
R3

Выпуск 24/07/2022

GNU Linux Pro

июл. 24, 2022

Оглавление

1	Темы	1
1.1	Содержимое	2
1.2	Расширение диска на виртуальной машине	2

1.1 Содержимое

1.1.1 Ссылки

1.2 Расширение диска на виртуальной машине

1.2.1 Расширение диска на виртуальной машине

ОЧЕНЬ ВАЖНЫЙ ДИСКЛЕЙМЕР

Я НЕ НЕСУ НИКАКОЙ ОТВЕТСТВЕННОСТИ ЗА ТО, ЧТО ДЕЛАЕТЕ ВЫ. ВСЁ, ЧТО ВЫ УВИДИТЕ В ЭТОЙ ТЕМЕ - ЧАСТНЫЙ СЛУЧАЙ, КОТОРЫЙ РАБОТАЕТ У МЕНЯ, А У ВАС ЭТИ ЖЕ ДЕЙСТВИЯ МОГУТ ПРИВЕСТИ К ПОЛНОЙ ПОТЕРЕ ДАННЫХ ИЛИ ДРУГИМ УЖАСНЫМ ПОСЛЕДСТВИЯМ. ВСЁ ЧТО ВЫ ДЕЛАЕТЕ - НА ВАШ СТРАХ И РИСК.

Задача

Для начала определимся с задачей: есть виртуалка с линуксом и на ней нужно добавить место на файловой системе.

Разведка

Нужно определить вводные параметры, так как от них зависит всё. Вывод каждой команды обязательно скриншотьте, так как это поможет понять, как было раньше и, возможно, потом это поможет при восстановлении, если что-то пойдёт не так. Каждая инфраструктура в некотором роде уникальна и я не могу разобрать все варианты. Мы возьмём пару популярных случаев, вы поймёте логику и с остальными случаями справитесь сами. И так, начнём с результата и будем двигаться обратно:

Какую файловую систему нужно увеличить?

Ответ на этот вопрос должны знать вы. Зачастую это корневая файловая система, которая забивается данными, какими-нибудь логами и, если вам повезёт, вы успеете заранее добавить пространство. Ну или другой вариант, вам заранее говорят или вы сами предвидите, что в какой-то файловой системе, где лежат данные, в будущем добавится много данных и текущего выделенного пространства перестанет хватать.

Чтобы понять, какую файловую систему мы хотим увеличить, нужно посмотреть список примонтированных файловых систем:

```
df -h
```

```
[user@alma ~]$ df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                   970M         0   970M   0% /dev
tmpfs                      989M         0   989M   0% /dev/shm
tmpfs                      989M      8.5M   981M   1% /run
tmpfs                      989M         0   989M   0% /sys/fs/cgroup
/dev/mapper/almalinux_alma-root 14G       2.0G    12G  15% /
/dev/sda1                 1014M      184M    831M  19% /boot
tmpfs                      198M         0   198M   0% /run/user/0
tmpfs                      198M         0   198M   0% /run/user/1000
[user@alma ~]$
```

Так мы можем найти различные параметры, по которым можно будет найти целевую файловую систему. К примеру, если я хочу увеличить корневую файловую систему, то легче будет понять по точке монтирования - корень, т.е. знак слэш. Если, скажем, я не знаю, куда монтируется, но знаю что осталось мало места - то можно будет разобраться по таким параметрам как Use, Avail и Size, на сколько процентов используется, сколько места свободно и общий размер файловой системы соответственно. Я определил строку по тем или иным параметрам - и в первом столбце этой строки я увижу файловую систему - ответ на мой вопрос:

```
/dev/mapper/almalinux_alma-root
```

Но если вдруг вы так и не нашли, какую файловую систему нужно увеличить, можно понять по директории. К примеру, вы знаете, что на этом сервере база и она хранит все данные в директории /var/lib, и что для этой базы нужно больше места. Найдём файловую систему по этой директории:

```
df -h /var/lib
```

```
[user@alma ~]$ df -h /var/lib/
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/almalinux_alma-root 14G    2.0G    12G   15% /
[user@alma ~]$
```

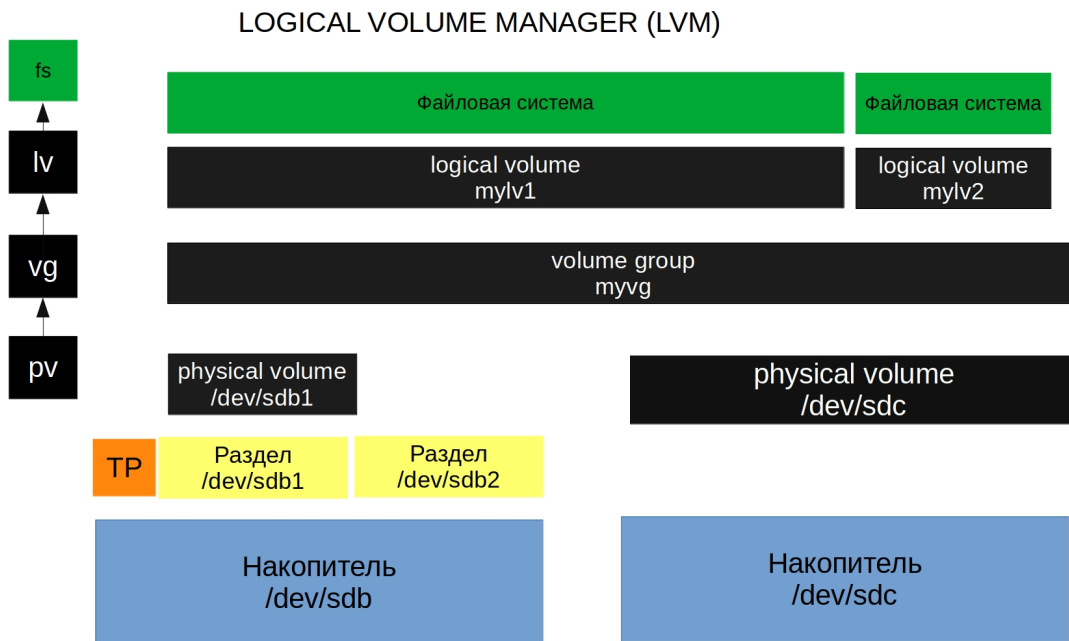
Тут мы видим, что `/var/lib` относится к корневой файловой системе.

И это только вершина айсберга. Следующий вопрос:

Где находится файловая система?

Понятно что на диске, но в современных операционных системах могут быть различные прослойки, с которыми тоже нужно считаться.

К примеру, LVM:



У нас есть диск, на диске есть раздел, на разделе есть физический том, поверх физического тома построена группа томов, на которых строится логический том, поверх которого ставится файловая система. И это ещё очень простой вариант. Если добавить сюда шифрование, рейды и прочее будет целый небоскрёб. Файловая система это всегда последний этаж. И если вы вдруг вздумаете расширить пентхаус, то нужно будет предварительно расширить каждый из нижестоящих этажей, начиная с фундамента.

Возвращаясь к нашему вопросу, давайте не будем гадать и введёт команду:

```
lsblk
```



```
[user@alma ~]$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   16G  0 disk
├─sda1                              8:1    0    1G  0 part /boot
├─sda2                              8:2    0   15G  0 part
│   └─almalinux_alma-root          253:0    0 13.4G  0 lvm /
│       └─almalinux_alma-swap      253:1    0   1.6G  0 lvm [SWAP]
sr0                                  11:0    1 1024M  0 rom
```

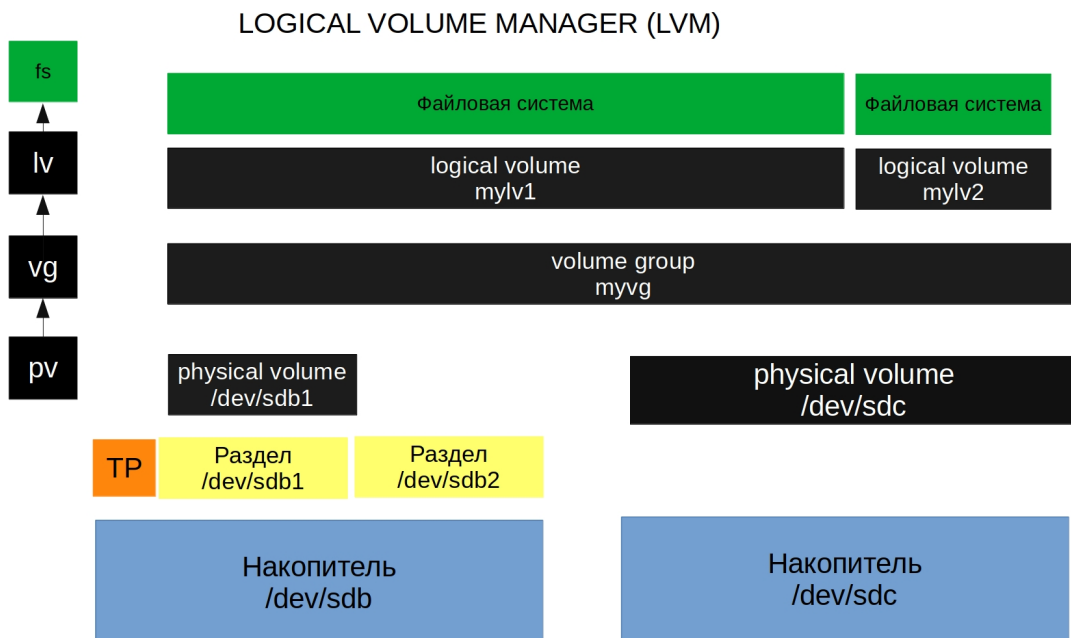
Она поможет и определить небоскрёб, какой этаж на каком стоит, и найти, на чём находится наша файловая система. У нас всё просто - та файловая система, которую мы нашли, находится на LVM. Видно, что LVM ссылается на sda2, который является разделом диска sda.

```
└─stratis-1-private-c648fcd3780341bc9b96171bfe1308be-physical-cachesub 253:10  0   16M  0 stratis
    └─stratis-1-private-c648fcd3780341bc9b96171bfe1308be-physical-cache 253:5    0  508M  0 stratis
        └─stratis-1-private-c648fcd3780341bc9b96171bfe1308be-physical-cache 253:6    0 1020M  0 stratis
            └─stratis-1-private-c648fcd3780341bc9b96171bfe1308be-flex-thinmeta 253:7    0   16M  0 stratis
                └─stratis-1-private-c648fcd3780341bc9b96171bfe1308be-thinpool-pool 253:9    0  972M  0 stratis
                    └─stratis-1-c648fcd3780341bc9b96171bfe1308be-thin-fs-8d18731d410943f7ae60df9a5c1d3cd3 253:11  0    1T  0 stratis
└─stratis-1-private-c648fcd3780341bc9b96171bfe1308be-flex-thindata 253:8    0  972M  0 stratis
    └─stratis-1-private-c648fcd3780341bc9b96171bfe1308be-thinpool-pool 253:9    0  972M  0 stratis
        └─stratis-1-c648fcd3780341bc9b96171bfe1308be-thin-fs-8d18731d410943f7ae60df9a5c1d3cd3 253:11  0    1T  0 stratis
└─stratis-1-private-c648fcd3780341bc9b96171bfe1308be-flex-mdv 253:10  0   16M  0 stratis
sdf                                  8:80    0    2G  0 disk
sr0                                  11:0    1 1024M  0 rom
```

И если вдруг вам показалось, что это просто и в этом любой дурак разберётся, то вспомните тему про stratis. Тот же самый lsblk, но нифига не понятно.

Вспомните математику - главное не ответ, а правильное решение. Поэтому мы пойдём по длинному пути. И так, мы определили, что целевая файловая система находится на LVM. В вашей задаче вместо LVM могут быть другие прослойки. У каждой прослойки свои инструменты для запроса информации, в случае с lvm это lvs, vgs и pvs.

Вспоминаем наш небоскрёб с LVM:



Мы начали с последнего этажа - файловой системы. Зачастую файловая система и нижележащий слой адресуются одинаково, т.е. наш путь `/dev/mapper/almalinux_almalinux-root` это и файловая система, и логический том, на котором она находится. Представьте это как бутылку с водой - нижележащий слой это просто оболочка, внутри которой находятся данные.

Спускаясь ниже, мы должны найти, на какой группе томов находится наш логический том, и из каких физических томов состоит эта группа.

Для первого шага нужна команда `lvs` с логическим томом:

```
sudo lvs /dev/mapper/almalinux_almalinux-root
```

```
[user@alma ~]$ sudo lvs /dev/mapper/almalinux_almalinux-root
LV VG Attr LSize Pool Origin Data%
root almalinux_almalinux -wi-ao---- 13.39g
[user@alma ~]$
```

Тут мы определили, что наш логический том относится к группе `almalinux_almalinux`.

Для следующего шага нужна команда `vgs` с этой группой:

```
sudo vgs almalinux_almalinux
```

```
[user@alma ~]$ sudo vgs almalinux_almalinux
VG #PV #LV #SN Attr VSize VFree
almalinux_almalinux 1 2 0 wz--n- <15.00g 0
[user@alma ~]$
```

Тут мы можем найти информацию, сколько места всего в этой группе и сколько свободно. Если у вас есть достаточно свободного места в группе, то спускаться ниже нет необходимости, так как пространство можно сразу взять отсюда. Однако, скорее всего, свободного места в группе у вас не будет, поэтому спускаемся ниже.

Берём команду pvs:

```
sudo pvs
```

```
[user@alma ~]$ sudo pvs
PV          VG          Fmt  Attr  PSize   PFree
 /dev/sda2  almalinux_alma  lvm2  a--   <15.00g    0
[user@alma ~]$
```

И смотрим, к каким физическим томам относится наша группа. В моём случае это всего один физический том - /dev/sda2, у вас их может быть несколько, даже на разных дисках. В случае с LVM можно увеличить любой, либо добавить ещё один физический том. Но лично я не сторонник создания ещё одних физических томов, дисков и разделов, так как это только усложняет фундамент. Сегодня вы добавите один диск, завтра ещё один и в итоге будет какая-то каша. Лучше увеличить один из физических томов.

Как и в случае с файловой системой и нижележащим логическим томом, также и физический том относится к нижележащему - они делят один адрес, как бутылка с водой. В нашем случае /dev/sda2 это как второй раздел диска sda, так и логический том. Так мы понимаем, что файловая система в нашем случае относится к диску sda, и его мы должны увеличить.

И так, мы добрались до самого фундамента. Но и здесь нужно разведать информацию, так как популярны два вида фундамента - две таблицы разделов - и тут есть свои нюансы.

Для начала выведем информацию о диске с помощью fdisk:

```
sudo fdisk -l /dev/sda
```

```
[user@alma ~]$ sudo fdisk -l /dev/sda
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 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
Disklabel type: dos
Disk identifier: 0xc14aeb0f

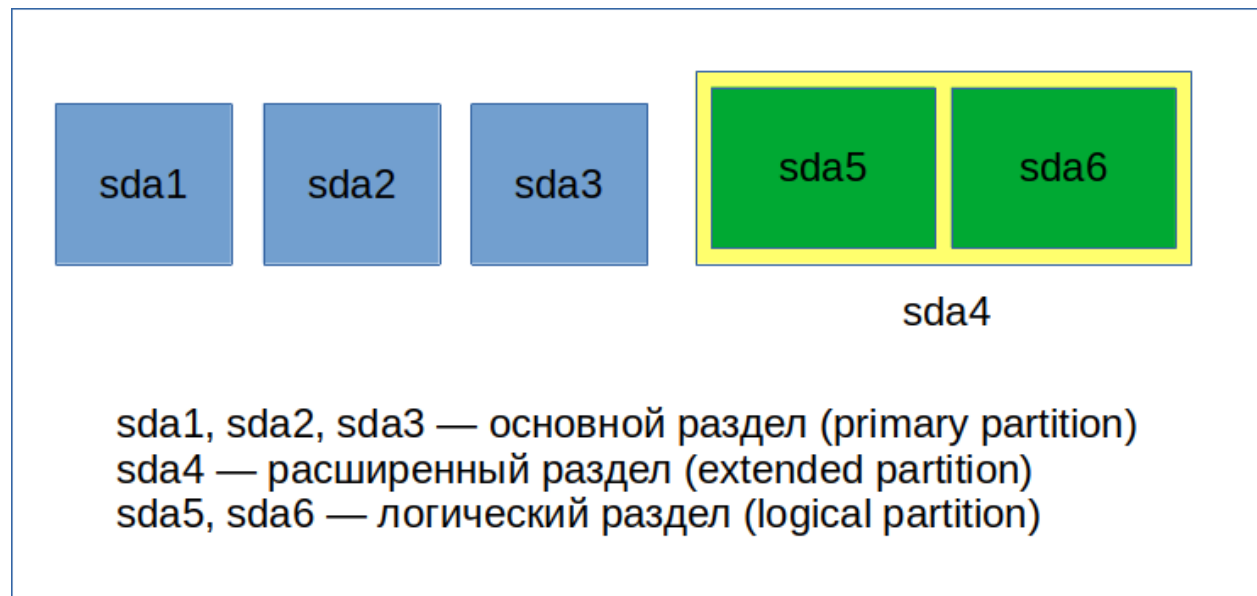
Device      Boot    Start        End    Sectors   Size Id Type
/dev/sda1   *           2048    2099199    2097152    1G 83 Linux
/dev/sda2             2099200  33554431  31455232   15G 8e Linux LVM
[user@alma ~]$
```

Во-первых, очень желательно, чтобы раздел, который мы хотим увеличить, был крайним. Таблицы разделов очень банально устроены - они хранят информацию где раздел начинается и где заканчивается. К примеру, в моём случае увеличить sda2 будет легко - просто поменять значение End на большее. А

вот с sda1 такое не прокатит - сразу после его последнего сектора идёт первый сектор sda2. И сделать, чтобы он закончился в одном месте, а потом продолжился в другом - не получится. И если вдруг была бы задача увеличить sda1, то это было бы довольно сложно, пришлось бы двигать всё содержимое. А если это был бы LVM, то как решение можно было бы создать ещё один раздел и добавить в группу, чего мы не хотели бы делать, но это всяко лучше перемещения раздела. В нашем случае всё просто, но, возможно, в будущем я разберу более проблемные случаи.

Помимо положения раздела также может сыграть роль тип таблицы разделов. Есть два популярных типа - DOS, который часто называют MBR, и GPT. С GPT проблем особо не будет, не считая крайность раздела, а вот с DOS есть ещё один нюанс - деление на основной, расширенный и логический разделы.

Очень многие не знают особенностей таблицы разделов DOS и из-за этого могут по ошибке создать большие проблемы. Поэтому давайте разберёмся.



В таблице разделов DOS может быть всего 4 раздела - их называют основными - primary. Но это ограничение стало создавать проблемы, из-за чего придумали обходной механизм - один из этих разделов использовать как эдакую коробку, внутри которой можно создавать другие разделы. Вот этот раздел-коробка называют расширенным разделом, а разделы внутри него называют логическими разделами. Сам по себе расширенный раздел нельзя использовать как обычный, он почти никакого пространства не занимает и является просто ссылкой на логические. Расширенный раздел не обязательно четвёртый, им может быть и второй, и третий раздел, номер не важен. Но говоря с точки зрения увеличения пространства, если целевая файловая система находится внутри логического раздела - к примеру sda6, то вам сначала надо будет увеличить коробку, а уже потом sda6. Т.е. сначала нужно расширить sda4, а уже потом sda6.

```
[user@alma ~]$ sudo fdisk -l /dev/sda
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 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
Disklabel type: dos
Disk identifier: 0xc14aeb0f

Device      Boot    Start        End    Sectors   Size Id Type
/dev/sda1   *           2048    2099199    2097152    1G 83 Linux
/dev/sda2           2099200  33554431  31455232   15G 8e Linux LVM
[user@alma ~]$
```

Возвращаясь к нашему случаю, у нас хоть и таблица разделов DOS, но всего 2 раздела, оба из них основные, поэтому никаких дополнительных нюансов. Просто увеличиваем sda, а потом sda2.

Но sda - это имя диска в самой системе и оно выдаётся динамически при каждом включении. Если у вас подключен один диск - ничего искать и не надо. А если дисков много? Легче всего будет понять по размеру диска. Если же у вас подключено много дисков с одинаковым размером - тут конечно придётся повозиться и нужна информация как с системы, так и с гипервизора.

С точки зрения гостевой системы информацию о диске можно найти с помощью утилиты lsscsi:

```
lsscsi -i
```

```
[user@alma ~]$ lsscsi -i
[0:0:0:5]    disk      QEMU      QEMU HARDDISK    2.5+  /dev/sda    0QEMU_QEMU_HARDDISK_drive-scsi0-0-0-5
[user@alma ~]$ ls -l /sys/class/block/sda
lrwxrwxrwx. 1 root root 0 Jul 23 18:15 /sys/class/block/sda -> ../../devices/pci0000:00/0000:00:02.3/0000:04:00.0/virtio3/host0/target0:0:0/0:0:0:5/block/sda
[user@alma ~]$
```

Но на каких-то дистрибутивах её может и не стоять по умолчанию, тогда можно посмотреть через sysfs:

```
ls -l /sys/class/block/sda
```

Опять же, тут всё индивидуально и зависит от гипервизора. На примере с QEMU/KVM можно определить по unit номеру:

```
Details XML
<disk type="file" device="disk">
  <driver name="qemu" type="qcow2"/>
  <source file="/home/doctor/vms/alma.qcow2" index="1"/>
  <backingStore/>
  <target dev="sda" bus="scsi"/>
  <alias name="scsi0-0-0-5"/>
  <address type="drive" controller="0" bus="0" target="0" unit="5"/>
</disk>
```

Я специально его поменял, чтобы был не 0 - выставил 5-кой. И в системе эта 5-ка видна в наших командах.

```
[user@alma ~]$ lsscsi -i
[0:0:0:5] disk QEMU QEMU HARDDISK 2.5+ /dev/sda 0QEMU_QEMU_HARDDISK_drive-scsi0-0-0-5
[user@alma ~]$ ls -l /sys/class/block/sda
lrwxrwxrwx. 1 root root 0 Jul 23 18:15 /sys/class/block/sda -> ../../devices/pci0000:00/0000:00:02.3/0000:04:00.0/virtio3/host0/target0:0:0/0:0:0:5/block/sda
[user@alma ~]$
```

Для остальных гипервизоров нужно гуглить другие методы. Но это всё слишком специфичный случай, поэтому это мы опустим.

Окей, диск, который нам нужно увеличить, мы тоже определили. Следующий вопрос:

Кто управляет диском?

А точнее - на каком гипервизоре находится виртуальная машина и где находится нужный диск? От этого зависит как мы будем увеличивать его.

- Понятное дело, у каждого гипервизора свой интерфейс и свои команды
- Некоторые гипервизоры для изменения размера диска требуют выключения виртуалки, к примеру - VirtualBox. Другие, к примеру - ESXi - позволяют это делать на лету
- В некоторых случаях диск для виртуалки может быть выдан по сетевому протоколу, к примеру - iSCSI или Fibre Channel (оптоволоконно). Т.е. есть отдельная система хранения данных, на ней создали диск и его указали в гипервизоре для конкретной виртуалки. Зачастую, гипервизору выделяют большой виртуальный диск и он там сам управляет пространством для различных виртуалок, но также возможно выделить диск напрямую виртуалке. И вот в таком случае гипервизор не может изменить размер диска, это уже нужно делать в системе хранения данных. Но это довольно специфичный случай.

Попробуем ответить на этот вопрос. Я буду использовать QEMU/KVM, так как в виртуалбоксе для изменения диска виртуалку надо выключать. К примеру, если в другом гипервизоре без выключения изменить размер диска или добавить новый диск, то гостевая система может не заметить эти изменения и нужно будет вводить дополнительные команды. Если систему выключить и включить, то эти нюансы отпадут. Нам же предпочтительнее без выключения, чтобы рассмотреть побольше.

В прошлом вопросе мы уже взяли информацию о диске с гипервизора, тут же можем посмотреть, где находится файл этого диска.

```
Details XML
<disk type="file" device="disk">
  <driver name="qemu" type="qcow2"/>
  <source file="/home/doctor/vms/alma.qcow2" index="1"/>
  <backingStore/>
  <target dev="sda" bus="scsi"/>
  <alias name="scsi0-0-0-5"/>
  <address type="drive" controller="0" bus="0" target="0" unit="5"/>
</disk>
```

Именно его нам нужно будет увеличить.

И так, всю необходимую информацию мы собрали. Теперь приступим к следующей стадии.

Разработка

Когда вы планируете какую-то комплексную задачу, чтобы во всём этом не запутаться, можно нарисовать блок-схему. Давайте зафиксируем всю собранную информацию на схеме.

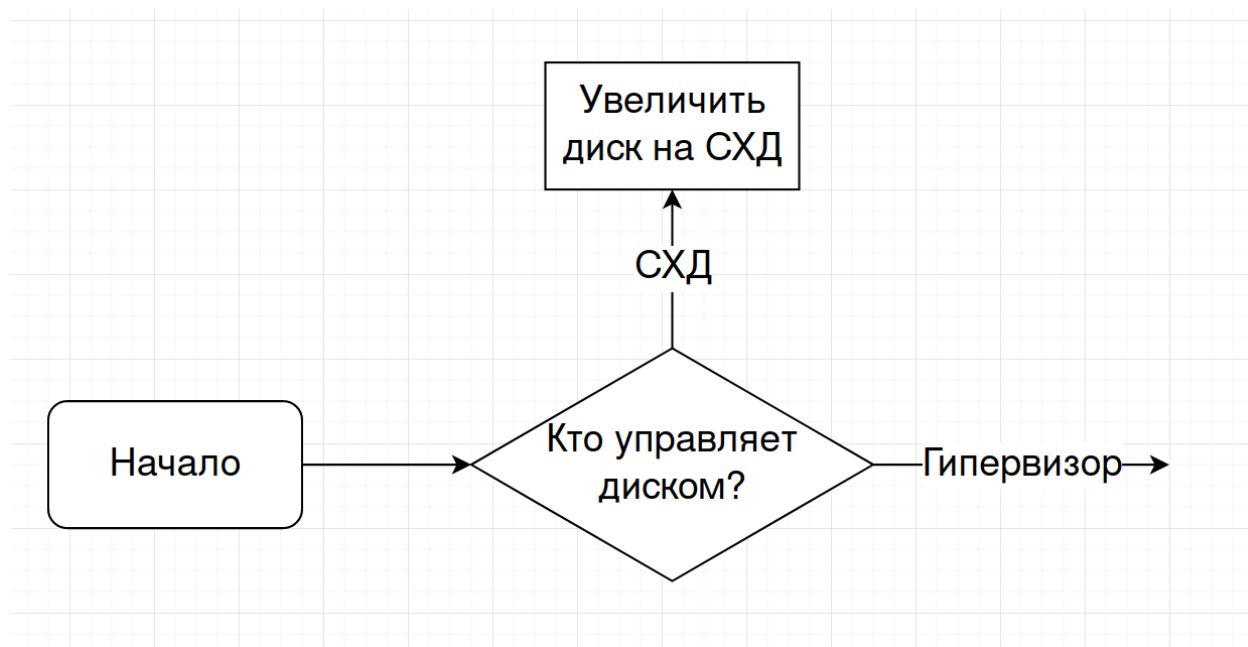
Так как мы задавались вопросами от обратного, т.е. от результата, то теперь переворачиваем порядок вопросов.

Начнём с вопроса:

- Кто управляет диском?



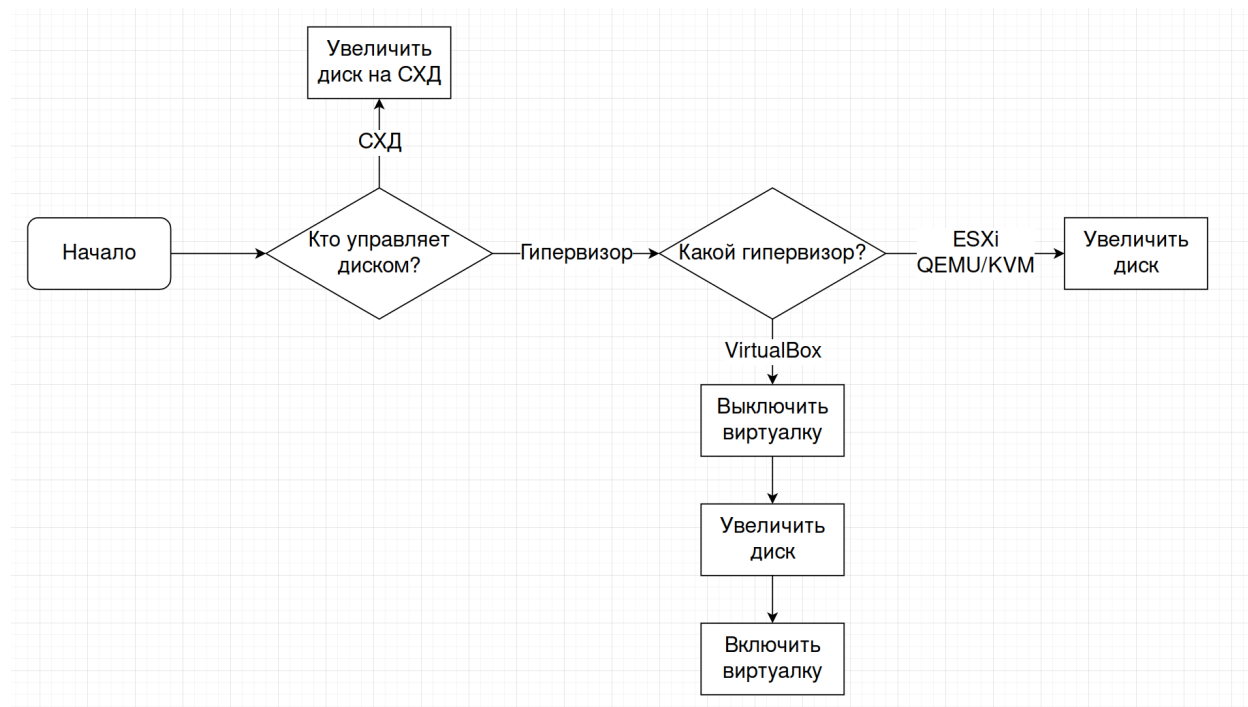
Ответов может быть много и они могут даже не касаться виртуализации. Но рассматривать все варианты в рамках одной темы нереально, поэтому мы предположим два варианта, касающиеся именно виртуализации: гипервизор и система хранения данных.



Ещё раз повторюсь, чтобы не было путаницы - когда я говорю «диск находится на СХД», речь не про общий диск, выделяемый хосту с гипервизором, а отдельный диск, выделяемый конкретной виртуалке. Это очень специфичный случай, но он подойдёт для общего понимания картины.

И так, если диском виртуалки управляет СХД, то увеличиваем его там. Если же диском управляет гипервизор - то нужно задаться ещё одним вопросом:

- Какой гипервизор?

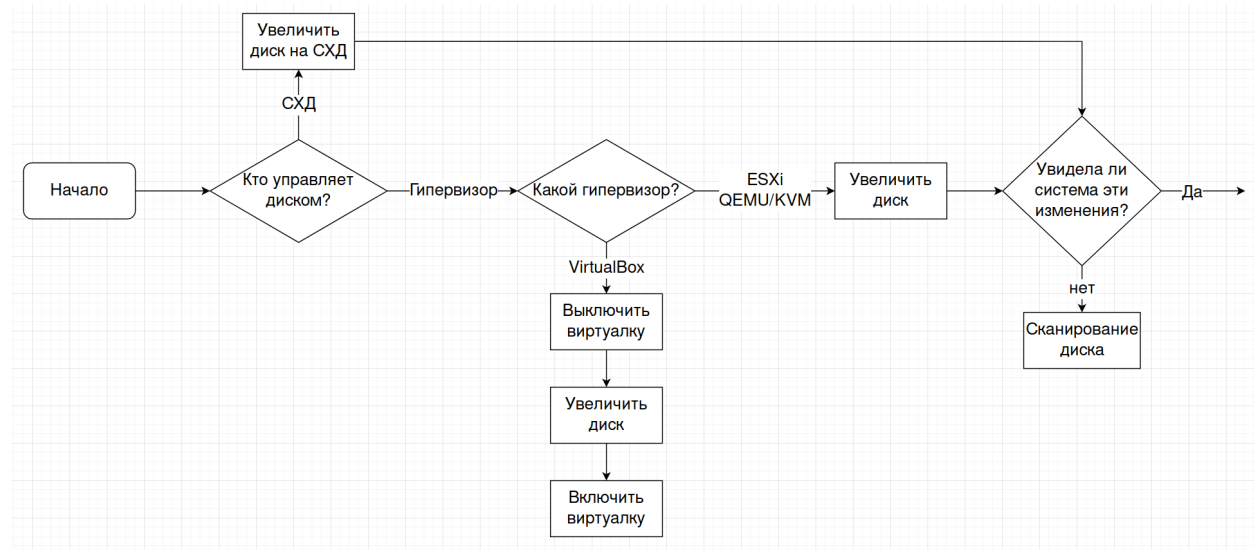


Как мы говорили, некоторые гипервизоры, к примеру - VirtualBox - не позволяют увеличить диск у работающей виртуалки. И в таком случае нужно будет выключить виртуалку, увеличить диск и заново включить виртуалку. Это не проблема для какой-нибудь тестовой или домашней системы. А в рабочей

среде не каждую систему можно безболезненно выключить, некоторые системы могут работать годами без перезагрузки. И в таком случае нужна возможность изменять диск на лету. Для примера, это можно сделать на гипервизорах ESXi или QEMU/KVM - просто на работающей виртуалке увеличиваем диск.

Но при изменении диска на лету нужно задаться ещё одним вопросом:

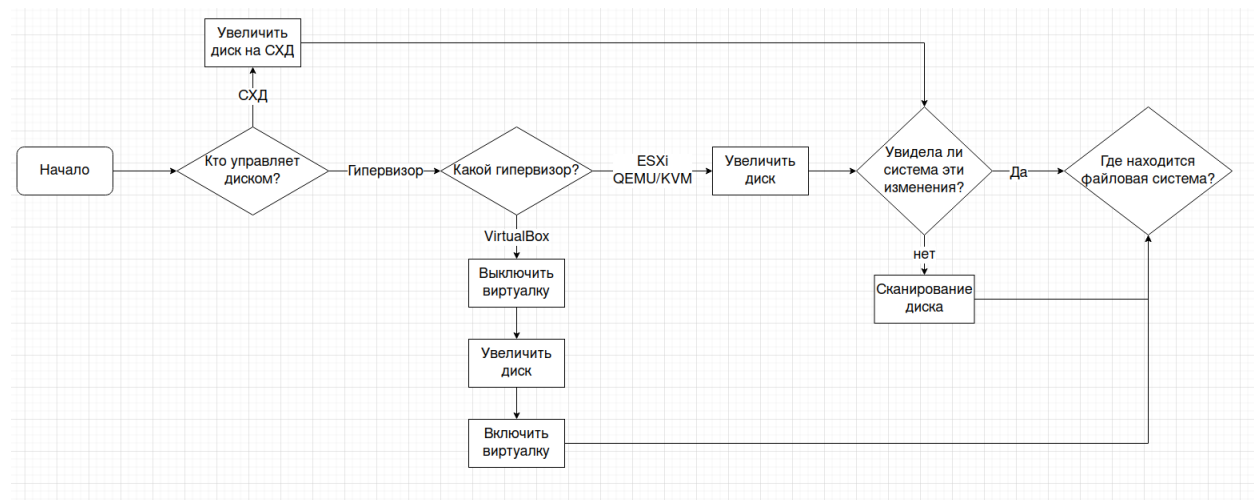
- Увидела ли система эти изменения?



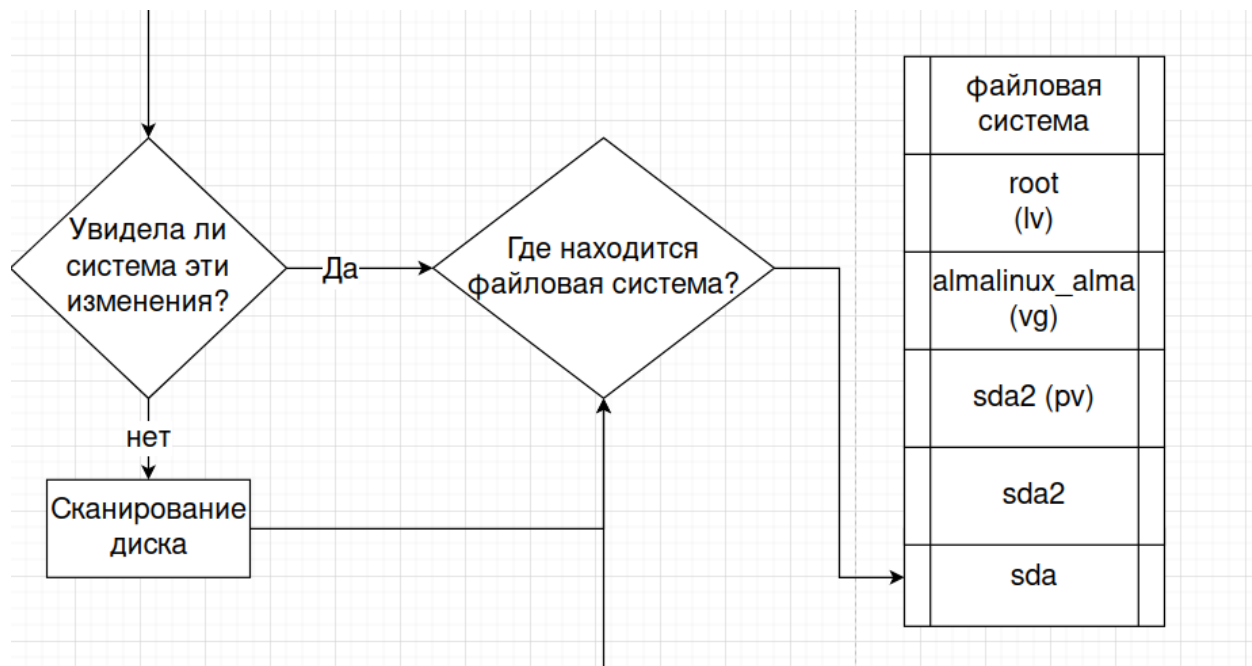
В зависимости от гипервизора и наличия гостевых дополнений, операционная система может увидеть это изменение, а может и не увидеть. Если нет - то нужно дополнительно вручную запустить сканирование диска в гостевой системе.

И после того как система увидит, что диск увеличился, возникает следующий вопрос:

- Где находится файловая система?

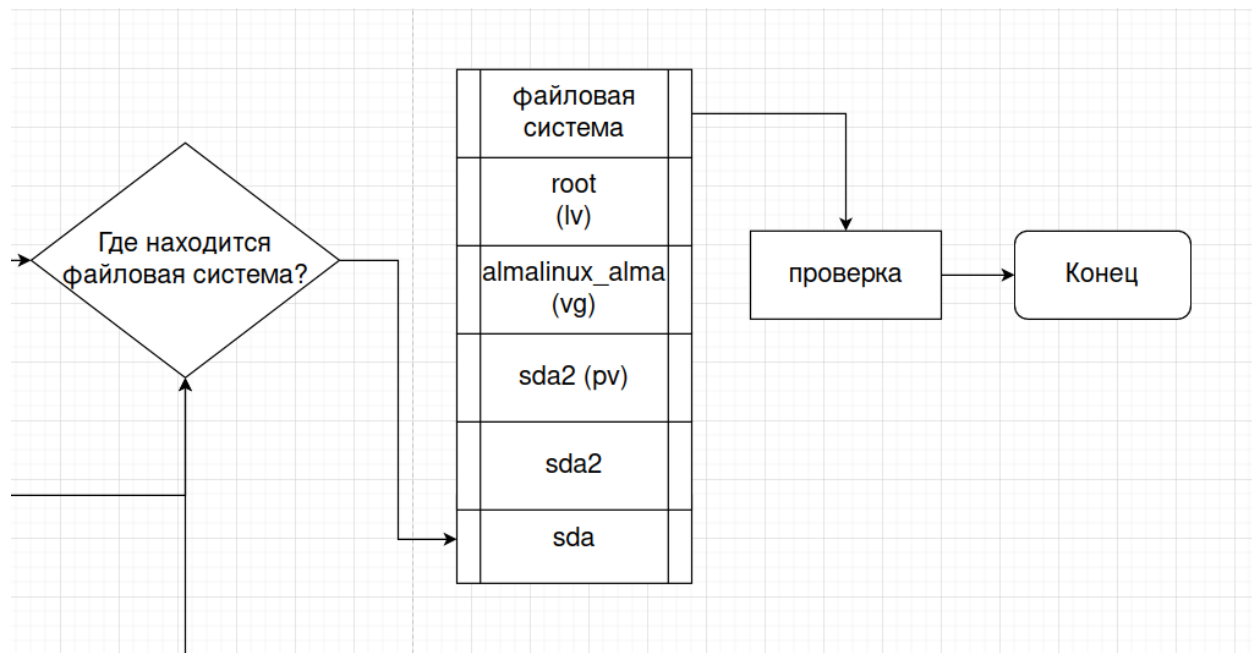


Ответом на этот вопрос будет башня, которую мы изучили.



Как мы и говорили, в каждой системе эта башня может отличаться, у кого-то больше этажей, у кого-то меньше. Но суть одна и та же - нужно поочерёдно увеличить каждый этаж, начиная с фундамента. Все предыдущие шаги были для увеличения диска sda. Увеличение каких-то этажей будет совмещено - скажем, при увеличении физического тома(pv) группа томов тоже увеличивается(vg).

Ну и после того как мы увеличили файловую систему, стоит убедиться, что всё сработало - проверим размер файловой системы.



Теперь зададимся вопросом:

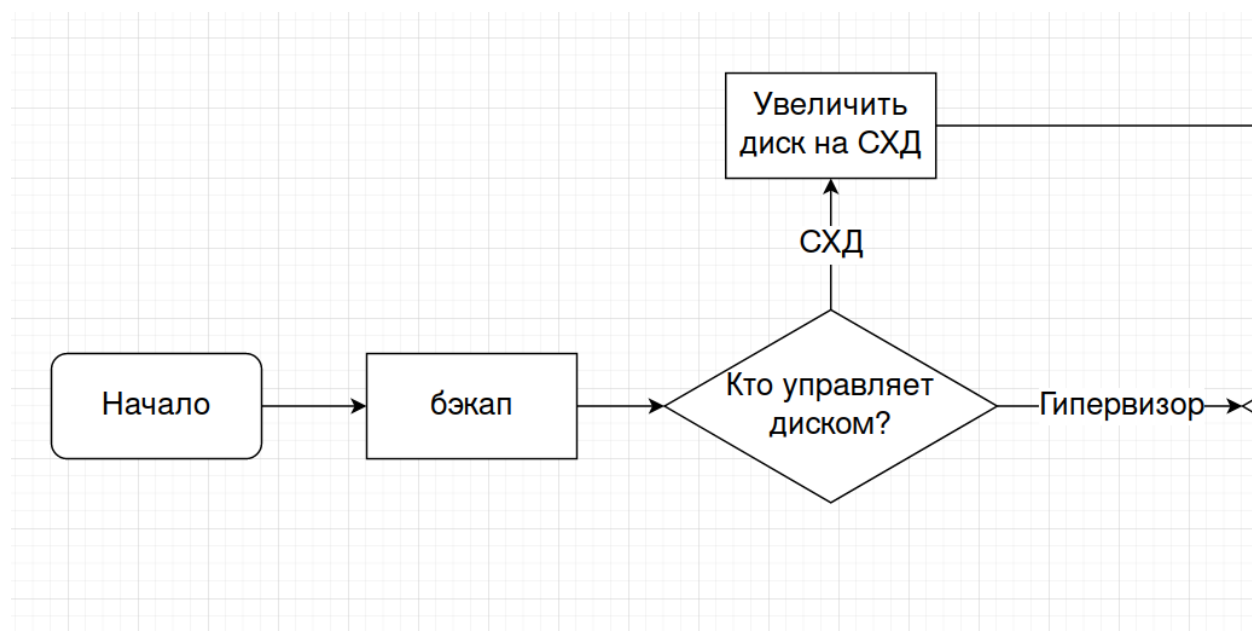
- А какие тут риски?

Ситуаций может быть много, но есть две особые, которые всё перекрывают - можно:

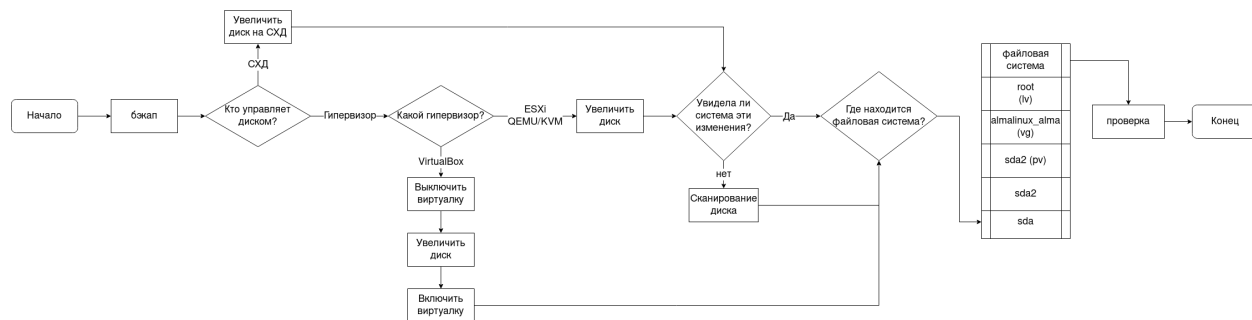
- сломать диск
- сломать файловую систему

К примеру, вместо увеличения диска можно случайно указать диск меньшего размера. Да, обычно есть какая-то защита от дурака, но представим, что не помогло и мы насильно уменьшили диск. Тут уже ничего не поделаешь, единственное спасение - это восстанавливать бэкап. Обычно это очень долго и мучительно, а для компании это может быть очень болезненно. А если бэкапа нет - страшно представить. Так что единственный, хоть и болезненный вариант - бэкап. Поэтому прежде всего убедитесь, что есть бэкап целевой системы. Ну и есть куча различных ошибок, которые можно совершить и поломать разметку диска, файловую систему и всё такое.

Так как риски огромные, то обязательно указываем бэкап в начале нашей схемы. Ну и если случится худший сценарий, сервис будет не работать, пользователи будут доставать вас звонками, пока вы будете возиться с восстановлением бэкапа. Это очень плохо скажется на ваших нервах, поэтому желательно подобные операции проводить в нерабочее время.



Итоговая блок-схема выглядит так:



И так, мы разработали план, мы оценили риски, теперь приступаем к реализации.

Реализация

Начальные шаги будут отличаться на разных системах в зависимости от гипервизора и расположения диска. Будем считать, что вы знаете как на вашем гипервизоре сделать бэкап виртуалки или увеличить диск, я же рассмотрю свой вариант на QEMU/KVM.

Бэкап

И так, в качестве бэкапа копируем диск, о котором мы узнали при разведке:

```
cp -v /home/doctor/vms/alma.qcow2{,.bkp}
```

```
[doctor@tardis]~[~/vms]
$ cp -v /home/doctor/vms/alma.qcow2{,.bkp}
'/home/doctor/vms/alma.qcow2' -> '/home/doctor/vms/alma.qcow2.bkp'
[doctor@tardis]~[~/vms]
$
```

Диск

Дальше увеличиваем диск до нужного размера:

```
virsh blockresize almalinux8 /home/doctor/vms/alma.qcow2 20GB
```

```
[doctor@tardis]~[~/vms]
$ virsh blockresize almalinux8 /home/doctor/vms/alma.qcow2 20GB
Block device '/home/doctor/vms/alma.qcow2' is resized
[doctor@tardis]~[~/vms]
$
```

Проверим, увидела ли система изменения? Для этого сойдёт fdisk:

```
sudo fdisk -l /dev/sda
```

```
[user@alma ~]$ sudo fdisk -l /dev/sda
Disk /dev/sda: 18.6 GiB, 20000000000 bytes, 39062500 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
Disklabel type: dos
Disk identifier: 0xc14aeb0f

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sda1   *           2048    2099199    2097152     1G 83 Linux
/dev/sda2             2099200   33554431   31455232    15G 8e Linux LVM
[user@alma ~]$
```

Сейчас мы видим 18 Гиббайт, а раньше было 16. Значит увеличилась. Кто-то может спросить - вроде до 20 увеличивали, почему же 18? Потому что мы выдали 20 гигабайт, а в fdisk-е видим это в гиббайтах.

Если же ваша система не увидела изменения диска, то, как мы и говорили, нужно просканировать диск на изменения. Это можно сделать с помощью следующей команды:

```
sudo su
echo 1 > /sys/block/sda/device/rescan
```

```
[user@alma ~]$ sudo su
[root@alma user]# echo 1 > /sys/block/sda/device/rescan
[root@alma user]#
```

Эта команда заставит ядро пересканировать информацию о диске и заметить, что, оказывается, у диска другой размер.

И так, фундамент у нас увеличен.

- [] filesystem
- [] root (lv)
- [] almalinux_alma (vg)
- [] sda2 (pv)
- [] sda2
- [x] sda

Раздел

Теперь нужно увеличить раздел sda2.

И так, заходим в fdisk:

```
sudo fdisk /dev/sda
p
```

и смотрим таблицу разделов.

```
[user@alma ~]$ sudo fdisk /dev/sda
[sudo] password for user:

Welcome to fdisk (util-linux 2.32.1).
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/sda: 18.6 GiB, 20000000000 bytes, 39062500 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
Disklabel type: dos
Disk identifier: 0xc14aeb0f

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sda1   *      2048   2099199   2097152    1G 83 Linux
/dev/sda2                2099200 33554431 31455232   15G 8e Linux LVM

Command (m for help): █
```

Как вы помните, тут важно, чтобы раздел был последним. В чём суть увеличения раздела - мы удаляем раздел и создаём заново. Не бойтесь, ничего не поломается если удалить раздел и сразу заново создать с тем же начальным сектором и чтобы конечный сектор был тем же, или больше. Начальный и конечный сектора видны в столбике Start и End. Т.е. пока вы находитесь в fdisk, никакие изменения на диск не пишутся, изменения сохраняются только при выходе.

И ещё раз напомним, что если у вас таблица разделов DOS и есть расширенный раздел, то сначала надо увеличить расширенный раздел, а уже потом логический раздел внутри расширенного.

В нашем случае расширенного раздела нет, поэтому начинаем с удаления второго раздела:

```
d
2
```

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

Partition 2 has been deleted.

Command (m for help): █
```

Теперь нужно заново создать второй раздел:

```

Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2): 2
First sector (2099200-39062499, default 2099200):
Last sector, +sectors or +size{K,M,G,T,P} (2099200-39062499, default 39062499):

Created a new partition 2 of type 'Linux' and of size 17.6 GiB.
Partition #2 contains a LVM2_member signature.

Do you want to remove the signature? [Y]es/[N]o: N

Command (m for help): █

```

```

n
p
2
Enter
Enter
N

```

Обратите внимание, что при создании первый сектор совпадает с тем, что был до удаления. Конечный сектор раньше был 33 миллиона, а теперь стал 39 миллионов, т.е. диапазон увеличился - а значит и раздел стал больше.

fdisk увидел, что на разделе есть метка LVM, и уточнил, стоит ли нам её стереть. Выбираем N, так как мы не хотим затрагивать никакие данные.

Прежде чем сохранить изменения нажимаем

```
p
```

и убеждаемся, что размер второго раздела увеличился. Если вы вдруг сделали что-то не так, размер стал меньше или номер раздела не тот или ещё что-то не так - то нажимаете q, чтобы выйти без сохранения.

```

Command (m for help): p

Disk /dev/sda: 18.6 GiB, 20000000000 bytes, 39062500 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
Disklabel type: dos
Disk identifier: 0xc14aeb0f

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sda1   *           2048    2099199    2097152     1G 83 Linux
/dev/sda2             2099200  39062499   36963300   17.6G 83 Linux

Command (m for help): 

```

Если же всё правильно, пишете w для сохранения изменений.

w

```

Command (m for help): w
The partition table has been altered.
Syncing disks.

[user@alma ~]$
[user@alma ~]$ sudo fdisk -l /dev/sda
[sudo] password for user:
Disk /dev/sda: 18.6 GiB, 20000000000 bytes, 39062500 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
Disklabel type: dos
Disk identifier: 0xc14aeb0f

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sda1   *           2048    2099199    2097152     1G 83 Linux
/dev/sda2             2099200  39062499   36963300   17.6G 83 Linux
[user@alma ~]$ 

```

При этом вы увидите сообщение, что таблица разделов была изменена. Если же выходит какая-либо ошибка, допустим, что не получается изменить таблицу разделов и изменения будут применены при перезагрузке, то, возможно, вы где-то ошиблись. Система не смогла изменить разделы, которые используются. Тут стоит проверить по скриншотам, которые вы обязательно должны были сделать при этапе разведки, а как было раньше. И сравнить как стало сейчас, не получилась ли каша в первых и последних секторах каждого раздела.

В нашем случае никаких ошибок не было, раздел увеличился, значит ставим ещё одну галочку.

- ☐ filesystem

- [] root (lv)
- [] almalinux_alma (vg)
- [] sda2 (pv)
- [x] sda2
- [x] sda

Физический том (pv)

Теперь очередь физического тома.

Посмотрим размер раздела и физического тома на нём:

```
sudo fdisk -l /dev/sda2
sudo pvs
```

```
[user@alma ~]$ sudo fdisk -l /dev/sda2
Disk /dev/sda2: 17.6 GiB, 18925209600 bytes, 36963300 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
[user@alma ~]$ sudo pvs
  PV          VG          Fmt  Attr  PSize    PFree
  /dev/sda2  almalinux_alma lvm2  a--  <15.00g    0
[user@alma ~]$
```

Как видите, LVM пока не заметила изменений. А значит нам нужно увеличить размер физического раздела с помощью команды pvresize:

```
sudo pvresize /dev/sda2
```

И проверить результат:

```
sudo pvs
```

```
[user@alma ~]$ sudo pvresize /dev/sda2
Physical volume "/dev/sda2" changed
1 physical volume(s) resized or updated / 0 physical volume(s) not resized
[user@alma ~]$ sudo pvs
  PV          VG          Fmt  Attr  PSize    PFree
  /dev/sda2  almalinux_alma lvm2  a--  17.62g  2.62g
[user@alma ~]$
```

Теперь видно, что размер физического тома увеличился.

- [] filesystem
- [] root (lv)
- [] almalinux_alma (vg)
- [x] sda2 (pv)
- [x] sda2

- [x] sda

Группа томов (vg)

Если посмотреть группу томов, то видно, что места стала больше, т.е. с группой томов нам никаких дополнительных операций делать не надо.

```
[user@alma ~]$ sudo vgs
VG                #PV #LV #SN Attr   VSize  VFree
almalinux_alma    1  2  0 wz--n- 17.62g 2.62g
[user@alma ~]$
```

- [] filesystem
- [] root (lv)
- [x] almalinux_alma (vg)
- [x] sda2 (pv)
- [x] sda2
- [x] sda

Логический том (lv)

Дальше нас интересует логический том. Для его увеличения используем команду `lvextend`:

```
sudo lvextend almalinux_alma/root -l +100%FREE -r
```

```
[user@alma ~]$ sudo lvextend almalinux_alma/root -l +100%FREE -r
Size of logical volume almalinux_alma/root changed from 13.39 GiB (3429 extents) to <16.02 GiB (4101 extents).
Logical volume almalinux_alma/root successfully resized.
meta-data=/dev/mapper/almalinux_alma-root isize=512    agcount=4, agsize=877824 blks
         =                               sectsz=512   attr=2, projid32bit=1
         =                               crc=1        finobt=1, sparse=1, rmapbt=0
         =                               reflink=1
data     =                               bsize=4096   blocks=3511296, imaxpct=25
         =                               sunit=0      swidth=0 blks
naming   =version 2                       bsize=4096   ascii-ci=0, ftype=1
log      =internal log                    bsize=4096   blocks=2560, version=2
         =                               sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                             extsz=4096   blocks=0, rtextents=0
data blocks changed from 3511296 to 4199424
[user@alma ~]$
```

Этой командой я выделил всё свободное место в группе томов этому разделу, и, как видно по выводу, его размер увеличился почти на 3 гигабайта. Тут же я использовал ключ `-r` для увеличения файловой системы на этом логическом томе.

- [] filesystem
- [x] root (lv)
- [x] almalinux_alma (vg)
- [x] sda2 (pv)

- [x] sda2
- [x] sda

Файловая система

Если же у вас была другая башня, скажем, без LVM, то вам пришлось бы увеличивать файловую систему самостоятельно. В зависимости от типа файловой системы, для ext4 команда была бы одной:

```
sudo resize2fs /dev/mapper/almalinux_alma-root
```

а для xfs другой:

```
sudo xfs_growfs /dev/mapper/almalinux_alma-root
```

```
[user@alma ~]$ sudo resize2fs /dev/mapper/almalinux_alma-root
resize2fs 1.45.6 (20-Mar-2020)
resize2fs: Bad magic number in super-block while trying to open /dev/mapper/almalinux_alma-root
Couldn't find valid filesystem superblock.
[user@alma ~]$ sudo xfs_growfs /dev/mapper/almalinux_alma-root
meta-data=/dev/mapper/almalinux_alma-root isize=512    agcount=5, agsize=877824 blks
        =                               sectsz=512   attr=2, projid32bit=1
        =                               crc=1        finobt=1, sparse=1, rmapbt=0
        =                               reflink=1
data      =                               bsize=4096   blocks=4199424, imaxpct=25
        =                               sunit=0      swidth=0 blks
naming    =version 2                       bsize=4096   ascii-ci=0, ftype=1
log       =internal log                   bsize=4096   blocks=2560, version=2
        =                               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none                             extsz=4096   blocks=0, rtextents=0
[user@alma ~]$
```

Т.е. у каждого типа файловой системы свои команды.

Теперь убедимся, что файловая система действительно выросла:

```
df -h /
```

```
[user@alma ~]$ df -h /
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/almalinux_alma-root  17G    2.0G    15G  13% /
[user@alma ~]$
```

Как видите, теперь размер корневой файловой системы больше.

- [x] filesystem
- [x] root (lv)
- [x] almalinux_alma (vg)
- [x] sda2 (pv)
- [x] sda2
- [x] sda

Итоги

В целом, вся эта задача не такая сложная. Но есть отягчающий фактор - ошибка может привести к потере данных. И так как нет людей, которые не ошибаются, всегда делайте бэкап заранее и проверяйте, что бэкапы рабочие.

Если вам показалось, что это долго и сложно, я просто сильно растянул тему, чтобы детальнее объяснить какие-то нюансы. В реальной среде вы наловчитесь и всё это будете делать автоматом за считанные минуты.

ОЧЕНЬ ВАЖНЫЙ ДИСКЛЕЙМЕР

Я НЕ НЕСУ НИКАКОЙ ОТВЕТСТВЕННОСТИ ЗА ТО, ЧТО ДЕЛАЕТЕ ВЫ. ВСЁ, ЧТО ВЫ УВИДИТЕ В ЭТОЙ ТЕМЕ - ЧАСТНЫЙ СЛУЧАЙ, КОТОРЫЙ РАБОТАЕТ У МЕНЯ, А У ВАС ЭТИ ЖЕ ДЕЙСТВИЯ МОГУТ ПРИВЕСТИ К ПОЛНОЙ ПОТЕРЕ ДАННЫХ ИЛИ ДРУГИМ УЖАСНЫМ ПОСЛЕДСТВИЯМ. ВСЁ ЧТО ВЫ ДЕЛАЕТЕ - НА ВАШ СТРАХ И РИСК.