ALIBABA CLOUD

阿里云

应用高可用服务

常见问题

文档版本：20220126

阿里云

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**。
粗体	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 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.流量防护

1.1. 应用防护规则常见问题

本文列举了应用防护规则常见的问题。

1. **请求链路页面添加规则是给单台机器添加还是给所有机器都添加？**

   添加规则会推送到所有机器。

2. **流控规则中的来源应用是什么意思？**

   Sentinel支持按调用来源限流。流控规则中来源应用（针对应用）指的是调用该资源的调用方标识，例如在Dubbo中就对应Dubbo Consumer的应用名称。默认来源应用设为 `default` ，代表不区分来源应用。

3. **流控规则中的每种流控模式是什么意思？**

   - 直接：直接按照当前资源的**调用来源**进行限流，若来源为 `default` 则不区分调用来源。
   - 关联：当两个资源之间具有资源争抢或者依赖关系的时候，这两个资源便具有了关联。关联限流会根据当前资源的**关联资源**进行限流。

     比如read_db和write_db这两个资源分别代表数据库读写，我们可以给read_db设置限流规则来达到写优先的目的：设置**关联资源**为write_db。这样当写库操作过于频繁时，读数据的请求会被限流。
   - 链路：根据调用链路入口限流。需要在规则中配置**入口资源**，即该调用链路入口的上下文名称。

4. **流控规则中的每种流控方式是什么意思？**

   - 快速失败：达到阈值时，立即拦截请求。
   - Warm Up：当流量突然增大的时候，希望系统从空闲状态到繁忙状态的切换的时间长一些，即如果系统在此之前长期处于空闲的状态，希望处理请求的速率缓慢增加，经过预期的时间以后，到达系统处理请求速率的设定值；默认会从配置QPS阈值的1/3开始慢慢往上增加QPS。
   - 排队等待：请求匀速通过，允许排队等待，通常用于消息队列削峰填谷等场景。比如配置了QPS为5，则代表请求每200 ms才能通过一个，多出的请求将排队等待通过。超时时间代表最大排队时间，超出最大排队时间的请求将会直接被拒绝。**注意排队等待模式下QPS不要超过1000（请求间隔1 ms）。**

5. **降级规则中的RT模式、异常比例模式是什么意思？**

   Sentinel熔断降级支持两种策略：

   - RT：按照**秒级的平均响应时间**进行降级，单位是毫秒。如果持续进入5个请求，对应的平均RT都持续超过降级阈值，则会被自动降级。
   - 异常比例：按照**秒级的异常比例**进行降级。当资源的每秒异常总数占通过量的比值超过阈值之后，资源进入降级状态。

   资源进入降级状态后，在配置的降级窗口时间内，请求都会快速失败。

6. **什么是系统保护规则？**

   系统保护规则可以从系统指标维度（入口QPS、RT、线程数、Load等）来进行流量控制，让系统尽可能跑在最大吞吐量水位的同时保证系统整体的稳定性。系统保护规则是整体维度的而不是资源维度的，并且仅对入口流量生效。入口流量指的是进入应用的流量，比如Web服务或Dubbo服务端接收的请求，都属于入口流量。

   系统规则支持的种类有以下几种：

- **系统Load（load1）**：按照系统load1以及实时的吞吐量进行流量控制。
- **总体平均RT**：当单台机器上所有入口流量的平均RT达到阈值即触发系统保护。
- **总体QPS**：当单台机器上所有入口流量的QPS达到阈值即触发系统保护。
- **总体线程数**：当单台机器上所有入口流量的并发线程数达到阈值即触发系统保护。

对于一个应用来说，每种相同的系统保护规则最多只能存在一条。

1.2. 应用防护页面看不到应用或资源？

本文介绍接入Agent后找不到应用的解决办法以及接入应用后无法看到接口数据的原因。

接入Agent之后，AHAS控制台看不到应用？

请参考以下步骤检查。

1. 确保选择了正确的地域（Region）。切换地域（Region）步骤请参见如何切换地域（Region）。
2. 确保引入了相应的依赖，并进行了正确的配置。请参见应用防护接入。
3. 确保本机时间正确，程序所在的机器时间要跟当地标准时间一致。
4. 检查应用是否是EDAS应用，如果是EDAS应用必须用Admin账号接入Agent，否则将无法接入成功。
5. 查看日志*/home/admin/logs/csp/sentinel-record.log*是否有报错，根据报错内容进行修改。
6. 如果日志或者目录不存在，则执行 `java -version` 查看内容。

   如下图所示，如果没有输出Agent相关信息，即Agent没有接入成功，文件没有挂载上。

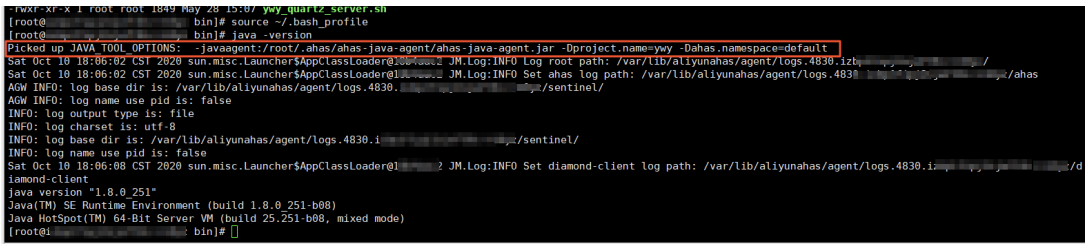
   ```
   [root@izbp1fipj2sjwfl0crrn0yz java]# java -version
   java version "1.8.0_251"
   Java(TM) SE Runtime Environment (build 1.8.0_251-b08)
   Java HotSpot(TM) 64-Bit Server VM (build 25.251-b08, mixed mode)
   [root@ z java]#
   ```

   解决方案如下：

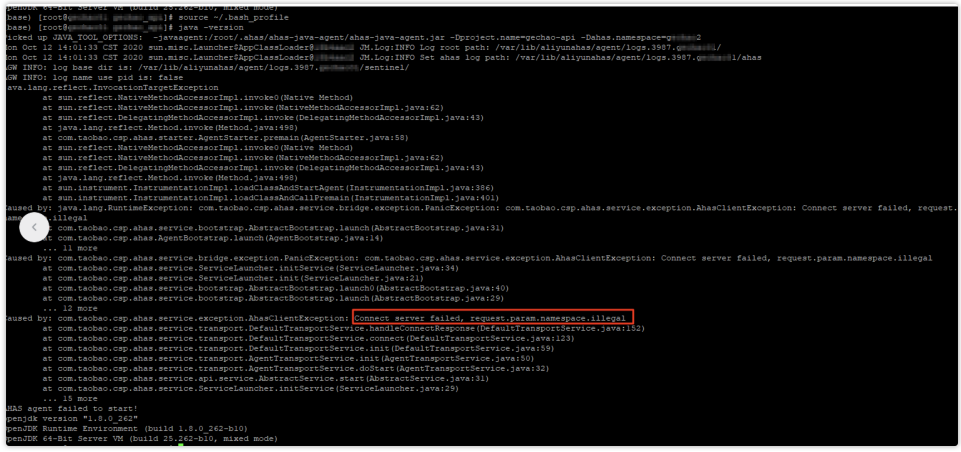
   i. 执行 `source ~/.bash_profile` 。

ii. 执行 `java -version` 。

- 若输出结果有Agent相关信息，则表示Agent接入成功，可以看到执行的Agent的内容，如下图所示。



- 若输出结果出现如下报错，则表示 `namespance` 不存在，请在AHAS控制台相应的Region下创建命令 `namespace` 下的环境或修改命令 `namespace` 为其他环境（例如default），请参见管理环境。



iii. 重启应用之后，在AHAS控制台选择对应的Region下查看，Agent即可生效。

AHAS控制台可以看到应用，但是没有数据显示？

请确保对应资源有访问量。资源要有访问量才会在请求链路页面显示。

- AHAS刚接入应用之后，应用防护页面各个应用数据显示为空，这是因为应用没有实时流量就不会有数据。此时您可以通过PTS发起少量压测流量或手动访问，例如通过浏览器访问应用即可看到请求，控制台中将显示对应的流量数据。
- 在应用防护页面中，展示的是最近5分钟的数据统计，所以可以看到近5分钟的趋势图。但接口数据展示的是实时数据，当实时数据为0时，接口详情将无数据显示。
- 请求链路页面显示的是单台机器上的所有调用链路数据，您可以通过下拉框切换机器查看不同机器的请求链路数据。

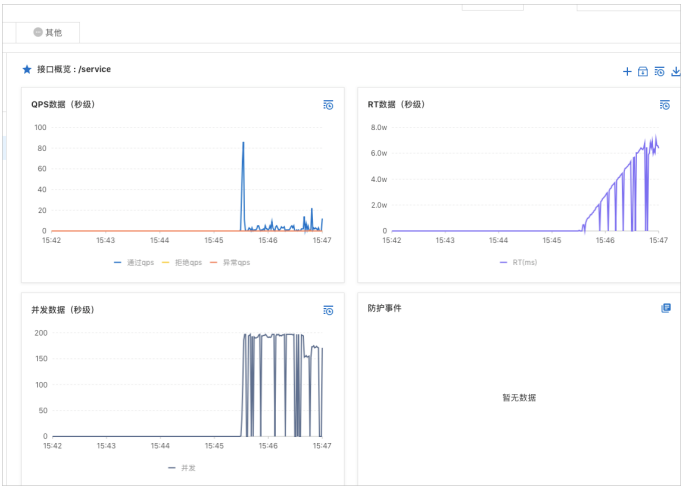
1.3. 如何理解并发、QPS和RT的关系？

本文介绍并发、QPS和RT的概念及其之间的关系。

在应用防护的数据图中，您可以看到每个接口都有以下3个指标：

- QPS：Query per Second，直接反应应用吞吐能力的指标。
- RT：Response Time，响应时间，表示处理请求快慢的指标。
- 并发：表示处理请求耗费的并发线程数。

如果处理能力弱，则响应时间长，即RT值较高。相同过来的请求，会导致并发线程数增加，QPS也会有所降低。如下图所示。



并发在不同的场景中表示不同的概念，例如以下场景：

- 压测中施压并发指并发虚拟用户数。
- SLB的并发指并发连接数。
- 服务端统计的并发指并发线程数。
- 流控规则中的并发指该资源正在执行的线程数，线程数模式按照资源的并发线程数进行流量控制。

关于压测的并发虚拟用户数和SLB的并发连接数概念，请参见FAQ。

1.4. 如何卸载应用防护的Java Agent和SDK埋点？

本文介绍如何卸载应用防护的Java Agent和SDK埋点。

背景信息

应用高可用服务AHAS会利用探针（Agent）从您的系统中采集所需的信息，包括以下两种Agent：

- 应用高可用探针（即AHAS Agent）：这是安装在操作系统上的独立进程，用来采集架构信息。使用架构感知、故障演练功能前需安装此探针。
- 应用高可用Java探针（即Java Agent）：这是针对JVM的Java探针，通过字节码增强技术进行实时监控和流量防护。如果需要使用流量防护功能，可安装此探针。

本文介绍的是如何卸载Java Agent，关于卸载AHAS Agent请参见卸载高可用探针（AHAS探针）。

卸载Java Agent

1. 在机器上使用安装时的账号执行 `sh ./ahas-agent.sh help` 。
   输出结果中会提示 `uninstall` 的用法。

   > **说明** 如果是EDAS应用，请使用admin账号执行。

2. 执行 `sh ./ahas-agent.sh uninstall` 卸载Java Agent。

```
[admin@iZbp1c          ~]$ sh ./ahas-agent.sh help

Usage: ./ahas-agent.sh COMMAND [OPTIONS]

Commands:
  install       Install latest java agent and setup ahas configuration automaticlly
  uninstall     Uninstall java agent and delete ahas configuration, you need to restart your app

Install Command Options:
  -a <value>    The application name that will show in ahas console
  -n <value>    The namespace your application belonging to
  -l <value>    The license used in public region for authentication

Examples:
  ./ahas-agent.sh install -a my-app -n default -l 1q2w3e4r5t6y
  ./ahas-agent.sh uninstall

[admin@iZbp1c          ~]$ sh  ./ahas-agent.sh uninstall
Start to uninstall ahas java agent
Removing ahas-java-agent.jar
Removing ahas.namespace
Removing ahas.license
Uninstall ahas java agent success
```

3. 重启应用。
   登录AHAS控制台，选择**流量防护 > 应用防护**，单击应用卡片，进入**应用详情**页，单击**接入节点**。在**接入节点**页面可以看到节点的健康状态为**失联**，同时QPS统计也会缺失对应的统计数据。这样代表Java Agent卸载成功。

卸载SDK埋点

1. 删除引入的依赖包。

   > 说明 引入的依赖包的参数可以参考AHAS控制台中**流量防护 > 应用防护**的**新应用接入**中的参数。

2. 删除启动参数。

   > 说明 启动参数可以参考AHAS控制台中**流量防护 > 应用防护**的**新应用接入**中的参数。

3. 重启应用。

1.5. Nginx接入Sidecar根据文档操作还是提示不兼容怎么办?

Condition

按照将Nginx接入流量防护文档中的步骤将Nginx接入流量防护操作，在CentOS 7操作系统下，Nginx版本为nginx-1.16.1，使用对应动态模块路径的*lib/centos7-nginx-1.16.1/ngx_sentinel_module.so*，但系统还是提示Nginx不兼容。如下图所示。

```
[root@ nginx]# ./sbin/nginx
nginx: [emerg] module "/opt/ahas-sentinel-sidecar-linux/lib/centos7-nginx-1.16.1/ngx_sentinel_modu
le.so" is not binary compatible in /opt/nginx/conf/nginx.conf:1
[root@ nginx]# ./sbin/nginx -V
nginx version: nginx/1.16.1
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
configure arguments: --prefix=/opt/nginx
```

Cause

Nginx自身限制编译动态模块*.so*文件时的Nginx版本和编译配置必须相同才兼容。当Nginx接入Sidecar时操作系统相同、Nginx版本相同，但编译配置不同时，也可能会不兼容Nginx。Sidecar安装包附带的*.so*版本是根据Yum EPEL仓库的Nginx打包的，如果使用从Yum EPEL安装的Nginx版本就可以使用默认提供的*.so*版本。

Remedy

操作步骤

1. 如果遇到类似不兼容的情况，请您把 `Nginx -V` （大写的V）输出的完整文本复制到用户钉钉群（群号：23196438）中，并联系我们，我们会为您单独提供一份对应的专属*.so*模块。

1.6. Nginx防护如何根据域名（host）来做流控？

本文介绍Nginx防护根据域名（host）做统计和流控的操作步骤。

Nginx防护在默认情况下，是按照API来进行统计和流控，如 `/api/login` 。Nginx大多数时候作为网关或者Proxy代理服务器转发使用，但不同的域名表示不同业务，有时需要根据不同域名来做统计和流控，可以在Nginx的HTTP配置中添加以下内容。

```
sentinel_entry $http_host
```

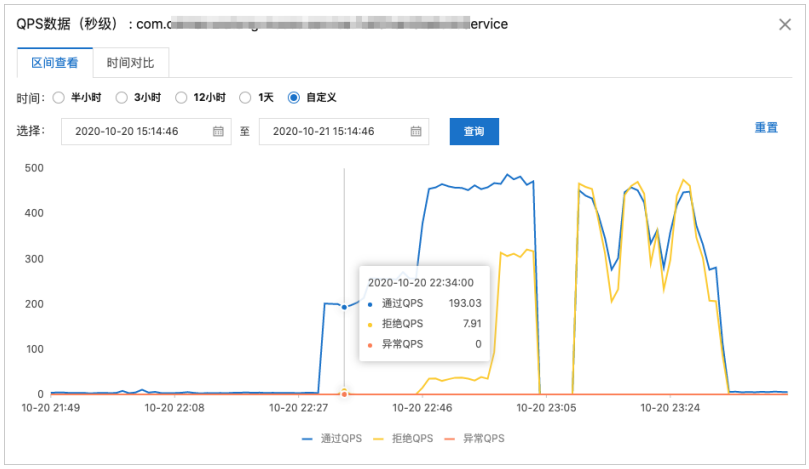
添加完配置之后，在AHAS控制台的Nginx防护接口详情中会看到对不同域名的统计，可以按照API接口一样的操作进行规则配置，详情请参见接口详情。

1.7. 应用共5个节点单机流控规则为100 QPS，为何总请求高于200 QPS时就触发了限流？

问题现象

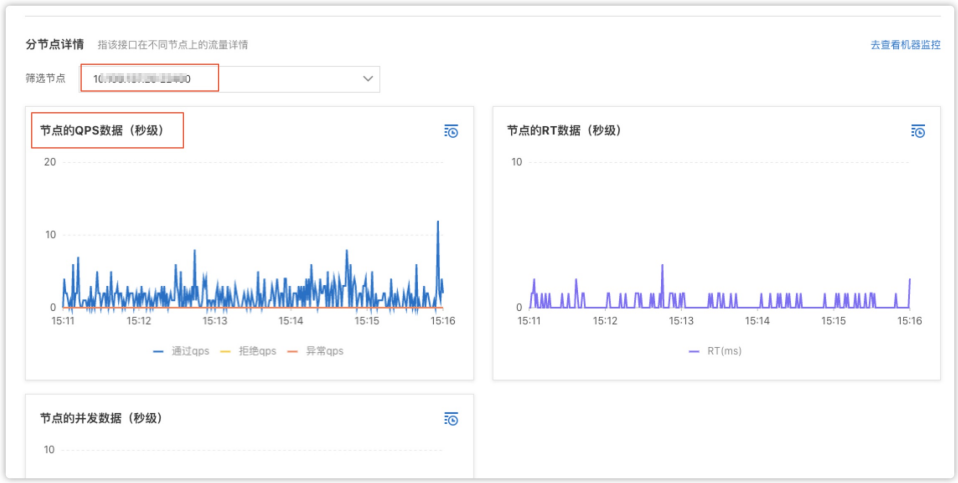
例如该应用一共接入了5个节点，流控规则配置为单机QPS达到100的时候触发限流，如下图所示。

但是当客户端总请求量高于200 QPS的时候，就发现有限流QPS，与预期设置不符。



排查方法

1. 登录AHAS控制台，在左侧导航栏选择**流量防护 > 应用防护**。
2. 单击资源卡片，进入**应用详情**页面。
3. 单击**接口详情**，搜索对应的接口，查看该接口的总QPS是否正常。
4. 在**接口详情**页面右下方的**分节点详情**区域，查看对应时间每个节点的QPS数据。



通过排查发现，虽然接入了5个节点，但总请求量高于200 QPS的时候，只有2个节点有流量，单机QPS超过100，故触发了单机限流。

解决方法

您有以下两种解决方式：

- 排查对应限流时间点上游的调度和日志，为何只请求到了某两个节点上。原因可能是SLB的健康检查或转发有问题、上游接口的调度出现反复重试或上游应用的调度出错等。找到对应原因，解决该问题，使每个节点均有流量，这样可以正常触发单机限流规则。
- 如果您想控制整个接口的总流量，可以通过设置集群流控规则的方式，这样流控规则就不会随着节点的变化而影响整体阈值。集群流控的详情请参见配置集群流控规则。

2.故障演练

2.1. 故障演练常见问题

欢迎您反馈在使用故障演练过程中遇到的问题，目前提供以下常见问题供您参考。

单个演练活动成功状态怎么判断？

因为一个活动可能会有很多目标机器，当所有机器执行完毕之后，如果有机器没有执行成功，那么这个演练活动就会被系统判断为失败。

单个演练活动是否成功取决于节点推进方式。如果是自动推进节点，那么系统运行成功就是成功；如果是手动推进节点，无论系统运行成功与否，都以用户的二次确认结果为准，如果用户允许继续流程，则当前活动是成功的。

怎么算一个演练任务是否成功？

只有当每个阶段的所有演练活动都运行成功了，才算演练任务成功。

在什么情况下会执行恢复操作？

- 用户配置了恢复操作。
- 用户主动结束了演练或者系统正常结束演练。
- 除了恢复阶段以外的任何一个阶段只要有一个活动运行失败了，并且当前活动不需要二次确认，则会立刻终止整个演练，并执行恢复操作。

关于故障演练计费的常见问题

- 什么是故障规则下发次数？

  一次故障注入即为一次故障规则下发。例如：对 10 台 ECS 注入 CPU 满载和磁盘填充两种故障，则故障注入次数为 10（ECS数）×2（故障场景数）= 20次，那么故障规则下发次数即为 20 次。

- 挂载 JavaAgent 是否计费？

  仅针对故障规则下发次数进行计费，挂载 JavaAgent 为故障注入的前置准备动作，不在计费范围内。

- 故障注入失败是否计费？

  不计费。仅对下发成功的故障规则进行计费，所以执行一次演练的费用是在演练执行成功后进行计算的，会去除下发失败的故障规则。

- 故障不生效是否计费？

  计费。故障不生效的原因较多，可能是参数配置不正确或无对应的请求命中，但故障规则已成功下发，故需要计费。

- 刚刚购买的资源包为什么会被扣减次数？

  因为购买前已产生欠费，购买资源包后会先扣减所欠的次数。

- 子账号的消费是否独立计费？

  不是，与主账号合并计费，子账号的所有消费均扣减主账号余额。

2.2. 为何Pod中仍存在已恢复故障的“僵尸进程”？

在K8s环境中，下发的故障已经被恢复了，但是Pod中仍存在该故障的“僵尸进程”。本文介绍该情况可能的原因以及解决方案。

可能原因

这是因为容器中存在PID Namespace隔离。在容器中，故障演练进程的父进程是PID=1的进程，容器中的一号进程不具有进程资源回收的能力，所以导致故障演练进程被终止之后，资源没有得到回收，从而成为僵尸进程。

解决方案

通过手动共享PID Namespace解决该问题。

在Pod的YAML文件中增加 `shareProcessNamespace: true` ，从而将Pod容器中PID=1的进程改为 `/pause` 进程，如下所示。

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubernetes.io/psp: ack.privileged
  labels:
    app: myapp
  name: myapp-pod
  namespace: ahas
spec:
  shareProcessNamespace: true
  containers:
  - command:
    - sh
    - -c
    - echo Hello Kubernetes! && sleep 3600
    image: busybox
    imagePullPolicy: Always
    name: myapp-container
    resources: {}
```

注意事项

共享PID Namespace会带来以下影响：

- **容器进程不再具有PID 1。**

  在没有PID 1的情况下，一些容器镜像会拒绝启动（例如使用systemd的容器），或者拒绝执行 `kill -HUP 1` 之类的命令来通知容器进程。在具有共享进程命名空间的Pod中，如果执行 `kill -HUP 1` ，将会把消息通知到Pod沙箱（本文示例中的Pod沙箱即 `/pause` ）。

- **进程对Pod中的其他容器可见。**

包括 `/proc` 中可见的所有信息，例如作为参数或环境变量传递的密码，这些信息仅受常规Unix权限的保护。

- **容器文件系统通过/proc/$pid/root链接对Pod中的其他容器可见。**

  这使调试更加容易，但也意味着文件系统安全性只受文件系统权限的保护。

2.3. 如何排查Java场景下故障注入不生效的问题

在对Java进程注入故障时，可能会出现故障注入失败的情况。为解决此类问题，在创建或编辑演练时，您可以在故障执行阶段选择开启Debug模式，并通过相关的日志信息来了解故障注入失败的原因。

开启Debug模式

在查看目标演练的故障注入日志前，您需要先确认该演练的故障执行步骤是否已开启Debug模式。若未开启，可按照以下步骤设置目标演练的故障执行步骤，开启Debug模式。

1. 登录AHAS控制台，在左侧导航栏选择**故障演练 > 我的空间**。
2. 在**我的空间**页面，单击目标演练名称，进入目标演练的详情页。
3. 在**配置**页签下，单击**编辑演练**。
4. 在**演练内容**区域，单击故障执行步骤。

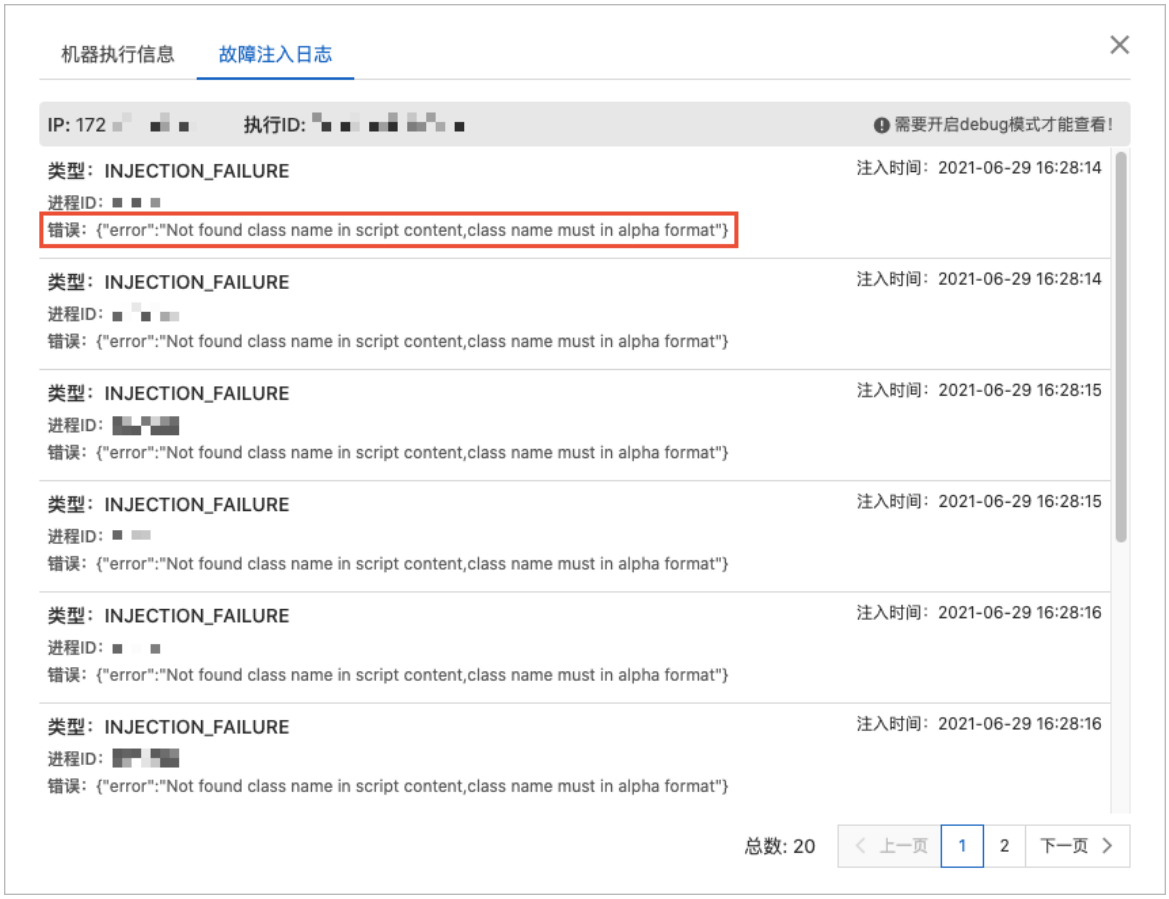
5. 在右侧弹出的对话框的**开启DEBUG**区域，选择**开启**。

6. 单击对话框底部的**关闭**，然后单击**保存**。

查看故障注入日志

1. 登录AHAS控制台，在左侧导航栏选择**故障演练 > 我的空间**。
2. 在**我的空间**页面，单击目标演练名称，进入目标演练的详情页。
3. 单击**演练记录**页签，然后在目标演练记录的**操作**列单击**查看详情**。
4. 在**执行情况**区域，单击故障执行节点，然后在右侧**机器**区域单击需要查看的机器地址。

5. 在弹出的对话框中，单击**故障注入日志**页签，查看故障执行的状态信息。
若状态类型为INJECTION_FAILURE，则表示故障执行失败。您可以通过**错误**区域显示的错误信息了解故障注入失败的原因。以下图为例，故障执行失败的原因是没有在脚本中找到类名。



3.架构感知

3.1. 探针安装常见问题

本文介绍探针安装和容器服务安装ack-ahas-pilot常见问题。

常见问题

本文介绍了以下常见问题：

- AHAS架构感知和故障演练探针limits配置
- ECS探针安装常见问题
- 容器服务安装ack-ahas-pilot常见问题1
- 容器服务安装ack-ahas-pilot常见问题2
- 容器服务安装ack-ahas-pilot常见问题3
- 容器服务安装ack-ahas-pilot常见问题4
- 容器服务安装ack-ahas-pilot常见问题5
- 容器服务安装ack-ahas-pilot常见问题6
- 容器服务安装ack-ahas-pilot常见问题7

AHAS架构感知和故障演练探针limits配置

问题现象

架构感知和故障演练功能需要使用AHAS Agent探针，AHAS Agent以Daemonset方式部署在每个节点上，默认的资源配置如下：

```
resources:
  requests:
    # resources.requests.cpu: cpu request
    cpu: 0.1
    # resources.requests.memory: memory request
    memory: 200Mi
  limits:
    # resources.limits.cpu: cpu limit
    cpu: 0.2
    # resources.limits.memory: memory limit
    memory: 360Mi
```

由于AHAS Agent占用的资源取决于节点上的Pods、进程、网络连接数等，所以在安装AHAS Agent时需要根据节点CPU和内存大小修改Agent limits资源配置。

解决方案

AHAS Agent的limits配置规则如下：

- CPU limits配置：
  - CPU≤8 Core，采用默认配置。
  - CPU＞8 Core，采用Core（核数）/40=CPU limits配置，例如节点CPU总核数是16，则CPU limits配置=16/40=0.4。

- Memory limits配置：
  - 节点内存Memory≤16 G，采用默认配置。
  - 节点内存Memory > 16 G，采用Memory（单位是G）×1024/48=Memory limits配置（单位是Mi）。例如Memory=64 G，则Memory limits配置=64×1024/48=1365 Mi。

综上所述，如果节点配置是12 Core、48 G，则limits配置如下：

```
resources:
  requests:
    # resources.requests.cpu: cpu request
    cpu: 0.1
    # resources.requests.memory: memory request
    memory: 200Mi
  limits:
    # resources.limits.cpu: cpu limit
    cpu: 0.3
    # resources.limits.memory: memory limit
    memory: 1024Mi
```

在Kubernetes集群中，常见ECS规格对应的AHAS Agent的资源limits配置建议如下表。

机器配置	CPU配置	Memory配置
4 vCPU 8 GiB	`resources.requests.cpu: 0.1` `resources.limits.cpu: 0.2`	`resources.requests.memory: 200Mi` `resources.limits.memory: 360Mi`
4 vCPU 16 GiB	`resources.requests.cpu: 0.1` `resources.limits.cpu: 0.2`	`resources.requests.memory: 200Mi` `resources.limits.memory: 360Mi`
8 vCPU 16 GiB	`resources.requests.cpu: 0.1` `resources.limits.cpu: 0.2`	`resources.requests.memory: 200Mi` `resources.limits.memory: 360Mi`

机器配置	CPU配置	Memory配置
96 vCPU 192 GiB	resources.requests.cpu: 0.1 resources.limits.cpu: 2.4	resources.requests.memory: 200Mi resources.limits.memory: 4096Mi

ECS探针安装常见问题

问题现象

探针安装引导页中无法查询到ECS机器。

可能原因

- 请确认地域选择是否正确。
- 架构感知探针仅支持VPC网络模式下的Linux机器，Window机器暂不支持。
- 自动安装模式仅支持ECS中自动安装了云助手的机器，未安装云助手的ECS会执行失败。

容器服务安装ack-ahas-pilot常见问题1

问题现象

安装ack-ahas-pilot时报以下错误。

```
Can't install release with errors: rpc error: code = Unknown desc = unable to decode ": no kind "CustomResourceDefinition: is registered for version "apiextensions.k8s.io/v1beta1”
```

问题原因

可能是Helm版本导致的问题，Helm最低要求2.12版本，解决方式是升级Helm版本。

解决方案

1. 登录到Kubernetes集群Master节点，请参见通过kubectl工具连接集群。
2. 执行如下命令，升级Helm的版本。

   ```
   helm init --tiller-image registry.cn-hangzhou.aliyuncs.com/acs/tiller:v2.16.3 --upgrade
   ```

   > **说明**
   > - 镜像地址可使用对应Region的VPC域名，比如杭州区域的机器可以替换为registry-vpc.cn-hangzhou.aliyuncs.com/acs/tiller:v2.16.3。
   > - 以上操作只升级Helm服务端的版本，客户端可以通过下载对应的Client Binary，请参见下载Helm 2.16.3版本的客户端。

3. Helm客户端和服务端的版本升级完毕后，执行如下命令，确认版本升级成功。

   ```
   helm version
   ```

   系统显示类似如下，确认Helm的版本为v2.16.3。

```
Client: &version.Version{SemVer:"v2.16.3", GitCommit:"2e55dbe1fdb5fdb96b75ff144a3394894
17b146b", GitTreeState:"clean"}
Server: &version.Version{SemVer:"v2.16.3", GitCommit:"2e55dbe1fdb5fdb96b75ff144a3394894
17b146b", GitTreeState:"clean"}
```

容器服务安装ack-ahas-pilot常见问题2

问题现象

安装ack-ahas-pilot时报以下错误。

```
Can't install release with errors: rpc error: code = Unknown desc = a release named ahas al
ready exists. Run: helm ls --all ahas; to check the status of the release Or run: helm del
--purge ahas; to delete it
```

问题原因

已安装AHAS。

解决方案

卸载已存在的AHAS，重新执行创建，卸载方式如下。

1. 容器服务管理控制台
2. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
3. 在集群管理页左侧导航栏，单击**发布**。
4. 在**Helm**页签删除AHAS应用。

容器服务安装ack-ahas-pilot常见问题3

问题现象

安装ack-ahas-pilot时报以下错误。

```
cannot re-use a name that is still in use
```

问题原因

已安装AHAS。

解决方案

卸载已存在的AHAS，重新执行创建，卸载方式如下。

1. 容器服务管理控制台
2. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
3. 在集群管理页左侧导航栏，单击**发布**。
4. 在**Helm**页签删除AHAS应用。

容器服务安装ack-ahas-pilot常见问题4

问题现象

安装ack-ahas-pilot时报以下错误。

```
Can't install release with errors: rpc error: code = Unknown desc = customresourcedefinitio
ns.apiextensions.k8s.io "chaosblades.chaosblade.io" is forbidden: User "system:serviceaccou
nt:kube-system:default" cannot delete resource "customresourcedefinitions" in API group "ap
iextensions.k8s.io" at the cluster scope
```

问题原因

用户集群中的helm server端安装的Tiller没有在具备集群角色（cluster admin）的服务账户的情况下部署，引发此问题。

解决方案

1. 登录到Kubernetes集群Master节点，请参见通过kubectl工具连接集群。
2. 执行以下命令。

```
kubectl create serviceaccount --namespace kube-system tiller
kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --ser
viceaccount=kube-system:tiller
kubectl patch deploy --namespace kube-system tiller-deploy -p '{"spec":{"template":{"sp
ec":{"serviceAccount":"tiller"}}}}'
```

容器服务安装ack-ahas-pilot常见问题5

问题现象

安装ack-ahas-pilot时报以下错误，无法删除chaosblade 的crd，执行 `kubectl delete crd chaosblades.chaosblade.io` 仍无法删除。

```
Can't install release with errors: rpc error: code = Unknown desc = object is being deleted
: customresourcedefinitions.apiextensions.k8s.io "chaosblades.chaosblade.io" already exists
```

解决方案

1. 执行以下命令，删除AHAS Agent CRD资源。

```
kubectl delete crd chaosblades.chaosblade.io --kubeconfig XXX
```

> **说明** 请替换XXX，如果本地配置了config文件，可以不添加kubeconfig参数。

2. 若长时间不返回，执行以下命令。

```
blades=($(kubectl get blade -n ahas --kubeconfig XXX | grep -v NAME | awk '{print $1}'
| tr '\n' ' ')) && kubectl patch blade $blades -n ahas --type merge -p '{"metadata":{"f
inalizers":[]}}' --kubeconfig XXX
```

容器服务安装ack-ahas-pilot常见问题6

问题现象

安装ack-ahas-pilot时报以下错误。

```
Can't install release with errors: rpc error: code = Unknown desc = unable to decode "": no
kind "CustomResourceDefinition" is registered for version "apiextensions.k8s.io/v1beta1"
```

问题原因

可能是Helm版本导致的问题，Helm最低要求2.12版本，解决方式是升级Helm版本。

解决方案

1. 登录到Kubernetes集群Master节点，请参见通过kubectl工具连接集群。
2. 执行如下命令，升级Helm的版本。

```
helm init --tiller-image registry.cn-hangzhou.aliyuncs.com/acs/tiller:v2.16.3 --upgrade
```

> 说明
> - 镜像地址可使用对应Region的VPC域名，比如杭州区域的机器可以替换为registry-vpc.cn-hangzhou.aliyuncs.com/acs/tiller:v2.16.3。
> - 以上操作只升级Helm服务端的版本，客户端可以通过下载对应的Client Binary，请参见下载Helm 2.16.3版本的客户端。

3. Helm客户端和服务端的版本升级完毕后，执行如下命令，确认版本升级成功。

```
helm version
```

系统显示类似如下，确认Helm的版本为v2.16.3。

```
Client: &version.Version{SemVer:"v2.16.3", GitCommit:"2e55dbe1fdb5fdb96b75ff144a339489417b146b", GitTreeState:"clean"}
Server: &version.Version{SemVer:"v2.16.3", GitCommit:"2e55dbe1fdb5fdb96b75ff144a339489417b146b", GitTreeState:"clean"}
```

容器服务安装ack-ahas-pilot常见问题7

问题现象

ack-ahas-pilot安装后，架构感知中无数据或者数据展示不全。

问题原因

- 部分AHAS Agent安装失败，请在**容器服务 > 应用 > 守护**进程集中选择AHAS命名空间，查看AHAS Agent是否存在异常。
- 请调整架构地图中顶部的视图筛选条件，选择合适的过滤条件。

4.多活容灾

4.1. MSHA基础常见问题

本文介绍MSHA的基础常见问题以及解决方案。

- 什么是MSHA？
- 如果用户自有机房，计划同城新建一个机房，应选用什么架构？
- 用户要做容灾架构，是否必须全套都使用阿里云产品？

什么是MSHA？

MSHA是Multi-site High Availability多活容灾的简称。它是容灾领域产品，在业务发展过程中支撑容灾架构演进及建设，覆盖同城多活、异地主备、异地应用双活、异地多活等容灾架构。

如果用户自有机房，计划同城新建一个机房，应选用什么架构？

同城新建机房，距离近且RT可控，可选用异地应用双活、异地灾备、异地双活、异地双读等架构。若用户对成本和改造敏感，双云场景下建议使用 “异地应用双活” 架构，单云多可用区场景下建议使用“同城多活”架构。

用户要做容灾架构，是否必须全套都使用阿里云产品？

不是的。MSHA组件化程度较高，且支持企业级和开源的组件。用户有以下策略可选择：

- 管控面使用MSHA能力，数据面流量入口使用MSHA ，数据面应用层相关组件自建。
- 管控面使用MSHA能力，数据面流量入口和数据库使用MSHA ， 数据面应用层相关组件自建。
- 管控面使用MSHA能力，数据面流量入口、微服务、消息等使用MSHA覆盖的组件，其他有定制化需求组件可通过自建或提交工单确认MSHA是否支持。

4.2. 同城多活常见问题

本文介绍同城多活的常见问题以及解决方案。

- 假设数据库主备，ECS双可用区部署，是否就是同城多活？
- 和EDAS、MSE集群流量同可用区优先的差异和优势？
- 消息是基于Shutdown机制实现多活么？
- 数据库是两个机房各一个主备，还是主机房一主一备，备机房一备的部署形态？
- 消息的主备容灾细节点是什么？
- MSHA能给客户业务带来什么价值？

假设数据库主备，ECS双可用区部署，是否就是同城多活？

不是。此方案仅做到同城“资源”双活，业务流量存在南北走向流量HTTP，东西走向流量RPC、消息、分布式任务等。资源双活在业务节点故障时，南北走向流量切走，东西走向流量仍旧持续进入，业务无法恢复，因此无法做到业务同城多活。

和EDAS、MSE集群流量同可用区优先的差异和优势？

类别	MSHA	EDAS、MSE
微服务日常场景	面向可用区级别，解决RT问题。	面向可用区级别，解决RT问题。
基本概念	逻辑集群概念	可用区概念
同可用区多个逻辑集群优先	支持	不支持
故障场景RPC切零	支持	不支持
支持的服务	• 微服务 • 消息 • 分布式任务	微服务

消息是基于Shutdown机制实现多活么？

不是。因为这样会涉及业务重启恢复，如果每次容灾演练都需要全部业务配合进行机器重启及顺序控制，这样是用户无法接受的。

数据库是两个机房各一个主备，还是主机房一主一备，备机房一备的部署形态？

一主一备。若新增备节点，成本可控下可多备节点。

消息的主备容灾细节点是什么？

Broker容灾策略核心在于数据追平、禁写、主节点切换。

MSHA能给客户业务带来什么价值？

MSHA在 “0-1-5-10”（事前预防-1分钟发现-5分钟决策-10分钟恢复）里面的 “0”、“5”、“10”提供价值，具体表现如下：

- “0”：基于MSHA隔离逻辑区域能力，业务代码发布、配置变更优先在其中之一的区域进行。生产部分进入小比例流量进行验证，持续一定时间后，若系统有问题则快速切流恢复，避免线上因代码、配置问题导致的大面积故障。
- “5”：基于MSHA流量封闭能力，业务故障时，在业务监控层面看到，A区域业务监控指标下跌，B区域正常。此时业务方可快速决策将A切流至B，避免定位问题及决策的时间开销。
- “10”：基于MSHA自上而下的流量规则管理和集成数据库同步能力，可分钟级完成容灾切换操作，恢复业务。

4.3. 异地多活常见问题

本文介绍异地多活的常见问题以及解决方案。

- 微服务在多活场景下如何解决？
- 消息在多活场景下如何解决？
- Redis等缓存在多活场景下怎么解决？
- 分布式调度任务在多活场景下如何解决的？
- MSHA数据面的性能指标是怎样的？
- 是否做好入口流量分发和数据同步，就能快速具备异地多活能力？

- 基于MSHA的业务改造成本是怎样的？
- 如何做好双活和之前业务的过渡态，如何控制风险？

微服务在多活场景下如何解决？

尽可能单元内自封闭，若无法自封闭，可基于Proxy或直连模式，对微服务进行打标，流量打标跨单元调用。

消息在多活场景下如何解决？

消息体打标，在Broker和Consumer侧进行多活处理，消息在单元间同步，避免数据丢失。

Redis等缓存在多活场景下怎么解决？

缓存本身不建议同步，基于数据库Binlog监听变化，使缓存失效，避免数据错乱。若业务需要同步，可考虑通过企业版Redis或DTS进行同步。

分布式调度任务在多活场景下如何解决的？

您可以考虑以下两个方案：

- 双活：2个单元都开启定时任务，捞取全量源数据，使用MSHA-Lignt-SDK过滤掉非本单元的数据后再执行任务。
- 主备方案：2个单元均开启定时任务，配置中心开关控制任务执行的主、备单元角色，主单元执行全量的定时任务，备单元定时任务忽略业务逻辑处理，仅保持运行状态。

MSHA数据面的性能指标是怎样的？

- MSFE性能可控，平均延迟在6 ms~8 ms之间。
- Agent RT同城基本没增长，异地场景增大1 ms~10 ms，与业务场景相关。

是否做好入口流量分发和数据同步，就能快速具备异地多活能力？

否。异地多活核心在于流量控制、数据同步、一站式管控、架构演进四个点。入口流量分发和数据同步仅仅只是完成了四分之一，还有四分之三需要进行建设，否则不具备可用点能力。

基于MSHA的业务改造成本是怎样的？

改造成本核心在于确定分流标示和范围，还包括后续的域名染色、Agent使用，以及Sequence基础改造的成本。

如何做好双活和之前业务的过渡态，如何控制风险？

MSHA具有标准的上线切换流程，简要概述为：

- 按照MSHA多活标准确定分流标和范围。
- 业务做好兼容，完成请求参数携带染色标的改造。
- 微服务、消息等框架组件切换到多活。
- 按标准的日常、预发、线上流程进行上线。

4.4. 异地双读常见问题

本文介绍异地双读的常见问题以及解决方案。

- 异地双读只保障读流量，如何保障写流量？

异地双读只保障读流量，如何保障写流量?

是的，异地双读只保障读流量，写流量可基于其他容灾架构保障。若没有其他容灾架构保障写流量，则默认写流量处理方案为单点写入。

读写流量按什么纬度区分读?

业务侧自行进行区分，可按不同域名、相同域名不同URI、不同IP方式进行区分。

异地双读应用场景是什么?

针对导购类、咨询类、查询类等读多写少型业务。

异地双读的业务改造成本有多大?

核心在于业务能区分出读类型业务，业务本身无代码改造成本，入口流量接入MSFE层，底层数据库同步。

4.5. 异地应用双活常见问题

本文介绍异地应用双活的常见问题以及解决方案。

混合云场景下，涉及云上云下两朵云，两朵云的注册中心如何建立容灾?

注册中心云上云下独立，过渡期可基于MSHA-rpc-sync进行单向同步可见。

配置中心如何建立容灾?

建议您在代码层面对配置中心进行双写改造，对控制台进行多写改造。若您不想改造，需保证同步工具配置最终一致，再考虑是自建还是阿里云支持建设，初步建议您自建。最佳实践场景中，建议您配置推送的时候先推送一个单元，长时间灰度验证没问题后再进行另一个单元操作，因此不推荐自动同步配置。

业务场景中以MySQL数据源为基准，会额外冗余到ES存储数据用于业务使用，ES数据是如何做同步?

目前DTS支持MySQL的同步，其他数据源建议不同步。一份数据无需浪费多份资源进行同步，基于同步的MySQL数据生成其他数据库相关数据即可。

异地应用双活架构是否需要业务改造?

不需要代码改造。仅需要业务接入MSHA-agent即可具备能力。

4.6. 管控面常见问题

本文介绍在MSHA管控面的常见问题以及解决方案。

MSFE创建集群错误怎么办？

您可以从以下方面进行排查：

- 实例需要开启安全组，其中80端口用于用户流量（挂载SLB），8090端口用于MSHA管控。

  公共云开放地址如下：

  - 杭州：100.104.58.192/26
  - 上海：100.104.66.192/26
  - 北京：100.104.133.64/26
  - 深圳：100.104.255.128/26

- CentOS必须是7u，不能是其它版本。集群网络类型要和ECS网络类型相同。

确保VPC互通的场景？

在MSFE与回源应用利用内网连通的情况下，需要确保两个单元间的VPC互通（SLB和ECS都在同一个VPC内最佳），并且带宽足够。

关于HTTP/HTTPS的注意事项？

- 如果到达MSFE的流量是HTTPS（例如MSFE位于用户流量第一跳），那么必须在MSFE配置域名的证书（配置顶级域名即可）。因为MSFE需要解析报文，并做路由寻址。
- 回源应用推荐使用HTTP，将证书终结在MSFE是合理的。因为MSFE到后端应用一般是内网，不需要加密。这也是目前支持的唯一一种方式。
- 回源应用也可以使用HTTPS，这种是在MSFE到应用的链路不可靠的情况下使用，目前暂不支持。

关于SLB的注意事项？

- 用户若需自行配置SLB监听，建议配置TCP监听即可。
- 若您一定要使用HTTP监听，则需注意以下风险：
  - 流量会经过SLB的Tengine，有一层性能损耗。
  - 流量会经过SLB的Tengine，对于会话保持、连接管理、报文长度、超时时间等无法进行管控。
  - 若配置了健康检查，MSFE的Tengine的健康检查路径为*ip/status*；路径错误会导致SLB会把MSFE的Tengine剔除，从而不能正常响应。

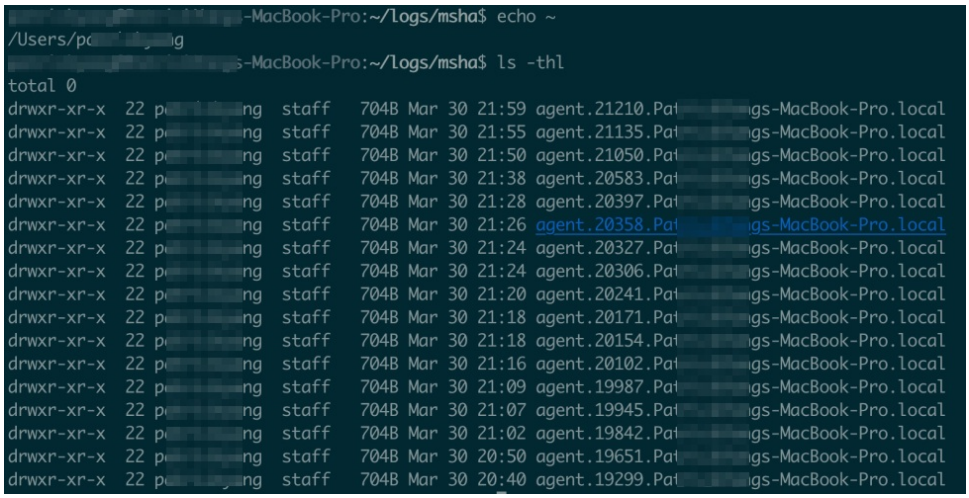
4.7. 数据面常见问题

本文介绍MSHA数据面的常见问题以及解决方案。

MSHA-agent日志在哪里？

MSHA-agent日志打印目录（每个进程一个日志目录）：*~/logs/msha/agent.$processId.$localIdentity/*。

MSHA-agent日志目录如下图所示：



您需要主要关注的日志文件：

- *agent.log*：启动日志。
- *agent_plugin.log*：Agent插件AOP切面日志。
- *agent_error.log*：Agent异常日志。

同城多活单元格流量切零后，企业版RocketMQ的*ons.log*中为什么会出现日志 `brokerName=msha_mock_queueBrokerName` ？

日志如下：

```
2021-03-31 23:59:34,034 INFO RocketmqClient
- topicRouteTable.put.Topic = MQ_INST_1815555058395900_BbCHFzvM%xiniao_pra_callback,
TopicRouteData
[TopicRouteData
 [orderTopicConf=null, queueDatas=
  [QueueData
    [brokerName=msha_mock_queueBrokerName, readQueueNums=8, writeQueueNums=8, perm=6, topic
SynFlag=0]
  ],
brokerDatas=
 [BrokerData
   [brokerName=msha_mock_brokerName, brokerAddrs={0=1.2.3.4:80}]
 ],
filterServerTable={}
 ]
]
```

这只是RocketMQ更新地址池信息*topicRouteTable.put*的日志，里面打印的 `brokerName=msha_mock_brokerName` （已经是MSHA-agent地址池过滤后放入的一个虚拟brokerName了）。这个日志说明流量切零是生效的，这台机器已经不会再消费消息了。

异地多活如何查看MSHA-agent是否从HTTP流量中提取出了路由标？

查看*agent_plugin.log*日志，会打印出Servlet切面before和after的2行日志，从下面示例中 `routeId:JoeHeader` 可以看出提取出的路由标为 `JoeHeader` 。

```
2021-05-13 15:12:39,750 INFO http-nio-7004-exec-1
com.alibaba.msha.agent.plugin.servlet.trace.ServletEnhancerHandler
- beforeAdvice invoked, url:http://localhost:7004/echo/xx, httpRouteIdConfigs:{5bec72a6-8e0
c-4ad1-9846-87e2442bd034=HttpRouteIdConfig{routeType='default', routeIdSources=[arg, header
, cookie], routeIdKey='key123yyybbbb'}}, httpHeader:{host=localhost:7004, user-agent=curl/7
.54.0, accept=*/*, key123yyybbbb=JoeHeader}, httpCookie:{}
2021-05-13 15:12:39,751 INFO http-nio-7004-exec-1
com.alibaba.msha.agent.plugin.servlet.trace.ServletEnhancerHandler
- routeId found in HTTP flow, routeId:JoeHeader, routeIdSource:header, routeIdKey:key123yyy
bbbb
```

4.8. MSHA产品和开源产品对比

本文介绍MSHA商业产品与开源产品之间在架构、通用能力、辅助能力等方面的异同。

架构支持

架构	商业产品	开源产品
异地双活	□	□
同城多活	□	□
异地应用双活	□	□

异地双活

组件	功能大类	具体功能说明	商业产品	开源产品
网关	路由	范围路由	□	□
网关	路由	精准路由	□	□
服务层SpringCloud	路由	范围路由	□	□
服务层SpringCloud	路由	精准路由	□	□
服务层SpringCloud	路由	单元保护（不属于本单元的调用直接失败）	□	□
服务层Dubbo	路由	范围路由	□	□
服务层Dubbo	路由	精准路由	□	□
服务层Dubbo	路由	单元保护（不属于本单元的调用直接失败）	□	□

组件	功能大类	具体功能说明	商业产品	开源产品
消息层MQ	路由	发布打标（路由ID）	□	□
消息层MQ	路由	消费过滤（依据路由ID计算消费单元）	□	□
消息层MQ	Failover	切流时重置位点，消息不丢	□	□
数据层MySQL	路由	数据保护，防止多点写脏数据	□	□
数据层MySQL	Failover	路由ID维度：切换数据源和延迟禁写，包括： • PolarDB • DRDS	□	□

通用能力

组件	功能大类	具体功能说明	商业产品	开源产品
网关	路由	条件路由（入口流量调度）	□	□
注册中心同步	服务发现	跨集群的服务同步	□	□
切流模块	命令通道	控制面和数据面的信道打通	□	□
切流模块	切流	一键切流（完成动作编排、规则计算和下发）	□	□
切流模块	切流	自定义动作	□	□

辅助能力

组件	功能大类	具体功能说明	商业产品	开源产品
网关	健康检查	网关实例巡检，异常透出。	□	□
网关	健康检查	Upstream巡检，自动摘除和恢复。	□	□
网关	发布	支持版本管理和灰度发布。	□	□
注册中心同步	健康检查	实例巡检，异常透出。	□	□

组件	功能大类	具体功能说明	商业产品	开源产品
OPS	报警	对所有相关异常进行透出和报警。		

组件	功能大类	具体功能说明	商业产品	开源产品
网关	Metrics	通用指标和特定架构的指标。		
网关	Logging	结构化的AccessLog和ErrorLog。		
网关	Tracing	为每个请求生成Trace ID。		
注册中心同步	Metrics	任务粒度的同步QPS、RT、错误数。		
Agent	Logging	• 关键点记录（路由、禁写等）。 • 高频日志（排查）可开关。		
Agent	Metrics	各个模块切面的QPS、RT、错误率。		
数据层	Metrics	同步位点延迟。		
消息层	Metrics	同步位点延迟。		

可观测

5.其他

5.1. 查看License

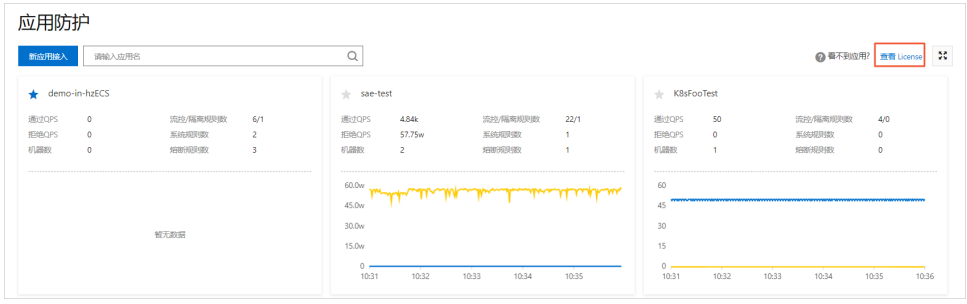
License作为客户端的启动参数，用于关联客户端跟阿里云账户的关系。本文介绍如何查看License。

背景信息

若用户的机器是阿里云在VPC网络环境内，且部署在杭州、上海、北京、深圳、张家口这些地域内，则无需配置License参数，默认会关联到机器所属用户下。其他情况下，用户则需要配置License参数关联到AHAS公网地域，才能使用AHAS。

查看License

1. 登录AHAS控制台，在左侧导航栏选择**流量防护 > 应用防护**。
2. 在**应用防护**页面右上角单击**查看License**。



注意事项

- 2020年05月20日之前开通AHAS的用户，不同地域License可能不同步，若需要接入到AHAS公网，请先在AHAS控制台页面的左上角选择公网，再查看License。

- License属于用户敏感信息，请勿泄露给他人使用。

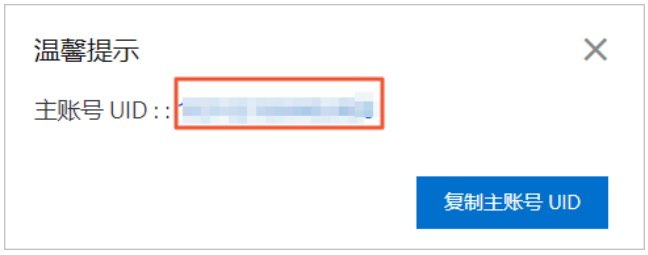
5.2. 如何查看主账号UID？

在AHAS的日常操作或维护中，系统可能会需要您提供主账号UID，即阿里云账号ID。本文介绍查看您主账号UID的方法。

方法一

您可以通过**概述**页的**查看主账号UID**获取信息。

1. 登录AHAS控制台。
2. 在**概述**页的右上角单击**查看主账号UID**，获取到您的UID信息。



方法二

若您当前登录账号是主账号，则可以通过账号管理获取主账号UID。

1. 登录AHAS控制台。
2. 单击控制台页面右上角的头像，进入**账号管理**页面。

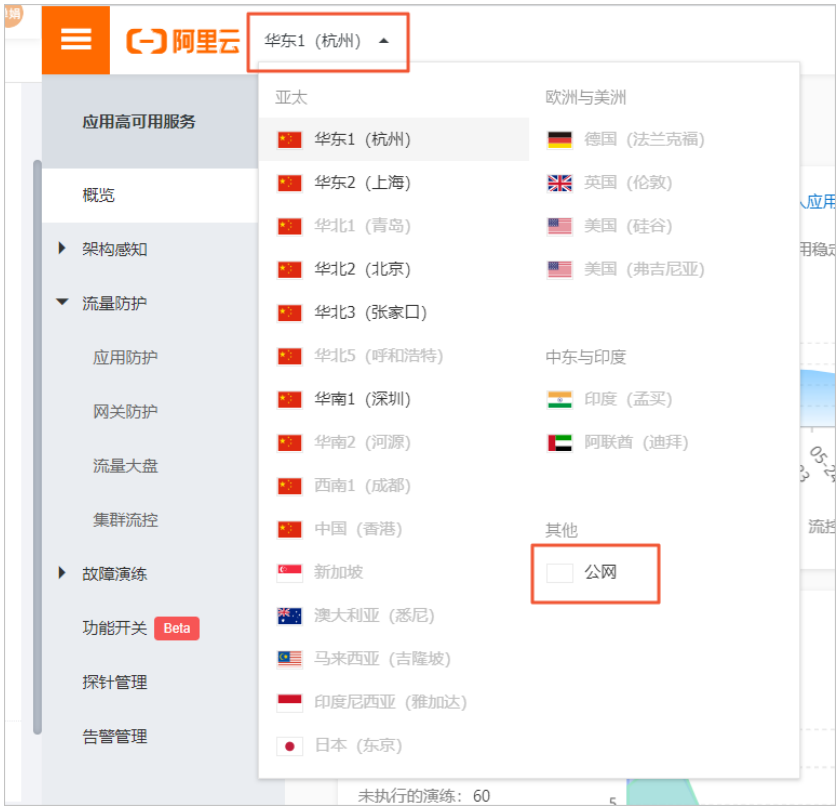
3. 在左侧导航栏选择**安全设置**，进入**安全设置**页面。页面中的**账号ID**即所需的账号UID。

5.3. 如何切换地域（Region）？

AHAS实例按地域（Region）划分，在日常操作维护中，您可能因机器在不同的地域而需要切换控制台中的地域（Region），本文介绍在AHAS控制台如何切换地域（Region）。

操作步骤

1. 登录AHAS控制台。
2. 单击控制台左上方的地域，在列表中选择所需的地域（Region）。
   - 如果您有阿里云北京、杭州、上海或深圳等地VPC的机器，可以选择对应的地域（Region）。
   - 如果不属于第一种情况，可选择**公网**。



5.4. 如何结束 AHAS Agent 进程？

AHAS 为应用高可用探针（即 AHAS Agent）提供进程自动拉起功能，即定时检查 AHAS Agent 进程是否存在，如不存在，自动拉起该进程。这一功能保证了进程可用性，避免进程因异常挂掉或机器重启等原因，需要手动拉起的情况。

所以，您在服务器进程中结束 AHAS Agent 无法永久结束该进程，AHAS 会在定时检查后或机器重启时，自动拉起该进程。

如果您某段时间不需要使用 AHAS Agent，建议您登录AHAS 控制台，在**管理**页面卸载 AHAS Agent。

5.5. 如何查看消费情况？

本文介绍如何在AHAS控制台查看您的消费情况。

如何在控制台查看资源状况？

1. 登录AHAS控制台。
2. 在**概述**页面，右上角的**资源状况**区域，展示了当前资源状况。
   - **流量防护**的**在线节点数**为当前活跃节点数。
   - **剩余资源**为剩余按量抵扣资源包资源。
   - **资源包数量**则为当前有效的包月资源包或按量抵扣资源包数量，单击数字可查看资源包具体信息。

如何查看消费账单？

1. 登录AHAS控制台。
2. 在顶部菜单栏选择**费用 > 费用账单**。

3. 在**费用账单**页面单击**账单明细**页签。
4. 在**账单明细**页签查看您的消费账单。

如何查看资源包使用情况？

1. 登录AHAS控制台。
2. 在顶部菜单栏选择**费用 > 费用账单**。

3. 在左侧导航栏选择**资源管理 > 资源包**，在**资源包总览**页签查看资源包情况。
4. 在**资源包管理**页面单击**使用明细**页签查看资源包使用明细。

5.6. 公网常见问题

本文介绍关于公网环境的一些常见问题。

1. **ECS经典网络无法连接到对应地域（Region）的控制台？**

   经典网络只能连接到公网控制台。请在控制台左上角选择**公网**环境，并在程序启动参数中加入 `-Dahas.license=${license}` ，然后在公网控制台查看应用。

2. **公网模式下AHAS控制台看不到数据？**

在公网模式下AHAS控制台看不到数据，且查看 `~/logs/csp` 下的日志，出现如下错误信息。表示节点上网络无法通过公网访问*ahas-proxy.aliyuncs.com*。可以通过尝试 `ping ahas-proxy.aliyuncs.com` 来进行网络验证，并确认服务器网络是否能通公网。

5.7. RAM 子账号访问 AHAS 控制台

借助访问控制RAM的RAM用户，您可以实现权限分割的目的，按需为RAM用户赋予不同权限，并避免因暴露阿里云账号密钥造成的安全风险。

背景信息

出于安全考虑，您可以为阿里云账号（主账号）创建 RAM 用户（子账号），并根据需要为这些子账号赋予不同的权限，这样就能在不暴露主账号密钥的情况下，实现让子账号各司其职的目的。在本文中，假设企业 A 希望让部分员工处理日常运维工作，则企业 A 可以创建 RAM 用户，并为 RAM 用户赋予相应权限，此后员工即可使用这些 RAM 用户登录控制台 。

AHAS 允许您借助 RAM 用户实现分权，即为该子账号开启控制台登录权限，并授予 **AliyunAHASFullAccess** 权限。

步骤一：创建 RAM 用户

首先需要使用阿里云账号（主账号）登录 RAM 控制台并创建 RAM 用户。

1. 登录 RAM 控制台，在左侧导航栏中选择**人员管理 > 用户**，并在**用户**页面上单击**新建用户**。
2. 在**新建用户**页面的**用户账号信息**区域框中，输入**登录名称**和**显示名称**。

   > **说明** 登录名称中允许使用小写英文字母、数字、“.”、“_”和“-”，长度不超过 128 个字符。显示名称不可超过 24 个字符或汉字。

3. （可选）如需一次创建多个用户，则单击**添加用户**，并重复上一步。
4. 在**访问方式**区域框中，勾选**控制台密码登录**，并设置自动生成默认密码或自定义登录密码、登录时是否要求重置密码，以及是否开启 MFA 多因素认证。单击**确定**。
5. 在**手机验证**对话框中单击**获取验证码**，并输入收到的手机验证码，然后单击**确定**。创建的 RAM 用户显示在**用户**页面上。

后续步骤

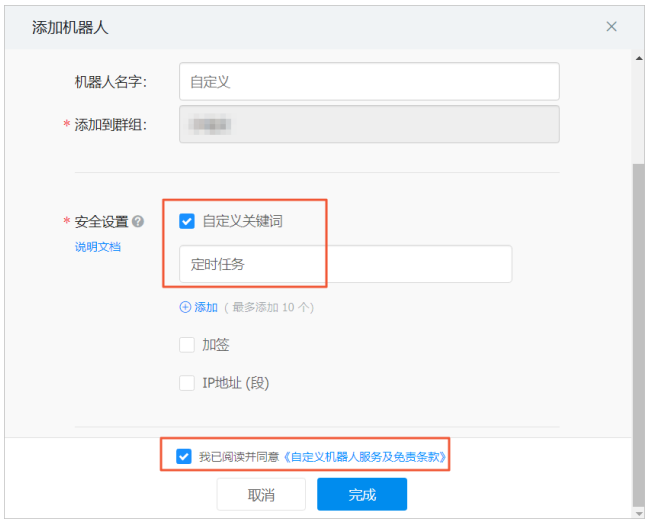
授权完成后，即可用子账号登录子账号 AHAS 控制台。

5.8. 如何获取自定义机器人Webhook地址？

本文以PC端钉钉为例为您介绍如何获取自定义机器人Webhook地址。

操作步骤

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**面板中选择**智能群助手 > 添加机器人**。
3. 在**群机器人**页面，选择**添加机器人**。
4. 在**群机器人**页面选择添加**自定义**。
5. 在**机器人详情**对话框中单击**添加**，然后在**添加机器人**对话框中编辑机器人头像和名称，选择**安全设置**为**自定义关键词**，**关键词**填写**定时任务**。选中**我已阅读并同意《自定义机器人服务及免责条款》**，然后单击**完成**。



6. 在**添加机器人**对话框中复制生成的机器人Webhook地址。

> **注意** 请保管好此Webhook地址，不要公布在外部网站上，泄露后有安全风险。