SOFAStack

服务网格
运维指南

产品版本：AntStack Plus 1.13.1

文档版本：20230707

蚂蚁集团
ANT GROUP

法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

商标声明

免责声明

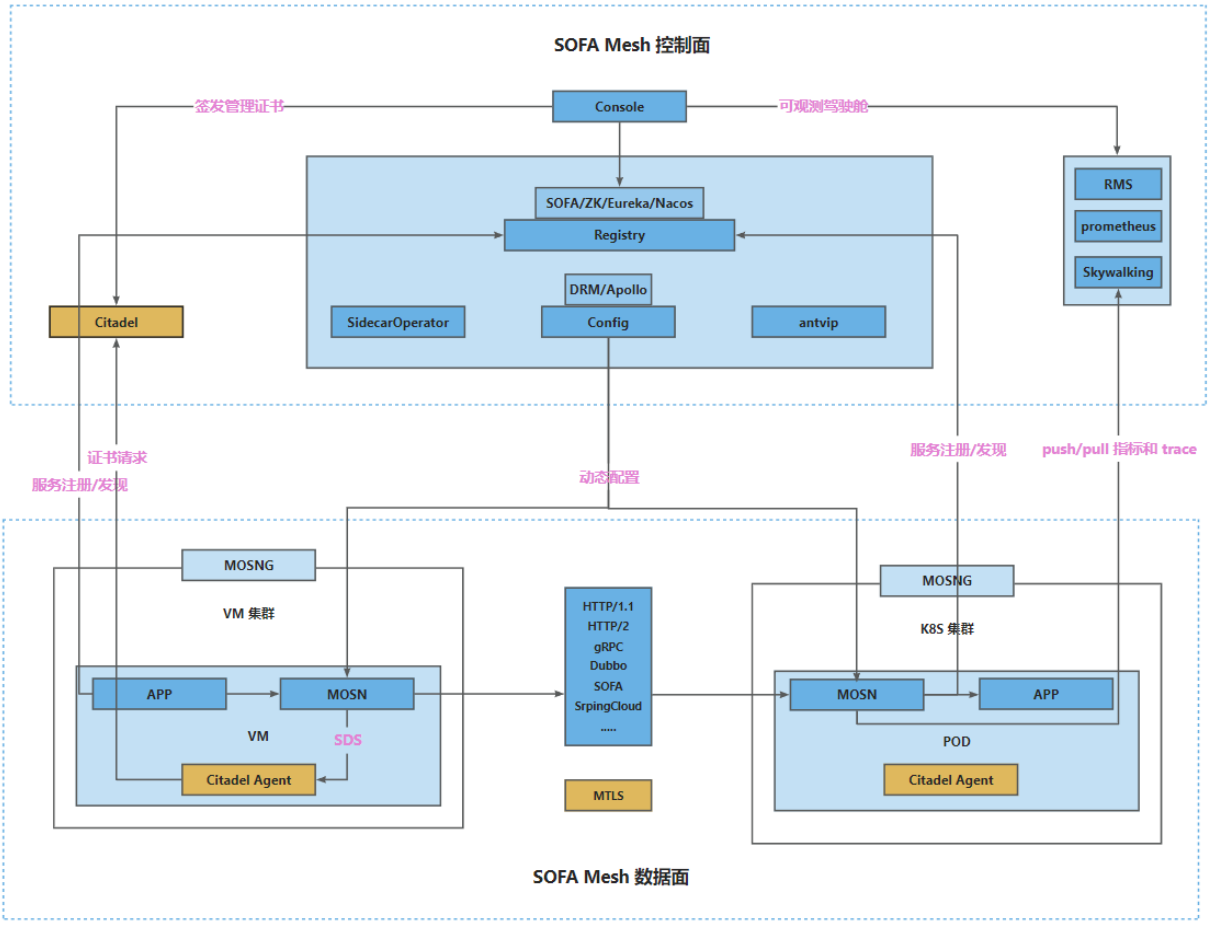
通用约定

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.系统组件

1.1. 组件角色

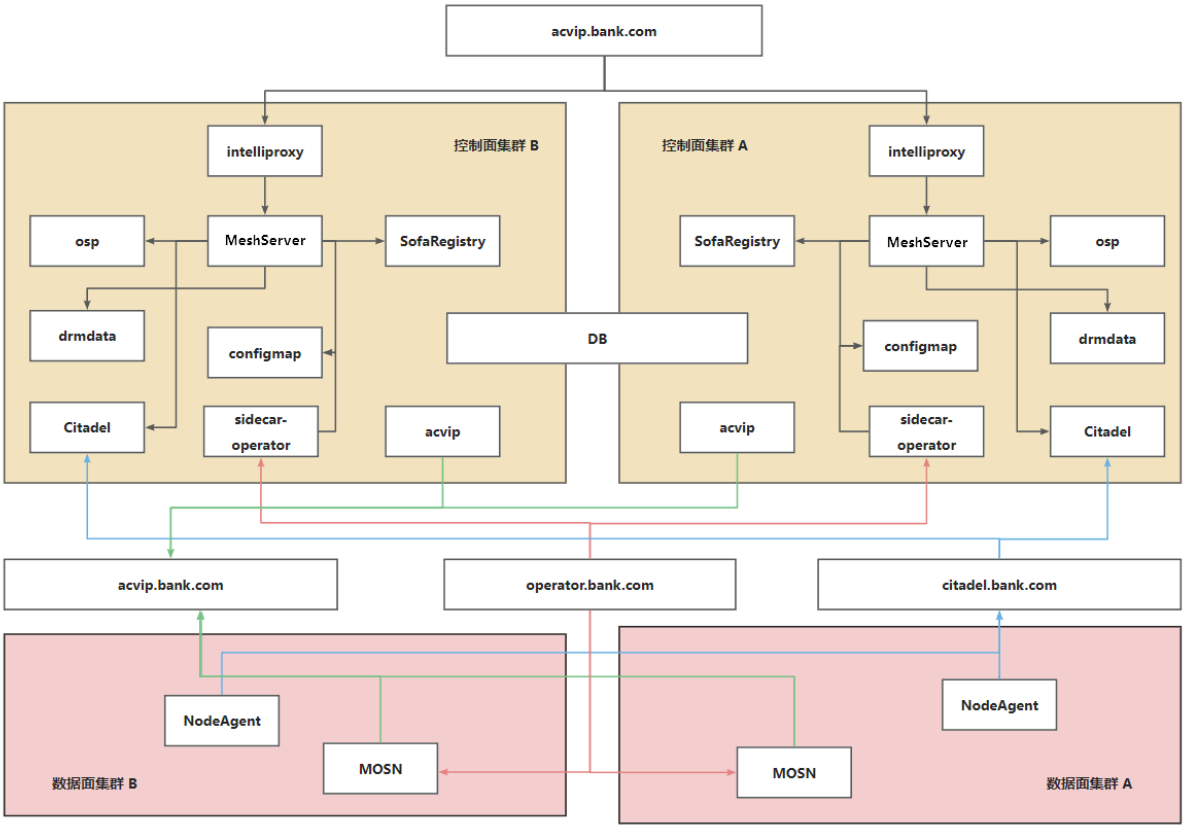


服务网格产品包括以下组件：

- **CloudMeshConsole**：服务网格控制台，查看应用服务状态、Pod 实例状态、Sidecar 状态、Sidecar 监控、管理身份认证密钥。
- **OpenAPI**：提供 K8s 集群操作接口，用于管理各业务 cr 资源。
- **Citadel**：监听 APIServer 的 cr 资源，下发 cr 资源、证书给 nodeagent。相关 Sidecar 包括 MIST 和 ODP。
- **Citadel-Agent**：监听 cr 资源、证书状态，下发给对应 node 的 Pod。相关 Sidecar 包括 MOSN。
- **Inspector**：从 APIServer 中查询 Pod 列表，根据 Pod 的连接信息远程获取 Pod 的状态。
- **SidecarOperator**：负责管理 sidecar 自动注入流程处理。
- **Config**：动态配置下发，管理服务治理配置或者安全配置。
- **RMS**：可观测性平台
- **skywalking**：trace 收集组件。
- **prometheus**：metrics 平台。

1.2. 部署拓扑

部署架构



资源清单说明

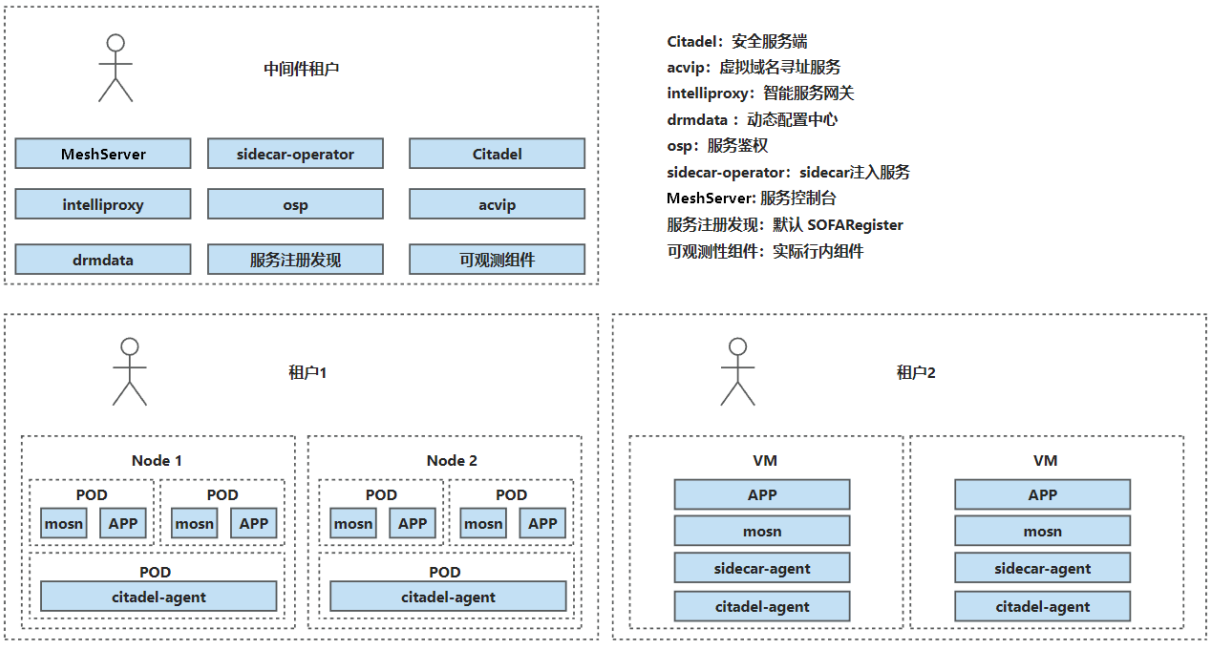
应用	规格	数量	总 CPU 核数	总 MEM（GB）	系统磁盘（GB）	说明
Citadel	CPU：2 C 内存：4 GB 磁盘：50 GB	4	8	16	200	单实例 500 连接数（即 1 台物理机），共 4 台物理机。
Meshnode-agent	CPU：1 C 内存：1 GB 磁盘：50 GB	1	1*N	1*N	50*N	需根据实际物理机数量调整。
intelliproxy	CPU：2 C 内存：4 GB 磁盘：50 GB	4	8	16	200	-

osp	CPU：2 C 内存：4 GB 磁盘：50 GB	4	8	16	200	-
acvip	CPU：2 C 内存：4 GB 磁盘：50 GB	4	8	16	200	-
MeshServer	CPU：2 C 内存：4 GB 磁盘：50 GB	4	8	16	200	-
drmdata	CPU：2 C 内存：4 GB 磁盘：50 GB	4	8	16	200	-
sidecar-operator	CPU：2 C 内存：4 GB 磁盘：50 GB	4	8	16	200	-
MOSN	CPU：2 C 内存：4 GB 磁盘：50 GB	1	2*N	4*N	50*N	-

1.3. 部署方案

服务网格部署分为中间件租户部署和用户部署。

部署方案图如下：



在中间件租户会部署以下组件：

- 微服务管控：服务治理后台，用户可以在微服务管控系统上查看服务实例信息、配置服务限流、路由规则等。
- Sidecar 管理：边车管理后台，用户可以在该系统上查看 Sidecar 信息进行运维操作等。
- 服务注册中心：服务信息都会注册到该系统，同时也提供服务发现接口供调用者做服务寻址。
- DRM：动态配置中心，用于动态下发用户的配置。
- Sidecar Operator：边车运维系统，负责注入、升级 Sidecar。
- OSP：运维支持系统，主要提供和管理中台的元数据服务，是被其他应用所依赖的基础服务系统。如 tenant、workspace 管理中心、用户等。
- ACVIP：虚拟域名寻址服务,提供中间件服务端寻址的facade，并且可以进行不通纬度的服务端灰度。
- Intelliproxy：智能服务网关，用来转发路由请求、统一权限和 Cookie 等切面管理，提供统一访问入口。

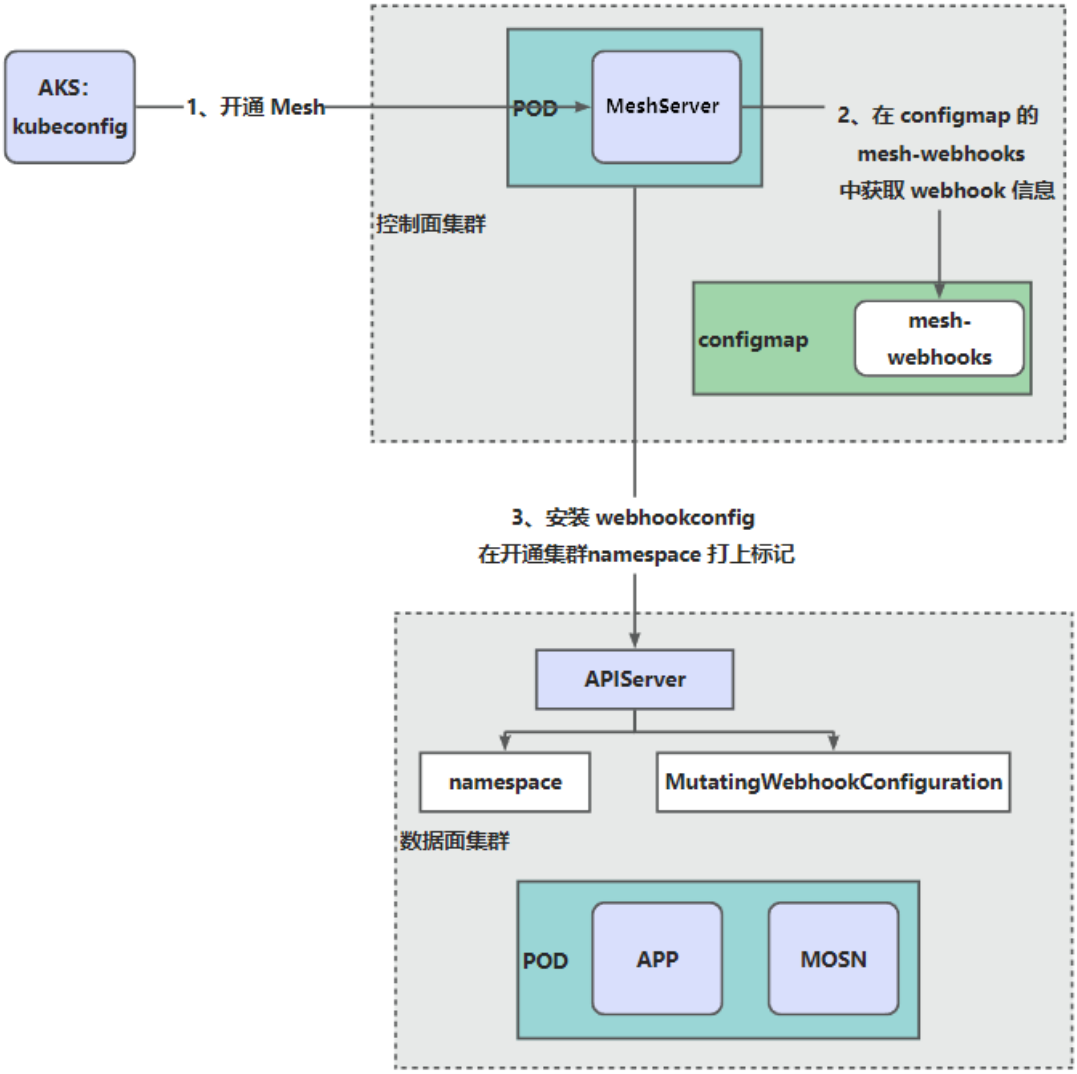
在用户租户会部署以下组件：

- MOSN：在用户应用的每个 Pod 中会注入 MOSN，用于拦截服务的流量，从而提供服务治理、流量调拨、通信加密等服务网格的能力。
- citadel-agent：安全组件的 daemon set，提供证书下发和轮换服务。
- sidecar-agent：负责接管所有 Sidecar的生命周期。

1.4. 核心组件交互

1.4.1. 开通集群

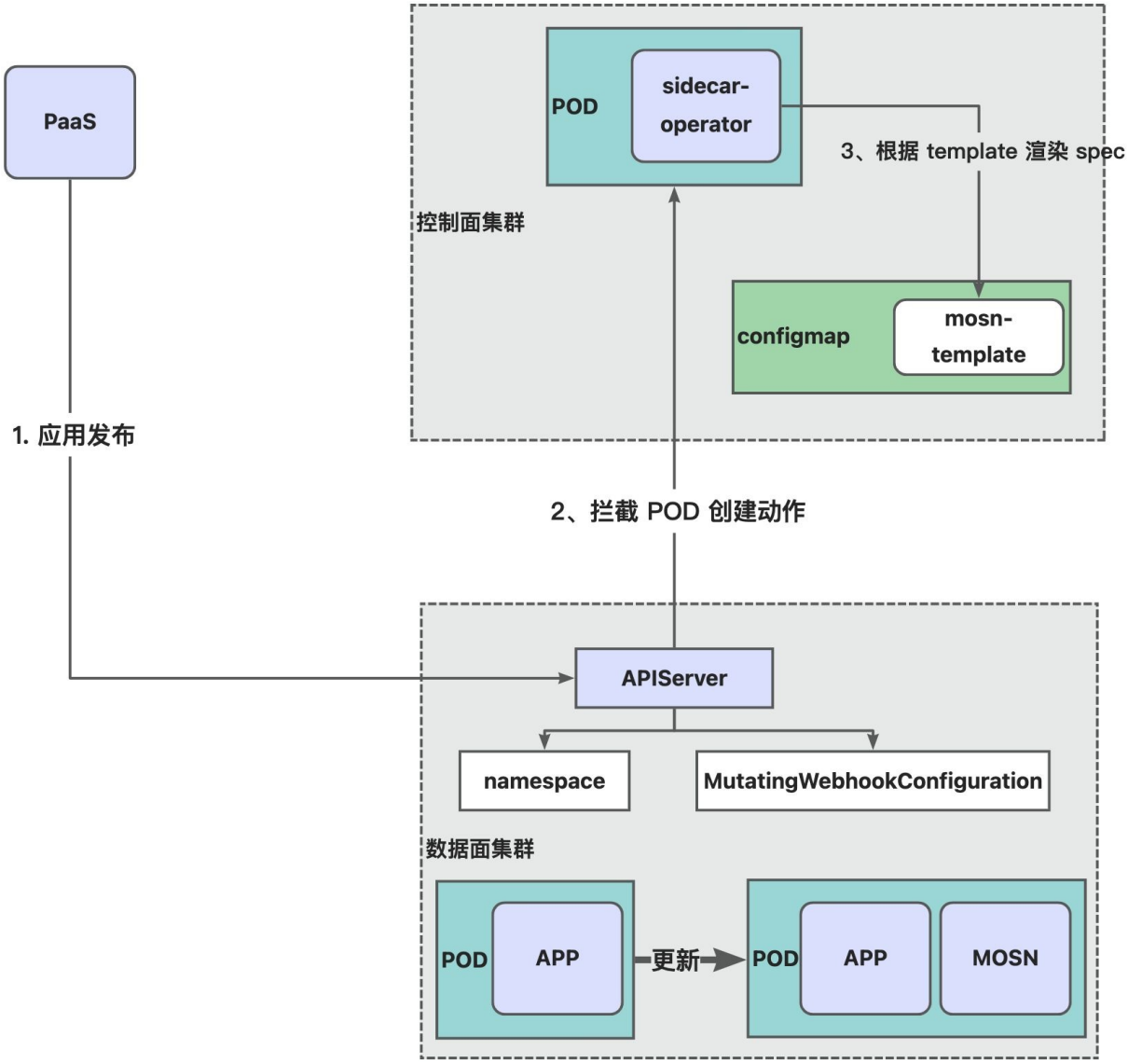
开通集群时涉及的流程和组件如下：



1. 任何一个 paas 平台或者手动录入，获取数据面的 kubeconfig 。
2. 在控制面读取 configmap 中 mesh-webhook 信息生成 webhook 基本信息。
3. CloudMesh 会通过 kubeconfig 向对应的数据面集群 apply 对应的配置，主要包括 webhook 的配置和 namespace 信息。

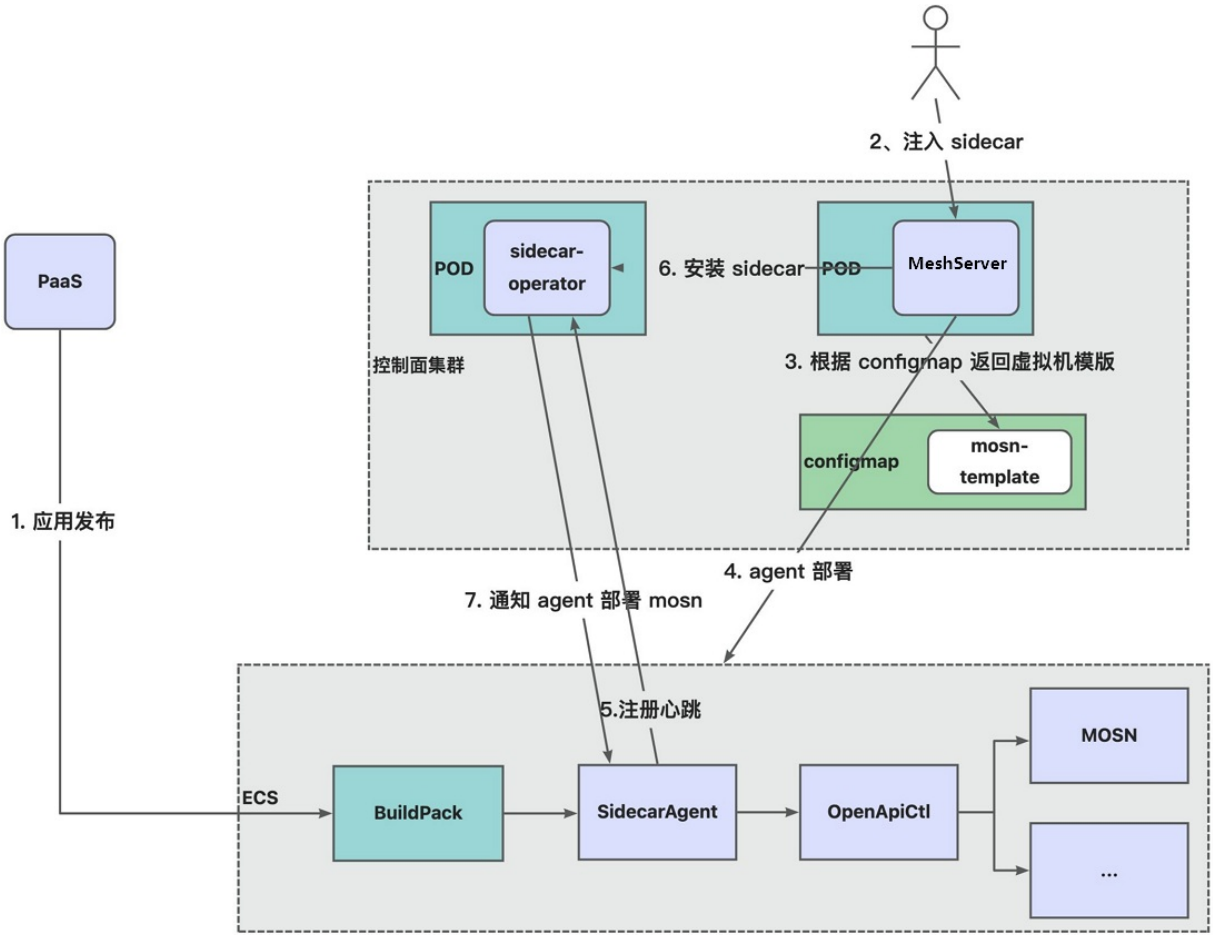
1.4.2. Sidecar 注入

容器场景



1. 在 PaaS 等平台进行应用的发布，发布时会将 deployment 或 StatefulSet 等资源 apply 到 APIServer，最后映射到 Pod 资源。
2. 由于在开通集群时已经设置了 webhook，所以 Pod 的 Create 事件会通过 APIServer 的 hook 机制转发到 Sidecar-Operator。
3. Sidecar-Operator 会根据是否注入 Sidecar 的白名单规则，选择是否注入 Sidecar，并且通过 configmap 中对应的 Sidecar 模版进行渲染，最后将渲染完成的 `pod.yaml` 返回给 APIServer。

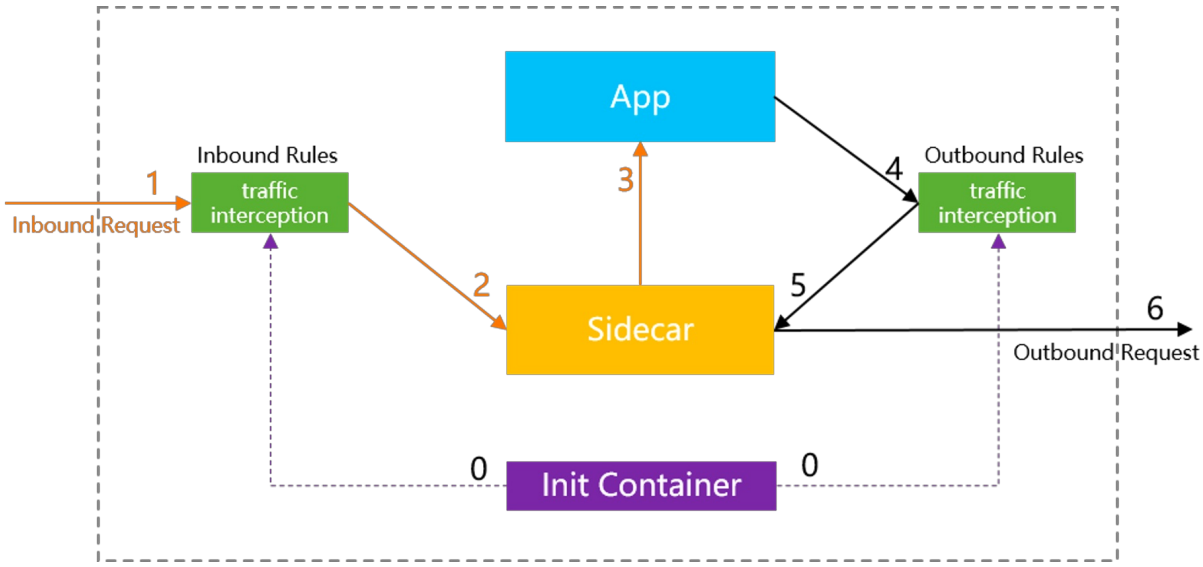
虚拟机场景



1. 用户选择支持 Mesh 的技术栈进行应用发布，并且指定开启 Mesh。
2. 在发布应用时，技术栈在应用部署过程中判断是否开启 Mesh。如开启，则安装 Sidecar Agent，并且通知 MeshServer 控制台进行 Sidecar 的注入。
3. 控制台会根据 configmap 模版生成注入配置后，MeshServer 会登陆虚拟机部署 sidecarAgent 进程。
4. Sidecar-Agent 上报元数据给 Sidecar-Operator，包括 tenant、workspace、instance、指定的 label 作为标签等，Sidecar-Operator 把元数据存储为一个单独的 configMap。
5. MeshServer 发请求，调用启动 Sidecar-Operator 的接口启动 Sidecar 。
6. Sidecar-Operator 根据 MeshServer 传过来的 IP、tenant、workspace 等字段从缓存中获取到对应的元数据，并将元数据中的 label 等字段作为选择器，获取 Sidecar 模版信息，然后发起部署。
7. 技术栈继续部署业务应用。
8. 业务应用发现当前处于 MOSN 环境时就会自动向 MOSN 发布或订阅服务，从而实现流量接管。

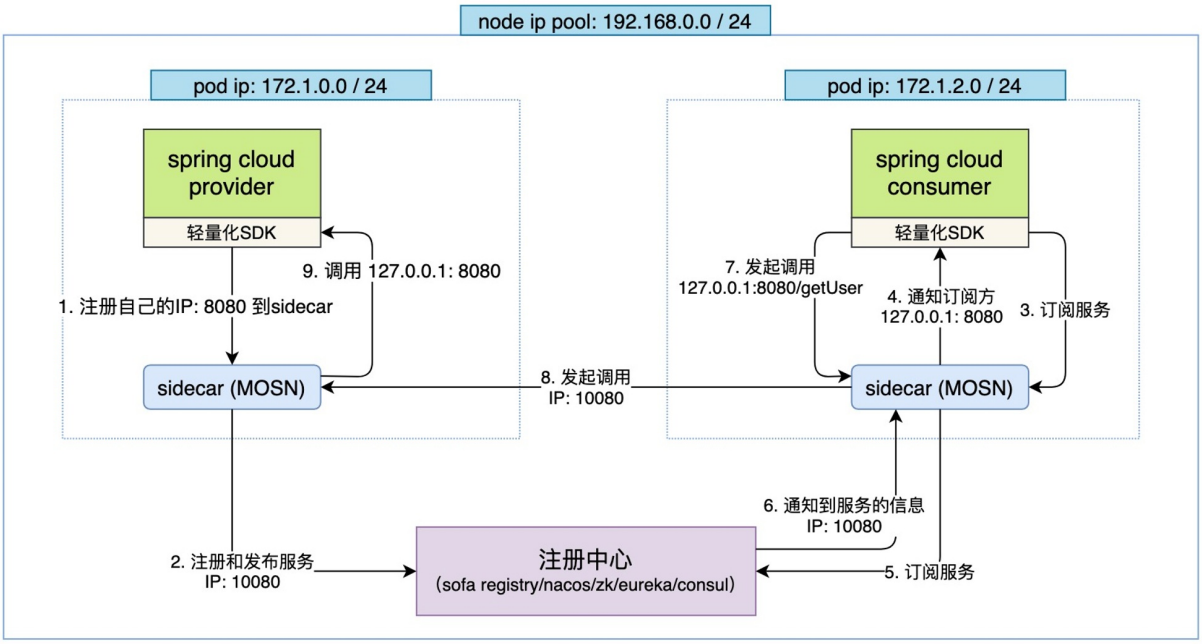
1.4.3. 透明劫持

流量劫持



1. 在容器初始化过程中，iptable 规则会写入 Pod 中，并随着请求流入进行规则甄别，判断端口是否命中规则。
2. 命中规则的服务会自动转发到 Sidecar 对应协议端口中。
3. 请求经过 Sidecar 处理结束，转发给 App。
4. App 发出请求，通过 iptable 规则确认是否需要转发到 sidecar 。
5. sidecar 收到 App 发出的请求，进行流量处理后转发出去。
6. 请求发给对应的节点。

端口欺骗

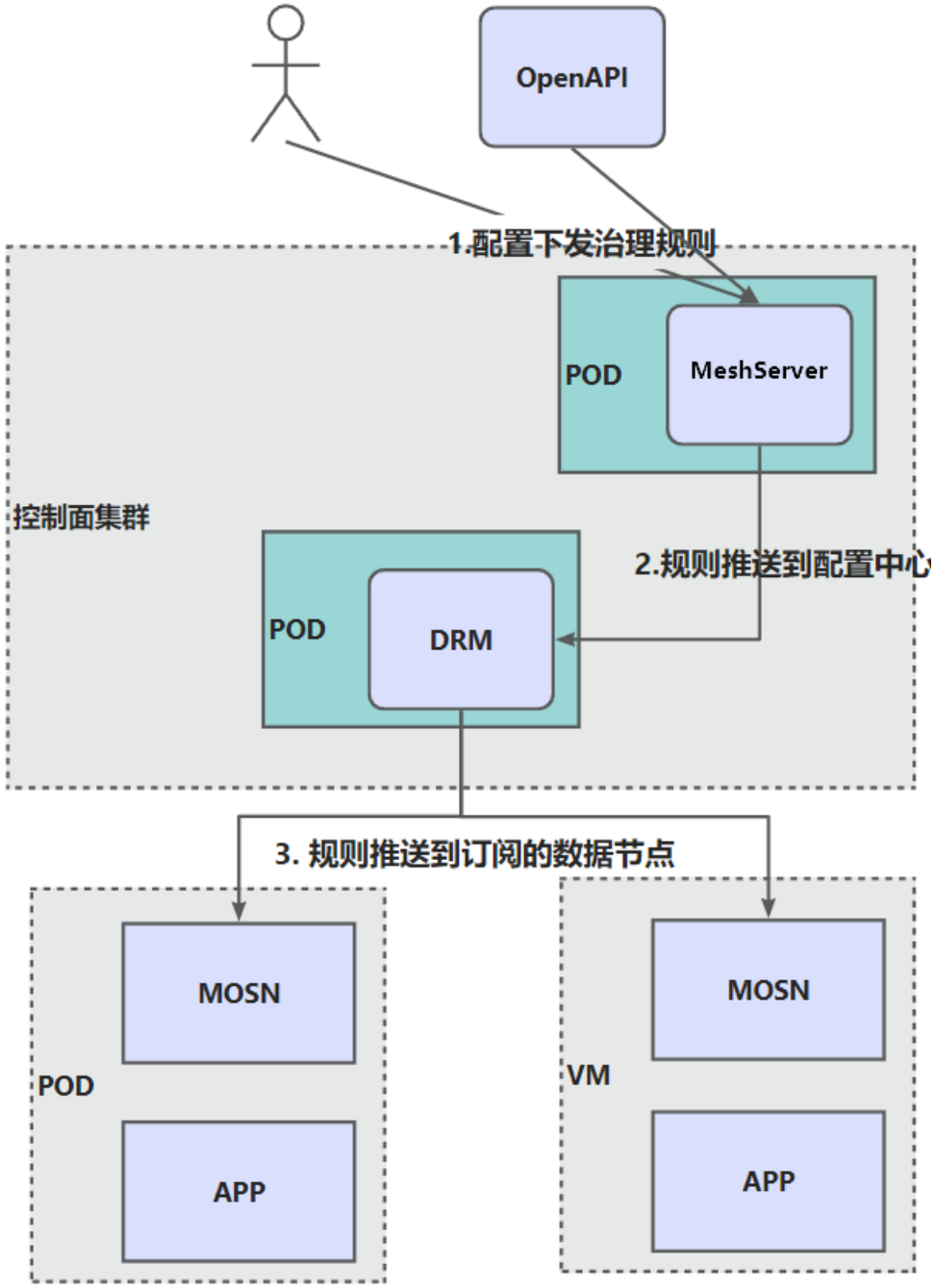


1. provider 发起注册，将 `自身 IP:8080` 注册到 Sidecar。
2. Sidecar 将请求转发给注册中心，其中注册的信息由 `provider IP:8080` 修改为 `provider IP:10080`（10080 端口是 MOSN 监听的端口）。

3. consumer 向 Sidecar 发起订阅。
4. Sidecar 收到订阅后通知 consumer 订阅成功，订阅地址为 `127.0.0.1:8080`（MOSN 监听的地址）。
5. Sidecar 向注册中心发起订阅。
6. 注册中心通知 Sidecar 订阅成功，订阅地址为 `provider IP:8080`。
7. consumer 向 `127.0.0.1:8080` 发起调用。
8. Sidecar 将请求转发给 `provider IP:8080` 节点。
9. 节点 Sidecar 收到对端 Sidecar 发过来的请求，转发给 `127.0.0.1:8080` 的 provider。

1.4.4. 服务治理

服务治理涉及的流程与组件如下：

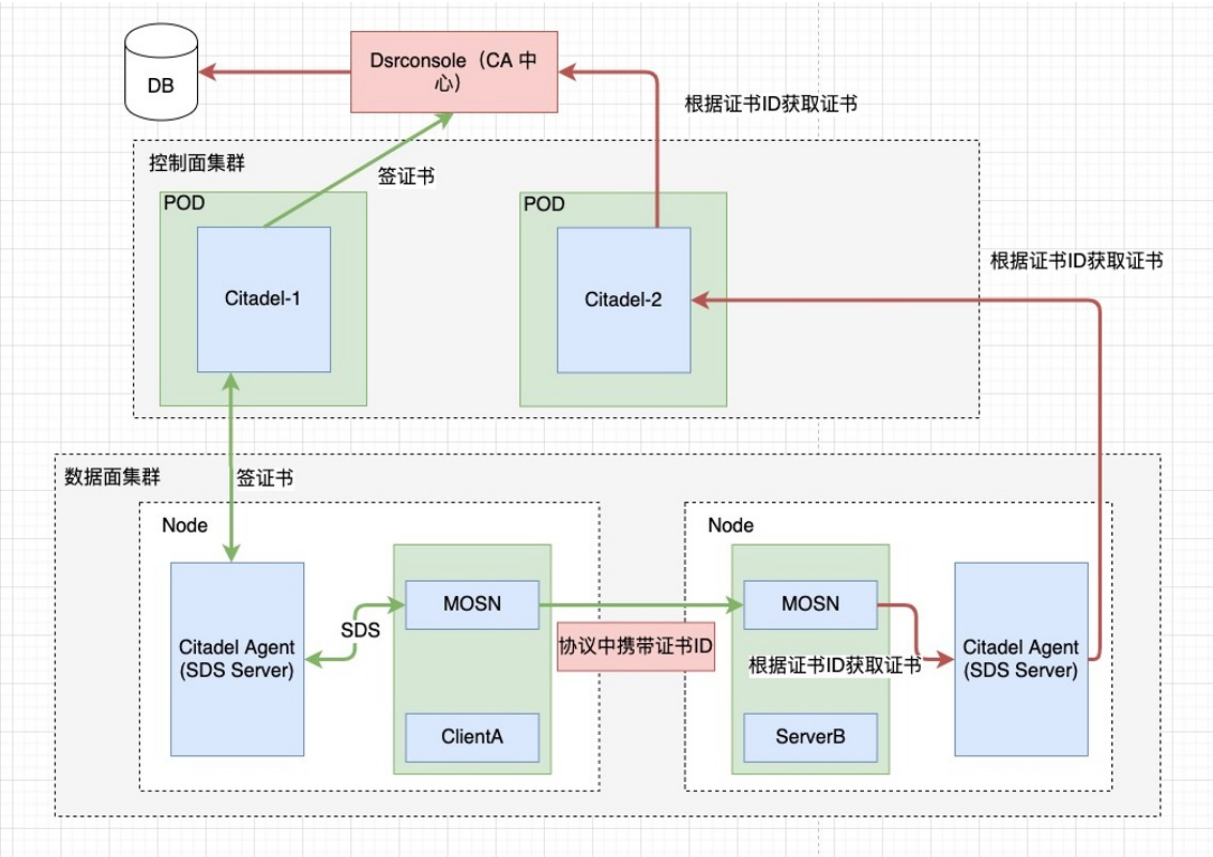


1. 用户通过 OpenAPI 或控制台配置治理规则。
2. MeshServer 会把订阅的规则 ID 和规则信息发送到 DRM 配置中心。
3. DRM 根据订阅 ID 将配置信息下发到 MOSN 的订阅节点。

1.4.5. 安全功能

证书颁发和获取流程

证书的颁发和获取流程如下：



1. MOSN 收到 DRM 下发的配置后，会根据下发的 SDS 配置向 Citadel Agent 发起签发证书（SDS）请求。
2. Citadel Agent 收到请求后会根据 ConnectID，从 Citadel MCP Server 下发的 Pod 信息中获取 MOSN 应用信息，然后向 Citadel 发起签发证书（SDS）请求（请求上下文中将携带应用信息），获取应用级证书。
3. Citadel 收到请求后，会将请求转发给 DsrCA（dsrconsole）。
4. DsrCA 收到请求后，将根据应用名称查询应用级证书返回给 Citadel。若证书不存在，将根据应用名称新签证书。
5. Citadel 收到响应后，依次返回给 Citadel Agent，Citadel Agent 再推送给 MOSN，完成证书签发流程。

> **说明**
>
> DsrCA 作为CA中心，具备签发证书、证书轮转、证书吊销等能力。 同时也会将 CA 证书，写入到 ConfigMap（dsr-ca-security）中，用于 DsrCA 与 Citadel 的 TLS 通信，保证签发链路的安全性。

2.日常运维

2.1. 监控和预警

2.1.1. 预警项运维动作

Citadel

预案名称	类型	架构域名称	风险	有无业务影响	触发条件	正常值	预案动作	影响场景与执行效果
熔断/限流/降级	应急	中间件-微服务和Mesh	中	有	内存使用率超过80%。	citadel_mcp_push_concurrency：1000 citadel_mcp_rate_limit：0.1s	修改 Configmap 中的 citadel_mcp_push_concurrency 值至 500，citadel_mcp_rate_limit 值至 0.5s，实现降级。	影响：超过阈值的 nodeAgent 需重连 citadel，配置下发及时性下降。 执行效果：内存消耗下降。
日志降级	应急	中间件-微服务和Mesh	低	无	IO 高水位	info	通过 Pilot 的 admin 端口调整日志级别：`curl 'http://{PodIP}:8080/logging?all=error'`	执行效果：只输出 error 级别的日志，IO 降低。
宕机重启	应急	中间件-微服务和Mesh	低	无	容器状态异常	-	尝试杀掉并重启容器。	实例重启

- 查看配置命令，示例如下：

```
GET curl":15014/admin/v1/dynamic_config"
Response:ConnectionFreq=1s
ConnectionBurstSize=100
PushConcurrency=1000
```

- 修改日志级别，示例如下：

```
curl "localhost:15014/logging"
```

- Configmap 配置，示例如下：

```
customParams:
citadelMcpRateLimit: "0.5s"
citadelMcpPushConcurrency: "500"
```

Nodeagent

预案名称	类型	架构域名城	风险	有无业务影响	触发条件	正常值	预案动作	影响场景与执行效果
日志降级	应急	中间件-微服务和mesh	低	无	IO 高水位	info	通过nodeagent 的 admin 端口调整日志级别：curl '127.0.0.1:15014/logging? default=error'	执行效果：只输出 error 级别的日志，IO 降低。

MOSN

预案名称	类型	架构域名城	风险	有无业务影响	触发条件	正常值	预案动作
日志降级	应急	中间件-微服务和mesh	低	无	IO 高水位	info	通过 nodeagent 的 admin 端口调整日志级别： `curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/json' -d'{"log_level":"ERROR","log_path":"./logs/mosn/default.log"}'`

2.1.2. 日常巡检项

在日常巡检时，您需要检查监控预警是否有效，检查预案脚本是否需要因为部署变更而调整。

2.2. 系统日志

2.2.1. 日志文件清单

MeshServer 、OSP、Intellproxy、AntVIP、MOSN、Citadel、Citadel-Agent 组件前往 `/home/admin/logs/` 目录查看相关日志。

2.2.2. 日常巡检项

> **说明**
>
> Mesh 1.11.0 及以后的版本会自动清理日志，无需进行以下操作。

1. 登录 **容器** 后，进入日志目录。
2. 如果发现日志没有滚动且超出了 512 MB，并且宿主机磁盘有报警，请按以下操作清理日志。

```
$cd /home/admin/logs/
$du -sh *
$echo "" > *.log
$rm *.log
```

3.常见问题

3.1. 监控告警手册

控制面

MeshServer

应用监控

监控指标	指标说明	类型	预警触发条件	告警等级	应急运维措施
checkservice	进程存活总量	应用	alive 当前时间 = 0.0	P0（紧急）	重启
checkservice_12200_8080_8341_alive	端口探活	应用	dead 最近2分钟持续 > 0.0	P1（重要）	重启
jvmgc	JVMGC 时间	应用	fgc_count 当前时间 > 1.0	P2（次要）	扩容
CPU	CPU 使用率	应用	cpu_util 最近5分钟持续 > 60.0	P2（次要）	扩容
Mem	内存使用率	应用	mem_util 当前时间 > 80.0	P2（次要）	扩容
DISK	磁盘使用率	应用	max_partition_util 当前时间 > 80.0	P2（次要）	扩容
应用 DAL 报错	访问 DB 报错	应用	fail 最近2分钟持续>0	P2（次要）	检查 DB 连接。
应用 CE 积压	线程池积压	应用	count 最近2分钟持续 > 0.0	P2（次要）	重启
应用 Error	应用Error统计告警	应用	count 最近2分钟持续 > 0.0	P2（次要）	查看 common-error 确认具体报错原因。

应用 SAL 报错	调用外部服务异常统计	应用	fail 最近2分钟持续 > 0.0	P2（次要）	查看 common-error 确认具体报错原因。

业务监控

监控指标	指标说明	类型	预警触发条件	告警等级	应急运维措施
app_sync_data	注册中心同步应用数据	业务	count 最近3分钟求和与上3分钟和环比 下跌超过 50.0	P2（次要）	检查和注册中心的链接是否正常或尝试重启。
bolt_heartbeat_error	Bolt 协议心跳报错	业务	count 当前时间 > 0.0	P2（次要）	重启
slow_slq	慢 SQL 告警	业务	count 当前时间 > 10.0	P2（次要）	检查数据库压力。
registry_auth_failed	用户鉴权失败	业务	count 当前时间 > 10.0	P2（次要）	确认是否有恶意访问。
rest_client_error	访问外部 REST 报错	业务	count 当前时间 > 10.0	P2（次要）	查看日志确认具体报错原因。
rest_server_error	调用 dsr 的 REST 接口报错	业务	count 当前时间 > 10.0	P2（次要）	查看日志确认具体报错原因。
rpc_client_error	访问外部 RPC 报错	业务	count 当前时间 > 10.0	P2（次要）	查看日志确认具体报错原因。
rpc_server_error	调用 dsr 的 RPC 接口报错	业务	count 当前时间 > 10.0	P2（次要）	查看日志确认具体报错原因。

SofaRegistry

产品	应用	监控类别	监控项名称	业务说明	使用场景	风险等级	告警配置	配置说明

SOFA	registry-meta	基础监控	System(系统指标)	系统指标	系统指标	P1	CPU使用率 > 90%内存使用率 > 90%磁盘使用率 > 90%	监控基础应用监控
SOFA	registry-meta	基础监控	checkService（端口指标）	端口探活	端口探活	P1	9610	监控基础应用监控
SOFA	registry-meta	业务监控	meta 错误日志	通用错误日志	通用错误日志	p4	最近5分钟持续大于0	/home/admin/logs/regisry/meta/common-error.log
SOFA	registry-meta	业务监控	metaRenew_fail_prod	跟data/session续约	跟data/session续约	p4	最近3分钟持续大于0	/home/admin/logs/registry/meta/common-error.log左起第 1 个 } 右至 ! 白名单 reNew error
SOFA	registry-meta	业务监控	dataNodeList_is_empty_prod	data列表为空	data列表为空	p1	当前值大于0	/home/admin/logs/registry/meta/registry-metrics.log 左起第 3 个 = 右至 } 白名单：{ 左起第 1 个 = 右至, 白名单：metaNodeList,sessionNodeList
SOFA	registry-meta	业务监控	meta_health_check_fail_prod	健康检查失败	健康检查失败	p1	最近3分钟持续大于0	/home/admin/logs/registry/meta/registry-metrics.log 左起第 1 个 = 右至, 白名单：healthCheck 左起第 3 个 = 右至, 白名单 false
SOFA	registry-meta	业务监控	meta_push_status_is_closed_prod	推送关闭	推送关闭	p4	最近3分钟持续大于0	/home/admin/logs/registry/meta/registry-metrics.log 左起第 1 个 = 右至 , 白名单：pushSwitch 左起第 3 个 = 右至 {， 白名单：closed

SOFA	registry-meta	业务监控	sessionNodeList_is_empty_prod	session列表为空	session列表为空	p1	当前时间的值 > 0	/home/admin/logs/registry/meta/registry-metrics.log 左起第 3 个 = 右至 } 白名单：{ 左起第 1 个 = 右至 , 白名单：metaNodeList,dataNodeList
SOFA	registry-meta	业务监控	meta_gc_cms_prod	meta gc	meta gc	p4	最近10分钟求和大于5	/home/admin/logs/gc.log 左起第 6 个 右至: 白名单 CMS-initial-mark
SOFA	registry-meta	业务监控	meta_role_change_prod	meta角色变化	meta角色变化	p4	-	/home/admin/logs/registry/meta/registry-raft.log左起第3个[右至] 白名单：StateMachineAdapter
SOFA	registry-meta	业务监控	metaNodeList_is_empty_prod	meta列表	meta列表	p1	当前时间值大于零	/home/admin/logs/registry/meta/registry-metrics.log左起第3个=右至} 白名单 {左起第1个=右至, sessionNodeList,dataNodeList
SOFA	registry-session	基础监控	System(系统指标)	系统指标	系统指标	P1	CPU使用率 > 90%内存使用率 > 90%磁盘使用率 > 90%	监控基础应用监控
SOFA	registry-session	基础监控	checkService（端口指标）	端口探活	端口探活	P1	9600 9603	监控基础应用监控
SOFA	registry-session	业务监控大盘	sessionPub_total	PUB总数	业务PUB数据时	不涉及	不涉及	/home/admin/logs/registry/session/registry-console.log 左起4个[右至] 白名单 Count， 对列求平均值左起第8个 右至,

SOFA	registry-session	业务监控大盘	session_Sub_total	SUB总数	业务SUB数据时	不涉及	不涉及	/home/admin/logs/registry/session/registry-console.log 左起4个[右至] 白名单 Count， 对列求平均值左起第5个 右至，
SOFA	registry-session	业务监控大盘	session_connect_total	链接总数	客户端连接数	不涉及	不涉及	/home/admin/logs/registry/session/registry-console.log 左起4个[右至] 白名单 Count， 对列求平均值左起第14个 右至
SOFA	registry-session	业务监控	session_ClientOff_notifyDataFail_prod	客户端断链session通知data失败	客户端断链	低	最近30分钟求和>10	/home/admin/logs/registry/session/common-error.log 白名单 CliendOff 白名单failed
SOFA	registry-session	业务监控	session_receive_sub_check_data_finally_giveUp(missing_client_data)(no_servers)_prod	session放弃校验数据任务	异步任务定时检查	低	当前值大于0	/home/admin/logs/registry/session/common-default.log 白名单 SubscriberRegisterFetchTaskDispatcher

SOFA	registry-session	业务监控	session_setZeroBySubFail_prod	数据访问缓存不存在，会远程获取，设置version为0，是为了让定期任务，在去跟DataServer轮巡时，能发现version落后，然后重新拿	获取数据时	低	最近60分钟求和大于0	/home/admin/logs/registry/session/common-error.log 左起8个 右至(白名单checkAndUpdateInterestVersionZero
SOFA	registry-session	业务监控	session_error_prod	错误总数	error日志	p4	最近10分钟>200	/home/admin/logs/registry/session/common-error.log
SOFA	registry-session	业务监控	sessionRenew_fail_prod	数据定义校验失败，会定时发起	定时任务校验数据	低	最近两分钟大于0	/home/admin/logs/registry/session/common-error.log左起3个 右至 白名单reNew左起第4个 右至 白名单node求行数
SOFA	registry-session	业务监控	session_push_error_prod	推送失败报错	根据链接向sub方推荐链接	低	最近5分钟大于200	/home/admin/logs/registry/session/common-push-error.log求行数

SOFA	registry-session	业务监控	sessionSyncConsoleError_prod	向MeshServer 同步报错	定时向MeshServer 同步	低	最近60分钟大于200	/home/admin/logs/registry/session/common-error.log左起第3[右至]，白名单ConsoleServiceImpl
SOFA	registry-session	业务监控	sessionConDataFailed_Prod	链接data失败	session向data发起建联	p3	最近5分钟大于0	/home/admin/logs/registry/session/common-error.log左起第2个-右至-，白名单connectDataServer
SOFA	registry-session	业务监控	sessionConMetaFailed_Prod	连接meta失败	session向meta建联	p3	最近5分钟大于0	/home/admin/logs/registry/session/common-error.log左起第2个-右至-，白名单connectMetaServer
SOFA	registry-session	业务监控	sessionRefreshLeader_fail_prod	向meta获取leader失败	定时向meta刷新leader	低	最近3分钟持续大于0	/home/admin/logs/registry/session/common-error.log左起第2个 右至白名单Refresh leader failed左起第3个 右至 白名单leader
SOFA	registry-session	业务监控	session_pub_notifyDataFail_prod	客户端pub到session通知data	定时通知datapub信息	低	最近30分钟求和大于0	/home/admin/logs/registry/session/common-error.log左起第2个 右至白名单PublishData左起第3个 右至 白名单failed
SOFA	registry-session	业务监控	session_setZeroByPushExceed_prod	session设置版本0重试	session设置版本0重试	低	最近60分钟求和大于50	/home/admin/logs/registry/session/registry-push.log左起第3个 右至白名单Retry左起第1个-右至 白名单exceeded
SOFA	registry-session	业务监控	鉴权失败	pub/sub鉴权失败	pub/sub鉴权失败	低	最近1分钟大于10	/home/admin/logs/registry/session/registry-session.log左起第3个 右至: 白名单is refused by authCheck

SOFA	registry-data	基础监控	System(系统指标)	系统指标	系统指标	P1	CPU使用率 > 90%内存使用率 > 90%磁盘使用率 > 90%load5 > 3	监控基础应用监控
SOFA	registry-data	基础监控	checkService（端口指标）	端口探活	端口探活	P1	9620	监控基础应用监控
SOFA	registry-data	业务监控大盘	dataPub_total_prod	PUB总数	业务PUB数据统计周期内平均	不涉及	不涉及	/home/admin/logs/registry/data/cache-digest.log 左起5个[右至]白名单 datum
SOFA	registry-data	业务监控	dataDatum_total_prod	数据总量	数据总量	不涉及	>100000	/home/admin/logs/registry/data/cache-digest.log 左起5个[右至]白名单 datum， 对列求平均值左起第9个 右至
SOFA	registry-data	业务监控	data_common_error_prod	报错总量	报错监控	P3	最近5分钟持续>0 短信报警最近10分钟求和 >100 短信报警只要满足一个条件就报警	/home/admin/logs/registry/data/common-error.log 黑名单 左起2个，右至 黑名单 invokeWithCallback
SOFA	registry-data	业务监控	dataRefreshLeader_fail_prod	data从meta获取选取leader失败	data从meta获取新选的leader失败	P3	最近3分钟持续>0 短信报警	/home/admin/logs/registry/data/common-error.log 左起2个右至 白名单 Refresh leader failed，左起3个右至，白名单leader

SOFA	registry-data	业务监控	dataRenew_fail_prod	数据更新失败	-	P3	最近3分钟持续>0 短信报警	/home/admin/logs/registry/data/common-error.log 左起5个，右至白名单 ReNewNodeTask
SOFA	registry-data	业务监控	data_health_check_fail_prod	健康检查	服务健康检查	P3	最近 3 分钟持续 > 0 短信告警.	/home/admin/logs/registry/data/registry-metrics.log 左起第1个=右至 白名单 healthCheck，左起第3个=右至 白名单 false
SOFA	registry-data	业务监控	dataRenewEvict_Expired_prod	数据剔除超时	数据更新及时性检查	P4	最近 60 分钟求和 > 100 短信告警	/home/admin/logs/registry/data/registry-datum-lease.log 左起第2个 右至常用分隔符 白名单 Evict，
SOFA	registry-data	业务监控	dataRenew_Snapshot_prod	更新快照	更新快照数量	P4	最近 60 分钟求和 > 100 短信告警	/home/admin/logs/registry/data/registry-renew.log 左起第4个 右至! 白名单 different
SOFA	registry-data	业务监控	data_SessionNotifierInvoke_Error_prod	通知session回调	通知session回调失败数量	P4	最近 10 分钟求和 > 100 短信告警.最近 10 分钟持续 > 0 短信告警. 只要满足一个就发报警	/home/admin/logs/registry/data/common-error.log 左起第3个[右至] SessionServerNotifier，左起第2个 右至常用分隔符 invokeWithCallback
SOFA	registry-data	业务监控	data_gc_cms_prod	gc	gc报警	P4	CMS-initial-mark	/home/admin/logs/gc.log 左起第6个 右至: CMS-initial-mark
SOFA	registry-data	业务监控	dataRenewEvict_Noheartbeat_prod	异步更新数据剔除无新跳	异步更新数据剔除无新跳	P4	最近 60 分钟求和 > 100 短信告警	/home/admin/logs/registry/data/registry-datum-lease.log/home/admin/logs/registry/data/registry-datum-lease.log 左起第2个 右至 Evict, 左起第5个 右至 no

SOFA	registry-data	业务监控	data_SessionNotifierRetryExceed_Error_prod	通知seesion更新重试次数	数据时效	P4	最近 10 分钟求和 > 0 短信告警	/home/admin/logs/registry/data/common-error.log 左起第3个[右至]，SessionServerNotifier；左起第2个 右至常用分隔符 retryTimes，左起第4个 右至! exceeded
SOFA	registry-data	业务监控	data_LocalDataServerChangeEvent_prod	数据节点变化	数据节点变化	P4	最近 10 分钟求和 > 2 短信告警	/home/admin/logs/registry/data/registry-data.log 左起第3个[右至] DataServerChangeEventHandler，左起第6个 右至，LocalDataServerChangeEvent

Intelliproxy

系统监控

监控指标	指标说明	类型	预警触发条件	告警等级	应急运维措施
checkservice	进程存活总量	应用	alive 当前时间 = 0.0	P0（紧急）	重启
checkservice_80_alive	端口探活	应用	alive 当前时间 = 0.0	P1（重要）	重启
jvmgc	JVMGC时间	应用	fgc_count 当前时间 > 1.0	P2（次要）	扩容
Cpu	CPU使用率	应用	cpu_util 最近5分钟持续 > 80.0	P2（次要）	扩容
Mem	内存使用率	应用	mem_util 当前时间 > 85.0	P2（次要）	扩容
DISK	磁盘使用率	应用	max_partition_util 当前时间 > 85.0	P2（次要）	扩容

业务监控

监控指标	指标说明	告警对象	类型	预警触发条件	告警等级	日志格式	应急运维措施
access日志监控	请求响应码监控	INTELLIPROXY-intelliproxy	业务	响应码500、400、403 一分钟出现超过 100 次	P2（次要）	2022-11-14 17:50:14.577 -|200|POST|http://region-sofastack-middleware-hzfinprod-osp.cloud.alipaycs.net/openapi/ms/openapi/sofa/ms/ddcs/attributes/query|osp|10.101.53.26:80|1391|-|23ms|-|	查看 access.log 判断异常服务，联系相关业务。

数据面

MOSN

系统指标

监控指标	监控指标说明	类型	预警触发条件	应急运维措施
MOSN CPU	MOSN CPU使用率	应用	>80	扩容
MOSN mem	MOSN 内存使用率	应用	>80	扩容
MOSN Error	MOSN 异常数，统计 MOSN 日志中 ERROR 的行数。	应用	连续2分钟>100	查看报错日志，具体分析。
MOSN Health	MOSN 健康状态，执行 `/home/admin/mosn/bin/process_checker.sh`	应用	连续3次不通过	重启
MOSN Port	MOSN 端口探活，检查34901、34902、34903 等端口（协议端口如 12200 等业务可自定义）	应用	连续3次端口异常	重启

业务指标

监控指标	监控指标说明	类型	预警触发条件	应急运维措施

MOSN RPC 调用量	采集 /home/admin/logs/tracelog /mosn/rpc-client-digest.log 行数	业务	count 每分钟>6 万（1Ktps）	扩容
MOSN RPC 调用耗时	采集 /home/admin/logs/tracelog /mosn/rpc-client-digest.log 耗时字段求平均	业务	count 最近 1 分钟求平均 >100	扩容
MOSN RPC 调用成功率	采集 /home/admin/logs/tracelog /mosn/rpc-client-digest.log 取是否成功字段求总数	业务	fail 最近3分钟持续>0	查看日志确认具体报错原因。
MOSN RPC 被调用量	采集 /home/admin/logs/tracelog /mosn/rpc-server-digest.log 行数	业务	count 每分钟>6W(1Ktps)	扩容
MOSN RPC 被调用耗时	采集 /home/admin/logs/tracelog /mosn/rpc-server-digest.log 耗时字段求平均	业务	count 最近1分钟求平均 >100	扩容
MOSN RPC 被调用成功率	采集 /home/admin/logs/tracelog /mosn/rpc-server-digest.log 取是否成功字段求占比	业务	fail 最近3分钟持续>0	查看日志确认具体报错原因。

3.2. SOFAStack 常见问题

SOFAStack 是否可以部署在虚拟机、物理机上？

可以，SOFA 部署支持物理机、虚机 VM。

SOFAStack 当前是否只支持阿里云飞天 ACK？

SOFA 部署支持多云异构、目前支持华为云、开源 Openstack 等。

SOFAStack 是否可以被客户侧现有系统集成？

可以，SOFAStack 提供开放标准 API 接口，供客户侧系统调用集成。

3.3. 问题排查思路

准备工作

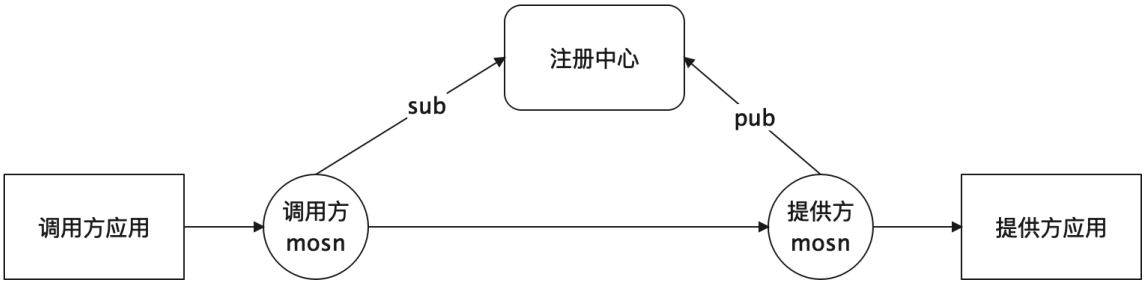
排查要素

排查问题前，请先明确以下几个要素：

- 客户端应用
- 服务端应用
- 调用的接口（dataId）
- traceId

调用链路

Mesh调用链路如下：



关键日志

MOSN 日志如下：

- /home/admin/logs/mosn/default.log：MOSN 流水日志。
- /home/agmin/logs/mosn/endpoint.access.log：MOSN 注册中心交互流水日志。

日志示例如下：

排查思路

1. 先根据当前异常，筛选出当前链路的几个要素（客户端应用、服务端应用、调用的接口（dataId））。
2. 根据 Mesh 控制台的 **服务目录** 页面查看当前 dataId 对应的服务提供者是否存在。

   - 如提供者不存在，则大概率是服务注册问题，请查看提供方应用是否正常、健康检查是否通过等。
   - 如果提供者存在，则大概率是客户问题，继续下一步。
3. 根据 traceId 查询 MOSN 的 default 日志，查看具体的错误信息。

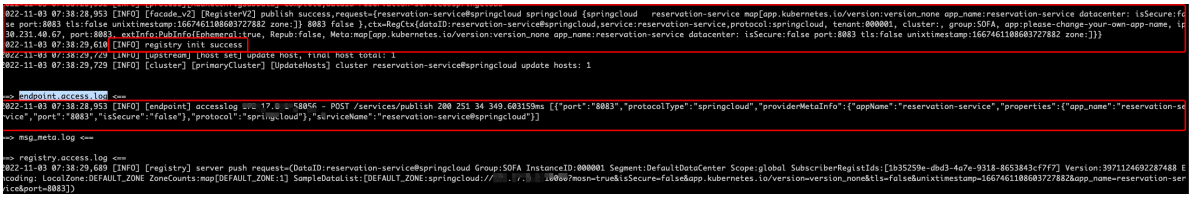
```
grep {traceId} /home/admin/logs/mosn/default.log
```

注意根据时间选择对应的日志文件。如果日志中没有明显的异常，可能是服务端问题，可继续按照 traceId 到服务端 MOSN 的 default.log 里查找

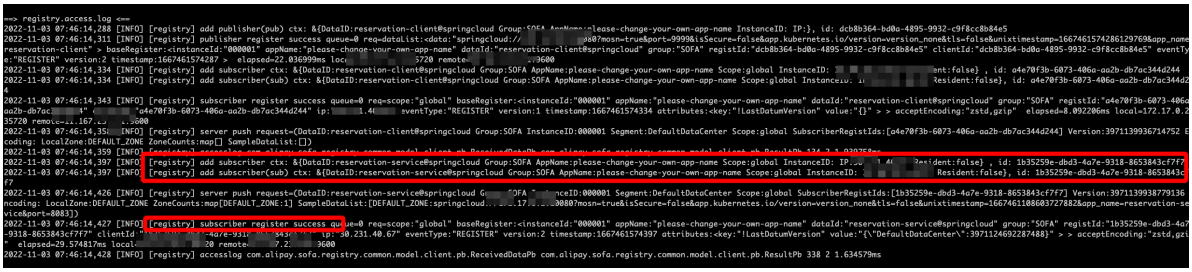
4. 若以上步骤未解决，请提供以下信息，并联系售后技术支持协助排查。
   - 客户端应用
   - 服务端应用
   - 调用的接口（dataId）
   - traceId
   - sofastack地址
   - 根据以上步骤排查的基本情况

附录：标准日志范例

- 服务端 pub 操作日志记录



- 客户端 sub 操作日志几率

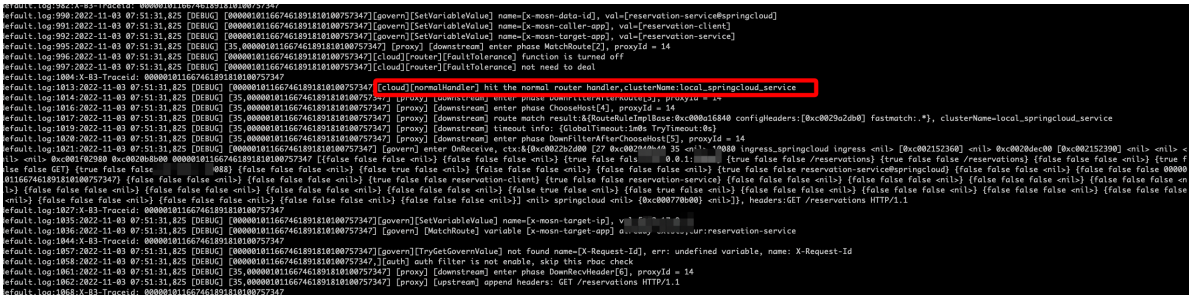


- RPC 通信记录

  如果没有开启 access 日志，需要借助 debug 日志排查，开启 debug 日志命令如下：

```
curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/j
son' -d'{"log_level":"DEBUG","log_path":"./logs/mosn/default.log"}'
```

  请求 MOSN 详细日志如下所示：



3.4. 常见问题排查

透明劫持

1. 登陆机器查看 iptable 配置。

```
iptables -L
```

2. 如果不存在配置，则确认以下问题：
   - 确认 Sidecar 注入模板是否配置正确，是否已开启。
   - 虚机场景，确认纳管的虚机账户是否有修改 iptables 权限。

MOSN 启动失败

排查思路一：通过 kubectl log 查看日志，确认问题

通过以下命令查看 default 日志， 确认启动失败原因。

```
 kubectl --kubeconfig=121sit -n cmbc exec -it mesh-demo-python-server-v1-0 -- tail -f ./logs/mosn/default.log
```

若日志为以下内容，则说明为租户问题，有两种可能：

- 手动注入时，未声明租户，也就是 annotaion 中没有声明 instance_id。解决方案如下：
  - 添加上对应的 `cafe.sofastack.io/instanceid: "{租户ID}"` 。
  - 登录控制台，在控制台注入 。
- 当前租户未在控制台开通当前产品。相关操作请参见 租户开通产品。

排查思路二：若查看日志不方便，尝试进入容器，手动启动后，查看日志

- 编辑 Sidecar 模板， 删除 lifecycle poststart 中的健康检查脚本，保证 Pod 不会被脚本重启。
- 编辑 Sidecar 模板， 重新规定 command，尝试不启动 MOSN 进程。容器拉起后，进入容器，手动启动 MOSN 观察报错。

例如将 commad 设置为 `["tail","-f","/dev/null"]` ：

```
  - name: command-demo-container
    image: debian
    command: ["tail","-f","/dev/null"]
    args: ["HOSTNAME", "KUBERNETES_PORT"]
```

服务治理

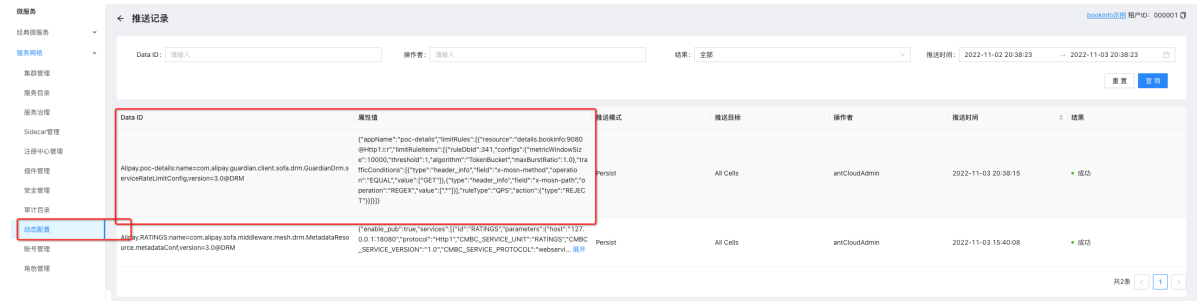
服务治理总体的排查思路如下：

1. 查看控制台配置是否正确。

   例如：应用名称是否配置正确、规则配置是否为预期的。

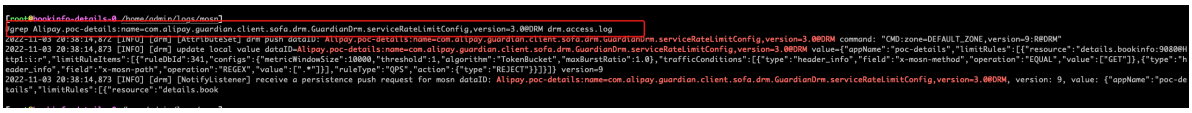
2. 查看控制台下发的配置是否下发成功。

   如下发失败应该是 DRM 服务异常。



3. 检查对应的规则是否下发到了 MOSN。

   可通过 `/home/admin/logs/mosn/drm.access.log` 查看规则下发内容。如未收到，则应该是 DRM 服务故障或应用名字填写错误。



4. 通过服务治理对应的关键字，判断生效的原因。

服务限流不生效

1. 确认 MOSN 是否收到控制面下发的指令。

   若未收到，则排查环境变量中 APPNAME 是否和控制台配置不一致。

   ```
   grep ${控制台下发DataID} drm.access.log
   ```

   此场景下需要开启 flowcontrol-default 文件日志 debug 日志级别进行排查。

   ```
   // 默认流控日志
   curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/j
   son' -d'{"log_level":"DEBUG","log_path":"/home/admin/logs/mosn/flowcontrol-default.log"}'
   ```

2. 查看日志 `flowcontrol-default.log` 请求参数匹配，排查 IsMatch 日志中是否控制台参数配置错误。

   ```
   // 参数解析
   2022-11-04 09:21:21,747 [DEBUG] [ParseResource] resource:&base.ResourceWrapper{name:"deta
   ils.bookinfo:9080@Http1:i:r", classification:2, flowType:0},sourceIp:172.16.0.40,targetAp
   p:poc-details,test:false,resType:2,trafficType:0
   // 参数生成
   2022-11-04 09:21:21,747 [INFO] [AcquireBusinessInfo] contextRefinedData=map[interface {}]
   interface {}{"X-CALLER-IP":"172.16.0.40", "X-TARGET-APP":"poc-details", "loadTest":false,
   "requestCtx":(*context.valueCtx)(0xc00365d020), "requestHeaders":http.RequestHeader{Reque
   stHeader:(*fasthttp.RequestHeader)(0xc00366f580)}, "system.sourceApp":"", "system.sourceI
   p":"172.16.0.40", "system.targetApp":"poc-details", "system.trafficType":"online", "syste
   m.uniqueId":""}
   // 参数匹配
   2022-11-04 09:21:21,747 [DEBUG] [IsMatch] leftVal(GET),op(EQUAL),rightVal([GET])
   // 请求参数匹配
   2022-11-04 09:21:21,747 [DEBUG] [IsMatch] leftVal(/details/0),op(REGEX),rightVal([.*])
   ```

3. 查看日志 `flowcontrol-block-stat.log`，如有请求命中纪录日志，则表示限流规则已命中。

```
2022-11-04 09:26:03,420 [ERROR] [flow blocked] |details.bookinfo:9080@Http1:i:r#f8238ada|
LocalLimitSlot|0|<nil>|maxPermits:1storedPermits:0genPermits:0permits:1|{"id":0,"ruleDbId
":341,"Resource":null,"thresholdType":"invokeTimes","threshold":1,"timeMs":10000,"mode":"
CONTROL","maxBurstRatio":1,"httpPath":"","httpMethod":"","zoneSpec":null}|
```

服务熔断不生效

1. 确认 MOSN 是否收到控制面下发的指令。

   若未收到，则排查环境变量中 APPNAME 是否和控制台配置不一致。

   ```
   grep ${控制台下发DataID} drm.access.log
   ```

   此场景下需要开启 flowcontrol-default 文件日志 debug 日志级别进行排查。

   ```
   // 默认流控日志
   curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/j
   son' -d'{"log_level":"DEBUG","log_path":"/home/admin/logs/mosn/flowcontrol-default.log"}'
   ```

2. 查看日志 `flowcontrol-default.log` 请求参数匹配，排查 IsMatch 日志中是否控制台参数配置错误。

   ```
   // 参数解析
   2022-11-04 09:21:21,747 [DEBUG] [ParseResource] resource:&base.ResourceWrapper{name:"deta
   ils.bookinfo:9080@Http1:i:r", classification:2, flowType:0},sourceIp:172.16.0.40,targetAp
   p:poc-details,test:false,resType:2,trafficType:0
   // 参数生成
   2022-11-04 09:21:21,747 [INFO] [AcquireBusinessInfo] contextRefinedData=map[interface {}]
   interface {}{"X-CALLER-IP":"172.16.0.40", "X-TARGET-APP":"poc-details", "loadTest":false,
   "requestCtx":(*context.valueCtx)(0xc00365d020), "requestHeaders":http.RequestHeader{Reque
   stHeader:(*fasthttp.RequestHeader)(0xc00366f580)}, "system.sourceApp":"", "system.sourceI
   p":"172.16.0.40", "system.targetApp":"poc-details", "system.trafficType":"online", "syste
   m.uniqueId":""}
   // 参数匹配
   2022-11-04 09:21:21,747 [DEBUG] [IsMatch] leftVal(GET),op(EQUAL),rightVal([GET])
   // 请求参数匹配
   2022-11-04 09:21:21,747 [DEBUG] [IsMatch] leftVal(/details/0),op(REGEX),rightVal([.*])
   ```

3. 查看日志 `flowcontrol-block-stat.log`，如有请求命中纪录日志，则表示熔断规则已命中。

   ```
   2022-11-04 11:19:02,318 [ERROR] [flow blocked] |reviews.allsite.alipay.net:9080@Http1:o:r
   #40f1a27a|FuseSlot|0|Block(ALWAYS_BLOCK)||{"id":0,"ruleDbId":14000001,"Resource":{"name":
   "reviews.allsite.alipay.net:9080@Http1","resourceType":2,"trafficType":1,"env":"","condit
   ionId":"40f1a27a"},"mode":"CONTROL","meltAlgorithm":"classic","recoveryAlgorithm":"classi
   c","triggerWindowMillis":10000,"floorRequestCount":2,"probeNum":2,"meltWindowMillis":1000
   0,"threshold":1,"triggerType":"errorRatio","zoneSpec":null}|
   ```

服务路由不生效

1. 确认 MOSN 是否收到控制面下发的指令。

   若未收到，则排查环境变量中 APPNAME 是否和控制台配置不一致。

   ```
   grep ${控制台下发DataID} drm.access.log
   ```

   此场景下需要开启 flowcontrol-default 和 routerule 日志 debug 日志级别进行排查。

```
// 默认流水日志
curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/j
son' -d'{"log_level":"DEBUG","log_path":"./logs/mosn/default.log"}'
// routerule.log 规则日志
curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/j
son' -d'{"log_level":"DEBUG","log_path":"/home/admin/logs/mosn/routerule.log"}'
```

2. 查看 routerule.log 日志是否命中路由规则，如果没有命中，请查看规则配置是否正确。

```
// 获取陆游规则
2022-11-04 10:39:52,528 [DEBUG] [77705df9691e3218][cloud][router]get router rule,serviceI
d:reviews.allsite.alipay.net:9080@Http1,ok:true,rule:{true [{20001772 true true [] 0xc003
c37f00}]}
// 当前 service id 命中规则
2022-11-04 10:39:52,528 [DEBUG] [77705df9691e3218][cloud][router] hit the service gov rou
ter handler,status:0
```

3. 过滤 default 日志 loadBalance 信息，查看负载均衡策略。已命中的节点，请查看命中的节点是否具备当前规则信息。

```
2022-04-20 16:21:15,634 [DEBUG] [][Router][loadBalance][GovernLoadBalance], originHosts =
[10.0.207.212]
2022-04-20 16:21:15,634 [DEBUG] [][Router][loadBalance][GovernLoadBalance], service =
2022-04-20 16:21:15,634 [DEBUG] [upstream] [cluster manager] clusterSnapshot.loadbalancer
.ChooseHost result is 10.0.207.212:15006, cluster name = python-server.cmbc:8888@Http1
2022-04-20 16:21:15,634 [DEBUG] [1921,0a00c8031650442875613180739] [proxy] [upstream] app
end headers: GET /hello HTTP/1.1
```

服务降级不生效

1. 确认 MOSN 是否收到控制面下发的指令。

   若未收到，则排查环境变量中 APPNAME 是否和控制台配置不一致。

```
grep ${控制台下发DataID} drm.access.log
```

   此场景下需要开启 downgrade 文件日志 debug 日志级别进行排查。

```
curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/j
son' -d'{"log_level":"DEBUG","log_path":"logs/mosn/downgrade.log"}'
```

2. 查看日志 downgrade.log 请求参数匹配，排查 requirement 日志中是否控制台参数配置错误。

```
842806:2022-11-04 11:40:35,726 [DEBUG] [10964446,6458fd5216675332357262272 49] [requiremen
t][Match] equal requirement:&{destination.service.name SYSTEM details.allsite.alipay.net:
9080@Http1},actualValue:127.0.0.1:9080@Http1
842910:2022-11-04 11:40:35,731 [DEBUG] [10964448,ec8ed46d99c2ea0a] [requirement][Match] e
qual requirement:&{destination.service.name SYSTEM details.allsite.alipay.net:9080@Http1}
,actualValue:details.allsite.alipay.net:9080@Http1
842971:2022-11-04 11:40:35,737 [DEBUG] [10964449,ec8ed46d99c2ea0a] [requirement][Match] e
qual requirement:&{destination.service.name SYSTEM details.allsite.alipay.net:9080@Http1}
,actualValue:reviews.allsite.alipay.net:9080@Http1
```

3. 查看日志 `downgrade.log`，如有有请求命中纪录日志，则表示降级规则命中。

```
// 命中规则 & 观察者模式
2022-11-04 11:25:30,730 [INFO] [6a85a896224af0e8] [downgradeFilter] downgrade rule is act
ive and trafficConditions matched, but mode is not reject
// 命中规则 & 拒绝
2022-11-04 11:25:30,730 [INFO] [6a85a896224af0e8][downgradeFilter] downgrade rule is acti
ve and hijack response
```

故障隔离不生效

1. 确认 MOSN 是否收到控制面下发的指令。

   若未收到，则排查环境变量中 APPNAME 是否和控制台配置不一致。

   ```
   grep ${控制台下发DataID} drm.access.log
   ```

   此场景下需要开启 fault_tolerance 文件日志 debug 日志级别进行排查。

   ```
   curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/j
   son' -d'{"log_level":"DEBUG","log_path":"./logs/mosn/fault_tolerance.log"}'
   ```

2. 查看日志 downgrade.log 请求参数匹配，排查 requirement 日志中是否控制台参数配置错误。

   ```
   842806:2022-11-04 11:40:35,726 [DEBUG] [10964446,6458fd5216675332357262272 49] [requiremen
   t][Match] equal requirement:&{destination.service.name SYSTEM details.allsite.alipay.net:
   9080@Http1},actualValue:127.0.0.1:9080@Http1
   842910:2022-11-04 11:40:35,731 [DEBUG] [10964448,ec8ed46d99c2ea0a] [requirement][Match] e
   qual requirement:&{destination.service.name SYSTEM details.allsite.alipay.net:9080@Http1}
   ,actualValue:details.allsite.alipay.net:9080@Http1
   842971:2022-11-04 11:40:35,737 [DEBUG] [10964449,ec8ed46d99c2ea0a] [requirement][Match] e
   qual requirement:&{destination.service.name SYSTEM details.allsite.alipay.net:9080@Http1}
   ,actualValue:reviews.allsite.alipay.net:9080@Http1
   ```

3. 查看日志 `fault_tolerance.log` ，如有有请求命中纪录日志，则表示隔离规则命中。

   ```
   2022-11-04 13:27:18,845 [DEBUG] [705cc6e5a3d20986,][Tolerance][FaultToleranceChooseFilter
   ] match traffic conditions or without traffic conditions
   ```

故障注入不生效

1. 确认 MOSN 是否收到控制面下发的指令。

   若未收到，则排查环境变量中 APPNAME 是否和控制台配置不一致。

   ```
   grep ${控制台下发DataID} drm.access.log
   ```

   此场景下需要开启 fault_inject 文件日志 debug 日志级别进行排查。

   ```
   curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/j
   son' -d'{"log_level":"DEBUG","log_path":"logs/mosn/fault_inject.log"}'
   ```

2. 查看日志 `fault_inject.log` 请求参数匹配，排查 requirement 日志中是否控制台参数配置错误。

```
2022-11-04 13:35:10,253 [DEBUG] [IsMatch]leftVal:11.158.125.203,op:EQUAL,rightVal:[11.158
.125.224]
2022-11-04 13:35:10,253 [DEBUG] [match] resource:reviews.allsite.alipay.net:9080@Http1:i:
r,key:reviews.allsite.alipay.*:i:r,conditions:[]
2022-11-04 13:35:10,253 [DEBUG] [match] resource:reviews.allsite.alipay.net:9080@Http1:i:
r,key:*.*:i:r,conditions:[]
```

3. 查看日志 `fault_inject.log`，如有有请求命中纪录日志，则表示故障注入规则命中。

```
fault_inject.log.2022-11-02:31024:2022-11-02 15:50:37,546 [INFO] [0c9ffb112a16e4c3][hit]
resourceName:reviews.allsite.alipay.net:9080@Http1:i:r#334b674a, headers:GET /reviews/0 H
TTP/1.1
```

安全加密

排查思路

1. 从 `security.access.log` 中排查下发配置是否正常， 配置是否命中当前节点。
   - 若没有收到配置，说明控制台配置的规则错误。
   - 若下发正确，进入下一步。
2. 从 `default.log` 中判断逻辑执行是否正常，是否有 error or warn。
   - 常见错误，无法连接 nodeagent，报 /var/run/cloudmesh 拨号失败。
   - 是否正确配置 sds 路径?
   - 更新 listener 是否有报错?
   - 镜像是否支持国密?
   - 协议是否正确配置?

   若都没问题， 进入下一步。
3. 若发现无法连接 nodeagent，查看 nodeagent 日志，排查 nodeagent 错误。
   - nodeagent 是否能正常连接 citadel。
   - citadel 状态是否正常。
   - citadel 地址是否配错。
   - citadel 命名空间是否正确配置。
4. 检查抓包命令是否正确。

排查线索

- 服务端（default.log）正常日志:

```
2022-03-29 19:44:08,764 [INFO] [drm-gray] gray data value OnChange,appName=python-client,
dataId=Alipay.python-client:name=com.alipay.sofa.middleware.mesh.drm.security.TLSRules,ve
rsion=3.0@DRM
2022-03-29 19:44:08,764 [INFO] [mtls] [sds provider] add a new sds provider default
2022-03-29 19:44:08,765 [INFO] [sds][subscribe] init sds stream client success
2022-03-29 19:44:08,960 [INFO] [mtls] [sds provider] provider default receive a cerificat
e set
2022-03-29 19:44:08,960 [INFO] [mtls] [sds provider] provider ROOTCA receive a validation
set
2022-03-29 19:44:08,961 [INFO] [mtls] [sds] update tls context success
2022-03-29 19:44:08,961 [INFO] update cluster tls config, do not need to update tls state
```

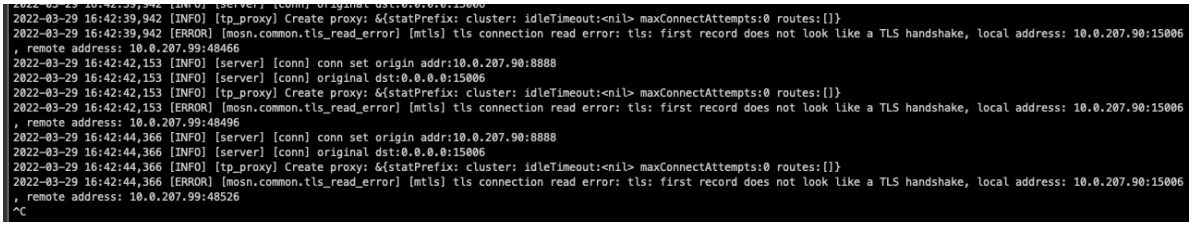
- 可用于判断策略是否下发。
- 可用于是否判断是否拉取到了证书。
- 可用区判断是否更新了 TLS 配置。

- 客户端正常日志：

```
2022-06-08 10:49:58,719 [INFO] [cluster] [primaryCluster] [UpdateHosts] cluster com.alipa
y.sofa.ms.service.HelloService@dubbo update hosts: 1
2022-06-08 10:49:58,719 [INFO] [upstream] [host set] update host, final host total: 1
2022-06-08 10:49:58,719 [INFO] [cluster] [primaryCluster] [UpdateHosts] cluster com.alipa
y.sofa.ms.service.BenchmarkService@dubbo update hosts: 1
2022-06-08 10:50:00,178 [INFO] [upstream] [cluster manager] 172.19.12.31:30800 tls state
changed
```

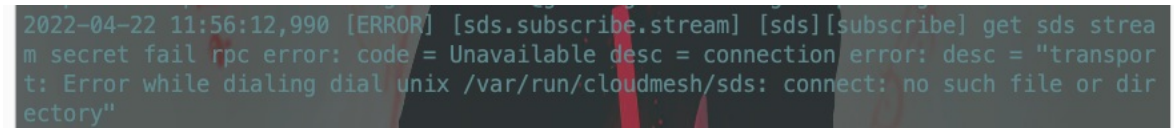
报错排查

- 透明劫持报错



透明劫持不支持安全加密。

- 找不到 `/var/run/cloudmesh/sds` 文件



`/var/run/cloudmesh/sds` 为 citadel-agent Unix domain 文件，缺少时，会导致无法获取证书。该文件是 citadel-agent 创建时创建的。缺失时，查看 citadel-agent 是否存在(`kubectl get ds -A |grep nodeagent`):

- 不存在可能是有人关闭了集群，导致资源被删除了，重新开通集群即可。
- citadel-agent 存在时，依然无法连接，则可能是 citadel-agent 未正常运行， 常见原因为 citadel mcp 出现错误， 需排查 citadel-agent 启动失败原因。

服务鉴权不生效

1. MOSN 是否收到控制面下发的指令？

   若未收到，则排查环境变量中 APPNAME 是否和控制台配置不一致。

   ```
   grep ${控制台下发DataID} drm.access.log
   ```

   此场景下需要开启 default 文件日志 debug 日志级别进行排查。

   ```
   curl -X POST http://127.0.0.1:34901/api/v1/update_loglevel -H 'Content_Type:application/json' -d'{"log_level":"DEBUG","log_path":"./logs/mosn/default.log"}'
   ```

2. 查看日志 `default.log` 请求参数匹配，排查 requirement 日志中是否控制台参数配置错误。

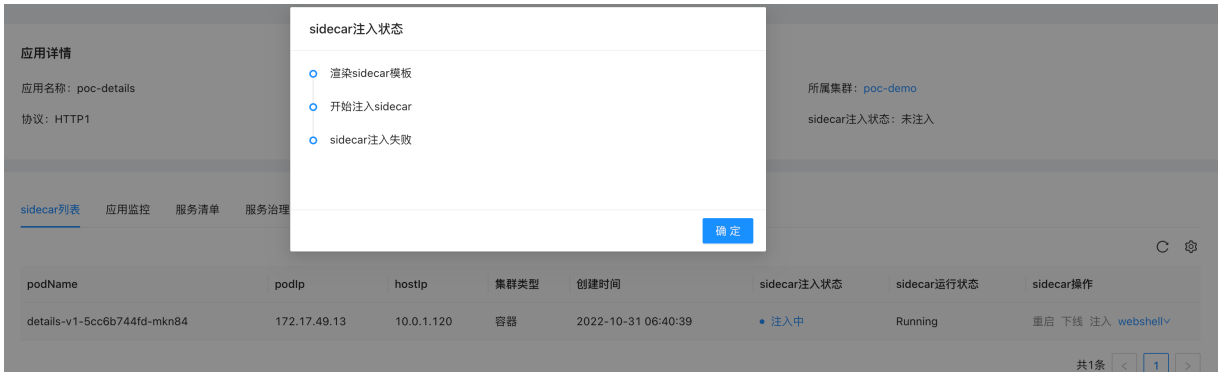
   ```
   // 请求的服务名字
   2022-11-04 14:07:44,911 [DEBUG] [175d2722c092cbd9,][rbac] got svc from variable, svc=reviews.allsite.alipay.net:9080@Http1
   // 请求协议
   2022-11-04 14:07:44,911 [DEBUG] [175d2722c092cbd9,][rbac] svcType=Http1
   // 请求的服务名字
   2022-11-04 14:07:44,911 [DEBUG] [175d2722c092cbd9,][rbac] got current service name, svc=reviews.allsite.alipay.net:9080@Http1
   // 请求详细信息
   2022-11-04 14:07:44,911 [DEBUG] [175d2722c092cbd9,][rbac] request header=map[Accept:*/* Accept-Encoding:gzip, deflate Connection:keep-alive Host:reviews.allsite.alipay.net:9080 User-Agent:python-requests/2.21.0 X-B3-Sampled:1 X-B3-Spanid:175d2722c092cbd9 X-B3-Traceid:175d2722c092cbd9 service:reviews.allsite.alipay.net:9080@Http1 sofa_head_method_name:GET sofa_head_target_app:poc-reviews]
   ```

3. 查看日志 `downgrade.log`，如有有请求命中纪录日志，则表示服务鉴权规则命中。

   ```
   default.log:2022-11-04 14:05:28,208 [INFO] [a911a2037b340fe1,][rbac] request has ben reject
   ```

服务注入失败

容器注入失败



1. 查看 dsr pod 错误日志。

   若有错误日志，根据错误提示解决问题。

```
[root@dsrconsole-0 /home/admin/logs/dsrconsole]
# tail -f common-error.log
        at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:163)
        at org.springframework.transaction.interceptor.TransactionAspectSupport.invokeWithinTransaction(TransactionAspectSupport.java:295)
        at org.springframework.transaction.interceptor.TransactionInterceptor.invoke(TransactionInterceptor.java:98)
        at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:186)
        at org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercept(CglibAopProxy.java:689)
        at com.alipay.antcloud.dsrconsole.core.service.impl.AuditLogServiceImpl$$EnhancerBySpringCGLIB$$9c66f811.record(<generated>)
        at com.alipay.antcloud.dsrconsole.endpoint.filter.AuditLogFilter.lambda$filter$0(AuditLogFilter.java:186)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:853)
```

2. 查看 pod 所在的 namespace 是否带有 `sidecar-inject: enabled` 标志。

   若不携带，则排查 K8s 审计日志是否被其他组件篡改。如有篡改，可以通过重新开通集群暂时解决。

```
whj at B-54N7MD6R-1941 in ~/workflow/aliyun_dev
$ k9s get ns bookinfo -oyaml
apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: "2022-08-26T12:10:19Z"
  labels:
    kubernetes.io/metadata.name: bookinfo
    sidecar-inject: enabled
  name: bookinfo
  resourceVersion: "1172289"
  uid: a4e52031-7afa-4dd2-9134-c6f8f58473e0
spec:
  finalizers:
  - kubernetes
status:
  phase: Active
```

3. 查看 MutatingWebhookConfiguration 中的 sidecar-operator-webhook 是否存在。

   若不存在，则排查 K8s 审计日志是否被其他组件删除。如果被删除，可以通过重新开通集群暂时解决。

```
whj at B-54N7MD6R-1941 in ~/workflow/aliyun_dev
$ k9s get MutatingWebhookConfiguration  sidecar-operator-webhook -oyaml
apiVersion: admissionregistration.k8s.io/v1
kind: MutatingWebhookConfiguration
metadata:
  creationTimestamp: "2022-08-29T03:04:33Z"
  generation: 34
  name: sidecar-operator-webhook
  resourceVersion: "27170016"
  uid: be7e529b-e96a-40c9-8708-e038d3004e36
webhooks:
- admissionReviewVersions:
  - v1
  - v1beta1
  clientConfig:
    caBundle: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM4RENDQWRpZ0F3SUJBZ0lRR0VDOWdGdUxnVDhLenBwOVBz
XdaWEpoZEc5eU1JSUJJakFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBCk1JSUJDZ0tDQVFFQTMrdTJiMHhpRlFEYmFrTTU3V3M3Qkh
SlV0Ykp6bFVXNEhTYzRCYXdBemJYSmpBNlg3OEg5N3RheWZDcTRFclpNVzFPCnZPM3pxdXdubERTZEIzRlBMUzEzdFQ5dzlVaTZYU2
zWFROWjI0Z3RlQmhkVjZ5NXl3SURBUUFCbzBBd1BqQU9CZ05WSFE4QkFmOEVCQU1DCkJhQXdEQVlEVlIwVEFRSC9CQUl3QURBZUJnT
9WVUJRRUJlWDdlOCt3SG5EM3B1UjMwRHh3eStHYzF3aVpGMTRiSkR1L0t2Tm5RMzI1djlYCnErRDBkcExIM1ZHM3BodjhseVkxckJG
WlHcWtuaVZRV29XWmtTbHhTelFwYzduckhUei9pWlgrRmMyOWxKZzc5ckY5VVlzUkc2SGxmUVFHClUzRzBQaFI1WTJ4bFl0am5PRFB
    url: https ,                  /inject
  failurePolicy: Fail
  matchPolicy: Exact
  name: inject.k8s.alipay.com
  namespaceSelector: {}
  objectSelector: {}
  reinvocationPolicy: Never
  rules:
  - apiGroups:
    - ""
    apiVersions:
    - v1
    operations:
    - CREATE
    resources:
    - pods
    scope: '*'
  sideEffects: None
  timeoutSeconds: 30
```

4. 修改日志 MutatingWebhookConfiguration 中的 sidecar-operator-webhook 字段为 `failurePolicy: Fail`，然后重新执行注入。

```
$ k9s get MutatingWebhookConfiguration  sidecar-operator-webhook -oyaml
apiVersion: admissionregistration.k8s.io/v1
kind: MutatingWebhookConfiguration
metadata:
  creationTimestamp: "2022-08-29T03:04:33Z"
  generation: 34
  name: sidecar-operator-webhook
  resourceVersion: "27170016"
  uid: be7e529b-e96a-40c9-8708-e038d3004e36
webhooks:
- admissionReviewVersions:
  - v1
  - v1beta1
  clientConfig:
    caBundle: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM4RENDQWRpZ0F3SUJ
XdaWEpoZEc5eU1JSUJJakFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBCk1JSUJDZ0tDQVFFQT
SlV0Ykp6bFVXNEhTYzRCYXdBemJYSmpBNlg3OEg5N3RheWZDcTRFclpNVzFPCnZPM3pxdXdub
zWFROWjI0Z3RlQmhkVjZ5NXl3SURBUUFCbzBBd1BqQU9CZ05WSFE4QkFmOEVCQU1DCkJhQXdE
9WVUJRRUJlWDdlOCt3SG5EM3B1UjMwRHh3eStHYzF3aVpGMTRiSkR1L0t2Tm5RMzI1djlYCnE
WlHcWtuaVZRV29XWmtTbHhTelFwYzduckhUei9pWlgrRmMyOWxKZzc5ckY5VVlzUkc2SGxmUV
    url: https://[redacted]/inject
  failurePolicy: Fail
  matchPolicy: Exact
  name: inject.k8s.alipay.com
  namespaceSelector: {}
  objectSelector: {}
  reinvocationPolicy: Never
  rules:
  - apiGroups:
    - ""
    apiVersions:
    - v1
    operations:
    - CREATE
    resources:
    - pods
    scope: '*'
  sideEffects: None
  timeoutSeconds: 30
```

5. 查看 pod 重启的 deployment 或者 sts 报错信息，根据提示处理问题。

```
status:
  availableReplicas: 1
  conditions:
  - lastTransitionTime: "2022-09-20T09:16:46Z"
    lastUpdateTime: "2022-09-20T09:16:46Z"
    message: Deployment has minimum availability.
    reason: MinimumReplicasAvailable
    status: "True"
    type: Available
  - lastTransitionTime: "2022-08-29T02:32:25Z"
    lastUpdateTime: "2022-10-31T07:02:41Z"
    message: Created new replica set "details-v1-bb9c59d6d"
    reason: NewReplicaSetCreated
    status: "True"
    type: Progressing
  - lastTransitionTime: "2022-10-31T07:02:44Z"
    lastUpdateTime: "2022-10-31T07:02:44Z"
    message: 'Internal error occurred: failed calling webhook "inject.k8s.alipay.com":
      failed to call webhook: Post "https://172.      .141/inject?timeout=30s": dial
      tcp      1      141:443: connect: no route to host'
    reason: FailedCreate
    status: "True"
    type: ReplicaFailure
  observedGeneration: 8
  readyReplicas: 1
  replicas: 1
  unavailableReplicas: 1
```

- 如果报错是 IP 问题，排查 IP 是不是不通或者不存在。可以通过手动签证解决。
- 如果报错是域名无法解析，修正 K8s 域名解析或者域名修正为 Pod IP 后手动重新签证解决。
- 若未发现上述问题，请查看 sidecar-operator 日志，然后根据提示，修正遇到的问题。

虚拟机注入失败

1. 查看 dsr Pod 错误日志。

   若有错误日志，根据错误提示解决问题。

```
[root@dsrconsole-0 /home/admin/logs/dsrconsole]
# tail -f common-error.log
        at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:163)
        at org.springframework.transaction.interceptor.TransactionAspectSupport.invokeWithinTransaction(TransactionAspectSupport.java:295)
        at org.springframework.transaction.interceptor.TransactionInterceptor.invoke(TransactionInterceptor.java:98)
        at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:186)
        at org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercept(CglibAopProxy.java:689)
        at com.alipay.antcloud.dsrconsole.core.service.impl.AuditLogServiceImpl$$EnhancerBySpringCGLIB$$9c66f811.record(<generated>)
        at com.alipay.antcloud.dsrconsole.endpoint.filter.AuditLogFilter.lambda$filter$0(AuditLogFilter.java:186)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:853)
```

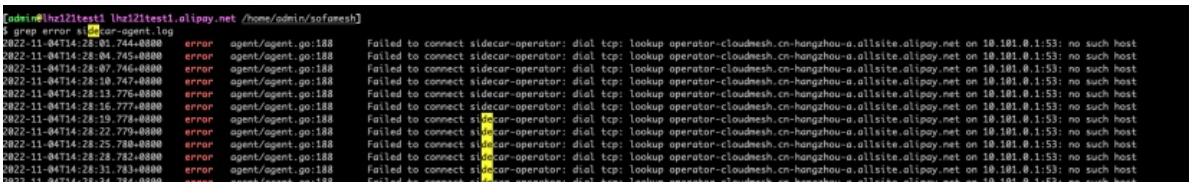
2. 查看 sidecar-agent 部署情况。

```
ps aux |grep sidecar-agent
```

   若不存在，请查看根目录 `sofamesh.log` 日志。

```
admin@lhz121test1:~ (kubectl)   ⌘1   ~ (-zsh)
[INFO] Using /home/admin/sofamesh as the sofa-mesh working directory.
[INFO] Downloading sofa-mesh installation artifacts...
[INFO] Setting up runtime parameters...
[INFO] Registry Ip is.          6
[INFO] Starting sidecar-agent...
[INFO] write keepalive cron use crontab -e
^[[32m[INFO] Ready to go!^[[0m
~
```

3. 查看 sidecar-agent 运行日志 `/home/admin/sofamesh/sidecar-agent.log`，如果有报错请跟进报错解决。



4. 查看 MOSN 进程是否启动，若不存在请排查 sidecar-agent 日志是否安装失败。

5. 查看 MOSN 日志是否启动其中成功，若报错请根据日志提示解决问题。

```
[admin@lhz121test1 lhz121test1.alipay.net /home/admin/logs/mosn]
$ tail -f out.log
2022-11-04 14:20:49,760
2022-11-04 14:21:49,760
GC forced
gc 10914 @1309085.630s 0%: 0.077+14+0.005 ms clock, 0.15+0.69/14/32+0.010 ms cpu, 32->32->27 MB, 54 MB goal, 4 P
2022-11-04 14:22:49,761
2022-11-04 14:23:49,763
GC forced
gc 10915 @1309205.657s 0%: 0.088+19+0.031 ms clock, 0.17+0/18/35+0.063 ms cpu, 31->31->27 MB, 54 MB goal, 4 P
2022-11-04 14:24:49,763
2022-11-04 14:25:11,452 [INFO] [actuator] publish mosn booting event
```

注入失败问题排查示例

- Sidecar 注入失败（控制台显示注入成功，但未注册）。

  开通集群需要注册 webhook，API Server 监听到 Pod 时，就会去调用 sidecar-operator 注入MOSN了。但是如果 webhook 注册的有问题，常见的是 Sidecar-Operator 地址没有正确配置，特别是手动部署 Mesh 产品时，就会出现 Sidecar 无法注入。

  i. 查看 api-server 日志， 查看是否有错误。

```
or converting to "v1" in scheme "k8s.io/kubernetes/pkg/api/legacyscheme/scheme.go:30"
I0411 20:29:43.062073       1 client.go:360] parsed scheme: "passthrough"
I0411 20:29:43.062098       1 passthrough.go:48] ccResolverWrapper: sending update to cc: {[{https://10.0.13.245:2379  <nil> 0 <nil>}] <n
il> <nil>}
I0411 20:29:43.062105       1 clientconn.go:948] ClientConn switching balancer to "pick_first"
I0411 20:29:51.406328       1 client.go:360] parsed scheme: "passthrough"
I0411 20:29:51.406365       1 passthrough.go:48] ccResolverWrapper: sending update to cc: {[{https://10.0.13.242:2379  <nil> 0 <nil>}] <n
il> <nil>}
I0411 20:29:51.406374       1 clientconn.go:948] ClientConn switching balancer to "pick_first"
I0411 20:29:57.056725       1 client.go:360] parsed scheme: "passthrough"
I0411 20:29:57.056749       1 passthrough.go:48] ccResolverWrapper: sending update to cc: {[{https://10.0.13.244:2379  <nil> 0 <nil>}] <n
il> <nil>}
I0411 20:29:57.056757       1 clientconn.go:948] ClientConn switching balancer to "pick_first"
W0411 20:30:11.868766       1 dispatcher.go:171] Failed calling webhook, failing open inject.k8s.alipay.com: failed calling webhook "inje
ct.k8s.alipay.com": Post "https://sidecar-operator-service.sofamesh/inject?timeout=30s": dial tcp: lookup sidecar-operator-service.sofame
sh on 100.100.2.138:53: no such host
E0411 20:30:11.868790       1 dispatcher.go:172] failed calling webhook "inject.k8s.alipay.com": Post "https://sidecar-operator-service.s
ofamesh/inject?timeout=30s": dial tcp: lookup sidecar-operator-service.sofamesh on 100.100.2.138:53: no such host
I0411 20:30:22.301804       1 client.go:360] parsed scheme: "passthrough"
I0411 20:30:22.301834       1 passthrough.go:48] ccResolverWrapper: sending update to cc: {[{https://10.0.13.245:2379  <nil> 0 <nil>}] <n
il> <nil>}
I0411 20:30:22.301842       1 clientconn.go:948] ClientConn switching balancer to "pick_first"
I0411 20:30:26.524398       1 client.go:360] parsed scheme: "passthrough"
I0411 20:30:26.524428       1 passthrough.go:48] ccResolverWrapper: sending update to cc: {[{https://10.0.13.242:2379  <nil> 0 <nil>}] <n
il> <nil>}
I0411 20:30:26.524437       1 clientconn.go:948] ClientConn switching balancer to "pick_first"
I0411 20:30:35.032705       1 client.go:360] parsed scheme: "passthrough"
I0411 20:30:35.032733       1 passthrough.go:48] ccResolverWrapper: sending update to cc: {[{https://10.0.13.244:2379  <nil> 0 <nil>}] <n
il> <nil>}
```

  如果有报错，确实是地址配置错误，查看 webhook 地址是不是配错了。

  ii. 查看 webhook 配置。

```
kubectl get MutatingWebhookConfiguration
```

- Sidecar 注入失败，出现如下报错：

这种情况一般是控制面初始化 webhook 时，使用的域名或者 IP 和现在注入的集群的 webhook 用的域名或者 IP 不一致，修改为一致的域名或 IP 即可。

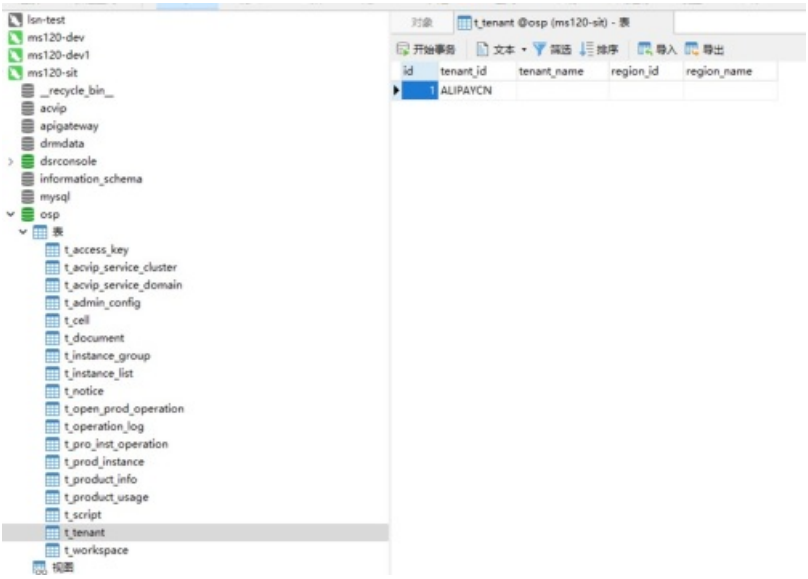
3.5. 常见问题解决方案

3.5.1. mock 场景下，OSP 如何创建新租户

增加租户

修改 OSP 的 t_tenant 表，增加新租户：

```
INSERT INTO `t_tenant` (`tenant_id`, `tenant_name`, `region_id`, `region_name`) VALUES ('AN
TCLOUD', '', '', '');
```



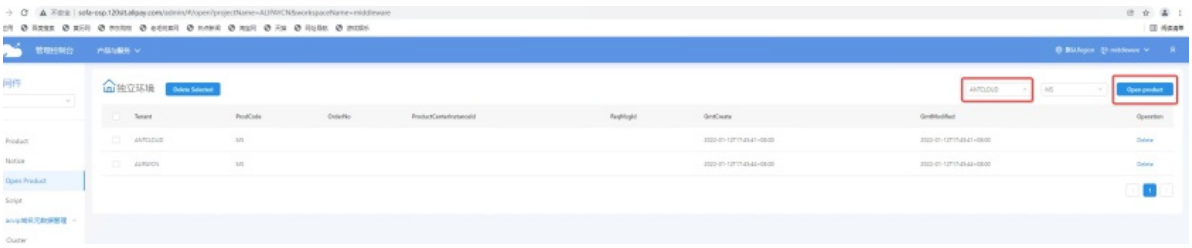
增加 workspace

OSP 的 t_workspace 表，增加新租户下的 workspace：

```
INSERT INTO `t_workspace` (`tenant_id`, `workspace_id`, `workspace_name`, `operator`, `gmt_
create`, `gmt_modified`) VALUES ('ANTCLOUD', 'sit', '', '', '2017-06-14 13:41:35', NULL);
```

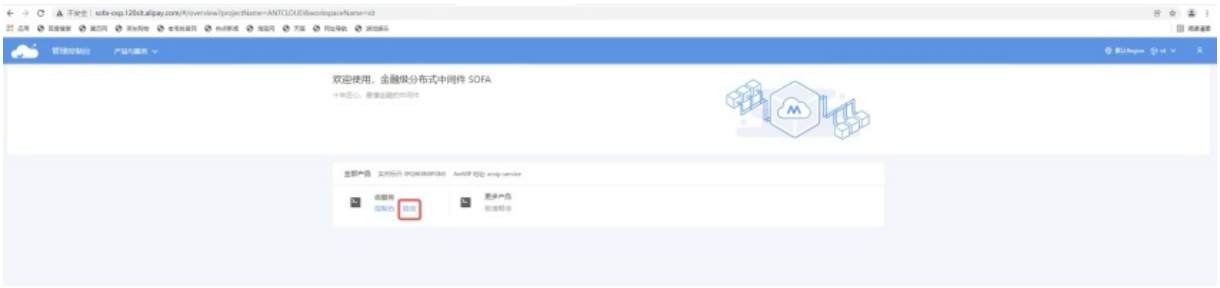
租户开通产品

1. 打开 OSP 页面，切换到 /admin/#/cluster 的 path。
2. 进入 OSP 管理页面，选择租户，点击开通产品。



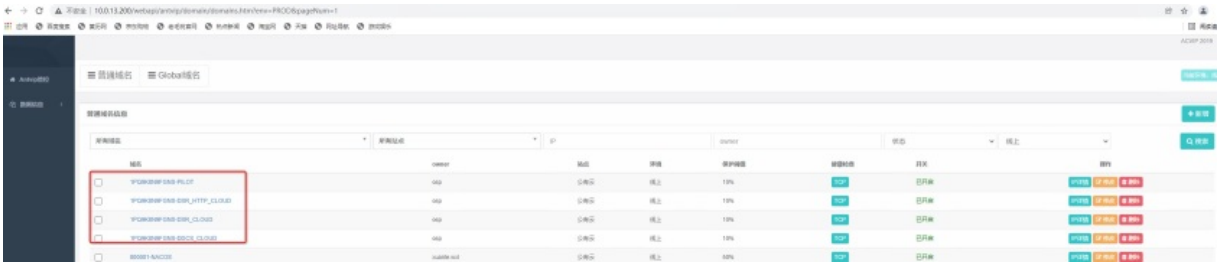
worksapce 启动微服务

切换到对应的租户及 workspace，进入 OSP 的普通页面，点击启动。



确认 ACVIP 上有数据

打开 ACVIP 控制台，查看是否有对应 instanceId 的 domain。



3.5.2. MOSN 常用运维命令

看看 MOSN 的运行状态

```
curl http://localhost:34901/api/v1/container_state
{"container_name":"mosn-sidecar-container","state":1}
```

state = 1 表示正常。如果您想进一步看每个模块的状态，可以使用如下命令：

```
curl http://localhost:34901/api/v1/health
{"health":true,"components":[{"name":"antvip","health":true,"message":""},{"name":"registry
","health":true,"message":""},{"name":"drm","health":true,"message":""},{"name":"zoneclient
","health":true,"message":""}]}
```

查看当前部署的版本

```
curl http://localhost:34901/api/v1/version
```

修改日志级别

```
curl  -d '{"log_level":"DEBUG","log_path":"./logs/mosn/default.log"}'   127.0.0.1:34901/api
/v1/update_loglevel
```

可用的日志级别如下：

- FATAL：输出会导致应用程序退出的严重错误事件日志。
- ERROR：输出不影响系统继续运行的错误和异常信息日志。
- WARN：删除会出现潜在错误的信息。部分信息可能不是错误信息，但需要给程序员的一些提示。
- INFO：输出一些您感兴趣的或者重要的信息，这个可以用于生产环境中输出程序运行的一些重要信息，这些信息在粗粒度级别上突出应用程序的运行过程。不能滥用，避免打印过多的日志。
- DEBUG：主要用于开发过程中打印一些运行信息。一般在调试应用程序时使用。
- TRACE：输出跟踪日志。日志消息的粒度太细，一般不会使用。

获取运行时配置

参数	说明	示例
无	获取全部运行时配置。	/api/v1/config_dump
router=${router_config_name}	获取指定的路由名称的配置。	/api/v1/config_dump? router=sofa_egress_bolt_router
allrouters	获取所有路由配置。	/api/v1/config_dump?allrouters
cluster=${cluster_name}	获取指定的 cluster 配置。	/api/v1/config_dump? cluster=local_bolt_service
allclusters	获取所有 clusters 配置。	/api/v1/config_dump?allclusters
mosnconfig	获取静态配置，不包含详细的路由 cluster 信息。	/api/v1/config_dump?mosnconfig
listener=${listener_name}	获取指定 Listener 配置。	/api/v1/config_dump? listener=ingress_sofa_xfire
alllisteners	获取所有 Listener 的配置。	/api/v1/config_dump?alllisteners

获取当前 Feature

参数	说明	示例

无	获取所有 Features 状态。	/api/v1/features
key	获取指定的 Feature 状态，只支持一个 key。	/api/v1/features?key=1

获取环境变量

参数	说明	示例
key	key为环境变量名，支持多个。	/api/v1/env?key=1&key=2

当前 MOSN 内存中处于运行态的 RouteRule

```
curl http://localhost:34901/api/v1/routerule/get
```

抓包命令

```
#INGRESS
tcpdump -i any -n port 30800  -vvv -A
```

查看请求链路

当请求不通或者服务治理不生效时，通过日志来排查错误。打开日志的 Debug 级别后，您可以通过 traceId 查看整个请求路径，把每个环节日志捞出来，分析有哪些报错。

```
cd /home/admin/logs/mosn
//xxxx 为 traceId， 即可查请求状况
grep "xxxxxx" -i -r ./*
```

3.5.3. 云游 Sidecar-Operator 手动签证书方案

操作步骤

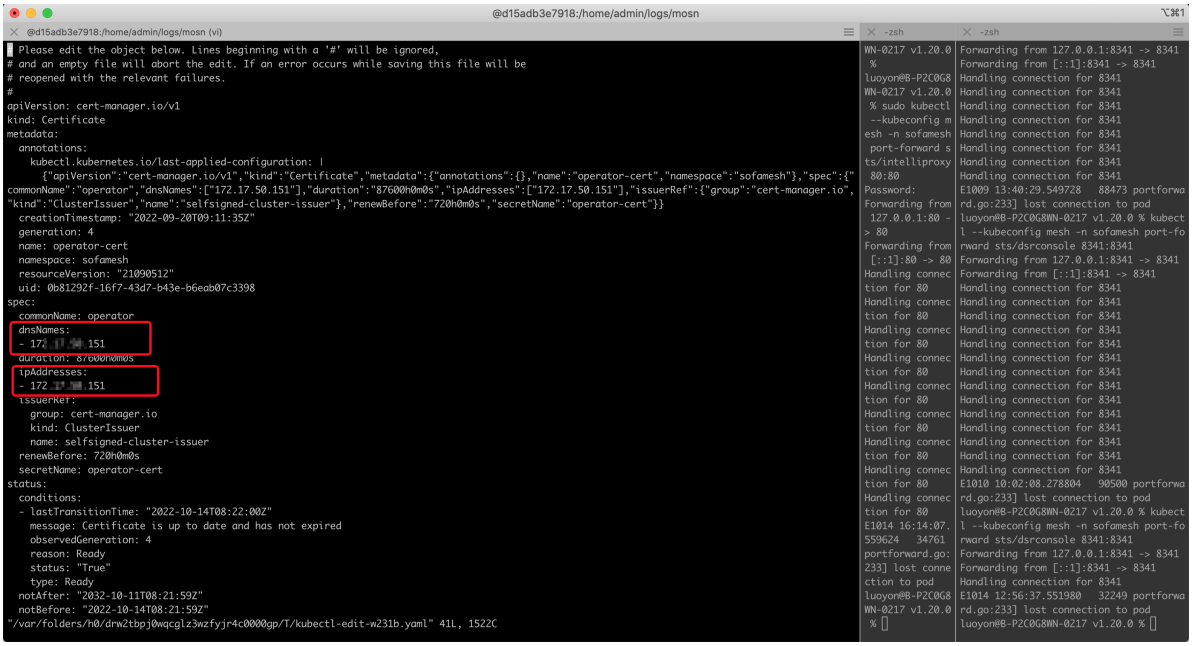
1. 执行以下命令，找到对应的 Certificate。

```
kubectl -n ${namespace} get Certificate
```

```
luoyon@B-P2C0G8WN-0217 v1.20.0 %
luoyon@B-P2C0G8WN-0217 v1.20.0 %
luoyon@B-P2C0G8WN-0217 v1.20.0 % kubectl --kubeconfig mesh -n sofamesh get Certificate
NAME                          READY   SECRET                                  AGE
jaeger-operator-serving-cert  True    jaeger-operator-webhook-server-cert     24d
operator-cert                 True    operator-cert                           23d
luoyon@B-P2C0G8WN-0217 v1.20.0 %
luoyon@B-P2C0G8WN-0217 v1.20.0 %
```

2. 执行以下命令修改签证书地址、dnsName 域名、ipAddresses 对应服务 IP 地址。

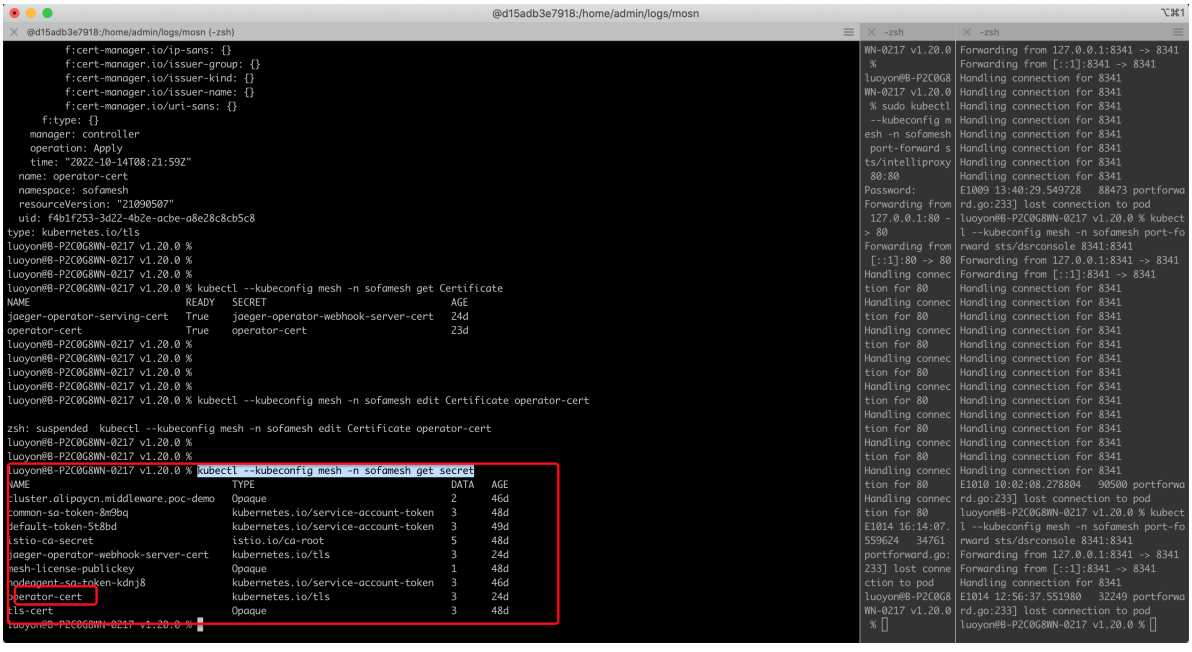
```
kubectl -n ${namespace} edit Certificate operator-cert
```



3. 执行以下命令，找到对应的 secret。

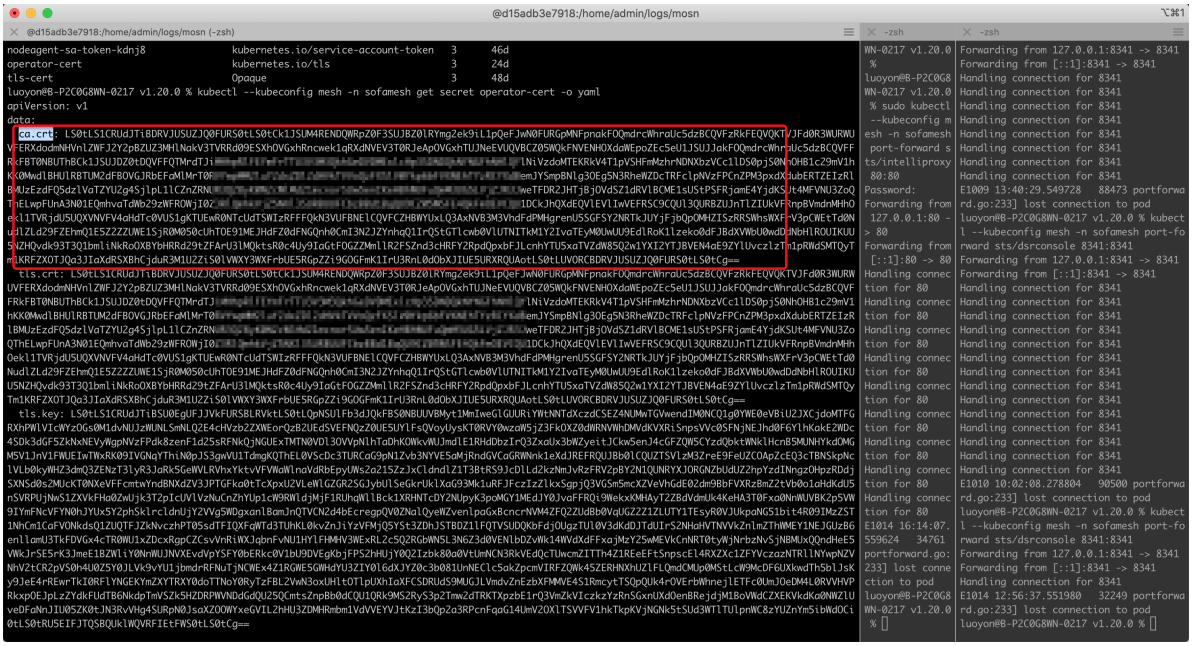
```
kubectl -n ${namespace} get secret
```

name 与 Certificate 中的 secretName 一致。



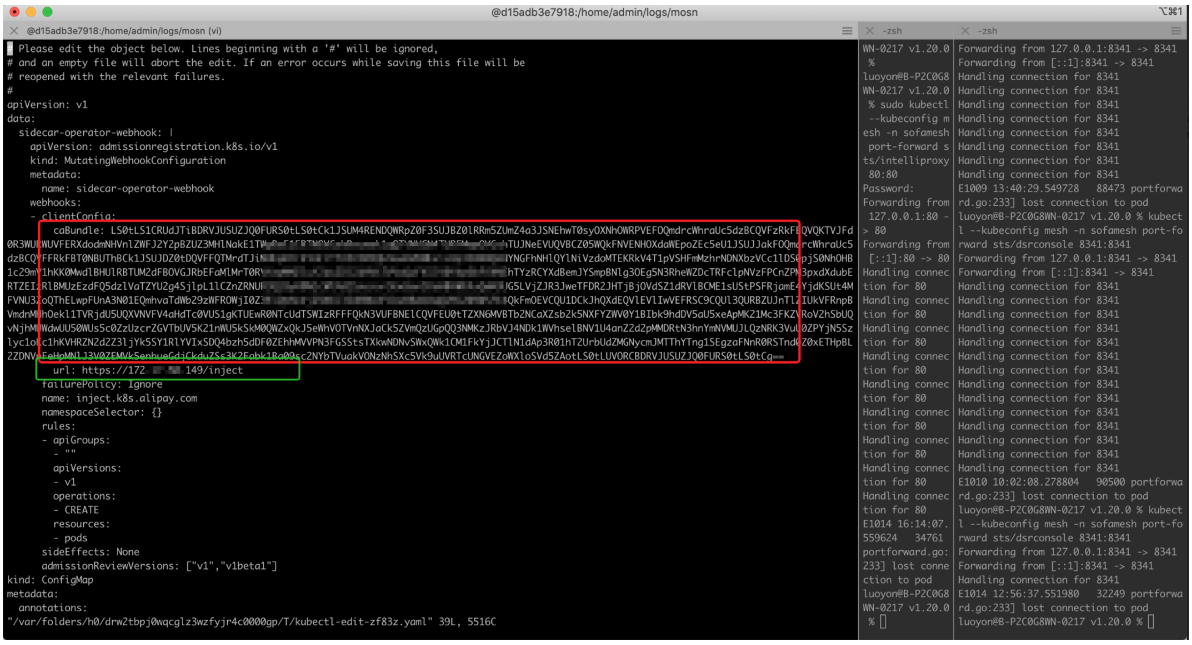
4. 执行以下命令，拷贝 `ca.crt` 至 `cm mesh-webhooks` 。

```
kubectl -n ${namespace} get secret ${secretName} -o yaml
```



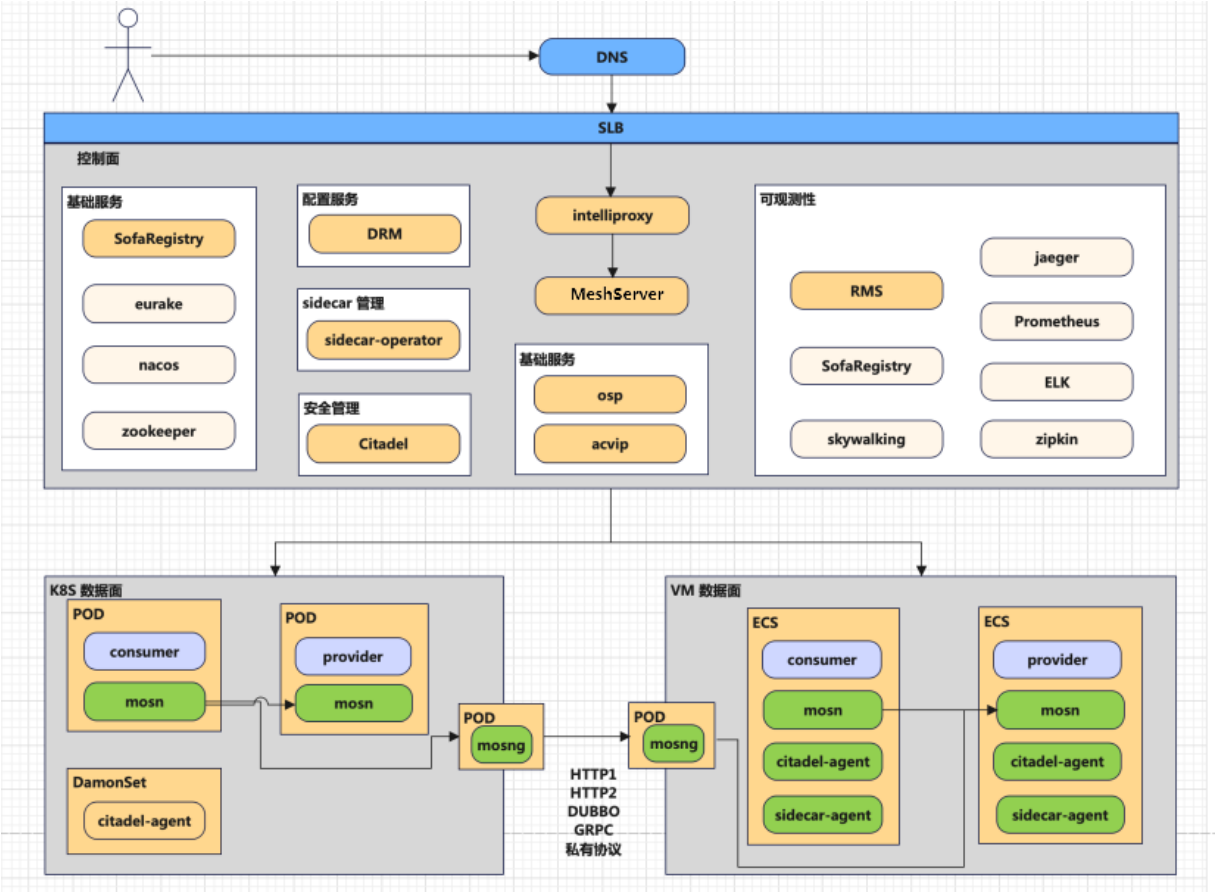
5. 执行以下命令，用 `ca.crt` 替换 `caBundle` ，修正 URI 的地址信息。

```
kubectl -n ${namespace} edit cm mesh-webhooks
```



3.6. 控制面组件不可用会产生哪些影响

Mesh 及 Mesh 依赖的系统拓扑如下：



原厂组件

类别	组件名称	组件功能	不可用影响
基础组件	Intelliproxy	智能服务网关，用来转发路由请求，统一权限和 cookie 等切面管理，提供统一访问入口 。	浏览器无法访问 Mesh 控制台。 影响范围如下： • 不影响新老 Sidecar。 • 不影响新应用启动。
基础组件	OSP	运维支持系统，主要提供和管理中台的元数据服务，是被其他应用所依赖的基础服务系统。如tenant、workspace 管理中心、用户等。	控制台鉴权失败，产生不可用的报错。影响范围如下： • 不影响新老 Sidecar。 • 不影响新应用启动。

基础组件	ACVIP	虚拟域名寻址服务，提供中间件服务端寻址的 facade，并且可以进行不同纬度的服务端灰度。	ACVIP 保存了 DRM、SessionServer、Pilot、GateWay 的地址，不可用时会导致 MOSN 无法获取上述组件的服务地址。 影响范围如下： • 不影响已有 Sidecar。 • 新增 Sidecar 无法与 Pilot、注册中心、DRM 等枢组件建连。 • 应用可以带 Sidecar 启动，但是启动了之后无法执行 pub/sub 操作，无法接收规则，相当于新应用不可用。
产品层	MeshServer	服务治理中心管控台	MeshServer 不可用。 影响范围如下： • 不影响新老 Sidecar。 • MeshServer 同时扮演 CA（Certificate Authority） 的角色。 • MeshServer 不可用会导致新的安全链路无法建连，无法通信。
配置中心	DRM	动态配置中心	DRM 配置下发不可用，导致服务限流、熔断、鉴权、降级、故障注入、故障隔离不可用。 影响范围如下： • 不影响应用启动。 • 影响规则下发。 • Sidecar 内存中已有的规则不影响，新规则无法下发。
安全管理	Citadel	安全服务端，提供 CA 和 MCP 服务。	Citadel 不可用，导致安全能力丧失。 影响范围如下： • 已有的加密链路不受影响。 • 新的链路无法加密、无法通信。
Sidecar 管理	Sidecar-Operator	Sidecar 注入服务，控制容器和虚拟机下的 Sidecar 注入。	Operator 不可用，导致 Sidecar注入失败。 影响范围如下： • 不影响应用启动。 • 不影响本身的 SDK 逻辑。

注册中心	SofaRegistry Session SofaRegistry Data SofaRegistry Meta	SOFA 注册中心	MeshServer 和 DRM、OpenAPI 之间的互相发现依靠 SOFA 注册中心，如果注册中心不可用，不影响控制面的可用性。 影响范围如下： • 不影响新老 Sidecar。 • 不影响应用启动。
MySQL/OB	数据库	控制面共用数据库	DB 不可用，导致 MeshServer、ACVIP、OSP 无法进行 DB 读写。控制台不可用。影响范围如下： • 新增 Sidecar 启动时，无法从ACVIP拉取配置，等同于ACVIP不可用。 • 已有 Sidecar 无影响。 • 新增 Sidecar 无法与 Pilot、注册中心、DRM 等中枢组件建连。 • 应用可以带 Sidecar 启动，但启动了之后无法进行 pub/sub 操作，无法 • 接收规则，相当于新应用不可用。
可观测性	RMS	业务实时监控。一整套海量日志实时分析解决方案，以日志、REST 接口、Shell 脚本等作为数据采集来源，提供设备、应用、业务等各种视角的监控能力，为线上系统可用率提供有效保障。	-

数据面集群组件

容器组件

组件	组件功能	不可用影响
filebeat（可选）	ELK 日志收集组件	日志无法采集。
MOSN	Sidecar 应用	应用无法通信。
Citadel-Agent	安全组件的 daemon set，提供证书下发和轮换服务。	安全功能将不可用。

虚拟机组件

组件	组件功能	不可用影响
Sidecar-Agent	负责接管所有 Sidecar 的生命周期。	Sidecar 无法启用。
filebeat（可选）	收集制定日志文件并传输至 kafka。	日志无法采集。
MOSN	Sidecar 应用	应用无法通信。
Citadel-Agent	安全组件的 daemon set，提供证书下发和轮换服务。	安全功能不可用。