



链滴

阿里云服务器 格式化和挂载数据盘到文件夹

作者: [yohyow](#)

原文链接: <https://ld246.com/article/1460352896036>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

步骤-4-格式化和挂载数据盘

如果您在创建实例时选择了数据盘，在登录实例后，系统需要先格式化数据盘，然后挂载数据盘

另外，您还可以根据业务需要，对数据盘进行多分区配置。建议使用系统自带的工具进行分区操作。

注意：云服务器 ECS 仅支持对数据盘进行二次分区，而不支持对系统盘进行二次分区（不管是 Windows 还是 Linux 系统）。如果您强行使用第三方工具对系统盘进行二次分区操作，可能引发未知风险，如系统崩溃、数据丢失等。

本操作适用于非 I/O 优化+SSD云盘 Linux (Redhat、CentO、Debian、Ubuntu) 实例。对于 I/O 优化实例+SSD云盘数据盘分区挂载，建议使用https://ld246.com/forward?goto=http%3A%2F%2Fhelp.aliyun.com%2Fknowledge_detail%2F6689611.html 工具：auto_fdisk_ssd.sh 脚本

使用远程连接工具，输入用户名 root 和密码登录到实例。

运行 fdisk -l 命令查看数据盘。**注意**：在没有分区和格式化数据盘之前，使用 df -h 命令是无法看到数据盘的。在下的示例中，有一个 5 GB 的数据盘需要挂载。如果执行了 fdisk -l 命令后，没有发现 /dev/xvdb，则表示您的实例没有数据盘，因此无需挂载。可以忽略本章。

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

```
Disk /dev/xvda: 42.9 GB, 42949672960 bytes  
255 heads, 63 sectors/track, 5221 cylinders  
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
Logical/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x00078f9c
```

```
Device Boot      Start         End      Blocks  Id System  
/dev/xvda1 *          1         222      222  1 Linux
```

```
Disk /dev/xvdb: 5368 MB, 536870912 bytes  
255 heads, 63 sectors/track, 652 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
Logical/O size (minimum/optimal): 512 bytes / 512 bytes
```

-number">512 bytes

Disk identifier: 0x0000000

</code></pre><p></p>

执行下列命令，对数据盘进行分区。

<pre><code class="hljs nginx">fdisk /dev/xvdb</code></pre>

</code></pre>

<p>根据提示，依次输入 n，p，1，两回车，wq，分区就开始了。</p>

<pre><code class="hljs sql">[root@xxx ~]# fdisk /dev/xvdb</code>

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0x33eb5059.

Changes will remain in memory only, until you decide to write them.

After that, of course, the previous content won't be recoverable.

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

<p>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').</p>

<p>Command (m for hel

Command action

e extended

p primary partition

p

Partition number (1-4): 1

First cylinder (1-652), default ass=1:

Using default value 1

Last cylinder, +cylinders size{K,M,G} (1-652), default 652):

Using default value 652</p>

<p>Command (m for hel

The partition table has been altered!</p>

</code><p><code class="hljs sql">Calling ioctl() to re read partition table.

Syncing disks.

</code></pre><p></p>

执行 `fdisk -l` 命令，查看新的分区。新分区 `xvdb1` 已经创建好如下面示例中的 `/dev/xvdb1`。

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

Disk `/dev/xvda`: 42.9 GB, 42949672960 bytes

255 heads, 63 sectors/track, 5221 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

Logical/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00078f9c

```
Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *          1          222    41940992   83  Linux
```

Disk `/dev/xvdb`: 5368 MB, 5368709120 bytes

255 heads, 63 sectors/track, 652 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

Logical/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x33eb505

```
Device Boot      Start         End      Blocks   Id  Sys
em
/dev/xvdb1          1          52    5237158   83  Linux
```

执行下列命令，对新分区进行格式化。格式化所需时间取决于数据盘大小。您也可自主决定选用其他文件格式，如 `ext14` 等。

```
mkfs.ext3 /dev/xvdb1
```

执行下列命令写入新分区信息。

我的系统是 Ubuntu 12.04 不支持 `barrier`，所以对该系统正确的命令是：

```
echo '/dev/xvdb1 /mnt ext3 default 0 0' >> /etc/fstab
```

```
其他版本可以用 echo '/dev/xvdb1 /mnt ext3 barrier=0 0' >> /etc/fstab
```

完成后，可以使用 `cat /etc/fstab` 命令查看。

如果需要把数据盘单独挂载到某个文件夹，比如单独用来存放网页，可以修改以上命令中的 `/mnt` 部分。

例如：我需要挂载到 `/alidata/datatmp` 下，则命令为

```
echo '/dev/xvdb1 /alidata/datatmp
ext3 defaults 0 0' >> /etc/fstab
顺便下创建软连接示例命令 ln -s /alidata/datatmp /alidata/data 给datatmp创建了一个data软连接。
```

执行 `mount /dev/xvdb1 /mnt` 挂载新分区，然后执行 `df -h` 查看分区。如果出现数据盘信息，说明挂载成功，可以使用新分区了。

```
[root@xxx ~]# mount /dev/xvdb1 /mnt
```

```
[root@xxx ~]# df -h
```

```
Filesystem      Size  Used Avail Use% Mounted on
```

```
/dev/xvda1      40G  1.5G   36G   4% /
```

```
tmpfs           498M    0 498M   0% /dev/shm
```

```
/dev/xvdb1      5.0G  139M  4.6G   3% /alidata/datatmp
```