

腾讯云云服务器

镜像

产品文档



腾讯云

【版权声明】

©2013-2017 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

文档声明.....	2
镜像.....	4
镜像概述.....	4
镜像类型.....	7
创建自定义镜像.....	10
复制镜像.....	12
共享自定义镜像.....	14
取消共享自定义镜像.....	16
删除自定义镜像.....	17
Linux 操作系统安装 cloud-init.....	18
强制导入镜像.....	25
导入镜像.....	33
Linux导入镜像检查virtio驱动.....	34

镜像

镜像概述

什么是镜像？

腾讯云镜像提供启动云服务器实例所需的所有信息。指定需要的镜像后可以从该镜像启动所需任意数量的实例，也可以根据需求从任意多个不同的镜像启动实例。通俗的说，镜像就是云服务器的“装机盘”。

镜像类型

腾讯云提供的镜像包括以下几种：

- 公有镜像：所有用户均可使用，涵盖大部分主流操作系统；
- 服务市场镜像：所有用户均可使用，除操作系统外还集成了某些特定应用程序；
- 自定义镜像：仅创建者和共享对象可以使用，由现有运行的实例创建而来或由外部导入而来；
- 共享镜像：由其他用户共享而来的镜像，仅能用作创建实例。

更多镜像类型介绍详见 [镜像类型](#)。

镜像部署 VS 手动部署

	镜像部署	手动部署
部署时长	3 - 5 分钟	1 - 2 天
部署过程	根据成熟的服务市场方案或已使用过的方案，快速创建合适的云服务器。	选择合适的操作系统、数据库、应用软件、插件等，并需要安装和调试。
安全性	除共享镜像来源需要用户自行甄别，其他公共镜像、自定义镜像、服务市场镜像都经过腾讯云测试和审核。	依赖开发部署人员的水平。
适用情况	公共镜像：正版操作系统，包含腾讯云提供的初始化组件； 服务市场镜像：快速依据成熟的搭建方案搭建个性化应用环境； 自定义镜像：快速创建跟已有云服	完全自行配置，无基础设置。

	镜像部署	手动部署
	务器相同软件环境，或进行环境备份 ； 共享镜像：快速创建跟其他用户已有云服务器相同软件环境。	

镜像应用

- 部署特定软件环境

使用共享镜像、自定义镜像、服务市场镜像都能帮助快速搭建特定的软件环境，免去了自行配置环境、安装软件等繁琐且耗时的的工作，并能满足建站、应用开发、可视化管理等多种个性化需求，让云服务器“即开即用”，省时方便。

- 批量部署软件环境

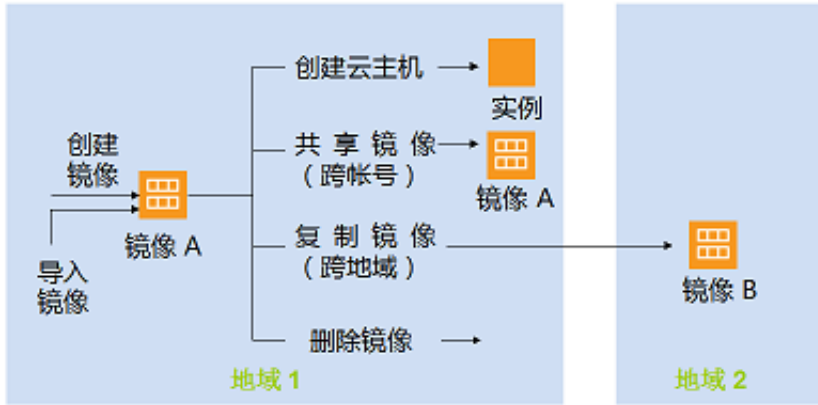
通过对已经部署好环境的云服务器实例制作镜像，然后在批量创建云服务器实例时使用该镜像作为操作系统，云服务器实例创建成功之后便具有和之前云服务器实例一致的软件环境，以此达到批量部署软件环境的目的。

- 服务器运行环境备份

对一台云服务器实例制作镜像备份运行环境。若该云服务器实例使用过程中因软件环境被损坏而无法正常运行，则可以使用镜像恢复。

镜像生命周期

下图总结了自定义镜像的生命周期。创建或导入了一个新自定义镜像之后，用户可以将其用于启动新实例（用户也可从现有的公共镜像或服务市场镜像启动实例）。自定义镜像可以被复制到同帐号的其他地域下，成为该地域下独立的镜像。用户还可以将自定义镜像共享给其他用户。



自定义镜像的生命周期

镜像类型

用户可以基于以下特性选择镜像：

- 位置（请参阅 [地域和可用区](#)）
- 操作系统类型
- 架构（32 位或 64 位）

根据不同来源，腾讯云提供镜像类型有：公共镜像、自定义镜像、共享镜像、服务市场镜像。

公共镜像

公共镜像是由腾讯云官方提供、支持和维护的镜像，包含基础操作系统和腾讯云提供的初始化组件，所有用户均可使用。

公共镜像特质：

- 操作系统类型：自由选择（如：基于 Linux 类型系统或 Windows 类型系统），并定期更新。
- 软件支持：集成腾讯云提供的软件包（如 API 等），并支持多版本的 Java、MySQL、SQL Server、Python、Ruby、Tomcat 等常见软件及其完全权限。
- 安全：
提供的操作系统完全合法合规，均使用官方正版操作系统。腾讯云内部专业安全运维团队制作，经过严格测试并可选内置腾讯云安全组件。
- 限制：暂无使用限制。
- 费用：除部分海外地域的 Windows 类型镜像需收取一定 License 费用外，其余全部免费。

服务市场镜像

[服务市场镜像](#)是由第三方服务商提供的镜像。经过腾讯云审核后发布，所有用户均可使用。

服务市场镜像特质：

- 操作系统类型：多种常规的操作系统等预安装环境。
- 软件支持：集成了如 PHP、Java、FTP、Nginx、Docker、WordPress、Discuz

等常用的热门软件环境。

- 安全：

入驻服务商均经过腾讯云严格甄选并签订入驻协议，镜像均经过服务商与腾讯云严格测试，保证镜像内容的安全性。

- 限制：暂无使用限制。

- 费用：

费用由镜像提供商定价，既有免费镜像，也有付费镜像。使用付费镜像启动实例时，费用包含实例费用和镜像费用两部分。

自定义镜像

自定义镜像 是用户通过镜像制作功能制作，或通过镜像导入功能导入的镜像。仅创建者与共享者可以使用。

自定义镜像特质：

- 应用场景：对一个已经部署好应用的云服务器实例创建镜像，以此快速创建更多包含相同配置的实例。
- 功能支持：支持用户自由创建、复制、分享和销毁。
- 限制：每个地域下最多支持 10 个自定义镜像。
- 费用：
创建可能涉及费用，具体的价格以创建实例时显示的价格为准。跨地域复制自定义镜像暂不收费。

更多操作方法与限制，请参阅 [创建自定义镜像](#)、[复制自定义镜像](#)、[共享自定义镜像](#)、[取消共享自定义镜像](#)、[导入自定义镜像](#)。

共享镜像

共享镜像 是由其他腾讯云用户通过镜像共享功能，将其自定义镜像共享给当前用户。
被共享的镜像将在被共享用户的原镜像相同地域下展示。

共享镜像特质：

- 应用场景：帮助其他用户快速创建云主机。
- 功能支持：共享镜像仅可用于创建云主机，不可以进行修改名称、复制、共享等其他操作。
- 安全：共享的镜像不经过腾讯云审核，可能存在安全风险。因此，强烈建议不要接受未知来源镜像。

- 限制：每个自定义镜像最多可共享给 50 个腾讯云用户。镜像共享仅支持共享到对方账户相同地域下。

更多操作方法与限制，请参阅 [共享自定义镜像](#)、[取消共享自定义镜像](#)。

创建自定义镜像

创建概述

常规思路

您可以先从一个公共镜像或服务市场镜像中启动实例，然后连接到您的实例并自行部署软件环境。实例正常运行的情况下，您可以根据自己的需要在其基础上创建新的自定义镜像，之后您可以使用这个镜像启动更多与原实例具有相同自定义项的新实例。

最佳实践

- 关机实例：
制作自定义镜像时，请先将实例关机，以保证镜像与当前实例部署环境完全一致。
- 数据迁移：
如果您需要在启动新实例时同时保留原有实例数据盘上的数据，则您可以先对数据盘做 [快照](#)，在启动新实例时使用该数据盘快照创建新的 CBS 数据盘。有关更多信息，请参阅 [快照创建云硬盘](#)。

创建限制

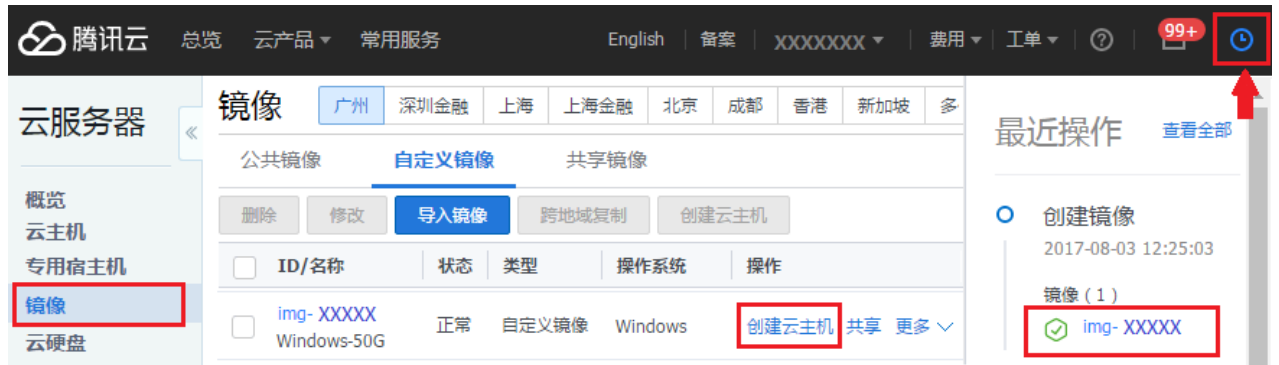
- 每个地域暂支持 10 个自定义镜像。
- 每个用户每天操作次数不能超过 5 次（含 API 操作，按每个地域计数）。

创建方法

使用控制台从实例创建

1. 登录 [云主机控制台](#)。
2. 关机实例。勾选需要关机的实例，单击上方【关机】。
3. 在需要制作镜像的实例右侧单击【更多】，单击【制作镜像】。
4. 在弹出框中，输入“镜像名称”和“镜像描述”，单击【确定】提交创建。
5. 鼠标移至控制台右上角【最近操作（时钟图标）】，即可查看创建进度。
6. 创建成功后，单击左侧导航栏【镜像】，或单击【最近操作（时钟图标）】中镜像 ID，即可跳转至镜像列表查看详细信息。

7. 若需要购买与之前相同镜像的服务器。在镜像列表中，镜像右侧单击【创建云主机】。



使用 API 创建

用户可以使用 CreateImage 接口创建自定义镜像，具体内容可以参考 [创建镜像 API](#)。

复制镜像

复制概述

常规思路

复制镜像功能 帮助用户快速 跨地域

部署相同的云服务器实例。您可以跨地域复制镜像，然后通过复制在新地域下的镜像创建云服务器。

复制说明

- 复制的镜像须为自定义镜像，需先创建自定义镜像，操作方法详见 [创建自定义镜像](#)。
- 复制镜像支持国内至国内复制（暂不支持香港），国外至国外复制，需要从国内复制镜像到国外，或者从国外复制镜像到国内请联系售后。
- 跨地域复制镜像暂不收费。
- 大于 50GB 的自定义镜像暂不支持跨地域复制。
- 复制镜像等待时间为 10 分钟到 30 分钟。

复制方法

使用控制台复制

1. 登录 [云主机控制台](#)。
2. 单击左侧导航窗格中的【镜像】。
3. 选择需要被复制的原始镜像地域，如：广州。
4. 单击头部【自定义镜像】选项卡。
5. 在需要被复制镜像的右侧单击【更多】，单击【跨地域复制】（若需要批量操作，可勾选所有需要被复制的镜像，单击顶部【跨地域复制】）。
6. 在弹框中选择要复制的目的地域，单击【确定】。
7. 复制成功后，目的地域的镜像列表将能显示名称相同的镜像（ID 不同）。
8. 在镜像列表中，镜像右侧单击【创建云主机】，即可创建相同的云服务器实例。

使用 API 复制

用户可以使用 SyncCvmImage 接口复制镜像，具体内容可以参考 [复制镜像 API](#)。

共享自定义镜像

概述

共享镜像 是将自己已经创建好的 自定义镜像 共享给 其他用户 使用。用户可以方便地从其他用户那里获得共享镜像，并从中获得需要的组件及添加自定义内容。

注意：

腾讯云无法保证其他用户共享镜像的完整性或安全性，我们建议用户只使用来自可靠来源的共享镜像。

限制

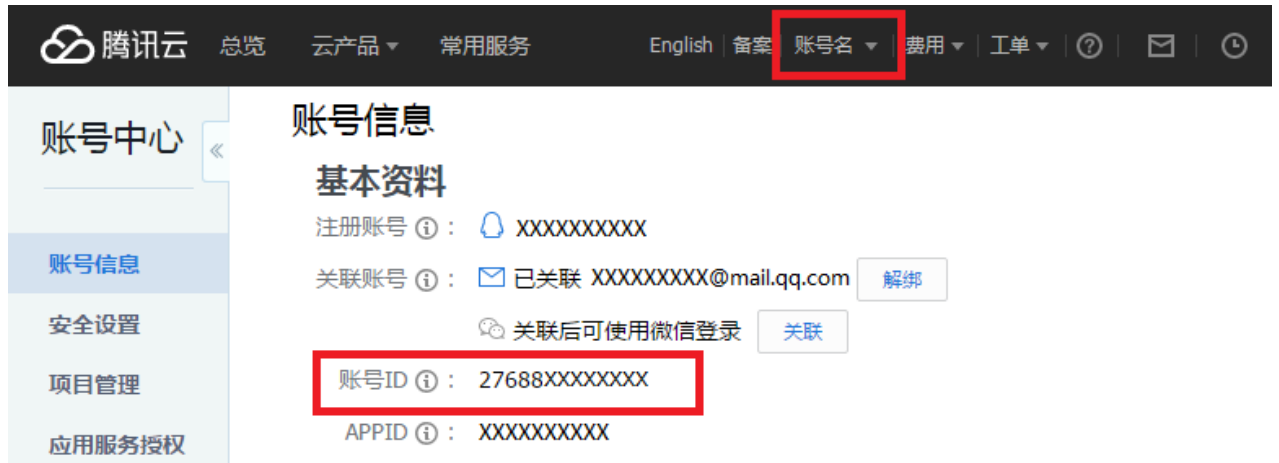
- 每个镜像最多可以共享给 50 个用户。
- 共享镜像不能更改名称和描述，仅可用于创建云服务器实例。
- 共享给其他用户的镜像不占用自身镜像配额。
- 共享给其他用户的镜像可以删除，但需先取消该镜像所有的共享，取消共享操作详见 [取消共享自定义镜像](#)。获取的共享镜像不可删除。
- 镜像支持共享到对方账户相同地域内；若需共享到不同地域，需先复制镜像到不同地域再进行共享。已共享镜像
- 不可将获取的镜像共享给其他用户。

操作

前提条件

腾讯云共享镜像通过对端账号唯一 ID 识别。获取对方用户的帐号唯一 ID 。可以通知该用户通过以下方式获取：

1. 登录 [云主机控制台](#) ，单击右上角账号名称。
2. 查看账号信息中的账号 ID 。
3. 通知对方将账号 ID 发送给自己。



通过控制台共享

1. 登录 [云主机控制台](#)。
2. 单击左侧导航窗格中的【镜像】。
3. 单击顶部【自定义镜像】选项卡。
4. 在列表中选中您要共享的自定义镜像，单击右侧【共享】。
5. 在弹出框中输入对方账户 ID，单击【共享】。
6. 通知对方登录 [云主机控制台](#)，并选择【镜像】 - 【共享镜像】，即可查看共享给其的镜像。
7. 若需要共享给多个用户，请重复上述步骤。

通过 API 共享

用户可以使用 ShareImage 接口共享镜像，具体内容可以参考 [共享镜像 API](#)。

取消共享自定义镜像

用户可以随时终止共享给其他人镜像的共享状态，从而决定不再共享给某个其他用户。此操作不会影响其他用户已经使用这个共享镜像创建的实例，但其他用户无法再查看此镜像，也无法使用此镜像创建更多实例。

通过控制台取消共享镜像

1. 登录 [云主机控制台](#)。
2. 单击左侧导航窗格中的【镜像】。
3. 单击【自定义镜像】选项卡。
4. 在列表中您需要取消共享的自定义镜像条目后，选择【更多】 - 【取消共享】按钮。
5. 在新页面中，在已共享对方帐号右侧单击【取消共享】，确认后即可完成取消镜像的共享。

通过 API 取消共享镜像

用户可以使用 CancelShareImage 接口取消共享镜像，具体内容可以参考 [取消共享镜像 API](#)。

删除自定义镜像

注意事项

使用完自定义镜像后，用户可以选择删除。执行删除操作前，请您注意以下事项：

- 删除自定义镜像后，无法通过此镜像创建实例，但不影响已启动的实例。如果您需要删除所有从此镜像启动的实例，可参考 [回收实例](#) 或 [销毁实例](#)。
- 已共享的镜像无法删除，需要先取消所有共享后方可删除，取消共享操作详见 [取消共享自定义镜像](#)。
- 仅自定义镜像能被删除，公共镜像和共享镜像均无法主动删除，获取的镜像亦不可删除。

通过控制台删除

1. 登录 [云主机控制台](#)。
2. 单击左侧导航窗格中的【**镜像**】。
3. 单击【**自定义镜像**】选项卡。
4. 删除单个镜像：列表右侧单击【**更多**】-【**删除**】，确认删除操作即可。
5. 删除多个镜像：列表中勾选所有要删除的镜像，单击顶部【**删除**】按钮，确认删除操作后即可删除所有选中镜像。

无法删除时会提示原因。

通过 API 删除

用户可以使用 DeleteImages 接口共享镜像，具体内容可以参考 [删除镜像 API](#)。

Linux 操作系统安装 cloud-init

cloud-init 主要提供了一个实例首次初始化时的自定义配置的能力，如果导入的镜像没有安装cloud-init服务，通过该镜像启动的实例就不会被正常初始化，因此该镜像就会导入失败。

安装cloud-init的方式有两种，[手工下载cloud-init源码包的方式安装](#) 和 [直接使用软件源上面的 cloud-init 包安装](#)。在导入Linux系统镜像前，请确保您的镜像内部已经按照如下方式之一正确安装了cloud-init服务。

一、手工下载cloud-init源码包的方式安装

1、前提条件

- 安装cloud-init的服务器外网能通

2、下载cloud-init源码包

1) 下载地址：<https://launchpad.net/cloud-init/+download>

2) 建议安装版本：cloud-init-17.1.tar.gz（注：cloud-init-17.1 版本与腾讯云的兼容性最好，在正常安装的情况下可以保证使用该镜像创建出来的云服务器所有配置项都能够正常初始化。）

3、安装cloud-init

3.1、安装cloud-init 依赖包

```
setuptools
jinja2
prettytable
oauthlib
configobj
pyyaml
requests
jsonpatch
jsonschema
```

six

3.2、解压并安装

```
cd ./cloud-init-17.1
python setup.py build
python setup.py install --init-system systemd
```

注：--init-system的可选参数有：(systemd, sysvinit, sysvinit_deb, sysvinit_freebsd, sysvinit_openrc, sysvinit_suse, upstart) [default: None]，需要根据当前操作系统使用的自启动服务管理方式是什么进行选择，如果选择出错则cloud-init服务无法开机自启动，本例以systemd自启动服务管理为例。

3.3、修改cloud-init 配置文件

根据不同操作系统，从以下链接下载 cloud.cfg 将 /etc/cloud/cloud.cfg 的内容进行替换。

- [ubuntu 操作系统的 cloud.cfg](#)
- [centos 操作系统的 cloud.cfg](#)
- 其他操作系统待补充

3.4、添加 syslog 用户

```
useradd syslog
```

3.5、设置 cloud-init 服务开机自启动

3.5.1、若操作系统是systemd自启动管理服务

ubuntu 或 debian 操作系统特殊执行

```
ln -s /usr/local/bin/cloud-init /usr/bin/cloud-init
```

所有操作系统都执行

```
systemctl enable cloud-init-local.service
systemctl start cloud-init-local.service
systemctl enable cloud-init.service
systemctl start cloud-init.service
systemctl enable cloud-config.service
systemctl start cloud-config.service
systemctl enable cloud-final.service
systemctl start cloud-final.service
systemctl status cloud-init-local.service
systemctl status cloud-init.service
systemctl status cloud-config.service
systemctl status cloud-final.service
```

centos 和 redhat 操作系统特殊执行

将 /lib/systemd/system/cloud-init-local.service 文件内容替换为如下：

```
[Unit]
```

```
Description=Initial cloud-init job (pre-networking)
```

```
Wants=network-pre.target
After=systemd-remount-fs.service
Before=NetworkManager.service
Before=network-pre.target
Before=shutdown.target
Conflicts=shutdown.target
RequiresMountsFor=/var/lib/cloud
```

```
[Service]
```

```
Type=oneshot
ExecStart=/usr/bin/cloud-init init --local
ExecStart=/bin/touch /run/cloud-init/network-config-ready
RemainAfterExit=yes
TimeoutSec=0
```

```
# Output needs to appear in instance console output
```

```
StandardOutput=journal+console
```

```
[Install]
```

```
WantedBy=cloud-init.target
```

将 `/lib/systemd/system/cloud-init.service` 文件内容替换为如下：

```
[Unit]
```

```
Description=Initial cloud-init job (metadata service crawler)
Wants=cloud-init-local.service
Wants=sshd-keygen.service
Wants=sshd.service
After=cloud-init-local.service
After=systemd-networkd-wait-online.service
After=networking.service
After=systemd-hostnamed.service
Before=network-online.target
Before=sshd-keygen.service
Before=sshd.service
```

```
Before=systemd-user-sessions.service
```

```
Conflicts=shutdown.target
```

```
[Service]
```

```
Type=oneshot
```

```
ExecStart=/usr/bin/cloud-init init
```

```
RemainAfterExit=yes
```

```
TimeoutSec=0
```

```
# Output needs to appear in instance console output
```

```
StandardOutput=journal+console
```

```
[Install]
```

```
WantedBy=cloud-init.target
```

3.5.2、若操作系统是sysvinit自启动管理服务

```
chkconfig --add cloud-init-local
```

```
chkconfig --add cloud-init
```

```
chkconfig --add cloud-config
```

```
chkconfig --add cloud-final
```

```
chkconfig cloud-init-local on
```

```
chkconfig cloud-init on
```

```
chkconfig cloud-config on
```

```
chkconfig cloud-final on
```

二、直接使用软件源上面的 cloud-init 包安装

执行以下安装命令安装

```
apt-get/yum install cloud-init
```

注：直接通过apt-get 或 yum 命令安装的cloud-init

版本默认为当前操作系统配置的软件源里面默认的cloud-init版本，通常情况下和cloud-init 17.1 版本存会存在比较大的差异，使用这种方式安装的镜像创建出来的实例可能会存在部分配置项初始化不符合预期的情况，建议使用方案一：手工下载cloud-init源码包的方式进行安装。

修改cloud-init 配置文件

根据不同操作系统，从以下链接下载 cloud.cfg 将 /etc/cloud/cloud.cfg 的内容进行替换。

- [ubuntu 操作系统的 cloud.cfg](#)
- [centos 操作系统的 cloud.cfg](#)
- 其他操作系统待补充

三、安装完之后的操作

注：以下操作执行完成后请勿重启服务器，否则需重新执行下以下操作。

```
cloud-init init --local  
rm -rf /var/lib/cloud
```

ubuntu 或 debian 操作系统特殊操作

```
rm -rf /etc/network/interfaces.d/50-cloud-init.cfg
```

修改 /etc/network/interfaces 为如下内容：

```
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).
```

```
source /etc/network/interfaces.d/*
```


强制导入镜像

当用户的linux镜像因为某些原因无法[安装cloudinit](#)时，可以使用

强制导入镜像

功能完成镜像的导入。此时，腾讯云无法对用户的虚拟机进行初始化配置，因此需要用户设置脚本自行根据腾讯云提供的配置文件对虚拟机进行配置。

限制条件及配置方法

申请权限

在您使用本功能前，请确保您已经开通了导入镜像权限。若您需要开通权限，请联系商务经理，并提交相关信息至工单系统申请。

镜像限制条件

- 镜像仍需要满足[导入镜像](#)中关于linux镜像导入的镜像的限制（cloudinit除外）。
- 导入镜像的系统分区未满。
- 导入的镜像不得存在可以被远程利用的漏洞。
- 建议用户用强制导入镜像创建实例成功后立即修改密码。

镜像导入配置方法

用户强制导入的镜像并没有使用cloudinit，因此无法自动进行配置。腾讯云提供了包含了配置信息的cdrom设备供用户自行配置，用户需要挂载cdrom，然后读取

```
mount_point/os_action/os.conf
```

的信息进行配置。如果用户有使用其他配置数据、UserData的需要，可以直接读取

```
mount_point/
```

下的文件。

os.conf配置文件内容

os.conf的基本内容如下：

```
hostname=VM_10_20_xxxx
password=GRSgae1fw9frsG.rfrF
eth0_ip_addr=10.104.62.201
eth0_mac_addr=52:54:00:E1:96:EB
eth0_netmask=255.255.192.0
eth0_gateway=10.104.0.1
dns_nameserver="10.138.224.65 10.182.20.26 10.182.24.12"
```

以上信息仅参数名有参考意义，参数值仅做示例。

其中各个参数的意义如下：

参数名称	参数意义
hostname	主机名
password	加密过的密码
eth0_ip_addr	eth0网卡的局域网IP
eth0_mac_addr	eth0网卡的MAC地址
eth0_netmask	eth0网卡的子网掩码
eth0_gateway	eth0网卡的网关
dns_nameserver	dns解析服务器

配置脚本解析

注意事项

- 脚本应该开机自动执行。
- 脚本挂载

/dev/cdrom

, 然后读取挂载点下的

```
os_action/os.conf
```

文件获取配置信息。

- 腾讯云放置到cdrom中的密码为加密后的密码，用户可以使用

```
chpasswd -e
```

的方式设置。注意，加密后的密码可能包含特殊字符，建议放置到文件中后以

```
chpasswd -e < passwd_file
```

的方式设置。

- 使用强制导入镜像制作的实例再制作镜像时，需要保证脚本依然会被执行，以保证实例正确配置。或者在实例中安装cloudinit。

示例

腾讯云提供一份基于centos的示例脚本，用户可以根据示例脚本制作针对自己镜像的配置脚本，需要注意：

- 该脚本需要在导入镜像前正确放置到系统中。
- 该脚本不适合所有操作系统，用户需要根据自己的操作系统进行相对修改已满足语义。
- 该脚本需要设置为开机启动才能正确配置，请根据操作系统的类型实现该要求。（例如将脚本

```
os_config
```

放置到/etc/init.d/目录下 然后执行以下命令。)

```
chmod +x /etc/init.d/os_config
```

```
chkconfig --add os_config
```

然后通过

```
chkconfig --list
```

检查

```
os_config
```

是否已经被添加到启动服务中。

- 用户需要自行保证脚本执行正确，如果镜像导入后遇到无法ssh连接实例，没有网络连接等问题首先尝试通过控制台连接到实例，重新执行脚本排查问题，如仍然无法处理，请联系客服。

以下为

```
os_config
```

脚本示例，用户可根据实际情况修改脚本。

```
#!/bin/bash
### BEGIN INIT INFO
# Provides: os-config
# 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: config of os-init job
# Description: run the config phase without cloud-init
### END INIT INFO

#####user settings#####

cdrom_path='/dev/cdrom'

load_os_config() {
```

```
mount_path=$(mktemp -d /mnt/tmp.XXXX)
mount /dev/cdrom $mount_path
if [[ -f $mount_path/qcloud_action/os.conf ]]; then
. $mount_path/qcloud_action/os.conf
if [[ -n $password ]]; then
passwd_file=$(mktemp /mnt/pass.XXXX)
passwd_line=$(grep password $mount_path/qcloud_action/os.conf)
echo root:${passwd_line#*=} > $passwd_file
fi
return 0
else
return 1
fi
}

cleanup() {
umount /dev/cdrom
if [[ -f $passwd_file ]]; then
echo $passwd_file
rm -f $passwd_file
fi
if [[ -d $mount_path ]]; then
echo $mount_path
rm -rf $mount_path
fi
}

config_password() {
if [[ -f $passwd_file ]]; then
chpasswd -e < $passwd_file
fi
}
```

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

```
config_network() {
  /etc/init.d/network stop
  cat << EOF > /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
IPADDR=$eth0_ip_addr
NETMASK=$eth0_netmask
HWADDR=$eth0_mac_addr
ONBOOT=yes
GATEWAY=$eth0_gateway
BOOTPROTO=static
EOF
  if [[ -n $hostname ]]; then
    sed -i "/^${eth0_ip_addr}.*d" /etc/hosts
    echo "${eth0_ip_addr} $hostname" >> /etc/hosts
  fi
  /etc/init.d/network start
}
```

```
config_gateway() {  
    sed -i "s/^GATEWAY=.*GATEWAY=$eth0_gateway" /etc/sysconfig/network  
}
```

```
#####init#####
```

```
start() {  
    if load_os_config ; then  
        config_password  
        config_hostname  
        config_dns  
        config_network  
        cleanup  
        exit 0  
    else  
        echo "mount ${cdrom_path} failed"  
        exit 1  
    fi  
}
```

```
RETVAL=0
```

```
case "$1" in  
    start)  
        start  
        RETVAL=$?  
        ;;  
    *)  
        echo "Usage: $0 {start}"  
        RETVAL=3  
        ;;  
esac
```

```
exit $RETVAL
```


导入镜像

除了使用 [创建自定义镜像](#)

功能外，腾讯云同时支持使用导入功能。可将本地或其他平台的服务器系统盘镜像文件导入至 CVM 自定义镜像中。导入后可以使用该导入镜像创建云服务器或对已有云服务器重装系统。

导入准备

申请权限

在您使用本功能前，请确保您已经开通了导入镜像权限。若您需要开通权限，请联系商务经理，并提交相关信息至工单系统申请。

准备镜像文件

您需提前准备好符合导入限制的镜像文件。

- Linux 系统类型镜像限制：

镜像属性	条件
操作系统	

Linux导入镜像检查virtio驱动

云主机系统内核需要支持virtio驱动（包括块设备驱动

virtio_blk

和网卡驱动

virtio_net

）才能在腾讯云上正常运行，对于未编译进内核的

virtio_blk

驱动，还需要包含在

initramfs(或者initrd)

文件中，云主机才能正常工作，本文将说明导入镜像前如何检查以及修复镜像中对 virtio 驱动的支持。

内核支持virtio驱动检查

以

Centos7

为例详细说明如何确定当前内核是否支持

virtio

驱动

（1）确认当前内核是否支持

virtio

驱动

```
grep -i virtio /boot/config-$(uname -r)
```

如下图所示：当前内核包含了

virtio_blk

和

virtio_net

驱动，并且是以模块形式编译的（

```
CONFIG_VIRTIO_BLK=m
```

，表示编译成为内核模块，等于y表示编译进内核），如果这一步没有找到

virtio_net

或

virtio_blk

的驱动信息，那么该镜像 不支持 导入腾讯云。

```
[root@centos7 ~]# grep -i virtio /boot/config-$(uname -r)
CONFIG_VIRTIO_BLK=m
CONFIG_SCSI_VIRTIO=m
CONFIG_VIRTIO_NET=m
CONFIG_VIRTIO_CONSOLE=m
CONFIG_HW_RANDOM_VIRTIO=m
CONFIG_VIRTIO=m
# Virtio drivers
CONFIG_VIRTIO_PCI=m
CONFIG_VIRTIO_PCI_LEGACY=y
CONFIG_VIRTIO_BALLOON=m
CONFIG_VIRTIO_INPUT=m
# CONFIG_VIRTIO_MMIO is not set
```

如果内核支持

virtio

驱动 (

virtio_blk

和

virtio_net

都支持) , 且

virtio_blk

驱动编译进入了内核 (即

CONFIG_VIRTIO_BLK=y

) , 则该内核支持导入 , 不需要后续确认 , 如果

virtio_blk

驱动是编译成内核模块的 (即

CONFIG_VIRTIO_BLK=m

)，则需要继续后续确认步骤，确认

virtio_blk

驱动正确包含进了

initramfs (或initrd)

文件中。

(2) 确认

initramfs

中是否包含

virtio_blk

驱动

```
lsinitrd /boot/initramfs-$(uname -r).img | grep virtio
```

如下图所示，

initramfs

中包含了

virtio_blk

驱动，以及其所依赖的

virtio.ko

、

virtio_pci.ko

、

virtio_ring.ko

，这样

initramfs

包含驱动正常，该镜像可以导入。

```
[root@centos7 ~]# lsinitrd /boot/initramfs-$(uname -r).img | grep virtio
-rw-r--r-- 1 root root 27885 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/block/virtio_blk.ko
-rw-r--r-- 1 root root 53533 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/char/virtio_console.ko
-rw-r--r-- 1 root root 49605 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/net/virtio_net.ko
-rw-r--r-- 1 root root 29253 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/scsi/virtio_scsi.ko
drwxr-xr-x 2 root root 0 Nov 10 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/virtio
-rw-r--r-- 1 root root 17989 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/virtio/virtio.ko
-rw-r--r-- 1 root root 35461 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/virtio/virtio_pci.ko
-rw-r--r-- 1 root root 22757 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/virtio/virtio_ring.ko
[root@centos7 ~]#
```

(3) 如果

initramfs

中未找到相关的

virtio

信息，则需要重新制作

initramf

s文件

1) CentOS 7 操作方法

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

```
mkinitrd -f --with=virtio_blk --with=virtio_pci /boot/initramfs-$(uname -r).img $(uname -r)
```

```
[root@centos7 ~]# lsinitrd /boot/initramfs-$(uname -r).img | grep virtio
Arguments: -f --add-drivers ' virtio-blk'
-rw-r--r-- 1 root root 27885 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/block/virtio_blk.ko
-rw-r--r-- 1 root root 53533 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/char/virtio_console.ko
-rw-r--r-- 1 root root 49605 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/net/virtio_net.ko
-rw-r--r-- 1 root root 29253 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/scsi/virtio_scsi.ko
drwxr-xr-x 2 root root 0 Jun 22 16:48 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/virtio
-rw-r--r-- 1 root root 17989 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/virtio/virtio.ko
-rw-r--r-- 1 root root 35461 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/virtio/virtio_pci.ko
-rw-r--r-- 1 root root 22757 Oct 25 2016 usr/lib/modules/3.10.0-327.36.3.el7.x86_64/kernel/drivers/virtio/virtio_ring.ko
[root@centos7 ~]#
```

2) Redhat5/Centos5 操作方法

a. 通过如下方式确定initrd文件中是否包含驱动信息

```
mkdir -p /tmp/initrd && cd /tmp/initrd
```

```
zcat /boot/initrd-$(uname -r).img | cpio -idmv
```

```
find . -name "virtio*"
```

b. 如果需要重新制作

```
initrd
```

文件，执行以下命令

```
cp /boot/initrd-$(uname -r).img /boot/initrd-$(uname -r).img.bak
```

```
mkinitrd -f --with=virtio_blk --with=virtio_pci /boot/initrd-$(uname -r).img $(uname -r)
```

3) Debian/Ubuntu 操作方法

a. 检查virtio驱动情况

```
lsinitramfs /boot/initrd.img-$(uname -r) | grep virtio
```

b. 如果initramfs中未包含，则执行以下步骤修复

```
echo -e "virtio_pci\nvirtio_blk" >> /etc/initramfs-tools/modules  
update-initramfs -u
```