

# SOFAStack

## 资金安全监控 部署指南

产品版本：AntStack Plus 1.11.0


文档版本：20221008

# 法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

## 商标声明

 蚂蚁集团 ANT GROUP 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

## 免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

# 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。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 <b>确定</b> 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.部署前须知	05
2.前置事项	06
3.云游 Global 解决方案	07
4.现场前置准备	09
5.云游 Local 部署	11
6.部署验证	15
7.OSS 数据源配置	22
8.常见问题	26
9.技术术语	28

# 1.部署前须知

## 版本信息

资金安全监控：1.121.0。

## 拓扑说明

标准单机房部署。

## 部署资源及规格

资源类型	资源名	规格	实例数	备注
数据库（RDS）	sofaclrisk_db	4C/8G/100GB	1	
缓存（Redis）	redis	Redis 5.0/4GB	1	
负载均衡（SLB）	sofaclriskslb	公网	1	
服务器（ECS）	sofaclrisk	4C/8G/100GB	4	<ul style="list-style-type: none"><li>离线核对使用该规格即可。</li><li>文件核对可扩容至 10 台，预计可支撑千万级别的表、文件进行小时级核对。</li></ul>
对象存储（OSS）	antfinorm-prod	100G+（视用量）	/	部署完成后配置，依赖客户 OSS。若客户没有 OSS 则无法使用文件核对能力。
大数平台（MaxCompute）	/	离线核对场景，MaxCompute 用量视用户具体数据量决定。	/	不需要配置，依赖该设施。若客户没有 MaxCompute，则无法使用离线核对能力。

## 2.前置事项

前置依赖及说明如下：

依赖类别	依赖项	依赖程度
其他产品集	身份-访问管理 / IAM	强依赖。 <ul style="list-style-type: none"><li>依赖 OpenAPI 两个参数：open_api_accessKey、open_api_accessSecret。</li><li>依赖平台租户 ID。</li></ul>
	开放平台 / OP	强依赖。通过 OP 强依赖其中的以下应用： <ul style="list-style-type: none"><li>ACM：获取用户权限。</li><li>IAM：获取操作员、租户信息。</li></ul>
	实时监控 / RealTime Monitoring Service	弱依赖。 应用监控及告警、自定义监控及告警。
	沟通 / Gotone	弱依赖。 资金安全告警邮件、钉钉通知。
其他依赖	外网	弱依赖。 资金安全告警钉钉通知。

## 3. 云游 Global 解决方案

### Global 导出解决方案交付物

#### 添加产品解决方案

1. 在 交付升级 页面，新增解决方案。



2. 进入创建的解决方案。
3. 单击 添加产品 > 自定义添加。



4. 在 自定义添加产品 页面，完成以下配置。



- 产品：输入 资金安全监控/clriskmonitor - CLRISKMONITOR。
- 产品版本：选择相应的版本。

- 部署拓扑：选择 标准单机房部署。
- 部署规格：根据实际情况，选择 标准生产-单机房 或 标准 POC。

5. 单击 提交审批。

## 导出交付物

1. 单击右上角 导出交付物。



2. 保存交付物以备云游 Local 导入。

## 导出应用镜像

1. 在云游 Local 导入解决方案后，在 解决方案元数据 查看应用的镜像。



2. 本地执行依次执行 `docker pull` 和 `docker save`，如需离线导镜像，则执行如下命令：

```
docker pull acs-reg.alipay.com/acloud/sofaclrisk: XXXXXXXX-XXXXXXXXXXXXX

// 保存tar以备local上传
docker save -o acs-reg.alipay.com/acloud/sofaclrisk:XXXXXXXX-XXXXXXXXXXXXX.tar docker pull acs-reg.alipay.com/acloud/sofaclrisk:XXXXXXXX-XXXXXXXXXXXXX
```

3. 执行以下命令保存 `sofaclrisk:XXXXXXXX-XXXXXXXXXXXXX.tar` 以备云游 Local 上传。

```
docker save -o sofaclrisk:XXXXXXXX-XXXXXXXXXXXXX.tar acs-reg.alipay.com/acloud/sofaclrisk:XXXXXXXX-XXXXXXXXXXXXX
```



## 4. 现场前置准备

### 数据库

```
CREATE DATABASE `sofaclrisk_db`
```

连接信息在云游配置补全。

- datasource\_url
- datasource\_username
- datasource\_password
- datasource\_database

### Redis

先获取 Redis 实例（使用已有或新建（新建时需配置白名单）），再获取 Redis 实例密码。

记录 Redis 实例地址、密码、端口以备参数规划使用。

#### 注意

配置 Redis 密码需使用密文，加密方式如下：

- 通过如下接口修改 \${password} 为明文密码。

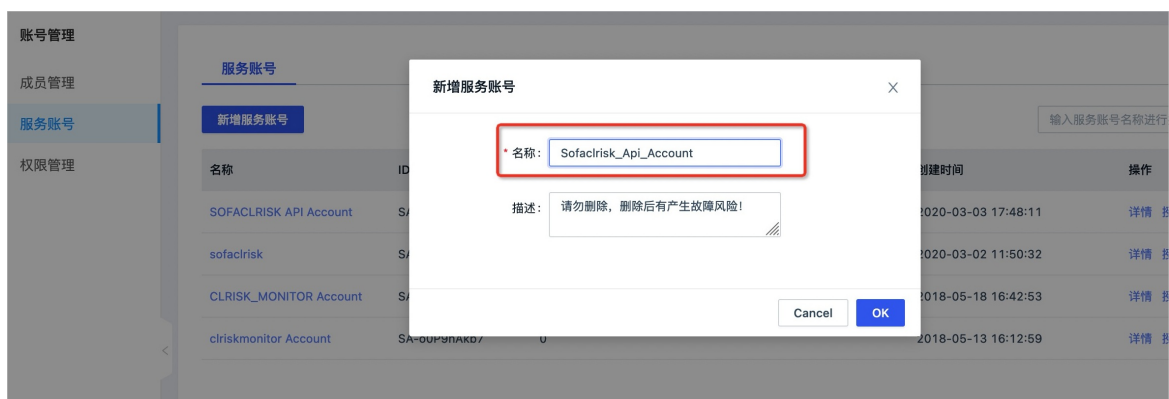
```
curl 'http://127.0.0.1:8080/godWindow/encryptTool/encryptRedisPassword?godKey=jgnmlxtxqqtgybj&password=${password}'
```

- 联系技术支持人员修改。

## IAM

### 创建服务账号、授权及 AKSK

1. 打开 admin 地址页面，切换至 ROC 租户创建服务账号。



- 名称：Sofaclrisk\_Api\_Account。
  - 描述：TRaaS 资金安全监控生产 API 调用使用。
2. 单击创建的账号，记录 AK 和 SK 以备参数规划使用。



3. 将 AK、SK 明文密码转为密文备用，操作方式如下：

- 增加如下接口说明。将 `${OpAk}` 替换为明文 AK，将 `${OpSk}` 替换为明文 SK。

```
curl 'http://127.0.0.1:8080/godWindow/encryptTool/encryptOPAkSk?godKey=jgnmlxtxqqtgyb  
&j&ak=${OpAk}&sk=${OpSk}'
```

- 联系技术支持人员修改。

## 文件存储

确认文件存储类型，当前版本支持 OSS，使用用户的 OSS 存储，在产品部署好之后配置。

### 注意

需要在页面配置一个特定名称的 OSS 数据源配置，名称为 atfinorm-prod。

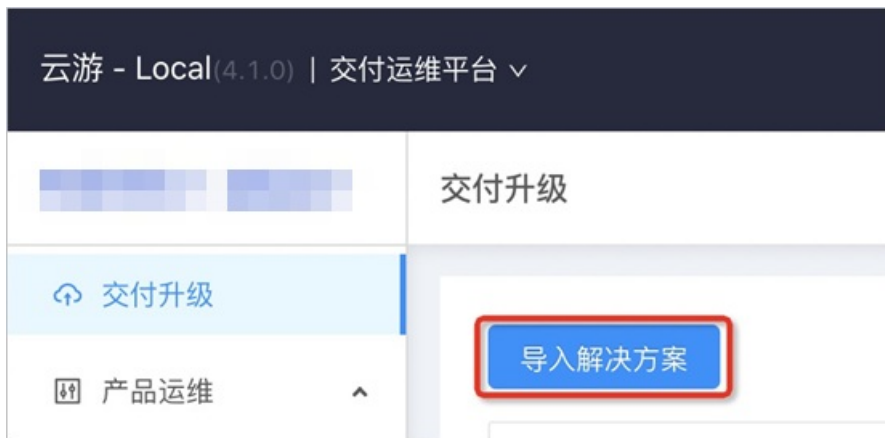
## 实时监控 RMS

- 中枢监控配置端口 8080\80 产品探活告警。
- 中枢监控配置 sof aclrisk FGC 频度告警，建议一分钟超过 10 次 FGC 告警。

## 5.云游 Local 部署

### 导入解决方案

1. 在 交付升级 页面，单击 导入解决方案。



2. 使用 导出交付物 中下载下来的解决方案，进行上传。



### 数据库规划

#### 创建数据库实例 & Schema

1. 单击 数据库 > 物理实例视角，根据实际需要创建实例。



2. 单击 数据库 > Schema 视角，选择创建的数据库实例，单击 新增 Schema。

### ? 说明

如果已创建 Schema，请忽略该步骤。



3. 选择 **手动输入**，填写 Schema 名称、账号、密码。

- **Schema 名称**：sofaclrisk\_db。
- **账号**：输入账号。获取方式请参考 [数据库](#)。
- **密码**：输入密码。获取方式请参考 [数据库](#)。

4. 单击 **确认**。

## 数据库规划

1. 单击 **编辑**。



## 2. 选择 使用已有数据库实例，添加 sofaclrisk\_db schem。

编辑数据库实例

数据库使用策略: ☐ 自动新建数据库实例 ☒ 使用已有数据库实例

当前实例: zhc1 [查看数据库实例详情](#)

实例类型		状态	RUNNING	创建时间	2020-11-25 19:10:28
数据库类型		规格	customize-speccode		
CPU	2 核	数据库内存	4096 MB	数据库空间	500 GB

已添加的 Schema (1):

选择未部署的 Schema (0 / 0):

Schema 名称	产品码	当前数据库实例	状态
暂无数据			

取消 确定

## 3. 单击 确定，确认数据库规划。

## 负载均衡规划

单击 确认负载均衡规划。

## 参数确认

## 产品公共参数

参数名	示例	说明
op.ak	RElyMzxxxxxxxxxx=	OpenAPI AccessKey。在 <a href="#">创建服务账号、授权及 AKSK</a> 中获取。
op.sk	M0xxxxxxxxxxxxTU4	OpenAPI SecretKey。在 <a href="#">创建服务账号、授权及 AKSK</a> 中获取。
spring.redis.host	r.aliyunrds.com	Redis host
spring.redis.password	NTY0QkM3ODIzRUFEMjB CMjNBREE4NTg4OUFGMk U1RjA=	Redis Password <div><div>说明</div><p>该 Password 是密文，请将 Redis 密码设置为 Sofaclrisk123，然后填写左侧示例值即可。如果密码设置失败，请联系技术支持生成密码对应的密文。</p></div>

参数名	示例	说明
url.domain	http://sofaclriskweb.galaxy.ant-galaxy.com/#/sofaclrisk	该参数用于确认发邮件时跳转的平台地址，请按照实际情况修改成站点对应的域名 <code>sofaclriskweb.xxx.xxx.xxx</code> 。

## 部署发布

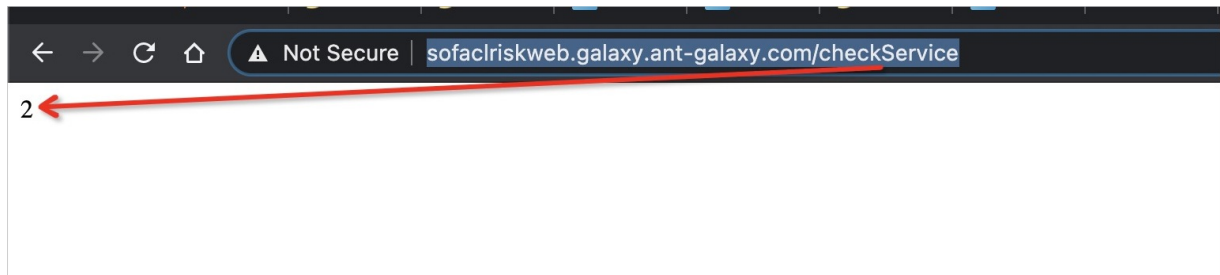
发布服务上线，按正常流程部署发布即可。

## 6. 部署验证

在 **产品基线** 中找到资金安全监控产品，找到其中 sofaclrisk 的公网域名，该域名地址即是资金安全监控的控制台地址。

### 验证服务正常

访问 <http://sofaclriskweb.galaxy.ant-galaxy.com/checkService>（请将域名替换成相应域名），若返回 2，说明正常服务。

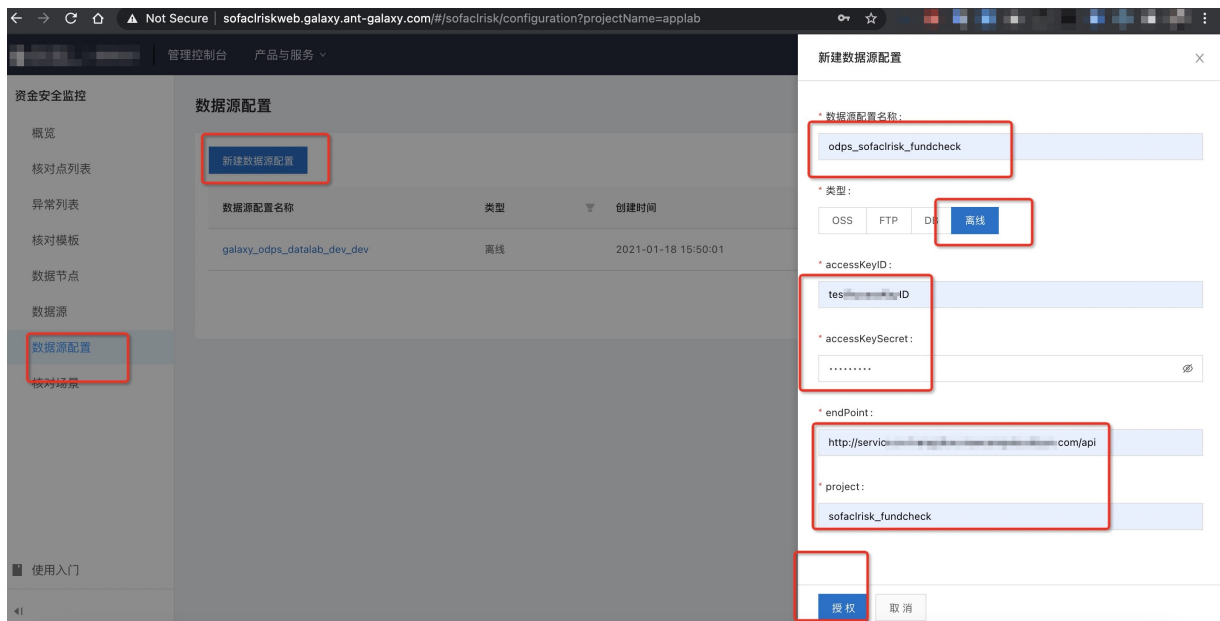


### 验证可正常配置核对点

详细步骤请参考《资金安全监控-使用指南》。

### 新建数据源配置

配置信息（AK、SK、endPoint、project）请通过客户 MaxCompute 获取。

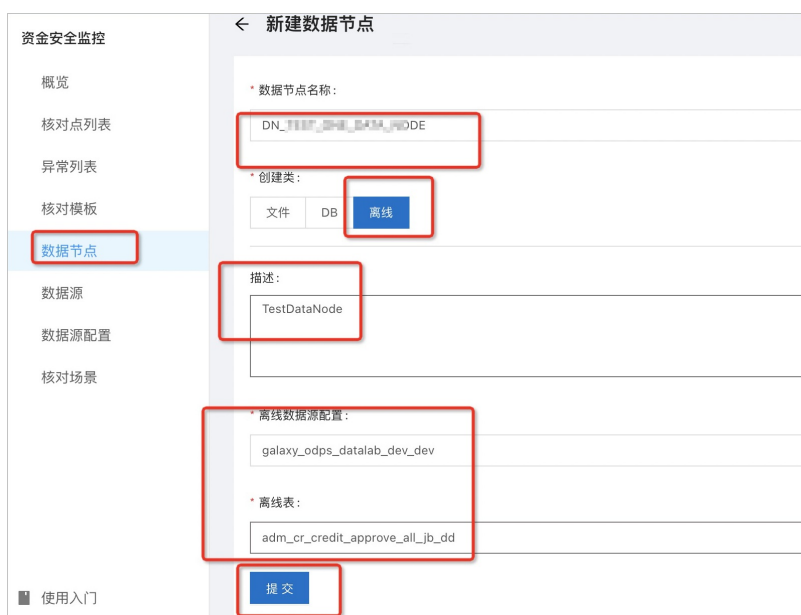


### 新建数据节点

1. 单击 **数据节点 > 新建数据节点**。



2. 在 新建数据节点 页面，完成参数配置。



## 新建核对点

1. 单击 核对点列表 > 新建核对点。



2. 在 新建核对点 页面，完成参数配置。



资金安全监控

概览

核对点列表

异常列表

核对模板

数据节点

数据源

数据源配置

核对场景

使用入门

新建核对点

基本信息

\* 核对点名称:

业务对账\_...

\* 核对场景:

资金安全交付测试

描述:

新建核对点测试

\* 负责人:

...

核对配置

\* 核对方式:

文件-文件 文件-DB DB-DB 离线

\* 数据节点:

...st

\* 核对类型:

☒ 小时核对 ☐ 天核对

\* 延迟执行时长:

15 分钟

提交

## 编写规则

更多 SQL 样例可参考《资金安全监控-技术白皮书》。

SQL 可以查出数据即可，您可以通过调试功能验证是否可查出数据。示例如下：

```
SELECT *
FROM schema.ods_promo_delta_mm promo
WHERE promo.dt = '{yyyyMMdd-5m}'
  AND promo.hh='{hh-5m}'
  AND promo.mm='{mm-5}'
  AND promo.account_type <>'NEW'
  AND promo.activity_id='FUNDRIK201910191445001'
;
```

资金安全监控

概览  
核对点列表  
异常列表  
核对模板  
数据节点  
数据源  
数据源配置  
核对场景

< 修改核对规则

\* 规则名称:  
[规则名称]\_核对规则1

描述:  
[规则名称]\_核对规则1 的描述

节点信息  
adm\_galaxy\_prin\_dd

\* 规则文本 (按照 SQL 语法进行编写):  
1 select \* from adm\_galaxy\_prin\_dd limit 3;

提交

## 查看异常信息

不会立即产生异常信息，须等待运行周期。规则可以检出数据才会产生。

### ? 说明

运行周期是建核对点时设置的运行周期。

资金安全监控

概览  
核对点列表  
异常列表  
核对模板  
数据节点  
数据源  
数据源配置  
核对场景

异常列表

全部异常 全部核对场景 请输入规则名称关键字

ID	规则名称	核对场景	创建时间	核对类型	处理时长 (秒)	差异数量	状态
ISSUE00000006	[规则名称]_核对规则1	交付测试	2020-12-24 16:18:00	小时核对	--	858	待处理
ISSUE00000004	资金安全交付测试	资金安全交付测试	2020-10-20 16:18:00	小时核对	--	1330	待处理
ISSUE00000005	[规则名称]_核对规则1	交付测试	2020-12-21 02:18:00	小时核对	306062	258	已处理
ISSUE00000003	资金安全交付测试	资金安全交付测试	2020-10-20 15:54:00	小时核对	1384	1	已处理
ISSUE00000002	资金安全交付测试	资金安全交付测试	2020-10-16 15:18:00	小时核对	--	74	待处理

## 验证可正常发送钉钉告警

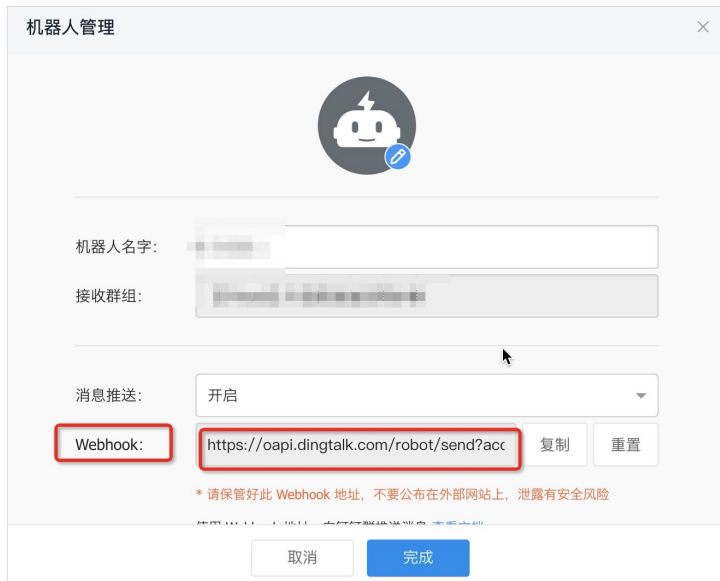
钉钉告警发送邮件会调用公网钉钉接口，需要资金安全监控运行的容器可以访问

`https://oapi.dingtalk.com` 域名，即需要打通外网访问。验证步骤如下：

1. 在测试钉钉群中添加机器人。



2. 在 **机器人管理** 页面，记录机器人的 Webhook，并勾选安全设置加签、记录签名秘钥。



机器人管理

机器人名字: [输入框]

接收群组: [输入框]

消息推送: 开启

Webhook: https://oapi.dingtalk.com/robot/send?acc 复制 重置

\* 请保管好此 Webhook 地址，不要公布在外部网站上，泄露有安全风险

取消 完成



机器人管理

Webhook: https://oapi.dingtalk.com/robot/send?acc 复制 重置

\* 请保管好此 Webhook 地址，不要公布在外部网站上，泄露有安全风险

使用 Webhook 地址，向钉钉群推送消息 [查看文档](#)

\* 安全设置

自定义关键词 ☐

说明文档 ☒ 加签

SEC8...4db16 重置

密钥如上，签名方法请参考 [说明文档](#)

☐ IP地址 (段)

移除机器人

取消 完成

### 3. 调用以下接口验证机器人是否可用。

```
curl 'http://127.0.0.1:8080/godWindow/notice/sendDingTalk?godKey=jgnmltxqqtgybj&webhook=${webhook}&secret=${secret}&content=元棒元棒的!&targetUrl=http://www.alipay.com'
```

```
// 参数中的特殊字符需要 urlEncode 一下 (webhookURL里的&变为%26)，以下为实例
// /godWindow/notice/sendDingTalk?godKey=jgnmltxqqtgybj&webhook=https://oapi.dingtalk.com/robot/send?access_token=338fcf509ac3945c853551fc9ab4408b49c56ffa794cdf611ac3f88d4e
be1bb%26secret=SEC822c3111d811ab214be423e1dfb8c4db169f1662c14aa1bbbe4877b9ae9651c1&secret=SEC822c3111d811ab214be423e1dfb8c4db169f1662c14aa1bbbe4877b9ae9651c1&content=%E5%85%83%E6%A3%92%E5%85%83%E6%A3%92%E7%9A%84!&targetUrl=http://www.alipay.com
```

其中，将 `${webhook}` 替换成步骤二中记录的 Webhook，将 `${secret}` 替换成步骤二中记录的签名密钥。

调用结束后，钉钉群中会收到一条内容为 [元棒元棒的!] 的推送，点击推送跳转至

`http://www.alipay.com`。

## 验证可正常发送邮件

调用以下接口验证邮件发送功能是否可用。

```
curl 'http://127.0.0.1:8080/godWindow/notice/sendEmail?godKey=jgnmltxqqtgybj&receiverEmail=${receive_mail_address}'
```

其中，将 `${receive_mail_address}` 替换为接收邮件人的邮箱地址。

调用结束后，会收到一封 [异常处理通知] 的告警邮件。

## 7.OSS 数据源配置

文件核对引擎执行时会使用客户的存储空间来保存临时文件。本节主要介绍文件核对在计算节点运行过程中使用的存储空间及相关协议文件的部署操作。

### 注意

bucketName 必须设置为 `[atfinorm-prod]`。如果一定要修改该名称，可以在云游参数配置中搜索值为 `atfinorm-prod` 的配置项进行修改。

### 初始化 OSS

1. 登录资金安全监控控制台。
2. 在左侧导航栏上，单击 **数据源配置**。
3. 单击 **新建数据源配置**，在右侧面板上配置以下信息：

新建数据源配置

×

\* 数据源配置名称:

atfinorm-prod

\* 类型:

OSS

FTP

DB

离线

需要您提供OSS账号的授权, 用以查询OSS下的文件

\* accessKeyId:

\* accessKeySecret:

.....

\* bucketName:

atfinorm-prod

\* endpoint:

http://oss-.....s.com

授权

取消

参数

说明

参数	说明
数据源配置名称	<p>数据源配置的名称。租户内唯一，长度为 1-100 个字符，可包含中英文、数字、下划线（_）和短横线（-）。此处输入 <code>atfinorm-prod</code>。</p> <div> <b>注意</b> 输入的数据源配置名称必须是 <code>atfinorm-prod</code>。如果要修改该名称，可以登录云游在参数配置中搜索值为 <code>atfinorm-prod</code> 的配置项进行修改。</div>
类型	<p>支持 <b>OSS</b>、<b>FTP</b>、<b>DB</b>、<b>离线</b> 四种类型。可根据您的实际业务需求自行选择。此处选择 <code>OSS</code>。</p> <div> <b>注意</b> 类型一旦选择确认后，将不能更改。</div>
accessKeyId	提供 OSS 账号的授权，用于查询 OSS 下的文件。需要从用户自己的 OSS 中获取，获取方式请参见 <a href="#">获取 AccessKey</a> 。
accessKeySecret	提供 OSS 账号的授权，用于查询 OSS 下的文件。需要从用户自己的 OSS 中获取，获取方式请参见 <a href="#">获取 AccessKey</a> 。
bucketName	输入数据源配置名称。此处输入 <code>atfinorm-prod</code> 。
endpoint	输入 project 地址。

#### 4. 单击 授权。

## 初始化协议文件

您可以通过阿里云租户控制台进行 OSS 文件管理，初始化协议文件。具体操作步骤如下：

#### 1. 新建 OSS 目录。您需要在根目录下创建如下文件夹：

- `finorm/hashed/`：用于存放哈希文件。
- `finorm/protocol/`：用于存储文件协议。
- `finorm/test/`：用于存放测试文件。



2. 下载 [协议描述文件](#)，并上传至 `finorm/protocol/` 目录。

## 8. 常见问题

### 部署后健康检查失败

#### 可能的原因

- Redis 没有设置访问白名单导致访问失败。
- Redis 密码设置错误。
- 加密配置不正确。

#### 解决方案

- 增加 Redis 访问白名单。
- 查看 `/home/admin/logs/sofaclrisk/common-error.log`，如果能找到如下日志，说明 Redis 密码相关设置有误。

```
at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:1175) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.ConstructorResolver.resolveAutowiredArgument(ConstructorResolver.java:857) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.ConstructorResolver.createArgumentArray(ConstructorResolver.java:760) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
... 25 more
Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'redisServiceImpl': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'redisClient' defined in class path resource [com/alipay/sofaclrisk/service/config/RedisConfig.class]: Bean instantiation via factory method failed; nested exception is org.springframework.beans.BeanInstantiationException: Failed to instantiate [org.springframework.data.redis.core.RedisTemplate]: Factory method 'redisClient' threw exception; nested exception is java.lang.NumberFormatException: For input string: " at org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessProperties(CommonAnnotationBeanPostProcessor.java:325) ~[spring-context-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1404) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:592) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:515) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(AbstractBeanFactory.java:320) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:318) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:199) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.config.DependencyDescriptor.resolveCandidate(DependencyDescriptor.java:277) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDependency(DefaultListableBeanFactory.java:1255) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:1175) ~[spring-beans-5.1.14.RELEASE.jar!/:5.1.14.RELEASE]
at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$AutowiredFieldElement.inject(AutowiredAnnotationBeanPostProcessor.java:586)
```

- 检查密码是否设置错误。如下示例中密码多复制了 AK。

```
[root@sjpd-clriskmonitor-sofaclrisk-0 sofaclrisk]# env | grep redis
spring.redis.database=1
spring.redis.timeout=10000
spring.redis.host=r-9yof40dec9174ca4.redis.rds.ops.aliyun-sh.ac
secret.redis=59hax2bds1988b6m
spring.redis.lock.database=0
spring.redis.port=6379
spring.redis.auth.on=true
spring.redis.password=RUI2RkQwNTdCNEZEQ0ZDMkY4NzY5OTZQ0RCRTY1MOM=AK
```

- 检查密码是否已开启加密配置，需要将 `spring.redis.auth.on` 配置为 `true`（开启密码验证）。

```
[root@sjpd-clriskmonitor-sofaclrisk-0 sofaclrisk]# env | grep redis
spring.redis.database=1
spring.redis.timeout=10000
spring.redis.host=r-9yof40dec9174ca4.redis.rds.ops.aliyun-sh.ac
secret.redis=59hax2bds1988b6m
spring.redis.lock.database=0
spring.redis.port=6379
spring.redis.auth.on=true
spring.redis.password=RUI2RkQwNTdCNEZEQ0ZDMkY4NzY5OTZQ0RCRTY1
```

- 检查 `spring.redis.password` 配置，是否已设置成密文。您可以参考 [Redis](#) 获取密文。

## 邮件发送失败

### 可能的原因

可能是由于 `op.ak/op.sk` 配置项错误导致资金安全监控平台通过 Gotone 发送邮件，如果

`op.ak/op.sk` 配置错误可能导致接口调用失败。

### 解决方案

- 检查 `op.ak/op.sk` 配置项是否已配置成密文。您可以参考 [Redis](#) 获取密文。
- 您可以参考 [验证可正常发送邮件](#) 验证邮件发送是否成功。

## 钉钉告警发送失败

### 可能的原因

可能是由于外网不通导致。资金安全监控平台通过直接调用公网钉钉 API 发送钉钉告警，如果外网不通可能造成钉钉邮件发送失败。

### 解决方案

- 检查外网网络是否已打通。
- 您可以参考 [验证可正常发送钉钉告警](#) 验证钉钉告警发送是否成功。

## 9. 技术术语

术语	说明
资金安全监控	SOFAStack 的一款风险防控类型的产品，能帮助您发现资金损失的风险。该产品可以通过执行您自定义的规则，以小时频率，准实时等多种时效策略，发现资金类数据问题，向您发出告警。您可以在第一时间收到告警，根据异常数据排查问题，分析原因，进而解决问题。
规则	资金安全监控产品中的规则，也叫核对规则，是数据核对的最小单位。可结合分类来管理。规则主要由 ODPS 表和 SQL 脚本构成，另外，您还可设置一些规则相关信息。规则的核对逻辑由 SQL 体现。
任务	资金安全监控产品中的任务（Task）指核对任务，即核对规则周期性执行的任务。每个周期有一组任务实例，每次执行对应一个任务实例。
任务周期	也叫核对周期，指核对任务的调度周期。如核对频率为天核对，就是一天执行一次，例如 2018.5.29 执行的核对任务对应周期 20180529；如核对频率为小时核对，就是一小时执行一次，例如 2018.5.29 日 10 点执行的核对任务对应周期 2018052910。
异常	核对任务运行中发现数据有差异，系统会创建异常事件（Issue）。注意：任务重跑时可能会重复产生差异。
差异数据	也叫差异，指核对任务执行时（执行 SQL）发现的不匹配核对规则的数据。您可以在异常详情里查看在某个核对周期中发现的抽样差异数据。
差异数量	核对任务执行时发现的差异数据的总数。注意：您在异常详情页看到的差异数量，是在当次任务周期中发现的差异数据总数，但是在差异数据（抽样数据），仅展现最多 10 条抽样的差异数据。
核对频率	核对频率有：按天核对、小时核对、5 分钟核对和自定义核对。 <ul style="list-style-type: none"><li>• 小时核对：一小时执行一次核对。</li><li>• 天核对：一天执行一次核对。</li><li>• 5 分钟核对：5 分钟执行一次核对。</li><li>• 自定义核对：根据用户配置的 cron 表达式触发核对执行。</li></ul>