应用性能管理 2.0 最佳实践

文档版本01发布日期2025-02-20





版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

商标声明

NUAWE和其他华为商标均为华为技术有限公司的商标。 本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束,本文档中描述的全部或部 分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,华为云计算技术有限公司对本文 档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文 档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址: 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编: 550029

网址: <u>https://www.huaweicloud.com/</u>



目录

1 最佳实践汇总	1
2 定位请求异常原因	3
3 调用链搜索 span 信息	5
4 接入 APM	11
4.1 云下业务接入云上 APM	
4.2 公网如何接入 APM	
4.3 Kubernetes 如何接入 APM	14
4.3.1 通过修改 deployment.yaml 接入 APM	
4.3.2 通过安装 Helm 方式接入 APM	
4.4 IDE 如何接入 APM	
4.5 Windows IIS 如何安装 JavaAgent	23
5 调用链与日志关联	26
6 结果查询页面内嵌到客户自建系统	28
7 如何使用 APM Profiler 定位性能问题	30
8 如何使用 Profiler 定位 OOM 问题	
9 CCE 容器部署 APM 探针支持对 APM 的 AK/SK 加密	40



本文汇总了应用运维管理(APM,Application Performance Management)常见应用 场景的操作实践,为每个实践提供详细的方案描述和操作指导,帮助用户轻松使用 APM。

表1-1 APM 最佳实践一览表

最佳实践	说明
定位请求异常原因	在外部请求激增、负载突变等场景下,极易出现应用性 能问题。APM作为云应用性能问题诊断服务,拥有强大 的分析工具,通过拓扑图、调用链可视化地展现应用状 态、调用过程、用户对应用的各种操作,快速定位问题 和改善性能瓶颈。
调用链搜索span信息	在分布式架构下,微服务之间的调用情况日趋复杂,在 外部请求响应变慢、部分请求异常等场景下,想要快速 定位哪个环节存在异常,您可以在业务在调用链路查询 页面,通过Traceld精确查询调用链路详细情况,或结合 多种条件筛选查询调用链路。
接入APM	云下业务接入云上APM
	用户需要云下业务接入云上APM,但云专线无法打通网 络。因此,用户需要用代理的方式接入APM,本文介绍 了云下业务如何接入云上APM的操作过程。
	公网如何接入APM
	介绍公网如何接入APM的操作步骤。
	Kubernetes如何接入APM
	1. 如果使用CCE或开源Kubernetes部署应用,可以 <mark>通过</mark> 修改deployment.yaml接入APM,即可在指标、调 用链界面上查看应用情况。
	2. 如果使用通用的开源Kubernetes部署应用,可以 <mark>通过</mark> 安装Helm方式接入APM,即可在指标、调用链界面 上查看应用情况。

最佳实践	说明
	IDE如何接入APM 介绍IDE如何接入APM的操作步骤。
	Windows IIS如何安装JavaAgent Windows IIS如何安装JavaAgent。
调用链与日志关联	应用性能管理(APM)与云日志服务(LTS)关联,您可 以在 LTS中关联调用链的TracelD 信息,当应用出现故障 时,可以通过调用链的TracelD 快速关联到业务日志,及 时定位分析并解决问题。
结果查询页面内嵌到客 户自建系统	APM支持将界面嵌入到客户自建系统。通过IAM服务的 联邦代理机制实现用户自定义身份代理,再将登录链接 嵌入至客户自建系统实现无需在华为云官网登录就可在 自建系统界面查看。
如何使用APM Profiler 定位性能问题	APM Profiler是一种持续性能剖析工具,可以帮助开发者 准确找到应用程序中消耗资源最多的代码位置。
如何使用Profiler定位 OOM问题	介绍如何使用APM Profiler定位OOM问题。
CCE容器部署APM探针 支持对APM的AK/SK加 密	在CCE容器中部署APM探针时,会出现明文的AK/SK。出于安全考虑,需要对AK/SK进行加密。本文介绍如何加密 明文AK/SK的操作步骤。

2 定位请求异常原因

背景信息

在外部请求激增、负载突变等场景下,极易出现应用性能问题,比如外部请求响应变 慢、部分请求异常等。快速识别发现、定位处理应用性能问题成为越来越常见的日常 运维场景。

APM作为云应用性能问题诊断服务,拥有强大的分析工具,通过拓扑图、调用链可视 化地展现应用状态、调用过程、用户对应用的各种操作,快速定位问题和改善性能瓶 颈。

例如,通过APM拓扑功能可视化服务间的调用关系,迅速找到有问题的实例;通过 APM调用链功能下钻到服务内部,根据出现问题的方法调用链路,确认问题根因。

适用场景

- 应用日常巡检,监控应用时延、吞吐量、错误数等性能指标。
- 应用异常调用快速定位。

操作步骤

- 步骤1 登录应用性能管理控制台。
- 步骤2 在左侧导航栏选择"应用监控 > 指标"。
- **步骤3**选择"接口调用"页签,进入监控页面,查看接口调用页面中各类指标,调用次数、 错误次数、时延等信息。

图 2-1 查看接口调用

1948	💷 LubanAPM /	🔄 App / 👸 apm-	access / 🖂 wulan	/ 更多节点	-							
应用指标 ③	括扑 接口调用	基础监控	异常 外部调用	通信协议	数据库	Web容器	消息队列					
□ 虚用: ・ ・ ・	① 全部案例	• Ø	• URL <u>始控</u>	*				2023/04/19 16:44 0	3MT+08:00 2023/04/	19 17:04 GMT+08:00	 if20910 	• C
 											0	
◎ 标签: 全部 ▼	接口維度汇总	状态码确度汇总	集群编度汇总	認知								
请输入组件或环境名称 Q	接口機成定总	最近一笔数据 2023/04	/19 17:01:00 GMT+08.0	10								
	Q 16.7.898	前两列内容										۲
🖃 💷 LubanAPM												
- 🚍 ն App	url			meth	od	调用次数	平均响应时间(错误数	量大并发	昼鸞凋用(ms)	apdex	
😑 😚 apm-access	IX. /apm2/he	alth/v1/health-check	ĸ	GET		144	0.31	0	1	0	1.00	
🖾 wulan 🔛 🏵 🛈										10 景凤 :	共1条 <	1 >

步骤4 单击出现问题的url请求,进入调用链搜索界面。

图 2-2 接口详情

接口維度汇总	1.总 状态码维度汇总 集群维度汇总 总宽											
接口维度汇总 -最近一笔数据 2022-01-07 10:06:35												
url	method	调用次数	平均响应	错误数	最大并发	最慢调用(ms)	0ms-10ms	10ms-10	100ms-5	500ms-1s	1s-10s	10s-n
/user/login	POST	2	128309.00	2	4	128332	0	0	0	0	0	2
/user/validate	POST	2	127244.50	2	4	127264	0	0	0	0	0	2

步骤5 在调用链搜索界面,查看失败/高时延调用链。

图 2-3 查看调用链

调用链 ⑦			© 15209% ▼ C < 177 min
查询条件	请	空 ⓒ 6 乐记录	
92%	default (BBA)	· 清水水功 · 清水大牧 · 明正时间 · · · · · · · · · · · · · · · · · · ·	
区域	● 編东	 <u>POST_User/nalidate</u> 2022/01/07 1007:14 GMT-0600 	500 127354 ms Toce ID: 60-1641521234348-20
应用	vmall-user-service	BOST UNBERVAILED Second (1) - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	500 127251 ms Trace ID: 60-1641521229582-17
环境	default	POST /user/validate	9 500 127221 ms
新生	apm-demo(192.168.0.222)	2022/01/07 10:05:38 GMT+08:00	Trace ID: 60-1641521138225-14
	Bart 1101	> POST Austrivalidate 2002/01/07 10:05:29 GART-REDO	500 127269 ms Trees ID: 60-1641521125472-9
1639	0	> POST user /relister zeozynym fisikki europa	500 127224 ms Trace ID: 60-1641521040154-8
	/user/validate	> POST /user/walkdate	9 500 127264 ms
请用方法	POST	2022/01/07 10:03:49 GAIT+66:00	Trace ID: 60-1641521029374-1
秋志田		6 祭记录	

步骤6 单击url,获取调用链详细信息,确定问题根因。

图 2-4 调用链详情

《 認知 Trace ID 60-1641521138226-14 約約15/形 127228ms	点形数 3个 消度 5 篇 开始时间 2022/01/07 10:05:38:226 GMT+08:00				
	wer und peakstavice und das avere 1777280(0007)(genetation, 1777280(0707)(genetation, 177780(0707)(genetation, 177780(
All from the second model and a later (1) as small one sector	defects (7) an annual dae constant (defects (7)				
and shows a management of the management of		10 0:00	410 1920	anes	Terms
Image: Source in the second	enters (J) • • mercaneterine (neuronal) metalogi 0 ms 12 127228 ms	28 ms 高用 vmail-product-se	API 类型 Torrcat	4802 R	更多信息 宜若 •
200 Jan • • • • • • • • • • • • • • • • • • •	300.01 • • • • • • • • • • • • • • • • • • •	28 ms B/H vmail-product-se vmail-product-se	API 武型 Torrcat Jdkhttpc	####2 22 22	更多信息 查查 • 查查 •
20 / Part ************************************	300.01 • • • • • • • • • • • • • • • • • • •	28 ms BH vmail-product-se vmail-product-se vmail-user-service	API 222 Torncat Jekhttpc. Torncat		更多位度 成型 * 成型 * 成型 *
20 years ************************************	300.00 * conservations (mon.1) 2020.00 * conservations (mon.1)	28 ms Biff vmail-product-se vmail-product-se vmail-user-service vmail-user-service	API 328 Torncat Idkhttpc. Torncat Idkhttpc.	####28 © © © ©	更多信息 重要 * 重要 * 重要 * 重要 *
Set Set and public set (not (n)) Set (not (n)) Set (not (n)) Set Set (not (n)) Set (n)) Set (n)) Set (not (n)) Set (n)) Set (n)) Set (n)) Set (n)) <td< td=""><td>(1) • • • • • • • • • • • • • • • • • • •</td><td>28 ms Biff vmsl-product-se vmsl-product-se vmsl-user-service vmsl-user-service vmsl-user-service</td><td>API 2021 Torncat Jokhttpc Torncat Jokhttpc</td><td>48828 20 20 20 20 20 20 20 20 20 20 20 20 20</td><td>E8449 122 * 122 * 122 * 122 * 122 *</td></td<>	(1) • • • • • • • • • • • • • • • • • • •	28 ms Biff vmsl-product-se vmsl-product-se vmsl-user-service vmsl-user-service vmsl-user-service	API 2021 Torncat Jokhttpc Torncat Jokhttpc	48828 20 20 20 20 20 20 20 20 20 20 20 20 20	E8449 122 * 122 * 122 * 122 * 122 *

----结束

3 调用链搜索 span 信息

背景信息

在分布式架构下,微服务之间的调用情况日趋复杂,在外部请求响应变慢、部分请求 异常等场景下,想要快速定位哪个环节存在异常,您可以在业务在调用链路查询页 面,通过Traceld精确查询调用链路详细情况,或结合多种条件筛选查询调用链路。

操作步骤

- 步骤1 登录APM控制台。
- 步骤2 在左侧导航栏选择"应用监控 > 调用链",进入调用链界面。

步骤3 输入如下查询条件,单击"查找Trace",页面右侧展示查找结果。

图 3-1 调用链查询结果

调用链 ⑦				2023/08/16 15:37 GMT+08:00 — 2023/08/16 15:57 GMT+08:00 ③ 送20分钟 * C 萨用户搁着
查调条件	APM (BEL)	潮空	 ○ 19961 新记录 ● 清水成功 ● 清水成功	< 1 2 3 4 5 6 … >
ĭ	•	*	> POST/sem2/masker/t/frear/beat 02 2022-08-16 15:57:05	200 6 ms Trace ID: 1005422-1092172025205-105925
	更多 •		> POST/gen2/master/v1/register [2] 2023-08-16 15:57:04	200 15 ms Trace ID: 1005422-1092172824942-108924
URL	Real URL	٣	> <u>POST/sem2imster/v1/mearbeat</u> 2023-08-10 15:57:04	200 5 ms Trace ID: 1005422-1692172624800-186923
稻油	0 0		> E051_isen2imatkr/t/fmarkeat @ 2023-08-10 15:57:04	200 5 ms Trace ID: 1006422-1092172824723-180922
调用方法	POST	•	> <u>POST/serv2/mesthety1/mearbeat</u> 2023-08-16 15:57:04	200 12 ms Trace ID: 1005422-1692172524499-105921
07.02.59	19-1705020964,33790		> <u>POST/icm2imsterv1meetbest</u>	200 2 ms Trace ID: 1005422-1692172624490-166920
响应时间 是否异常	ms ·	ms •	> <u>POST.ison2imatket/16martbeat</u> 2023-08-10 15:57:04	200 9 ms Trace ID: 1005422-1692172824468-188919
Trace ID		0	> <u>POST/sem2imenter/v1/mearbeat</u> 2023-08-16 15:57:04	200 5 ms Trace ID: 1005422-1692172524455-168918
	④ 其他靈術条件		> POST/gen2/master/stmartbeat 02 2023-08-16 15:57:04	200 16 ms Trace ID: 1005422-1692172624146-169917
重线Trace			> <u>POST/icen2/imaster/v1/hear/beat</u> 2023-08-10 15:57:04	200 3 ms Trace ID: 1005422-1692172824128-168916

表 3-1 调用链查询条件

查询条件	具体含义	必填/选填
应用	调用链所在应用。	必填
区域	调用链所在区域。	必填
组件	调用链所在组件。	选填

查询条件	具体含义	必填/选填
环境	调用链所在环境。	选填
实例	调用链所在实例。	选填
URL	调用链的URL,分为Rest URL和Real URL两种 搜索条件:Rest URL为restful风格的URL, URL中带有变量名称,如/apm/get/{id};Real URL为实际访问的URL。	选填
精确	对URL是否精确匹配,开启状态下为精确查询 URL,不开启则进行模糊查询。	选填
调用方法	调用链的HttpMethod。	选填
状态码	调用链返回的HTTP状态码。	选填
响应时间	调用链的响应时间范围,可以填写最小响应时 间和最大响应时间搜索调用链,两个值都可以 为空。	选填
是否异常	调用链是否有异常。	选填
Trace ID	调用链的TracelD,填写该搜索条件后,其他搜 索条件全部失效,只根据该TracelD搜索。	选填

步骤4 单击"其他查询条件",展示"自定义参数"、"全局Trace ID"以及"应用码"三个 查询条件。

图 3-2 其他条件

查询条件	清	≅ ⊙
应用	vmall-demo (默认)	
区域	•	
	更多 🔻	
URL	Rest URL -	
精确	• •	
	更多 ▼	
响应时间	ms - ms	
是否异常	全部 ▼	
Trace ID		?
自定义参数		
全局Trace ID		
应用码		
查找Trace		

表 3-2 调用链查询条件

查询条件	具体含义	必填/选填
自定义参数	已配置url监控项的拦截header指定key值、拦 截url参数指定key值、拦截cookie指定key值参 数后,在这里可以设置key=value进行搜索。	选填
全局Trace ID	调用链的全局TracelD,填写该搜索条件后,其 他搜索条件全部失效,只根据该全局TracelD搜 索。	选填
应用码	已配置url监控项的业务code采集长度限制、解 析业务code的key、业务code的正确值参数 后,会采集相应的应用码,这里可以根据应用 码进行搜索。	选填

自定义参数

使用方法

a. 配置url监控项的拦截header指定key值、拦截url参数指定key值、拦截cookie 指定key值参数。具体方法参见配置url监控项。

- b. 在"自定义参数"后的框中,填写对应的参数以及参数值。
- c. 单击"查找Trace",页面右侧展示查找结果。

图 3-3 自定义参数查询结果

MHH IF O				202300-10 15 00 0M1+00 00 - 202300-10 15 20 0M1+00 00 00 32200344 • C
查询条件))空 (○ 19740 祭记录 ■ #4757 ■ #4757 ■ #57555 月 用台2555 月 用台2555 月	< 1 2 3 4 5 6 … ≯
应用	APM (既认)	¥		
15.18 ⁶	• 1	٣	> <u>FOST Learn24pt/set/set/set/10/and</u> IS 2023-08-16-15-28-28	200 17 ms Trace ID: 1725503-1692170371766-125744
	更多 •		> POST Atem2/conformed/v1/troaster R 2023-08-16 15:28:28	200 5 ms Trace ID: 1725474-1692170908342-46156
URL	Rest URL	Ŧ	POST Jacon 2/30/Video/metric/il/trend 2023-08-16 15.28.28	200 12 ms Trace ID: 1725503-1692170871596-125742
6010			> EOST /sem2seivisevimetric/vitrand IX 2023-08-16 15 28 27	200 92 ms Trace ID: 1725503-1692170871290-125740
	<u>₩</u> \$ •		POST-/sem2/conformd/v1/register 2023-08-16 15:28:27	200 6 ms Trace ID: 1725474-1682170997921-46155
- 响应时间 - 黑云母军	ms -	ms	POST Japan 2/mszter/v1/tregister 2023-08-16 15.28.27	• 200 16 ms Trace ID: 1005422-1692170907792-184188
Trace ID	ak 10°	0	FORT_instm2/ac/instant/metric/rithmenf IR 2023-08-16 15 28 27	200 13 ms Trace ID: 1725503-1692170871047-125738
自定义参数	httpMethod=POST		> EOST Jaen Zaolivawimetricvititrand IR 2023-08-16 15:28:27	200 148 ms Trace ID: 1725503-1092170870097-125737
全局Trace ID			POST Jacon 2/30/viewstmetricy1/thrend 2023-08-16 15/28/27	200 69 ms Trace ID: 1725503-1692170870372-125735
应用码			> POST-teon/260/viewimehic/vitrend IR 2023-08-16 15:28:26	200 11 ms Trace ID: 1725503-1892170870198-125733
查找Trace			EOST.Jaem2JaeViewimetrich1ftrand R 2023-08-10 15:28:26	200 16 ms Trace ID: 1725503-1092170070195-125732

- 全局Trace ID使用方法 使用方法
 - a. 单击待查看的调用链前的 >, 查看全局Trace ID。

图 3-4 获取全局 Trace ID

POST/apm2/apm2/apm2/apm2/apm2/apm2/apm2/apm2			200 5 ms Trace ID: 1726222-1692171632315-13082	
	組件 实例 Real URL	apm-ap/Ivu/an apm2-apm-api-7b6tcftc96-h552((**********************************	金局Trace ID v-1739115-1692171595585-139022	

- b. 在"全局Trace ID"后的框中,填写全局Trace ID。
- c. 单击"查找Trace",页面右侧展示查找结果。

图 3-5 查询全局 Trace ID 结果

查询条件	清空	(◎ 1 条记录	
应用	APM (BSJ.)		請求成功 ■ 請求失效 助空时间 = 产生时间 =	
×1	• •		POST_sen2iseViewimitifizivititend R 2023-00-16 15:40:32	200 5 ms Trace ID: 1726222-1692171632315-13082
	<u>9</u> 2 ▼		Elite april-april /rolan 家門 apri2-april april-2-april april-2-april april 2-april 3-april 2-april 3-april 2-april 3-april 2-april 3-april 2-april 3-april 2-april 3-april	
URL	Rest URL +		Real URL 🗍 (POST) /apm2/api/View/metricv1/thrend	
精神	0		1 条记录	
	更多 ▼			
响应时间	ms - ms			
是否异常	全部 👻			
Trace ID		0		
自定义参数				
全局Trace ID	v-1739115-1692171595585-139(
应用码				
查找Trace				

• 应用码使用方法

使用方法

a. 配置url监控项的业务code采集长度限制、解析业务code的key、业务code的 正确值参数。具体方法参见<mark>配置url监控项</mark>。

图 3-6 url 监控项

f (URL监控) 监控配置 、、				*, X	
url采集配置	規整方式 ① 新增	匹配表达式	规整后展示url	操作	
uri配置	url ④ 新增	响应时间间的		操作	
采集黑名单配置	匹配方式 ④ 新増	匹配表达式		操作	
业务code采集新	- 0 +				
解析业务code的	ikey bizcode 日 resultcode 日 ④ 新環				
业务code的正确	e CBC.0000 ロロロン ロロロロロロロロロロロロロロロロロロロロロロロロロロロロロ				
错误状态码定义	 ○ 秋恣弱400以上統计为撤误 ● 状況 図 使用默认值 ② 	5码500以上统计为错误			
Url自动规整	● 杏 ○ 是				
				30	定 取消

- b. 在左侧导航栏选择"应用监控 > 调用链",进入调用链界面。
- c. 单击 🖪 ,查看对应的业务code的值。业务code的值即"应用码"。

图 3-7 查看业务 code

洞用鏃 ①			2023/08/21 14 53 GMT+08/00 — 2023/08/21 15 13 GMT+08/00 の 近203/3HP ・ C IF用中
查询条件	清空	© 2089275 \$+22m	< 1 2 3 4 5 6 … >
应用	G H *	● 備水成功 ● 備水共取 端回时间 # 产生时间 #	
(Est)	• :	GET /red/obc/contaisearchouer/service/r/i/commonsearch/UK 2023-06-21 15 13 01	200 1 ms Trace ID: 1452385-1692601091178-1172624
	118 •	> SELTAN 2023-86	× 200 69 ms H0214-1602001972410-4290
URL	Rest URL *	查看词用参数 2023-66	260 0 mi 103-1992601901097-125770
精神	0	httpMethod:	[] 現利命令 200 5 ms
		2023-60 Fost	v.1715247-1692001600081- 1557439
		POSTA x-cse-src-microservice:	200 i 3ms 日期第二章 300年年日 12501 12501 1250
明白的间	ms - ms	> GET/he CESContractService	200 0 ms
是四异常	金部 ▼	bizCode:	□ 現100 · 1
Trace ID		D SELC 2023-68 CBC.0000	200 0 ma 105-195260195092-164395
	 再位重用条件 	> <u>GET /rec</u> prepareTime:	□ 現地会中 200 i 1 ma □ 20.102201050005-4240555
		GET.tre	200 6 ms
122CTrace		handlersRequestTime:	1010440-1022001350704-57
		> POSTA 2023-66 1m	200 4 mm v-1715527-1695001974972- 1556977
		> GELTAR handlersResponseTime:	□ 思知会令 200 i 0 ma 163-1692601950043-164350
		Cos > POST # 2023-00-cs releases	200 3 ma Have no 11 Have 17. 1092/01/30/042, 101/2340
		> POST /** 10 10 10 10 10 10 10 10 10 10 10 10 10	200 100 ms Town ID - 1714570, 105290 1050704

d. 在"应用码"后的框中,填写应用码。

图 3-8 查找 code 对应的调用链

调用链 ②		2023/08/21 14 54 GMT+00.00 — 2023/08/21 15:14 GMT+00.00 ② 近2059時 🔹 🖸 印用中部
亚 词条件	第至 ○ 610418 我已就 前来忘去	< 1 2 3 4 5 6 >
Est •	 POST-IntellibricitionsRepairies/ValueInterview/ValueI	200 2 ms Trace ID: 1714057-1022002055004-194151
11.0 ·	> GRT hentibolockonemionaholidataeceana/hashkilidataec 2023-08-21 15:14:15	200 1 mm Trace ID: 1601963-1602962054800-976764
URL Rest URL	POST Institutionmentsbetroenicer/situatimentsuer.coll.Edl 2022-06-21-15-14-14	200 6 ms Trace ID: v=1715247-1852822054611-
1830 O	> DOST Institution communicativity and a statement of the	1399478 2001 15 mm Trace ID: v-1715241-1692602054612-
Es ·	> POST Institucióticomanstalessenice/Hibbs_comission/check/ (\$ 2023-06-21 15:14:14	135616 2001 1 ms Trace ID: v-1715241-16226054612-
構築時間 ms -	ms POST hukkickickonstandukklaisseninekklaifistopatoffisipatoffisiopatoffisiopatoffisiopatoffisiopatoffisiopatoffisiopatoff	130816 2001 3 mi Trace ID: 1715182-1692562052586-96536
Trace ID	> 9ET institubilitieseminantalainesemine/ritubilitiepeinessetespilitien R ○ 2022-06-21 is 14 14 R	200 5 ms Trace ID: v-1715241-1822862654866-
自定义参数	> POST hink to be constantial ensemble in the control cardon environment in the 2023-00-21 15:14:14	1309915 200 23 mm Trace ID: v-1715327-16925902054614-
土間Trace ID	> <u>QET institubultikommandatalessen/ise/s/instanter/details/justamential/</u>	1300031 800 15 ms Trace KD: v-1715241-1822802054586-
四用65 CBC.0660	> POST hmit kolchbereducktalamenisek liefferingestelleningestelleningestelleningestellening für 2022-00-21 15:14:14	1330914 201 2 mi Trace ID: 1715283-1692592054595-350859
M192Trace	> <u>QET and/clob/bb/db/db/db/db/db/services/hite/access/bask/hite/access/ba</u>	200 2 ms Trace ID: 1631063-1632802054555-076761
	SET I welicholdssambaremysen/set/iluntamet/secoutilizet/gr IB 2022-00-21 1514-14	200 3 mm Tince ID: 1715342-1682902054446-1258717
	SET institution manufactures environment for a set of the set of t	200 22 ms



----结束



4.1 云下业务接入云上 APM

背景信息

用户需要云下业务接入云上APM,但云专线无法打通网络。因此,用户需要用代理的 方式接入APM,不知道如何操作。

配置方法

接入APM的机器与APM服务网络无法连通,可以接入代理。

步骤1 配置代理

- 1. 登录AOM 2.0控制台。
- 2. 在菜单栏选择"采集管理",进入"采集管理"界面。
- 3. 在左侧导航栏中,选择"UniAgent管理 > 代理区域管理",进入代理区域管理页 面。
- 4. 单击"添加代理机",配置相关参数信息。

图 4-1 添加代理机

添加代理机

★ 代理区域	test33	•	
<mark>★</mark> 主机	uniagent-test-0003	•	С
	温馨提示: 这里的主机必须是已安装的Uni	Agen	ıt主机。
★ 代理IP]
<mark>★</mark> 端口			

表 4-1 添加代理机参数说明

参数	说明	示例
代理区域	选择已创建的 <mark>代理区</mark> <mark>域</mark> 。	region
主机	选择已安装的UniAgent 主机。	-
代理IP	配置代理机的IP地址	-
端口	端口号,必须小于或等 于65535	-

5. 单击"确认",完成代理机添加。

步骤2 配置JavaAgent

1. 将javaagent下载到需要接入APM机器的任意目录。

示例命令:

curl -O https://xxx/apm-javaagent-x.x.x.tar Agent 2.4.1下载方法: curl -k https://apm2-javaagent-xx-xx-x.obs.xx-xxx-4.xxxx.xxx/ apm_agent_install2.sh -o apm_agent_install.sh && bash apm_agent_install.sh -ak {APM_AK} -sk {APM_SK} -masteraddress https://xx.xx.xx:41333 -obsaddress https://apm2-javaagent-xx-xxxxx.obs.xx-xxxx-x.xxxx.xxx -version 2.4.1; history -cw; history -r

2. 执行tar命令解压javaagent。

示例命令:

tar -xvf apm-javaagent-x.x.x.tar

3. 修改javaagent中的apm.config配置文件。将apm.proxy写入配置文件中,如下图 所示。

图 4-2 配置文件

■ *apm.config - i∃事本	-	×
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)		
master.address=https://1 1333		1
access.key=ag71		
secret.key=vRv		 Shu
log.level=debug		
event.thread.count=3		
apm.proxy=		
#access.address=		
app.name=helloworld		
#instance.name=		
#env={{env}}		
#env.tag=		
#decrypt.className=com demo.DecryptDemo		
#decrypt.methodName=decrypt		

- Agent 2.4.1及之后版本支持采用代理接入。格式: apm.proxy=ip:port(此处为aom界面的ip:port)。
- 获取AK/SK的操作步骤,请参见<mark>访问密钥</mark>。
- 获取master.address的操作步骤,请参见<mark>探针接入地址</mark>。

步骤3 重启应用。

 修改java进程启动脚本。
 在服务启动脚本的java命令之后,配置apm-javaagent.jar包所在路径,并指定 java进程的组件名。
 添加-javaagent参数示例: java -javaagent:/xxx/apm-javaagent/apm-javaagent.jar=appName={appName}

2. 重启应用。

----结束

4.2 公网如何接入 APM

前提条件

- 1. 已购买华为云弹性云服务器ECS作为跳板机。
- 2. 弹性云服务器已绑定弹性IP地址。
- 推荐CentOS 6.5 64bit及其以上版本的镜像, 最低规格为1vCPUs | 1GB, 推荐规 格为2vCPUs | 4GB。
- 4. 推荐使用iptables作为跳板机转发实现。

操作步骤

请先在华为云上购买一台弹性云服务器作为跳板机,然后执行如下操作。

- 步骤1 登录弹性云服务器,修改跳板机ECS的安全组规则。
 - 1. 在ECS详情页,单击安全组页签,进入安全组列表页。
 - 2. 单击具体的安全组名,单击"更改安全组规则",进入安全组详情页。
 - 在该安全组详情页,单击"入方向规则 > 添加规则",按表4-2添加安全组规则。

表 4-2 安全组规则

方向	协议	端口	说明
入方向	ТСР	41333,41335	JavaAgent发送数据到跳 板机的端口列表。

- 步骤2 获取APM上报地址,参见探针接入地址。
- 步骤3 以root用户登录跳板机,执行iptables转发命令。

如果没有iptables相关服务,需要先安装,命令如下。

yum install iptables-services systemctl stop firewalld.service systemctl disable firewalld.service systemctl mask firewalld.service

- 开启数据转发功能。

 #编辑文件
 vim /etc/sysctl.conf
 # 增加一行
 net.ipv4.ip_forward=1
 # 使数据转发功能生效
 sysctl -p

 Y本机、(跳板机,) 端口的数据
- 将本机(跳板机)端口的数据转发到上报到APM的端口。
 # 编辑文件
 vim /etc/sysconfig/iptables
 # *filter新增

-A INPUT -p tcp -m state --state NEW -m tcp --dport 41333 -j ACCEPT -A INPUT -p tcp -m state --state NEW -m tcp --dport 41335 -j ACCEPT # *新增NAT规则 -A OUTPUT -p tcp --dport 41333 -j DNAT --to-destination {上报到APM的主机IP}:41333 -A PREROUTING -p tcp --dport 41333 -j DNAT --to-destination {上报到APM的主机IP}:41333 -A POSTROUTING -d {上报到APM的主机IP}/32 -p tcp --dport 41333 -j SNAT --to-source {跳板机ip} -A OUTPUT -p tcp --dport 41335 -j DNAT --to-destination {上报到APM的主机IP}:41335 -A PREROUTING -p tcp --dport 41335 -j DNAT --to-destination {上报到APM的主机IP}:41335 -A POSTROUTING -d {上报到APM的主机IP}/32 -p tcp --dport 41335 -j SNAT --to-source {跳板机ip} # 如果存在以下规则,需要删除 -A FORWARD -j REJECT --reject-with icmp-host-prohibited 3. 重启iptables。 systemctl restart iptables 验证端口转发是否成功。 4. curl -kv https://{跳板机ip}:41333 curl -kv https://{跳板机ip}:41335 步骤4 修改javaagent中的apm.config配置文件。 master.address=https://{跳板机公网ip}:41333 access.address={跳板机公网ip}:41335 步骤5 重启应用。

----结束

4.3 Kubernetes 如何接入 APM

4.3.1 通过修改 deployment.yaml 接入 APM

如果使用CCE或开源Kubernetes部署应用,可以通过修改deployment.yaml接入 APM,即可在指标、调用链界面上查看应用情况。

前提条件

部署APM Agent时,必须确保接入APM的机器与APM服务网络连通,Agent才能正常工作。

可使用Telnet命令测试目标机器与APM服务器网络是否连通。例如,以检查华北-北京 四区域的连通性为例,请登录应用所部署的机器,并输入命令**telnet 100.125.12.108** 41333,其他区域地址请参考探针接入地址。

操作步骤

编辑deployment.yaml。

步骤1 在volumes中增加一个emptyDir。

volumes: - name: paas-apm2 emptyDir: {}

步骤2 在containers.volumeMounts中增加moutPath。

volumeMounts:

- name: paas-apm2

mountPath: /paas-apm2/javaagent/

步骤3 在env中增加JAVA_TOOL_OPTIONS环境变量。

env:

```
    name: JAVA_TOOL_OPTIONS
    value: '-javaagent:/paas-apm2/javaagent/apm-javaagent.jar'
```

步骤4 新增initContainers。

nitCont - nam imag com - /b - '-c - cc app_nat	ainers: e: init-javaagent e: {swrAddress}/op_svc_apm/javaagent:{agentVersion} nand: in/sh ' /paas-apm2/javaagent/apm-javaagent; /bin/sh init-config.sh -master_address {masterAddress} - me {appName} -access_key {accessKey} -access_value {secretKey} -business {business} -env {env} urces:
limi cp m rec cp m volu - na m term term imag	ts: u: 250m emory: 250Mi uests: u: 250m emory: 250Mi meMounts: ume: paas-apm2 ountPath: /var/init/javaagent inationMessagePath: /dev/termination-log inationMessagePolicy: File gePullPolicy: Always itercFruur: default

----结束

参数说明

参数	参数说明
swrAddress	必填,探针镜像地址。组成方式为swr. {regionld}.myhuaweicloud.com,例如:上海一的swrAddress 为swr.cn-east-3.myhuaweicloud.com。
agentVersion	必填,探针版本。参考 JavaAgent<mark>更新说明</mark>,比如:2.4.8- x86_64。
masterAddress	必填,探针上报地址。参考 <mark>探针接入地址</mark> ,比如:上海一 https://100.125.4.27:41333。
appName	必填,组件名称。
accessKey	必填,APM上报AK,从 APM控制台->系统管理->访问密钥获 取 。
secretKey	必填,APM上报SK,从 APM控制台->系统管理->访问密钥获 取 。
business	选填,所属应用名称。
env	选填,环境名称,用于区分所属环境,可用于搜索和配置。

配置示例

spec: replicas: 1

selector: matchLabels: app: vmall-admin version: v1 template: metadata: creationTimestamp: null labels: app: vmall-admin version: v1 spec: volumes: - name: vol-171526714525730640 hostPath: path: /vmall/vmall-deploy/vmall-data/app/vmall-admin/logs type: " - name: paas-apm2 emptyDir: {} initContainers: name: init-javaagent image: swr.cn-north-4.myhuaweicloud.com/op_svc_apm/javaagent:2.4.8-x86_64 command: - /bin/sh - '-c' - cd /paas-apm2/javaagent/apm-javaagent; /bin/sh init-config.sh -master_address https:// 100.125.12.108:41333 -app_name vmall-admin -access_key {AK} -access_value {SK} -business vmall-demo resources: limits: cpu: 250m memory: 250Mi requests: cpu: 250m memory: 250Mi volumeMounts: - name: paas-apm2 mountPath: /var/init/javaagent terminationMessagePath: /dev/termination-log terminationMessagePolicy: File imagePullPolicy: Always monitorGroup: default containers: - name: vmall-admin image: swr.cn-north-4.myhuaweicloud.com/vmall-org/vmall-admin:1.0.0 env: - name: PAAS_APP_NAME value: vmall-admin - name: PAAS_NAMESPACE value: default - name: PAAS_PROJECT_ID value: e49731583bc54978aa7b4da1a2196e46 - name: CLOUDDEPLOY_SYS_INCREMENT value: '1719279081896' - name: JAVA_TOOL_OPTIONS value: '-javaagent:/paas-apm2/javaagent/apm-javaagent/apm-javaagent.jar' resources: limits: cpu: '1' memory: 2Gi requests: cpu: 500m memory: 1Gi volumeMounts: - name: vol-171526714525730640 mountPath: /var/logs - name: paas-apm2 mountPath: /paas-apm2/javaagent/ terminationMessagePath: /dev/termination-log terminationMessagePolicy: File imagePullPolicy: Always

restartPolicy: Always terminationGracePeriodSeconds: 30 dnsPolicy: ClusterFirst securityContext: {} imagePullSecrets: - name: default-secret schedulerName: default-scheduler tolerations: - key: node.kubernetes.io/not-ready operator: Exists effect: NoExecute tolerationSeconds: 300 - key: node.kubernetes.io/unreachable operator: Exists effect: NoExecute

tolerationSeconds: 300

通过 APM 管理应用

应用启动后,等待3分钟应用数据就会呈现在APM界面中,此时登录APM,您可以在 APM进行应用性能优化,详细操作请参考《应用性能管理用户指南(2.0)》。

4.3.2 通过安装 Helm 方式接入 APM

如果使用通用的开源Kubernetes部署应用,可以通过安装Helm方式接入APM,即可在 指标、调用链界面上查看应用情况。

前提条件

1. 确保接入的节点与APM服务网络连通。可使用Telnet命令测试目标机器与APM服务器网络是否连通。

例如,以检查华北-北京四区域,且代码源选择"增强型探针"的连通性为例,请 登录应用所部署的机器,并输入命令telnet 100.125.12.108:41333,其他区域地 址请参考<mark>表4-3</mark>。

表 4-3 增强型探针接入比	地址
----------------	----

区域	接入地址
华北-北京四	https://100.125.12.108:41333
华北-北京一	https://100.125.57.101:41333
华东-上海一	https://100.125.4.27:41333
华东-上海二	https://100.125.140.105:41333
华南-广州	https://100.125.143.102:41333
亚太-新加坡	https://100.125.4.25:41333
华南-广州-友好用户环 境	https://100.125.4.30:41333
华北-乌兰察布一	https://100.125.4.30:41333
华北-乌兰察布汽车一	https://100.125.254.17:41333
西南-贵阳一	https://100.125.0.90:41333

区域	接入地址
中国-香港	https://100.125.6.106:41333

- 2. 请确保您的Kubernetes集群版本为1.19及以上。
- 3. 进程所需最大堆内存250 MB,最大CPU使用0.25 Core。

操作步骤

- 步骤1 安装Helm3。
 - 执行使用脚本安装。
 curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
 - 二进制版本安装。
 - a. 下载 需要的版本。
 - b. 执行如下命令,解压。 tar -zxvf helm-v3.0.0-linux-amd64.tar.gz
 - c. 在解压目录中找到helm程序,移动到需要的目录中。
 mv linux-amd64/helm /usr/local/bin/helm

步骤2 安装探针。

支持无状态和有状态两种类型的应用接入,以下安装步骤以无状态为例。

- 1. 下载apm-operator安装包。 wget 'https://apm2-javaagent-cn-north-4.obs.cn-north-4.myhuaweicloud.com/apm-operatorstable.tar.gz'
- 2. 解压apm-operator安装包。 tar -xvf apm-operator-stable.tar.gz
- 3. 编辑安装包下的operatorK8s/apm-operator/values.yaml,根据实际情况修改参数后进行保存。

accessKeyld: "" accessKeySecret: "" masterAddress: "" swrAddress: "" agentVersion: "" operatorVersion: "" namespace: "apm-operator" targetPort: "9443" selfCheckPort: "9445"

表 4-4 编辑安装包参数说明

参数名称	是否必 填	说明
accessKeyld	是	APM的上报AK,从 APM控制台->系统管理->访 问密钥获取。
accessKeySecret	是	APM的上报SK,从 APM控制台->系统管理->访问 密钥获取。
masterAddress	是	探针上报地址,参考 <mark>表4-3</mark> 。例如:上海一的 masterAddress为https://100.125.4.27:41333。

参数名称	是否必 填	说明
swrAddress	是	探针镜像地址。组成方式为swr. {regionld}.myhuaweicloud.com,比如:上海一 的swrAddress为swr.cn- east-3.myhuaweicloud.com。
agentVersion	是	探针版本。参考 JavaAgent更新说明 ,比如: 2.4.8-x86_64。
operatorVersion	是	apm-operator版本,非必要不修改,使用 stable。
namespace	是	命名空间,默认为apm-operator,非必要不修 改。
targetPort	是	目标端口,默认为9443,非必要不修改。
selfCheckPort	是	检测端口,默认为9445,非必要不修改。

4. 使用脚本安装。

cd operatorK8s chmod 755 apm-operator-install.sh ./apm-operator-install.sh

步骤3 修改应用YAML文件。

- 1. 编辑应用的YAML文件。 kubectl edit deployment {Deployment名称} -o yaml
- 2. 在spec.template.metadata层级下添加以下内容。
 - labels: apmJavaAgentEnable: "on" apmAppName: "<appName>" apmBusiness: "<apmBusiness>" apmEnv: "<apmEnv>" apmJavaAgentVersion: "<agentVersion>"

表 4-5 spec.template.metadata 层级参数说明

参数名称	是否必 填	说明
apmJavaAgentE nable	是	如果需要接入APM探针,需要配置此参数,且设 置为"on"。
apmAppName	是	组件名称,请将 <appname>替换为组件名称。</appname>
apmBusiness	否	所属应用名称,默认为default,请将 <apmbusiness>替换为应用名称。</apmbusiness>
apmEnv	否	环境名称,用于区分所属环境,可用于搜索和配置。默认为default,请将 <apmenv>替换为环境名称。</apmenv>
apmJavaAgentV ersion	否	默认为values下的agentVersion,可指定 agentVersion。

3. 保存配置,应用将自动重启,可以查询YAML确保配置生效。 kubectl get deployment {deployment名称} -o yaml

步骤4 通过APM管理应用。

应用启动后,等待3分钟应用数据就会呈现在APM界面。此时登录APM,您可以在 APM进行应用性能优化,详细操作请参考**《用户指南》**。

----结束

常见问题

- 如果kubectl没有配置成功,helm install时会出现如下报错。
 # helm install prometheus/ --generate-name
 WARNING: This chart is deprecated
 Error: Kubernetes cluster unreachable: Get "http://localhost:8080/version?
 timeout=32s": dial tcp [::1]:8080: connect: connection refused
 解决办法:给节点配置kubeconfig,配置方法请参见使用kubectl连接集群。
- 同一个集群下需要安装不同的Agent版本,需要怎么处理? 解决办法:通过labels中的apmJavaAgentVersion指定版本实现。

4.4 IDE 如何接入 APM

前提条件

已按APM公网接入指导完成公网接入。

操作步骤

- 步骤1 下载APM探针。
 - 1. 登录管理控制台。
 - 2. 单击左侧 ,选择"管理与监管 > 应用性能管理 APM",进入APM服务页面。
 - 3. 在左侧导航栏中选择"应用监控 > 应用列表"。

图 4-3 接入应用

应用性能管理		应用列表 ①		© #6	BRANK D	11回日秋 股入2用
香香中心 9月27日	•	APMEEMopentalemetryRiskywalking, WBBER,BKSS				×
应用监控		●年北-乌兰察布-二零三				
加州州寨		应用:APM (IRSA)		Q INAXMITHE	-	企业版
(#01966)		相称 i 环境 ①	Agentitic	操作	- L	R927112013
Agentitititi		apm-nodėjo-demo delaut	0 1 0 正常 案紙 停止	2526 1628 1809	198815	企业物研究AgentB
新34回22 1000 App(回22 1000	÷	TestApp profiler203test	0 3 0 正常 能给 99注	575 QE 899	B 10 1	Agent*-j@t
配量管理 系统管理	Ţ	TestApp delsuit	0 0 0 正常 奏後 停止	539 QE 899	2(881)Q	2024/09/25
后台管理	•	apm-sw-analysis wulan	2 0 0 正常 陶纸 停止	1519 1628 1899	Biefe	0/3600 Agent*/j-8j
		apm-sw-store waten	2 0 0 亚常 業績 停止	1525 1238 1899	3(839)14	2024/10/31
		apm-archive water	2 0 0 正常 蜀碑 停止	558 (102) 1890	預计 541 天	SHRR

4. 单击"接入应用",进入接入应用页面。

图 4-4 接入应用

基本信息				
* 区域	* 应用	1		
乌兰察布二零三	▼ APM			•
选择接入方式				
* 后端语言				
2 IS Java 2 Node js				
- 代码源				
增强型探针 提供更丰富、稳定的性能。 3 Skywalking 分布式系统的	I <mark>公測</mark> Skywalking ⁾ 立用程序性能	OpenTelemetry <a>Number 2018 开源可观测框架	A CONTRACTOR OF CONTRACTOR OFO	
数据接入				
				4
curl -k https://javaagent.obs.cn-north-7.ulanqab.huawei.cc masteraddress https://100.79.29.96:443 -obsaddress http	m/apm_agent_install2.sh -o apm_; s://javaagent.obs.cn-north-7.ulanqa	agent_install.sh && bash apm_age b.huawei.com -version latest; hist	ent_install.sh -ak (API ory -cw; history -r	/I_AK} -sk {APM_SK} - □

- 在左侧导航栏选择"系统管理 > 访问密钥",进入访问密钥页面,获取 JavaAgent的APM_AK和APM_SK。详细操作参见访问密钥。
- 6. 用从访问密钥页面中获取的AK/SK替换复制的JavaAgent命令的APM_AK和 APM_SK。
- 步骤2 执行git bash命令。在本地电脑D盘agent目录下,将复制的安装JavaAgent命令执行。
- 步骤3 修改apm.config文件中的master.address、access.address以及business参数值。

图 4-5 修改配置文件



步骤4 单击下拉菜单,选择"编辑配置"。

1. 修改IDE的运行/调试配置。

륞 ApmMasterApp 🔻 <u>1</u>-G C 编辑配置.. tBeatTask.java skService 运行配置 dpmAccessApp dpmAlarmApp dpmApiApp dpmArchiveApp d ApmDeliverApp ApmFrontApp nterApp 🔬 🔬 ApmMergeApp dpmPuTaskApp ApmStoreApp ApmSwAccessApp ass); ApmSwAnalysisApp dpmSwApiApp dpmSwStoreApp dpmTaskApp dpmTransferApp ApmMasterApp chSize, String region, i

图 4-6 修改运行/调试配置

2. 在运行/调试配置页面,左侧导航栏中,选择"ApmMasterApp"。右侧"构建并运行"中添加"-javaagent:D:\agent\apm-javaagent\apm-javaagent\apm-javaagent.jar=appName=apm-master"。

图 4-7 修改构建并运行

■ 运行/调试配置		×
+ − T to the set of t	名称(N): ApmMasterApp 运行于: ★本地が騒 ▼ 管理目标 运行配置可以在本地或目标上执行: 例如,在 Docker 智器中或使用 SSH 在运程主机上执行:	存储为项目文件(S) 🌲
<pre>@ ApmArchiveApp @ ApmDeliverApp @ ApmFrontApp @ ApmMasterApp</pre>	构建井延行 java 8 'apn-master-app' 根共的 SDK ▼ ↓ -cp apn-master-app	修改选项(M) ~ Alt+M
≪ ApmMergeApp ≪ ApmPuTaskApp	-javaagent:D:\agent\apm-javaagent\apm-javaagent.jar=appName=apm-master	Ξ μ ^η
《ApmSwAccessApp 《ApmSwAnakysSapp 《ApmSwStoreApp 《ApmTaskApp 《ApmTaskApp 《ApmTaskApp	com.huawei.hwclouds.lubanops.apm.master.ApmMasterApp VM 远现、'Java' 命令的 CLI 实务、示例: -ea -Xmx2048m. Alt+V 有效配置文件: 逗号分隔的配置文件列表 屈动時打开运行/喝試工員會口 × 將带有 *provided* 范围的依赖坝添加到她路经 ×	
· · · · · · · · · · · · · · · · · · ·	検定	取消 应用(A)

3. 单击"确定"。

步骤5 重启服务。如果在应用下拉菜单中找到创建的"APM",则接入成功。

图 4-8	接入成功
-------	------

Relation	🌐 APM / 😚 apm-master / 🖾 defau	11(夏多节点~	12 用户指用	
磁用版材 ③ ■ 空用 APM ・ ② 区域 乌兰原ホニ専三 ・	招扑 Profiler性能分析 2010 基礎	<u>編成 </u> 戸田 2024年0718 19:53 GMT-88 の — 202440718 27:13 GMT-88 の 〇 近2015日	• C	
♦ 标签: 全部 *	jvm能思			
请输入出件描环境名称 Q	javaagent)後本	2.4.5		
🖃 💷 APM	启动时间	2024-03-18 20:12:58 GMT+06:00		
😑 🤴 apm-master	启动参数	[-javaagent:D:/agent/apm-javaagent/apm-javaagent jar=appName=apm-master, -XX:TieredStopAtLevel=1, -Xverity:none, -Dspring output ansi.enab		
🖾 default 🖾 🕲 🗓	El default 単 ② む java映物路径 C:Program FiesJavaljdkt 8.0_251/bin C:WINDOWS/SunJavabin,C:WINDOWS/system52,C:WINDOWS/C:WINDOWS/System52,C:WINDOWS/C:WINDOWS/System52,C:WINDOWS/C:WINDOWS/System52,C:WINDOWS/C:WINDOWS/System52,C:WINDOWS/C:WINDOWS/System52,C:WINDOWS/C:WINDOWS/System52,C:WINDOWS/C:WINDOWS/C:WINDOWS/System52,C:WINDOWS/C/WINDOWS/C:WINDOWS/C/WINDOWS/C:WINDOWS/C/WINDOWS/C/WINDOWS/C:WINDOWS/C/WINDOWS/C/WINDOWS/C/WINDOWS/C:WINDOWS/C/WIND			
	java版本	1.8.0_251		
	java规范版本	1.2		
	操作系统	Windows 10		
	操作系统版本	10.0		
	arch	amd64		
	处理器个数	12		
	SDK版本			

----结束

4.5 Windows IIS 如何安装 JavaAgent

前提条件

• 部署APM Agent时,必须确保接入APM的机器与APM服务网络连通,Agent才能正常工作。

可使用Telnet命令测试目标机器与APM服务器网络是否连通。

例如,以检查华北-北京四区域,且接入方式选择"增强型探针"的连通性为例, 请登录应用所部署的机器,并输入命令telnet 100.125.12.108:41333,其他区域 地址请参考探针接入地址。如果接入APM的机器与APM服务网络无法连通,可以 接入代理。

 选择"系统管理 > 访问密钥"进入访问密钥页面,查看获取接入javaagent所需的 AK/SK。

图 4-9 获取 AK/SK

应用性能管理	il.	间密钥 ①					0 用户推荐
1919中心 ~							
应用监控 ~		11年3月回初世書,会市未設	實世靈风险、为了所导安全性、建议您是需要换并安排	善保存访问密钥。			×
1921-839 <mark>8449</mark> ~							
RORAZED Beta		① 使用这问密码上传数编时建设	对话间密闭进行如密处理。具体步骤 非专制处文性。				×
App显短 Beta ~			WART VALUE AND BUILDING				(ast) initialization
配置管理 ン							
系统管理 へ		访问把明ID(AK)	运用性能管查访问整明(SK)	描述	8969369	秋恣	授作
话问题明		go NP	💿 0****1	a	2023-08-24 14:15:06	0 屈用	编辑 停用 動脉
进行政策		Jy51 , X	⊙ 2****0	R	2022-05-12 16:29:54	◎ 屈問	编辑 经相关
探针统计							

操作步骤

步骤1 登录管理控制台。

- 步骤2 单击左侧 💳 ,选择"管理与监管 > 应用性能管理 APM",进入APM服务页面。
- 步骤3 在左侧导航栏中选择"应用监控 > 应用列表"。

步骤4 单击"接入应用",进入接入应用页面。

图 4-10 接入应用

应用性能管理		应用列表 ⑦									ロ用作指導 ご 返回日板	被入应用
資留中心 省留列表	^	APW已无地opentalemetry形	lskywalking, ittillERAB	10.32								×
应用监控	^	●印北-乌兰奈布·汽车—	• ###-###	● #41-31用目	● 华北-乌兰原布—	● 彩北-乌兰寮布二零—	●出北-乌兰奈布二零二	●徐东-上海一	♥除东・上海二	◆印度・广州	● 徐庵-广州-友好用户环境	۰
应用列表												
1215		应用 aprotest-AOM (銀い)	~					谢哈入兴地学的	R	00	企业版	_
调用线		創件 环境 ③			AgentRED		1915				- Subtrinants	
探针管理		lettuce			1 0 0		15H 183	-				11/1
(E)张温珍 Deta	~	1991			止果 陶镜 停止							
App <u>G210</u> Beta	× v	mysic) test			1 0 0 正常 業後 停止		1514 Q.M	29				
配置管理 系统管理	š	< profiler_247 test			1 0 0 正常 南班 停止		1531 B3	894				
		profiler_245 test			1 0 0 正常 南线 停止		500 Q.	259				

步骤5 复制JavaAgent OBS下载地址。



步骤6 下载JavaAgent到需要接入APM机器的任意目录。下载方法如下:

- 参考探针下载地址下载apm-javaagent。
- 下载在浏览器中执行命令: {obsaddress}/apm2_javaagent/apm-javaagentx.x.x.tar

~	0	新标签	页			×	+											
÷	O thtps://apm2-javaagent-cn-north-5.obs.cn-north-5.myhuaweicloud.com/apm2_javaagent/apm-javaagent-2.4.9.tar																	
	伏鏡相	眹	口前	台地址	🗅 dt	C	可信构建	🗅 otel	🗅 apm	口 华为云	し 続期会	合書云	L HCS环境	口作	□ 个人培养	D IA	口项目	C \$

在CMD中执行如下命令。
 curl -kv {obsaddress}/apm2_javaagent/apm-javaagent-x.x.x.tar -output apm-javaagent.tar

步骤7 解压apm-javaagent.tar。

enar	12.80.00.00	~=	~~
boot	2023/9/22 15:11	文件夹	
core	2023/9/22 15:11	文件夹	
ext	2023/9/22 15:11	文件夹	
lib	2023/9/22 15:11	文件夹	
plugine	2023/0/22 15:11	☆ <i>0</i> +±±	
apm.config	2023/9/25 15:17	CONFIG 文件	1 KB
🕹 apm-javaagent.jar	2023/9/20 20:29	Executable Jar File	17 KB
init-config.sh	2023/9/20 20:29	Shell Script	2 KB

步骤8 修改javaagent中的apm.config配置文件。master.address配置请参见探针接入地址, 将AK/SK写入配置文件中,如下图所示。

图 4-12 写入 AK/SK

	-	×
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)		
master.address=https://1 1333		^
access.key=ag7183		
secret.key=vR		 Shu
log.level=debug		
event.thread.count=3		
apm.proxy=http://iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii		
#access.address=		
app.name=helloworld		
#instance.name=		
#env={{env}}		
#env.tag=		
#decrypt.className=com demo.DecryptDemo		
#decrypt.methodName=decrypt		

步骤9 添加启动命令并重启应用。

图 4-13 添加启动命令并重启应用

```
2. 茶加用却命令并重用症用
Agent安装大约需要1分钟、安装完毕后,在应用服务器的启动脚本中添加以下JVM参数,然后重新启动您的应用。
java-javaagent-K带针安装路径>/apm-javaagent.jar=appName=<递仟名称>,env=<环境名称>,envTag=<环境标签>,business=<应用名称>,subBusiness=< ロ

子应用名称>-jar

用の应用>,jar

店誌命令示例: java-javaagent/icot/my-dir/apm-javaagent/jar=appName=my-service_env=der_-jar test-2.0.0-SNAPSHOT.jar ロ
```



5 调用链与日志关联

适用范围

常用的一些日志框架logback, log4j等。

举例

<property name="LOG_PATTERN" value="%d{yyyy-MM-dd HH:mm:ss.SSS}} | gtraceid: %X{apm-gtraceid} | traceid: %X{apm-traceid} | spanId: %X{apmspanid}">

</property>

APM 服务中调用链相关的参数说明

- 1. apm-traceid: apm服务采集到调用链的唯一标识。
 - 图 5-1 采集调用链的唯一标识

查询条件	清空	◎ 1 新记录	
应用	APM (BGJ.)	▲ 请求成功 ▲ 请求代表 病应时间 # 产生时间 #	PM实际采样到的调用链的唯一标识:trace-id
⊠m(• •	EOST ///lacm2/coenari//vew/metric/med/ E 2020-10-00 10 10:300	0 200 47 ms Trace ID: 1033977-1009532129695-200769
	更多 •	1 乐记录	
URL	Rest URL +		
163 9			
	更多 ▼		
响应时间	ms · ms		
是否异常	\$5 v		
Trace ID	1833977-1698632129865-29076	3	
	④ 其他查询条件		
遭找Trace			

- 2. apm-gtraceid: apm服务中未被采样到的调用关系的唯一标识 。
 - apm服务的调用链具有一定采样率,所以用apm-gtrace-id来表示未被采样的调用 链的唯一标识。
- 3. apm-spanid:在某个调用链的微服务之间调用,表示某一个微服务的id,示例如下。

图 5-2 调用链的微服务之间调用



6 结果查询页面内嵌到客户自建系统

背景信息

APM支持将界面嵌入到客户自建系统。通过IAM服务的联邦代理机制实现用户自定义 身份代理,再将登录链接嵌入至客户自建系统实现无需在华为云官网登录就可在自建 系统界面查看。

前提条件

用户自定义创建身份代理并创建登录地址FederationProxyUrl,详细可参考IAM服务的 文档"创建自定义身份代理"。

操作步骤

用户创建自定义身份代理成功后,您需要执行如下步骤,实现页面的内嵌。

步骤1 将FederationProxyUrl中的console_service_url修改为云服务console地址。

console_service_url示例:

对应业务模 块	示例url
应用监控- 指标	https://console.huaweicloud.com/apm2/? region={regionId}&cfModuleHide=header_sidebar_floatlayer#/ console/appindex/business/detail?leftMenuCollapsed=true
应用监控- 调用链	https://console.huaweicloud.com/apm2/? region={regionId}&cfModuleHide=header_sidebar_floatlayer#/ console/appchain?leftMenuCollapsed=true
链路追踪- 指标	https://console.huaweicloud.com/apm2/? region={regionId}&cfModuleHide=header_sidebar_floatlayer#/ console/trace/metric/environment/view?leftMenuCollapsed=true
链路追踪- 调用链	https://console.huaweicloud.com/apm2/? region={regionId}&cfModuleHide=header_sidebar_floatlayer#/ console/trace/chain?leftMenuCollapsed=true

对应业务模 块	示例url
前端监控- 前端站点	https://console.huaweicloud.com/apm2/? region={regionId}&cfModuleHide=header_sidebar_floatlayer#/ console/rum/app?leftMenuCollapsed=true
App监控- App列表	https://console.huaweicloud.com/apm2/? region={regionId}&cfModuleHide=header_sidebar_floatlayer#/ console/mobile/list?leftMenuCollapsed=true

参数名称	说明
regionId	表示当前您所在的区域,可在华为云官网登录云服务后在浏览器的地 址栏中获取。例如cn-north-4。
cfModuleHi de	值header_sidebar_floatlayer,表示隐藏华为云console页面页头页脚 和菜单栏。
leftMenuCo llapsed	true表示隐藏左侧菜单,false表示不隐藏左侧菜单。

步骤2 使用iframe将apm页面嵌入用户系统,示例代码如下:

<iframe src="\${FederationProxyUrl}" ref="Frame" scrolling="auto" width="100%" height="100%"</iframe>

----结束

了 如何使用 APM Profiler 定位性能问题

APM Profiler 是一种持续性能剖析工具,可以帮助开发者准确找到应用程序中消耗资源最多的代码位置。

前提条件

- 1. APM Agent 已接入,操作方法参见开始监控JAVA应用。
- 2. Profiler功能已开启,操作方法参见Profiler性能分析。
- 3. 登录应用性能管理控制台。

如何查询并解决 CPU 升高问题

- 步骤1 在左侧导航栏选择"应用监控 > 指标"。
- 步骤2 在界面左侧树单击待查看基础监控环境后的 🖾 。
- 步骤3 单击"基础监控",切换至基础监控页签,监控项选择"JVM监控"。

图 7-1 查看 JVM 监控

le Pente	🌐 APM / 🔄 App / 🎧 apm-access / 🔃 wulan / 更多节点 *		19 用户用
27用編年 回 血用 APM (別は) ・ ・ ・ ・ ・	Kith Profilent生産に分析 (2013) 接口同用 Matticke Frie E = apm2-apm-access・ C = JVM品社 ・	※部項用 通信的议 取留加 Web皆編 消息別,列 活起:1控用用 2024/04/17 11.66 0417-06	00-202404/18 11:06 04/7-08:00 〇 近1天 🔹 (
% 5≝ ±#	48 ¹²	≡ 编程状态	=
	 当前該理政 ◆ 死時提理政 ◆ 守护規理政 ◆ 保有目記該理政 10,000 	< 1/2 > WAITING\$5:558722 TERMINATEDE5 250	ESERER - RUNNABLES < 1/3 >
- 💷 APM	8,000	200	
- E App	6,000	150	
🖂 wulan 🔛 🕲 🗓	4,000	100	
	2,000	50	
	04-17 04-17 04-17 04-17 04-17 04-17 04-17 04-18 04-18	04-18 04-18 04-18 04-18 04-18 04-18	04-17 04-18 04-18 04-18 04-18 04-18 04-18 04-18

步骤4 找到 "cpu(%)",发现CPU持续高达80%以上。

图 7-2 cpu(%)



步骤5 单击"Profiler性能分析",切换至Profiler性能分析页签。

步骤6 单击"性能分析",Profiler性能分析页面,类型选择"CPU Time"。

图 7-3 Profiler 火焰图



步骤7 分析火焰图数据 从火焰图中可以看到, java.util.LinedList.node(int) 方法占用了 66% 的 CPU, 而相应的业务代码方法是 countPages(List)。

图 7-4 Profiler 火焰图分析

	com.hu	awei.performance.demo.negative.Controller.countPa	
	ges(Lis	t) (66.92%, 42m37s950ms)	piit(
	Package	com.huawei.performance.demo.negative	NOL
om huawei performance demo negative Controller countPages(List) (66.92%, 42m37s950ms)	Class	Controller countPages(List)	a. 18
ava.util.LinkedList.get(int) (66.89%, 42m36s900ms)			NOR
ava.util.LinkedList.node(int) (66.88%, 42m36s550ms)	CPU Time		
	Self	950ms(0.02%)	lina
	Total	42m37s950ms(66.92%)	10

步骤8 分析业务代码,结合代码可以发现该方法countPages(List)是对入参集合list进行下标 遍历,而通过火焰图运行时数据发现,传入的是 LinkedList,而LinkedList底层数据结 构是链表,通过下标遍历效率会非常差。

图 7-5 代码分析

			Returns the promition wode at the specified element index.
		e e	Node <e> node(int index) {</e>
	private long countPages(List <book> list) {</book>		<pre>// assert isElementIndex(index):</pre>
¢			
<u>é</u>			Node <e> x = first;</e>
			<pre>for (int i = 0; i < index; i++)</pre>
¢.			x = x.next:
<u> </u>			
			Node <e> x = last;</e>
ê 1			for (int i = size - 1; i > index; i)
}			
2			
	and the second sec		
	业务代码		} JDK代码

步骤9 修复代码,将list的遍历算法从普通的下标for循环改为增强的for循环。

图 7-6 修复代码

<pre>private long countPages(List<book> list) {</book></pre>		private long countPages(List <book> list) {</book>
		<pre>count += book.getPageCount();</pre>

步骤10 优化后,重复步骤4-步骤5,发现CPU使用率<1%。



图 7-7 优化后 CPU(%)

----结束

如何查询并解决内存升高问题

前提条件:开启测试程序,同时设定heap的大小为2g(-Xms2g -Xmx2g)。

- 步骤1 在左侧导航栏选择"应用监控 > 指标"。
- 步骤2 在界面左侧树单击待查看基础监控环境后的 🖾 。
- **步骤3** 单击"基础监控",切换至基础监控页签,监控项选择"GC监控",非常频繁的进行gc操作。

文档版本 01 (2025-02-20)

图 7-8 查看 GC 监控



步骤4 监控项选择"JVM监控",查看JVM监控。

图 7-9 查看 JVM 监控



步骤5 单击"Profiler性能分析",切换至Profiler性能分析页签。

步骤6 单击"性能分析",Profiler性能分析页面,实例选择"Allocated Memory"。根据右侧Self排序排查,找到分配内存最多的方法。

图 7-10 内存火焰图



步骤7 查看代码,发现LargeEnum是个枚举类,定义了大量的常量。由于枚举类的方法 values() 底层是通过数组clone实现的,即每次调用values()方法,底层会复制一个枚 举数组,所以会导致频繁分配堆内存,频繁GC。

图 7-11 查看代码



步骤8 问题修复,将values定义为一个常量,避免频繁调用enum.values()。

1	↓ ∠ Side-by-side viewer • Ignore imports and formating • Highlight lines • ÷ ± □ ∅, ◎ ?			2 di Britanilia na barana dan barana kata dan sa kata di ana sa kata da ang kata da ang kata da ang kata da sa kata
*	REGION_998("REGION_998", new byte[1024 * 1024],"北京region: 998	1002	1882	REGION_998("REGION_998", new byte[1024 * 1024],"北京region: 9
	REGION_999("REGION_999", new byte[1024 * 1024],"北京region: 999			REGION_999("REGION_999", new byte[1024 * 1024],"北京region: 9
	public final String regionName;			public final String regionName;
	public final String description;			public final String description;
	LargeEnum(String regionName, byte[] bytes, String description)			LargeEnum(String regionName, byte[] bytes, String description
	<pre>public static LargeEnum fromValue(String name) {</pre>		1013	
				<pre>public static LargeEnum fromValue(String name) {</pre>
	<pre>if (item.regionName.equals(name)) {</pre>			
				<pre>if (item.regionName.equals(name)) {</pre>
}				
1				

图 7-12 问题修复

步骤9 重复<mark>步骤3-步骤6</mark>,发现GC次数大幅下降,并且火焰图中以找不到enum.values()内存分配。



图 7-13 优化后 GC 监控

图 7-14 优化后性能分析火焰图



----结束

如何查询并解决接口响应慢问题

- 步骤1 在左侧导航栏选择"应用监控 > 指标"。
- 步骤2 在界面左侧树单击待查看基础监控环境后的 🖾 。
- **步骤3**单击"接口调用",切换至接口调用页签。通过APM的接口调用功能发现接口响应 慢,平均响应时间80s左右。

图 7-15 接口调用

🧾 CodeBot / 🚞 RefactorBot / 🚞 CloudProfiler / 🌍	CP-Processor / 应	alpha-szv-kunpeng-c	om / 更多节点 >					☞ 用户指南
拓扑 Profiler性能分析 🔼 接口调用 基础监控	异常 数据库	缓存 Web容器						
□ • mysql-negative-cas ▼ □ • URL监控	Ŧ			2024/04/07 19:20 GN	1T+08:00 — 2024/04	/07 19:40 GMT+08:00	③ 近20分钟	• C
接口维度汇总 状态码维度汇总 集群维度汇总	总览							
接口继度汇总								
Q 输入默认前两列内容搜索				1				⊚ ±
url	method	调用次数	平均响应时间(ms)	错误数	最大并发	最慢调用(ms)	apdex	
/user/{{numeric}}	GET	1386	80248.41	0	200	96754	0.00	
				-		10 🔻 劉凤, :	共1条 〈	1 >

- 步骤4 单击"Profiler性能分析",切换至Profiler性能分析页签。
- **步骤5** 单击"性能分析",Profiler性能分析页面,实例选择"Latency",输入接口所在的方法。

图 7-16 性能分析

Com.ndawer.pronier.mysql.demo.comroner.webController.registeroser			0				
Total (100%, 16h19m59s414ms)	方法路径	Self 47	Total J≣				
org.apache.tomcat.util.threads.TaskThread\$WrappingRunnable.run() (99.07%, 16h10m51s952ms)	pthread cond wait()	13h42	13h42				
org.apache.tomcat.util.threads.ThreadPoolExecutor\$Worker.run() (99.07%, 16h10m51s952ms)	pthread cond timedwait()	2h32m	2h32m				
org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor\$Worker) (99.07%, 16h10m51s952ms)	iava lang Throwable fill in sta	11s281	1m42s				
org apache tomcat util net SocketProcessorBase run() (99.07%, 16h10m51s952ms)	java Jang StrictMath Iog1n/double)	110146	140201				
org.apache.tomcat.util.net.NioEndpoint\$SocketProcessor.doRun() (99.07%, 16h10m51s952ms)	Control and Control and Control of Control o	40+754	193231				
org.apache.coyote.AbstractProtocol\$ConnectionHandler.process(SocketWrapperBase,SocketEvent) (99.07%, 16h10m51s952ms)	Codeneapiiid_statt()	105754	105754				
org.apache.coyote.AbstractProcessorLight.process(SocketWrapperBase,SocketEvent) (99.07%, 16h10m51s952ms)	leee/54_log()	9s453ms	125158				
org.apache.coyote.http11.Http11Processor.service(SocketWrapperBase) (99.07%, 16h10m51s952ms)	ieee754_rem_pio2()	9s128ms	9s483ms				
org apache catalina.connector.CoyoteAdapter.service(Request,Response) (99.07%, 16h10m51s952ms)	AllocTracer::trapHandler()	8s735ms	15s162				
org.apache.catalina.core.StandardEngineValve.invoke(Request,Response) (99.07%, 16h10m51s952ms)	BacktraceBuilder::push()	7s500ms	11s256				
org.apache.catalina.valves.ErrorReportValve.invoke(Request,Response) (99.07%, 16h10m51s952ms)	java.lang.String.contains(CharS	6s108ms	6s108ms				
org.apache.catalina.core.StandardHostValve.invoke(Request,Response) (99.07%, 16h10m51s952ms)	pthread_cond_signal()	5s681ms	5s681ms				
org apache catalina authenticator AuthenticatorBase invoke(Request,Response) (99.07%, 16h10m51s952ms)	SymbolTable::lookup()	5s110ms	6s270ms				
org apache catalina.core.StandardContextValve.invoke(Request,Response) (99.07%, 16h10m51s952ms)	frame::sender()	4s650ms	20s846				
org apache catalina.core.StandardWrapperValve.invoke(Request,Response) (99.07%, 16h10m51s952ms)	set_high()	4s354ms	4s354ms				

步骤6 排查调用栈,寻找耗时的方法。如下图,NegativeWorkService#handle中 executeUpdate()方法耗时最多。

图 7-17 排查调用栈



步骤7 排查NegativeWorkService#handle方法,发现根因是循环内执行数据库插入操作。

图 7-18 排查 NegativeWorkService#handle 方法





图 7-19 问题修复



步骤9 观察接口调用平均响应时间,从80s减少到0.2s。

图 7-20 优化后查询接口调用平均响应时间

拓扑 Profiler性能分析 🔠 接口调用 基础监控	异常 数据库 !	缓存 Web容器						
🖸 • mysql-positive-case 👻 🗐 • URL监控	*			2024/04/07 20:11 GMT+0	08:00 — 2024/04/07	20:31 GMT+08:00	⑤ 近20分钟	• (
接口维度汇总 状态码维度汇总 集群维度汇总	总览							
								@ _
3817/3A/4/89/PG79F3T8138.64								0 0
url	method	週田次数	亚物响应时间(ms)	10010236	是大井发	最慢適田(ms)	andex	
uri /user/{{numeric}}	GET	调用次数 1280302	平均响应时间(ms) 187.30	错误数 0	最大并发 200	最 惯调用(ms) 2213	apdex 0.81	
uri /uset/((numeric))	GET	调用次数 1280302	平均响应时间(ms) 187.30	(研究数 0	最大井发 200	最間调用(ms) 2213 10 ▼ 急(页,	apdex 0.81 共1条 <	1 >

----结束

8 如何使用 Profiler 定位 OOM 问题

背景

服务所在容器频繁重启,通过自监控发现重启前fullgc次数会突增(约每分钟20次)。

使用 Profiler 定位 OOM 问题

- 步骤1 登录管理控制台。
- 步骤2 单击左侧 ,选择"管理与监管 > 应用性能管理 APM",进入APM服务页面。
- 步骤3 在左侧导航栏选择"应用监控 > 指标"。
- 步骤4 在界面左侧树单击待查看Profiler性能分析环境后的 2 。
- 步骤5 单击"Profiler性能分析",切换至Profiler性能分析页签。
- 步骤6 单击"性能分析",进入性能分析页面。
- **步骤7** 选择类型: Memory,范围: All Code,聚合维度: Method,发现有两个方法占用了 较多内存。



步骤8单击"方法路径"列,对应的方法名找到该方法的调用栈,向上找到调用此方法的业务代码。



步骤9 定位到业务代码块,该代码使用了一个缓存来存放每个instance的信息。通过自监控查 看该sql的调用,发现每分钟会调10万次,进一步证实是缓存失效了。

当查询instance信息时,会先从缓存查,如果查不到再从mysql查。instance信息查到 之后存入缓存,防止频繁访问数据库。

步骤10 检查代码发现缓存的key是一个类,该类没有重写equals和hashCode方法。因此,导 致缓存通过key去获取value时,会根据key的地址来判断该key是否在缓存中存在。而 每次传参进来的key地址都不一样,所以从缓存中查找失败,只能从mysql查找,然后 又不断往缓存中存,最终导致OOM问题。



----结束

解决方案

给作为key的类重写equals和hashCode方法。该类有个uuid属性,不同instance的uuid不一样,因此可以用uuid来判断两个instance是否相同。

CCE 容器部署 APM 探针支持对 APM 的 AK/SK 加密

컙뫂

在CCE容器中部署APM探针时,出于安全考虑,可以对APM的AK/SK进行加密。

操作步骤

- 步骤1 生成一个含有解密方法的jar包,假设jar包名为demo.jar,内置解密类为 com.demo.DecryptDemo,解密方法为decrypt(注意decrypt的是静态方法)。然后 将该jar打包一个镜像,上传到镜像仓,获取密钥方法参见<mark>访问密钥</mark>。
- 步骤2 在CCE deployment yaml中添加一个initContainer属性,如下所示。

示例:

- 步骤1中上传的镜像地址为swr.cn-north-5.myhuaweicloud.com/ 1. hwstaff_pub_apmpaasw3/decrypt:v2
- 解密的类名为com.demo.DecryptDemo,解密方法为decrypt。 2.

按如下方式添加一个initContainer,注意替换加粗部分。

initContainers: - name: init-secret image: swr.cn-north-5.myhuaweicloud.com/hwstaff_pub_apmpaasw3/decrypt:v2 command: - /bin/sh - '-C' - cp /root/com.demo.DecryptDemo.jar /var/init/secret/apm-javaagent/ext; sed -i 's %#decrypt.className=.*%decrypt.className=com.demo.DecryptDemo%g' /var/init/secret/apmjavaagent/apm.config; sed -i 's%#decrypt.methodName=.*%decrypt.methodName=decrypt%g' /var/ init/secret/apm-javaagent/apm.config; resources: limits: cpu: 100m memory: 100Mi requests: cpu: 100m memory: 100Mi volumeMounts: - name: paas-apm2 mountPath: /var/init/secret

添加该initContainer可以实现将jar包复制到apm-javaagent/ext目录下,以及修改 配置文件的目的。

步骤3 在apm页面获取AK和SK,然后对sk进行加密处理,将AK和加密后的SK替换yaml文件中的如下值。

性能管理配置						
安装探针	不启用 APM2.0 探针 APM1.0 探针					
探针版本						
	探针版本 latest 并不映射最新版本,探针版本能力请参见版本说明。 版本说明					
探针升级策略	重启自动升级 ~					
APM 环境	请榆入环境名称 (选填)					
APM 应用	vpceptest v Q 您可以创建APM 应用 C , 完成后点击刷新按钮					
子应用	请输入子应用(选填)					
接入密钥	AK:					
	^{sk:} [····································					
	已自动获取密钥信息。可前往 APM 服务 🖸 查看接入密钥					

步骤4 保存配置对CCE实例进行升级即可。

----结束