SOFAStack

AntStack Plus

物理机底座安装手册

产品版本：AntStack Plus 1.11.0

文档版本：20220929

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。

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 `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.准备工作 06
2.部署架构 08
3.环境要求 09
4.部署流程 14
4.1. 使用 AntBoot 安装 AKEPlus 和 AntStackPlus 14
4.1.1. 安装前须知 14
4.1.2. 安装 Antboot 14
4.1.3. 加载 AKEPlus 和 AntStackPlus 安装包 15
4.1.4. 编辑配置文件 15
4.1.5. 部署 AKEPlus 和 AntStack Plus 34
4.1.6. 后续操作 36
4.1.7. 附录：端口汇总 39
4.2. 云游 Global 创建环境 40
4.2.1. 在云游 Global 上新建客户、云和环境 40
4.2.2. Global 修改环境设置 44
4.3. 云游 Local 环境初始化 49
4.3.1. 新增环境 49
4.3.2. 修改站点系统环境变量 56
4.4. 云游 Local 发布 AntStack Plus 其他必要组件 57
4.4.1. 部署 yunyouoperator 和 apwebhook 57
4.4.2. 部署云游 Agent 62
4.4.3. 部署 ADNS Operator 64
4.4.4. 部署 IAM 和 OP 65
4.5. 云游接入 IAM 68
4.5.1. 云游 Local 成员管理接入 IAM 68
4.5.2. 云游接入 IAM 审计功能 72

4.6. 使用云游的一键部署发布 SOFAStack 产品 ---------- 73

1.准备工作

适用版本

本文档适用于 **AntStack Plus 1.11.0 版本在物理机、虚拟机、ECS 等云主机上的安装。**

AntStack Plus 1.11.0 包含的产品，以 Global 上产品集定义为准：

名称	版本	交付产物
Antboot	1.0.1	antboot-1.0.1.tar.gz
AKE3/AKEPLUS	2.1.0	AKE-Plus-2.1.0.aap
AntStack DNS/ADNS	1.18.0	AntStack-Plus-1.11.0.aap
Captain Plus/CAPTAINPLUS	1.4.0	
YUNYOU Plus/YUNYOUPLUS	1.16.1	
IAM/IAM	2.43.12	不含
OP/OP	0.31.4	

准备交付产物

制作解决方案

为了和以前的交付方式兼容，云游 Operator、adns Operator、IAM 和 OP 依然由云游 Local 发布，因此需要在 Global 导出解决方案。

产品名	产品码	版本	拓扑	规格
云游 Plus	YUNYOUPLUS	1.16.1	operator 部署	标准生产-单机房
AntStack DNS	ADNS	1.18.1	adns Operator 部署	标准生产-单机房
身份认证和权限管理	IAM	2.43.12	私有云建站(全量 PaaS 平台)	标准生产-单机房

开放网关	OP	0.31.4	专有云模式部署	标准生产-单机房

- AntStack Plus 1.9.0 开始，AntStack Plus 产品集都打包在了 aap 包内，直接下载即可推送到本地镜像仓库，无需从 Global 单独下载镜像。
- 镜像中心 localregistry 域名为 `acs-reg.alipay.com` 。

AntBoot 安装介质

名称	安装包
Antboot 安装包	antboot-service-1.0.1.tar.gz
AKE Plus 2.1.0 产品安装包	AKE-Plus-2.1.0-RC.aap
AntStack Plus 1.11.0 产品安装包	AntStack-Plus-1.11.0-UNKNOWN.aap

将以上文件上传到要安装的集群的某个节点上。后续操作都在该节点执行。

建议下载 Extra RPMs 包以方便日常使用，没有这些软件不影响系统安装或使用。如果客户集群有可用的 yum 源，建议优先使用 yum 源提供的软件。

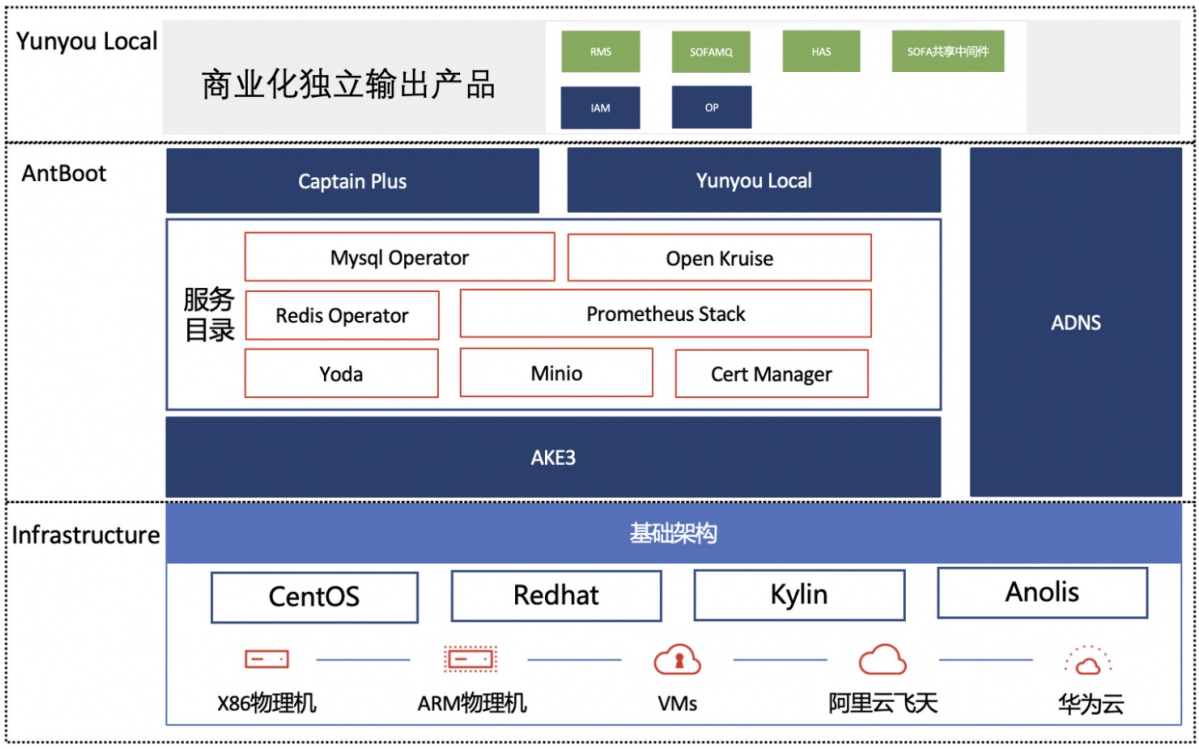
软件包中包含如下软件：

- bash completion：可以为 antboot、kubectl 等配置 bash 命令行自动补全。
- sshpass：方便配置免密等。

2.部署架构

AntStack Plus 底座部署升级，物理机环境使用 AntBoot 交付。AntBoot 可以拉起 AKE3 集群，并将云游 Local 和 Captain Plus 部署在 AKE3 集群内。

- 使用 Captain Plus 对 AKE3 集群进行运维（扩展集群节点、重启等）。
- 使用云游 Local 部署商业输出产品（SOFAStack 支持使用云游一键部署功能）。
- ADNS 会使用 Docker 部署在 worker 节点（coredns 不会调度到 ADNS master/slave 所在节点）。



3.环境要求

目前 Antboot 支持在 ECS 主机、虚拟机或者物理机上安装 AKE Plus 和 AntStack Plus。

资源需求

以下资源配置根据生产环境进行推荐：

机器视角

- Master 节点

节点角色	CPU	内存	数量	存储
Master 节点	16c	32 GB	3	mysql-data 20 GB
Master 节点（不需要 MySQL）	12c	24 GB	3	-

  - 其中 mysql-data 数据盘来自 Yoda VG，20 GB 的磁盘需求指不考虑和其他产品共用。
  - 不需要 MySQL 是指由客户为 AntStack Plus 提供数据库，Antboot 不再额外安装数据库。

- Worker 节点

  Worker 节点可按产品的资源需求添加。

  每个节点的磁盘要求：

目录	备注	推荐大小	预期的增长
/home/t4	○ 用于活跃容器运行时（包括 Pod）和本地镜像存储。 ○ 保存 etcd、master 和节点容器的数据。	最低 200 GB，推荐 500 GB 以上	○ 集群元数据增长较慢。 ○ 日志文件可能会快速增长，可以通过日志轮转控制占用量。
/var/lib/kubelet	kubelet 默认工作目录，会存放临时卷的文件（包括 emptyDir)、plugin 文件等。	100 GB	增长受运行容器容量的限制。

- 本地存储磁盘限制

  用于 Yoda（Open Local）VG 的磁盘根据产品需求需要单独挂载。Yoda 挂载盘可以是单独的磁盘分区，或者单独的磁盘（无须设置 ext4、xfs 等文件系统）。

> **说明**
>
> 根据集群规模推荐大小配置。

- 网卡限制

  主网卡名必须小于等于 5 位。由于 CNI 插件需要在网卡名加后缀，若超过 Linux 内核网卡名长度限制会导致 CNI 插件运行失败。

服务视角

服务	CPU	内存	数量	PV（Yoda存储）	备注
yunyou	2c	4 GB	2	20 GB * 2	对应 ops 角色
captain	2c	4 GB	2	20 GB * 2	对应 ops 角色
adns	2c	4 GB	4		对应 ADNS 角色
yunyou operator	2c	4 GB	2		
adns operator	2c	4 GB	2		
mysql	4c	8 G.	2	20 GB	
cert manager	0.4c	0.5 GB	2		
open kruise	0.2c	0.5 GB	2		
yoda	2c	2 GB	2		
hybridnet	1c	1.5 GB	2		
AKE master	8c	16 GB	3		
kube-prometheus-stack	-	-	-	300G * 3	硬性要求，如果安装监控产品需要磁盘

推荐的操作系统

请参考如下兼容性说明：

操作系统	AKE-Plus-2.x (k8s.1.22)
centos 7.4	支持
centos 7.6	支持
centos 7.8	支持
centos 7.9	支持
alios 4000（内核 4.19）	支持
alios 2008（内核 3.10）	不支持（containerd 版本不支持）
Anolis os 7.9	支持

环境需要的配置

配置节点免密

Antboot 安装的节点需要对其他节点配置免密。如果没有 sshpass 需要先安装：

- 使用 yum 源提供的软件

```
yum install sshpass -y
```

- 下载 Extra RPMs 包

  下载并解压之后，在解压目录安装 sshpass：

```
yum localinstall ./sshpass-*.rpm
```

配置免密方式如下：

- 以 root 用户部署

```
# 将下面的 IP 列表换成其他节点的 IP
cat > ip_list <<'EOF'
192.168.10.241
192.168.10.242
192.168.10.243
EOF

ROOT_PWD="<root password>"
ssh-keygen -t rsa -N '' -f /root/.ssh/id_rsa
cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys
cat ip_list | xargs -n 1 -I {} sshpass -p $ROOT_PWD ssh -oStrictHostKeyChecking=no root@{
} exit
cat ip_list | xargs -n 1 -I {} sshpass -p $ROOT_PWD ssh -oStrictHostKeyChecking=no root@{
} mkdir -p /root/.ssh
cat ip_list | xargs -n 1 -I {} sshpass -p $ROOT_PWD scp /root/.ssh/authorized_keys root@{
}:/root/.ssh/authorized_keys
```

- 以非 root 用户部署

```
# 当前非root用户
SUDO_USER_NAME=`id -un`
SUDO_USER_PWD="<当前sudo用户密码>"

# 将下面的 IP 列表换成其他节点的 IP
cat > ip_list <<'EOF'
192.168.10.241
192.168.10.242
192.168.10.243
EOF

mkdir -p ~/.ssh

if [ ! -f "~/.ssh/id_rsa" ]; then
    ssh-keygen -t rsa -N '' -f ~/.ssh/id_rsa
fi

cat ip_list |xargs -n 1 -I {} sshpass -p $SUDO_USER_PWD ssh  -oStrictHostKeyChecking=no $
SUDO_USER_NAME@{} exit
cat ip_list |xargs -n 1 -I {} sshpass -p $SUDO_USER_PWD ssh  -oStrictHostKeyChecking=no $
SUDO_USER_NAME@{} mkdir -p /home/$SUDO_USER_NAME/.ssh
cat ip_list |xargs -n 1 -I {} sshpass -p $SUDO_USER_PWD scp  ~/.ssh/id_rsa.pub  $SUDO_USE
R_NAME@{}:/home/$SUDO_USER_NAME/.ssh/authorized_keys
cat ip_list |xargs -n 1 -I {} sshpass -p $SUDO_USER_PWD scp  ~/.ssh/id_rsa.pub  $SUDO_USE
R_NAME@{}:/home/$SUDO_USER_NAME/.ssh/authorized_keys

# 让 root 用户这个密钥登录
cat ip_list |xargs -n 1 -I {} sshpass -p $SUDO_USER_PWD ssh  -oStrictHostKeyChecking=no $
SUDO_USER_NAME@{} sudo mkdir -p "/root/.ssh"
cat ip_list |xargs -n 1 -I {} sshpass -p $SUDO_USER_PWD ssh  -oStrictHostKeyChecking=no $
SUDO_USER_NAME@{} sudo cp /home/$SUDO_USER_NAME/.ssh/authorized_keys /root/.ssh/authorize
d_keys
```

关闭 SELinux

- 物理机或 VM
  i. 查看 SE-Linux 状态：

```
$ sestatus
SELinux status:                 disabled
```

  ii. 如果不是 `disabled` 状态，需修改 `/etc/selinux/config` 配置，将 SELINUX 改成 `disabled` ，然后重启机器。

```
vim /etc/selinux/config
SELINUX=disabled
```

- 安装 LVM2（可选：如果需要安装 `yoda/open-local` ）

  在所有需要提供 yoda 磁盘的主机上执行：

```
yum install -y lvm2
```

4.部署流程

4.1. 使用 AntBoot 安装 AKEPlus 和 AntStackPlus

4.1.1. 安装前须知

AntBoot 部署需要安装：

- hybridnet
- open-local
- 云游 Local
- Captain Plus
- localregistry
- ADNS
- cert-manager（adns Operator需要）

还可以部署：

- Prometheus（可选）
- MySQL（可选）

  虽然 AntBoot 中自带的 MySQL Operator，但目前只支持部分但产品输出场景，未经过大规模生产使用，使用前请先联系 AntStack 产品经理授权，否则无法给与您 AntStack Plus 的支持。
- openkruise（可选）

4.1.2. 安装 Antboot

操作步骤

1. 下载 Antboot 安装包。
2. 解压安装包。

   ```
   tar xzfv antboot-services-<version>.tar.gz
   ```

3. 执行安装脚本。

   ```
   cd antboot-service/
   bash -x install.sh
   ```

4. 查看 Antboot 状态。

   执行如下命令：

   ```
   systemctl status antbootd
   ```

   显示状态为 `active (running)` 即为可用。

4.1.3. 加载 AKEPlus 和 AntStackPlus 安装包

执行 `antboot load package PACKAGE_FILE` 命令加载要安装的安装包，其中 `PACKAGE_FILE` 替换为您 AAP 产品包的路径。

命令示例如下：

```
antboot load package /path/to/AKE-Plus-<version>-GA.aap
antboot load package /path/tp/AntStack-Plus-<version>-GA.aap
```

若初次加载有问题，可使用 `--override` 选项覆盖已加载版本：

```
antboot load package --override /path/to/AKE-Plus-<version>-GA.aap
antboot load package --override /path/tp/AntStack-Plus-<version>-GA.aap
```

4.1.4. 编辑配置文件

执行 `antboot edit config` 命令编辑配置，编辑时会启动一个 vim 进行交互，编辑完成后保存退出即可。如果您还没有编辑过配置，Antboot 会读取已经加载的安装包然后生成一份 starter 配置模板，注意 starter 配置中的值不一定是正确的，请根据文档编辑。

节点配置

- 节点通用配置

  在 `nodes.commonconfig` 中配置，表示所有节点共同使用的配置：

  ```
  [[nodes.common.vg]]
  name = "open-local-pool-0"
  pvDisks = [ "/dev/vdb" ]

  [nodes.common.ssh]
  port = 22
  user = "root"
  privateKeyFile = "/root/.ssh/id_rsa"
  ```

  - vg：节点磁盘的 vg
    - name： 从 AKE3 2.0.0 开始使用 `open-local-pool-0` 。

      AKE 2.0 之后 vg 名称应以 open-local-pool 开头，后面数字任意。不要再以 yoda 开头。
    - pvDisks：yoda 所使用的磁盘列表。
  - ssh： 所有节点的 ssh 登录配置
    - user：ssh 用户。
    - port：ssh 端口。
    - privateKeyFile：免密密钥路径，对 root 用户通常为 `/root/.ssh/id_rsa` 。

- 节点信息配置

  添加节点，每个节点至少提供以下配置：

```
[[nodes.hosts]]
ip = "10.0.0.1"
roles = [ "antboot", "worker", "ops" ]
```

  - ip：填写节点 IP，格式为单个 IP（例如 `192.168.0.1`），或者 IP 范围（例如 `192.168.0.1-192.168.0.5`）。
  - roles：为节点的角色。详情请参见下方节点角色规划说明。

- 节点角色规划

  您需要规划如下角色：

  - antboot：为 Antboot 所在的节点，register 会部署在 Antboot 节点上。
  - ops：会在 ops 节点安装 Docker。
  - adns：会在 ADNS 节点部署 ADNS，最多 2 个节点为 ADNS 节点。
  - worker：即 K8s 的 Worker 节点。
  - master：即 K8s 的 Master 节点。

  下面给出几种推荐的规划场景：

  - 生产环境高可用，推荐场景：生产环境。

    节点充裕（节点数不少于 5 个）时，可将系统服务分散到不同的节点。

节点	角色
node-1	antboot、 master
node-2	master
node-3	master
node-4	ops、adns、 worker
node-5	ops、adns、 worker
node-6 及其他	worker

- 最小高可用部署，推荐场景：测试集群、SIT 环境、UAT 集群

  至少 4 个节点，保证所有的服务至少有 2 个副本的高可用实例。

节点	角色
node-1	antboot、 ops、 master
node-2	master、 ops、 adns
node-3	master
node-4	worker、 adns

- 3 节点 Master，推荐场景：本地测试环境、POC 环境

  最少 3 个节点，保证 AKE 高可用，Captain、YunYou、ADNS 等产品 2 副本。

node-1	antboot、 master、 ops
node-2	master、 ops，adns
node-3	master、 adns

- 单节点 Master，仅用于资源有限的 POC，本地测试等

节点	角色
node-1	antboot, worker
node-2	ops,master,adns
0 到多个其他节点	worker

- 节点角色规则

  规则列表如下：

规则 key	是否可以忽略	说明	违反时的处理建议

antboot_node_role	否	必须有且只有 1 个 antboot 角色。	配置 1 个 antboot 角色。
antboot_locality	仅当确认为误报时	Antboot 服务没有安装在配置了 antboot 角色的节点。	确认 Antboot 是否绑定在部署所在的节点。如果确认是误报，可以忽略。
adns_antboot_mutually_exclusive	否	adns 和 antboot 不能在同一个节点，在没有开启 secure registry 时，adns 和 registry 服务 80 端口冲突。	i. 将 adns 规划到其他节点。 ii. 开启 secure registry。
no_master	否	没有配置 master 节点。	配置至少 1 个 master 节点
master_less_than_3	是	少于 3 个 master 节点。	增加 master 节点。 如果资源有限，并且悉知单个 master 节点的风险，可以忽略。
adns_placement	是	adns 和 coredns 有 53 端口冲突，请根据建议修改角色规划。	○ 一共只有 1 个节点，则不能安装 adns。 ○ 一共 3 个节点，至少有 1 个 master 节点不能同时是 adns 节点。 ○ 一共有 4 个节点或更多，至少有 2 个 master 节点不能同时是 adns 节点。

建议：

- adns 和 antboot 不在同一节点。
- master 最好有 3 节点。
- ops、adns 建议有 2 个节点。

全局配置

注意

- 红色加星号的配置是必填的，请关注。
- 红色的配置表示需要确认。

key	global.clusterName

默认值	ake
描述	集群名称，显示在 Captain 页面上，双机房部署时两个机房的此名称必须不同。
修改建议	根据客户需求修改，或者不修改。
配置示例	[global] clusterName = "ake"

dnsZone

key	global.dnsZone
默认值	antplus.com
描述	集群 DNS 域，apiserver、yunyou、captain 等会根据这个域注册域名到 ADNS。
修改建议	根据客户需求修改，或者不修改。
配置示例	[global] dnsZone = "antplus.com"

key	global.timezone
默认值	默认使用系统时区。
描述	系统时区，为数据库配置时区。
修改建议	根据客户需求修改，或者不修改。
配置示例	[global] timezone = "Asia/Shanghai"

key	global.iaasType

默认值	baremetal
描述	集群平台类型,设置该平台参数，Antboot 会安装对应平台的组件。
修改建议	重要：根据环境选择。 baremetal"：物理机与虚拟机。 ack"：公有云。 acas"：飞天 私有云。 aws"：aws。
配置示例	[global] iaasType= "baremetal"

AKE 集群配置

产品配置

安装包级别的配置：

key	**ake.serviceSubnet**
默认值	172.16.0.0/16
描述	Kubernetes 的 service 的网段。
修改建议	一般不修改。 如果需要修改，不应与 Pod、节点网段重叠。如果修改此配置，同时也需要修改 **ake.clusterDnsIP** 配置。
配置示例	[ake] serviceSubnet = "172.16.0.0/16"

key	**ake.clusterDnsIP**
默认值	172.16.0.100
描述	Kubernetes 的 DNS入口IP
修改建议	一般不修改。 如果需要修改，必须与 **ake.serviceSubnet** 网段重叠。

配置示例	[ake] clusterDnsIP= "172.16.0.100"

key	**ake.useAntbootRepoOnly**
默认值	false
描述	离线安装的情况下，仅使用 antboot yum 软件源，原有 yum 仓库源文件会移动到 `/root/repos.backup/` 中。
修改建议	一般不修改。 如果离线状态中无法正常安装软件包，需要移除系统自带 centos repo文件，否则无法安装，软件报错。
配置示例	[ake] useAntbootRepoOnly=true

key	**ake.ntpIP**
默认值	""
描述	当需要开启 NTP 同步时间功能，需要指定同步的源 IP，不指定表示不使用 NTP 功能。
修改建议	一般不修改。 当开启 NTP 时，ops1 会使用该源服务器；ops2 会使用 ops1 当源。其他节点由 ops1 及 ops2 共同当源同步时间。
配置示例	[ake] ntpIP="192.168.122.1"

key	**ake.gpuShared**
默认值	true
描述	是否开启 GPU 共享，默认开启。

修改建议	一般不修改。
配置示例	[ake] gpuShared=true

key	ake.gpuRepeated
默认值	4
描述	GPU 超卖倍数，当开启 GPU 共享时生效。
修改建议	一般不修改。
配置示例	[ake] gpuRepeated=4

Hybridnet 网络插件配置

默认使用 Hybridnet 作为 Kubernetes 的网络插件，Antboot会创建一个网络。

key	ake.hybridnet.networkType
默认值	Overlay
描述	网络模式
修改建议	参考前文 Overlay 和 Underlay 的对比，SOFA 一定要使用 underlay。
有效值	Overlay 或者 Underlay。
配置示例	[ake.hybridnet] networkType = "Overlay"

key	ake.hybridnet.interfaces *
默认值	-

描述	节点的网卡名，如果每个节点网卡名不同，用逗号分开，例如 (eth0,bond0)。
修改建议	根据集群配置，必须是已经存在的网卡。 可以使用 `ip link show` 来查看主机上有哪些网卡。
配置示例	[ake.hybridnet] interfaces = "eth0"

key	**ake.hybridnet.netID ***
默认值	-
描述	hybridnet 组网使用的 vlan/vxlan 的 net id。
修改建议	• Overlay 模式下，vxlan id 不能为 0，可填 4。 • Underlay 模式下，无须配置。见Underlay网络配置。
配置示例	[ake.hybridnet] netID = 4

key	**ake.hybridnet.cidr**
默认值	10.16.0.0/24
描述	Pod 网段。 如果使用 Overlay 模式，不应与**节点网段**重叠。
修改建议	• Overlay 模式下，如果不与主机网段重叠，可以不修改。 • Underlay 模式下，无须配置。详情请参见 Underlay网络配置。
配置示例	[ake.hybridnet] cidr = "10.16.0.0/24"

hybridnet overlay 网络配置示例：

```
[[nodes.hosts]]
ip = "192.168.10.7"
roles = [ "antboot", "master", "ops" ]

[[nodes.hosts]]
ip = "192.168.10.8"
roles = [ "ops", "master", "adns" ]

[[nodes.hosts]]
ip = "192.168.10.9"
roles = [ "master","adns"]

[ake.hybridnet]
enable = true
networkType = "Overlay"
interfaces = "eth0"
netID = 4
```

Underlay网络配置

当网络类型使用 Underlay 时需要配置 [[ake.hybridnet.networks]] 参数，ovlerlay网络无须配置。

key	[[ake.hybridnet.networks]].name
默认值	无
描述	Underlay 网络段的名称。
修改建议	尽量简短,由英文数字组成，多个网络段名称不可重复。
配置示例	[[ake.hybridnet.networks]] name="net1"

key	[[ake.hybridnet.networks]].netID
默认值	无
描述	Underlay 网络段的网络的 ID。填 0 表示不使用 vlan，而公用主网络设置可以填写701，使用 vlan 规划网络。
修改建议	必须提供。

配置示例	[[ake.hybridnet.networks]] netID= 701

key	[[ake.hybridnet.networks]].subnet
默认值	无
描述	由集群网络管理员规划的 vlan IP 范围地址，其中 IP 必须在大二层 IP，且上层交换机支持 vlan 功能。
修改建议	至少提供一组子网设置。
配置示例	[[ake.hybridnet.networks]] name="net2" netID= 701 subnet=[{cidr="192.168.3.0/24",gateway="192.168.3.1",start="192.168.3.2",end="192.168.3.20"},] cidr：网络段。 gateway：规划的上层网关。 start：开启IP（包括）。 end：结束IP（包括）。

hybridnet underlay网络配置示例：

```
[[nodes.hosts]]
ip = "192.168.10.7"
#这里net1角色表示该节点绑定net1网络
roles = [ "antboot", "master", "ops","net1" ]

[[nodes.hosts]]
ip = "192.168.10.8"
roles = [ "ops", "master", "adns","net1"  ]

[[nodes.hosts]]
ip = "192.168.10.9"
#这里net2角色表示该节点绑定第二个net2网络
roles = [ "master","adns","net2"]

[ake.hybridnet]
enable = true
#这里选择Underlay时,必须设置[[ake.hybridnet.networks]]网络
networkType = "Underlay"
interfaces = "eth0"

#配置多个网络
[[ake.hybridnet.networks]]
name="net1"
netID= 701
#子网可以支持多个网段,
subnet=[
{cidr="192.168.1.0/24",gateway="192.168.1.1",start="192.168.1.2",end="192.168.1.20"},
{cidr="192.168.2.0/24",gateway="192.168.2.1",start="192.168.2.2",end="192.168.2.20"},
]

[[ake.hybridnet.networks]]
name="net2"
netID= 701
subnet=[
{cidr="192.168.3.0/24",gateway="192.168.3.1",start="192.168.3.2",end="192.168.3.20"},
]
```

Antstack Plus 配置

产品配置

key	antstack.dbUrl
默认值	无
描述	为 AntStack Plus 指定外部数据库。如果指定该参数，AntStack Plus 的组件会使用该数据库并且不在安装 MySQL。

修改建议	可不填
配置示例	[antstack] dbUrl = "***"

key	**antstack.dbUser**
默认值	无
描述	外部数据库用户。
修改建议	可不填
配置示例	[antstack] dbUser = "user"

key	**antstack.dbPassword**
默认值	无
描述	外部数据库用户对应的密码，密码不要携带 `:&$` 符号。
修改建议	可不填
配置示例	[antstack] dbPassword = "***"

key	**antstack.dbPort**
默认值	3306
描述	外部数据库端口。
修改建议	可不填

配置示例	[antstack] dbPort = "3306"

key	**antstack.storageClass**
默认值	open-local-lvm
描述	antstack 产品使用的存储类型。公有云、私有云安装环境时。必须根据集群规格提供正确的 storageClass。使用 openlocal 存储的，这里可以不用填写。
修改建议	可不填
配置示例	[antstack] storageClass= "open-local-lvm"

mysql

如果没有提供 dbURL 等数据库配置，Antboot 会安装一个主备高可用的 MySQL 集群。

key	**antstack.mysql.rootPassword**
默认值	无
描述	为 Antboot 部署的 MySQL 指定密码，密码不要携带 `:&$` 符号。
修改建议	可不填
配置示例	[antstack.mysql] rootPassword = "***"

key	**antstack.mysql.storageSize**
默认值	5 GB
描述	默认5 GB。 共有云环境由于阿里云最小磁盘限制至少分配 20 GB。

修改建议	建议根据需求修改
配置示例	[antstack.mysql] storageSize="10 GB"

adns

key	antstack.adns.forwardDNSServer
默认值	-
描述	转发的 dns server。如果没有则不填。
修改建议	根据客户需求，如果客户提供上游 dns server 则可以配置；也可以参考主机上的 `/etc/resovle.conf` ，多个地址用英文逗号（,）隔开。
配置示例	[antstack.adns] forwardDNSServer = "10.16.0.1,10.16.0.2"

key	**antstack.adns.masterVips**
默认值	默认无配置，表示不开启 VIP。
描述	如果使用 vip，master 的 vip 用英文逗号（,）分割，例如(192.168.10.20,192.168.10.21)。如果填写，则 **antstack.adns.slaveVips** 也需要填写。在 ECS 场景不需要填写，在 物理机、Vmware 场景需要填写，且 vip 需要与主机网卡处于同一网络下。
修改建议	
配置示例	[antstack.adns] masterVips = "192.168.10.20,192.168.10.21"

key	**antstack.adns.slaveVips**
默认值	默认无配置，表示不开启 VIP。

描述	如果使用 vip，slave 的 vip 用英文逗号（,）分割，例如(192.168.10.20,192.168.10.21)。如果填写，则 **antstack.adns.slaveVips** 也需要填写。在 ECS 场景不需要填写，在 物理机、Vmware 场景需要填写，且 vip 需要与主机网卡处于同一网络下。
修改建议	
配置示例	[antstack.adns] slaveVips = "192.168.10.20,192.168.10.21"

key	**antstack.adns.interface**
默认值	默认无配置，表示不开启 VIP
描述	节点绑定 vip 的网卡名称。如果配置了 **antstack.adns.masterVips** 和 **antstack.adns.slaveVips**，需要配置该参数。
修改建议	
配置示例	[antstack.adns] interface = "eth0"

minio（可选）

注意

如果安装 prometheus 则建议开启 minio。

key	**antstack.minio.enable**
默认值	true
描述	是否安装 minio，默认安装。如需关闭请改为 fasle（注意不要加引号）。
修改建议	根据资源和需求配置。

配置示例	[antstack.minio] enable = false

key	**antstack.minio.minioAccessKey**
默认值	minio
描述	minio API 访问账户，默认 minio。
修改建议	
配置示例	[antstack.minio] minioAccessKey = "minio"

key	**antstack.minio.minioSecretKey**
默认值	awesome-minio
描述	minio API 访问密钥，默认为 awesome-minio，至少 8 位。
修改建议	建议根据需求修改。
配置示例	[antstack.minio] minioSecretKey = "awesome-minio"

key	**antstack.minio.storage**
默认值	100Gi
描述	每个 minio 实例使用的存储到大小。
修改建议	建议根据需求修改。

配置示例	[antstack.minio] storage = "100GiB"

key	**global.dnsZone**
默认值	antplus.net
描述	集群 DNS 域，apiserver、yunyou、captain 等会根据这个域注册域名到 ADNS。 同时也是 minio 子域名，会渲染到 minio 环境参数 MINIO_DOMAIN 中。 渲染结果：MINIO_DOMAIN = minio.antplus.net (默认)。
修改建议	根据客户需求修改，或者不修改。
配置示例	[global] dnsZone = "antplus.net"

prometheus （可选）

默认安装 prometheus stack 和 grafana，可以通过 enable 开关关闭。

key	**antstack.prometheus.enable**
默认值	true
描述	是否安装 prometheus stack 和 grafana，默认安装。如需关闭请改为 fasle（注意不要加引号）。
修改建议	根据资源和需求配置。
配置示例	[antstack.prometheus] enable = false

key	**antstack.prometheus.prometheusStorage**
默认值	250GiB

描述	prometheus 每个实例需要的存储空间，默认 250 GiB 共 3 个实例需求，即需要 750 GiB。
修改建议	根据集群资源配置
配置示例	[antstack.prometheus] prometheusStorage = "250GiB"

提交配置

在 `antboot edit config` 开启的 vim session 中编辑完成后，直接保存退出即可。完成后的配置示例如下：

```
[[nodes.common.vg]]
name = "yoda-ssd"
pvDisks = [ "/dev/vdb1" ]

[nodes.common.ssh]
port = 22
user = "root"
privateKeyFile = "/root/.ssh/id_rsa"

[[nodes.hosts]]
ip = "10.0.0.1"
roles = [ "antboot", "worker", "ops" ]

[[nodes.hosts]]
ip = "10.0.0.2"
roles = [ "ops", "worker", "adns" ]

[[nodes.hosts]]
ip = "10.0.0.3"
roles = [ "master" ]

[global]

[ake.hybridnet]
networkType = "Overlay"
interfaces = "eth0"
netID = 4

[antstack.adns]
forwardDNSServer = "10.10.0.1"
interface = "eth0"
```

您可以使用 `--normalize` 选项检查配置，Antboot 会解析 TOML 配置，并进行格式化和标准化校验，输出的结果为 YAML 格式的完整配置，这份配置将会用于安装。完整命令如下：

```
antboot get config --normalize
```

4.1.5. 部署 AKEPlus 和 AntStack Plus

部署测试（可选）

部署前可以用 `dryrun` 选项进行测试，每个任务会渲染各自的执行参数，并进行校验：

```
antboot install --dryrun --all
```

如果一切正常，应如下图输出所示，输出都是 ok。

```
  task is in pending state, and will be executed

[ok] AntStack-Plus-1.11.0.after-deploy:kruise
Explain:
  task is in pending state, and will be executed

[ok] AntStack-Plus-1.11.0.after-deploy:scheduling-insight
Explain:
  task is in pending state, and will be executed

[ok] AntStack-Plus-1.11.0.Install:AntStack-Plus-1.11.0.after-deploy
Explain:
  this is a workflow type

[ok] AntStack-Plus-1.11.0.install-yunyou:yunyou-args-reader
Explain:
  task is in pending state, and will be executed

[ok] AntStack-Plus-1.11.0.install-yunyou:apyunqing
Explain:
  task is in pending state, and will be executed

[ok] AntStack-Plus-1.11.0.Install:AntStack-Plus-1.11.0.install-yunyou
Explain:
  this is a workflow type

[ok] AntBootWorkflow.Install:AntStack-Plus-1.11.0.Install
Explain:
  this is a workflow type


[root@i39f13400.sqa.eu95 /root]
```

部署步骤

1. 安装所有已经加载的包。

   ```
   antboot install --all
   ```

   `antboot install` 仅发送安装命令，安装流程是在 antbootd（Antboot 服务端）执行。

   `antboot install` 执行后会自动从服务端请求日志。您可以通过 `Ctrl + C` 中断日志显示，这并不会停止安装。如果需要停止安装，可以执行 `antboot cancel` 。

   如果部署报错后中断，可以再次执行 `antboot install --all` ，部分任务会自动重试。

2. 查看部署进度。

您可以通过以下命令查看部署进度：

- 查看正在执行的 workflow 日志

```
antboot logs wotkflow current
```

- 查看正在执行的 workflow 状态

```
antboot get workflow current
```

当所有任务的 State 状态都是 ready 时，即表示安装完成。

```
  ├─ add-worker                    AKE-Plus-2.1.0.install    add_cluster_worker    ready
  ├─ remove-default-cni-conf       AKE-Plus-2.1.0.install    exec                  ready
  ├─ add-kubeconfig-to-ftp         AKE-Plus-2.1.0.install    exec                  ready
  ├─ install-kubeconfig            AKE-Plus-2.1.0.install    exec                  ready
  ├─ restart-hybridnet-daemon      AKE-Plus-2.1.0.install    exec                  ready
  ├─ render-underlay-network       AKE-Plus-2.1.0.install    render-gotmpl         ready
  ├─ deploy-underlay-network       AKE-Plus-2.1.0.install    exec                  ready
  └─ open-local                    AKE-Plus-2.1.0.install    helm                  ready
Name:        AntStack-Plus-1.11.0.Install
State:       --
Progress:    3/3
Tasks:
  Name                                      ProcessName                          TaskType          State
  * AntStack-Plus-1.11.0.antstack           AntStack-Plus-1.11.0.Install         workflow          ready
  ├─ load-antstack-images                   AntStack-Plus-1.11.0.antstack        exec              ready
  ├─ minio                                  AntStack-Plus-1.11.0.antstack        helm              ready
  ├─ apply-antstack-namespace               AntStack-Plus-1.11.0.antstack        kube-manifest     ready
  ├─ write-db-info                          AntStack-Plus-1.11.0.antstack        kube-manifest     ready
  ├─ adns                                   AntStack-Plus-1.11.0.antstack        adns              ready
  ├─ update-dns-setting                     AntStack-Plus-1.11.0.antstack        exec              ready
  ├─ restart-coredns                        AntStack-Plus-1.11.0.antstack        exec              ready
  └─ captain                                AntStack-Plus-1.11.0.antstack        helm              ready
  * AntStack-Plus-1.11.0.after-deploy       AntStack-Plus-1.11.0.Install         workflow          ready
  ├─ import-cluster                         AntStack-Plus-1.11.0.after-deploy    import-cluster    ready
  ├─ cert-manager                           AntStack-Plus-1.11.0.after-deploy    helm              ready
  ├─ kruise                                 AntStack-Plus-1.11.0.after-deploy    helm              ready
  └─ scheduling-insight                     AntStack-Plus-1.11.0.after-deploy    helm              ready
  * AntStack-Plus-1.11.0.install-yunyou     AntStack-Plus-1.11.0.Install         workflow          ready
  ├─ yunyou-args-reader                     AntStack-Plus-1.11.0.install-yunyou  yunyou_args       ready
  └─ apyunqing                              AntStack-Plus-1.11.0.install-yunyou  helm              ready

[root@i39f13400.sqa.eu95 /root]
```

日志也会输出“workflow execute successfully”：

```
ace=default, status=deployed,
[DONE] AntStack-Plus-1.11.0.after-deploy/kruise 14:58:48.215 task exited: ready
[RUN] AntStack-Plus-1.11.0.after-deploy/scheduling-insight 14:58:48.215 task is in pending state, and will be executed
AntStack-Plus-1.11.0.after-deploy/scheduling-insight 14:58:48.216 install helm release; chart=scheduling-insight, vers
.1.0,
AntStack-Plus-1.11.0.after-deploy/scheduling-insight 14:58:54.698 Pod[kube-system/scheduling-insight-88b46cd56-n6fsx]
ready in the running state; chart=scheduling-insight, version=1.1.0, rule=Deployment[kube-system/scheduling-insight],
ondition=allReady,
AntStack-Plus-1.11.0.after-deploy/scheduling-insight 14:58:54.698 Install complete; chart=scheduling-insight, version=
, release=scheduling-insight, namespace=kube-system, status=deployed,
[DONE] AntStack-Plus-1.11.0.after-deploy/scheduling-insight 14:58:54.699 task exited: ready
AntStack-Plus-1.11.0.after-deploy 14:58:54.699 workflow execute successfully;
[DONE] AntStack-Plus-1.11.0.Install/AntStack-Plus-1.11.0.after-deploy 14:58:54.700 task exited: ready
[RUN] AntStack-Plus-1.11.0.Install/AntStack-Plus-1.11.0.install-yunyou 14:58:54.700 this is a workflow type
[RUN] AntStack-Plus-1.11.0.install-yunyou/yunyou-args-reader 14:58:54.702 task is in pending state, and will be execute
[DONE] AntStack-Plus-1.11.0.install-yunyou/yunyou-args-reader 14:58:54.721 task exited: ready
[RUN] AntStack-Plus-1.11.0.install-yunyou/apyunqing 14:58:54.721 task is in pending state, and will be executed
AntStack-Plus-1.11.0.install-yunyou/apyunqing 14:58:54.721 install helm release; chart=apyunqing, version=1.9.0,
AntStack-Plus-1.11.0.install-yunyou/apyunqing 15:09:14.899 Pod[antstack/apyunqing-0] is already in the running state;
StatefulSet[antstack/apyunqing], stopCondition=allReady, chart=apyunqing, version=1.9.0,
AntStack-Plus-1.11.0.install-yunyou/apyunqing 15:09:14.899 Pod[antstack/apyunqing-1] is already in the running state;
=apyunqing, stopCondition=allReady, rule=StatefulSet[antstack/apyunqing], version=1.9.0,
AntStack-Plus-1.11.0.install-yunyou/apyunqing 15:09:14.899 Install complete; chart=apyunqing, version=1.9.0, release=a
ing, status=deployed, namespace=antstack,
[DONE] AntStack-Plus-1.11.0.install-yunyou/apyunqing 15:09:14.900 task exited: ready
AntStack-Plus-1.11.0.install-yunyou 15:09:14.900 workflow execute successfully;
[DONE] AntStack-Plus-1.11.0.Install/AntStack-Plus-1.11.0.install-yunyou 15:09:14.901 task exited: ready
AntStack-Plus-1.11.0.Install 15:09:14.901 workflow execute successfully;
[DONE] AntBootWorkflow.Install/AntStack-Plus-1.11.0.Install 15:09:14.904 task exited: ready
AntBootWorkflow.Install 15:09:14.904 workflow execute successfully;
```

- 查看 package 的 State

```
$ antboot get package
Name        Version     Code     State        Selected
AKE-Plus    1.4.1       ake      installed    true
```

package 的 State 应为 installed。

- 查看除 sigma 相关 Pod 外的其他 Pod 状态

```
kubectl get po -A | grep -vP "sigma-cluster|sigma-machine-operator"
```

其他操作命令

- 停止安装

```
antboot cancel
# 或者
antboot stop workflow current
```

该命令会停止正在执行的所有的 workflow，正在运行中的任务会收到取消的通知。

- 继续安装

```
antboot install --all
```

该命令会跳过已经执行完成的任务，但是如果某个任务的配置被修改了，它也会被重新执行（即使已经完成）。

4.1.6. 后续操作

添加 registry-zone

```
#此脚本在ags节点/opt/ags/adns/zone.sh
#cat dns-zone.yaml

curl -H 'Content-Type: application/json' -d '{
"name": "'$1'",
"defaultTTL": 600,
"description": "'$1' test zone",
"type": "hosted",
"soa": {
"mname": "master.'$1'",
"rname": "hostmaster.'$1'",
"serial": 1,
"refresh": 3600,
"retry": 1800,
"expire": 604800,
"minimumTTL": 60
},
"nsRRs": [{
"name": "'$1'.",
"type": "NS",
"ttl": 60,
"value": "ns1.'$1'."
}],
"glueRRs": [{
"name": "ns1.'$1'.",
"type": "A",
"ttl": 60,
"value": "1.1.1.1"
}]
}
' http://100.**.**.65:80/zone/addzone
##替换 100.**.**.65 为 adns vip

sh  dns-zone.yaml alipay.com
```

```
##cat dns.sh
#!/bin/sh
if [ ! $# -eq 3 ];then
        echo "Usage:$0 add <add|remove> name ip"
        exit 1
fi
action=$1
name=$2
ip=$3
curl "http://100.**.**.65:80/cgi-bin/dnsapi.cgi?action=$action&name=$name&ip=$ip"
##替换 100.**.**.65 为 adns vip
sh  dns.sh add acs-reg.alipay.com  ip

#注释所有节点 hosts 文件中 acs-reg.alipay.com
```

Captain 登录

Captain 部署完成后默认部署一个 NodePort Service，用于访问服务。查看 NodePort 命令如下：

```
$ kubectl get svc -n antstack -l "app.kubernetes.io/name=captain"
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP    PORT(S)         AGE
captain             ClusterIP   172.**.**.165   <none>         82/TCP          25h
captain-nodeport    NodePort    172.**.**.45    <none>         82:30082/TCP    25h
```

查看其中 TYPE 为 NodePort 记录的 82 端口的主机映射端口，示例中为 30082。打开浏览器访问：`<主机IP>:<node-port>`，其中主机 IP 可以是任意一台机群主机的 IP 地址，`node-port` 是上面查到的端口。例如：`http://10.0.0.1:30082`。

通过 Node ip+82端口访问（推荐）

```
kubectl get po -A  -owide |grep  captain
```

```
[root@i39f13400.sqa.eu95 /root]
#kubectl get po -A  -owide |grep  captain
antstack        captain-0                                  1/1     Running     0          46m    11.[redacted].193
  11.[redacted].193   <none>           <none>
antstack        captain-1                                  1/1     Running     0          44m    100.[redacted].65
  100.[redacted].65   <none>           <none>
antstack        captain-dbinit--1-rfzmz                    0/1     Completed   0          47m    11.[redacted].27
  11.[redacted].27    <none>           <none>
```

云游登录

云游要求必须通过域名来访问，域名格式是：`yunyou.<global.dnsZone>`。其中 `global.dnsZone` 是您在安装时配置的域名。如果没有配置，则是默认值 `antplus.com`。登录默认用户名和密码为：`antcloudAdmin/antcloudYunyou`。

1. 绑定运用域名。

   您需要通过 hosts 文件绑定，将任意一台集群主机的 IP 绑定到域名即可，例如：

   ```
   echo "10.0.0.1 yunyou.antplus.com" >> /etc/hsots
   ```

2. 查找 NodePort。

   Yunyou 部署完成后默认部署一个 NodePort Service，用于访问服务。查看 NodePort 命令如下：

   ```
   $ kubectl get svc -n antstack -l "app.kubernetes.io/name=apyunqing"
   NAME                TYPE        CLUSTER-IP      EXTERNAL-IP    PORT(S)         AGE
   apyunqing           ClusterIP   172.**.**.210   <none>         81/TCP          25h
   apyunqing-nodeport  NodePort    172.**.**.83    <none>         81:30081/TCP    25h
   ```

   查看其中 TYPE 为 NodePort 记录的 81 端口的主机映射端口，示例中为 30081。打开浏览器访问：`<域名>:<node-port>`，其中域名是步骤 1 绑定的域名，`node-port` 是上面查到的端口。比如：`http://yunyou.antplus.com:30081`

通过 Node ip+81端口访问（推荐）

```
kubectl get po -A  -owide |grep  apyunqing
```

```
[root@i39f13400.sqa.eu95 /root]
#kubectl get po -A  -owide |grep  apyunqing
antstack        apyunqing-0                                  1/1     Running     0          26m    11.[redacted].193
  11.[redacted].193    <none>           <none>
antstack        apyunqing-1                                  1/1     Running     0          24m    100.[redacted].65
  100.[redacted].65    <none>           <none>
antstack        apyunqing-dbinit--1-m4wlp                    0/1     Completed   0          32m    100.[redacted].200
  100.[redacted].200   <none>           <none>

[root@i39f13400.sqa.eu95 /root]
```

4.1.7. 附录：端口汇总

- Kubelet Node Port 端口范围：30000~32767。
- 底座服务监听端口：

服务	主机端口
云游	81、7001
ADNS	53、54、80
dns-agent	15353、15354
Captain	82、8080
apiserver	6443
ETCD	2379、2380
Kubelet	10248、10250、10255
kube-proxy	10249
kube-schedule	10251
kube-controller	10252
registry	80
containerd	36838
coredns	53、8080

服务	主机端口
hybridnet	11021、9898

- LB 端口

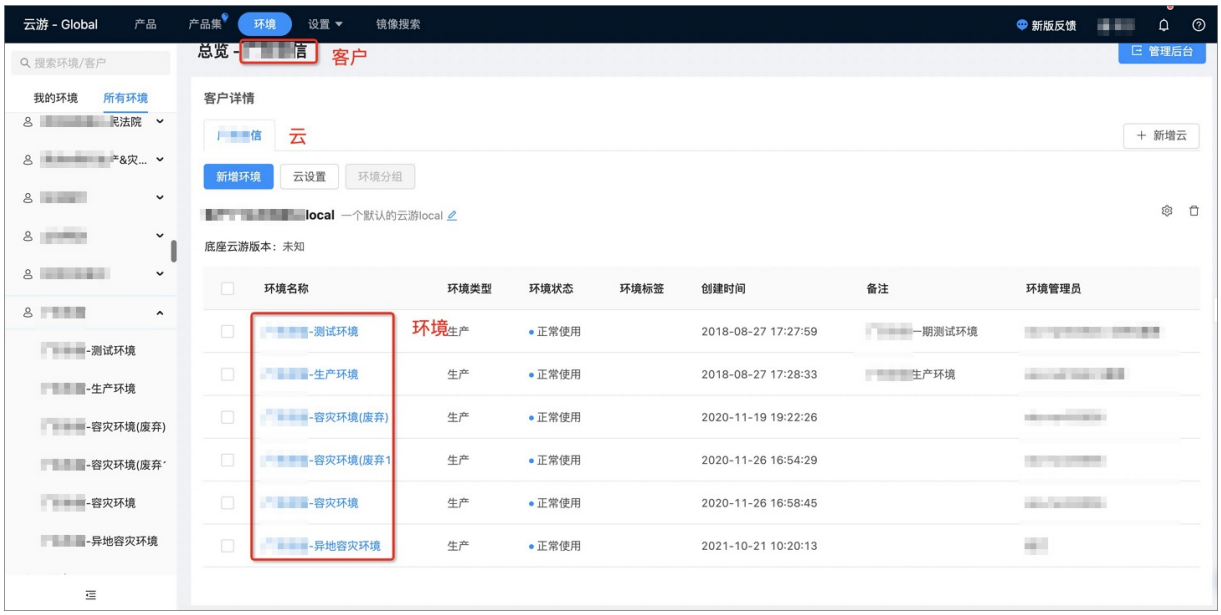
产品	ADNS 中域名	ADNS 中 IP	LB 后端 IP	主机端口（LB 后端监听端口）	LB 前端端口
云游 Local	yunyou.*.*.net	LB 地址	任意两台集群主机 IP	apyunqing 的 nodeport	81、7001
CaptainPlus	captain.*.*.net	LB 地址	任意两台集群主机 IP	Captain 的 nodeport	82、7001
AKE3 kube-apiserver	apiserver.*.*.net	LB 地址	三台 master 的 IP	6443	6443
IAM	iam.*.*.net admin.*.*.net	LB 地址	任意两台集群主机 IP	aciamweb 的 nodeport	80
OP	op.*.*.net	LB 地址	任意两台集群主机 IP	acprodgw 的 nodeport	80

4.2. 云游 Global 创建环境

4.2.1. 在云游 Global 上新建客户、云和环境

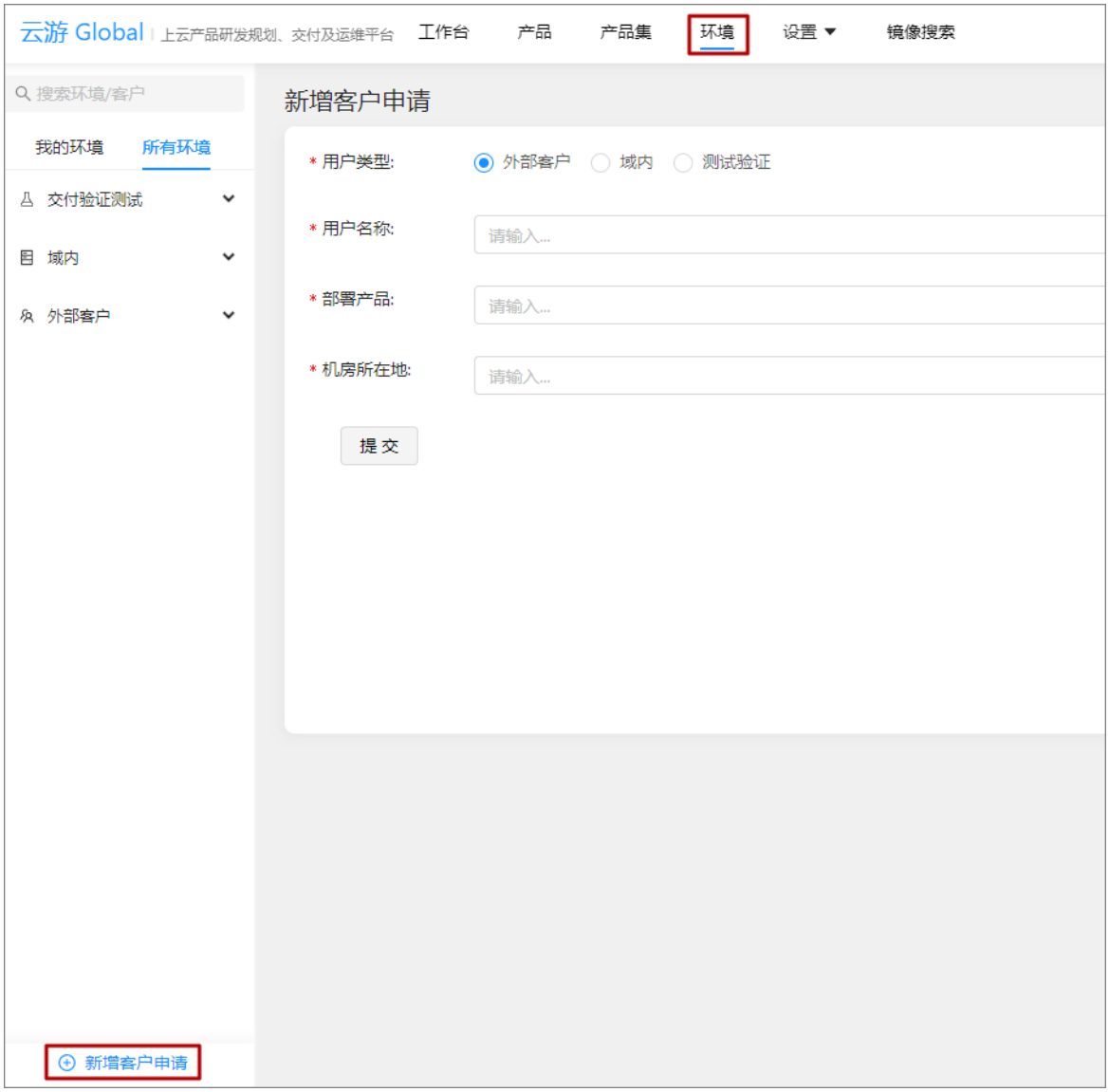
客户、云、环境是从左到右一对多的关系，一个客户允许拥有多个云游 Local，一个云游 Local 可以管理多个环境。本文介绍如何在云游 Global 上新建客户、云和环境。

下面的总览图展示了三者的关系：



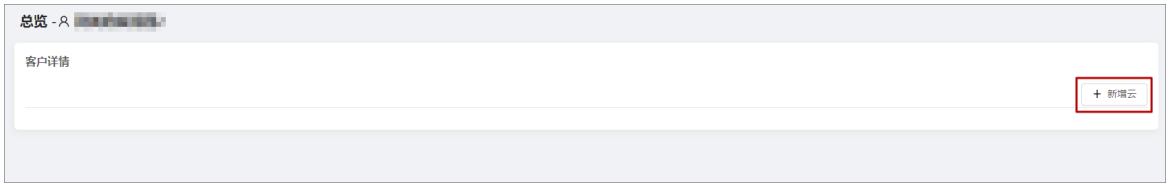
新建客户

1. 登录云游 Global。
2. 单击上方菜单栏的环境，然后在左侧导航栏单击 **新增客户申请**。
3. 输入客户信息后单击 **提交**。



新增云

1. 在云游 Global 左侧导航栏单击刚刚创建的客户名称。
2. 单击 **新增云**。



3. 填写 **云名称** 和 **云标识** 后，单击 **提交审批**。

> **说明**
>
> 一个客户可以新增多个云。

新增环境

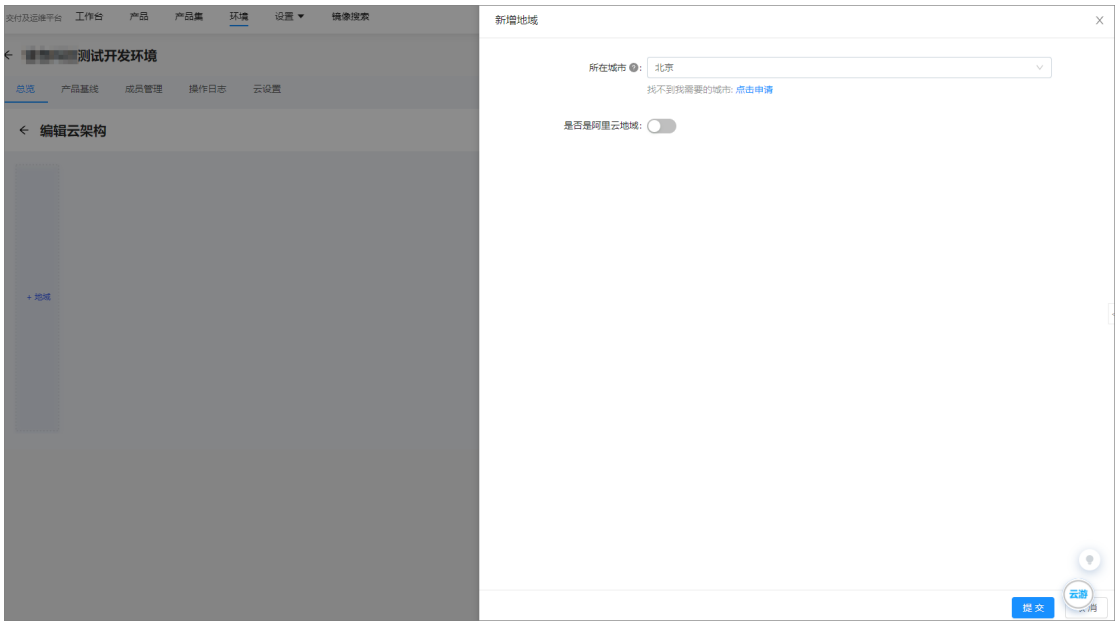
1. 添加机房信息。

   新建环境时，请先在云设置中添加对应的地域机房等信息。目前 AntStack Plus 中没有太多使用到云架构中的内容，填写只为了元数据记录以及保障流程正确，您只需简单填写一下机房和地域信息即可。

   i. 在客户详情中，单击 **云设置**。

   ii. 在 **总览** 页签单击 **编辑云架构**。

iii. 点击 **+地域**，然后编辑地域信息后单击 **提交**。



2. 新增环境。

   i. 在客户详情中单击 **新增环境**。

   ii. 按流程填写环境信息后，单击 **提交审批**。

> **说明**
>
> 自 2020 年 10 月开始，新增的环境基本都是 Plus 环境。除少数 AKE 1.x、2.x 版本的历史站点外，Plus 环境请选择 **单机房**，资源类型可以直接确认。

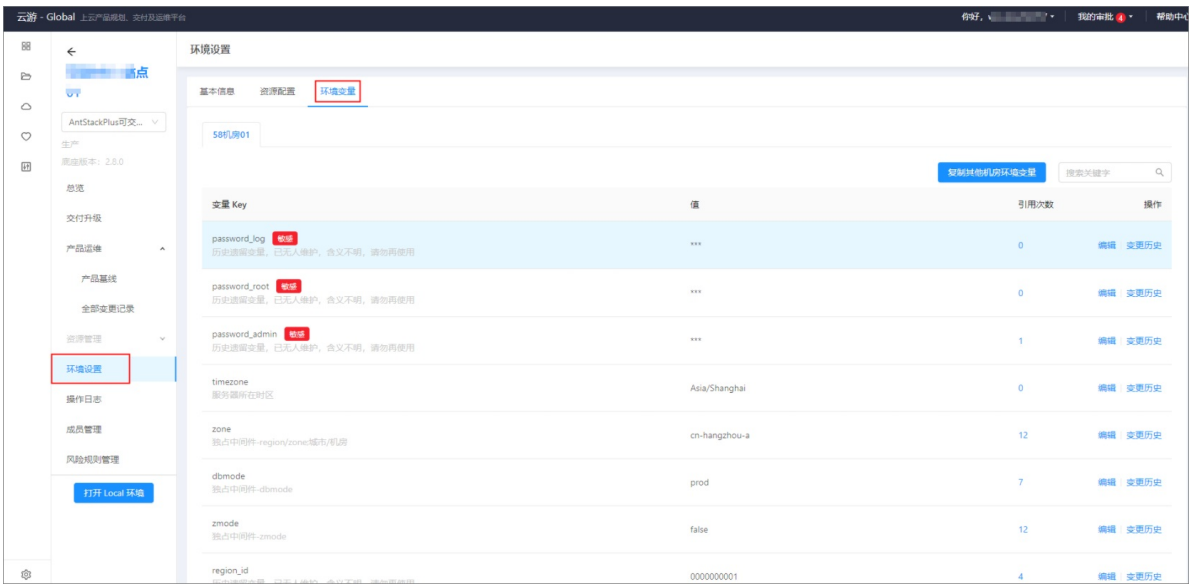
4.2.2. Global 修改环境设置

准备工作

- 环境添加完成以后，对于使用 Plus 底座的环境一定要打上标签。某些使用持久化存储的产品会进行导出解决方案的适配。

请管理员联系技术支持处理。

- 云游导出解决方案时，部分产品依赖一些环境变量，您需要在云游 Global 的 **环境设置** > **环境变量** 中填写好环境变量以后，再导出解决方案。



您可以参考下方表格填写环境变量：

key	value	备注
cell	default	双机房部署需要修改，单机房保持默认。 双机房示例：defaultA、defaultB

key	value	备注
cloudAccessKey	${中枢账号的 AK}	当前站点的 AccessKey。 本演示环境设置为：antCloudAdmin
cloudAccessSecret	${中枢账号的 AKS}	当前站点的 AccessSecret。 本演示环境设置为：antCloud123
com_alipay_env	shared	是否共享中间件。 可选值：shared、unshared
com_alipay_instanceid	000001	中间件实例 ID。
com_antcloud_antvip_endpoint	${acvip-inner 的域名}	中间件 Endpoint。 当前演示环境设置为：acvip-inner.antcloud.net
currentEnv	AntStack	当前环境标识，名称自定义。 该演示环境设置为：AntStack
datacenter	default	共享中间件-当前机房。双机房部署需要修改，单机房保持默认。 双机房示例：cn-shenzhencn-wuhan
dbmode	prod	独占中间件-dbmode。
deploy_obproxy	false	控制 obproxy 是否启动。
deploy_type	private	站点的部署模式（私有云或公有云）。 枚举值: private、public
domain	antplus.com	填写为自身实际环境的域名。
domainname	antplus.com	填写为自身实际环境的域名。
iaas_auth_enable	false	历史遗留变量，已无人维护，含义不明。 当前环境设置为：false

key	value	备注
iaas_type	AntStack	IaaS 类型，区别大小写。
keyseed	bk9KOXNhclByM1h5ZEphWVNM********wRTRGMTA2NzcxQjc4OEVFNjU1QUVBMjQzMERERjYwNTgzODQxRjI0NUU2OTQ1QQ==	说明 可暂时不填写，待解决方案导入到云游 Local 以后，在云游 Local 生成后回填到云游 Global 中。生成方式见下方说明。
log_level	info	无
obproxy_config_server_url	http://{OB 域名或 IP}/services?Action=GetObProxyConfig&User_ID=alibaba-inc&UID=ocpmaster	obproxy 启动时依赖的 OCP config url。
parent_domain	antplus.com	填写为自身实际环境的域名。
password_admin	abcABC123admin	历史遗留变量，保持默认即可。
password_log	paas123456	历史遗留变量，保持默认即可。
password_root	abcABC123root	历史遗留变量，保持默认即可。
protocol	http://	保持默认值，直接确认即可。
region_id	0000000001	Region ID 信息。
seed	M0M0QUE4NDk3QzdFREYyM********RDlBMDM1MzEwMDFGMjg2QzU1RUYxMTcxNjJGRkM0MDlDQ0Y0RDI2OQ==	说明 可暂时不填写，待解决方案导入到云游 Local 以后，在云游 Local 生成后回填到云游 Global 中。生成方式见下方说明。
site	antfin	站点信息。

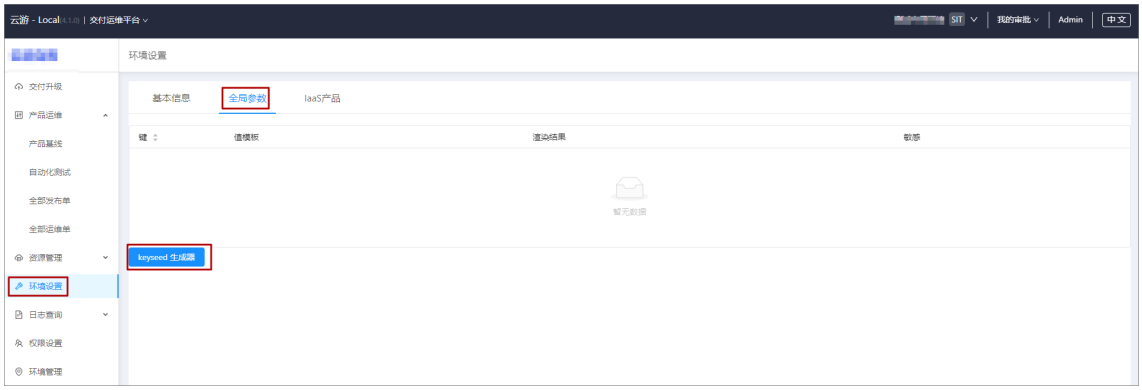
key	value	备注
timezone	Asia/Shanghai	国内站点默认即可。
TZ	Asia/Shanghai	云游 2.8.10 支持，解决监控国际化问题。
yunyou_access_key	yunyou_access_key	历史遗留变量，保持默认即可。
yunyou_access_secret	yunyou_access_secret	历史遗留变量，保持默认即可。
zmode	false	独占中间件-zmode。
zone	cn-hangzhou-a	双机房部署需要修改，单机房保持默认。 双机房示例：gz00agz00b

物理机的 AccessKey、AccessSecret 需要使用 Minio 的 AK、SK：

keyseed 和 seed 生成步骤：

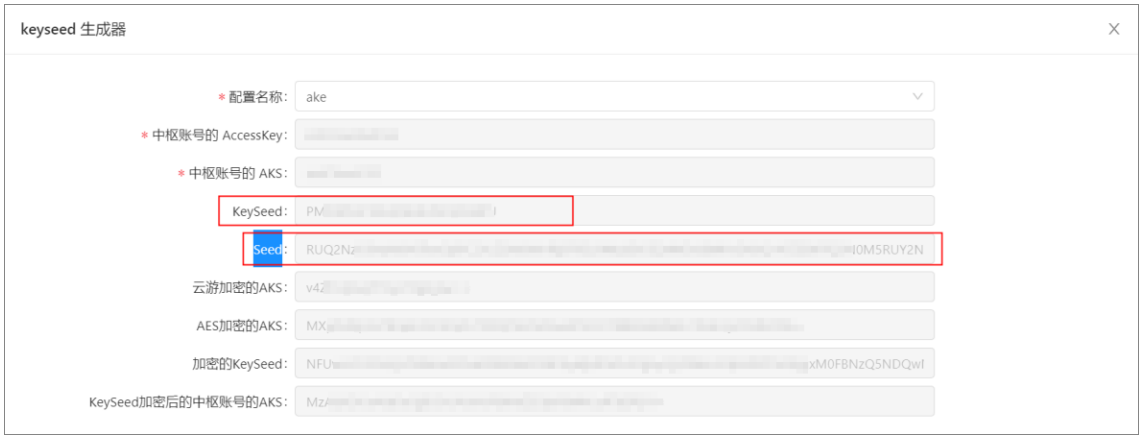
i. 登录云游 Local。

ii. 单击 **环境设置**，然后在 **全局参数** 页签单击 **keyseed 生成器**。



iii. 在 **keyseed 生成器** 对话框填写配置名称、中枢账号的 AK 和 AKS 后，单击 **获取加密参数**。

在加密参数中即可获得 keyseed 和 seed。

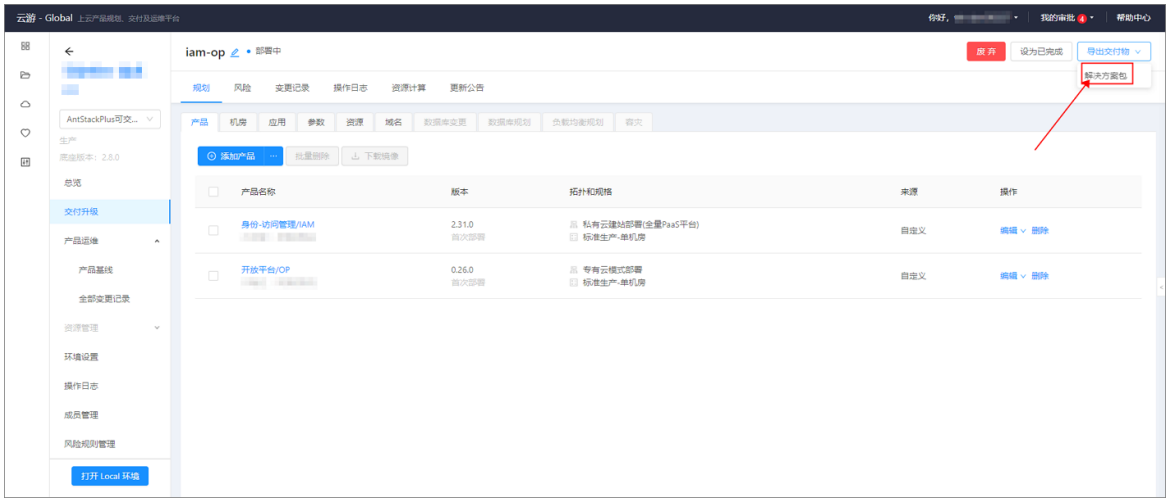


4.3. 云游 Local 环境初始化

4.3.1. 新增环境

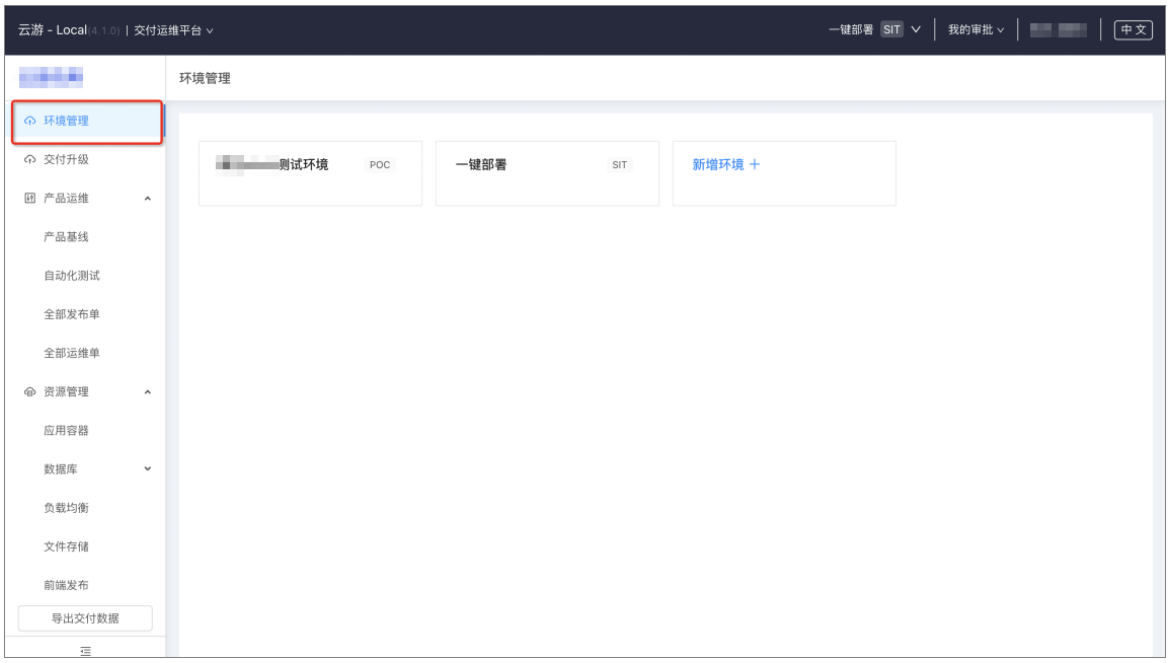
操作步骤

1. 在云游 Global 新建解决方案，添加产品后，导出解决方案。



2. 登录云游 Local 控制台。

3. 在左侧导航栏单击 **环境管理**，然后单击 **新增环境**。



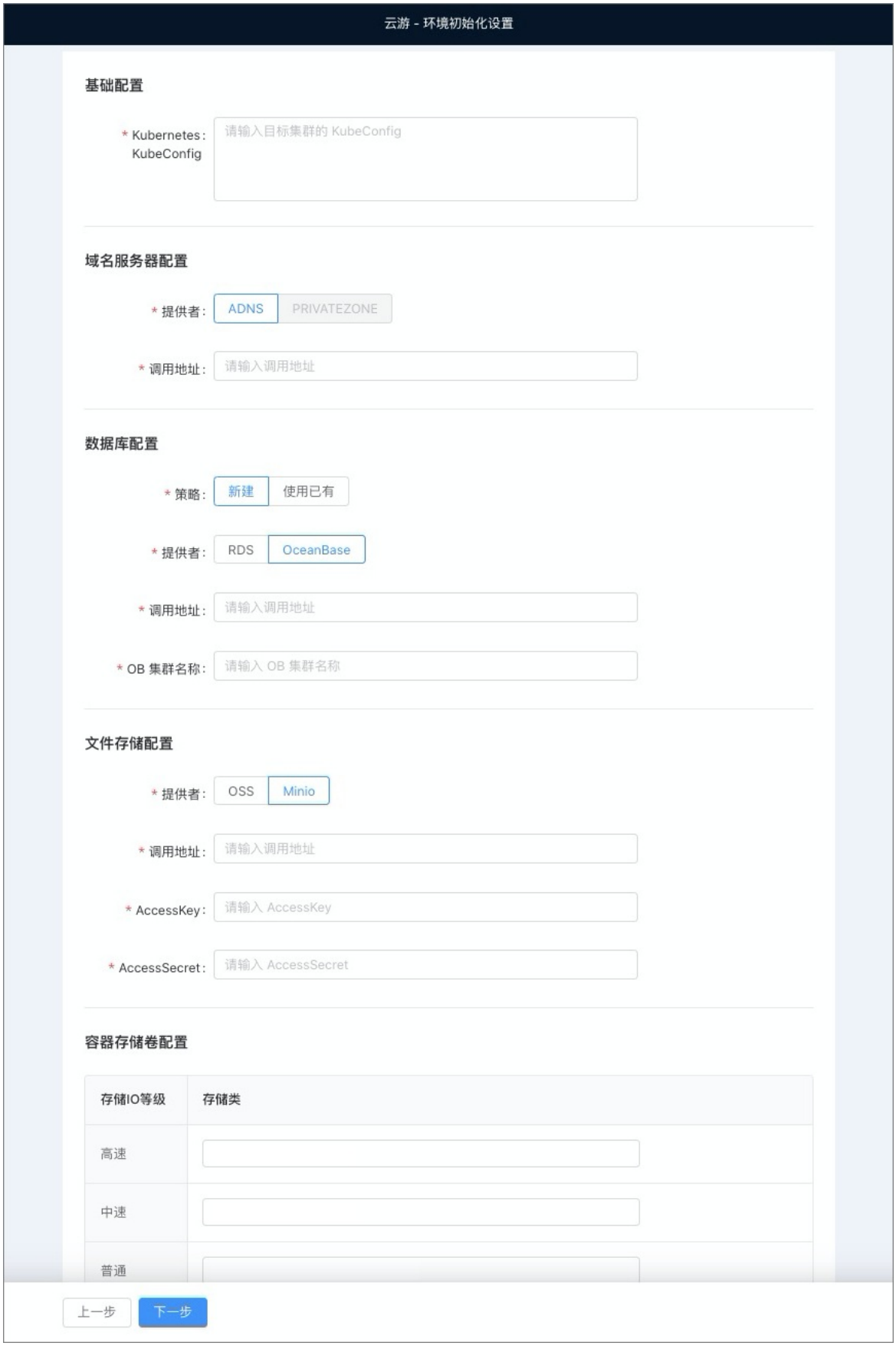
4. 在 **导入解决方案** 步骤上传步骤 1 导出的解决方案，然后单击 **下一步**。
5. 在 **环境基本信息** 步骤填写 **基本配置** 和 **底座配置**，然后单击 **下一步**。

云游 - 环境初始化设置
导入解决方案
2 环境基本信息
3 环境资源配置
4 环境初始化确认
基本信息
名称：
客户：蚂蚁金融云
站点：
是否带线：否
环境类型：POC
基本配置
* 资源名前缀：
所有创建的容器/数据库实例等的前缀
* 全局域名：
请输入全局域名，例：alipey.net
* 访问地址：
http://
t:81
底座配置
* 选择底座：
阿里云（公有云）
阿里云（专有云）
第三方
* 地域(Region)：
请输入地域，例：cn-hangzhou
* 可用区(ZONE)：
请输入可用区，例：cn-hangzhou-a
上一步
下一步

参数	说明

基本配置	**资源名前缀**	填写环境的唯一标示，多机房下该字段不可重复。 一般填写用户环境的简称。
	全局域名	当前环境产品对外提供访问的域名，双机房下使用同一套。 您可以通过 Antboot 查询： `antboot get config --normalize | grep dnsZone`
	访问地址	填写云游 Local 的访问地址。 建议使用默认值。访问云游时建议是通过 81 端口访问，这里带出来的端口即 81。
底座配置	**选择底座**	选择 **第三方**。
	地域 和 **可用区**	根据实际部署架构填写 Region 和 Zone，会用作 SOFA 应用流量地域收敛及机房收敛的依据。 同城双机房下，两个环境应地域相同，可用区不同。

6. 在 **环境资源** 配置步骤填写 Kubeconfig 及各资源配置后，单击 **下一步**。


云游 - 环境初始化设置
基础配置
* Kubernetes: KubeConfig
请输入目标集群的 KubeConfig
域名服务器配置
* 提供者:
ADNS
PRIVATEZONE
* 调用地址:
请输入调用地址
数据库配置
* 策略:
新建
使用已有
* 提供者:
RDS
OceanBase
* 调用地址:
请输入调用地址
* OB 集群名称:
请输入 OB 集群名称
文件存储配置
* 提供者:
OSS
Minio
* 调用地址:
请输入调用地址
* AccessKey:
请输入 AccessKey
* AccessSecret:
请输入 AccessSecret
容器存储卷配置
存储IO等级
存储类
高速
中速
普通
上一步
下一步

参数		说明
基础配置	**Kubernetes KubeConfig**	填写 ACK 集群的 Kubeconfig。 在集群上执行 `kubectl config view --raw` 复制输出的内容。
域名服务器配置	**提供者**	需选择 **ADNS**。
	调用地址	配置 ADNS 的地址。 如果配置了 VIP，填写 masterVips 的地址的其中一个 IP。您可以通过 `antboot get config --normalize | grep masterVips` 命令查询；若未配置 VIP，则填写规划了 adns 角色的一个节点的 IP。您可以通过 `antboot get config` 命令查看哪个节点有 adns 这个角色。
数据库配置	**策略**	对于多机房场景，第一个机房一般选 **新建**，后续机房选择 **使用已有**，并复用第一个机房的连接信息。
	提供者	选择 **OceanBase**。
	调用地址	填写为产品规划的 OceanBase 数据库的地址，没有可以不修改。
	OB 集群名称	填写 OceanBase 的集群名称。
文件存储配置	**提供者**	需选择 **Minio**。
	调用地址	如果没有安装 Minio，则填写 `http://127.0.0.1`，否则填写 Minio 的服务地址。

	AccessKey 和 AccessSecret	填写访问 Minio 的 AccessKey 和 AccessSecret。
		您可以通过
		`antboot get config --normalize | grep minioAccessKey`
		和
		`antboot get config --normalize | grep minioSecretKey`
		命令查询。

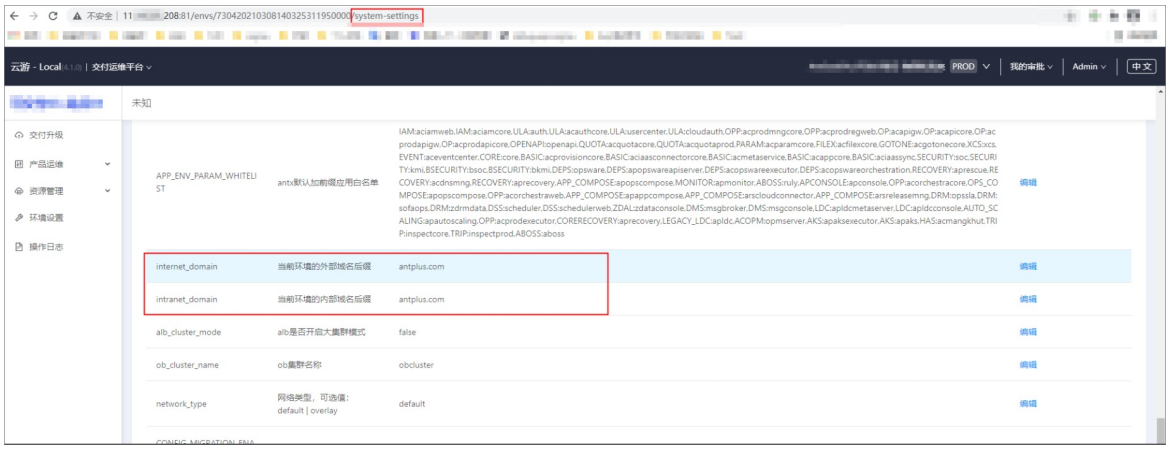
容器存储卷配置	-	物理机下， 使用 open-local 存储，三种存储类型均可配置为 `open-local-lvm-default` 。 可通过 `kubectl get sc` 命令查看。
安全设置	**keystone 加密**	设置为 Local。
Agent 配置	**调用地址**	Antboot没有部署 apyunqing Agent ，所以填写为 `127.0.0.1:80` 。

7. 在 **环境初始化** 步骤确认信息无误后，单击 **完成设置**。

4.3.2. 修改站点系统环境变量

操作步骤

1. 将当前访问域名的最后一部分修改为 system-settings 进入系统配置页面。如图所示：



2. 配置环境域名后缀。

   将 internet_domain 和 intranet_domain 项修改为您当前环境的外部域名。

3. 查看持久化存储类映射。

   确认 storage_io_level_to_class 项为：

```
{"normal":"open-local-lvm-default","medium":"open-local-lvm-default","high":"open-local
-lvm-default"}
```

> **注意**
>
> storage_io_level_to_class 定义了普通、中等、高速到具体存储类的映射关系，存储类必须位于集群内。您可以单击 **容器持久化存储**，然后单击 **存储类** 查看集群支持的具体存储类。

4.4. 云游 Local 发布 AntStack Plus 其他必要组件

4.4.1. 部署 yunyouoperator 和 apwebhook

产品用途

- yunyou operator
  - 监听 SLB 后端挂载。
  - SLB 摘流。
- apwebhook
  - 监听 anytunnel 后端挂载。
  - 在环境变量中注入 Pod index、Pod name、Pod dnsname。

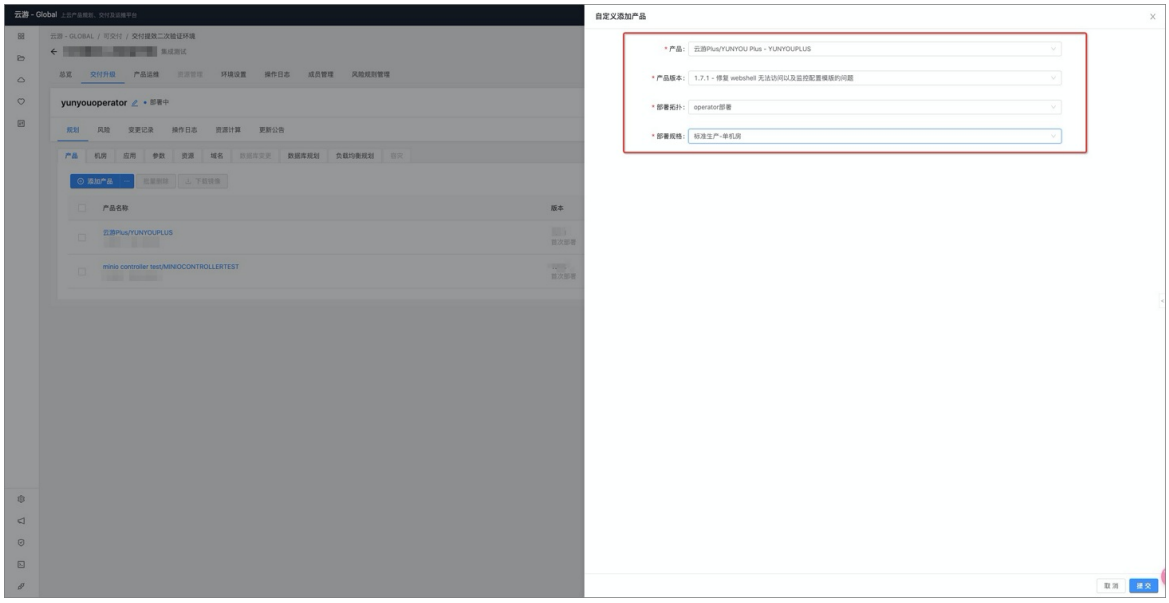
安装步骤

注意

- 云游 Local 1.9.x~1.11.x 版本，选择 operator 部署拓扑制作解决方案，只安装 yunyouoperator。
- 云游 Local 1.12.x 及以上版本，选择 operator 部署拓扑制作解决方案安装 yunyouoperator 和 webhook。

1. 制作解决方案。

   在云游 Global 上制作 YUNYOUPLUS 的解决方案，产品版本为：1.13.0；部署拓扑为：operator；部署规格为：标准生产-单机房。



2. 云游 Local 导入解决方案。

   在云游 Local 导入制作的解决方案。其中，确认参数时，您需要注意确认以下参数：

   - **产品公共参数** 中确认 DB 信息：

   - **应用启动参数** 中确认 operator 启动参数：

应用名	键	值模版	描述

应用名	键	值模版	描述
yunyouoperator	K8S_PROVIDER	CLOUD_THIRDPAPTY	■ 公有云填：ALIYUN_PUBLIC ■ 专有云填：ALIYUN_DEDICATED ■ 物理机/第三方填：CLOUD_THIRDPARTY
yunyouoperator	SLB_CONTROLLER_ENABLE	FALSE	开启对于 anytunnel 类型 SLB 挂载控制，与云游 Local 的 **是否支持 anytunnel** 保持一致。物理机下填写 FALSE。
yunyouoperator	IS_ENI_MODE		现场确定，物理机下该参数无需配置，维持默认值即可。专有云 3.12 版本需设为 FALSE，3.14 及以上版本需设为 TRUE。
yunyouoperator	ALIYUN_ACCESS_KEY		现场确定，物理机下该参数无需配置，维持默认值即可。专有云与云游 Local 保持一致。
yunyouoperator	ALIYUN_ACCESS_SECRET		现场确定，物理机下该参数无需配置，维持默认值即可。专有云与云游 Local 未加密的 secret 保持一致（从 configseed 获取）。
yunyouoperator	ALIYUN_ASAPI_HOST		现场确定，物理机下该参数无需配置，维持默认值即可。专有云通过天基获取，例如： `server.asapi.cn-neimeng-121e-d01.intra.121e.shuguang.com/asapi/v3` 。
yunyouoperator	ALIYUN_ECS_HOST		现场确定，物理机下该参数无需配置，维持默认值即可。专有云通过天基获取。
yunyouoperator	ALIYUN_SLB_HOST		现场确定，物理机下该参数无需配置，维持默认值即可。专有云通过天基获取。

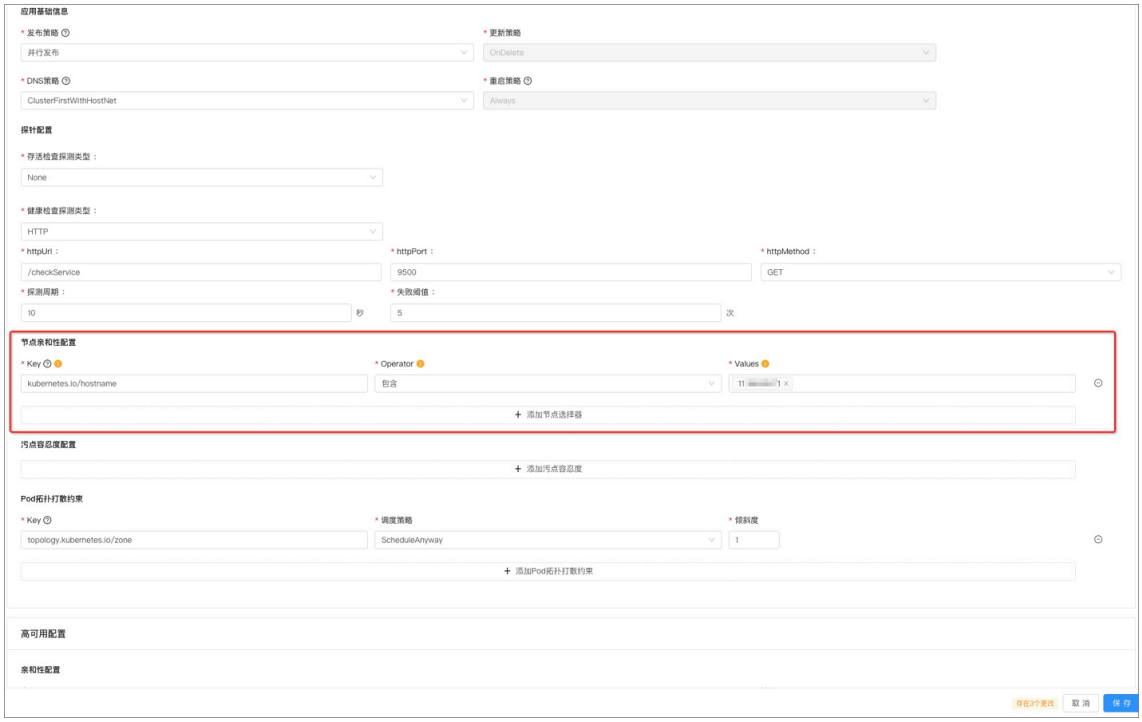
- **应用启动参数** 中确认 YUNYOUPLUS 域名：

> **说明**
>
> 如果没有部署云游 Agent，可以忽略此步骤。

	产品码	应用名	键	值模板	渲染结果	操作
已确认	YUNYOUPLUS	apyunqingplus	ANTCLOUD_SOFA_domain_name	yun[redacted]-eu95-x86.alipay.net	yuny[redacted]chine-eu95-x86.alipay.net	编辑

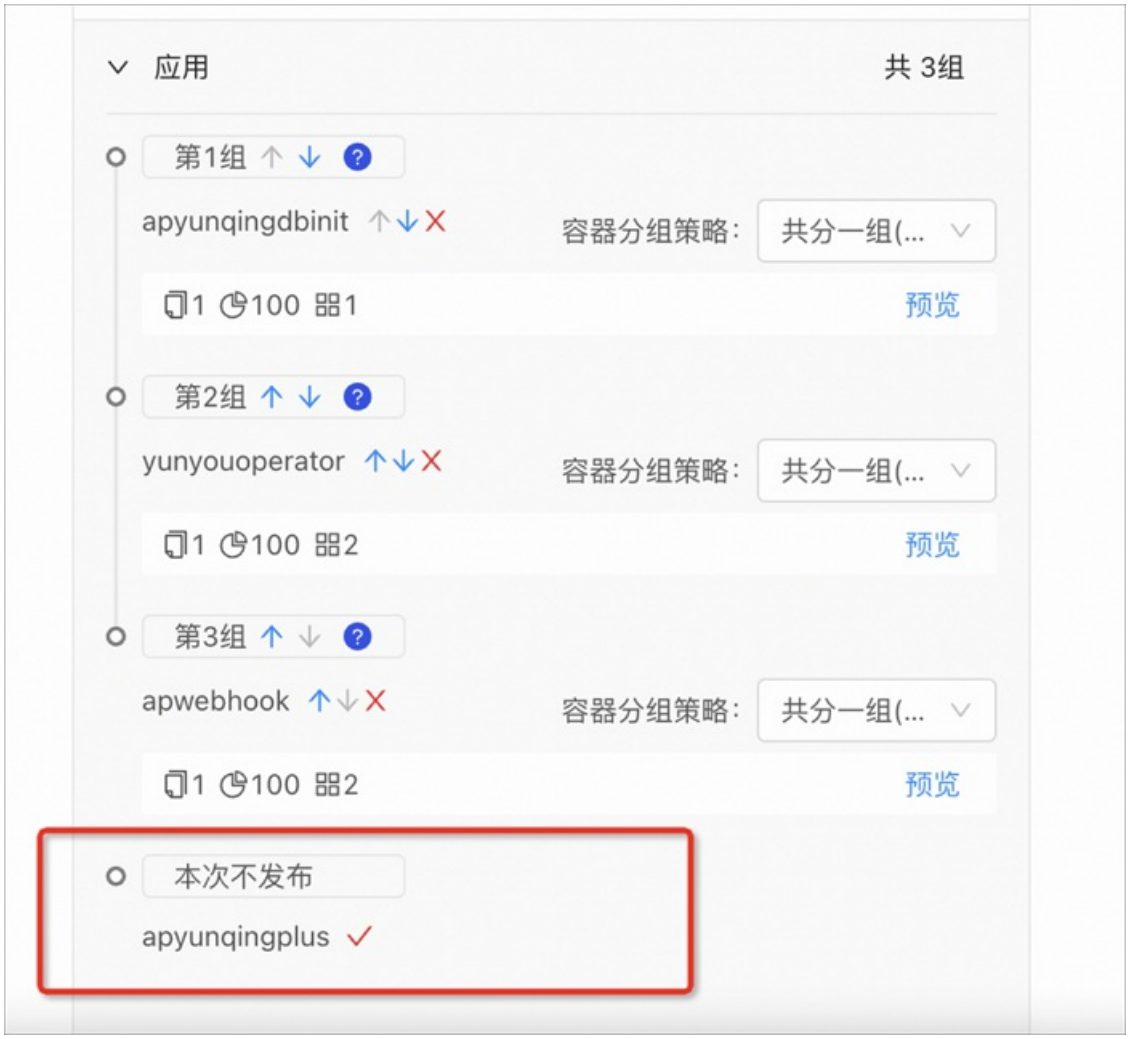
- 在 **应用配置规划** 页签修改 apyunqingplus 的本地参数，添加云游所在机器的 IP（多实例云游需要添加多个）：

> **说明**
>
> 如果没有部署云游 Agent 或者云游部署在AKE集群中，节点亲和性配置可以随便填写。



3. 发布部署。
   - 1.11.x 版本调整发布顺序
     - 第一组先发布 apyunqingdbinit，再发布 yunyouOperator。
     - 把 apyunqingplus 放在第二组，发布完 yunyouOperator 后直接跳过发布 apyunqingplus。
     - apwebhook 可以不发布，如果确实需要发布，可参见 附录：1.11.x 版本部署云游 Webhook。
   - 1.13.x 版本调整发布顺序

     调整方式如下：
     - 第一组先发布 apyunqingdbinit
     - 第二组先发布 yunyouOperator 、再发布 apwebhook。



yunyouoperator 发布成功即表示其工作正常。

备机房安装

由于云游双机房共库，安装备机房时不需要 dbinit，第一组先发布 yunyouOperator 、再发布 apwebhook，也不需要发布 apyunqingplus。

附录：1.11.x 版本部署云游 Webhook

1. 在某个 master 节点中下载 Deployment 文件。
2. 创建云游文件。

```
kubectl create ns yunyou
```

3. 解压步骤 1 下载的文件。

```
tar -xf yunyou_webhook.tar
```

4. 执行脚本文件。

```
sh deployment/webhook-create-signed-cert.sh --service env-injector-webhook-svc --secret
env-injector-webhook-certs --namespace yunyou
```

5. 创建 YAML 文件。

```
cat deployment/mutatingwebhook.yaml | deployment/webhook-patch-ca-bundle.sh > deploymen
t/mutatingwebhook-ca-bundle.yaml
```

6. 部署 Pod。

```
kubectl apply -f deployment/configmap.yaml  -n yunyou
kubectl apply -f deployment/deployment.yaml  -n yunyou
kubectl apply -f deployment/service.yaml  -n yunyou
kubectl apply -f deployment/mutatingwebhook-ca-bundle.yaml  -n yunyou
```

7. 验证云游 Webhook 是否部署正常。
   i. 测试云游的可用性。

      在 ops 执行 `docker ps` 命令，观察云游容器是否已经正常拉起。

   ii. 待容器状态变为 healthy 后，通过浏览器访问 `http://{yunyou地址:81}` 查看是否可以正常访问。

4.4.2. 部署云游 Agent

您可以通过云游 Agent 对云游 Local 版本进行管理、部署和升级。

前置条件

- 云游 Local 版本不低于 1.10.0，建议使用 1.11 及以上版本。
- 云游 Agent 版本不低于 1.10.0，建议使用 1.11 及以上版本。

注意事项

- 云游 Local 1.11.x 版本必须部署云游 Agent。
- 如果云游 Local 1.13.x 版本是以 Docker 部署方式安装在 ops 节点上，也建议按照 1.11.x 版本的步骤安装云游 Agent。

操作步骤

1. 开启 Docker 的 2375 端口。

> **说明**
>
> 不同版本 Docker 开启此端口的方法略有差异，本文以 alidocker 为例。

```
vim /etc/sysconfig/docker
```

修改如下参数：

```
// 开启 2375 端口。
-H tcp://0.0.0.0:2375
// 关闭 tls
#--tlsverify --tlscacert=/root/.dockercert/ca.pem --tlscert=/root/.dockercert/cert.pem
--tlskey=/root/.dockercert/key.pem
```

2. 重启 Docker。

```
systemctl reenable docker
systemctl daemon-reload
systemctl restart docker
```

3. 在部署云游的机器上部署云游 Agent。

```
mkdir /var/metrics
docker run -td --name yunyou-agent --network host -v /var/metrics:/var/metrics <云游agent镜像>
```

4. 在云游 Local 上配置 Agent 服务地址。

   i. 登录云游 Local。

ii. 修改环境参数。

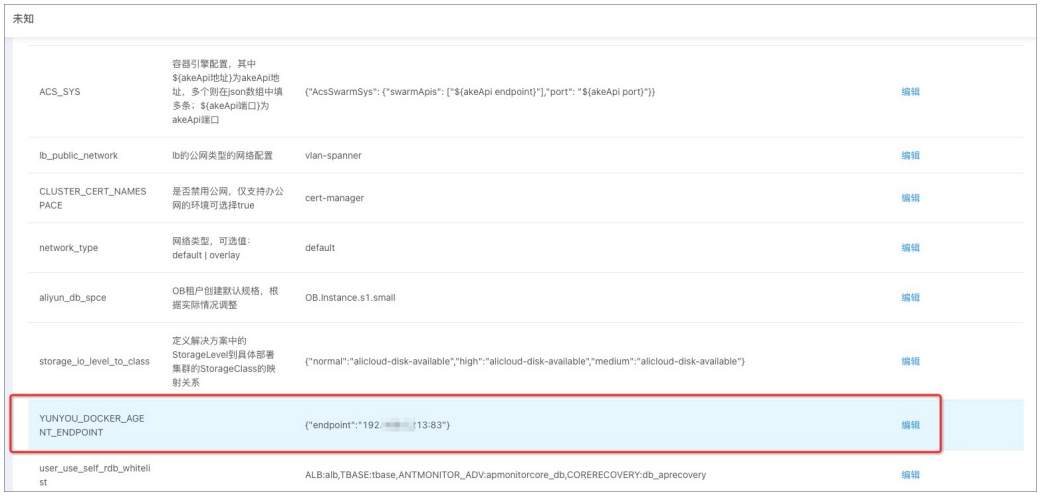
- 1.10.0 版本

  a. 将云游域名的最后一部分修改为 `system-setting` ，例如：

  `http://yunyou.com:81/envs/7304202******25957110000/system-setting` 。

  b. 将 `YUNYOU_DOCKER_AGENT_ENDPOINT` 字段值修改为 agent 地址信息。

  格式为： `{"endpoint":"<ip:port>"}` ，如下图所示：



- 1.11.0 及以上版本

  a. 在左侧导航栏，单击 **环境设置**。

  b. 将 Yunyou agent 值修改为 Agent 的地址信息。

  格式为： `ip:port` ，如下图所示。

> **注意**
>
> 如果修改失败或没有前端入口，则按照 1.10.0 版本方式进行配置。

4.4.3. 部署 ADNS Operator

注意事项

- 全局域名只有一个，所有的产品域名都会写入全局域名中。
- 如果产品域名对应 A 记录的 IP 为 F5 IP，则产品发布后需要通过云游页面修改域名对应的 IP。
- ADNS Operator 安装完成后，重新发布已发布的产品或发布新产品会创建 dnsrecord，且会影响已注册的产品域名。发布完成后，需要通过云游界面修改域名对应 IP（改为 F5 IP）
- 已创建 dnsrecord的产品，重新发布后：
  - 如果对应的 dnsrecord enable 字段被设为 `false`（容灾切换场景下会设为 `false`），会更新整个 dnsrecord。发布完成后，需要通过云游界面修改域名对应 IP（改为 F5 IP）。
  - 如果 dnsrecord enable 字段为 `true`，不会更新 dnsrecord。

前提条件

已部署 ADNS，且已知 ADNS master 的 SLB IP。

操作步骤

1. 制作解决方案。

   在云游 Global 上制作 ADNS 的解决方案，产品版本为：1.16.3；部署拓扑为：adns Operator；部署规格为：标准生产-单机房。

2. 在云游 Local 上导入解决方案。

   其中，确认参数时，您需要注意确认以下参数：

应用名	键	值模版	描述
hestiamanager	DNSMasterIP	192.168.1.1	ADNS master 的 SLB IP 地址，填写一个即可。
hestiamanager	DNSMasterPort	80	ADNS master 的 API 端口，一般是 80 端口。

4.4.4. 部署 IAM 和 OP

物理机 standalone 部署

产品版本和拓扑规格

产品名	版本	拓扑	规格
IAM	2.43.12	私有云建站部署（全量PaaS平台）	标准生产-单机房

op	0.31.4	专有云模式部署	标准生产-单机房

部署 IAM

使用 standalone 模式部署 IAM 时，确认下参数无误后，直接发布即可。部署流程请参见 部署 yunyouoperator 和 apwebhook。IAM 涉及参数如下：

应用名	键	值模板	描述
iam-init	aliyun_account	NOT_NEED	阿里云账号，单软件独立输出模式下不用填，保持默认值；可以修改该参数重跑 iam-init。
aciamcore	deploy_mode	private	-
aciamweb	deploy_mode	private	-
iam-init	deploy_mode	private	-

发布完成后，您还需要将 IAM 的域名 `admin.{domain_name}` 域名注册一下，IP 地址对应为 iamweb 的 VIP。执行完成后需执行测试用例，验证产品功能。

部署 OP

使用 standalone 模式部署 OP 时，需修改如下参数：

应用名	键	值模板	描述
opinit	new_site	true	是否是新增站点，决定 IAM 相关 API 的初始化。 非常重要的参数，请慎重填写。

参数修改完成后，直接发布即可。发布完成后需执行测试用例，验证产品功能。

物理机 FullStack 部署

产品版本和拓扑规格

产品名	版本	拓扑	规格

iam	2.43.12	私有云建站部署（全量PaaS平台）	标准生产-单机房
op	0.31.4	专有云模式部署	标准生产-单机房

部署 IAM

第一次发布：使用 standalone 模式，无需参数修改，直接发布即可。

第二次发布：MS 发布以后，重新渲染参数变为 FullStack 模式，此时需要二次发布 IAM，并重跑一下 iam-init。您可以修如下参数重跑 iam-init。

应用名	键	值模板	描述
iam-init	aliyun_account	NOT_NEED	阿里云账号，单软件独立输出模式下不用填，保持默认值；可以修改该参数重跑 iam-init。
aciamcore	deploy_mode	private	-
aciamweb	deploy_mode	private	-
iam-init	deploy_mode	private	-

发布完成后，您还需要将 IAM 的域名 `admin.{domain_name}` 域名注册一下，IP 地址对应为 iamweb 的 VIP。执行完成后需执行测试用例，验证产品功能。

部署 OP

使用 FullStack 模式部署 OP 时，需修改如下参数：

应用名	键	值模板	描述
opinit	new_site	true	是否是新增站点，决定 IAM 相关 API 的初始化。 非常重要的参数，请慎重填写。

参数修改完成后，直接发布即可。发布完成后需执行测试用例，验证产品功能。

当 MS 发布完成，且自动化测试过了以后。需二次发布 IAM、OP、OneConsole。发布完成后执行测试用例，验证产品功能。

4.5. 云游接入 IAM

4.5.1. 云游 Local 成员管理接入 IAM

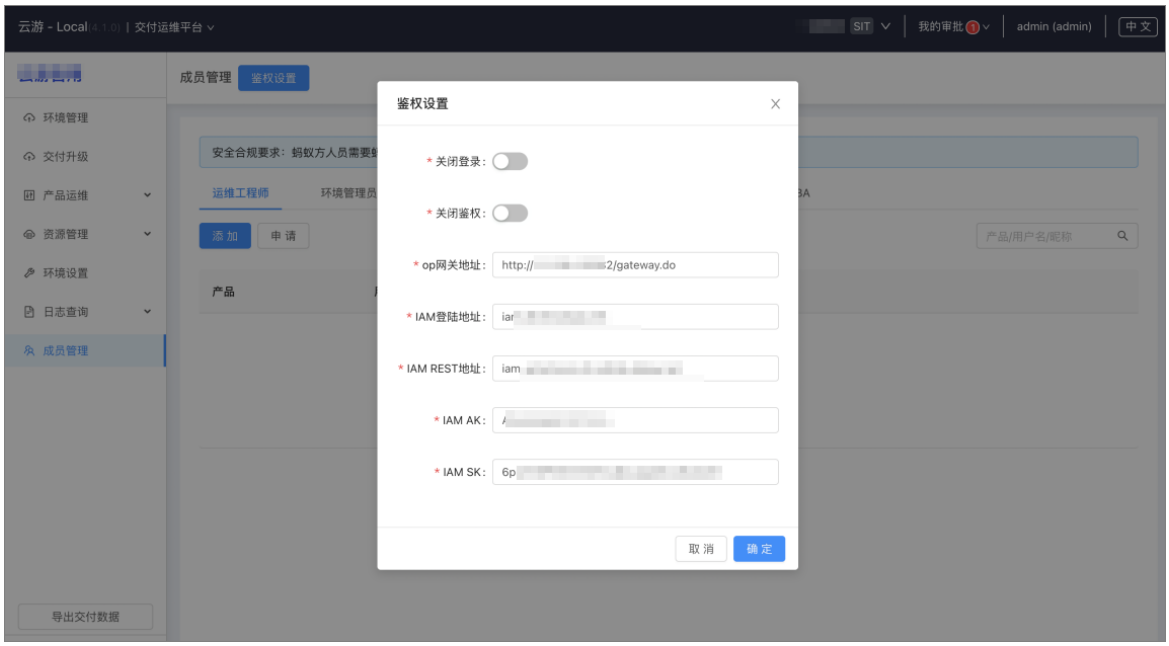
云游 Local 1.13.0 及以上版本

> **注意**
>
> - 自 1.13.0 版本开始，云游 Local 支持多租户功能。
> - 自 1.13.0 版本开始，调用云游 Lcoal 的 API 可以使用内嵌的 gateway，但更推荐通过 OP 来调用。

1. 在 IAM 的 ROC 租户下，为云游创建一个服务账号并授予租户管理员角色。

   创建服务账号及获取 AK（AccessKey ID）、SK（AccessKey Secret） 的方法请参见 IAM 使用指南中《管理服务账号》的内容。

2. 登录云游 Local 控制台。
3. 在左侧导航栏单击 **权限设置**，然后单击 **鉴权设置**。
4. 配置以下参数后，单击 **确定**。



参数	说明
关闭登录	选择是否通过 IAM 登录。 选择开启。

参数	说明
关闭鉴权	选择是否通过 IAM 鉴权。 选择开启。
op 网关地址	配置当前环境的 OP（OpenAPI） 的 LB。格式为： `http://{当前环境的 OP的 LB（必须是域名）}/gateway.do` 。
IAM 登录 地址	当前环境 IAM 控制台的 LB 域名（必须是域名）。
IAM REST 地址	当前环境的 aciamcore 应用的内网 LB 域名（必须是域名）。
IAM AK	云游服务账号的 AK。
IAM SK	云游服务账号的 SK。

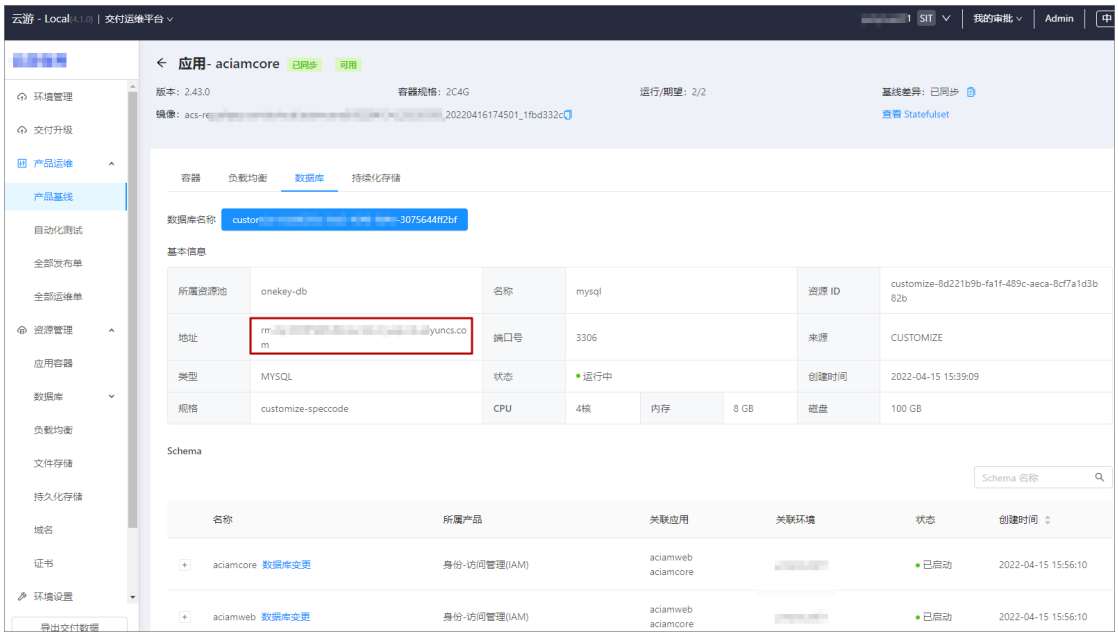
5. 单击 **确定**。

云游 Local 1.13.0 之前版本

> **注意**
>
> 云游 Local 1.13.0 版本以下版本，调用云游 Lcoal 的 API 需访问内嵌的 gateway。

1. 元数据初始化。
   i. 在云游 Local 上完成 IAM 的部署和验证。

ii. 在云游 Local 上获取 IAM 产品的 aciamcore 数据库的相关连接信息。



iii. 通过 MySQL 命令行初始化云游相关角色权限元数据到 aciamcore 数据库中。

```
mysql -P${db_port} -h${db_url} -u${db_user} -p${db_password} --default-character-se
t='utf8' aciamcore < aciamcore_init.sql
```

参数说明如下：

参数	说明
${db_port}	填写 aciamcore 的数据库访问端口。
${db_url}	填写 aciamcore 的数据库地址。
${db_user}	填写 aciamcore 的数据库用户名。
${db_password}	填写 aciamcore 的数据库访问密码。
aciamcore_init.sql	是具体的初始化 SQL 文件。单击 aciamcore_init_v1 4.sql.zip 下载该文件。

示例如下：

```
mysql -P3306 -h172.**.**.28 -uaciam**** -pPaaS1**** --default-character-set='utf8'
aciamcore < aciamcore_init.sql
```

2. 开启登录鉴权。

元数据初始化完成后，您可以调用以下 API 开启云游的鉴权。

```
curl -i -v -X PUT -H 'Content-Type:application/json' http://${yunyou_ip}:81/api/ops/system-params/open-auth -d '{"noLogin":false,"noAuth": false,"restUrl":"${aciamcore_vip}","loginUrl":"${aciamweb_domain}"}'
```

参数说明如下：

参数	说明
${yunyou_ip}	填写云游 Local 的 IP 地址。
${aciamcore_vip}	填写当前环境 aciamcore 应用的内网 VIP，例如：10.0.0.1
${aciamweb_domain}	填写当前环境 IAM 控制台的访问域名。
isNoLogin	是否开启 IAM 登录，取值如下： ○ true：不开启 IAM 登录。 ○ false：开启 IAM 登录。 此项填写为 false。
isNoAuth	是否开启 IAM 鉴权，取值如下： ○ true：不开启 IAM 鉴权。 ○ false：开启 IAM 鉴权。 此项填写为 false。

示例如下：

```
curl -i -v -X PUT -H 'Content-Type:application/json' http://192.168.2.227:81/api/ops/system-params/open-auth -d '{"noLogin":false,"noAuth":fasle,"restUrl":"192.168.2.48","loginUrl":"admin.****.aliyuncs.com"}'
```

API 访问成功后，预计 1 分钟内生效。

说明

- 您也可以在 apyunqing 数据库中开启鉴权，命令如下：

```
update system_parameter set value='{"noLogin":false,"noAuth":fasle,"restUrl":"192.168.2.48","loginUrl":"admin.****.aliyuncs.com"}'  where type='auth_config';
```

- 如果您后续不再通过 IAM 鉴权，可在 apyunqing 中执行如下命令关闭：

```
update system_parameter set value='{"noLogin":true,"noAuth": true}' where type='auth_config';
```

4.5.2. 云游接入 IAM 审计功能

> **说明**
>
> 本文以云游 Plus 1.10.2 版本为例，请根据实际情况修改版本。

准备工作

- 下载 apyunqingplus 镜像。

```
docker run --net=host -d -v /home/yunyou/staticfiles:/home/staticfiles acs-reg.alipay.com
/acloud/apyunqingplus:EI62154649_20211122_20211122161132_cdd9ca69
```

- 下载 apyunqingsetup 镜像。

```
docker run --net=host -d -v /home/yunyou/staticfiles:/home/staticfiles acs-reg.alipay.com
/acloud/apyunqingsetup:master_20211022091743_dcb7629d
```

- 备份 apyunqing 的数据库。
- 导入云游 Plus 1.10.2 的镜像到镜像仓库。

```
docker push acs-reg.alipay.com/acloud/registry:1.1
```

升级云游

1. 登录云游 Local 服务器。
2. 进入 `apyunqing-compose` 目录。

   ```
   cd /home/yunyou/staticfiles/apyunqing-compose
   ```

3. 修改 `apyunqing-compose` 目录下的 `config.json` 文件。

   在 `config.json` 中添加以下环境变量：

   ```
   "ANTCLOUD_SOFA_iam_customer_id=0000000001",
   "ANTCLOUD_SOFA_audit_address=iamcore.jr.alipay.net",
   ```

   `audit_address` 对应地址为可用的 aciamcore 地址。

4. 升级云游 Plus 镜像。

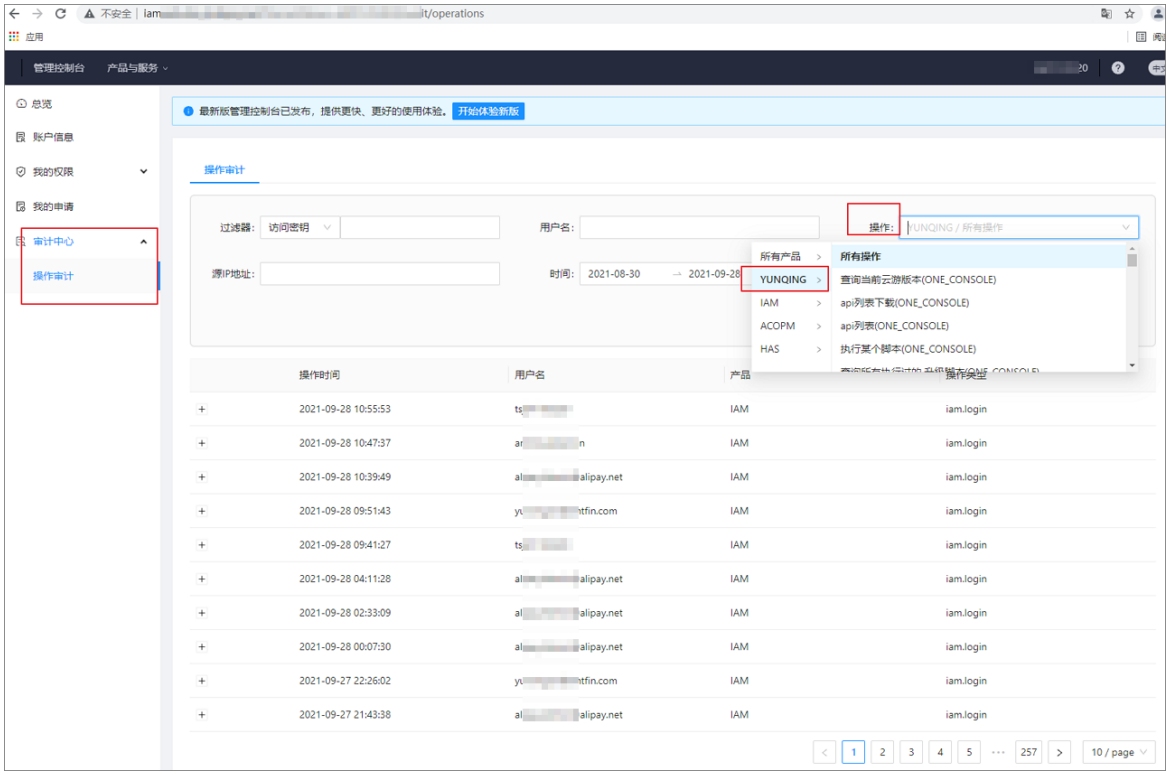
   ```
   python yunqing-setup.py y --image <apyunqingplus 镜像>
   ```

导入审计元数据

1. 下载 SQL 脚本。
   - operation_type.sql
   - operation_type_onex_extra.sql
2. 在对应环境的数据库 aciamcore 中执行步骤 1 下载的 SQL 脚本来导入已经准备好的审计元数据。

查看审计结果

1. 登录 IAM 控制台。
2. 在左侧导航栏选择 **审计中心 > 操作审计**。
3. 在 **操作** 区域选择 **YUNQING**。



附录：回滚方案

1. 登录云游 Local 服务器。
2. 使用之前镜像对云游 Local 进行回滚。

```
cd /home/yunyou/staticfiles/apyunqing-compose
## 回滚云游 Plus 镜像
python yunqing-setup.py y --image <apyunqingplus 上一版本镜像>
```

3. 还原数据库。

```
mysql -uapyunqing -pAntcloud2018 -h127.0.0.1 apyunqing < apyunqing.dump --force
```

4.6. 使用云游的一键部署发布 SOFAStack 产品

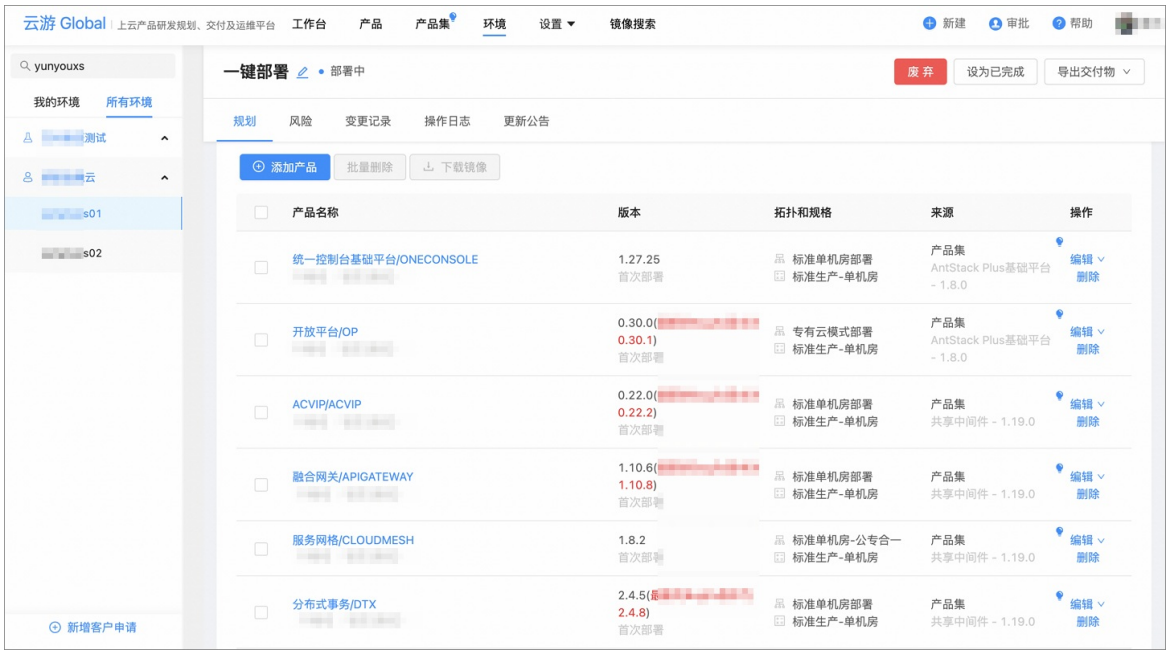
注意事项

- 仅适用于云游 Local 1.13.x 及以上版本。
- 仅适用于交付站点时首次安装站点所有的产品。
- 仅适用于通过产品成熟度的产品，目前支持 SOFAStack 产品线。

操作步骤

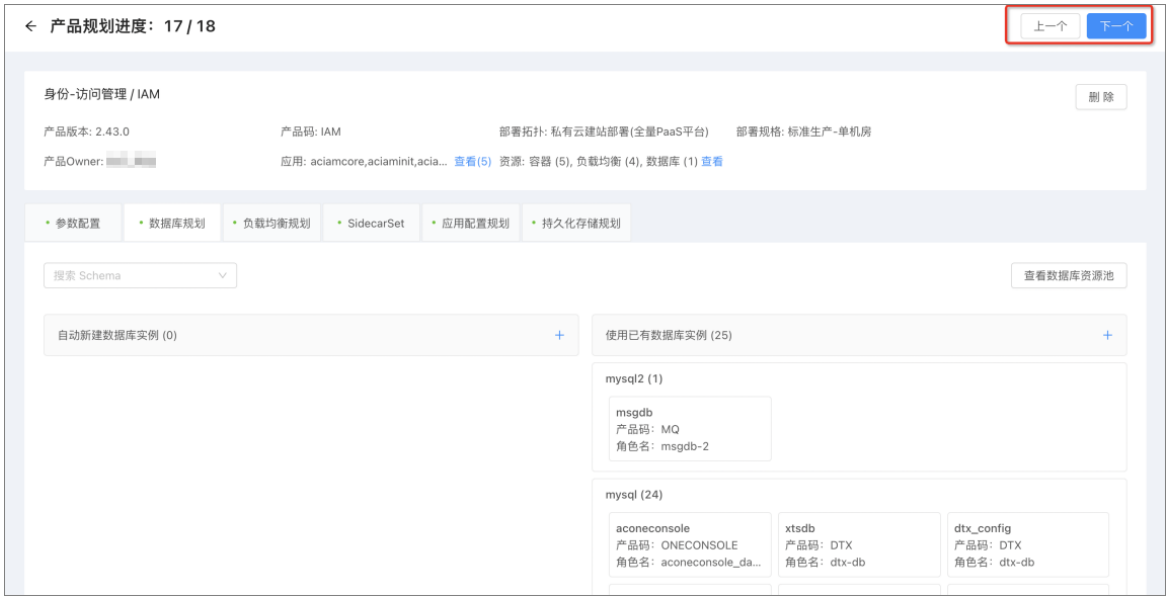
1. 在云游 Global 中创建环境，将首次交付的所有产品添加在一个解决方案中。

   您可以将 SOFA 中间件、IAM、OP 放在一个解决方案中。



2. 导出解决方案。
3. 登录云游 Local。
4. 在左侧导航栏单击 **交付升级**，然后单击 **一键部署**。
5. 单击 **导入产品**，然后将之前导出的部署方案上传。

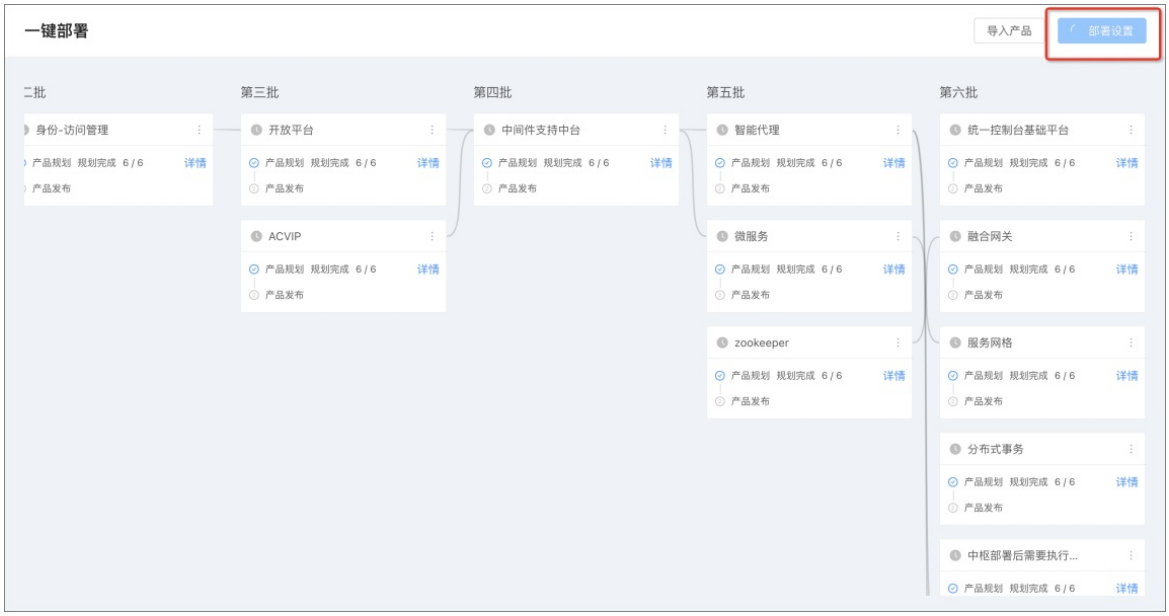
6. 单击 **开始规划**，然后依次完成所有产品部署前任务的确认。
   - 和原先的发布、批量发布逻辑一致，您需要依次完成每个产品的 **参数配置**、**数据库规划**、**负载均衡规划**、**持久化存储规划** 和 **应用配置化规划**。
   - 单击 **上一个** 或 **下一个** 可切换待规划的产品。
   - 该产品不需要在本次发布，可单击 **删除** 将其从本次发布队列中踢除。



注意

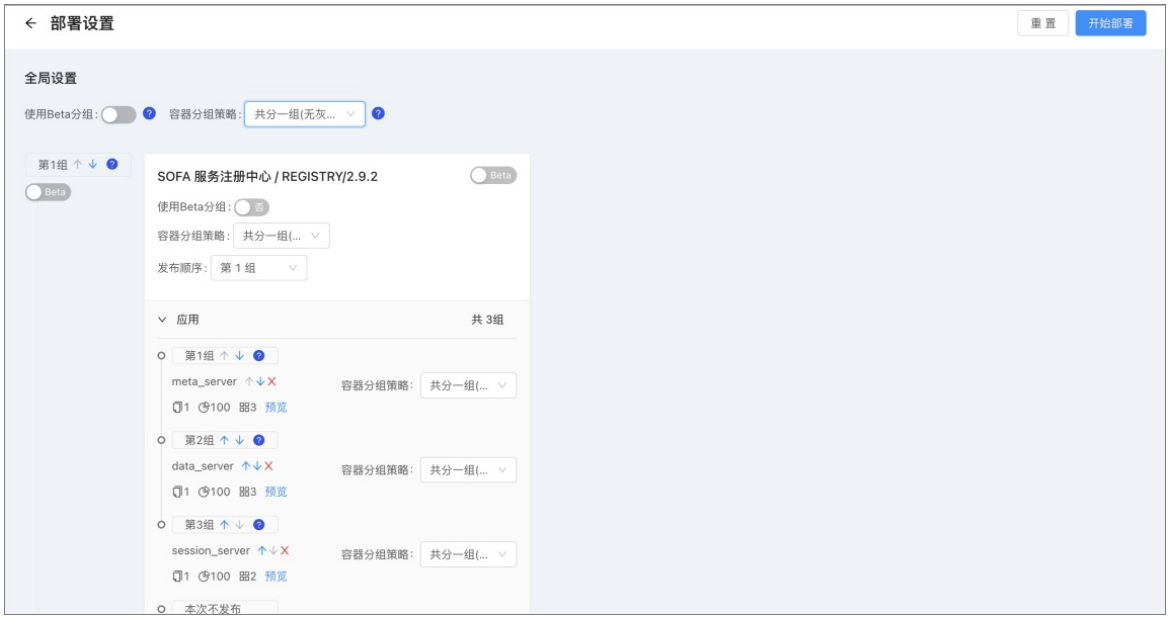
- 未通过一键部署导入的产品，且未发布过，会被加入到一键部署的任务中。如果不需要发布，需要您手动从本次一键发布中删除。
- 未通过一键部署导入的产品，且已经发布过（产品基线中会出现），无论其正在发布中或者发布失败，都不会被加入到一键部署的任务中。

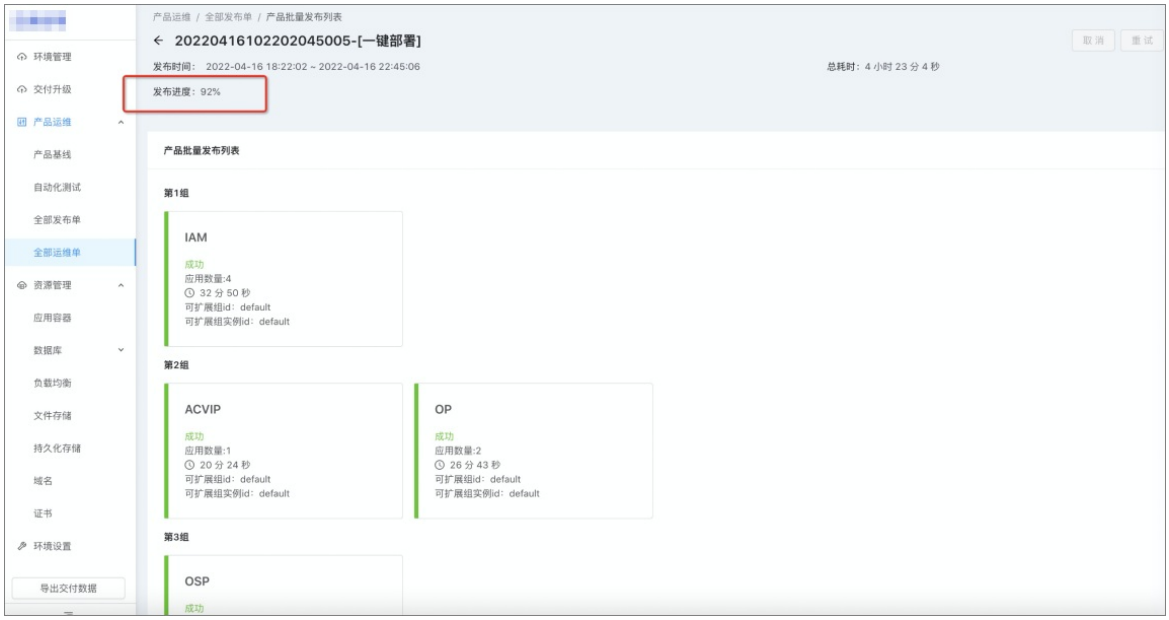
7. 全部规划设置完成后，单击 **部署设置**。



8. 在 **部署设置** 页面调整发布策略。

   首次发布时，**容器分组策略** 可以设置为 **共分一组**，让多实例并行发布；**使用 Bate 分组** 可以不开启。



9. 单击 **开始部署**。

   您可以在发布单中查看整体发布进度和每个产品部署的结果。