

# 阿里云 云服务器 ECS 镜像

文档版本：20190223

# 法律声明

---

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，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。
3. 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。阿里云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，并在阿里云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阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

## 通用约定

格式	说明	样例
	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。	 <b>禁止：</b> 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 <b>警告：</b> 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 <b>说明：</b> 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	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。	单击 <b>确定</b> 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
##	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
[ ]或者[a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ }或者{a b}	表示必选项，至多选择一个。	<code>swich {stand   slave}</code>

# 目录

---

法律声明.....	I
通用约定.....	I
1 公共镜像.....	1
1.1 镜像发布记录.....	1
1.2 Aliyun Linux 17.01 特性说明.....	4
1.3 已知问题.....	5
2 镜像发布记录.....	6
2.1 创建自定义镜像.....	6
2.1.1 使用实例创建自定义镜像.....	6
2.1.2 使用Packer创建自定义镜像.....	7
2.1.3 使用 Packer 创建并导入本地镜像.....	9
2.2 修改自定义镜像信息.....	15
2.3 导入镜像.....	15
2.3.1 镜像规范检测工具.....	15
2.3.2 导入镜像必读.....	19
2.3.3 安装 cloud-init.....	22
2.3.4 安装virtio驱动.....	24
2.3.5 定制 Linux 自定义镜像.....	28
2.3.6 转换镜像格式.....	33
2.3.7 导入自定义镜像.....	34
2.4 复制镜像.....	37
2.5 共享镜像.....	38
2.6 导出镜像.....	39
3 云市场镜像.....	42
4 镜像相关开源工具.....	44
5 公共镜像.....	45
6 自定义镜像.....	46

# 1 公共镜像

## 1.1 镜像发布记录

本文介绍镜像的特性与版本的发布记录。

2019年2月18日

镜像版本	版本说明
CentOS 7.6	<ul style="list-style-type: none"><li>· 镜像ID: centos_7_06_64_20G_alibase_20190218.vhd</li><li>· 内核版本: 3.10.0-957.5.1.el7.x86_64</li><li>· 发布地域: 华东1, 华北2</li><li>· 更新内容: 更新最新补丁包</li></ul>

2019年1月3日

镜像版本	版本说明
Debian 9.6	<ul style="list-style-type: none"><li>· 镜像ID: debian_9_06_64_20G_alibase_20190103.vhd</li><li>· 内核版本: 4.9.0-8-amd64</li><li>· 发布地域: 所有地域</li><li>· 更新内容: 开启systemd-networkd服务</li></ul>

2018年12月22日

镜像版本	版本说明
Windows Server Version 1809	<ul style="list-style-type: none"><li>· 镜像ID:<ul style="list-style-type: none"><li>- winsvr_64_dtcC_1809_zh-cn_40G_alibase_20181222.vhd (中文版)</li><li>- winsvr_64_dtcC_1809_en-us_40G_alibase_20181222.vhd (英文版)</li></ul></li><li>· 发布地域: 所有地域</li><li>· 更新内容:<ul style="list-style-type: none"><li>- 镜像更新至2018年12月份补丁KB4483235</li><li>- 默认已执行Sysprep封装</li></ul></li></ul>

镜像版本	版本说明
Windows Server 2008 R2	<ul style="list-style-type: none"> <li>· 镜像ID: win2008r2_64_ent_sp1_en-us_40G_alibase_20181222.vhd (英文版)</li> <li>· 发布地域: 所有地域</li> <li>· 更新内容: <ul style="list-style-type: none"> <li>- 镜像更新至2018年12月份安全补丁KB4471318。在使用RDP连接时, Windows客户端需升级到最新补丁</li> <li>- .NET Framework 升级到4.7.2</li> <li>- 内置执行Sysprep封装</li> </ul> </li> </ul>

2018年12月20日

镜像版本	版本说明
Windows Server 2008 R2	<ul style="list-style-type: none"> <li>· 镜像ID: win2008r2_64_ent_sp1_zh-cn_40G_alibase_20181220.vhd (中文版)</li> <li>· 发布地域: 所有地域</li> <li>· 更新内容: <ul style="list-style-type: none"> <li>- 镜像更新至2018年12月份安全补丁KB4471318。在使用RDP连接时, Windows客户端需升级到最新补丁</li> <li>- .NET Framework 升级到4.7.2</li> <li>- 内置执行Sysprep封装</li> </ul> </li> </ul>
Windows Server 2012 R2	<ul style="list-style-type: none"> <li>· 镜像ID: <ul style="list-style-type: none"> <li>- win2012r2_64_dtc_9600_zh-cn_40G_alibase_20181220.vhd (中文版)</li> <li>- win2012r2_64_dtc_9600_en-us_40G_alibase_20181220.vhd (英文版)</li> </ul> </li> <li>· 发布地域: 所有地域</li> <li>· 更新内容: <ul style="list-style-type: none"> <li>- 镜像更新至2018年12月份安全补丁KB4471320。在使用RDP连接时, Windows客户端需升级到最新补丁</li> <li>- .NET Framework 升级到4.7.2</li> <li>- 内置执行Sysprep封装</li> </ul> </li> </ul>

镜像版本	版本说明
Windows Server 2016	<ul style="list-style-type: none"> <li>· 镜像ID:               <ul style="list-style-type: none"> <li>- win2016_64_dtc_1607_zh-cn_40G_alibase_20181220.vhd (中文版)</li> <li>- win2016_64_dtc_1607_en-us_40G_alibase_20181220.vhd (英文版)</li> </ul> </li> <li>· 发布地域: 所有地域</li> <li>· 更新内容:               <ul style="list-style-type: none"> <li>- 镜像更新至2018年12月份安全补丁KB4471321。在使用RDP连接时, Windows客户端需升级到最新补丁</li> <li>- .NET Framework 升级到4.7.2</li> <li>- 内置执行Sysprep封装</li> </ul> </li> </ul>

2018年12月12日

镜像版本	版本说明
CentOS 7.6	<ul style="list-style-type: none"> <li>· 镜像ID: centos_7_05_64_20G_alibase_20181212.vhd</li> <li>· 内核版本: 3.10.0-957.1.3.el7.x86_64</li> <li>· 发布地域: 所有地域</li> <li>· 更新内容: 更新最新系统补丁</li> </ul>
Debian 9.6	<ul style="list-style-type: none"> <li>· 镜像ID: debian_9_06_64_20G_alibase_20181212.vhd</li> <li>· 内核版本: 4.9.0-8-amd64</li> <li>· 发布地域: 所有地域</li> <li>· 更新内容:               <ul style="list-style-type: none"> <li>- 更新最新系统补丁</li> <li>- 更新cloud-init版本</li> <li>- 开启chrony时间同步服务</li> <li>- 设置GRUB_TIMEOUT=1</li> </ul> </li> <li>· 已知问题: <a href="#">经典网络配置问题</a></li> </ul>
Ubuntu 18.04	<ul style="list-style-type: none"> <li>· 镜像ID: ubuntu_18_04_64_20G_alibase_20181212.vhd</li> <li>· 内核版本: 4.15.0-42-generic</li> <li>· 发布地域: 所有地域</li> <li>· 更新内容:               <ul style="list-style-type: none"> <li>- 更新最新系统补丁</li> <li>- 更新cloud-init版本</li> <li>- 开启chrony时间同步服务</li> <li>- 设置GRUB_TIMEOUT=1</li> </ul> </li> </ul>

2018年12月10日

镜像版本	版本说明
CentOS 7.5	<ul style="list-style-type: none"> <li>· 镜像ID: centos_7_05_64_20G_alibase_20181210.vhd</li> <li>· 内核版本: 3.10.0-862.3.3.el7.x86_64</li> <li>· 发布地域: 所有地域</li> <li>· 更新内容: <ul style="list-style-type: none"> <li>- 更新最新系统补丁</li> <li>- 更新cloud-init版本</li> <li>- 开启chrony时间同步服务</li> <li>- 默认禁用密码登录</li> <li>- 设置GRUB_TIMEOUT=1</li> </ul> </li> </ul>

## 1.2 Aliyun Linux 17.01 特性说明

Aliyun Linux 镜像是公共镜像的一种，是阿里云针对 ECS 实例提供的定制化原生操作系统。本文介绍 Aliyun Linux 镜像 17.01（Golden Toad）版本的特性。

- 集成阿里云内核组维护的 4.4 系列 Kernel
  - 支持 swap 内存压缩可保证显著减少 I/O；
  - 支持动态补丁机制 kpatch，升级内核无需重启系统；
  - 支持超大内存 kdump（至多 3 TB）；
  - 支持 Automatic NUMA；
  - 用户态 vsyscall 获取时间戳性能提升；
  - 全面支持 Docker 技术栈；
  - 支持 pv panic，方便协助云用户快速定位内核问题；
  - 重新设计的无锁 TCP listener，提升 2 ~ 3 个数量级防 SYN 攻击能力。
- 高效创建和启动
  - 基于虚拟化级优化栈创建实例，有效降低创建虚拟机所需时间。
  - OS 搭载阿里云定制启动服务，提升启动速率，使创建到可登录时间控制在 30 秒内。
- 文件系统

新增支持文件系统：XFS、Btrfs、AUFS 和 OverlayFS。
- 全新工具链

GCC 4.8.x glibc 2.17 GDB 7.6.1。



- 安全提升
  - OpenSSH 支持 chroot shell 登录，限制用户登录在一个受限的根文件系统中。
  - OpenSSH 可指定多个验证条件以加强登录访问。
- 兼容CentOS 7.2

## 1.3 已知问题

本文介绍了不同发行平台的阿里云镜像的已知故障、故障涉及范围以及解决方法。

### Debian：经典网络配置问题

- 问题描述：因为Debian系统默认禁用了systemd-networkd服务，经典网络类型实例无法通过DHCP（Dynamic Host Configuration Protocol）模式自动分配IP。
- 涉及镜像ID：debian\_9\_06\_64\_20G\_alibase\_20181212.vhd
- 修复方案：您需要运行下列命令来解决此问题。

```
systemctl enable systemd-networkd
systemctl start systemd-networkd
```

## 2 镜像发布记录

---

### 2.1 创建自定义镜像

#### 2.1.1 使用实例创建自定义镜像

您可以基于实例创建一个自定义镜像，把实例中的所有磁盘，包括系统盘和数据盘中的数据，完整地复制到自定义镜像中。

创建自定义镜像的过程中，该实例的每块磁盘都自动创建一个快照，这些快照组合起来构成一个自定义镜像，如下图所示。

您也可以基于快照创建自定义镜像。请参考 [#unique\\_10](#)。

#### 操作须知

- 请将实例中的敏感数据删除之后再创建自定义镜像，避免数据安全隐患。
- 创建镜像的过程中，不要改变实例的状态，不要停止、启动或者重启实例，避免创建失败。
- 使用带数据盘的自定义镜像创建实例时，默认根据镜像创建数据盘；如果创建实例的同时创建了数据盘，则默认根据挂载点情况使用该镜像的数据盘快照创建实例的数据盘。
- 支持导出带数据盘的自定义镜像。
- 更换系统盘时，不能使用带数据盘的自定义镜像。

#### 操作步骤

1. 登录 [ECS 管理控制台](#)。
2. 在实例列表页面顶部，选择目标实例所在的地域。
3. 单击左侧导航栏中的 实例。
4. 找到需要的实例。单击列表最右侧的 更多 > 磁盘和镜像 > 创建自定义镜像。
5. 输入镜像名称和描述信息。
6. 单击 创建。

所有磁盘的快照全部创建结束后，镜像才能使用。请耐心等待。

#### 后续操作

成功创建自定义镜像后，您可能想 [#unique\\_11](#)。

## 2.1.2 使用Packer创建自定义镜像

Packer是一款轻量级的镜像定义工具，能够运行在常用的主流操作系统（如Windows、Linux和macOS）上。参阅本文安装并使用Packer，轻松创建自定义镜像。



说明：

本文描述仅适用于运行Linux操作系统的服务器。

### 1. 安装Packer

您可以进入 [Packer 官网下载页面](#)，选择下载与您操作系统对应的版本，参阅以下步骤或者访问 [Packer 官方安装说明](#) 安装Packer。

1. 连接并登录到Linux服务器。如果您的服务器为ECS Linux实例，可参阅文档 [使用用户名密码验证连接 Linux 实例](#)。
2. 执行命令 `cd /usr/local/bin` 进入 `/usr/local/bin` 目录。



说明：

`/usr/local/bin` 目录为环境变量目录，您可以将Packer安装到该目录下或其他已添加到环境变量的目录下。

3. 执行命令 `wget https://releases.hashicorp.com/packer/1.1.1/packer_1.1.1_linux_amd64.zip` 获取Packer安装包。您可以访问 [Packer 下载页面](#) 获取其他版本Packer安装包。
4. 执行命令 `unzip packer_1.1.1_linux_amd64.zip` 解压文件。
5. 执行命令 `packer -v` 验证Packer安装状态。若Linux服务器返回Packer版本号，表示您已正确安装Packer。若Linux服务器提示 `command not found` 表示Packer未正确安装。

### 2. 定义Packer模板

使用Packer创建自定义镜像时，需要创建一个JSON格式的模板文件。在该模板文件中，您需要指定创建自定义镜像的 [Alicloud Image Builder（生成器）](#) 和 [Provisioners（配置器）](#)。Packer具有多种配置器，可用于配置自定义镜像的内容生成方式，以下以常用的 [Shell](#) 配置器为例，定义Packer模板。

在Linux服务器中创建名为alicloud的json文件并粘贴以下内容。

```
{
  "variables": {
    "access_key": "{{env `ALICLOUD_ACCESS_KEY`}}",
    "secret_key": "{{env `ALICLOUD_SECRET_KEY`}}",
  },
  "builders": [{
    "type": "alicloud-ecs",
```

```

    "access_key": "{{user `access_key`}}",
    "secret_key": "{{user `secret_key`}}",
    "region": "cn-beijing",
    "image_name": "packer_basic",
    "source_image": "centos_7_02_64_20G_alibase_20170818.vhd",
    "ssh_username": "root",
    "instance_type": "ecs.n1.tiny",
    "internet_charge_type": "PayByTraffic",
    "io_optimized": "true"
  }],
  "provisioners": [{
    "type": "shell",
    "inline": [
      "sleep 30",
      "yum install redis.x86_64 -y"
    ]
  }]
}

```

您需要自定义以下参数值。

参数	描述
access_key	您的AccessKeyID。更多详情，参阅 <a href="#">创建AccessKey</a> 。  <div>  <b>说明:</b>            由于AccessKey权限过大，为防止错误操作，建议您 <a href="#">创建 RAM 用户</a>，并使用RAM子账号 <a href="#">创建AccessKey</a>。         </div>
secret_key	您的AccessKeySecret。更多详情，参阅 <a href="#">创建AccessKey</a> 。
region	创建自定义镜像时使用临时资源的地域。
image_name	自定义镜像的名称。
source_image	基础镜像的名称，可以从阿里云公共镜像列表获得。
instance_type	创建自定义镜像时生成的临时实例的类型。
internet_charge_type	创建自定义镜像时临时实例的公网带宽付费类型。
provisioners	创建自定义镜像时使用的 <a href="#">Packer 配置器</a> 类型。

### 3. 使用Packer创建自定义镜像

指定Packer模板文件生成自定义镜像：

1. 导入您的AccessKeyID：运行 `export ALICLOUD_ACCESS_KEY=您的 AccessKeyID`。
2. 导入您的AccessKeySecret：运行 `export ALICLOUD_SECRET_KEY=您的 AccessKeySecret`。
3. 执行命令 `packer build alicloud.json` 创建自定义镜像。

示例运行结果如下，以下示例将创建含Redis的自定义镜像：

```
alicloud-ecs output will be in this color.
==> alicloud-ecs: Prevalidating alicloud image name...
alicloud-ecs: Found image ID: centos_7_02_64_20G_alibase_20170818.vhd
==> alicloud-ecs: Start creating temporary keypair: packer_59e44f40-
c8d6-0ee3-7fd8-b1ba08ea94b8
==> alicloud-ecs: Start creating alicloud vpc
-----
==> alicloud-ecs: Provisioning with shell script: /var/folders/3q/
w38xx_js6cl6k5mwkrqsnw7w0000gn/T/packer-shell257466182
alicloud-ecs: Loaded plugins: fastestmirror
-----
alicloud-ecs: Total                                     1.3
MB/s | 650 kB 00:00
alicloud-ecs: Running transaction check
-----
==> alicloud-ecs: Deleting temporary keypair...
Build 'alicloud-ecs' finished.
==> Builds finished. The artifacts of successful builds are:
--> alicloud-ecs: Alicloud images were created:
cn-beijing: m-2ze12578be1oa4ovs6r9
```

下一步

[#unique\\_11](#)。

参考链接

- 访问阿里云GitHub Packer仓库 [packer-provider](#) 获取更多信息。
- 参阅 [Packer官方文档](#) 了解更多Packer使用详情。

## 2.1.3 使用 Packer 创建并导入本地镜像

Packer是一款轻量级的镜像定义工具，能够运行在常用的主流操作系统（如Windows、Linux和macOS）上。参阅本文安装并使用Packer，轻松创建自定义镜像。

自行创建本地镜像并上传到云平台是一个非常繁复的过程，此处示范了如何在 Ubuntu 16.04 服务器中创建 CentOS 6.9 本地镜像并上传到阿里云云平台，如果您需要制作其他操作系统类型的镜像，请根据需要 自定义Packer模板。

前提条件

- 您已经 [创建 AccessKey](#)，用于输出到配置文件 centos.json 里。



说明：

由于 AccessKey 权限过大，为防止数据泄露，建议您 [创建 RAM 用户](#)，并使用 RAM 用户子账号 [创建 AccessKey](#)。

- 您已经 [开通OSS服务](#)，便于上传镜像文件到云平台。

## 使用示例

1. 运行 `egrep "(svm|vmx)" /proc/cpuinfo` 确保您的源服务器或者虚拟机是否支持 KVM。

当返回下列信息时，表示支持 KVM。

```
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good
nopl xtopology nonstop_tsc aperfmperf tsc_known_freq pni pclmulqdq
dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm
pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave
avx f16c rdrand lahf_lm abm 3dnowprefetch epb intel_pt tpr_shadow
vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2 smep bmi2
erms invpcid mpx rdseed adx smap clflushopt xsaveopt xsavec xgetbv1
xsaves dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
mca cmov
```

2. 若您的源服务器或者虚拟机支持 KVM，运行以下命令安装 KVM。

```
sudo apt-get install qemu-kvm qemu virt-manager virt-viewer libvirt
-bins bridge-utils #安装 KVM 及相关控件。
sudo virt-manager #开启 virt-manager。
```

如果可以通过 GUI 创建虚拟机，表示您已经成功安装 KVM。

3. 安装 Packer。

您可以参阅 [使用 Packer 创建自定义镜像](#) 部分内容安装 Packer。

4. 依次执行以下命令定义 Packer 模板：



说明：

以下示例制作的本地镜像基于 CentOS 6.9 操作系统，如果您需要创建其他类型操作系统的镜像，您可以根据需要 [自定义 Packer 模板](#) 配置文件 `centos.json`。

```
cd /user/local #切换所在目录。
wget https://raw.githubusercontent.com/alibaba/packer-provider/
master/examples/alicloud/local/centos.json #下载阿里云官方发行的
centos.json 配置文件。
wget https://raw.githubusercontent.com/alibaba/packer-provider/
master/examples/alicloud/local/http/centos-6.9/ks.cfg #下载阿里云官方
发行的 ks.cfg 配置文件。
mkdir -p http/centos-6.9 #创建目录。
mv ks.cfg http/centos-6.9/ #移动文件 ks.cfg 到目录 http/centos-6.9
中。
```

5. 依次执行以下命令创建本地镜像。

```
export ALICLOUD_ACCESS_KEY=您的 AccessKeyID #导入您的 AccessKeyID。
export ALICLOUD_SECRET_KEY=您的 AccessKeySecret #导入您的 AccessKeyS
ecret。
```

```
packer build centos.json #创建本地镜像。
```

示例运行结果如下：

```
qemu output will be in this color.
==> qemu: Downloading or copying ISO
      qemu: Downloading or copying: http://mirrors.aliyun.com/centos/
6.9/isos/x86_64/CentOS-6.9-x86_64-minimal.iso
.....
==> qemu: Running post-processor: alicloud-import
      qemu (alicloud-import): Deleting import source https://oss-cn-
beijing.aliyuncs.com/packer/centos_x86_64
Build 'qemu' finished.
==> Builds finished. The artifacts of successful builds are:
--> qemu: Alicloud images were created:
cn-beijing: XXXXXXXX
```

6. 等待一段时间后，在对应地域的 [ECS 控制台镜像详情页](#) 查看上传结果，如示例中的华北二。

## 后续操作

您可以使用该自定义镜像创建 ECS 实例，具体步骤参阅 [#unique\\_11](#)。

## 自定义 Packer 模板

以上 [使用示例](#) 制作的本地镜像基于 CentOS 6.9 操作系统，如果您需要创建其他类型操作系统的镜像，您可以根据需要自定义 Packer 模板。

例如，以下 JSON 文件是基于 Packer 模板生成的，用于创建 CentOS 6.9 镜像。

```
{
  "variables": {
    "box_basename": "centos-6.9",
    "build_timestamp": "{{isotime \"20060102150405\"}}",
    "cpus": "1",
    "disk_size": "4096",
    "git_revision": "__unknown_git_revision__",
    "headless": "",
    "http_proxy": "{{env `http_proxy`}}",
    "https_proxy": "{{env `https_proxy`}}",
    "iso_checksum_type": "md5",
    "iso_checksum": "af4a1640c0c6f348c6c41f1ea9e192a2",
    "iso_name": "CentOS-6.9-x86_64-minimal.iso",
    "ks_path": "centos-6.9/ks.cfg",
    "memory": "512",
    "metadata": "floppy/dummy_metadata.json",
    "mirror": "http://mirrors.aliyun.com/centos",
    "mirror_directory": "6.9/isos/x86_64",
    "name": "centos-6.9",
    "no_proxy": "{{env `no_proxy`}}",
    "template": "centos-6.9-x86_64",
    "version": "2.1.TIMESTAMP"
  },
  "builders": [
    {
      "boot_command": [
        "<tab> text ks=http://{{ .HTTPIP }}:{{ .HTTPPort }}/{{user `ks_path`}}<enter><wait>"
      ],
      "boot_wait": "10s",
    }
  ]
}
```

```
        "disk_size": "{{user `disk_size`}}",
        "headless": "{{ user `headless` }}",
        "http_directory": "http",
        "iso_checksum": "{{user `iso_checksum`}}",
        "iso_checksum_type": "{{user `iso_checksum_type`}}",
        "iso_url": "{{user `mirror`}}/{{user `mirror_directory`}}/{{
user `iso_name`}}",
        "output_directory": "packer-{{user `template`}}-qemu",
        "shutdown_command": "echo 'vagrant'|sudo -S /sbin/halt -h -p
",
        "ssh_password": "vagrant",
        "ssh_port": 22,
        "ssh_username": "root",
        "ssh_wait_timeout": "10000s",
        "type": "qemu",
        "vm_name": "{{ user `template` }}.raw",
        "net_device": "virtio-net",
        "disk_interface": "virtio",
        "format": "raw"
    }
  ],
  "provisioners": [{
    "type": "shell",
    "inline": [
      "sleep 30",
      "yum install cloud-util cloud-init -y"
    ]
  }],
  "post-processors": [
    {
      "type": "alicloud-import",
      "oss_bucket_name": "packer",
      "image_name": "packer_import",
      "image_os_type": "linux",
      "image_platform": "CentOS",
      "image_architecture": "x86_64",
      "image_system_size": "40",
      "region": "cn-beijing"
    }
  ]
}
```

## Packer 生成器参数说明

**使用示例** 中使用的生成器为 QEMU Builder，该生成器用于创建虚拟机镜像。其必需参数如下所示。

必需参数	类型	描述
iso_checksum	string	操作系统的 ISO 文件的校验和。Packer 在启动挂载了 ISO 的虚拟机之前检查该参数。至少指定 iso_checks um 或者 iso_checksum_url 参数；当您指定了参数 iso_checksum 时，参数 iso_checksum_url 的值被自动忽略。



必需参数	类型	描述
iso_checks um_type	string	已指定的操作系统的 ISO 文件的校验和类型。取值范围： <ul style="list-style-type: none"> <li>· none：忽略校验和过程。不推荐指定该值</li> <li>· md5</li> <li>· sha1</li> <li>· sha256</li> <li>· sha512</li> </ul>
iso_checks um_url	string	一个包含了操作系统 ISO 文件校验和的检验和文件的链接（URL），样式为 GNU 或者 BSD。您需要至少指定 iso_checksum 或者 iso_checksum_url 参数；当您指定了参数 iso_checksum 时，参数 iso_checks um_url 的值被自动忽略。
iso_url	string	一个指向 ISO 并包含安装镜像的 URL。这个 URL 可以是一个 HTTP 链接,也可以是一个文件路径： <ul style="list-style-type: none"> <li>· 如果是 HTTP 链接，Packer 从 HTTP 链接下载文件并缓存运行。</li> <li>· 如果是 IMG 或者 QCOW2 文件链接，QEMU 直接在文件基础上启动。当您指定了文件路径时，请将参数 disk_image 置为 true。</li> </ul>
headless	boolean	Packer 默认通过启动图形化虚拟机界面构建 QEMU 虚拟机。当您 headless 置为 True 时，将启动一个没有控制台的虚拟机。

其余可选参数可参阅 Packer [QEMU Builder](#) 页面。

#### Packer 配置器部分参数说明

**使用示例** 中使用的配置器中包含了 Post-Processor 模块，该模块实现自动化上传本地镜像到 ECS 云平台。Post-Processor 模块的必需参数如下所示。

必需参数	类型	描述
access_key	string	您的 AccessKeyID。由于 AccessKey 权限过大，为防止数据泄露，建议您 <a href="#">创建 RAM 用户</a> ，并使用 RAM 用户子账号 <a href="#">创建 AccessKey</a> 。
secret_key	string	您的 AccessKeySecret。由于 AccessKey 权限过大，为防止数据泄露，建议您 <a href="#">创建 RAM 用户</a> ，并使用 RAM 用户子账号 <a href="#">创建 AccessKey</a> 。
region	string	本地镜像上传的地域，如本示例中的 cn-beijing。请参阅 <a href="#">地域和可用区</a> 查看更多地域信息。

必需参数	类型	描述
image_name	string	您的本地镜像的名称。 <ul style="list-style-type: none"><li>· 由 [2, 128] 位英文或中文字符组成。</li><li>· 必须以大小写字母或中文字符开始。</li><li>· 可以包含数字、下划线 ( _ ) 或者连字符 ( - ) 。</li><li>· 不能以 http:// or https:// 开头。</li></ul>
oss_bucket_name	string	您的 OSS Bucket 名称。若您指定了一个不存在的 Bucket 名称，Packer 上传镜像时自动创建一个同名 Bucket。
image_os_type	string	镜像类型，取值范围： <ul style="list-style-type: none"><li>· linux</li><li>· windows</li></ul>
image_platform	string	镜像发行版本，如本示例中的 CentOS。
image_architecture	string	镜像发行版本的架构平台，取值范围： <ul style="list-style-type: none"><li>· i386</li><li>· x86_64</li></ul>
format	string	镜像的格式，取值范围： <ul style="list-style-type: none"><li>· RAW</li><li>· VHD</li></ul>

其余可选参数可参阅 Packer [Alicloud Post-Processor](#) 页面。

## 下一步

您可以使用该镜像创建 ECS 实例，参阅 [#unique\\_11](#)。

## 参考链接

- 您可以参阅 [Packer 官方文档](#) 了解 Packer 使用详情。
- 您可以访问 GitHub Packer 仓库 [packer](#) 获取更多信息。
- 您可以访问 GitHub Alibaba 仓库 [opstools](#) 获取更多信息。
- 您可以访问 GitHub Alibaba&Packer 仓库 [packer-provider](#) 获取更多信息。
- 您可以参阅 [Anaconda Kickstart](#) 了解配置文件 ks.cfg 的更多信息。

## 2.2 修改自定义镜像信息

创建了自定义镜像后，您可以重新修改自定义镜像的名称和描述。

### 操作步骤

1. 登录 [ECS管理控制台](#)。
2. 在左侧导航栏中，选择 快照和镜像 > 镜像。
3. 选择地域。
4. 找到需要编辑的 自定义镜像。
5. 单击 图标，重新填写镜像名称。
6. 单击 编辑描述，在弹出的对话框中：
  - 自定义镜像描述：重新填写镜像描述。
  - （可选）标签：重新选择标签。
7. 单击 保存 完成修改自定义镜像描述信息。

您也可以使用ECS API [ModifyImageAttribute](#) 修改自定义镜像信息。

## 2.3 导入镜像

### 2.3.1 镜像规范检测工具

本文通过操作示例、参数介绍和输出详解为您介绍如何使用镜像规范检测工具，自动定位不符合阿里云规范的操作系统设置，适用于导入自定义镜像等场景。

#### 背景信息

ECS支持使用自定义镜像创建实例。该自定义镜像可以是您基于线下服务器、虚拟机或者其他云平台的云主机创建的。您导入的镜像需要满足一定镜像要求，更多详情，请参阅[导入镜像必读](#)。

为缩短镜像制作及创建实例的周期，我们推荐您使用 ECS 镜像规范检测工具，本文简称检测工具，制作符合规范的镜像。镜像规范检测工具能基于服务器环境检测各项配置指标、定位不合规原因、生成TXT和JSON格式检测报告并给出解决意见。

#### 使用限制

目前检测工具仅支持检测Linux类型镜像，包括Ubuntu、CentOS、Debian、RedHat、SLES（SUSE Linux Enterprise Server）、OpenSUSE、FreeBSD、CoreOS和Others Linux发行版本。

## 使用示例

以CentOS 7.4 64位服务器为例。

1. 登录您的服务器、虚拟机或者其他云平台的云主机。
2. [下载](#)检测工具。
3. 使用root权限运行image\_check，避免检测工具无法读取有权限控制的配置文件。

```
chmod +x image_check
sudo image_check -p [目标路径]
```



## 说明:

您可以使用-p [目标路径]指定检测报告的生成路径，不指定则默认输出在检测工具所在的路径。

4. 等待检测工具检测系统配置。

```
Begin check your system...
The report is generating.
-----
The information you need to input when you import your image to
Alibaba Cloud website:
Current system: CentOS          # 系统信息1: 服务器操作系统
Architecture: x86_64          # 系统信息2: 系统架构
System disk size: 42 GB       # 系统信息3: 服务器系统盘容量
-----
# 检测项
Check driver                   [ OK ]
Check shadow file authority    [ OK ]
Check security                 [ OK ]
Check qemu-ga                  [ OK ]
Check network                  [ OK ]
Check ssh                      [ OK ]
Check firewall                 [ OK ]
Check filesystem               [ OK ]
Check device id                [ OK ]
Check root account             [ OK ]
Check password                 [ OK ]
Check partition table          [ OK ]
Check lvm                      [ FAILED ]
Check lib                      [ OK ]
Check disk size                [ OK ]
Check disk use rate            [ WARNING ]
Check inode use rate           [ OK ]
-----
15 items are OK
1 items are failed
1 items are warning
-----
The report is generated: /root/image_check_report_2018-05-14_18-18-
10.txt
Please read the report to check the details
```

5. 在相应位置查看检测报告。报告格式为image\_check\_report\_日期\_时间.txt或image\_check\_report.json。

## 检测项

为避免使用您的自定义镜像创建的ECS实例功能不全，检测工具主要检测服务器中以下配置项。

检测项	不合规后果	合规建议
driver	无法正常启动ECS实例	安装虚拟化驱动，如 <a href="#">安装 virtio 驱动</a>
/etc/shadow	无法修改密码文件，进而无法根据自定义镜像创建ECS实例	不要使用chattr命令锁定/etc/shadow文件
SELinux	无法正常启动ECS实例	不要通过修改/etc/selinux/config开启 SELinux
qemu-ga	ECS所需要的部分服务不可用，实例功能不全	卸载qemu-ga
network	ECS实例网络功能不稳定	关闭或删除Network Manager，并打开 network服务
ssh	无法从控制台 <a href="#">远程连接</a> ECS实例	打开SSH服务，不要设置PermitRootLogin
firewall	系统无法自动配置您的ECS实例环境	关闭防火墙Iptables、Firewalld、IPFILTER (IPF)、 IPFIREWALL (IPFW)或PacketFilter (PF)
file system	无法 <a href="#">扩容磁盘</a>	<ul style="list-style-type: none"> <li>· 推荐使用xfs、ext3、ext4文件系统</li> <li>· 容许使用ext2、ufs、udf文件系统</li> <li>· ext4文件系统不要使用64 bit feature</li> </ul>
root	无法使用用户名和密码远程连接ECS实例	保留root账号
passwd	ECS实例无法新增用户	保留passwd命令，或重新安装passwd
分区表	无法正常启动ECS实例	采用MBR分区
Logical Volume Manager (LVM)	无法正常启动ECS实例	更换为其他分区服务
/lib	无法自动化配置ECS实例	/lib和/lib64的指向位置不能为绝对路径，修改链接文件/lib和/lib64的指向位置为相对路径

检测项	不合规后果	合规建议
system disk	N/A	增加系统盘容量，系统盘空间最佳为40GiB至500GiB。并在导入镜像时根据镜像的虚拟文件大小而非使用容量配置系统盘容量
disk usage	无法为ECS实例安装必要的驱动或服务	保持足够的磁盘空间
inode usage	无法为ECS实例安装必要的驱动或服务	保持足够的inode资源

检测工具会根据检测项给出OK、FAILED或者WARNING检测结果。

- OK：检测项均符合要求。
- FAILED：检测项不符合要求，使用该自定义镜像创建的ECS实例无法正常启动。建议您修复报错项后再制作镜像，以提高实例启动效率。
- WARNING：检测项不符合要求，使用该自定义镜像创建的ECS实例可以安全启动，但ECS无法通过有效途径干预您的实例配置。您可以选择立即修复，也可以暂时忽略问题前往制作镜像环节。

## 输出项

检测系统环境后，会输出TXT和JSON两种格式检测报告。输出路径可以使用-p [目标路径]指定，不指定则默认输出在检测工具所在的路径。

- TXT格式报告名称为image\_check\_report\_日期\_时间.txt，报告内容包括服务器配置信息以及检测结果。以CentOS 7.4 64位服务器为例：

```
The information you need to input when you import your image to
Alibaba Cloud Website:
Current system is: CentOS          #服务器操作系统
Architecture: x86_64             #系统架构
System disk size: 42 GB          #服务器系统盘容量
-----
Check driver                      #检测项名称
Pass: kvm drive is exist          #检测结果
Alibaba Cloud supports kvm virtualization technology
We strongly recommend installing kvm driver.
```

- JSON格式报告名称为image\_check\_report.json，报告内容包括服务器配置信息以及检测结果。以CentOS 7.4 64位服务器为例：

```
"platform": "CentOS",           \\服务器操作系统
"os_big_version": "7",          \\操作系统版本号（高位）
"os_small_version": "4",        \\操作系统版本号（低位）
"architecture": "x86_64",       \\系统架构
"system_disk_size": "42",        \\服务器系统盘容量
"version": "1.0.2",             \\检测工具版本
"time": "2018-05-14_19-18-10",  \\检测时间
"check_items": [{
```

```
"name": "driver",          \\检测项名称
"result": "OK",           \\检测结果
"error_code": "0",        \\错误码
"description": "Pass: kvm driver exists.", \\描述
"comment": "Alibaba Cloud supports kvm virtualization
technology. We strongly recommend installing kvm driver."
  ]]
}
```

## 下一步

1. 查看[导入镜像注意事项](#)
2. [安装virtio驱动](#)
3. (可选) [转换镜像格式](#)
4. [导入镜像](#)
5. [使用自定义镜像创建实例](#)

## 2.3.2 导入镜像必读

为保证导入镜像的可用性，提高镜像导入效率，您需要考虑ECS对导入镜像的限制条件。

根据操作系统不同，如 [Windows类型镜像](#) 和 [Linux类型镜像](#)，导入镜像的注意事项不同。

### Windows类型镜像

#### 重要建议

- 导入Windows操作系统的镜像前，请确认文件系统的完整性。
- 请检查系统盘的剩余空间，确保系统盘没有被写满。
- 关闭防火墙，并放行RDP 3389端口。
- administrator账号的登录密码必须是：长度为 8 至 30 个字符，必须同时包含大小写英文字母、数字和特殊符号。特殊符号可以是()<`~!@#%&^\*-\_+=|{}[]; ‘<>,.?/。其中，Windows实例不能以斜线号 (/) 为密码首字符。
- 根据镜像的虚拟磁盘大小而非使用容量配置导入的系统盘大小，系统盘容量范围支持40GiB–500GiB。
- 请勿修改关键系统文件。

#### 支持项

- 支持多分区系统盘。
- 支持NTFS文件系统，支持MBR分区。
- 支持RAW、qcow2和VHD格式镜像。



#### 说明：

导入其他格式的镜像前，请先[#unique\\_31](#)后再导入。建议您转换成传输容量较小的VHD格式。

- ECS支持导入包含以下操作系统版本的镜像：
  - Microsoft Windows Server 2016
  - Microsoft Windows Server 2012 R2（标准版）
  - Microsoft Windows Server 2012（标准版和数据中心版）
  - Microsoft Windows Server 2008 R2（标准版、数据中心版和企业版）
  - Microsoft Windows Server 2008（标准版、数据中心版和企业版）
  - 含Service Pack 1 (SP1) 的Windows Server 2003（标准版、数据中心版和企业版）或更高版本

#### 不支持项

- 不支持在镜像中安装qemu-ga，否则会导致ECS所需要的部分服务不可用。
- 不支持Windows XP、专业版和企业版Windows 7、Windows 8和Windows 10。

#### Linux类型镜像

##### 重要建议

- 导入Linux操作系统的镜像前，请确认文件系统的完整性。
- 请检查系统盘的剩余空间，确保系统盘没有被写满。
- 关闭防火墙，并放行TCP 22端口。
- 安装虚拟化平台XEN或者KVM驱动。
- 建议 [安装 cloud-init](#)，以保证能成功配置hostname、NTP 源和yum源。
- 需要开启DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）。
- root账号的登录密码必须是8-30个字符，并且同时包含大写或小写字母、数字和特殊符号。其中特殊字符可以是 ( ) ` ~ ! @ # \$ % ^ & \* \_ + = | { } [ ] : ; ' < > , . ? /
- 请勿修改关键系统文件，如 /sbin、/bin 和 /lib\* 等目录。

#### 支持项

- 支持RAW、qcow2和VHD格式镜像。



说明:

导入其他格式的镜像前，请先 [#unique\\_31](#) 后再导入。建议您转换成传输容量较小的VHD格式。

- 支持xfs、ext3和ext4文件系统，支持MBR分区。



说明:



支持的ext4文件系统中不能包含特性（feature） 64bit，并且特性 project 和 quota 不能成对出现。您可以运行 `tune2fs -l <ext4文件系统盘所在路径> | grep features` 命令查看ext4文件系统中包含的特性列表。

#### 不支持项

- 不支持多个网络接口。
- 不支持IPv6地址。
- 不支持调整系统盘分区，目前只支持单个根分区。

#### 注意事项

根据您导入的Linux系统镜像是否为标准平台镜像，您需要注意不同的问题。

- 我们定义官方发布的操作系统发行版本为 标准平台镜像，目前支持的系统版本包括Aliyun Linux、CentOS 5/6/7、CoreOS 681.2.0+、Debian 6/7、FreeBSD、OpenSUSE 13.1、RedHat、RHEL（Red Hat Enterprise Linux）、SUSE Linux 10/11/12和Ubuntu 10/12/13/14。
- 不在ECS提供的公共镜像列表里的操作系统平台镜像为 非标准平台镜像。虽然镜像来自标准平台，但是系统关键性配置文件、系统基础环境 and 应用方面没有遵守标准平台要求的镜像。如果您需要使用非标准平台镜像，在导入镜像时您只能选择：
  - Others Linux：ECS统一标识为其他系统类型。如果导入Others Linux平台镜像，ECS不会对所创建的实例做任何处理。如果您在制作镜像前开启了DHCP，ECS会自动为您配置网络。完成实例创建后，您需要通过管理控制台的 [远程连接](#) 功能连接实例，再自行配置IP、路由和密码等。
  - Customized Linux：定制版镜像。导入Customized Linux镜像后，请按照ECS标准系统配置方式配置实例的网络和密码等。更多详情，请参见 [#unique\\_38](#)。

名称	标准平台镜像	非标准平台镜像
系统关键配置文件要求	<ul style="list-style-type: none"> <li>· 请勿修改 <code>/etc/issue*</code>，否则ECS无法正常识别系统发行版，从而创建系统失败。</li> <li>· 请勿修改 <code>/boot/grub/menu.lst</code>，否则无法启动ECS实例。</li> <li>· 请勿修改 <code>/etc/fstab</code>，否则无法加载异常分区从而无法启动ECS实例。</li> <li>· 请勿修改 <code>/etc/shadow</code>为 只读，否则将导致无法修改密码文件，从而创建系统失败。</li> <li>· 请勿修改 <code>/etc/selinux/config</code> 开启SELinux，否则系统无法启动。</li> </ul>	不符合标准平台镜像的要求

名称	标准平台镜像	非标准平台镜像
应用要求	不支持在镜像中安装 qemu-ga，否则会导致ECS所需要的部分服务不可用。	不符合标准平台镜像的要求

### 2.3.3 安装 cloud-init

如果您需要使用已有镜像创建ECS实例，可以通过导入镜像功能完成配置。在导入Linux类型镜像时，为了保证导入的镜像hostname、NTP和Yum源配置成功，建议您预先在源服务器、虚拟机或者云主机上安装cloud-init。

#### 注意事项

- 目前，支持安装cloud-init的Linux发行版包括CentOS、Debian、Fedora、FreeBSD、Gentoo、RHEL（Red Hat Enterprise Linux）、SLES（SUSE Linux Enterprise Server）和Ubuntu。
- 若您的源服务器、虚拟机或者云主机已经安装了cloud-init，您需要确认cloud-init版本号高于0.7.9，包含低版本cloud-init的镜像会引起相关实例配置失败，例如NTP、HostName 和 Yum等。
  1. 登录源服务器、虚拟机或者云主机。
  2. 运行 `cloud-init --version` 查询cloud-init版本。

如果您的cloud-init版本号高于0.7.9，表示该版本可用。否则您需要 [安装cloud-init](#)。

#### 前提条件

安装cloud-init之前，您的源服务器、虚拟机或者云主机必须安装以下软件。

- git：下载cloud-init源码包  
yum安装方式：`yum install git`
- Python2.7：cloud-init运行和安装的基础  
yum安装方式：`yum install python`
- pip：安装cloud-init依赖的Python库  
yum安装方式：`yum install python-pip`

此处列举了 yum 安装方式。zypper 或者 apt-get 的包管理器安装方式与 yum 安装方式类似。

#### 安装cloud-init

您可以按以下步骤安装cloud-init：

1. 登录待导入镜像的源服务器、虚拟机或者云主机。

2. 运行 `git clone https://git.launchpad.net/cloud-init` 从cloud-init官网下载cloud-init源码包。
3. 运行 `cd cloud-init` 进入cloud-init目录。
4. 运行 `python setup.py install` 安装cloud-init的安装文件setup.py。
5. 运行 `vi /etc/cloud/cloud.cfg` 修改配置文件cloud.cfg。

将 `cloud_init_modules` 之前的配置修改成如下：

```
# Example datasource config
# The top level settings are used as module
# and system configuration.
# A set of users which may be applied and/or used by various
modules
# when a 'default' entry is found it will reference the 'default_us
er'
# from the distro configuration specified below
users:
  - default
user:
  name: root
  lock_passwd: False
# If this is set, 'root' will not be able to ssh in and they
# will get a message to login instead as the above $user
disable_root: false
# This will cause the set+update hostname module to not operate (if
true)
preserve_hostname: false
syslog_fix_perms: root:root
datasource_list: [ AliYun ]
# Example datasource config
datasource:
  AliYun:
    support_xen: false
    timeout: 5 # (defaults to 50 seconds)
    max_wait: 60 # (defaults to 120 seconds)
#     metadata_urls: [ 'blah.com' ]
# The modules that run in the 'init' stage
cloud_init_modules:
```

## 排查故障

不同镜像缺少的库可能不同，您都可以通过pip安装，安装完之后再次重复以上 [步骤4](#)。

### 缺少six和oauthlib库

- 安装过程中，如果出现以下信息，表示Python缺少six库。您可以使用pip安装six库：`pip install six`。

```
File "/root/cloud-init/cloudinit/log.py", line 19, in <module>
import six
```

```
ImportError: No module named s )
```

- 安装过程中，如果出现以下信息，表示Python缺少oauthlib库。您可以使用pip安装oauthlib库：`pip install oauthlib`。

```
File "/root/cloud-init/cloudinit/url_helper.py", line 20, in <module>
    import oauthlib.oauth1 as oauth1
ImportError: No module named oauthlib.oaut )
```

提示错误时没有明确缺少的依赖库

如果安装报错时，没有明确提示缺少哪些依赖库时，您可以根据cloud-init的requirements.txt文件里显示的库，并运行`pip install -r requirements.txt`安装所有依赖库。

#### 后续步骤

您可以 [导入自定义镜像](#)。

#### 参考链接

cloud-init [Alibaba Cloud \(AliYun\)](#)

## 2.3.4 安装virtio驱动

为避免部分服务器、虚拟机或者云主机的操作系统在导入自定义镜像后，创建的ECS实例无法启动，您需要在导入镜像前检查是否需要在源服务器中安装virtio驱动。

#### 无需安装virtio驱动的镜像

从本地[导入自定义镜像](#)时，阿里云会自动处理导入的自定义镜像的virtio驱动的操作系统有：

- Windows Server 2008
- Windows Server 2012
- Windows Server 2016
- CentOS 6/7
- Ubuntu 12/14/16
- Debian 7/8/9
- SUSE 11/12

以上列表的镜像，默认已安装virtio驱动的系统，需要注意[修复临时文件系统](#)。

#### 需要安装virtio驱动的镜像

其他不在以上列表的操作系统，您需要在导入镜像之前，为源服务器安装virtio驱动。

检查服务器内核是否支持virtio驱动

1. 运行 `grep -i virtio /boot/config-$(uname -r)` 检查当前操作系统的内核是否支持virtio驱动。



说明:

- 如果在输出信息中没有找到 VIRTIO\_BLK 及 VIRTIO\_NET 的信息，表示该操作系统没有安装virtio相关驱动，暂时不能直接导入阿里云云平台。您需要为您的服务器编译安装virtio驱动。
- 如果参数 CONFIG\_VIRTIO\_BLK 及 CONFIG\_VIRTIO\_NET 取值为 y，表示包含了virtio驱动，您可以参阅 [导入镜像必读](#) 直接 [导入自定义镜像](#) 到阿里云。
- 如果参数 CONFIG\_VIRTIO\_BLK 及 CONFIG\_VIRTIO\_NET 取值为 m，需要进入第2步。

2. 执行命令 `lsinitrd /boot/initramfs-$(uname -r).img | grep virtio` 确认virtio驱动是否包含在临时文件系统initramfs或者initrd中。



说明:

- 截图表明，initramfs已经包含了virtio\_blk驱动，以及其所依赖的virtio.ko、virtio\_pci.ko和virtio\_ring.ko，您可以参阅 [导入镜像必读](#) 直接 [导入自定义镜像](#) 到阿里云。
- 如果临时文件系统initramfs没有包含virtio驱动，则需要修复临时文件系统。

### 修复临时文件系统

通过检查，发现源服务器内核支持virtio驱动，但是临时文件系统initramfs或者initrd中没有包含virtio驱动时，需要修复临时文件系统。以CentOS等为例。

- CentOS/RedHat 5

```
mkinitrd -f --allow-missing \  
--with=xen-vbd --preload=xen-vbd \  
--with=xen-platform-pci --preload=xen-platform-pci \  
--with=virtio_blk --preload=virtio_blk \  
--with=virtio_pci --preload=virtio_pci \  
--with=virtio_console --preload=virtio_console \
```

- CentOS/RedHat 6/7

```
mkinitrd -f --allow-missing \  
--with=xen-blkfront --preload=xen-blkfront \  
--with=virtio_blk --preload=virtio_blk \  
--with=virtio_pci --preload=virtio_pci \  
--with=virtio_console --preload=virtio_console \
```

```
/boot/initramfs-$(uname -r).img $(uname -r)
```

- Debian/Ubuntu

```
echo -e 'xen-blkfront\nvirtio_blk\nvirtio_pci\nvirtio_console' >> \
/etc/initramfs-tools/modules
mkinitramfs -o /boot/initrd.img-$(uname -r)"
```

## 编译安装virtio驱动

此处以Redhat服务器为例，为您示范如何编译安装virtio驱动。

### 下载内核安装包

1. 运行 `yum install -y ncurses-devel gcc make wget` 安装编译内核的必要组件。
2. 运行 `uname -r` 查询当前系统使用的内核版本，如示例中的4.4.24-2.a17.x86\_64。
3. 前往 [Linux内核列表页面](#) 下载对应的内核版本源码，如示例中的4.4.24开头的linux-4.4.24.tar.gz的网址为 <https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.4.24.tar.gz>。
4. 运行 `cd /usr/src/` 切换目录。
5. 运行 `wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.4.24.tar.gz` 下载安装包。
6. 运行 `tar -xzf linux-4.4.24.tar.gz` 解压安装包。
7. 运行 `ln -s linux-4.4.24 linux` 建立链接。
8. 运行 `cd /usr/src/linux` 切换目录。

### 编译内核

1. 依次运行以下命令编译内核。

```
make mrproper
symvers_path=$(find /usr/src/ -name "Module.symvers")
test -f $symvers_path && cp $symvers_path .
cp /boot/config-$(uname -r) ./config
make menuconfig
```

2. 出现以下界面时，开始打开virtio相关配置：



说明：

选 \* 配置表示编译到内核，选 m 配置表示编译为模块。

a. 使用空格勾选Virtualization项。

确认是否勾选了KVM（Kernel-based Virtual Machine）选项。

```
Processor type and features --->
  [*] Paravirtualized guest support --->
      --- Paravirtualized guest support
  (128) Maximum allowed size of a domain in gigabytes
  [*] KVM paravirtualized clock
  [*] KVM Guest support
```

```
Device Drivers --->
  [*] Block devices --->
  <M> Virtio block driver (EXPERIMENTAL)
  -* Network device support --->
      <M> Virtio network driver (EXPERIMENTAL)
```

b. 按下Esc键退出内核配置界面并根据弹窗提示保存.config文件。

c. [检查](#) virtio相关配置是否已经正确配置。

d. 若检查后发现暂未设置virtio相关配置，运行以下命令手动编辑.config文件。

```
make oldconfig
make prepare
make scripts
make
make install
```

e. 运行以下命令查看virtio驱动的安装情况。

```
find /lib/modules/"$(uname -r)"/ -name "virtio.*" | grep -E "
virtio.*"
grep -E "virtio.*" < /lib/modules/"$(uname -r)"/modules.builtin
```



说明:

如果任一命令输出virtio\_blk、virtio\_pci.virtio\_console等文件列表，表明您已经正确安装了virtio驱动。

## 下一步

检查virtio驱动后，您可以：

- [#unique\\_44](#)。
- [导入自定义镜像](#)。



## 2.3.5 定制 Linux 自定义镜像

当您的镜像操作系统不在阿里云所支持的已有平台类型中，亦不能安装 cloud-init 时，您可以在导入自定义镜像时选择 Customized Linux（定制版镜像）。阿里云将定制版 Linux 镜像当作无法识别的操作系统类型，缺少初次启动ECS实例时所必要的标准配置信息，需要您导入镜像前根据本文描述在定制版镜像中添加解析脚本，便于初次启动时自动化配置实例。

### 限制条件

- Customized Linux 镜像的第一个分区必须可以被写入。
- Customized Linux 镜像的第一个分区类型支持 FAT32、EXT2、EXT3、EXT4 或 UFS。
- Customized Linux 镜像的虚拟文件大小必须大于 5 GiB。
- Customized Linux 镜像有如下安全要求：
  - 不能存在可被远程利用的高危漏洞。
  - 使用控制台的 [管理终端](#) 登录实例时，如果存在初始默认密码，必须在首次登录时修改，修改密码之前不允许进入实例做任何操作。
  - 不支持默认 SSH 密钥对，初始 SSH 密钥对必须由阿里云随机生成。

### 配置方法

1. 在镜像的第一个分区的根目录下新建目录 `aliyun_custom_image`。

使用该 Customized Linux 镜像所创建的实例初次启动时，阿里云会在 `aliyun_custom_image` 目录的 `os.conf` 文件中写入实例相关配置信息。如果不存在 `os.conf` 文件，系统则自动创建。

2. 在镜像中创建一份解析脚本，用以解析 `os.conf` 文件的系统配置。参阅 [解析脚本注意事项](#) 和 [解析脚本示例](#) 编写脚本。

### os.conf 文件示例

#### 经典网络类型实例

```
hostname=iZ23r29djmjZ
password=cXdlcjEyMzQK
eth0_ip_addr=10.171.254.123
eth0_mac_addr=00:8c:fa:5e:14:23
eth0_netmask=255.255.255.0
eth0_gateway=10.171.254.1
eth0_route="10.0.0.0/8 10.171.254.1;172.16.0.0/12 10.171.254.1"
eth1_ip_addr=42.120.74.105
eth1_mac_addr=00:8c:fa:5e:14:24
eth1_netmask=255.255.255.0
eth1_gateway=42.120.74.1
eth1_route="0.0.0.0/0 42.120.74.1"
dns_nameserver="7.7.7.7 8.8.8.8"
```

参数说明如下表所示：



参数名称	参数说明
hostname	主机名参数。
password	密码参数，Base64 编码的字符串。
eth0_ip_addr	eth0 网卡 IP 地址。
eth0_mac_addr	eth0 网卡 MAC 地址。
eth0_netmask	eth0 网卡掩码。
eth0_gateway	eth0 网卡默认网关。
eth0_route	eth0 内网路由列表，默认用半角分号分隔。
eth1_ip_addr	eth1 网卡 IP 地址。
eth1_mac_addr	eth1 网卡 MAC 地址。
eth1_netmask	eth1 网卡掩码。
eth1_gateway	eth1 网卡默认网关。
eth1_route	eth1 公网路由列表，默认用半角分号分隔。
dns_nameserver	DNS 地址列表，默认用空格分隔。

#### 专有网络 VPC 类型实例

```
hostname=iZ23r29djmjZ
password=cXdlcjEyMzQK
eth0_ip_addr=10.171.254.123
eth0_mac_addr=00:8c:fa:5e:14:23
eth0_netmask=255.255.255.0
eth0_gateway=10.171.254.1
eth0_route="0.0.0.0/0 10.171.254.1"
dns_nameserver="7.7.7.7 8.8.8.8"
```

参数说明如下表所示：

参数名称	参数说明
hostname	主机名参数。
password	密码参数，Base64 编码的字符串。
eth0_ip_addr	eth0 网卡 IP 地址。
eth0_mac_addr	eth0 网卡 MAC 地址。
eth0_netmask	eth0 网卡掩码。
eth0_gateway	eth0 网卡默认网关。
eth0_route	eth0 内网路由列表，默认用半角分号分隔。
dns_nameserver	DNS 地址列表，默认用空格分隔。

## 解析脚本注意事项

实例初次启动时，正常情况下阿里云自动将配置项的相关信息写入第一个分区的根目录下 `aliyun_custom_image` 目录的 `os.conf` 文件中。配置 Customized Linux 镜像必须要在镜像中创建预定义解析脚本，用以从 `os.conf` 文件中读取实例配置信息并完成实例配置。以下为解析脚本需要满足的条件。

- 开机启动：解析脚本需要设置成开机自启动，例如，将解析脚本存放在 `/etc/init.d/` 目录下。
- 配置项取值规则：如 [os.conf 文件示例](#) 的配置项所述，VPC 与经典网络实例的配置项数量和部分配置项的取值规则均有所不同。
- 配置文件读取路径：Customized Linux 镜像在创建 I/O 优化实例或非 I/O 优化实例时，为第一个分区所分配的设备名默认不一样。所以在解析脚本中最好可以用 `uuid` 或 `label` 识别第一个分区的设备。用户密码为 Base64 编码的字符串，设置密码时需要做相关处理。
- 判断 VPC 或经典网络：解析脚本判断该网络类型时，可以查看是否存在 `eth1_route` 或其他 `eth1` 相关的配置项。判断出当前实例的网络类型后再有针对性地解析和处理。
  - VPC 类型实例在 `os.conf` 文件的 `eth0_route` 参数中配置默认公网路由。
  - 经典网络类型实例在 `os.conf` 文件的 `eth1_route` 参数中配置默认公网路由，内网路由配置在 `eth0_route` 中。
- 配置优化：`os.conf` 文件中的配置在实例的整个生命周期中执行一次即可，解析脚本执行成功后建议删除 `os.conf` 文件。同时解析脚本如果没有读取到 `os.conf` 文件配置，则不执行文件中的配置。
- 自定义镜像处理：根据 Customized Linux 镜像创建的实例再制作自定义镜像时，镜像中也会包含这个开机启动脚本。阿里云会在实例第一次启动时写入 `os.conf` 配置，解析脚本在检测到该配置时即可执行相关配置。
- 修改相关配置时的处理：当实例的配置信息通过阿里云的控制台或 API 发生变更时，阿里云将相关信息写入到 `os.conf` 文件中，解析脚本将被再次执行从而下发这些更改。

## 解析脚本示例

以下为以 CentOS 操作系统为例的解析脚本示例，仅供参考，您需要根据实际的操作系统类型调整脚本内容。在使用脚本前，务必在镜像中调试脚本，并保证调试通过。

```
#!/bin/bash

### BEGIN INIT INFO
# Provides:          os-conf
# Required-Start:    $local_fs $network $named $remote_fs
# Required-Stop:
# Should-Stop:
# Default-Start:     2 3 4 5
```

```
# Default-Stop:      0 1 6
# Short-Description: The initial os-conf job, config the system.
### END INIT INFO

first_partition_dir='/boot/'
os_conf_dir=${first_partition_dir}/aliyun_custom_image
os_conf_file=${os_conf_dir}/os.conf

load_os_conf() {
    if [[ -f $os_conf_file ]]; then
        . $os_conf_file
        return 0
    else
        return 1
    fi
}

cleanup() {
    # ensure $os_conf_file is deleted, to avoid repeating config system
    rm $os_conf_file >& /dev/null
    # ensure $os_conf_dir is existst
    mkdir -p $os_conf_dir
}

config_password() {
    if [[ -n $password ]]; then
        password=$(echo $password | base64 -d)
        if [[ $? == 0 && -n $password ]]; then
            echo "root:$password" | chpasswd
        fi
    fi
}

config_hostname() {
    if [[ -n $hostname ]]; then
        sed -i "s/^HOSTNAME=.*HOSTNAME=$hostname/" /etc/sysconfig/network
        hostname $hostname
    fi
}

config_dns() {
    if [[ -n $dns_nameserver ]]; then
        dns_conf=/etc/resolv.conf
        sed -i '/^nameserver.*/d' $dns_conf
        for i in $dns_nameserver; do
            echo "nameserver $i" >> $dns_conf
        done
    fi
}

is_classic_network() {
    # vpc: eth0
    # classic: eth0 eth1
    grep -q 'eth1' $os_conf_file
}

config_network() {
    /etc/init.d/network stop
    config_interface eth0 ${eth0_ip_addr} ${eth0_netmask} ${eth0_mac_addr}
}

config_route eth0 ${eth0_route}
if is_classic_network ; then
    config_interface eth1 ${eth1_ip_addr} ${eth1_netmask} ${eth1_mac_a
ddr}
fi
```

```

    config_route eth1 ${eth1_route}
    fi
    /etc/init.d/network start
}

config_interface() {
    local interface=$1
    local ip=$2
    local netmask=$3
    local mac=$4
    inteface_cfg="/etc/sysconfig/network-scripts/ifcfg-${interface}"
    cat << EOF > $inteface_cfg
DEVICE=$interface
IPADDR=$ip
NETMASK=$netmask
HWADDR=$mac
ONBOOT=yes
BOOTPROTO=static
EOF
}

config_default_gateway() {
    local gateway=$1
    sed -i "s/^GATEWAY=.* /GATEWAY=$gateway/" /etc/sysconfig/network
}

config_route() {
    local interface=$1
    local route=$2
    route_conf=/etc/sysconfig/network-scripts/route-${interface}
    > $route_conf
    echo $route | sed 's;/\n/ | \'
    while read line; do
        dst=$(echo $line | awk '{print $1}')
        gw=$(echo $line | awk '{print $2}')
        if ! grep -q "$dst" $route_conf 2> /dev/null; then
            echo "$dst via $gw dev $interface" >> $route_conf
        fi
        if [[ "$dst" == "0.0.0.0/0" ]]; then
            config_default_gateway $gw
        fi
    done
}

##### sysvinit service portal #####

start() {
    if load_os_conf ; then
        config_password
        config_network
        config_hostname
        config_dns
        cleanup
        return 0
    else
        echo "not load $os_conf_file"
        return 0
    fi
}

RETVAL=0

case "$1" in
    start)

```

```
        start
        RETVAL=$?
    ;;
    *)
        echo "Usage: $0 {start}"
        RETVAL=3
    ;;
esac
exit $RETVAL
```

## 2.3.6 转换镜像格式

ECS只支持导入RAW、VHD和qcow2格式的镜像文件。其他镜像文件，需要转换格式后再导入。本文介绍如何利用qemu-img工具将其它格式的镜像文件转换成VHD或RAW格式。qemu-img能将RAW、qcow2、VMDK、VDI、VHD (vpc)、VHDX、qcow1或QED格式的镜像转换成VHD格式，也可以实现RAW和VHD格式的互相转换。

本地为Windows操作系统

按以下步骤安装qemu-img并转换镜像文件格式：

1. 下载qemu-img并完成安装。本示例安装路径为C:\Program Files\qemu。
2. 为qemu-img配置环境变量：
  - a. 选择 开始 > 计算机，右键单击 属性。
  - b. 在左侧导航栏里，单击 高级系统设置。
  - c. 在 系统属性 对话框里，单击 高级 页签，并单击 环境变量。
  - d. 在 环境变量 对话框里，在 系统变量 部分找到 Path，并单击 编辑。如果 Path 变量不存在，单击 新建。
  - e. 添加系统变量值：
    - 如果是 编辑系统变量，在 变量值 里，添加 C:\Program Files\qemu，不同的变量值之间以半角分号 (;) 分隔。
    - 如果是 新建系统变量，在 变量名 处输入 Path，在 变量值 处输入 C:\Program Files\qemu。
3. 测试是否成功配置环境变量：打开Windows 命令提示符，运行qemu-img --help，如果显示正常即表示已配置环境变量。
4. 在 命令提示符 里，使用cd [源镜像文件所在的目录]切换文件目录，例如cd D:\ConvertImage。

5. 运行 `qemu-img convert -f qcow2 -O raw centos.qcow2 centos.raw` 命令转换镜像文件格式，其中：

- `-f` 的参数值为源镜像文件的格式。
- `-O`（必须是大写）的参数值为目标镜像格式、源镜像文件名称和目标文件名称。

转换完成后，目标文件会出现在源镜像文件所在的目录下。

本地为Linux操作系统

按以下步骤安装 `qemu-img` 并转换镜像文件格式：

1. 安装 `qemu-img`，示例如下：

- 如果本地为Ubuntu操作系统，运行 `apt install qemu-img`。
- 如果本地为CentOS操作系统，运行 `yum install qemu-img`。

2. 运行 `qemu-img convert -f qcow2 -O raw centos.qcow2 centos.raw` 命令转换镜像文件格式，其中：

- `-f` 的参数值为源镜像文件的格式。
- `-O`（必须是大写）的参数值为目标镜像格式、源镜像文件名称和目标文件名称。

转换完成后，目标文件会出现在源镜像文件所在的目录下。

常见问题

如果安装 `qemu-img` 报错时，而且没有明确提示缺少哪些依赖库时，您可以根据 `cloud-init` 的 `requirements.txt` 文件里显示的库，并运行 `pip install -r requirements.txt` 安装所有依赖库。

下一步

[导入自定义镜像](#)

## 2.3.7 导入自定义镜像

将本地物理镜像文件导入ECS适用于在云上部署业务的场景。导入的自定义镜像会出现在相应地域下您的自定义镜像列表里，您可以使用这份镜像创建ECS实例和更换系统盘。



说明：

- 导入自定义镜像是相对耗时的任务，完成的时间取决于镜像文件的大小和当前导入任务并发数，需要您耐心等待。

- 导入自定义镜像时，系统会自动生成一份快照，您可以在 快照列表 里看到快照信息。导入镜像任务未完成前，快照的状态会显示为 失败。任务完成后，状态会自动更新为 可用。这份快照的容量为导入的自定义镜像文件大小，与导入自定义镜像时设置的系统盘大小无关。

### 前提条件

在导入镜像前，您应该已经完成如下工作：

- 参阅 [导入镜像必读](#)、[#unique\\_38](#) 和 [#unique\\_31](#) 了解镜像限制和要求。
- [开通OSS服务](#)。
- （可选）如果您使用的是RAM子账号，应预先联系主账号授权 [AliyunECSImageImportDefaultRole](#) 角色策略获得操作许可。

### 操作步骤

在阿里云管理控制台上导入自定义镜像的步骤如下：

1. 使用OSS的第三方工具客户端、API或者OSS SDK上传制作好的自定义镜像。关于如何上传大于5 GiB的文件，请参阅OSS [分片上传和断点续传](#)。
2. 登录 [ECS管理控制台](#)。
3. 在左侧导航栏里，选择 快照和镜像 > 镜像。
4. 在镜像列表页，单击 导入镜像。
5. 在 导入镜像 对话框里，单击提示信息第3步里的 确认地址。
6. 在 云资源访问授权 窗口，勾选 `AliyunECSImageImportDefaultRole` 和 `AliyunECSExportDefaultRole`，单击 同意授权 授权ECS服务访问您的OSS资源。
7. 在镜像列表页，再次单击 导入镜像。

8. 在 导入镜像 对话框里，完成以下设置：

- 镜像所在地域：上传镜像文件的OSS Bucket所在的地域。
- OSS Object 地址：从OSS控制台获取的镜像文件Object地址。更多详情，请参阅OSS [下载文件](#)。
- 镜像名称：指定自定义镜像文件导入后显示的名称。长度为2-128个字符，以大小写字母或中文开头，可包含数字、点号（.）、下划线（\_）、半角冒号（:）和连字符（-）。
- 操作系统：根据您的镜像操作类型选择 Windows 或 Linux。如果您要导入的是非标准平台自定义镜像，应选择 Linux。
- 系统盘大小：系统盘大小范围为40 GiB-500 GiB。
- 系统架构：64位操作系统选择 x86\_64，32位操作系统选择 i386。
- 系统平台：下拉列表里提供的选项由 操作系统的选项决定，选择适用于您的镜像的系统平台。
  - Windows：支持Windows Server 2003、Windows Server 2008和Windows Server 2012。
  - Linux：支持CentOS、SUSE、Ubuntu、Debian、FreeBSD、CoreOS、Aliyun、Customized Linux和Others Linux（请 [提交工单](#) 确认是否支持）。
  - 如果您的镜像的操作系统是根据Linux内核定制开发的，请 [提交工单](#) 联系我们。
- 镜像格式：仅支持RAW，qcow2和VHD格式，推荐使用传输内容更小的qcow2或者VHD格式。
- 镜像描述：填写镜像描述信息，便于后续管理。
- 添加数据盘镜像：同时导入包含数据盘的自定义镜像，支持的数据盘容量范围为5 GiB-2000 GiB。

9. 信息确认无误后，单击 确定，创建一个导入自定义镜像的任务。

10. (可选) 您可以在您导入地域的镜像列表中查看导入进度。在任务完成前，您都可以通过 [任务管理](#)，找到该导入的自定义镜像，取消导入镜像任务。

您也可以使用ECS API [#unique\\_53](#) 导入自定义镜像。

下一步

[#unique\\_11](#)

参考链接

- [自定义镜像FAQ](#)
- [使用 Packer 创建并导入本地镜像](#)



## 2.4 复制镜像

复制镜像适用于跨地域部署应用，您可以在不同地域的ECS实例中运行同一镜像环境。您可以将某一地域中的一份自定义镜像复制到其他任何阿里云地域，但是任务完成时间取决于网络传输速度和并发任务的排队数量。

### 注意事项

- 复制镜像时，会同时在目标地域生成对应的快照，随后在目标地域根据快照生成自定义镜像。因此，会产生不同地域间的数据传输流量费。目前未对这部分流量收费，具体收费时间以阿里云公告为准。
- 复制镜像后，您会在另一地域获得同样配置的一份自定义镜像。但是相关角色授权和服务授权信息会丢失，也不会包含原来设置过的#unique\_56。

### 操作步骤

在ECS管理控制台上复制镜像的步骤如下：

1. 登录[ECS管理控制台](#)。
2. 选择地域。
3. 在左侧导航栏中，选择快照和镜像 > 镜像。
4. 选中需要复制的镜像，镜像类型必须是自定义镜像，在操作列中，单击复制镜像。



#### 说明：

如果您要复制的自定义镜像大于200 GiB，需要提交工单处理。当您单击复制镜像时，系统会引导您提交工单。

5. 在复制镜像对话框中，您可以看到选中的自定义镜像的ID，您需要完成以下设置：
  - a. 选择目标地域。
  - b. 填写镜像在目标地域显示的自定义镜像名称和自定义镜像描述。
  - c. 单击确定。
6. 切换到目标地域，查看自定义镜像的复制进度。当进度显示为100%时，说明任务完成。



#### 说明：

如果进度没到100%，状态为创建中时，您可以随时点击取消复制。取消任务后，目标地域的镜像列表中不再显示该自定义镜像。

您也可以使用ECS API [CopyImage](#)和[CancelCopyImage](#)完成复制镜像任务。

## 下一步

当镜像的状态变为可用时，您可以使用复制的镜像[创建实例](#)或[#unique\\_62](#)。

您也可以查看在目标地域中查看从源地域复制过来的快照信息。

## FAQ

[复制镜像FAQ](#)

## 2.5 共享镜像

制作了自定义镜像后，您可以将镜像共享给其他阿里云用户使用。共享镜像能明显缩短 ECS 新用户的适应时间，使得新用户能基于您的自定义镜像，快速创建 ECS 实例并建立业务环境。并且共享镜像不会占用被共享者的镜像使用额度。

### 注意事项

您只能共享自己的自定义镜像，无法共享其他用户共享给您的镜像。每份自定义镜像最多可以共享给 50 个用户，并且只能在同一阿里云地域中共享镜像，不支持跨地域共享镜像。

共享镜像之前，请确认自定义镜像已清除敏感数据和文件。



说明：

ECS 无法保证共享镜像的完整性和安全性，使用共享镜像时请确认镜像来自于可信任的共享者或共享账号，并需要自行承担风险。根据共享镜像创建实例后，请务必[连接实例](#)检查镜像的完整性和安全性。

### 删除共享镜像造成的影响

如果您的自定义镜像已经共享给其他账号，您需要预先解除该自定义镜像的全部共享关系后才能删除镜像。删除被共享的自定义镜像后：

- 被共享者无法通过管理控制台或者 ECS API 查询到该镜像，也无法使用该镜像创建 ECS 实例和更换系统盘。
- 被共享者使用共享镜像创建的 ECS 实例无法重新初始化系统盘。

### 共享镜像

在管理控制台上共享镜像的步骤如下：

1. 登录 [ECS 管理控制台](#)。
2. 选择地域。
3. 在左侧导航栏中，选择 快照和镜像 > 镜像。

4. 选择一份 自定义镜像，在 操作 列中，单击 共享镜像。
5. 在共享镜像页面，账号类型 选择 `aliyunUid`，在 账号 对话框中输入待共享的用户 ID。您可以参阅 [附录：如何获取账号 ID](#) 获取 `aliyunUid`。



说明：

您也可以点击 取消共享 解除与该用户的镜像共享关系。取消共享后，该用户将无法查询和使用该镜像；如果该用户已经使用这份共享镜像创建了实例，实例将无法[重新初始化系统盘](#)。

6. （可选）被共享者可以在ECS管理控制台同一地域中的 快照和镜像 > 镜像 > 共享镜像 页面查看共享镜像。

您也可以使用 ECS API [ModifyImageSharePermission](#) 和 [DescribeImageSharePermission](#) 完成共享镜像。

#### 下一步

共享镜像后，被共享者可以使用该镜像创建一台或多台实例。

1. 登录 [ECS 管理控制台](#)。
2. 参阅 [快速入门步骤 2：创建ECS实例](#) 创建一台或多台实例，并在选择镜像时选择 共享镜像。

被共享者还可以使用该镜像为实例 [#unique\\_62](#)。

#### 附录：如何获取账号 ID

您可以通过如下步骤查询您的账户ID。

1. 登录阿里云管理控制台。
2. 鼠标移至右上角用户头像，如 `funCustomer*****@aliyun.com`，在弹出的账号菜单中单击 安全设置。
3. 找到账号ID。

## 2.6 导出镜像

导出ECS自定义镜像适用于搭建本地测试环境，或者为私有云环境提供相同环境的场景。



说明：

- 导出自定义镜像是相对耗时的任务，需要您耐心等待。完成的时间取决于自定义镜像文件的大小和当前导出任务的并发数。
- 导出自定义镜像需要使用[对象存储OSS](#)，所以会产生一定的OSS存储和下载的流量费用。更多详情，请参见《OSS文档》[计量项和计费项](#)。

## 限制条件

您需要注意如下限制条件：

- 不支持导出通过[云市场镜像](#)的系统盘快照创建的自定义镜像。
- 导出的自定义镜像中可以包含数据盘快照，但数据盘不能超过4块，单块数据盘容量最大不能超过500 GiB。
- 使用导出的全镜像#*unique\_72*时，您需要自行确认`/etc/fstab`记录的文件设备是否与导出的数据盘快照信息互相对应。

## 前提条件

在导出自定义镜像前，您应该已经完成如下工作：

- [提交工单](#)申请导出权限，并说明导出自定义镜像的使用场景。
- 请确保您已经开通OSS服务，而且自定义镜像所在地域里有可用的OSS Bucket。更多详情，请参见《OSS文档》[创建存储空间](#)。

## 操作步骤

在ECS管理控制台上导出一份自定义镜像的步骤如下：

1. 登录[ECS管理控制台](#)。
2. 在左侧导航栏里，选择快照和镜像 > 镜像。
3. 选择地域。
4. 找到需要导出的自定义镜像，在操作列中，单击导出镜像。
  - a. 在导出镜像对话框里，单击提示信息里的确认地址。
  - b. 在云资源访问授权窗口，单击同意授权授权ECS服务访问您的OSS资源。
5. 回到ECS管理控制台首页，在自定义镜像列表页操作列中，再次单击导出镜像。
6. 在导出镜像对话框中：
  - 选择一个与自定义镜像所属地域相同的OSS Bucket。
  - 为自定义镜像的Object名称设置一个前缀。例如，您可以将Demo设为前缀，则导出的自定义镜像文件，在OSS Bucket中的名称即为*Demo-[#####]*。
7. 单击确定开始导出自定义镜像。

8. （可选）取消导出自定义镜像任务。在任务完成前，您都可以通过[任务管理](#)，在相应地域下找到导出自定义镜像的任务，取消导出自定义镜像。

您也可以使用ECS API [ExportImage](#)和[CancelTask](#)导出自定义镜像。

#### 下一步

导出的自定义镜像包含数据盘快照时，您的OSS中会出现多个文件。文件名带有`system`的表示系统盘快照，文件名带有`data`的表示数据盘快照。数据盘快照会有与数据盘对应的标识，即磁盘的挂载点，如`xvdb`或者`xvdc`。

1. 登录[OSS管理控制台](#)查询导出自定义镜像的结果。
2. 参见《[OSS文档](#)》[下载自定义镜像文件](#)。



#### 说明:

目前镜像文件的导出格式默认为`.raw.tar.gz`，解压后为`.raw`格式。如果您的本地计算机为Mac OS X系统，推荐您使用`gnu-tar`解压工具。

## 3 云市场镜像

---

镜像，相当于云服务器ECS的装机盘。您可以选择自行配置环境和安装软件，也可以购买云市场镜像快速获得预装的云服务器运行环境或软件应用，满足建站、应用开发、可视化管理等个性化需求，让ECS即开即用，省时方便。

### 云市场镜像的优点

不管是创建实例时，还是创建实例后，您都可以切换使用云市场镜像。云市场镜像部署相比自定义部署，有如下优点：

### 创建实例时选择云市场镜像

如果您是新购云服务器的用户，推荐您使用云市场镜像。

1. 前往 [ECS购买页](#)。
2. 参阅 [创建ECS实例](#) 完成选型配置，在 镜像 配置项处，选择 镜像市场 > 从镜像市场选择（含操作系统）。
3. 您可以从 镜像市场 > 精选镜像 页面选择系统为您推荐的最热门镜像以及您可能喜欢的操作系统及应用软件。
4. 如果没有找到需要的镜像，您可以点击 镜像分类 并从相应分类中选择合适的镜像。镜像市场提供了操作系统、运行环境、管理监控、建站系统、应用开发、数据库、服务器软件、企业应用和云安全10类镜像供您选择。
5. 您还可以通过搜索镜像关键字，或按操作系统和架构筛选出适合您的镜像。
6. 选择您需要的镜像，完成 [创建ECS实例](#) 所需的其他配置后，确认订单 并 创建实例。

### 在云市场选购镜像后创建实例

1. 前往 [云市场](#)。
2. 选择您需要的镜像环境并单击 立即购买。
3. 如果您未登录阿里云控制台，需要登录后才能成功跳转。
4. 参阅 [创建ECS实例](#) 完成选型配置。

## 使用云市场镜像更换操作系统

若您已购买了云服务器，需要使用镜像来部署运行环境，或者安装软件应用，操作如下：



说明：

更换镜像会导致系统盘数据丢失，建议您在更换操作系统前备份数据。更多详情，请参阅 [#unique\\_79](#)。

1. 登录 [ECS管理控制台](#)。
2. 停止目标实例。
3. 参阅 [#unique\\_62](#)，并在更换系统盘 页面的 镜像类型 处选择 镜像市场，即可使用所需的镜像。

## 4 镜像相关开源工具

---



## 5 公共镜像

---

## 6 自定义镜像

---