

How to remount a data volume on Linux after a resize

Contents

I. General.....	2
A. Description.....	2
B. Disclaimer.....	2
II. Attach a data volume to a CentOS VM and prepare the volume for use.....	3
A. Create a 20GB data volume and attach it to the VM.....	3
B. Prepare the 20GB data volume for use.....	4
C. Create a test file in the data volume partition.....	7
III. Resize the data volume and remount it.....	8
A. Resize the data volume.....	8
B. Remount the resized volume/partition.....	9
1. Unmount the partition.....	9
2. Delete the former 20GB partition.....	9
3. Create the new 30GB partition.....	11
4. Remount the partition.....	12
C. Check the test file.....	12

I. General

A. Description

- With Linux operating systems, when resizing an additional volume or disk, few steps are required before it can be fully used. Once these steps are completed, the Linux operating system can see the new size of the volume and access it.
- In this FAQ, we will explain how to remount the resized partition through an example.

We will first:

- o use a VM with CentOS 7
- o create a data volume of 20GB
- o attach the data volume to the VM
- o create a partition on the data volume
- o create a Linux file system on the partition
- o mount the data volume to a new folder: /home/partition (any folder you wish to mount the volume to)
- o create a test file on the partition and add a content in the test file

Then, we will:

- o resize the data volume to 30GB on the GIOCloud portal
- o unmount the partition
- o resize the data volume partition (delete the 20GB partition and create the 30GB partition)
- o scan the file system of the partition
- o resize the file system of the partition
- o remount the partition

And finally, we will test by:

- o checking the size of the new partition
- o checking if the test file is still in the partition
- o checking if the content of the test file is still the same

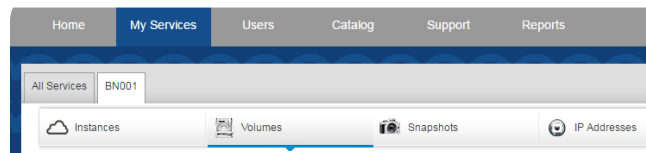
B. Disclaimer

- By resizing a volume, data should not be lost. However, if there was already a corruption in the file system of the partition before the resize, it may result on a data loss after the resize of the file system. Make sure to proceed with great care and note that Leap Solutions Asia will not be held responsible for any data loss resulting from this procedure.
- To check if your partition has any corrupted blocks or errors before any attempt to resize a volume, for CentOS, type: **e2fsck -f -n /dev/sd****
(** Volume letter and Partition number of the data volume)

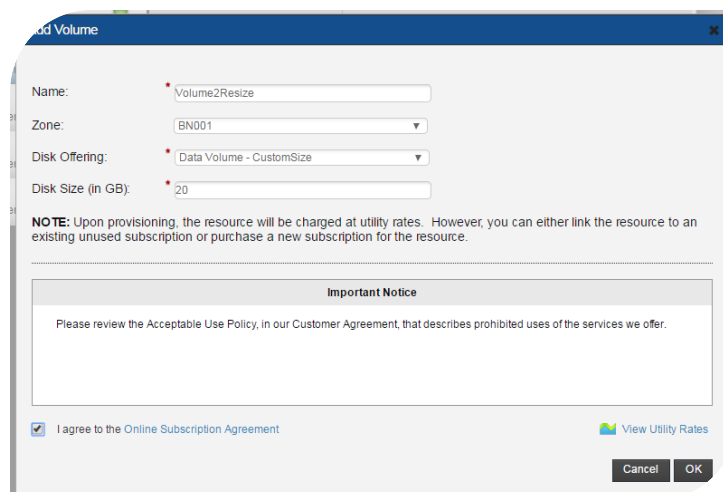
II. Attach a data volume to a CentOS VM and prepare the volume for use

A. Create a 20GB data volume and attach it to the VM

- From the **GIOCloud Portal**, click on the **My Services** tab located on the top horizontal menu
- Click on **Manage Resources**
- Click on the **Volumes** tab



- Create a new volume of 20Gb by clicking on **Add New** and filling up the information, agree on the **Online Subscription Agreement**, and then click on **OK**

Add Volume

Name:

Zone:

Disk Offering:

Disk Size (in GB):

NOTE: Upon provisioning, the resource will be charged at utility rates. However, you can either link the resource to an existing unused subscription or purchase a new subscription for the resource.

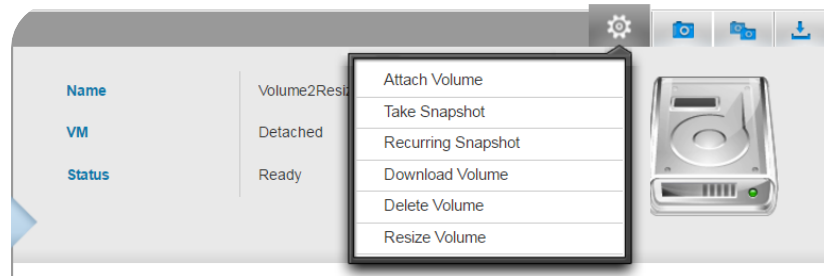
Important Notice

Please review the Acceptable Use Policy, in our Customer Agreement, that describes prohibited uses of the services we offer.

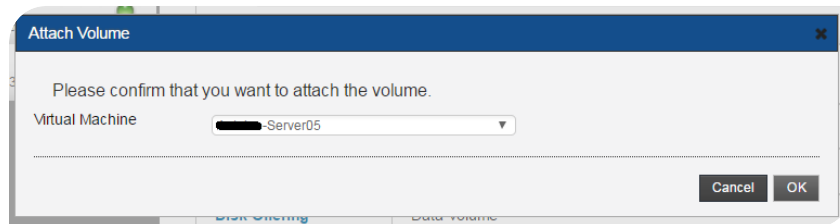
I agree to the Online Subscription Agreement [View Utility Rates](#)

Cancel OK

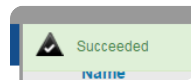
- Attach the new created 20GB data volume to the VM by clicking on the **gear icon** and by selecting **Attach Volume**



- Select your VM and click on **OK**



- The volume is attached to the VM once it says that it has **Succeeded**



B. Prepare the 20GB data volume for use

- Note that we will be using a newly created CentOS 7 VM with the default template provided on Leap GIO Public
- Connect to the VM by SSH or console then type: **ls /dev/sd***
Only one volume is visible, which is the **root volume (sda)** with two partitions (sda1 and sda2)

```
login as: root
Authenticating with public key "rsa-key-20170627"
Last login: Tue Jun 27 15:14:44 2017 from 103.213.207.5
[root@-Server05 ~]# ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2
```

- After a reboot of the VM or a scan, you should be able to see the second volume by typing the same command: **ls /dev/sd***
Two volumes are visible, the root volume (sda) with two partitions (sda1 and sda2) and the second volume (sdb) with no partition because the data volume has never been used

```
[root@xxxxxxxx-Server05 ~]# ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sdb
```

- Create a partition on the data volume by using: **fdisk /dev/sdb**

```
root@xxxxxxxx-Server05 ~]# fdisk /dev/sdb
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): █
```

- Type **n** to add a new partition

```
root@xxxxxxxx-Server05 ~]# fdisk /dev/sdb
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): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): █
```

- Type **p** for a primary partition in our case and by default we take the whole size of the volume so type **Enter** and **Enter**

```
root@xxxxxxxx-Server05 ~]# fdisk /dev/sdb
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): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Using default value 2048
Last sector, +sectors or +size(K,M,G) (2048-41943039, default 41943039):
Using default value 41943039
Partition 1 of type Linux and of size 20 GiB is set

Command (m for help): █
```

- Type **w** to write (save) and exit

```

[root@Server05 ~]# fdisk /dev/sdb
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): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039):
Using default value 41943039
Partition 1 of type Linux and of size 20 GiB is set

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

Calling ioctl() to re-read partition table.
Syncing disks.

```

- If you believe that you have made a mistake, you can type **q** to quit without saving and repeat the above steps to create a partition and always make sure to type **w** to save your change to the disk
- If you type: **ls /dev/sd***
You should be able to see the new created partition on the volume (sdb1)

```

[root@Server05 ~]# ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2  /dev/sdb  /dev/sdb1

```

- We now have a new disk installed, it is visible to CentOS and we have configured a Linux partition on the disk. The next step is to create a Linux file system on the partition so that the operating system can use it to store files and data. In our example, we will use **ext4** file system. Type **mkfs.ext4 /dev/sdb1**

```

[root@Server05 ~]# mkfs.ext4 /dev/sdb1
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
1310720 inodes, 5242624 blocks
262131 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

```

- We now create a directory as a mount point and mount the partition to that directory. Here, we will mount it under **/home/partition**
To create the new directory, type: **mkdir /home/partition**
To mount the partition into that directory, type: **mount /dev/sdb1 /home/partition**

```
[root@Server05 ~]# mkdir /home/partition
[root@Server05 ~]# mount /dev/sdb1 /home/partition
```

- By typing only **mount | grep sdb1**, you can see if the new partition has correctly been mounted (only if a value is returned when typing the command)

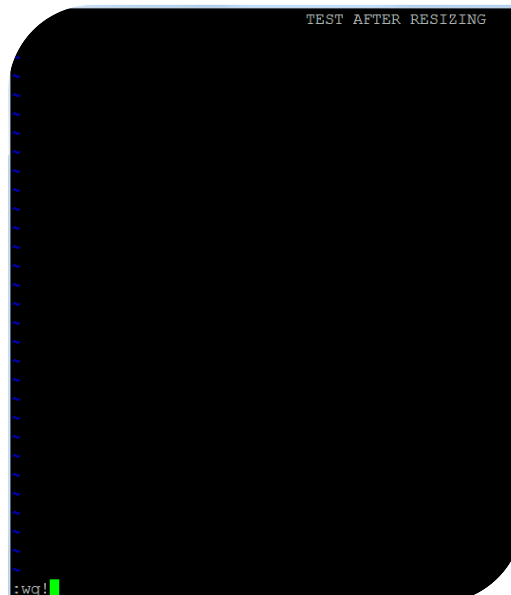
```
[root@Server05 ~]# mount | grep sdb1
/dev/sdb1 on /home/partition type ext4 (rw,relatime,data=ordered)
```

C. Create a test file in the data volume partition

- Create a test file in the **sdb1** partition. Type: **vi /home/partition/test_file**

```
[root@Server05 ~]# vi /home/partition/test_file
```

- We add the below text, then save and exit from **test_file**

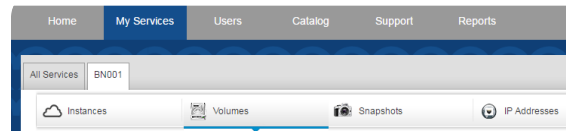


```
TEST AFTER RESIZING
:wq!
```

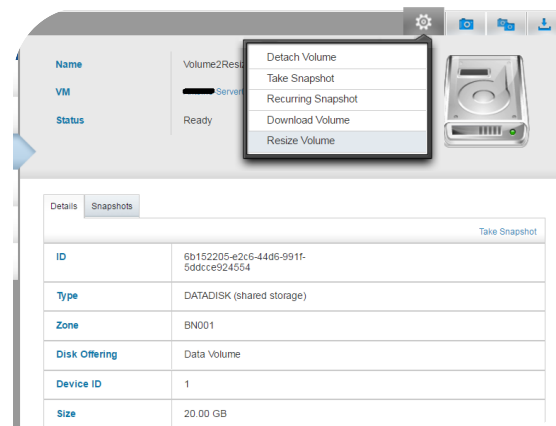
III. Resize the data volume and remount it

A. Resize the data volume

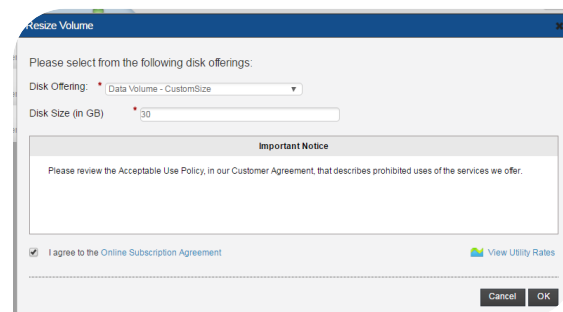
- From the **GIOCloud Portal**, click on the **My Services** tab located on the top horizontal menu
- Click on **Manage Resources**
- Click on the **Volumes** tab



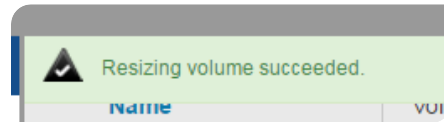
- Select the data volume to resize from the left menu, click on the gear icon, and select



- Select the **disk offering**, input the **new size** of the volume, agree on the **Online Subscription Agreement** and click on **OK**. You cannot downsize a volume. We use 30GB in our example, so we resize from 20GB to 30GB



- Wait for the resize to complete



B. Remount the resized volume/partition

1. Unmount the partition

- Connect to the VM where the resized volume needs to be attached by SSH or console
- Type: **df -h**. We can see here that the size of the partition (sdb1) is still the former size (20GB) and that the partition is still mounted

```
[root@centos7-Server05 ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/cl_centos73-root 47G    1.3G    46G   3% /
devtmpfs                   230M          0   230M   0% /dev
tmpfs                       241M          0   241M   0% /dev/shm
tmpfs                       241M    4.5M   236M   2% /run
tmpfs                       241M          0   241M   0% /sys/fs/cgroup
/dev/sda1                  1014M    184M    831M  19% /boot
/dev/sdb1                   20G      45M     19G   1% /home/partition
tmpfs                       49M          0    49M   0% /run/user/0
```

- Unmount the partition (except if it was already unmounted). Type: **umount /dev/sdb1**. Check with **df -h** and make sure **/dev/sdb1** is not visible anymore

```
[root@centos7-Server05 ~]# umount /dev/sdb1
[root@centos7-Server05 ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/cl_centos73-root 47G    1.3G    46G   3% /
devtmpfs                   230M          0   230M   0% /dev
tmpfs                       241M          0   241M   0% /dev/shm
tmpfs                       241M    4.5M   236M   2% /run
tmpfs                       241M          0   241M   0% /sys/fs/cgroup
/dev/sda1                  1014M    184M    831M  19% /boot
tmpfs                       49M          0    49M   0% /run/user/0
```

2. Delete the former 20GB partition

- Type: **fdisk /dev/sdb** then **p** to see the partition on the data volume. The partition is still the former size (20GB)

```

root@leap-Server05 ~]# fdisk /dev/sdb
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): p

Disk /dev/sdb: 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: 0x7f9aadb5

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1             2048     41943039     20970496   83   Linux

Command (m for help): █

```

- Delete the partition by typing **d** and make sure it is deleted by typing **p** again to check that the volume has no partition. **NOTE:** This will **NOT** delete the data on the data volume/disk/partition

```

Command (m for help): d
Selected partition 1
Partition 1 is deleted

Command (m for help): p

Disk /dev/sdb: 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: 0x7f9aadb5

   Device Boot      Start         End      Blocks   Id  System

```

- Save and exit by typing **w**

```

Disk /dev/sdb: 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: 0x7f9aadb5

   Device Boot      Start         End      Blocks   Id  System

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

Calling ioctl() to re-read partition table.
Syncing disks.

```

3. Create the new 30GB partition

- Type: **fdisk /dev/sdb** then **n** to create a new partition. Type **p** for primary partition, then **Enter**, **Enter**, and **Enter** for default value. You can already see the new size of 30GB

```

root@Server05 ~]# fdisk /dev/sdb
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): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-62914559, default 2048):
Using default value 2048
Last sector, +sectors or +size[K,M,G] (2048-62914559, default 62914559):
Using default value 62914559
Partition 1 of type Linux and of size 30 GiB is set

Command (m for help): █
  
```

- You can double check that the partition is created by typing **p**. Then, type **w** to save and exit

```

Command (m for help): p

Disk /dev/sdb: 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 label type: dos
Disk identifier: 0x7f9aadb5

   Device Boot      Start         End      Blocks   Id  System
 /dev/sdb1          2048     62914559     31456256   83   Linux

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

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

- Check the file system in read-only mode by typing: **e2fsck -f -n /dev/sdb1**. The **-n** option is to run the file system check in read-only mode. Nothing will be changed or repaired if an error is found. **If an error is found during this step, we warn our customers that continuing with the following steps may result in data loss since corrupted blocks/errors will be automatically repaired with the next command.**
- **WARNING**, data loss may occur here if there were any corrupted blocks or errors ---- Check the file system and repair errors/corrupted blocks by typing: **e2fsck -f /dev/sdb1**
- Resize the file system by typing: **resize2fs /dev/sdb1**

```
[root@~ -Server05 ~]# e2fsck -f /dev/sdb1
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/sdb1: 12/1310720 files (0.0% non-contiguous), 126323/5242624 blocks
[root@~ -Server05 ~]# resize2fs /dev/sdb1
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/sdb1 to 7864064 (4k) blocks.
The filesystem on /dev/sdb1 is now 7864064 blocks long.
```

4. Remount the partition

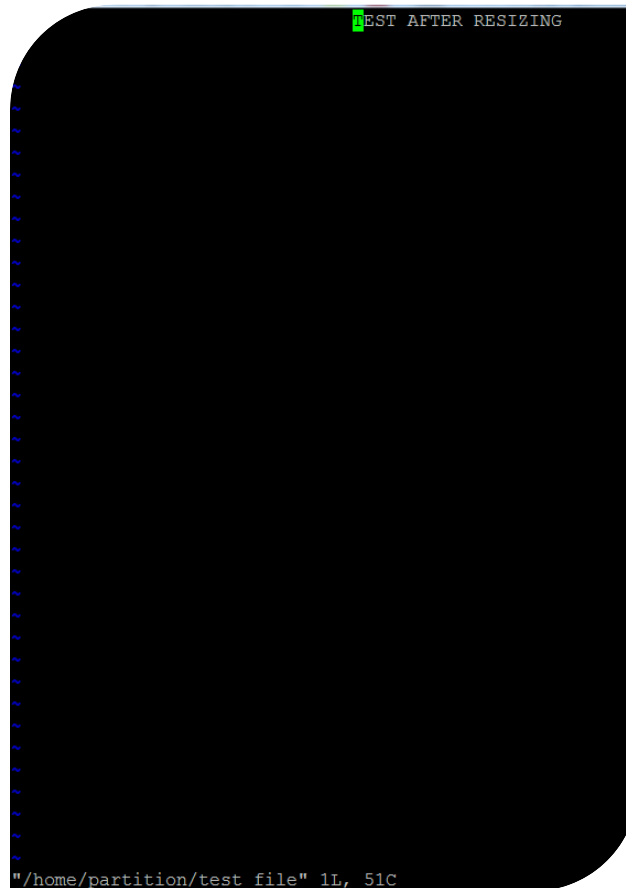
- Remount the partition by typing: **mount /dev/sdb1 /home/partition**
- Check that the partition is mounted and that the new size is correct: **df -h**

```
[root@~ -Server05 ~]# mount /dev/sdb1/ /home/partition
[root@~ -Server05 ~]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/cl_centos73-root 47G    1.3G    46G   3% /
devtmpfs                   230M          0   230M   0% /dev
tmpfs                       241M          0   241M   0% /dev/shm
tmpfs                       241M    4.5M   236M   2% /run
tmpfs                       241M          0   241M   0% /sys/fs/cgroup
/dev/sda1                   1014M    184M    831M  19% /boot
tmpfs                       49M          0    49M   0% /run/user/0
/dev/sdb1                   30G     44M    28G   1% /home/partition
```

C. Check the test file

- To check if the test file is still present, type: **ls -lh /home/partition/**
- To check if the data in the file is still present, type: **vi /home/partition/test_file**
- If everything went well, the file is still present and the data inside has not been deleted

```
[root@-Server05 ~]# ls -lh /home/partition/  
total 20K  
drwx----- 2 root root 16K Jun 28 15:26 lost+found  
-rw-r--r-- 1 root root 51 Jun 28 15:57 test file  
[root@-Server05 ~]# vi test_file
```



If you have any questions please check our FAQ section. If you still cannot find what you are looking for or believe that there is a careless mistake in this document, please contact our support at support@leapsolutions.co.th or send us your inquiry through our [Inquiry Form](#) located on your Web Portal.