admin/edas-agent/bin/startup.sh` 将 Agent 启动。排查 Agent 异常退出的原因：

- 查看 `/home/admin/edas-agent/logs/agent.log` 中是否有错误信息。
- 确定系统内存是否足够，如果系统内存紧张，会导致系统 oom kill。具体请搜索：Linux oom kill。如果出现 oom kill，建议查看系统内存使用情况，调整内存分配。

## 9.1.7. 如果 Ali-Tomcat 突然容器退出，该如何处理

### Condition

Ali-Tomcat 突然容器退出。

### Remedy

### 操作步骤

1. 登录 EDAS 控制台启动对应的应用，排查 Ali-Tomcat 异常退出的原因。
  - 查看 /home/admin/tomcat（安装目录）/logs/catalina.out 中是否有错误信息
  - 确定系统内存是否足够，如果系统内存紧张，会导致系统 oom kill。具体请搜索：linux oom kill。如果出现 oom kill，建议查看系统内存使用情况，调整内存分配策略

## 9.1.8. 资源未及时同步怎么办？

### Remedy

#### 操作步骤

1. 请在控制台单击资源管理 > ECS > 同步 ECS 按钮进行同步。
2. 同步之后还是无数据，请访问 <https://ak-console.aliyun.com>。
3. ak,sk 正常则需要联系技术支持人员进行排查。

## 9.1.9. 安装 Agent 的过程中发现卸载 Java

### 问

为什么安装 EDAS Agent 的过程中会将原有安装的 Java 卸载掉？

### 答

在安装 EDAS Agent 的过程中，除了安装 EDAS 运行环境需要的 Agent 之外，还会安装其依赖环境，如：JDK，由于目前 EDAS 支持 jdk1.7 与 jdk 1.8 两个 Java 版本，在安装 EDAS Agent 的过程中，提供了命令行参数选择；即如果指定了 `-java 7` 则安装 JDK 1.7，`-java 8` 则安装 JDK 1.8。默认会选择 1.7 进行安装。

在第一次安装的过程中，将会全部下载并初始化一遍，如果被安装的机器上已经安装有相应版本的 JDK，则不会重新安装，否则，会将之前的版本卸载之后重新下载相应版本安装。

之后的升级过程（第二次执行 `install.sh` 脚本），分为如下情况进行讨论：

- 执行参数加入了 `-force` 选项，则会将所有的组件卸载重新安装，包括 JDK。
- 执行参数没有加 `-force` 选项，默认只会重新安装 EDAS Agent 组件，然而，针对 JDK 的 `-java` 选项，又有两种情况需要处理：
  - 如果指定的 JDK 版本与之前安装的版本一致，则不会执行任何操作。
  - 如果指定的 JDK 版本与安装版本不一致，则会重新卸载安装并指定版本。

## 9.1.10. 日志采集器相关问题

1. 在一个 VPC 中，是否可以有多台机器安装日志采集器？  
答：推荐使用多个采集器托管一个 VPC 集群内的机器，尤其是在大集群中。
2. 安装完采集器之后，采集器所占用的带宽是否计费？  
答：由于我们使用的是内网带宽，阿里云不会把这一部分流量计入账单的。
3. 选择安装采集器的机器是否必须安装 EDAS Agent？  
答：不是必须的，但是我们推荐在安装了 EDAS Agent 的机器上安装采集器。
4. 如果不安装日志采集器，会有什么样的影响？  
答：体现在 EDAS 的服务中，如果获取不到数据，所有和视图以及调用链相关的服务均变得不可用；这部分功能包括监控（基础监控、服务监控）、报警、调用链和扩容缩容等。

5. 在安装过程的最后阶段，老是出现 `setup sproxy failed` 字样，是否是真的失败了？  
答：由于安装脚本只是简单的检查连接是否建立，而从模块启动到建立需要一个较长的时延，所以会出现这个问题。建议在安装完成一分钟之后，手动检测端口（8002）是否建立成功。
6. 支持日志采集器的系统有哪些？  
答：目前我们仅支持Cent OS 7.0以上的64位系统和Ali-Linux 15.1以上的64位系统。
7. VPC 环境调用链查询不到，监控无数据。  
答：
  - 调用链和监控数据需要有访问请求流量才会产生，若无客户访问网站，则不会有曲线。
  - 确定机器的 8182 端口打开。执行 `netstat -ant|grep 8182` 查看是否端口开放。

## 9.2. 应用管理

### 9.2.1. 重启服务器后 EDAS Agent 是否能自动重启？

是的，重启服务器后 EDAS Agent、应用都会自动重启。

### 9.2.2. EDAS Agent 为什么无法启动？

EDAS Agent 无法启动的话请参照以下步骤进行排查。

1. 在服务器上使用 `ping` 命令检查 EDAS Console 服务器是否可达：

```
ping edas-internal.console.aliyun.com
```

。
2. 请检查您的安全令牌文件是否被正确设置：

```
cat /home/admin/.spas_key/default
```

。

### 9.2.3. EDAS 是否支持在同一个实例上安装多个应用？

在 EDAS 产品中，应用的部署有两种类型。

- ECS 独占实例：在一台独立的 ECS 实例上，仅允许部署单独一个应用。
- Docker 实例：EDAS 使用 Docker 技术，在一台独立的 ECS 机器上创建多个 Docker 实例，允许在每一个 Docker 实例上部署一个应用。

### 9.2.4. 为什么应用运行时进程突然消失了？

这个问题通常由操作系统物理内存耗尽或应用运行的Java虚拟机进程Crash导致，本文以Linux操作系统为例说明如何解决。

#### 操作系统物理内存耗尽，触发操作系统OOM Killer

当操作系统物理内存和交换空间不够用时，操作系统的OOM Killer机制（默认打开）就会选择性地杀死进程，那么它是怎样知道要先杀死哪个进程呢？其实Linux的每个进程都有一个`oom_score`（位于`/proc/<pid>/oom_score`），这个值越大，就越有可能被OOM Killer选中并杀死。

当一个进程被OOM Killer杀死以后会向操作系统日志写入杀死的进程PID等信息，所以判断进程是否被OOM Killer，就可以通过搜索操作系统日志来得知。

ECS集群进程被OOM Killer的日志：

```
[Wed Aug 31 16:36:42 2017] Out of memory: Kill process 43805 (keystone-all) score 249 or sacrifice child
      [Wed Aug 31 16:36:42 2017] Killed process 43805 (keystone-all) total-vm:4446352 kB, anon-rss:4053140kB, file-rss:68kB
      [Wed Aug 31 16:56:25 2017] keystone-all invoked oom-killer: gfp_mask=0x280da, oorder=0, oom_score_adj=0
      [Wed Aug 31 16:56:25 2017] keystone-all cpuset=/ mems_allowed=0
      [Wed Aug 31 16:56:25 2017] CPU: 2 PID: 88196 Comm: keystone-all Not tainted 3.10.0-327.13.1.el7.x86_64 #1
```

Swarm集群进程被OOM Killer的日志:

```
Memory cgroup out of memory: Kill process 20911 (beam.smp) score 1001 or sacrifice child
Killed process 20977 (sh) total-vm:4404kB, anon-rss:0kB, file-rss:508kB
```

总结起来搜索命令为:

```
root# grep -i 'killed process' /var/log/messages
```

或者:

```
root# egrep "oom-killer|total-vm" /var/log/messages
```

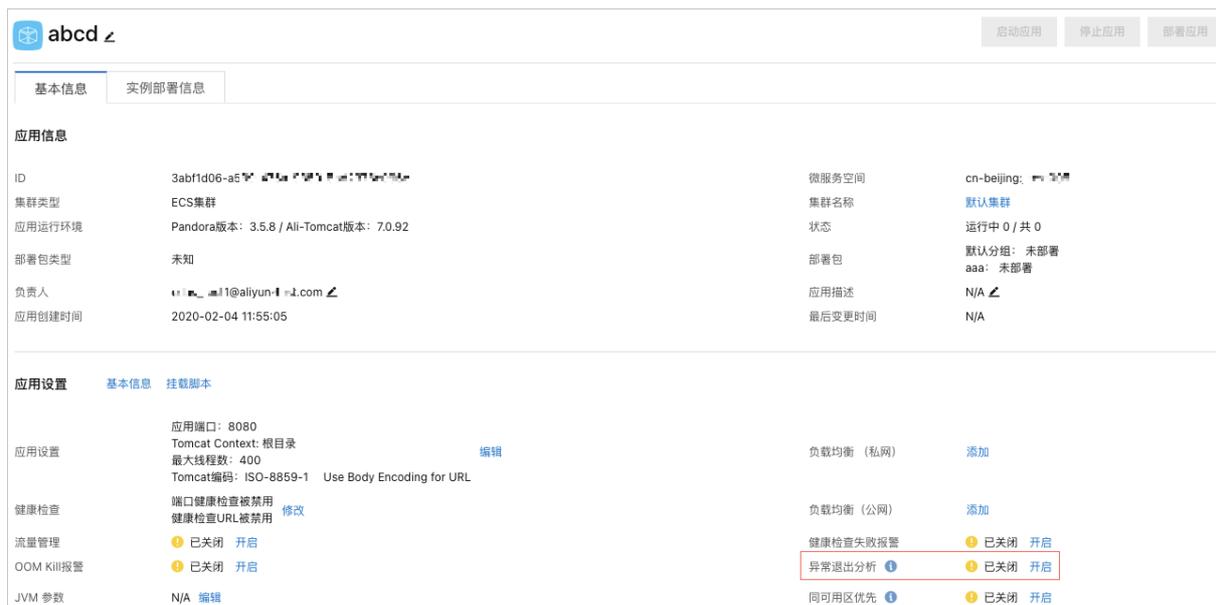
遇到此问题时, 解决办法有以下几个:

- 增大ECS实例物理内存或者减少被杀死的进程分配的内存值。
- 检查ECS实例是否挂载了Swap交换分区 (ECS实例默认不挂载Swap交换分区), 如果未挂载 (阿里云ECS环境大多数OOM Killer的问题都因为未挂载Swap交换分区, 相比性能, 进程健康性更加重要) 请搜索Linux挂载Swap交换分区的方法并自行挂载Swap交换分区。

## 应用运行的Java虚拟机进程Crash (异常退出)

Java虚拟机通常会由于异常的JNI调用、C Heap OOM、其他Bug等原因在运行时进程Crash, 发生此问题时, 会在当前JVM进程的工作目录 (通过`pwdx <jvm_pid>`命令可查) 生成一个`hs_err_<jvm_pid>.log`。通常, 从这个日志文件中即可查到Java虚拟机Crash时执行的线程或原因 (必要时需要通过允许生成coredump文件进一步分析)。

另外, 在EDAS控制台应用的基本信息页面, 应用设置区域的异常退出分析右侧单击开启, 配合应用监控告警功能, Java虚拟机进程退出时, 就会发出告警, 然后登录ECS实例查找相关日志, 分析具体原因。



## 9.2.5. 应用部署时提供的发布包 URL 是不是可以随意设置?

应用部署包的 URL 必须保证您的服务器是可以下载的。

## 9.2.6. 启停和部署应用等管理操作失败了怎么办?

一般来说，管理应用的操作失败是由于 EDAS Agent 没有正常运行导致。

### 操作步骤

1. 登录EDAS控制台。
2. 在左侧导航栏中选择资源管理 > ECS。
3. 在实例列表页面中查看 Agent 状态。
  - 如果该实例上已经安装了 Agent，其状态为在线、Docke 在线等。
  - 如果未安装 Agent 或安装失败，其状态为未知或异常。
4. 对于 Agent 状态为异常的情况，想排查具体原因请参考Agent安装后控制状态“未知”或“异常”。

## 9.2.7. 已买的 ECS 实例没有出现在实例列表中怎么办?

在创建应用时，已经购买的 ECS 实例没有出现在实例列表中怎么办？首先，您需检查购买的 ECS 实例上是否有成功安装 EDAS Agent；然后，需检查是否将购买的 ECS 实例已经添加到您要创建应用的集群中。

### 操作步骤

1. 登录EDAS控制台。
2. 在左侧导航栏中选择资源管理 > ECS。
3. 在实例列表页面中查看要创建应用的 ECS 的 Agent 状态。
  - 如果已经安装了 Agent，其状态为在线或 Docker 在线。
  - 如果未安装 Agent 或安装失败，其状态为未知或异常。
    - 如果 Agent 状态为未知，即该 ECS 实例上未安装 EDAS Agent，请为ECS实例安装EDAS Agent。

- 如果 Agent 状态为异常，则该 ECS 实例上安装 EDAS Agent 失败，想排查具体原因请参考[Agent 安装后控制状态“未知”或“异常”](#)。
4. 确保该 ECS 实例上已经安装 EDAS Agent 后，在 EDAS 控制台左侧导航栏中选择资源管理 > 集群，然后单击您要创建应用所在的集群名称，在集群详情页面查看您想要创建应用的 ECS 是否在该集群中。
    - 如果已经存在，请提交工单或联系 EDAS 技术支持人员。
    - 如果不存在，请在集群中添加 ECS 实例，具体操作请参见[使用控制台创建 ECS 集群](#)。

## 结果验证

按上述步骤完成操作后，重新创建应用，在选择集群后，您名下想要部署应用的 ECS 实例如果已经在实例列表中则说明 ECS 实例已成功导入集群。

## 9.2.8. 为什么在 EDAS 控制台上实例的状态是“未知”？

EDAS Agent 会定期汇报心跳数据给 EDAS Console，如果 Agent 停止汇报状态，则某段时间后该机器将会被判定为未知状态。通常而言，该问题是由于 Agent 停止导致。

## 9.2.9. 服务可以正常调用，但是服务列表不显示

接口 API 中存在泛型，但是泛型没有指定具体的类型，导致解析服务列表失败，请修改对应的代码。

## 9.2.10. 服务列表显示正常，但是调用失败怎么办？

1. 查看服务提供者对应的分组是否已经创建，如果没有创建则会鉴权失败。
2. 需要查看 /home/admin/logs/hsf/hsf.log，确认具体的错误码，根据错误码查询[定位及解决 HSF 问题](#)。

## 9.2.11. 删除应用后是否可以恢复？

不可以，删除应用操作不可逆，所有的数据都会被清除。

## 9.2.12. 如何进行分批发布或者分组发布

如果一个应用有多台实例，可以分批发布，可以分组发布，也可以基于不同实例分组进行分批发布。

### 操作步骤

1. 应用创建完成后，在应用详情页面，单击部署应用。
2. 在部署应用对话框中设置文件上传方式并上传 WAR/JAR 包，然后填写版本。
3. 设置发布目标分组。在发布目标分组右侧的下拉菜单中选择不同分组，即可进行分组发布。

如果下拉菜单中没有其它分组，例如 beta 分组，需要新建实例分组：

- i. 在应用详情页面的实例部署信息页签中单击创建新分组，在新建分组对话框中填写分组名（如 beta 分组），单击创建。
- ii. 返回实例部署信息页面，在默认分组下的实例右侧单击更换分组，在更换分组对话框中选择新建的分组（beta 分组），单击更换。
- iii. 然后依次将需要发布的实例添加到该分组中。

关于实例分组的详细操作，请参考[使用控制台管理 ECS 集群中的应用实例分组](#)。

4. 在批次右侧的下拉菜单中，选择发布批次。
5. 选择分批方式，然后单击发布。即可基于分组和批次进行应用发布了。

### 9.2.13. 应用部署在多个实例后，集群中会话如何共享？

目前 EDAS 尚未提供分布式 Session 管理的功能，您可以借助一些可以存储共享存储信息的缓存系统（例如：云数据库 Memcache 版，Redis 等）来进行管理。

### 9.2.14. 应用部署在多个实例后，如何实现负载均衡？

- EDAS 中提供 HTTP 服务的 Web 应用，需要使用 SLB 来进行负载均衡，您可以在 EDAS 的应用配置页面，进行 SLB 的配置。具体配置步骤，请参考[负载均衡概述](#)。
- 对于 EDAS 的 RPC 服务提供者对应的应用，您无需考虑负载均衡问题，EDAS 原生支持对 RPC 服务提供者的负载均衡。

### 9.2.15. 如何分批升降级应用的运行环境

对应用的运行环境版本进行升降级，会造成应用的不可用。您可以选择在访问流量小的时候进行，也可以选择分批进行。相关操作请参见[升级或降级运行时环境](#)。

### 9.2.16. 在 EDAS 控制台上对应用进行操作无响应

在 EDAS 控制台上对应用进行操作无响应是由于 EDAS Agent 进程退出导致，这个问题在新版 EDAS Agent 中已经修复。

请登录到您的实例，以 admin 账户执行 `/home/admin/edas-agent/bin/startup.sh`。如果应用还是无法操作，请单击重置。

### 9.2.17. 创建应用时的安全组规则

在 EDAS 上创建的应用所部署的 ECS 实例会使用您的账号默认启用如下两个安全组规则：

- 监控告警日志采集：端口范围：8182/8182；适用网络类型：经典网络；
- 健康检查健康检查：
  - 授权类型：100.104.0.0/16。
  - 适用网络类型：VPC。
  - 生效时间：2018年3月23日。

EDAS 保证不会将您的账号用于任何其它用途。如果您对该策略有顾虑，可以终止应用创建流程。

### 9.2.18. ECS集群应用日志清理机制是什么

在EDAS中部署的应用的日志的清理机制有以下常见问题。

#### 在EDAS中部署的应用的日志是否会被定期清理？

仅应用运行环境是EDAS-Container x.x.x，且是ECS集群、WAR包格式时（会使用到Ali-Tomcat），EDAS的定时任务脚本才会清理 *Ali-Tomcat* 日志目录下的日志（具体以ECS集群应用部署的ECS主机上使用admin用户执行 `crontab -l` 看到的每天凌晨3点10分的定时清理任务为准），有定时清理脚本的情况：

```
[admin@iZbp1bzvz55uz2x8s0bXXXX ~]$ crontab -l
10 3 * * * /home/admin/edas-agent/bin/rotator.sh
```

ECS集群但应用运行环境非EDAS-Container x.x.x、Swarm集群、容器服务K8s集群的应用日志以及可能产生的Tomcat日志目前还不会被清理。

## EDAS的日志定时清理脚本会清理哪些路径下的日志？清理的规则又是什么？

- `/home/admin/taobao-tomcat*`目录下那些包含log、out关键字的，且文件大于1 KB、最后修改时间是5天前的日志文件会被清理掉，同时，会轮询 `/home/admin/taobao-tomcat*/catalina.out`文件。
- `/home/admin/configclient`目录下那些包含log关键字的，且文件大于1 KB、最后修改时间是7天前的日志文件会被清理掉。
- `/home/admin/logs/diamond-client`目录下那些包含log关键字的，且文件大于1 KB、最后修改时间是7天前的日志文件会被清理掉。
- `/home/admin/logs`目录下（包括子目录）那些以.log 后缀结尾的，且文件大小大于500 MB、且当文件个数大于3个且较旧的日志文件会被清理掉。

## EDAS的日志定时清理脚本无法满足使用需求，日志中仍然会产生很多日志怎么办？

- Ali-Tomcat基于Apache Tomcat定制，在日志记录这方面与Apache Tomcat保持一致，应用部署在EDAS中，运行环境无论是EDAS-Container x.x.x（内含Ali-Tomcat），还是Apache Tomcat，应用都不宜向Tomcat的 `catalina.out`日志文件写入较多的日志，或者将应用自定义的日志输出到Tomcat的日志目录 `$CATALINA_HOME/logs`，频繁写入，可能会导致ECS主机 `/home`挂载点目录对应的磁盘空间耗尽。
- 建议在应用中使用Java自带或者开源日志框架来管控应用日志，并使用日志框架去轮询和控制应用日志文件保存的数量、大小、个数等，避免应用日志写入过多耗尽磁盘空间。

## 9.2.19. 变更流程问题排查指南

当您在EDAS上进行应用部署、启动、扩容/缩容等操作时，应用变更记录将整个变更过程以可视化的方式展现给您。您可以直观的看到变更流程的进度。如果变更失败，您可以在变更详情中查询变更流程中哪些步骤执行失败了，以及失败的原因。

### 排查变更失败问题

执行应用变更操作后，可以跳转到应用详情页，通过变更流程排查变更失败问题。

1. 在应用详情页面左侧的导航栏中单击变更记录。

创建时间	结束时间	变更类型	描述信息	变更状态	变更人	来源	操作
2020-12-31 15:00:07	2020-12-31 15:00:08	部署应用	版本: 1590044655025   镜像名称: [REDACTED]	● 执行失败	[REDACTED]	console	<a href="#">查看</a>

如果某次变更失败，则在变更状态列会显示执行失败。

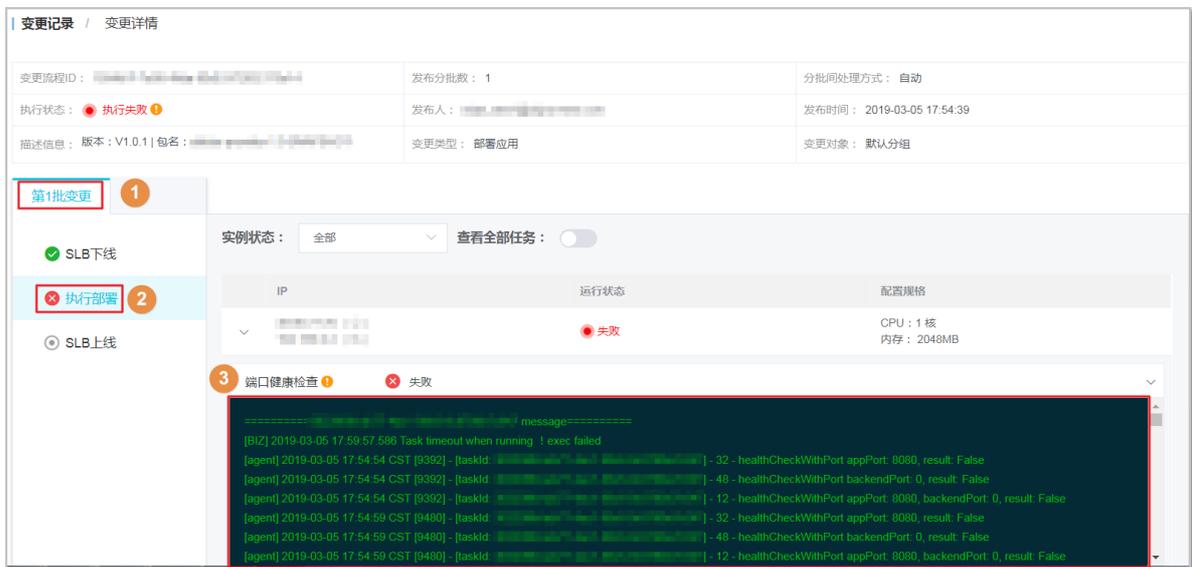
2. 在变更记录页面执行失败的变更记录的操作列单击查看。
3. 在变更详情页面的第一批部署（取决于您在部署该应用时设置的分批数。如果您设置了2批部署，则会有第一批部署和第二批部署两个页面。）区域单击失败的阶段（如执行部署），在右侧实例中查看失败步骤的具体日志，定位问题。



## 应用变更典型异常及处理

下面介绍应用变更时的几种典型异常如何通过变更流程排查及处理。

### ● 端口健康检查异常导致变更失败



#### 可能原因

- 应用启动时应该被正常占用的65000端口不可访问。
- 客户的Tomcat容器没有正常启动。

#### 处理方法

查看应用日志是否有相应的异常日志。

### ● URL健康检查失败导致变更失败

如果用户配置了健康检查URL，应用在启动时会检查这个URL是否可以正常访问。在收到Http code为200的返回码后，应用发布流程才会继续进行。如果在3分钟的健康检查周期内都收到错误的响应（非200），则认为URL健康检查失败。



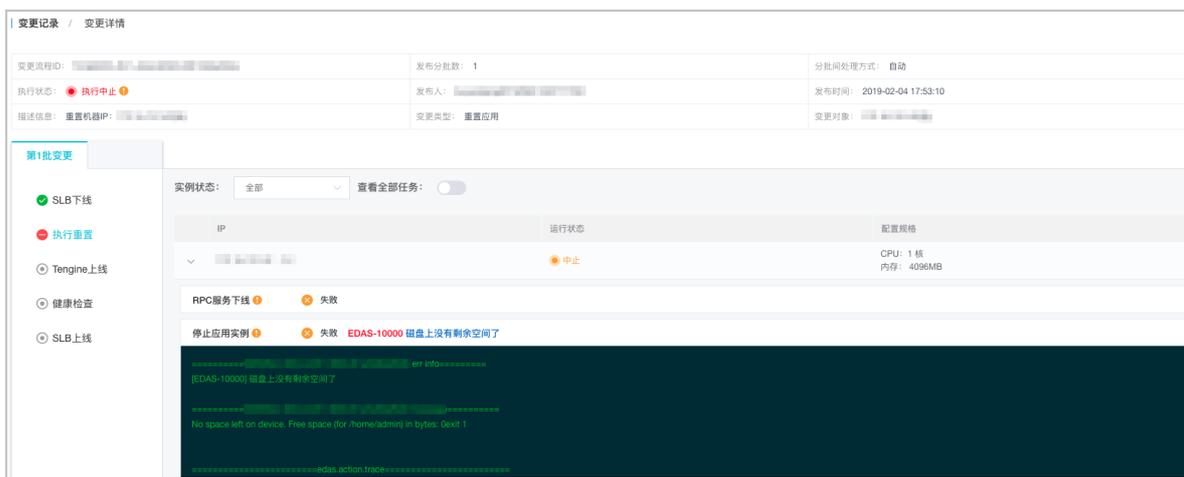
### 处理方法

- 应用启动失败，需要检查应用日志是否正常启动。
- 检查URL地址是否配置出错。
- 检查应用是否配置了拦截器，拦截了对HTTP健康检查URL的请求。

处理完成后，再次部署应用。

### ● 磁盘满导致变更失败

磁盘满会导致在变更流程中，应用长时间处于某个阶段。



### 报错信息

[EDAS-10000] 磁盘上没有剩余空间了 或 No space left on device. Free space (for /home/admin) in bytes: 0exit 1。说明机器的磁盘满了，一般是应用的业务日志将磁盘打满了，此时需要用户清理应用日志。

### 处理方法

- ECS集群应用需要登录ECS清理磁盘上的日志。
- Swarm集群下的Docker应用则需要重置应用。重置后，容器会被停止、删掉，然后重建。这样磁盘空间就会被释放出来。

## 9.2.20. EDAS应用无法访问页面及RPC服务无法调用

本文主要介绍企业级分布式应用服务EDAS应用无法访问页面及RPC服务无法调用的处理方案。

### 问题描述

通过HTTP方式无法访问页面，远程过程调用RPC服务无法调用。

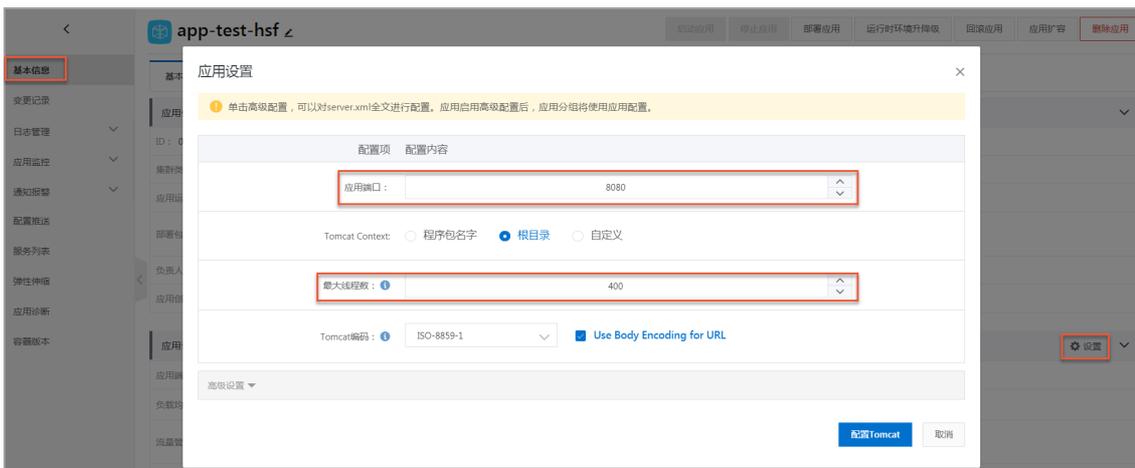
### 问题原因

1. 网络问题、应用配置路径错误、应用本身错误等。
2. 服务器到注册中心网络不可达、服务权限限制、服务调用超时等。

### 解决方案

1. 通过HTTP方式无法访问页面。
  - 通过服务器本机 `curl/ping` 等命令确认网页URL是否正常。如果正常，检查访问机到服务机的网络，将网络问题解决后进行重试。
  - 如果不是网络问题，请检查Tomcat日志错误，解决所有日志错误为止。

- 排查HTTP路径，特别是应用设置的Context配置和端口配置。



- 执行如下命令，查看Java程序进程ID。

```
ps -ef |grep java
```

- 排查程序运行是否正常，依次执行如下命令，通过JVM工具排查。

```
sudo jstack -F 进程ID
sudo jstat -class [vmid] [interval] [count];
jmap -heap 2083
```

- [vmid]: VM的进程号，即当前运行的Java进程号。
- [interval]: 间隔时间，单位为秒或毫秒。
- [count]: 打印次数，如果缺省则打印无数次。

## 2. RPC服务无法调用。

- 检查服务调用应用和发布应用是同一个账号，网络可达。服务器端口是否被屏蔽导致服务无法访问。
- 检查服务Tomcat应用日志错误，解决所有日志错误，直到服务列表能正常查询到服务。
- 查看/home/admin/logs/hsf/hsf.log日志，确认具体的错误码，具体错误及解决方案，请参见[HSF错误码](#)。

## 9.2.21. 快速入门常见问题

如果您在使用EDAS体验创建、部署应用时遇到了问题，可以按本文提供的方法尝试解决。

### 如何选择ECS集群和Kubernetes集群

创建应用第一步，面对集群类型选择时，您可以根据以下几个指标来选择适合的集群类型：

**说明** 如果您在选择集群类型时感到困惑，建议您选择ECS集群。

集群类型	部署形态	收费标准	运维成本	与云原生的结合	与云效的结合
------	------	------	------	---------	--------

集群类型	部署形态	收费标准	运维成本	与云原生的结合	与云效的结合
ECS集群	单个ECS实例上部署一个应用。	<ul style="list-style-type: none"> <li>• IaaS层：按使用的ECS的规格计费，详情请参见<a href="#">计费概述</a>。</li> <li>• PaaS层：在EDAS上按应用实例（部署了应用的ECS实例）计费，详情请参见<a href="#">价格说明</a>。</li> </ul>	贴近传统的Linux运维操作，门槛低，可操作性强。	无	云效支持在ECS集群中部署应用，详情请参见 <a href="#">使用云效2020部署Java应用至ECS集群</a> 。
Kubernetes集群	单个Pod上部署一个应用，单个ECS实例上可以运行多个Pod。	<ul style="list-style-type: none"> <li>• IaaS层：按使用的容器服务Kubernetes集群计费，详情请参见<a href="#">产品计费</a>。</li> <li>• PaaS层：在EDAS上按应用实例（部署了应用的Pod）计费，详情请参见<a href="#">价格说明</a>。</li> </ul>	需要对虚拟化、容器技术和Kubernetes集群有基本的了解和使用经验。	与云原生天然结合，可以一个镜像，多处运行。	云效支持在Kubernetes集群中部署应用，详情请参见 <a href="#">使用云效2020进行持续集成和部署(K8s)</a> 。

### 使用新购ECS实例创建应用时，提示余额不足

● 现象



● 可能的原因

当使用新购买的实例来创建应用时，EDAS将为您代购按量付费的ECS实例，当前账号余额需要不少于100元。而您的账号余额小于100元，导致购买按量付费ECS失败。

- 解决办法
  - 到[阿里云用户中心-充值](#)页面充值，保证账号余额大于等于100元。
  - 提前购买ECS，再在EDAS中使用已购买的ECS创建部署应用。购买ECS实例请参见[创建ECS实例](#)。

## 为什么自定义环境中只有实例规格

- 现象  
在应用配置页面选择后只能看到实例规格，看不到VPC和具体ECS实例。



- 可能的原因  
当前账号在当前地域下，没有创建过VPC、微服务空间、集群和ECS，所以只需选择实例规格，EDAS会帮您创建默认的VPC和指定规格的ECS实例，并在EDAS中创建默认微服务空间和集群。
- 解决办法  
如果您对网络、服务隔离和ECS有具体的需求，可以先创建VPC、微服务空间、集群和ECS实例，则在该页面可以选择相应的资源部署应用。

## 集群导入失败

- 现象  
在应用配置页面选择自定义，在实例区域选择使用已有实例，再选择未导入到集群的实例，最后在创建应用的时候遇到导入机器到集群失败的报错，如图所示。



- 可能的原因  
选择的ECS的操作系统不在EDAS支持的范围内。
- 解决办法  
重新选择ECS实例，操作系统需要是Aliyun Linux 2.1903 64位或CentOS 7.2/7.3/7.4/7.6 64位的。

## 如何修改应用端口

- 在ECS集群中部署的应用，可以在应用详情页中修改应用配置中的Tomcat端口来指定应用端口。如果遇到指定无效，请参见[为什么在代码中配置的Tomcat端口不生效？](#)。
- 在Kubernetes集群中部署的应用，暂时只能通过代码中指定或者通过挂载SLB的方式来修改应用端口。

如果在使用新流程创建应用时遇到任何问题，欢迎您加入钉钉群进行反馈或咨询。钉钉群号：21958624

## 9.2.22. 应用部署时的默认启动命令参数

应用部署时，为了优化应用的性能，保证应用可以使用中间件和应用管理相关的功能，EDAS会默认添加启动命令参数。

配置项	配置值	解析	类别
alicloud.deployment.mode	EDAS_MANAGED	添加此配置后，会自动将中间件组件的服务端地址替换为EDAS提供的服务端地址，同时填充鉴权信息。	微服务
spas.identity	/home/admin/.spas_key/default	中间件鉴权信息所在的文件，与tenant.id一一对应。	微服务
address.server.domain	addr-sh-internal.edas.aliyun.com	中间件地址服务器的地址，每个Region对应一个，通过此地址去查找中间件服务端的真实地址。	微服务
address.server.port	8080	中间件地址服务器的地址端口，通过此地址去查找中间件服务端的真实地址。	微服务

配置项	配置值	解析	类别
configserver.client.port	8000	ConfigServer服务端的地址，配置后，Config Server Client会去连接服务端的这个端口。	微服务
dpath.id.group	default	全链路灰度需要使用到的参数。	微服务
dpath.id	44990d8b-4e65-41af-b11a-733253847d2d	全链路灰度需要使用到的参数。	微服务
-javaagent	/home/admin/.opt/ArmsAgent/arms-bootstrap-1.7.0-SNAPSHOT.jar	默认在Java应用启动时挂载一个Java Agent。包含免费版本的ARMS，以及服务无损下线、离群摘除等微服务能力。	微服务
project.name	实际值	应用ID。	应用管理
ahas.project.name	实际值	应用名，AHAS组件会从这里获取。	应用管理
tenant.id	实际值	租户ID，与命名空间一一对应，鉴权和隔离使用。	应用管理
pandora.accept.foreign.ip	FALSE	Pandora的控制端口是否允许外部IP调用，默认关闭，确保安全。	中间件
JM.LOG.RETAIN.COUNT	7	com.taobao.middleware.logger.Logger的配置，最大保存7个日志文件。	<ul style="list-style-type: none"> <li>日志</li> <li>中间件</li> </ul>
JM.LOG.FILE.SIZE	300 MB	com.taobao.middleware.logger.Logger的配置，每个文件最大的值为300 MB。	<ul style="list-style-type: none"> <li>日志</li> <li>中间件</li> </ul>
tomcat.monitor.http.binding.host	实际值	tomcat.monitor.http.binding.host Tomcat monitor绑定的地址，一般是本机地址。	<ul style="list-style-type: none"> <li>日志</li> <li>中间件</li> </ul>
jdk.tls.ephemeralDHKey Size	2048	将TLS的DH密钥大小扩展到2048位，增强安全性。	Tomcat
java.security.egd	file:/dev/./urandom	配置Java随机数生成的种子（urandom的u是“unlocked”，非阻塞的随机数发生器），加速Tomcat启动。	Tomcat

配置项	配置值	解析	类别
org.apache.tomcat.util.http.ServerCookie.ALLOW_EQUALS_IN_VALUE	TRUE	添加配置后, Cookie的值可以包含等号 (=)。	Tomcat
org.apache.tomcat.util.http.ServerCookie.ALLOW_HTTP_SEPARATORS_IN_V0	TRUE	添加配置后, Cookie的值可以包含HTTP分隔符, 如正斜线 (/)。	Tomcat
atalina.logs	/home/admin/taobao-tomcat-production-7.0.59.3/logs	Tomcat日志在所在的文件夹目录地址。	Tomcat
atalina.base	/home/admin/taobao-tomcat-production-7.0.59.3	Tomcat所在的地址。	Tomcat
atalina.home	/home/admin/taobao-tomcat-production-7.0.59.3	Tomcat所在的地址。	Tomcat
--server.context-path	/	设置Spring Boot应用Tomcat的context path。	Tomcat
--server.port	8080	设置Spring Boot应用Tomcat监听的端口。	Tomcat
server.tomcat.uri-encoding	ISO-8859-1	设置Spring Boot应用Tomcat的uri-encoding编码。	Tomcat
--server.tomcat.max-threads	400	设置Spring boot应用最大线程池为400。	Tomcat
java.util.logging.config.file	/home/admin/taobao-tomcat-production-7.0.59.3/conf/logging.properties	设定Java自带日志组件配置的文件。	日志
java.util.logging.manager	org.apache.juli.ClassLoaderLogManager	设定Java自带的日志组件的Manager。	日志
java.io.tmpdir	/home/admin/taobao-tomcat-production-7.0.59.3/temp	Java临时文件存储的地址。	通用
atalina.vendor	alibaba	没有实质性的意义, 就是表征这个Ali-Tomcat是阿里巴巴开发的。	其他

## 9.3. 监控报警

### 9.3.1. 本地使用jconsole或jvisualvm监控ECS实例上的JVM进程

当监控各应用使用的Java虚拟机的堆内存、线程、MBean时，除了使用EDAS为ECS集群和Swarm集群中部署的应用提供的应用诊断功能以外，还可以使用JDK自带的jconsole或者jvisualvm（这两个小程序默认位于`$JAVA_HOME/bin`目录下，jvisualvm也可以使用开源版）。

**说明** 该方法适用于在本地（办公电脑环境）通过jconsole或者jvisualvm远程获取运行在ECS实例上的JVM进程的配置与运行时信息，且创建ECS实例时需要为实例分配公网IPv4地址。

1. 登录EDAS 控制台。
2. 在左侧导航栏中选择应用列表。
3. 在应用列表页面单击具体应用名称。
4. 在应用详情基本信息页面的应用设置区域，单击JVM参数的编辑。
5. 在应用设置对话框的自定义区域，设置自定义参数，然后单击配置JVM参数。



-Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.port=一个 1024 以上的端口 -Dcom.sun.management.jmxremote.rmi.port=同 com.sun.management.jmxremote.port 属性设置的端口 -Djava.rmi.server.hostname=ECS 实例的公网 IP 地址

6. 设置完成后，返回应用详情页面，可以在基本信息页面的应用设置区域下方看到配置的JVM参数。
7. 重启应用，使配置的参数生效。

## 在ECS实例上查看JVM进程是否打开了JMX Remote端口

1. 登录[台 EDAS 控制台](#)。
2. 在左侧导航栏中选择[应用列表](#)。
3. 在[应用列表](#)页面单击具体应用名称。
4. 在[应用详情实例部署信息](#)页面查看ECS实例的IP地址。
5. 使用SSH工具登录ECS实例，然后执行下面的命令，查看应用JVM进程是否打开了JMX Remote端口。

```
[admin@izbp146****ubo8si0ihkz ~]$ netstat -antp | grep 10000
      (Not all processes could be identified, non-owned process info
      will not be shown, you would have to be root to see it all.)
tcp    0      0 0.0.0.0:10000      0.0.0.0:*          LISTEN      645
7/java
```

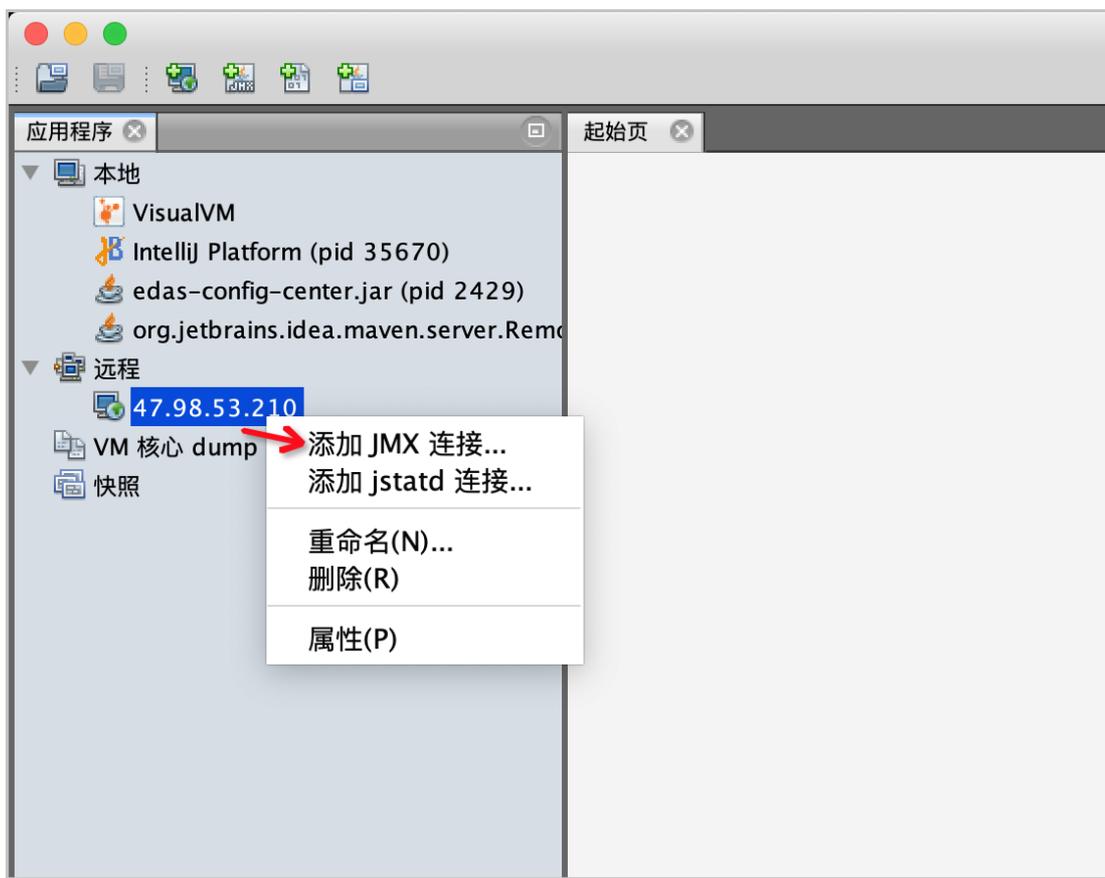
如果JMX Remote端口未打开，请检查JVM进程命令行是否包含上面的JVM属性参数。  
正常的打开了JMX Remote端口的JVM进程示例如下：

```
[admin@izbp146****ubo8si0ihkz ~]$ ps -ef|grep tomcat | grep -v grep
admin      6457      1   1 15:46 ?        00:03:18 //bin/java -Djava.util.logging.config.file=/home/admin/taobao-tomcat-production-7.0.59.3/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Dcatalina.vendor=alibaba -Djava.security.egd=file:/dev/./urandom -Dlog4j.defaultInitOverride=true -Dorg.apache.tomcat.util.http.ServerCookie.ALLOW_EQUALS_IN_VALUE=true -Dorg.apache.tomcat.util.http.ServerCookie.ALLOW_HTTP_SEPARATORS_IN_V0=true -Dproject.name=blc8a3ea-309b-43e8-b716-f52008784801 -Dtenant.id=e19f8e62-84db-4ff3-aae6-b5750a6a248a -Dalicloud.deployment.mode=EDAS_MANAGED -Dpandora.accept.foreign.ip=false -Dlog4j.defaultInitOverride=false -Dspas.identity=/home/admin/.spas_key/default -Daddress.server.domain=addr-hz-internal.edas.aliyun.com -Daddress.server.port=8080 -Dconfigserver.client.port=8000 -DJM.LOG.RETAIN.COUNT=7 -DJM.LOG.FILE.SIZE=300MB -Dtomcat.monitor.http.binding.host=172.16.129.92 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.port=10000 -Dcom.sun.management.jmxremote.rmi.port=10000 -Djava.rmi.server.hostname=47.98.53.210 -Dcatalina.logs=/home/admin/taobao-tomcat-production-7.0.59.3/logs -Dignore.endorsed.dirs= -classpath /home/admin/taobao-tomcat-production-7.0.59.3/bin/bootstrap.jar:/home/admin/taobao-tomcat-production-7.0.59.3/bin/tomcat-juli.jar -Dcatalina.base=/home/admin/taobao-tomcat-production-7.0.59.3 -Dcatalina.home=/home/admin/taobao-tomcat-production-7.0.59.3 -Djava.io.tmpdir=/home/admin/taobao-tomcat-production-7.0.59.3/temp org.apache.catalina.startup.Bootstrap start
```

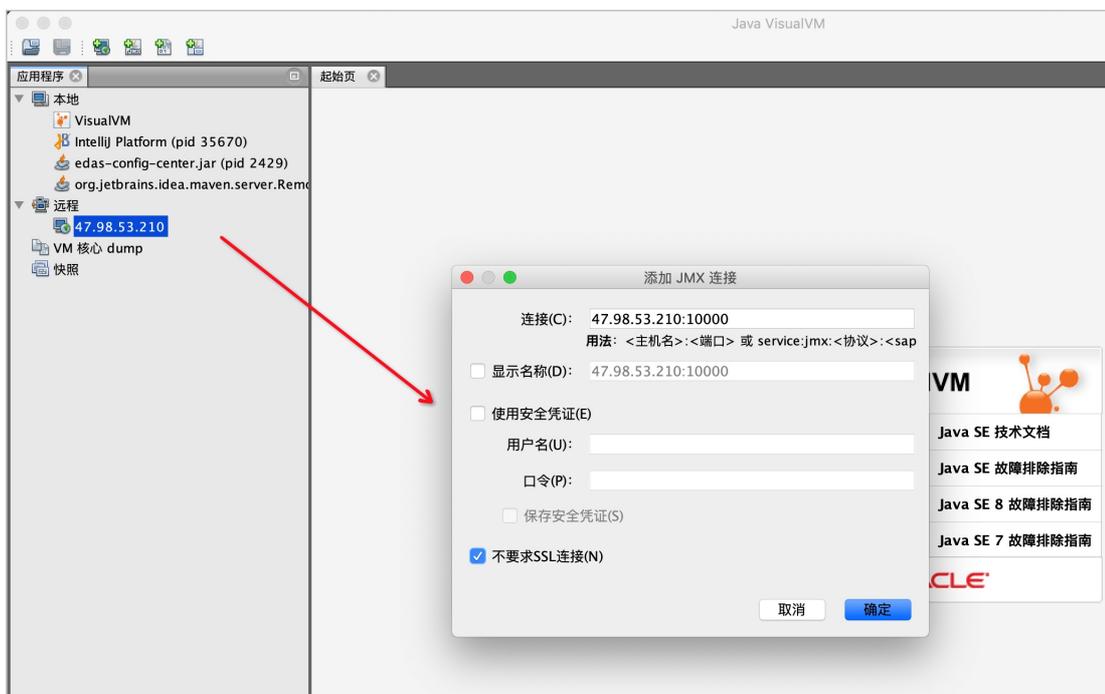
6. 在[应用详情的实例部署信息](#)页面，单击具体实例名称。
7. 在[云服务器管理控制台](#)的实例详情页单击[安全组](#)页签。
8. 在[安全组列表](#)页签单击[配置规则](#)，添加JMX Port的安全组规则。
9. 在本地打开jconsole或者jvisualvm，添加该远程ECS实例上JVM的JMX监控。  
下面以 jvisualvm为例进行配置说明：

 **说明** 本文中的配置仅为测试样例，未开启身份安全验证，如果需要在正式环境使用，请开启安全验证。由于jconsole、jvisualvm是JDK提供的监控工具，与EDAS产品并无关系，更多配置请参考jconsole或者jvisualvm官方文档。

i. 添加JMX连接。



ii. 配置远程JMX连接。



iii. 查看远程JVM连接监控。



## 9.4. 日志诊断常见问题

### 9.4.1. Java 应用 CPU 使用率高

Java 应用程序运行中，有时会遇到 Java 应用（JVM）进程 CPU 使用率高的情况。在这种情况下，Java 应用的性能通常会下降，我们可以借助一些工具或命令进行分析诊断，找到并优化造成 Java 应用（JVM）进程 CPU 使用率高的原因。

#### 1. 使用 edas-agent 自带的命令诊断

EDAS 为导入 ECS 集群中的 ECS 提供了一个可以直接显示出应用进程中使用 CPU 的线程及其 StackTrace，可以帮助用户快速找到造成应用进程 CPU 使用率高的问题原因。

使用 root 账号通过 SSH 登录到 CPU 高的应用进程所在的 ECS，然后切换到 admin 账号，执行 `edas busy-threads` 即可查看到应用进程中消耗 CPU 高的线程（默认前 5 个）：

```
[root@izbp19o2g75lcdht92iaaeZ ~]# su - admin
[admin@izbp19o2g75lcdht92iaaeZ ~]$ edas busy-threads
09/28/19 22:57:07 [INFO] EXECUTING: busy-threads
[1] Busy(4.6%) thread(3222/0xc96) stack of java process(3221) under user(admin):
"main" #1 prio=5 os_prio=0 tid=0x00002ab68004e800 nid=0xc96 runnable [0x00002ab67c1df000]
  java.lang.Thread.State: RUNNABLE
    at java.net.PlainSocketImpl.socketAccept(Native Method)
    at java.net.AbstractPlainSocketImpl.accept(AbstractPlainSocketImpl.java:409)
    at java.net.ServerSocket.implAccept(ServerSocket.java:545)
    at java.net.ServerSocket.accept(ServerSocket.java:513)
    at org.apache.catalina.core.StandardServer.await(StandardServer.java:490)
    at org.apache.catalina.startup.Catalina.await(Catalina.java:819)
    at org.apache.catalina.startup.Catalina.start(Catalina.java:765)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:309)
    at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:443)
[2] Busy(0.9%) thread(2725/0xaa5) stack of java process(2721) under user(admin):
"DestroyJavaVM" #13 prio=5 os_prio=0 tid=0x00002ba81004c000 nid=0xaa5 waiting on condition
[0x0000000000000000]
  java.lang.Thread.State: RUNNABLE
[3] Busy(0.0%) thread(3221/0xc95) stack of java process(3221) under user(admin):
[4] Busy(0.0%) thread(2721/0xaa1) stack of java process(2721) under user(admin):
```

该脚本的使用帮助如下：

```
[admin@izbp19o2g75lcdht92iaaeZ component]$ edas busy-threads -h
Usage: busy-threads.sh [OPTION]... [delay [count]]
Find out the highest cpu consumed threads of java, and print the stack of these threads.
Example:
  busy-threads.sh          # show busy java threads info
  busy-threads.sh 1       # update every 1 seconds, (stop by eg: CTRL+C)
  busy-threads.sh 3 10   # update every 3 seconds, update 10 times
Options:
  -p, --pid <java pid>    find out the highest cpu consumed threads from the specifed java process,
                           default from all java process.
  -c, --count <num>       set the thread count to show, default is 5
  -a, --append-file <file> specify the file to append output as log
  -s, --jstack-path <path> specify the path of jstack command
  -F, --force              set jstack to force a thread dump
                           use when jstack <pid> does not respond (process is hung)
  -m, --mix-native-frames set jstack to print both java and native frames (mixed mode)
  -l, --lock-info          set jstack with long listing. Prints additional information about locks
  --cpu_period             period of time for collecting cpu stats, in secs, default 2 (this option is valid only when --current option is set)
  --current                use current cpu stats rather than that since start
  -h, --help              display this help and exit
```

常用的几个参数及选项（注意使用的是 admin 账号执行）：

```
#每隔 2 秒执行一次，共执行 5 次（每次默认显示应用进程中前 5 个使用 CPU 高的线程）
edas busy-threads 2 5
#显示指定 Java 进程的前 5 个使用 CPU 高的线程：
edas busy-threads -p <jvm_pid>
#显示 Java 进程中前 10 个使用 CPU 高的线程：
edas busy-threads -c 10
#计算最近的 2 秒种内 CPU 使用较高的线程：
edas busy-threads --current
```

## 2. 使用开源工具诊断

在非 ECS 集群的环境，可以直接使用下面这个开源的脚本来找到占用指定进程中排名前几位 CPU 高的线程：

```
admin$ wget --no-check-certificate https://raw.githubusercontent.com/oldratlee/useful-scripts/release/show-busy-java-threads
admin$ chmod +x show-busy-java-threads
admin$ ./show-busy-java-threads
```

具体使用方法，请参见 [show-busy-java-threads](#)。

除了这个脚本以外，还可以使用阿里巴巴的 Java 问题综合诊断工具 Arthas。该工具也可以用来显示指定 JVM 进程中使用 CPU 的排名前几位的线程。

```
#显示当前连接的 JVM 进程中 CPU 占用排名前 3 名的线程及 StraceTrace 信息：
admin$ thread -n 3
"as-command-execute-daemon" Id=29 cpuUsage=75% RUNNABLE
  at sun.management.ThreadImpl.dumpThreads0(Native Method)
  at sun.management.ThreadImpl.getThreadInfo(ThreadImpl.java:440)
  at com.taobao.arthas.core.command.monitor200.ThreadCommand$1.action(ThreadCommand.java:58)
  at com.taobao.arthas.core.command.handler.AbstractCommandHandler.execute(AbstractCommandHandler.java:238)
  at com.taobao.arthas.core.command.handler.DefaultCommandHandler.handleCommand(DefaultCommandHandler.java:67)
  at com.taobao.arthas.core.server.ArthasServer$4.run(ArthasServer.java:276)
  at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
  at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
  at java.lang.Thread.run(Thread.java:745)
Number of locked synchronizers = 1
- java.util.concurrent.ThreadPoolExecutor$Worker@6cd0b6f8
"as-session-expire-daemon" Id=25 cpuUsage=24% TIMED_WAITING
  at java.lang.Thread.sleep(Native Method)
  at com.taobao.arthas.core.server.DefaultSessionManager$2.run(DefaultSessionManager.java:85)
"Reference Handler" Id=2 cpuUsage=0% WAITING on java.lang.ref.Reference$Lock@69ba0f27
  at java.lang.Object.wait(Native Method)
  - waiting on java.lang.ref.Reference$Lock@69ba0f27
  at java.lang.Object.wait(Object.java:503)
  at java.lang.ref.Reference$ReferenceHandler.run(Reference.java:133)
```

更多参数及解释，详见 `thread` 命令。

### 3. 使用传统方式诊断

在没有互联网连接的主机还可以使用下面原始的方式来获取 CPU 占用较高的线程：

```
#在执行 JVM 进程 CPU 高时，每隔 3-5 秒执行一次，执行 3-5 次。
admin$ top -Hbp <jvm_pid> -d 1 -n 1 >> top.<jvm_pid>.txt && jstack <jvm_pid> >> jstack.*jvm
_pid*.txt
```

例如：

```
admin$ top -Hbp 22801 -d 1 -n 1 >> top.22801.txt && jstack 22801 >> jstack.22801.txt
```

1. 从收集到的 `top.xxxxx.txt` 中找到 CPU 占用率最高的线程 ID（注意跟 `jstack.xxxxx.txt` 文件中的线程堆栈信息一一对应）。
2. 将这些线程 ID（十进制的数字）转换成十六进制（可以用 `printf %x 1234`）。
3. 用这些转换后的十六进制的线程 ID 去 `jstack.xxxxx.txt` 文件中搜索，即可找到对应线程的信息（这方面资料较多，这里不再赘述）。