

云手机服务器

最佳实践

文档版本

04

发布日期

2026-08-27



版权所有 © 华为技术有限公司 2026。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目 录

1 云手机接入最佳实践	1
1.1 创建云手机服务器	1
1.2 连接云手机并获取云手机画面	6
1.3 云手机内管理程序保活方案	8
1.4 云手机应用部署	12
1.4.1 获取云手机列表	12
1.4.2 为单台云手机安装应用	13
1.4.3 生成应用版本 tar 包并推至 OBS 桶	14
1.4.4 部署应用	14
1.4.5 更新应用版本	16
2 云手机共享应用管理最佳实践	18
2.1 共享应用特性综述	18
2.2 共享空间中的应用包管理	20
2.2.1 配置共享空间	20
2.2.2 准备共享应用 tar 包	20
2.2.2.1 获取云手机列表	20
2.2.2.2 为单台云手机安装应用	21
2.2.2.3 生成应用版本 tar 包并推至 OBS 桶	22
2.2.3 推送应用 tar 包至服务器共享空间	23
2.2.4 从服务器共享空间删除应用	24
2.3 共享应用使用场景	24
2.3.1 云手机按需安装	25
2.3.1.1 应用安装	25
2.3.1.2 应用更新	25
2.3.1.3 应用卸载	26
2.3.1.4 应用批量卸载	26
2.3.2 云手机预装应用	26
2.3.2.1 预装应用安装	27
2.3.2.2 预装应用更新	27
2.3.2.3 预装应用卸载	27
2.3.3 云手机部署文件	28
2.3.3.1 文件预置	28
2.3.3.2 文件按需部署	29

2.3.3.3 已部署文件更新.....	30
2.3.3.4 取消文件预置.....	30
2.3.3.5 使用多个配置文件包部署文件.....	30
2.4 appctrl 命令.....	31
2.4.1 appctrl 命令的执行方法.....	31
2.4.2 appctrl install 安装应用.....	32
2.4.3 appctrl start 安装并启动应用.....	32
2.4.4 appctrl uninstall 卸载应用.....	33
2.4.5 appctrl clear 批量卸载应用.....	34
2.4.6 appctrl applyConfig 部署文件.....	34
3 批量安装应用至云手机.....	35
4 修改云手机定位.....	39
5 使用云手机摄像头.....	42
6 国内云手机服务器导流海外.....	45
7 委托 CPH 操作 OBS 桶.....	50
8 云手机 AOSP 版本切换.....	54

1

云手机接入最佳实践

1.1 创建云手机服务器

操作步骤

1.

登录CPH控制台，进入购买云手机服务器页面。
2.

根据界面提示，完成基础配置，如表1-1。

表 1-1 基础配置参数说明

参数	参数说明	样例
计费模式	服务器仅支持一种计费类型：包年/包月。	包年/包月
区域	不同区域的云服务产品之间内网互不相通。建议您选择最靠近您业务的区域，这样可以减少网络时延、提高访问速度。 需要注意：服务器购买成功后不能更换区域。	中国-香港
可用区	指在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。 - 如果您需要提高应用的高可用性，建议您将服务器创建在不同的可用区。 - 如果您需要较低的网络时延，建议您将服务器创建在相同的可用区。	可用区1
服务器类型	请根据业务场景进行选择。详情请参见“云手机服务器规格”。	云手机服务器 physical.rx1.xlarge

参数	参数说明	样例
实例规格	请根据业务场景进行选择。	rc1.se
手机镜像	云手机实例运行的操作系统，目前只支持Android系统，其他手机系统由于商业授权缘故，不能提供。 说明 查看私有手机镜像列表需要具有对应的细粒度权限：ims:images:list。	AOSP7.1.1
购买量	- 服务器的数量，一次最多可购买10台服务器。 - 购买时长：可选取的时间范围为1个月~3年。	服务器数量：1 购买时长：6个月

3. 单击“下一步：网络配置”。根据界面提示，完成网络配置。
推荐您使用自定义网络配置，如表1-2所示。

表 1-2 自定义网络配置

参数	参数说明	样例
网络	在下拉列表中选择可用的虚拟私有云、子网，并设置私有IP地址的分配方式。 请确定您的登录用户至少具有“VPC ReadOnlyAccess”或“CPH AgencyDependencyAccess”权限。 云手机网络使用虚拟私有云（VPC）提供的网络，包括子网、安全组等。您可以选择使用已有的虚拟私有云网络，或者创建新的虚拟私有云。 ● 暂不分配IPv6地址/自动分配IPv6地址：当且仅当选择部分区域、部分规格的云手机服务器、且VPC子网开启了IPv6功能时，该参数可见。子网如何开启IPv6功能，请参见IPv4/IPv6双栈网络。系统默认分配IPv4地址，当选择“自动分配IPv6地址”后，网卡的IP地址为IPv4/IPv6双栈类型。 ● 暂不配置/选择需要的共享带宽：在同一VPC内，云手机服务器可通过IPv6地址在双栈服务器之间进行内网访问。如需访问外网，您需要在下拉列表选择您的共享带宽，将IPv6地址加入您的共享带宽。此时云手机服务器可以通过IPv6地址与互联网上的IPv6网络进行访问。 说明 ● 在创建云手机服务器时，一旦开启IPv6功能，开启成功后，不能修改。 ● 由于VPC限制，上海二不支持开启IPv4/IPv6双栈。 ● IPv6暂不支持使用独享带宽。 ● 单个共享带宽默认能加入的IPv6地址数量最多为20个，双栈云手机服务器IPv6网卡数量和VIP数量相同。若您要购买e0v100等多VIP规格的云手机服务器时，建议您提前申请扩大共享带宽的容量配额。 ● IPv6不支持规格为RX1的服务器。	无
安全组授权	云手机服务器服务将为您创建一个“cph_admin_trust”委托，该委托包含的权限为：“CPH AgencyDependencyAccess”。 了解更多请参考权限管理。 云手机服务器服务将使用该委托完成如下操作： ● 为云手机实例创建弹性网卡、弹性公网IP、虚拟IP。 ● 为云手机服务器创建默认安全组，并设置安全组，开放端口范围，该端口开放范围将映射到每个云手机/云游戏实例以支持实例开放应用访问端口。 说明 同一VPC下的弹性云服务器默认无法通过1-9999端口访问云手机/云游戏实例，若想放开此限制，请参考使用自定义网络的安全组授权规则添加高优先级的安全组规则。	无

参数	参数说明	样例
弹性公网IP	● 现在购买：为云手机购买新的弹性公网IP。 ● 使用已有：为云手机分配已有弹性公网IP。	现在购买
线路	● 静态BGP中的网络结构发生变化，运营商无法在第一时间自动调整网络设置以保障用户的体验度。 ● 全动态BGP可根据设定的寻路协议第一时间自动优化网络结构，以保持客户使用的网络持续稳定、高效。	全动态BGP
公网带宽	仅在新购买弹性公网IP场景下可选择： ● 按流量计费：按照实际使用的流量来计费。 ● 加入共享带宽：一个带宽中可以加入多个弹性公网IP，多个弹性公网IP共用一个带宽。	加入共享带宽
带宽大小	可选带宽范围：1-300 Mbit/s。	300 Mbit/s
带宽名称	“公网带宽”选择“加入共享带宽”时，请在下拉框中选择已有的共享带宽名称。	bandwidth h-001

4. 单击“下一步：高级配置”。根据界面提示，完成高级配置，如表1-3所示。

表 1-3 参数说明

参数	参数说明	样例
名称	为您购买的服务器和云手机实例命名，名称不可重复。 命名规则：购买服务器后，名称自动按序增加数字后缀；服务器对应的云手机实例则自动按序增加5位数字后缀。 例如，您购买了1台服务器，该服务器对应60台云手机，名称输入CPH，则服务器名称为CPH-1，云手机实例的名称为CPH-1-00001~CPH-1-00060。	CPH

参数	参数说明	样例
密钥对	使用密钥对（Key Pair）进行远程登录身份验证。 - 如果您已经创建过密钥对，并且本地已妥善保存私钥文件（.pem格式），可以在下拉列表选择已有密钥对。 - 如果您从未创建过密钥对，请单击“新建密钥对”，跳转至云服务器控制台进行新建。然后返回该页面，刷新下拉列表，选择创建好的密钥对。 私钥用于远程登录身份认证，为保证安全，私钥文件（.pem格式）只能下载一次，请妥善保管。更多关于密钥对的介绍请参见“创建密钥对”。 说明 - 请确保账号具有查询密钥对的细粒度权限 ecs:serverKeyPairs:list。 - 若账号需要创建密钥对，请确保账号具有创建密钥对的细粒度权限 ecs:serverKeyPairs:create。	KeyPair-test
应用端口	高级配置勾选“应用端口”时，该参数可配。适用于云手机需要对外提供服务的场景。 - 应用名称：支持大小写英文字母。不能为关键字“ADB”，大小写的任意组合都不行。 - 端口号：在0~65535之间。 - 公网访问 - 勾选时，表示可通过公网访问云手机该应用端口，即云手机对应的端口以及相对应的服务器公网端口直接暴露在公网，无需鉴权即可互通。 - 不勾选时，表示只能通过租户的私有网络访问。 注意 - 请谨慎勾选“公网访问”，确保在勾选前已做好安全控制。 - 云手机服务未对客户选择打开的端口做任何安全鉴权。	key 10001 不勾选

- 单击“下一步：确认订单”，您可再次核对信息。
 - 确认无误后，单击“立即购买”。
 - 如果还需要修改，单击“上一步”，修改参数。
- 根据界面提示，完成支付。

大约需要20~30分钟，购买服务器成功后，系统会自动创建好云手机。

云手机状态变为“运行中”时，表示云手机实例可使用。

1.2 连接云手机并获取云手机画面

Airtest是跨平台UI自动化编译器，可使用Airtest工具快速获取云手机画面。

前提条件

- 已购买并通过ADB方式登录云手机。详细操作请参见“[购买云手机服务器](#)”。
- 已在本地PC安装Airtest工具。

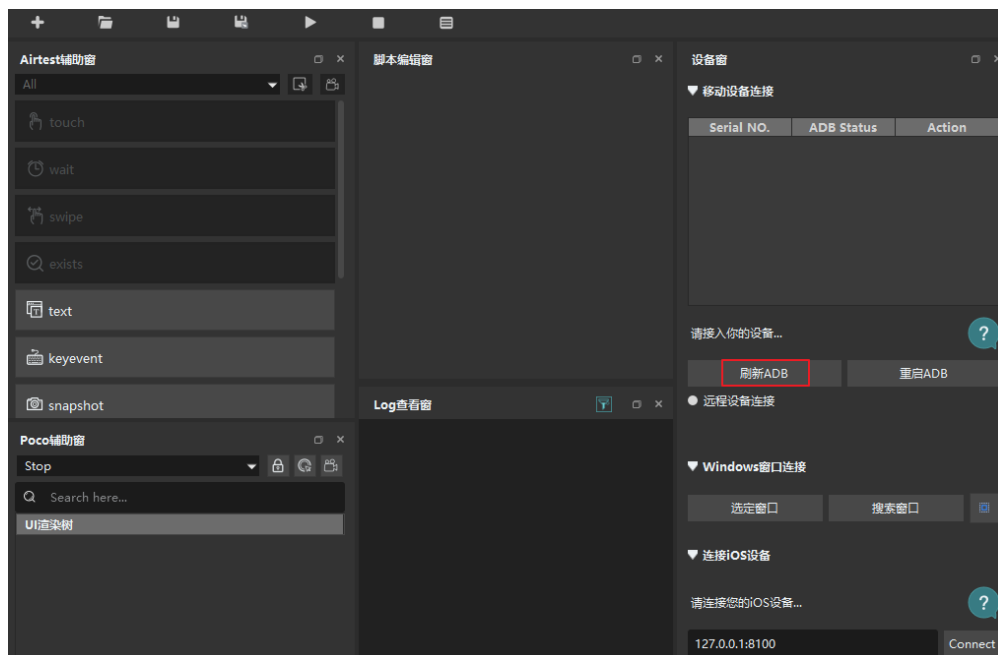
说明

- 登录Airtest官网（<https://airtest.netease.com/>），下载符合您操作系统的版本并安装。
- 已关闭ADB连接的命令行窗口，并保证SSH隧道建立成功。

操作步骤

- 在Airtest主页单击“刷新ADB”，出现已连接的移动设备：

图 1-1 Airtest 主页



- 如果没有出现您想要连接的设备，可选择下方的“远程设备连接”，然后手动输入对应云手机的ADB连接命令，如[图1-2](#)。

adb connect 127.0.0.1:1234

其中，1234为建立SSH隧道时所使用的本地空闲端口。

图 1-2 远程设备连接



单击右侧的“连接”后，“移动设备连接”中即会出现所需连接的云手机。

 注意

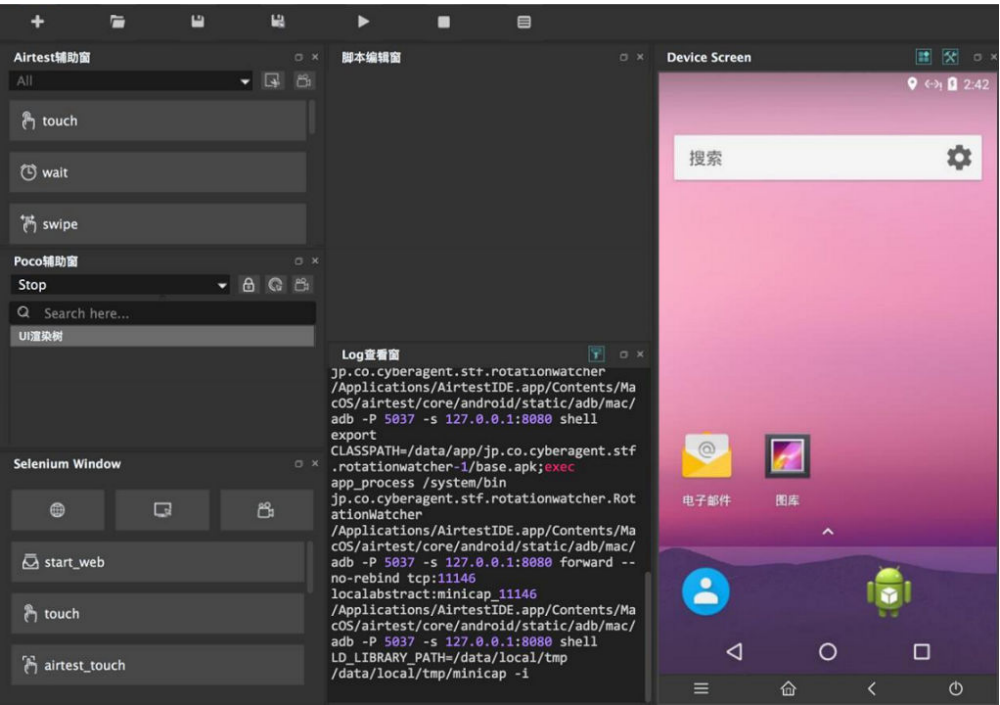
请确保ADB连接的命令行窗口已关闭，否则会连接失败；并且保证SSH隧道建立成功，否则即使已识别出移动设备，“ADB Status”也会出现“offline”状态，导致无法获取云手机画面。

3. 在已识别的移动设备列表中单击对应设备右侧的“connect”，即可获取云手机画面：

图 1-3 移动设备列表

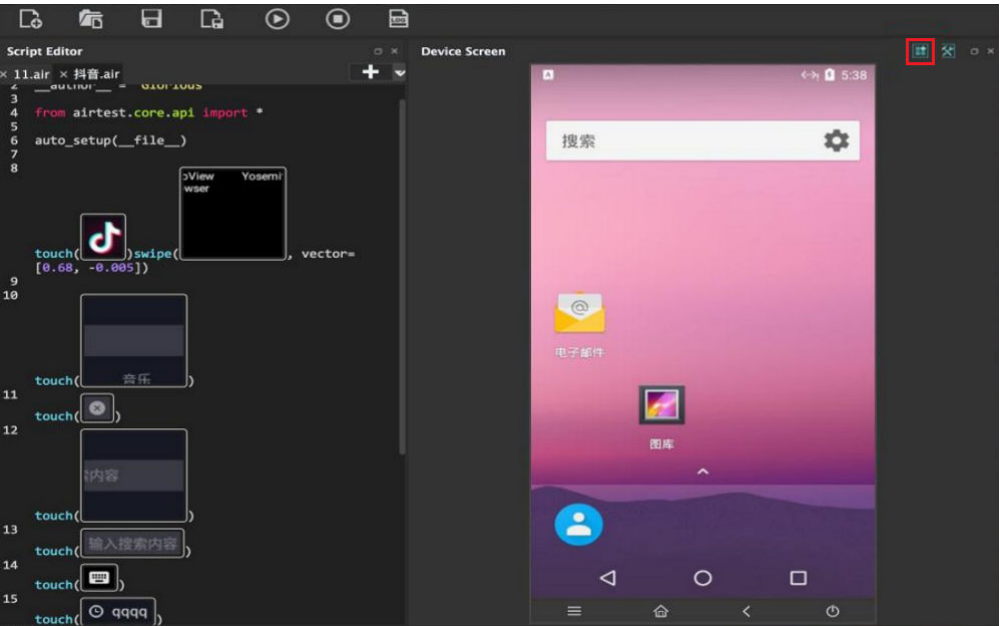


图 1-4 云手机画面



4. 若通过ADB连接了多台云手机，需要切换画面，可单击右上角的切换图标进行切换。

图 1-5 切换云手机



1.3 云手机内管理程序保活方案

管理程序形态分类

- 在云手机内的管理程序，基本分为两类：
- Android APK Service 形态，即以Android无UI的Service的形式在云手机后台运行。
- Android JNI Native 形态，即用Android JNI开发的Executable的二进制可执行程序。

管理程序保活方案

两种形态的管理程序保活方案都依赖一个关键点：extend_custom.sh，即在云手机可内置的钩子脚本。详细信息请参见[开机回调脚本](#)。

钩子脚本 extend_custom.sh 的机制说明：

云手机在系统 boot_complete 之后会检查 /data/local/tmp 目录下是否有 extend_custom.sh 脚本，如果有则会执行此脚本。客户可以在此脚本中执行自己文件的操作和移动，也可以启动和管理自己的程序等，但是有个限制条件是脚本执行超时时间是10s。

1. Service形态的管理程序保活

Service形态的程序由于系统机制限制，首次安装之后，如果不启动是属于处于停止状态的，而开机启动完成的广播使用了FLAG_EXCLUDE_STOPPED_PACKAGES，会使得处于停止状态的应用收不到开机启动完成广播。因此，service管理程序的的安装和首次启动都需要依赖在云手机 /data/local/tmp 目录下内置 extend_custom.sh 脚本来实现，后续重启开机就可以通过接收开机启动完成广播来自启动。

自启动可以在AndroidManifest.xml里配置接收开机广播并在对应的广播处理里启动Service。

```
<!-- example code-->
```

```
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<!-- 适配 8.0及以上版本 -->
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>

<receiver android:name=".DemoServiceReceiver">
    <intent-filter android:priority="1000">
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</receiver>
```

开机广播处理并启动Service:

```
// example code
public class DemoServiceReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (!intent.getAction().equals("android.intent.action.BOOT_COMPLETED")) {
            return;
        }
        Intent serviceIntent = new Intent(context, DemoService.class);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            context.startForegroundService(serviceIntent);
        } else {
            context.startService(serviceIntent);
        }
    }
}
```

service 程序的保活可以在对应的 Service 类里的 onStartCommand 函数里返回 START_STICKY，这样的话，当 Service 进程被系统 kill 后可以被重新拉起。

```
// example code
public class DemoService extends Service {
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        //other todo...

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) { //适配8.0及以上
            String channel_id = "MyTestService-id";
            String channel_name = "MyTestService-name";
            NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
            NotificationChannel channel = new NotificationChannel(channel_id, channel_name,
                NotificationManager.IMPORTANCE_HIGH);
            if (manager != null) {
                manager.createNotificationChannel(channel);
            }
            Notification notification = new
                Notification.Builder(this).setChannelId(channel_id).setSmallIcon(R.mipmap.ic_launcher).build();
            startForeground(100, notification);
        }
        return START_STICKY;
    }
}
```

安装和首次启动依赖的 extend_custom.sh 脚本实现，示例如下（示例中默认apk文件也在 /data/local/tmp 目录下）：

```
# example code

# !/system/bin/sh

# APK名称
ApkName=service.apk
# APK包名
PackageName=com.huawei.myapplication
# Service
ServiceName=com.huawei.myapplication/.MyService
```

```
InstallService() {
    count=`pm list packages | grep $PackageName |wc -l`
    if [ $count -le 0 ]; then
        echo "$ApkName is not installed, now install."
        ret=`pm install $ApkName`
        Succ="Success"
        if [ "$ret" == "$Succ" ]; then
            echo "install succeed."
        else
            echo "install failed."
            exit 1
        fi
    else
        echo "$ApkName is already installed."
        echo "Service start by boot_complete broadcast."
        exit 1
    fi
    return 0
}

StartService() {
    ret=`am startservice -n $ServiceName`
    Err="Error: Not found; no service started."
    contain=$(echo $ret | grep "${Err}")
    if [[ "$contain" != "" ]]; then
        echo "Service start failed."
        exit 1
    else
        echo "Service start succeed."
    fi
    return 0
}

main() {
    # 安装
    InstallService

    echo "To start when first installed."
    # 启动
    StartService
}
```

2. Native形态的管理程序保活

Native 管理程序需要被放在云手机 /system/bin 目录下，并给予可执行权限，Native 程序依赖的二进制so库需要放在 system/lib 和 system/lib64 目录下。

Native程序、二进制so库文件等都可以通过共享存储或共享应用的方式推进手机的数据目录，然后利用 extend_custom.sh 脚本将自己的文件放置在对应的目录下。针对使用诉求不同，有以下几种方式：

（1）系统级保活

可以实现 init_custom.rc 文件，并将文件移动到 /data/local/目录下，然后需要重启手机生效，重启手机后系统会扫描到 init_custom.rc 文件并在 boot_completed 之后启动 NativeDemo（示例中 Native 管理程序名），如果 NativeDemo 进程异常退出或被kill都会被系统再次拉起。

```
on property:sys.boot_completed=1
    start NativeDemo

service NativeDemo /system/bin/NativeDemo
    user root
    group root
    disabled
    writepid /dev/cpuset/system-background/tasks
```

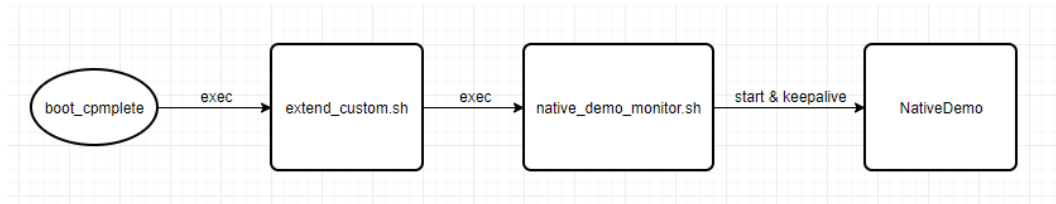
系统级保活方式有以下优缺点：

优点：系统级的自启动和保活，可靠性高，拉起实时性高。

缺点：需要重启一次手机才可生效。

（2）用户级保活

如果既需要开机自启又需要保活管理程序，而且不希望重启手机生效，可以先在 extend_custom.sh 脚本中执行一个管理程序的检测保活脚本 native_demo_monitor.sh，由管理程序的检测保活脚本真正实现 NativeDemo 管理程序的启动和保活。



extend_custom.sh 钩子脚本实现示例：

```
# example code
#!/system/bin/sh

[[ -f /data/local/tmp/native_demo_monitor.sh ]] && sh /data/local/tmp/native_demo_monitor.sh &
```

native_demo_monitor.sh 脚本实现示例：

```
# example code
#!/system/bin/sh

file_dir="/data/demo"

CopyFile() {
    cp -rf $file_dir/NativeDemo /system/bin/
    chmod 755 /system/bin/NativeDemo

    cp $file_dir/lib*.so /system/lib64/
}

CheckIfCopyFile() {
    if [ -s $file_dir ]; then
        echo "file exist"
        CopyFile
    else
        echo "file not exist"
    fi
}

Start() {
    nohup NativeDemo &
}

KeepAlive() {
    while do
        echo "check NativeDemo proc"
        proc_count=`ps | grep NativeDemo | wc -l`
        if [ $proc_count -le 0 ]; then
            echo "start NativeDemo"
            Start
        else
            echo "NativeDemo already started"
        fi
        sleep 5
    done
}
```

```
main() {
    CheckIfCopyFile
    KeepAlive
}

main
```

用户级保活方式有以下优缺点：

优点：不需要重启手机即可生效。

缺点：拉起的实时性不如系统级，依赖检查的循环间隔时间。

1.4 云手机应用部署

1.4.1 获取云手机列表

参考《云手机服务器API参考》的“[查询云手机列表](#)”章节获取云手机列表。

接口示例

```
GET https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones?
phone_name={phone_name}&server_id={server_id}&status={status}&offset={offset}&limit={limit}&type={type}
Header:
Content-Type: application/json
X-Auth-Token: ${token}
```

其中，

- CPH Endpoint为Endpoint列表中CPH对应区域的终端节点，如中国-香港为cph.ap-southeast-1.myhuaweicloud.com。
- project_id为云手机服务器所属区域对应的项目ID，如083e9f825e80f50c2f96c0045edc70e8。可通过如下方式获取：
 - a. 登录管理控制台。
 - b. 单击右上角用户名下的“我的凭证”。
 - c. 在“API凭证”页面的项目列表中获得项目ID。

图 1-6 获取项目 ID



- URL中"?"之后的部分为可选参数。
- \$token为[获取Token](#)接口的响应结果。

接口使用示例

```
GET https://cph.cn-north-4.myhuaweicloud.com/v1/083e9f825e80f50c2f96c0045edc70e8/cloud-phone/phones
Header:
Content-Type: application/json
X-Auth-Token: ${token}
```

说明

`${token}`需要换成实际获取到的token值。

1.4.2 为单台云手机安装应用

参考《云手机服务器API参考》的“[安装apk](#)”章节为单台云手机安装应用。

前提条件

- 云手机服务器所在Region的OBS桶中已经存放好对应的APK安装包。否则，需要上传安装包，可参考[场景二：快速通过OBS Browser+上传下载文件](#)进行上传。
- 确保已经配置好OBS桶的授权策略，若未配置请参考[委托CPH操作OBS桶](#)章节来完成配置。

接口示例

```
POST https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones/commands
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "command": "install",
  "content": "-t -r obs://{bucket_name}/{object_path}",
  "phone_ids": [
    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  ]
}
```

其中，

- CPH Endpoint、project_id、`${token}`等参数取值可以参考[获取云手机列表](#)章节的说明获取。
- bucket_name为对象存储服务OBS的桶名，object_path为apk安装包的存放路径。
- phone_ids为需要安装应用的云手机ID（通过[获取云手机列表](#)获得，可以填多个，填多个则为多台云手机安装apk应用）。

接口使用示例

```
POST https://cph.cn-east-3.myhuaweicloud.com/v1/081ceeb7fb800f0c2f4cc004bb39c2f7/cloud-phone/phones/commands
Content-Type: application/json
X-Auth-Token: ${token}
{
  "command": "install",
  "content": "-t -r obs://yzw-apk-install/apk/com.hermes.bgame.apk",
  "phone_ids": [
    "bdc2f2e960164dd9a2765374afeea300"
  ]
}
```

- yzw-apk-install为OBS桶名，apk/com.hermes.bgame.apk为安装包存放路径，obs://yzw-apk-install/apk/com.hermes.bgame.apk为安装包全路径。

- `${token}`需要换成实际获取到的token值。

1.4.3 生成应用版本 tar 包并推至 OBS 桶

前提条件

- 需确保云手机已安装对应的应用。
- 确保已配置好OBS桶的授权策略，具体操作请参考[委托CPH操作OBS桶](#)章节。

接口示例

```
POST https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones/batch-storage
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "storage_infos": [{
    "phone_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "include_files": [
      "/data/app/${package_name}-*",
      "/data/data/${package_name}",
      "/data/media/0/Android/data/${package_name}"
    ],
    "bucket_name": "${bucket_name}",
    "object_path": "apk/${package_name}_${version_name}.tar"
  }]
}
```

其中，

- CPH Endpoint、project_id、`${token}`、bucket_name、object_path等参数可以参考[为单台云手机安装应用](#)章节的说明获取。
- phone_id为安装了对应应用的云手机ID。
- include_files中的三个元素需要填写手机中的绝对路径。
- 如果该安装包为xapk类型，则需要在include_files中增加“/data/media/obb/\${package_name}”路径。
- object_path为tar包上传至OBS桶的目标路径。

须知

1. object_path中apk为obs桶中已存在文件夹，`${package_name}`为当前应用的包名，`${version_name}`为当前应用的版本号，版本号可自行定义。
2. 针对共享应用场景，部分应用在启动后，会进行资源文件的在线下载。对于此类应用，在执行当前操作前，建议先启动一次应用，待资源文件和补丁包下载完成后，再执行当前操作。后续以共享应用的方式安装到云手机后，该应用的启动过程可省去资源文件下载的过程。

1.4.4 部署应用

存储 2.0 机器（推荐）

推送tar包至服务器，即将文件“apk/\${package_name}_\${version_name}.tar”推送到服务器（`${server_id1}`和`${server_id2}`）的共享应用中。

- 接口示例

```
POST https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones/share-apps
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "package_name": "${package_name}"
  "bucket_name": "${bucket_name}",
  "object_path": "apk/${package_name}_${version_name}.tar",
  "server_ids": [
    "${server_id1}",
    "${server_id2}"
  ]
}
```

其中，

- CPH Endpoint、project_id、\${token}、bucket_name、object_path等参数可以参考[1.4.2-为单台云手机安装应用](#)的说明获取。
- package_name为应用在安卓系统中的包名，例如：com.miniteck.miniworld。
- object_path为tar包上传的目的路径。
- package_name为当前应用的包名，version_name为当前应用的版本号。

说明

apk为任意已存在文件夹，\${package_name}_\${version_name}.tar中除.tar后缀外，其余内容需根据实际名称来修改。

- server_ids为部署应用版本的目标服务器ID列表，可以填写多个，服务器ID可通过[“查询云手机服务器列表”](#)接口获得。

- 使用示例

参考《云手机服务器API参考》的[“推送共享应用”](#)章节。

存储 1.0 机器

推送tar包至服务器，即将文件“apk/\${package_name}_\${version_name}.tar”推送到服务器（\${server_id1}和\${server_id2}）的共享存储中。

- 接口示例

```
POST https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones/share-files
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "bucket_name": "${bucket_name}",
  "object_path": "apk/${package_name}_${version_name}.tar",
  "server_ids": [
    "${server_id1}",
    "${server_id2}"
  ]
}
```

其中，

- CPH Endpoint、project_id、\${token}、bucket_name、object_path等参数可以参考[4.2 为单台云手机安装应用](#)的说明获取。
- object_path为tar包上传的目的路径。
- package_name为当前应用的包名，version_name为当前应用的版本号。

📖 说明

apk为任意已存在文件夹，`\${package\_name}\_\${version\_name}.tar`中除tar后缀外，其余内容需根据实际名称来修改。

- server_ids为部署应用版本的目标服务器ID列表，可以填写多个，服务器ID可通过“[查询云手机服务器列表](#)”接口获得。
- 使用示例
参考《云手机服务器API参考》的“[推送共享存储文件](#)”章节。
- 后续操作
参考《云手机服务器API参考》“[重置云手机](#)”接口，批量重置所有云手机。

1.4.5 更新应用版本

存储 2.0 的机器（推荐）

确保手机已经卸载应用

推送最新的应用

随后，在需要时调用appctl start会启动最新版本应用

存储 1.0 的机器

更新应用版本，需要先删除服务器上的旧版本应用，然后重新部署新版本应用。

删除旧版本应用

- 接口示例：
POST [https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones/share-files](#)
Header:
Content-Type: application/json
X-Auth-Token: \${token}
Body:

```
{
  "file_paths": [
    "/data/app/${package_name}-1",
    "/data/app/${package_name}-2",
    "/data/data/${package_name}",
    "/data/media/0/Android/data/${package_name}"
  ],
  "server_ids": [
    "${server_id1}",
    "${server_id2}"
  ]
}
```

删除服务器（`\${server\_id1}`和`\${server\_id2}`）共享存储中的文件，文件列表包含：“/data/app/\${package_name}-1”，“/data/app/\${package_name}-2”，“/data/data/\${package_name}”，“/data/media/0/Android/data/\${package_name}”。

其中，

- CPH Endpoint、project_id、\${token}等参数可以参考[为单台云手机安装应用](#)的说明获取。
- file_paths的内容等同于[生成应用版本tar包并推至OBS桶](#)中的include_files，其中package_name为当前应用的包名。

- server_ids为部署应用版本的目标服务器ID列表,可以填写多个, 服务器ID可通过“[查询云手机服务器列表](#)”接口获得。
- 使用示例:
参考《云手机服务器API参考》的“[删除共享存储文件](#)”章节。

部署新版本应用

参考[部署应用](#)来部署应用。

2 云手机共享应用管理最佳实践

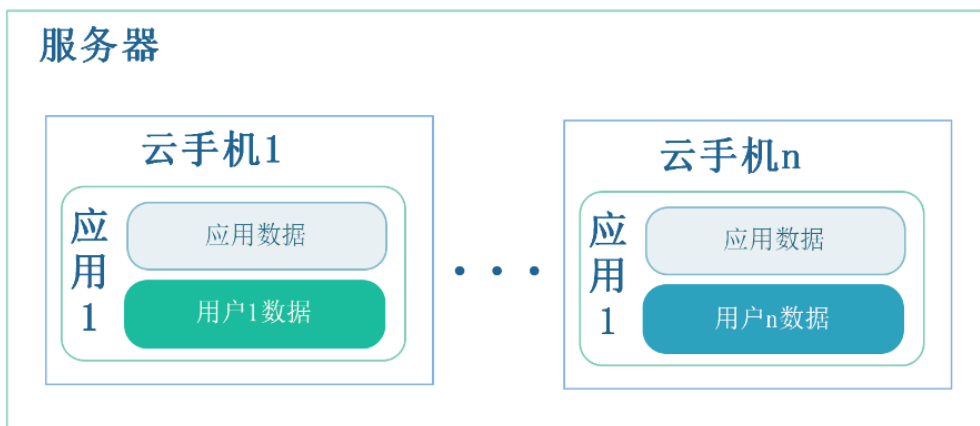
2.1 共享应用特性综述

共享应用是什么？

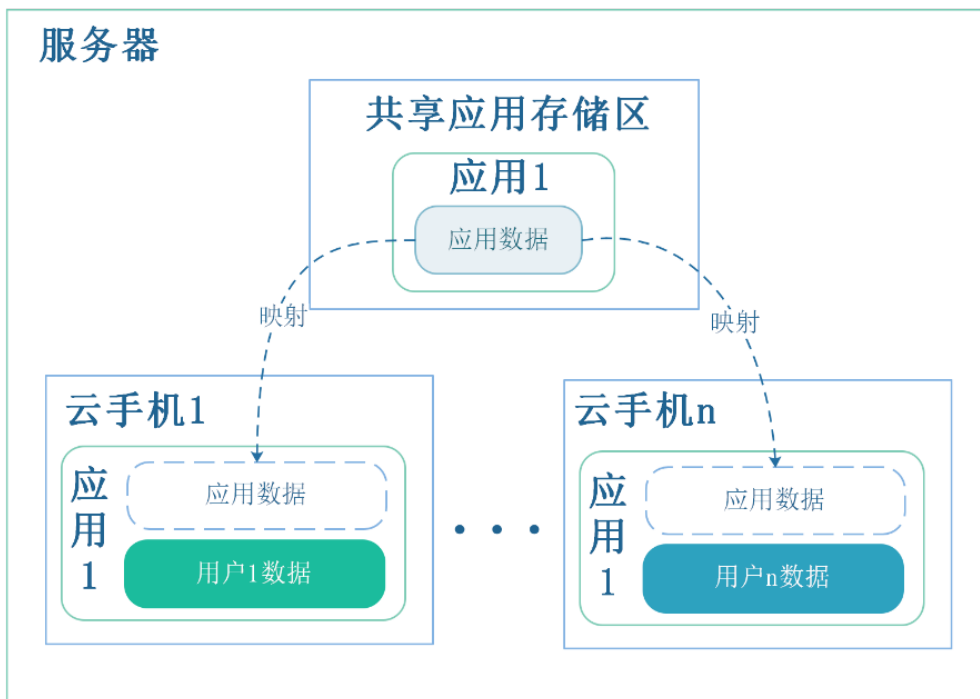
云手机服务器提供一块共享应用存储区（简称共享空间），服务器所属用户只需要下载/更新一次应用，然后将应用数据打包上传至指定服务器的共享空间，即可以实现该服务器上所有的云手机实例共享该应用。

图 2-1 共享应用

原生应用：



共享应用：



对于云手机服务器用户而言，共享应用可以避免同样的应用数据在多台云手机里重复下载、存储、更新，降低存储和网络成本。除此之外，共享应用还有以下优点：

- 部署方式灵活可控：云手机应用既可批量部署，也可按云手机实例单台部署。
- 方便快捷极致体验：将更新后的应用直接共享，简化安装流程，免除更新操作，真正实现即连即用。

共享应用有哪些典型场景？

1. 对于需要为服务器上的云手机实例按需安装、更新应用的场景，请参考[云手机按需安装](#)。

2. 对于需要为整台服务器上所有的云手机实例预安装、更新相同应用的场景，请参考[云手机预装应用](#)
3. 对于非应用类文件，可以使用共享应用的能力，实现文件的批量部署的场景，请参考[云手机部署文件](#)。

2.2 共享空间中的应用包管理

在使用共享应用前，首先需要确保为云手机服务器配置了合适的共享空间，共享空间大小在购买服务器时进行配置；其次，需要将应用的相关文件打包推送到服务器的共享空间中；当某应用不再使用时，可以将共享空间中的应用删除，以便释放共享空间来推送其他应用。

2.2.1 配置共享空间

在购买服务器实例页面，选定区域、服务器类型、实例规格后，配置共享空间大小。建议在实际业务共享应用所需空间大小的基础上，上浮30%来配置共享空间的大小，以满足后续应用更新对共享空间的需求。

图 2-2 配置共享空间



2.2.2 准备共享应用 tar 包

2.2.2.1 获取云手机列表

参考《云手机服务器API参考》的“[查询云手机列表](#)”章节获取云手机列表。

接口示例

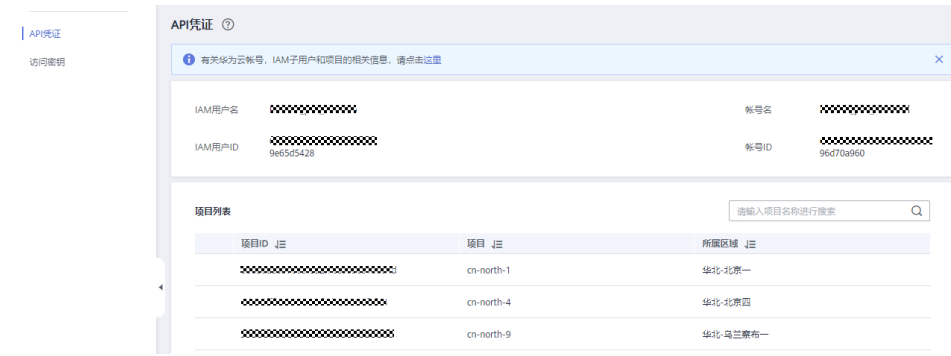
```
GET https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones?
phone_name={phone_name}&server_id={server_id}&status={status}&offset={offset}&limit={limit}&type={type}
Header:
Content-Type: application/json
X-Auth-Token: ${token}
```

其中，

- CPH Endpoint为Endpoint列表中CPH对应区域的终端节点，如中国-香港为cph.ap-southeast-1.myhuaweicloud.com。

- project_id为云手游服务器所属区域对应的项目ID，如083e9f825e80f50c2f96c0045edc70e8。可通过如下方式获取：
 - a. 登录管理控制台。
 - b. 单击右上角用户名下的“我的凭证”。
 - c. 在“API凭证”页面的项目列表中获取项目ID。

图 2-3 获取项目 ID



- URL中"?"之后的部分为可选参数。
- \$token为[获取Token](#)接口的响应结果。

接口使用示例

```
GET https://cph.cn-north-4.myhuaweicloud.com/v1/083e9f825e80f50c2f96c0045edc70e8/cloud-phone/phones
Header:
Content-Type: application/json
X-Auth-Token: ${token}
```

说明

`\${token}`需要换成实际获取到的token值。

2.2.2.2 为单台云手机安装应用

参考《云手机服务器API参考》的“[安装apk](#)”章节为单台云手机安装应用。

前提条件

- 云手机服务器所在Region的OBS桶中已经存放好对应的APK安装包。否则，需要上传安装包，可参考[场景二：快速通过OBS Browser+上传下载文件](#)进行上传。
- 确保已经配置好OBS桶的授权策略，若未配置请参考[委托CPH操作OBS桶](#)章节来完成配置。

接口示例

```
POST https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones/commands
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "command": "install",
  "content": "-t -r obs://{bucket_name}/{object_path}",
  "phone_ids": [
    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  ]
}
```

```
    ]  
  }
```

其中，

- CPH Endpoint、project_id、\${token}等参数取值可以参考[获取云手机列表](#)章节的说明获取。
- bucket_name为对象存储服务OBS的桶名，object_path为apk安装包的存放路径。
- phone_ids为需要安装应用的云手机ID（通过[获取云手机列表](#)获得，可以填多个，填多个则为多台云手机安装apk应用）。

接口使用示例

```
POST https://cph.cn-east-3.myhuaweicloud.com/v1/081ceeb7fb800f0c2f4cc004bb39c2f7/cloud-phone/  
phones/commands  
Content-Type: application/json  
X-Auth-Token: ${token}  
{  
  "command": "install",  
  "content": "-t -r obs://yzw-apk-install/apk/com.hermes.bgame.apk",  
  "phone_ids": [  
    "bdc2f2e960164dd9a2765374afeea300"  
  ]  
}
```

- yzw-apk-install为OBS桶名，apk/com.hermes.bgame.apk为安装包存放路径，obs://yzw-apk-install/apk/com.hermes.bgame.apk为安装包全路径。
- \${token}需要换成实际获取到的token值。

2.2.2.3 生成应用版本 tar 包并推至 OBS 桶

前提条件

- 需确保云手机已安装对应的应用。
- 确保已配置好OBS桶的授权策略，具体操作请参考[委托CPH操作OBS桶](#)章节。

接口示例

```
POST https://{CPH Endpoint}/v1/{project_id}/cloud-phone/phones/batch-storage  
Header:  
Content-Type: application/json  
X-Auth-Token: ${token}  
Body:  
{  
  "storage_infos": [{  
    "phone_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",  
    "include_files": [  
      "/data/app/${package_name}-*",  
      "/data/data/${package_name}",  
      "/data/media/0/Android/data/${package_name}"  
    ],  
    "bucket_name": "${bucket_name}",  
    "object_path": "apk/${package_name}_${version_name}.tar"  
  }]  
}
```

其中，

- CPH Endpoint、project_id、\${token}、bucket_name、object_path等参数可以参考[为单台云手机安装应用](#)章节的说明获取。

- phone_id为安装了对应应用的云手机ID。
- include_files中的三个元素需要填写手机中的绝对路径。
- 如果该安装包为xapk类型，则需要在include_files中增加“/data/media/obb/\${package_name}”路径。
- object_path为tar包上传至OBS桶的目标路径。

须知

1. object_path中apk为obs桶中已存在文件夹，\${package_name}为当前应用的包名，\${version_name}为当前应用的版本号，版本号可自行定义。
2. 针对共享应用场景，部分应用在启动后，会进行资源文件的在线下载。对于此类应用，在执行当前操作前，建议先启动一次应用，待资源文件和补丁包下载完成后，再执行当前操作。后续以共享应用的方式安装到云手机后，该应用的启动过程可省去资源文件下载的过程。

2.2.3 推送应用 tar 包至服务器共享空间

调用接口将OBS桶内的应用tar包推送至云手机服务器共享空间。

调用示例

```
POST https://${CPH_Endpoint}/v1/${projectId}/cloud-phone/phones/share-apps
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "package_name": "${package_name}",
  "bucket_name": "${bucket_name}",
  "object_path": "apk/${package_name}_${version_name}.tar",
  "pre_install_app": 0,
  "server_ids": [
    "${server_id1}",
    "${server_id2}"
  ]
}
```

其中，

- \${bucket_name}为对象存储服务OBS的桶名，object_path为应用tar包在OBS桶中的存放路径。
- pre_install_app 为1表示将待推送的应用设置为预装应用，0表示该应用为非预装。（预装应用的具体说明请参考[云手机预装应用](#)）。如果应用被设置为预装应用，当云手机重置后，应用会自动安装到云手机中。
- server_ids为准备接受应用tar包推送的服务器ID列表。一次可指定多个服务器ID。
- 该接口的更多说明，请参考[推送共享应用](#)。

须知

1. 同一款应用可多次推送，后推送的版本作为该应用的最新版本，多个版本可同时存在于共享空间中。
2. package_name为应用的真实包名，不能被修改。如果是渠道包，可能会存在渠道包后缀，可以在[获取云手机列表](#)中已安装该应用的云手机中查看真实包名。

例如：

官网下载包名：com.huawei.xxxx

其他渠道下载包名：com.huawei.xxxx.huawei

2.2.4 从服务器共享空间删除应用

当云手机服务器上的所有云手机均不再使用某一共享应用时，需要从云手机服务器的共享空间中删除该共享应用，以释放存储空间。

约束与限制

云手机服务器上所有云手机均已卸载该共享应用。

调用示例

```
DELETE "https://${CPH Endpoint}/v1/${projectId}/cloud-phone/phones/share-apps"
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "package_name": "com.huawei.xxxx",
  "server_ids": ["1678567b8bab40f937112xxxxx","1234567b8bab40ffb711xxxxx"]
}
```

其中，package_name 为应用的真实包名，server_ids为服务器ID列表，可以指定多个服务器ID。

详细步骤请参考[删除共享应用](#)。

须知

1. 在删除共享应用前，请确认服务器上所有云手机均已卸载该共享应用。如果没有卸载，会导致共享空间中的应用无法删除，调用删除接口失败时会返回未卸载该应用的云手机列表信息。
2. 如果共享空间中存在某款共享应用的多个版本，并且旧版本没有被任何一台云手机安装，该旧版本会被定期自动清理。

2.3 共享应用使用场景

共享应用tar包推送到共享空间后，可通过共享应用命令实现共享应用的快速安装、更新、卸载。

2.3.1 云手机按需安装

2.3.1.1 应用安装

业务场景示例

- 云游戏场景：用户接入后直接进入某款游戏。
- 个人云手机场景：用户接入后先看到云手机画面，按需选择安装共享空间中已有的应用。

操作步骤

在云手机中执行appctrl命令（参考[appctrl命令](#)）。

1. 使用appctrl install命令安装应用
使用指导：appctrl install {包名}
调用示例：appctrl install com.huawei.xxxx
2. 使用appctrl start命令安装并启动应用
使用指导：appctrl start {包名}
调用示例：appctrl start com.huawei.xxxx

须知

当共享空间某个应用有多个版本时，appctrl install 和appctrl start会自动选择该应用最新版本进行安装。

2.3.1.2 应用更新

业务场景示例

- 当某款应用发布新版本，需对共享应用版本进行更新。
- 当某款游戏发布了新的在线资源更新，需对共享应用中的资源提前进行更新，避免云游戏场景下用户等待在线资源的下载和安装。

操作步骤

1. 参考[获取云手机列表](#)、[为单台云手机安装应用](#)在单台手机上安装应用最新的版本。
2. 如果应用存在资源更新，请启动应用并完成资源文件的下载和更新。
3. 参考[生成应用版本tar包并推至OBS桶](#)和[推送应用tar包至服务器共享空间](#)生成并推送tar包到服务器共享空间。
4. 执行appctrl命令进行更新（参考[appctrl命令](#)）。
 - 执行appctrl start 命令，云手机会安装该应用共享空间中最新版本并启动。
 - 执行appctrl install命令，云手机会安装该应用共享空间中最新版本。

说明

1. 当共享空间中某应用没有更新版本时，appctrl start和 appctrl install不会对已安装版本进行重复安装。
2. 对于采用共享方式安装的应用，如果启动应用后采用在线更新的方式更新资源，该更新后的资源会占用云手机的存储空间。因此当应用存在必要的在线更新时，建议及时对共享空间中的应用版本和在线资源进行更新。

2.3.1.3 应用卸载

操作步骤

在云手机中执行appctrl命令，对已安装在云手机中的共享应用进行卸载（参考[appctrl命令](#)）。

使用指导：在云手机中执行appctrl uninstall {包名}

例如：卸载应用

```
appctrl uninstall com.huawei.xxxx
```

须知

appctrl uninstall命令，仅适用于共享应用，通过原生接口安装的应用无法使用该命令卸载。

2.3.1.4 应用批量卸载

业务场景示例

客户分配的云手机时间到期，需要回收进入待分配区，并进行共享应用批量卸载，卸载上一个玩家安装的共享应用。

操作步骤

在云手机中执行appctrl命令，对使用appctrl命令安装的所有共享应用进行卸载（参考[appctrl命令](#)）。

使用指导：在云手机中执行appctrl clear

须知

appctrl clear命令，仅适用于使用appctrl install和appctrl start命令安装的所有共享应用，原生接口安装的应用和预装的共享应用无法使用该命令卸载。

2.3.2 云手机预装应用

2.3.2.1 预装应用安装

业务场景示例

服务器上所有云手机都要安装某些应用，并且云手机重置后默认安装这些应用。

操作步骤

1. 参考[推送应用tar包至服务器共享空间](#)，推送需要预装的应用tar包到服务器共享空间，将pre_install_app设置为1。
2. 重置云手机，所有预装应用会自动安装。

须知

- 预装应用的安装过程会占用一定的手机重置时间。
- 若云手机正在使用中，无法重置进行重置，也可使用appctrl install命令直接安装应用。
- 取消应用的预装，参考[推送应用tar包至服务器共享空间](#)，再次推送该应用的tar包并将pre_install_app参数设置为0；再次重置云手机时，该应用不会自动安装到云手机中。

2.3.2.2 预装应用更新

操作步骤

1. 参考[为单台云手机安装应用](#)，安装新的应用版本。
2. 启动应用更新下载资源。
3. 参考[生成应用版本tar包并推至OBS桶](#)，生成共享应用tar包并推送到OBS桶中。
4. 参考[推送应用tar包至服务器共享空间](#)，推送需要预装的应用tar包到服务器共享空间，将pre_install_app设置为1。
5. 重启或重置云手机，所有预装应用会自动更新。

须知

预装应用的更新过程会占用一定的手机重启和重置时间。

2.3.2.3 预装应用卸载

业务场景示例一

单台或部分手机不再使用预装的应用。

操作步骤

1. 参考[应用卸载](#)，卸载手机上的对应应用。

须知

卸载应用后，重置手机应用会重新自动安装。

业务场景示例二

服务器上所有的手机都不再使用预装的应用。

操作步骤

1. 参考[应用卸载](#)，卸载服务器上所有手机的对应应用。
2. 参考[从服务器共享空间删除应用](#)，从服务器共享应用区域删除应用。

2.3.3 云手机部署文件

除了共享应用的使用场景外，部分用户业务上需要在云手机中部署文件，部署前需要将这类文件打成tar包，下文中统称这种tar包为配置文件包。

2.3.3.1 文件预置

业务场景示例

需要在服务器上的所有云手机中预置文件。例如：保活脚本文件、应用行为监控文件等。

操作步骤

1. 参考[获取云手机列表](#)，从列表选定一台云手机，将需要打包的文件放入到该云手机的目标路径下。
例如：服务器上云手机需要在/data/local/huawei/、/data/local/tmp/路径下分别部署test.txt和test.sh文件。将test.txt、test.sh放入选定手机的/data/local/huawei/、/data/local/tmp/目录下。
2. 将文件打包成配置文件tar包并上传到OBS桶。

调用示例：

```
POST https://${CPH_Endpoint}/v1/${project_id}/cloud-phone/phones/batch-storage
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "storage_infos": [{
    "phone_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "include_files": [
      "/data/local/huawei/test.txt",
      "/data/local/tmp/test.sh"
    ],
    "bucket_name": "${bucket_name}",
    "object_path": "apk/${package_name}_${version_name}.tar"
  }]
}
```

其中，

- include_files中的元素为需要打包的文件在云手机中的绝对路径。

- object_path中\${package_name}仅支持com.cph.config、com.cph.config.level1、com.cph.config.level2中的一种，\${version_name}用于识别配置包版本，可自行定义。
3. 参考[推送应用tar包至服务器共享空间](#)，将配置文件tar包推送到目标云手机服务器。

调用示例：

```
POST https://${CPH_Endpoint}/v1/${projectId}/cloud-phone/phones/share-apps
Header:
Content-Type: application/json
X-Auth-Token: ${token}
Body:
{
  "package_name": "${package_name}",
  "bucket_name": "${bucket_name}",
  "object_path": "apk/${package_name}_${version_name}.tar",
  "pre_install_app": 1,
  "server_ids": [
    "${server_id1}",
    "${server_id2}"
  ]
}
```

推送时需将"pre_install_app"参数设置为1。

4. 重置或重启云手机，自动部署配置文件包中的文件到云手机对应位置。

须知

1. 配置文件包部署到云手机会占用云手机存储空间，建议配置文件不要过大。
2. 建议所需文件放置在云手机固定目录，方便文件统一管理。然后将这些文件都打包在com.cph.config包中。
3. 如果当前支持的三个配置文件包均需预置，云手机在重置或重启时，会按照com.cph.config、com.cph.config.level1、com.cph.config.level2的顺序依次部署，如果有相同文件，则会依次覆盖。

2.3.3.2 文件按需部署

业务场景示例

服务器上只有某些云手机需要部署文件，云手机正在运行状态，直接部署文件到云手机。

操作步骤

1. 参考[文件预置](#)的1和2，将配置文件打成配置文件tar包并上传到OBS桶。
2. 参考[推送应用tar包至服务器共享空间](#)，将配置文件tar包推送到目标云手机服务器。
推送时将"package_name"配置为支持的配置文件包包名，将"pre_install_app"配置为0。
3. 参考[appctrl命令](#)，在云手机中调用appctrl applyConfig {配置文件包名}命令。

2.3.3.3 已部署文件更新

业务场景示例

该服务器上云手机需要更新已部署过的文件。

如：/data/local/tmp/test.sh脚本有更新，云手机需要部署最新的脚本。

操作步骤

1. 参考[文件预置](#)的1、2、3，将新配置文件打成tar包并推送到服务器共享空间。
2. 更新文件。
方式一：重启云手机，自动更新文件。
方式二：执行appctl applyConfig命令更新文件（参考[appctl命令](#)）。

须知

1. 如果配置文件和自身业务有关联，如：配置文件与运行应用的版本有相互依赖关系，当应用更新时需要对配置文件进行更新。
2. 由于重置云手机只会部署最后一次推送的配置文件包中的文件，更新配置文件包时，需要将所有期望生效的配置文件全量打包。
例如：已部署的配置文件包中有1.txt、2.txt，当前需要更新2.txt。您需要将1.txt和已更新的2.txt全量打包并推送。如果仅打包更新后的2.txt并推送，重置仅会部署2.txt。

2.3.3.4 取消文件预置

业务场景示例

服务器上的所有云手机重置时均不再需要部署文件。

操作步骤

1. 参考[文件预置](#)的1、2、3，将配置文件tar包推送到目标云手机服务器。
推送时将3中的"pre_install_app"参数设置为0。
2. 重置云手机，该包中文件将不再部署。

2.3.3.5 使用多个配置文件包部署文件

业务场景示例一

所有服务器需要预置公共文件，某些服务器需要预置特殊配置文件，处理不同业务。

例如：1.txt、2.txt为公共配置文件，需要预置到所有服务器的云手机中。同时需要预置特定配置文件a.txt，仅对服务器组A生效；特定配置文件b.txt，仅对服务器组B生效。

此时需将1.txt、2.txt打包，并以com.cph.config为包名推送到所有服务器；

将a.txt单独打包，并以com.cph.config.level1为包名推送到服务器组A的服务器；

将b.txt单独打包，并以com.cph.config.level1为包名推送到服务器组B的服务器；

如果需要更新1.txt、2.txt，则更新该文件后重新打包，并以com.cph.config为包名推送的服务器。

如果需要更新a.txt/b.txt，则更新该文件后重新打包，并以com.cph.config.level1为包名推送到对应服务器。

操作步骤

1. 参考[文件预置](#)的1、2、3，将配置文件tar包推送到目标云手机服务器。
推送时将3中的"pre_install_app"参数设置为1。
2. 参考[文件预置](#)的4完成文件部署到云手机。

业务场景示例二

服务器上的所有云手机已预置公共文件，其中某些云手机需要额外增加特定配置文件。

例如：服务器中所有云手机已预置公共配置文件1.txt、2.txt，某些云手机需要额外部署特定配置文件3.txt、4.txt。

此时需将1.txt、2.txt打包，并以com.cph.config为包名推送到服务器；将特定配置文件3.txt、4.txt打包，并以com.cph.config.level1为包名推送到服务器。

操作步骤：

1. 参考[文件按需部署](#)的1、2，将配置文件tar包推送到目标云手机服务器。
推送com.cpm.config.level1时，需将2中的"pre_install_app"云手机单独配置设置为0。
2. 参考[文件按需部署](#)的3完成文件部署到云手机。

2.4 appctrl 命令

2.4.1 appctrl 命令的执行方法

可通过以下三种方式在云手机中执行appctrl命令：

1. 通过ADB连接云手机执行命令，请参考[一键式ADB连接](#)。
2. 调用API接口在云手机中执行命令，请参考[执行异步adb命令](#)。
3. 云手机中的业务程序或脚本直接调用appctrl命令。

说明

针对AOSP11 v10.0.0及AOSP14 v5.0.0之后的自定义镜像版本，appctrl已实现免root运行。对于其他版本的镜像，请确保业务程序或脚本拥有root权限以保障正常运行。

自AOSP11 v10.0.0起，共享应用支持多用户功能，而该版本之前的旧镜像则不支持此功能。因此，若将系统从新镜像回退旧镜像（重启换镜像方式），由于旧镜像不支持多用户功能，非User_0用户相关数据将会丢失。建议结合业务情况，使用重置换镜像方式进行回退。

2.4.2 appctrl install 安装应用

使用场景

安装应用到云手机。

前置条件

应用的tar包已经推送到云手机服务器。多用户场景，需要先创建多用户。

使用指导

- 单用户模式

语法：

```
appctrl install <package> [-v {共享应用版本号}]
```

参数：

- <package>：应用包名。
- -v：可选，指定应用版本号。

示例：

```
# 安装指定包名的应用（默认版本）
appctrl install com.huawei.xxxx
# 安装指定包名及版本的应用
appctrl install com.huawei.xxxx -v 1.0.0
```

- 多用户模式（适用于AOSP11共享应用环境）

语法：

```
appctrl install [--user <id>] <package> [-v {共享应用版本号}]
```

参数：

- --user <id>：可选，指定用户ID。
- <package>：应用包名。
- -v：可选，指定应用版本号。

示例：

```
# 在指定用户下安装指定包名应用
appctrl install --user 10 com.huawei.xxxx
# 在指定用户下安装指定包名及版本的应用
appctrl install --user 10 com.huawei.xxxx -v 1.0.0
```

2.4.3 appctrl start 安装并启动应用

使用场景

安装应用到云手机，并启动应用。

前置条件

应用的tar包已经推送到云手机服务器。多用户场景，需要创建并启动多用户。

使用指导

- 单用户模式

语法：

```
appctrl start <package>
```

参数：

<package>：应用包名。

示例：

```
# 安装并启动指定包名应用
appctrl start com.huawei.xxxx
```

- 多用户模式（适用于AOSP11共享应用环境）

语法：

```
appctrl start [--user <id>] <package>
```

参数：

- --user <id>：可选，指定用户ID。
- <package>：应用包名。

示例：

```
# 在指定用户下安装并启动指定包名应用
appctrl start --user 10 com.huawei.xxxx
```

须知

对于服务类应用，没有activity启动入口，须先使用appctrl install命令先安装应用，再使用am命令启动前后台服务。

2.4.4 appctrl uninstall 卸载应用

使用场景

云手机中卸载应用。

使用指导

- 单用户模式

语法：

```
appctrl uninstall <package>
```

参数：

<package>：应用包名。

示例：

```
# 卸载指定应用
appctrl uninstall com.huawei.xxxx
```

- 多用户模式（适用于AOSP11共享应用环境）

语法：

```
appctrl uninstall [--user <id>] <package>
```

参数：

- --user <id>：可选，指定用户ID。
- <package>：应用包名。

示例：

```
# 在指定用户下卸载应用
appctrl uninstall --user 10 com.huawei.xxxx
```

2.4.5 appctrl clear 批量卸载应用

使用场景

批量卸载所有非预安装的共享应用。

使用指导

- 单用户模式

语法：

```
appctrl clear
```

说明：批量卸载用户0的非预装共享应用。

- 多用户模式（适用于AOSP11共享应用环境）

语法：

```
appctrl clear [--user <id> | -a]
```

参数：

- --user <id>：可选，指定用户ID。
- -a：可选，卸载所有用户的非预装共享应用。

示例：

```
# 批量卸载用户10的共享应用
```

```
appctrl clear --user 10
```

```
# 批量卸载用户 0 的非预装应用，以及其它用户的共享应用
```

```
appctrl clear -a
```

2.4.6 appctrl applyConfig 部署文件

使用场景

直接部署文件至云手机。

前置条件

- 配置文件tar包已经推送到云手机服务器，并且为固定支持的包名。
- 仅支持包名为com.cph.config、com.cph.config.level1、com.cph.config.level2。

使用指导

调用appctrl applyConfig {配置文件包名}

例如：appctrl applyConfig com.cph.config.level2

3 批量安装应用至云手机

在一台云手机内安装APP后，可以通过调用接口的方式将此APP共享安装至多台云手机，省去重复安装的时间。

 说明

假设这台安装了APP的云手机为种子云手机。

约束与限制

- 该方案仅适用于非qemu规格的云手机，即实例规格名称中不包含“qemu”字样。

图 3-1 实例规格

实例规格	规格名称	vCPUs 内存 Disk	屏幕分辨率	手机开数 ?	EIP/VIP个数 ?
☐	rx1.cp.c60.d10.e0v60	2核 3.5 GB 10 GB	1280x720	60	0/60
☒	rx1.cp.c60.d32.e1v1.qemu	2核 3.0 GB 32 GB	960x540	60	1/1
☐	rx1.cp.c30.d16.e3v3	2核 7.0 GB 16 GB	1280x720	30	3/3
☐	rx1.cp.c60.d10.e1v1	2核 3.5 GB 10 GB	1280x720	60	1/1
☐	rx1.cp.c60.d10.e1v1.br60	2核 3.5 GB 10 GB	1280x720	60	1/1
当前服务器规格 64核 256 GB physical.rx1.xlarge					
当前实例规格 2核 3.0 GB 32 GB rx1.cp.c60.d32.e1v1.qemu 960x540					

- 存储1.0和存储2.0服务器可通过控制台的服务器详情页面查看区分。

图 3-2 服务器详情

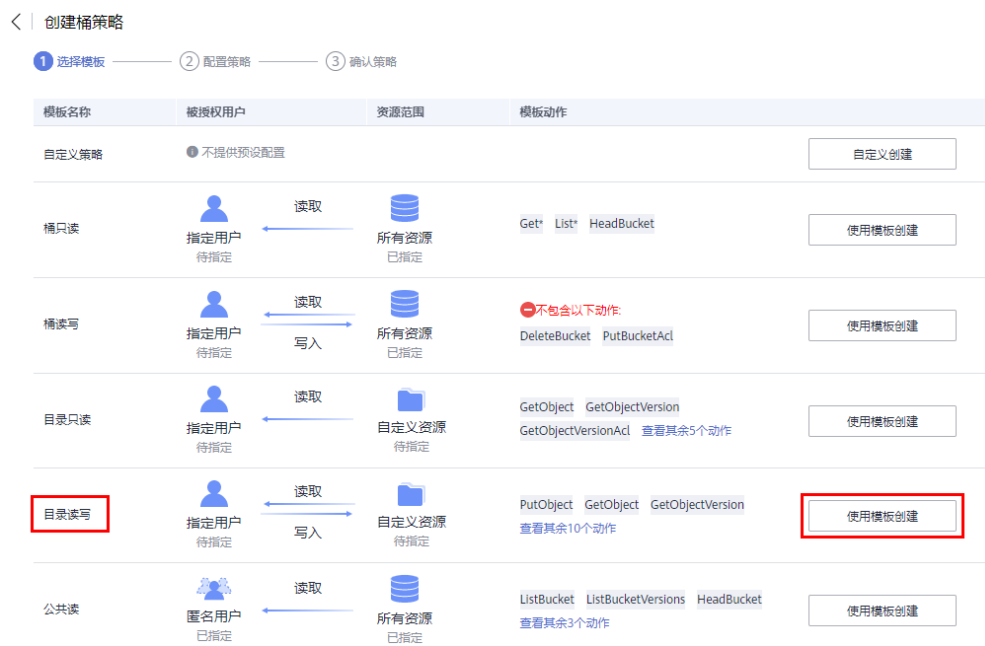
服务器名称	拉曼-圣地亚哥	服务器ID	可用区2
区域	rc3-ve-pro 8 vCPUs 16 GB 1920x1080	可用区	KeyPair-2012
实例规格	60/60	密钥对	共享存储
手机开数	弹性公网IP (私有)	共享存储	GPSSD 1000GB 存储2.0
IP地址	弹性公网IP (私有)	EIP/VIP个数	1/1
IPv6地址	—	网络类型	自定义网络
虚拟私有云	vpc-4406-junbin	子网网络	subnet-4414-junbin
带宽大小	300 Mbits	带宽名称	—
安全组	system-cgr-sg	企业项目	default
订单号	—	到期时间	—

- 种子云手机必须为一台未进行过任何操作的云手机。如果云手机上存在历史操作，请[重置云手机](#)。

操作步骤

- 登录[OBS控制台](#)，参考“[批量控制](#)”创建一个用来存放文件的OBS桶，并为桶设置访问权限。
注意：为了成功上传文件到桶，在创建桶策略时请设置为“目录读写”模式，如图3-3所示。

图 3-3 创建桶策略



- 使用ADB方式连接种子云手机，并为其安装需要的APP。
详细指导请参考“[如何在单台云手机中安装APP?](#)”。
- 通过调用接口的方式，导出种子云手机数据并打包上传至步骤1中的OBS桶。
curl命令如下：

```
curl -i -k -X POST "https://${CPH Endpoint}/v1/${projectId}/cloud-phone/phones/batch-storage" -H "Content-Type: application/json" -H "X-Auth-Token: $token" -d '{
  "storage_infos": [
    {
      "phone_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "include_files": [
        "/data/app/${package-name}-1",
        "/data/data/${package-name}",
        "/data/media/0/Android/data/${package-name}"
      ],
      "bucket_name": "${bucket_name}",
      "object_path": "${your_dir}/${package-name}.tar"
    }
  ]
}'
```

- 其中，
- phone_id为种子云手机的ID。
 - bucket_name为导出数据所存储的OBS桶名。

- object_path为导出数据所存储的OBS路径。

注意

如果想全量打包云手机APP数据，必须要包含以下3个路径：

- /data/app/\${package-name}-1
其中\${package-name}后面可能接的不是-1，需要根据云手机中的实际情况填写。
- /data/data/\${package-name}
- /data/media/0/Android/data/\${package-name}

示例：

```
curl -i -k -X POST "https://${CPH_Endpoint}/v1/${projectId}/cloud-phone/phones/batch-storage" -H
"Content-Type: application/json" -H "X-Auth-Token: $token" -d '
{
  "storage_infos": [
    {
      "phone_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
      "include_files": [
        "/data/app/com.taptap-1",
        "/data/data/com.taptap",
        "/data/media/0/Android/data/com.taptap"
      ],
      "bucket_name": "your-bucket-name",
      "object_path": "your/dir/taptap.tar"
    }
  ]
}'
```

4. 进入步骤1中的OBS桶，在“对象”中查看并确认步骤3中打包上传的文件是否已全部上传成功。确认成功后执行下一步。
5. 此步骤需区分“存储1.0”和“存储2.0”服务器。

- 存储2.0服务器（推荐）

通过调用接口的方式，将OBS桶内的文件推送至服务器。

curl命令示例如下：

```
curl -i -k -X POST "https://${CPH_Endpoint}/v1/${projectId}/cloud-phone/phones/share-apps" -H
"Content-Type: application/json" -H "X-Auth-Token: $token" -d '
{
  "package_name": "com.miniteck.miniworld",
  "pre_install_app": 0,
  "bucket_name": "your-bucket-name",
  "object_path": "your/dir/miniworld.tar",
  "server_ids": ["1678567b8bab40f93711234cb8","1234567b8bab40ffb711234cb"]
}'
```

其中，

- bucket_name、object_path分别对应步骤3中的内容。
- pre_install_app为是否预装应用，非必选参数，默认不预装。0代表不预装，1代表预装。非预装应用需要手动安装，预装应用重置云手机会自动安装。
- server_ids为接受文件推送的服务器ID列表。指定多个服务器ID，可以实现多台服务器上的云手机均安装APP的诉求。

说明

该接口的更多说明，请参考“[推送共享应用](#)”。

- 存储1.0服务器

通过调用接口的方式，将OBS桶内的文件推送至服务器的共享存储目录。

curl命令示例如下：

```
curl -i -k -X POST "https://${CPH Endpoint}/v1/${projectId}/cloud-phone/phones/share-files" -H "Content-Type: application/json" -H "X-Auth-Token: $token" -d '{
  "bucket_name": "your-bucket-name",
  "object_path": "your/dir/taptap.tar",
  "server_ids": ["1678567b8bab40f93711234cb8","1234567b8bab40ffb711234cb"]
}'
```

其中，

- bucket_name、object_path分别对应步骤3中的内容。
- server_ids为接受文件推送的服务器ID列表。指定多个服务器ID，可以实现多台服务器上的云手机均安装APP的诉求。

说明

该接口的更多说明，请参考“[推送共享存储文件](#)”。

6. 此步骤需区分“存储1.0”和“存储2.0”服务器

- 存储2.0的服务器（推荐）

应用推送时为预装应用，批量重置对应的云手机，应用即可安装。非预装应用，在云手机中执行安装应用命令：appctrl install package_name，其中的包名对应[第五步](#)中的package_name，例如：com.miniteck.miniworld。

注意

应用推送时为预装应用，应用卸载后，重置云手机应用会再次安装。

- 存储1.0的服务器

进入云手机控制台，单击已接受文件推送的服务器名称，在“实例管理”中，勾选所有需要安装以上APP的云手机实例，单击“重置”。

注意

此步骤为重置，并非重启。

执行结果

存储1.0的机器在重置手机成功后均已安装APP；存储2.0的机器无需重置，后续需要时直接调用 appctrl start package_name来启动应用。

4 修改云手机定位

功能介绍

影响手机定位功能的信息分为四类，分别为GPS经纬度数据、基站信息数据、WiFi信息及IP地址，云手机支持其中的GPS经纬度数据和WiFi信息模拟，用户可根据业务场景，选择其中一种或多种进行定位模拟。

定位模式	精度	常见场景
GPS	高	地图、导航等需要持续定位的应用软件
WiFi	中	一般用于GPS和基站的辅助定位手段

前提条件

已购买并通过ADB方式登录云手机。详细操作请参见“[购买云手机服务器](#)”。

GPS 经纬度数据模拟

1. GPS经纬度获取
- 云手机GPS经纬度数据模拟基于WGS84坐标系，使用其它坐标系如GCJ02(高德、QQ地图)，BD09(百度地图) 注入会产生偏差。
- 获取GPS定位信息示例：
- ```
mLocationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
mLocationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 0,
locationListener);
private final LocationListener locationListener = new LocationListener() {
 public void onLocationChanged(Location location) {
 // location包含gps数据经纬度等信息
 }
};
```
2. GPS数据注入方式
- shell命令注入

在手机adb shell环境里执行echo "参数列表（参数之间用“:”分隔）"> /data/gps/fifo。

参数说明如[表4-1](#)所示。

表 4-1 GPS 数据注入参数

| 参数        | 说明                                                | 是否必须 | 缺省值        | 约束                                     |
|-----------|---------------------------------------------------|------|------------|----------------------------------------|
| latitude  | 纬度。<br>北纬为正，南纬为负。                                 | 必须   | 22.657501  | 取值范围：-90.000000~90.000000<br>单位：度（°）   |
| longitude | 经度。<br>东经为正，西经为负。                                 | 必须   | 114.055939 | 取值范围：-180.000000~180.000000<br>单位：度（°） |
| altitude  | 海拔高度                                              | 可选   | 51.0       | 单位：米                                   |
| speed     | 速度                                                | 可选   | 0.0        | 单位：米                                   |
| bearing   | 方向角。<br>0°表示正北方向，90°表示正东方向，180°表示正南方向，270°表示正西方向。 | 可选   | 30.0       | 取值范围：0.0~360.0<br>单位：度（°）              |
| accuracy  | 定位精度                                              | 可选   | 90.0       | 单位：米                                   |

举例：假设需修改的位置是东经114.055939度，北纬22.657501度。

```
adb -s 127.0.0.1:本地空闲端口 shell "echo 'longitude=114.055939:latitude=22.657501' > /data/gps/fifo"
```

- GPS数据持续注入（建议使用非阻塞注入方式）

使用自有或第三方SDK应用代码持续注入GPS数据，需使用O\_NONBLOCK非阻塞方式注入。

举例：

```
#define GPSFifoName "/data/gps/fifo"
if((fifo_fd = open(GPSFifoName, O_WRONLY | O_NONBLOCK)) < 0) {
 ALOGE("open fifo \"%s\" (write) fail, error = %s\n", GPSFifoName, strerror(errno));
 return;
}
//注意cmd参数间需要通过冒号“:”进行分隔
char* cmd = "longitude=113.370592:latitude=23.123642";
len = strlen(cmd);
if(write(fifo_fd, cmd, len) != len) {
 ALOGE("%s: write \"%s\" to \"%s\" fail: %s", FUNCTION, cmd, GPSFifoName, strerror(errno));
}
```

WiFi 信息模拟

1. WiFi信息获取

获取WiFi信息示例：

```
WifiManager wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
WifiInfo wifiConnection = wifiManager.getConnectionInfo();
```

```
if (wifiConnection != null) {
 String bssid = wifiConnection.getBSSID();
}
```

## 2. WiFi信息注入

针对当前连接WiFi BSSID，可以通过设置属性com.cph.wifi.bssid进行注入。

举例：

```
setprop com.cph.wifi.bssid 02:00:00:00:00:00
```

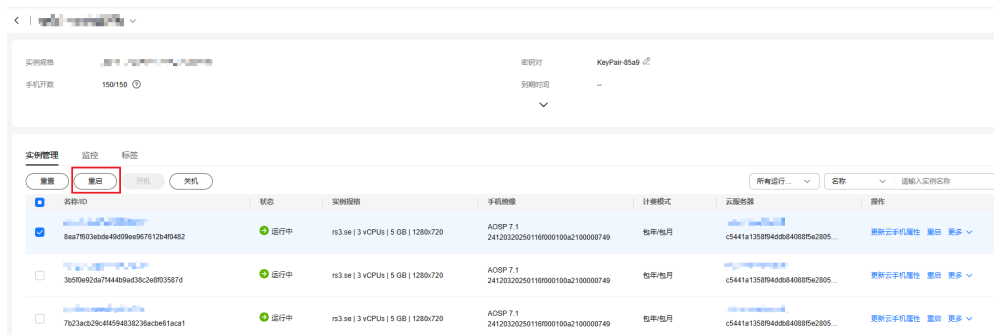
# 5 使用云手机摄像头

## 步骤 1：替换手机镜像

1. 查看云手机镜像[最新动态](#)，选择一个2020年10月9日（包含）之后的镜像，复制镜像ID。
2. 登录[CPH控制台](#)，切换到您的资源所在region。
3. 单击服务器管理，查看云手机服务器列表。



4. 进入云手机服务器详情页，选择其中一台云手机，单击“重启”按钮。



5. 勾选更新手机镜像复选框，然后填入[第一步](#)所选的镜像ID。



6. 单击“确定”，完成单台云手机更新镜像操作。

## 步骤 2：上传图片到手机

上传一张图片到云手机的“/data/local/tmp/”目录下。您有两种方式上传图片，此处以“/path/to/local”目录下的“pic.jpeg”为例。

- **方式一：通过adb push命令推送图片**  
首先通过adb连接到云手机，然后执行以下命令：  
**adb push </path/to/local/pic.jpeg> /data/local/tmp/pic.jpeg**  
**adb shell chmod 644 /data/local/tmp/pic.jpeg**
- **方式二：调用云手机api接口推送图片**  
请参考[ADB命令推送文件](#)来推送图片文件。

### 须知

- 上传的图片尺寸保证比例为 480(宽) \* 640(高)，比例不为480\*640时，图片在摄像头中可能会被缩放。
- 图片只支持jpeg和png格式，请不要用其他格式的图片。并且路径一定要是“/data/local/tmp/”目录下。
- 图片权限至少为644(rw-r--r--)。

## 步骤 3：设置手机属性

您有两种方式设置手机属性。

**方式一：adb连接到云手机，然后执行adb命令**

**adb shell setprop com.cph.cam\_local\_pic\_path /data/local/tmp/pic.jpeg**

使用此这种方式，重启手机后属性失效。

**方式二：调用云手机api接口设置属性**

参考[更新云手机属性](#)来设置，将” com.cph.cam\_local\_pic\_path ” :” /data/local/tmp/pic.jpeg ” 属性设置到手机中。属性将被持久化，重启手机属性依然保留。

## 步骤 4：测试摄像头

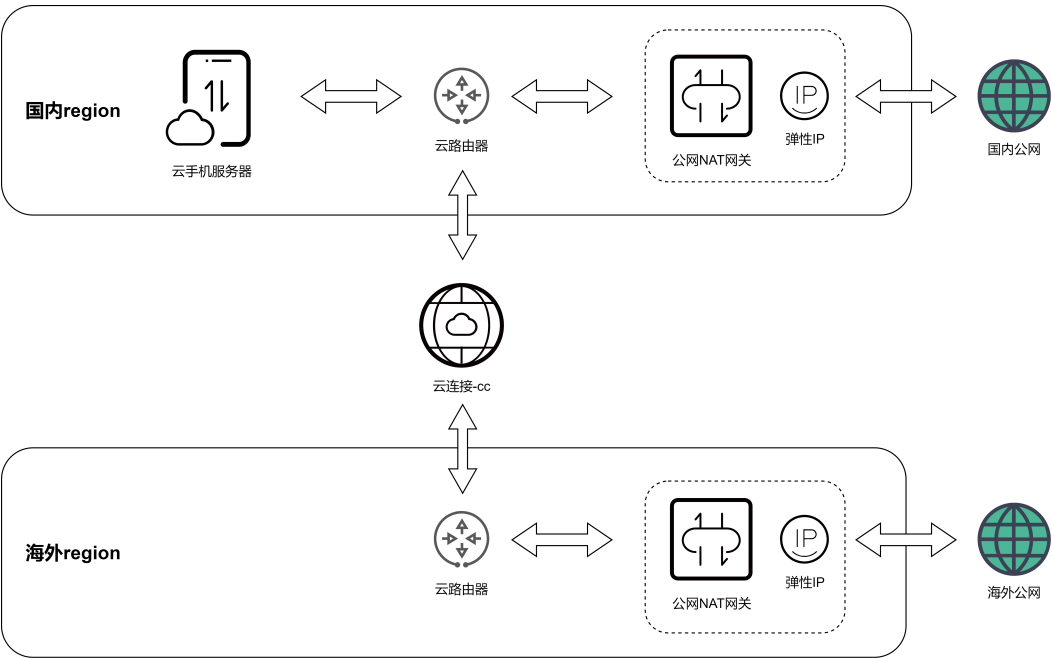
安装任意一款需要调用摄像头的APP，打开APP查看取景框是否成功显示您设置的图片。

### 说明

当前云手机只支持后置摄像头。

# 6 国内云手机服务器导流海外

导流方案整体结构如图：



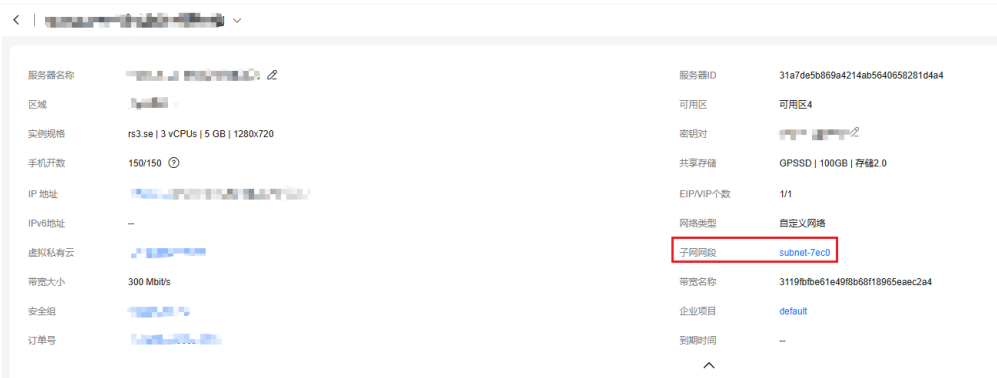
## 约束与限制

- 该方案仅适用于没有EIP的云手机服务器，即实例规格中EIP个数为0的规格（VIP个数不限）。

| 规格名称                                                  | vCPUs   运行内存   机身存储    | 屏幕分辨率    | 手机开数 ? | EIP/VIP个数 ? |
|-------------------------------------------------------|------------------------|----------|--------|-------------|
| <input type="radio"/> rx1.cp.c15.d46.e1v1             | 4 vCPUs   16GB   46GB  | 1280x720 | 15     | 1/1         |
| <input checked="" type="radio"/> rx1.cp.c60.d10.e0v60 | 2 vCPUs   3.5GB   10GB | 1280x720 | 60     | 0/60        |

## 操作步骤

- 参考云连接服务的“[跨境申请管理](#)”章节申请跨境专线资质。待审批通过后，执行后续步骤。
- 登录[CPH控制台](#)，进入云手机服务器管理，单击需要导流的云手机服务器，进入详情页面，找到“子网网段”。



3. 单击子网名称，进入子网页面，找到“子网IPv4网段”，记录下云手机服务器所在的子网地址，例如192.168.0.0/24。



4. 选择“网络 > NAT网关 NAT”。
5. 切换到想要出海外公网的Region，例如“中国-香港”。
6. 参考NAT网关“[通过公网NAT网关的SNAT规则访问公网](#)”，购买EIP、公网NAT网关，并配置好SNAT规则、路由表增加目的地址0.0.0.0/0到NAT网关。  
其中配置SNAT规则时，使用场景选择“云专线/云连接”，并输入第3步记录的云手机服务器子网网段。

## 添加SNAT规则

- 当云主机同时配置弹性公网IP服务和NAT网关服务时，数据均通过弹性公网IP转发。 [参考链接](#)
- SNAT规则和DNAT规则一般面向不同的业务，如果使用相同的EIP，会面临业务相互抢占问题，请尽量避免。
- SNAT规则不能和全端口的DNAT规则共用EIP。

|         |      |
|---------|------|
| NAT网关名称 | test |
|---------|------|

### \* 使用场景

虚拟私有云

云专线/云连接

|     |   |     |   |   |   |   |   |    |   |
|-----|---|-----|---|---|---|---|---|----|---|
| 192 | . | 168 | . | 0 | . | 0 | / | 24 | Ⓢ |
|-----|---|-----|---|---|---|---|---|----|---|

- 选择“网络 > 云连接CC”。
- 参考云连接CC“[跨区域同账号VPC互通](#)”，创建云连接实例，并配置加载云手机所在子网和第6步购买的NAT网关所在子网。购买“跨大区互通”带宽包，例如“中国大陆”到“亚太”，并绑定到云连接实例。示例如下图：

[基本信息](#)
[网络实例](#)
[带宽包](#)
[域间带宽](#)
[路由信息](#)
[标签](#)

## 加载网络实例

中国-香港

 vpc-4575

+

华东-上海-

vpc--test

+

[←](#) | [购买带宽包](#) [?](#)

### 计费模式

包年/包月

计费模式选定后无法修改。

\* 名称

bandwidthPackge-a09e

### 计费方式

### 按带宽计费

### 互通类型

## 大区内互通

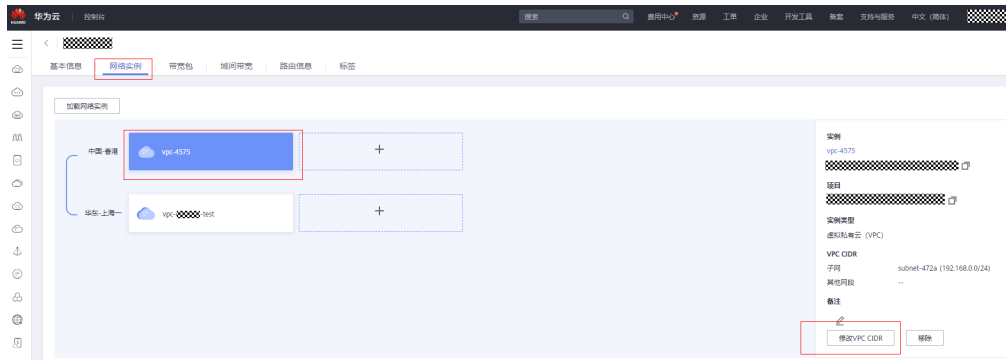
## 跨大区互通

互通大区

中国大陆

亚太

- 在云连接“网络实例”页面选中已加载的海外vpc（示例中为“中国-香港”），然后单击“修改VPC CIDR”。



在弹出的对话框中选择“高级配置”，输入0.0.0.0/0，单击“添加”按钮，然后确定。

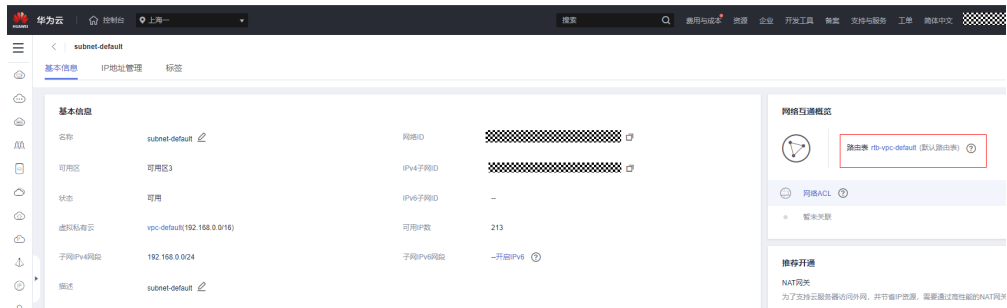


以上步骤完成后，国内云手机服务器的流量就已经全部导流到云连接，使用海外EIP出口。可以在云手机服务器内测试公网出口IP，验证配置是否成功。

如果不需要国内流量分流，以下步骤无需执行。

配置云手机服务器国内流量分流（可选）

- 参考以上“操作步骤”第6步，在国内云手机所在Region，购买EIP、公网NAT网关，并配置好SNAT规则。此处有一个差异，不需要配置路由表目的地址0.0.0.0/0到NAT网关。
- 参考以上“操作步骤”第2、3步，进入云手机服务器详情页面，找到“路由表”。



- 单击路由表名称，进入路由表页面，单击“添加路由”。



4. 在弹出的对话框中，目的地址输入需要分流到国内的IP地址或网段，下一跳类型选择“NAT网关”，下一跳选择第1步购买的公网NAT网关，然后确定。

添加路由



5. 如果有其它IP地址或网段需要分流，重复第4步添加。
- 以上步骤完成后，从云手机服务器内访问配置了分流的IP地址时，流量会从国内的EIP出口，其它流量则会导流到云连接从海外EIP出口。

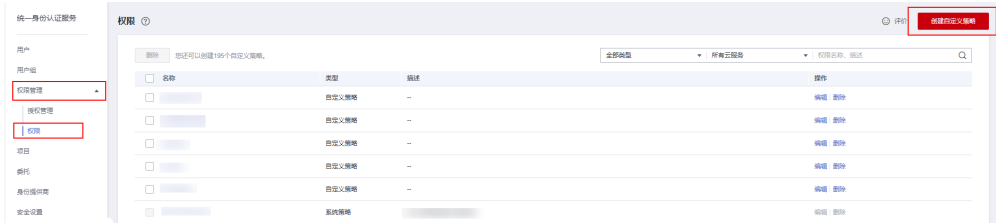
# 7 委托 CPH 操作 OBS 桶

管理员可以通过IAM创建自定义策略，给OBS桶授予自定义策略来进行精细的访问控制，并将此策略委托给CPH服务，以完成云手机数据备份恢复、应用安装等功能。

## 操作步骤

- 创建自定义策略
- 1. 登录IAM控制台。
- 2. 在左侧导航树中，选择“权限管理 > 权限”，单击右上方的“创建自定义策略”。

图 7-1 创建自定义策略



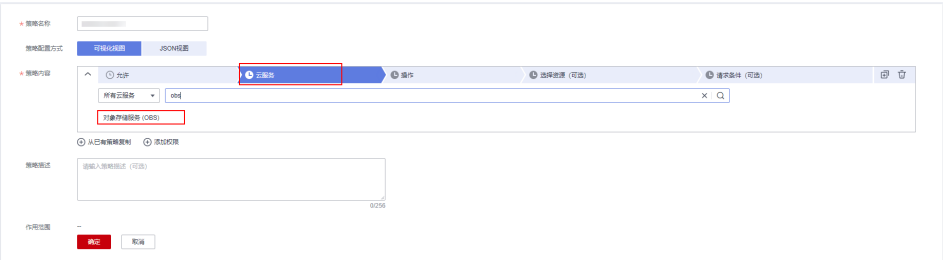
- 3. 输入“策略名称”。
- “策略配置方式”选择“可视化视图”。

图 7-2 可视化视图



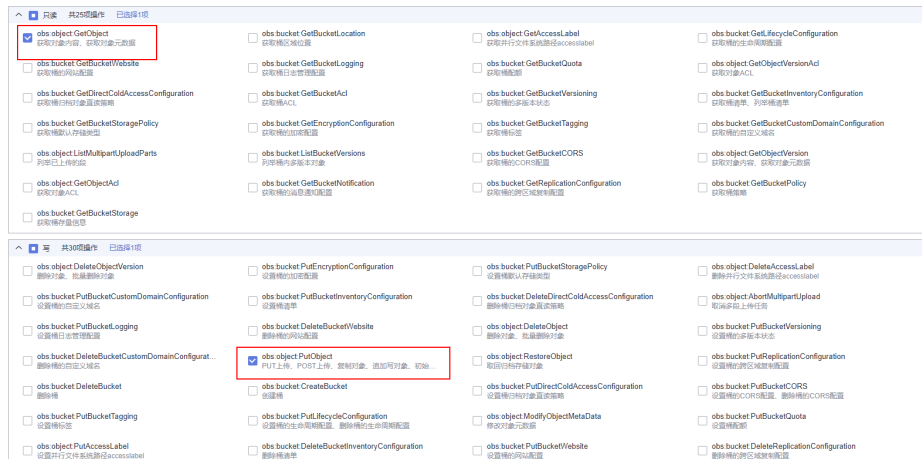
- 4. 在“策略内容”下配置策略。
- a. 选择“允许”。
- b. 单击“云服务”，选择“对象存储服务 OBS”。

图 7-3 配置策略



- c. 选择“操作”，根据需求勾选授权项。关于授权项请参考[对象相关授权项](#)。委托CPH操作OBS桶，至少指定以下两项操作：  
**"obs:object:GetObject",**  
**"obs:object:PutObject",**

图 7-4 勾选授权项



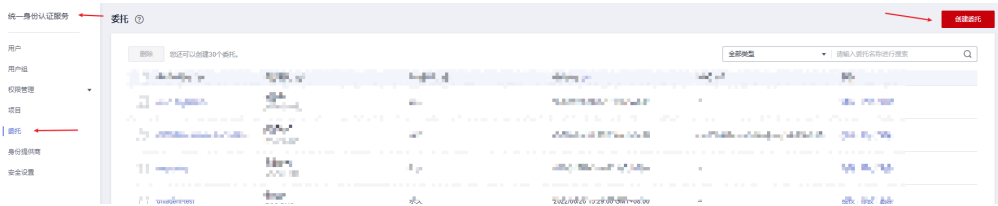
- d. 选择“所有资源”。如需选择“特定资源”，请参考[创建自定义策略](#)章节中关于“特定资源”的说明。

图 7-5 选择资源



5. 单击“确定”，完成创建。
- 创建委托
  - 1. 登录[IAM控制台](#)。
  - 2. 在左侧导航树中，选择“委托”，单击“创建委托”。

图 7-6 委托



3. 参照下图填写参数内容，单击“下一步”。

图 7-7 创建委托

委托 / 创建委托

\* 委托名称

cph\_obs\_agency

\* 委托类型

☐ 普通帐号

将帐号内资源的操作权限委托给其他华为云帐号。

☒ 云服务

将帐号内资源的操作权限委托给华为云服务。

\* 云服务

云手机 CPH

\* 持续时间

永久

描述

请输入委托信息。

0/255

下一步

取消



委托名称必须填写“cph\_obs\_agency”。

4. 选中步骤[创建自定义策略](#)中创建的自定义策略，单击“下一步”。

图 7-8 选择策略



5. 选择“全局服务资源”，单击“确定”委托创建完成。



# 8 云手机 AOSP 版本切换

本文将指导您如何切换云手机AOSP镜像版本。

## 前提条件

仅手机镜像为AOSP7的服务器支持切换AOSP镜像版本。

## 注意事项

1. 版本切换是高风险操作，切换前请参考[导出云手机数据](#)备份用户数据。如需回退，重置原有镜像后参考[恢复云手机数据](#)恢复数据即可。
2. 版本切换时，Android版本号相关的属性值（如ro.build.version.release属性和ro.build.fingerprint中Android版本字段）会自动变更为目标镜像对应的版本号。

## 升级 AOSP 版本

### 方法一（保留用户数据）

云手机重启接口支持切换AOSP镜像版本，同时保留用户数据，操作详情请参考[重启云手机实例](#)。

注意：

- 重启接口支持将低版本镜像升级成高版本，但不支持将高版本镜像降级为低版本。
- 如果您没有保留用户数据的强烈诉求，推荐您使用方法二切换AOSP版本，应用不兼容的风险更小。

### 方法二（不保留用户数据）

云手机重置接口支持切换AOSP镜像版本，操作详情请参考[重置云手机实例](#)。

注意：

- 重置接口支持将低版本镜像升级成高版本，也支持将高版本镜像回退为低版本。

## 回退 AOSP 版本

因为应用兼容原因，只支持低版本切换高版本时保留用户数据。回退低版本只能使用不保留用户数据的重置接口，操作详情请参考[重置云手机实例](#)。

### 说明

若您通过重置云手机实例的方式更换低版本镜像，某些版本adb客户端需要先执行断开命令：  
`adb disconnect ip:port`，再重新执行连接命令：`adb connect ip:port`，才能看到手机画面。