



# 云硬盘 产品文档





# 文档目录

## 产品概述

产品概述

典型使用场景

产品功能

云硬盘类型简介

## 产品优势

可靠

弹性

高性能

易用性

快照备份

## 产品功能

## 使用指引

## 新手指引

新手指引

前提条件

新建云硬盘

连接到云服务器实例

格式化、创建文件系统并写入文件

连接到 Linux 类型实例

连接到 Windows 类型实例

## 云硬盘的生命周期

创建云硬盘

创建云硬盘

创建弹性云盘

使用控制台创建弹性云盘

使用快照创建弹性云盘

挂载云硬盘

使用控制台将弹性云硬盘连接到实例

部分已创建的云服务器实例无法识别弹性云盘的解决方案

使用自定义镜像及数据盘快照启动新实例时自动挂载数据盘（linux）

使用自定义镜像及数据盘快照启动新实例时自动挂载数据盘（windows）

初始化云硬盘

Linux系统下初始化云硬盘

Linux系统下初始化云硬盘



前提条件

查看已挂载的硬盘

(可选) 对磁盘进行分区操作

挂载硬盘

Windows系统下初始化云硬盘

Windows系统下初始化云硬盘

前提条件

Windows 2012 磁盘联机、分区和格式化

联机磁盘

(可选) 格式化磁盘

(可选) 对磁盘分区

使用自定义镜像及数据盘快照启动新实例时自动挂载数据盘

扩容云硬盘

扩容云硬盘

扩容类型为云硬盘的数据盘

通过云硬盘控制台扩容CBS数据盘

通过云服务器控制台扩容CBS数据盘

扩容类型为云硬盘的系统盘

扩容文件系统

扩容Linux文件系统

扩容Linux文件系统

GPT分区云硬盘扩容后修改分区指引

新空间格式化成一个独立GPT分区

查看数据盘信息

卸载已挂载数据盘

数据盘分区

格式化新建分区

新空间增加到已有分区中(GPT分区格式)

查看数据盘信息

卸载已挂载数据盘

数据盘分区

检查扩容后分区的文件系统

扩容文件系统

挂载新分区

MBR分区云硬盘扩容后修改分区指引

MBR分区云硬盘扩容后修改分区指引

新空间格式化成一个独立分区

查看数据盘信息



- 卸载所有已挂载的分区
- 数据盘分区
- 查看新分区
- 格式化新分区并创建文件系统
- 挂载新分区
- 添加新分区信息
- 将新空间增加到已有分区空间中
- 将新空间增加到已有分区空间中
- 卸载正在使用的硬盘分区
- 下载一键扩容工具
- 执行扩容工具
- 重新挂载扩容后的分区

#### 扩容Windows文件系统

- 扩容Windows文件系统
- 前提条件
- 新空间格式化成一个独立分区
- 新空间增加到已有分区空间中

#### 卸载弹性云盘

- 卸载弹性云盘
- 使用控制台卸载弹性云盘

#### 删除云硬盘

- 删除云硬盘
- 删除弹性云盘
- 删除弹性云盘

#### 快照

##### 什么是快照

- 什么是快照
- 快照的作用
- 使用快照的业务场景
- 注意事项

##### 创建快照

- 创建快照
- 使用控制台创建快照

##### 快照回滚

- 快照回滚
- 使用控制台回滚快照

##### 快照创建云硬盘

- 快照创建云硬盘



使用控制台从快照创建云硬盘

删除快照

删除快照

使用控制台删除快照

定期快照

定期快照策略说明

操作指引

创建定期快照策略

关联云硬盘

将自动快照转为永久保留的快照

共享快照

操作场景

注意事项

操作步骤

监控云硬盘

最佳实践

如何衡量云硬盘的性能

多块弹性云盘构建RAID组

多块弹性云盘构建LVM逻辑卷

常见问题

一般性问题

云硬盘使用和生命周期问题

快照常见问题

硬盘性能常见问题

词汇表



# 产品概述

## 产品概述

最近更新时间: 2023-03-13 15:06:22

云硬盘（Cloud Block Storage）为云服务器实例提供高效可靠的存储设备，它是一种高可用、高可靠、低成本、可定制化的块存储设备，可以作为云服务器的独立可扩展硬盘使用。它提供数据块级别的数据存储，采用三副本的分布式机制，为CVM提供数据可靠性保证。

CBS支持在可用区内自动复制，将您的数据备份在不同机器上，从而免除单个机器故障带来的数据丢失等问题，提高数据的可用性。您可通过控制台轻松购买、调整、管理您的云硬盘设备，并通过构建文件系统创建出大于单块CBS设备容量大小的存储空间。

建行云云硬盘（CBS）提供数据块级别的存储，可以挂载到同一可用区中任何运行中的实例上。云硬盘通常用作需要频繁和细粒度更新的数据（如文件系统、数据库等）的主存储设备，具有高可用、高可靠、高性能的特点。

建行云对用户可以使用的云硬盘数量和总存储量有相应的限制。了解相关限制的更多信息，请参阅使用约束。

云硬盘的生命周期独立于 CVM 实例，始终不受实例运行时间的影响；同时，用户可以将多块云硬盘连挂载至同一个实例，也可以将云硬盘从实例中断开并挂载到另一个实例。云硬盘支持随时在同一可用区的云服务器上进行挂载和卸载，挂载后即可被云服务器视为数据盘使用。



# 典型使用场景

最近更新时间: 2023-03-13 15:06:18

- 1.申请完的云服务器在使用过程中发现硬盘空间不够，可以通过申请一块或多块云硬盘挂载至云服务器上满足存储容量需求。
- 2.申请云服务器时不需要额外的存储空间，有存储需求的时候再通过申请云硬盘扩展云服务器的存储容量。
- 3.在多个服务器之间存在数据交换的诉求时，可以通过卸载云硬盘（数据盘）并重新挂载到其他服务器上来满足。
- 4.可以通过申请多块云硬盘并配置 LVM 逻辑卷来实现突破单块云硬盘存储容量上限。
- 5.可以通过申请多块云硬盘并配置 RAID 策略来实现突破单块云硬盘IO能力上限。



# 产品功能

最近更新时间: 2023-03-13 15:06:18

用户可灵活选择磁盘种类，并自行在磁盘上进行存储文件、搭建数据库等操作。

1. 2种磁盘选择：高性能云硬盘和 SSD 云硬盘。
2. 弹性挂载/卸载：所有云硬盘类型均支持弹性挂载、卸载，可在云服务器上挂载多块云硬盘搭建大容量的文件系统。
3. 弹性扩容：单盘最大支持 16 TB，您可随时对云硬盘进行扩容。
4. 快照备份：支持创建快照和快照回滚，及时备份关键数据；支持使用快照创建磁盘，可快速实现业务部署。



# 云硬盘类型简介

最近更新时间: 2023-03-13 15:06:17

- **高性能云硬盘**

高性能云硬盘是建行云推出的混合型存储类型，通过 Cache 机制提供接近固态存储的高性能存储能力，同时采用三副本的分布式机制保障数据可靠性。高性能云硬盘适用于高数据可靠性要求、普通中度性能要求的 Web/App 服务器、业务逻辑处理、中小型建站等中小型应用场景。

- **SSD 云硬盘**

SSD 云硬盘是建行云基于 NVMe SSD 存储介质提供的全闪型存储类型，其采用三副本的分布式机制，提供低时延、较高随机 IOPS 和吞吐量的 I/O 能力。SSD 云硬盘适用于对 I/O 性能有较高要求的场景。



# 产品优势

## 可靠

最近更新时间: 2023-03-13 15:06:17

在每个存储写入请求返回给用户之前，CBS 就已确保数据已被成功写入三份，且跨机架存储。后台数据复制机制能够保证任何一个副本故障时快速进行数据迁移恢复，时刻确保用户数据3份副本可用，可靠性领先业内水平，确保核心业务数据不丢失，为您提供安全放心的数据存储服务。



# 弹性

最近更新时间: 2023-03-13 15:06:17

您可以自由配置存储容量，按需扩容，且无须中断业务。

单磁盘容量为10 ~ 16000GB，单CVM 累计可挂载320TB 的存储，单CBS集群存储容量无上限，高效应对TB/PB级数据的大数据处理场景。



# 高性能

最近更新时间: 2023-03-13 15:06:17

SSD云硬盘采用NVMe标准高性能SSD，单盘提供26000随机 IOPS，260MB/s 吞吐，实现了超强性能与超高可靠性的集合。

高性能的SSD云盘助您轻松支撑业务侧高吞吐量的DB访问。



# 易用性

最近更新时间: 2023-03-13 15:06:17

通过简单的创建、挂载、卸载、删除等操作即可管理及使用云硬盘，节省人工管理部署成本。



# 快照备份

最近更新时间: 2023-03-13 15:06:17

您可以通过拍摄CBS云硬盘的时间点快照来备份您的数据，通过使用CBS快照创建新的CVM实例，大幅提升业务的灵活性。



# 产品功能

最近更新时间: 2023-03-13 15:06:17

建行云提供多样化存储设备，用户可灵活选择磁盘种类，并自行在磁盘上进行存储文件、搭建数据库等操作。

- 弹性挂载/卸载：普通云硬盘支持弹性挂载、卸载，可为一台云服务器搭建大容量的文件系统。
- 弹性扩容：单盘最大支持16TB的空间，您可随时对硬盘进行扩容。
- 快照备份：支持创建快照和快照回滚，及时备份关键数据；支持使用快照创建磁盘，可快速实现业务部署。



# 使用指引

最近更新时间: 2023-03-13 15:06:17

- 1.系统盘单块容量为 50GB-1024GB，以 1GB 为步长增长；弹性云硬盘（数据盘）单块容量为 10GB-16000GB，以 10GB 为步长增长。
- 2.单台云服务器最多支持挂载 20 块弹性云硬盘（数据盘），随云服务器一起创建并挂载的数据盘最多为 10 块。
- 3.单次最多可创建 50 块弹性云硬盘（数据盘）。
- 4.云服务器和云硬盘必须在同一个可用区内。
- 5.当对云硬盘进行读写时，IOPS、吞吐量可能其中一个指标先达到产品设计上限，具体哪个先到达上限与用户读写时的 block size 大小有关。具体的性能说明如下：
  - (1) 高性能云硬盘，能够达到的最大随机 IOPS 为 6000，最大吞吐量为150MB/s。能够达到的最大随机 IOPS = MIN (1800 + 存储容量 (GB) ×8, 6000) 能够达到的最大吞吐量 = MIN (100 + 存储容量 (GB) ×0.15, 150) MB/S
  - (2) SSD 云硬盘，能够达到的最大随机 IOPS 为 26000，最大吞吐量为260MB/s。例如，1TB 的 SSD 云硬盘，当使用 block size 为 4KB/8KB 的 I/O 读写时，IOPS 先达到最大值 26000（此时吞吐量达不到最大值 260MB/s）。当使用 block size 为 16KB 的 I/O 读写时，吞吐量先达到最大值 260MB/s（此时 IOPS 达不到最大值 26000）。
- 6.快照只能回滚到创建快照的源云硬盘上。云服务器关机状态下，系统盘快照可以回滚；云服务器关机状态或者关联云盘未挂载情况下，数据盘快照可以回滚。
- 7.只有数据盘快照可用来创建新的数据盘，使用快照新创建的云硬盘容量必须大于或等于创建快照的云硬盘容量。
- 8.快照只包含数据信息，不包含主机内存数据。
- 9.云硬盘创建后不可更改可用区，不支持跨可用区故障迁移。
10. 云服务器上操作系统缺省情况下不备份，如果针对标准云服务器上数据确需进行备份，可在可研时申请使用云硬盘的快照功能，合理设置定期备份频率，单个用户云硬盘数量（包含系统盘、数据盘）和快照数量比例不超过1:4，用户的单个定期快照策略绑定云盘数不超过500，避免在同一时间点大批量制作快照而导致快照失败。应用应定期对备份数据进行验证。



# 新手指引

## 新手指引

最近更新时间: 2023-03-13 15:14:10

本文档将以在武汉地域下创建一块空的弹性云盘 cbs-test（有关什么是弹性云盘可以参考云硬盘的分类，有关如何创建包含数据的云硬盘请参考快照创建云硬盘）、连接到云服务器实例、创建文件系统并向其中写入一个名为 yuncb.txt 的文件为例，帮助用户了解如何初步使用建行云云硬盘。



# 前提条件

最近更新时间: 2023-03-13 15:14:10

要根据本文档进行云硬盘的使用，请先确保您在创建云硬盘的地域和可用区（武汉一区）下有可用的运行状态云服务器实例。有关如何申请并启动云服务器实例，请参考[申请并启动云服务器](#)。



# 新建云硬盘

最近更新时间: 2023-03-13 15:14:10

本例通过控制台新建一块弹性普通云盘，有关如何创建云硬盘的更多内容，可以参考[创建云硬盘](#)。

1. 登录云硬盘控制台，点击开始新建。
2. 在弹出框中选择【武汉一区】、【云硬盘】、【20GB】点击【确定】按钮。
3. 在页面点击【确认】完成新建。您可在云硬盘列表页查看新建的云硬盘，您刚刚新建的弹性云盘默认未命名，并显示为待挂载状态。点击编辑小图标，将它命名为“cbs-test”。



# 连接到云服务器实例

最近更新时间: 2023-03-13 15:14:10

1. 登录建行云控制台。
2. 进入【云服务器】-【云硬盘】选项卡。
3. 在云硬盘列表页，点击刚刚创建的云硬盘后的【更多】-【挂载到云主机】按钮。
4. 在弹出框中选择需要挂载到的云服务器（见前提条件），点击【确定】按钮。



# 格式化、创建文件系统并写入文件 连接到 Linux 类型实例

最近更新时间: 2023-03-13 15:14:10

1. 执行 `fdisk -l` 命令，查看连接到实例的盘设备名称。找到创建的 20 GB 云硬盘，假设这里它的设备名为 `/dev/vdb`。

2. 格式化该设备（本例使用 EXT4 文件系统）：执行 `mkfs.ext4 /dev/vdb` 命令。

3. 挂载到 `/data` 挂载点下，执行以下命令：

```
mount /dev/vdb /data
```

4. 进入该设备，在其中写入一个名为 `yunccb.txt` 的文件，执行如下命令：

```
cd /data
```

```
vi yunccb.txt
```

编辑态下写入一些内容，如：“This is my first test”。按 ESC 退出编辑态后输入 `wq` 保存刚刚更改的内容，此时执行 `ls` 命令可发现刚刚的文件已经写入盘中了。

有关 Linux 系统下分区、格式化及创建文件系统的更多内容，请参考Linux系统下初始化云硬盘。

# 连接到 Windows 类型实例

最近更新时间: 2023-03-13 15:14:10

1. 找到类似（【开始】-）【服务器管理】-【存储/计算机管理】-【磁盘管理】的路径进入磁盘管理。
  2. 在创建的空弹性云盘上右键点击，选择【联机】。
  3. 右键点击该磁盘，在弹出的快捷菜单中选择【新建简单卷】，根据向导提示进行操作，输入整块磁盘的大小，点击【下一步】，选择文件系统（磁盘大于2TB时一定要选择GPT分区形式），格式化分区，点击【下一步】，点击【完成】按钮。
  4. 显示正在格式化，待格式化完成后，该弹性云盘即可被使用。
  5. 进入创建的盘中，新建一个名为 yunccb.txt的文件，输入您需要的内容，点击【保存】按钮。即完成写入文件操作。
- 有关 Windows 系统下系统分区、格式化、联机及创建文件系统的更多内容，请参考章节Windows系统下初始化云硬盘。



# 云硬盘的生命周期

## 创建云硬盘

## 创建云硬盘

最近更新时间: 2023-03-13 15:14:10

您可以创建云硬盘并将它连接到同一个可用区内的任何CVM实例。通过块储存设备映射，云硬盘被云服务器实例识别并使用。同时，用户也可基于先前创建的快照来启动新的云硬盘。有关更多信息，请参阅 [快照创建云硬盘](#)，云硬盘在创建后不需要预热即可达到其最高性能。

根据云硬盘类型的不同，创建云硬盘的方式也不同，有关云硬盘类型的更多内容请参考云硬盘的分类。



# 创建弹性云盘

## 使用控制台创建弹性云盘

最近更新时间: 2023-03-13 15:14:10

1. 登录云硬盘控制台，单击即可开始申请。
2. 在弹出框中选择地域/可用区、容量、申请数量单击【确定】按钮。
3. 在页面单击【确认】完成申请。您可在云硬盘列表页查看已申请的云硬盘，您刚刚申请的弹性云盘默认未命名，并显示为待挂载状态。

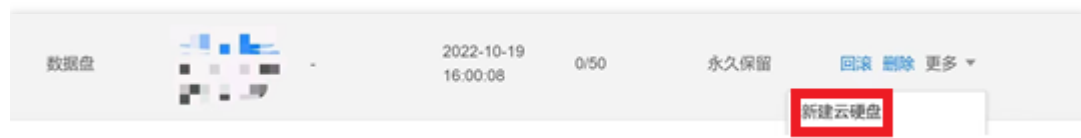


# 使用快照创建弹性云盘

最近更新时间: 2023-03-13 15:14:10

如果您需要在新创建的磁盘中保留数据盘快照数据，则可以选择这种方式。

1) 登录快照控制台，在您要使用的快照后，单击【新建云硬盘】即可开始申请。



2) 在弹出框中选择地域/可用区、容量、申请数量，单击【确定】按钮。

3) 在页面单击【确认】完成申请。您可在云硬盘列表页查看已申请的云硬盘，您刚刚申请的弹性云硬盘默认名称为 From snap-xxxxxxx，并显示为待挂载状态。



# 挂载云硬盘

## 使用控制台将弹性云硬盘连接到实例

最近更新时间: 2023-03-13 15:23:51

目前支持对作为数据盘的普通弹性云硬盘云盘进行挂载，不可挂载系统盘。

1. 登录建行云控制台。
2. 进入【云服务器】-【云硬盘】选项卡。
3. 在云硬盘列表页，点击状态为待挂载、支持挂载/卸载的云硬盘后的【更多】-【挂载到云主机】按钮进行单盘挂载；或在云硬盘列表页，勾选状态为待挂载、支持挂载/卸载的云硬盘，点击顶部【挂载】按钮进行批量挂载。
4. 在弹出框中选择需要挂载到的云服务器，点击【确定】按钮，等待挂载完毕即可登录云服务器查看云硬盘挂载状况。

云硬盘在挂载完后并不能马上使用，需要进行分区、格式化等一系列操作。具体操作方式请见：Linux系统下初始化云硬盘，Windows系统下初始化云硬盘。



# 部分已创建的云服务器实例无法识别弹性云盘的解决方案

最近更新时间: 2023-03-13 15:23:51

目前提供的所有镜像已经支持弹性云盘的连接/解挂操作。请注意，拔盘（卸载）前请先执行umount（Linux）或脱机（Windows）操作，否则可能出现再次挂载时无法识别的问题。



# 使用自定义镜像及数据盘快照启动新实例时自动挂载数据盘（linux）

最近更新时间: 2023-03-13 15:23:51

在启动新的云服务器实例时，如果用户指定 自定义镜像 及数据盘快照，建行云云硬盘可以支持启动云服务器实例后自动挂载（即不需要进行一系列的添加、分区、格式化等操作可直接读写数据盘）。用户需要在制作自定义镜像和数据盘快照前在原实例上进行一些操作，下文将详细描述。

在Linux系统下如果用户希望指定数据盘快照生产出来的云硬盘能够自动挂载至新的云服务器实例，指定的自定义镜像和数据盘快照必须满足以下要求：

数据盘在制作快照前 必须已经格式化过，也即在原云服务器上已经mount成功。

系统盘在制作自定义镜像前，需要在 /etc/rc.local 文件中添加以下命令，将数据盘挂载点写入文件中：

```
mkdir -p <mount-point>
```

```
mount <device-id> <mount-point>
```

其中：<mount-point>请填入文件系统的挂载点如/mydata，<device-id>请填入用户的实际文件分区位置，如/dev/vdb(无分区有文件系统的设备名)和/dev/vdb1(有分区有文件系统)。

只有同时满足以上两个条件才能保证新启动的Linux云服务器实例数据盘可以被自动识别和挂载。

# 使用自定义镜像及数据盘快照启动新实例时自动挂载数据盘（windows）

最近更新时间: 2023-03-13 15:23:51

在启动新的云服务器实例时，如果用户指定自定义镜像及数据盘快照，建行云云硬盘可以支持启动云服务器实例后自动挂载（即不需要进行一系列的添加、分区、格式化等操作可直接读写数据盘）。用户需要在制作自定义镜像和数据盘快照前在原实例上进行一些操作，下文将详细描述。

在Windows系统下如果用户希望指定数据盘快照生产出来的云硬盘能够自动挂载至新的云服务器实例，指定的自定义镜像和数据盘快照必须满足以下要求：

- 自定义镜像中的 SAN 策略为：onlineAll。建行云目前提供的 Windows 公有镜像已默认进行相关设置，但仍建议用户在制作自定义镜像前检查下此配置，检查方法如下：

```
PS C:\Users\Administrator>
PS C:\Users\Administrator> diskpart

Microsoft DiskPart 版本 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
在计算机上: [REDACTED]

DISKPART> san

SAN 策略: 使共享磁盘脱机

DISKPART> san policy=onlineall

DiskPart 已成功更改用于当前操作系统的 SAN 策略。

DISKPART> san

SAN 策略: 全部联机
```

- 数据盘在制作快照前必须已经被格式化为ntfs或fat32格式。

只有同时满足以上两个条件才能保证新启动的Windows云服务器实例数据盘可以被自动识别和联机。



# 初始化云硬盘

## Linux系统下初始化云硬盘

## Linux系统下初始化云硬盘

最近更新时间: 2023-03-13 15:43:25

实例可以识别连接的云硬盘并将其视为普通硬盘。用户可以使用任何文件系统将云块存储设备格式化、分区及创建文件系统。此后任何写入文件系统的数据均写入云硬盘中，并且对使用该设备的应用程序是透明的。本文档通过一个示例演示如何在云主机中使用块存储设备，并对一些需要注意的事项进行了详细的说明。您还可以从Windows系统下初始化云硬盘获得有关Windows实例上使用云硬盘的指引。

本示例使用的云主机启动时一并申请了一块20GB的系统盘和一块30GB的数据盘（即非弹性云硬盘），而后又在云硬盘控制台申请了一块10GB的弹性云盘。



# 前提条件

最近更新时间: 2023-03-13 15:43:25

- 请确保您已进行将云硬盘连接到CVM实例及登录Linux实例。
- 格式化后，数据盘中的数据将被全部清空。请在格式化之前，确保数据盘中没有数据或对重要数据已进行备份。为避免服务发生异常，格式化前请确保云服务器已停止对外服务。
- 当用户申请了多块云硬盘时，建议您对存放重要数据的弹性云盘设置自定义名称
- 用户可以在云硬盘控制台中根据自定义名称或者关联的云主机内网 IP 快速查找云硬盘。

本示例中，弹性云盘ID为ins-kjo6azag，名称为弹性云盘使用演示。

# 查看已挂载的硬盘

最近更新时间: 2023-03-13 15:43:25

## 1. 运行fdisk -l命令查看硬盘信息。

此时可以看到随云主机创建的非弹性云硬盘vdb和刚刚挂载的弹性云盘vdc都尚未格式化。

```
brw-rw---- 1 root disk    253,  16 Jul 21 16:25 vdb
brw-rw---- 1 root disk    253,  32 Jul 21 16:43 vdc
[root@VM_28_66-centos ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       20G   1.3G   18G   7% /
devtmpfs        912M    0   912M   0% /dev
tmpfs           921M   24K   921M   1% /dev/shm
tmpfs           921M   8.3M   912M   1% /run
tmpfs           921M    0   921M   0% /sys/fs/cgroup
tmpfs          185M    0   185M   0% /run/user/0
[root@VM_28_66-centos ~]# fdisk -l

Disk /dev/vda: 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 label type: dos
Disk identifier: 0x0005fc9a

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1    *        2048     41943039     20970496   83   Linux

Disk /dev/vdb: 32.2 GB, 32212254720 bytes, 62914560 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
```

## 2. 执行ls -l /dev/disk/by-id/ 命令，可以在此处看到弹性云盘与设备名的对应关系。注意，非弹性云盘目前不会在这里显示任何信息。

```
tmpfs          185M    0   185M   0% /run/user/0
[root@VM_28_66-centos ~]# ll /dev/disk/by-id/
total 0
lrwxrwxrwx 1 root root 9 Jul 21 16:43 virtio-disk-bm42ztpm -> ../../vdc
[root@VM_28_66-centos ~]#
```

硬盘从未进行初始化时，您需要先创建文件系统，然后才能够使用它。从快照创建的云硬盘中可能已经含有文件系



统，如果您在现有的文件系统上创建新的文件系统，则将覆盖原有的全部数据。

使用 `file -s device` 命令可列出特殊信息，例如文件系统类型。

```
sudo file -s /dev/xvdf
```

```
/dev/xvdf: data
```

如果前面的命令的输出仅显示该设备的 `data`，则说明设备上没有文件系统，您需要创建一个文件系统。您可继续下面的所有步骤。如果在包含文件系统的设备上运行此命令，则输出将有所不同，若返回形如 `Linux rev 1.0 ext4 filesystem data` 的输出则说明此硬盘上已经创建了文件系统，您可以跳过分区和格式化操作。

若磁盘已经初始化，可从 `System` 字段判断是否需要创建新的文件系统。如果 `System` 字段显示 `EXT3`、`EXT4` 等文件系统类型，则不需新建文件系统，可以跳过分区和格式化操作步骤。

## (可选) 对磁盘进行分区操作

最近更新时间: 2023-03-13 15:43:22

1. 对云硬盘进行分区操作。当然用户也可以无需分区直接进行格式化操作。这里我们演示了将弹性云盘划分为两个分区使用。执行以下命令：

```
fdisk /dev/vdb
```

按照界面的提示，依次输入“n”(新建分区)、“p”(新建扩展分区)、“1”(使用第1个主分区)，两次回车(使用默认配置)，输入“wq”(保存分区表)，回车开始分区。

这里是以创建 1 个分区为例，开发者也可以根据自己的需求创建多个分区。

```
[root@VM_124_230_centos ~]# fdisk /dev/vdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x2d8cd07a.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-104025, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-104025, default 104025):
Using default value 104025

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

calling ioctl() to re-read partition table.
Syncing disks.
[root@VM_124_230_centos ~]#
```



使用“fdisk -l”命令，即可查看到，新的分区 vdb1 已经创建完成。

```
[root@VM_124_230_centos ~]# fdisk -l

Disk /dev/vda: 8589 MB, 8589901824 bytes
255 heads, 63 sectors/track, 1044 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: 0xcd6e8236

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *          1         1044     8385898+   83   Linux

Disk /dev/vdb: 53.7 GB, 53687091200 bytes
16 heads, 63 sectors/track, 104025 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2d8cd07a

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1          1        104025     52428568+   83   Linux

Disk /dev/vdc: 2147 MB, 2147483648 bytes
16 heads, 63 sectors/track, 4161 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/vdc doesn't contain a valid partition table
[root@VM_124_230_centos ~]#
```

Command (m for help): p

```
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 label type: dos
Disk identifier: 0x53d03ce6
```

| Device    | Boot | Start    | End      | Blocks  | Id | System   |
|-----------|------|----------|----------|---------|----|----------|
| /dev/vdc1 |      | 2048     | 12584959 | 6291456 | 83 | Linux    |
| /dev/vdc2 |      | 12584960 | 20971519 | 4193280 | 5  | Extended |
| /dev/vdc5 |      | 12587008 | 20971519 | 4192256 | 83 | Linux    |

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

[root@VM\_28\_66\_centos ~]#

2. 分区后执行 `ls -l /dev/disk/by-id` 命令，可以看到以下内容：

```
[root@VM_28_66_centos ~]#  
[root@VM_28_66_centos ~]# ls -l /dev/disk/by-id/  
total 0  
lrwxrwxrwx 1 root root 9 Jul 21 16:58 virtio-disk-bm42ztpm -> ../../vdc  
lrwxrwxrwx 1 root root 10 Jul 21 16:58 virtio-disk-bm42ztpm-part1 -> ../../vdc1  
lrwxrwxrwx 1 root root 10 Jul 21 16:58 virtio-disk-bm42ztpm-part2 -> ../../vdc2  
lrwxrwxrwx 1 root root 10 Jul 21 16:59 virtio-disk-bm42ztpm-part5 -> ../../vdc5  
[root@VM_28_66_centos ~]#
```

运行 `mkfs.ext4 device_name` 命令格式化并创建 ext4 文件系统。用设备名称（例如，`/dev/vdb`）替换 `device_name`。根据应用程序的要求或操作系统的限制，您可以选择其他文件系统类型，如 ext3 或 XFS。

```
[root@VM_28_66_centos ~]#  
[root@VM_28_66_centos ~]# mkfs.ext3 /dev/vdc1  
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  
393216 inodes, 1572864 blocks  
78643 blocks (5.00%) reserved for the super user  
First data block=0  
Maximum filesystem blocks=1610612736  
48 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  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (32768 blocks): done  
Writing superblocks and filesystem accounting information: done  
  
[root@VM_28_66_centos ~]# mkfs.ext3 /dev/vdc5  
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  
262144 inodes, 1048064 blocks  
52403 blocks (5.00%) reserved for the super user  
First data block=0  
Maximum filesystem blocks=1073741824  
32 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  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (16384 blocks): done  
Writing superblocks and filesystem accounting information: done
```

# 挂载硬盘

最近更新时间: 2023-03-13 15:43:22

```
mkdir /data/part1 -p # 创建示例挂载点
mkdir /data/part5 -p # 创建示例挂载点
mount /dev/vdc1 /data/part1 # 将vdc1挂载到/data/part1处
mount /dev/vdc5 /data/part5 # 将vdc5挂载到/data/part5处
touch /data/part1/disk-bm42ztpm-part1.txt # 创建一个空文件用于后续演示
touch /data/part5/disk-bm42ztpm-part5.txt # 创建一个空文件用于后续演示
yum install tree -y # 安装一个用于展示目录结构的工具
tree /data # 查看/data目录结构
此时可以看到如下的结构树:
```

```
[root@VM_28_66_centos ~]#
[root@VM_28_66_centos ~]# tree /data
/data
|-- part1
|   |-- disk-bm42ztpm-part1.txt
|   |-- lost+found
|-- part5
|   |-- disk-bm42ztpm-part5.txt
|   |-- lost+found

4 directories, 2 files
[root@VM_28_66_centos ~]#
```

执行 `lsblk -f` 命令查询文件系统UUID和挂载点信息:

```
[root@VM_28_66_centos ~]#
[root@VM_28_66_centos ~]# lsblk -f
NAME        FSTYPE LABEL UUID                                MOUNTPOINT
vda
`-vda1 ext3          49f819fd-e56d-48a4-86d3-7ebe0a68ec88 /
vdb ext3          e95f9d9c-d968-4da4-822a-f8e2ef140e4e
vdc
|-vdc1 ext3          970da23c-52bb-4208-b989-4b717a89c7d3 /data/part1
|-vdc2
`-vdc5 ext3          2a5294d9-ba68-4025-a288-f13ca8003b0a /data/part5
[root@VM_28_66_centos ~]#
```

在启动新的云服务器实例时，如果用户指定 自定义镜像 及数据盘快照，建行云云硬盘可以支持启动云服务器实例后自动挂载（即不需要进行一系列的添加、分区、格式化等操作可直接读写数据盘）。用户需要在制作自定义镜像和数据盘快照前在原实例上进行一些操作，下文将详细描述。

在 Linux 系统下如果用户希望指定数据盘快照生产出来的云硬盘能够自动挂载至新的云服务器实例，指定的自定义镜像和数据盘快照必须满足以下要求：

- 数据盘在制作快照前必须 已经格式化过，也即在原云服务器上已经 mount 成功。
- 系统盘在制作自定义镜像前，需要在 `/etc/rc.local` 文件中添加以下命令，将数据盘挂载点写入文件中：



```
mkdir -p <mount-point>
```

```
mount <device-id> <mount-point>
```

其中：<mount-point>请填入文件系统的挂载点如 /mydata，<device-id>请填入用户的实际文件分区位置，如 /dev/vdb(无分区有文件系统的设备名) 和 /dev/vdb1(有分区有文件系统)。

只有同时满足以上两个条件才能保证新启动的 Linux 云服务器实例数据盘可以被自动识别和挂载。



# Windows系统下初始化云硬盘

## Windows系统下初始化云硬盘

最近更新时间: 2023-03-13 15:43:22

实例可以识别连接的云硬盘并将其视为普通硬盘。用户可以使用任何文件系统将云块存储设备格式化、分区及创建文件系统。此后任何写入文件系统的数据均写入云硬盘中，并且对使用该设备的应用程序是透明的。数据盘默认为脱机状态，未做分区和格式化时无法使用。本教程将引导您进行 Windows 系统分区以及格式化。

虽然不同的Windows版本在进入“磁盘管理”界面的路径不同，但进入磁盘管理界面后对于磁盘分区格式化的操作基本一致。

本文将从 Windows 2012系统来引导用户进行数据盘挂载、分区和格式化。



# 前提条件

最近更新时间: 2023-03-13 15:43:22

- 请确保您已将云硬盘连接到CVM实例并登录Windows实例。
- 格式化后，数据盘中的数据将被全部清空。请在格式化之前，确保数据盘中没有数据或对重要数据已进行备份。为避免服务发生异常，格式化前请确保云服务器已停止对外服务。
- 当用户申请了多块云硬盘时，建议您对存放重要数据的弹性云盘设置自定义名称。
- 用户可以在云硬盘控制台中根据自定义名称或者关联的云主机内网IP快速查找云硬盘。

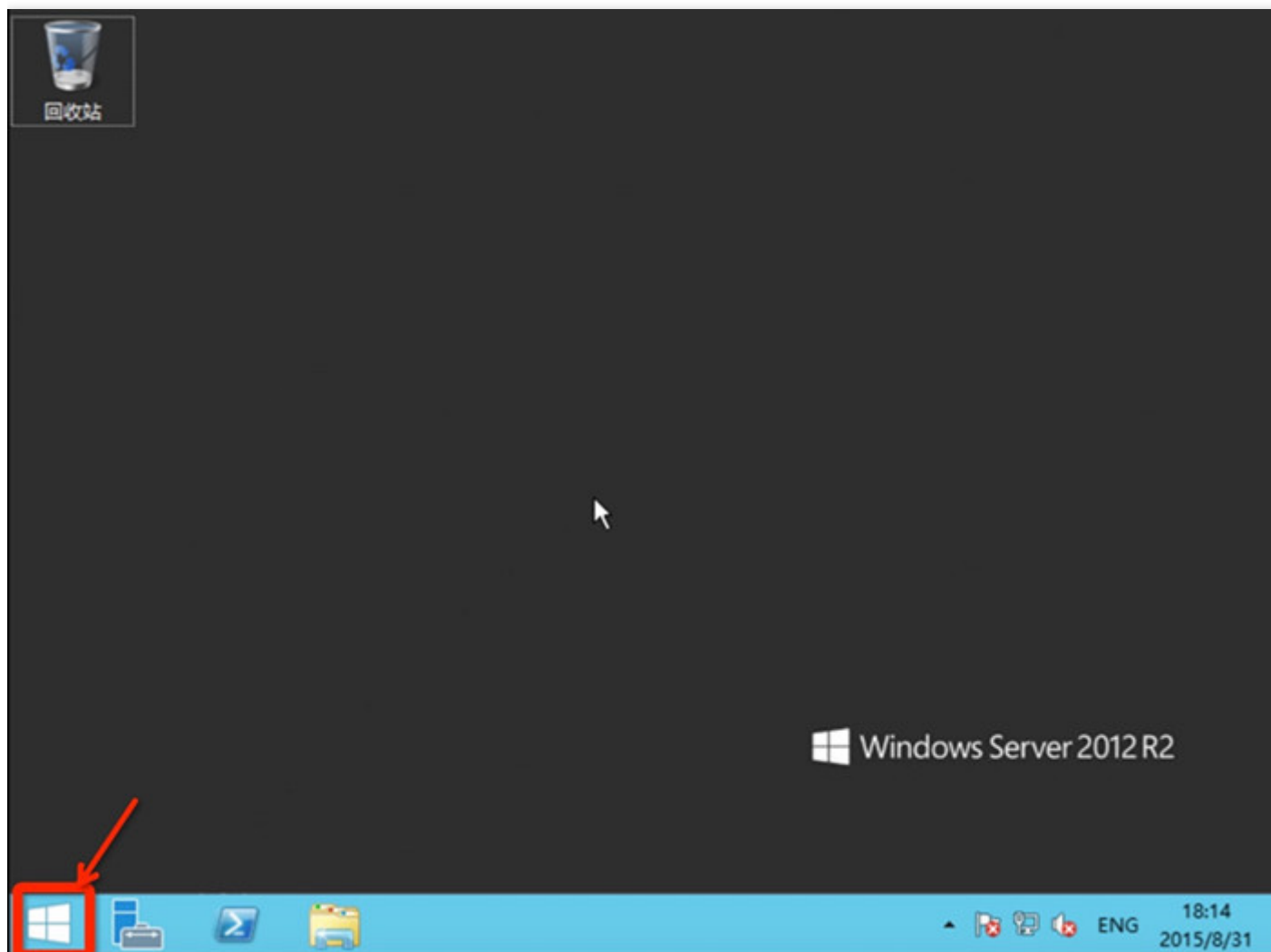
# Windows 2012 磁盘联机、分区和格式化

## 联机磁盘

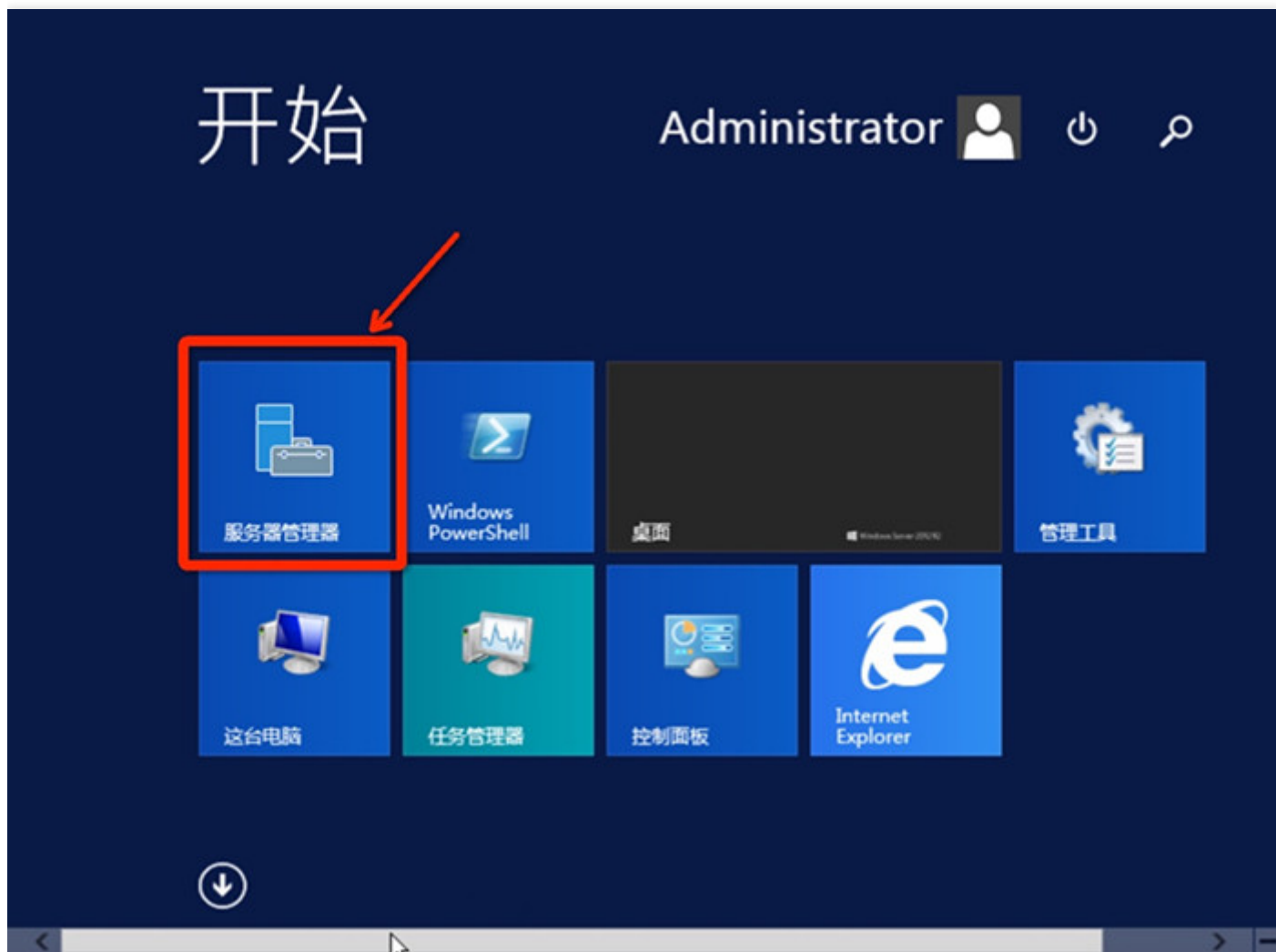
最近更新时间: 2023-03-13 15:43:22

Windows 2012 进入磁盘管理的路径为：【开始】－【服务器管理】－【工具】－【计算机管理】－【磁盘管理】。

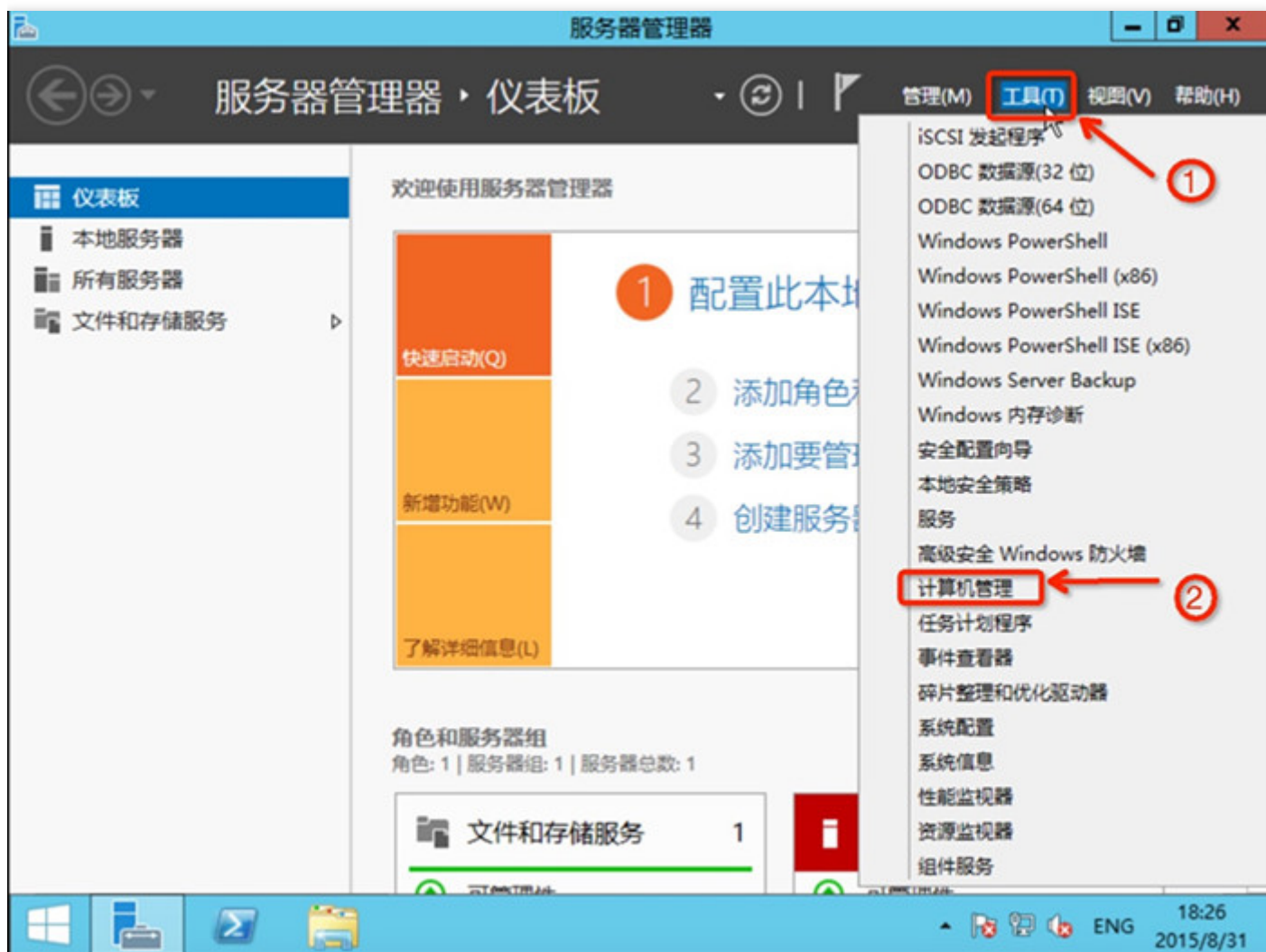
点击【开始】按钮：



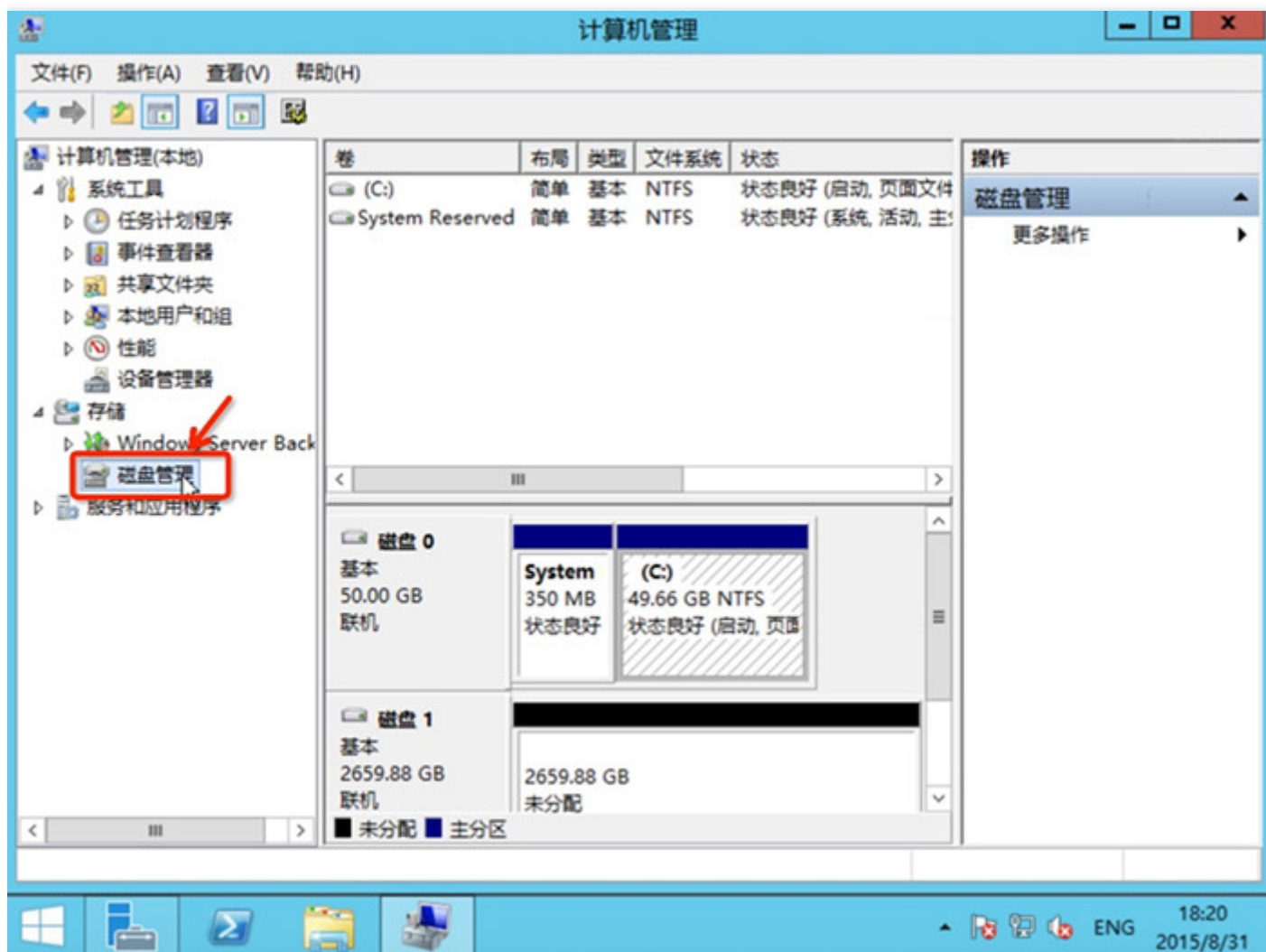
点击【服务器管理】：



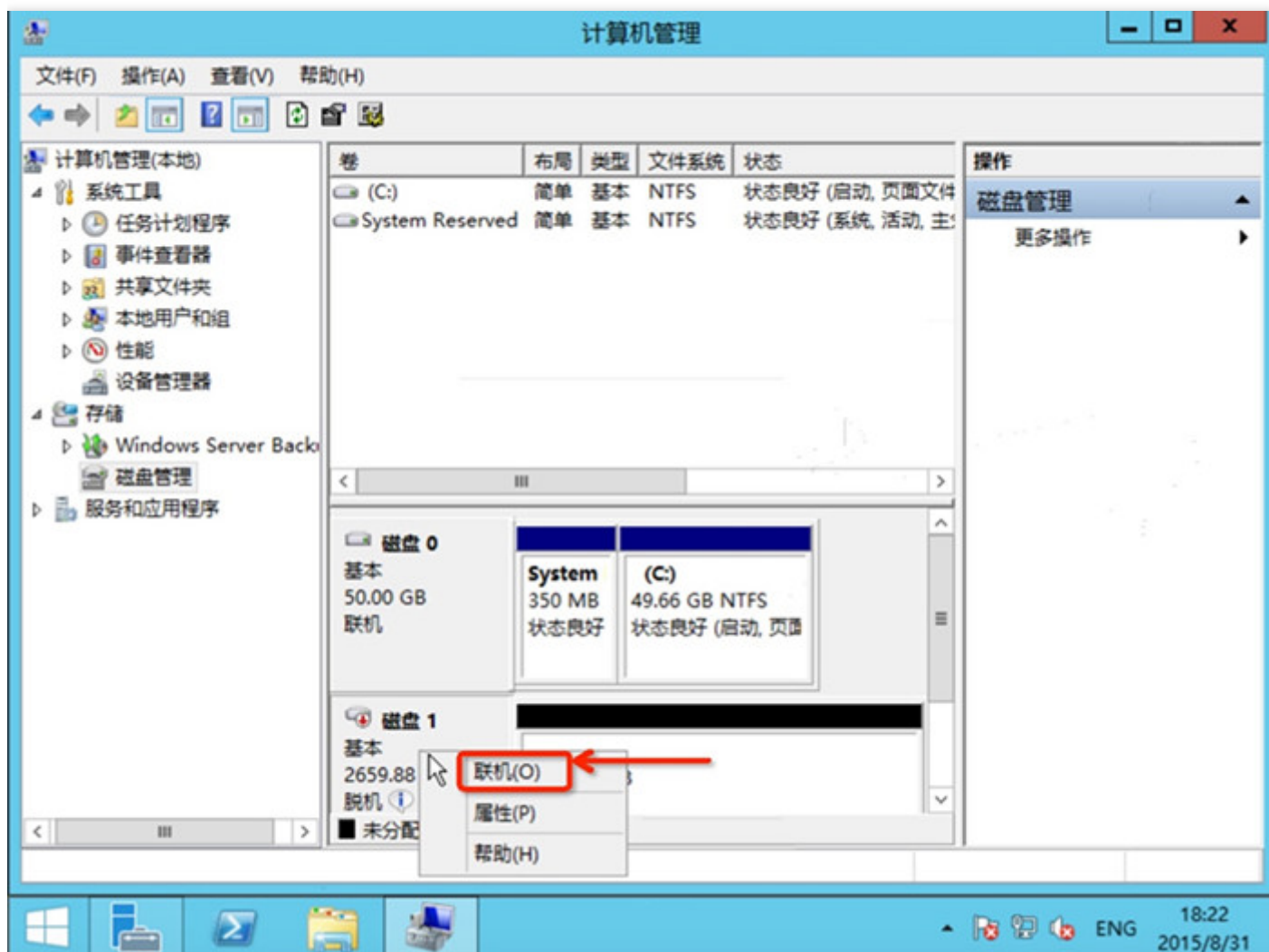
点击【工具】－【计算机管理】：



点击【磁盘管理】：



如下图所示：在磁盘1上右键点击，选择【联机】：

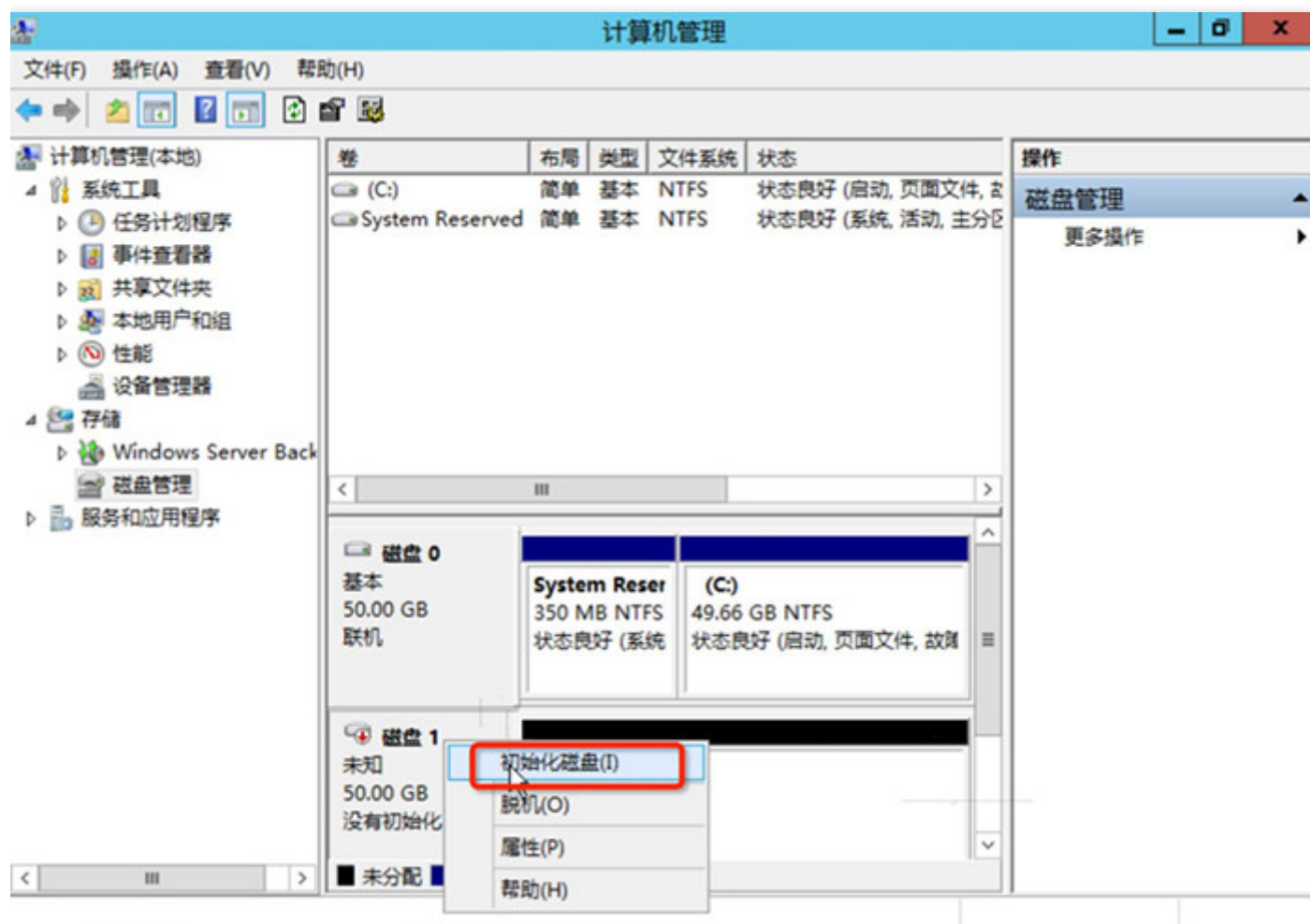


若磁盘上已经有数据（即不为空白磁盘情况下），用户可忽略下面的操作。对一个非空磁盘进行重新格式化或分区将清除所有原有数据。

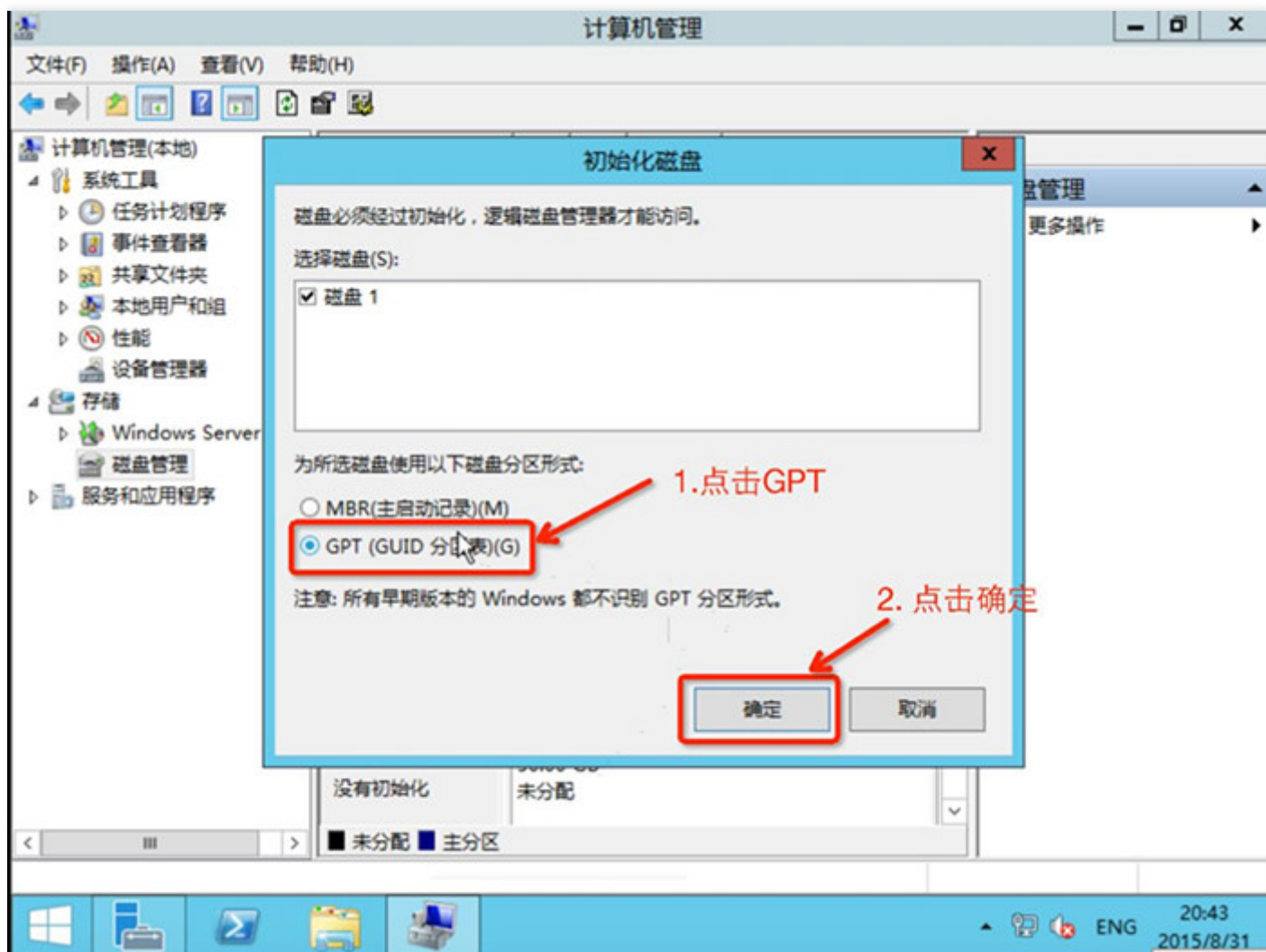
# (可选) 格式化磁盘

最近更新时间: 2019-11-30 15:11:34

右键点击，选择【初始化磁盘】：



根据分区方式的不同，选择【GPT】或【MBR】，点击【确定】按钮：



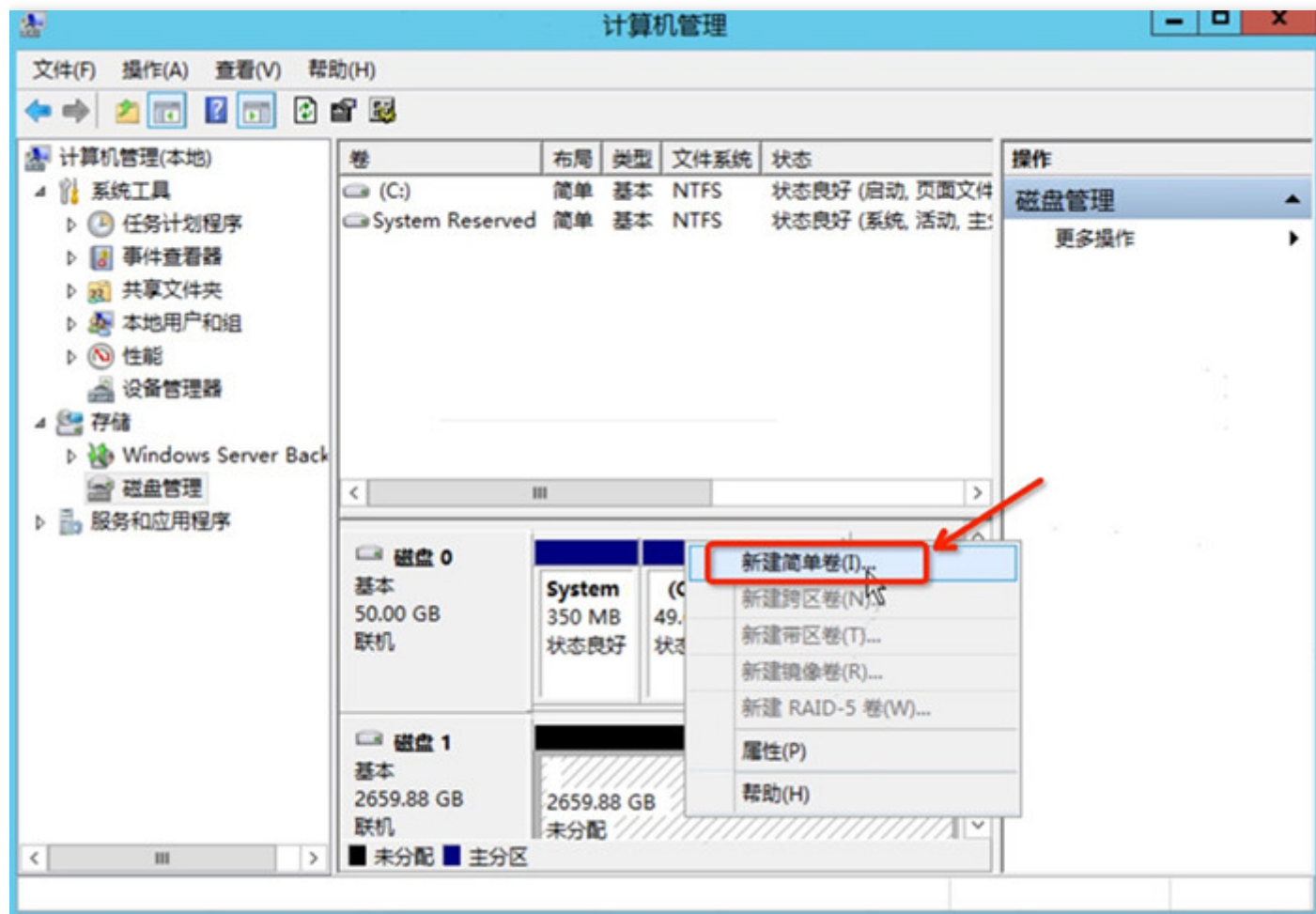
注意：磁盘大于 2TB 时仅支持 GPT 分区形式。若您不确定磁盘后续扩容是否会超过该值，则建议您选择 GPT 分区；若您确定磁盘大小不会超过该值，则建议您选择 MBR 分区以获得更好的兼容性。



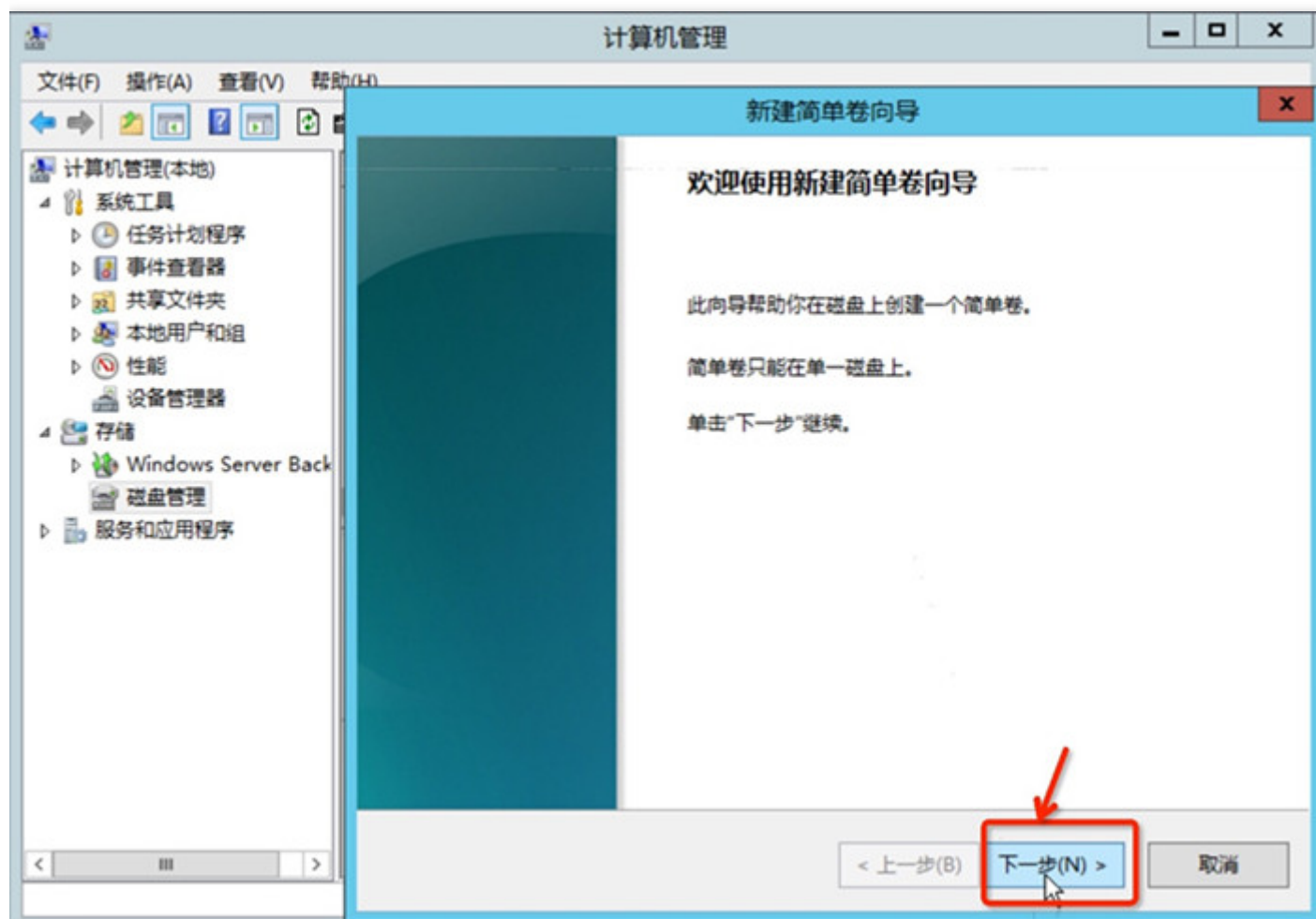
## (可选) 对磁盘分区

最近更新时间: 2019-11-30 15:11:34

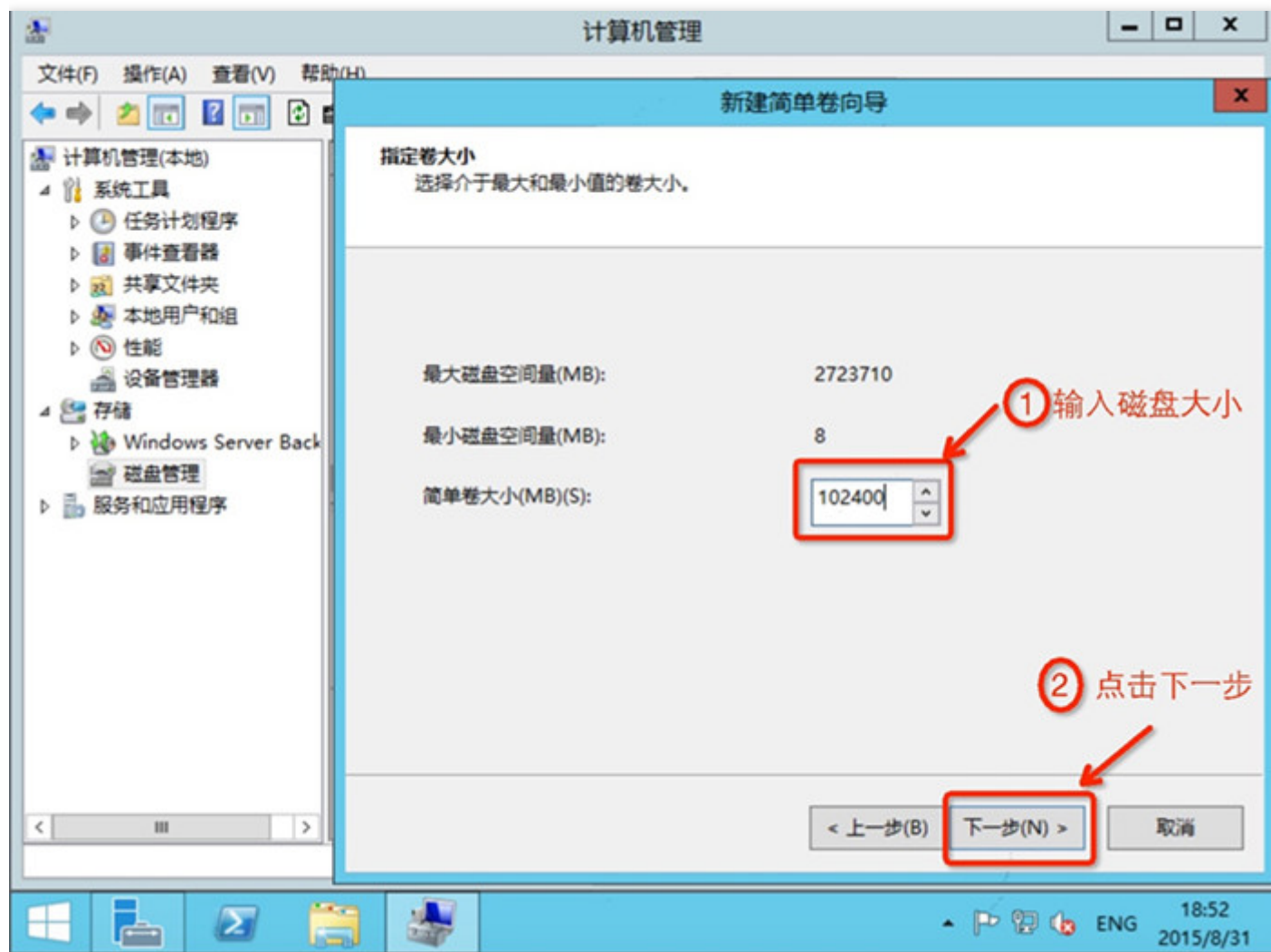
在未分配的空间处右击，选择【新建简单卷】：



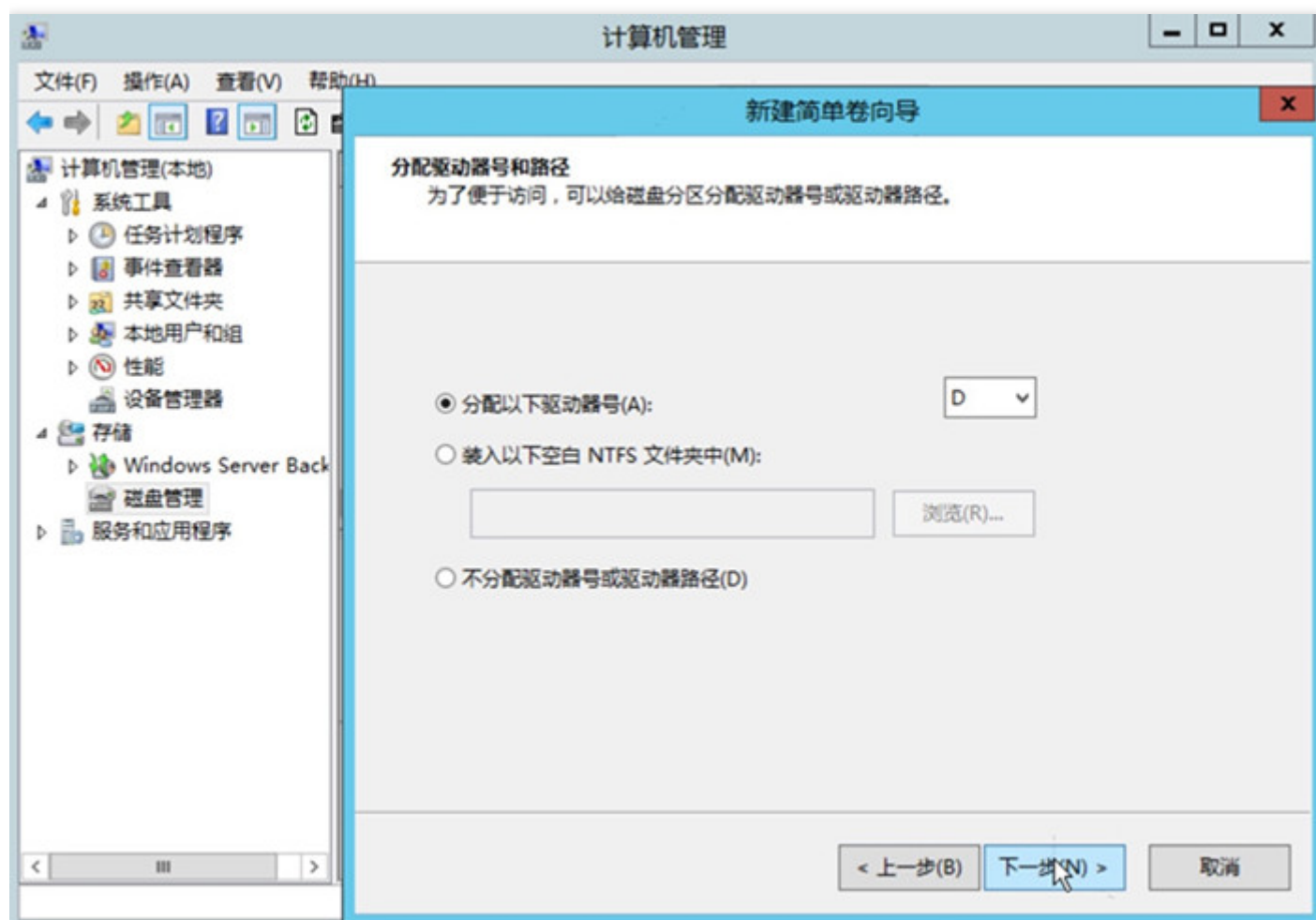
在弹出的“新建简单卷向导”窗口中，点击【下一步】：



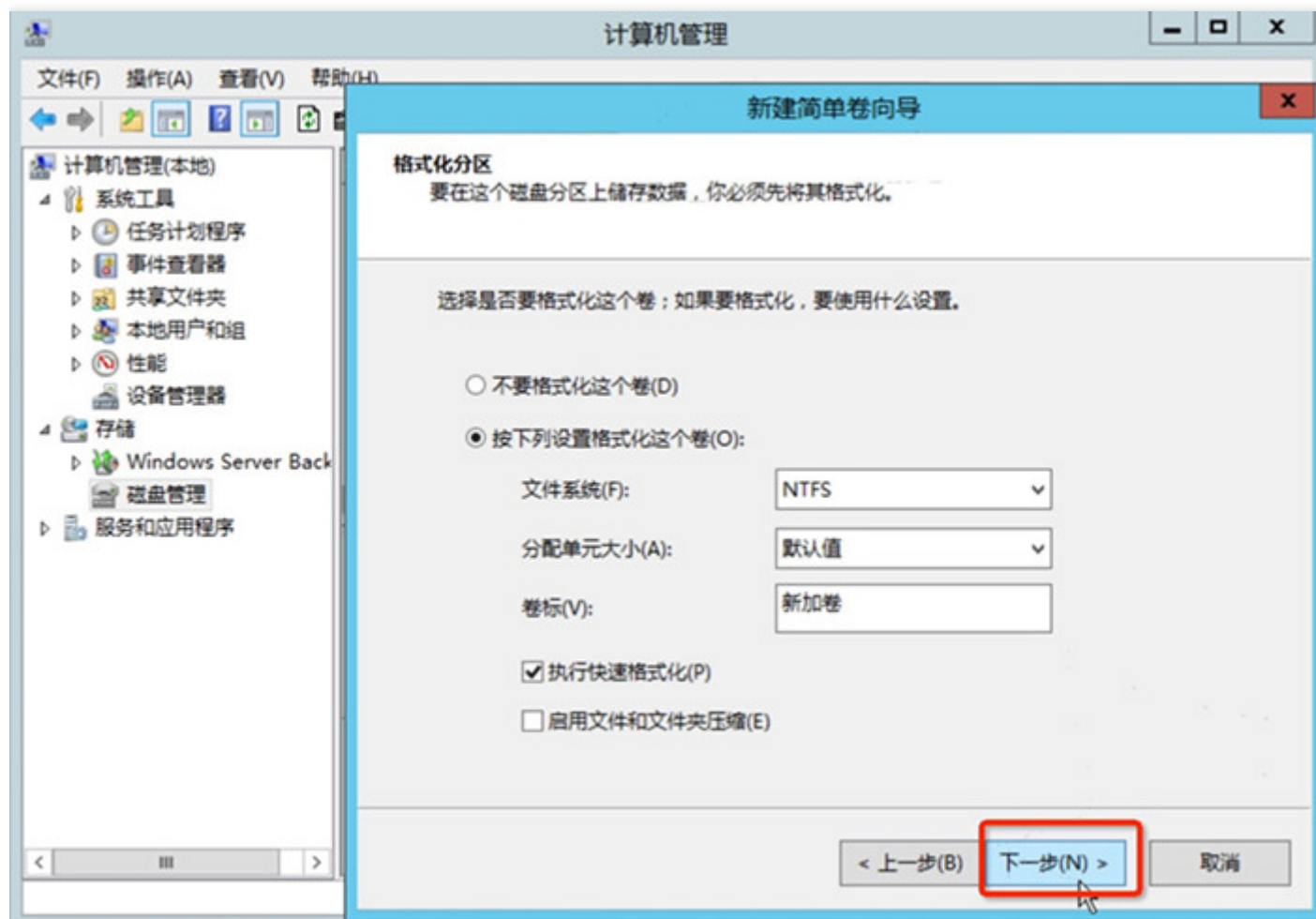
输入分区所需磁盘大小，点击【下一步】：



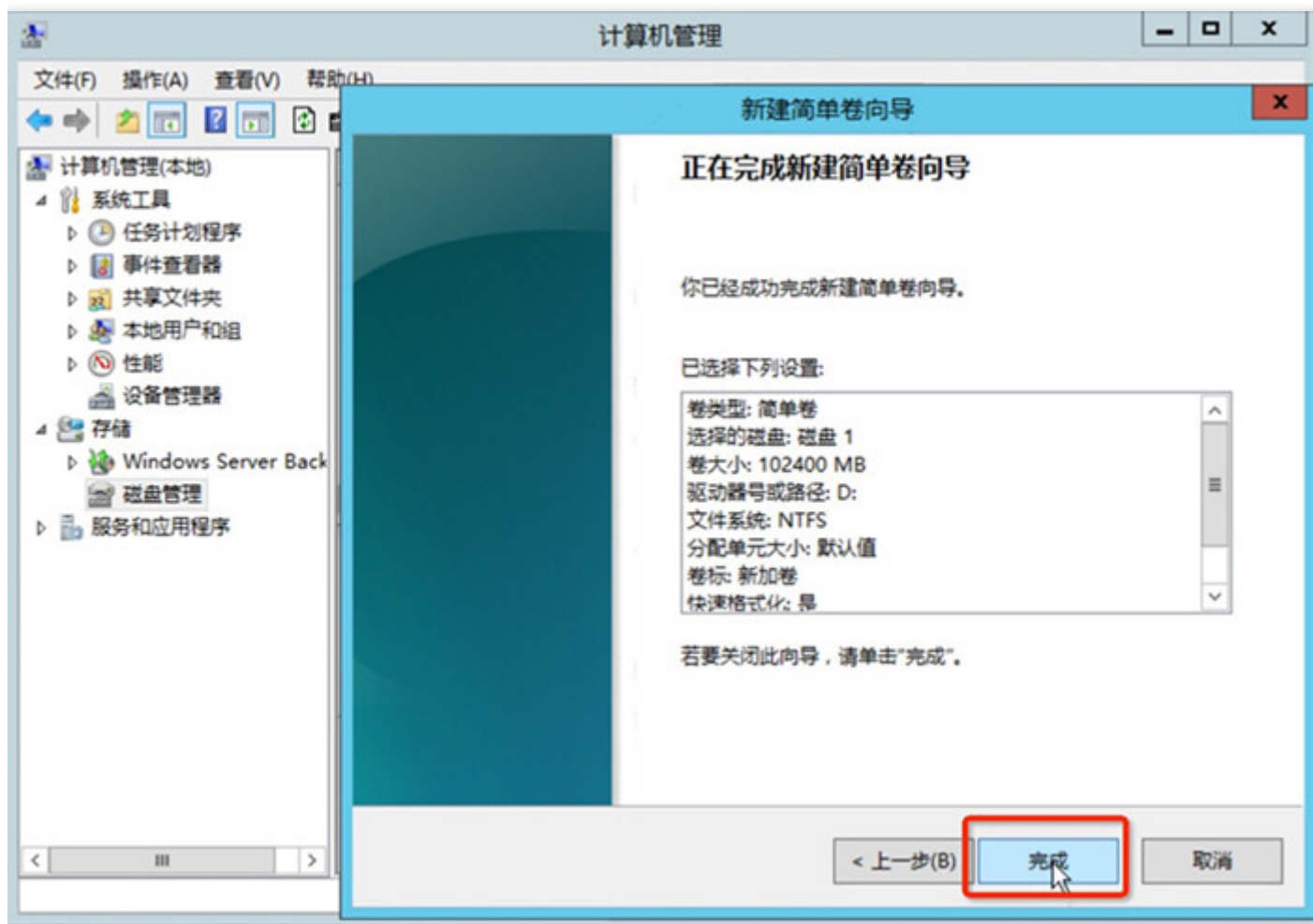
输入驱动器号，点击【下一步】：



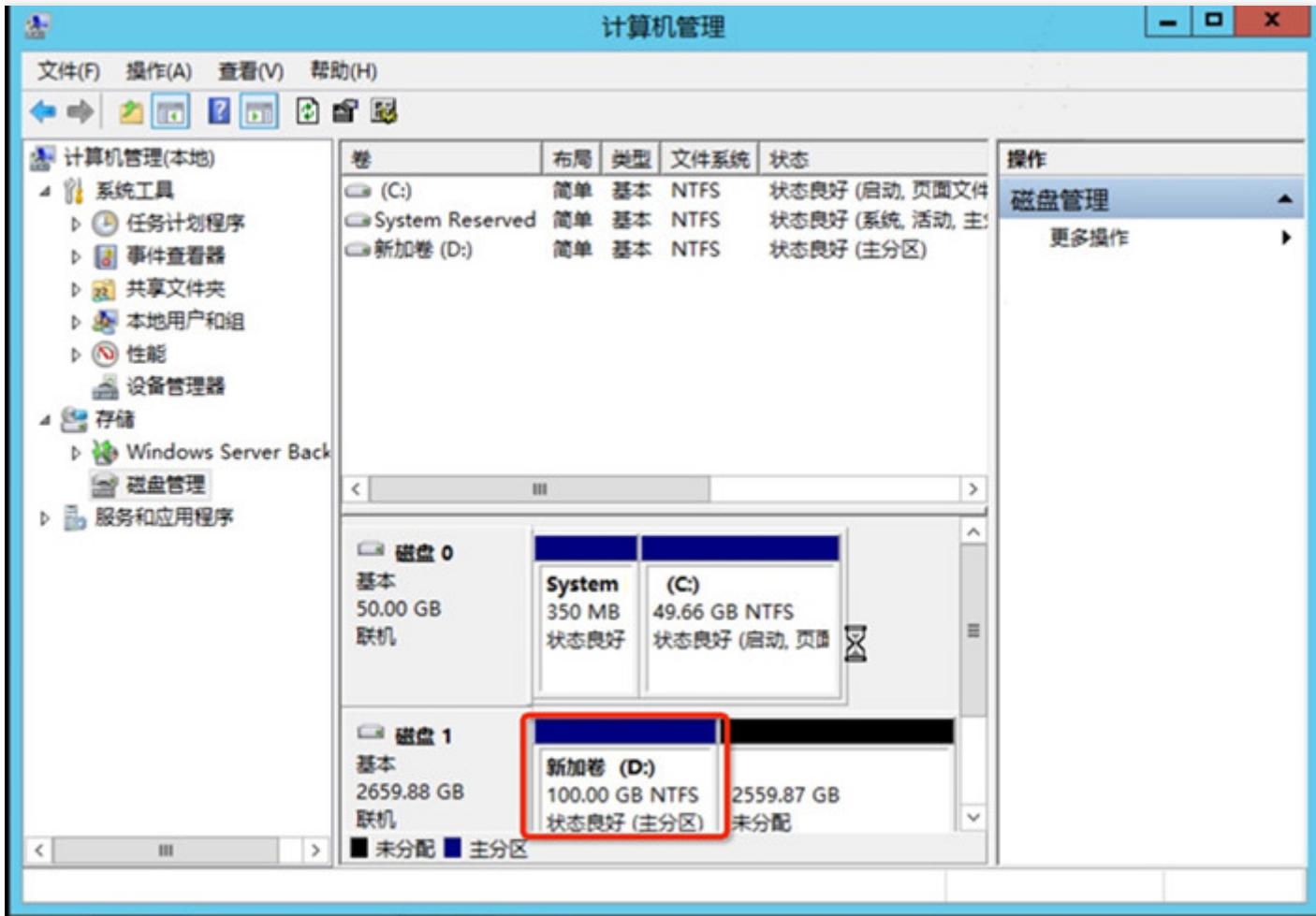
选择文件系统，格式化分区，点击【下一步】：



完成新建简单卷，点击【完成】：



查看新分区：



# 使用自定义镜像及数据盘快照启动新实例时自动挂载数据盘

最近更新时间: 2023-03-13 15:43:22

在启动新的云服务器实例时，如果用户指定自定义镜像及数据盘快照，建行云云硬盘可以支持启动云服务器实例后自动挂载（即不需要进行一系列的添加、分区、格式化等操作可直接读写数据盘）。用户需要在制作自定义镜像和数据盘快照前在原实例上进行一些操作，下文将详细描述。

在 Windows 系统下如果用户希望指定数据盘快照生产出来的云硬盘能够自动挂载至新的云服务器实例，指定的自定义镜像和数据盘快照必须满足以下要求：

- 自定义镜像中的 SAN 策略为：onlineAll。建行云目前提供的 Windows 公有镜像已默认进行相关设置，但仍建议用户在制作自定义镜像前检查下此配置，检查方法如下：

```
PS C:\Users\Administrator>
PS C:\Users\Administrator> diskpart

Microsoft DiskPart 版本 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
在计算机上: [REDACTED]

DISKPART> san

SAN 策略: 使共享磁盘脱机

DISKPART> san policy=onlineall

DiskPart 已成功更改用于当前操作系统的 SAN 策略。

DISKPART> san

SAN 策略: 全部联机
```

- 数据盘在制作快照前必须已经被格式化为 ntfs 或 fat32 格式。

只有同时满足以上两个条件才能保证新启动的 Windows 云服务器实例数据盘可以被自动识别和联机。



# 扩容云硬盘

## 扩容云硬盘

最近更新时间: 2023-03-13 15:53:45

云硬盘是云上可扩展的存储设备，用户可以在创建云硬盘后随时扩展其大小，以增加存储空间，同时不失去云硬盘上原有的数据。

云硬盘扩容完成后，需扩展分区及文件系统。您可将扩容部分的容量划分至已有分区内，或者将扩容部分的容量格式化成为独立的新分区。

如果您的硬盘分区为 MBR 格式，扩容到超过 2TB 大小时 MBR 分区格式已不支持。建议您新建一块数据盘，使用 GPT 分区后通过复制的方式将数据拷贝至新盘上。



# 扩容类型为云硬盘的数据盘

## 通过云硬盘控制台扩容CBS数据盘

最近更新时间: 2023-03-13 15:53:45

1. 打开云服务器 CVM控制台。
2. 单击导航窗格中的【云硬盘】。
3. 只有显示为【未挂载】状态且【支持挂载/卸载】的硬盘可以扩容，点击末尾【更多】-【扩容】按钮，选择需要的新大小（必须大于或等于当前大小）即完成了实体硬盘的扩容操作。  
对于已经连接到了实例的弹性云盘请先执行 卸载云硬盘 操作



# 通过云服务器控制台扩容CBS数据盘

最近更新时间: 2023-03-13 15:53:45

1. 打开云服务器 CVM控制台。
2. 单击导航窗格中的【云主机】。
3. 只有显示为【关机】状态且系统盘、数据盘均为云硬盘的实例可以扩容，点击末尾【更多】-【云主机设置】-【调整硬盘】按钮，选择需要的新大小（必须大于或等于当前大小）即完成了实体硬盘的扩容操作。  
对于正在运行的系统盘、数据盘均为云硬盘的实例，需要扩容请先执行实例关机操作。



# 扩容类型为云硬盘的系统盘

最近更新时间: 2023-03-13 15:53:45

系统盘类型为云硬盘时，支持对系统盘的扩容，但仅可以通过对云服务器进行重装系统操作

# 扩容文件系统

## 扩容Linux文件系统

## 扩容Linux文件系统

最近更新时间: 2023-03-13 15:53:45

云硬盘是云上可扩展的存储设备，用户可以在创建云硬盘后随时扩展其大小，以增加存储空间，同时不失去云硬盘上原有的数据。要达到扩容并使用扩容空间的目的，用户在扩容实体云硬盘大小之后，还需要扩展其上的文件系统以识别新增可用的空间。您可以根据下面的步骤进行：

### 1) 扩容实体云硬盘大小。

在执行了扩容实体云硬盘大小操作后，用户可以通过检查文件系统大小来查看实例是否识别了更大的云硬盘更空间。在 Linux 上，可以使用 `df -h` 命令检查文件系统大小。如果没有看到云硬盘大小变成扩容后的值，则需要扩容此文件系统，以便实例可以使用新的空间。

### 2) 扩容分区。

使用以下命令确认云硬盘在扩容前使用的分区表形式：

```
fdisk -l
```

若结果如下两图所示（根据操作系统不同略有不同），则说明云服务器扩容前为GPT分区方式，后续操作请参见GPT分区云硬盘扩容后修改分区指引。

```
Disk /dev/vdb: 32.2 GB, 32212254720 bytes, 7864320 sectors
Units = sectors of 1 * 4096 = 4096 bytes
Sector size (logical/physical): 4096 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk label type: dos
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks    Id  System
/dev/vdb1           1      2621439     10485756    ee    GPT
```

WARNING: GPT (GUID Partition Table) detected on '/dev/vdb'! The util fdisk doesn't support GPT. Use GNU Parted.

若结果如下图所示（根据操作系统不同略有不同），则说明云服务器扩容前为MBR分区方式，后续操作请参见MBR分区云硬盘扩容后修改分区指引。



```
[root@VM_20_49_tlinux ~]# fdisk -l
```

Disk /dev/xvda: 8589 MB, 8589934592 bytes  
255 heads, 63 sectors/track, 1044 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: 0xf5a25329

| Device     | Boot | Start | End  | Blocks   | Id | System |
|------------|------|-------|------|----------|----|--------|
| /dev/xvda1 | *    | 1     | 1044 | 8385898+ | 83 | Linux  |

Disk /dev/xvdc: 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: 0x06a4a875

| Device     | Boot | Start | End  | Blocks    | Id | System |
|------------|------|-------|------|-----------|----|--------|
| /dev/xvdc1 |      | 1     | 6526 | 52420063+ | 83 | Linux  |

3) 扩容文件系统。



# GPT分区云硬盘扩容后修改分区指引

## 新空间格式化成一个独立GPT分区

## 查看数据盘信息

最近更新时间: 2019-11-30 15:11:28

执行命令parted 磁盘路径 print命令来确认云硬盘的容量变化。如在过程中收到如下提示,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)?
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 188743680 blocks) or continue with
the current setting?
Fix/Ignore? Fix
```

```
[root@VM_164_30_centos ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name      Flags
  1      1049kB  10.7GB  10.7GB  ext3         primary
```

这里扩容后的云硬盘大小为107GB, 已有分区的大小为10.7GB。



# 卸载已挂载数据盘

最近更新时间: 2019-11-30 15:11:28

执行以下命令确认该云硬盘是否还有分区已挂载：

`mount | grep '磁盘路径'`

```
[root@VM_164_30_centos ~]# mount | grep '/dev/vdb'
/dev/vdb1 on /data type ext3 (rw)
```

这里云硬盘上有一个分区(vdb1)挂载在/data上，需要将其解挂。

使用以下命令解挂：

`umount 挂载点`

本例中即执行`umount /data`进行卸载。

注：要将云硬盘上所有分区的文件系统都解挂，如vdb1、vdb2..... 再次使用`mount | grep '/dev/vdb'` 命令来确认此硬盘上所有分区的文件系统都已解挂。

```
[root@VM_164_30_centos ~]# mount | grep '/dev/vdb'
[root@VM_164_30_centos ~]#
```



# 数据盘分区

最近更新时间: 2019-11-30 15:11:28

确认云硬盘所有分区均已卸载后，执行以下命令新建一个分区： parted 磁盘路径 这里输入parted /dev/vdb。 接下来输入print来查看分区信息，记住已有分区的End值，以此值作为下一个分区的起始偏移值：

```
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

| Number | Start  | End    | Size   | File system | Name    | Flags |
|--------|--------|--------|--------|-------------|---------|-------|
| 1      | 1049kB | 10.7GB | 10.7GB | ext3        | primary |       |

接下来执行以下命令新建一个主分区，此分区将从已有分区的末尾开始，覆盖硬盘所有的新增空间。mkpart primary start end 本例使用mkpart primary 10.7GB 100% 再次执行print可发现新分区已经新建成功，键入quit即可退出parted工具：

```
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

| Number | Start  | End    | Size   | File system | Name    | Flags |
|--------|--------|--------|--------|-------------|---------|-------|
| 1      | 1049kB | 10.7GB | 10.7GB | ext3        | primary |       |
| 2      | 10.7GB | 107GB  | 96.6GB |             | primary |       |



# 格式化新建分区

最近更新时间: 2019-11-22 23:17:53

执行以下命令格式化上述新建的分区，用户可以自行决定文件系统的格式，如ext2、ext3等。mkfs.[fstype] [分区路径]

这里使用命令mkfs.ext3 /dev/vdb2对新分区进行格式化，文件系统为EXT3。



# 新空间增加到已有分区中(GPT分区格式)

## 查看数据盘信息

最近更新时间: 2019-11-30 15:11:28

执行命令parted 磁盘路径 print命令来确认云硬盘的容量变化。如在过程中收到如下提示,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)?
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 188743680 blocks) or continue with
the current setting?
Fix/Ignore? Fix
```

```
[root@VM_164_30_centos ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End     Size    File system  Name      Flags
  1      1049kB  10.7GB  10.7GB  ext3         primary
```

这里扩容后的云硬盘大小为107GB, 已有分区的大小为10.7GB。



# 卸载已挂载数据盘

最近更新时间: 2019-11-30 15:11:23

执行以下命令确认该云硬盘是否还有分区已挂载: `mount | grep '磁盘路径'`

```
[root@VM_164_30_centos ~]# mount | grep '/dev/vdb'
/dev/vdb1 on /data type ext3 (rw)
```

这里云硬盘上有一个分区(vdb1)挂载在/data上, 需要将其解挂。使用以下命令解挂: `umount 挂载点` 本例中即执行 `umount /data` 进行卸载。注: 要将云硬盘上所有分区的文件系统都解挂, 如vdb1、vdb2..... 再次使用 `mount | grep '/dev/vdb'` 命令来确认此硬盘上所有分区的文件系统都已解挂。

```
[root@VM_164_30_centos ~]# mount | grep '/dev/vdb'
[root@VM_164_30_centos ~]#
```

# 数据盘分区

最近更新时间: 2023-03-13 16:34:42

确认云硬盘所有分区均已卸载后，执行以下命令，将原分区删除并以同样的起始偏移新建一个分区：parted [磁盘路径] 接下来输入units，将显示和操纵单位变成sector（默认为GB），输入print来查看分区信息，记住已有分区的Start值。删除分区并新建后，Start值必须与这个相同，否则数据将会丢失。

```
(parted) unit s
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

| Number | Start | End       | Size      | File system | Name    | Flags |
|--------|-------|-----------|-----------|-------------|---------|-------|
| 1      | 2048s | 20969471s | 20967424s | ext3        | primary |       |

执行以下命令删除原有分区：rm [分区Number] 由上图可知云硬盘上有一个分区，Number号为“1”，执行rm 1结果如下图：

```
(parted) rm 1
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

| Number | Start | End | Size | File system | Name | Flags |
|--------|-------|-----|------|-------------|------|-------|
|--------|-------|-----|------|-------------|------|-------|

输入mkpart primary [原分区起始扇区] 100%新建一个主分区。本例中使用mkpart primary 2048s 100%，此主分区从第2048个扇区开始（必须与删除之前的分区一致），100%表示此分区到磁盘的最末尾。如果出现如图状态请输入Ignore：

```
(parted) mkpart primary 2048s 100%
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? Ignore
```



再次输入print可发现新分区已经新建成功，输入quit即可退出parted工具：

```
(parted) mkpart primary 2048s 100%
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

| Number | Start | End        | Size       | File system | Name    | Flags |
|--------|-------|------------|------------|-------------|---------|-------|
| 1      | 2048s | 209713151s | 209711104s | ext3        | primary |       |



# 检查扩容后分区的文件系统

最近更新时间: 2019-11-30 15:11:23

使用以下命令检查扩容后的分区： `e2fsck -f` 分区路径 前述步骤中本例已新建了分区1，使用`e2fsck -f /dev/vdb1` 进行操作。结果如下：

```
[root@VM_164_30_centos ~]# e2fsck -f /dev/vdb1
e2fsck 1.41.12 (17-May-2010)
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/vdb1: 11/655360 files (0.0% non-contiguous), 79696/2620928 blocks
```



# 扩容文件系统

最近更新时间: 2019-11-04 02:13:11

执行以下命令进行分区上文件系统的扩容操作：`resize2fs 分区路径`



# 挂载新分区

最近更新时间: 2019-11-30 15:11:23

执行以下命令挂载分区：

mount 分区路径 挂载点

这里通过mount /dev/vdb1 /data命令手动挂载新分区，并使用df -h命令查看，出现以下信息说明挂载成功，即可以查看到数据盘了。

```
[root@VM_164_30_centos ~]# mount /dev/vdb1 /data
[root@VM_164_30_centos ~]# df -hl
```

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/vda1  | 7.9G | 902M | 6.6G  | 12%  | /          |
| tmpfs      | 1.9G | 0    | 1.9G  | 0%   | /dev/shm   |
| /dev/vdb1  | 99G  | 163M | 94G   | 1%   | /data      |



# MBR分区云硬盘扩容后修改分区指引

## MBR分区云硬盘扩容后修改分区指引

最近更新时间: 2023-03-13 16:34:42

MBR分区的云硬盘进行扩容后，您可以选择： – 将新增的容量空间建立成独立的新分区同时原有分区保持不变 – 扩容原有分区至新增的容量空间（包括未分区直接格式化的场景），并且保持原有分区的数据不丢失。

以上两种场景，在您的Linux云服务器的云硬盘升级成功之后，都可以通过Linux下的分区扩容工具 (fdisk/e2fsck/resize2fs)，执行一系列命令，完成分区扩容，并且保证原数据不会丢失。需要注意的是，不管是添加新分区还是扩容到已有分区都需要先将此磁盘的所有已挂载分区umount再执行扩容操作，这样内核才能识别出新的分区表。

请注意，由于MBR的限制，选择任何一种方式时，请保持任意分区的大小不超过2TB（若您扩容后的空间已经大于2TB则不可选择第二种方式）。

# 新空间格式化成一个独立分区

## 查看数据盘信息

最近更新时间: 2019-11-30 15:11:23

执行命令df -h查看已挂载的数据盘分区信息，以及命令fdisk -l查看数据盘扩容后未分区的信息：

```
[root@VM_20_49_tlinux ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.9G  2.1G  5.5G  28% /
tmpfs           4.0G    0  4.0G   0% /dev/shm
/dev/xvdc1       50G  180M   47G   1% /data
[root@VM_20_49_tlinux ~]#
```

```
[root@VM_20_49_tlinux ~]# fdisk -l

Disk /dev/xvda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0xf5a25329

   Device Boot      Start         End      Blocks    Id  System
/dev/xvda1    *           1         1044      8385898+   83  Linux

Disk /dev/xvdc: 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: 0x06a4a875

   Device Boot      Start         End      Blocks    Id  System
/dev/xvdc1           1         6526     5242063+   83  Linux
```



# 卸载所有已挂载的分区

最近更新时间: 2019-11-22 23:17:45

执行以下命令卸载所有已挂载的分区：

`umount 挂载点`

这里使用 `umount /data` 卸载所有已挂载分区。

# 数据盘分区

最近更新时间: 2019-11-30 15:11:23

确认云硬盘所有分区均已卸载后, 执行以下命令新建一个新分区: `fdisk [硬盘路径]`

本例使用`fdisk /dev/xvdc`命令, 按照界面的提示依次输入“p”(查看现有分区信息)、“n”(新建分区)、“p”(新建主分区)、“2”(新建第2个主分区), 两次回车(使用默认配置), 输入“w”(保存分区表), 开始分区:

```
[root@VM_20_49_tlinux /]#  
[root@VM_20_49_tlinux /]# fdisk /dev/xvdc  
  
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to  
switch off the mode (command 'c') and change display units to  
sectors (command 'u').  
  
Command (m for help): p  
  
Disk /dev/xvdc: 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: 0x06a4a875  
  
   Device Boot      Start         End      Blocks   Id  System  
/dev/xvdc1         1         6526     52420063+  83  Linux  
  
Command (m for help): n  
Command action  
  e   extended  
  p   primary partition (1-4)  
p  
Partition number (1-4): 2  
First cylinder (6527-13054, default 6527):  
Using default value 6527  
Last cylinder, +cylinders or +size{K,M,G} (6527-13054, default 13054):  
Using default value 13054  
  
Command (m for help): w  
The partition table has been altered!  
  
Calling ioctl() to re-read partition table.  
Syncing disks.  
[root@VM_20_49_tlinux /]#
```

这里是以创建1个分区为例, 用户也可以根据自己的需求创建多个分区。



# 查看新分区

最近更新时间: 2019-11-30 15:11:23

使用以下命令查看新分区: `fdisk -l`

```
[root@VM_20_49_tlinux /]# fdisk -l

Disk /dev/xvda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 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: 0xf5a25329

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *           1         1044      8385898+   83   Linux

Disk /dev/xvdc: 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: 0x06a4a875

   Device Boot      Start         End      Blocks   Id  System
/dev/xvdc1           1         6526      52420063+   83   Linux
/dev/xvdc2        6527       13054      52436160    83   Linux
```

这里新的分区xvdc2已经创建完成。



# 格式化新分区并创建文件系统

最近更新时间: 2019-11-30 15:11:23

在进行分区格式化时，用户可以自行决定文件系统的格式，如ext2、ext3等。这里以“ext3”为例，使用命令mkfs.ext3 /dev/xvdc2对新分区进行格式化。

```
[root@VM_20_49_tlinux /]# mkfs.ext3 /dev/xvdc2
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3278576 inodes, 13109040 blocks
655452 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
401 block groups
32768 blocks per group, 32768 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

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

This filesystem will be automatically checked every 20 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@VM_20_49_tlinux /]#
```



# 挂载新分区

最近更新时间: 2019-11-30 15:11:23

使用以下命令创建新的挂载点： mkdir 新挂载点 并执行以下命令挂载新分区到新挂载点上： mount 新分区路径 新挂载点 这里使用命令mkdir /data1创建data1目录，再通过mount /dev/xvdc2 /data1命令手动挂载新分区后，用df -h命令查看，出现以下信息说明挂载成功，即可以查看到数据盘了：

```
[root@VM_20_49_tlinux /]# mkdir /data1
[root@VM_20_49_tlinux /]# mount /dev/xvdc2 /data1
[root@VM_20_49_tlinux /]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.9G  2.1G  5.5G  28% /
tmpfs           4.0G   0    4.0G   0% /dev/shm
/dev/xvdc1       50G   180M   47G   1% /data
/dev/xvdc2       50G   180M   47G   1% /data1
[root@VM_20_49_tlinux /]#
```



# 添加新分区信息

最近更新时间: 2019-11-30 15:11:23

如果希望云服务器在重启或开机时能自动挂载数据盘，必须将分区信息添加到/etc/fstab中。如果没有添加，则云服务器重启或开机后都不能自动挂载数据盘。执行以下命令添加信息： `echo '/dev/xvdc2 /data1 ext3 defaults 0 0' >> /etc/fstab` 执行 `cat /etc/fstab` 命令查看，出现以下信息表示添加分区信息成功：

```
[root@VM_20_49_tlinux /]# echo '/dev/xvdc2 /data1 ext3 defaults 0 0' >> /etc/fstab
[root@VM_20_49_tlinux /]# cat /etc/fstab
/dev/xvda1          /                    ext3              noatime,acl,user_xattr 1 1
/dev/xvdb           swap                swap              defaults 0 0
/dev/xvdc1          /data               ext3              noatime,acl,user_xattr 1 2
tmpfs               /dev/shm            tmpfs             defaults          0 0
proc                /proc              proc              defaults          0 0
sysfs               /sys               sysfs             noauto           0 0
debugfs             /sys/kernel/debug  debugfs           noauto           0 0
devpts              /dev/pts            devpts            mode=0620,gid=5  0 0
/dev/xvdc2          /data1              ext3              defaults          0 0
[root@VM_20_49_tlinux /]#
```



# 将新空间增加到已有分区空间中

# 将新空间增加到已有分区空间中

最近更新时间: 2019-11-04 02:20:05

若原有的硬盘分区为一个MBR分区(可以看到vdb1,vdc1等字样), 同时在此分区上制作了文件系统。或原有的硬盘没有分区, 直接在此硬盘上制作了文件系统。这两种情况都可以选择使用自动扩容工具进行扩容。自动扩容工具适用于Linux操作系统, 用于将扩容时新扩的云硬盘存储空间添加到已存在的文件系统中, 扩容能够成功必须满足下面3个条件: - 文件系统是ext2/ext3/ext4 - 当前文件系统不能有错误 - 扩容后的磁盘大小不超过2TB 下面介绍自动扩容工具的使用方法。



# 卸载正在使用的硬盘分区

最近更新时间: 2019-11-30 15:11:13

执行以下命令卸载分区：umount 挂载点

```
[root@VM_136_143 ~]# umount /data/
[root@VM_136_143 ~]# df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/vda1              8254240    6832388    1002560   88% /
[root@VM_136_143 ~]#
```



# 下载一键扩容工具

最近更新时间: 2023-03-13 16:34:42

执行以下命令下载工具:

wget -O /tmp/devresize.py <http://mirrors.yun.ccb.com/install/virts/devresize.py>

# 执行扩容工具

最近更新时间: 2023-03-13 16:34:42

执行以下命令进行扩容：

python /tmp/devresize.py 硬盘路径

请注意，这里硬盘路径是需要扩容的云硬盘，而不是分区名。若您的文件系统在vdb1上，则应执行python /tmp/devresize.py /dev/vdb

```
[root@VM_136_143 ~]# python /tmp/devresize.py /dev/vdb
It will resize (/dev/vdb1).
This operation may take from several minutes to several hours, continue? [Y/n]
Y
Backup MBR to /tmp/MBR_2015-05-22_16:56:18_bak
e2fsck 1.41.12 (17-May-2010)
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/vdb1: 591/656640 files (0.0% non-contiguous), 87719/2621440 blocks
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/vdb1 to 13107198 (4k) blocks.
The filesystem on /dev/vdb1 is now 13107198 blocks long.
```

若输出“The filesystem on /dev/vdb1 is now XXXXX blocks long.”则表示扩容成功。

若输出的是“[ERROR] – e2fsck failed!!”，请先用fsck对文件系统所在分区进行修复，可以执行以下命令进行自动修复：fsck -a 分区路径

请注意这里与前一个命令不同，需要填写的是文件系统所在分区。若您的文件系统在vdb1上，则应执行fsck -a /dev/vdb1。

修复成功后，再使用python /tmp/devresize.py 硬盘路径来使用扩容工具进行扩容。



# 重新挂载扩容后的分区

最近更新时间: 2019-11-30 15:11:13

执行以下命令挂载扩容后的分区： mount 分区路径 挂载点 并通过以下命令查看扩容后的分区容量： df -h

这里通过mount /dev/vdb1 /data命令手动挂载扩容后的分区(如果原先是没有分区的，执行mount /dev/vdb /data)，用df -h命令查看，出现以下信息说明挂载成功，即可以查看到数据盘了：

```
[root@VM_136_143 ~]# mount /dev/vdb1 /data/
[root@VM_136_143 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       7.9G  6.6G  930M  88% /
/dev/vdb1       50G   2.2G   45G   5% /data
[root@VM_136_143 ~]# ll /data/
total 2050024
drwx----- 2 root root      16384 May 25 17:45 lost+found
-rw-r--r-- 1 root root 1048576000 May 25 17:46 test1.txt
-rw-r--r-- 1 root root 1048576000 May 25 17:46 test2.txt
```

再执行ll /data命令，

可以查看到，扩容后原分区的数据没有丢失，新增加的存储空间已经扩容到文件系统中。



# 扩容Windows文件系统

## 扩容Windows文件系统

最近更新时间: 2019-11-22 20:26:46

Windows 云硬盘扩容有以下两种场景：

- 对于新增的容量空间，建立独立的新分区，老的分区保持不变。
- 扩容旧的分区至新增的空量空间，并且保持老分区的数据不丢失。

以上两种场景在您的 windows 云硬盘升级成功之后（看到云硬盘容量变化），都可以通过 Windows 下的分区扩容工具分区助手完成分区扩容，并且保证原数据不会丢失。



# 前提条件

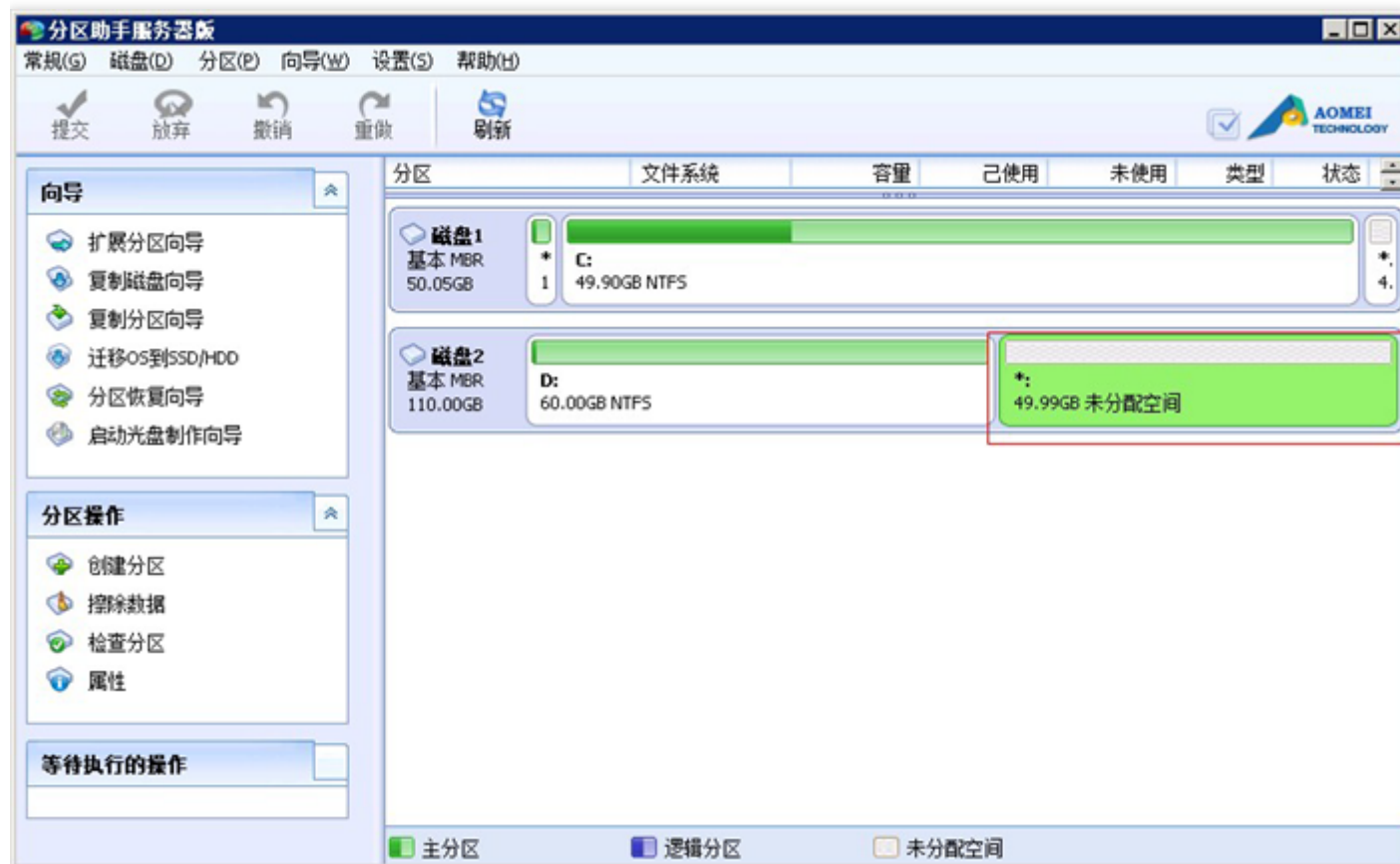
最近更新时间: 2023-03-13 16:34:42

- 用户需要先下载分区助手。
- 然后完成扩容实体云硬盘操作。
- 若此云硬盘上没有经过格式化和创建文件系统，直接在原有空白云硬盘基础上增加了容量，用户可以直接参考章节 Windows 系统下初始化云硬盘的相关操作。

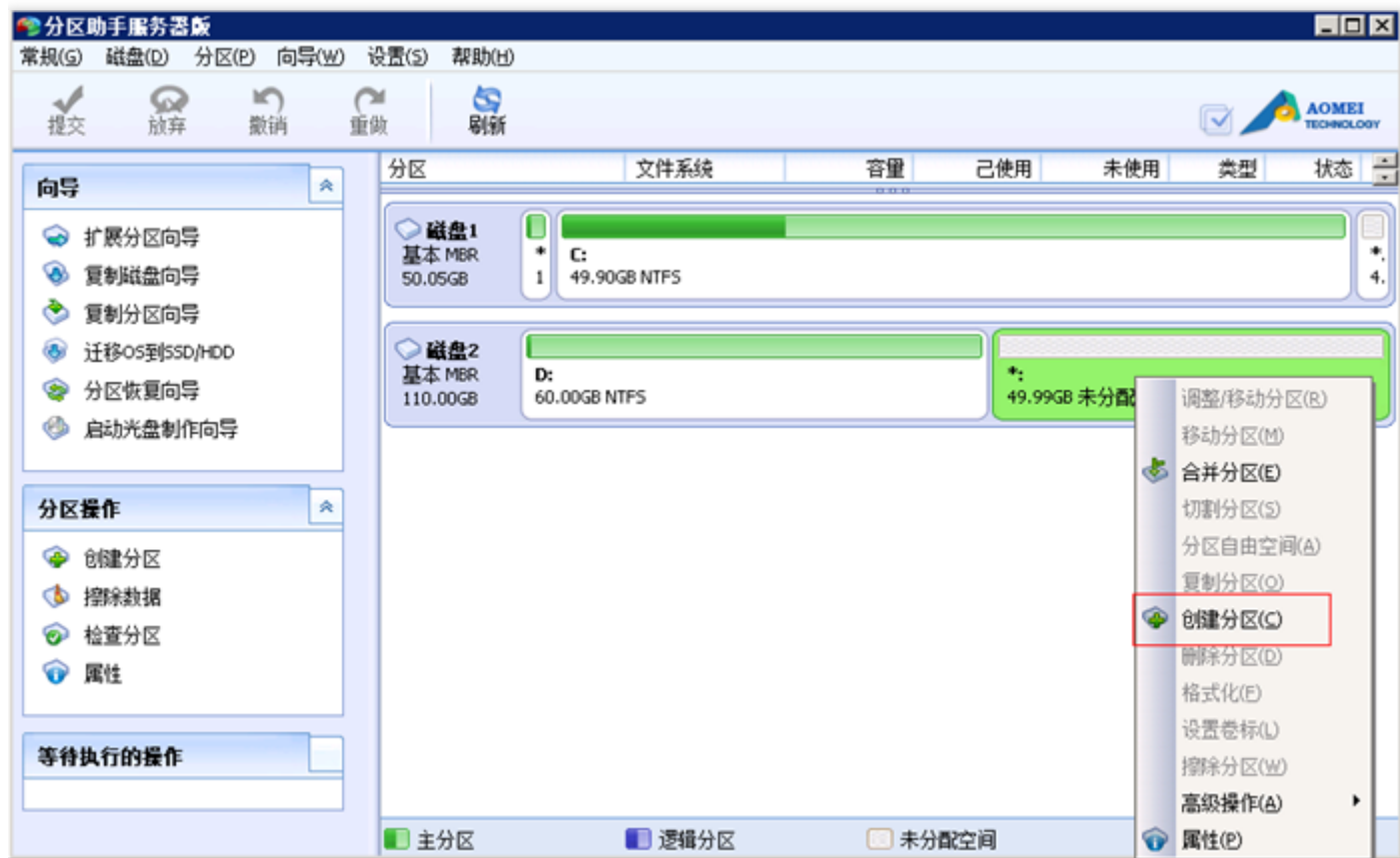
# 新空间格式化成一个独立分区

最近更新时间: 2019-11-30 15:11:13

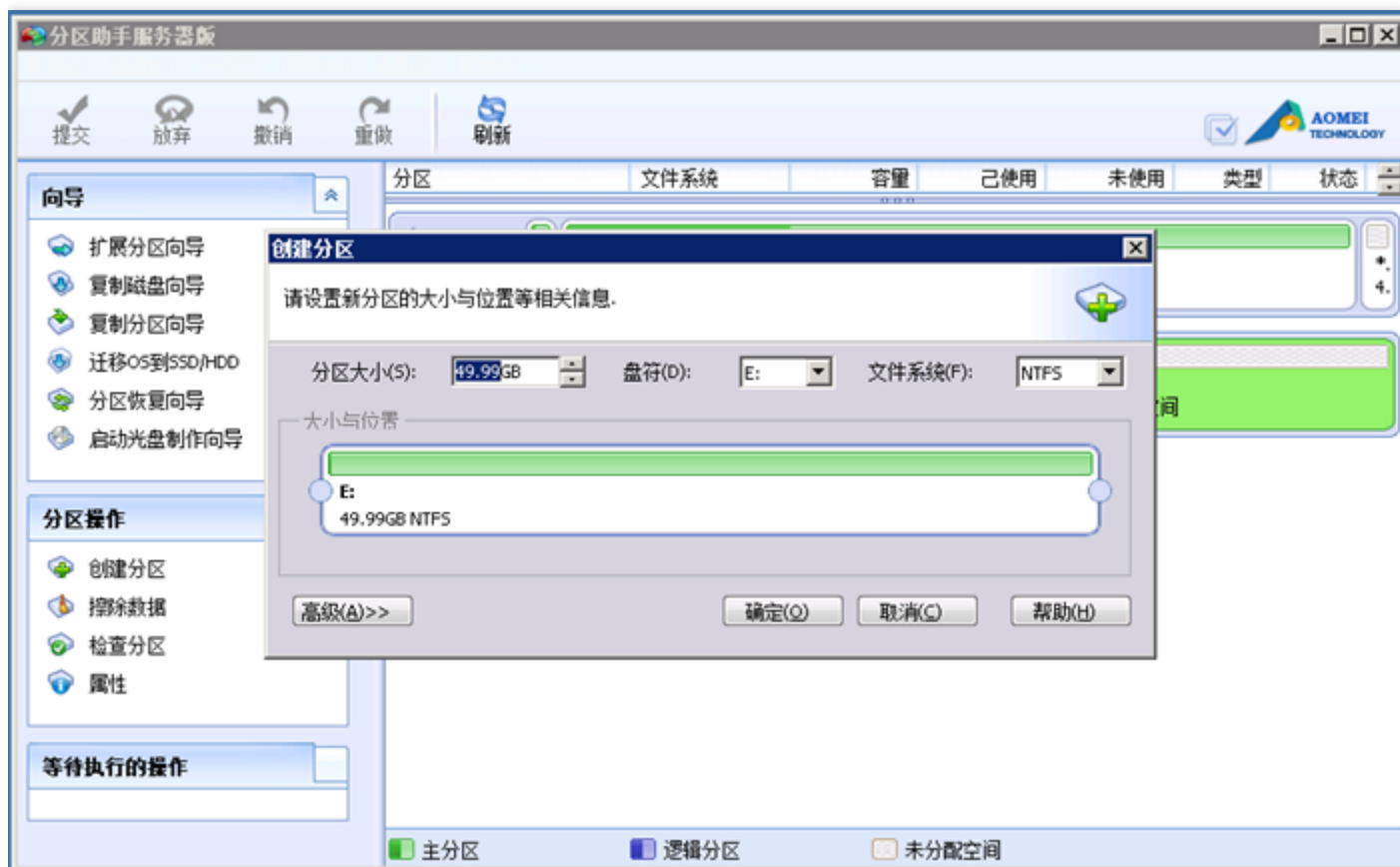
打开分区助手可以看到新扩容未使用的磁盘空间：



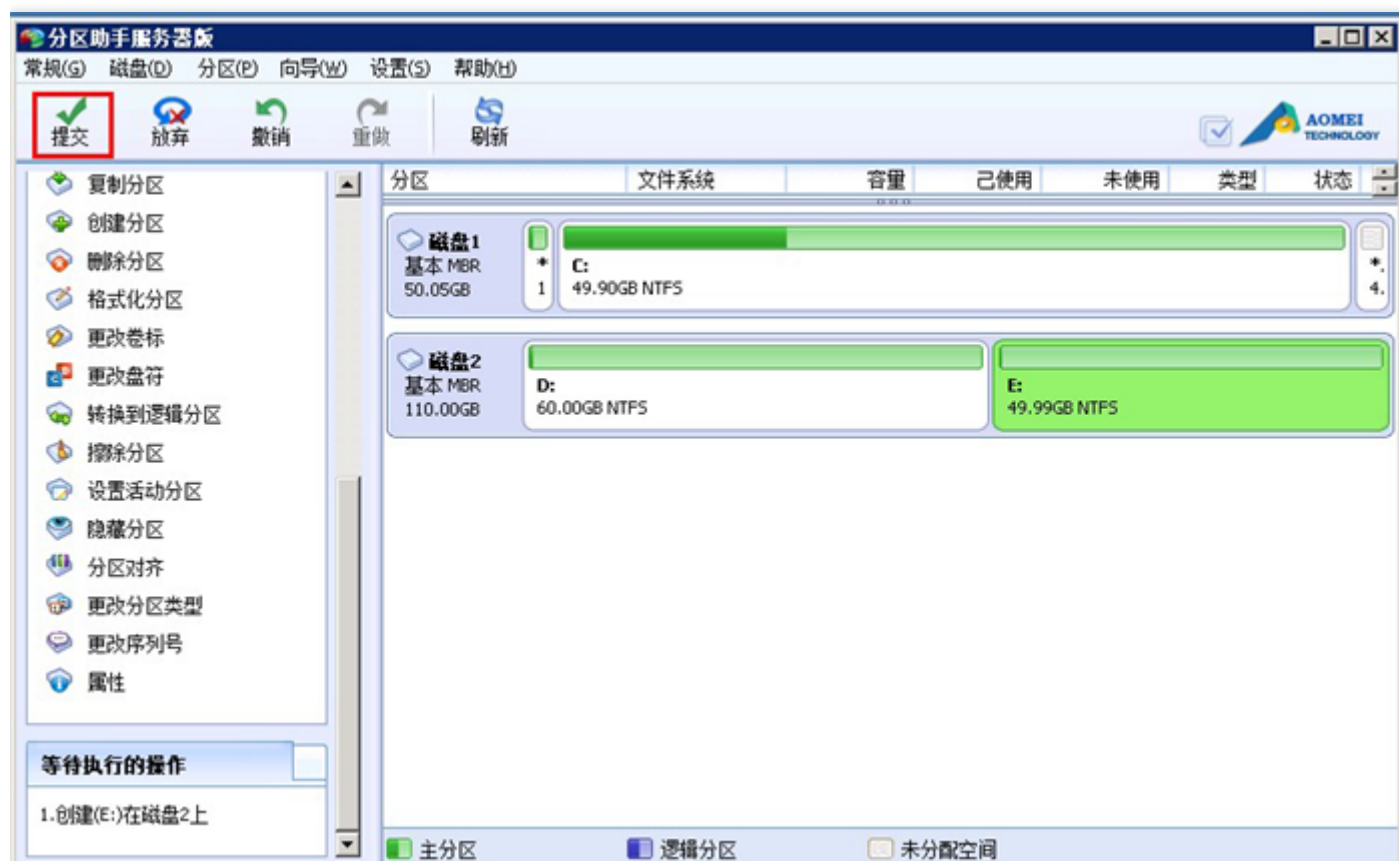
右键选中未使用的磁盘空间，选择【创建分区】：



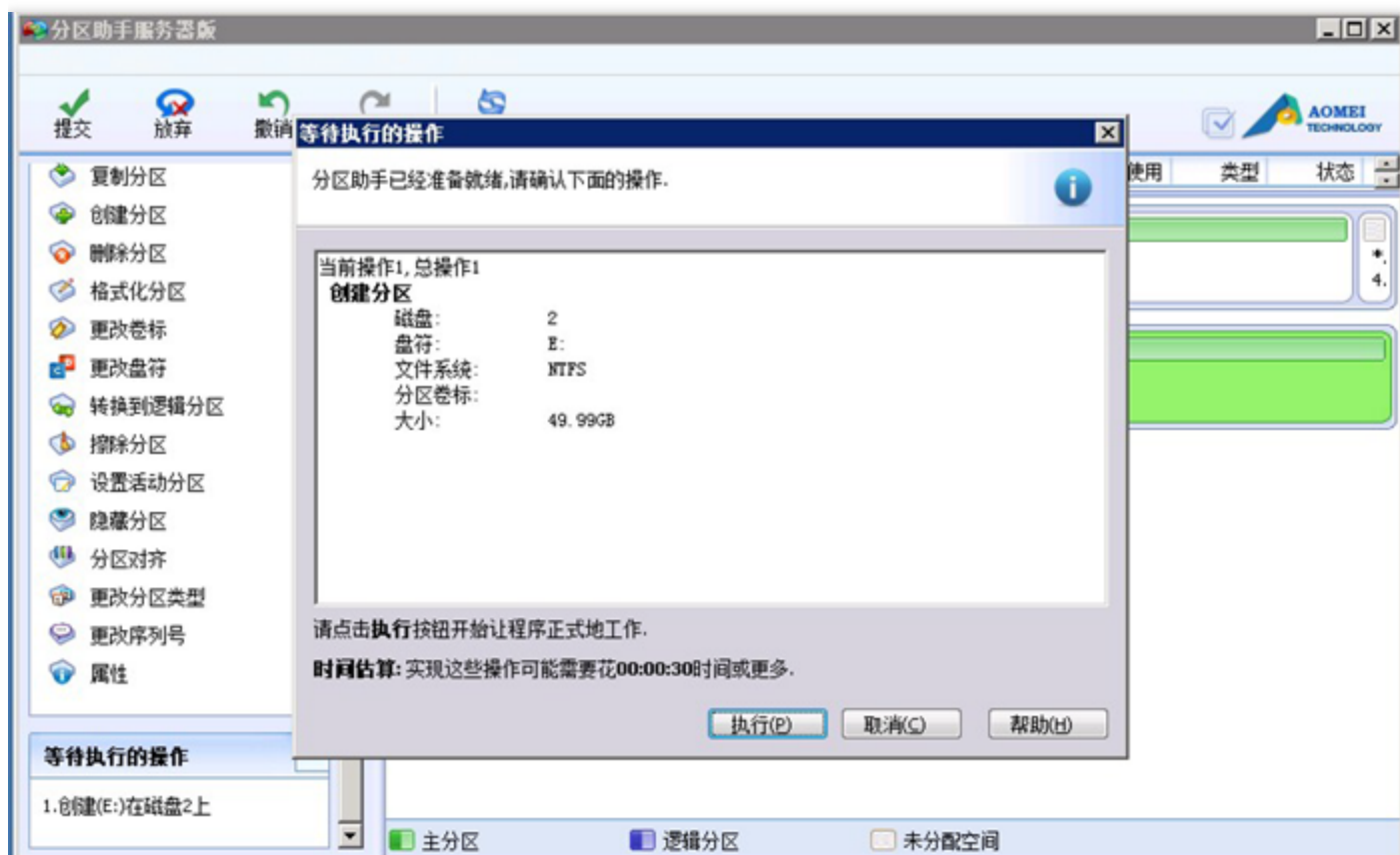
在弹出框中，输入需要的分区大小、盘符和文件系统，然后点击【确定】按钮：



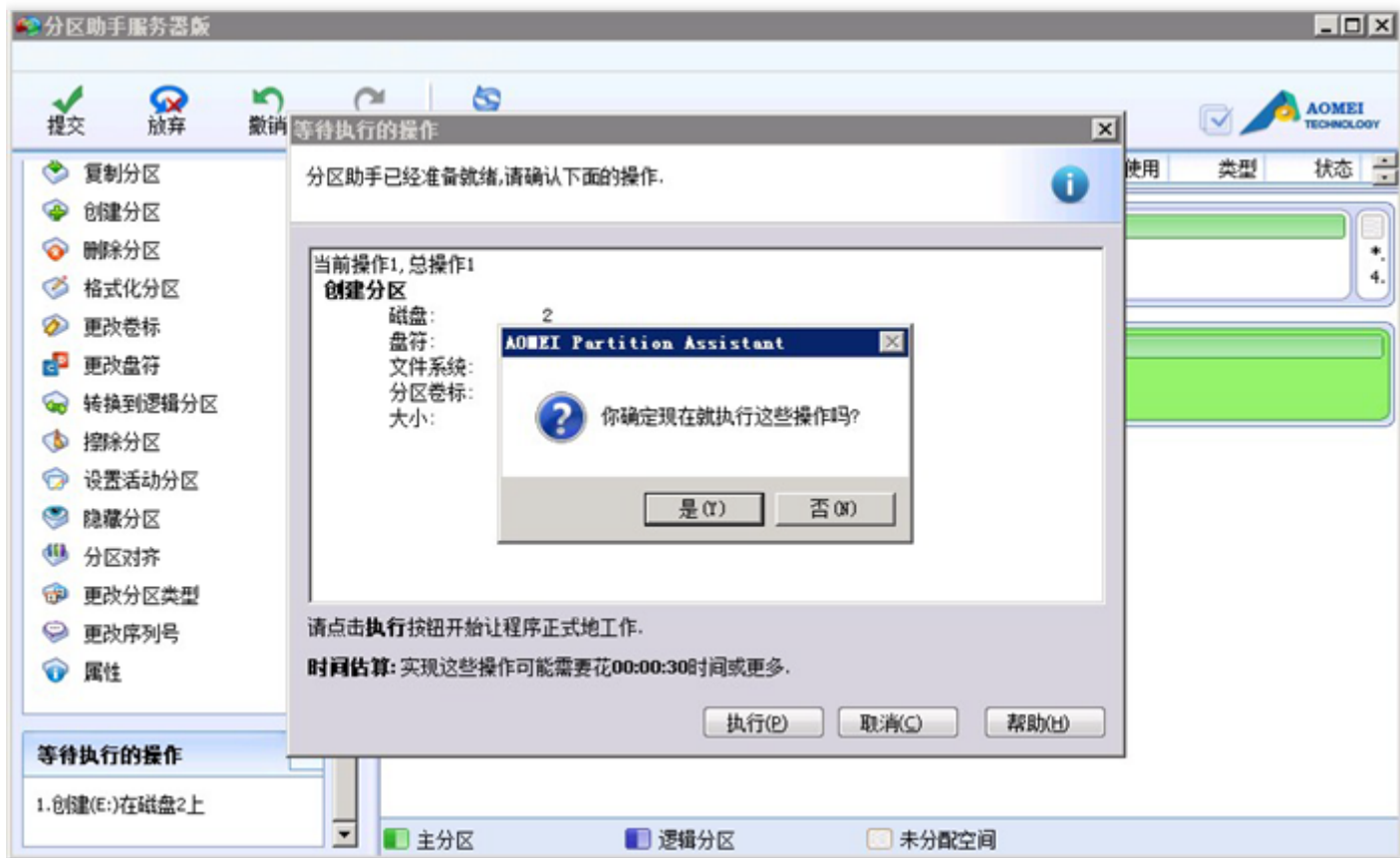
在左上角点击【提交】任务按钮：



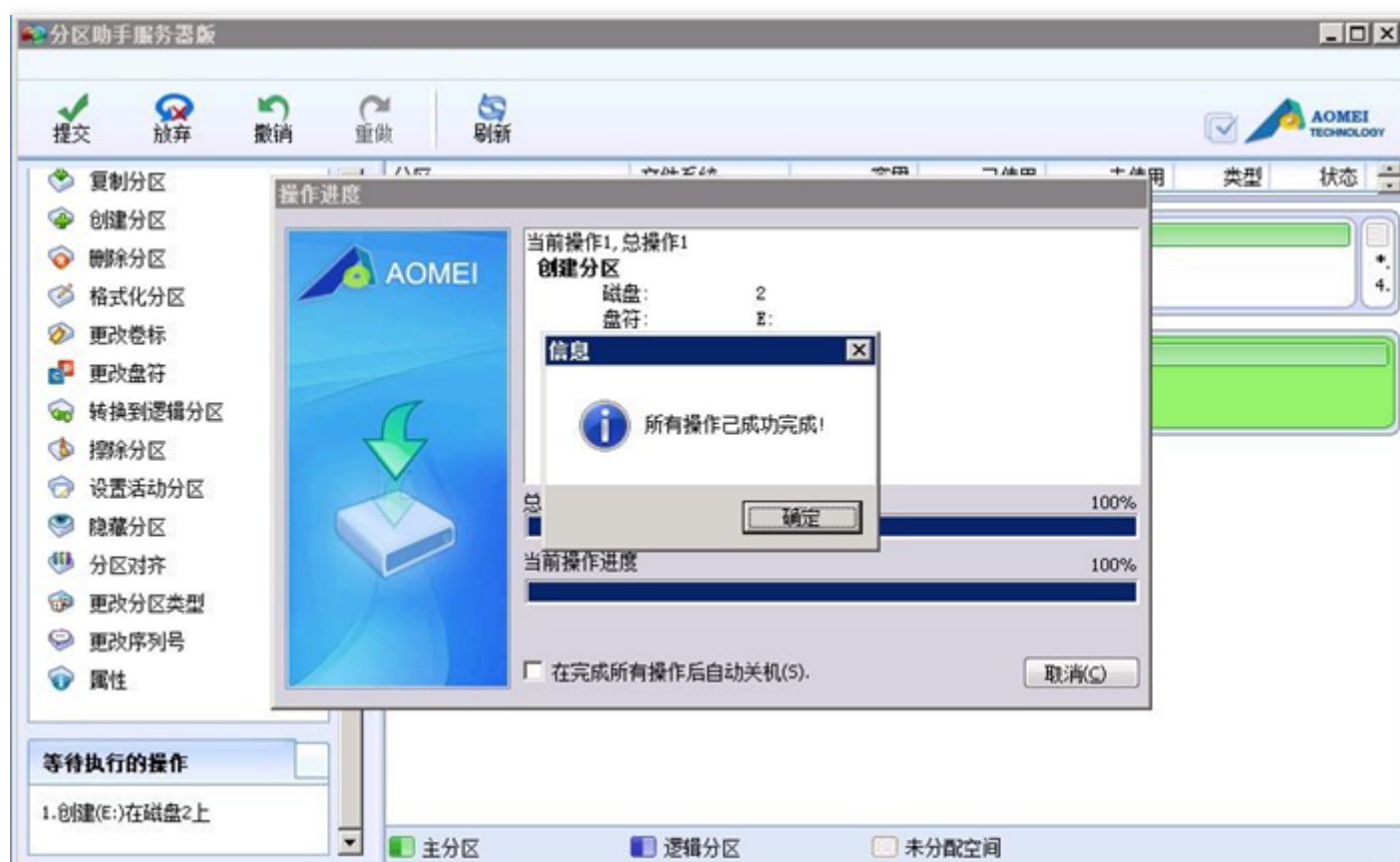
在弹出框中，确认格式分区的信息无误后，点击【执行】按钮：



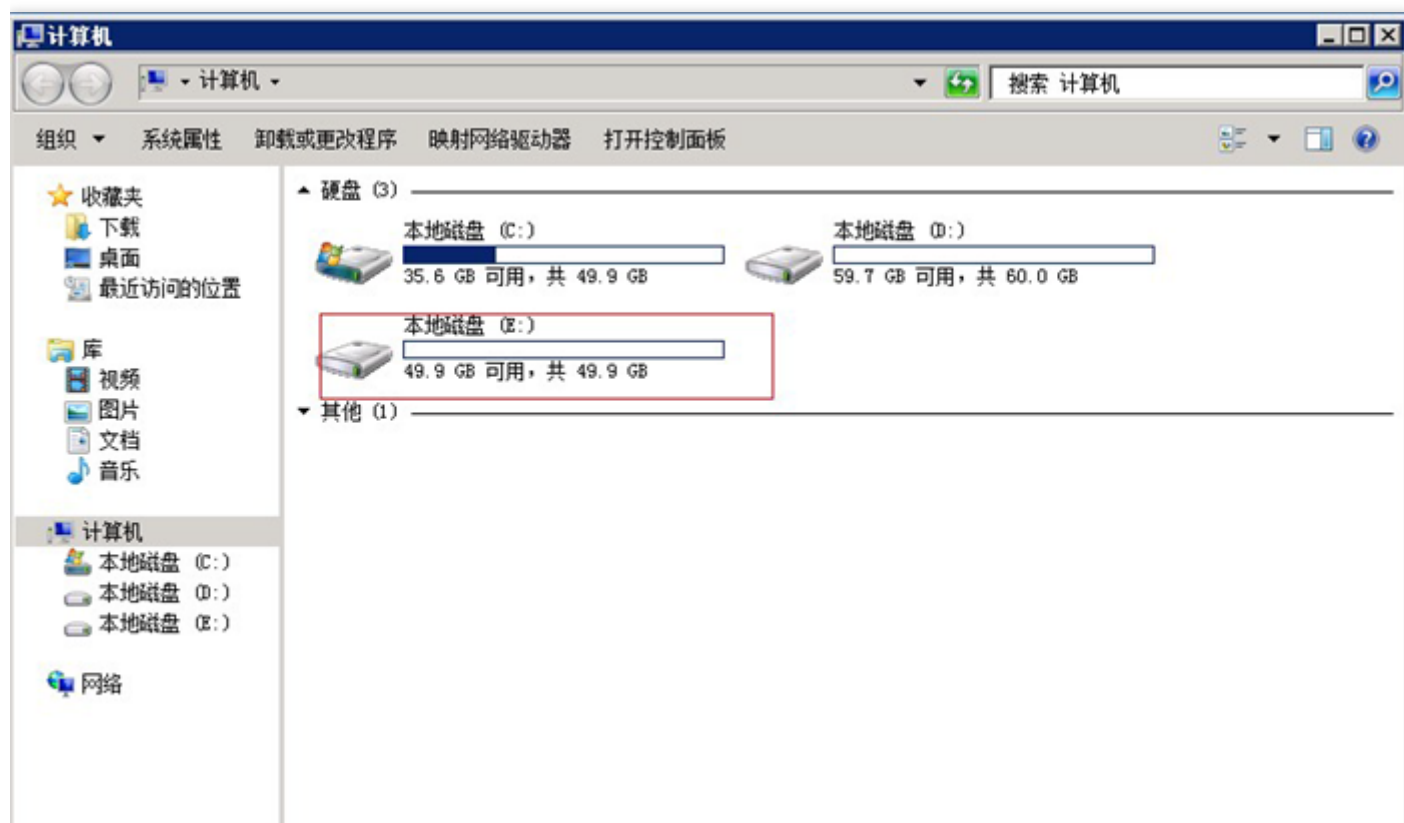
再次确认格式分区的信息无误后，在弹出框中点击【是】按钮：



创建完成后，点击【确认】按钮：



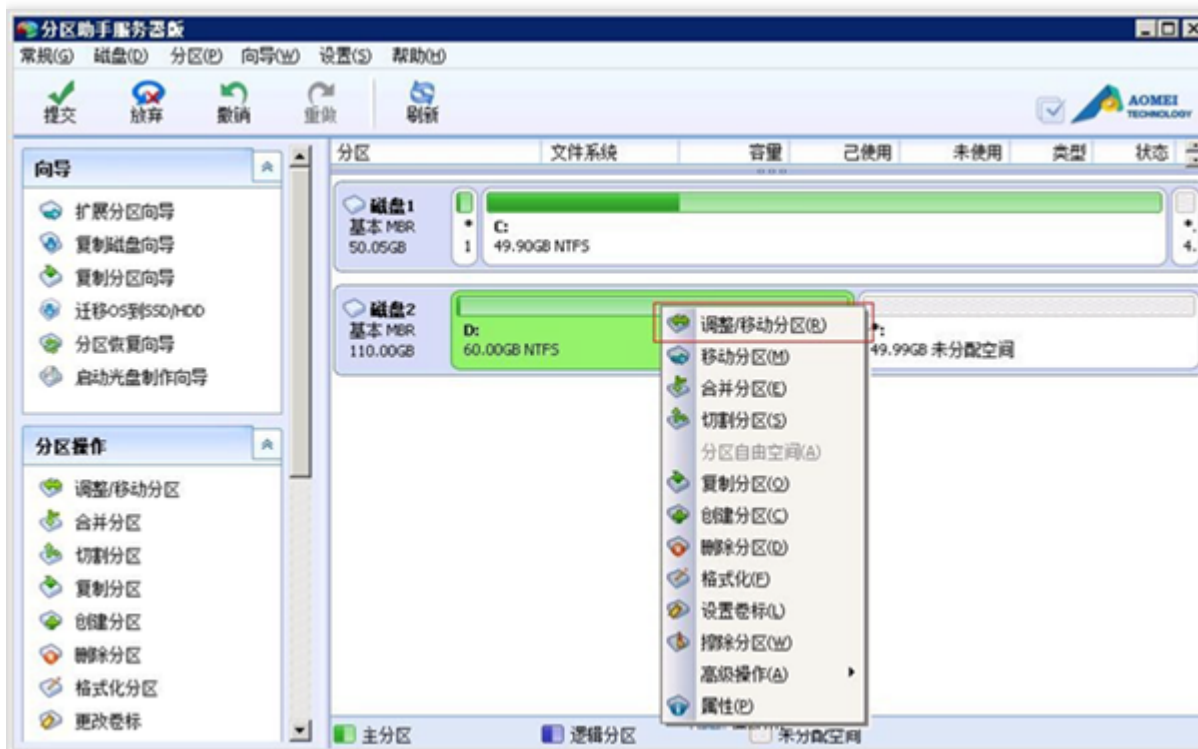
打开【我的电脑】可以看到新创建的磁盘分区(此例中新创建的是E盘):



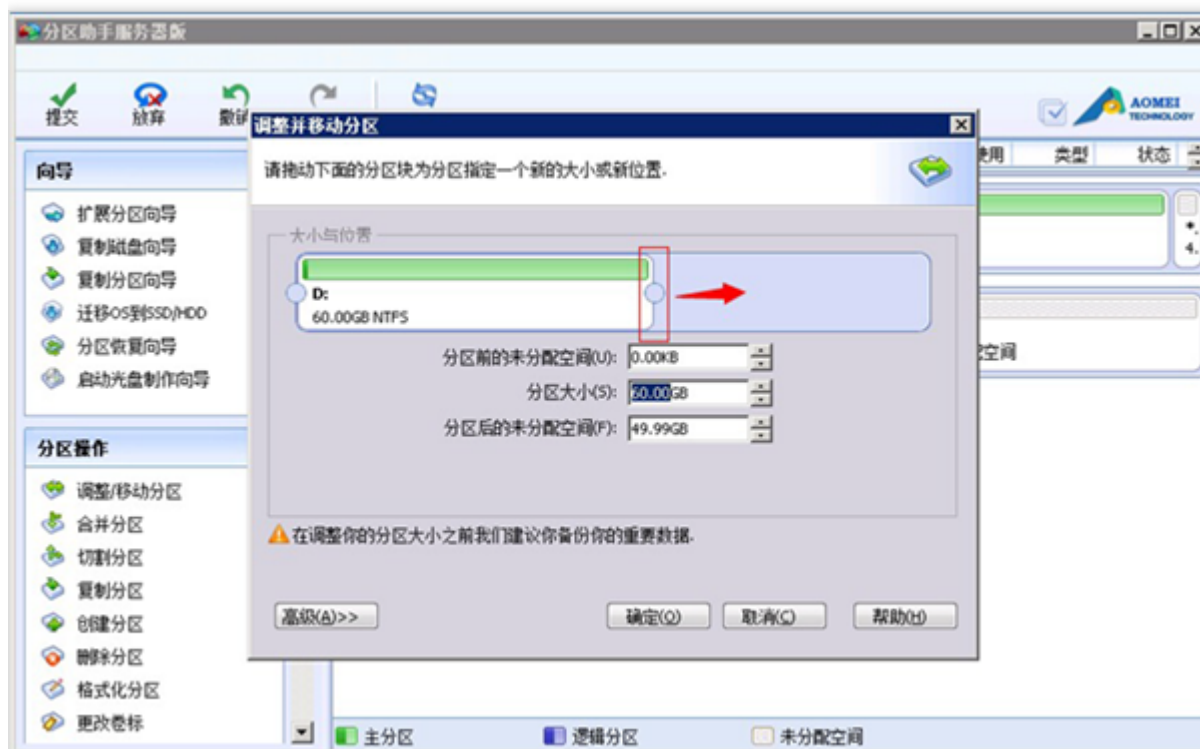
# 新空间增加到已有分区空间中

最近更新时间: 2019-11-30 15:11:13

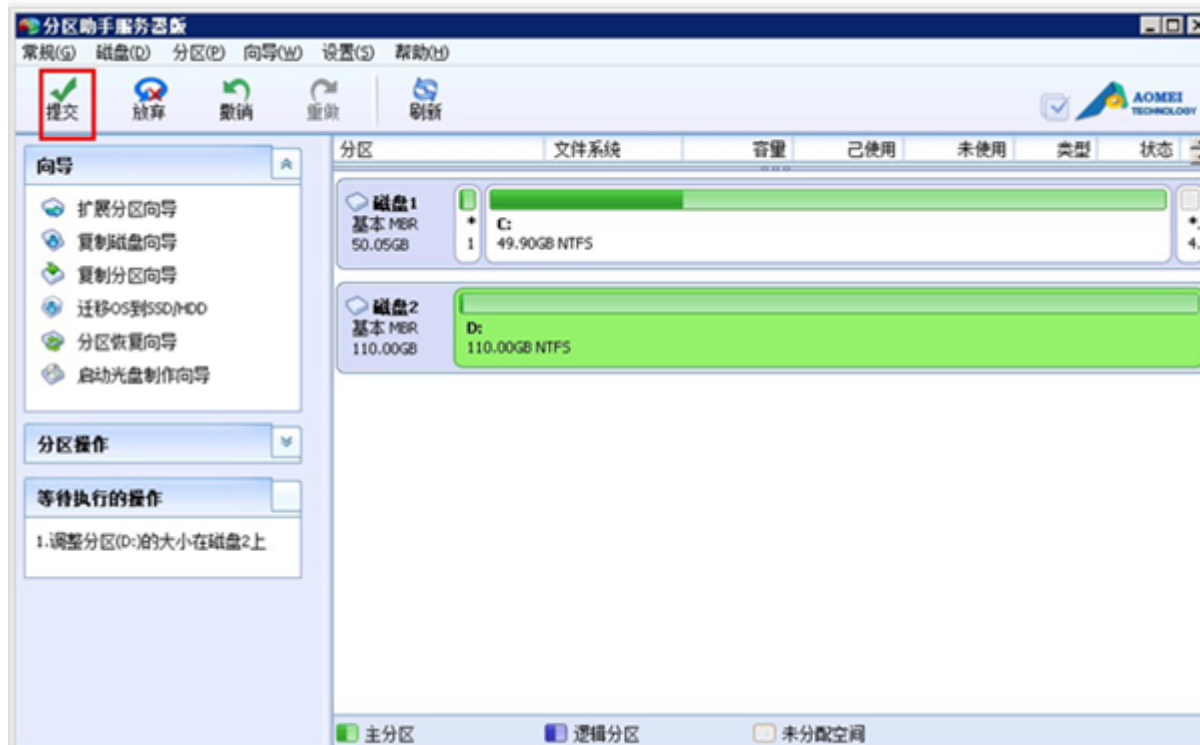
右键点击需要扩容的分区，选择【调整/移动分区(R)】：



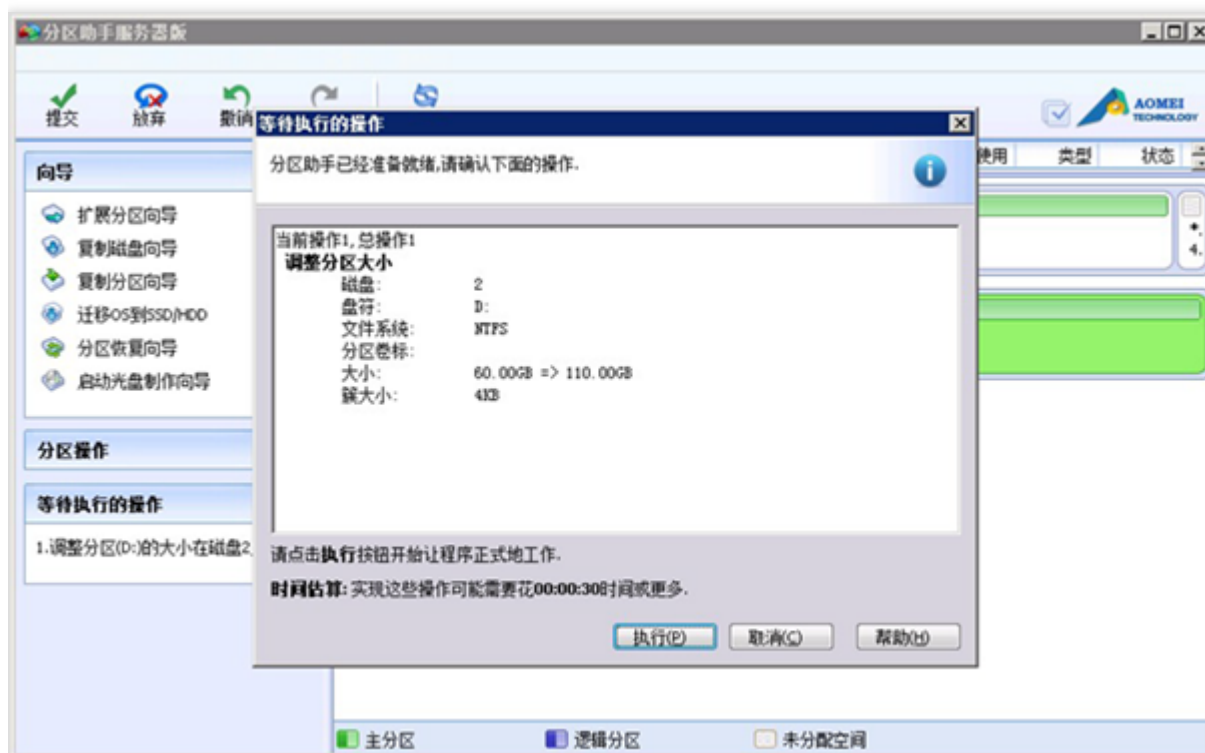
在弹出框中，如图所示，向右拖动小箭头调整分区需要的空间大小，然后点击【确认】按钮：



在左上角点击【提交】按钮：



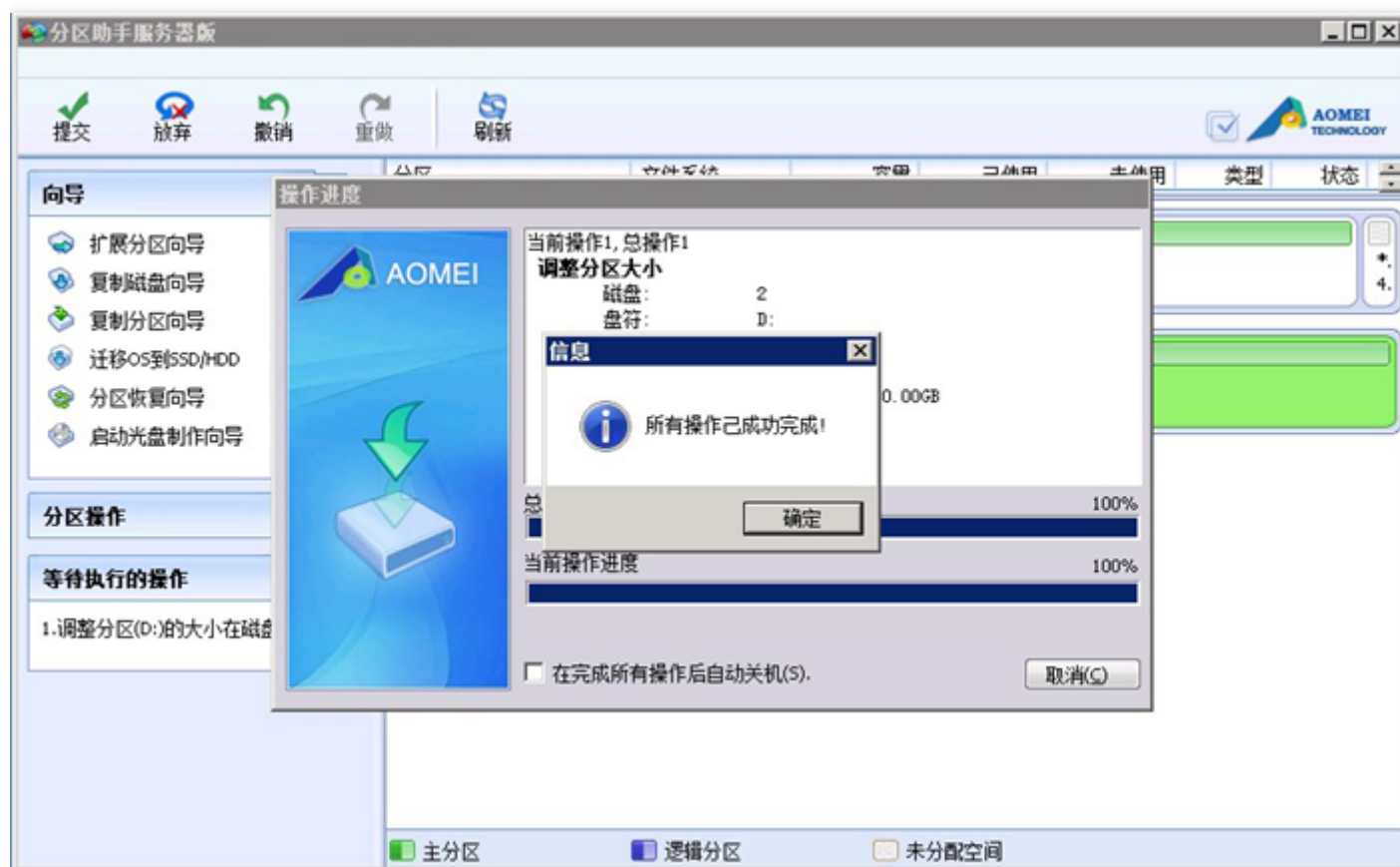
在弹出框中，确认分区扩容信息正确无误后，点击【确认】按钮：



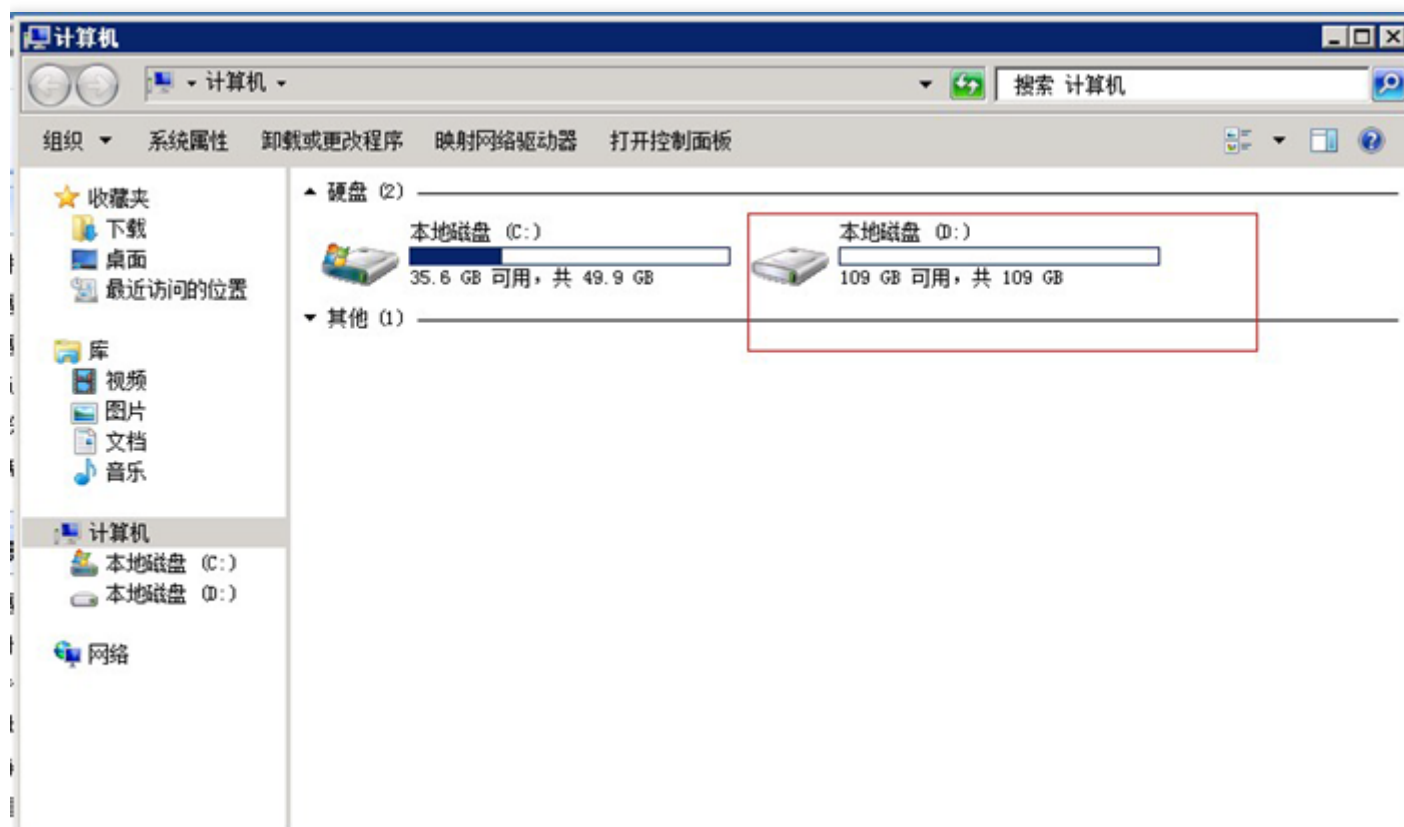
再次确认分区扩容信息后，点击【是】按钮：



等待分区扩容完成后，点击【确定】：



打开【我的电脑】，可以查看到扩容后的分区变化(此例中扩容的是D盘，由60G扩容至109G)：





# 卸载弹性云盘

## 卸载弹性云盘

最近更新时间: 2019-11-04 02:27:44

当您需要将 弹性云盘 挂载到另一台实例上使用时，您可以主动地从实例断开该弹性云盘，并将其连接到其他实例上。卸载弹性云盘并不会清除该硬盘上的数据。

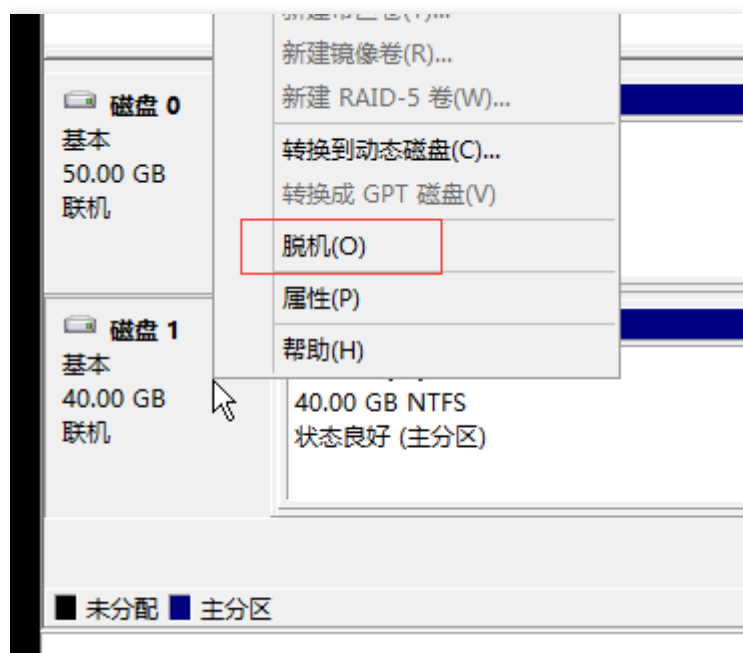
# 使用控制台卸载弹性云盘

最近更新时间: 2019-11-30 15:11:13

目前支持对作为数据盘的普通弹性云盘进行卸载，不可卸载系统盘。

卸载数据盘时，请确保您了解以下事项：

- 在 Windows 操作系统下，为了保证数据完整性，建议您暂停对该磁盘的所有文件系统的读写操作，否则未完成读写的数据会丢失。解挂弹性云盘时需要先将磁盘设为脱机状态，否则在不重启云主机的情况下，您可能将无法再次挂载弹性云盘。



- 在 Linux 操作系统下，您需要先登录实例，并对需要卸载的弹性云硬盘进行 unmount 操作，命令执行成功后再进入控制台对磁盘进行卸载操作。若未进行 unmount 操作直接被强制解挂后，关机时和开机时可能会出现以下问题：



```
Checking filesystems
/dev/vda1: clean, 35630/524288 files, 335690/2096474 blocks
fsck.ext3: Unable to resolve 'UUID=dabe8ee8-221b-44c7-9074-4d3f8fc4ae44'
fsck.ext3: No such file or directory while trying to open /dev/disk/by-id/virtio-disk-ezy5q5l6-part5
/dev/disk/by-id/virtio-disk-ezy5q5l6-part5:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

[FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

```
Checking filesystems
/dev/vda1: clean, 35630/524288 files, 335690/2096474 blocks
fsck.ext3: Unable to resolve 'UUID=dabe8ee8-221b-44c7-9074-4d3f8fc4ae44'
fsck.ext3: No such file or directory while trying to open /dev/disk/by-id/virtio-disk-ezy5q5l6-part5
/dev/disk/by-id/virtio-disk-ezy5q5l6-part5:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

[FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

1. 登录建行云控制台。
2. 进入【云服务器】-【云硬盘】选项卡。
3. 在云硬盘列表页，点击状态为已挂载、支持挂载/卸载的云硬盘后的【更多】-【卸载】按钮进行单盘卸载；或在云硬盘列表页，勾选状态为已挂载、支持挂载/卸载状态的云硬盘，点击顶部【挂载】按钮进行批量挂载。
4. 在弹出的对话框中确认警告事项，点击【确认】按钮。



# 删除云硬盘

## 删除云硬盘

最近更新时间: 2023-03-13 16:34:38

由于云硬盘的生命周期根据类型不同而有差异：弹性云盘的生命周期独立于云服务器实例，因此可以独立于 CVM 实例而销毁；非弹性云硬盘的生命周期跟随云主机，只能在云主机销毁时被删除。



# 删除弹性云盘

## 删除弹性云盘

最近更新时间: 2023-03-13 16:34:38

弹性云盘的生命周期独立于云服务器实例，可以独立于 CVM 实例而销毁。



# 快照

## 什么是快照

## 什么是快照

最近更新时间: 2023-03-13 16:34:38

快照是建行云提供的一种数据备份方式，通过对指定云硬盘进行完全可用的拷贝，使该备份独立于云硬盘的生命周期。快照包括硬盘在拷贝开始的时间点的映像。快照不占据用户的存储空间，从而进一步确保了备份的可靠性。快照属于增量备份，这意味着仅保存设备上在最新快照之后有更改的数据，这将尽可能缩短创建快照所需的时间，且可以节省存储成本。

可以基于快照创建新的云硬盘，这样云硬盘在初始状态就具有快照中的数据，是原始云硬盘的精确副本。快照具有地域属性，您只能在需要创建云硬盘 的同一地域下使用快照。有关更多信息，请参阅快照创建云硬盘。



# 快照的作用

最近更新时间: 2023-03-13 16:34:38

1. 能够进行在线数据的实时副本 快照是对云硬盘的完全可用拷贝。在该盘在出现问题时，可以快速恢复到未出问题前的状况。重大变更前对磁盘做快照，若变更失败可用于回滚。
2. 关键里程碑的持久备份 快照可以用于保存业务数据的持久备份，可以保留业务数据的里程碑状态。
3. 快速创建新硬盘 当磁盘A创建了快照文件，用户可使用该快照文件，快速克隆多块磁盘，达到快速部署服务器的目的。



# 使用快照的业务场景

最近更新时间: 2023-03-13 16:34:38

快照是一种便捷高效的数据保护服务手段，推荐应用于以下业务场景中：

- 数据日常备份：系统盘、数据盘的日常备份，您可以利用快照定期备份重要业务数据，以应对误操作、攻击或病毒等导致的数据丢失风险。
- 快速数据恢复：更换操作系统、应用软件升级或业务数据迁移等重大操作前，您可以创建一份或多份数据快照。一旦升级或迁移过程中出现任何问题，即可以通过数据快照及时将业务恢复到正常的系统数据状态。
- 生产数据的多副本应用：您可以通过创建生产数据快照，为数据挖掘、报表查询、开发测试等应用提供近实时的真实生产数据。



# 注意事项

最近更新时间: 2023-03-13 16:34:38

数据库业务：Flush & Lock Table 文件系统：Sync 操作。



# 创建快照

## 创建快照

最近更新时间: 2023-03-13 16:34:38

数据被写入了云硬盘之后，您可以定期创建快照作为数据备份使用。建行云使用增量的方式创建快照，即仅创建与上一次快照相比更改了的数据，因此在数据量改动不大的情况下能够在较快的时间内完成快照的创建。尽管快照是以增量方式保存的，但是快照删除流程不会影响您使用任何快照数据，未删除的快照均能将云硬盘恢复至该快照状态。

您可以在云硬盘的任何状态下创建快照，但是，快照只能保存创建时间点时已经写入云硬盘的数据，若应用程序或其他进程在该时刻正在向该云硬盘写入数据，可能无法被保存下来。如果您可以将所有文件写入暂停一段时间并创建快照，则该快照应该是完整的。如果无法暂停，则建议您将该云硬盘从实例中卸载、创建快照然后重新连接到 CVM 实例上。



# 使用控制台创建快照

最近更新时间: 2019-11-22 20:26:36

1. 打开云服务器CVM控制台 。
2. 单击导航窗格中的【云硬盘】。
3. 点击需要创建快照的列表项后的【创建快照】按钮。
4. 等待快照创建。



# 快照回滚

## 快照回滚

最近更新时间: 2023-03-13 16:34:38

使用快照的数据回滚云硬盘可以使该云硬盘的数据恢复到创建快照时的状态，在某些变更导致数据错误或数据丢失时非常有用。

快照仅能在其创建来源的云硬盘上进行回滚操作，如果需要在其他云硬盘上获得快照数据，请使用 快照创建云硬盘 功能。

需要注意的是：使用快照回滚云硬盘时，如果云硬盘正挂载在子机上使用，子机需要处于关机状态；如果云硬盘处于非挂载状态则可以直接回滚。



# 使用控制台回滚快照

最近更新时间: 2019-11-22 23:31:17

1. 打开云服务器CVM控制台。
2. 单击导航窗格中的【快照】。
3. 在快照列表中选中磁盘想要回滚的快照，点击【回滚】。



# 快照创建云硬盘

## 快照创建云硬盘

最近更新时间: 2023-03-13 16:48:30

从快照创建的云硬盘将在初始状态下就拥有快照的全部数据，因此不需要再进行分区、格式化及创建文件系统 等操作，因为格式化会擦除云硬盘上的所有数据。用户在使用快照创建云盘，并 连接到云服务器实例 后可以正常读写快照上的所有数据。因此，快照是数据共享和迁移的重要方式。



# 使用控制台从快照创建云硬盘

最近更新时间: 2023-03-13 16:48:30

1. 打开云服务器CVM控制台。
2. 单击导航窗格中的【快照】。
3. 在快照列表选中创建磁盘所要使用的数据盘快照，点击【使用快照创建磁盘】。
4. 在弹出框中选择地域/可用区、容量、申请数量和，点击【确定】按钮。

注：

- 新申请的磁盘容量默认等于快照大小，您可以调整容量大于默认值。
- 单次最多可创建50块弹性云硬盘。



# 删除快照

## 删除快照

最近更新时间: 2023-03-13 16:48:27

对于不会再使用到的数据状态，您可以删除该时刻创建的快照。删除快照时仅删除该快照专有的数据，不会对创建快照的云硬盘有任何影响。并且，删除某块云硬盘较早时间创建的快照也不会影响您使用较晚时间的快照，亦即：建行云提供的每份快照数据都可以单独地还原云硬盘至该时刻数据状态。



# 使用控制台删除快照

最近更新时间: 2023-03-13 16:48:27

1. 打开云服务器CVM控制台。
2. 单击导航窗格中的【快照】。
3. 点击需要删除的快照项后的【删除】按钮。  
或批量删除：
4. 打开云服务器CVM控制台。
5. 单击导航窗格中的【快照】。
6. 勾选所有您要删除的快照（确保快照没有在任务状态中），点击删除。



# 定期快照

## 定期快照策略说明

最近更新时间: 2023-03-13 16:48:27

**对象：**所有 CBS 云硬盘（包括系统盘和数据盘）。

**执行策略：**自动快照时间点可具体到每小时-每天，执行策略保存后，长期有效。执行策略可修改，修改成功后立即生效。

**定期销毁（重要）：**定期快照，根据创建规则，可设定定期销毁的时间点（默认保留7天后自动删除），也可设置为长期保留。

**批量：**先设定定期快照策略，可勾选多块云硬盘，针对多块云硬盘批量执行同一策略。

**命名规则：**自动快照的命名：snap\_yyyyMMdd\_HH，yyyyMMdd 是当天的日期，HH 为小时，例如 snap\_20220418\_11 表示 2022年4月18日11时创建的自动快照。您也可以自行修改快照命名。

**生命周期（重要）：**快照的生命周期有2种，1、手动创建的快照，生命周期默认为长期保存；2、定期快照，根据创建规则，可设定定期销毁的时间点，该时间点可在快照详情页更改为长期保存，需要手动触发。

**快照冲突：**自动快照与自定义快照使用上没有冲突，当正在对某一块云盘执行自动快照时，用户需要等待自动快照完成后，才能手动创建快照（反之同理）；如果云盘数据量大，一次打快照时长超过两个自动快照时间点间隔，则下一个时间点不打快照自动跳过。例如：用户设置9:00、10:00、11:00为自动快照时间点，9:00打快照的时候时长为70分钟，也就是10:10才打完，那10:00预设时间点将不打快照，下个快照时间点为11:00。

**ASP：**ASP指的是定期快照策略，即Auto Snapshot Policy。

**地域属性：**快照的地域属性为地域，即使用武汉标准区快照文件可创建武汉标准一区、武汉标准二区的云硬盘。

**保存周期：**对于自动快照，控制台会显示回收的倒计时时间。对于手动创建快照，显示永久保留。自动快照可手动点击改为永久保留。

**ASP暂停功能：**ASP自动快照策略提供手动触发暂停的功能，暂停后，则不会再自动创建快照。但自动快照的生命周期不受影响，到期后会自动退订。例如生命周期为7天，到期后依然清退。

**操作日志：**操作日志会显示所有自动快照的创建过程，与手动添加的快照相同。

最近更新时间: 2023-03-13 16:48:27

您可在快照控制台创建定期快照策略。

| 定期快照策略                       |        | 武汉                    |                                     |                     |                                                               |
|------------------------------|--------|-----------------------|-------------------------------------|---------------------|---------------------------------------------------------------|
| 新建                           |        | 删除                    |                                     | 支持快照策略ID、名称搜索       |                                                               |
| ID/名称                        | 关联磁盘数量 | 策略详情                  | 定期快照                                | 创建时间                | 操作                                                            |
| ecs-1ba0f1ee<br>ecs-1ba0f1ee | 0个     | 每周日，03:00，自动创建快照， ... | <input checked="" type="checkbox"/> | 2020-08-07 14:51:28 | <a href="#">修改策略</a> <a href="#">关联云硬盘</a> <a href="#">删除</a> |
| ecs-afac9d47<br>ecs-afac9d47 | 0个     | 每周日，04:00，自动创建快照， ... | <input checked="" type="checkbox"/> | 2020-08-07 11:34:22 | <a href="#">修改策略</a> <a href="#">关联云硬盘</a> <a href="#">删除</a> |
| ecs-02e9beae<br>ecs-02e9beae | 0个     | 每周一、三、五，14:00，自动创...  | <input checked="" type="checkbox"/> | 2020-08-07 11:07:42 | <a href="#">修改策略</a> <a href="#">关联云硬盘</a> <a href="#">删除</a> |



新建快照策略



名称

所属地域

武汉

备份日期

☐ 每周日

☐ 每周一

☐ 每周二

☐ 每周三

☐ 每周四

☐ 每周五

☐ 每周六

备份时间点

☐ 00:00

☐ 01:00

☐ 02:00

☐ 03:00

☐ 04:00

☐ 05:00

☐ 06:00

☐ 07:00

☐ 08:00

☐ 09:00

☐ 10:00

☐ 11:00

☐ 12:00

☐ 13:00

☐ 14:00

☐ 15:00

☐ 16:00

☐ 17:00

☐ 18:00

☐ 19:00

☐ 20:00

☐ 21:00

☐ 22:00

☐ 23:00

快照保留时间

☒ 保留

-

7

+

天后自动删除

☐ 永久保留

首次备份时间

确定

取消



# 关联云硬盘

最近更新时间: 2023-03-13 16:48:27

您可以将云硬盘添加到定期策略中，实现自动快照能力。

关联云硬盘

选择云硬盘

搜索云硬盘名称

ID/名称

磁盘属性

容量

☐

disk-0y57fszm  
tesgs

系统盘

50G

☐

disk-19ewnla8  
数据盘

系统盘

20G

☐

disk-qum2m3gm  
ssssssss

数据盘

210G

☒

disk-8hvjbv52

数据盘

40G

☐

zhijiangliu\_resize\_t

数据盘

110G

☐

disk-g7kjpjyw  
aaaaa

数据盘

170G

该磁盘未挂载到服务器上，定期快照策略将暂停对该磁盘制作快照

支持按住Shift进行多选

已选择(1)

ID/名称

磁盘属性

容量

disk-8hvjbv52  
123dsf

数据盘

40G

×

确定

取消



# 将自动快照转为永久保留的快照

最近更新时间: 2023-03-13 16:48:27

## 快照信息

|      |                                     |
|------|-------------------------------------|
| 名称   | 111                                 |
| 快照ID | snap-4kua54r1                       |
| 地域   | 武汉                                  |
| 大小   | 20GB                                |
| 关联磁盘 | <a href="#">disk-3i8clzx0</a> (未命名) |
| 快照类型 | 数据盘                                 |
| 保留时间 | 永久保留                                |
| 创建时间 | 2022-08-23 15:01:10                 |



快照列表

console.yun.ccb.com/cvm/snapshot

模组在Windows上...

建行云

总览 云产品

tianlu@10... 费用

删除

支持快照ID、名称搜索

| ID/名称                                  | 状态 | 大小  | 快照类型 | 关联磁盘          | 磁盘挂载云... | 创建时间                | 快照共享状态 | 保留时间 | 操作       |
|----------------------------------------|----|-----|------|---------------|----------|---------------------|--------|------|----------|
| <input type="checkbox"/> snap_3wop6m8z | 正常 | 50G | 数据盘  | disk-p71566j6 | -        | 2022-10-19 16:00:08 | 0/50   | 永久保留 | 回滚 删除 更多 |
| <input type="checkbox"/> snap_3wop6m8z | 正常 | 50G | 数据盘  | disk-p71566j6 | -        | 2022-09-14 16:00:09 | 0/50   | 永久保留 | 回滚 删除 更多 |
| <input type="checkbox"/> snap_3wop6m8z | 正常 | 20G | 数据盘  | disk-p71566j6 | -        | 2022-08-23 15:01:10 | 0/50   | 永久保留 | 回滚 删除 更多 |
| <input type="checkbox"/> snap_3wop6m8z | 正常 | 50G | 数据盘  | disk-p71566j6 | -        | 2022-08-20 16:00:47 | 0/50   | 永久保留 | 回滚 删除 更多 |
| <input type="checkbox"/> snap_3wop6m8z | 正常 | 50G | 数据盘  | disk-p71566j6 | -        | 2022-08-04 16:00:09 | 0/50   | 永久保留 | 回滚 删除 更多 |
| <input type="checkbox"/> snap_3wop6m8z | 正常 | 50G | 数据盘  | disk-p71566j6 | -        | 2022-08-03 17:09:31 | 0/50   | 永久保留 | 回滚 删除 更多 |
| <input type="checkbox"/> snap-3wop6m8z |    |     |      | disk-p71566j6 |          | 2022-02-21          |        |      |          |

Windows taskbar

20:08



# 共享快照 操作场景

最近更新时间: 2023-03-13 16:51:22

共享快照是将已经创建好的数据盘快照共享给其他用户使用。用户可通过简单的操作相互共享快照，并利用获取的快照快速创建云硬盘并挂载使用。



# 注意事项

最近更新时间: 2023-03-13 16:51:22

- 每个快照最多可以共享给50个用户。
- 由系统盘创建的快照，不支持共享。
- 已共享给其他用户的快照需取消共享才可删除，您可单击目标快照 ID 进入快照详情页取消共享。
- 快照支持共享到对方账号相同地域内。
- 不支持将获取的共享快照再次共享给其他用户。



# 操作步骤

最近更新时间: 2023-03-13 16:51:22

## 1.获取对端账号唯一 ID

建行云共享快照通过对端主账号唯一 ID 进行识别，即您首先需获取对方所使用的建行云账号的主账号的唯一 ID。  
获取途径如下：

- (1) 登录云服务器控制台。
- (2) 单击右上角的账号名称，选择账号信息。
- (3) 在“账号信息”管理页面，查看并记录主账号的“账号ID”。
- (4) 通知对方将获取到的账号 ID 发送给自己。

## 2.进行快照共享

您可通过控制台，共享快照给其他用户：

- (1) 进入快照列表页面。
- (2) 选择需共享的快照所在行右侧的更多 > 共享快照。如下图所示：

| 删除                       |       | 云硬盘类型: 数据盘 多个关键字用竖线“ ”分隔，多个过滤标签用回车键分隔 |             |      |        |      |                        |                   | Q                        | 刷新 | 设置 | 下载 |
|--------------------------|-------|---------------------------------------|-------------|------|--------|------|------------------------|-------------------|--------------------------|----|----|----|
| <input type="checkbox"/> | ID/名称 | 状态                                    | 硬盘属性        | 关联硬盘 | 硬盘挂载实例 | 关联镜像 | 创建时间                   | 保留时间              | 操作                       |    |    |    |
| <input type="checkbox"/> | 快照名称  | 正常                                    | 数据盘<br>10GB | 测试   | -      | -    | 2021-05-30<br>09:05:16 | 将于2021-06-29 自动删除 | 回滚 删除 查看快照链<br>更多        |    |    |    |
| <input type="checkbox"/> | 快照名称  | 正常                                    | 数据盘<br>10GB | 测试   | -      | -    | 2021-05-29<br>09:05:12 | 将于2021-06-28 自动删除 | 新建云硬盘<br>创建自定义镜像<br>共享快照 |    |    |    |

- (3) 在弹出的“共享快照”窗口中，输入对端账号唯一 ID。
  - (4) 单击共享即可共享该快照。
  - (5) 通知对方前往 快照列表 页面，并选择共享快照页签，即可查看已共享的快照。
- 如需共享给多个用户，请重复上述步骤。



# 监控云硬盘

最近更新时间: 2021-09-01 17:37:45

为云硬盘提供良好的监控环境是保持数据高可靠性的重要部分。用户可以使用云监控监控已经挂载在实例上的云硬盘，所以需要收集云硬盘统计数据时，请先执行将云硬盘连接到 CVM 实例操作。通过云监控，用户可以查看云硬盘的指标数据，并分析和设置有关云硬盘的告警。目前云监控为云硬盘提供了如下的监控指标：

| 指标名称               | 指标中文名称   | 计算方式                   | 指标含义                                 | 单位   | 统计粒度<br>(period) |
|--------------------|----------|------------------------|--------------------------------------|------|------------------|
| disk_read_iops     | 硬盘读IOPS  | 块存储在统计周期内读 IOPS的平均值    | 每秒从块存储读到内存中的IO次数                     | 次数   | 5s、10s、60s、300s  |
| disk_read_traffic  | 硬盘读流量    | 块存储在统计周期内读吞吐量的平均值      | 数据从块存储读取到内存中的速率                      | MB/s | 5s、10s、60s、300s  |
| disk_write_iops    | 硬盘写IOPS  | 块存储在统计周期内写 IOPS的平均值    | 每秒从内存写到块存储中的IO次数                     | 次数   | 5s、10s、60s、300s  |
| disk_write_traffic | 硬盘写流量    | 块存储在统计周期内写吞吐量的平均值      | 数据从内存写入到块存储中的速率                      | MB/s | 5s、10s、60s、300s  |
| disk_await         | 硬盘IO等待时间 | 块存储在统计周期内 ioawait 的平均值 | 在采样周期内有百分之几的时间 CPU 空闲并且有仍未完成的 I/O 请求 | ms   | 5s、10s、60s、300s  |
| disk_svctm         | 硬盘IO服务时间 | 块存储在统计周期内 svctm 的平均值   | IO 服务时间                              | ms   | 5s、10s、60s、300s  |
| disk_util          | 硬盘IO繁忙比率 | 块存储在统计周期内 io_util 的平均值 | 硬盘有IO操作的时间（即非空闲时间）的比率                | %    | 5s、10s、60s、300s  |

有关具体的监控指标说明，请参考云监控产品文档。

云监控从运行状态下的云服务器实例中收集硬盘的原始数据，并将数据展示为易读的图标形式。统计数据默认保存



一个月，用户可以观察一个月来磁盘的对应情况，更好地了解使用量和读写等方面的数据。

用户可以从 [云监控控制台入口](#) 或 [云监控 API](#) 获取数据，同时控制台还提供了相应指标的可视化图标。有关更多内容，可以参考[获取特定指标的监控数据](#) 和 [查看监控图表](#)。



# 最佳实践

## 如何衡量云硬盘的性能

最近更新时间: 2023-03-13 17:54:11

### 1.衡量指标

由于不同应用程序的工作负载不同，若未提供足够的 I/O 请求来充分利用云硬盘时，可能无法达到云硬盘的最大性能。

一般使用以下指标衡量云硬盘的性能：

- IOPS：每秒读/写次数，单位为次（计数）。存储设备的底层驱动类型决定了不同的 IOPS。
- 吞吐量：每秒的读写数据量，单位为MB/s。
- 时延：I/O 操作的发送时间到接收确认所经过的时间，单位为秒。

### 2.测试工具

FIO 是测试磁盘性能的工具，用来对硬件进行压力测试和验证，本文以 FIO 为例。

使用 FIO 时，建议配合使用 libaio 的 I/O 引擎进行测试。请您自行安装 FIO 和 libaio。

警告：

- 请不要在系统盘上进行 FIO 测试，避免损坏系统重要文件。
- 为避免底层文件系统元数据损坏导致数据损坏，请不要在业务数据盘上进行测试。
- 请确保/etc/fstab文件配置项中没有被测硬盘的挂载配置，否则将导致云服务器启动失败。

### 3.测试对象建议

- 建议在空闲的、未保存重要数据的硬盘上进行 FIO 测试，并在测试完后重新制作被测硬盘的文件系统。
- 测试硬盘性能时，建议直接测试裸数据盘（如 /dev/vdb）。
- 测试文件系统性能时，推荐指定具体文件测试（如 /data/file）。

### 4.测试示例

不同场景的测试公式基本一致，只有 rw、iodepth 和 bs（block size）三个参数的区别。例如，每个工作负载适合最佳 iodepth 不同，取决于您的特定应用程序对于 IOPS 和延迟的敏感程度。

参数说明：

| 参数名      | 说明                                                                                                                                                         | 取值样例   |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| bs       | 每次请求的块大小。取值包括4k，8k，16k等。                                                                                                                                   | 4k     |
| ioengine | I/O 引擎。推荐使用 Linux 的异步 I/O 引擎。                                                                                                                              | libaio |
| iodepth  | 请求的 I/O 队列深度。                                                                                                                                              | 1      |
| direct   | 指定 direct 模式。 <ul style="list-style-type: none"><li>• True（1）表示指定 O_DIRECT 标识符，忽略 I/O 缓存，数据直写。</li><li>• False（0）表示不指定 O_DIRECT 标识符。默认为 True（1）。</li></ul> | 1      |
|          |                                                                                                                                                            |        |



|                 |                                                                                                  |          |
|-----------------|--------------------------------------------------------------------------------------------------|----------|
| rw              | 读写模式。取值包括顺序读（read）、顺序写（write）、随机读（randread）、随机写（randwrite）、混合随机读写（randrw）和混合顺序读写（rw, readwrite）。 | read     |
| time_based      | 指定采用时间模式。无需设置该参数值，只要 FIO 基于时间来运行。                                                                | N/A      |
| runtime         | 指定测试时长，即 FIO 运行时长。                                                                               | 600      |
| refill_buffers  | FIO 将在每次提交时重新填充 I/O 缓冲区。默认设置是仅在初始时填充并重用该数据。                                                      | N/A      |
| norandommap     | 在进行随机 I/O 时，FIO 将覆盖文件的每个块。若给出此参数，则将选择新的偏移量而不查看 I/O 历史记录。                                         | N/A      |
| randrepeat      | 随机序列是否可重复。<br>• True（1）表示随机序列可重复。<br>• False（0）表示随机序列不可重复。默认为 True（1）。                           | 0        |
| group_reporting | 多个 job 并发时，打印整个 group 的统计值。                                                                      | N/A      |
| name            | job 的名称。                                                                                         | fio-read |
| size            | I/O 测试的寻址空间。                                                                                     | 100GB    |
| filename        | 测试对象，即待测试的磁盘设备名称。                                                                                | /dev/sdb |

常见用例如下：

• bs = 4k iodepth = 1：随机读/写测试，能反映硬盘的时延性能 执行以下命令，测试硬盘的随机读时延。

```
fio -bs=4k -ioengine=libaio -iodepth=1 -direct=1 -rw=randread -time_based -runtime=600 -refill_buffers
-norandommap -randrepeat=0 -group_reporting -name=fio-randread-lat --size=10G -
filename=/dev/vdb
```

执行以下命令，测试硬盘的随机写时延。

```
fio -bs=4k -ioengine=libaio -iodepth=1 -direct=1 -rw=randwrite -time_based -runtime=600 -
refill_buffers -norandommap -randrepeat=0 -group_reporting -name=fio-randwrite-lat --size=10G -
filename=/dev/vdb
```

执行以下命令，测试 SSD 云硬盘的随机混合读写时延性能。

```
fio --bs=4k --ioengine=libaio --iodepth=1 --direct=1 --rw=randrw --time_based --runtime=100 --
refill_buffers --norandommap --randrepeat=0 --group_reporting --name=fio-read --size=1G --
filename=/dev/vdb
```



测试结果如下图所示：

```
fio-read: (g=0): rw=randrw, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=1
fio-3.1
Starting 1 process
Jobs: 1 (f=1): [m(1)][100.0%][r=3411KiB/s,w=3603KiB/s][r=852,w=900 IOPS][eta 00m:00s]
fio-read: (groupid=0, jobs=1): err= 0: pid=2377: Thu Jun 13 18:23:47 2019
  read: IOPS=880, BW=3523KiB/s (3607kB/s) (344MiB/100001msec)
    slat (nsec): min=2905, max=62479, avg=5254.61, stdev=2075.46
    clat (usec): min=205, max=6921, avg=463.65, stdev=259.48
      lat (usec): min=209, max=6925, avg=469.13, stdev=259.56
    clat percentiles (usec):
      | 1.00th=[ 245], 5.00th=[ 269], 10.00th=[ 293], 20.00th=[ 375],
      | 30.00th=[ 400], 40.00th=[ 416], 50.00th=[ 437], 60.00th=[ 457],
      | 70.00th=[ 478], 80.00th=[ 498], 90.00th=[ 545], 95.00th=[ 619],
      | 99.00th=[ 2057], 99.50th=[ 2376], 99.90th=[ 3294], 99.95th=[ 4015],
      | 99.99th=[ 6259]
    bw ( KiB/s): min= 2168, max= 4024, per=100.00%, avg=3522.64, stdev=310.95, samples=200
    iops       : min=  542, max= 1006, avg=880.66, stdev=77.74, samples=200
  write: IOPS=877, BW=3511KiB/s (3595kB/s) (343MiB/100001msec)
    slat (nsec): min=2981, max=58808, avg=5377.71, stdev=2079.36
    clat (usec): min=421, max=10492, avg=659.10, stdev=219.96
      lat (usec): min=428, max=10496, avg=664.70, stdev=220.05
    clat percentiles (usec):
      | 1.00th=[ 490], 5.00th=[ 523], 10.00th=[ 545], 20.00th=[ 562],
      | 30.00th=[ 578], 40.00th=[ 594], 50.00th=[ 611], 60.00th=[ 635],
      | 70.00th=[ 660], 80.00th=[ 693], 90.00th=[ 783], 95.00th=[ 914],
      | 99.00th=[ 1516], 99.50th=[ 1926], 99.90th=[ 3261], 99.95th=[ 3982],
      | 99.99th=[ 5342]
    bw ( KiB/s): min= 2296, max= 4008, per=100.00%, avg=3510.84, stdev=305.46, samples=200
    iops       : min=  574, max= 1002, avg=877.71, stdev=76.36, samples=200
  lat (usec)  : 250=0.76%, 500=40.51%, 750=51.04%, 1000=5.03%
  lat (msec)  : 2=1.90%, 4=0.71%, 10=0.05%, 20=0.01%
  cpu         : usr=0.50%, sys=1.52%, ctx=175841, majf=0, minf=29
  IO depths   : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
    submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
    complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
```

- bs = 128k iodepth = 32：顺序读/写测试，能反映硬盘的吞吐性能

执行以下命令，测试硬盘的顺序读吞吐带宽。

```
fio -bs=128k -ioengine=libaio -iodepth=32 -direct=1 -rw=read -time_based -runtime=600 -refill_buffers
-norandommap -randrepeat=0 -group_reporting -name=fio-read-throughput --size=10G -
filename=/dev/vdb
```

执行以下命令，测试硬盘的顺序写吞吐带宽。

```
fio -bs=128k -ioengine=libaio -iodepth=32 -direct=1 -rw=write -time_based -runtime=600 -refill_buffers
-norandommap -randrepeat=0 -group_reporting -name=fio-write-throughput --size=10G -
filename=/dev/vdb
```

执行以下命令，测试 SSD 云硬盘的顺序读吞吐性能。

```
fio --bs=128k --ioengine=libaio --iodpth=32 --direct=1 --rw=write --time_based --runtime=100 --
refill_buffers --norandommap --randrepeat=0 --group_reporting -rw --size=1G --filename=/dev/vdb
```



测试结果如下图所示：

```
fio-rw: (g=0): rw=write, bs=(R) 128KiB-128KiB, (W) 128KiB-128KiB, (T) 128KiB-128KiB, ioengine=libaio, iodepth=32
fio-3.1
Starting 1 process
Jobs: 1 (f=1): [W(1)][100.0%][r=0KiB/s,w=260MiB/s][r=0,w=2082 IOPS][eta 00m:00s]
fio-rw: (groupid=0, jobs=1): err= 0: pid=2679: Thu Jun 13 18:27:32 2019
write: IOPS=2081, BW=260MiB/s (273MB/s) (25.4GiB/100045msec)
  slat (nsec): min=2847, max=72524, avg=7739.21, stdev=3233.07
  clat (usec): min=1033, max=250494, avg=15341.09, stdev=28854.07
  lat (usec): min=1041, max=250503, avg=15349.03, stdev=28853.95
  clat percentiles (usec):
    | 1.00th=[ 1565],  5.00th=[ 1860], 10.00th=[ 2057], 20.00th=[ 2311],
    | 30.00th=[ 2540], 40.00th=[ 2769], 50.00th=[ 2999], 60.00th=[ 3326],
    | 70.00th=[ 3818], 80.00th=[ 5014], 90.00th=[82314], 95.00th=[84411],
    | 99.00th=[86508], 99.50th=[86508], 99.90th=[87557], 99.95th=[88605],
    | 99.99th=[90702]
  bw ( KiB/s): min=265708, max=319488, per=100.00%, avg=266498.36, stdev=3766.74, samples=200
  iops        : min= 2075, max= 2496, avg=2082.01, stdev=29.43, samples=200
  lat (msec)  : 2=8.46%, 4=64.09%, 10=11.90%, 20=0.18%, 50=0.01%
  lat (msec)  : 100=15.37%, 500=0.01%
  cpu         : usr=5.34%, sys=1.90%, ctx=63555, majf=0, minf=28
  IO depths   : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=100.0%, >=64=0.0%
    submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
    complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.1%, 64=0.0%, >=64=0.0%
    issued rw: total=0,208238,0, short=0,0,0, dropped=0,0,0
    latency   : target=0, window=0, percentile=100.00%, depth=32

Run status group 0 (all jobs):
  WRITE: bw=260MiB/s (273MB/s), 260MiB/s-260MiB/s (273MB/s-273MB/s), io=25.4GiB (27.3GB), run=100045-100045msec

Disk stats (read/write):
vdb: ios=42/207998, merge=0/0, ticks=21/3173469, in queue=3174831, util=99.95%
```

- bs = 4k iodepth = 32：随机读/写测试，能反映硬盘的 IOPS 性能

执行以下命令，测试硬盘的随机读 IOPS。

```
fio -bs=4k -ioengine=libaio -iodepth=32 -direct=1 -rw=randread -time_based -runtime=600 -
refill_buffers -norandommap -randrepeat=0 -group_reporting -name=fio-randread-iops --size=10G -
filename=/dev/vdb
```

执行以下命令，测试硬盘的随机写 IOPS。

```
fio -bs=4k -ioengine=libaio -iodepth=32 -direct=1 -rw=randwrite -time_based -runtime=600 -
refill_buffers -norandommap -randrepeat=0 -group_reporting -name=fio-randwrite-iops --size=10G -
filename=/dev/vdb
```



测试 SSD 云硬盘的随机读 IOPS 性能。如下图所示：

```
[root@VM_16_21_centos ~]# fio -bs=4k -ioengine=libaio -iodepth=32 -direct=1 -rw=randread -time_based
-runtime=300 -refill_buffers -norandommap -randrepeat=0 -group_reporting -name=fio-randread --siz
e=100G -filename=/dev/vdc
fio-randread: (g=0): rw=randread, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=lib
aio, iodepth=32
fio-3.1
Starting 1 process
Jobs: 1 (f=1): [r(1)][100.0%][r=18.8MiB/s,w=0KiB/s][r=4804,w=0 IOPS][eta 00m:00s]
fio-randread: (groupid=0, jobs=1): err= 0: pid=2689: Tue Jul 16 13:39:33 2019
read: IOPS=4800, BW=18.8MiB/s (19.7MB/s) (5626MiB/300017msec)
  slat (usec): min=2, max=3183, avg= 7.55, stdev=14.34
  clat (usec): min=209, max=777668, avg=6656.04, stdev=45385.49
    lat (usec): min=371, max=777673, avg=6664.08, stdev=45385.46
  clat percentiles (usec):
    | 1.00th=[ 635], 5.00th=[ 832], 10.00th=[ 938], 20.00th=[ 1090],
    | 30.00th=[ 1237], 40.00th=[ 1352], 50.00th=[ 1467], 60.00th=[ 1582],
    | 70.00th=[ 1713], 80.00th=[ 1991], 90.00th=[ 9372], 95.00th=[ 19006],
    | 99.00th=[ 38536], 99.50th=[341836], 99.90th=[734004], 99.95th=[750781],
    | 99.99th=[767558]
  bw ( KiB/s): min= 256, max=38424, per=100.00%, avg=19202.16, stdev=13626.10, samples=600
  iops        : min= 64, max= 9606, avg=4800.52, stdev=3406.52, samples=600
  lat (usec)  : 250=0.01%, 500=0.12%, 750=2.64%, 1000=10.86%
  lat (msec)  : 2=66.45%, 4=5.87%, 10=4.59%, 20=6.49%, 50=2.32%
  lat (msec)  : 100=0.01%, 500=0.36%, 750=0.26%, 1000=0.05%
  cpu         : usr=1.38%, sys=5.00%, ctx=233328, majf=0, minf=63
  IO depths   : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=100.0%, >=64=0.0%
    submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
    complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.1%, 64=0.0%, >=64=0.0%
    issued rwts: total=1440148,0,0, short=0,0,0, dropped=0,0,0
    latency   : target=0, window=0, percentile=100.00%, depth=32

Run status group 0 (all jobs):
  READ: bw=18.8MiB/s (19.7MB/s), 18.8MiB/s-18.8MiB/s (19.7MB/s-19.7MB/s), io=5626MiB (5899MB), run=
300017-300017msec

Disk stats (read/write):
  vdc: ios=1440052/0, merge=0/0, ticks=9478008/0, in_queue=9485217, util=99.98%
```



# 多块弹性云盘构建RAID组

最近更新时间: 2023-03-13 17:54:11

## 1.RAID 简介

RAID 可以将多个磁盘组合起来构成一个磁盘阵列组，提高数据的读写性能和可靠性。操作系统只会将磁盘阵列组当作一个硬盘来使用。目前 RAID 有多种等级，根据选择版本不同，磁盘阵列组相较于一块容量相当的大硬盘有增强数据集成度、增强容错功能、增加处理量或容量等优势。

注意：

- 请及时对即将到期的弹性云硬盘进行续费操作，以免由于弹性云硬盘到期被系统强制隔离对 RAID 产生影响。
- 建议创建 RAID 1、RAID 01、RAID 10 时，使用相同大小的分区，减少磁盘空间的浪费。

RAID 0、RAID 1、RAID 01 和 RAID 10之间的相同点与差异点如下表所示：

| RAID 级别 | 简介                                                 | 优/缺点                                                                                                                                                                                                                                                    | 使用场景（建议）                                       |
|---------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| RAID 0  | 存储模式：<br>数据分段存放在不同的磁盘中。<br>虚拟盘大小：<br>阵列中所有盘容量之和。   | <ul style="list-style-type: none"><li>• 优点：<br/>读写可以并行。理论上读写速率可达单个磁盘的 N 倍（N 为组成 RAID 0 的磁盘个数），但实际上受限于文件大小、文件系统大小等多种因素。</li><li>• 缺点：<br/>没有数据冗余，即便只有单个磁盘损坏，在最严重的情况下也有可能所有数据的丢失。</li></ul>                                                               | 对 I/O 性能要求很高，并且已通过其他方式对数据进行备份处理或者不需要进行数据备份的场景。 |
| RAID 1  | 存储模式：<br>数据被镜像存储在多个磁盘中。<br>虚拟盘大小：<br>阵列中容量最小的盘的容量。 | <ul style="list-style-type: none"><li>• 优点：<ul style="list-style-type: none"><li>○ 读取速度快。</li><li>○ 数据可靠性较高，单个磁盘的损坏不会导致数据的不可修复。</li></ul></li><li>• 缺点：<ul style="list-style-type: none"><li>○ 磁盘利用率最低。</li><li>○ 写入速度受限于单个磁盘的写入速度。</li></ul></li></ul> | 对读性能要求较高，并且需要对写入的数据进行备份处理的场景。                  |
| RAID 01 | 先用多个盘构建 RAID 0，再用多个 RAID 0 构建 RAID 1。              | <ul style="list-style-type: none"><li>• 优点：<br/>同时具备 RAID 0 和 RAID 1 的优点。</li><li>• 缺点：<ul style="list-style-type: none"><li>○ 成本相对较高，至少需要使用四块盘。</li><li>○ 单磁盘的损坏会导致同组的磁盘都不可用。</li></ul></li></ul>                                                      | 推荐使用 RAID 10。                                  |
| RAID 10 | 先用多个盘构建 RAID 1，再用多                                 | <ul style="list-style-type: none"><li>• 优点：<br/>同时具备 RAID 0 和 RAID 1 的优点。</li></ul>                                                                                                                                                                     |                                                |



|  |                      |                            |  |
|--|----------------------|----------------------------|--|
|  | 个 RAID 1 构建成 RAID 0。 | • 缺点：<br>成本相对较高，至少需要使用四块盘。 |  |
|--|----------------------|----------------------------|--|

## 2.构建 RAID

说明：

本文以在 CentOS 云服务器中使用四块弹性云硬盘构建 RAID 0 为例。不同操作系统、不同 RAID 级别的操作可能不同，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或 RAID 相关文档。

Linux 内核提供用于管理 RAID 设备的 md 模块，可以直接使用 mdadm 工具来调用 md 模块创建 RAID 0。

```
[root@VM_63_126_centos ~]# fdisk -l | grep /dev/vd | grep Linux | grep -v vda
/dev/vdc1          1          20805      10485688+   83   Linux
/dev/vdc2          20806       27046       3145464    83   Linux
/dev/vdd1          1          20805      10485688+   83   Linux
/dev/vde1          1          20805      10485688+   83   Linux
/dev/vdf1          1          20805      10485688+   83   Linux
[root@VM_63_126_centos ~]#
```

- (1) 以 root 用户登录云服务器。
- (2) 执行以下命令，安装 mdadm。

```
yum install mdadm -y
```



安装成功结果如下图所示：

```
[root@VM_63_126_centos ~]# yum install mdadm -y
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
--> Package mdadm.x86_64 0:3.3.2-5.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch             Version           Repository         Size
=====
Installing:
mdadm                  x86_64           3.3.2-5.el6       os                 345 k
=====
Transaction Summary
=====
Install      1 Package(s)

Total download size: 345 k
Installed size: 800 k
Downloading Packages:
mdadm-3.3.2-5.el6.x86_64.rpm                                | 345 kB    00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : mdadm-3.3.2-5.el6.x86_64                      1/1
  Verifying  : mdadm-3.3.2-5.el6.x86_64                      1/1

Installed:
mdadm.x86_64 0:3.3.2-5.el6

Complete!
```

(3) 执行以下命令，使用 mdadm 创建 RAID 0。

```
mdadm --create /dev/md0 --level=0 --raid-devices=4 /dev/vd[cdef]1
```

创建成功结果如下图所示：

```
[root@VM_63_126_centos ~]# mdadm --create /dev/md0 --level=0 --raid-devices=4 /dev/vd[cdef]1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

(4) 执行以下命令，使用 mkfs 创建文件系统（以使用 EXT3 文件系统为例）。

```
mkfs.ext3 /dev/md0
```

创建成功结果如下图所示：

```
[root@VM_63_126_centos ~]# mkfs.ext3 /dev/md0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=512 blocks
2621440 inodes, 10477056 blocks
523852 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
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

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

This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

(5) 依次执行以下命令，挂载文件系统。

```
mkdir md0/
```

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

```
tree md0
```

挂载成功结果如下图所示：

```
[root@VM_63_126_centos ~]# mount /dev/md0 md0/
[root@VM_63_126_centos ~]# tree md0
md0
|-- lost+found
1 directory, 0 files
```

(6) 执行以下命令，查看文件系统详情，并记录其 UUID（以c5d8a204:c28853ba:3882e9f8:62d078de为例）。如下图所示：

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

```
[root@VM_163_144_centos ~]# mdadm --detail --scan
ARRAY /dev/md0 metadata=1.2 name=VM_163_144_centos:0 UUID=c5d8a204:c28853ba:3882e9f8:62d078de
```

(7) 执行以下命令，编辑 mdadm 配置文件。vi /etc/mdadm.conf (8) 在编辑模式下，输入相关内容。

对于弹性云硬盘，建议输入以下配置：



DEVICE /dev/disk/by-id/virtio-弹性云盘1ID-part1

DEVICE /dev/disk/by-id/virtio-弹性云盘2ID-part1

DEVICE /dev/disk/by-id/virtio-弹性云盘3ID-part1

DEVICE /dev/disk/by-id/virtio-弹性云盘4ID-part1

ARRAY 逻辑设备路径 metadata= UUID=

本文以逻辑设备路径是 /dev/md0，metadata 是1.2为例，则输入：

DEVICE /dev/disk/by-id/virtio-弹性云盘1ID-part1

DEVICE /dev/disk/by-id/virtio-弹性云盘2ID-part1

DEVICE /dev/disk/by-id/virtio-弹性云盘3ID-part1

DEVICE /dev/disk/by-id/virtio-弹性云盘4ID-part1

ARRAY /dev/md0 metadata=1.2 UUID=3c2adec2:14cf1fa7:999c29c5:7d739349

(9) 按 Esc，输入:wq保存并退出。



# 多块弹性云盘构建LVM逻辑卷

最近更新时间: 2023-03-13 17:54:11

## 1.LVM 简介

逻辑卷管理（Logical Volume Manager, LVM）通过在硬盘和分区之上建立一个逻辑层，将磁盘或分区划分为相同大小的 PE（Physical Extents）单元，不同的磁盘或分区可以划归到同一个卷组（VG, Volume Group），在 VG 上可以创建逻辑卷（LV, Logical Volume），在 LV 上可以创建文件系统。

相较于直接使用磁盘分区的方式，LVM 的优势在于弹性调整文件系统的容量：

- 文件系统不再受限于物理磁盘的大小，可以分布在多个磁盘中。例如，您可以申请3块4TB的弹性云硬盘并使用 LVM 创建一个将近12TB的超大文件系统。
- 可以动态调整逻辑卷大小，不需要对磁盘重新分区。当 LVM 卷组的空间无法满足您的需求时，您可以单独申请弹性云硬盘并挂载到相应的云服务器上，然后将其添加到 LVM 卷组中进行扩容操作。

## 2.构建 LVM

说明：

本文以使用3块弹性云硬盘通过 LVM 创建可动态调整大小的文件系统为例。如下图所示：

```
[root@VM_63_126_centos ~]# fdisk -l | grep vd | grep -v vda | grep -v vdb
Disk /dev/vdc: 10.7 GB, 10737418240 bytes
Disk /dev/vdd: 10.7 GB, 10737418240 bytes
Disk /dev/vde: 10.7 GB, 10737418240 bytes
```

### 步骤 1 创建物理卷 PV

(1)以 root 用户 登录 Linux 云服务器。

(2)执行以下命令，创建一个 物理卷（Physical Volume, PV）。

`pvcreate <磁盘路径1> ... <磁盘路径N>`

本文以 /dev/vdc、/dev/vdd 和 /dev/vde 为例，则执行：

`pvcreate /dev/vdc /dev/vdd /dev/vde`

创建成功则如下图所示：

```
[root@VM_63_126_centos ~]# pvcreate /dev/vdc /dev/vdd /dev/vde
Physical volume "/dev/vdc" successfully created
Physical volume "/dev/vdd" successfully created
Physical volume "/dev/vde" successfully created
```

(3)执行以下命令，查看现在系统中的物理卷。

`lvmdiskscan | grep LVM`

```
[root@VM_63_126_centos ~]# lvm diskscan | grep LVM
/dev/vdc [ 10.00 GiB] LVM physical volume
/dev/vdd [ 10.00 GiB] LVM physical volume
/dev/vde [ 10.00 GiB] LVM physical volume
3 LVM physical volume whole disks
0 LVM physical volumes
```

## 步骤 2 创建卷组 VG

执行以下命令，创建 VG。

`vgcreate [-s <指定PE大小>] <卷组名> <物理卷路径>`

本文以创建一个名为“lvm\_demo0”的卷组为例，则执行：

`vgcreate lvm_demo0 /dev/vdc /dev/vdd`

创建成功则如下图所示：

```
[root@VM_63_126_centos ~]# vgcreate lvm_demo0 /dev/vdc /dev/vdd
Volume group "lvm_demo0" successfully created
```

当提示“Volume group “<卷组名>” successfully created”时，表示卷组创建成功。

o 卷组创建完成后，可执行以下命令，向卷组中添加新的物理卷。

`vgextend 卷组名 新物理卷路径`

添加成功则如下图所示：

```
[root@VM_63_126_centos ~]# vgextend lvm_demo0 /dev/vdf
Volume group "lvm_demo0" successfully extended
```

o 卷组创建完成后，可执行 `vgs`、`vgdisplay` 等命令查看当前系统中的卷组信息。如下图所示：

```
[root@VM_63_126_centos ~]# vgs
VG      #PV #LV #SN Attr   VSize  VFree
lvm_demo 3   0   0 wz--n- 29.99g 29.99g
```

## 步骤 3 创建逻辑卷 LV

执行以下命令，创建 LV。

`lvcreate -L <逻辑卷大小> <VG名称>`

本文以创建一个8GB的名为“lv\_0”的逻辑卷为例，则执行：

`lvcreate -L 8G -n lv_0 lvm_demo0`

创建成功则如下图所示：

```
[root@VM_63_126_centos ~]# lvcreate -L 8G -n lv_0 lvm_demo0
Logical volume "lv_0" created
```

说明：



执行 pvs 命令，可查看到此时只有 /dev/vdc 被使用了8GB。如下图所示：

```
[root@VM_63_126_centos ~]# pvs
PV          VG          Fmt  Attr PSize  PFree
/dev/vdc    lvm_demo    lvm2 a--  10.00g  2.00g
/dev/vdd    lvm_demo    lvm2 a--  10.00g 10.00g
/dev/vde    lvm_demo    lvm2 a--  10.00g 10.00g
```

#### 步骤 4 创建并挂载文件系统

(1) 执行以下命令，在创建好的逻辑卷上创建文件系统。

```
mkfs.ext3 /dev/lvm_demo/lv_0
```

(2) 执行以下命令，挂载文件系统。

```
mount /dev/lvm_demo/lv_0 vg0/
```

挂载成功则如下图所示：

```
[root@VM_63_126_centos ~]# mount /dev/lvm_demo/lv_0 vg0/
[root@VM_63_126_centos ~]# mount | grep lvm
/dev/mapper/lvm_demo-lv_0 on /root/vg0 type ext3 (rw)
```

#### 步骤 5 动态扩展逻辑卷及文件系统大小

注意：

仅当 VG 容量有剩余时，LV 容量可动态扩展。扩展 LV 容量后，需一并扩展创建在该 LV 上的文件系统的大小。

(1) 执行以下命令，扩展逻辑卷大小。

```
lvextend [-L +/- <增减容量>] <逻辑卷路径>
```

本文以向逻辑卷“lv\_0”扩展4GB容量为例，则执行：

```
lvextend -L +4G /dev/lvm_demo/lv_0
```

扩展成功则如下图所示：

```
[root@VM_63_126_centos vg0]# lvextend -L +4G /dev/lvm_demo/lv_0
Size of logical volume lvm_demo/lv_0 changed from 8.00 GiB (2048 extents) to 12.00 GiB (3072 extents).
Logical volume lv_0 successfully resized
```

说明：

执行 pvs 命令，可查看到此时 /dev/vdc 已被完全使用，/dev/vdd 被使用了2GB。如下图所示：

```
[root@VM_63_126_centos vg0]# pvs
PV          VG          Fmt  Attr PSize  PFree
/dev/vdc    lvm_demo    lvm2 a--  10.00g  0
/dev/vdd    lvm_demo    lvm2 a--  10.00g  7.99g
/dev/vde    lvm_demo    lvm2 a--  10.00g 10.00g
```

(2) 执行以下命令，扩展文件系统。

```
resize2fs /dev/lvm_demo/lv_0
```

扩展成功则如下图所示：



```
[root@VM_63_126_centos vg0]# resize2fs /dev/lvm_demo/lv_0
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/lvm_demo/lv_0 is mounted on /root/vg0; on-line resizing required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/lvm_demo/lv_0 to 3145728 (4k) blocks.
The filesystem on /dev/lvm_demo/lv_0 is now 3145728 blocks long.

[root@VM_63_126_centos vg0]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       7.9G 1019M  6.5G  14% /
/dev/mapper/lvm_demo-lv_0
                12G   549M   11G   5% /root/vg0
```

扩展成功后，可执行以下命令，查看逻辑卷的容量是否变为12GB。

df -h



# 常见问题

## 一般性问题

最近更新时间: 2023-03-13 18:04:26

问：当CVM销毁时，数据会发生什么情况？

答：由于本地盘的生命周期完全跟随CVM实例，因此当CVM实例销毁时存储于本地盘中的数据将一并被销毁。存储在云硬盘（CBS）上的数据独立于实例的生命周期，您可以自由选择是否需要实例生命周期之外保留 CBS 卷。因此，我们建议您使用 CBS 来存储需要保存较长时间的数据。

问：我应该选择哪种 CBS 类型？

答：在选择硬盘类型前，请先确认您需要用来进行什么样的工作类型数据存储：

- 如果您的硬盘用来存储中小型数据库、Web/App应用、交易型工作负载等常规场景数据，则推荐您使用高性能云硬盘来获得最佳性价比。
- 如果您的硬盘用来存储大型核心数据库、OLTP业务、NoSQL数据库等工作负载较高、对性能要求较高的场景数据，则推荐您使用 SSD 云硬盘来获得最高的性能。

问：在成功申请后，是否能更换云硬盘的类型？

答：现版本支持弹性云硬盘类型从高性能云硬盘升级为SSD云硬盘，您可以对数据进行快照备份后，通过【调整云硬盘类型】将高性能云硬盘升级为SSD云硬盘。

问：在成功申请后，是否能调整云硬盘的容量？

答：云硬盘支持容量调整功能，可扩大但不可缩小容量。



# 云硬盘使用和生命周期问题

最近更新时间: 2023-03-13 18:04:26

问：是否支持多个实例访问同一块云硬盘？

答：虽然您可以将多达 20 块云硬盘挂载到一个实例，但是目前暂不支持多个实例访问同一块云硬盘。

问：能否将多块硬盘逻辑上合并成一块硬盘以获得更好性能？

答：可以。您可以将挂载到CVM实例的多块云硬盘一起条带化，以实现超越单硬盘的更高的性能。详细内容请参考多块弹性云盘构建LVM逻辑卷。

问：卸载云硬盘时会丢数据吗？

答：云硬盘中的数据不因挂载或卸载而发生改变。为了保持数据一致，我们强烈建议：

- 在Windows操作系统下，为了保证数据完整性，建议您暂停对该磁盘的所有文件系统的读写操作，否则未完成读写的数据会丢失。
- 在Linux操作系统下，您需要登录云服务器实例并对该磁盘进行umount命令行操作，命令执行成功后再进入控制台对磁盘进行卸载操作。

问：卸载云硬盘时该盘会丢数据吗？

答：我们强烈建议：在Windows操作系统下，为了保证数据完整性，建议您暂停对该磁盘的所有文件系统的读写操作，否则未完成读写的数据会丢失。在Linux操作系统下，您需要登录云服务器实例并对该磁盘进行umount命令行操作，命令执行成功后再进入控制台对磁盘进行卸载操作。

问：为什么挂载云硬盘时找不到我想挂载的云服务器？

答：请确保您的云服务器实例和磁盘处于同一个地域的同一个可用区，云硬盘只能在同一个地域的同一个可用区内进行挂载。同时确保您的实例没有被释放。

问：在同一台子机上挂载了几块相同大小、相同类型的云硬盘，应该如何区分这些盘与实际申请的cbs的对应关系？

答：对于Linux操作系统，您可以通过执行ls -l /dev/disk/by-id命令查看到弹性云盘与设备名之间的对应关系：

```
[root@VM_63_126_centos ~]# ls -l /dev/disk/by-id/
total 0
lrwxrwxrwx 1 root root 9 Mar  1 17:31 virtio-disk-35t32l8g -> ../../vdf
lrwxrwxrwx 1 root root 9 Mar  1 17:31 virtio-disk-je13nl0g -> ../../vdc
lrwxrwxrwx 1 root root 9 Mar  1 17:31 virtio-disk-jwz43lpg -> ../../vde
lrwxrwxrwx 1 root root 9 Mar  1 17:31 virtio-disk-punhzcju -> ../../vdd
```

对于Windows操作系统，您可以通过执行wmic diskdrive get caption,deviceid,serialnumber或者wmic path win32\_physicalmedia get SerialNumber,Tag命令查看：

```
[root@VM_63_126_centos ~]# ls -l /dev/disk/by-id/
total 0
lrwxrwxrwx 1 root root 9 Mar  1 17:31 virtio-disk-35t32l8g -> ../../vdf
lrwxrwxrwx 1 root root 9 Mar  1 17:31 virtio-disk-je13nl0g -> ../../vdc
lrwxrwxrwx 1 root root 9 Mar  1 17:31 virtio-disk-jwz43lpg -> ../../vde
lrwxrwxrwx 1 root root 9 Mar  1 17:31 virtio-disk-punhzcju -> ../../vdd
```



问：云硬盘使用上有什么注意事项？

答：

- 对于独立申请的云硬盘，在使用fstab配置静态文件系统信息时，文件系统标识应使用文件系统的 UUID 或者 label，防止由于多个云硬盘在同一子机上多次挂载/卸载后导致云盘在子机中的内核名称发生变化。
- 请特别注意，如果云硬盘先于云服务器到期，那么该云硬盘将会在到期时一段时间内被限流、解挂和回收。为防止对您的业务产生影响，请注意及时对其续费。
- 如果云硬盘从云服务器上移除对您的关键业务不会产生严重影响，那么在配置fstab时可以考虑使用nofail选项，防止由于云硬盘从云服务器上移除后导致系统重启时报错。
- 建议您在 Windows 操作系统下使用云盘前先在 diskpart 内执行san policy=OnlineAll操作。
- 从 Windows 操作系统中解挂云硬盘时建议您先中断对该盘的所有读写操作，并执行offline（脱机）操作。

# 快照常见问题

最近更新时间: 2023-03-13 18:04:26

问：是否需要卸载硬盘或中断所有读写才能拍摄快照？

答：否，您可以在硬盘连接和使用期间实时拍摄快照，不会影响您正常的业务。不过，快照只能捕获已写入 CBS 卷的数据，不包含应用程序或操作系统已在内存中缓存的数据。为了确保快照中捕获了所有应用程序的数据，我们建议您先彻底地暂停对硬盘的IO操作后进行快照制作。对于用作系统盘的 CBS 卷，我们建议您先进行关机，以便能拍摄完整的快照。

问：能否读取以前的快照来恢复？

答：每个快照都有一个唯一的全局标识，用户可以根据任何现有的快照回滚或创建新的 CBS 卷。

问：快照回滚是否需要关机？

答：对于已经挂载在子机上的硬盘，回滚时需要关闭子机；对于未挂载的硬盘则可以直接执行回滚操作。

问：制作快照是否会影响硬盘性能？

答：制作快照时根据硬盘写入数据的数据量情况将对硬盘读写性能有较小影响，因此我们推荐您在业务低谷时期进行快照操作。

问：制作快照到快照可用需要多久？

答：快照制作的时间根据您硬盘写入量的情况、底层的读写情况等各种因素变化，较难预测。不过您无需担心，制作快照并不影响您正常使用硬盘。

问：为什么有的快照无法创建镜像？

答：系统盘和数据盘均可以创建快照，但只有系统盘快照可以创建自定义镜像，再使用已创建的镜像创建一个或多个环境相同的实例。

问：销毁云硬盘的时候，关联快照会被删除吗？

答：不会，销毁云硬盘只会导致数据盘快照无法回滚，关联快照不会被删除，如需删除相关快照，请前往控制台或使用API进行删除。



# 硬盘性能常见问题

最近更新时间: 2023-03-13 18:04:26

问：什么是随机 IO？什么是顺序 IO？

答：IO要么是随机的，要么是顺序的。随机 IO 是指访问地址不连续，而是随机分布在磁盘 LUN 的寻址空间内。主要产生随机 IO 的业务包括：OLTP 业务、SQL、即时通信业务等。

顺序 IO 指的是读写操作连续从相邻的地址访问数据，按逻辑块逐个进行。在顺序 IO 访问中，硬盘寻道时间大幅缩短，因为读写磁头基本不需要移动就可以访问下一个块。如数据备份、写日志流水等业务大部分产生的是顺序 IO。

问：什么是IOPS？

答：IOPS (Input/Output Per Second)即每秒的输入输出量(或读写次数)，是衡量硬盘性能的重要指标之一。IOPS 是指单位时间内系统能处理的 I/O 请求数量，一般以每秒处理的 I/O 请求数量为单位，I/O 请求通常为读或写数据操作请求。

传统磁盘本质上一种机械装置，如FC, SAS, SATA磁盘，转速通常为5400/7200/10000/15000 rpm不等。影响磁盘的关键因素是磁盘服务时间，即磁盘完成一个I/O请求所花费的时间，它由寻道时间、旋转延迟和数据传输时间三部分构成。

常见的，一块 7200 转的机械硬盘能提供75-150的IOPS，15000转的机械硬盘能提供175-210的IOPS(具体的数值视访问模式，如顺序、随机，以及IO大小等因素决定) 问：应用程序读写的 IO 大小是否会影响我获得的 IOPS 性能？

答：会。对于给定的资源，您获得的 IOPS 取决应用程序读取和写入操作的 IO 大小。通常在 IO 大小为 256K 等小块读写时，能将硬盘的 IOPS 性能充分应用起来。

问：应用程序读写的 IO 大小是否会影响我获得的吞吐性能？

答：会。对于给定的资源，您获得的吞吐量取决于应用程序读取和写入操作的 IO 大小。通常在 IO 大小为 1M 等大块读写时，能将硬盘的吞吐性能充分应用起来。



# 词汇表

最近更新时间: 2023-03-13 18:04:26

## \*\*\* 弹性云硬盘\*\*

拥有独立生命周期，可在同地域同可用区不同云服务器实例之间自由挂载、卸载的云硬盘（暂不支持同时在多个云服务器实例上挂载），也称为弹性云盘。

## \*\*\* 非弹性云硬盘\*\*

非弹性云盘是指跟云服务器一起创建且生命周期跟随云服务器的云硬盘，不支持弹性挂载。

## \*\*\* 回滚\*\*

程序或数据处理错误，将程序或数据恢复到上一次正确状态的行为。回滚包括程序回滚和数据回滚等类型。

## \*\*\* IOPS\*\*

IOPS (Input/Output Per Second) 即每秒的输入输出量（或读写次数），是衡量硬盘性能的重要指标之一。IOPS 是指单位时间内系统能处理的 I/O 请求数量，一般以每秒处理的 I/O 请求数量为单位，I/O 请求通常为读或写数据操作请求。传统磁盘本质上一种机械装置，如 FC，SAS，SATA 磁盘，转速通常为5400/7200/10000/15000rpm不等。影响磁盘的关键因素是磁盘服务时间，即磁盘完成一个 I/O 请求所花费的时间，它由寻道时间、旋转延迟和数据传输时间三部分构成。通常情况下，单块转速为7200rpm的机械硬盘能提供75 – 150的 IOPS，单块转速为15000rpm的机械硬盘能提供175 – 210的 IOPS。具体的数值视访问模式（如顺序、随机）以及 I/O 大小等因素而定。

## \*\*\* 全量快照\*\*

对磁盘第一次创建的快照，保存磁盘数据的全量。

## \*\*\* 吞吐量\*\*

对网络、设备、端口、虚电路或其他设施，单位时间内成功地传送数据的数量。

## \*\*\* 云硬盘快照\*\*

保存了云硬盘在某一个时间点的副本，可以使用快照将云硬盘还原到快照创建的时间点。

## \*\*\* 增量快照\*\*

在全量快照或上一次增量快照后，以后每次的快照只需备份与前一次相比增加或者被修改的文件，称为增量快照。