

# 阿里云 EDAS 用户指南

---

针对 EDAS v2.3.0

©Alibaba EDAS 项目组

2016/4/5

---

# 目录

1	前言	1
2	产品架构	1
3	专业术语	2
4	EDAS 功能特性	2
5	产品功能详细介绍	3
5.1	分布式应用服务化	3
5.1.1	分布式应用服务化	3
5.1.2	服务框架整体结构	3
5.1.3	发布 HSF 服务	4
5.1.4	消费 HSF 服务	4
5.2	EDAS Session Framework	5
5.2.1	引入 jar 包依赖	5
5.2.2	Webx MVC 框架注意事项	5
5.2.3	添加 Filter	6
5.2.4	添加 Filter Mapping	7
5.3	EDAS Agent 的安装	8
5.4	应用生命周期管理	9
5.4.1	创建应用	9
5.4.2	部署应用	11
5.4.3	启动应用	12
5.4.4	回滚应用	13
5.4.5	应用扩容	14
5.4.6	停止应用	14
5.5	实时日志	15

---

5.6	容器诊室.....	15
5.7	服务列表.....	16
5.8	链路分析.....	17
5.9	容量规划.....	19
5.10	限流降级.....	20
5.11	基础监控.....	22
5.12	应用监控.....	23
5.13	帐号管理.....	24
5.14	资源组.....	24

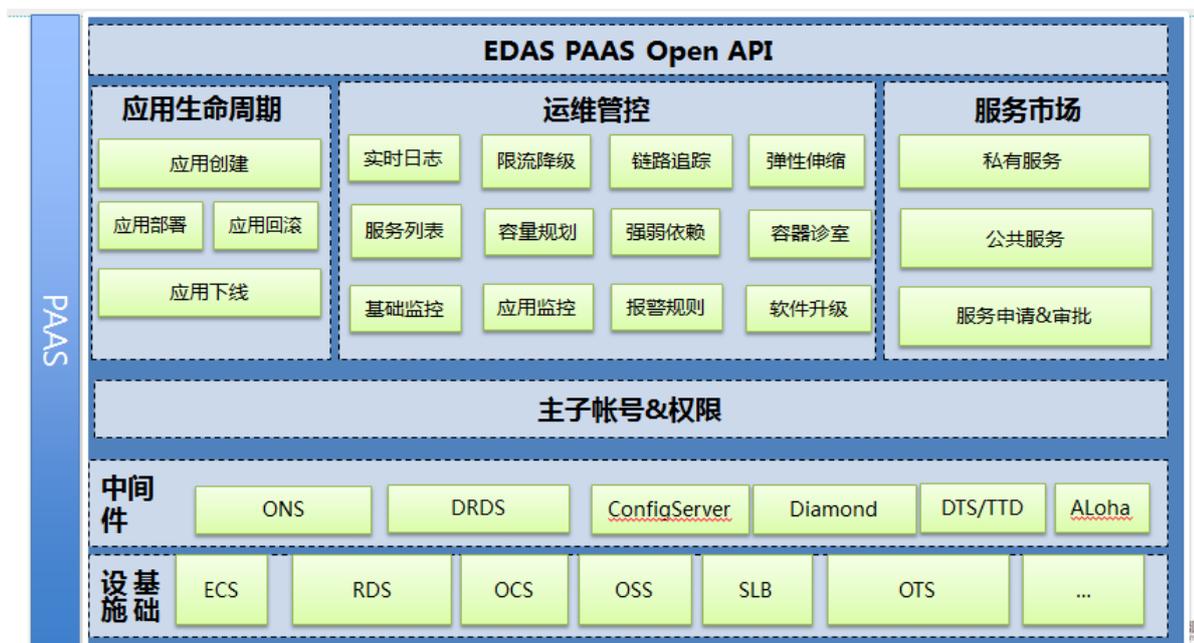
# 1 前言

本文档旨在描述阿里云 EDAS 产品的基本概念，以及如何使用。

## 2 产品架构

EDAS ( 企业级分布式应用服务，Enterprise Distributed Application Service ) 是一个以阿里巴巴中间件团队的多款久经沙场的分布式产品作为核心基础组件构建的企业云计算解决方案，其充分利用阿里云现有的资源管理和服务体系，引入中间件整套成熟的分布式计算框架 ( 包括分布式服务化、链路追踪和稳定性组件等 )，以应用为中心，帮助企业级客户轻松构建大型分布式应用服务。

EDAS 对于分布式应用强大的管理能力，能够非常方便的帮助企业级应用实现一站式应用生命周期的管理，体现到包括服务治理、基础监控、应用监控和应用诊断在内的一系列配套管理服务；同时，EDAS 独创的链路分析，限流降级、容量规划和自动压测等一系列稳定性模块，将极大的提升企业客户对大型分布式应用的管理能力,利用弹性伸缩,轻松应对各种流量高峰.整体架构图如下:



---

## 3 专业术语

- Agent

安装于用户 ECS 上，负责 EDAS 控制台与用户 ECS 之间的通信，以此来实现对应用的管理。

- HSF

EDAS 产品中分布式服务化子模块的名字，是一个高性能的服务化框架，全称 High Speed FrameWork。

## 4 EDAS 功能特性

企业级用户通过使用 EDAS PAAS 平台，能够轻松自如的操作所有的应用，平台具有应用生命周期管理，以及强大的分布式服务的运维管控功能，目前其产品核心功能包括：

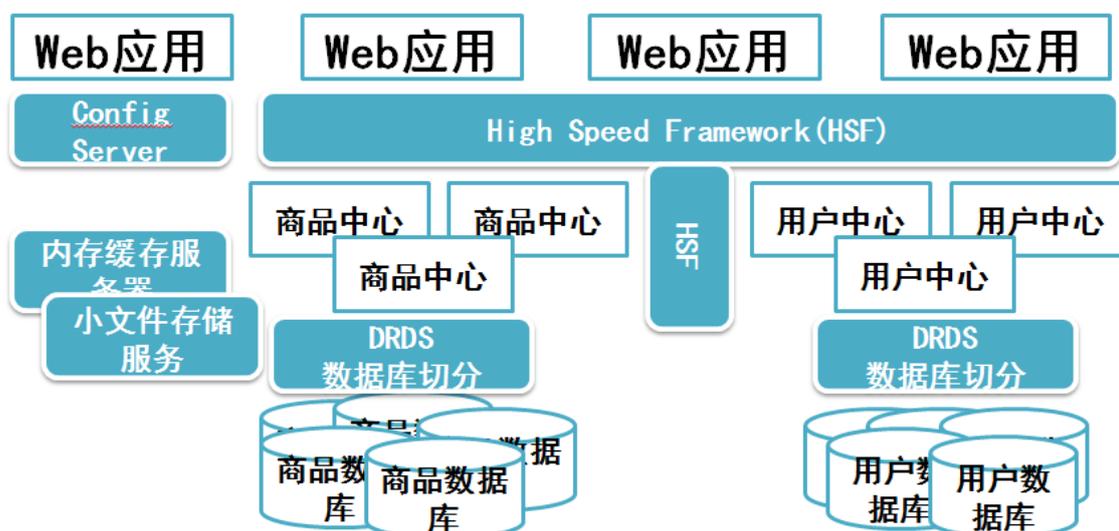
- 分布式应用服务化
- 应用生命周期管理
- 实时日志
- 容器诊室
- 服务列表
- 链路分析
- 容量规划
- 限流降级
- 基础监控
- 应用监控
- 报警设置
- 账号管理

## 5 产品功能详细介绍

### 5.1 分布式应用服务化

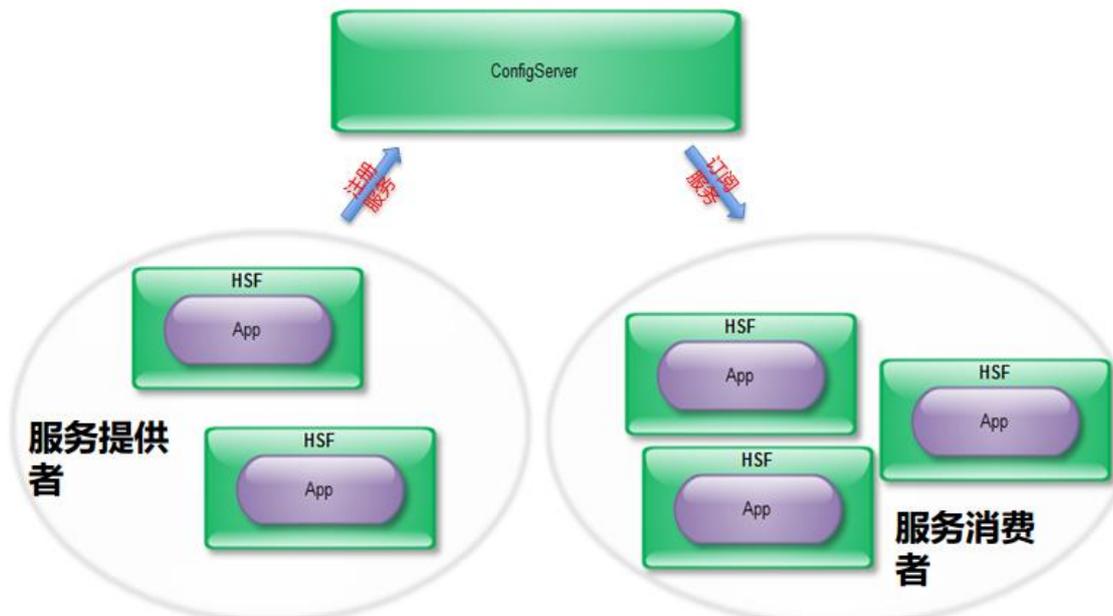
#### 5.1.1 分布式应用服务化

将集中式系统改造成分布式系统的过程中，需要面临的第一个问题就是服务化。在服务化的过程中，首先需要应用进行业务逻辑的整合，把一个庞大的系统切分成以服务为粒度的多个子模块，从而垂直拆分业务。通常将业务功能点以服务的方式对外暴露出高内聚低耦合的接口，并将这些功能切分由专门的团队进行维护。如下图所示：



#### 5.1.2 服务框架整体结构

在 EDAS 产品中，服务化框架的整体结构如下所示：



### 5.1.3 发布 HSF 服务

发布 HSF 服务的配置示例为：

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:hsf="http://www.taobao.com/hsf"
  xmlns="http://www.springframework.org/schema/beans"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
    http://www.taobao.com/hsf
    http://www.taobao.com/hsf/hsf.xsd" default-autowire="byName">
  <bean id="itemService" class="com.alibaba.edas.carshop.itemcenter.ItemServiceImpl" />
  <!-- 提供一个服务示例 -->
  <hsf:provider id="itemServiceProvider" interface="com.alibaba.edas.carshop.itemcenter.ItemService"
    ref="itemService" version="for-test" group="your-namespace">
  </hsf:provider>
</beans>
```

### 5.1.4 消费 HSF 服务

消费 HSF 服务的配置示例为：

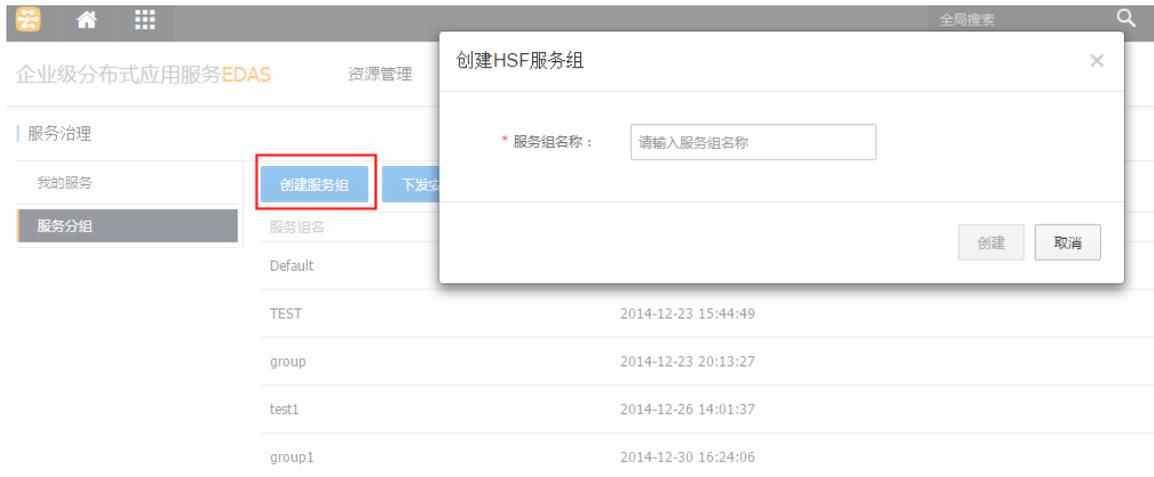
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:hsf="http://www.taobao.com/hsf"
  xmlns="http://www.springframework.org/schema/beans"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
```

```

    http://www.taobao.com/hsf
    http://www.taobao.com/hsf/hsf.xsd" default-autowire="byName">
<!-- 消费一个服务示例 -->
<hsf:consumer id="item" interface="com.alibaba.edas.carshop.itemcenter.ItemService"
version="for-test" group="your-namespac">
</hsf:consumer>
</beans>

```

注：发布服务的分组，以及消费服务的分组要在同一个分组，才能调通，配置中的 testHSFGroup 就是的分组名，这个分组名在 EDAS 平台上是有权限的，不能随意配置，需要在控制台上创建：



## 5.2 EDAS Session Framework

EDAS 提供应用 session 信息的集中管理,我们可以使用阿里云 OCS 服务来存储用户 session 信息,通过 EDAS Session Framework 可以自动处理用户 session 信息的 ocs 读取和写入.EDAS Session Framework 使用方法如下:

### 5.2.1 引入 jar 包依赖

```

<dependency>
    <groupId>com.taobao.middleware</groupId>
    <artifactId>edas-session</artifactId>
    <version>1.0-SNAPSHOT</version>
</dependency>

```

### 5.2.2 Webx MVC 框架注意事项

Webx 的应用 ,需要注释掉 webx.xml 中的 session,因为如果保留这个 session ,webx mvc 框架不会使用 EDAS Session

```
<services:request-contexts xmlns="http://www.alibaba.com/schema/services/request-contexts">
    <basic/>
    <buffered/>
    <lazy-commit/>
    <parser/>
    <set-locale defaultLocale="zh_CN" defaultCharset="UTF-8"/>
    <!-- <session>
        <id>
            <cookie path="/" maxAge="0" httpOnly="true"/>
        </id>
        <stores>
            <session-stores:cookie-store id="temporaryCookie">
                <session-stores:cookie name="tmp"/>
            </session-stores:cookie-store>
        </stores>
        <store-mappings>
            <match name="*" store="temporaryCookie"/>
        </store-mappings>
    </session> -->
</services:request-contexts>
```

### 5.2.3 添加 Filter

```
<filter>
    <filter-name>sessionFilter</filter-name>
    <filter-class>com.taobao.edas.session.TaobaoSessionFilter</filter-class>
    <init-param>
        <param-name>OCSHost</param-name>
        <param-value>${OCSHost}</param-value> <!--OCS 控制台上的“内网地址”-->
    </init-param>
    <init-param>
        <param-name>OCSPort</param-name>
```

```

        <param-value>${OCSPort}</param-value> <!-- OCS 实例的端口-->
    </init-param>
    <!-- 如果在 OCS 控制台上设置了免密码访问，OCSUsername 和 OCSPassword 两部分可以
删除-->
    <init-param>
        <param-name>OCSUsername</param-name>
        <param-value>${OCSUsername}</param-value> <!-- OCS 控制台上的“访问账号
“-->
    </init-param>
        <init-param>
            <param-name>OCSPassword</param-name>
            <param-value>${OCSPassword}</param-value> <!-- 把 OCS 邮件或短信中提供
的“密码”使用 Base64 加密后的字符串。 ->
        </init-param>
    </filter>

```

注：使用 Base64 加密 ocs 访问密码的示例代码

```

String ocsPassword = "plainWords";
byte[] encoded = org.apache.commons.codec.binary.Base64.encodeBase64(ocsPas
sword.getBytes());
String result = new String(encoded);

```

## 5.2.4 添加 Filter Mapping

```

<filter-mapping>
    <filter-name>sessionFilter</filter-name>
    <url-pattern>*.htm</url-pattern>
</filter-mapping>
<filter-mapping>
    <filter-name>sessionFilter</filter-name>
    <url-pattern>*.html</url-pattern>
</filter-mapping>
<filter-mapping>

```

```
<filter-name>sessionFilter</filter-name>
<url-pattern>*.do</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>sessionFilter</filter-name>
  <url-pattern>*.jsp</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>sessionFilter</filter-name>
  <url-pattern>*. *</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>sessionFilter</filter-name>
  <url-pattern>*</url-pattern>
</filter-mapping>
```

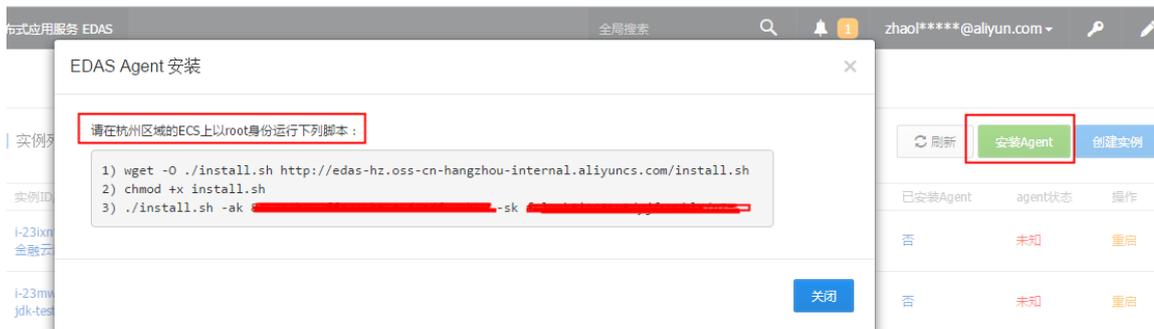
## 5.3 EDAS Agent 的安装

EDAS 会自动同步用户的 ECS 资源进入到 EDAS 中,当您登入 EDAS 控制台时,您在资源管理页面就可以看到.但如果您还没有在这些 ECS 机器上安装 EDAS Agent,那您创建应用或者扩容应用将无法看到资源列表.

EDAS 的 Agent 的安装方法,点击资源管理导航条,点击云服务器 ECS,选择不同的 Region,点击安装 Agent

注:

- a. 不同的 Region,有不同的脚本下载地址
- b. 脚本要用 root 帐号执行
- c. 安装脚本默认是装 java 6,如果您的应用需要 java 7 or java 8,请添加参数即可  
`./install.sh -ak ak -sk sk -java 7`
- d. 如果您想升级您本地已安装的 edas agent,也可以再次执行这个安装脚本



## 5.4 应用生命周期管理

生命周期管理 EDAS 产品体系中非常重要的一部分，用于基于该功能能够完成对一个应用在发布与运行过程的全面管理，包括：应用的创建、部署、启动、回滚、扩容和停止下线等。

### 5.4.1 创建应用

进入“应用列表”界面，如图所示：

企业级分布式应用服务EDAS						
资源管理	应用管理	服务市场	链路分析	账号管理		
应用列表						创建应用
应用名称	所在区域	负责人	最近运行时间	运行状态	运行实例/部署实例	操作
market-provider	北京	zhaolinjnu	7天22小时21分钟	运行中	1/1	管理
shihuaitem	北京	zhaolinjnu	35天21小时11分钟	运行中	1/1	管理
UDP	青岛	zhaolinjnu	17小时51分钟	运行中	1/1	管理

点击“创建应用”按钮，出现应用创建对话框，如下图所示：

创建应用
✕

**\* 应用运行环境:** EDAS-Container 2.3

**\* 应用名:** 请输入应用名称

**\* 应用所在区域:** 北京

**应用健康检查 <sup>?</sup>:** http://127.0.0.1:8080/healthCheck.htm

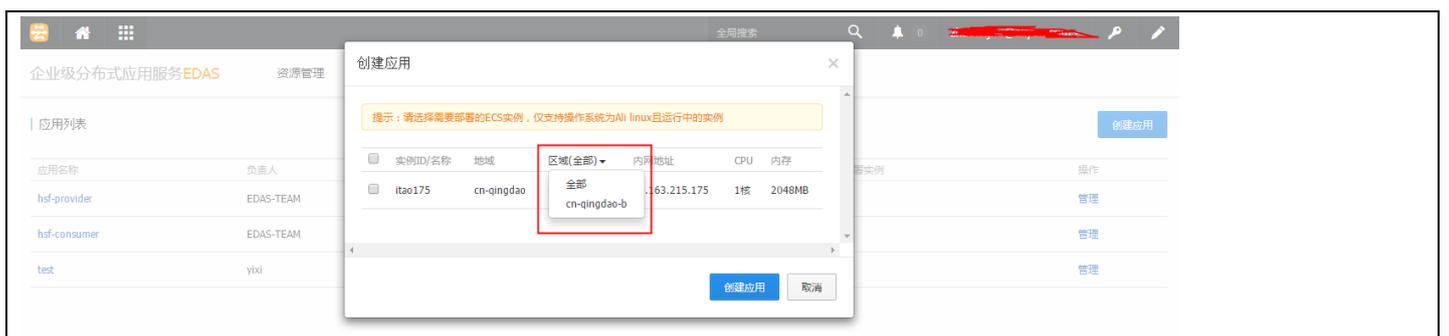
**备注:** 限制在256字以内

下一步
取消

**说明：**

1. 应用运行环境：是指 EDAS 底层提供的为 Java Web 应用运行的容器环境，第一次使用推荐选择最新版本即可。
2. 应用名：应用的名字，通常是一个英文字符串。
3. 应用所在区域：选择该应用后续需要部署的区域，请在北京、青岛、杭州和深圳中选取一个
5. 应用健康检查：该项是一个 URL 地址，EDAS 后台会通过访问该 URL 来实现对每个应用的健康检查，用户请根据各自应用实际情况给出一个可访问的 URL 地址。
6. 备注: 应用描述

完成应用信息的填写后，点击“下一步”按钮，进入用户 ECS 实例选取界面，如下图所示：



在该界面中,可以根据此 Region 的多个可用区进行过滤选择,方便客户应用同城多个可用区的部署支持,勾选 ECS 实例后,点击“创建应用”按钮完成应用的创建。此时“应用管理”界面中就会出现刚刚创建的应用,这个应用下的每台机器上会自动初始化 Java 环境,以及 EDAS 底层容器环境(包括 Tomcat 运行容器和中间件技术栈)。现在,你已经创建好一个应用,如下图所示:

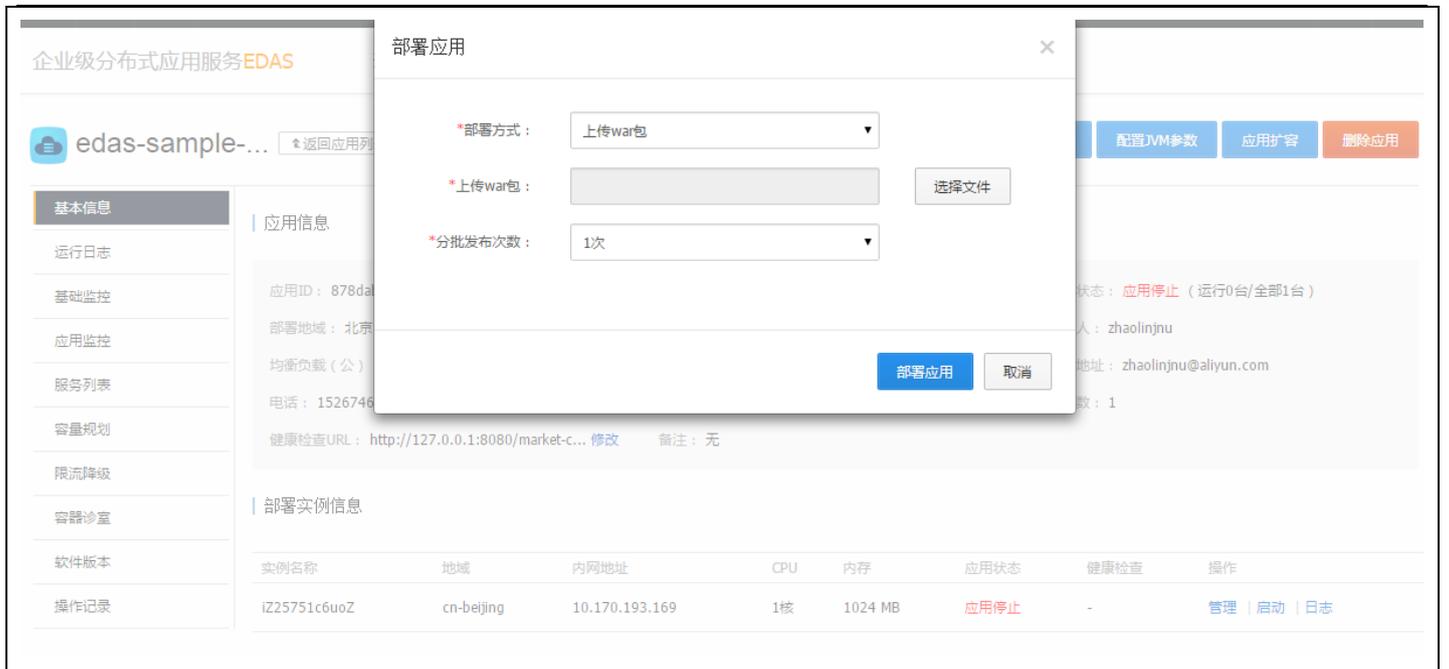
应用名称	所在区域	负责人	最近运行时间	运行状态	运行实例/部署实例	操作
market-provider	北京	zhaolinjnu	7天22小时21分钟	运行中	1/1	管理
shihuaitem	北京	zhaolinjnu	35天21小时11分钟	运行中	1/1	管理
UDP	青岛	zhaolinjnu	17小时51分钟	运行中	1/1	管理
edas-sample-app		zhaolinjnu	0小时0分钟	应用停止	0/1	管理

## 5.4.2 部署应用

点击“应用管理”界面中对应应用的“管理”按钮,进入应用管理界面,如下图所示:

实例名称	地域	内网地址	CPU	内存	应用状态	健康检查	操作
iZ25751c6uoZ	cn-beijing	10.170.193.169	1核	1024 MB	应用停止	-	管理   启动   日志

点击“部署应用”,出现应用 WAR 包上传界面,如下图所示:

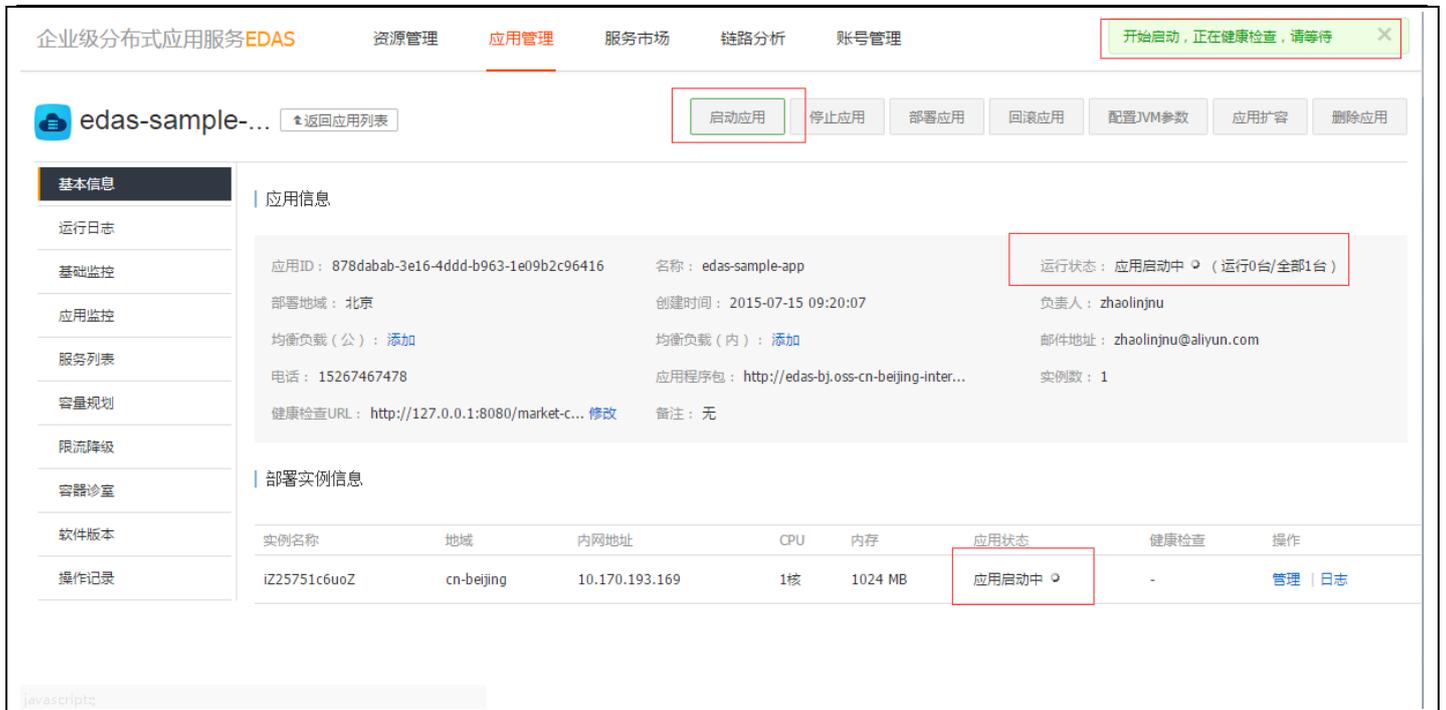


点击“上传文件”按钮，进行 WAR 上传，最后点击“部署应用”按钮来进行应用的部署，如下图所示：



### 5.4.3 启动应用

完成应用部署之后，就可以启动应用，点击“启动应用”按钮，EDAS 就会为您启动应用，如下图所示：

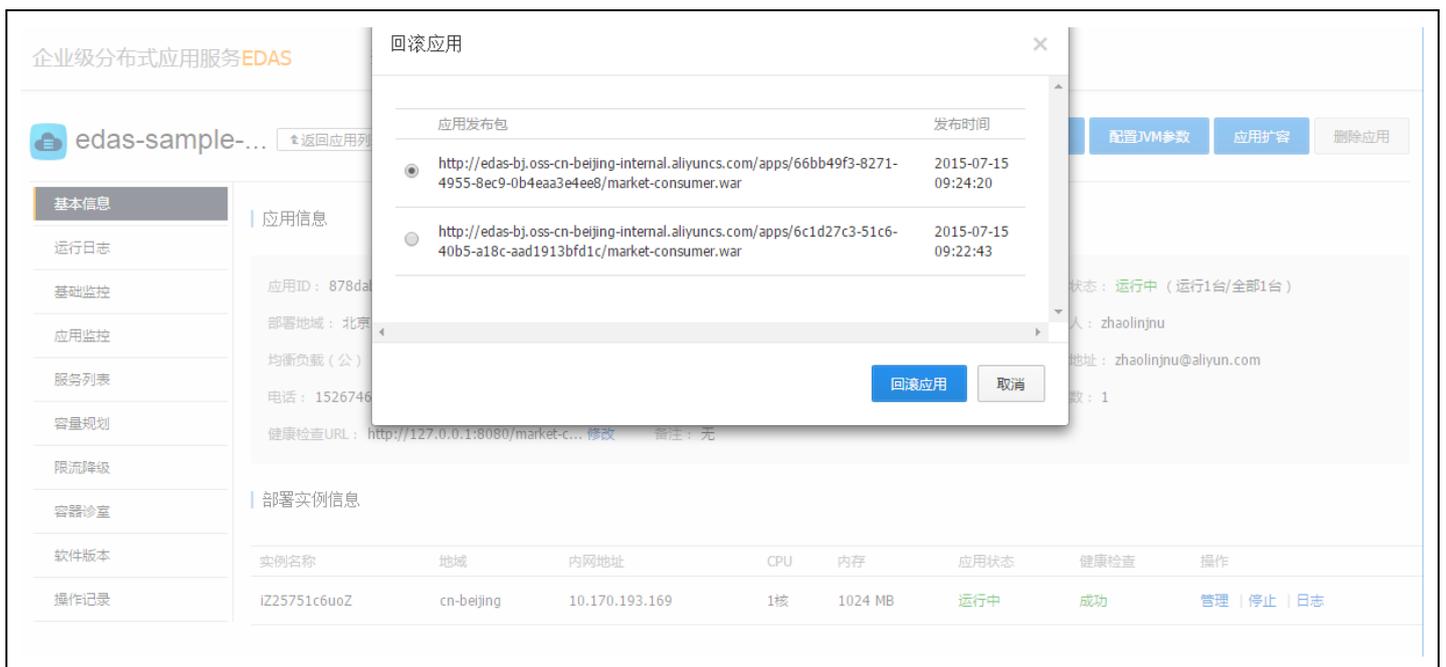


从上图中可以看出一些启动应用过程中的状态变化，当应用启动完毕，就可以看到对应的运行状态都变成“运行中”。

至此，完成应用的部署与启动。

#### 5.4.4 回滚应用

如果用户在应用发布之后发现需要进行应用的回滚，可以使用 EDAS 控制台的“回滚应用”功能。具体的，点击“回滚应用”按钮，出现应用发布包版本选择界面，如下图所示：



从中选择出本次回滚的应用包版本，最后点击“回滚应用”即可完成。

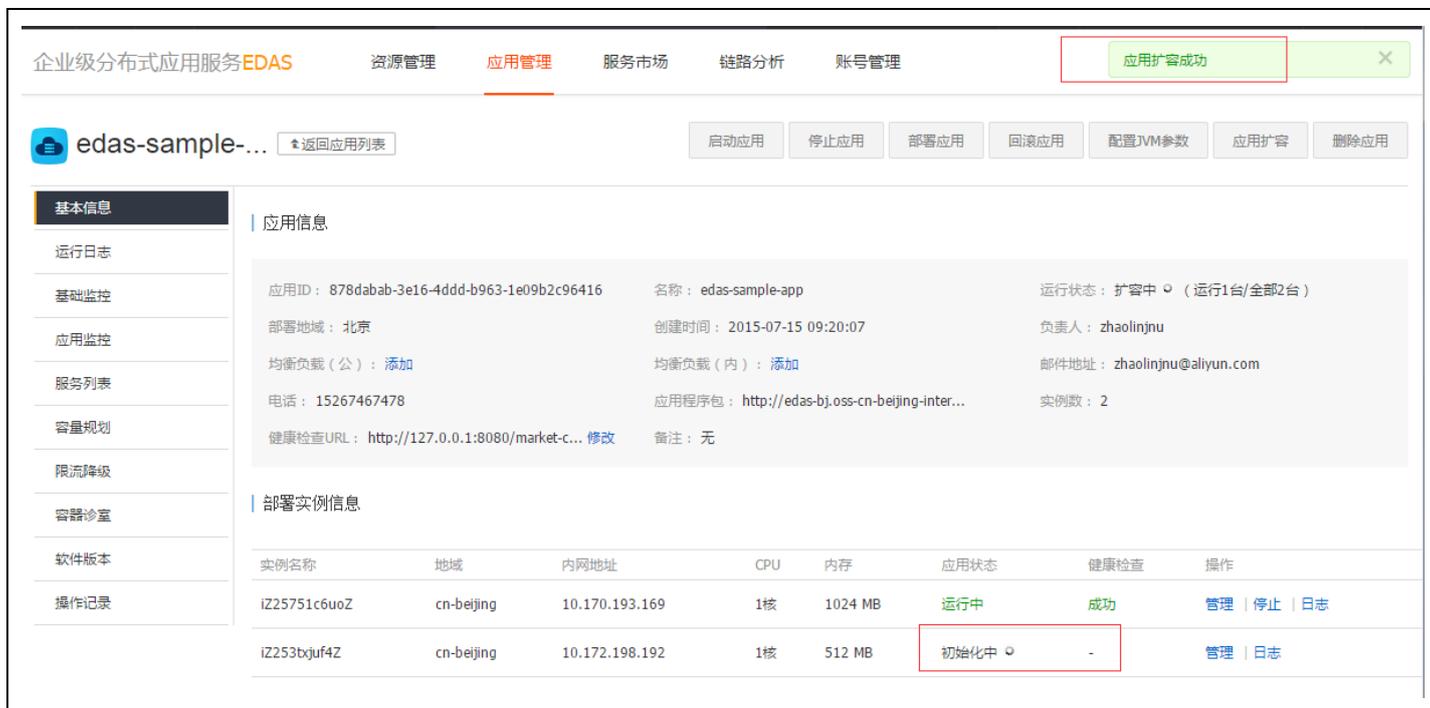
## 5.4.5 应用扩容

假如用于需要使用多台机器来同时部署同一个应用，可以使用“应用扩容”功能完成。

点击应用管理界面的“应用扩容”按钮，出现应用扩容界面，如下图所示：



在该操作界面上，选取应用扩容的目标 ECS 机器，点击“扩容”按钮，完成应用扩容，如下图所示：



## 5.4.6 停止应用

如果需要停止某一个应用，点击“停止应用”按钮即可，如下图所示。

企业级分布式应用服务EDAS 资源管理 **应用管理** 服务市场 链路分析 账号管理

停止应用成功

edas-sample-... 返回应用列表

启动应用 **停止应用** 部署应用 回滚应用 配置JVM参数 应用扩容 删除应用

**基本信息**

应用信息

应用ID: 878dabab-3e16-4ddd-b963-1e09b2c96416 名称: edas-sample-app 运行状态: 应用正在停止 (运行0台/全部2台)  
 部署地域: 北京 创建时间: 2015-07-15 09:20:07 负责人: zhaolinjnu  
 均衡负载(公): 添加 均衡负载(内): 添加 邮件地址: zhaolinjnu@aliyun.com  
 电话: 15267467478 应用程序包: http://edas-bj.oss-cn-beijing-inter... 实例数: 2  
 健康检查URL: http://127.0.0.1:8080/market-c... 修改 备注: 无

部署实例信息

实例名称	地域	内网地址	CPU	内存	应用状态	健康检查	操作
iZ25751c6uoZ	cn-beijing	10.170.193.169	1核	1024 MB	应用正在停止	-	管理   日志
iZ253tbjuf4Z	cn-beijing	10.172.198.192	1核	512 MB	应用正在停止	-	管理   日志

## 5.5 实时日志

在 EDAS 里查看一个应用的基本详情，左侧的菜单栏里有实时日志的功能，这个功能可以帮助开发者查看这个应用集群的任意一台机器上的日志文件的最新日志，帮助开发者快速诊断问题。

企业级分布式应用服务EDAS 资源管理 **应用管理** 服务市场 链路分析 账号管理

edas-sample-... 返回应用列表

**运行日志**

ECS实例ID/名称/IP: i-25751c6uo/iZ25751c6uoZ/10.170.193.169 应用路径: 尚未设置 设置路径

实例IP	文件名称	大小	时间	操作
10.170.193.169	/home/admin/taobao-tomcat-7.0.59/logs/catalina.log.2015-07-15	0.02 MB	2015-07-15 09:23:11	在线查看
10.170.193.169	/home/admin/taobao-tomcat-7.0.59/logs/catalina.out	0.06 MB	2015-07-15 09:23:11	在线查看
10.170.193.169	/home/admin/taobao-tomcat-7.0.59/logs/localhost.log.2015-07-15	0 MB	2015-07-15 09:23:11	在线查看

点击在线查看，就可以看这个日志文件的最新日志。

## 5.6 容器诊室

客户的应用是部署，并运行在我们的 EDAS-Container 容器中，我们的应用容器提供几个诊断功能：

- 内存变化

如下图所示，可以查看任何一台应用服务器堆内存的变化的历史曲线，方便诊断 Java 内存问题



- 类加载

Java 应用会依赖很多的 jar 包，这些 jar 包，哪些被应用容器加载，通过此功能，会展示出来。这个功能将帮助用户诊断类冲突问题

- 线程

列举容器里面的线程，各个线程的状态，以及线程名

- 连接器

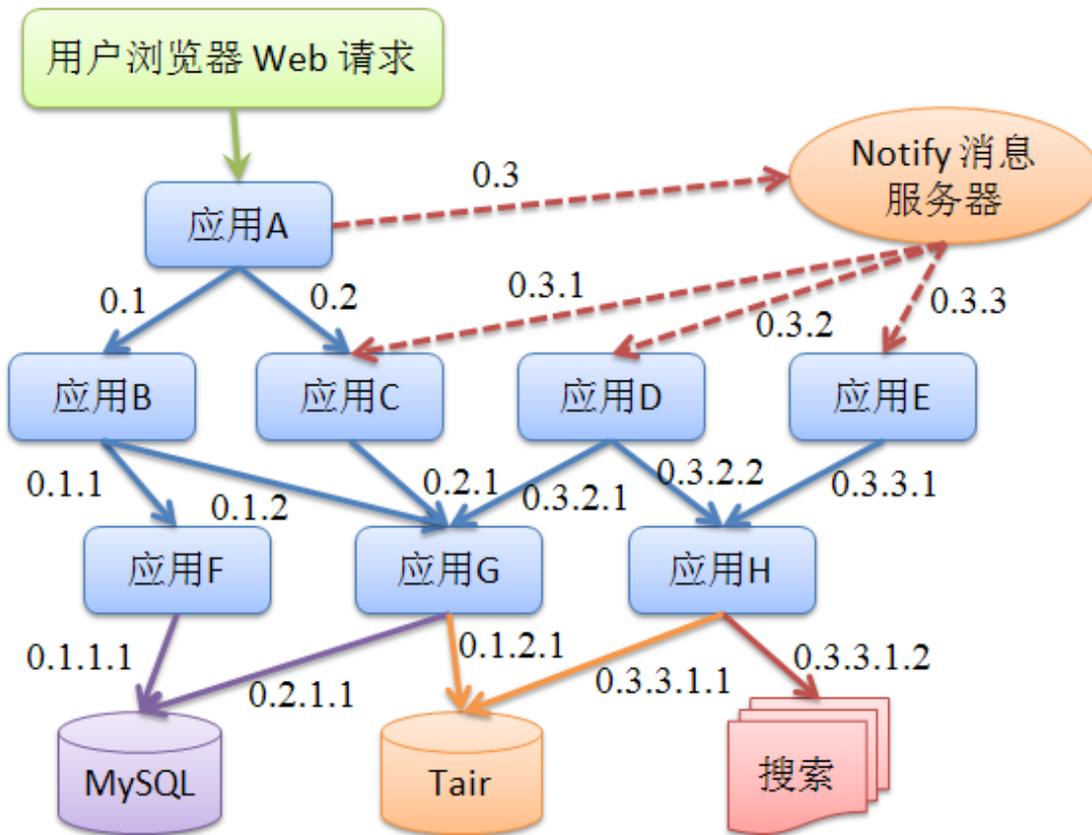
每个连接器当前处理的请求的一些数据

## 5.7 服务列表

每个应用发布了哪些 HSF 服务，消费了哪些 HSF 服务，可以通过此功能可以查看。此应用发布的服务，可以对其中一个比较重要的服务，不需要修改应用代码，在 EDAS 上进行动态分组配置，比如服务组 A，服务组 B，这样就可以控制不同的 consumer 连接不同服务组中的服务，方便隔离。

## 5.8 链路分析

在于服务化的架构体系中，对于任何一次业务上的请求，底层都会有很多远程的 HSF 服务化调用，访问数据库，收发消息，或者其它的操作，通过链路分析的功能，可以准确的描绘出上层一次用户请求，所经历的所有系统，服务，调用所花的时间，是否有错误。



使用方法如下：

第一步：进入 EDAS 控制台。



## 第二步：进入链路分析

企业级分布式应用服务

调用链查询

应用名称：请输入应用名称进行搜索 机器IP：请输入IP进行搜索 服务名称：请输入服务名称进行搜索 调用类型：全部

调用出错过滤：否 调用超时： ms 时间：2016-02-26 13:28:57 至 2016-02-26 13:58:57 搜索

TraceId	开始时间	操作
---------	------	----

进入调用链查询，在输入框中输入应用名，并指定时间段，即可查询这段时间内该应用所有的调用情况。如下所示：

企业级分布式应用服务 | 调用链查询

应用名称:  机器IP:  服务名称:  调用类型:

调用出错过滤:  调用超时:  ms 时间:  至

TraceId	开始时间	操作
65c84c7f14564663866714775d0356	2016-02-26 13:59:46	<a href="#">查看详情</a>
65c84c7f14564663864044774d0356	2016-02-26 13:59:46	<a href="#">查看详情</a>
65c84c7f14564663761554773d0356	2016-02-26 13:59:36	<a href="#">查看详情</a>
65c84c7f14564663759494772d0356	2016-02-26 13:59:35	<a href="#">查看详情</a>
65c84c7f14564663657064771d0356	2016-02-26 13:59:25	<a href="#">查看详情</a>
65c84c7f14564663654974770d0356	2016-02-26 13:59:25	<a href="#">查看详情</a>

### 第三步 点击查看调用详情

应用服务 | 调用链查询

应用名称:  机器IP:  服务名称:  调用类型:

调用出错过滤:  调用超时:  ms 时间:  至

TraceId	开始时间	操作
65c84c7f14564663866714775d0356	2016-02-26 13:59:46	<a href="#">查看详情</a>
65c84c7f14564663864044774d0356	2016-02-26 13:59:46	<a href="#">查看详情</a>
65c84c7f14564663761554773d0356	2016-02-26 13:59:36	<a href="#">查看详情</a>
65c84c7f14564663759494772d0356	2016-02-26 13:59:35	<a href="#">查看详情</a>

点击 查看详情，即可看到调用详情，如下所示：

企业级分布式应用服务 | 调用链详情

调用链TraceId:

应用名	IP	类型	状态	大小	服务/方法	时间轴	耗时
姬望测试专用web (勿删)	10.172.250.40	HTTP	OK	0B	http://localhost:8080/portal/echo.htm		398ms
姬望测试专用service (勿删)	10.171.112.16	HSF	OK	57B	com.alibaba.edas.test.EdasDemoService@echo~E		3ms

## 5.9 容量规划

一个应用系统，所提供的 HSF 服务，在当前的应用机器集群下，最大可以支撑多少的 QPS 调用，通过 EDAS 容量规划的自动压测功能，即可以测算出当前系统的容量。这个将方便我们对未来流量增长情况下，我们对应用所需机器数的一个预估，这将变得更加科学准确。

## 5.10 限流降级

一个应用系统，会提供很多的 HSF 服务，对于这些 HSF 服务，可以配置限流降级规则，限制其它应用，对此服务的调用，可以从 QPS，以及线程的两个维度来进行设置。这个功能，将可以帮助用户，在应对流量高峰时，系统能以最大的支撑能力平稳运行。

### 1. 添加限流规则:

### 添加限流规则 ×

---

当前应用：

\* 需要限流的接口：

\* 需要限流的方法：

\* 被限流的应用：

\* 限流粒度：

\* 限流阈值：

---

### 2. 在应用当中添加如下的依赖:

```
<dependency>
  <groupId>cglib</groupId>
  <artifactId>cglib-nodep</artifactId>
  <version>2.2(或者更新)</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.aop</artifactId>
```

```
<version>2.5.6(或者更新)</version>
</dependency>
```

### 3. 添加接入方式:

- a. 给 hsf provider 限流, 请引入以下 Bean

```
<bean id="customEditorConfigurer"
      class="org.springframework.beans.factory.config.CustomEditorConfigurer">
  <property name="propertyEditorRegistrars">
    <list>
      <bean class="com.taobao.csp.sentinel.entrypoint.entrance.HSFSpringProviderBeanRegistrar"/>
    </list>
  </property>
</bean>

<bean id="traceMethodInterceptor" class="com.taobao.csp.sentinel.entrypoint.proxy.TraceMethodInterceptor"/>

<bean id="traceProxyTypeHolder" class="com.taobao.csp.sentinel.entrypoint.ProxyTypeHolder">
  <property name="beanTypes">
    <list>
      <value>com.taobao.hsf.app.spring.util.HSFSpringConsumerBean</value>
    </list>
  </property>
</bean>

<bean id="traceBeanTypeAutoProxyCreator" class="com.taobao.csp.sentinel.entrypoint.BeanTypeAutoProxyCreator">
  <property name="proxyTypeHolder" ref="traceProxyTypeHolder"/>
  <property name="interceptorNames">
    <list>
      <value>traceMethodInterceptor</value>
    </list>
  </property>
```

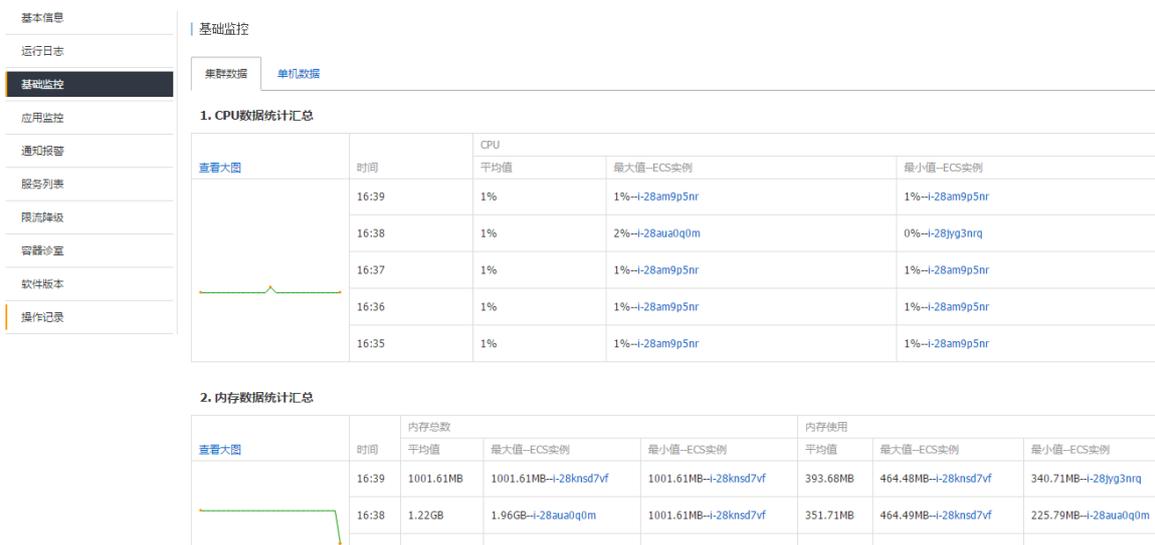
```
</bean>
```

b. 为 web 限流，请在 web.xml 里添加下列代码(对 URL 的限流,当前在 EDAS 里面暂不支持):

```
<filter>
  <filter-name>CommonFilter</filter-name>
  <filter-class>com.taobao.csp.sentinel.entrypoint.entrance.CommonFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>CommonFilter</filter-name>
  <url-pattern>*.htm</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>CommonFilter</filter-name>
  <url-pattern>*.jspx</url-pattern>
</filter-mapping>
```

## 5.11 基础监控

应用的基础监控模块，将会帮助用户，以应用的维度，查看集群机器的平均 load,平均网络读写，平均磁盘读写等指标，以及发现每一个指标，哪台机器出现了最高值。我们也提供单机的监控指标视图，方便客户了解单机各指标的历史发展变化。



---

## 5.12 应用监控

应用监控模块，会提供应用全方位的监控信息，主要包括的核心功能点如下：

- 系统概要

以下几个数据概要

- Http 入口

如何对您应用当中使用的 URL 进行跟踪,只需要在您的 web 应用的 web.xml 配置文件中添加如下的配置:

```
<filter>
  <filter-name>EagleEyeFilter</filter-name>
  <filter-class>com.taobao.eagleeye.EagleEyeFilter</filter-class>
  <init-param>
    <param-name>useLocalIp</param-name>
    <param-value>true</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>EagleEyeFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

- 提供的 HSF 服务(我提供的 HSF 服务)

Hsf 数据,您无须做任何配置,将自动采集

- HSF 服务调用来源(谁调了我?)

Hsf 数据,您无须做任何配置,将自动采集

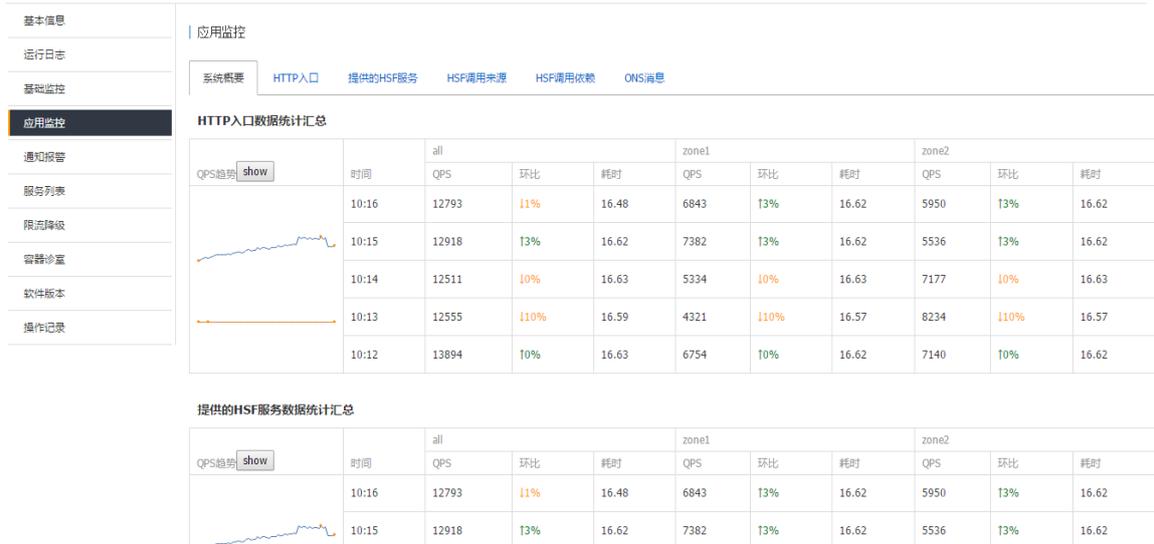
- HSF 服务调用依赖(我调了谁?)

Hsf 服务调用依赖数据,您无须做任何配置,将自动采集

- ONS 消息发送量

暂不支持

- DRDS 数据库访问量



## 5.13 帐号管理

EDAS 作为一个企业级的 PAAS 平台，将会为客户提供主子帐号功能，以及细粒度的角色，权限控制，满足企业开发人员，以及运维人员分工协作的需求。

用户管理 [绑定用户](#)

用户  [搜索](#)

用户	联系人	角色	授权应用	操作
james.tj@alibaba-inc.com	james.tj	什么都有的权限	hsf-provider hsf-consumer test	<a href="#">管理角色</a>   <a href="#">授权应用</a>   <a href="#">解绑</a>
nyan_bjut@aliyun.com	nyan_bju	--	--	<a href="#">管理角色</a>   <a href="#">授权应用</a>   <a href="#">解绑</a>

## 5.14 资源组

在资源管理功能模块里面,可以创建资源组,资源组里可以添加 ECS,SLB 等资源.然后在帐号管理里面,可以给予帐号绑定所能使用的资源组,这样子帐号在创建应用,或者扩容应用时,只能看到授权使用的资源.注:主帐号,可以看到所有资源.

企业级分布式应用服务 EDAS 全局搜索

企业级分布式应用服务EDAS 资源管理

云服务器ECS 负载均衡SLB 资源组管理

实例列表 杭州

资源组名称 描述

### 创建资源组

\* 资源组名称：  
长度限制为2-32个字符。

资源组描述：  
您最多可以输入64个字符。

资源组创建完成后，请在资源组管理页面为该资源组绑定ECS、SLB实例。

确定 关闭