阿里云
云监控

最佳实践

文档版本：20190131

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
	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。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
*## *	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。	`swich` *`{stand | slave}`*

目录

1 报警模板最佳实践

本文旨在通过一个具体案例讲解企业用户如何使用报警模板，高效管理好各业务使用的云资源的报警规则。

背景信息

当您的云账号下拥有很多服务器和云产品的资源时，怎样才能快速的为这些资源创建报警规则，在报警规则不合理时修改报警规则。本文将通过具体案例为您讲解大企业用户如何通过使用报警模板和应用分组，提升效率并管理好各业务使用云资源的报警规则。

使用报警模板的准备工作

使用报警模板前，我们先了解一下报警规则配置在应用分组和配置在单个实例上的区别以及报警模板如何提升配置规则的效率。

- 报警规则配置在应用分组和配置在单实例上的区别：
  - 创建报警规则时资源范围可以选择“实例”或者“应用分组”，如果选择“应用分组”，那么报警规则的作用范围就是整个应用分组内的所有资源。您的业务需要扩容或者缩容时，只需要将相应资源移入或移出应用分组，而不需要增加或删除报警规则。如果需要修改报警规则，也只需要修改这一条报警规则，就会在组内所有实例上生效。
  - 如果您选择将报警规则创建在实例上，那么该规则只对单一实例有效。修改报警规则时也只对单一实例生效。当实例增多时报警规则会变得难以管理。
- 报警模板极大的提升了配置报警规则的效率。
  - ECS、RDS、SLB等基础服务在配置报警时，监控项和报警阈值相对固定，为这些需要报警的指标建立模板后，新增业务时，创建好应用分组后直接将模板应用在分组上，即可一键创建报警规则。
  - 当您需要批量新增、修改、删除报警规则时，也可以修改模板后，将模板统一应用在分组上，极大的节省操作时间。

使用报警模板的实施步骤

注意事项

当您的账号下服务器和其他云产品实例非常多时，首先建议您按照业务视角为资源创建不同的应用分组，然后通过应用分组来批量管理资源。

操作步骤

下面我们以一个常见的电商网站后台业务为例，介绍如何使用报警模板和应用分组，快速将业务的云上监控报警体系搭建起来。

1. 首先，我们创建一个名为“电商后台模块报警模板”的报警模板。

   a. 登录云监控控制台。

   b. 单击左侧导航栏中报警服务下的报警模板，进入报警模板页面。

   c. 单击右上角的创建报警模板按钮，进入创建报警模板页面。

   d. 配置模板基本信息：输入模板名称和描述。

   e. 配置报警策略：选择产品类型，添加并设置报警规则，将业务模块需要的报警策略添加到报警模板中。

   f. 点击确认按钮，保存报警模板配置。

2. 创建报警联系人和报警联系组。

   a. 登录云监控控制台。

   b. 单击左侧导航栏中报警服务下的报警联系人，进入报警联系人管理页面。

c. 单击右上角的新建联系人按钮，填写手机、邮箱等信息。添加手机和邮箱时需要对手机和邮件进行验证，防止您填写了错误的信息，无法及时收到报警通知。

d. 在报警联系人管理页面，单击页面上方的报警联系组页签，切换到报警联系组列表。

e. 单击右上角的新建联系组，弹出新建联系组页面。

f. 填写组名并选择需要加入组中的联系人即可。

3. 创建一个名为“库存管理线上环境”的应用分组，并选择刚才创建的报警模板。

a. 登录云监控控制台。

b. 单击左侧导航栏中的应用分组，进入应用分组页面。

c. 单击右上角的创建组按钮，进入创建应用分组页面。

d. 配置基本信息：输入应用分组名称，选择联系人组。联系人组即报警联系组，用于接收报警通知。

e. 配置监控报警：选择报警模板（用于对组内的实例初始化报警规则）和报警级别。启用初始化安装监控插件，即在新生成ECS 实例后，会对实例安装云监控插件，以便采集监控数据。

f. 配置动态添加实例：库存管理这块业务使用的云资源，我们以最常见的服务器+数据库+负载均衡资源组合为例。通过制定动态匹配规则添加云服务器ECS实例，支持根据ECS实例

名称进行字段的“包含”、“前缀”、“后缀”匹配，符合匹配规则的实例会加入到应当前用分组（包含后续新创建的实例）。最多可以添加三条动态匹配规则，规则之间可以是“与”、“或”的关系。点击添加产品，可继续制定云数据库RDS版和负载均衡的动态匹配规则。

g. 点击创建应用分组按钮，完成分组的创建。进入到该应用分组详情页，即可看到符合匹配规则的实例已添加到您所创建的应用分组内。

2 如何通过钉钉群接收报警通知

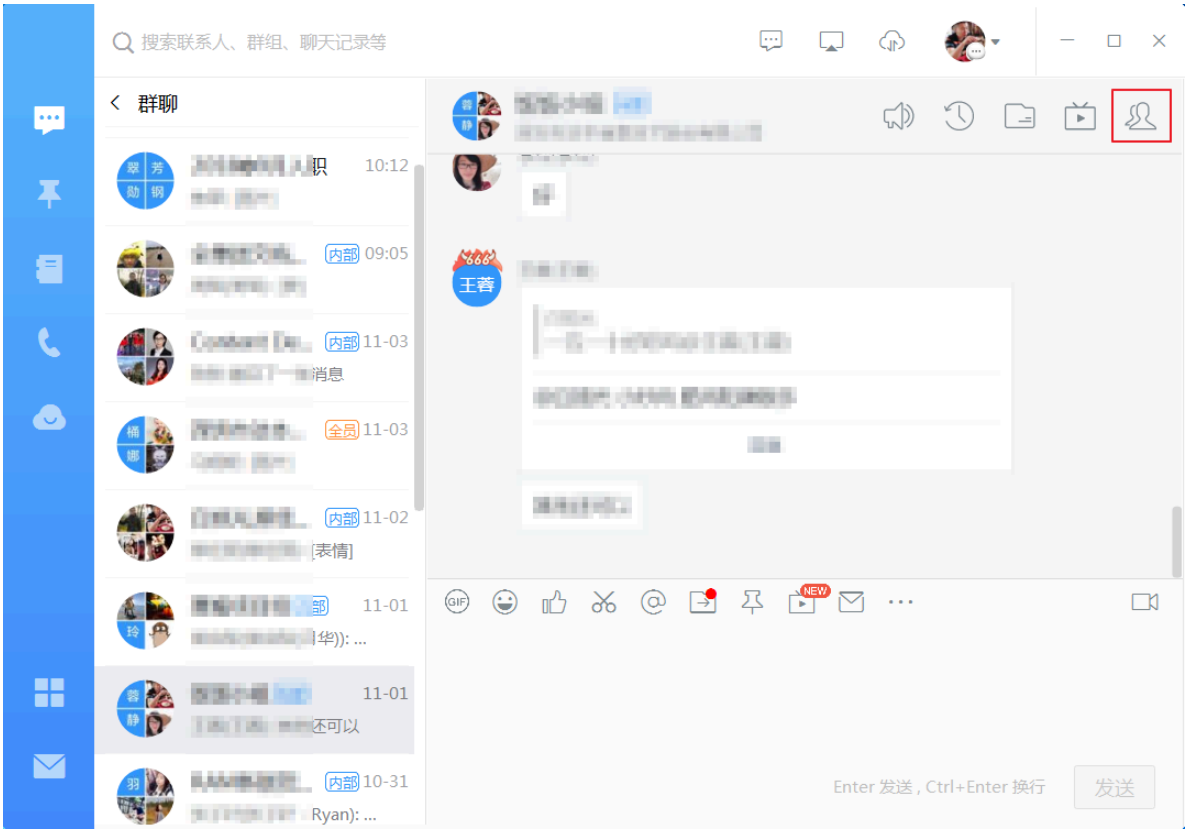
云监控新增钉钉群接收报警通知的功能，您可以按照以下指引设置钉钉群接收报警通知。

已经创建的报警规则，只需要在报警联系人中增加钉钉机器人的回调地址，即可收到钉钉群报警，无需修改报警规则。

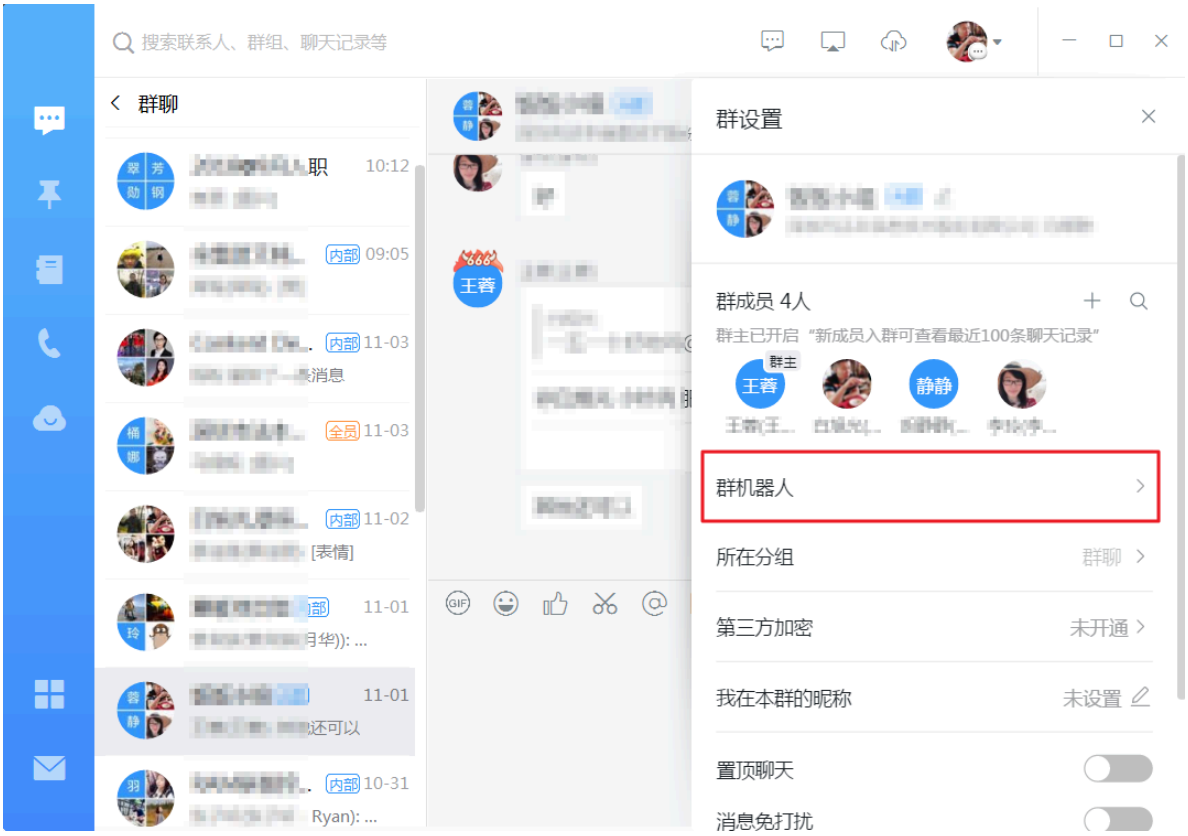
在已有的报警联系人上新增钉钉机器人后，即可通过钉钉群接收联系人之前通过邮件、短信收到的全部报警规则了。

创建钉钉机器人（PC版）

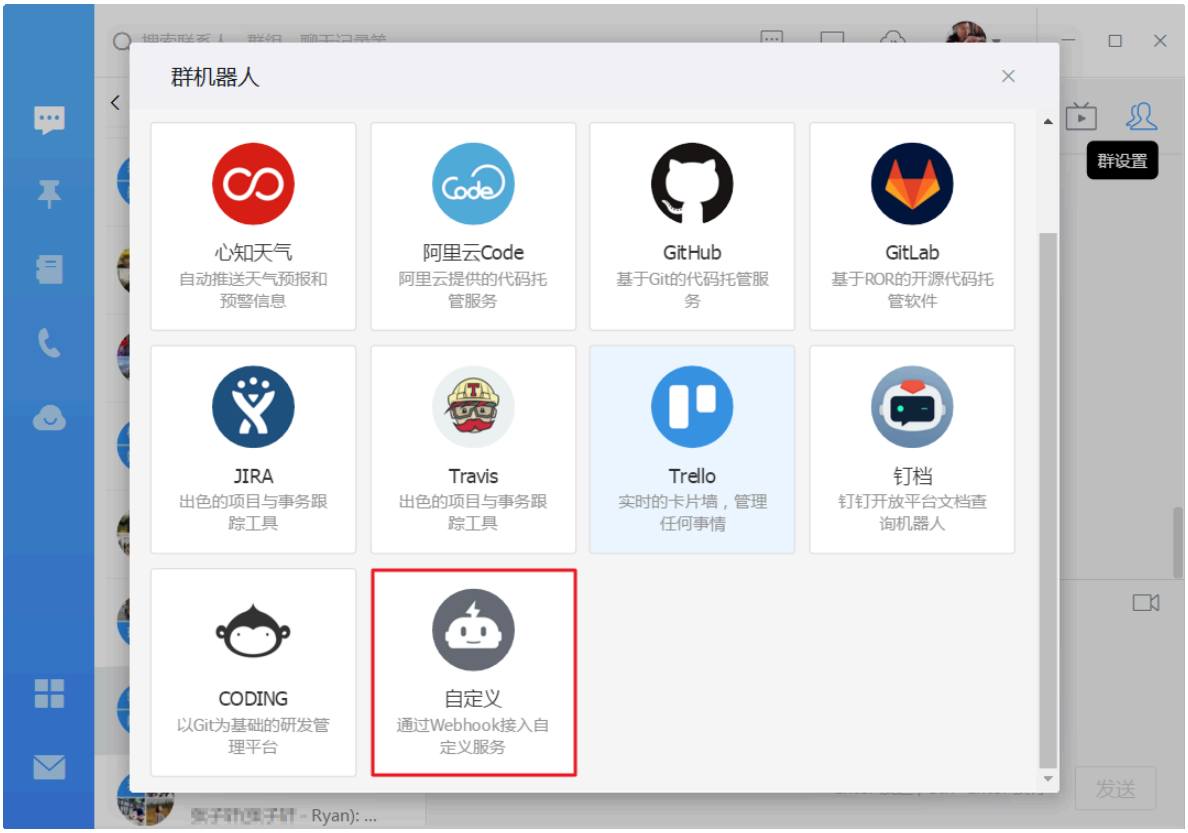
1. 在PC版中打开您要接收报警通知的钉钉群。



2. 单击右上角的群设置图标，打开群设置弹窗。



3. 单击群机器人，打开群机器人窗口，单击自定义，创建一个用于接收报警通知的钉钉机器人。



4. 在机器人详情窗口，单击添加，进入添加机器人窗口。

5. 输入机器人名字，例如云监控报警通知，单击完成即可。

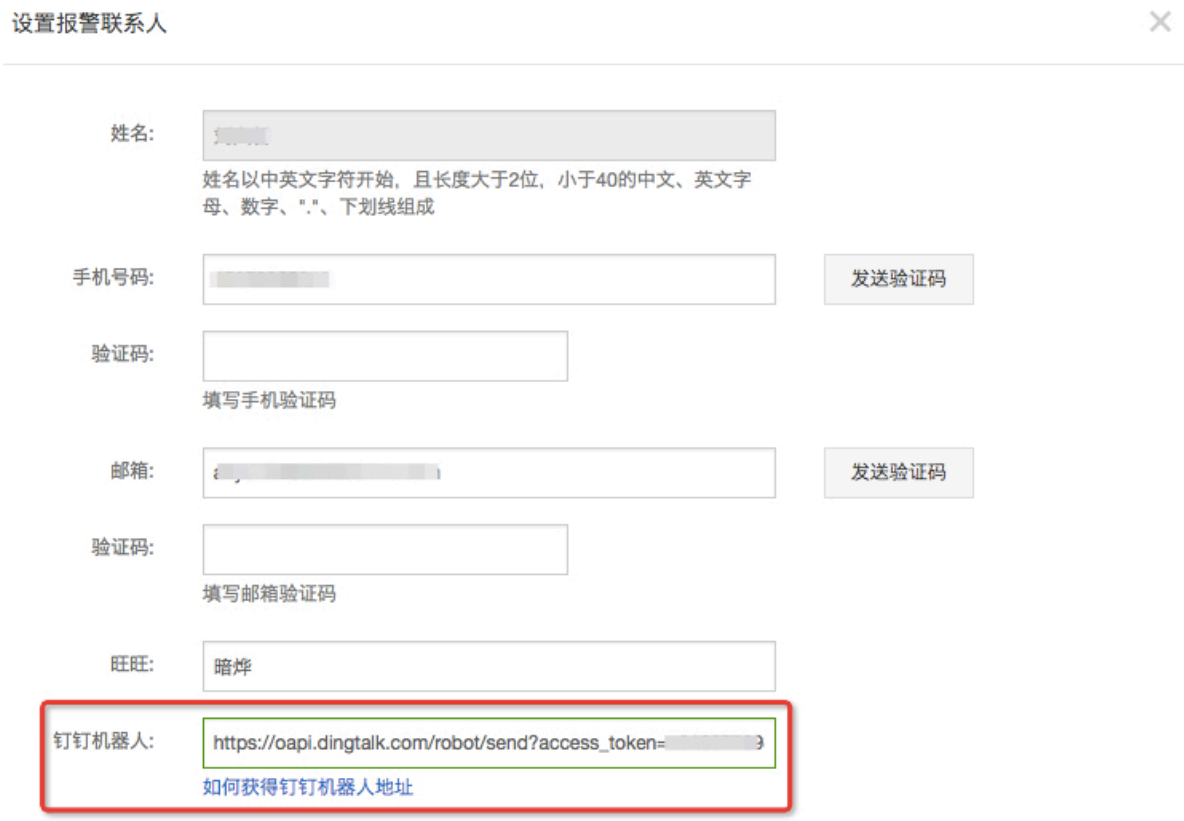
6. 单击复制，复制webhook地址，单击完成，即可完成添加机器人。

在报警联系人中添加钉钉机器人

将上述创建好的钉钉机器人webhook地址添加在报警联系人中，该联系人所在联系组对应的报警规则即可通过钉钉群接收报警通知。

1. 登录云监控控制台。
2. 单击左侧导航栏中报警服务下的报警联系人，进入报警联系人管理页面。

3. 单击编辑，打开设置报警联系人窗口，在已有联系人中添加钉钉机器人的回调地址，或者单击新建联系人，创建包含钉钉机器人的联系人。


设置报警联系人
姓名:
姓名以中英文字符开始，且长度大于2位，小于40的中文、英文字母、数字、"."、下划线组成
手机号码:
发送验证码
验证码:
填写手机验证码
邮箱:
发送验证码
验证码:
填写邮箱验证码
旺旺:
暗烨
钉钉机器人:
https://oapi.dingtalk.com/robot/send?access_token=
如何获得钉钉机器人地址

3 创建容器实例监控大盘

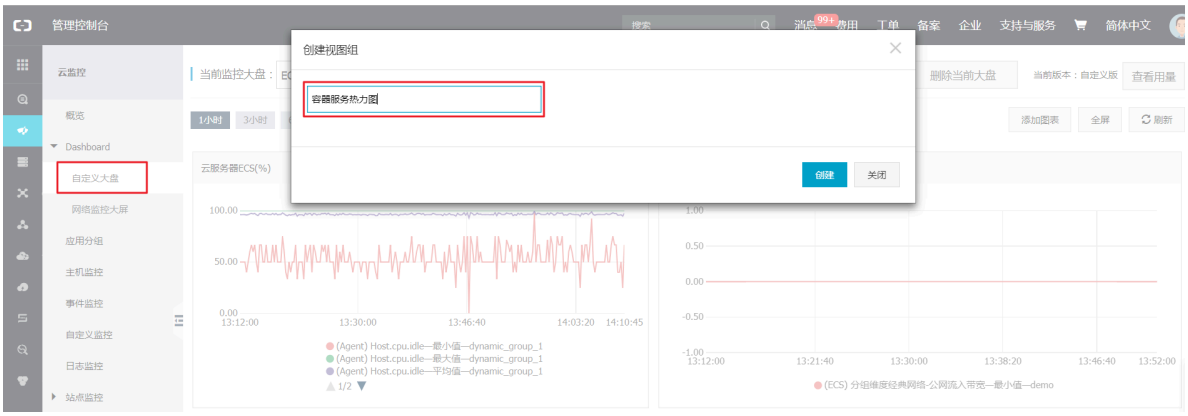
应用场景

随着上云不断深入，越来越多的企业级用户选择将服务直接部署在容器服务里，容器实例越来越多，用户期望能够有一个大图显示所有容器实例的热力负载情况，便于随时掌握全局情况。

您可以通过云监控的Dashboard和容器服务监控两者结合满足这个需求场景。

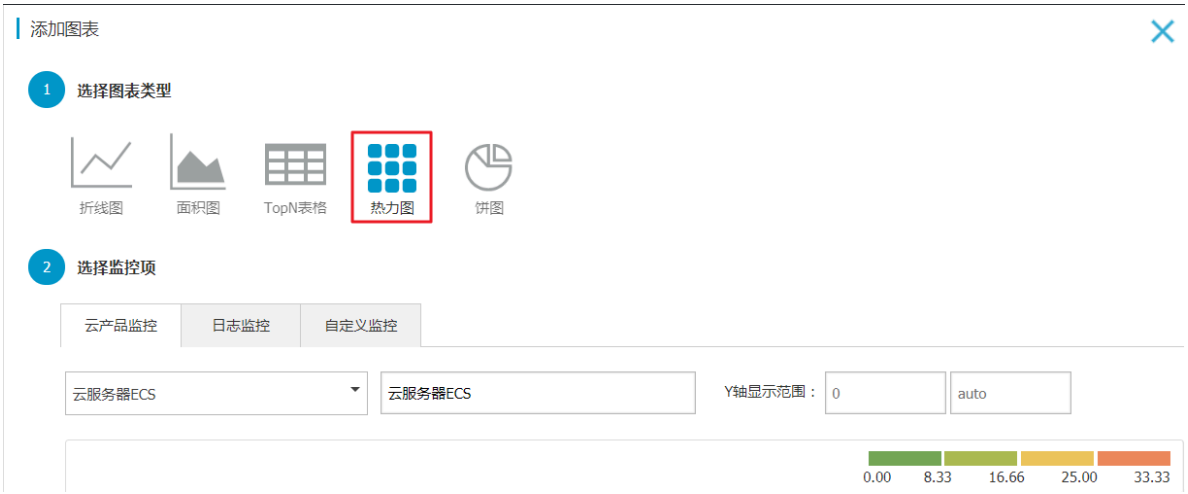
操作步骤

1. 登录云监控控制台。
2. 单击左侧导航栏中Dashboard下的自定义大盘，进入当前监控大盘页面。
3. 单击右上角的创建监控大盘，创建一个容器监控大盘。

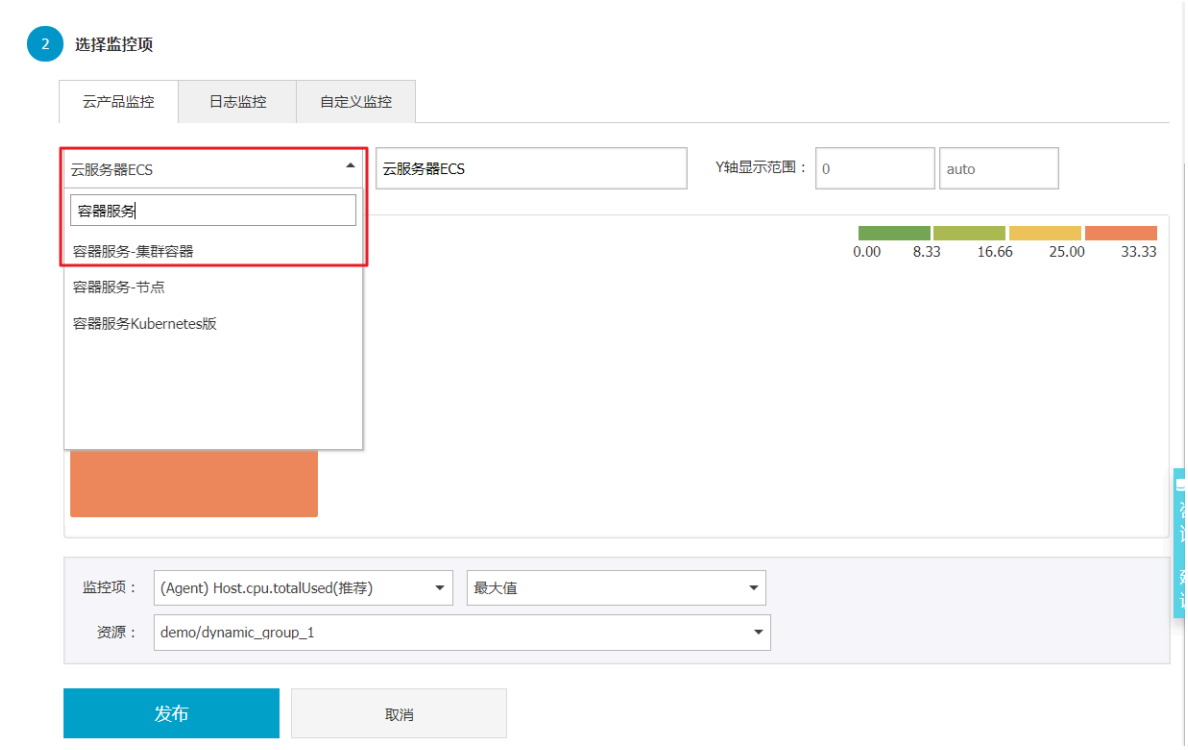


4. 在创建好的大盘内，点击 添加图表，进入添加图表页面。

5. 选择图表类型：选择热力图。



6. 选择监控项：在云产品监控页签产品下拉框中选择容器服务，并选择相应的监控项和资源。



7. 单击发布按钮，完成图表添加，效果图如下：

8. （可选）单击右上角全屏按钮，可以查看全屏效果图。

云监控Dashboard监控大盘支持添加多个图表，每个图表相对独立，因此可以添加多种产品（ECS、RDS、SLB等）的监控图表到一个大盘，然后对大盘进行投屏。

4 日志监控最佳实践

4.1 日志关键字的监控与报警

目的

统计业务日志中关键字的数量，并在统计数量达到一定条件时报警是业务日志的常见需求之一。本教程的目的是通过一个具体案例介绍如何对存储在日志服务产品中的数据进行关键字统计和报警。参照本教程的介绍，您可以快速掌握日志的关键字统计、查询图表可视化和设置报警流程。

实战案例

- 使用前提

  1. 首先需要您将本地日志收集到日志服务（Log Service）中，如果您未使用过阿里云日志服务产品，可查看日志服务快速入门了解产品。
  2. 需要确保主账号的AccessKey是激活状态。AccessKey保持激活状态后您才能授权云监控读取您的日志数据。

     激活方法：登录阿里云控制台，将鼠标移至页面右上角您的用户名上方，在显示的菜单中单击 AccessKeys，在弹出的确认对话框中单击继续使用AccessKey以进入 AccessKey管理页面 。创建密钥对（Access Key），确认状态已设置为启用。

- 统计日志关键字

  在使用日志监控前，您需要确保收集到日志服务中的日志已经被切分为Key-Valve格式。参考常见日志格式常见日志格式的处理方法。

  日志样例

```
2017-06-21 14:38:05 [INFO] [impl.FavServiceImpl] execute_fail and
run time is 100msuserid=
2017-06-21 14:38:05 [WARN] [impl.ShopServiceImpl] execute_fail, wait
 moment 200ms
2017-06-21 14:38:05 [INFO] [impl.ShopServiceImpl] execute_fail and
run time is 100ms,reason:user_id invalid
2017-06-21 14:38:05 [INFO] [impl.FavServiceImpl] execute_success,
wait moment ,reason:user_id invalid
2017-06-21 14:38:05 [WARN] [impl.UserServiceImpl] execute_fail and
run time is 100msuserid=
2017-06-21 14:38:06 [WARN] [impl.FavServiceImpl] execute_fail, wait
moment userid=
```

```
2017-06-21 14:38:06 [ERROR] [impl.UserServiceImpl] userid=, action
=, test=, wait moment ,reason:user_id invalid
```

收集到日志服务中的日志被切分成如下字段：

Key	Value
content	2017-06-21 14:38:05 [INFO] [impl.FavServiceImpl] execute_fail and run time is 100msuserid=
content	2017-06-21 14:38:05 [WARN] [impl.ShopServiceImpl] execute_fail, wait moment 200ms
content	2017-06-21 14:38:06 [ERROR] [impl.ShopServiceImpl] execute_success:send msg,200ms
content	……

1. 授权云监控只读权限

a. 进入云监控首页，选择日志监控功能。

b. 按照页面提示，点击这里进行授权。初次使用日志监控功能时需要授权，后续不再需要授权。授权后云监控会获得读取您日志数据的权限，并且仅用于按照您配置的处理规则进行日志数据处理的用途。

2. 配置统计方式

a. 授权后可进入如下日志监控列表页面。

b. 点击新建日志监控，进入创建页面。

c. 关联资源，选择您需要进行关键字统计的日志服务资源。

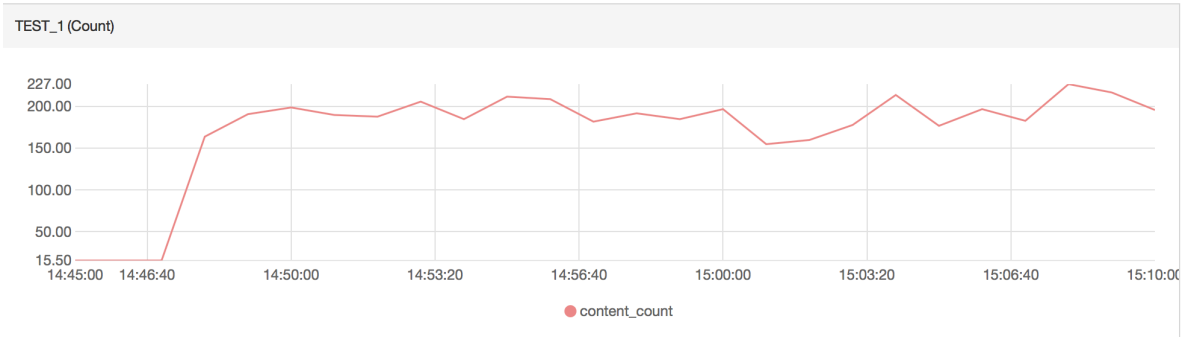
d. 预览数据：如果您选择的日志服务中已经写入数据，可以在第二步分析日志的预览框中查看到原始的日志数据。

e. 分析日志，本步骤用于定义如何处理日志数据。不支持日志的字段名称为中文。这里以统计ERROR关键字数量为例，统计日志每分钟出现的ERROR关键字数量。通过日志筛选过滤出content中包含ERROR关键字的日志记录，并通过统计方法中的计数（Count）方法计算筛选后的记录数。

f. 点击确定，保存配置。

3. 查看统计数据

创建完日志监控以后，等待3-5分钟即可查看统计数据。查看方法是进入日志监控的指标列表页面，点击操作中的监控图表查看监控图。

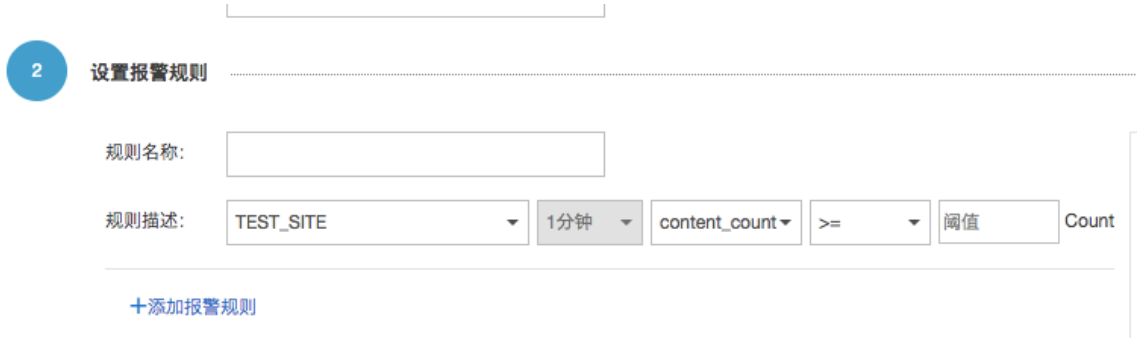


4. 设置报警规则

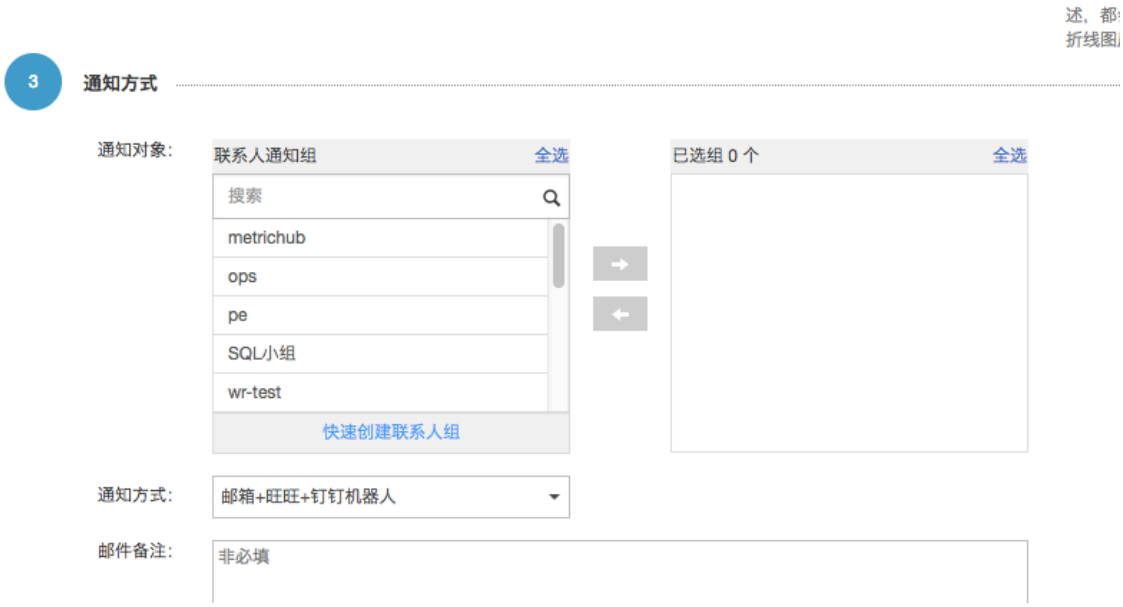
a. 进入日志监控的指标列表页面，点击操作中的报警规则，进入报警规则列表页面。

b. 点击页面右上角的新建报警规则按钮，进入创建报警规则页面。

c. 为报警规则命名，并在规则描述中配置需要报警的情况。



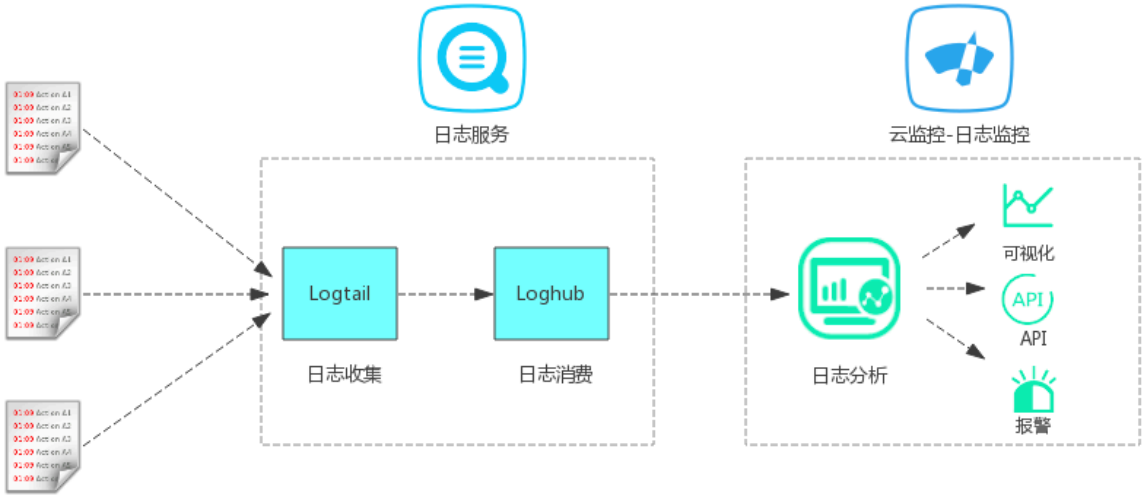
d. 选择需要报警的联系人组和通知方式并确认保存，即可完成报警规则的设置。



4.2 业务日志的统计监控与报警

目的

本教程的目的是通过一个具体实例介绍如何对存储在日志服务中的数据进行数据统计、形成可视化监控图、设置报警。

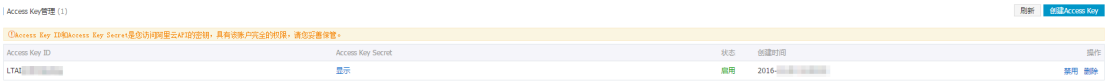


实战案例

- 使用前提

1. 首先需要您将本地日志收集到日志服务（Log Service）中，如果您未使用过阿里云日志服务产品，可查看日志服务快速入门了解产品。
2. 需要确保主账号的AccessKey是激活状态。AccessKey保持激活状态后您才能授权云监控读取您的日志数据。

   - 激活方法：登录阿里云控制台，将鼠标移至页面右上角您的用户名上方，在显示的菜单中单击 AccessKeys 。在弹出的确认对话框中单击继续使用AccessKey以进入AccessKey管理页面 。创建密钥对（Access Key），确认状态已设置为启用。



- 创建日志监控

在使用日志监控前，需要您确保收集到日志服务中的日志已经被切分为Key-Valve格式。参考常见日志格式常见日志格式的处理方法。

1. 授权云监控只读权限

   a. 登录云监控控制台，选择日志监控。

   b. 按照页面提示，点击这里进行授权。初次使用日志监控功能时需要授权，后续不再需要授权。授权后云监控会获得读取您日志数据的权限，并且仅用于按照您配置的处理规则进行日志数据处理的用途。

2. 创建日志监控

a. 授权后可进入如下日志监控列表页面。

b. 点击新建日志监控，进入创建页面。

c. 关联资源，选择您需要进行监控统计的日志服务资源。

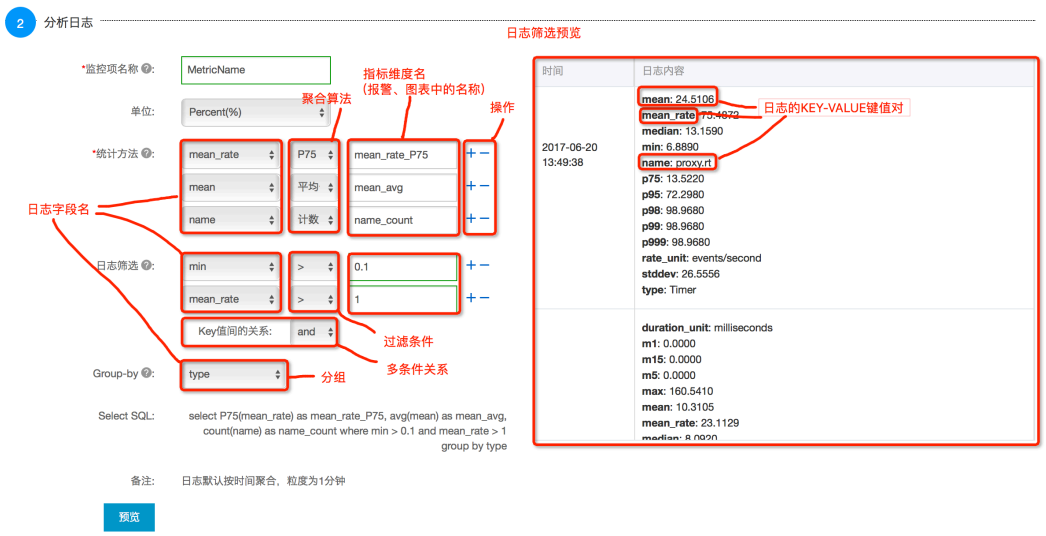
d. 预览数据：如果您选择的日志服务中已经写入数据，可以在第二步分析日志的预览框中查看到原始的日志数据。

时间	日志内容
2017-06-21 14:16:01	**content**: 2017-06-21 14:16:01 [ERROR] [impl.FavServiceImpl] userid=, action=, test=:send msg,userid=
2017-06-21 14:16:01	**content**: 2017-06-21 14:16:01 [WARN] [impl.UserServiceImpl] execute_success:send msg,200ms
2017-06-21 14:16:02	**content**: 2017-06-21 14:16:02 [WARN] [impl.UserServiceImpl] execute_fail, wait moment user id=
2017-06-21 14:16:02	**content**: 2017-06-21 14:16:02 [ERROR] [impl.FavServiceImpl] execute_success and run time is 100ms,reason:user_id invalid
2017-06-21 14:16:02	**content**: 2017-06-21 14:16:02 [ERROR] [impl.FavServiceImpl] userid=, action=, test= and run time is 100ms200ms
2017-06-21 14:16:02	**content**: 2017-06-21 14:16:02 [WARN] [impl.UserServiceImpl] userid=, action=, test=, wait moment 200ms

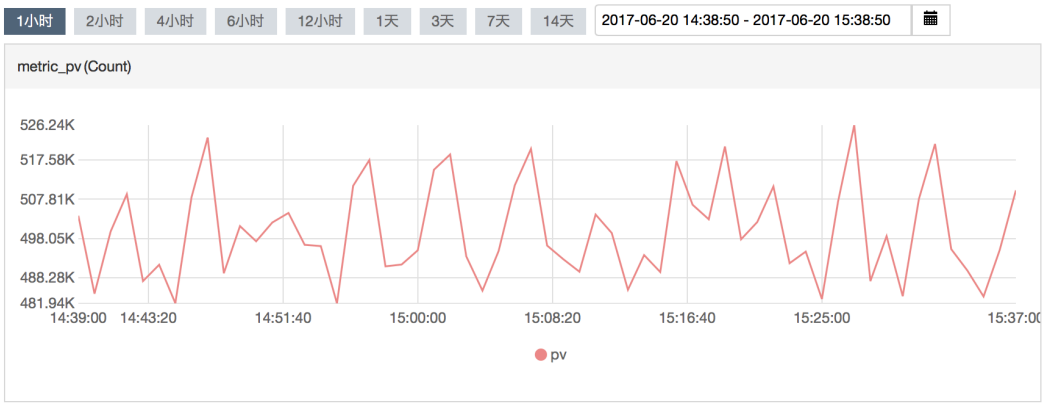
预览数据功能需要您开启日志服务的索引功能。具体可点击如下图所示的开启日志索引链接，进入查询页面后在页面右上角点击开启索引。

e. 分析日志，本步骤用于定义如何处理日志数据。不支持日志的字段名称为中文。

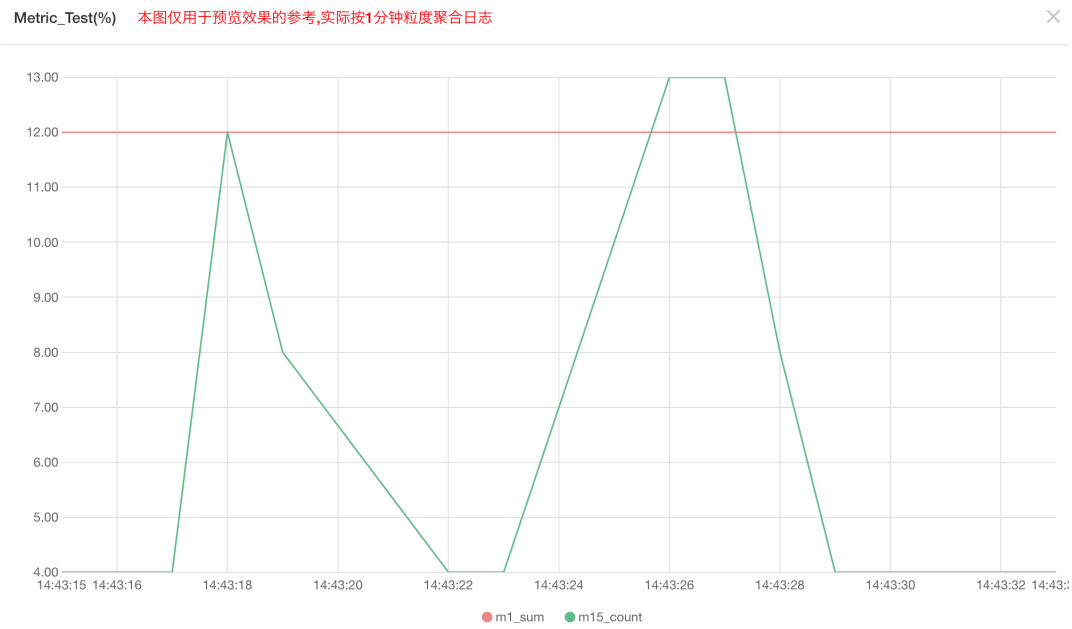
- 监控项名称: 定义一个监控指标的名称。支持数字、字母、下划线。
- 单位：可以根据数据含义选择一个单位，会显示在监控图的Y轴上。
- 统计方法：每分钟根据选定的统计方法对日志数据进行聚合处理。如果字段值是数值型，可以使用所有统计方法，否则只能使用计数和countps两种聚合算法。



- 日志筛选：对日志数据进行过滤，相当于SQL中的where条件。选择过滤的日志字段名不能包含中文。
- Group by：类似SQL的group by功能，根据指定日志字段对数据进行分组后再按照聚合算法聚合。支持不对数据进行Group by。以下是不Group by和Group by的结果展示，分别计算日志的每分钟整体PV和按http 返回码分类的各返回码PV。



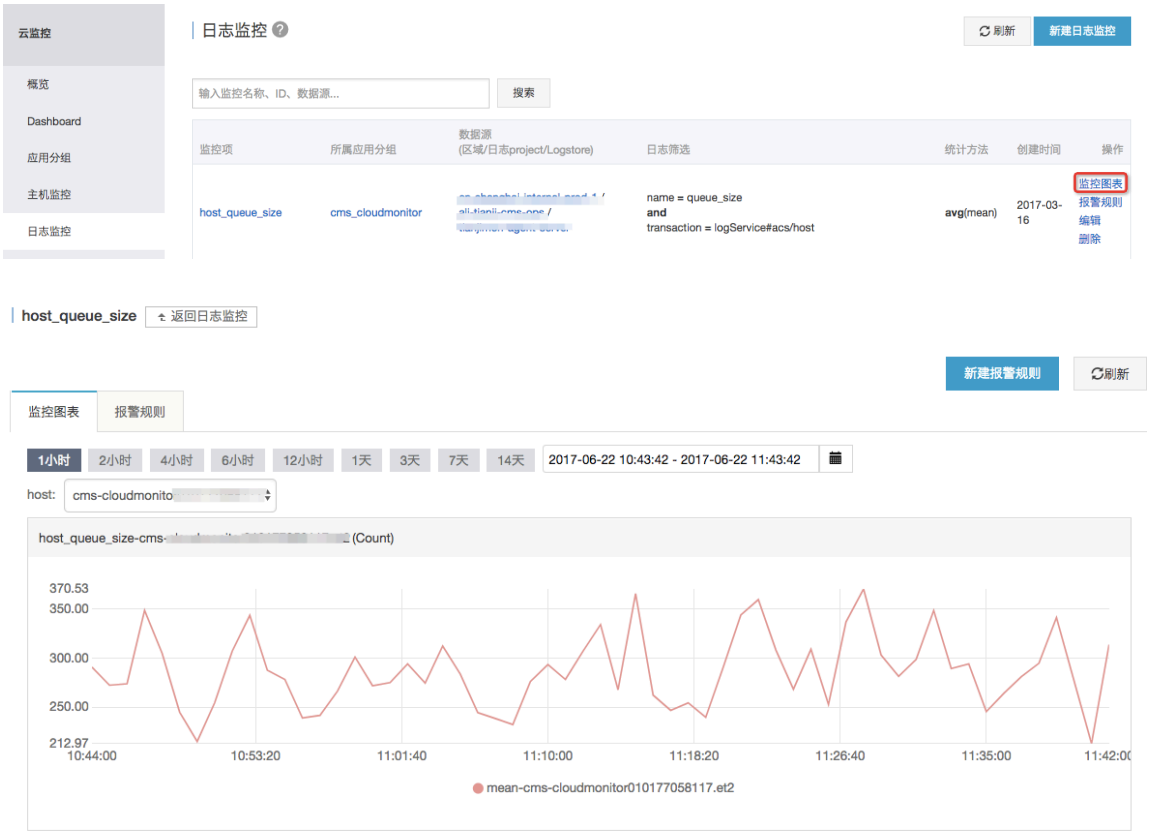
- 预览：实际统计会按1分钟进行聚合计算，预览中为方便您调试，按1秒为单位进行计算（只计算最近100条日志数据）。预览目前不支持Group by功能。



f. 创建报警：本步骤为可选，可以在创建日志监控时设置报警，也可后续需要时再创建报警规则。规则描述选择您在分析日志时统计方法中定义的值。默认为您发送邮件、旺旺和钉钉机器人通知。如果您需要更复杂的报警设置，可在创建好日志监控指标后，通过报警规则页面创建规则。

3. 查看监控数据

创建完日志监控以后，等待3-5分钟即可查看监控数据。查看方法是进入日志监控的指标列表页面，点击操作中的监控图表查看监控图。



4. 设置报警规则

您在创建完日志监控后，可以后续再为添加的指标创建报警规则。

a. 进入日志监控的指标列表页面，点击操作中的报警规则进入报警规则管理页面。

b. 点击页面右上角的新建报警规则按钮，进入报警规则创建页面。

c. 设置报警阈值、触发条件、通知方式和通知对象等主要配置。

d. 点击确认保存设置。

4.3 网站访问日志数据统计与报警

场景

使用ECS搭建网站，并且将网站的访问日志（比如Nginx，Apache）收集到阿里云日志服务后，您可以使用日志监控统计QPS、状态码（HTTP CODE）、响应时间（rt）等指标，并对这些指标设置报警规则。

下文以Nginx的AccessLog为例，说明如何使用日志监控统计网站的QPS、状态码、响应时间。

日志字段

```
192.168.1.2 - - [10/Jul/2015:15:51:09 +0800] "GET /ubuntu.iso HTTP/1.0
" 0.032 129 200 168 "-" "Wget/1.11.4 Red Hat modified"
```

以上述日志为例，在日志服务中配置提取如下字段。

字段	字段样例	说明
time	2015-06-10 15:51:09	日志时间戳，日志系统默认值字段。
rt	0.032	执行时间,单位为秒，精度为毫秒。
URL	/ubuntu.iso	访问的URL。
status	200	HTTP 返回码。
body	168	返回客户端的HTTP body大小，不包含header。

说明:

URL的值不要包含参数GET请求“？”后面的值，如果是Rest风格则不要包含资源定位符，否则无法进行单URL的QPS统计。

如果您是第一次使用日志监控功能，在使用日志监控功能前，需要授权云监控读取可以读取您的日志，详情可参考授权日志监控。

统计网站总QPS或各个URL的QPS

1. 登录云监控，进入日志监控页面后，点击页面右上角的“新建日志监控”，进入配置页面。
2. 在关联资源中选择需要统计的网站访问日志数据源。

3. 在分析日志中，统计方法选择任意一个日志字段，countps计算方式，和日志筛选条件”。如果统计网站的总QPS，则group by不需要填写内容。如果统计各个URL的QPS，group by选择“URL”字段。（URL的个数需要在1000以内，否则会导致没有监控数据。）

统计网站的整体响应时间分布

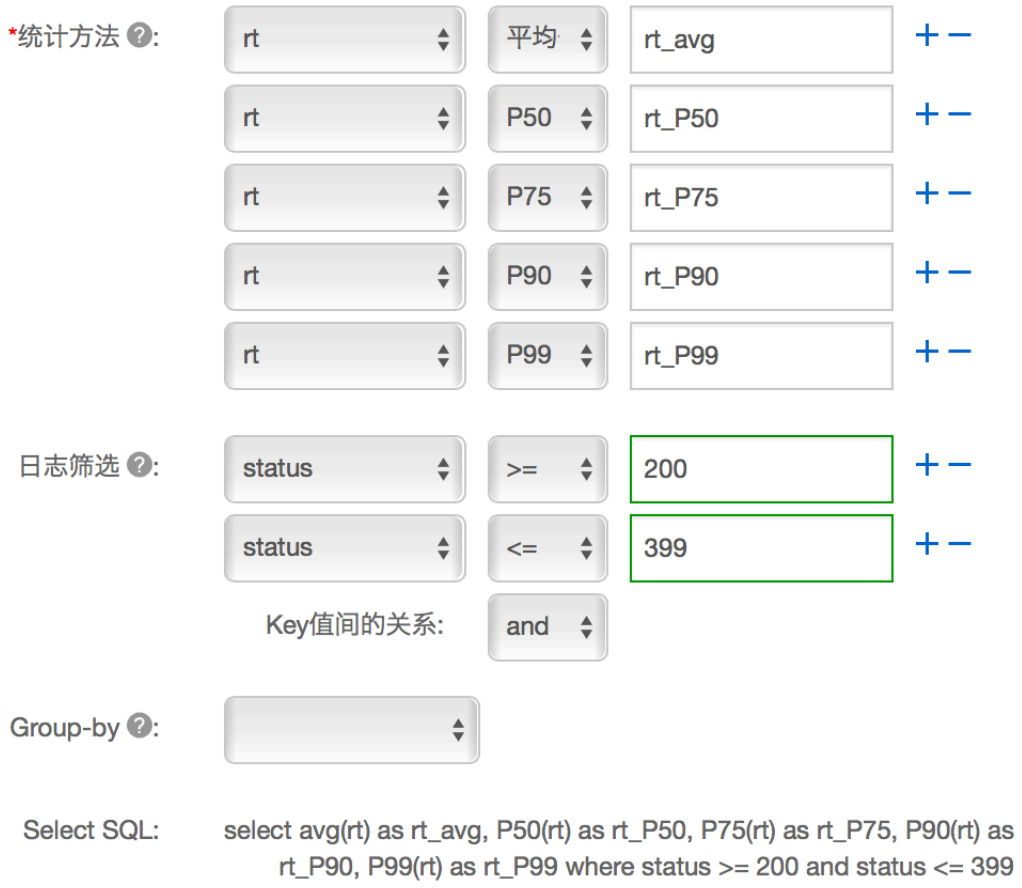
1. 登录云监控，进入日志监控页面后，单击页面右上角的“新建日志监控”，进入配置页面。
2. 在关联资源中选择需要统计的网站访问日志数据源。

3. 在分析日志中，统计方法选择rt字段，并且根据实际需求选择求和、P50、P75、P90、P99等计算方法。日志过滤和group by不需要填写内容。
   - 选择平均，表示1分钟内的平均时间。

- 选择P50，表示1分钟内rt的中位数。
- 选择P75，表示1分钟内75%的rt小于此值。
- 选择P90，表示1分钟内90%的rt小于此值。
- 选择P99，表示1分钟内99%的rt小于此值。

统计网站HTTP访问请求为2XX/3XX情况下的响应时间分布

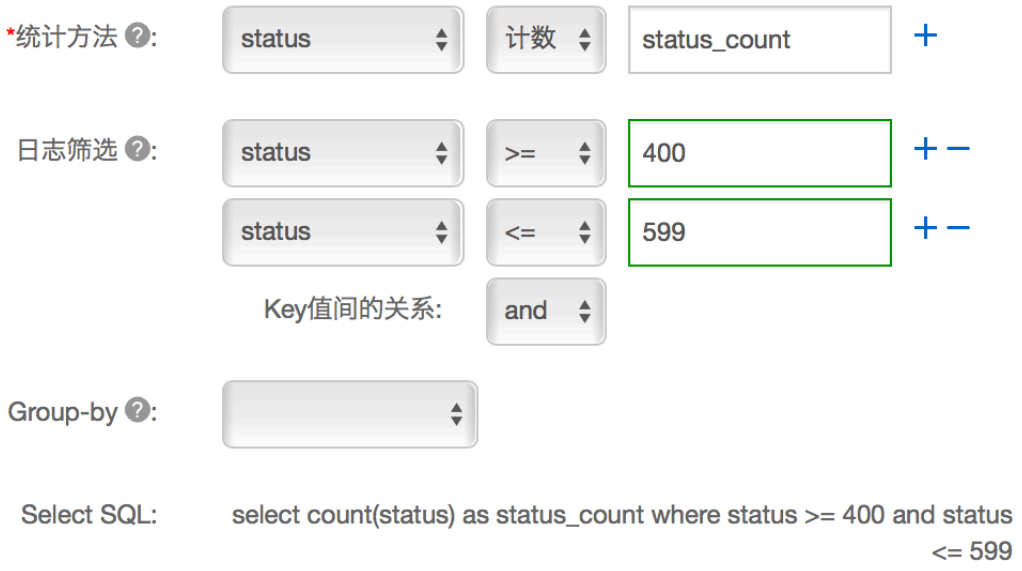
1. 登录云监控，进入日志监控页面后，单击页面右上角的“新建日志监控”，进入配置页面。
2. 在关联资源中选择需要统计的网站访问日志数据源。
3. 在分析日志中，统计方法选择rt，并且根据实际需求选择求和、P50、P75、P90、P99等计算方法。日志过滤和group by不需要填写内容。
   - 选择平均，表示1分钟内的平均时间。
   - 选择P50，表示1分钟内rt的中位数。
   - 选择P75，表示1分钟内75%的rt小于此值。
   - 选择P90，表示1分钟内90%的rt小于此值。
   - 选择P99，表示1分钟内99%的rt小于此值。
4. 在日志筛选中按下图选择各参数 。



5. 如果统计网站的整体2XX/3XX 响应时间分布，group by不需要填写内容。如果统计网站下各个URL的2XX/3XX 响应时间分布，group by选择“URL”字段。URL的个数需要在1000以内，否则会出现没有监控数据。）

统计网站HTTP访问请求的4XX、5XX状态码的个数

1. 登录云监控，进入日志监控页面后，单击页面右上角的“新建日志监控”，进入配置页面。
2. 在关联资源中选择需要统计的网站访问日志数据源。
3. 在分析日志中，统计方法选择`status`，并选择计数的计算方式。
4. 在日志筛选中按下图选择各参数 。



5. 如果统计网站的整体4XX/5XX 响应个数，group by不需要填写内容。如果统计网站下各个URL的4XX/5XX 响应个数，group by选择URL。URL的个数需要在1000以内，否则会出现没有监控数据。）
6. 单击确认保存设置。

5 内网监控最佳实践

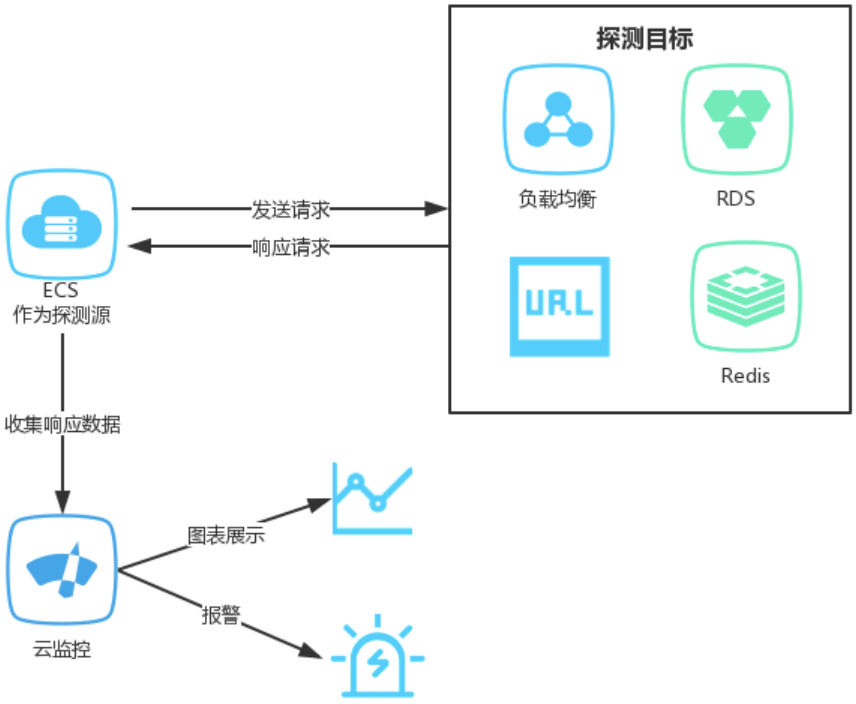
本文旨在通过具体案例介绍如何使用云监控实现内网监控的目的。

背景信息

随着越来越多的用户从经典网络迁移到更安全、更可靠的VPC网络环境，如何监控VPC内部服务是否正常响应就成为需要关注的问题。本文将通过具体案例说明如何监控VPC内ECS上的服务是否可用、VPC内ECS到RDS、Redis的连通性如何、VPC内SLB是否正常响应。

内网监控准备工作

内网监控的原理如下图所示：



首先需要您在服务器上安装云监控插件，然后通过控制台配置监控任务，选择已安装插件的机器作为探测源，并配置需要探测的目标URL或端口。完成配置后，作为探测源的机器会通过插件每分钟发送一个HTTP请求或Telnet请求到目标URL或端口，并将响应时间和状态码收集到云监控进行报警和图表展示。

内网监控的实施步骤

注意事项

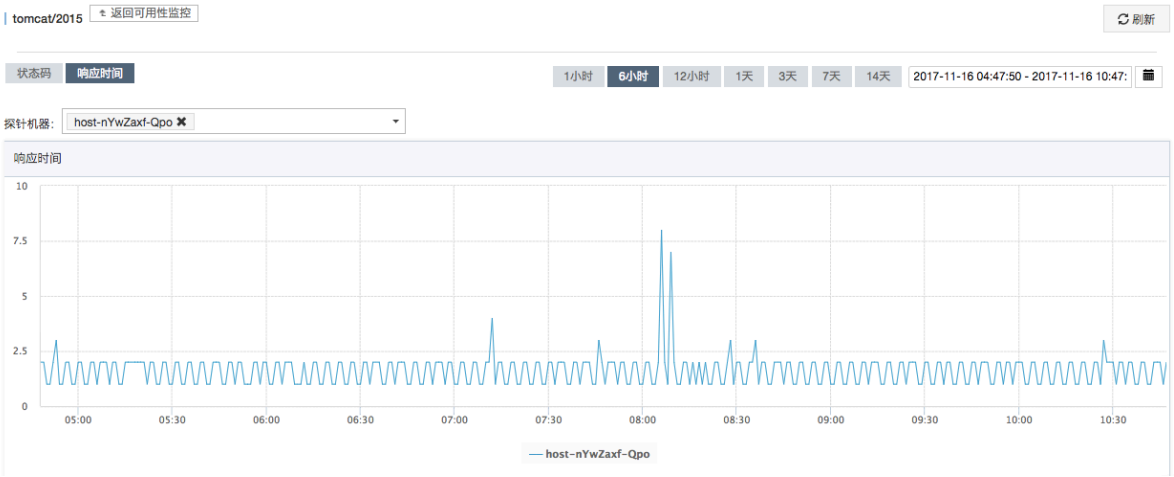
- 作为探测源的服务器需要安装云监控插件。
- 需要创建应用分组，并将作为探测源的服务器加入到分组中。

操作步骤

1. 登录云监控控制台。
2. 单击左侧导航栏中的应用分组，进入应用分组页面。
3. 选择需要创建可用性监控的应用分组，进入应用分组详情页面。
4. 单击页面左侧导航栏中的可用性监控，进入可用性监控页面。
5. 单击页面右上角的新建配置按钮，进入编辑页面。
   - 需要监控VPC内ECS本地进程是否响应正常时，可在探测源中选中所有需要监控的ECS，在探测目标中填写`localhost:port/path`格式的地址，进行本地探测。
   - 需要监控VPC内SLB是否正常响应时，可选择与SLB在同一VPC网络内的ECS作为探测源，在探测目标中填写SLB的地址进行探测。
   - 需要监控VPC内ECS后端使用的RDS或Redis是否正常响应时，可将与ECS在同一VPC网络内的RDS或Redis添加到应用分组，并在探测源中选择相应的ECS，在探测目标中选择RDS或Redis实例。

6. 单击确定后，可以在任务对应的监控图表中查看探测结果，并在探测失败时收到报警通知。

7. 单击任务列表中的监控图表，可查看监控详情。


tomcat/2015
返回可用性监控
刷新
状态码
响应时间
1小时
6小时
12小时
1天
3天
7天
14天
2017-11-16 04:47:50 - 2017-11-16 10:47:
探针机器:
host-nYwZaxf-Qpo
响应时间
10
7.5
5
2.5
0
05:00
05:30
06:00
06:30
07:00
07:30
08:00
08:30
09:00
09:30
10:00
10:30
host-nYwZaxf-Qpo

6 如何创建容器服务Kubernetes版的Pod报警规则

本文为您介绍如何对容器服务Kubernetes的Pod及一组Pod的聚合指标设置报警规则。

背景信息

云监控新增容器服务Kubernetes版报警功能，通过监控容器服务的 CPU 使用率、入带宽等监控项，帮助您获取容器服务的使用情况。在您创建容器服务后，云监控自动开始对其监控，您登录云监控的容器服务Kubernetes版页面即可查看监控详情。当您对监控项设置报警规则后，即可在数据异常时收到报警通知。

创建容器服务Kubernetes版报警规则的准备工作

在创建容器服务Kubernetes版报警规则之前，建议您先创建好容器服务，然后在云监控中创建应用分组和报警联系人\报警联系组。

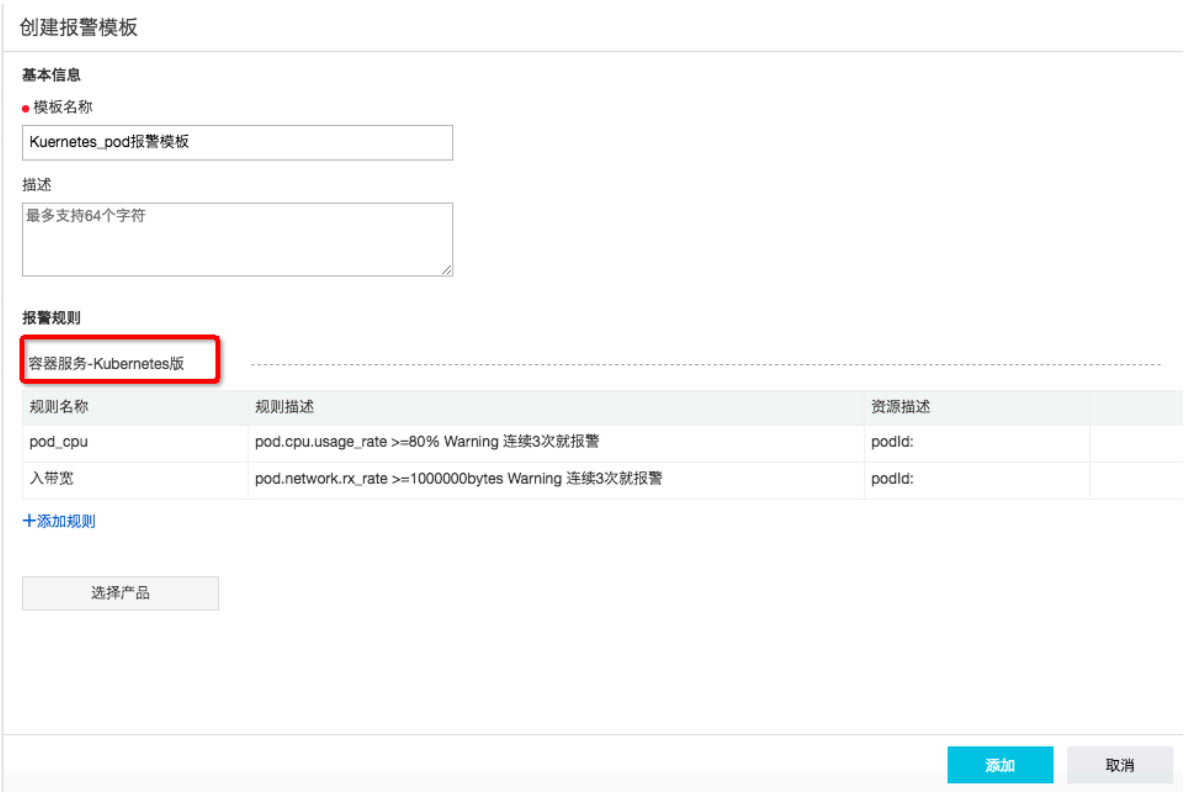
创建容器服务Kubernetes版报警规则的实施步骤

注意事项

- 监控数据最多保存31天。
- 最多可连续查看14天的监控数据。

创建容器服务Kubernetes版报警模板

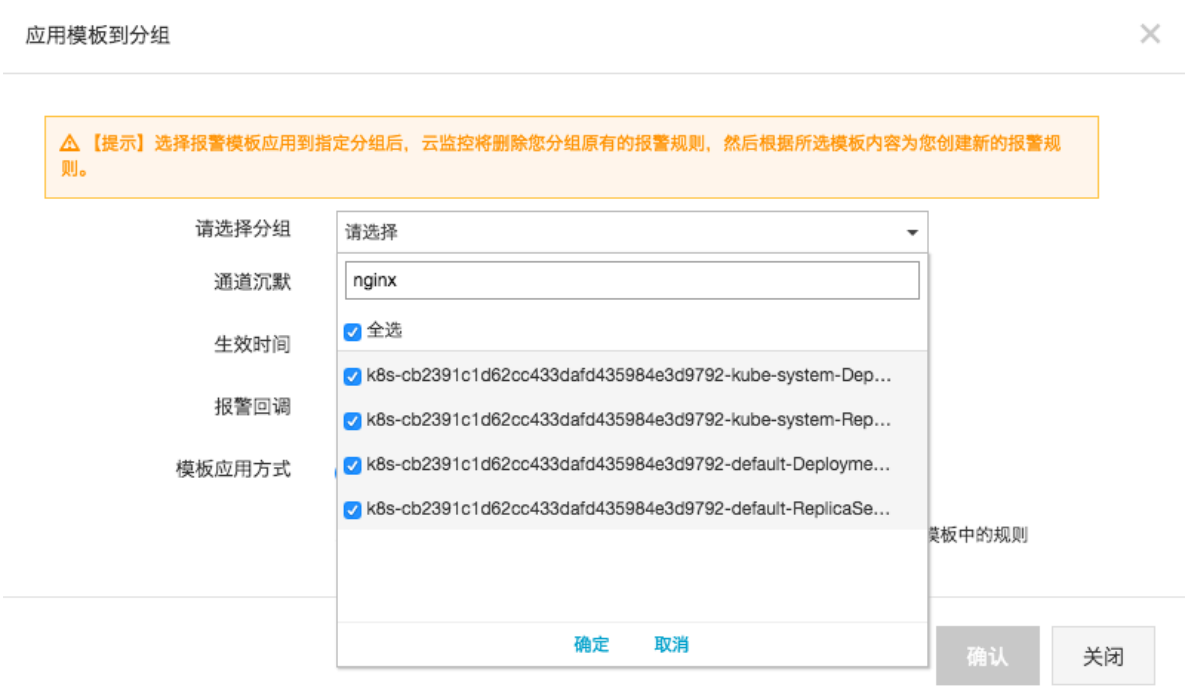
1. 登录云监控控制台。
2. 单击左侧导航栏中报警服务下的报警模板，进入报警模板页面。
3. 单击右上角的创建报警模板按钮，进入创建报警模板页面。



4. 配置模板信息：产品名称选择容器服务-Kubernetes版，并配置相关监控项。
5. 单击添加按钮，完成创建报警模板。

将模板应用在Kubernetes分组上

1. 登录云监控控制台。
2. 单击左侧导航栏中报警服务下的报警模板，进入报警模板页面。
3. 选择上面创建的容器服务Kubernetes版报警模板，单击操作栏中的应用到分组。



4. 选择需要创建报警规则的分组，单击确认按钮即可。

后续操作

Kubernetes应用分组的报警通知会默认发送给“云账号报警联系人”分组，如果您所有的Kubernetes应用报警通知只需要发送给同一个联系人组，则直接修改“云账号联系人”分组中的接收人即可。

直接修改联系人组的操作步骤：

1. 登录云监控控制台。
2. 单击左侧导航栏中报警服务下的报警联系人，进入报警联系人页面。
3. 单击报警联系组页签，切换到报警联系组页面。

4. 单击修改图标，进入编辑组页面，修改联系人后，单击确定按钮即可。

如果您不同的Kubernetes应用的报警通知，需要发送给不同联系人组，则需要进入应用分组，修改应用分组关联的报警联系人组。

通过应用分组修改关联联系人组的操作步骤：

1. 登录云监控控制台。
2. 单击左侧导航栏中应用分组，进入应用分组页面。
3. 通过搜索找到Kubernetes的分组，单击分组名称，进入分组详情页。

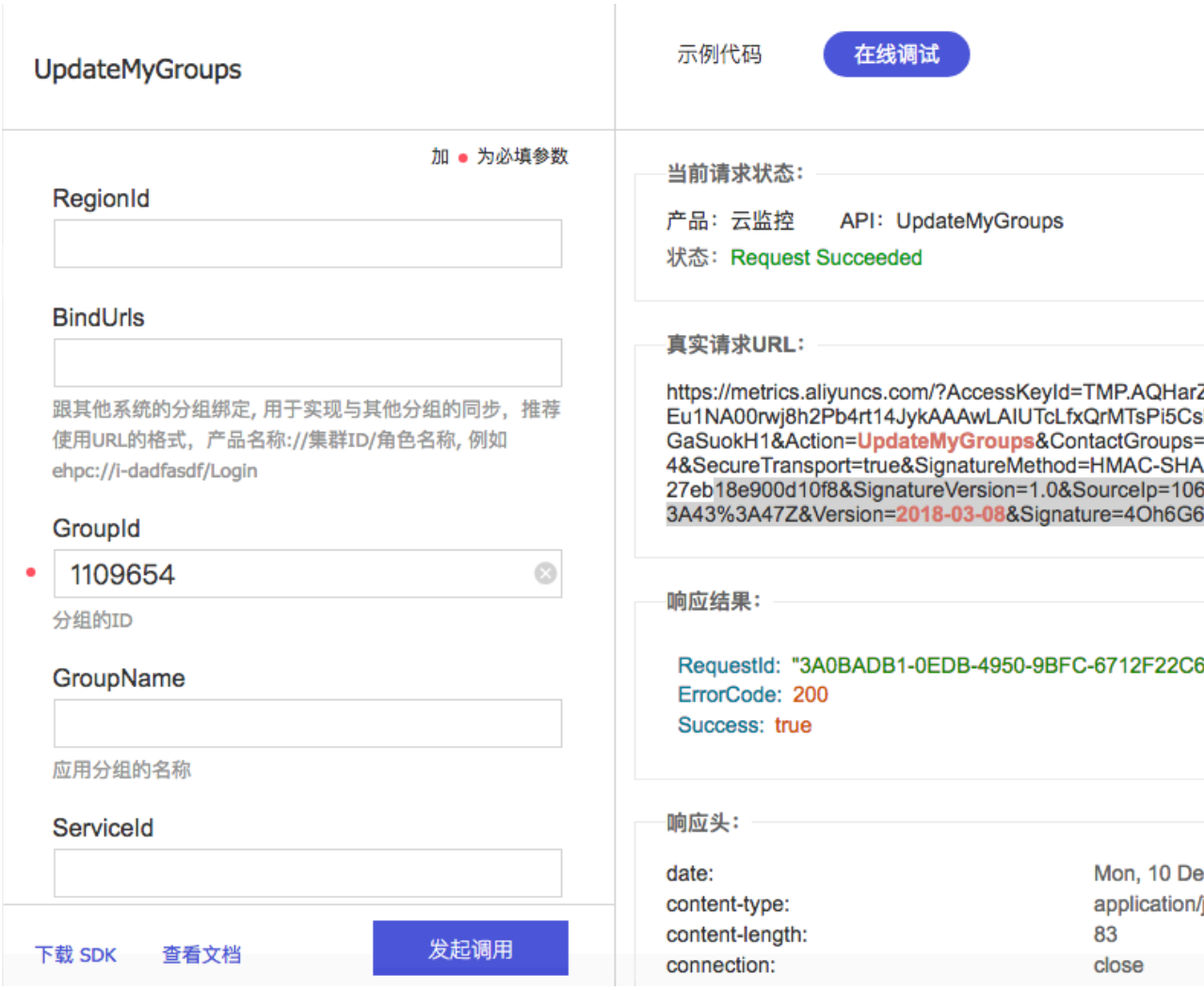
4. 单击修改图标，修改联系人组后，单击确定即可，该分组中的所有报警通知，将发送给新的报警联系人组。

如果您需要修改多个Kubernetes应用分组的联系人组，可以通过OpenAPI工具进行修改。

通过OpenAPI工具修改联系人组的操作步骤：

1. 登录*OpenAPI Explorer*。

2. 在左侧云产品列表中，单击云监控，在搜索框中输入UpdateMyGroups，在搜索结果中单击该API，进入修改应用分组的API调用页面。



3. 在GroupId输入框中，输入需要修改的分组ID。
4. 在ContactGroups输入框中，输入需要接收报警通知的联系人组。
5. 单击发起调用按钮即可。

7 ECS状态变化事件的自动化运维最佳实践

本文通过实践案例为您介绍云监控如何利用MNS消息队列实现自动化处理ECS状态变化事件。

背景信息

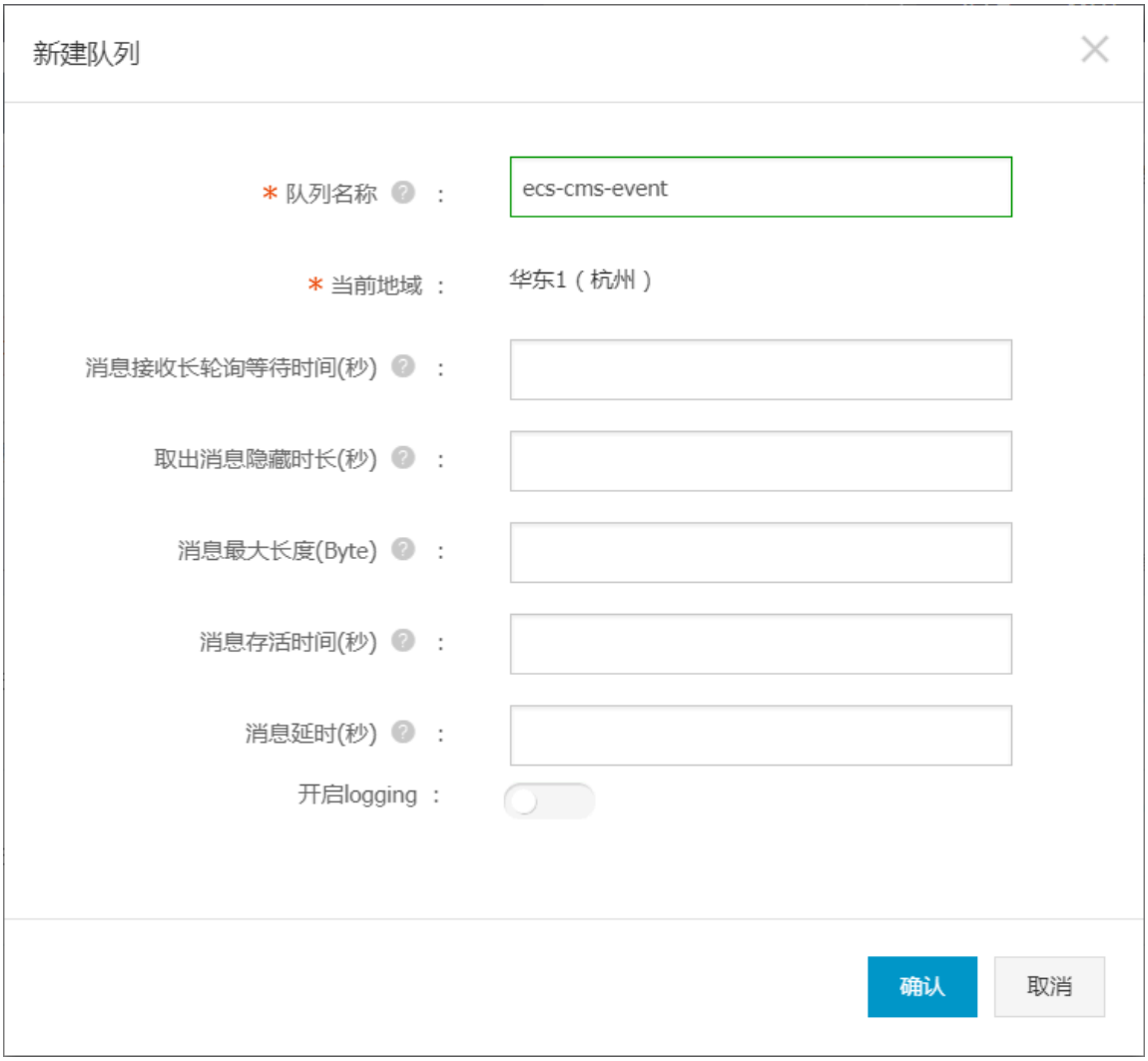
阿里云ECS在已有的系统事件的基础上，通过云监控新发布了状态变化类事件和抢占型实例的中断通知事件。每当ECS实例的状态发生变化的时候，都会触发一条ECS实例状态变化事件。这种变化包括您在控制台/OpenAPI/SDK操作导致的变化，也包括弹性伸缩或欠费等原因而自动触发的变化，还包括因为系统异常而触发的变化。

云监控以前发布的系统事件，主要针对告警后人工介入的场景，而这次新发布的事件属于正常类的信息通知，适合自动化的审计运维等场景。为了自动化处理ECS状态变化事件，云监控提供了两种主要途径：一种是通过函数计算，另一种是通过MNS消息队列。本文将为您介绍利用MNS消息队列自动化处理ECS事件的三种最佳实践。

自动化处理ECS状态变化事件的准备工作

- 创建消息队列

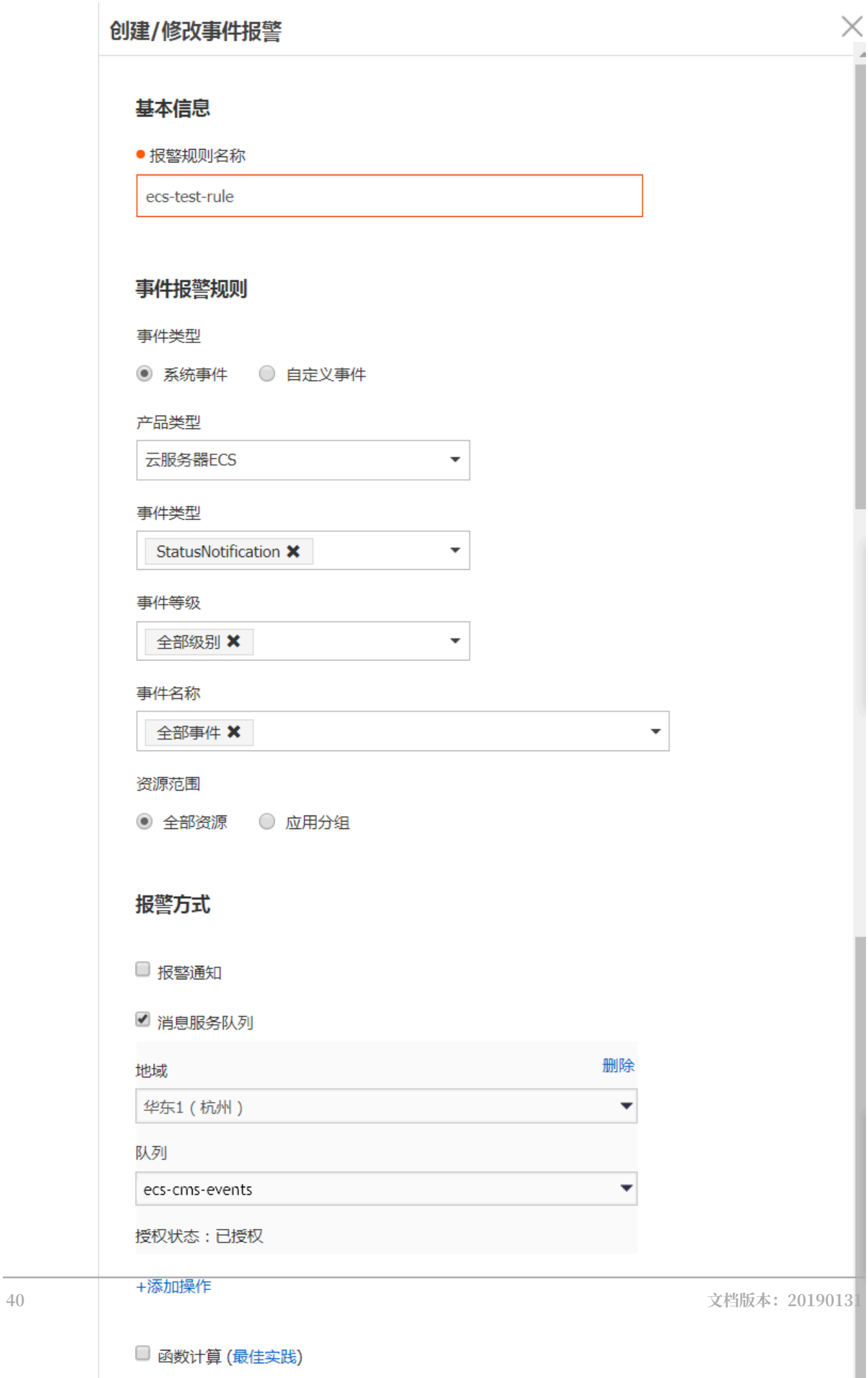
  1. 登录*MNS*控制台。
  2. 在队列列表页面，选择地域，单击右上角的创建队列，进入新建队列页面。



3. 输入队列的名称（例如“ecs-cms-event”）等信息，单击确认即可完成创建消息队列。

- 创建事件报警规则

1. 登录云监控控制台。
2. 单击左侧导航栏中的事件监控，进入事件查询页面
3. 单击报警规则页签，然后单击右上角的创建事件报警，弹出创建/修改事件报警对话框。


创建/修改事件报警
基本信息
报警规则名称
ecs-test-rule
事件报警规则
事件类型
系统事件
自定义事件
产品类型
云服务器ECS
事件类型
StatusNotification
事件等级
全部级别
事件名称
全部事件
资源范围
全部资源
应用分组
报警方式
报警通知
消息服务队列
地域
删除
华东1（杭州）
队列
ecs-cms-events
授权状态：已授权
+添加操作
函数计算 (最佳实践)

4. 在基本信息区域，填写报警规则名称，例如如“ecs-test-rule”。
5. 设置事件报警规则：选择事件类型为系统事件。
   - 产品类型、事件等级、事件名称：产品类型选择云服务器ECS，事件类型选择StatusNotification，其余按照实际情况填写。
   - 资源范围：选择全部资源时，任何资源发生相关事件，都会按照配置发送通知；选择应用分组时，只有指定分组内的资源发生相关事件时，才会发送通知。
6. 在报警方式中，选择消息队列，然后选择地域和队列（例如ecs-cms-event）。
7. 完成以上设置后，单击确定按钮即可完成创建事件报警规则。

· 安装Python依赖

本文所有的代码均使用Python 3.6测试通过，您也可以使用Java等其他编程语言。

请使用Pypi安装以下Python依赖:

- aliyun-python-sdk-core-v3>=2.12.1
- aliyun-python-sdk-ecs>=4.16.0
- aliyun-mns>=1.1.5

自动化处理ECS状态变化事件的实施步骤

云监控会把云服务器ECS所有的状态变化事件都投递到MNS里面，接下来我们需要通过编写代码从MNS获取消息并进行消息处理。

实践一：对所有ECS的创建和释放事件进行记录

目前ECS控制台无法查询已经释放的实例。如果您有查询需求，可以通过ECS状态变化事件把所有ECS的生命周期记录在自己的数据库或者日志里。每当创建ECS时，会首先发送一个Pending事件，每当释放ECS时，会最后发送一个Deleted事件。我们需要对这两种事件进行记录。

1. 编辑一个Conf文件。需包含mns的endpoint（可以登录MNS的控制台，在队列列表页，单击获取Endpoint得到）、阿里云的access key和secrect、region id（例如cn-beijing）以及mns queue的名字。

```
class Conf:
    endpoint = 'http://<id>.mns.<region>.aliyuncs.com/'
    access_key = '<access_key>'
    access_key_secret = '<access_key_secrect>'
    region_id = 'cn-beijing'
    queue_name = 'test'
    vsever_group_id = '<your_vserver_group_id>'
```

2. 使用MNS的SDK编写一个MNS Client用来获取MNS消息。

```
# -*- coding: utf-8 -*-
import json
```

```
from mns.mns_exception import MNSExceptionBase
import logging
from mns.account import Account
from . import Conf


class MNSClient(object):
    def __init__(self):
        self.account =  Account(Conf.endpoint, Conf.access_key, Conf
.access_key_secret)
        self.queue_name = Conf.queue_name
        self.listeners = dict()

    def regist_listener(self, listener, eventname='Instance:
StateChange'):
        if eventname in self.listeners.keys():
            self.listeners.get(eventname).append(listener)
        else:
            self.listeners[eventname] = [listener]

    def run(self):
        queue = self.account.get_queue(self.queue_name)
        while True:
            try:
                message = queue.receive_message(wait_seconds=5)
                event = json.loads(message.message_body)
                if event['name'] in self.listeners:
                    for listener in self.listeners.get(event['name
']):
                        listener.process(event)
                queue.delete_message(receipt_handle=message.
receipt_handle)
            except MNSExceptionBase as e:
                if e.type == 'QueueNotExist':
                    logging.error('Queue %s not exist, please create
 queue before receive message.', self.queue_name)
                else:
                    logging.error('No Message, continue waiting')


class BasicListener(object):
    def process(self, event):
        pass
```

上述代码只是对MNS消息进行拉取，调用Listener消费消息之后删除消息，后面的实践也会用到。

3. **注册一个Listener进消费指定事件。这个简单的Listener判断收到Pending和Deleted事件时，打印一行日志。**

```
 # -*- coding: utf-8 -*-
import logging
from .mns_client import BasicListener


class ListenerLog(BasicListener):
    def process(self, event):
        state = event['content']['state']
        resource_id = event['content']['resourceId']
        if state == 'Panding':
```

```
            logging.info(f'The instance {resource_id} state is {
state}')
        elif state == 'Deleted':
            logging.info(f'The instance {resource_id} state is {
state}')
```

Main函数可以这么写：

```
mns_client = MNSClient()

mns_client.regist_listener(ListenerLog())

mns_client.run()
```

实际生产环境下，可能需要把事件存储在数据库里，或者利用SLS日志服务，方便后期的搜索和审计。

实践二：ECS的关机自动重启

在某些场景下，ECS会非预期的关机，您可能需要自动重启已经关机的ECS。

为了实现这一目的，我们复用实践一里面的MNS Client，添加一个新的Listener。当收到Stopped事件的时候，对该ECS执行一个Start命令。

```
# -*- coding: utf-8 -*-
import logging
from aliyunsdkecs.request.v20140526 import StartInstanceRequest
from aliyunsdkcore.client import AcsClient
from .mns_client import BasicListener
from .config import Conf


class ECSClient(object):
    def __init__(self, acs_client):
        self.client = acs_client

    # 启动ECS实例
    def start_instance(self, instance_id):
        logging.info(f'Start instance {instance_id} ...')
        request = StartInstanceRequest.StartInstanceRequest()
        request.set_accept_format('json')
        request.set_InstanceId(instance_id)
        self.client.do_action_with_exception(request)


class ListenerStart(BasicListener):
    def __init__(self):
        acs_client = AcsClient(Conf.access_key, Conf.access_key_secret
, Conf.region_id)
        self.ecs_client = ECSClient(acs_client)

    def process(self, event):
        detail = event['content']
        instance_id = detail['resourceId']
        if detail['state'] == 'Stopped':
```

```
            self.ecs_client.start_instance(instance_id)
```

在实际生产环境下，执行完Start命令后，可能还需要继续接收后续的Starting/Running/Stopped等事件，再配合计时器和计数器，进行Start成功或失败之后的处理。

实践三：抢占型实例释放前，自动从SLB移除

抢占型实例在释放之前五分钟左右，会发出释放告警事件，您可以利用这短暂的时间运行一些业务不中断的逻辑。例如，主动从SLB的后端服务器中去掉这台即将被释放的抢占型实例，而不是被动等待实例释放后SLB的自动处理。

我们还是复用实践一的MNS Client，添加一个新的Listener，当收到抢占型实例的释放告警时，调用SLB的SDK。

```
# -*- coding: utf-8 -*-
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.request import CommonRequest
from .mns_client import BasicListener
from .config import Conf


class SLBClient(object):
    def __init__(self):
        self.client = AcsClient(Conf.access_key, Conf.access_key
_secret, Conf.region_id)
        self.request = CommonRequest()
        self.request.set_method('POST')
        self.request.set_accept_format('json')
        self.request.set_version('2014-05-15')
        self.request.set_domain('slb.aliyuncs.com')
        self.request.add_query_param('RegionId', Conf.region_id)

    def remove_vserver_group_backend_servers(self, vserver_group_id,
instance_id):
        self.request.set_action_name('RemoveVServerGroupBackendServers
')
        self.request.add_query_param('VServerGroupId', vserver_gr
oup_id)
        self.request.add_query_param('BackendServers',
                                     "[{'ServerId':'" + instance_id +
 "','Port':'80','Weight':'100'}]")
        response = self.client.do_action_with_exception(self.request)
        return str(response, encoding='utf-8')


class ListenerSLB(BasicListener):
    def __init__(self, vsever_group_id):
        self.slb_caller = SLBClient()
        self.vsever_group_id = Conf.vsever_group_id

    def process(self, event):
        detail = event['content']
        instance_id = detail['instanceId']
        if detail['action'] == 'delete':
```

```
            self.slb_caller.remove_vserver_group_backend_servers(self.
vsever_group_id, instance_id)
```

：

抢占型实例释放告警的event name与前面不同，应该是“Instance:PreemptibleInstanceInterruption”， mns_client.regist_listener(ListenerSLB(Conf.vsever_group_id)， 'Instance:PreemptibleInstanceInterruption')

在实际生产环境下，您可能需要再申请一台新的抢占型实例，挂载到SLB上，来保证服务能力。