SOFAStack

数据访问代理
部署指南

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

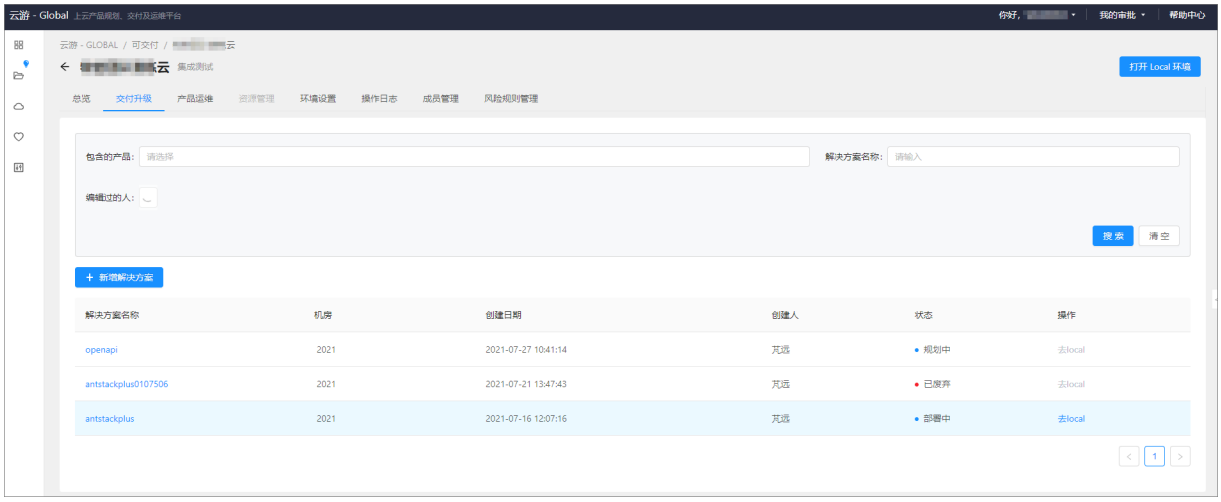
1.环境初始化 06
2.准备工作 13
3.部署底座 17
3.1. 部署 Docker 17
3.2. 部署 AntStack DNS 17
3.3. 部署 Captain 20
3.4. 部署 Registry 21
4.部署中枢集群 22
4.1. 获取 ReleaseVersion 相关资源 22
4.2. 创建 ReleaseVersion 版本 23
4.3. 创建元集群 24
4.4. 配置集群网络 31
4.5. 扩容节点 32
4.6. 为节点添加标签 33
4.7. 部署 charts 插件 34
4.8. 对外暴露 Grafana 服务 36
4.9. 底座产品域名透出 37
5.部署云游 39
5.1. 部署云游 39
5.2. 添加新站点 40
6.部署核心产品 47
6.1. 产品版本和拓扑规格 47
6.2. 部署身份访问 IAM 47
6.3. 部署 OpenAPI（OP） 47
7.部署 SOFA 产品 49
7.1. 产品版本和拓扑规格 49

7.2. 部署 ZooKeeper 49
7.3. 部署 Registry 49
7.4. 部署 ACVIP 49
7.5. 部署 OSP 50
7.6. 部署 IntelliProxy 50
7.7. 发布 MS 51
7.8. 部署 ODP 51

1.环境初始化

准备解决方案

在云游 Global 上面创建项目，并添加解决方案和产品，如下图所示：



准备介质

准备好前期所需要的 Docker 镜像以及配置文件和 RPM 包，上传到 ops1 指定目录。

云游介质准备

在云游 Global 中找到云游产品，下载相关镜像。步骤请参见 下载产品介质。

非云游介质准备

非云游介质准备非云游介质主要包括 Registry、Captain、yunyou、chart 部署等相关镜像。

文件名	获取方式
containerd 依赖软件包	在 2103 可交付验证钉钉群下载 `containerd.io-1.3.7-3.1.el7.x86_64.tar.gz` 。
captain-init.tar	镜像地址： `acs-reg.alipay.com/acloud/captain-init:20210402174057-ER61887887_release_20210402.d399cc5`
yunyou_setup.tar	镜像地址： `acs-reg.alipay.com/acloud/apyunqingsetup:1c89a3ed-1614743628409`

yunyou_webhook.tar	镜像地址： `acs-reg.alipay.com/acloud/apwebhook:6c604f96-1617098502874`
yunyou_dbinit.tar	镜像地址： `acs-reg.alipay.com/acloud/apyunqingdbinit:50c6f657-1617952348733`
registry.1.1.tar	镜像地址：`acs-reg.alipay.com/acloud/registry:1.1`
registry-conf.yaml	下载地址：registry-config.yml
cafeoperator	镜像地址： `acs-reg.alipay.com/acloud/cafeoperator:1a449caa-1591581544579`

chart	通过 Docker 获取文件和镜像，命令如下：
	`docker run -v $PWD/.:/helm_data  acs-reg.alipay.com/acloud/akepluscharts:4754e72d-1616646372722`
	
	`docker run acs-reg.alipay.com/acloud/akepluscharts:4754e72d-1616646372722 images`

device_name-1.1.8.sh	下载地址：device_name-1.1.8.sh
pssh	下载地址：pssh-2.3.1.tar.gz
kubectl	下载地址：kubectl.zip
netshoot.tar	镜像地址：`acs-reg.alipay.com/acloud/netshoot:latest`
netshoot.yaml	下载地址：netshoot.yaml

中枢集群 ReleasedVersion	• 镜像、二进制文件和 YAML 通过 Docker 获取：
	`docker run  acs-reg.alipay.com/acloud/releasedversion:1a0d18bb-1615793425674  -r antcloud -t meta-1.20-containerd -l`
	
	`docker run acs-reg.alipay.com/acloud/releasedversion:1a0d18bb-1615793425674 -r antcloud -t meta-1.20-containerd -p http://11.166.85.208:8888/data  > releaseversion.yaml`
	• selfdef_for_meta.txt

用户集群 ReleasedVersion	镜像、二进制文件和 YAML 通过 Docker 获取：
	`docker run acs-reg.alipay.com/acloud/releasedversion:1a0d18bb-1615793425674 -r antcloud -t meta-1.16-containerd -p http://158.**.**.11:8888/data  > releaseversion.yaml`
	`#上面的 158.**.**.11:8888 换成自己的文件服务器 IP 和端口。`
	
	`docker run acs-reg.alipay.com/acloud/releasedversion:1a0d18bb-1615793425674 -r antcloud/meta-1.16 -t meta -l`

spanner 镜像	镜像地址：`acs-reg.alipay.com/acloud/spanner:1.1.0` 需要打 tag 为 `hub.kf.zjnx.net/sofastack/hub.kf.zjnx.net/sofastack/spanner:1.1.0` ，并推送到中枢集群的hub。
nfcc 镜像	镜像地址：`acs-reg.alipay.com/acloud/nfccagent:1.1.0` 需要打 tag 为 `hub.kf.zjnx.net/sofastack/nfccagent:1.1.0`，并推送到中枢集群的 hub。

cafe-kubernetes-extension 镜像	镜像地址：
	`acs-reg.alipay.com/aks/cafe-kubernetes-extension:pure-20201105174024-v1.0.1-20201105.39a4a45`
	需要打 tag 为
	`hub.kf.zjnx.net/sofastack/cafe-kubernetes-extension:pure-20201105174024-v1.0.1-20201105.39a4a45`
	，并推送到中枢集群的 hub。

flyover 镜像	镜像地址：`acs-reg.alipay.com/acloud/flyover:v1.6.0` 需要打 tag 为 `hub.kf.zjnx.net/sofastack/flyover:v1.6.0`，并推送到中枢集群的 hub。

节点配置

服务器初始化

- 网络配置 bond
  - 在所有节点执行以下命令：
    - 关闭 networkManager 服务。

      ```
      systemctl stop NetworkManager
      systemctl disable NetworkManager
      ```

    - 执行脚本 `device_name-1.1.8.sh` 。
  - ops1 对所有 App 节点做免密操作。

    执行以下命令：

    ```
    ssh-keygen -t rsa -m PEM
    ```

- 关闭 IPv6

  i. 通过 pssh 脚本关闭所有节点的 IPv6 功能。

  ```
  ./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'echo "net.ipv6.conf.all.disable_ipv6 = 1" >> /etc/sysctl.conf'

  ./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'echo "net.ipv6.conf.default.disable_ipv6 = 1" >> /etc/sysctl.conf'

  ./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'echo "net.ipv6.conf.lo.disable_ipv6 = 1" >> /etc/sysctl.conf'

  ./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'sysctl -p'
  ```

  ii. 进入每台节点执行以下命令：

  ```
  sudo vim /etc/default/grub
  ```

增加以下内容：

iii. 执行以下命令，并重启所有 App 节点的物理机。

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
sudo reboot
```

iv. 通过 `ifconfig` 命令确认 IPv6 已关闭。

- 关闭 swap、selinux 和 ssh 反解析
  - 关闭 swap

```
./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'sysctl -w vm.swappiness=0'

./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'echo "vm.swappiness=0" >> /etc/sysctl.conf'
```

  - 关闭 selinux 和防火墙

```
./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'setenforce 0

./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config'

./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'systemctl stop firewalld.service'

./pssh-2.3.1/bin/pssh -i -A -O "StrictHostKeyChecking=no" -h /root/alliplist 'systemctl disable firewalld.service'
```

  - 关闭 ssh 反解析

```
#找到选项UseDNS ，取消注释，改为no
UseDNS no

#重启sshd服务
/etc/init.d/sshdrestart
systemctl restart sshd
```

- 修改 hostname 并分发 hosts 文件

  i. 修改 hostname。

```
hostnamectl set-hostname zjnxkfops1ops01
```

  ii. 将所有 App 节点和 OB 节点的解析及镜像中心解析添加到 hosts 文件里，并下发到所有节点。

- 修改 ntpserver

```
[root@zjnxkfops01 ~]# vi /etc/ntp.conf
driftfile /var/lib/ntp/drift
restrict default nomodify notrap nopeer noquery
# Permit all access over the loopback interface.This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
# Hosts on local network are less restricted.
restrict 158.220.102.0 mask 255.255.255.0 nomodify notrap
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool
(http://www.pool.ntp.org/join.html).
server 158.220.145.95 iburst minpoll 4 maxpoll 6

./pssh-2.3.1/bin/pscp-O "StrictHostKeyChecking=no" -h /root/alliplist /etc/ntp.conf /etc/
ntp.conf

./pssh-2.3.1/bin/pssh -i-O "StrictHostKeyChecking=no" -h /root/alliplist 'ntpdate 158.220
.145.95'

./pssh-2.3.1/bin/pssh -i-O "StrictHostKeyChecking=no" -h /root/alliplist 'hwclock -w'

./pssh-2.3.1/bin/pssh -i -O "StrictHostKeyChecking=no" -h /root/alliplist 'systemctl enab
le ntpd'

./pssh-2.3.1/bin/pssh -i -O "StrictHostKeyChecking=no" -h /root/alliplist 'systemctl star
t ntpd'
```

> **注意**
>
> 如果客户不提供 NTP server，需要自行在 ops server 上部署 NTP 服务并修改上述配置中的 NTP Server 的 IP。

- 磁盘分区初始化

  i. 在所有 App 节点执行以下操作：

```
parted /dev/sdb

mklabel gpt

parted /dev/sdb

mkpart primary 1 1919G

parted /dev/sdb

name 1 /home/t4

mkfs -t ext4 /dev/sdb1

echo "/dev/sdb1 /home/t4 ext4 defaults 0 0" >> /etc/fstab

mkdir /home/t4

mount -a
```

ii. App 节点挂盘。

```
[root@zjnxkfops01 ~]# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                   8:0    0 446.1G  0 disk
├─sda1                8:1    0   512M  0 part /boot/efi
├─sda2                8:2    0   512M  0 part /boot
└─sda3                8:3    0 445.1G  0 part
  ├─vg_os-lv_root 253:0    0 441.1G  0 lvm  /
  └─vg_os-lv_swap 253:1    0     4G  0 lvm
sdb                   8:16   0   1.8T  0 disk
sdc                   8:32   0   3.5T  0 disk
[root@zjnxkfops01 ~]# lsblk
```

iii. 将 Sdc 做成 vg 卷。

初始化磁盘分区表：

```
parted /dev/sdc

mklabel msdos

pvcreate /dev/sdc

vgcreate yoda-ssd /dev/sdc
```

拷贝介质

1. 登录 ops1 节点。
2. 将准备好的介质拷贝到 `/home/t4/install-images` 下。

```
mkdir /home/t4/install-images/

cd /home/t4/install-images/
```

2.准备工作

本文介绍安装物理机底座时需完成的准备工作。

准备机器资源

您需要准备如下数量的服务器：

名称	机器数量	说明
ops 节点	2 台	用于部署 Yunyou、Captain、DNS、镜像中心等服务。
master 节点	3 台	用于部署 K8s 的 master 节点。节点规格要求至少 8C16G（8 核 CPU、16G 内存）。
node 节点	动态调整	根据实际环境需要，动态调整节点数量。

服务器要求如下：

- 服务器操作系统内核版本 3.10。
- 服务器目录 `/home/t4` 需单独挂载，否则磁盘利用率很容易达到 100%。
- 节点除了系统盘和 `/home/t4` 单独挂载以外，还需要至少有三台节点拥有独立的磁盘以做成卷组。

准备数据库

部署基础组件之前，需提前规划好基础产品的数据库，需创建的数据如下：

数据库名（schema）	对应基础产品名	说明
antdns	AntStack DNS	ADNS 产品对应的数据库。
captain	Captain Plus	Captain Plus 产品对应的数据库。
apyunqing	云游 Plus	云游 Plus 产品对应的数据库。

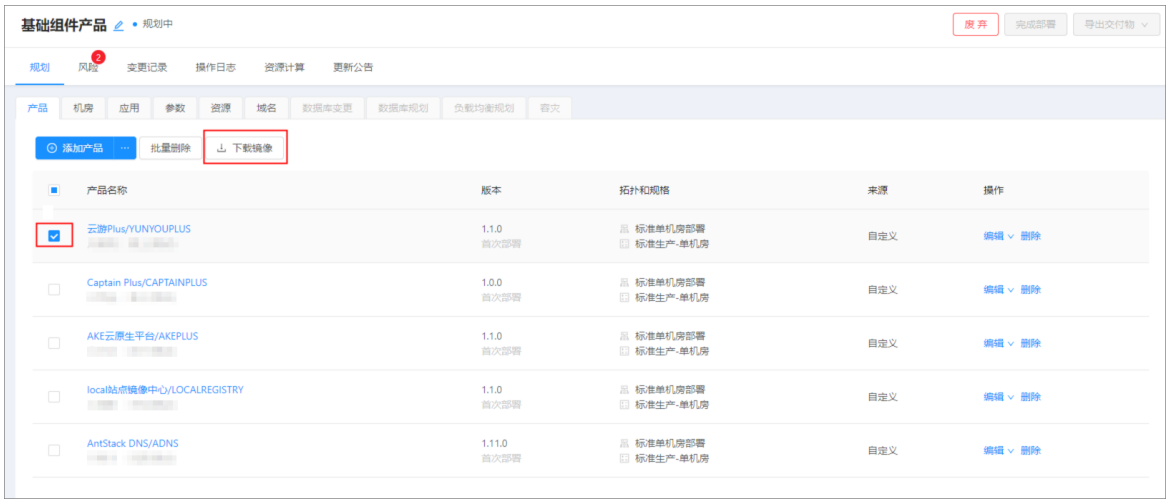
制作解决方案

通过云游 Global 制作基础组件解决方案，需包含的产品如下：

产品名	产品码	规格	拓扑
AKE 云原生平台	AKEPLUS	标准单机房部署	标准生产-单机房
Captain Plus	CAPTAINPLUS	标准单机房部署	标准生产-单机房
云游 Plus	YUNYOUPLUS	标准单机房部署	标准生产-单机房
Local 站点镜像中心	LOCALREGISTRY	标准单机房部署	标准生产-单机房
AntStack DNS	ADNS	标准单机房部署	标准生产-单机房

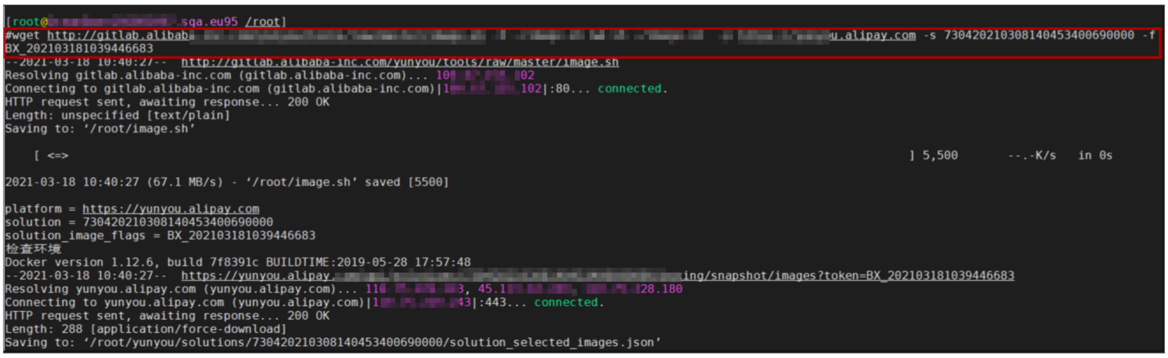
下载产品介质

1. 登录云游 Global 控制台。
2. 进入目标环境，然后在左侧导航栏单击 **交付升级**。
3. 单击 **规划**，然后单击 **产品** 页签。
4. 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。



5. 在 **镜像下载地址** 对话框单击 **复制**。

6. 在域内跳板机（11.166.85.97）上执行刚刚获取的命令。



```
[root@[redacted]sqa.eu95 /root]
#wget http://gitlab.alibab[redacted]u.alipay.com -s 730420210308140453400690000 -f
BX_202103181039446683
--2021-03-18 10:40:27--  http://gitlab.alibaba-inc.com/yunyou/tools/raw/master/image.sh
Resolving gitlab.alibaba-inc.com (gitlab.alibaba-inc.com)... 10[redacted]02
Connecting to gitlab.alibaba-inc.com (gitlab.alibaba-inc.com)|1[redacted]102|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: '/root/image.sh'

    [ <=>                                                                    ] 5,500       --.-K/s   in 0s

2021-03-18 10:40:27 (67.1 MB/s) - '/root/image.sh' saved [5500]

platform = https://yunyou.alipay.com
solution = 730420210308140453400690000
solution_image_flags = BX_202103181039446683
检查环境
Docker version 1.12.6, build 7f8391c BUILDTIME:2019-05-28 17:57:48
--2021-03-18 10:40:27--  https://yunyou.alipay.[redacted]ing/snapshot/images?token=BX_202103181039446683
Resolving yunyou.alipay.com (yunyou.alipay.com)... 11[redacted]3, 45.1[redacted]28.180
Connecting to yunyou.alipay.com (yunyou.alipay.com)|1[redacted]43|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 288 [application/force-download]
Saving to: '/root/yunyou/solutions/730420210308140453400690000/solution_selected_images.json'
```

7. 进入镜像文件下载目录，将 `images.tar` 文件复制到自己环境内的 ops 机器中。

8. 下载补充镜像。

   您需要通过以下链接下载补充镜像，并上传至自己环境内的 ops 机器中。待 docker 安装好以后，使用 docker load 将镜像解压到本地。

   - catpain_init 镜像

     ```
     acs-reg.alipay.com/acloud/captain-init:20210402174057-ER61887887_release_20210402.d399cc5
     ```

   - apyunqing_init 镜像

     ```
     acs-reg.alipay.com/acloud/apyunqingdbinit:ed1686e2-1614837535410
     ```

   - apyunqing-compose 镜像

     ```
     acs-reg.alipay.com/acloud/apyunqing-compose:20190905100331-f234450
     ```

   - akeplus-charts 镜像

     ```
     acs-reg.alipay.com/acloud/akepluscharts:0192d3ae-1616760965943
     ```

注意

- 域外交付时，为了满足高可用的要求，所有底座的基础服务都需要在两台 ops 机器分别部署。由于 ADNS 的端口默认是 80，Registry 的端口默认也是 80，导致 adns 和 Registry 无法部署到同一台机器，需要分别在两台 ops 机器部署。
- 本次演练环境只记录了底座拉起的操作步骤，并不满足高可用的要求，域外交付时，请根据现场情况规划部署。该演练环境 ops 节点使用规划如下：
  - ops1 用来部署 adns、Captain、Yunyou。
  - ops2 用来部署 Registry。
- 上传镜像时，请注意根据规划分别将要部署的产品镜像上传到 ops1 或 ops2。

初始化服务器

1. 在 ops1 机器生产秘钥对，并拷贝公钥到所有节点，实现 ops 机器到所有节点免密。

i. 创建 ssh-keypair。

命令如下：

```
ssh-keygen -t rsa -N '' -f /root/.ssh/id_rsa
cd /root/.ssh
cat id_rsa.pub >> authorized_keys
chmod 0700 authorized_keys
```

ii. 将 authorized_keys 拷贝到所有节点。

2. 在所有 osp 机器上配置 NTP 服务，保证所有节点时间一致。
3. 检查机器挂盘情况。

注意

- 您必须确保 `/home/t4` 目录单独挂盘。
- 本次迭代须保证除去系统盘和 `/home/t4` 单独挂载以外，节点有独立的磁盘可以做卷组。

3.部署底座

3.1. 部署 Docker

安装物理机底座时，您需要在两台 ops 机器上分别安装 Docker。

操作步骤

1. 下载 DockerCE 安装包。
2. 将安装包解压，然后进入 docker 文件夹执行安装命名。

```
yum localinstall  -y ./*
```

3. 安装成功后，修改 DockerCE 配置。

```
vi /usr/lib/systemd/system/docker.service
```

将 ExecStart 替换为如下内容：

```
ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock -s ove
rlay2 -g /var/lib/docker --live-restore --log-opt max-size=100m --insecure-registry=acs
-reg.alipay.com
```

4. 重启 Docker。

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

5. 修改 Containerd 配置。

> **注意**
>
> 如需将跳板机加入集群，必须执行此步骤。

```
mkdir -p /etc/containerd  && containerd config default | tee /etc/containerd/config.tom
l
sed -i 's/var\/lib\/containerd/home\/t4\/containerd/g' /etc/containerd/config.toml
systemctl daemon-reload
systemctl restart containerd
systemctl restart docker
```

6. 执行以下命令将之前上传到 ops 的镜像加载到本地。

```
 docker load -i {镜像名}
```

3.2. 部署 AntStack DNS

本文介绍如何部署 AntStack DNS 服务。

配置步骤

1. 在 ops 机器中增加 master 和 slave 配置文件。

   文件内容如下：

   - master.env

```
DNS_MASTER_INIT_ROLE=ACTIVE
DNS_DB_DRIVER=mysql
DNS_SLAVE_AGENT_PORT=15353
DNS_DB_CONN_URL=apyunqing:Antcloud2021@tcp(11.166.79.70:3306)/antdns?interpolateParam
s=True    ### apyunqing修改为自己环境的数据库用户名，Antcloud2021修改为自己环境的数据库密码
，11.166.79.70修改为自己数据库的url地址
DNS_MNG_PORT=80
DNS_MASTER_IPS=11.**.**.208,11.**.**.209    ### 请修改为自己环境 ops 机器的 IP。
DNS_SLAVE_IPS=11.**.**.208,11.**.**.209    ### 请修改为自己环境 ops 机器的 IP。
SELF_IP=11.**.**.208    ### 请修改为当前所在 ops 节点的 IP（ops1 和 ops2 要区分开）。
DNS_AGENT_PORT=15454
DNS_SERVICE_PORT=54
LOG_LEVEL=info
DNS_CLUSTER_NAME=default
KEEPALIVED_VIRTUAL_IPS=11.**.**.210,11.**.**.211    ### 请修改为自己环境 DNS master 的
VIP。
DNS_SLAVE_SERVICE_PORT=53
USE_VRRP=yes
DNS_DEFAULT_FORWARDERS=10.**.**.1    ### 物理机环境默认的 DNS，根据实际情况修改为自己环境系
统默认的即可。
DNS_PRESETUP_ZONES=antplus.com    ### 要注册的 zone 信息，后面部署 AKE 集群时会用到。根据
需要，修改为自己喜欢的 zone 即可。
DNS_UPDATE_ACL=192.**.**.0/24
LISTEN_ON_VIRTUAL_IPS=yes
KEEPALIVED_USE_UNICAST=no
KEEPALIVED_INTERFACE=bond0
```

> **注意**
>
> 您必须删除示例中的注释内容，文件内容中不要出现中文。

   - slave.env

```
SELF_IP=11.**.**.208    ### 请修改为当前所在 ops 节点的 IP（ops1 和 ops2 要区分开）。
DNS_MASTER_IPS=11.**.**.208,11.**.**.209    ### 请修改为自己环境 ops 机器的 IP。
DNS_SLAVE_IPS=11.**.**.208,11.**.**.209    ### 请修改为自己环境 ops 机器的 IP。
DNS_SERVICE_PORT=53
DNS_AGENT_PORT=15353
DNS_MASTER_SERVICE_PORT=54
DNS_MASTER_AGENT_PORT=15454
DNS_DEFAULT_FORWARDERS=10.**.**.1    ### 物理机环境默认的 DNS，根据实际情况修改为自己环境系统默认的即可。
DNS_PRESETUP_ZONES=antplus.com    ### 要注册的 zone 信息，后面部署 AKE 集群时会用到。根据需要，修改为自己喜欢的 zone 即可。
USE_VRRP=yes
KEEPALIVED_VIRTUAL_IPS=11.**.**.212,11.**.**.213    ### 请修改为自己环境 dnssalve 的 VIP。
LISTEN_ON_VIRTUAL_IPS=yes
TARGET=dnsagent
KEEPALIVED_USE_UNICAST=no
KEEPALIVED_INTERFACE=bond0
```

2. 在 ops1、ops2 机器上部署 ANDS。

   执行以下命令前，确认 DNS 镜像已存在。

   - 部署 dnsmaster:

```
docker run --net=host -d --restart=always --privileged -m=4G -l com.alipay.acs.container.cpus=2  --env-file=./master.env -v /opt/dnsmaster/named/:/var/named/chroot/ -v /home/admin/dnsmng/log/ -v /home/admin/dnsagent/log/ --name dnsmaster1  acs-reg.alipay.com/antdns/dnsmaster:1.9.0-9e5dc5b132f3
```

   - 部署 dnssalve:

```
docker run --net=host -d --restart=always --privileged -m=4G -l com.alipay.acs.container.cpus=2 --env-file=./slave.env  -v /opt/antdns/dnsslave/named/:/var/named/chroot/ -v /home/admin/dnsagent/log/  --name dnsslave1 acs-reg.alipay.com/antdns/dnsslave:1.9.0-a3e43bc54d47
```

3. 测试 DNS 的可用性。

   i. 将 3 台 master 节点注册域名。

```
curl -X POST -H "Content-Type: application/json" \
-d '{
"name": "apiserver.antplus.com.",    ### 修改为自己要注册的 APIServer 的域名。
"zone": "antplus.com",    ### 修改为自己发布 DNS 时注册的 zone。
"value": "11.**.**.202",    ### 修改为自己环境规划的 AKE 集群 master 节点的 IP 地址。
"ttl": 60,
"type": "A"
}' \
http://11.**.**.208:80/rr/addrr    ### 修改为自己环境 dnsmaster 的 IP 地址。
```

   ii. 修改 `/etc/resolv.conf` 文件中 nameserver 信息。

```
echo "nameserver 11.**.**.208" > /etc/resolv.conf    ### 修改为自己的 DNS 地址。
```

iii. 测试已注册域名的可用性。

使用 ping 命令测试刚注册的域名，观察是否可以正常解析。

```
ping apiserver.antplus.com
```

iv. 将 ops1 节点的 `/etc/resolv.conf` 文件复制到集群中所有节点。

3.3. 部署 Captain

本文介绍如何部署 Captain 控制台。

操作步骤

1. 在 ops1 部署 Captain 控制台。

   执行以下命令前，确认 Captain 控制台镜像已存在。

   - 部署 captain-init：

     ```
     docker run --rm --net host<captain-init 的镜像> bash -x initdb.sh<数据库域名或 IP><数据库端口><数据库用户名><数据库密码><数据库 schema>
     ```

     示例如下：

     ```
     docker run --rm --net host acs-reg.alipay.com/acloud/captainplusinit:c03bbde1-*******919  bash -x initdb.sh 11.**.**.70 3306 apyunqing Antcloud2021 captain
     ```

   - 部署 Captain：

     ```
     docker run --name<Captain 容器的名字> --net host -v /home/admin/data:/home/admin/data -v /var/run/:/host/var/run/ -v /tmp:/host/tmp -e app.env.app_root=/home/admin -e app.env.app_name=akeconsole -e app.env.log_level=info -e app.env.domain_name=localhost -e app.env.dbmode=dev -e app.env.db_url=<数据库 IP 地址和端口，形如**.**.**.**:3306) > -e app.env.db_username=<数据库用户名> -e app.env.db_connect_timeout=10000 -e app.env.db_socket_timeout=6000 -e app.env.db_max_active_connection=20 -e app.env.db_password=<数据库密码> -e app.env.db_schema=<数据库 schema> -e app.env.db_max_active_connection=100 -e app.env.registry_address=<镜像中心的 IP 地址和端口> -e app.env.cluster_name=default -e app.env.web_port=<Captain 页面的访问端口> -e app.env.docker_repo=acs-reg.alipay.com -it -d<Captain 的镜像> sh /opt/entrypoint.sh
     ```

     示例如下：

     ```
     docker run --name ake_console --net host -v /home/admin/data:/home/admin/data -v /var/run/:/host/var/run/ -v /tmp:/host/tmp -e app.env.app_root=/home/admin -e app.env.app_name=akeconsole -e app.env.log_level=info -e app.env.domain_name=localhost -e app.env.dbmode=dev -e app.env.db_url=11.**.**.70:3306 -e app.env.db_username=apyunqing -e app.env.db_connect_timeout=10000 -e app.env.db_socket_timeout=6000 -e app.env.db_max_active_connection=20 -e app.env.db_password=Antcloud2021 -e app.env.db_schema=captain -e app.env.db_max_active_connection=100 -e app.env.registry_address=11.166.85.218:80 -e app.env.cluster_name=default -e app.env.web_port=82 -e app.env.docker_repo=acs-reg.alipay.com -it -d acs-reg.alipay.com/acloud/captainplus:c03bbde1-1******985 sh /opt/entrypoint.sh
     ```

2. 测试 Captain 的可用性。

i. 在 ops 上执行 `docker ps`，观察 Captain 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://<Captain 地址:82>` 查看是否可以正常访问。

3.4. 部署 Registry

本文介绍如何部署 Registry。

> **注意**
>
> Registry 不能与 ADNS 部署在一个节点上。

操作步骤

1. 在 ops2 部署 Registry。

   执行以下命令前，请确认 Registry 和 apyunqing-compose 镜像已存在。

```
docker run --net=host -d -v  /home/yunyou/staticfiles:/home/staticfiles acs-reg.alipay.com/acloud/apyunqing-compose:20190905100331-f234450
tar -xvf /tmp/yunyou/staticfiles/apyunqing-compose.tar -C /home/t4
docker run -d  -v /home/t4/registry:/var/lib/registry  -v /home/t4/apyunqing-compose/registry-config.yml:/etc/docker/registry/config.yml --net=host --restart=always --name registry acs-reg.alipay.com/acloud/registry:1.1
```

2. 测试镜像中心是否可用。

   i. 在 `/etc/hosts` 文件中添加镜像中心的 hosts 解析。

```
echo "11.**.**.218 acs-reg.alipay.com" >> /etc/hosts  ## 11.**.**.218 修改为自身环境的镜像中心地址。
```

   ii. 确认是否可以正常推送镜像。

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

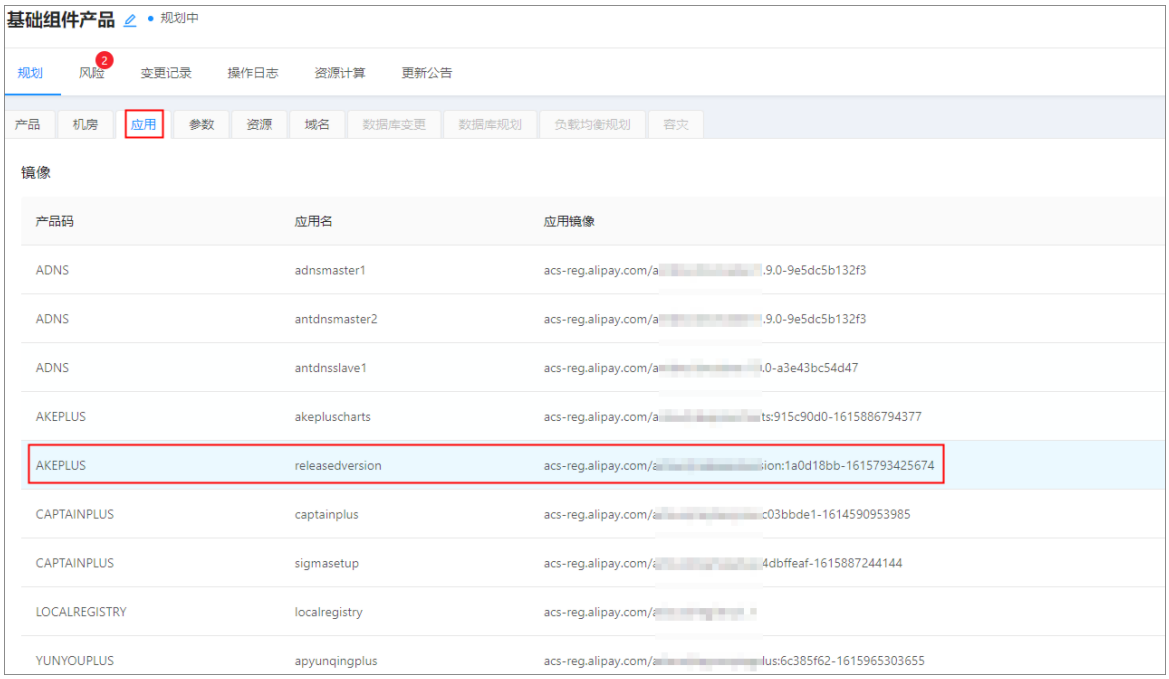
3. 确认镜像中心可用后，修改所有节点的 `/etc/hosts` 文件，添加镜像中心的 hosts 解析。

4.部署中枢集群

4.1. 获取 ReleaseVersion 相关资源

操作步骤

1. 在云游 Global 上获取 ReleaseVersion 镜像。



2. 在跳板机上执行以下命令获取镜像下载命令和二进制文件下载命令。

```
docker run  ${ ReleaseVersion 镜像地址}  -r antcloud -t meta-1.20-containerd -l
```

3. 下载镜像。

   在域内跳板机执行图片中 **导出依赖镜像** 显示的 `wget` 命令内容下载镜像。

```
wget http://alipay-offline.cn-hangzhou.alipay.aliyun-inc.com/mth/image.sh -O ~/image.sh
&& sh ~/image.sh -d '{"version":"1.0.0","targetRegistryDomain":"acs-reg.alipay.com","im
ages":["acs-reg.alipay.com/acloud/kube-apiserver-amd64:v1.20.0","acs-reg.alipay.com/acl
oud/kube-controller-manager:v1.20.0","acs-reg.alipay.com/acloud/kube-scheduler:v1.20.0"
,"acs-reg.alipay.com/acloud/machine-operator:0.1.3","acs-reg.alipay.com/acloud/cluster-
operator:v0.1.3","acs-reg.alipay.com/acloud/apyunqing-sigma-operator:20210311194117-mas
ter.ae9d8c2","acs-reg.alipay.com/acloud/kube-proxy:v1.20.0","acs-reg.alipay.com/acloud/
rama-manager:v0.2.1","acs-reg.alipay.com/acloud/rama-webhook:v0.2.1","acs-reg.alipay.co
m/acloud/rama-daemon:v0.1.0","acs-reg.alipay.com/acloud/etcd:v3.4.13-0","acs-reg.alipay
.com/cafe-kubernetes/coredns:v1.6.2","acs-reg.alipay.com/acloud/pause-amd64:3.1","acs-r
eg.alipay.com/acloud/cafecniservice:20200******5826-EI61586081_20200219.6*****52","acs-
reg.alipay.com/acloud/cni-plugins:201912021575278680-master.656e0e8c0","acs-reg.alipay.
com/acloud/flannel:v0.10.0","acs-reg.alipay.com/acloud/cni-plugins:201912021575278680-m
aster.656e0e8c0","acs-reg.alipay.com/acloud/flannel:v0.10.0"]}'
```

4. 将镜像拷贝到自己的 ops 机器上加载镜像，并将镜像推至自己的镜像中。
5. 下载依赖二进制文件。
   i. 在域内跳板机新建一个目录。
   ii. 进入目录执行图片中 **下载依赖二进制文件** 显示的 `wget` 命令内容下载二进制文件。

```
wget $(echo -e"http://alipay-offline.cn-hangzhou.alipay.aliyun-inc.com/mth/libsysco
nf-alipay-0.1.9-201******502.alios7.x86_64.rpm\nhttp://alipay-offline.cn-hangzhou.a
lipay.aliyun-inc.com/mth/kubernetes-cni-0.8.7-0.x86_64.rpm\nhttp://alipay-offline.c
n-hangzhou.alipay.aliyun-inc.com/mth/kubelet-1.20.0-0.x86_64.rpm\nhttp://alipay-off
line.cn-hangzhou.alipay.aliyun-inc.com/mth/containerd.io-1.3.7-3.1.el7.x86_64.tar.g
z\n")
```

6. 将下载的二进制文件拷贝到自己的 ops1 机器的 `/home/admin/data/` 目录。

4.2. 创建 ReleaseVersion 版本

操作步骤

1. 获取 ReleaseVersion Yaml 文件。

   在跳板机上执行以下命令：

```
docker run acs-reg.alipay.com/acloud/releasedversion:1a0d18bb-1615793425674 -r antcloud
-t meta-1.20-containerd -p http://11.**.**.208:8888/data  > releaseversion.yaml
 ##注意：将 11.**.**.208 修改为自己的 Captain 容器所在机器的 IP（ops1 机器的 IP），得到 relea
seversion.yaml 文件。
```

2. 将获取的 `releaseversion.yaml` 文件下载到本地，然后通过浏览器访问 Captain。
3. 在左侧导航栏单击 **版本**，然后单击 **创建**。

4. 单击 **上传已有的版本配置文件**，然后上传 `releaseversion.yaml` 文件。

5. 导入完成后，单击 **下一步**。
6. 按提示操作后，单击 **创建**。

4.3. 创建元集群

操作步骤

1. 登录 Captain 控制台。
2. 单击 **集群管理**，然后在 **创建集群** 下拉菜单单击 **新建元集群**。
3. 选择 **自定义**，然后单击 **下一步**。
4. 配置元集群的基本信息，然后单击 **下一步**。

参数	说明
集群名称	输入元集群名称。
DNS 域	输入 部署 AntStack DNS 步骤中规划的 zone。
API server 域名	API server 的域名由集群名和 DNS 域组成。 自动生成，若生成的域名与 部署 AntStack DNS 步骤中注册的 API server 的域名不一致时，需手动修改一致。
扩展 API Server 地址	其他用来访问 API Server 的地址，如额外挂载的负载均衡地址等。 没有扩展 API Server 时置空即可。
Service 子网	集群内 Service IP 的网段，例如 `10.0.0.0/24` 。 双机房部署时，注意网段不要相同。
使用外部 etcd	选择是否使用外部 etcd。

参数	说明
描述	填写元集群的描述信息。

5. 配置元集群的详细信息，然后单击 **下一步**。

参数	说明
版本配置	
安装包镜像	填写 获取 ReleaseVersion 相关资源 步骤获取的镜像。
集群版本套件	填写 创建 ReleaseVersion 版本 步骤创建生成的版本。
DNS 配置	DNS 插件能够根据 Service 自动注册 DNS 记录。默认开启。
DNS 控制器类型	选择 CoreDNS。
网络配置	
集群网络模式	选择 Rama。
节点主网卡	输入自己机器的主网卡名称。
启用 ebtables 特性	根据您集群的情况选择是否开启： ○ 集群是物理机时，建议关闭。 ○ 集群是虚拟机时，建议打开。
登录节点配置	
登录节点类型	选择 SSH 密钥。
SSH 端口	设置为 22。
用户名	输入 root。

参数	说明
私钥	填写 初始化服务器 步骤生成的所有节点都可以访问的私钥。
其他配置	
节点自定义脚本	参见下方自定义脚本内容。

> 说明
>
> 以上表格中未提及的配置均保持默认。

自定义脚本：

```
#---------------------------- AKE ----------------------------#
# AKE 维护：以下部分由 AKE 同学维护
# 1. 容器日志切割配置
cat > /etc/logrotate.d/docker-containers << EOF
# @xizhe Containerd::
/home/t4/sigma-slave/logs/kubelet.ERROR{
    daily
    size 500M
    missingok
    rotate 30
    nocompress
    copytruncate
    notifempty
}
/home/t4/sigma-slave/logs/kubelet.INFO{
    daily
    size 500M
    missingok
    rotate 30
    nocompress
    copytruncate
    notifempty
}
/home/t4/sigma-slave/logs/kubelet.WARNING{
    daily
    size 500M
    missingok
    rotate 30
    nocompress
    copytruncate
    notifempty
}
/home/t4/containerd/logs/containerd.log{
    daily
    size 500M
```

```
    missingok
    rotate 30
    nocompress
    copytruncate
    notifempty
}
EOF
# 2. AKE 依赖模块
ake_req_mod() {
   cat > /etc/modules-load.d/ake_req_mod.conf  << EOF
xt_conntrack
xt_addrtype
xt_comment
xt_nat
iptable_nat
iptable_filter
nf_defrag_ipv4
nf_nat_ipv4
nf_nat_masquerade_ipv4
nf_tables
nf_nat
nf_conntrack
ipt_MASQUERADE
# @xizhe Containerd
br_netfilter
overlay
EOF
}
# 3. AKE 依赖的其他服务
ake_req_service() {
    mkdir -p /home/t4
    # @宇志
    #yum install -b current -y libsysconf-alipay-0.1.8-20191028110732.alios7.x86_64
    # @昱盛
    systemctl stop nonf_conntrack
    systemctl disable nonf_conntrack
}
# 4. 关闭 networkManage service
disnetwork() {
    systemctl stop NetworkManager
    systemctl disable NetworkManager
}
ake_req_mod
ake_req_service
disnetwork
#---------------------------- ALB ----------------------------#
# ALB 维护：以下部分由 ALB 同学维护
touch /var/lock/subsys/local
ulimit -n 204800
/sbin/chkconfig --level 2345 irqbalance off
/etc/init.d/iptables  stop
/etc/init.d/irqbalance stop
/bin/sh /etc/quagga/set_irq_affinity.sh eth0
/bin/sh /etc/quagga/set_irq_affinity.sh eth1
```

```
#---------------------------- AKE&ALB ----------------------------#
# AKE&ALB 维护：因可能有参数冲突，以下部分由双方同学共同维护；涉及到参数修改部分，需双方协商。
set_sysctl() {
    # https://aone.alipay.com/project/532482/task/taskList?akProjectId=532482&#toPage=4
&pageSize=15&openTaskId=9642307
    cat << 'EOF' > /etc/sysctl.d/z99-alipay.conf
# para for alb/ake/tair/gaokeyong/sys-container/antq/middleware/
vm.max_map_count = 655360
vm.dirty_background_ratio = 10
vm.dirty_ratio = 20
vm.dirty_writeback_centisecs = 50
vm.dirty_expire_centisecs = 100
#vm.enable_legacy_mm=0
#vm.extra_free_kbytes=5242880
vm.page-cluster = 3
vm.min_free_kbytes = 204800
vm.overcommit_memory = 1
vm.swappiness = 0
vm.zone_reclaim_mode = 0
vm.oom_kill_allocating_task=1
vm.watermark_scale_factor = 200
net.ipv4.neigh.default.gc_thresh1=2048
net.ipv4.neigh.default.gc_thresh2=4096
net.ipv4.neigh.default.gc_thresh3=32768
net.ipv4.tcp_tw_timeout = 3
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 0
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_window_scaling = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_syn_retries = 2
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_retries1=3
net.ipv4.tcp_retries2=5
net.ipv4.ip_local_reserved_ports = 8080,8899,8888,8443,12200,19211,9001,8902
net.ipv4.ip_local_port_range = 10000 65000
# @xizhe Containerd
net.bridge.bridge-nf-call-iptables  = 1
net.ipv4.ip_forward                 = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.netfilter.nf_conntrack_max = 3000000
net.core.netdev_max_backlog = 30000
net.core.somaxconn = 51200
net.ipv4.tcp_max_syn_backlog = 204800
fs.inotify.max_user_watches = 1524288
fs.inotify.max_user_instances = 8192
fs.inotify.max_queued_events = 32768
kernel.core_pattern = /home/admin/logs/core.%e.%p.%h.%t
kernel.printk = 6 4 1 6
## namespaced sysctl
# kernel.msgmax
# kernel.msgmnb
# kernel.msgmni
```

```
# kernel.sem
# kernel.shmall
# kernel.shmmax
# kernel.shmmni
# kernel.shm_rmid_forced
# fs.mqueue.*
# net.*
EOF
    ##make sure nf_conntrack_max=1500000 when nf_conntrack module insmoded
    #spanner:https://yuque.antfin-inc.com/58-alb/dev-docs/spanner-kernel-conf
    echo "options nf_conntrack hashsize=375000" > /etc/modprobe.d/nf_conntrack.conf
    #requirement from ake2.0:
    echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
    echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
}
set_sysctl
#---------------------------- 系统运维 ----------------------------#
# 系统运维：以下由系统运维同学维护
# 1. 系统级日志切割配置
cat > /etc/logrotate.d/syslog-ng << EOF
/var/log/cron /var/log/maillog /var/log/messages /var/log/secure /var/log/spooler /var/
log/kern /var/log/boot.log
{
    daily
    rotate 7
    compress
    dateext
    missingok
    sharedscripts
    postrotate
        /bin/kill -HUP \$(cat /var/run/syslog-ng.pid 2> /dev/null) 2> /dev/null || true
    endscript
}
EOF
# 2. 系统用户初始化
ensure_user() {
    local user=$1
    local uid=$2
    local key_name=${user}_authed_key
    local key=${!key_name}
    local home
    if [[ $user == root ]]; then
        home=/root
    else
        home=/home/$user
    fi
    groupadd -g $uid $user &> /dev/null
    useradd -u $uid -g $uid -m -s /bin/bash $user &> /dev/null
    chmod 0755 $home
    chown -R $user:$user $home
}
# 3. limit 限制
set_limit() {
    cat << 'EOF' > /etc/security/limits.d/90-alipay.conf
```

```
admin    -    core  unlimited
root     -    core  unlimited
root     -    nofile  204800
admin    -    nofile  204800
root     -    nproc  1966050
admin    -    nproc  1966050
root     -    memlock unlimited
admin    -    memlock unlimited
EOF
}
ensure_user root 0
ensure_user log 505
ensure_user admin 500
set_limit
#-------------------------- END --------------------------#
```

6. 单击 **添加中控节点**，然后在 **添加中控节点** 面板配置以下参数：

参数	说明
IaaS 底座类型	选择 **AntStack 私有云**。
单个添加/批量导入	开启此项。
IP 列表	输入规划的 master 节点 IP 地址。
调度状态	选择 **允许调度**。
服务器角色	选择 **应用服务器**。
网络	置空此项。
互斥组	置空此项。
opsServer	关闭此项。

7. 单击 **确定**，然后单击 **预检**。
8. 预检全部通过后，单击 **创建**。

> **注意**
>
> 如果集群部署失败，请清理集群后重新创建集群，重建的集群名最好不要跟失败的完全一致。

4.4. 配置集群网络

集群创建后，您需要为集群配置网络。

操作步骤

1. 登录 Captain 控制台。
2. 单击 **集群管理**，然后单击目标集群。
3. 添加网络。
   i. 在左侧导航栏单击 **网络**，然后单击 **创建网络**。
   ii. 输入网络名称和 netID，然后单击 **确定**。

   域内物理机环境中，netID 默认使用 701。

   iii. 单击 **网络池** 页签，然后单击 **添加 Pool**。
   iv. 在 **添加 Pool** 面板配置以下参数：

参数	说明
Subnet Name	输入子网名称。
开始 IP	输入地址池开始 IP。
结束 IP	输入地址池结束 IP。
网关	输入子网网关。
子网	输入子网的网络地址，使用 CIDR 格式。例如：`10.0.0.0/24`。
macNatEnable	表示该容器网段是否需要伪装所在节点的 MAC。 如果在虚拟机场景下，且有 MAC 过滤的限制，则该开关需要开启，否则关闭即可。

> **注意**
>
> - 添加的子网必须满足与用到的物理机网络在同一个大网下。
> - 如果用到的物理机是不同网段的，那么添加网络也要对应创建多个网络，并添加对应的子网，然后将网络标签打到对应的机器。

   v. 单击 **确定**。
4. 添加网络标签。
   i. 在左侧导航栏选择 **集群管理** > **节点（Nodes）**。

ii. 选中目标节点，然后选择 **批量操作** > **修改标签**。

iii. 单击 **批量增加/覆写**。

- **新增键**：填写 network.alipay.com/network。
- **值**：填写步骤 3 创建的子网名称。

iv. 单击 **提交**。

> **注意**
>
> 因为之前创建集群时，还没有创建网络，master 节点没有打网络的标签，所以 master 节点也需要参考上面步骤添加一下网络标签。

4.5. 扩容节点

注意事项

- 请确保添加到集群中的节点有机器有单独的磁盘可以做卷组。
- 制作卷组时，如果机器只有一种 SSD 或 HDD 磁盘时，也可以只创建一种卷组。卷组制作请参考 Yoda 本地磁盘初始化。

操作步骤

1. 登录 Captain 控制台。
2. 单击 **集群管理**，然后单击目标集群。
3. 选择 **集群管理** > **节点（Nodes）**。
4. 单击 **扩容节点**。
5. 选择 **通用方式**，然后单击 **添加工作节点**。
6. 在 **添加工作节点** 面板配置以下参数后，单击 **确定**。

参数	说明
IaaS 底座类型	选择 AntStack 私有云。
单个添加/批量导入	开启
IP	输入规划的 master 节点 IP 地址。
调度状态	选择 **允许调度**。
服务器角色	选择 **应用服务器**。

参数	说明
网络	输入 配置集群网络 步骤创建的子网名称。
互斥组	置空此项。
opsServer	关闭此项。
自定义标签	置空此项。

7. 单击 **提交**。

4.6. 为节点添加标签

为保证集群的高可用，您需要为物理机节点添加标签。

打标规则

规划好整个 Region 的可用区，例如 A 机房所有的机器添加 Zone-A 标签，B 机房所有的机器添加 Zone-B 标签。

操作步骤

1. 划分可用区。

   当前 IDC 有机房 A 和机房 B 两个机房，您可以按照自定义规则划分机，例如将整个 IDC 划分为可用区 A 和可用区 B：

   - 可用区 A：包含所有机房 A 的机器。
   - 可用区 B：包含所有机房 B 的机器。

2. 给集群节点添加标签。

   您可以通过 Captain 控制台或命令行为节点打标。

   - Captain 控制台

     a. 登录 Captain 控制台。

     b. 选择 **集群管理** > **节点（Nodes）**。

     c. 批量选中机房 A 的节点，然后选择 **批量操作** > **修改标签**。

     d. 单击 **批量添加/覆写**，填写标签键和值。

     - 键：填写 topology.kubernetes.io。
     - 值：填写 Zone-A。

       实际环境中，请根据自己的需求填写标签值。

     e. 单击 **确定**。

   - 命令行

     输入以下命令为节点添加标签：

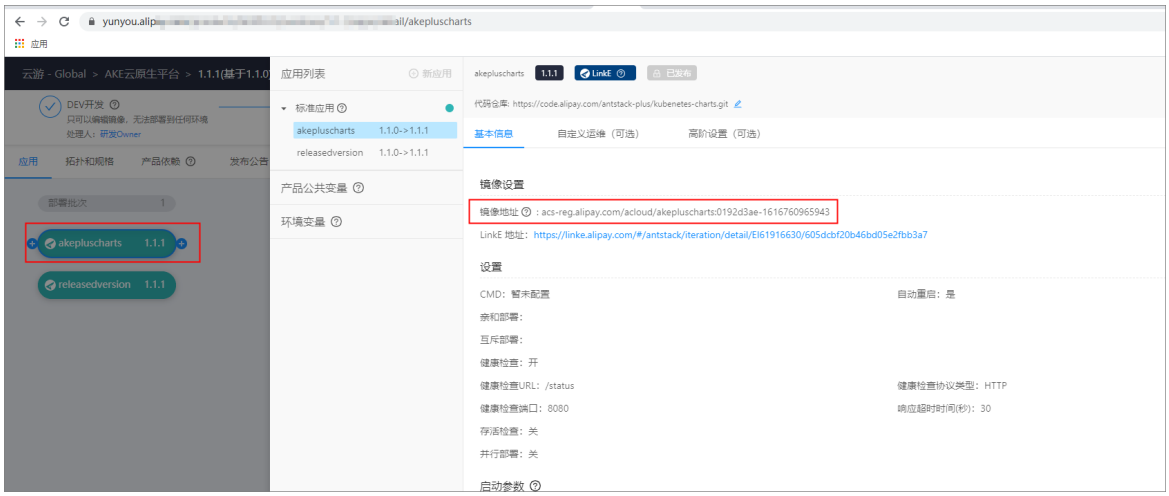
```
kubectl label node <节点名称>   topology.kubernetes.io/zone=Zone-A
```

4.7. 部署 charts 插件

部署charts 插件时部署的组件包括：Yoda、底座云原生监控（prom+grafana）、minio（对象存储）、npd（node problem detector，节点问题自动发现）、openkruise（webhook自动管理）等。

步骤一：获取 chart 包

1. 登录云游 Global。
2. 在左侧导航栏选择 **产品研发 > 产品版本管理**，然后单击 **全部产品**。
3. 找到并单击 **AKE 云原生平台**，然后单击 **产品版本**。
4. 单击 **1.X.X 版本**，然后单击目标版本名称。
5. 获取对应版本的 akepluscharts 的镜像地址。



6. 在域内跳板机上下载 charts 文件。

   命令如下：

```
docker run -v $PWD/.:/helm_data  <akepluscharts 镜像地址>
```

7. 将获取到的 charts 文件复制到本地。

步骤二：获取镜像

1. 在域内跳板机上获取镜像列表。

```
docker run <akepluscharts 镜像地址>
```

```
[root@OceanBase166085097.sga.eu95 /data/dzw/bbbb]
#docker run acs-reg.alipay.com/acloud/akepluscharts:[redacted] images
WARNING: IPv4 forwarding is disabled. Networking will not work.
project root: /root/kubernetes-charts
templating grafana ...
templating kube-prometheus-stack ...
templating kube-state-metrics ...
templating minio ...
templating node-problem-detector ...
templating openkruise ...
templating post-config ...
templating prometheus-node-exporter ...
templating yoda ...
acs-reg.alipay.com/acloud/alertmanager:v0.21.0
acs-reg.alipay.com/acloud/bats:v1.1.0
acs-reg.alipay.com/acloud/busybox:1.31.1
acs-reg.alipay.com/acloud/csi-node-driver-registrar:v1.3.0
acs-reg.alipay.com/acloud/csi-plugin:v1.18.8.45-512134ad-aliyun
acs-reg.alipay.com/acloud/csi-provisioner:v1.6.0-b6f763a43-ack
acs-reg.alipay.com/acloud/csi-resizer:v0.1.0
acs-reg.alipay.com/acloud/grafana:7.3.5
acs-reg.alipay.com/acloud/k8s-sidecar:1.1.0
acs-reg.alipay.com/acloud/kruise-manager:v0.7.0
acs-reg.alipay.com/acloud/kube-state-metrics:v1.9.7
acs-reg.alipay.com/acloud/kube-webhook-certgen:v1.5.0
acs-reg.alipay.com/acloud/mc:RELEASE.2020-11-25T23-04-07Z
acs-reg.alipay.com/acloud/minio:RELEASE.2020-12-03T05-49-24Z
acs-reg.alipay.com/acloud/node-exporter:v1.0.1
acs-reg.alipay.com/acloud/node-problem-detector:v0.8.5
acs-reg.alipay.com/acloud/prometheus-operator:v0.44.0
acs-reg.alipay.com/acloud/prometheus:v2.22.1
acs-reg.alipay.com/acloud/yoda-amd64:1.1.2-777e77c
acs-reg.alipay.com/acloud/prometheus-config-reloader:v0.44.0
```

2. 将镜像列表中的镜像下载到域内跳板机，并上传至自己环境的镜像中心。

步骤三：将 chart 包上传到应用商店

1. 登录 Captain 控制台。
2. 单击 **集群管理**，然后单击目标集群。
3. 选择 **应用商店** > **产品模板**。
4. 单击 **上传产品模板**，将获取的 charts 文件上传。

步骤四：安装 charts

1. 发布产品。

   发布的产品名称与应用模板名称需保持一致。发布产品的对应空间和发布顺序如下：

   - Yoda、minio，post-config、node-problem-detector 安装在 kube-system 下。
   - Grafana、kube-prometheus-stack、prometheus-node-export、kube-state-metrics 部署在 kube-plugin 命名空间。
   - kruise 部署在 default 下。
   - 发布顺序：Yoda > prometheus-node-export > kube-state-metrics > Grafana > kube-prometheus-stack > minio > node-problem-detector > post-config > kruise。

   以发布 Yoda 为例，操作步骤如下：

   i. 在 Captain 控制台的左侧导航栏，选择 **应用商店** > **产品实例**。

   ii. 在 **发布列表** 页面的 **命名空间** 下拉框，选择 kube-system 命名空间。

iii. 单击 **创建发布**，并配置以下参数：

参数	说明
产品名称	设置为 Yoda。
应用模板	选择 Yoda。
命名空间	无需配置。
版本	选择待发布应用的版本。
描述	为发布的应用添加描述信息。

iv. 单击 **提交**。

v. 单击 **开始/重新发布**，然后单击 **确定**。

按照上面的发布要求，依次在对应空间发布对应产品。

4.8. 对外暴露 Grafana 服务

您可以通过以下两种方式对外暴露 Grafana 服务：

NodePort方式

1. 创建 Service。

```
kubectl apply -f service.yaml -n kube-plugin
```

`service.yaml` 的内容如下：

```
apiVersion: v1
kind: Service
metadata:
  name: grafana-node-service
  labels:
    name: grafana-node-service
spec:
  type: NodePort       #代表类型为 NodePort。
  ports:
  - port: 3000         #端口和 clusterIP 对应，即 ip:8080。供内部访问。
    targetPort: 3000   #端口一定要和 container 暴露出来的端口对应。
    protocol: TCP
    nodePort: 31111    # 所有的节点都会开放此端口，此端口供外部调用。端口号大于 30000。
  selector:
    app.kubernetes.io/instance: grafana #选择器一定要选择容器的标签。
    app.kubernetes.io/name: grafana
```

2. 访问集群中任意一个 NodeIP:32222。
   该端口可自定义，但是需要符合规范。不要和其他的服务的 NodePort 冲突。

PodIp方式

如果您的容器网络和内部网络已打通，可以直接访问 `podIp:3000` 。

4.9. 底座产品域名透出

AKE 3 集群部署完成后，需要将底座基础产品的域名透出。

当前需要暴露以下外部可访问的域名：

- ake-grafana.{domain name}
- ake-prometheus.{domain name}
- ake-minio.{domain name}

其中， `domain name` 为 部署 AntStack DNS 步骤中填写的 zone。

操作步骤

1. 获取域名需要绑定的 IP 地址。
   - 获取 grafana 需要绑定的 IP，命令如下：

     ```
     kubectl get pod -nkube-plugin -o wide|grep "grafana"
     ```

     ```
     [root@h[redacted]u95 /root]
     #kubectl get pod -nkube-plugin -o wide|grep grafana
     grafana-575896569f-ds8hw                    2/2     Running   0          6d    11.[redacted].89    100.88.107.238   <none>           <none>

     [root@[redacted]u95 /root]
     #
     ```

   - 获取 prometheus 需要绑定的 IP，命令如下：

     ```
     kubectl get po -nkube-plugin -owide|grep "prometheus-stack-prometheus"
     ```

     ```
     [root@[redacted]5 /root]
     #kubectl get po -nkube-plugin -owide|grep "prometheus-stack-prometheus"
     prometheus-kube-prometheus-stack-prometheus-0       2/2     Running   1          6d1h   11[redacted]88    100.88.107.229   <none>           <none>

     [root@[redacted]95 /root]
     #
     ```

   - 获取 minio 需要绑定的 IP，命令如下：

     ```
     kubectl get po -nkube-system -owide |grep "minio"
     ```

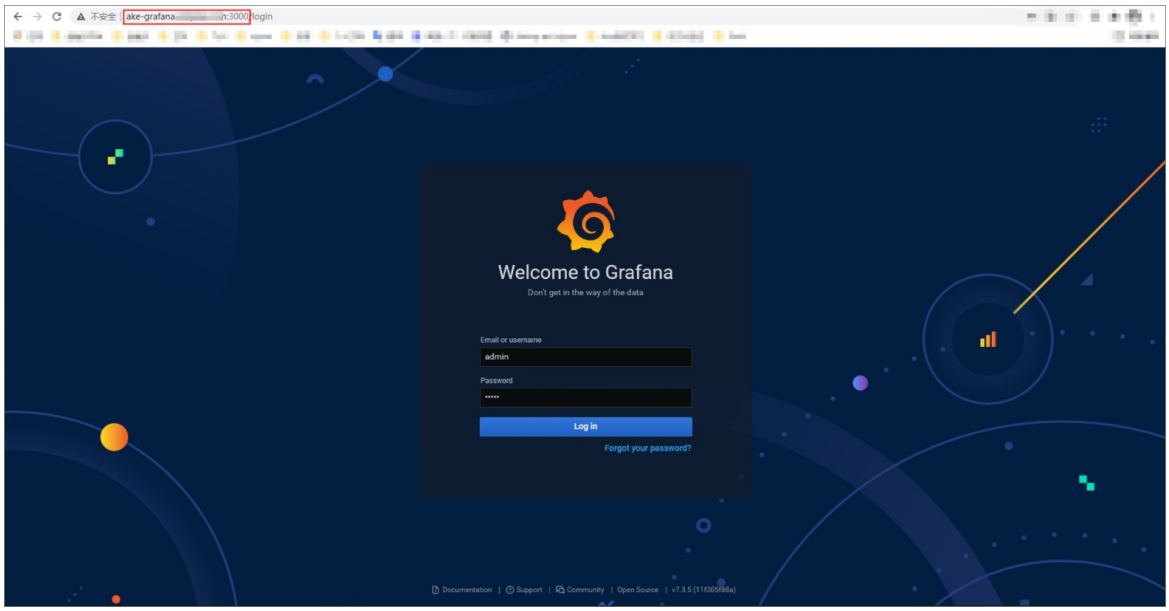
     ```
     [root@[redacted]u95 /root]
     #kubectl get po -nkube-system -owide |grep "minio"
     minio-0                                     1/1     Running   0          6d1h   11.[redacted].83    100.[redacted].229   <none>           <none>
     minio-1                                     1/1     Running   0          6d1h   11.[redacted].84    100.[redacted].229   <none>           <none>
     minio-2                                     1/1     Running   0          6d1h   11.[redacted].85    100.[redacted].229   <none>           <none>
     minio-3                                     1/1     Running   0          6d1h   11.[redacted].86    100.[redacted].229   <none>           <none>

     [root@[redacted]95 /root]
     #
     ```

2. 在本地 hosts 文件内，将需要绑定的域名和对应的 IP进行绑定。

   其中，minio 绑定获取的 4 个 IP 中的任意一个即可。

3. 测试 Grafana 是否可以访问。

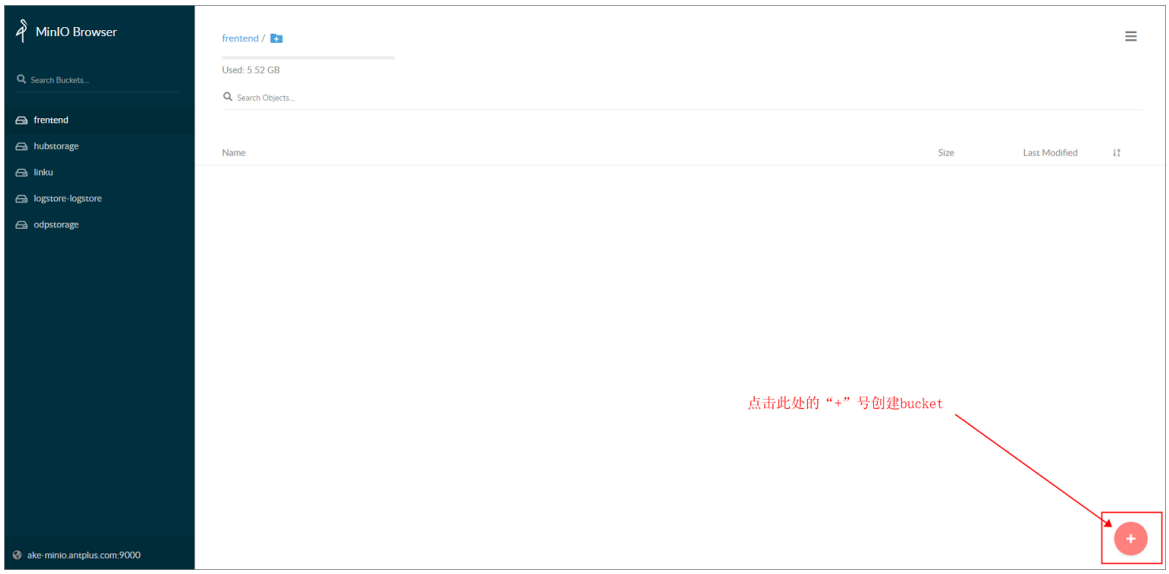
使用浏览器访问 `http://ake-grafana.{domain name}:3000` ，默认账号密码均为 admin。



4. 测试 minio 是否可以访问。

   使用浏览器访问 `http://ake-minio.{domain name}:9000` ，默认账号为：minio；默认密码为：awesome-minio。

5. 创建 bucket。
   在 minio 上创建 bucket，以备后面发布产品时使用。



5.部署云游

5.1. 部署云游

在 ops1 部署云游前，请确认 yunyou 和 apyunqing-compose 镜像已存在。

操作步骤

1. 初始化云游 db。

   i. 创建云游文件夹。

   ```
   mkdir -p  /home/yunyou/staticfiles
   ```

   ii. 下载 apyunqing-compose 镜像。

   ```
   docker run --net=host -d -v /home/yunyou/staticfiles:/home/staticfiles acs-reg.alip
   ay.com/acloud/apyunqing-compose:20190905100331-f234450
   ```

   iii. 解压 `apyunqing-compose.tar` 文件。

   ```
   cd /home/yunyou/staticfiles
   tar zxvf apyunqing-compose.tar
   ```

   iv. 修改 `config.json` 文件。

   ```
   cd apyunqing-compose
   vim config.json
   ```

   将下图中标记的内容修改为自身环境的数据库信息。

   v. 初始化数据库。

   ```
   python yunqing-setup.py i --image acs-reg.alipay.com/acloud/apyunqingdbinit:ed1686e
   2-16148375
   ```

2. 部署 apyunqing。

   ```
   python yunqing-setup.py y --image acs-reg.alipay.com/acloud/apyunqingplus:9c22893d-1616
   145099566
   ```

3. 部署云游 Webhook。
   i. 下载 Deployment 文件，并将文件拷复制到其中一台 master 节点中。
   ii. 部署 Webhook。

   命令如下：

```
kubectl create ns yunyou
tar -xf yunyou_webhook.tar
sh deployment/webhook-create-signed-cert.sh --service env-injector-webhook-svc --se
cret env-injector-webhook-certs --namespace yunyou
cat deployment/mutatingwebhook.yaml | deployment/webhook-patch-ca-bundle.sh > deplo
yment/mutatingwebhook-ca-bundle.yaml
kubectl create -f deployment/configmap.yaml  -n yunyou
kubectl create -f deployment/deployment.yaml  -n yunyou
kubectl create -f deployment/service.yaml  -n yunyou
kubectl create -f deployment/mutatingwebhook-ca-bundle.yaml  -n yunyou
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
```

4. 测试云游的可用性。
   i. 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://{云游地址:81}` 查看是否可以正常访问。

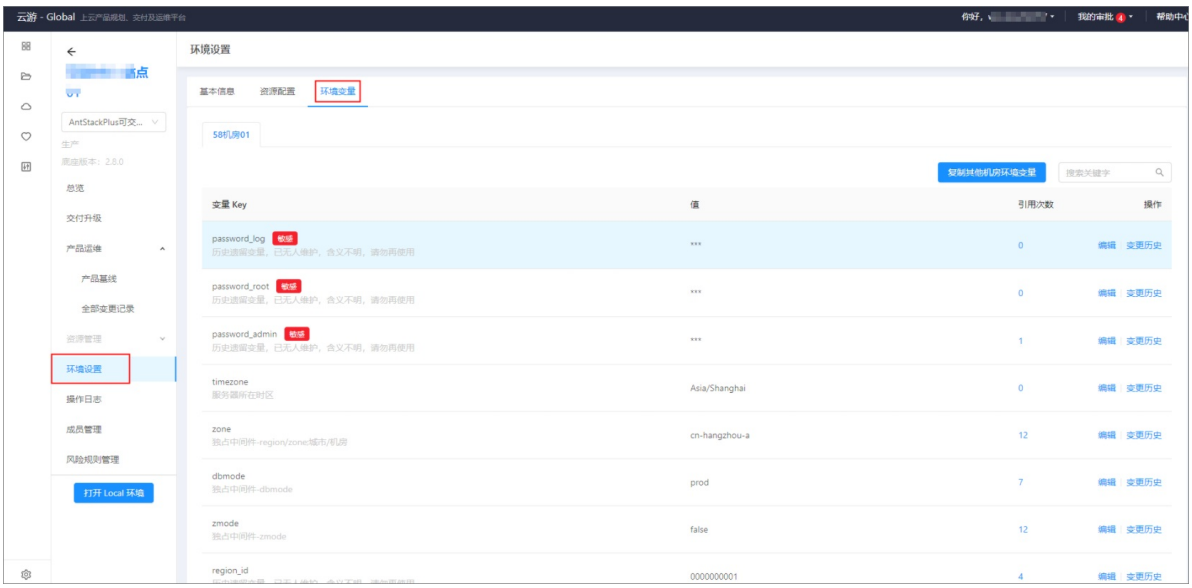
5.2. 添加新站点

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	bk9KOXNhclByM1h5ZEphWVNMMDE5RDkwRTRGMTA2NzcxQjc4OEVFNjU1QUVBMjQzMERERjYwNTgzODQxRjI0NUU2OTQ1QQ==	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	M0M0QUE4NDk3QzdFREYyM0FFNjhERjU4RDlBMDM1MzEwMDFGMjg2QzU1RUYxMTcxNjJGRkM0MDlDQ0Y0RDI2OQ==	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

keyseed 和 seed 生成步骤：

i. 登录云游 Local。

ii. 单击 **环境设置**，然后单击 **全局参数**。

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。

操作步骤

1. 获取云游解决方案。
   i. 登录云游 Global。
   ii. 在左侧项目列表，单击目标项目，然后单击目标环境。
   iii. 单击 **交付升级**，然后单击 **解决方案**。
   iv. 单击目标解决方案，然后单击 **添加产品**，添加身份访问管理 IAM 和 开放平台 OP 两个产品。
   v. 选择右上角 **导出交付物** > **解决方案包**。
2. 将刚导出的解决方案导入到云游 Local 中。
   i. 访问 `http://{yunyou地址:81}` 。
   ii. 单击 **交付升级**，然后单击 **导入解决方案**。
   iii. 按流程导入解决方案包。
   iv. 配置基础环境后，单击 **下一步**。

   其中 **主机房策略** 选择 **独立机房**；**资源名前缀** 填写自定义的容器、数据库实例等名称的前缀。

v. 配置资源池参数后，单击 **下一步**。

参数配置如下：

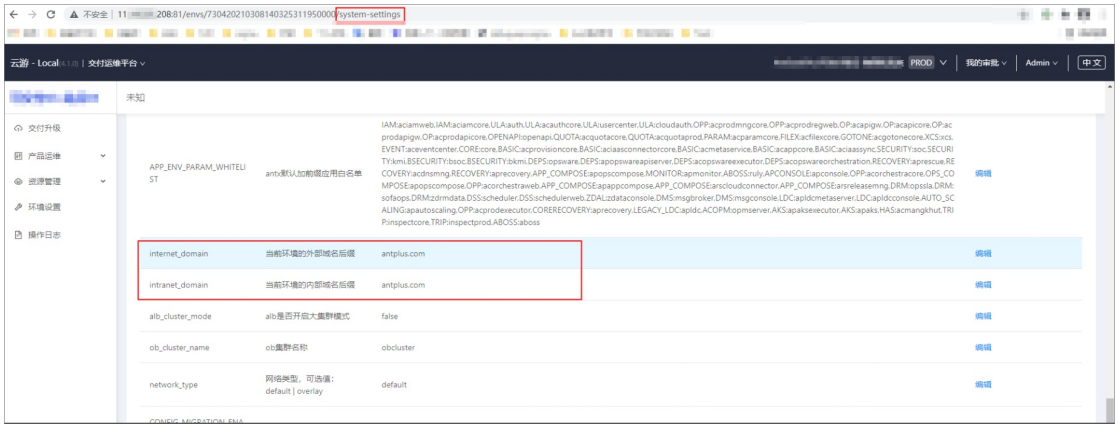
参数	说明
Kubernetes 版本	选择 Kubernetes。
k8s 提供商	选择 **第三方**。
策略	选择 **导入新集群**。
Kubernetes 集群证书（Kubeconfig）	填写从 master 节点的 `/etc/kubernetes/kubeconfig/admin.kubeconfig` 目录下获取的 kubeconfig。
数据库资源池配置	选择 **通过云游部署**。
负载均衡资源池配置	选择 **通过云游部署**。
文件存储资源池配置	选择 **通过云游部署**。

vi. 配置系统配置。

直接单击 **确认** 即可。系统配置可在导入解决方案后修改。

3. 修改站点系统环境变量。

i. 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：



ii. 配置环境域名后缀。

将 internet_domain 和 intranet_domain 项修改为 部署 AntStack DNS 步骤中填写的 zone。

iii. 配置持久化存储类映射。

将 storage_io_level_to_class 项修改为：

```
{"normal":"yoda-lvm-default","medium":"yoda-lvm-default","high":"yoda-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。

6.部署核心产品

6.1. 产品版本和拓扑规格

产品名	版本	拓扑	规格
IAM	2.31.1	私有云建站部署(全量PaaS平台)	标准生产-单机房
OP	0.26.0	专有云模式部署	标准生产-单机房

6.2. 部署身份访问 IAM

IAM 需要进行两次发布：

- 第一次发布：第一次发布是 standalone 模式，无需修改参数，直接在云游 Local 中发布即可。步骤可以参考 部署 OpenAPI（OP）。
- 第二次发布：微服务发布以后，重新渲染 IAM 的参数，变为 fullstack 模式。此时需要二次发布 IAM，并修如下参数重跑 iam-init。

应用名	键	值模板
iam-init	aliyun_account	NOT_NEED
aciamcore	deploy_mode	private
aciamweb	deploy_mode	private
iam-init	deploy_mode	private

IAM 发布完成后，您还需要注册 IAM 的域名 `admin.{domain_name}` ，IP 地址对应为 iamweb 的内网 VIP。

> **注意**
>
> 物理机环境的 IP 是容器 IP，不是 Service 的 VIP。

6.3. 部署 OpenAPI（OP）

本文介绍如何部署 OpenAPI（OP）服务。

操作步骤

1. 登录云游 Local。
2. 单击 **交付升级**，然后单击 OP 的解决方案。
3. 单击 **解决方案元数据**，然后单击 **参数**。
4. 根据您的需求修改 new_site 参数。

   找到 new_site 参数，然后单击 **编辑**，根据您的需求修改 **值模板**。

   您可以通过搜索栏快速查找 new_site 参数。
5. 单击 **解决方案部署**，然后单击 **开放平台 OP** 右侧的 **发布**。
6. 发布完成后，单击 **测试**。
7. 单击 **开始测试**，然后单击 **确定**。

   完成测试后，可单击 **测试单**，查看测试详情。

7.部署 SOFA 产品

7.1. 产品版本和拓扑规格

产品名	版本	拓扑	规格
ZooKeeper	3.15.0	标准单机房部署	标准生产-单机房
Registry	2.6.0	标准单机房部署	标准生产-单机房
ACVIP	0.20.0	标准单机房部署	标准生产-单机房
OSP	1.19.2	标准单机房部署	标准生产-单机房
IntelliProxy	1.10.4	标准单机房部署	标准生产-单机房
MS	2.40.0	标准单机房部署	标准生产-单机房

7.2. 部署 ZooKeeper

发布 ZooKeeper 前，您可以根据需求修改如下参数，步骤可以参考 部署 OpenAPI（OP）。

应用名	键	值模板	描述
ZooKeeper	INDEX_START_FROM_ZERO	true	配置从环境变量获取的 index 是否从 0 开始。ACK 模式必须填写。 大小写敏感。

参数修改完成后，直接在云游 Local 发布即可。

7.3. 部署 Registry

创建 Registry 发布单时，部署策略选择 **共分一组-无需确认**，然后将 meta-server 调整到第一组发布。发布步骤可参考 部署 OpenAPI（OP）。

发布完成后，您需要进入 meta-server 其中一个容器，手动执行 `curlhttp://localhost:9615/stopPushDataSwitch/close` 。执行完该任务才能去跑测试用例。

7.4. 部署 ACVIP

无需修改参数，直接发布即可。发布步骤可以参考 部署 OpenAPI（OP）。

7.5. 部署 OSP

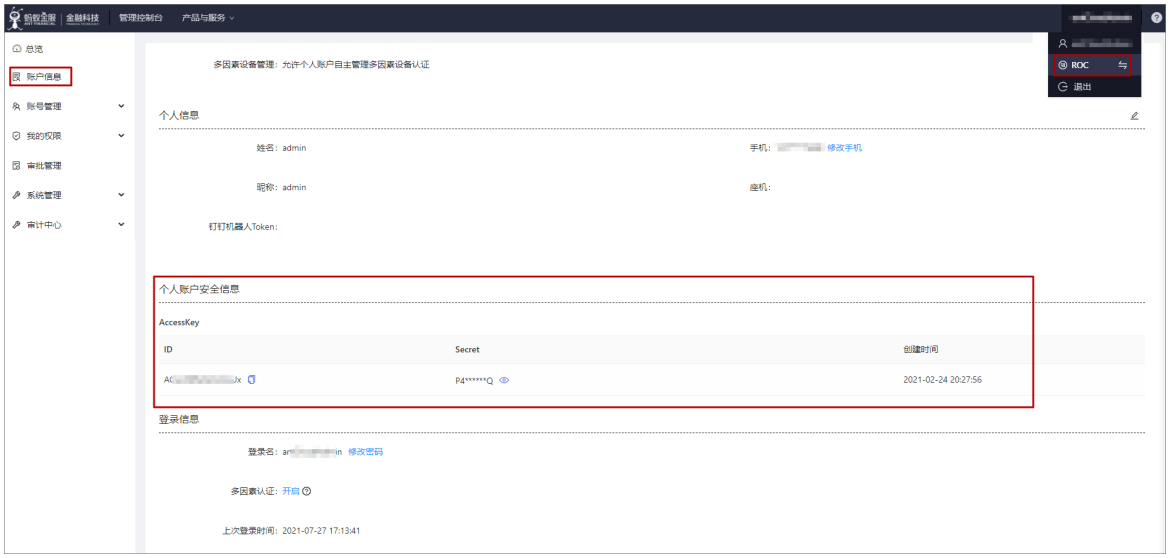
发布 OSP 前，您可以根据需求修改如下参数，步骤可以参考 部署 OpenAPI（OP）。

应用名	键	值模板	描述
ospaccoreinit	global_skip	false	在某些没有 OP 和 IAM 的环境，或希望人工执行元数据导入的环境（怕引发变更风险），可以将该值配置为 true，否则一律为 false。
osp	openapi.paasAccesskey	ACswUHb****	IAM 的 ROC 租户的 AccessKey ID。获取步骤如下所示。
osp	openapi.paasAccesssecret	tHtC4HoqM2Gz2QnRsbE****	IAM 的 ROC 租户的 AccessKey Secret。

参数修改完成后，直接在云游 Local 发布即可。

获取 AccessKey

1. 登录 IAM 控制台。
2. 选择租户环境为 ROC。
3. 在左侧导航栏，单击 **租户信息**，然后在 **个人账户安全信息** 区域查看 AccessKey 信息。



7.6. 部署 IntelliProxy

发布 IntelliProxy 时，无需修改参数，直接发布即可。发布步骤可以参考 部署 OpenAPI（OP）。

发布完成后，您需要在 ADNS 中注册如下域名：

```
192.**.**.104  sofa-osp.antplus.com
192.**.**.104  sofa-mq.antplus.com
192.**.**.104  sofa-ms.antplus.com
192.**.**.104  sofa-ts.antplus.com
192.**.**.104  sofa-dtx.antplus.com
192.**.**.104  sofa-dbp.antplus.com
192.**.**.104  sofa-sofamq.antplus.com
192.**.**.104  sofa-apigateway.antplus.com
```

IP 地址为 IntelliProxy 的内网 VIP，请根据实际环境修改。

> **注意**
>
> 物理机环境的 IP 是容器 IP，不是 Service 的 VIP。

7.7. 发布 MS

发布 MS 前，您可以根据需求修改如下参数，步骤可以参考 部署 OpenAPI（OP）。

应用名	键	值模板	描述
tsaccoreinit	global_skip	false	在某些没有 OP 和 IAM 的环境，或希望人工执行元数据导入的环境（怕引发变更风险），可以将该值配置为 true，否则一律为 false。
dsrconsole	ldc.switch	true	表示当前部署的环境是不是 LDC 环境。
antscheduler	cell.monitor.start	true	共享 LDC 模式中，是否开始 CELL 信息同步功能，依赖 OSP 1.19.0。

参数修改完成后，直接在云游 Local 发布即可。发布完成后，您需要访问 OSP 控制台，确保中间件租户下 middleware 和 core 工作空间的 MS 已启动。

7.8. 部署 ODP

发布部署

发布 ODP 前，您可以根据需求修改如下参数，步骤可以参考 部署 OpenAPI（OP）。

- 产品公共参数

产品名称	键	值模板	描述
ODP	ldc_enabled	false	是否为 IDC 环境。

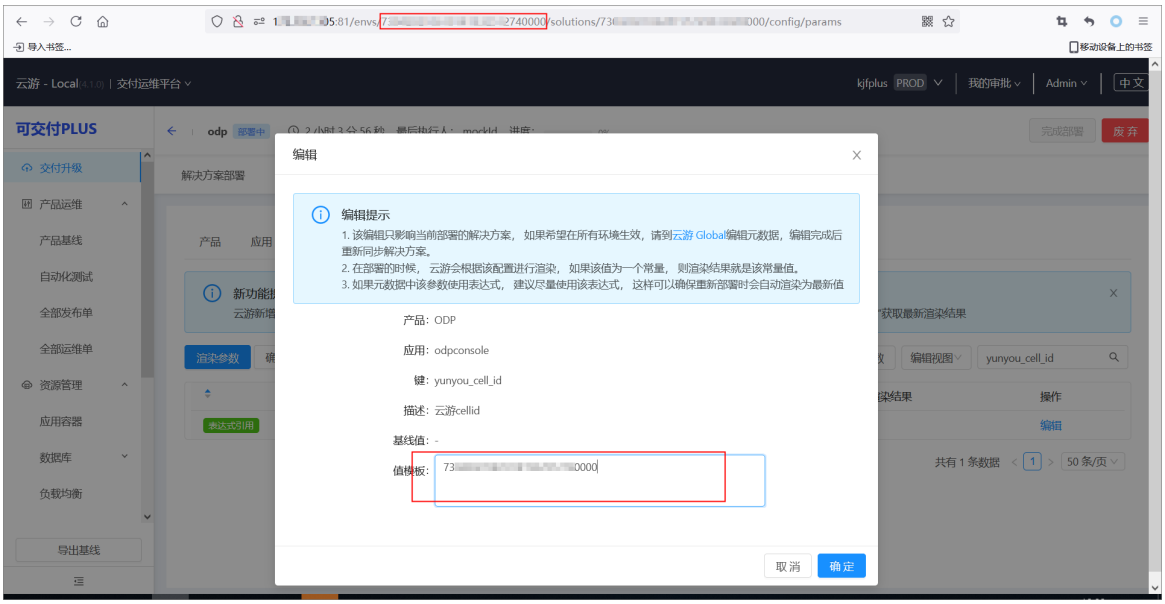
- 应用启动参数

应用名	键	值模板	描述
odpserver	ceresdb_address	127.0.0.1:5000	该迭代不输出 ceresdb，使用左侧默认支模板即可。
odpconsole	ceresdb_address	http://127.0.0.1:5000	该迭代不输出 ceresdb，使用左侧默认支模板即可。
odpconsole	yunyou_cell_id	73042021******0325311950000	填写自身环境云游 Local 的 envid。
odpconsole	yunyou_openapi_endpoint	11.**.**.208:81	填写自身环境云游 Local 的地址。
odpconsole	oss_ak	minio	访问 minio 的 AK。
odpconsole	oss_sk	awesome-minio	访问 minio 的 SK。
odpconsole	oss_endpoint	http://minio.kube-system:9000	访问 minio 的 Endpoint。
odpconsole	oss_bucket	odpstorage	填写 minio 中为 ODP 创建的 Bucket 名称，如果不存在，请访问 minio 手动创建。
odptest	dbptest.env.oss.enable	false	确认环境中 ODP 导入导出是否使用 OSS。
odptest	dbptest.openapi.dbpInstanceId	odp-XV******CNX	填写部署单元对应的实例 ID。

部分参数获取说明：

- yunyou_cell_id

  在云游 Local 地址栏中获取，如下图所示：



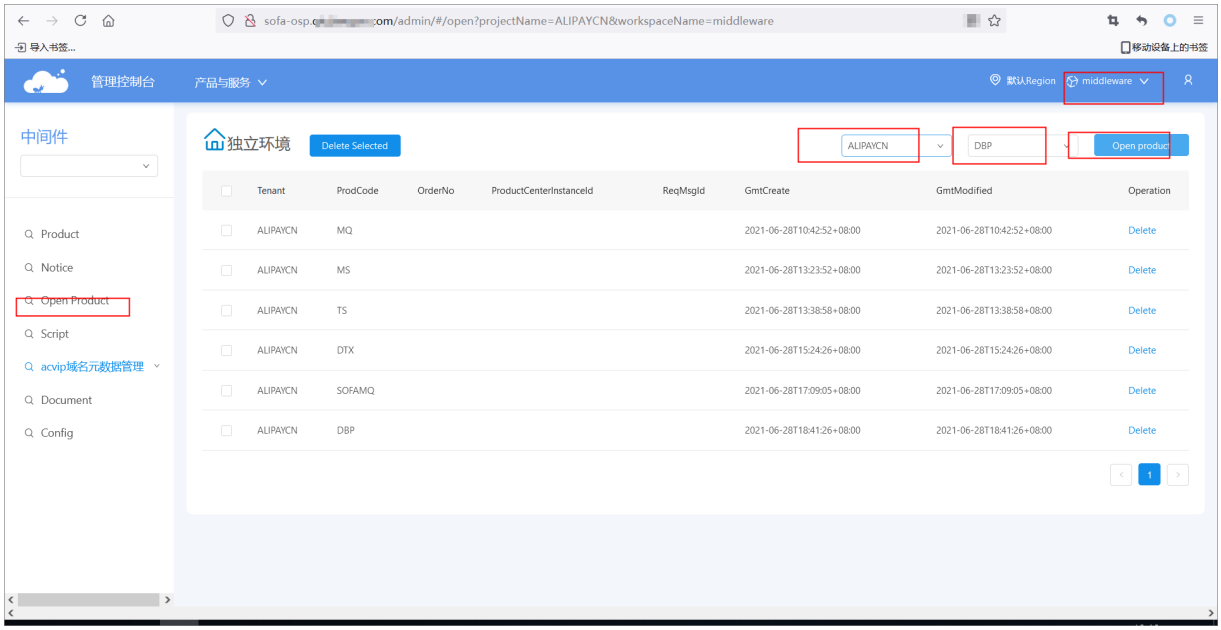
- dbptest.openapi.dbpInstanceId

  在产品的部署单元中获取，如下图所示：

参数修改完成后，直接发布即可。

后置任务

访问 sofa-osp 控制台，确保中间件租户下 middleware 和 core 工作空间的 ODP 已启动。



自动化测试

进行 ODP 的自动化测试前，请确认一下 `dbpconsole_db` 库的 `zdalproxy_sql_batch_job` 表中是否有 `pre_sql_batch_job_id` 这一列。

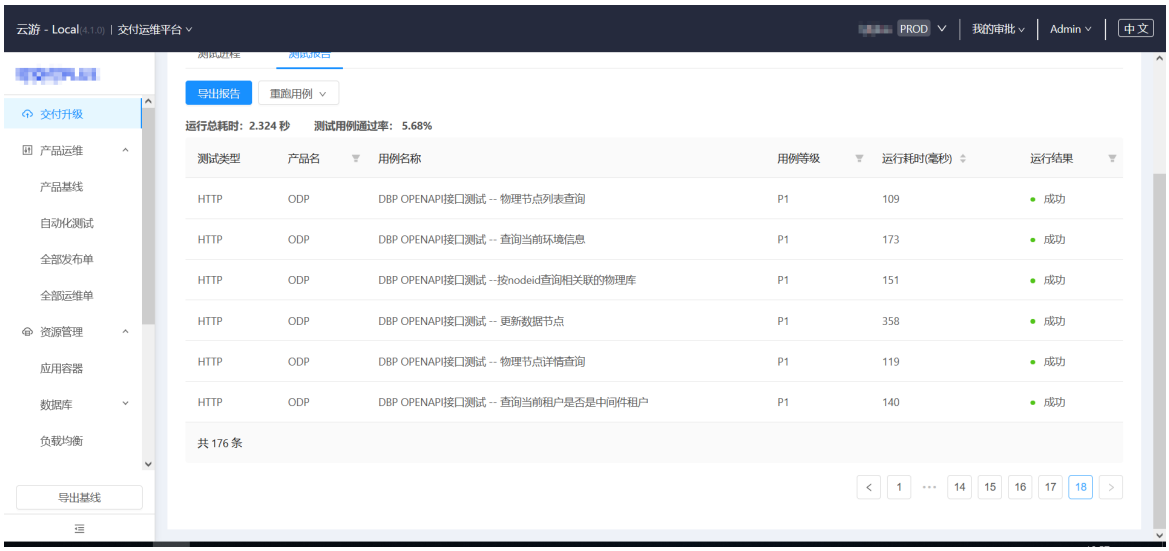
```
Database changed
MySQL [dbpconsole_db]> desc zdalproxy_sql_batch_job;
+----------------------+---------------------+------+-----+---------+-------+
| Field                | Type                | Null | Key | Default | Extra |
+----------------------+---------------------+------+-----+---------+-------+
| id                   | bigint(20) unsigned | NO   | PRI | NULL    |       |
| gmt_create           | datetime            | NO   |     | NULL    |       |
| gmt_modified         | datetime            | NO   |     | NULL    |       |
| cloud_instance_id    | varchar(100)        | NO   |     | NULL    |       |
| zdalproxy_instance_id | varchar(64)        | NO   |     | NULL    |       |
| schema_name          | varchar(128)        | NO   |     | NULL    |       |
| name                 | varchar(128)        | YES  |     | NULL    |       |
| description          | varchar(512)        | YES  |     | NULL    |       |
| type                 | int(11)             | NO   |     | NULL    |       |
| status               | int(11)             | NO   |     | NULL    |       |
| job_mode             | int(11)             | NO   |     | 0       |       |
| user_name            | varchar(128)        | NO   |     | NULL    |       |
| content              | text                | NO   |     | NULL    |       |
| error_msg            | varchar(512)        | YES  |     | NULL    |       |
| total_executions     | bigint(20)          | YES  |     | NULL    |       |
| finished_executions  | bigint(20)          | YES  |     | NULL    |       |
| run_instance_id      | varchar(20)         | YES  |     | NULL    |       |
| run_ip               | varchar(20)         | YES  |     | NULL    |       |
| last_start_time      | datetime            | YES  |     | NULL    |       |
| last_end_time        | datetime            | YES  |     | NULL    |       |
| pre_sql_batch_job_id | bigint(20) unsigned | YES  |     | NULL    |       |
+----------------------+---------------------+------+-----+---------+-------+
21 rows in set (0.05 sec)
```

如果没有，请执行如下 SQL 语句添加：

```
ALTER TABLE `zdalproxy_sql_batch_job` ADD COLUMN `pre_sql_batch_job_id` BIGINT(20) UNSIGNED
default NULL COMMENT '前置任务sqlBatchJobId';
```

1. 进行自动化测试。

因单元的信息未注入 Pod 实例，所以测试只成功了一页。



2. 修改参数。
   - 公共参数

产品名称	键	值模板	描述
ODP	ldc_enabled	false	是否为 IDC 环境。

   - 启动参数

     第一次发布时，dbptest.openapi.dbpInstanceId 值是 None-Deployed-DU_20190******335042，测试完成后，odptest 值会变成 odp-pbds*****dNT，所以需要修改参数重新进行测试。

应用名	键	值模板
odptest	dbptest.openapi.dbpInstanceId	odp-pbds*****dNT

3. 重新进行自动化测试。

   测试已正常。