

SOFAStack

消息队列 SOFAKafka 部署指南

产品版本：AntStack Plus 1.13.1


文档版本：20230707

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。 |
| 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 <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.前置条件 | 05 |
| 2.准备工作 | 06 |
| 2.1. 资源准备 | 06 |
| 2.2. 云游 Global 解决方案 | 06 |
| 2.3. 云游 Local 部署 | 08 |
| 2.4. 参数确认 | 12 |
| 3.发布部署 | 19 |
| 4.部署验证 | 20 |

1.前置条件

版本信息

SOFAKafka: 1.3.1。

归属产品集：共享中间件 1.22.0。

产品部署依赖

依赖产品列表：ACVIP、中间件支持中台（OSP）。

② 说明

本文档的相关部署需要在云游进行，请提前完成云游的部署。

部署产品

| 产品码 | 部署拓扑 | 部署规格 |
|-------|---------|----------|
| ACVIP | 标准单机房部署 | 标准生产-单机房 |
| OSP | 标准单机房部署 | 标准生产-单机房 |

2.准备工作

2.1. 资源准备

| 类型 | 应用 | 规格 | 实例数 |
|-------------|----------------|---|-----|
| 容器 | afterbroker | 2 C/4 GB/100 GB（双核 CPU、4 GB 内存、100 GB 硬盘） | 1 |
| | kafkabroker | 2 C/8 GB/100 GB | 3 |
| | kafkaconsole | 2 C/4 GB/100 GB | 1 |
| | kafkazookeeper | 2 C/4 GB/100 GB | 3 |
| | kafkarouter | 2 C/4 GB/100 GB | 1 |
| 负载均衡 | kafkabroker | 内网 | 1 |
| | kafkazookeeper | 内网 | 1 |
| 数据库（Schema） | sofakafkadb | 4 C/8 GB/100 GB（四核 CPU、8 GB 内存、100 GB 硬盘） | 1 |

重要

若执行重启容器操作，必须通过云游强制重启应用的操作。

2.2. 云游 Global 解决方案

新增解决方案

1. 登录云游 Global。
2. 进入目标环境后，在 交付升级 页面，单击 新增解决方案。
3. 在 新增解决方案页面，确定解决方案名称和部署范围，单击 确认。
4. 进入新增的解决方案，单击 添加产品。
5. 根据部署的产品，在 基于产品集添加产品页面 页面，填写相关配置，需部署的产品清单如下表所示：

| 产品码 | 部署拓扑 | 部署规格 |
|----------|---------|----------|
| REGISTRY | 标准单机房部署 | 标准生产-单机房 |
| ACVIP | 标准单机房部署 | 标准生产-单机房 |
| OSP | 标准单机房部署 | 标准生产-单机房 |

- 单击 **提交**。
- 选择新增解决方案中的添加的产品，在该页面右上方单击 **导出交付物**。
- 保存导出的交付物以备云游 Local 导入。



重要

新增的解决方案必须是云游的标准方案。

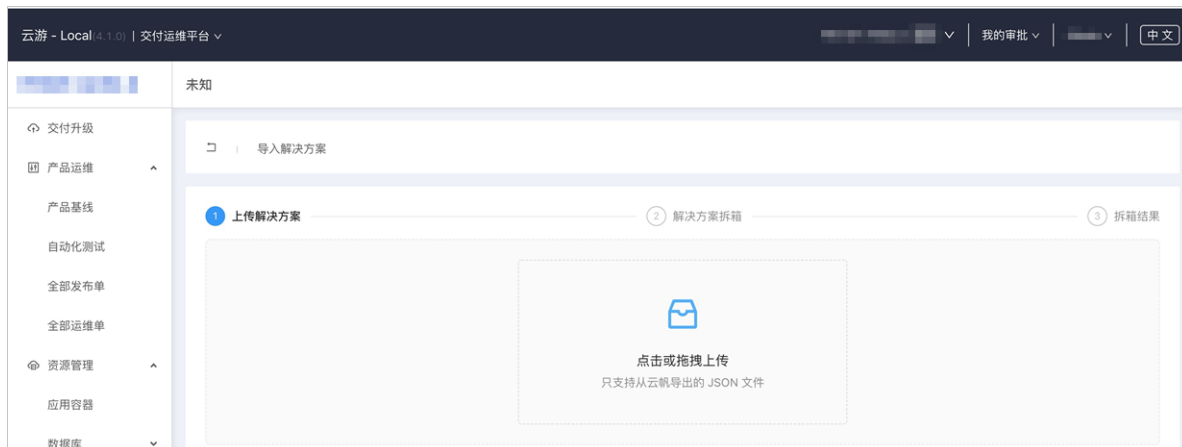
获取镜像下载地址

在 Global 控制台获取上述产品的镜像时，需安装完成 Docker，当 Docker 安装完成后，您可以通过 `docker pull` 命令下载这些镜像。

- 登录云游 Global 控制台。
- 进入目标环境后，在交付升级页面，单击 **规划**，然后单击 **产品** 页签。
- 选择要下载镜像的产品名，然后单击 **下载镜像** 获取下载链接。



- 在 **镜像下载地址** 对话框单击 **复制**。



2. 在 导入解决方案 页面点击 点击或拖曳上传，上传 [云游 Global 解决方案](#) 中导出的交付物。

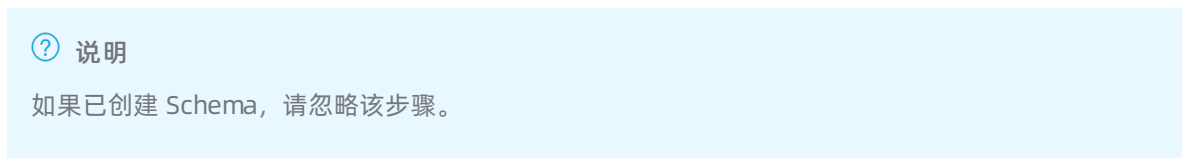
数据库规划

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。



3. 选择手动输入，在新增 Schema 面板下填写以下信息：

- Schema 名称：本例中输入 `sofakafkadb`。
- 账号：输入账号。
- 密码：输入密码。

4. 单击 确认。

编辑数据库实例

1. 在解决方案详情页面，单击 解决方案元数据 > 数据库规划。
2. 单击铅笔图标。

3. 选择 使用已有数据库实例，添加 sofakafkadb Schema。

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 ☐ 使用已有数据库实例组 ?

当前实例：[查看数据库实例详情](#)

| | | | | | |
|-------|-------|-------|--------------------|-------|---------------------|
| 实例类型 | 自定义 | 状态 | RUNNING | 创建时间 | 2021-12-12 10:10:10 |
| 数据库类型 | MYSQL | 规格 | customize-speccode | | |
| CPU | 4 核 | 数据库内存 | 8192 MB | 数据库空间 | 100 GB |

已添加的 Schema (1) :

选择未部署的 Schema (0 / 0) :

Schema 名称

| <input type="checkbox"/> | Schema 名称 | 产品码 | 当前数据库实例 | 状态 |
|----------------------------|-----------|-----|---------|----|
| <div><div></div>暂无数据</div> | | | | |

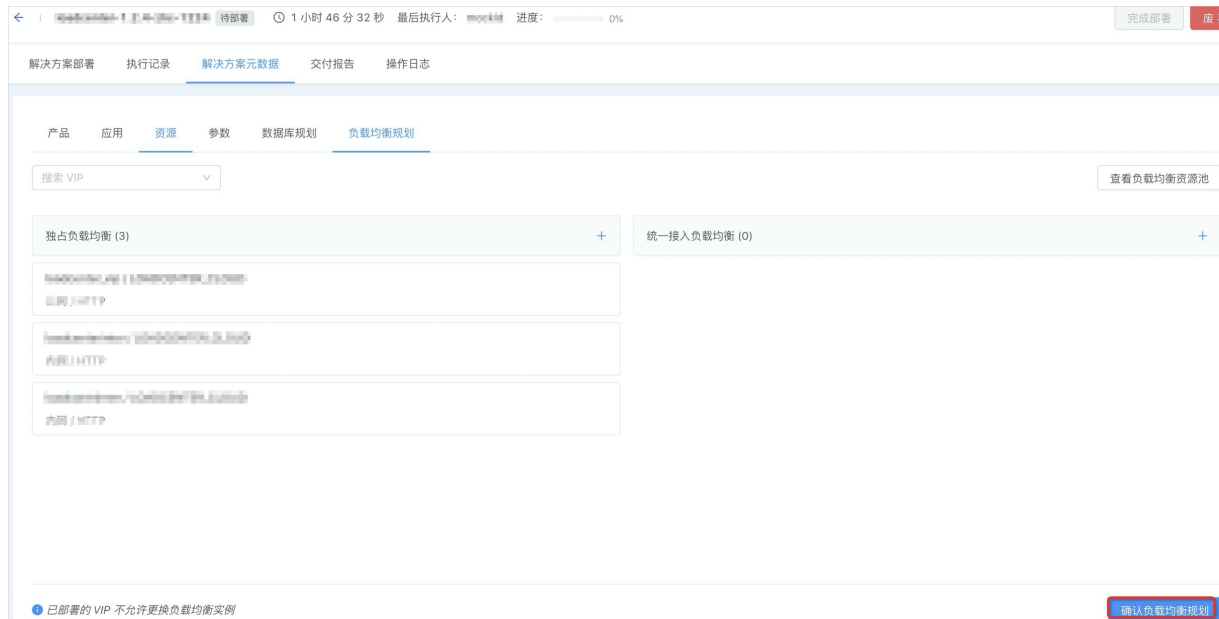
取消

确定

4. 单击 确定。

负载均衡规划

1. 在解决方案详情页面，单击 解决方案元数据 > 负载均衡规划。
2. 单击 确认负载均衡规划。



2.4. 参数确认

应用安装完成后，启动前请确认如下参数：

| 应用名 | 参数名 | 值 | 说明 |
|-----|-----------------------|---------------|-----------------------|
| | product_access_secret | *** | product access secret |
| | product_cluster_id | mockclusterid | product cluster id |
| | product_cname | sofakafka | 产品中文名 |
| | product | SOFAKAFKA | product |
| | product_order | 20 | product order |
| | product_enname | SOFAKAFKA | 产品英文名 |
| | product_description | SOFAKAFKA | product description |
| | product_code | SOFAKAFKA | product code |
| | product_index_url | /index.html | product index url |

| | | | |
|-------------|--------------------------|---|---|
| afterbroker | product_doc_url | / | product doc url |
| | acvip_domain_server_port | 9092 | kafka port |
| | acvip_domain_name | SOFAKAFKA | <code>\${instanceId-SOFAKAFKA}</code> 组成真正的寻址域名 |
| | global_skip | true | 公有云部署一次性任务必填参数 |
| | acvip_domain_servers | <pre> # if (\${res.SOFAKAFKA.app.kafkabroker.anytunnelVips}) \${res.SOFAKAFKA.app.kafkabroker.anytunnelVips} # else \${res.SOFAKAFKA.app.kafkabroker.ipList} # end </pre> | Broker 地址列表 |
| | Datacenter | <code>\${env.info.datacenter}</code> | 当前机房 |
| | osp_servers | <code>\${res.OSP.app.osp.containerIps}</code> | osp servers |
| kafkabroker | SOFAKAFKA_COMPONENT | SOFAKAFKA_BROKER | 组件名称（区分启动路径） |
| | APP_KEY | SINGLE | APP_KEY 映射所属集群 |
| | DATA_CENTER | <code>\${env.info.datacenter}</code> | 当前机房 |
| | CONSOLE_URL | <code>http://\${res.SOFAKAFKA.lb.kafkaconsole.vip}:8080</code> | 管控台地址 |
| | | | |

| | | | |
|--------------|-----------------------|---|---|
| | ZOOKEEPER_SERVERS | \${res.SOFAKAFKA.app.kafkazookeeper.containerNames} | Zookeeper 地址 |
| Kafkaconsole | app.db.driver | com.mysql.jdbc.Driver | 数据库引擎 |
| | app.profile.active | cloud | 加载 profile 文件 |
| | app.db.type | mysql | 数据库类型 |
| | app.instanceId | 00001 | 实例 ID |
| | app.db.username | \${res.SOFAKAFKA.db.sofakafkadb.sofakafkadb.user} | 数据库用户名 |
| | app.db.password | \${res.SOFAKAFKA.db.sofakafkadb.sofakafkadb.password} | 数据库密码 |
| | app.db.url | \${res.SOFAKAFKA.db.sofakafkadb.sofakafkadb.url} | 数据库 URL |
| | app.db.port | \${res.SOFAKAFKA.db.sofakafkadb.sofakafkadb.port} | 数据库端口 |
| | app.db.dbname | \${res.SOFAKAFKA.db.sofakafkadb.sofakafkadb.database} | 数据库库名 |
| | app.deployMode | \${prod.SOFAKAFKA.deploy_mode} | 部署模式。 <ul style="list-style-type: none"> 公有云：aliyun 私有云环境下：private |
| | app.endpoint.internal | \${res.ACVIP.lb.antvip-inner-slb.vip} | ACVIP 内部地址 |
| | app.endpoint.external | \${res.ACVIP.lb.antvip-inner-slb.vip} | ACVIP 外部地址 |

| | | | |
|----------------|-----------------------|---|----------------|
| | app.envDomain | sofa- sofakafka.\${env.info.do main} | 环境域名 |
| | app.broker.servers | \${res.SOFAKAFKA.app.ka fkabroker.containerNam es} | Broker 地址列表 |
| | app.zookeeper.servers | \${res.SOFAKAFKA.app.ka fkabroker.containerNam es} | ZK 地址 |
| | app.osp.url | \${res.OSP.lb.osp_interna l_lb.vip} | OSP 地址 |
| | aap.datacenter | \${env.info.datacenter} | 应用机房 |
| kafkazookeeper | SOFAKAFKA_COMPONENT | SOFAKAFKA_ZOOKEEPER | 组件名称（区分启动路径） |
| | APP_KEY | SINGLE | APP_KEY 映射所属集群 |
| | DATA_CENTER | \${env.info.datacenter} | 当前机房 |
| | CONSOLE_URL | http://\${res.SOFAKAFKA .lb.kafkaconsole.vip}:80 80 | 控制台地址 |
| | ZOOKEEPER_SERVERS | \${res.SOFAKAFKA.app.ka fkazookeeper.container Names} | Zookeeper 地址列表 |
| kafkarouter | SOFAKAFKA_COMPONENT | SOFAKAFKA_ROUTER | 组件名称（区分启动路径） |
| | CONSOLE_URL | http://\${res.SOFAKAFKA .lb.kafkaconsole.vip}:80 80 | 管控台地址 |
| | DATA_CENTER | \${env.info.datacenter} | 当前机房名称 |
| | | | |

| | | | |
|---------------|-------------------------|---|---|
| kafkacoreinit | op_provider_ak | MiddleWareProviderAK | <p>网关后端验签所用 AK。</p> <div> <p> 说明</p> <p>58 cloud (58 3套) 环境下填 ACxly8lOb8Nb****。</p> </div> |
| | op_product_instance_id | MiddleWareCluster-SOFAKAFKA | MiddleWareCluster-产品码 |
| | global_skip | false | 所有存量站点默认关闭 (true)，新增站点打开 (false)。 |
| | product_code | SOFAKAFKA | 产品码 |
| | op_region | MiddlewareRegion | 网关 Region 路由的 Region 名称 |
| | op_invoker_ak | Invoker-SOFAKAFKA | 下游调用 IAM 做鉴权时，所用的 accessKey。 |
| | op_invoker_sk | Invoker-SOFAKAFKA | 下游调用 IAM 做鉴权时，所用的 accessSecret。 |
| | aciamweb_container_ips | <pre>#if(\${env.prod.IAM.exist})) \${res.IAM.app.aciamweb.containerIps} #end</pre> | aciamweb 容器 IP 列表 |
| | aciamcore_container_ips | <pre>#if(\${env.prod.IAM.exist})) \${res.IAM.app.aciamcore.containerIps} #end</pre> | aciamcore 容器 IP 列表 |
| | op_container_ips | <pre>#if(\${env.prod.OP.exist})) \${res.OP.app.acprodapi.gw.containerIps} #end</pre> | OP 容器 IP 列表 |

| | | | |
|-----------|---------------------------|--|--|
| | op_route_endpoint | <pre> # if (\${env.system.zoneDomain}) http://sofa-osp.\${env.system.zoneDomain} # else http://sofa-osp.\${env.info.domain} # end </pre> | 网关路由的下游地址 |
| kafkatest | ANTCLOUD_OUTPUT_TYPE | 阿里云-公有云 | 金融云输出形态 |
| | run.environment | 测试环境 | 用例运行环境 |
| | com.alipay.instanceid | 000001 | InstanceId |
| | ENV_PROFILE | yunyou | 环境配置文件名称 |
| | com.alipay.env | shared | 共享中间件 |
| | openapi.productInstanceId | MiddleWareCluster-MQ | OP InstanceId |
| | openapi.accessKey | MiddleWareProviderAK | OP accessKey |
| | openapi.accessSecret | MiddleWareProviderSK | OP accessSecret |
| | openapi.version | 1.0 | OP Version |
| | http.test.projectName | ALIPAYCN | 中间件租户 |
| | http.test.workspaceName | middleware | 中间件 Workspace |
| | ui.test.adminAccount | antCloudAdmin | 空间管理员账号，如果不存在antCloudAdmin 账号，请交付人员修改为该环境的空间管理员账号。 |

| | | | |
|--|------------------------------|---|--|
| | ui.test.adminPassword | *** | 空间管理员密码，如果不存在antCloudAdmin 密码，请交付人员修改为该环境的空间管理员密码。 |
| | com.antcloud.antvip.endpoint | <pre> #if (\${env.prod.ACVIP.exist}) \${res.ACVIP.lb.antvip- inner-slb.vip} #else 127.0.0.1 #end </pre> | antvip endpoint |
| | openapi.endpoint | <pre> #if (\${env.prod.OP.exist}) http://\${res.OP.lb.antpr odgw_internal_lb.vip}/g ateway.do #else #end </pre> | OP 地址 |
| | http.test.portalUrl | <pre> \${env.info.protocol}adm in.\${env.info.domain} </pre> | portalUrl |

3.发布部署

操作步骤

1. 登录云游 Local。
2. 单击 交付升级，然后单击目标解决方案。
3. 单击 解决方案元数据，然后单击 参数。
4. 完成 资源准备 及 参数确认 后，单击 解决方案，检查参数和镜像变化。
5. 在该页面下将鼠标放在后面的感叹号上，会显示出当前基线和新解决方案之间的参数差异，确认无误后再进行下一步发布。

说明

请将该页面进行截图，把截图发给技术支持进行确认。

6. 单击 解决方案部署，然后单击 批量发布。
7. 选择需要发布的应用后，单击 确定，执行发布。

重要

需要开启 Beta 分组的开关，部署策略需要设置为分两组后开始部署应用。

后续操作

1. 登录 SOFA-OSP 控制台，单击 启动。
2. 确认中间件租户下 Middleware 和 Core 工作空间的 SOFAKAFKA 已启动，如下图所示：



4. 部署验证

1. 在云游平台 **全部发布单** 页面中找到刚刚执行的升级变更单，找到 SOFAKAFKA 升级发布的应用。
2. 当升级部署没有报错的情况下，发布单执行完成后，在云游启动自动化测试用例。

② 说明

若配置接口成功跑通，则发布部署成功。