

云硬盘

最佳实践

文档版本 01
发布日期 2018-12-21



版权所有 © 华为技术有限公司 2024。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 通过 LVM 管理云硬盘	1
1.1 通过 LVM 管理云硬盘方案概述.....	1
1.2 资源和成本.....	2
1.3 操作流程.....	2
1.4 实施步骤.....	3
1.4.1 安装 LVM.....	3
1.4.2 通过 LVM 创建逻辑卷.....	4
1.4.3 创建并挂载文件系统.....	7
1.4.4 使用未分配容量扩展逻辑卷的容量.....	10
1.4.5 扩容云硬盘扩展逻辑卷的容量.....	11
1.4.6 新增云硬盘扩展卷组的容量.....	13
2 处理 Windows 云服务器磁盘空间不足的方法	17
2.1 简介.....	17
2.2 使用系统自带磁盘清理工具清理磁盘空间.....	17
2.3 使用控制面板卸载不需要的程序.....	19
3 使用云硬盘组建 RAID 磁盘阵列	21
3.1 方案概述.....	21
3.2 资源规划.....	22
3.3 实施步骤.....	23
3.3.1 创建云服务器.....	23
3.3.2 创建并挂载云硬盘.....	24
3.3.3 使用 mdadm 工具创建云硬盘 RAID 阵列.....	25
3.3.4 配置云硬盘 RAID 阵列开机自启动.....	28
4 扩展磁盘分区和文件系统（Linux 内核低于 3.6.0）	30
4.1 分区和文件系统扩展前准备（Linux 内核低于 3.6.0）.....	30
4.2 扩展系统盘的分区和文件系统（Linux 内核低于 3.6.0）.....	34
4.3 扩展数据盘的分区和文件系统（Linux 内核低于 3.6.0）.....	41
4.4 扩展 SCSI 数据盘的分区和文件系统（Linux 内核低于 3.6.0）.....	60

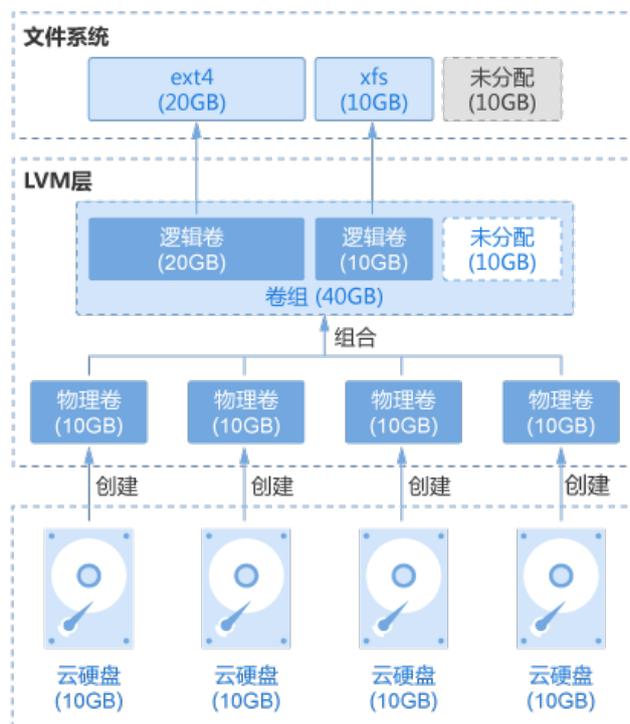
1 通过 LVM 管理云硬盘

1.1 通过 LVM 管理云硬盘方案概述

LVM是逻辑卷管理（Logical Volume Manager）的简称，它是Linux环境下对磁盘分区进行管理的一种机制。

LVM通过在云硬盘和文件系统之间添加一个逻辑层，来为文件系统屏蔽下层云硬盘分区布局，提高硬盘分区管理的灵活性，LVM的架构图如图1-1所示。

图 1-1 LVM 架构



从上面LVM架构图中可以看出，使用LVM管理云硬盘的基本过程如下：

1. 将云硬盘创建为物理卷
2. 将多个物理卷组合成卷组
3. 在卷组中创建逻辑卷
4. 在逻辑卷之上创建文件系统

通过LVM管理云硬盘之后，文件系统不再受限于云硬盘的大小，可以分布在多个云硬盘上，也可以动态扩容。

如果需要对图1-1中的“ext4”文件系统进行扩容，通常有两种做法：

- 卷组中未分配的空间充足时，直接扩展逻辑卷的容量
- 卷组中未分配的空间不足时，扩展卷组的容量，再扩展逻辑卷的容量

术语

- 物理卷（Physical Volume）
物理卷由云硬盘和LVM管理参数组成，是LVM的基本存储设备。
- 卷组（Volume Group）
卷组是将所有的物理卷首尾相连，组成的一个在逻辑上连续编址的大存储池。
- 逻辑卷（Logical Volume）
逻辑卷是卷组按照逻辑进行分区所得到的。

1.2 资源和成本

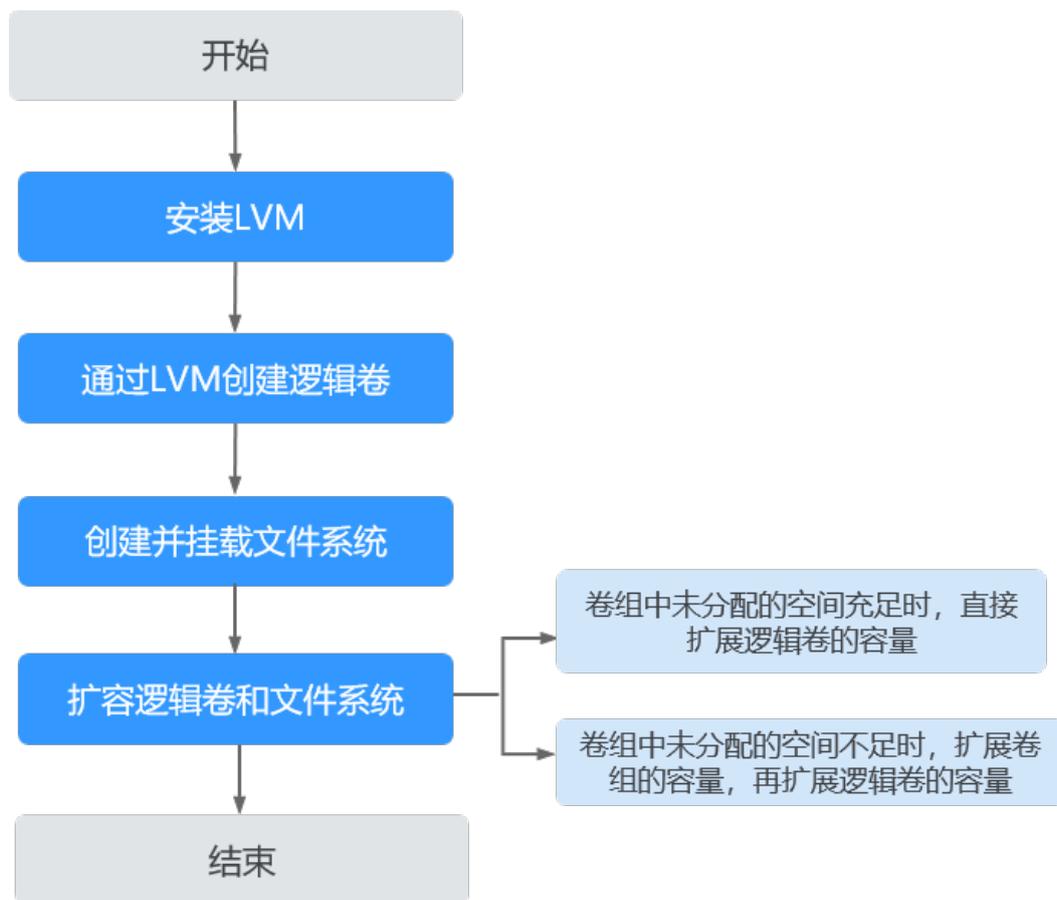
表 1-1 资源和成本规划

资源	资源说明	数量	每月费用
弹性公网IP	弹性云服务器需要绑定弹性公网IP	1	具体的计费方式及标准请参考 计费说明 。
弹性云服务器	操作系统：CentOS	1	具体的计费方式及标准请参考 计费说明 。
云硬盘	数据盘：10G	4	具体的计费方式及标准请参考 计费说明 。

1.3 操作流程

通过LVM管理云硬盘需要您先安装LVM，然后通过LVM创建逻辑卷、在逻辑卷上创建并挂载文件系统。

图 1-2 通过 LVM 管理云硬盘



1.4 实施步骤

1.4.1 安装 LVM

操作场景

默认情况下，弹性云服务器系统中并没有安装LVM，因此需要您手工安装。本章节将帮助您查询您的弹性云服务器是否安装了LVM以及如何去安装LVM。

本文以云服务器的操作系统为“CentOS 7.5 64位”为例，不同云服务器的操作系统的格式化操作可能不同，本文仅供参考。

前提条件

本指导假定您已有弹性云服务器，并为弹性云服务器绑定了弹性公网IP。

操作步骤

步骤1 以root用户登录弹性云服务器。

步骤2 执行以下命令，查看系统中是否安装了LVM管理工具。

```
rpm -qa |grep lvm2
```

```
[root@ecs-lvmtest ~]# rpm -qa |grep lvm2
lvm2-libs-2.02.177-4.el7.x86_64
lvm2-2.02.177-4.el7.x86_64
```

- 如果得到以上回显信息，则说明系统中已经安装了LVM工具，可直接查看[通过 LVM创建逻辑卷](#)。
- 如果未得到以上回显信息，则说明系统中未安装LVM工具，请执行[步骤3](#)完成安装。

步骤3 执行以下命令，并根据回显提示信息安装LVM工具。

```
yum install lvm2
```

回显类似如下信息：

```
.....
Installed:
  lvm2.x86_64 7:2.02.177-4.el7

Dependency Installed:
  device-mapper-event.x86_64 7:1.02.146-4.el7          device-mapper-event-libs.x86_64
  7:1.02.146-4.el7
  device-mapper-persistent-data.x86_64 0:0.7.3-3.el7    lvm2-libs.x86_64 7:2.02.177-4.el7

Dependency Updated:
  device-mapper.x86_64 7:1.02.146-4.el7              device-mapper-libs.x86_64 7:1.02.146-4.el7

Complete!
```

当回显显示“Complete!”时，表示安装LVM完成。

----结束

1.4.2 通过 LVM 创建逻辑卷

操作场景

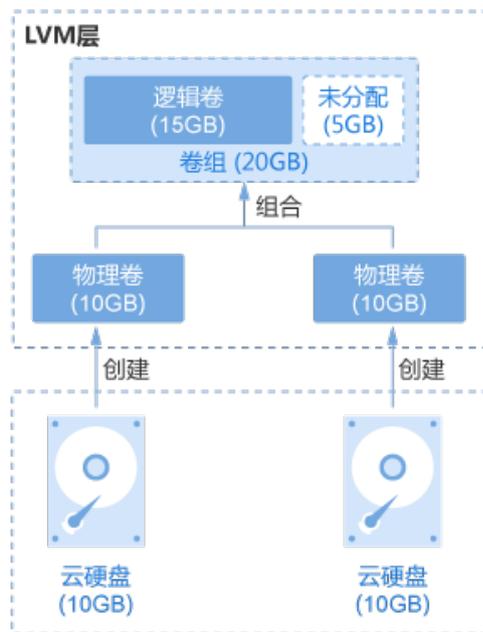
如[图1-3](#)所示，本章节指导您在两块10GB的云硬盘之上创建一个15GB的逻辑卷。

说明

创建逻辑卷使用的多块云硬盘，规格可以不同。

创建过程为依次创建物理卷、创建卷组和创建逻辑卷。

图 1-3 创建 LVM 逻辑卷示例



前提条件

安装LVM的弹性云服务器已经挂载了两块云硬盘。

操作步骤

步骤1 以root用户登录弹性云服务器。

步骤2 执行如下命令，查看磁盘并记录设备名称。

```
fdisk -l | grep /dev/vd | grep -v vda
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# fdisk -l | grep /dev/vd | grep -v vda
Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

可以看到云服务器上挂载的2块磁盘，设备名称分别为“/dev/vdb”、“/dev/vdc”。

步骤3 将云硬盘创建为物理卷。

1. 执行以下命令，将云硬盘创建为物理卷。

```
pvcreate 磁盘设备名1 磁盘设备名2 磁盘设备名3...
```

参数说明如下：

磁盘设备名：此处需要填写磁盘的设备名称，如果需要批量创建，可以填写多个设备名称，中间以空格间隔。

命令示例：

```
pvcreate /dev/vdb /dev/vdc
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# pvcreate /dev/vdb /dev/vdc
Physical volume "/dev/vdb" successfully created.
Physical volume "/dev/vdc" successfully created.
```

2. 执行如下命令，查看系统中物理卷的详细信息。

pvdisplay

回显类似如下信息：

```
[root@ecs-lvmtest ~]# pvdisplay
"/dev/vdc" is a new physical volume of "10.00 GiB"
--- NEW Physical volume ---
PV Name           /dev/vdc
VG Name
PV Size           10.00 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           dypyLh-xljj-PvG3-jD0j-yup5-O7SI-462R7C

"/dev/vdb" is a new physical volume of "10.00 GiB"
--- NEW Physical volume ---
PV Name           /dev/vdb
VG Name
PV Size           10.00 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           srv5H1-tgLu-GRTL-Vns8-GfNK-jtHk-Ag4HHB
```

可以看到系统有两个新的物理卷，物理卷名称分别为“/dev/vdc”、“/dev/vdb”。

步骤4 将多个物理卷组合成卷组。

1. 执行以下命令，创建卷组。

vgcreate 卷组名 物理卷名称1 物理卷名称2 物理卷名称3...

参数说明如下：

- 卷组名：可自定义，此处以vgdata为例。
- 物理卷名称：此处需要填写待添加进卷组的所有物理卷名称，中间以空格隔开。

命令示例：

vgcreate vgdata /dev/vdb /dev/vdc

回显类似如下信息：

```
[root@ecs-lvmtest ~]# vgcreate vgdata /dev/vdb /dev/vdc
Volume group "vgdata" successfully created
```

2. 执行如下命令，查看系统中卷组的详细信息。

vgdisplay

回显类似如下信息：

```
[root@ecs-lvmtest ~]# vgdisplay
--- Volume group ---
VG Name           vgdata
System ID
Format            lvm2
Metadata Areas    2
Metadata Sequence No 1
VG Access         read/write
VG Status         resizable
MAX LV            0
Cur LV           0
Open LV           0
```

```
Max PV          0
Cur PV         2
Act PV          2
VG Size         19.99 GiB
PE Size         4.00 MiB
Total PE        5118
Alloc PE / Size 0 / 0
Free PE / Size  5118 / 19.99 GiB
VG UUID         NLkZV7-hYYE-0w66-tnlt-Y6jL-lk7S-76w4P6
```

步骤5 在卷组中创建逻辑卷

1. 执行如下命令，创建逻辑卷。

```
lvcreate -L 逻辑卷大小 -n 逻辑卷名称 卷组名称
```

参数说明如下：

- 逻辑卷大小：该值应小于卷组剩余可用空间大小，单位可以选择“MB”或“GB”。
- 逻辑卷名称：可自定义，此处以lvdata1为例。
- 卷组名称：此处需要填写逻辑卷所在的卷组名称。

命令示例：

```
lvcreate -L 15GB -n lvdata1 vgdata
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# lvcreate -L 15GB -n lvdata1 vgdata
Logical volume "lvdata1" created.
```

2. 执行如下命令，查询系统中逻辑卷的详细信息。

```
lvdisplay
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# lvdisplay
--- Logical volume ---
LV Path                /dev/vgdata/lvdata1
LV Name                 lvdata1
VG Name                 vgdata
LV UUID                 c7mNcF-CdPW-5PLD-1gVj-QZpB-nHfy-PHXchV
LV Write Access         read/write
LV Creation host, time ecs-lvmtest.novalocal, 2018-11-29 11:28:18 +0800
LV Status                available
# open                  0
LV Size                 15.00 GiB
Current LE               3840
Segments                2
Allocation               inherit
Read ahead sectors      auto
 - currently set to     8192
Block device            252:0
```

----结束

1.4.3 创建并挂载文件系统

操作场景

在创建完逻辑卷之后，需要在逻辑卷之上创建文件系统并挂载文件系统到相应目录下。本章节指导您在逻辑卷之上创建“ext4”文件系统并挂载文件系统到“/Data1”目录下。

操作步骤

步骤1 以root用户登录弹性云服务器。

步骤2 执行如下命令，创建文件系统。

mkfs.*文件格式* *逻辑卷路径*

命令示例:

mkfs.ext4 /dev/vgdata/lvdata1

回显类似如下信息:

```
[root@ecs-lvmtest ~]# mkfs.ext4 /dev/vgdata/lvdata1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
983040 inodes, 3932160 blocks
196608 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2151677952
120 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

步骤3 执行如下命令，创建挂载目录。

mkdir *挂载目录*

命令示例:

mkdir /Data1

步骤4 执行如下命令，将文件系统挂载到目录下。

mount *逻辑卷路径* *挂载目录*

命令示例:

mount /dev/vgdata/lvdata1 /Data1

步骤5 执行如下命令，查询文件系统挂载信息。

mount | grep *挂载目录*

命令示例:

mount | grep /Data1

回显类似如下信息:

```
[root@ecs-lvmtest ~]# mount | grep /Data1
/dev/mapper/vgdata-lvdata1 on /Data1 type ext4 (rw,relatime,data=ordered)
```

“dev/mapper/vgdata-lvdata1”为文件系统路径，记录此路径，[步骤6](#)中需要使用此路径。

步骤6 执行以下步骤，设置云服务器系统启动时自动挂载文件系统。

如果未进行此项设置，在弹性云服务器重启时重新手动挂载文件系统。

1. 执行如下命令，查询文件系统的UUID。

blkid *文件系统路径*

以查询“dev/mapper/vgdata-lvdata1”的UUID为例：

blkid /dev/mapper/vgdata-lvdata1

回显类似如下信息：

```
[root@ecs-lvmtest ~]# blkid /dev/mapper/vgdata-lvdata1
/dev/mapper/vgdata-lvdata1: UUID="c6a243ce-5150-41ac-8816-39db54d1a4b8" TYPE="ext4"
UUID为"c6a243ce-5150-41ac-8816-39db54d1a4b8"。
```

2. 执行以下命令，打开“/etc/fstab”文件。

vi /etc/fstab

回显类似如下信息：

```
[root@ecs-lvmtest ~]# vi /etc/fstab

#
# /etc/fstab
# Created by anaconda on Tue Nov 7 14:28:26 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=27f9be47-838b-4155-b20b-e4c5e013cdf3 / ext4 defaults 1 1
UUID=2b2000b1-f926-4b6b-ade8-695ee244a901 /boot ext4 defaults 1 2
```

3. 按“i”进入编辑模式。
4. 将光标移至文件末尾，按“Enter”，添加如下内容。

```
UUID=c6a243ce-5150-41ac-8816-39db54d1a4b8 /Data1 ext4 defaults 0 0
```

内容说明如下：

- 第一列：UUID，此处填写1查询的UUID；
- 第二列：文件系统的挂载目录，此处填写步骤3创建的挂载目录“/Data1”；
- 第三列：文件系统的文件格式，此处填写步骤2设置的文件格式“ext4”；
- 第四列：挂载选项，此处以“defaults”为例；
- 第五列：备份选项，设置为“1”时，系统自动对该文件系统进行备份；设置为“0”时，不进行备份。此处以“0”为例；
- 第六列：扫描选项，设置为“1”时，系统在启动时自动对该文件系统进行扫描；设置为“0”时，不进行扫描。此处以“0”为例。

5. 按“Esc”，输入“:wq!”，并按“Enter”。

保存设置并退出vi编辑器。

步骤7 执行以下步骤，验证自动挂载功能。

1. 执行如下命令，卸载文件系统。

umount *逻辑卷路径*

命令示例：

umount /dev/vgdata/lvdata1

2. 执行如下命令，将/etc/fstab文件所有内容重新加载。

mount -a

3. 执行如下命令，查询文件系统挂载信息。

mount | grep 挂载目录

命令示例：

mount | grep /Data1

回显类似如下信息，说明自动挂载功能生效：

```
[root@ecs-lvmtest ~]# mount | grep /Data1
/dev/mapper/vgdata-lvdata1 on /Data1 type ext4 (rw,relatime,data=ordered)
```

----结束

1.4.4 使用未分配容量扩展逻辑卷的容量

操作场景

当逻辑卷容量不能满足用户需求时，可以扩展逻辑卷的容量。本指导假设您创建的15GB逻辑卷已经不能满足需求，需要增加4GB容量。

📖 说明

在进行扩容时，请确保需要扩容的逻辑卷所在的卷组有足够的空闲空间。如果卷组的空闲空间不能满足当前逻辑卷扩容的需求，请参考[扩容云硬盘扩展逻辑卷的容量](#)或者[新增云硬盘扩展卷组的容量](#)对卷组进行扩容。

操作步骤

步骤1 以root用户登录弹性云服务器。

步骤2 执行如下命令，扩展逻辑卷的容量。

lvextend -L +增加容量 逻辑卷路径

参数说明如下：

- 增加容量：该值应小于组卷剩余可用空间大小，单位可以选择“MB”或“GB”。
- 逻辑卷路径：此处需要填写待扩容的逻辑卷的路径。

命令示例：

lvextend -L +4GB /dev/vgdata/lvdata1

回显类似如下信息：

```
[root@ecs-lvmtest ~]# lvextend -L +4GB /dev/vgdata/lvdata1
Size of logical volume vgdata/lvdata1 changed from 15.00 GiB (3840 extents) to 19.00 GiB (4864 extents).
Logical volume vgdata/lvdata1 successfully resized.
```

此时只是扩展的逻辑卷的容量，在其之上的文件系统也要随之进行扩展才能使用。

步骤3 执行如下命令，扩展文件系统的容量。

resize2fs 逻辑卷路径

命令示例：

resize2fs /dev/vgdata/lvdata1

回显类似如下信息：

```
[root@ecs-lvmtest ~]# resize2fs /dev/vgdata/lvdata1
resize2fs 1.42.9 (28-Dec-2013)
```

```
Filesystem at /dev/vgdata/lvdata1 is mounted on /Data1; on-line resizing required  
old_desc_blocks = 4, new_desc_blocks = 28  
The filesystem on /dev/vgdata/lvdata1 is now 3657728 blocks long.
```

步骤4 执行如下命令，查看文件系统容量是否增加。

df -h

回显类似如下信息：

```
[root@ecs-lvmtest ~]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/vda2       39G  1.5G  35G   5% /  
devtmpfs       487M    0  487M   0% /dev  
tmpfs          496M    0  496M   0% /dev/shm  
tmpfs          496M  6.7M  490M   2% /run  
tmpfs          496M    0  496M   0% /sys/fs/cgroup  
/dev/vda1      976M  131M  779M  15% /boot  
tmpfs         100M    0  100M   0% /run/user/0  
/dev/mapper/vgdata-lvdata1 19G  44M  18G   1% /Data1
```

可以看到，文件系统“/dev/mapper/vgdata-lvdata1”的容量相比之前增加了4GB。

----结束

1.4.5 扩容云硬盘扩展逻辑卷的容量

操作场景

当逻辑卷容量不能满足用户需求时，可以扩展逻辑卷的容量。本指导假设您创建的19GB逻辑卷已经不能满足需求，通过扩容其中一个云硬盘，增加10GB容量。

操作步骤

步骤1 在控制台扩容云硬盘。

1. 登录管理控制台。
2. 选择“存储 > 云硬盘”。进入云硬盘页面。
3. 找到待扩容的云硬盘，扩容云硬盘。

关于扩容云硬盘的详细操作，请参见[扩容云硬盘容量](#)。

步骤2 以root用户登录弹性云服务器。

步骤3 执行以下命令，查看系统是否正确识别扩容后的磁盘。

fdisk -l

回显类似如下信息：

```
[root@ecs-lvmtest ~]# fdisk -l
Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000f1217

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *          2048     83886079     41942016   83  Linux

Disk /dev/vdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/vdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/vgdata-lvdata1: 20.4 GB, 20401094656 bytes, 39845888 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

扩容前/dev/vdb的容量是10GB，扩容后为20GB。

步骤4 执行如下命令，查看LVM的物理卷相关信息。

pvdisplay

回显类似如下信息：

```
[root@ecs-lvmtest ~]# pvdisplay
--- Physical volume ---
PU Name                /dev/vdb
VG Name                vgdata
PU Size                10.00 GiB / not usable 4.00 MiB
Allocatable            yes (but full)
PE Size                4.00 MiB
Total PE               2559
Free PE                0
Allocated PE           2559
PU UUID                QCBWMe-cHfp-2cAj-ZkUH-ghXM-SDJw-mu0rKI

--- Physical volume ---
PU Name                /dev/vdc
VG Name                vgdata
PU Size                10.00 GiB / not usable 4.00 MiB
Allocatable            yes
PE Size                4.00 MiB
Total PE               2559
Free PE                254
Allocated PE           2305
PU UUID                wJxNtf-k86g-fHY1-3ZiU-xLCZ-bG9a-nEo0FU
```

/dev/vdb的容量是10GB，说明物理卷容量未增加。

步骤5 执行如下命令，扩容该云硬盘对应的物理卷。

pvresize -v 磁盘设备名

命令示例：

pvresize -v /dev/vdb

回显类似如下信息：

```
[root@ecs-lvmtest ~]# pvresize -v /dev/vdb
archiving volume group "vgdata" metadata (seqno 3).
Resizing volume "/dev/vdb" to 41943040 sectors.
Resizing physical volume /dev/vdb from 2559 to 5119 extents.
Updating physical volume "/dev/vdb"
Creating volume group backup "/etc/lvm/backup/vgdata" (seqno 4).
Physical volume "/dev/vdb" changed
1 physical volume(s) resized or updated / 0 physical volume(s) not resized
```

说明/dev/vdb对应的物理卷扩容成功。

步骤6 如果还需要对现有分区进行扩容，执行如下命令，扩容对应逻辑卷。

lvextend -l +100%FREE 逻辑卷路径

命令示例：

```
lvextend -l +100%FREE /dev/vgdata/lvdata1
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# lvextend -l +100%FREE /dev/vgdata/lvdata1
Size of logical volume vgdata/lvdata1 changed from 19.00 GiB (4864 extents) to 29.99 GiB (7678 extents).
Logical volume vgdata/lvdata1 successfully resized.
```

步骤7 执行如下命令，扩展磁盘分区文件系统的大小。

resize2fs *逻辑卷路径*

命令示例：

```
resize2fs /dev/vgdata/lvdata1
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# resize2fs /dev/vgdata/lvdata1
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vgdata/lvdata1 is mounted on /Data1: on-line resizing required
old_desc_blocks = 3, new_desc_blocks = 4
[ 2591.781109] EXT4-fs (dm-0): resizing filesystem from 4988736 to 7862272 blocks
[ 2591.782411] EXT4-fs (dm-0): resized filesystem to 7862272
The filesystem on /dev/vgdata/lvdata1 is now 7862272 blocks long.
```

步骤8 执行如下命令，查看扩容最终结果。

lvdisplay

回显类似如下信息：

```
[root@ecs-lvmtest ~]# lvdisplay
--- Logical volume ---
LV Path                /dev/vgdata/lvdata1
LV Name                 lvdata1
VG Name                 vgdata
LV UUID                 5FCqyK-HBJE-apc1-F198-PUUu-9pEd-Gg5gMM
LV Write Access        read/write
LV Creation host, time ecs-lvmtest, 2020-06-04 17:13:26 +0800
LV Status               available
# open                  1
LV Size                 29.99 GiB
Current LE              7678
Segments                3                               29.99 GiB (7678 extents).
allocation              inherit
Read ahead sectors     auto
  - currently set to   8192
Block device            252:0
```

可以看到，逻辑卷容量“LV Size”已经增加10GB。

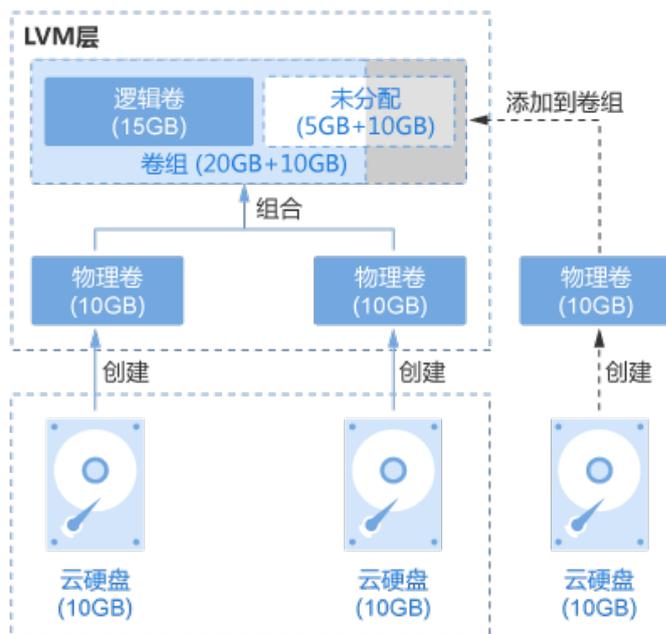
----结束

1.4.6 新增云硬盘扩展卷组的容量

操作场景

如图1-4所示，当LVM卷组的空间无法满足您的需求时，您可以通过新创建云硬盘、创建物理卷、将物理卷添加到卷组中等操作，对LVM卷组进行扩容。

图 1-4 扩容卷组示例



操作步骤

步骤1 创建云硬盘并挂载。

1. 登录管理控制台。
2. 选择“存储 > 云硬盘”。进入云硬盘页面。
3. 单击“购买磁盘”，创建云硬盘。
关于创建云硬盘的详细操作，请参见[云硬盘用户指南](#)。
4. 在云硬盘列表，找到新创建的云硬盘，单击“挂载”。
5. 选择云硬盘待挂载的云服务器，该云服务器必须与云硬盘位于同一个可用分区，通过下拉列表选择“挂载点”。

返回云硬盘列表页面，此时云硬盘状态为“正在挂载”，表示云硬盘处于正在挂载至云服务器的过程中。当云硬盘状态为“正在使用”时，表示挂载至云服务器成功。

步骤2 以root用户登录弹性云服务器。

步骤3 执行如下命令，查看系统中当前卷组的容量。

vgdisplay

回显类似如下信息：

```
[root@ecs-lvmtest ~]# vgdisplay
--- Volume group ---
VG Name          vgdata
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 3
VG Access        read/write
VG Status        resizable
MAX LV           0
```

```
Cur LV      1
Open LV     1
Max PV      0
Cur PV     2
Act PV      2
VG Size     19.99 GiB
PE Size     4.00 MiB
Total PE    5118
Alloc PE / Size 4864 / 19.00 GiB
Free PE / Size 254 / 1016.00 MiB
VG UUID     NLkZV7-hYYE-0w66-tnlt-Y6jL-lk7S-76w4P6
```

可以看到，当前卷组容量“VG Size”为19.99 GiB。

步骤4 执行如下命令，查看磁盘并记录设备名称。

```
fdisk -l | grep /dev/vd | grep -v vda
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# fdisk -l | grep /dev/vd | grep -v vda
Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdd: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

可以看到云服务器上新挂载的磁盘，设备名称为“/dev/vdd”。

步骤5 执行如下命令，将新挂载的磁盘创建为物理卷。

```
pvcreate 磁盘设备名
```

命令示例：

```
pvcreate /dev/vdd
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# pvcreate /dev/vdd
Physical volume "/dev/vdd" successfully created.
```

步骤6 执行如下命令，添加物理卷到卷组中，对卷组进行扩容。

```
vgextend 卷组名称 物理卷名称
```

命令示例：

```
vgextend vgdata /dev/vdd
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# vgextend vgdata /dev/vdd
Volume group "vgdata" successfully extended
```

步骤7 执行如下命令，查看系统中卷组的详细信息。

```
vgdisplay
```

回显类似如下信息：

```
[root@ecs-lvmtest ~]# vgdisplay
--- Volume group ---
VG Name          vgdata
System ID
Format           lvm2
Metadata Areas   3
Metadata Sequence No 4
VG Access        read/write
VG Status        resizable
```

```
MAX LV      0
Cur LV     1
Open LV     1
Max PV      0
Cur PV     3
Act PV      3
VG Size     <29.99 GiB
PE Size     4.00 MiB
Total PE    7677
Alloc PE / Size  4864 / 19.00 GiB
Free PE / Size  2813 / <10.99 GiB
VG UUID     NLkZV7-hYYE-0w66-tnlt-Y6jL-lk7S-76w4P6
```

可以看到，卷组容量“VG Size”已经增加10GB。

----结束

2 处理 Windows 云服务器磁盘空间不足的方法

2.1 简介

当云服务器的磁盘空间不足时，会影响云服务器的运行速度，降低使用体验。当磁盘空间不足时，此时你可以通过以下两种途径来处理：

- 清理磁盘空间，详细方法如下：
 - [使用系统自带磁盘清理工具清理磁盘空间](#)
 - [使用控制面板卸载不需要的程序](#)
- 扩容磁盘，详细方法如下：
 - 云硬盘用户指南 > 扩容状态为“正在使用”的云硬盘
 - 云硬盘用户指南 > 扩容状态为“可用”的云硬盘

本手册基于华为云云硬盘实践所编写，以操作系统为Windows 2016的云服务器为例。介绍了清理磁盘空间的常用操作。同时建议在日常使用中养成良好的磁盘使用习惯，定期清理冗余文件，有助于您节省磁盘空间。

- 定期将不常用的文件压缩保存，节省磁盘空间。
- 定期使用磁盘清理工具清理磁盘空间，删除不需要的文件，并定期清理回收箱。
- 卸载不需要的程序，释放磁盘空间。

2.2 使用系统自带磁盘清理工具清理磁盘空间

操作场景

本章节指导用户通过Windows操作系统自带的磁盘清理工具来清理空间不足的磁盘。

本文以操作系统为“Windows Server 2016 Standard 64bit”的云服务器为例。不同操作系统的云服务器清理步骤可能不同，本文仅供参考，具体操作步骤和差异请参考对应操作系统的产品文档。

操作步骤

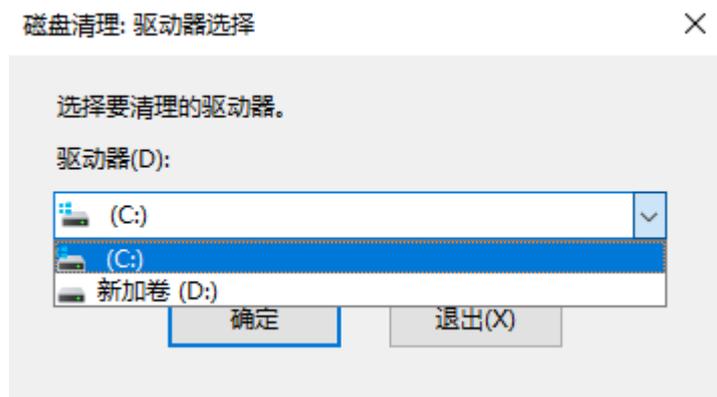
步骤1 在云服务器桌面，单击左下方开始图标。

弹出常用程序窗口。

步骤2 在左侧导航栏中，选择“Windows管理工具 > 磁盘清理”。

弹出“磁盘清理：驱动器选择”窗口。

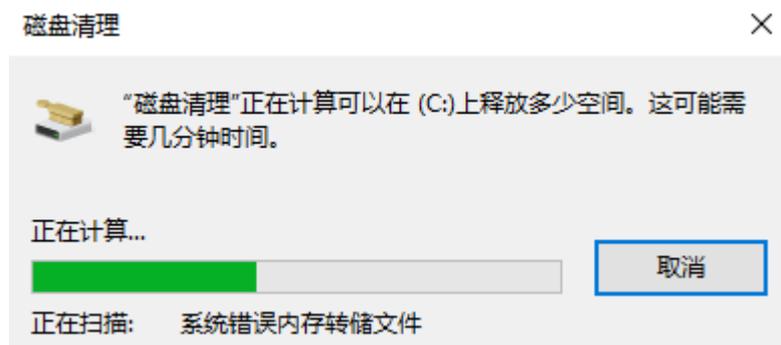
图 2-1 磁盘清理：驱动器选择



步骤3 在下拉框中，选择待清理的磁盘，以“C盘”为例。

弹出“磁盘清理”窗口，此时系统自动计算可在C盘上释放的空间。

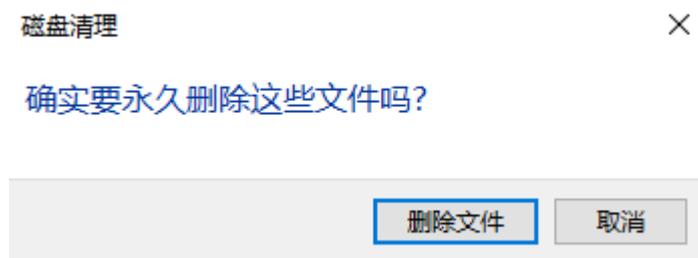
图 2-2 磁盘清理



步骤4 自动计算完成后，在弹窗中勾选要删除的文件，并单击“确定”。

弹出确认删除对话框。

图 2-3 确认删除



步骤5 单击“删除文件”，开始清理磁盘，释放磁盘空间。

---结束

2.3 使用控制面板卸载不需要的程序

操作场景

本章节指导用户通过控制面板卸载不需要的程序。

本文以操作系统为“Windows Server 2016 Standard 64bit”的云服务器为例。不同操作系统的云服务器清理步骤可能不同，本文仅供参考，具体操作步骤和差异请参考对应操作系统的产品文档。

操作步骤

步骤1 在云服务器桌面，单击左下方开始图标。

弹出常用程序窗口。

步骤2 在左侧导航栏中，选择“Windows系统 > 控制面板”。

弹出“所有控制面板项”窗口。

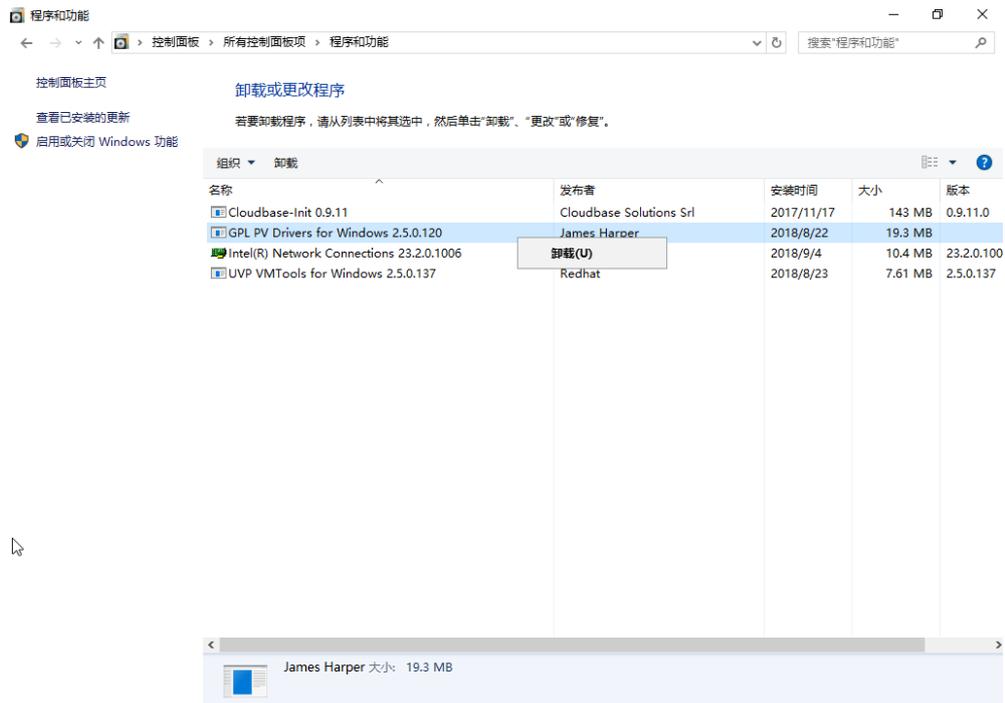
图 2-4 所有控制面板项



步骤3 在导航列表中，选择“程序和功能”。

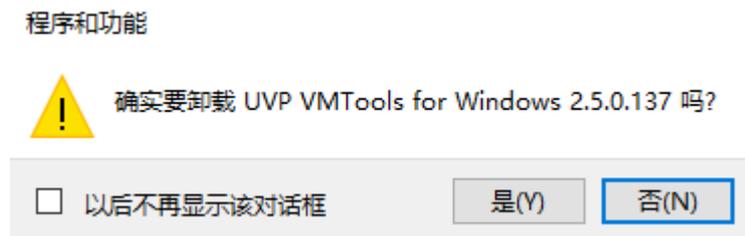
进入“程序和功能”窗口。

图 2-5 程序和功能



步骤4 在程序列表中，单击待卸载的程序，并右键单击“卸载”。
弹出确认卸载对话框。

图 2-6 确认卸载



步骤5 单击“是”，开始卸载程序，释放磁盘空间。

----结束

3 使用云硬盘组建 RAID 磁盘阵列

3.1 方案概述

独立冗余磁盘阵列（Redundant Array of Independent Disks, RAID），是把多块独立的磁盘（物理磁盘）按照一定的方式组合构建磁盘组（逻辑磁盘）。相比单个磁盘，能够提升存储性能和数据可靠性。

说明

此处的物理磁盘由云硬盘替代，将独立的云硬盘组成RAID阵列，原理与使用物理磁盘相同。

本手册以操作系统为CentOS 7.5的云服务器为例，共使用4块云硬盘介绍了组件RAID10的方法。RAID10是结合RAID1和RAID0两种磁盘阵列，先将磁盘两两组建成RAID1镜像，再组建RAID0将数据分条存储，至少需要4块磁盘。云服务资源的具体信息如下：

- 资源规划：[资源规划](#)
- 创建资源：[创建云服务器](#)、[创建并挂载云硬盘](#)

常见 RAID 磁盘阵列介绍

表 3-1 常见 RAID 磁盘阵列介绍

RAID 级别	简介	读写性能	安全性能	磁盘使用率	组建不同RAID阵列所需的最少磁盘数量
RAID 0	RAID0将数据分条存储在多个磁盘上，可实现并行读写，提供最快的读写速率。	多个磁盘并行读写获取更高性能	最差 没有冗余能力，一个磁盘损坏，整个RAID阵列数据都不可用	100%	两块

RAID 级别	简介	读写性能	安全性能	磁盘使用率	组建不同RAID阵列所需的最少磁盘数量
RAID 1	通过构造数据镜像实现数据冗余，阵列中一半的磁盘容量投入使用，另一半磁盘容量用来做镜像，提供数据备份。	读性能：与单个磁盘相同 写性能：需要将数据写入是两个磁盘，写性能低于单个磁盘	最高 提供磁盘数据的完整备份，当阵列中的一个磁盘失效时，系统可以自动采用镜像磁盘的数据	50%	两块
RAID 01	结合RAID0和RAID1两种磁盘阵列，先将一半磁盘组建成RAID0分条存储数据，再用另一半磁盘做RAID1镜像。	读性能：和RAID0相同 写性能：和RAID1相同	比RAID10的安全性能低	50%	四块
RAID 10	结合RAID1和RAID0两种磁盘阵列，先将磁盘两两组建成RAID1镜像，再组建RAID0将数据分条存储。	读性能：RAID0相同 写性能：RAID1相同	和RAID1的安全性能相同	50%	四块
RAID 5	RAID5不需要单独指定数据校验磁盘，而是将每块磁盘生成的校验信息分块存储至阵列中的每块磁盘中。	读性能：和RAID0相同 写性能：由于要写入奇偶校验信息，写性能低于单个磁盘	比RAID10的安全性能低	66.7%	三块

3.2 资源规划

本章节介绍了组建RAID10的云服务器及云硬盘资源的规划情况。

云服务器资源

本示例中创建了1台云服务器，参数配置如表3-2所示。

表 3-2 云服务器配置参数

参数	配置信息
名称	ecs-raid10
镜像	CentOS 7.5 64bit
规格	通用计算型、s2.medium.2 (1vCPUs、2GiB内存)
弹性公网IP地址	139.XX.XX.XX
私有IP地址	192.168.1.189

云硬盘资源

由于组建RAID10至少需要4块云硬盘，本示例中批量创建了4块云硬盘，并挂载至云服务。

3.3 实施步骤

3.3.1 创建云服务器

操作场景

本章节指导用户创建云服务器。本示例中共创建1台云服务器，配置情况请参见[资源规划](#)。

操作步骤

步骤1 登录管理控制台。

步骤2 选择“计算 > 弹性云服务器”。

进入弹性云服务器页面。

步骤3 单击“购买弹性云服务器”。

关于创建弹性云服务器的详细操作，请参见“弹性云服务器用户指南”。

创建云服务器过程中，以下参数请按照说明配置：

- 镜像：本示例中云服务器的镜像为“CentOS 7.5 64bit”。
- 弹性公网IP地址：云服务器需要访问公网时，必须使用弹性公网IP，由于本示例需要安装mdadm (multiple devices admin) 工具，因此必须开通弹性公网IP。请您根据实际情况购买或者使用已有。

本示例中使用新购买的IP，如[图3-1](#)所示。

图 3-1 配置弹性公网 IP

弹性公网IP 

如有互联网访问需求，请先规划您的弹性公网IP资源。单击[这里](#)查看弹性公网IP。

现在购买

使用已有

暂不购买

本示例创建1台云服务器，具体参数如表3-3所示。

表 3-3 云服务器规格

云服务器规格		计费模式	数量
规格	通用计算型 s2.medium.2 1vCPUs 2GiB	按需付费	1
镜像	CentOS 7.5 64bit		
系统盘	高IO, 40GB		
虚拟私有云	vpc-1a55		
安全组	Sys-default		
网卡	subnet-1a55(192.168.1.0/24)		
弹性公网IP	规格: 静态BGP 计费方式: 按带宽计费 带宽: 5Mbit/s		
云服务器名称	ecs-raid10		

----结束

3.3.2 创建并挂载云硬盘

操作场景

本章节指导用户批量创建4块云硬盘，并挂载至云服务器。

操作步骤

步骤1 登录管理控制台。

步骤2 选择“存储 > 云硬盘”。

进入云硬盘页面。

步骤3 单击“购买磁盘”，创建云硬盘。

关于创建云硬盘的详细操作，请参见“云硬盘用户指南”。

本示例为批量创建4块云硬盘，具体参数如图3-2所示。

图 3-2 云硬盘规格

购买磁盘 

服务选型 2 规格确认 3 完成

详情

产品类型	产品规格	计费模式	数量	价格
磁盘	区域 可用区 数据源 容量(GB) 磁盘类型 磁盘加密 磁盘模式 共享盘 磁盘名称 标签	广州 可用区2 暂不配置 普通IO 否 VBD 不共享 volume-raid10	4	¥0.0168/小时

步骤4 将云硬盘挂载至云服务器。

----结束

3.3.3 使用 mdadm 工具创建云硬盘 RAID 阵列

操作场景

本章节指导用户通过mdadm工具创建RAID阵列，以RAID10为例。

本文以云服务器的操作系统为“CentOS 7.5 64bit”为例。不同云服务器的操作系统的配置可能不同，本文仅供参考，具体操作步骤和差异请参考对应操作系统的产品文档。

操作步骤

步骤1 使用root用户登录云服务器。

步骤2 执行以下命令，查看磁盘并记录设备名称。

```
fdisk -l | grep /dev/vd | grep -v vda
```

回显类似如下信息：

```
[root@ecs-raid10 ~]# fdisk -l | grep /dev/vd | grep -v vda
Disk /dev/vdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vdd: 10.7 GB, 10737418240 bytes, 20971520 sectors
Disk /dev/vde: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

可以看到云服务器上挂载的4块磁盘，设备名称分别为“/dev/vdb”、“/dev/vdc”、“/dev/vdd”和“/dev/vde”。

步骤3 执行以下命令，安装mdadm工具。

```
yum install mdadm -y
```

说明

mdadm是Linux下的RAID管理工具，务必确保安装mdadm工具的云服务器已开通弹性公网IP。

回显类似如下信息：

```
[root@ecs-raid10 ~]# yum install mdadm -y
.....
Installed:
mdadm.x86_64 0:4.0-13.el7
```

```
Dependency Installed:
  libreport-filessystem.x86_64 0:2.1.11-40.el7.centos

Complete!
```

步骤4 执行以下命令，使用**步骤2**中查到的4块磁盘创建RAID阵列。

```
mdadm -Cv RAID阵列设备名 -a yes -n 磁盘数量 -l RAID级别 磁盘1设备名 磁盘2设备名 磁盘3设备名 磁盘4设备名...
```

参数说明如下：

- RAID阵列设备名：可自定义，此处以/dev/md0为例。
- 磁盘数量：根据实际情况填写，此处RAID10至少为4块。
不同的RAID阵列要求的最小磁盘数量不同，具体说明请参见[方案概述](#)。
- RAID级别：根据实际情况填写，此处以RAID10为例。
- 磁盘设备名：此处需要填写待组建RAID阵列的所有磁盘设备名，中间以空格隔开。

命令示例：

```
mdadm -Cv /dev/md0 -a yes -n 4 -l 10 /dev/vdb /dev/vdc /dev/vdd /dev/vde
```

回显类似如下信息：

```
[root@ecs-raid10 ~]# mdadm -Cv /dev/md0 -a yes -n 4 -l 10 /dev/vdb /dev/vdc /dev/vdd /dev/vde
mdadm: layout defaults to n2
mdadm: layout defaults to n2
mdadm: chunk size defaults to 512K
mdadm: size set to 10476544K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

步骤5 执行以下命令，格式化新建的RAID阵列。

```
mkfs.文件格式 RAID阵列设备名
```

命令示例：

```
mkfs.ext4 /dev/md0
```

回显类似如下信息：

```
[root@ecs-raid10 ~]# mkfs.ext4 /dev/md0
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=256 blocks
1310720 inodes, 5238272 blocks
261913 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2153775104
160 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
 4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

步骤6 执行以下命令，创建挂载目录。

```
mkdir 挂载目录
```

命令示例：

```
mkdir /RAID10
```

步骤7 执行以下命令，挂载RAID阵列设备名。

```
mount RAID阵列设备名 挂载目录
```

命令示例：

```
mount /dev/md0 /RAID10
```

步骤8 执行以下命令，查看RAID阵列的挂载结果。

```
df -h
```

回显类似如下信息：

```
[root@ecs-raid10 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda2        39G  1.5G  35G   5% /
devtmpfs         911M    0  911M   0% /dev
tmpfs            920M    0  920M   0% /dev/shm
tmpfs            920M  8.6M  911M   1% /run
tmpfs            920M    0  920M   0% /sys/fs/cgroup
/dev/vda1        976M  146M  764M  17% /boot
tmpfs            184M    0  184M   0% /run/user/0
/dev/md0         20G   45M  19G   1% /RAID10
```

步骤9 执行以下步骤，设置云服务器系统启动时自动挂载RAID阵列。

1. 执行以下命令，打开“/etc/fstab”文件。

```
vi /etc/fstab
```

2. 按“i”进入编辑模式。

回显类似如下信息：

```
[root@ecs-raid10 ~]# vi /etc/fstab

#
# /etc/fstab
# Created by anaconda on Tue Nov  7 14:28:26 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=27f9be47-838b-4155-b20b-e4c5e013cdf3 /          ext4 defaults    1 1
UUID=2b2000b1-f926-4b6b-ade8-695ee244a901 /boot      ext4 defaults    1 2
```

3. 参考以下示例，在文件的最后一行添加如下配置：

```
/dev/md0          /RAID10          ext4 defaults    0 0
```

4. 按“Esc”，输入“:wq!”，并按“Enter”。

保存设置并退出vi编辑器。

步骤10 执行以下命令，查看磁盘RAID阵列的信息。

```
mdadm -D RAID阵列设备名
```

命令示例：

```
mdadm -D /dev/md0
```

回显类似如下信息：

```
[root@ecs-raid10 ~]# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Thu Nov  8 15:49:02 2018
  Raid Level : raid10
  Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Thu Nov  8 16:15:11 2018
  State : clean
  Active Devices : 4
  Working Devices : 4
  Failed Devices : 0
  Spare Devices : 0

  Layout : near=2
  Chunk Size : 512K

Consistency Policy : resync

  Name : ecs-raid10.novalocal:0 (local to host ecs-raid10.novalocal)
  UUID : f400dbf9:60d211d9:e006e07b:98f8758c
  Events : 19

Number Major Minor RaidDevice State
 0   253   16     0   active sync set-A /dev/vdb
 1   253   32     1   active sync set-B /dev/vdc
 2   253   48     2   active sync set-A /dev/vdd
 3   253   64     3   active sync set-B /dev/vde
```

----结束

3.3.4 配置云硬盘 RAID 阵列开机自启动

操作场景

本章节指导用户在mdadm配置文件中添加新建RAID阵列的信息，例如设备名、UUID等。系统启动时，通过查询文件中配置的信息，启动运行可用的RAID阵列。

本文以云服务器的操作系统为“CentOS 7.5 64bit”为例。不同云服务器的操作系统的配置可能不同，本文仅供参考，具体操作步骤和差异请参考对应操作系统的产品文档。

操作步骤

步骤1 使用root用户登录云服务器。

步骤2 执行以下命令，查看RAID阵列的UUID等信息。

```
mdadm --detail --scan
```

回显类似如下信息：

```
[root@ecs-raid10 ~]# mdadm --detail --scan
ARRAY /dev/md0 metadata=1.2 name=ecs-raid10.novalocal:0 UUID=f400dbf9:60d211d9:e006e07b:98f8758c
```

步骤3 执行以下步骤，在mdadm文件中添加新建RAID阵列的信息。

1. 执行以下命令，打开“mdadm.conf”文件。

```
vi /etc/mdadm.conf
```

- 按“i”进入编辑模式。
- 参考以下示例，在文件最后添加如下配置：

```
DEVICE /dev/vdb /dev/vdc /dev/vdd /dev/vde
ARRAY /dev/md0 metadata=1.2 name=ecs-raid10.novalocal:0
UUID=f400dbf9:60d211d9:e006e07b:98f8758c
```

说明如下：

- DEVICE行：为组建RAID阵列的磁盘设备名，多个磁盘设备名以空格隔开。
- ARRAY行：此处填写[步骤2](#)中查到的RAID阵列信息。

说明

此处仅为本示例的信息，请根据RAID阵列的实际信息添加。

- 按“Esc”，输入“:wq!”，并按“Enter”。
保存设置并退出vi编辑器。

步骤4 执行以下命令，查看“mdadm.conf”文件是否修改成功。

more /etc/mdadm.conf

回显类似如下信息：

```
[root@ecs-raid10 ~]# more /etc/mdadm.conf
DEVICE /dev/vdb /dev/vdc /dev/vdd /dev/vde
ARRAY /dev/md0 metadata=1.2 name=ecs-raid10.novalocal:0 UUID=f400dbf9:60d211d9:e006e07b:98f8758c
```

可以看到[步骤3](#)中添加的信息，表示修改成功。

----**结束**

4 扩展磁盘分区和文件系统（Linux 内核低于 3.6.0）

4.1 分区和文件系统扩展前准备（Linux 内核低于 3.6.0）

通过云服务管理控制台扩容成功后，仅扩大了云硬盘的存储容量，因此需要参考本章节操作扩展分区和文件系统。

须知

- 扩容时请谨慎操作，误操作可能会导致数据丢失或者异常，建议扩容前对数据进行备份，可以使用CBR或者快照功能，CBR请参见[管理备份云硬盘](#)，快照功能请参见[创建快照（公测）](#)。
- 当操作系统内核低于3.6.0时，扩大已有MBR分区需要reboot重启，扩展分区和文件系统才会生效，会中断业务。reboot重启后，新增容量会自动扩展至系统盘末尾分区内。
- 如果您不希望重启弹性云服务器来扩展分区和文件系统，您可以先将该数据盘中的业务数据迁移至弹性云服务器中的其他磁盘，然后卸载该数据盘，并将其挂载至其他内核大于3.6.0的弹性云服务器上来扩展磁盘分区和文件系统，扩展完成后再将其挂载回原始弹性云服务器，最后再将业务数据迁移回该磁盘。迁移数据有风险，请提前做好备份。扩容内核大于3.6.0的弹性云服务器上的分区和文件系统请参见[扩展磁盘分区和文件系统（Linux）](#)。

扩展磁盘分区和文件系统前，请先检查磁盘的分区形式和文件系统，并根据磁盘的分区形式选择对应的操作指导。

1. 检查磁盘分区形式，请参见：
 - [方法一：使用fdisk命令查看分区形式和文件系统](#)
 - [方法二：使用parted命令查看分区形式和文件系统](#)
2. 选择操作指导，请参见[表4-1](#)。

表 4-1 分区和文件系统扩展场景说明

磁盘	场景	方法
系统盘	为扩容部分的云硬盘分配新的 MBR 分区	新增 MBR 分区
	将扩容部分的容量划分至已有的 MBR 分区内	扩大已有 MBR 分区（内核版本低于 3.6.0）
数据盘	为扩容部分的云硬盘分配新的 MBR 分区	新增 MBR 分区
	将扩容部分的容量划分至已有的 MBR 分区内	扩大已有 MBR 分区
	为扩容部分的云硬盘分配新的 GPT 分区	新增 GPT 分区
	将扩容部分的容量划分至已有的 GPT 分区内	扩大已有 GPT 分区
SCSI 数据盘	为扩容部分的云硬盘分配新的 MBR 分区	新增 MBR 分区
	将扩容部分的容量划分至已有的 MBR 分区内	扩大已有 MBR 分区

📖 说明

MBR 分区支持的磁盘最大容量为 2 TB，超过 2 TB 的部分无法使用。

如果当前磁盘采用 MBR 分区形式，并且需要将该磁盘扩容至 2 TB 以上投入使用。则必须将磁盘分区形式由 MBR 切换成 GPT，期间会中断业务，并且更换磁盘分区形式时会清除磁盘的原有数据，请在扩容前先对数据进行备份。

方法一：使用 fdisk 命令查看分区形式和文件系统

步骤1 执行以下命令，查看云服务器挂载的所有磁盘情况。

lsblk

回显类似如下信息：

```
[root@ecs-test-0001 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 253:0 0 40G 0 disk
└─vda1 253:1 0 40G 0 part /
vdb 253:16 0 150G 0 disk
└─vdb1 253:17 0 100G 0 part /mnt/sdc
```

本示例中数据盘“/dev/vdb”扩容前已有分区“/dev/vdb1”，将数据盘扩容50GB后，新增的容量还未划分磁盘分区，因此“/dev/vdb”显示150GB，“/dev/vdb1”显示100GB。

步骤2 执行以下命令，查看当前磁盘分区的分区形式。

fdisk -l

回显类似如下信息：

```
[root@ecs-test-0001 ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000bcb4e

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *        2048     83886079   41942016   83  Linux

Disk /dev/vdb: 161.1 GB, 161061273600 bytes, 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x38717fc1

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1          2048    209715199  104856576   83  Linux
```

“system”为“Linux”表示分区形式为MBR。“system”为“GPT”表示分区形式为GPT。

- 若回显中没有列出所有的磁盘分区，和步骤1中的信息不符合。可能原因是：磁盘已有分区为GPT，并且扩容后存在未分配分区空间，此时使用fdisk -l无法查看所有分区的信息，请参考方法二：使用parted命令查看分区形式和文件系统重新确认磁盘分区形式和文件系统。
- 若回显中列出了所有的磁盘分区，和步骤1中的信息符合，则继续执行以下操作。

步骤3 执行以下命令，查看磁盘分区的文件系统格式。

blkid 磁盘分区

命令示例：

blkid /dev/vdb1

```
[root@ecs-test-0001 ~]# blkid /dev/vdb1
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
```

“TYPE”为“ext4”：表示为/dev/vdb1的文件系统是ext4。

步骤4 执行以下命令，确认文件系统的状态。

ext*：**e2fsck -n 磁盘分区**

xfs：**xfs_repair -n 磁盘分区**

以“ext4”为例：

e2fsck -n /dev/vdb1

回显类似如下信息：

```
[root@ecs-test-0001 ~]# e2fsck -n /dev/vdb1
e2fsck 1.42.9 (28-Dec-2013)
Warning! /dev/vdb1 is mounted.
Warning: skipping journal recovery because doing a read-only filesystem check.
/dev/vdb1: clean, 11/6553600 files, 459544/26214144 blocks
```

文件系统状态为clean表示状态正常，若不是clean，请先修复问题后执行扩容操作。

----**结束**

方法二：使用 parted 命令查看分区形式和文件系统

步骤1 执行以下命令，查看云服务器挂载的所有磁盘情况。

lsblk

回显类似如下信息：

```
[root@ecs-test-0001 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 253:0 0 40G 0 disk
└─vda1 253:1 0 40G 0 part /
vdb 253:16 0 150G 0 disk
└─vdb1 253:17 0 100G 0 part /mnt/sdc
```

本示例中数据盘“/dev/vdb”扩容前已有分区“/dev/vdb1”，将数据盘扩容50GB后，新增的容量还未划分磁盘分区，因此“/dev/vdb”显示150GB，“/dev/vdb1”显示100GB。

步骤2 执行以下命令，指定一块磁盘，然后输入“p”，查看磁盘的分区形式。

parted 磁盘

以查看“/dev/vdb”的分区形式为例：

parted /dev/vdb

回显类似如下信息：

```
[root@ecs-test-0001 ~]# parted /dev/vdb
GNU Parted 3.1
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another
operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
Fix/Ignore/Cancel? Fix
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of
the space (an extra 104857600
blocks) or continue with the current setting?
Fix/Ignore? Fix
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 161GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start End Size File system Name Flags
1 1049kB 107GB 107GB ext4 test

(parted)
```

“Partition Table”表示当前磁盘的分区形式，msdos表示磁盘分区形式为MBR，gpt表示磁盘分区形式为GPT。

- 若系统出现以下Error，请输入“Fix”。

```
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that
another operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
```

GPT分区表信息存储在磁盘开头，为了减少分区表损坏的风险，同时在磁盘末尾会备份一份。当磁盘容量扩大后，末尾位置也会随之变化，因此需要根据系统提示输入“Fix”，将分区表信息的备份文件挪到新的磁盘末尾位置。

- 若系统出现以下Warning，请输入“Fix”。

```
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all
of the space (an extra 104857600
```

```
blocks) or continue with the current setting?  
Fix/Ignore? Fix
```

根据系统提示输入“Fix”，系统会自动将磁盘扩容部分的容量设置为GPT。

步骤3 查看完成后，输入“q”，退出parted模式。

---结束

4.2 扩展系统盘的分区和文件系统（Linux 内核低于 3.6.0）

操作场景

通过云服务管理控制台扩容成功后，仅扩大了云硬盘的存储容量，因此需要参考本章操作扩展分区和文件系统。

对于Linux操作系统而言，需要将扩容部分的容量划分至已有分区内，或者为扩容部分的云硬盘分配新的分区。

如果在云服务器关机的时候扩容了系统盘，则开机后，Linux系统盘的新增容量可能会自动扩展至末尾分区内，此时新增容量可以直接使用。

本文以内核低于3.6.0的“CentOS 6.5 64bit”操作系统为例，提供growpart和fdisk两种工具的扩容指导。不同操作系统的操作可能不同，本文仅供参考，具体操作步骤和差异请参考对应操作系统的产品文档。

- [扩大已有MBR分区（内核版本低于3.6.0）](#)
- [新增MBR分区](#)

查看Linux内核版本方法请参见[查看Linux操作系统内核版本](#)。

须知

- 扩容时请谨慎操作，误操作可能会导致数据丢失或者异常，建议扩容前对数据进行备份，可以使用CBR或者快照功能，CBR请参见[管理备份云硬盘](#)，快照功能请参见[创建快照（公测）](#)。
- 当操作系统内核低于3.6.0时，扩大已有MBR分区需要reboot重启，扩展分区和文件系统才会生效，会中断业务。reboot重启后，新增容量会自动扩展至系统盘末尾分区内。
- 如果您不希望重启弹性云服务器来扩展分区和文件系统，您可以先将该数据盘中的业务数据迁移至弹性云服务器中的其他磁盘，然后卸载该数据盘，并将其挂载至其他内核大于3.6.0的弹性云服务器上来扩展磁盘分区和文件系统，扩展完成后再将其挂载回原始弹性云服务器，最后再将业务数据迁移回该磁盘。迁移数据有风险，请提前做好备份。扩容内核大于3.6.0的弹性云服务器上的分区和文件系统请参见[扩展磁盘分区和文件系统（Linux）](#)。

前提条件

- 已通过管理控制台扩容系统盘容量，并已挂载至云服务器，请参见[扩容云硬盘容量](#)
- 已登录弹性云服务器。
 - 登录弹性云服务器请参见[登录弹性云服务器](#)。

- 登录裸金属服务器请参见[登录裸金属服务器](#)。

查看 Linux 操作系统内核版本

执行以下命令，查看Linux操作系统内核版本。

```
uname -a
```

需要根据Linux内核版本是否高于3.6.0选择对应的操作：

- 以“CentOS 7.4 64bit”为例，回显类似如下信息：
Linux ecs-test-0001 3.10.0-957.5.1.el7.x86_64 #1 SMP Fri Feb 1 14:54:57 UTC 2019 x86_64 x86_64
内核版本为3.10.0，高于3.6.0，具体操作请参见[扩展系统盘的分区和文件系统（Linux内核高于3.6.0）](#)。
- 以“CentOS 6.5 64bit”为例，回显类似如下信息：
Linux ecs-test-0002 2.6.32-754.10.1.el6.x86_64 #1 SMP Tue Jan 15 17:07:28 UTC 2019 x86_64
内核版本为2.6.32，低于3.6.0，需要reboot重启，才可以完成分区和文件系统扩展，具体操作请参见本章节内容。

扩大已有 MBR 分区（内核版本低于 3.6.0）

以“CentOS 6.5 64bit”操作系统为例，系统盘“/dev/vda”原有容量40GB，只有一个分区“/dev/vda1”。将系统盘容量扩大至100GB，本示例将新增的60GB划分至已有的MBR分区内“/dev/vda1”内。

步骤1 （可选）执行以下命令，安装growpart扩容工具。

```
yum install cloud-utils-growpart
```

📖 说明

可以用growpart命令检查当前系统是否已安装growpart扩容工具，若回显为工具使用介绍，则表示已安装，无需重复安装。

步骤2 执行以下命令，安装dracut-modules-growroot工具。

```
yum install dracut-modules-growroot
```

回显类似如下信息：

```
[root@ecs-test-0002 ~]# yum install dracut-modules-growroot
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
epel/metalink | 4.3 kB
00:00
* epel: pubmirror1.math.uh.edu
base | 3.7 kB
00:00
extras | 3.4 kB
00:00
updates | 3.4 kB
00:00
Package dracut-modules-growroot-0.20-2.el6.noarch already installed and latest version
Nothing to do
```

步骤3 执行以下命令，重新生成initramfs文件。

```
dracut -f
```

步骤4 执行以下命令，查看系统盘“/dev/vda”的总容量。

```
fdisk -l
```

回显类似如下信息:

```
[root@ecs-test-0002 ~]# fdisk -l

Disk /dev/vda: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0004e0be

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *          1         5222    41942016   83  Linux
```

步骤5 执行以下命令，查看系统盘分区“/dev/vda1”的容量。

df -TH

回显类似如下信息:

```
[root@ecs-test-0002 ~]# df -TH
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/vda1      ext4  43G  1.7G  39G   5% /
tmpfs          tmpfs 2.1G   0  2.1G   0% /dev/shm
```

步骤6 执行以下命令，重启云服务器。

reboot

待重启完成后，重新连接云服务器后，执行以下操作。

说明

重启云服务器后，扩展磁盘分区和文件系统即可生效，无需再执行“**resize2fs 磁盘分区**”命令来扩展分区和文件系统。

步骤7 执行以下命令，查看扩容后系统盘分区“/dev/vda1”的容量。

df -TH

回显类似如下信息:

```
[root@ecs-test-0002 ~]# df -TH
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/vda1      ext4 106G  1.7G  99G   2% /
tmpfs          tmpfs 2.1G   0  2.1G   0% /dev/shm
```

----结束

新增 MBR 分区

系统盘“/dev/vda”原有容量40GB，只有一个分区“/dev/vda1”。将系统盘容量扩大至80GB，本示例为新增的40GB分配新的MBR分区“/dev/vda2”。

步骤1 执行以下命令，查看磁盘的分区信息。

fdisk -l

回显类似如下信息:

```
[root@ecs-2220 ~]# fdisk -l

Disk /dev/vda: 85.9 GB, 85899345920 bytes, 167772160 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0008d18f
```

```
Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *        2048     83886079   41942016   83  Linux
```

表示当前系统盘“dev/vda”容量为80 GB，当前正在使用的分区“dev/vda1”为40 GB，新扩容的40 GB还未分配分区。

步骤2 执行如下命令之后，进入fdisk分区工具。

```
fdisk /dev/vda
```

回显类似如下信息：

```
[root@ecs-2220 ~]# fdisk /dev/vda
Welcome to fdisk (util-linux 2.23.2).
```

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Command (m for help):
```

步骤3 输入“n”，按“Enter”，开始新建分区。

回显类似如下信息：

```
Command (m for help): n
Partition type:
  p  primary (1 primary, 0 extended, 3 free)
  e  extended
```

表示磁盘有两种分区类型：

- “p”表示主分区。
- “e”表示扩展分区。

说明

磁盘使用MBR分区形式，最多可以创建4个主分区，或者3个主分区加1个扩展分区，扩展分区不可以直接使用，需要划分成若干个逻辑分区才可以使用。

磁盘使用GPT分区形式时，没有主分区、扩展分区以及逻辑分区之分。

步骤4 以创建一个主要分区为例，输入“p”，按“Enter”，开始创建一个主分区。

回显类似如下信息：

```
Select (default p): p
Partition number (2-4, default 2):
```

步骤5 以分区编号选择“2”为例，输入分区编号“2”，按“Enter”。

回显类似如下信息：

```
Partition number (2-4, default 2): 2
First sector (83886080-167772159, default 83886080):
```

步骤6 输入新分区的起始磁柱值，以使用默认起始磁柱值为例，按“Enter”。

系统会自动提示分区可用空间的起始磁柱值和截止磁柱值，可以在该区间内自定义，或者使用默认值。起始磁柱值必须小于分区的截止磁柱值。

回显类似如下信息：

```
First sector (83886080-167772159, default 83886080):
Using default value 83886080
Last sector, +sectors or +size{K,M,G} (83886080-167772159,default 167772159):
```

步骤7 输入新分区的截止磁柱值，以使用默认截止磁柱值为例，按“Enter”。

系统会自动提示分区可用空间的起始磁柱值和截止磁柱值，可以在该区间内自定义，或者使用默认值。起始磁柱值必须小于分区的截止磁柱值。

回显类似如下信息:

```
Last sector, +sectors or +size{K,M,G} (83886080-167772159,
default 167772159):
Using default value 167772159
Partition 2 of type Linux and of size 40 GiB is set
```

Command (m for help):

步骤8 输入“p”，按“Enter”，查看新建分区。

回显类似如下信息:

Command (m for help): p

```
Disk /dev/vda: 85.9 GB, 85899345920 bytes, 167772160 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0008d18f
```

```
Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *        2048     83886079   41942016   83  Linux
/dev/vda2          83886080  167772159   41943040   83  Linux
Command (m for help):
```

步骤9 输入“w”，按“Enter”，将分区结果写入分区表中。

回显类似如下信息:

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

```
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

表示分区创建完成。

说明

如果之前分区操作有误，请输入“q”，则会退出fdisk分区工具，之前的分区结果将不会被保留。

步骤10 执行以下命令，将新的分区表变更同步至操作系统。

partprobe

步骤11 执行以下命令，设置新建分区文件系统格式。

mkfs -t 文件系统 磁盘分区

- ext*文件系统命令示例:

以“ext4”文件格式为例:

mkfs -t ext4 /dev/vda2

回显类似如下信息:

```
[root@ecs-2220 ~]# mkfs -t ext4 /dev/vda2
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
2621440 inodes, 10485760 blocks
```

```
524288 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2157969408
320 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

- xfs文件系统命令示例:

mkfs -t xfs /dev/vda2

回显类似如下信息:

```
[root@ecs-2220 ~]# mkfs -t xfs /dev/vda2
meta-data=/dev/vda2          isize=512    agcount=4, agsize=2621440 blks
        =                   sectsz=512   attr=2,    projid32bit=1
        =                   crc=1      finobt=0, sparse=0
data      =                   bsize=4096  blocks=10485760, imaxpct=25
        =                   sunit=0    swidth=0 blks
naming   =version2          bsize=4096  ascii-ci=0 ftype=1
log      =internal log     bsize=4096  blocks=5120, version=2
        =                   sectsz=512   sunit=0 blks, lazy-count=1
realtime =none             extsz=4096  blocks=0, rtextents=0
```

格式化需要等待一段时间, 请观察系统运行状态, 若回显中进程提示为done, 则表示格式化完成。

步骤12 (可选) 执行以下命令, 新建挂载目录。

若需要挂载至新建目录下, 执行该操作。

mkdir 挂载目录

以新建挂载目录 “/opt” 为例:

mkdir /opt

步骤13 执行以下命令, 挂载新建分区。

mount 磁盘分区 挂载目录

以挂载新建分区 “/dev/vda2” 至 “/opt” 为例:

mount /dev/vda2 /opt

说明

新增加的分区挂载到不为空的目录时, 该目录下原本的子目录和文件会被隐藏, 所以, 新增的分区最好挂载到空目录或者新建目录。如确实要挂载到不为空的目录, 可将该目录下的子目录和文件临时移动到其他目录下, 新分区挂载成功后, 再将子目录和文件移动回来。

步骤14 执行以下命令, 查看挂载结果。

df -TH

回显类似如下信息:

```
[root@ecs-2220 ~]# df -TH
Filesystem Type Size Used Avail Use% Mounted on
/dev/vda1 ext4 43G 2.0G 39G 5% /
```

```
devtmpfs    devtmpfs 509M    0 509M    0% /dev
tmpfs       tmpfs    520M    0 520M    0% /dev/shm
tmpfs       tmpfs    520M    7.2M 513M    2% /run
tmpfs       tmpfs    520M    0 520M    0% /sys/fs/cgroup
tmpfs       tmpfs    104M    0 104M    0% /run/user/0
/dev/vda2   ext4     43G    51M 40G    1% /opt
```

📖 说明

云服务器重启后，挂载会失效。您可以修改“/etc/fstab”文件，将新建磁盘分区设置为开机自动挂载，请参见[设置开机自动挂载磁盘分区](#)。

----结束

设置开机自动挂载磁盘分区

您可以通过配置fstab文件，设置弹性云服务器系统启动时自动挂载磁盘分区。已有数据的弹性云服务器也可以进行设置，该操作不会影响现有数据。

本文介绍如何在fstab文件中使用UUID来设置自动挂载磁盘分区。不建议采用在“/etc/fstab”直接指定设备名（比如/dev/vdb1）的方法，因为云中设备的顺序编码在关闭或者开启弹性云服务器过程中可能发生改变，例如/dev/vdb1可能会变成/dev/vdb2，可能会导致弹性云服务器重启后不能正常运行。

📖 说明

UUID (universally unique identifier) 是Linux系统为磁盘分区提供的唯一的标识字符串。

步骤1 执行如下命令，查询磁盘分区的UUID。

blkid 磁盘分区

以查询磁盘分区“/dev/vdb1”的UUID为例：

blkid /dev/vdb1

回显类似如下信息：

```
[root@ecs-test-0001 ~]# blkid /dev/vdb1
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
```

记录下回显中磁盘分区“/dev/vdb1”的UUID，方便后续步骤使用。

步骤2 执行以下命令，使用VI编辑器打开“fstab”文件。

vi /etc/fstab

步骤3 按“i”，进入编辑模式。

步骤4 将光标移至文件末尾，按“Enter”，添加如下内容。

```
UUID=0b3040e2-1367-4abb-841d-ddb0b92693df /mnt/sdc          ext4 defaults    0 2
```

以上内容仅为示例，具体请以实际情况为准，参数说明如下：

- 第一列为UUID，此处填写**步骤1**中查询到的磁盘分区的UUID。
- 第二列为磁盘分区的挂载目录，可以通过**df -TH**命令查询。
- 第三列为磁盘分区的文件系统格式，可以通过**df -TH**命令查询。
- 第四列为磁盘分区的挂载选项，此处通常设置为defaults即可。
- 第五列为Linux dump备份选项。
 - 0表示不使用Linux dump备份。现在通常不使用dump备份，此处设置为0即可。

- 1表示使用Linux dump备份。
- 第六列为fsck选项，即开机时是否使用fsck检查磁盘。
 - 0表示不检验。
 - 挂载点为（/）根目录的分区，此处必须填写1。
根分区设置为1，其他分区只能从2开始，系统会按照数字从小到大依次检查下去。

步骤5 按“ESC”后，输入“:wq”，按“Enter”。

保存设置并退出编辑器。

步骤6 执行以下步骤，验证自动挂载功能。

1. 执行如下命令，卸载已挂载的分区。

```
umount 磁盘分区
```

命令示例：

```
umount /dev/vdb1
```

2. 执行如下命令，将“/etc/fstab”文件所有内容重新加载。

```
mount -a
```

3. 执行如下命令，查询文件系统挂载信息。

```
mount | grep 挂载目录
```

命令示例：

```
mount | grep /mnt/sdc
```

回显类似如下信息，说明自动挂载功能生效：

```
root@ecs-test-0001 ~]# mount | grep /mnt/sdc  
/dev/vdb1 on /mnt/sdc type ext4 (rw,relatime,data=ordered)
```

---结束

4.3 扩展数据盘的分区和文件系统（Linux 内核低于 3.6.0）

操作场景

通过云服务管理控制台扩容成功后，仅扩大了云硬盘的存储容量，因此需要参考本章节操作扩展分区和文件系统。

对于Linux操作系统而言，需要将扩容部分的容量划分至已有分区内，或者为扩容部分的云硬盘分配新的分区。

本文以“CentOS 7.4 64位”操作系统为例，提供MBR分区和GPT分区的操作指导。不同操作系统的操作可能不同，本文仅供参考，具体操作步骤和差异请参考对应操作系统的产品文档。

- [新增MBR分区](#)
- [扩大已有MBR分区](#)
- [新增GPT分区](#)
- [扩大已有GPT分区](#)

须知

- 扩容时请谨慎操作，误操作可能会导致数据丢失或者异常，建议扩容前对数据进行备份，可以使用CBR或者快照功能，CBR请参见[管理备份云硬盘](#)，快照功能请参见[创建快照（公测）](#)。
- 当操作系统内核低于3.6.0时，扩大已有MBR分区需要reboot重启，扩展分区和文件系统才会生效，会中断业务。reboot重启后，新增容量会自动扩展至系统盘末尾分区内。
- 如果您不希望重启弹性云服务器来扩展分区和文件系统，您可以先将该数据盘中的业务数据迁移至弹性云服务器中的其他磁盘，然后卸载该数据盘，并将其挂载至其他内核大于3.6.0的弹性云服务器上来扩展磁盘分区和文件系统，扩展完成后再将其挂载回原始弹性云服务器，最后再将业务数据迁移回该磁盘。迁移数据有风险，请提前做好备份。扩容内核大于3.6.0的弹性云服务器上的分区和文件系统请参见[扩展磁盘分区和文件系统（Linux）](#)。

前提条件

- 已通过管理控制台扩容系统盘容量，并已挂载至云服务器，请参见[扩容云硬盘容量](#)
- 已登录弹性云服务器。
 - 登录弹性云服务器请参见[登录弹性云服务器](#)。
 - 登录裸金属服务器请参见[登录裸金属服务器](#)。

新增 MBR 分区

数据盘“/dev/vdb”原有容量100GB，只有一个分区“/dev/vdb1”。将数据盘容量扩大至150GB，本示例为新增的50GB分配新的MBR分区“/dev/vdb2”。

步骤1 执行以下命令，查看磁盘的分区信息。

fdisk -l

回显类似如下信息：

```
[root@ecs-test-0001 ~]# fdisk -l
```

```
Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000bcb4e
```

Device	Boot	Start	End	Blocks	Id	System
/dev/vda1	*	2048	83886079	41942016	83	Linux

```
Disk /dev/vdb: 161.1 GB, 161061273600 bytes, 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x38717fc1
```

Device	Boot	Start	End	Blocks	Id	System
/dev/vdb1		2048	209715199	104856576	83	Linux

步骤2 执行以下命令，进入fdisk分区工具。

fdisk 磁盘

命令示例:

fdisk /dev/vdb

回显类似如下信息:

```
[root@ecs-test-0001 ~]# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help):
```

步骤3 输入“n”，按“Enter”，开始新建分区。

回显类似如下信息:

```
Command (m for help): n
Partition type:
  p  primary (1 primary, 0 extended, 3 free)
  e  extended
Select (default p):
```

表示磁盘有两种分区类型:

- “p”表示主分区。
- “e”表示扩展分区。

说明

磁盘使用MBR分区形式，最多可以创建4个主分区，或者3个主分区加1个扩展分区，扩展分区不可以直接使用，需要划分成若干个逻辑分区才可以使用。

磁盘使用GPT分区形式时，没有主分区、扩展分区以及逻辑分区之分。

步骤4 以创建一个主分区为例，输入“p”，按“Enter”。

回显类似如下信息:

```
Select (default p): p
Partition number (2-4, default 2):
```

“Partition number”表示主分区编号，可以选择2-4，由于1已被使用，此处从2开始。

步骤5 以分区编号选择“2”为例，输入分区编号“2”，按“Enter”。

回显类似如下信息:

```
Partition number (2-4, default 2): 2
First sector (209715200-314572799, default 209715200):
```

“First sector”表示起始磁柱值，可以选择209715200-314572799，默认为209715200。

步骤6 输入新分区的起始磁柱值，以使用默认起始磁柱值为例，按“Enter”。

系统会自动提示分区可用空间的起始磁柱值和截止磁柱值，可以在该区间内自定义，或者使用默认值。起始磁柱值必须小于分区的截止磁柱值。

回显类似如下信息:

```
First sector (209715200-314572799, default 209715200):
Using default value 209715200
Last sector, +sectors or +size{K,M,G} (209715200-314572799, default 314572799):
```

“Last sector”表示截止磁柱值，可以选择209715200-314572799，默认为314572799。

步骤7 输入新分区的截止磁柱值，以使用默认截止磁柱值为例，按“Enter”。

系统会自动提示分区可用空间的起始磁柱值和截止磁柱值，可以在该区间内自定义，或者使用默认值。起始磁柱值必须小于分区的截止磁柱值。

回显类似如下信息：

```
Last sector, +sectors or +size{K,M,G} (209715200-314572799, default 314572799):
Using default value 314572799
Partition 2 of type Linux and of size 50 GiB is set

Command (m for help):
```

步骤8 输入“p”，按“Enter”，查看新建分区。

回显类似如下信息：

```
Command (m for help): p

Disk /dev/vdb: 161.1 GB, 161061273600 bytes, 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x38717fc1

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048    209715199    104856576   83  Linux
/dev/vdb2    209715200    314572799     52428800   83  Linux

Command (m for help):
```

步骤9 输入“w”，按“Enter”，将分区结果写入分区表中。

回显类似如下信息：

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

说明

如果之前分区操作有误，请输入“q”，则会退出fdisk分区工具，之前的分区结果将不会被保留。

步骤10 执行以下命令，将新的分区表变更同步至操作系统。

partprobe

步骤11 执行以下命令，为新建分区设置文件系统。

mkfs -t 文件系统 磁盘分区

- ext*文件系统命令示例：

mkfs -t ext4 /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb2
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
```

```
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

- xfs文件系统命令示例:

mkfs -t xfs /dev/vdb2

回显类似如下信息:

```
[root@ecs-test-0001 ~]# mkfs -t xfs /dev/vdb2
meta-data=/dev/vdb2          isize=512    agcount=4, agsize=3276800 blks
        =                   sectsz=512   attr=2,    projid32bit=1
        =                   crc=1      finobt=0, sparse=0
data      =                   bsize=4096  blocks=13107200, imaxpct=25
        =                   sunit=0     swidth=0 blks
naming   =version2          bsize=4096  ascii-ci=0 ftype=1
log      =internal log     bsize=4096  blocks=6400, version=2
        =                   sectsz=512   sunit=0 blks, lazy-count=1
realtime =none             extsz=4096  blocks=0, rtextents=0
```

格式化需要等待一段时间, 请观察系统运行状态, 若回显中进程提示为done, 则表示格式化完成。

步骤12 (可选) 执行以下命令, 新建挂载目录。

若需要挂载至新建目录下, 执行该操作。

mkdir 挂载目录

以新建挂载目录 “/mnt/test” 为例:

mkdir /mnt/test

步骤13 执行以下命令, 挂载新建分区。

mount 磁盘分区 挂载目录

以挂载新建分区 “/dev/vdb2” 至 “/mnt/test” 为例:

mount /dev/vdb2 /mnt/test

📖 说明

新增加的分区挂载到不为空的目录时, 该目录下原本的子目录和文件会被隐藏, 所以, 新增的分区最好挂载到空目录或者新建目录。如确实要挂载到不为空的目录, 可将该目录下的子目录和文件临时移动到其他目录下, 新分区挂载成功后, 再将子目录和文件移动回来。

步骤14 执行以下命令, 查看挂载结果。

df -TH

回显类似如下信息:

```
[root@ecs-test-0001 ~]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/vda1       ext4      43G   1.9G   39G   5% /
devtmpfs        devtmpfs  2.0G   0   2.0G   0% /dev
tmpfs           tmpfs     2.0G   0   2.0G   0% /dev/shm
tmpfs           tmpfs     2.0G   9.1M   2.0G   1% /run
tmpfs           tmpfs     2.0G   0   2.0G   0% /sys/fs/cgroup
tmpfs           tmpfs     398M   0   398M   0% /run/user/0
/dev/vdb1       ext4     106G   63M  101G   1% /mnt/sdc
/dev/vdb2       ext4      53G   55M   50G   1% /mnt/test
```

📖 说明

云服务器重启后, 挂载会失效。您可以修改 “/etc/fstab” 文件, 将新建磁盘分区设置为开机自动挂载, 请参见[设置开机自动挂载磁盘分区](#)。

----结束

扩大已有 MBR 分区

须知

扩大已有分区不会删除数据盘上的数据, 但是需要通过umount命令先卸载已有分区, 因此会影响线上业务运行。

数据盘 “/dev/vdb” 原有容量150GB, 有两个分区 “/dev/vdb1 ” 和 “/dev/vdb2” 。将数据盘容量扩大至230GB, 本示例将新增的80GB划分至已有的MBR分区内 “/dev/vdb2” 内。

须知

扩容后的新增存储空间是添加在磁盘末尾的, 对具有多个分区的的磁盘扩容时, 只支持将新增容量划分至排在末尾的分区。

步骤1 执行以下命令, 查看磁盘的分区信息。

fdisk -l

回显类似如下信息:

```
[root@ecs-test-0001 ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000bcb4e

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *         2048     83886079     41942016   83  Linux

Disk /dev/vdb: 247.0 GB, 246960619520 bytes, 482344960 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x38717fc1
```

```
Device Boot      Start         End      Blocks   Id  System
/dev/vdb1        2048    209715199    104856576   83  Linux
/dev/vdb2        209715200  314572799     52428800   83  Linux
```

记录待扩大分区的起始磁柱值（Start）和截止磁柱值（End），本示例中待扩大分区“/dev/vdb2”的起始磁柱值为209715200，截止磁柱值为314572799。

查看回显中数据盘“/dev/vdb”的容量，扩容的容量是否已经包含在容量总和中。

- 若扩容的容量未在数据盘容量总和中，请参考[扩展SCSI数据盘的分区和文件系统（Linux内核低于3.6.0）](#)章节刷新系统内容。
- 若扩容的容量已在数据盘容量总和中，请记录待扩大分区的起始和截止磁柱值，这些值在后续重新创建分区时需要使用，记录完成后执行[步骤2](#)。

步骤2 执行以下命令，卸载磁盘分区。

```
umount 磁盘分区
```

命令示例：

```
umount /dev/vdb2
```

步骤3 执行以下命令，进入fdisk分区工具。

```
fdisk 磁盘
```

命令示例：

```
fdisk /dev/vdb
```

回显类似如下信息：

```
[root@ecs-test-0001 ~]# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).
```

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Command (m for help):
```

步骤4 执行以下步骤，删除待扩大的分区。

1. 输入“d”，按“Enter”，删除磁盘分区。

回显类似如下信息：

```
Command (m for help): d
Partition number (1,2, default 2):
```

2. 输入分区编号，此处以分区编号选择“2”为例，按“Enter”，删除磁盘分区。

回显类似如下信息：

```
Partition number (1,2, default 2): 2
Partition 2 is deleted
```

```
Command (m for help):
```

说明

删除分区后，请参考以下操作步骤扩大原有分区，则不会导致数据盘内数据的丢失。

步骤5 输入“n”，按“Enter”，开始新建分区。

回显类似如下信息：

```
Command (m for help): n
Partition type:
 p primary (1 primary, 0 extended, 3 free)
```

```
e extended
Select (default p):
```

表示磁盘有两种分区类型：

- “p”表示主分区。
- “e”表示扩展分区。

📖 说明

磁盘使用MBR分区形式，最多可以创建4个主分区，或者3个主分区加1个扩展分区，扩展分区不可以直接使用，需要划分成若干个逻辑分区才可以使用。

磁盘使用GPT分区形式时，没有主分区、扩展分区以及逻辑分区之分。

步骤6 此处分区类型需要与原分区保持一致，以原分区类型是“主分区”为例，输入“p”，按“Enter”，开始重新创建一个主分区。

回显类似如下信息：

```
Select (default p): p
Partition number (2-4, default 2):
```

“Partition number”表示主分区编号。

步骤7 此处分区编号需要与原分区保持一致，以原分区编号是“2”为例，输入分区编号“2”，按“Enter”。

回显类似如下信息：

```
Partition number (2-4, default 2): 2
First sector (209715200-482344959, default 209715200):
```

“First sector”表示起始磁柱值。

📖 说明

以下操作会导致数据丢失：

- 选择的起始磁柱值与原分区的值不一致。
- 选择的截止磁柱值小于原分区的值。

步骤8 此处必须与原分区保持一致，输入**步骤1**中记录的初始磁柱值209715200，此处该值也为默认值，按“Enter”。

回显类似如下信息：

```
First sector (209715200-482344959, default 209715200):
Using default value 209715200
Last sector, +sectors or +size{K,M,G} (209715200-482344959, default 482344959):
```

“Last sector”表示截止磁柱值。

步骤9 此处截止磁柱值应大于等于**步骤1**中记录的截止磁柱值314572799，以选择默认截止磁柱值482344959为例，按“Enter”。

回显类似如下信息：

```
Using default value 209715200
Last sector, +sectors or +size{K,M,G} (209715200-482344959, default 482344959):
Using default value 482344959
Partition 2 of type Linux and of size 130 GiB is set
```

```
Command (m for help):
```

表示分区完成。

步骤10 输入“p”，按“Enter”，查看分区的详细信息。

回显类似如下信息：

```
Command (m for help): p

Disk /dev/vdb: 247.0 GB, 246960619520 bytes, 482344960 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x38717fc1

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048    209715199    104856576   83  Linux
/dev/vdb2           209715200    482344959    136314880   83  Linux
```

```
Command (m for help):
```

步骤11 输入“w”，按“Enter”，将分区结果写入分区表中。

回显类似如下信息：

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

说明

如果之前分区操作有误，请输入“q”，则会退出fdisk分区工具，之前的分区结果将不会被保留。

步骤12 执行以下命令，将新的分区表变更同步至操作系统。

partprobe

步骤13 根据磁盘的文件系统，分别执行以下操作。

- 若磁盘文件系统为ext*，请执行以下步骤。
 - a. 执行以下命令，检查磁盘分区文件系统的正确性。

e2fsck -f 磁盘分区

命令示例：

e2fsck -f /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# e2fsck -f /dev/vdb2
e2fsck 1.42.9 (28-Dec-2013)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vdb2: 11/3276800 files (0.0% non-contiguous), 251790/13107200 blocks
```

- b. 执行以下命令，扩展磁盘分区文件系统的大小。

resize2fs 磁盘分区

命令示例：

resize2fs /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# resize2fs /dev/vdb2
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/vdb2 to 34078720 (4k) blocks.
The filesystem on /dev/vdb2 is now 34078720 blocks long.
```

- c. （可选）执行以下命令，新建挂载目录。
若需要挂载至新建目录下，执行该操作。

mkdir 挂载目录

以新建挂载目录“/mnt/test”为例：

mkdir /mnt/test

- d. 执行以下命令，挂载磁盘分区。

mount 磁盘分区 挂载目录

以挂载分区“/dev/vdb2”至“/mnt/test”为例：

mount /dev/vdb2 /mnt/test

📖 说明

新增加的分区挂载到不为空的目录时，该目录下原本的子目录和文件会被隐藏，所以，新增的分区最好挂载到空目录或者新建目录。如确实要挂载到不为空的目录，可将该目录下的子目录和文件临时移动到其他目录下，新分区挂载成功后，再将子目录和文件移动回来。

- 若磁盘文件系统为xfs，请执行以下步骤。

- a. （可选）执行以下命令，新建挂载目录。
若需要挂载至新建目录下，执行该操作。

mkdir 挂载目录

以新建挂载目录“/mnt/test”为例：

mkdir /mnt/test

- b. 执行以下命令，挂载磁盘分区。

mount 磁盘分区 挂载目录

以挂载分区“/dev/vdb2”至“/mnt/test”为例：

mount /dev/vdb2 /mnt/test

📖 说明

新增加的分区挂载到不为空的目录时，该目录下原本的子目录和文件会被隐藏，所以，新增的分区最好挂载到空目录或者新建目录。如确实要挂载到不为空的目录，可将该目录下的子目录和文件临时移动到其他目录下，新分区挂载成功后，再将子目录和文件移动回来。

- c. 执行以下命令，扩展磁盘分区文件系统的大小。

sudo xfs_growfs 磁盘分区

命令示例：

sudo xfs_growfs /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# sudo xfs_growfs /dev/vdb2
meta-data=/dev/vdb2          isize=512    agcount=4, agsize=3276800 blks
=                               sectsz=512   attr=2, projid32bit=1
=                               crc=1      finobt=0, spinodes=0
data =                          bsize=4096  blocks=13107200, imaxpct=25
=                               sunit=0    swidth=0 blks
naming  =version2              bsize=4096  ascii-ci=0 ftype=1
log     =internal              bsize=4096  blocks=6400, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                 extsz=4096  blocks=0, rtextents=0
data blocks changed from 13107200 to 34078720.
```

步骤14 执行以下命令，查看挂载结果。

df -TH

回显类似如下信息：

```
[root@ecs-test-0001 ~]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/vda1       ext4      43G   1.9G  39G   5% /
devtmpfs        devtmpfs  2.0G   0   2.0G   0% /dev
tmpfs           tmpfs     2.0G   0   2.0G   0% /dev/shm
tmpfs           tmpfs     2.0G   9.1M  2.0G   1% /run
tmpfs           tmpfs     2.0G   0   2.0G   0% /sys/fs/cgroup
tmpfs           tmpfs     398M   0   398M   0% /run/user/0
/dev/vdb1       ext4     106G   63M  101G   1% /mnt/sdc
/dev/vdb2       ext4     138G   63M  131G   1% /mnt/test
```

📖 说明

云服务器重启后，挂载会失效。您可以修改“/etc/fstab”文件，将新建磁盘分区设置为开机自动挂载，请参见[设置开机自动挂载磁盘分区](#)。

----结束

新增 GPT 分区

数据盘“/dev/vdb”原有容量100GB，只有一个分区“/dev/vdb1”。将数据盘容量扩大至150GB，本示例为新增的50GB分配新的GPT分区“/dev/vdb2”。

步骤1 执行以下命令，查看磁盘的分区信息。

lsblk

回显类似如下信息：

```
[root@ecs-test-0001 ~]# lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda   253:0   0  40G  0 disk
├─vda1 253:1   0  40G  0 part /
vdb   253:16  0 150G  0 disk
├─vdb1 253:17  0 100G  0 part /mnt/sdc
```

步骤2 执行以下命令，进入parted分区工具。

parted 磁盘

命令示例：

parted /dev/vdb

回显类似如下信息：

```
[root@ecs-test-0001 ~]# parted /dev/vdb
GNU Parted 3.1
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

步骤3 输入“unit s”，按“Enter”，设置磁盘的计量单位为磁柱。

步骤4 输入“p”，按“Enter”，查看当前磁盘分区情况。

回显类似如下信息：

```
(parted) unit s
(parted) p
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another operating system believes the disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
```

```
Fix/Ignore/Cancel? Fix
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of
the space (an extra 104857600
blocks) or continue with the current setting?
Fix/Ignore? Fix
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 314572800s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start End      Size      File system Name  Flags
  1      2048s 209713151s 209711104s ext4    test

(parted)
```

记录已有分区的截止磁柱值 (End)，本示例中已有分区“/dev/vdb1”的截止磁柱值为209713151s。

- 若系统出现以下Error，请输入“Fix”。

```
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that
another operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
```

GPT分区表信息存储在磁盘开头，为了减少分区表损坏的风险，同时在磁盘末尾会备份一份。当磁盘容量扩大后，末尾位置也会随之变化，因此需要根据系统提示输入“Fix”，将分区表信息的备份文件挪到新的磁盘末尾位置。

- 若系统出现以下Warning，请输入“Fix”。

```
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all
of the space (an extra 104857600
blocks) or continue with the current setting?
Fix/Ignore? Fix
```

根据系统提示输入“Fix”，系统会自动将磁盘扩容部分的容量设置为GPT。

步骤5 以为新增容量分配一整个分区为例，执行以下命令，按“Enter”。

```
mkpart 磁盘分区名称 起始磁柱值 截止磁柱值
```

命令示例：

```
mkpart data 209713152s 100%
```

由于**步骤4**中，已有分区“dev/vdb1”的截止磁柱值为“209713151s”，因此对于新增分区“dev/vdb2”，起始磁柱值设置为“209713152s”，截止磁柱值设置为“100%”。此处仅供参考，您可以根据业务需要自行规划磁盘分区数量及容量。

回显类似如下信息：

```
(parted) mkpart data 209713152s 100%
(parted)
```

说明

获取最大截止磁柱值的方法如下：

- 参考**步骤2~步骤4**，查询磁盘的最大截止磁柱值。
- 可以输入-1s或者100%，即默认为磁盘的最大截止磁柱值。

步骤6 输入“p”，按“Enter”，查看新建分区。

回显类似如下信息：

```
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 314572800s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

```
Disk Flags:

Number Start      End          Size         File system Name  Flags
 1    2048s    209713151s 209711104s ext4      test
 2    209713152s 314570751s 104857600s          data

(parted)
```

步骤7 输入“q”，按“Enter”，退出parted分区工具。

步骤8 执行以下命令，为新建分区设置文件系统。

mkfs -t 文件系统 磁盘分区

- ext*文件系统命令示例：

mkfs -t ext4 /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb2
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

- xfs文件系统命令示例：

mkfs -t xfs /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# mkfs -t xfs /dev/vdb2
meta-data=/dev/vdb2          isize=512    agcount=4, agsize=3276800 blks
        =                   sectsz=512   attr=2,    projid32bit=1
        =                   crc=1      finobt=0, sparse=0
data     =                   bsize=4096  blocks=13107200, imaxpct=25
        =                   sunit=0    swidth=0 blks
naming   =version2          bsize=4096  ascii-ci=0  ftype=1
log      =internal log     bsize=4096  blocks=6400, version=2
        =                   sectsz=512   sunit=0    blks, lazy-count=1
realtime =none              extsz=4096  blocks=0,  rtextents=0
```

格式化需要等待一段时间，请观察系统运行状态，若回显中进程提示为done，则表示格式化完成。

步骤9 (可选) 执行以下命令，新建挂载目录。

如果需要挂载至新建目录下，执行该操作。

mkdir 挂载目录

以新建挂载目录“/mnt/test”为例：

mkdir /mnt/test

步骤10 执行以下命令，挂载新建分区。

mount *磁盘分区* *挂载目录*

以挂载新建分区 “/dev/vdb2” 至 “/mnt/test” 为例：

mount /dev/vdb2 /mnt/test

说明

新增加的分区挂载到不为空的目录时，该目录下原本的子目录和文件会被隐藏，所以，建议将新增的分区挂载到空目录或者新建目录。如确实要挂载到不为空的目录，可将该目录下的子目录和文件临时移动到其他目录下，新分区挂载成功后，再将子目录和文件移动回来。

步骤11 执行以下命令，查看挂载结果。

df -TH

回显类似如下信息：

```
[root@ecs-test-0001 ~]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/vda1       ext4      43G   1.9G  39G   5% /
devtmpfs        devtmpfs 2.0G   0   2.0G   0% /dev
tmpfs           tmpfs     2.0G   0   2.0G   0% /dev/shm
tmpfs           tmpfs     2.0G   9.1M  2.0G   1% /run
tmpfs           tmpfs     2.0G   0   2.0G   0% /sys/fs/cgroup
tmpfs           tmpfs     398M   0   398M   0% /run/user/0
/dev/vdb1       ext4     106G   63M  101G   1% /mnt/sdc
/dev/vdb2       ext4     53G   55M   50G   1% /mnt/test
```

说明

云服务器重启后，挂载会失效。您可以修改 “/etc/fstab” 文件，将新建磁盘分区设置为开机自动挂载，请参见[设置开机自动挂载磁盘分区](#)。

----结束

扩大已有 GPT 分区

须知

扩大已有分区不会删除数据盘上的数据，但是需要通过umount命令先卸载已有分区，因此会影响线上业务运行。

数据盘 “/dev/vdb” 原有容量150GB，有两个分区 “/dev/vdb1 ” 和 “/dev/vdb2” 。将数据盘容量扩大至230GB，本示例将新增的80GB划分至已有的GPT分区内 “/dev/vdb2” 内。

扩容后的新增存储空间是添加在磁盘末尾的，对具有多个分区的的磁盘扩容时，只支持将新增容量划分至排在末尾的分区。

步骤1 执行以下命令，查看磁盘的分区信息。

lsblk

回显类似如下信息：

```
[root@ecs-test-0001 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
```

```
vda 253:0 0 40G 0 disk
└─vda1 253:1 0 40G 0 part /
vdb 253:16 0 230G 0 disk
└─vdb1 253:17 0 100G 0 part /mnt/sdc
└─vdb2 253:18 0 50G 0 part /mnt/test
```

查看回显中磁盘“/dev/vdb”的容量，扩容的容量是否已经包含在容量总和中。

- 若扩容的容量未在磁盘容量总和中，请参考[扩展SCSI数据盘的分区和文件系统 \(Linux内核低于3.6.0\)](#)章节刷新系统内容量。
- 若扩容的容量已在磁盘容量总和中，请执行[步骤2](#)。

步骤2 执行以下命令，卸载磁盘分区。

umount *磁盘分区*

命令示例：

umount /dev/vdb2

步骤3 执行以下命令，确认磁盘分区的卸载结果。

lsblk

回显类似如下信息：

```
[root@ecs-test-0001 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 253:0 0 40G 0 disk
└─vda1 253:1 0 40G 0 part /
vdb 253:16 0 230G 0 disk
└─vdb1 253:17 0 100G 0 part /mnt/sdc
└─vdb2 253:18 0 50G 0 part
```

步骤4 执行以下命令，进入parted分区工具。

parted *磁盘*

命令示例：

parted /dev/vdb

回显类似如下信息：

```
[root@ecs-test-0001 ~]# parted /dev/vdb
GNU Parted 3.1
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

步骤5 输入“unit s”，按“Enter”，设置磁盘的计量单位为磁柱。

步骤6 输入“p”，按“Enter”，查看当前磁盘分区情况。

回显类似如下信息：

```
(parted) unit s
(parted) p
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another
operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
Fix/Ignore/Cancel? Fix
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of
the space (an extra 167772160
blocks) or continue with the current setting?
Fix/Ignore? Fix
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 482344960s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

```
Disk Flags:

Number Start      End          Size         File system Name  Flags
 1    2048s    209713151s 209711104s  ext4     test
 2    209713152s 314570751s 104857600s  ext4     data

(parted)
```

记录待扩大分区“/dev/vdb2”的初始磁柱值（start）和截止磁柱值（End），在后续重新划分分区需要使用。本示例中待扩大分区的初始磁柱值为209713152s，截止磁柱值为314570751s。

- 若系统出现以下Error，请输入“Fix”。

```
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that
another operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
```

GPT分区表信息存储在磁盘开头，为了减少分区表损坏的风险，同时在磁盘末尾会备份一份。当磁盘容量扩大后，末尾位置也会随之变化，因此需要根据系统提示输入“Fix”，将分区表信息的备份文件挪到新的磁盘末尾位置。

- 若系统出现以下Warning，请输入“Fix”。

```
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all
of the space (an extra 104857600
blocks) or continue with the current setting?
Fix/Ignore? Fix
```

根据系统提示输入“Fix”，系统会自动将磁盘扩容部分的容量设置为GPT。

步骤7 输入“rm”和分区编号，此处以“2”为例，按“Enter”。

回显类似如下信息：

```
(parted) rm
Partition number? 2
(parted)
```

步骤8 重新划分分区，执行以下命令，按“Enter”。

mkpart *磁盘分区名称 起始磁柱值 截止磁柱值*

命令示例：

mkpart data 209713152s 100%

- 起始磁柱值需要和原分区的取值保持一致，输入**步骤6**中记录的起始磁柱值“209713152s”。
- 截止磁柱值必须大于原分区的取值，即大于**步骤6**中记录的截止磁柱值“314570751s”，此处以将所有新增容量划分至分区“dev/vdb2”为例，输入100%。

回显类似如下信息：

```
(parted) mkpart data 209713152s 100%
(parted)
```

📖 说明

以下操作会导致数据丢失：

- 选择的起始磁柱值与原分区的值不一致。
- 选择的截止磁柱值小于原分区的值。

步骤9 输入“p”，按“Enter”，查看分区信息。

回显类似如下信息：

```
(parted) p
Model: Virtio Block Device (virtblk)
```

```
Disk /dev/vdb: 482344960s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start      End          Size         File system  Name  Flags
 1     2048s    209713151s  209711104s  ext4        test
 2     209713152s  482342911s  272629760s  ext4        data

(parted)
```

步骤10 输入“q”，按“Enter”，退出parted分区工具。

步骤11 根据磁盘的文件系统，分别执行以下操作。

- 若磁盘文件系统为ext*，请执行以下步骤。
 - a. 执行以下命令，检查磁盘分区文件系统的正确性。

e2fsck -f 磁盘分区

命令示例：

e2fsck -f /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# e2fsck -f /dev/vdb2
e2fsck 1.42.9 (28-Dec-2013)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vdb2: 11/3276800 files (0.0% non-contiguous), 251790/13107200 blocks
```

- b. 执行以下命令，扩展磁盘分区文件系统的大小。

resize2fs 磁盘分区

命令示例：

resize2fs /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# resize2fs /dev/vdb2
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/vdb2 to 34078720 (4k) blocks.
The filesystem on /dev/vdb2 is now 34078720 blocks long.
```

- c. (可选) 执行以下命令，新建挂载目录。

若需要挂载至新建目录下，执行该操作。

mkdir 挂载目录

以新建挂载目录“/mnt/test”为例：

mkdir /mnt/test

- d. 执行以下命令，挂载磁盘分区。

mount 磁盘分区 挂载目录

以挂载分区“/dev/vdb2”至“/mnt/test”为例：

mount /dev/vdb2 /mnt/test

说明

新增加的分区挂载到不为空的目录时，该目录下原本的子目录和文件会被隐藏，所以，新增的分区最好挂载到空目录或者新建目录。如确实要挂载到不为空的目录，可将该目录下的子目录和文件临时移动到其他目录下，新分区挂载成功后，再将子目录和文件移动回来。

- 若磁盘文件系统为xfs，请执行以下步骤。
 - a. (可选) 执行以下命令，新建挂载目录。
若需要挂载至新建目录下，执行该操作。
mkdir 挂载目录
以新建挂载目录“/mnt/test”为例：
mkdir /mnt/test
 - b. 执行以下命令，挂载磁盘分区。
mount 磁盘分区 挂载目录
以挂载分区“/dev/vdb2”至“/mnt/test”为例：
mount /dev/vdb2 /mnt/test

📖 说明

新增加的分区挂载到不为空的目录时，该目录下原本的子目录和文件会被隐藏，所以，新增的分区最好挂载到空目录或者新建目录。如确实要挂载到不为空的目录，可将该目录下的子目录和文件临时移动到其他目录下，新分区挂载成功后，再将子目录和文件移动回来。

- c. 执行以下命令，扩展磁盘分区文件系统的大小。

sudo xfs_growfs 磁盘分区

命令示例：

sudo xfs_growfs /dev/vdb2

回显类似如下信息：

```
[root@ecs-test-0001 ~]# sudo xfs_growfs /dev/vdb2
meta-data=/dev/vdb2          isize=512    agcount=4, agsize=3276800 blks
        =                   sectsz=512   attr=2,    projid32bit=1
        =                   crc=1      finobt=0, spinodes=0
data     =                   bsize=4096  blocks=13107200, imaxpct=25
        =                   sunit=0     swidth=0 blks
naming   =version2          bsize=4096  ascii-ci=0  ftype=1
log      =internal         bsize=4096  blocks=6400, version=2
        =                   sectsz=512   sunit=0    blks, lazy-count=1
realtime =none             extsz=4096  blocks=0,  rtextents=0
data blocks changed from 13107200 to 34078720.
```

步骤12 执行以下命令，查看挂载结果。

df -TH

回显类似如下信息：

```
[root@ecs-test-0001 ~]# df -TH
Filesystem Type      Size  Used Avail Use% Mounted on
/dev/vda1  ext4     43G  1.9G  39G   5% /
devtmpfs  devtmpfs 2.0G   0  2.0G   0% /dev
tmpfs     tmpfs    2.0G   0  2.0G   0% /dev/shm
tmpfs     tmpfs    2.0G  9.1M  2.0G   1% /run
tmpfs     tmpfs    2.0G   0  2.0G   0% /sys/fs/cgroup
tmpfs     tmpfs    398M   0  398M   0% /run/user/0
/dev/vdb1  ext4    106G  63M  101G   1% /mnt/sdc
/dev/vdb2  ext4    138G  63M  131G   1% /mnt/test
```

📖 说明

云服务器重启后，挂载会失效。您可以修改“/etc/fstab”文件，将新建磁盘分区设置为开机自动挂载，请参见[设置开机自动挂载磁盘分区](#)。

----**结束**

设置开机自动挂载磁盘分区

您可以通过配置fstab文件，设置弹性云服务器系统启动时自动挂载磁盘分区。已有数据的弹性云服务器也可以进行设置，该操作不会影响现有数据。

本文介绍如何在fstab文件中使用UUID来设置自动挂载磁盘分区。不建议采用在“/etc/fstab”直接指定设备名（比如/dev/vdb1）的方法，因为云中设备的顺序编码在关闭或者开启弹性云服务器过程中可能发生改变，例如/dev/vdb1可能会变成/dev/vdb2，可能会导致弹性云服务器重启后不能正常运行。

📖 说明

UUID（universally unique identifier）是Linux系统为磁盘分区提供的唯一的标识字符串。

步骤1 执行如下命令，查询磁盘分区的UUID。

blkid 磁盘分区

以查询磁盘分区“/dev/vdb1”的UUID为例：

blkid /dev/vdb1

回显类似如下信息：

```
[root@ecs-test-0001 ~]# blkid /dev/vdb1
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
```

记录下回显中磁盘分区“/dev/vdb1”的UUID，方便后续步骤使用。

步骤2 执行以下命令，使用VI编辑器打开“fstab”文件。

vi /etc/fstab

步骤3 按“i”，进入编辑模式。

步骤4 将光标移至文件末尾，按“Enter”，添加如下内容。

```
UUID=0b3040e2-1367-4abb-841d-ddb0b92693df /mnt/sdc ext4 defaults 0 2
```

以上内容仅为示例，具体请以实际情况为准，参数说明如下：

- 第一列为UUID，此处填写**步骤1**中查询到的磁盘分区的UUID。
- 第二列为磁盘分区的挂载目录，可以通过**df -TH**命令查询。
- 第三列为磁盘分区的文件系统格式，可以通过**df -TH**命令查询。
- 第四列为磁盘分区的挂载选项，此处通常设置为defaults即可。
- 第五列为Linux dump备份选项。
 - 0表示不使用Linux dump备份。现在通常不使用dump备份，此处设置为0即可。
 - 1表示使用Linux dump备份。
- 第六列为fsck选项，即开机时是否使用fsck检查磁盘。
 - 0表示不检验。
 - 挂载点为（/）根目录的分区，此处必须填写1。
根分区设置为1，其他分区只能从2开始，系统会按照数字从小到大依次检查下去。

步骤5 按“ESC”后，输入“:wq”，按“Enter”。

保存设置并退出编辑器。

步骤6 执行以下步骤，验证自动挂载功能。

1. 执行如下命令，卸载已挂载的分区。

```
umount 磁盘分区
```

命令示例：

```
umount /dev/vdb1
```

2. 执行如下命令，将“/etc/fstab”文件所有内容重新加载。

```
mount -a
```

3. 执行如下命令，查询文件系统挂载信息。

```
mount | grep 挂载目录
```

命令示例：

```
mount | grep /mnt/sdc
```

回显类似如下信息，说明自动挂载功能生效：

```
root@ecs-test-0001 ~]# mount | grep /mnt/sdc  
/dev/vdb1 on /mnt/sdc type ext4 (rw,relatime,data=ordered)
```

----结束

4.4 扩展 SCSI 数据盘的分区和文件系统（Linux 内核低于 3.6.0）

操作场景

通过云服务管理控制台扩容成功后，仅扩大了云硬盘的存储容量，因此需要参考本章节操作扩展分区和文件系统。

对于Linux操作系统而言，需要将扩容部分的容量划分至已有分区内，或者为扩容部分的云硬盘分配新的分区。

本文以“CentOS 7.4 64位”操作系统为例，提供针对SCSI数据盘的MBR分区操作指导。不同操作系统的操作可能不同，本文仅供参考，具体操作步骤和差异请参考对应操作系统的产品文档。

- [新增MBR分区](#)
- [扩大已有MBR分区](#)

须知

- 扩容时请谨慎操作，误操作可能会导致数据丢失或者异常，建议扩容前对数据进行备份，可以使用CBR或者快照功能，CBR请参见[管理备份云硬盘](#)，快照功能请参见[创建快照（公测）](#)。
- 当操作系统内核低于3.6.0时，扩大已有MBR分区需要reboot重启，扩展分区和文件系统才会生效，会中断业务。reboot重启后，新增容量会自动扩展至系统盘末尾分区内。
- 如果您不希望重启弹性云服务器来扩展分区和文件系统，您可以先将该数据盘中的业务数据迁移至弹性云服务器中的其他磁盘，然后卸载该数据盘，并将其挂载至其他内核大于3.6.0的弹性云服务器上来扩展磁盘分区和文件系统，扩展完成后再将其挂载回原始弹性云服务器，最后再将业务数据迁移回该磁盘。迁移数据有风险，请提前做好备份。扩容内核大于3.6.0的弹性云服务器上的分区和文件系统请参见[扩展磁盘分区和文件系统（Linux）](#)。

前提条件

- 已通过管理控制台扩容系统盘容量，并已挂载至云服务器，请参见[扩容云硬盘容量](#)。
- 已登录弹性云服务器。
 - 登录弹性云服务器请参见[登录弹性云服务器](#)。
 - 登录裸金属服务器请参见[登录裸金属服务器](#)。

新增 MBR 分区

数据盘“/dev/sda”原有容量50GB，只有一个分区“/dev/sda1”。将数据盘容量扩大至100GB，本示例为新增的50GB分配新的MBR分区“/dev/sda2”。

步骤1 执行以下命令，查看磁盘的分区信息。

fdisk -l

回显类似如下信息：

```
[root@ecs-scsi ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000bcb4e

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *         2048     83886079     41942016   83  Linux

Disk /dev/sda: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x915ffe6a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1          2048    104857599     52427776   83  Linux
```

查看回显中SCSI数据盘“/dev/sda”的容量，扩容的容量是否已经包含在容量总和中。

- 若扩容的容量未在数据盘容量总和中，请执行**步骤2**刷新系统内容量。
- 若扩容的容量已在数据盘容量总和中，请执行**步骤3**。

步骤2（可选）执行以下步骤，刷新系统内SCSI数据盘的容量。

1. 执行以下命令，刷新云服务器内SCSI数据盘的容量。

```
echo 1 > /sys/class/scsi_device/%d:%d:%d:%d/device/rescan &
```

其中“%d:%d:%d:%d”为“/sys/class/scsi_device/”路径下的文件夹，执行 **ll /sys/class/scsi_device/**命令获取。

回显类似如下信息，“2:0:0:0”即为待获取的文件夹。

```
cs-xen-02:/sys/class/scsi_device # ll /sys/class/scsi_device/
total 0
lrwxrwxrwx 1 root root 0 Sep 26 11:37 2:0:0:0 -> ../../devices/xen/vscsi-2064/host2/target2:0:0/2:0:0:0/
scsi_device/2:0:0:0
```

命令示例：

```
echo 1 > /sys/class/scsi_device/2:0:0:0/device/rescan &
```

2. 刷新完成后，执行以下命令，再次查看磁盘分区信息。

```
fdisk -l
```

若扩容的容量已在数据盘容量总和中，请执行**步骤3**。

步骤3 执行以下命令，进入fdisk分区工具。

```
fdisk 磁盘
```

命令示例：

```
fdisk /dev/sda
```

回显类似如下信息：

```
[root@ecs-scsi ~]# fdisk /dev/sda
Welcome to fdisk (util-linux 2.23.2).
```

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Command (m for help):
```

步骤4 输入“n”，按“Enter”，开始新建分区。

回显类似如下信息：

```
Command (m for help): n
Partition type:
  p primary (1 primary, 0 extended, 3 free)
  e extended
Select (default p):
```

表示磁盘有两种分区类型：

- “p”表示主分区。
- “e”表示扩展分区。

说明

磁盘使用MBR分区形式，最多可以创建4个主分区，或者3个主分区加1个扩展分区，扩展分区不可以直接使用，需要划分成若干个逻辑分区才可以使用。

磁盘使用GPT分区形式时，没有主分区、扩展分区以及逻辑分区之分。

步骤5 以创建一个主分区为例，输入“p”，按“Enter”。

回显类似如下信息:

```
Select (default p): p
Partition number (2-4, default 2):
```

“Partition number”表示主分区编号，可以选择2-4，由于1已被使用，此处从2开始。

步骤6 以分区编号选择“2”为例，输入分区编号“2”，按“Enter”。

回显类似如下信息:

```
Partition number (2-4, default 2): 2
First sector (104857600-209715199, default 104857600):
```

“First sector”表示起始磁柱值，可以选择104857600-209715199，默认为104857600。

步骤7 输入新分区的起始磁柱值，以使用默认起始磁柱值为例，按“Enter”。

系统会自动提示分区可用空间的起始磁柱值和截止磁柱值，可以在该区间内自定义，或者使用默认值。起始磁柱值必须小于分区的截止磁柱值。

回显类似如下信息:

```
First sector (104857600-209715199, default 104857600):
Using default value 104857600
Last sector, +sectors or +size{K,M,G} (104857600-209715199, default 209715199):
```

“Last sector”表示截止磁柱值，可以选择104857600-209715199，默认为209715199。

步骤8 输入新分区的截止磁柱值，以使用默认截止磁柱值为例，按“Enter”。

系统会自动提示分区可用空间的起始磁柱值和截止磁柱值，可以在该区间内自定义，或者使用默认值。起始磁柱值必须小于分区的截止磁柱值。

回显类似如下信息:

```
Last sector, +sectors or +size{K,M,G} (104857600-209715199, default 209715199):
Using default value 209715199
Partition 2 of type Linux and of size 50 GiB is set
```

Command (m for help):

步骤9 输入“p”，按“Enter”，查看新建分区。

回显类似如下信息:

```
Command (m for help): p

Disk /dev/sda: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x915ffe6a
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		2048	104857599	52427776	83	Linux
/dev/sda2		104857600	209715199	52428800	83	Linux

Command (m for help):

步骤10 输入“w”，按“Enter”，将分区结果写入分区表中。

回显类似如下信息:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

```
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

📖 说明

如果之前分区操作有误，请输入“q”，则会退出fdisk分区工具，之前的分区结果将不会被保留。

步骤11 执行以下命令，将新的分区表变更同步至操作系统。

partprobe

步骤12 执行以下命令，为新建分区设置文件系统。

mkfs -t 文件系统 磁盘分区

- ext*文件系统命令示例：

mkfs -t ext4 /dev/sda2

回显类似如下信息：

```
[root@ecs-scsi ~]# mkfs -t ext4 /dev/sda2
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

- xfs文件系统命名示例：

mkfs -t xfs /dev/sda2

回显类似如下信息：

```
[root@ecs-scsi ~]# mkfs -t xfs /dev/sda2
meta-data=/dev/sda2          isize=512    agcount=4, agsize=3276800 blks
       =                   sectsz=512   attr=2,    projid32bit=1
       =                   crc=1        finobt=0, sparse=0
data     =                   bsize=4096   blocks=13107200, imaxpct=25
       =                   sunit=0      swidth=0 blks
naming   =version2          bsize=4096   ascii-ci=0 ftype=1
log      =internal log     bsize=4096   blocks=6400, version=2
       =                   sectsz=512   sunit=0 blks, lazy-count=1
realtime =none              extsz=4096   blocks=0, rtextents=0
```

格式化需要等待一段时间，请观察系统运行状态，若回显中进程提示为done，则表示格式化完成。

步骤13 (可选) 执行以下命令，新建挂载目录。

如果需要挂载至新建目录下，执行该操作。

mkdir 挂载目录

以新建挂载目录 “/mnt/test” 为例：

```
mkdir /mnt/test
```

步骤14 执行以下命令，挂载新建分区。

```
mount 磁盘分区 挂载目录
```

以挂载新建分区 “/dev/sda2” 至 “/mnt/test” 为例：

```
mount /dev/sda2 /mnt/test
```

说明

新增加的分区挂载到不为空的目录时，该目录下原本的子目录和文件会被隐藏，所以，建议将新增的分区挂载到空目录或者新建目录。如确实要挂载到不为空的目录，可将该目录下的子目录和文件临时移动到其他目录下，新分区挂载成功后，再将子目录和文件移动回来。

步骤15 执行以下命令，查看挂载结果。

```
df -TH
```

回显类似如下信息：

```
[root@ecs-scsi ~]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/vda1       ext4      43G   2.0G   39G   5%  /
devtmpfs        devtmpfs  509M   0   509M   0%  /dev
tmpfs           tmpfs     520M   0   520M   0%  /dev/shm
tmpfs           tmpfs     520M   7.2M  513M   2%  /run
tmpfs           tmpfs     520M   0   520M   0%  /sys/fs/cgroup
tmpfs           tmpfs     104M   0   104M   0%  /run/user/0
/dev/sda1       ext4      53G   55M   50G   1%  /mnt/sdc
/dev/sda2       ext4      53G   55M   50G   1%  /mnt/test
```

说明

云服务器重启后，挂载会失效。您可以修改 “/etc/fstab” 文件，将新建磁盘分区设置为开机自动挂载，请参见[设置开机自动挂载磁盘分区](#)。

----结束

扩大已有 MBR 分区

须知

扩大已有分区不会删除数据盘上的数据，但是需要通过umount命令先卸载已有分区，因此会影响线上业务运行。

SCSI数据盘 “/dev/sda” 原有容量100GB，有两个分区 “/dev/sda1 ” 和 “/dev/sda2”。将数据盘容量扩大至150GB，本示例将新增的50GB划分至已有的MBR分区 “/dev/sda2” 内。

扩容后的新增存储空间是添加在磁盘末尾的，对具有多个分区的的磁盘扩容时，只支持将新增容量划分至排在末尾的分区。

步骤1 执行以下命令，查看磁盘的分区信息。

```
fdisk -l
```

回显类似如下信息:

```
[root@ecs-scsi ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000bcb4e

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *          2048     83886079     41942016   83  Linux

Disk /dev/sda: 161.1 GB, 161061273600 bytes, 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x915ffe6a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            2048    104857599     52427776   83  Linux
/dev/sda2    104857600    209715199     52428800   83  Linux
```

记录待扩大分区的起始磁柱值 (Start) 和截止磁柱值 (End), 本示例中待扩大分区 “/dev/sda2” 的起始磁柱值为 104857600, 截止磁柱值为 209715199。

查看回显中 SCSI 数据盘 “/dev/sda” 的容量, 扩容的容量是否已经包含在容量总和中。

- 若扩容的容量未在数据盘容量总和中, 请执行[步骤2](#)刷新系统内容量。
- 若扩容的容量已在数据盘容量总和中, 请记录待扩大分区的起始和截止磁柱值, 这些值在后续重新创建分区时需要使用, 记录完成后执行[步骤3](#)。

步骤2 (可选) 执行以下步骤, 刷新系统内 SCSI 数据盘的容量。

1. 执行以下命令, 刷新云服务器内 SCSI 数据盘的容量。

```
echo 1 > /sys/class/scsi_device/%d:%d:%d:%d/device/rescan &
```

其中 “%d:%d:%d:%d” 为 “/sys/class/scsi_device/” 路径下的文件夹, 执行 **ll /sys/class/scsi_device/** 命令获取。

回显类似如下信息, “2:0:0:0” 即为待获取的文件夹。

```
cs-xen-02:/sys/class/scsi_device # ll /sys/class/scsi_device/
total 0
lrwxrwxrwx 1 root root 0 Sep 26 11:37 2:0:0:0 -> ../../devices/xen/vscsi-2064/host2/target2:0:0/2:0:0:0/scsi_device/2:0:0:0
```

命令示例:

```
echo 1 > /sys/class/scsi_device/2:0:0:0/device/rescan &
```

2. 刷新完成后, 执行以下命令, 再次查看磁盘分区信息。

```
fdisk -l
```

若扩容的容量已在数据盘容量总和中, 请记录待扩大分区的起始和截止磁柱值, 这些值在后续重新创建分区时需要使用, 记录完成后执行[步骤3](#)。

步骤3 执行如下命令, 卸载磁盘分区。

```
umount 磁盘分区
```

命令示例:

```
umount /dev/sda2
```

步骤4 执行以下命令，进入fdisk分区工具。

fdisk 磁盘

命令示例：

fdisk /dev/sda

回显类似如下信息：

```
[root@ecs-scsi ~]# fdisk /dev/sda
Welcome to fdisk (util-linux 2.23.2).
```

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Command (m for help):
```

步骤5 执行以下步骤，删除待扩大的分区。

1. 输入“d”，按“Enter”，删除磁盘分区。

回显类似如下信息：

```
Command (m for help): d
Partition number (1,2, default 2):
```

2. 输入分区编号，此处以分区编号选择“2”为例，按“Enter”，删除磁盘分区。

回显类似如下信息：

```
Partition number (1,2, default 2): 2
Partition 2 is deleted
```

```
Command (m for help):
```

说明

删除分区后，请参考以下操作步骤扩大原有分区，则不会导致数据盘内数据的丢失。

步骤6 输入“n”，按“Enter”，开始新建分区。

回显类似如下信息：

```
Command (m for help): n
Partition type:
  p primary (1 primary, 0 extended, 3 free)
  e extended
Select (default p):
```

表示磁盘有两种分区类型：

- “p”表示主分区。
- “e”表示扩展分区。

说明

磁盘使用MBR分区形式，最多可以创建4个主分区，或者3个主分区加1个扩展分区，扩展分区不可以直接使用，需要划分成若干个逻辑分区才可以使用。

磁盘使用GPT分区形式时，没有主分区、扩展分区以及逻辑分区之分。

步骤7 此处分区类型需要与原分区保持一致，以原分区类型是主要分区为例，输入“p”，按“Enter”，开始重新创建一个主分区。

回显类似如下信息：

```
Select (default p): p
Partition number (2-4, default 2):
```

“Partition number”表示主分区编号。

步骤8 此处分区编号需要与原分区保持一致，以原分区编号是“2”为例，输入分区编号“2”，按“Enter”。

回显类似如下信息：

```
Partition number (2-4, default 2): 2
First sector (104857600-314572799, default 104857600):
```

“First sector”表示起始磁柱值。

📖 说明

以下操作会导致数据丢失：

- 选择的起始磁柱值与原分区的值不一致。
- 选择的截止磁柱值小于原分区的值。

步骤9 此处必须与原分区保持一致，输入**步骤1**或者**步骤2**记录的起始磁柱值104857600，此处该值也为默认值，按“Enter”。

回显类似如下信息：

```
First sector (104857600-314572799, default 104857600):
Using default value 104857600
Last sector, +sectors or +size{K,M,G} (104857600-314572799, default 314572799):
```

“Last sector”表示截止磁柱值。

步骤10 此处截止磁柱值应大于等于**步骤1**或者**步骤2**中记录的截止磁柱值209715199，以选择默认截止磁柱值314572799为例，按“Enter”。

回显类似如下信息：

```
Last sector, +sectors or +size{K,M,G} (104857600-314572799, default 314572799):
Using default value 314572799
Partition 2 of type Linux and of size 100 GiB is set
```

Command (m for help):

表示分区完成。

步骤11 输入“p”，按“Enter”，查看分区的详细信息。

回显类似如下信息：

```
Command (m for help): p

Disk /dev/sda: 161.1 GB, 161061273600 bytes, 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x915ffe6a
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		2048	104857599	52427776	83	Linux
/dev/sda2		104857600	314572799	104857600	83	Linux

Command (m for help):

步骤12 输入“w”，按“Enter”，将分区结果写入分区表中。

回显类似如下信息，表示分区创建完成。

```
Command (m for help): w
The partition table has been altered!
```

Calling ioctl() to re-read partition table.

```
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
```

```
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

📖 说明

如果之前分区操作有误，请输入“q”，则会退出fdisk分区工具，之前的分区结果将不会被保留。

步骤13 执行以下命令，将新的分区表变更同步至操作系统。

partprobe

步骤14 根据磁盘的文件系统，分别执行以下操作。

- 若磁盘文件系统为ext*，请执行以下步骤。
 - a. 执行以下命令，检查磁盘分区文件系统的正确性。

e2fsck -f 磁盘分区

命令示例：

e2fsck -f /dev/sda2

回显类似如下信息：

```
[root@ecs-scsi ~]# e2fsck -f /dev/sda2
e2fsck 1.42.9 (28-Dec-2013)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda2: 11/3276800 files (0.0% non-contiguous), 251790/13107200 blocks
```

- b. 执行以下命令，扩展磁盘分区文件系统的大小。

resize2fs 磁盘分区

命令示例：

resize2fs /dev/sda2

回显类似如下信息：

```
[root@ecs-scsi ~]# resize2fs /dev/sda2
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/sda2 to 26214400 (4k) blocks.
The filesystem on /dev/sda2 is now 26214400 blocks long.
```

- c. (可选) 执行以下命令，新建挂载目录。

若需要挂载至新建目录下，执行该操作。

mkdir 挂载目录

以新建挂载目录“/mnt/test”为例：

mkdir /mnt/test

- d. 执行以下命令，挂载磁盘分区。

mount 磁盘分区 挂载目录

以挂载分区“/dev/sda2”至“/mnt/test”为例：

mount /dev/sda2 /mnt/test

📖 说明

新增加的分区挂载到不为空的目录时，该目录下原本的子目录和文件会被隐藏，所以，建议将新增的分区挂载到空目录或者新建目录。如确实要挂载到不为空的目录，可将该目录下的子目录和文件临时移动到其他目录下，新分区挂载成功后，再将子目录和文件移动回来。

- 若磁盘文件系统为xfs，请执行以下步骤。
 - a. (可选) 执行以下命令，新建挂载目录。
若需要挂载至新建目录下，执行该操作。
mkdir 挂载目录
以新建挂载目录“/mnt/test”为例：
mkdir /mnt/test
 - b. 执行以下命令，挂载磁盘分区。
mount 磁盘分区 挂载目录
以挂载分区“/dev/vdb2”至“/mnt/test”为例：
mount /dev/vdb2 /mnt/test

📖 说明

新增加的分区挂载到不为空的目录时，该目录下原本的子目录和文件会被隐藏，所以，新增的分区最好挂载到空目录或者新建目录。如确实要挂载到不为空的目录，可将该目录下的子目录和文件临时移动到其他目录下，新分区挂载成功后，再将子目录和文件移动回来。

- c. 执行以下命令，扩展磁盘分区文件系统的大小。

sudo xfs_growfs 磁盘分区

命令示例：

sudo xfs_growfs /dev/sda2

回显类似如下信息：

```
[root@ecs-scsi ~]# sudo xfs_growfs /dev/sda2
meta-data=/dev/sda2      isize=512    agcount=4, agsize=3276800 blks
                =               sectsz=512   attr=2,   projid32bit=1
                =               crc=1      finobt=0, spinodes=0
data      =               bsize=4096  blocks=13107200, imaxpct=25
                =               sunit=0     swidth=0 blks
naming    =version2     bsize=4096  ascii-ci=0  ftype=1
log       =internal    bsize=4096  blocks=6400, version=2
                =               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none        extsz=4096  blocks=0,  rtextents=0
data blocks changed from 13107200 to 26214400df .
```

步骤15 执行以下命令，查看挂载结果。

df -TH

回显类似如下信息：

```
[root@ecs-scsi ~]# df -TH
Filesystem Type Size Used Avail Use% Mounted on
/dev/vda1 ext4 43G 2.0G 39G 5% /
devtmpfs devtmpfs 509M 0 509M 0% /dev
tmpfs tmpfs 520M 0 520M 0% /dev/shm
tmpfs tmpfs 520M 7.2M 513M 2% /run
tmpfs tmpfs 520M 0 520M 0% /sys/fs/cgroup
tmpfs tmpfs 104M 0 104M 0% /run/user/0
/dev/sda1 ext4 53G 55M 50G 1% /mnt/sdc
/dev/sda2 ext4 106G 63M 101G 1% /mnt/test
```

📖 说明

云服务器重启后，挂载会失效。您可以修改“/etc/fstab”文件，将新建磁盘分区设置为开机自动挂载，请参见[设置开机自动挂载磁盘分区](#)。

----**结束**

设置开机自动挂载磁盘分区

您可以通过配置fstab文件，设置弹性云服务器系统启动时自动挂载磁盘分区。已有数据的弹性云服务器也可以进行设置，该操作不会影响现有数据。

本文介绍如何在fstab文件中使用UUID来设置自动挂载磁盘分区。不建议采用在“/etc/fstab”直接指定设备名（比如/dev/vdb1）的方法，因为云中设备的顺序编码在关闭或者开启弹性云服务器过程中可能发生改变，例如/dev/vdb1可能会变成/dev/vdb2，可能会导致弹性云服务器重启后不能正常运行。

📖 说明

UUID (universally unique identifier) 是Linux系统为磁盘分区提供的唯一的标识字符串。

步骤1 执行如下命令，查询磁盘分区的UUID。

blkid 磁盘分区

以查询磁盘分区“/dev/vdb1”的UUID为例：

blkid /dev/vdb1

回显类似如下信息：

```
[root@ecs-test-0001 ~]# blkid /dev/vdb1
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
```

记录下回显中磁盘分区“/dev/vdb1”的UUID，方便后续步骤使用。

步骤2 执行以下命令，使用VI编辑器打开“fstab”文件。

vi /etc/fstab

步骤3 按“i”，进入编辑模式。

步骤4 将光标移至文件末尾，按“Enter”，添加如下内容。

```
UUID=0b3040e2-1367-4abb-841d-ddb0b92693df /mnt/sdc ext4 defaults 0 2
```

以上内容仅为示例，具体请以实际情况为准，参数说明如下：

- 第一列为UUID，此处填写**步骤1**中查询到的磁盘分区的UUID。
- 第二列为磁盘分区的挂载目录，可以通过**df -TH**命令查询。
- 第三列为磁盘分区的文件系统格式，可以通过**df -TH**命令查询。
- 第四列为磁盘分区的挂载选项，此处通常设置为defaults即可。
- 第五列为Linux dump备份选项。
 - 0表示不使用Linux dump备份。现在通常不使用dump备份，此处设置为0即可。
 - 1表示使用Linux dump备份。
- 第六列为fsck选项，即开机时是否使用fsck检查磁盘。
 - 0表示不检验。
 - 挂载点为(/)根目录的分区，此处必须填写1。
根分区设置为1，其他分区只能从2开始，系统会按照数字从小到大依次检查下去。

步骤5 按“ESC”后，输入“:wq”，按“Enter”。

保存设置并退出编辑器。

步骤6 执行以下步骤，验证自动挂载功能。

1. 执行如下命令，卸载已挂载的分区。

umount 磁盘分区

命令示例：

umount /dev/vdb1

2. 执行如下命令，将“/etc/fstab”文件所有内容重新加载。

mount -a

3. 执行如下命令，查询文件系统挂载信息。

mount | grep 挂载目录

命令示例：

mount | grep /mnt/sdc

回显类似如下信息，说明自动挂载功能生效：

```
root@ecs-test-0001 ~]# mount | grep /mnt/sdc  
/dev/vdb1 on /mnt/sdc type ext4 (rw,relatime,data=ordered)
```

----**结束**