

百 G 级 Oracle 服务器 迁移最佳实践



场景描述

本文介绍了如何通过 CMS-SMS 主机迁移服务，将部署有单机版百 G 级别的 Oracle 服务器迁移至天翼云 ECS，为迁移提供最佳实践指南。主机迁移服务（Server Migration Service, SMS）是一种 P2V/V2V 迁移服务，该服务能够协助用户将 X86 物理服务器或者私有云、公有云平台上的虚拟机迁移到天翼云弹性云主机上，从而轻松将服务器上的文件、应用、数据库等迁移至天翼云。本实践指南以在天翼云上部署的单机版百 G 级的 Oracle 云主机为例，展示如何在不中断业务的情况下，使用 CMS-SMS 工具快速将 Oracle 数据库热迁移至天翼云。

解决问题：

1、在线热迁移，快速迁移上云。

产品列表

• 弹性云主机 ECS

2、简化上云流程，摒弃繁杂的人工操作过程，自动完成数据迁移。

3、支持稽核修复，保障业务数据“0”丢失。

- 云迁移服务器迁移服务 CMS-SMS
- 虚拟私有云 VPC
- 弹性 IP EIP
- 云硬盘 EVS

文档修订记录

版本编号	时间	作者	修订说明
V1.0	2024年7月17日	林松	创建文档
V2.0	2024年7月24日	刘锦玮	修改文档
V2.1	2024年8月5日	黄锦堃	批注与修订文档

百 G 级 Oracle 服务器迁移最佳实践

最佳实践概述

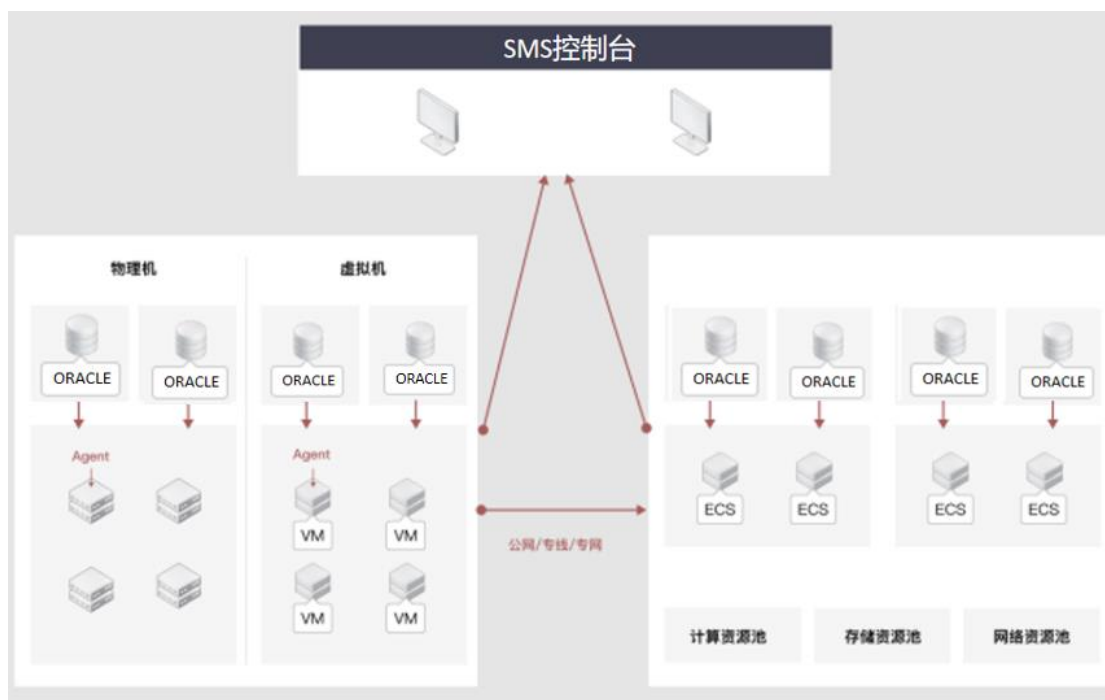
概述

通过服务器迁移服务（CMS-SMS, Server Migration Service）将部署有百 G 级别的单机 Oracle 服务器迁移至天翼云 ECS。CMS-SMS 通过特定的迁移客户端将迁移源主机的所有数据以文件形式传输至目标机，结合“全量+增量”的传输模式，帮助客户在本地业务不停的情况下快速热迁移至天翼云。同时，为了确保数据传输的完整性和准确性，迁移工具还提供数据稽核和数据修复服务，实现 Oracle 数据库快速、安全、高效迁移至天翼云。上云后的数据库系统可以充分发挥云上资源弹性可伸缩的优势，按需提供服务，节约资源使用的成本。

应用场景

1. 适用于中小型企业客户将业务系统数据上云，简化上云步骤；
2. 其他云数据库数据迁移至天翼云，整合云厂商资源。

技术架构图



方案优势

- 兼容性强，支持国内外主流的操作系统。
- 基于自研迁移工具 CMS-SMS，能够完成在不同场景下的迁移任务。
- 灵活易操作，提供用户自服务页面进行灵活简单操作，客户仅需进行简单的配置，即可调整迁移模式、选择迁移对象等功能。
- 智能监控、高可靠性，在操作系统以及数据库迁移的过程中，对任务进行实时监控，包括展示任务总进度、阶段进度、预计完成时间等信息。针对任务过程中出现的问题进行告警处理。
- 日志与审计功能，CMS-SMS 提供完善的日志与审计功能，记录每一次迁移任务的操作记录、数据传输情况，帮助管理员对迁移过程进行全面监控和审计，确保数据迁移的可靠性和安全性。
- 数据加密，在传输过程中支持对数据进行加密，保障数据安全。

名词解释

- 弹性云主机:(Elastic Cloud Server, 简称 ECS)是基于天翼云成熟稳定的虚拟化技术打造的一种可随时获取、弹性可扩展的计算服务。云主机由 CPU、内存、镜像、云硬盘组成,同时结合 VPC、安全组、数据多副本保存等能力,打造一个既高效又可靠安全的计算环境,确保服务持久稳定运行。
- 云硬盘:(Elastic Volume Service, 简称 EVS)是天翼云一种可弹性扩展的块存储设备,可以为弹性云主机和弹性裸金属服务器提供高性能、高可靠的块存储服务。天翼云硬盘规格丰富,满足不同场景的业务需求,适用于文件系统、数据库、开发测试等场景。用户可以在线操作及管理云硬盘,并可以像使用传统服务器硬盘一样,对挂载到云主机的云硬盘做格式化、创建文件系统等操作。
- 弹性 IP:(Elastic IP, 简称 EIP)是可以独立申请的公网 IP 地址,包括公网 IP 地址与公网出口带宽服务。可以与云主机、物理机、负载均衡、NAT 网关等云产品动态绑定和解绑,实现云资源的互联网访问。拥有多种灵活的计费方式,并且通过 IPv6 带宽提供 IPv6 访问公网能力,可以满足各种业务场景的需要。
- 服务器迁移服务(Server Migration Service, 简称 SMS)是天翼云自主研发的一种 P2V/V2V 迁移平台,可将物理服务器、私有云、公有云平台上的单台或多台源主机迁移到天翼云,从而帮助客户解决上云成本高、操作复杂、迁移周期长、业务中断等迁移难题。
- 虚拟私有云:(Virtual Private Cloud, 简称 VPC)是基于天翼云创建的自定义私有网络,为用户提供一个逻辑隔离的区域,构建一个安全可靠、可配置和管理的虚拟网络环境。专有网络由逻辑网络设备(如虚拟路由器,虚拟交换机)组成,可以通过专线/VPN 等连接方式与传统数据中心组成一个按需定制的网络环境,实现应用的平滑迁移上云。
- 源机:源机是指在进行服务器迁移或数据迁移过程中,作为迁移起始点的服务器或设备。它包含了需要迁移的数据、应用、配置等信息。在迁移过程中,源机的信息会被复制到目标机,以实现数据的迁移或服务器

的替换。

- **目标机：**目标机是指在进行服务器迁移或数据迁移过程中，作为迁移终点的服务器或设备。它是迁移后数据、应用、配置等信息的新的承载者。在迁移完成后，目标机将替代源机，继续提供服务或运行应用。目标机通常具备与源机相同或更好的性能和配置，以确保迁移后的系统能够正常运行并满足业务需求。

前置条件

天翼云账号准备

执行本文操作之前，请完成以下准备工作：

- 注册天翼云账号，并完成实名认证。您可以登录天翼云控制中心，并前往实名认证页面（<https://www.ctyun.cn/console/user/realname/result>）查看是否完成实名认证。
- 天翼云账户余额需要大于 100 元。您可以登录天翼云管理中心，并前往账户总览页面（<https://www.ctyun.cn/console/expense/home>）查看账户余额。

天翼云资源准备

- 开通弹性云主机 ECS、云硬盘 EVS、虚拟网络 VPC、弹性 IP EIP、云迁移 CMS、服务器迁移服务 CMS-SMS。
- 源端数据做好备份工作，或采用快照进行备份，或采用拷贝进行备份，本文不对此操作进行详细描述。
- 为了便于演示，本文中涉及资源开通时，均默认选择按量付费模式。

开通以下产品或服务：

产品或服务	本文示例	备注
虚拟网络 VPC	开通服务：免费	
云硬盘 EVS	开通服务：按需计费	普通 IO，40G；普通 IO，200G

云迁移 CMS	开通服务：免费	
服务器迁移服务（CMS-SMS）	开通服务：免费	
弹性云主机 ECS	开通服务：按需计费	规格：8 核 16 G，用于单机数据库、目的端云主机
弹性 IP EIP	开通服务：按需计费	宽带大小：10Mbps

注意： 本文仅作为实践演示，具体环境以用户实际需求为准。

网络准备

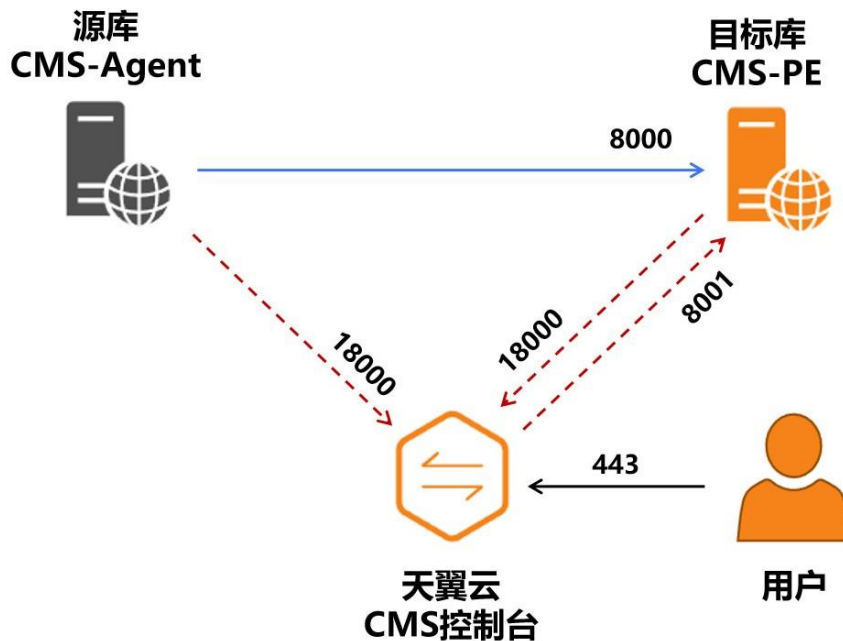
●源库与目标库需通过开放端口或防火墙白名单实现以下网络环境：

1. 源库（此处指需要迁移的源机）

- 1) 出方向能够访问目标库
- 2) 出方向能够访问 CMS 控制台

2. 目标库（此处指迁移后的目标机）

- 1) 入方向 8000 端口放行源库
- 2) 入方向 8001 端口放行 CMS 控制台
- 3) 出方向能够访问 CMS 控制台



1. 源端 Oracle 服务器搭建

1.1 开通云主机

按照天翼云的操作步骤，在规划的网络环境下，购买 1 台云主机，用于搭建单机版 Oracle。规格为：8C16G，Ubuntu，系统盘 40G，数据盘为 200G。

1.2 搭建 Oracle 数据库

本次最佳实践通过采用 docker 容器自动化安装的方式部署 oracle 数据库。

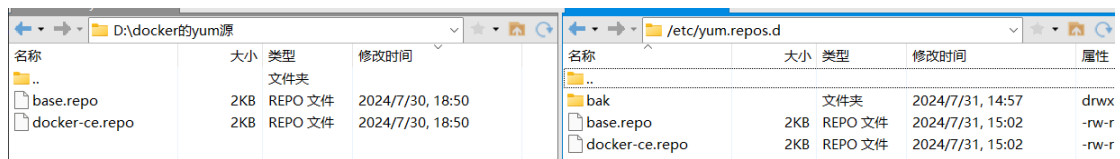
（注：本文提供的 Oracle 相关安装命令和配置过程仅供参考，不作为用户指导或操作 oracle 部署手册。用户在实际操作中应根据具体环境和需求，结合官方文档和技术支持进行配置和安装。本文档不对因使用本方法所导致的任何数据丢失、系统故障或其他潜在问题承担责任。请在执行操作之前备份您的数据，并在需要时寻求专业技术支持。）

步骤 1 关闭防火墙

```
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

步骤 2 导入 docker 的 yum 源，建立元数据缓存

把下载的 base.repo 和 docker-ce.repo 导入云主机中



建立元数据缓存

```
[root@localhost yum.repos.d]# ll
总用量 8
drwxr-xr-x. 2 root root 220 7月 31 14:57 bak
-rw-r--r--. 1 root root 252 7月 31 15:02 base.repo
-rw-r--r--. 1 root root 2081 7月 31 15:02 docker-ce.repo
[root@localhost yum.repos.d]# yum makecache fast
正在设置缓存: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.cloud.tencent.com
 * centosplus: mirrors.cloud.tencent.com
 * epel: mirrors.cloud.tencent.com
 * extras: mirrors.cloud.tencent.com
 * updates: mirrors.cloud.tencent.com
base                               | 3.6 kB | 00:00:00
centosplus                          | 2.9 kB | 00:00:00
docker-ce-stable                    | 3.5 kB | 00:00:00
epel                                 | 4.3 kB | 00:00:00
extras                              | 2.9 kB | 00:00:00
updates                             | 2.9 kB | 00:00:00
(1/10): base//x86_64/group_gz      | 152 kB | 00:00:00
(2/10): docker-ce-stable//x86_64/updateinfo | 55 B | 00:00:00
(3/10): epel//x86_64/group         | 399 kB | 00:00:00
(4/10): centosplus//x86_64/primary_db | 8.3 MB | 00:00:00
(5/10): extras//x86_64/primary_db  | 253 kB | 00:00:00
(6/10): docker-ce-stable//x86_64/primary_db | 152 kB | 00:00:00
(7/10): epel//x86_64/primary_db   | 8.7 MB | 00:00:00
(8/10): updates//x86_64/primary_db | 27 MB | 00:00:00
(9/10): epel//x86_64/updateinfo    | 1.0 MB | 00:00:02
(10/10): base//x86_64/primary_db   | 6.1 MB | 00:00:15
元数据缓存已建立
```


步骤 3 安装并启动 docker-ce

```
[root@localhost yum.repos.d]# yum install -y docker-ce
已加载插件: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirrors.cloud.tencent.com
* centosplus: mirrors.cloud.tencent.com
* epel: mirrors.cloud.tencent.com
* extras: mirrors.cloud.tencent.com
* updates: mirrors.cloud.tencent.com
正在解决依赖关系
--> 正在检查事务
--> 软件包 docker-ce.x86_64.3.26.1.4-1.el7 将被 安装
--> 正在处理依赖关系 container-selinux >= 2:2.74, 它被软件包 3:docker-ce-26.1.4-1.el7.x86_64 需要
--> 正在处理依赖关系 containerd.io >= 1.6.24, 它被软件包 3:docker-ce-26.1.4-1.el7.x86_64 需要
--> 正在处理依赖关系 docker-ce-cli, 它被软件包 3:docker-ce-26.1.4-1.el7.x86_64 需要
--> 正在处理依赖关系 docker-ce-rootless-extras, 它被软件包 3:docker-ce-26.1.4-1.el7.x86_64 需要
--> 正在检查事务
--> 软件包 container-selinux.noarch.2.2.119.2-1.911c772.el7_8 将被 安装
--> 软件包 containerd.io.x86_64.0.1.6.33-3.1.el7 将被 安装
--> 软件包 docker-ce-cli.x86_64.1.26.1.4-1.el7 将被 安装
--> 正在处理依赖关系 docker-buildx-plugin, 它被软件包 1:docker-ce-cli-26.1.4-1.el7.x86_64 需要
--> 正在处理依赖关系 docker-compose-plugin, 它被软件包 1:docker-ce-cli-26.1.4-1.el7.x86_64 需要
--> 软件包 docker-ce-rootless-extras.x86_64.0.26.1.4-1.el7 将被 安装
--> 正在处理依赖关系 fuse-overlayfs >= 0.7, 它被软件包 docker-ce-rootless-extras-26.1.4-1.el7.x86_64 需要
--> 正在处理依赖关系 slirp4netns >= 0.4, 它被软件包 docker-ce-rootless-extras-26.1.4-1.el7.x86_64 需要
--> 正在检查事务
--> 软件包 docker-buildx-plugin.x86_64.0.0.14.1-1.el7 将被 安装
--> 软件包 docker-compose-plugin.x86_64.0.2.27.1-1.el7 将被 安装
--> 软件包 fuse-overlayfs.x86_64.0.7.2-6.el7_8 将被 安装
--> 正在处理依赖关系 libfuse3.so.3(FUSE_3.2)(64bit), 它被软件包 fuse-overlayfs-0.7.2-6.el7_8.x86_64 需要
--> 正在处理依赖关系 libfuse3.so.3(FUSE_3.0)(64bit), 它被软件包 fuse-overlayfs-0.7.2-6.el7_8.x86_64 需要
--> 正在处理依赖关系 libfuse3.so.3()(64bit), 它被软件包 fuse-overlayfs-0.7.2-6.el7_8.x86_64 需要
--> 软件包 slirp4netns.x86_64.0.0.4.3-4.el7_8 将被 安装
--> 正在检查事务
--> 软件包 fuse3-libs.x86_64.0.3.6.1-4.el7 将被 安装
--> 解决依赖关系完成

依赖关系解决

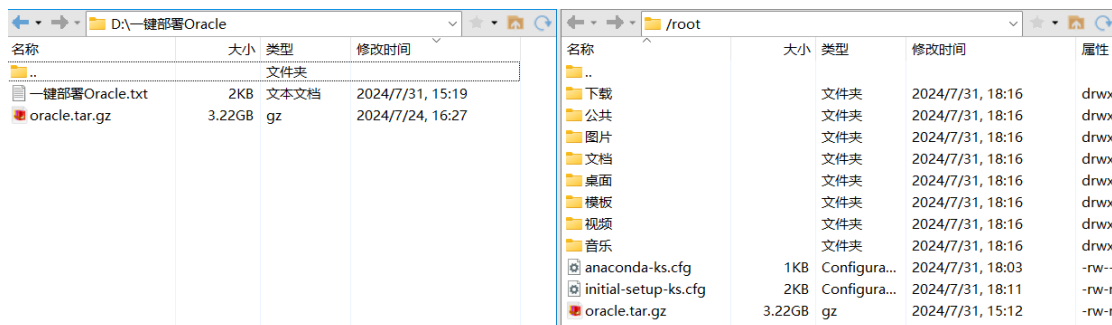
已安装:
docker-ce.x86_64 3:26.1.4-1.el7

作为依赖被安装:
container-selinux.noarch 2:2.119.2-1.911c772.el7_8      containerd.io.x86_64 0:1.6.33-3.1.el7      docker-buildx-plugin.x86_64 0:0.14.1-1.el7      docker-ce-cli.x86_64 1:26.1.4-1.el7
docker-ce-rootless-extras.x86_64 0:26.1.4-1.el7      docker-compose-plugin.x86_64 0:2.27.1-1.el7      fuse-overlayfs.x86_64 0:0.7.2-6.el7_8      fuse3-libs.x86_64 0:3.6.1-4.el7
slirp4netns.x86_64 0:0.4.3-4.el7_8

完毕!
[root@localhost yum.repos.d]# systemctl start docker
[root@localhost yum.repos.d]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since 2024-07-31 15:10:03 CST; 8s ago
     Docs: https://docs.docker.com
    Main PID: 57107 (dockerd)
      Tasks: 10
     Memory: 33.2M
     CGroup: /system.slice/docker.service
             └─57107 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

步骤 4 一键安装部署 Oracle

将 oracle.tar.gz 安装包导入云主机中



```
[root@localhost ~]# ll
总用量 3374544
-rw-----. 1 root root      1523 7月  31 2024 anaconda-ks.cfg
-rw-r--r--. 1 root root      1571 7月  31 2024 initial-setup-ks.cfg
-rw-r--r--. 1 root root 3455522816 7月  31 15:12 oracle.tar.gz
drwxr-xr-x. 2 root root        6 7月  31 2024 公共
drwxr-xr-x. 2 root root        6 7月  31 2024 模板
drwxr-xr-x. 2 root root        6 7月  31 2024 视频
drwxr-xr-x. 2 root root        6 7月  31 2024 图片
drwxr-xr-x. 2 root root        6 7月  31 2024 文档
drwxr-xr-x. 2 root root        6 7月  31 2024 下载
drwxr-xr-x. 2 root root        6 7月  31 2024 音乐
drwxr-xr-x. 2 root root        6 7月  31 2024 桌面
```

将 oracle.tar.gz 文件中的 Docker 镜像导入到 Docker 本地镜像库中

```
[root@localhost ~]# docker load -i oracle.tar.gz
40c24f62a02f: Loading layer [=====>] 233.7MB/233.7MB
49c80855196a: Loading layer [=====>] 465.1MB/465.1MB
a2c532d8cc36: Loading layer [=====>] 34.3kB/34.3kB
99d7f2451a1a: Loading layer [=====>] 37.38kB/37.38kB
cf2aa31a6ca9: Loading layer [=====>] 2.757GB/2.757GB
Loaded image: 10.11.0.199:18080/database/oracle:12.2
[root@localhost ~]# docker run -d --name oracle \
> -p 11521:1521 -p 15500:5500 -p 12484:2484 \
> -e ENABLE_ARCHIVELOG=true \
> -e ENABLE_FORCE_LOGGING=true \
> -e ENABLE_TCPS=true \
> 10.11.0.199:18080/database/oracle:12.2
764ad5b439b96e8d21b15fe423fe3b964427b7bb9f7f3ff051b7b2b9c4a651aa
```

进入到 Oracle 容器执行以下命令：

```
docker exec -it oracle_container_name /bin/bash
```

其中，替换 oracle_container_name 为本次使用的容器名称

```
[root@localhost ~]# docker exec -it oracle /bin/bash
[oracle@764ad5b439b9 /]$
```

使用 SQL*Plus 命令进入 Oracle 数据库：

```
sqlplus username/password@//localhost:1521/your_service_name
```

替换 username 为数据库用户名；

替换 password 为数据库密码；

替换 your_service_name 为数据库名称。

```
[oracle@764ad5b439b9 /]$ sqlplus system/oradoc_db1@//localhost:1521/ORCLPDB1.localdomain
SQL*Plus: Release 12.2.0.1.0 Production on Wed Jul 31 07:41:08 2024
Copyright (c) 1982, 2016, Oracle. All rights reserved.
Last Successful login time: Wed Jul 31 2024 07:19:30 +00:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
SQL>
```

2. CMS-SMS 迁移环境准备

2.1 开通目标端云主机

步骤 1 在控制中心切换到目标端所在资源池，点击弹性云主机



步骤 2 点击创建云主机



步骤 3 按照规划输入基础配置

- 付费方式：按量付费
- 地域：福建-福州 25
- 实例名称：cms-sms
- 主机名称：cms-sms

计费模式 包年/包月 按量付费 ?

* 地域 福建 - 福州25 ?

* 企业项目 default ? 如需创建新的企业项目，您可以[点击去创建](#)

* 虚拟私有云 default_vpc(10.0.0.0/8) C
 该VPC尚未开通IPv6
 如需创建新的VPC，您可[前往控制台创建](#)

* 实例名称 cms-sms ?

* 主机名称 cms-sms ?
 创建多台云主机时，系统自动增加后缀，例如：我的云主机-0001

- 规格：8 核 16G

在使用 CMS-SMS 迁移工具进行迁移时，建议目标机与源机的配置尽量保持一致。为了确保迁移程序的正常运行，源端主机应预留约 1-2GB 的内存用于迁移程序的占用。本文中源端主机的规格为 8 核 16GB 内存，因此目标主机选取相同规格的配置进行开通。

- 镜像类型：公共镜像

- 镜像：CMS-PE-LINUX-V2-mini (2GB)

源机为 Ubuntu 操作系统，属于 Linux 类型的操作系统，此处目标机镜像选择“CMS-PE-LINUX-V2-mini (2GB)”即可。

- 存储：系统盘 40G

- 购买数量：1



步骤 4 配置完成，单击下一步：网络配置

- 网卡： default_network(10.0.0.0/24)

- 内网 IP 地址 (IPV4)：自动分配内网 Ipv4 地址

- 安全组：default 安全组
- 弹性 IP：自动分配
- IP 版本： Ipv4
- 宽带：10M



步骤 5 配置完成，单击下一步：高级配置

- 登录方式：密码
- 创建密码：立即创建

弹性云主机

1 基础配置 2 网络配置 3 高级配置

* 登录方式 密码 密钥对
 可使用密码通过控制台登录云主机

* 创建密码

* 用户名:

* 密码

* 确认密码

云主机组:

购买量 1 台 配置费用 **¥ 2.58** /小时 ①

- ✓ 8~30 个字符
- ✓ 必须同时包含三项 (大写字母、小写字母、数字、`[-!@#%&*_+=|00~<>./` 中的特殊符号)
- ✓ 不能以斜线号 (/) 开头
- ✓ 不能包含3个及以上连续字符, 如abc、123
- ✓ Linux镜像不能包含镜像用户名 (root)、用户名的倒序 (toor)、用户名大小写变化 (如RoOt/rOot等)

步骤 6 配置完成，单击下一步：确认配置后，单击立即购买

弹性云主机

1 基础配置 2 网络配置 3 高级配置

配置

配置	基础配置	网络配置	高级配置
云主机名称	cms-sms	地域	福建 - 福州25
企业项目	default	虚拟私有云	default_vpc(10.0.0.0/8)
规格	计算增强型 c6.2xlarge.2 8核 16G	系统盘	通用型SSD 40GB
安全组	default	弹性IP	—
带宽	10Mbps	主网卡	default_subnet(10.0.0.0/24)
计费模式	按量付费	镜像	CTyunOS 3-23.01 64位
IP版本	—	IP版本	IPv4
登录方式	密码	云主机组	—

您还可以创建 50 台云主机。

配置费用 **¥ 2.58** /小时 ①

2.2 迁移源端云主机下载、安装并启动迁移 agent

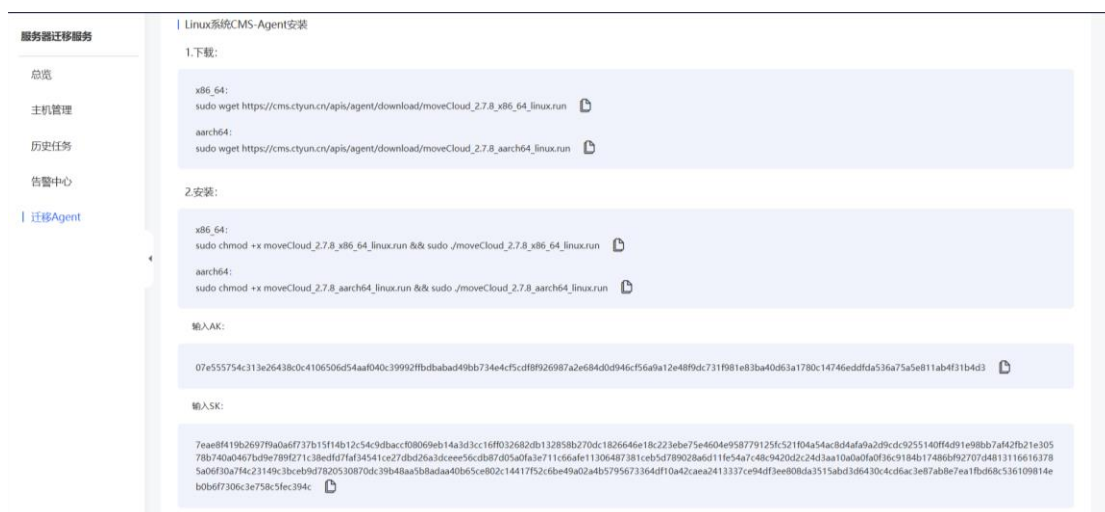
2.2.1 打开“CMS-SMS”控制台，点击“迁移 agent”，浏览迁移前检查项，并认真核对



2.2.2 下载迁移 agent 程序

服务器迁移服务支持 Linux 和 Windows 两种操作系统的同构迁移。根据云主机的操作系统和处理器架构选择对应的迁移 agent 安装程序，复制对应的下载指令并在源机上运行。

本次实践选择 Linux 操作系统 X86 版本的迁移 agent 安装程序。



执行指令，下载 agent 安装程序：

```
sudo wget https://cms.ctyun.cn/apis/agent/download/moveCloud_2.7.8_x86_64_linux.run
```

```
[root@cms-best-platform ~]# sudo wget https://cms.ctyun.cn/apis/agent/download/moveCloud_2.7.8_x86_64_linux.run
--2024-08-02 07:08:36-- https://cms.ctyun.cn/apis/agent/download/moveCloud_2.7.8_x86_64_linux.run
Resolving cms.ctyun.cn (cms.ctyun.cn)... 203.56.24.243
Connecting to cms.ctyun.cn (cms.ctyun.cn)|203.56.24.243|:443... connected.
HTTP request sent, awaiting response... 200
Length: 65737284 (63M) [application/octet-stream]
Saving to: 'moveCloud_2.7.8_x86_64_linux.run'

100%[=====>] 65,737,284  1.15MB/s  in 56s

2024-08-02 07:09:33 (1.13 MB/s) - 'moveCloud_2.7.8_x86_64_linux.run' saved [65737284/65737284]

[root@cms-best-platform ~]#
```

2.2.3 安装迁移 agent

回到天翼云“迁移 agent”界面，复制安装指令：



在迁移源机上执行安装指令：

```
agent: sudo chmod +x moveCloud_2.7.8_x86_64_linux.run && sudo ./moveCloud_2.7.8_x86_64_linux.run
```

```
[root@cms-best-platform ~]# sudo chmod +x moveCloud_2.7.8_x86_64_linux.run && sudo ./moveCloud_2.7.8_x86_64_linux.run
Please enter AK: █
```

2.2.4 安装过程输入 AK 和 SK。

回到天翼云“迁移 agent”界面，可复制 AK 和 SK：

Linux系统CMS-Agent安装

1. 下载:

```
x86_64:
sudo wget https://cms.ctyun.cn/apis/agent/download/moveCloud_2.7.8_x86_64_linux.run
aarch64:
sudo wget https://cms.ctyun.cn/apis/agent/download/moveCloud_2.7.8_aarch64_linux.run
```

2. 安装:

```
x86_64:
sudo chmod +x moveCloud_2.7.8_x86_64_linux.run && sudo ./moveCloud_2.7.8_x86_64_linux.run
aarch64:
sudo chmod +x moveCloud_2.7.8_aarch64_linux.run && sudo ./moveCloud_2.7.8_aarch64_linux.run
```

输入AK:

```
07e555754c313e26438c0c4106506d54aaf040c39992ffbdbabad49bb734e4c5cdf8f926987a2e684d0d946cf56a9a12e48f9dc731f981e83ba40d63a1780c14746eddffa536a75a5e811ab4f31b4d3
```

输入SK:

```
7ee8f419b2697f9a0a6f737b15f14b12c54c9dbaccf08069eb14a3d3c16ff0ee56522eb0705ea274b7966e6ff9d43e8e75e4604e958779125fc521f04a54ac8d4afa9a2d9cd:9255140ff4d91e98bb7af42fb21e30578b740a0467b9d9e789f271c38edfd7faf34541ce27dbd26a3dceee56c8db87d05a0fa3e711c66afe11306487381ceb5d789028a6d11fe54a7c48c9420d2c24d3aa10a0a0fa0f36c9184b17486bf92707d48131166163785a06f30a7f4c23149c3bceb9d7820530870dc39b48aa5b8adaa40b65ce802c14417f52c6be49a02a4b5795673364df10a42c2aa2413337ce94df3ee808da3515abd3d6430c4c6dc3e87ab8e7ea1fbd68c536109814eb0b6f7306c3e758c5fec394c
```

在迁移源机上输入对应的 AK 和 SK

```
[root@cms-best-platform ~]# sudo chmod +x moveCloud_2.7.8_x86_64_linux.run && sudo ./moveCloud_2.7.8_x86_64_linux.run
Please enter AK:07e555754c313e26438c0c4106506d54aaf040c39992ffbdbabad49bb734e4c5cdf8f926987a2e684d0d946cf56a9a12e48f9dc731f981e83ba40d63a1780c14746eddffa536a75a5e811ab4f31b4d3
Please enter SK:7ee8f419b2697f9a0a6f737b15f14b12c54c9dbaccf08069eb14a3d3c16ff0ee56522eb0705ea274b7966e6ff9d43e8e75e4604e958779125fc521f04a54ac8d4afa9a2d9cd:9255140ff4d91e98bb7af42fb21e30578b740a0467b9d9e789f271c38edfd7faf34541ce27dbd26a3dceee56c8db87d05a0fa3e711c66afe11306487381ceb5d789028a6d11fe54a7c48c9420d2c24d3aa10a0a0fa0f36c9184b17486bf92707d48131166163785a06f30a7f4c23149c3bceb9d7820530870dc39b48aa5b8adaa40b65ce802c14417f52c6be49a02a4b5795673364df10a42c2aa2413337ce94df3ee808da3515abd3d6430c4c6dc3e87ab8e7ea1fbd68c536109814eb0b6f7306c3e758c5fec394c
[INFO] Program deployment completed!
[INFO] Usage: sudo movecloud {start [-y|n]|stop|restart [-y|n]|status|uninstall|version|proxy [ip:port|remove]}
[root@cms-best-platform ~]#
```

2.2.5 启动迁移 agent 程序

输入AK:

```
07e555754c313e26438c0c4106506d54aaf040c39992ffbdbabad49bb734e4c5cdf8f926987a2e684d0d946cf56a9a12e48f9dc731f981e83ba40d63a1780c14746eddffa536a75a5e811ab4f31b4d3
```

输入SK:

```
7ee8f419b2697f9a0a6f737b15f14b12c54c9dbaccf08069eb14a3d3c16ff0ee56522eb0705ea274b7966e6ff9d43e8e75e4604e958779125fc521f04a54ac8d4afa9a2d9cd:9255140ff4d91e98bb7af42fb21e30578b740a0467b9d9e789f271c38edfd7faf34541ce27dbd26a3dceee56c8db87d05a0fa3e711c66afe11306487381ceb5d789028a6d11fe54a7c48c9420d2c24d3aa10a0a0fa0f36c9184b17486bf92707d48131166163785a06f30a7f4c23149c3bceb9d7820530870dc39b48aa5b8adaa40b65ce802c14417f52c6be49a02a4b5795673364df10a42c2aa2413337ce94df3ee808da3515abd3d6430c4c6dc3e87ab8e7ea1fbd68c536109814eb0b6f7306c3e758c5fec394c
```

3. 示例:

```
[root@localhost ~]# sudo chmod +x moveCloud_x.x.x_x86_64_linux.run && sudo ./moveCloud_x.x.x_x86_64_linux.run
Please enter AK:9d1a29f54e63877ade9d9e6e2878c94d1e2b05d0b13ae979042a2bcf29ace7e912350d9cd3df483977b88f8af1bfe313
Please enter SK:0b018c1465a29fd53eb9bea6b44110352b20d8cc041c39453175b30de2a3df4af8801e91636743e00867b70632fa4fb6693495016317f8db6670a9b9dbb6857b27e4e171884dbd98a1a6a74b47e1fdf3497134cee4eece80bca34bb11431bcbab7a6b8de607a1a04a06cb498116631137dff2a8d5583f003cf4b27
[INFO] Program deployment completed!
[INFO] Usage: sudo movecloud {start [-y|n]|stop|restart [-y|n]|status|uninstall|version}
```

4. 启动:

```
sudo movecloud start
```

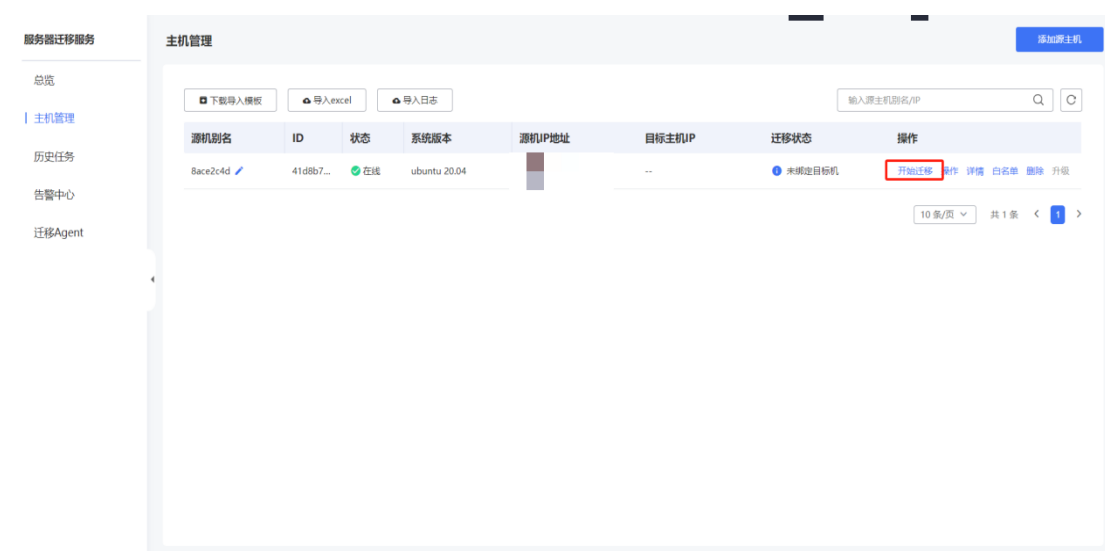
在迁移源机启动迁移 agent 程序：
movecloud start

```
[root@cms-best-platform ~]# sudo movecloud start
Starting moveCloud server...
start movecloud success
```

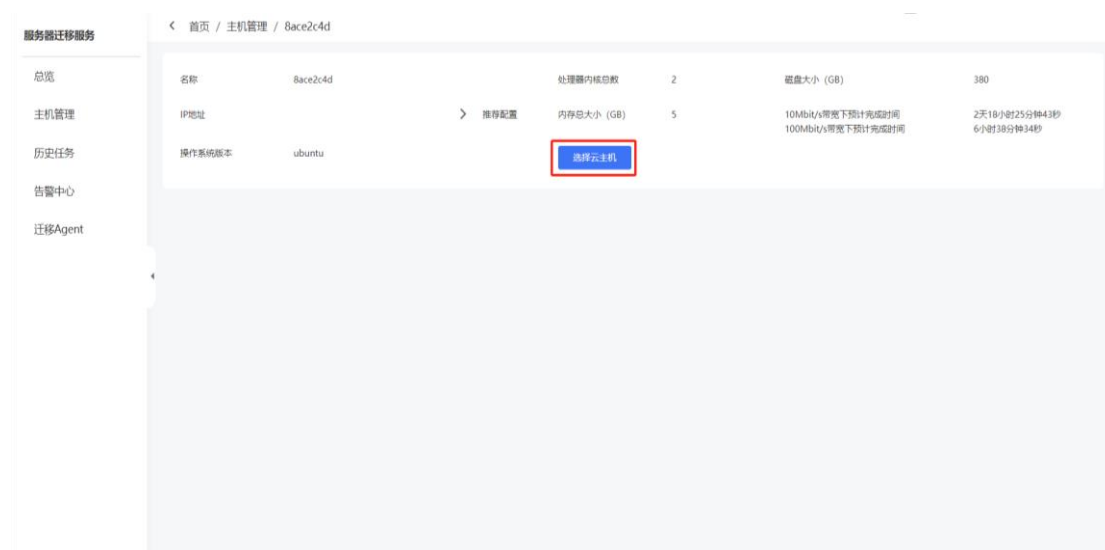
3.使用 CMS-SMS 进行服务器迁移

3.1 源机绑定目标机

步骤 1 点击“主机管理”查看已经安装 agent 程序的源机，并点击“开始迁移”。显示“该源机还未绑定目标机，是否前往绑定”，点击“前往”。

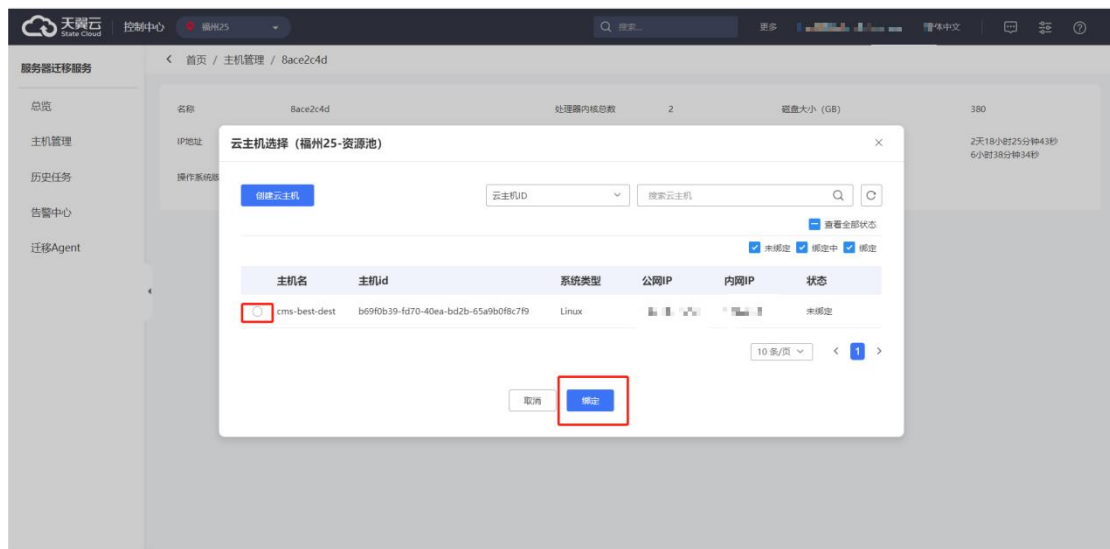


步骤 2 点击“选择云主机”。

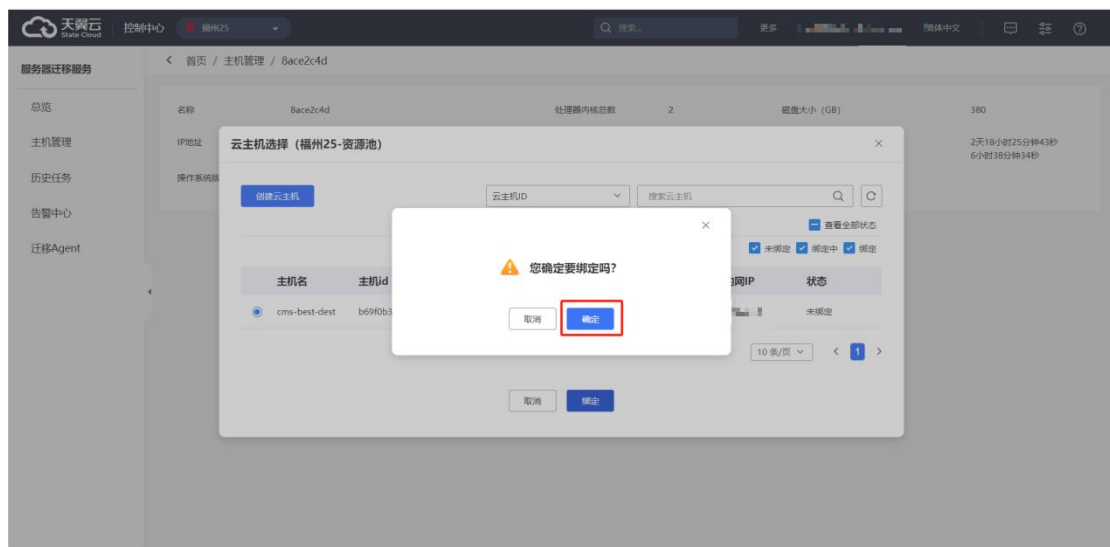


步骤 3 跳转到“云主机选择”界面，左上角选择对应的资源池，查找并勾选创建好的目的端云主机。目的端资源池云主机数量过多，支持通过云主机 ID 或名称等搜索对应的目的端云主机。勾选完成之后需仔细核对目的端云主机的“主机

名”、“主机 ID”等信息。点击“绑定”，随后点击“确定”，即完成目的端云主机的绑定。



本次实践操作的源机为 Ubuntu 操作系统，是基于 Linux 的操作系统，对应目标机选择专用的云迁移镜像“CMS-PE-LINUX-V2-mini (2GB)”。



注意：迁移目标机最低配置不低于 2 核 4G。若目标机内存不足 4G，则会出现提示“目标机推荐内存总大小 4，当前内存总大小 2，不满足”的提示信息。如出现这个此提示，升级迁移目标机配置即可。

注意：推荐目标机与源机的配置尽量保持一致，源端机需预留约 1-2G 内存用于迁移程序占用。如源机配置低于 2 核 4G，目标机配置可以设置为 2 核 4G 及以上。

步骤 4 绑定成功后，进入任务配置界面。

了解是否有增量的需求。如判断在数据迁移的过程中将发生数据变化(增删改查)，

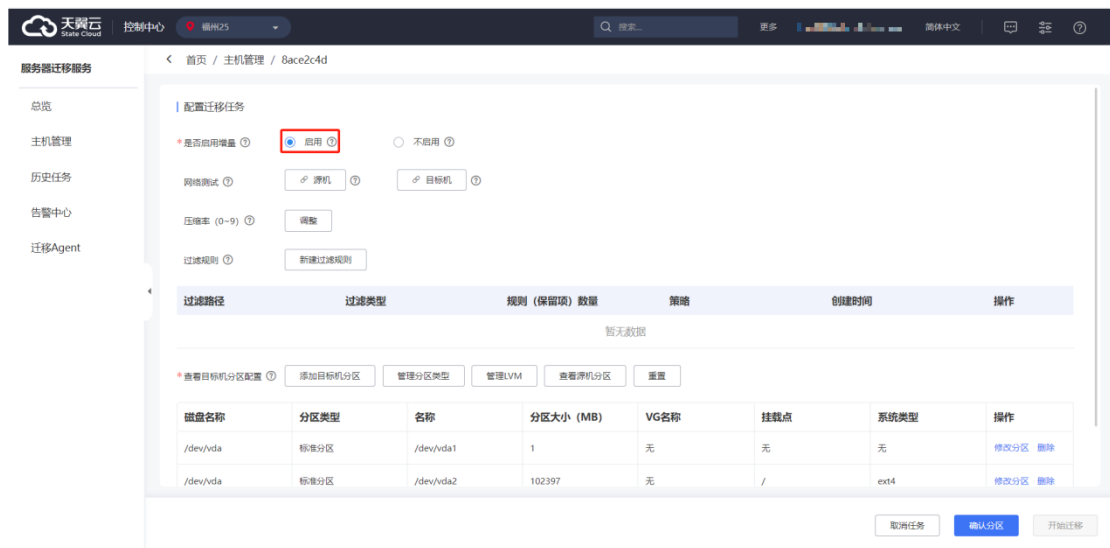
即可开启“启用增量”。反之，则无需开启。

接下来，检查源机以及目标机的网络连通性。点击“源机”和“目标机”两个按钮，如按钮图标变为“√”图标，则源机以及目标机的网络可以连通。

反之，则不成功，需排查源机与目标机 IP 以及端口等是否配置正确。

在数据迁移过程中，若对迁移数据量的规模及迁移效率有明确要求，建议调整“压缩率”设置。具体而言，压缩率数值越趋近于 0，表示数据压缩程度越高，所需传输的数据量相应减少，从而可能提升数据传输速度。若无特定需求，则无需调整此设置。

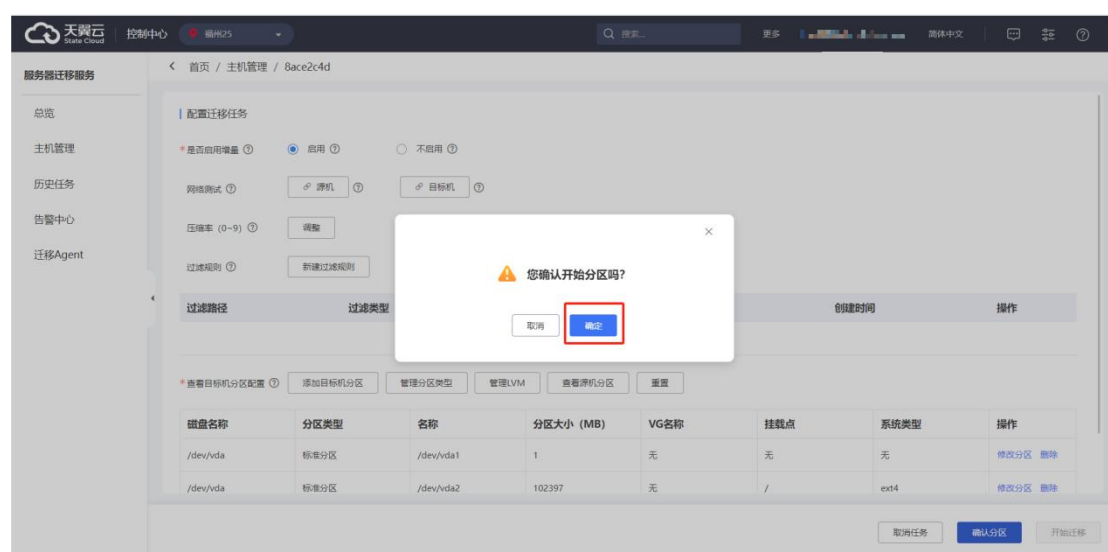
此外，针对特定盘符中不需要迁移的文件，可配置过滤规则以排除这些文件，确保迁移过程的精确性和效率。同样地，若无需进行此类过滤操作，则无需额外设置。



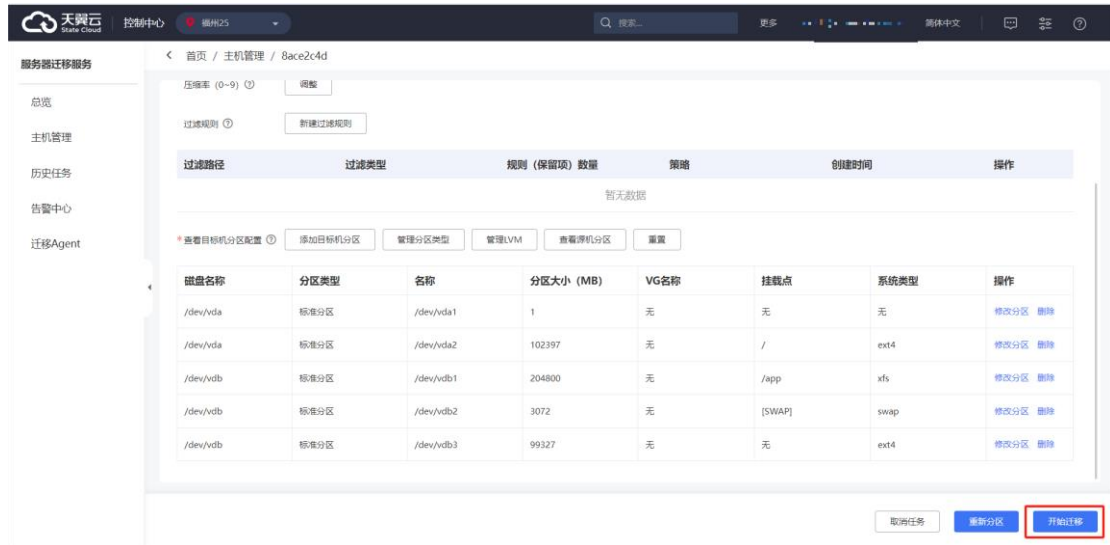
步骤 5 核对源机和目标机分区情况。如源机与目标机的系统盘和数据盘大小完全一致，系统自动分区一般不会出现有问题。如目标机分区与源机分区不一致，则需要手动分区。（目标机的分区只能比源机大），分区完成后，点击“确认分区”。



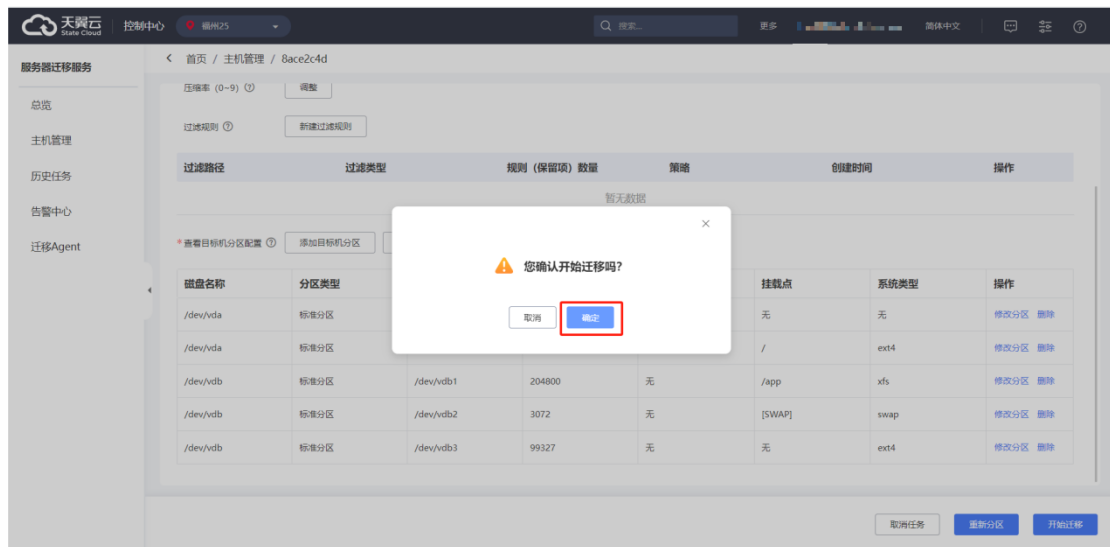
步骤 6 跳出选择框，点击“确认”



步骤 7 待分区结束后，点击“开始迁移”

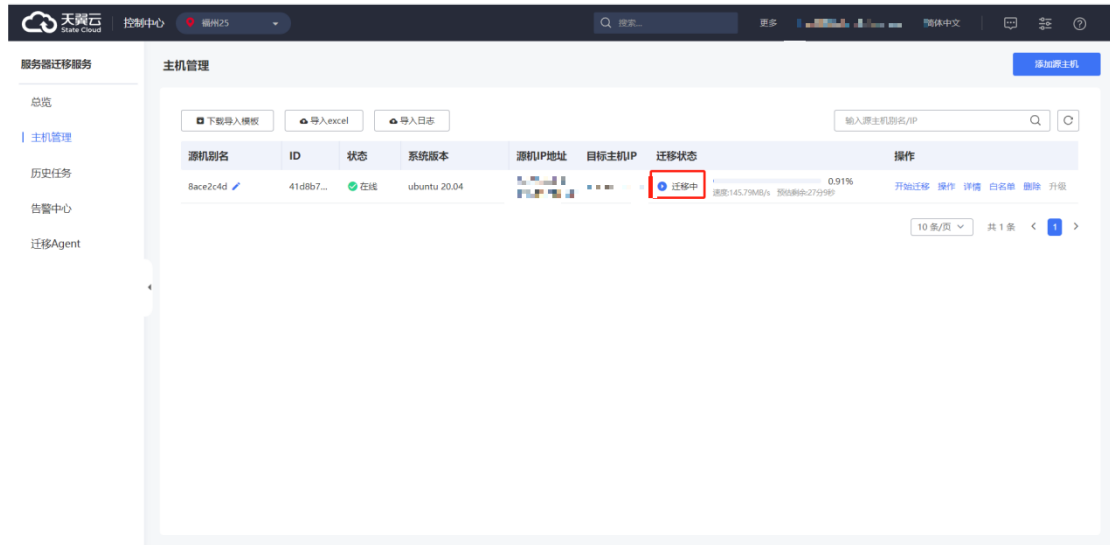


步骤 8 跳出选择框，点击“确认”

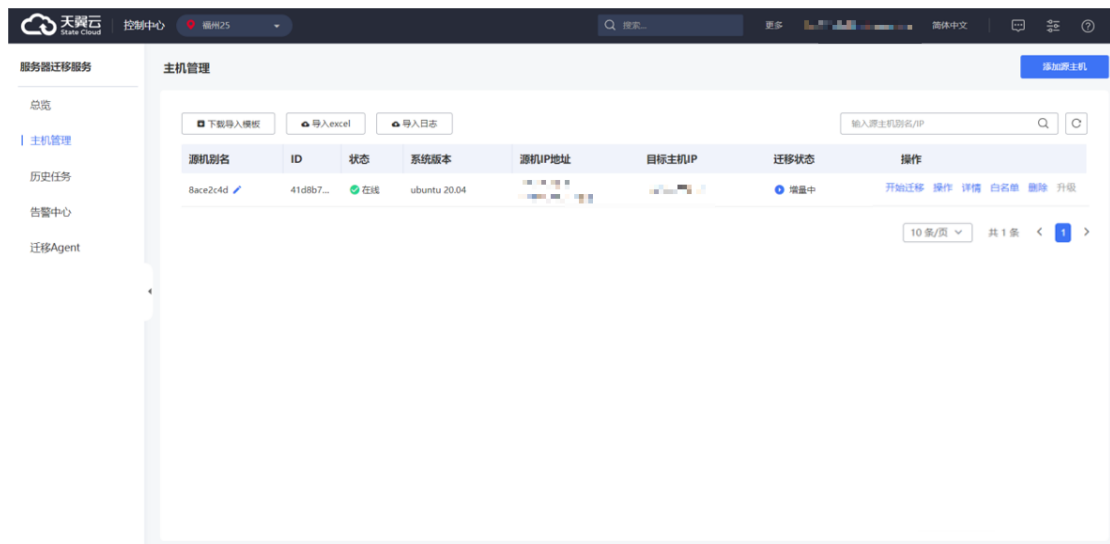


3.2 开始迁移任务

步骤 1 在确认数据迁移操作后，系统将自动跳转回主机管理界面，用户在此界面上不仅能实时观察到源机器的迁移状态已转变为“迁移中”，同时还能获取详尽的迁移详情，包括但不限于迁移代理（agent）的当前状态、实时的数据传输速度，以及基于当前速度预估的迁移剩余时间等关键信息，以便于全面监控和管理迁移进程。



步骤 2 全量迁移阶段完成后，出现提示，进入增量迁移阶段，单击“确认”。进入增量阶段后，每五分钟进行一次增量传输。



步骤 3 模拟迁移过程中，源机数据库在业务运行时的增删改查操作，在增量阶段往源机数据库写入数据（在表 my_table 中插入 1000 条数据），具体效果如下图所示。

```
SQL>
SQL> SELECT COUNT(*) AS record_count
FROM my_table; 2

RECORD_COUNT
-----
          100

SQL> SELECT COUNT(*) AS record_count
FROM my_table; 2

RECORD_COUNT
-----
         1100

SQL> █
```

3.3 oracle 应用割接

步骤 1 准备进行迁移业务割接操作，首先停止源端机数据库服务。

（注意：本停库方式适用于本系统使用的 Oracle 数据库 12C 版本，该停库方式仅做参考。实际环境中因涉及环境及版本复杂，相关操作和命令可能会有所不同，请参考相应版本的官方文档或联系数据库管理员以获取准确的信息和操作步骤。本文档不对因使用本方法所导致的任何数据丢失、系统故障或其他潜在问题承担责任。请在执行操作之前备份您的数据，并在需要时寻求专业技术支持。）

进入源端机 oracle 所在容器

```
docker exec -it oracle_container_name /bin/bash
```

```
[root@localhost ~]# docker exec -it oracle /bin/bash
[oracle@764ad5b439b9 /]$ █
```

使用 SQL*Plus 命令进入 Oracle 数据库：

```
sqlplus username/password@//localhost:1521/your_service_name
```

```
[oracle@764ad5b439b9 /]$ sqlplus system/oradoc_db1@//localhost:1521/ORCLPDB1.localdomain
SQL*Plus: Release 12.2.0.1.0 Production on Wed Jul 31 07:41:08 2024
Copyright (c) 1982, 2016, Oracle. All rights reserved.
Last Successful login time: Wed Jul 31 2024 07:19:30 +00:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
SQL> █
```

使用“SHUTDOWN NORMAL;” 关停数据库，此操作将确保数据库在关闭前完成所

有事务的处理，并将数据落盘。（注意： 执行“SHUTDOWN NORMAL;”后，请耐心等待数据库完成关闭过程。这包括等待所有活动事务完成、回滚未提交的事务、以及将数据文件同步到磁盘等操作。确保数据库完全关闭后，再执行后续步骤。）

```
SQL> SHUTDOWN NORMAL;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

在确认业务系统无任何增量数据写入，并且数据库已完全关闭后，停止数据库容器。执行命令“docker stop oracle”，并设置尽可能长的等待时间，以确保容器内的进程有足够的时间安全退出

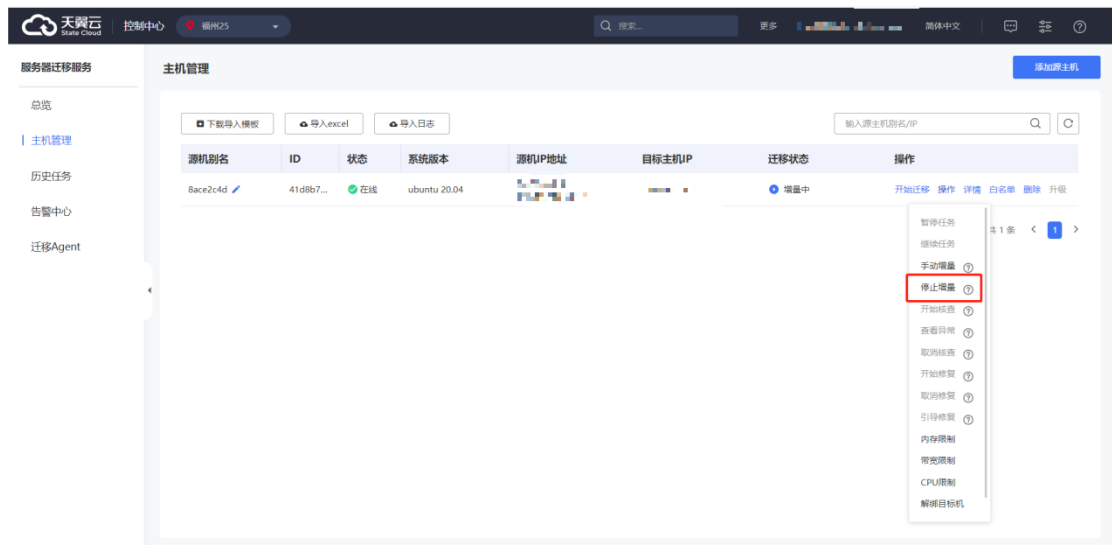
```
root@cms-best-platform:~# docker stop oracle
oracle
root@cms-best-platform:~# docker ps -a | grep oracle
b2236a1e49 - 1b.11.9.199:19880/database/oracle:12.2 22 hours ago Exited (137) 15 seconds ago
root@cms-best-platform:~#
```

在确认数据库容器已完全停止后，关闭 Docker 服务，执行命令“systemctl stop docker.socket”

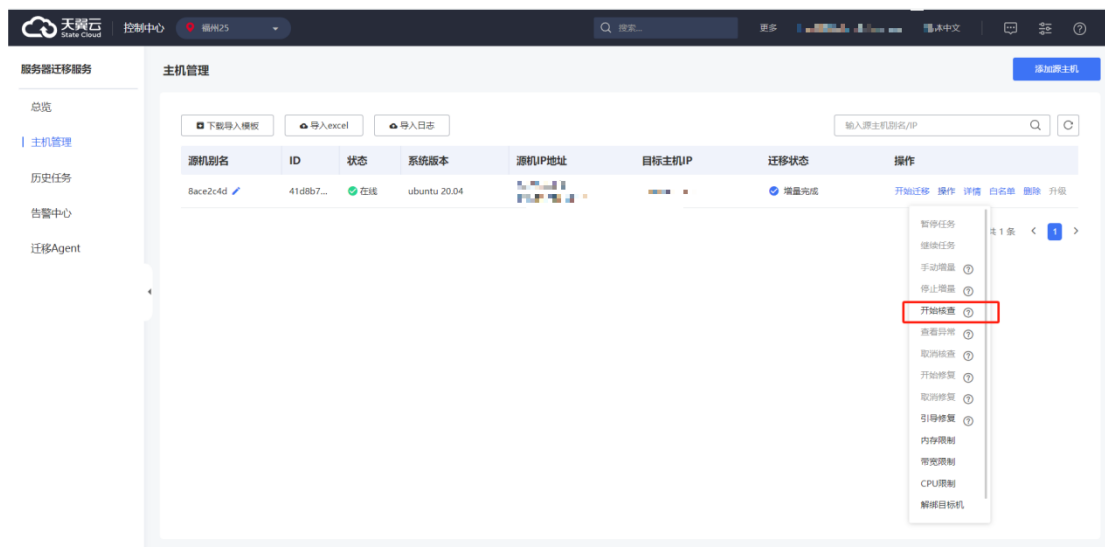
```
root@cms-best-platform:~# systemctl stop docker.socket
root@cms-best-platform:~# systemctl status docker.socket
● docker.socket - Docker Socket for the API
   Loaded: loaded (/lib/systemd/system/docker.socket; enabled; vendor preset: enabled)
   Active: inactive (dead) since Wed 2024-07-17 17:22:52 CST; 5s ago
   Triggers: ● docker.service
   Listen: /run/docker.sock (Stream)

Jul 17 10:58:41 cms-best-platform systemd[1]: Starting Docker Socket for the API.
Jul 17 10:58:41 cms-best-platform systemd[1]: Listening on Docker Socket for the API.
Jul 17 17:22:52 cms-best-platform systemd[1]: docker.socket: Succeeded.
Jul 17 17:22:52 cms-best-platform systemd[1]: Closed Docker Socket for the API.
root@cms-best-platform:~#
```

步骤 2 点击“停止增量”。跳出选项框，提示需进行源端应用停止。确认步骤一源端应用成功停止后，单击“确认”



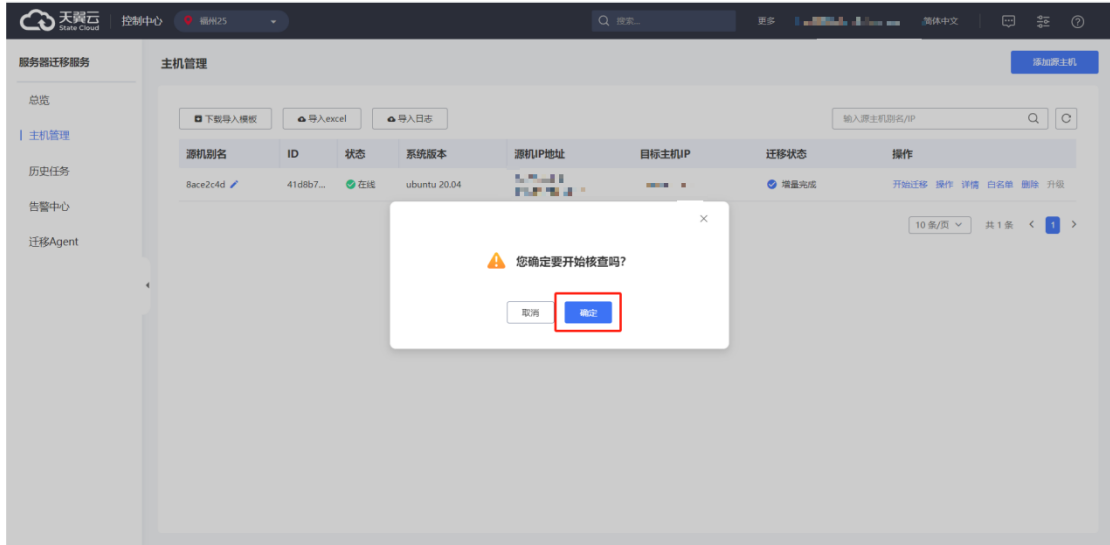
步骤 3 等待增量完成，单击“开始核查”



步骤 4 全选标志位，单击“确定”。

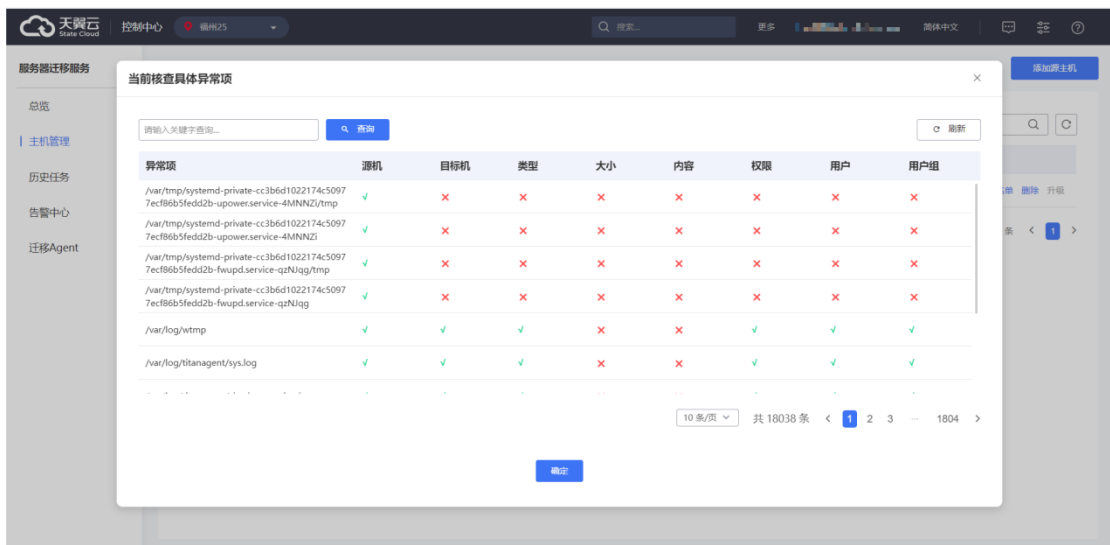
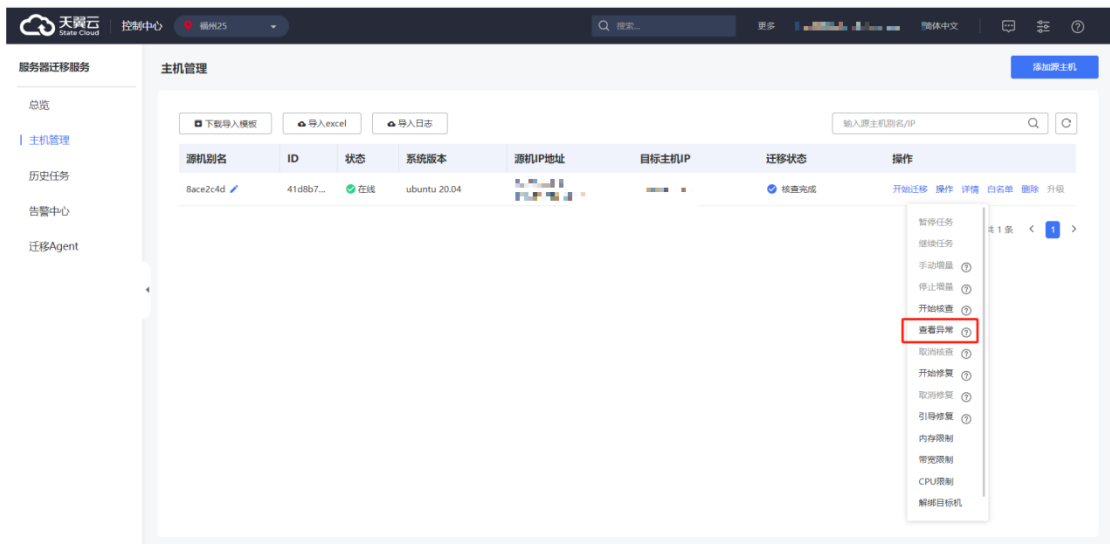


单击“确定”。

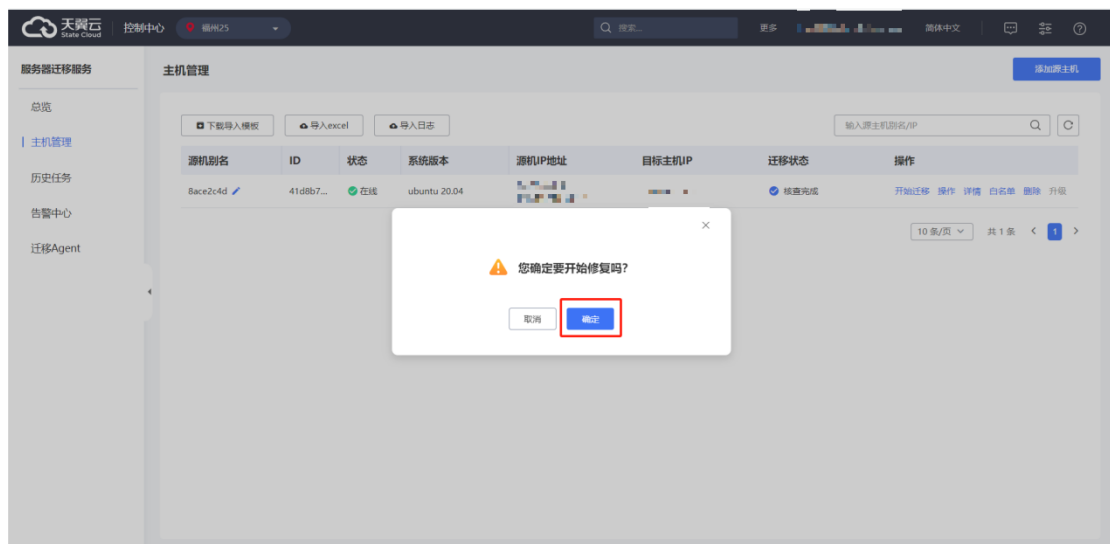
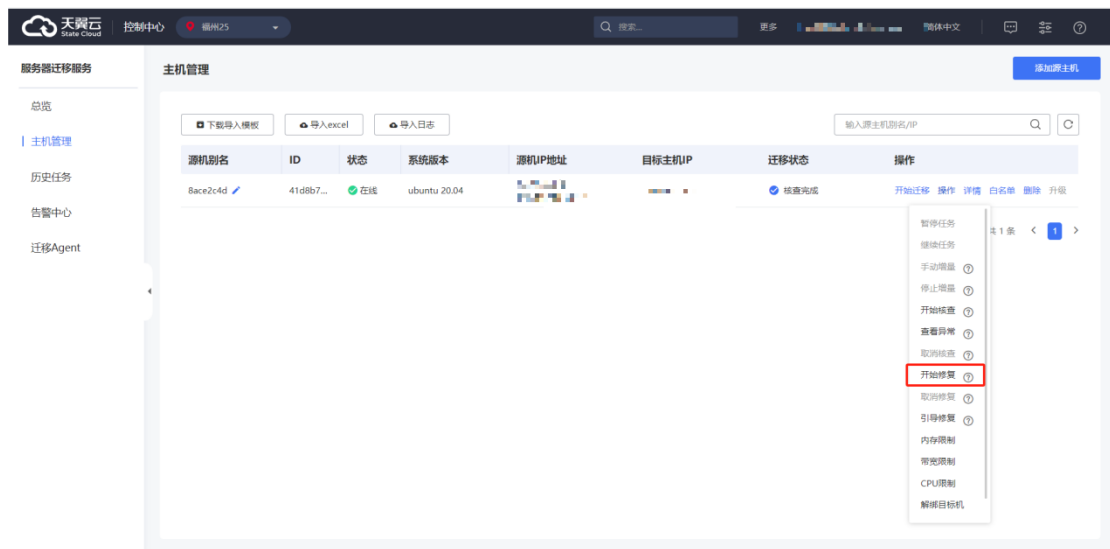


步骤 5 进入核查修复阶段，等待核查完成，单击“查看异常”。

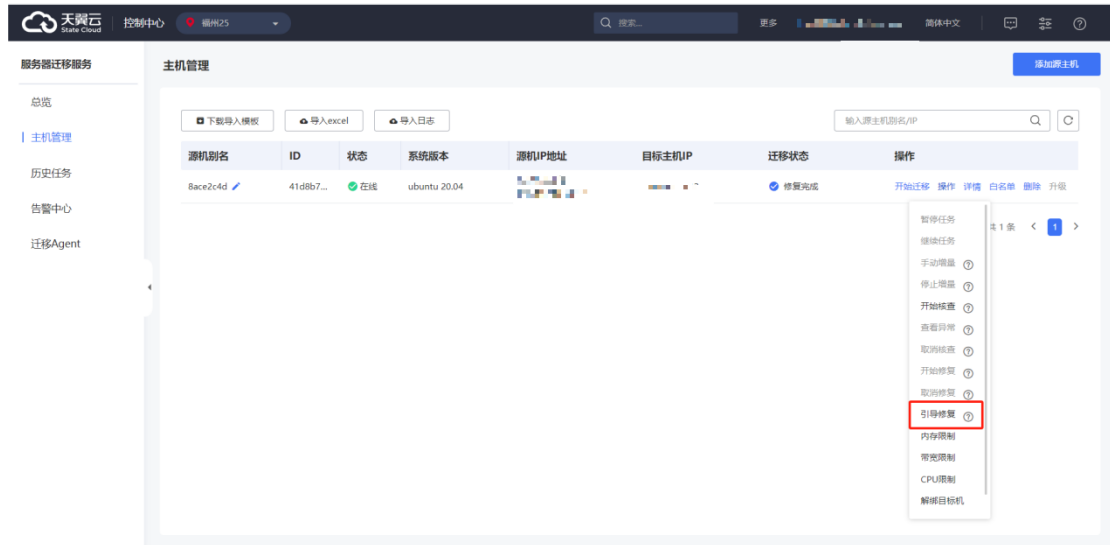
查看当前核查具体异常项，确认完成后，单击“确定”。



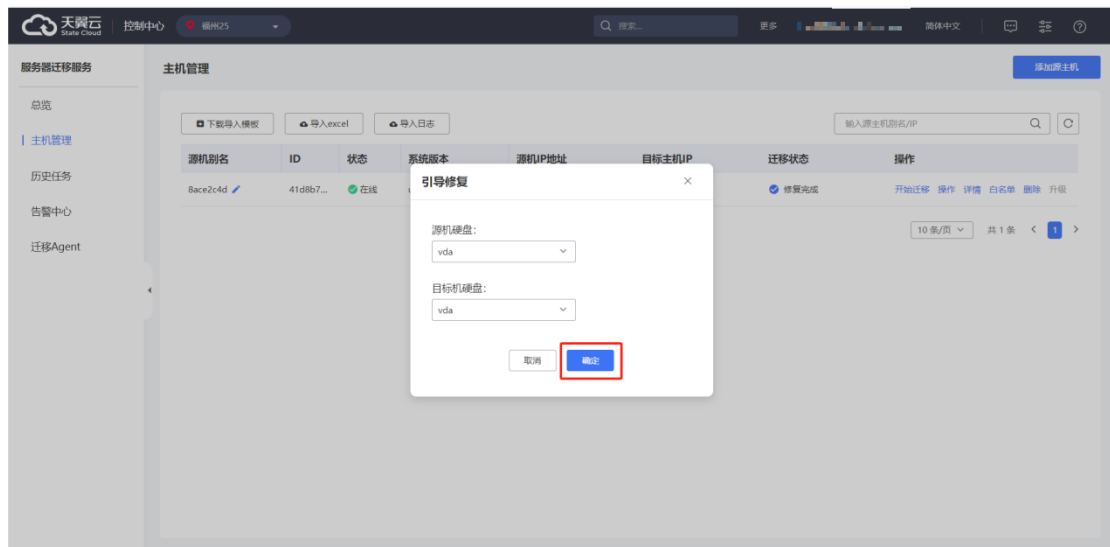
步骤6 单击“开始修复”，进入修复阶段，等待修复完成。



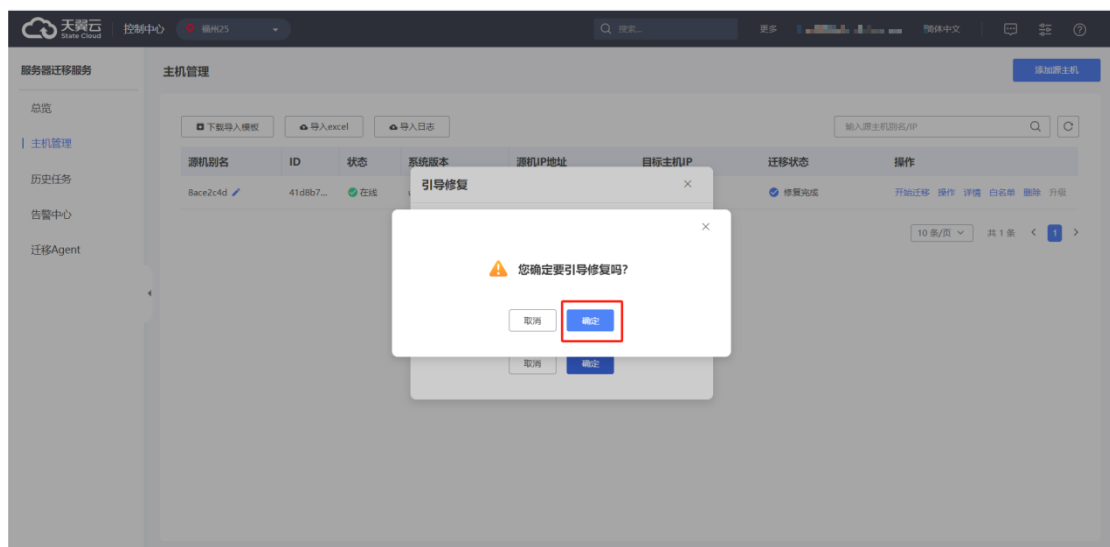
步骤7 单击“引导修复”。



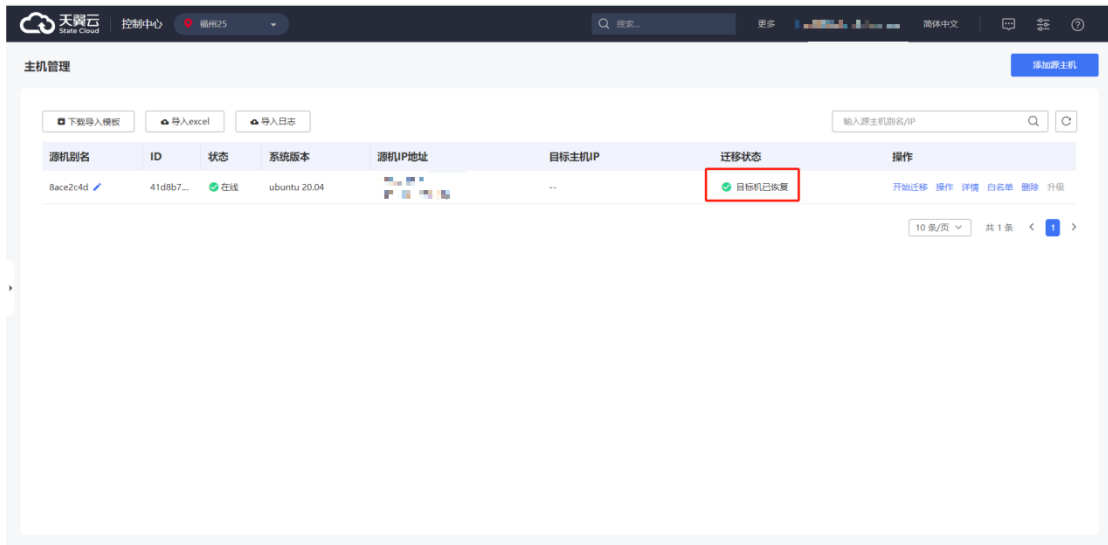
选择源机以及目标机对应的引导盘，单击“确定”。



单击“确定”。



目标机恢复中，等待目标机恢复完成。



目标机恢复完成。

注意：引导修复为切换目标机业务的操作，即将原业务切换至目标机上启动运行。

3.4 迁移后 oracle 业务启动

步骤 1 进入云主机控制台，选择目标端云主机，点击远程登陆。

步骤 2 输入账号密码，检查账号密码等软件配置均和源机一致。

步骤 3 检查目标机磁盘空间使用率是否与源机基本一致

注意：迁移工具会自动将磁盘中的碎文件进行重新写入和重组，因此磁盘的占用大小将略低于源机，而迁移后的文件大小总量是同源机一致的。

执行命令：df -TH

```
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=ABE72B9D-1400-1C44-9FDD-CA90331AE00B
root@cms-best-platform:~# df -B1
Filesystem      1B-blocks      Used    Available Use% Mounted on
udev            8070127616          0   8070127616   0% /dev
tmpfs           1623605248          0   1621700608   1% /run
/dev/vda2       105356070912  65537937408  35077201920  66% /
tmpfs           8118009856          0   8118009856   0% /dev/shm
tmpfs           5242880            0    5242880   0% /run/lock
tmpfs           8118009856          0   8118009856   0% /sys/fs/cgroup
/dev/vdb        214643507200  155155283968  59488223232  73% /app
tmpfs          1623601152          0   1623601152   0% /run/user/1000
/dev/vdc2      101917515776   93033177688   3659968512   97% /u01
overlay        105356070912  65537937408  35077201920  66% /var/lib/docker/overlay2/77d2c7389f6b530c1abb2aeb0c2a6cf3a930b595e879d5db2e01886a4d3bf40a/merged
overlay        105356070912  65537937408  35077201920  66% /var/lib/docker/overlay2/3f81ff3bf6b02bc46a4401b92c41b53eab0e70de64f635e80e5383e09ef06/merged
overlay        105356070912  65537937408  35077201920  66% /var/lib/docker/overlay2/85ee9dd385ef32242d0f6a70e66365525bf4b54579f5af7b2d49a31d50f41d9/merged
overlay        105356070912  65537937408  35077201920  66% /var/lib/docker/overlay2/9fa93c17546951d8a09188d8de8cad799f7511850015029f87b7a67682ad1108/merged
overlay        105356070912  65537937408  35077201920  66% /var/lib/docker/overlay2/8ea55cc38595572454d1d5d2713c577cbade353d8995f9d1335b9d38eaae6a8/merged
overlay        105356070912  65537937408  35077201920  66% /var/lib/docker/overlay2/32479796a35c48c913576e464c0b1e296374e709b8491b920baaf808b164365a/merged
overlay        105356070912  65537937408  35077201920  66% /var/lib/docker/overlay2/5f047f1848d96008ed68cb89085b0b31fbc4c003093e5b1f16c1ca23444fe77d/merged
overlay        105356070912  65537937408  35077201920  66% /var/lib/docker/overlay2/71beae654ac935c25b26e703ecac66ec4735fc7e68e2837bd019c6ed50bb0dce/merged
root@cms-best-platform:~#
```

```
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=390B9FB6-70FD-EA40-BD2B-65A9B0F8C7F9
root@cms-best-platform:~# df -B1
Filesystem      1B-blocks      Used    Available Use% Mounted on
udev            8070131712         0   8070131712    0% /dev
tmpfs          1623605248      1835008   1621770240    1% /run
/dev/vda2      105489817600  65580593152  34523844608    66% /
tmpfs          8118009856         0   8118009856    0% /dev/shm
tmpfs          5242880         0    5242880    0% /run/lock
tmpfs          8118009856         0   8118009856    0% /sys/fs/cgroup
/dev/vdb3      1023201688960  93044527104  4051271680    96% /ub1
/dev/vdb1      214643507200  153670729728  6097277472    72% /app
overlay        105489817600  65580593152  34523844608    66% /var/lib/docker/overlay2/d4b30970c0ef5f6c30f46070f3d8bd0c522b988e76809f0d79055ac870abe8ce/merged
overlay        105489817600  65580593152  34523844608    66% /var/lib/docker/overlay2/85ee9dd385efe32242d0fea70e66365525bf4b54579f5affb2d49a31d50f41d9/merged
overlay        105489817600  65580593152  34523844608    66% /var/lib/docker/overlay2/3f81ff3bf6b02bc46a4401b92c41b53eab6b0e70de64f6350e80e53e3e09ef06/merged
overlay        105489817600  65580593152  34523844608    66% /var/lib/docker/overlay2/77d2c7389f0b530c1abb2eeb9c2abc3e938b595e879d5db2e01880a4d3bf40a/merged
overlay        105489817600  65580593152  34523844608    66% /var/lib/docker/overlay2/8ea55c38505572454dd1d5d2713c577cbade353d8965f9d1335b6d38eaae6a8/merged
overlay        105489817600  65580593152  34523844608    66% /var/lib/docker/overlay2/32479796a35c48c913576e464c0b1e296374e709b8491b920baaf800b164365a/merged
tmpfs          1623601152         0   1623601152    0% /run/user/1000
overlay        105489817600  65580593152  34523844608    66% /var/lib/docker/overlay2/5f047f1848d96008ed68cb89085b0b31fbc4c003093e5b1f16c1ca23444fe77d/merged
overlay        105489817600  65580593152  34523844608    66% /var/lib/docker/overlay2/71beaa654ac935c25b26e703ecac66ec4735fc7e68e2837bd019c6ed50bb0dce/merged
root@cms-best-platform:~#
```

步骤 4 查看数据库服务状态，如果数据库为关闭状态则开启数据库。

启动容器服务：systemctl stop docker.socket

启动数据库：docker start oracle

步骤 5 检查增量数据是否同步登录并查看源端数据库中的数据情况，my_table 表中的记录数为 1100 条。

```
SQL> SELECT COUNT(*) AS record_count
  2 FROM my_table;

RECORD_COUNT
-----
          1100

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
[oracle@b4236baale49 /]$ exit
exit
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=ABE72B9D-1400-1C44-9FFD-CA90331AE0DB
root@cms-best-platform:~#
```

登录并查看目标机启动的数据库，查看数据库中的数据情况，my_table 表中的记录数为 1100 条，与源机相同。

```
SQL> SELECT COUNT(*) AS record_count
  2 FROM my_table;

RECORD_COUNT
-----
          1100

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
[oracle@b4236baale49 /]$
[oracle@b4236baale49 /]$ sudo lshw | grep uuid
bash: sudo: command not found
[oracle@b4236baale49 /]$ exit
exit
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=390B9FB6-70FD-EA40-BD2B-65A9B0F8C7F9
root@cms-best-platform:~#
```

步骤 6 检查数据库版本是否一致

检查源机数据库系统版本：

```
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=ABE72B9D-1400-1C44-9FFD-CA90331AE0DB
root@cms-best-platform:~# docker exec -it b4236baale49 bash
[oracle@b4236baale49 /]$ sqlplus -v

SQL*Plus: Release 12.2.0.1.0 Production

[oracle@b4236baale49 /]$ █
```

检查目标机数据库系统版本，同源机一致：

```
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=390B9FB6-70FD-EA40-BD2B-65A9B0F8C7F9
root@cms-best-platform:~# docker exec -it b4236baale49 bash
[oracle@b4236baale49 /]$ sqlplus -v

SQL*Plus: Release 12.2.0.1.0 Production

[oracle@b4236baale49 /]$ █
```

步骤 7 检查操作系统版本是否一致

检查源机操作系统版本：

```
root@cms-best-platform:~# cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.6 LTS"
NAME="Ubuntu"
VERSION="20.04.6 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.6 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=ABE72B9D-1400-1C44-9FFD-CA90331AE0DB
root@cms-best-platform:~# █
```

检查目标机操作系统版本，同源机一致：

```
root@cms-best-platform:~# cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.6 LTS"
NAME="Ubuntu"
VERSION="20.04.6 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.6 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=390B9FB6-70FD-EA40-BD2B-65A9B0F8C7F9
root@cms-best-platform:~# █
```

步骤 8 检查数据库文件总数是否一致

检查源机 oracle 数据库所在挂载点的文件总数：


```
root@cms-best-platform:~# find /u01/ -type f | wc -l
82644
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=ABE72B9D-1400-1C44-9FFD-CA90331AE0DB
root@cms-best-platform:~#
```

检查目标机 oracle 数据库所在挂载点的文件总数, 同源机一致:

```
root@cms-best-platform:~# find /u01/ -type f | wc -l
82644
root@cms-best-platform:~# sudo lshw | grep uuid
configuration: boot=normal family=Virtual Machine uuid=390B9FB6-70FD-EA40-BD2B-65A9B0F8C7F9
root@cms-best-platform:~#
root@cms-best-platform:~#
```