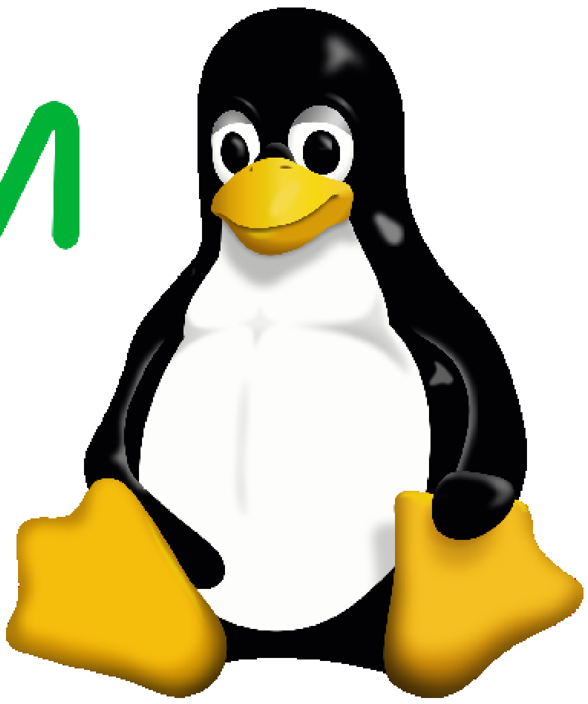
Build an immutable backup repository for Veeam Backup & Replication. Part 3



This guide will show you, step by step, how to create and implement a disk-based immutable Veeam backup repository from scratch. In this part: Prepare the Linux server for Veeam.

Introduction

Purpose of these articles

You are a Windows administrator running Veeam Backup & Replication and wish to raise protection against malware attacks and hackers without reverting to shuffle or rotate physical media.

This you can accomplish by *immutable backups* stored on a physical server running Linux. However, you have no Linux servers running and don't want to.

But, like it or not, that is your only option, as the XFS file system is the only one capable of immutability, and XFS only runs under Linux.

Thus, a Linux server is a must. When you have accepted this fact, then what? Where to start?

Like me, you have about zero experience with Linux and, therefore, hesitate to set up a Linux server, indeed in a production environment.

If so, this guide is for you. Here, nothing about Linux is taken for granted.

Sections

The guide has been split in eight parts. This allows you to skip parts you are either familiar with or wish to implement later if at all.

Requirements

You are familiar with:

- the usual tasks administering at least a small network with one Windows Server
- *Veeam Backup & Replication* and have it installed and running
- the command line - from PowerShell, Command Prompt, or even DOS

> *Veeam Backup & Replication* is assumed to be of *version 11* or later. It can be a licensed trial or paid version or even the free *Community Edition*.

XFS and the virtual air gap

The XFS file system was introduced by SGI in 1993 for its IRIX 5.0 operation system which was based on UNIX System V Release 4.

XFS was ported to Linux in 2001. As SGI ceased operations in 2009, Linux is today the only operating system supporting XFS.

Why is this important? Because XFS is the only file system offering immutability:

> Once the file is set immutable, this file is impervious to change for any user. Even the root cannot modify, remove, overwrite, move or rename the file. You will need to unset the immutable attribute before you can tamper with the file again.

For the details about handling this, study *Dan Nanni*'s blog on Xmodulo: How to make a file immutable on Linux.

Applying immutability to your backup files hosted on a physical server introduces a virtual *air gap* in your backup chain, protecting the backup files from anything else than direct physical access. This way, the backup files will be protected from any attack caused by advanced malware or possible hackers.

The effect is the same as if you back up to tape or DVD and, when done, remove the media from its drive.

Part 3. Prepare the Linux server for Veeam

In this section we will prepare a Linux server running Ubuntu Server (as installed in Part 2) for use with Veeam.

Remote control

First step is to establish a secure connection to the Linux server to be used for remote control via Windows PowerShell on your Windows administrator machine.

So, open PowerShell and enter this command:

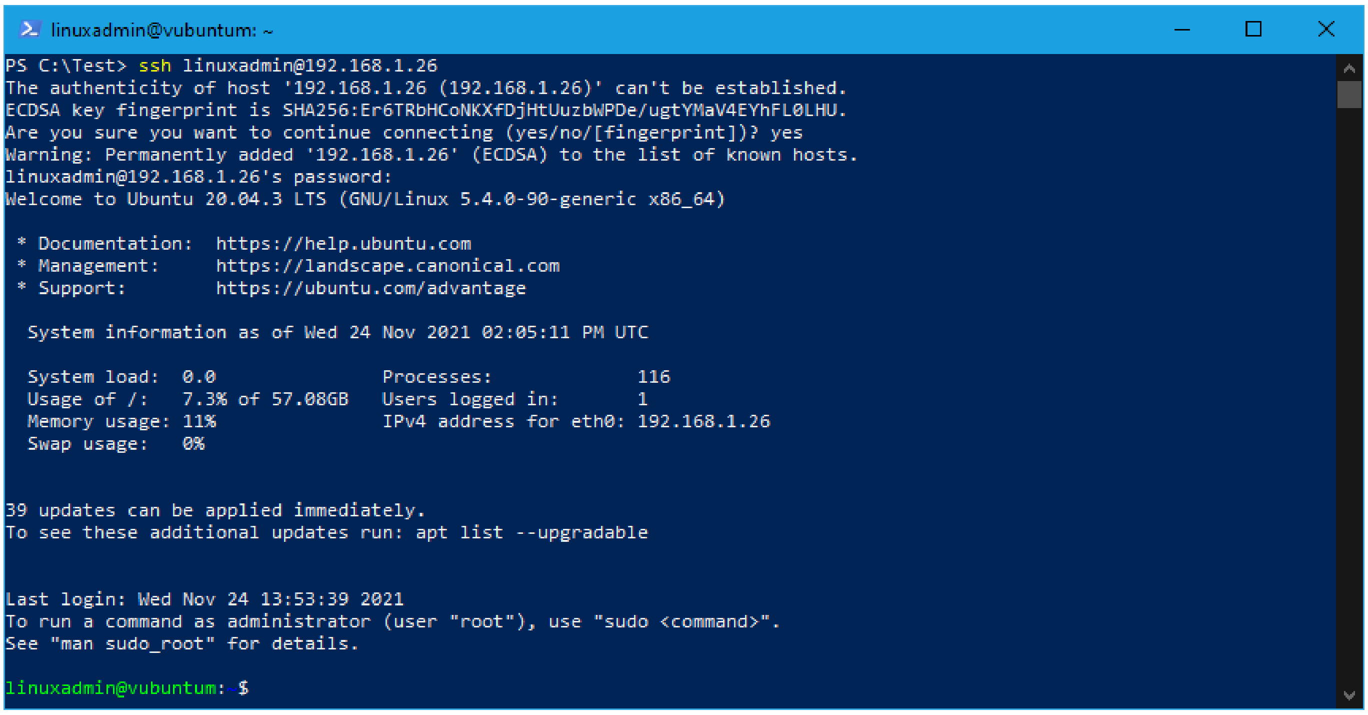
```
ssh linuxadmin@ip-address
```

or, if you have recorded the Linux server in DNS:

```
ssh linuxadmin@hostname
```

The first time, before login, ssh will ask for confirmation of the key to use for the secure connection. Type **yes** (net just **y**) and press Enter and enter the password.

The connection will open, and the welcome message will be displayed:



```
PS C:\Test> ssh linuxadmin@192.168.1.26
The authenticity of host '192.168.1.26 (192.168.1.26)' can't be established.
ECDSA key fingerprint is SHA256:Er6TRbHCoNKXfDjHtUuzbWPDe/ugtYMaV4EYhFL0LHU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.26' (ECDSA) to the list of known hosts.
linuxadmin@192.168.1.26's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-90-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed 24 Nov 2021 02:05:11 PM UTC

  System load:  0.0               Processes:             116
  Usage of /:   7.3% of 57.08GB   Users logged in:       1
  Memory usage: 11%               IPv4 address for eth0: 192.168.1.26
  Swap usage:   0%


39 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Wed Nov 24 13:53:39 2021
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

linuxadmin@vubuntum:~$
```

Updates and basic configuration

As the very first, install updates and upgrades. An upgrade will install both.

To view the list of available upgrades, enter this command:

```
apt list --upgradable
```

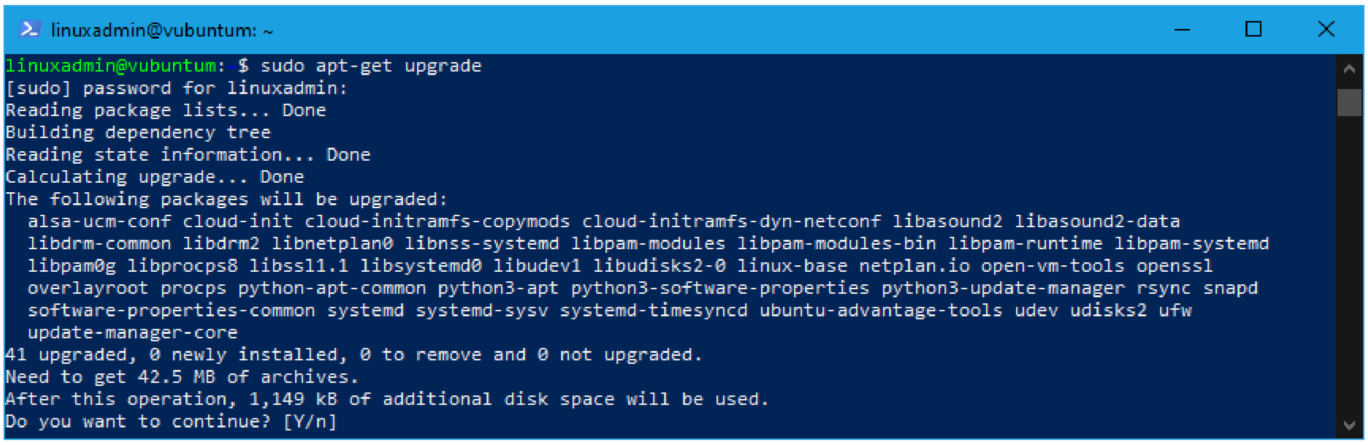
This command is not "dangerous" - it reads only and doesn't change anything. The command to upgrade does, however, thus must be called including the sudo command.

> **sudo**
>
> Where we in Windows have **RunAs**, Linux has **sudo**. Whenever a command needs elevated user rights, precede it with sudo.
>
> When called the first time in a session, it will ask for your password, then use this throughout the session while you are active.

Installing upgrades will modify the system, therefore *sudo* must be used to run the *upgrade* command:

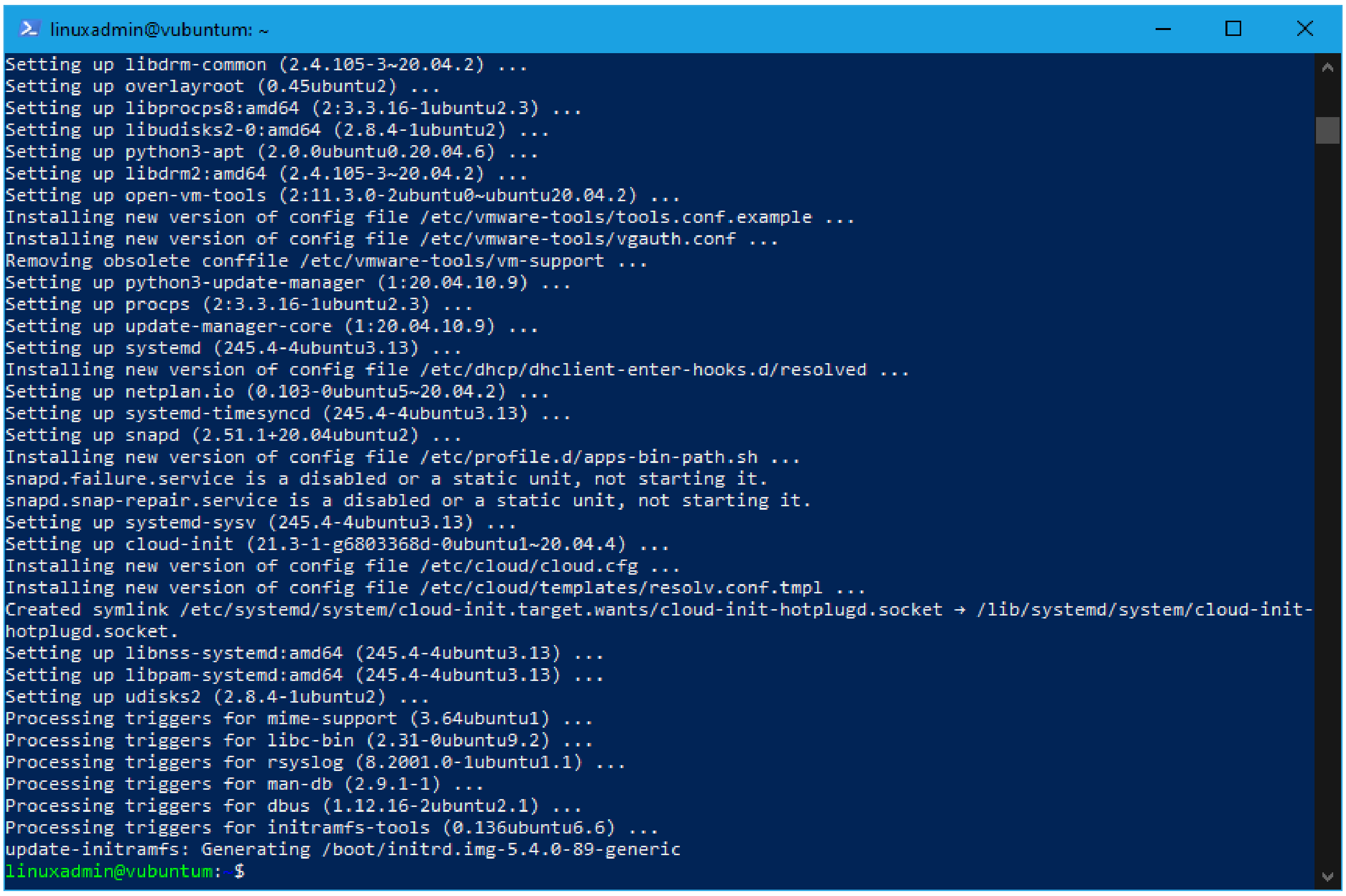
```
sudo apt-get upgrade
```

At first, it will display a compressed list of the potential upgrades and ask for your confirmation:



```
linuxadmin@vubuntum: $ sudo apt-get upgrade
[sudo] password for linuxadmin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  alsa-ucm-conf cloud-init cloud-initramfs-copymods cloud-initramfs-dyn-netconf libasound2 libasound2-data
  libdrm-common libdrm2 libnetplan0 libnss-systemd libpam-modules libpam-modules-bin libpam-runtime libpam-systemd
  libpam0g libprocps8 libssl1.1 libsystemd0 libudev1 libudisks2-0 linux-base netplan.io open-vm-tools openssl
  overlayroot procps python-apt-common python3-apt python3-software-properties python3-update-manager rsync snapd
  software-properties-common systemd systemd-sysv systemd-timesyncd ubuntu-advantage-tools udev udisks2 ufw
  update-manager-core
41 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 42.5 MB of archives.
After this operation, 1,149 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Press **Y** and Enter, and the upgrades will now be installed:



```
Setting up libdrm-common (2.4.105-3~20.04.2) ...
Setting up overlayroot (0.45ubuntu2) ...
Setting up libprocps8:amd64 (2:3.3.16-1ubuntu2.3) ...
Setting up libudisks2-0:amd64 (2.8.4-1ubuntu2) ...
Setting up python3-apt (2.0.0ubuntu0.20.04.6) ...
Setting up libdrm2:amd64 (2.4.105-3~20.04.2) ...
Setting up open-vm-tools (2:11.3.0-2ubuntu0~ubuntu20.04.2) ...
Installing new version of config file /etc/vmware-tools/tools.conf.example ...
Installing new version of config file /etc/vmware-tools/vgauth.conf ...
Removing obsolete conffile /etc/vmware-tools/vm-support ...
Setting up python3-update-manager (1:20.04.10.9) ...
Setting up procps (2:3.3.16-1ubuntu2.3) ...
Setting up update-manager-core (1:20.04.10.9) ...
Setting up systemd (245.4-4ubuntu3.13) ...
Installing new version of config file /etc/dhcp/dhclient-enter-hooks.d/resolved ...
Setting up netplan.io (0.103-0ubuntu5~20.04.2) ...
Setting up systemd-timesyncd (245.4-4ubuntu3.13) ...
Setting up snapd (2.51.1+20.04ubuntu2) ...
Installing new version of config file /etc/profile.d/apps-bin-path.sh ...
snapd.failure.service is a disabled or a static unit, not starting it.
snapd.snap-repair.service is a disabled or a static unit, not starting it.
Setting up systemd-sysv (245.4-4ubuntu3.13) ...
Setting up cloud-init (21.3-1-g6803368d-0ubuntu1~20.04.4) ...
Installing new version of config file /etc/cloud/cloud.cfg ...
Installing new version of config file /etc/cloud/templates/resolv.conf.tmpl ...
Created symlink /etc/systemd/system/cloud-init.target.wants/cloud-init-hotplugd.socket → /lib/systemd/system/cloud-init-
hotplugd.socket.
Setting up libnss-systemd:amd64 (245.4-4ubuntu3.13) ...
Setting up libpam-systemd:amd64 (245.4-4ubuntu3.13) ...
Setting up udisks2 (2.8.4-1ubuntu2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.6) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-89-generic
linuxadmin@vubuntum:~$
```

That finishes the upgrades and updates.

Timezone

By default, the local timezone will be set for timezone UTC. If that is not yours, adjust the timezone.

To view the current setting, run this command:

```
timedatectl
```

That will list the information like this:

```
               Local time: Fri 2021-11-19 10:01:08 UTC
           Universal time: Fri 2021-11-19 10:01:08 UTC
                 RTC time: Fri 2021-11-19 10:01:08
                Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

If you know your timezone, run this command:

```
sudo timedatectl set-timezone CET
```

If you don't, look it up:

```
timedatectl list-timezones
```

However, as Linux has very good information about timezones, the list is overwhelming. Reduce it to those of your continent/region, for example for *Europe*, by passing (that's done with the *pipe* symbol, **|**) the list to the Linux command *grep*, which is a filter command:

```
timedatectl list-timezones | grep Europe
```

> Be very aware, that - contrary to Windows - everything in Linux is *case-sensitive*. Thus, if you in the command above use *europe*, nothing will be listed.

From the returned list, find the region/city closest to you, say, *Europe/Copenhagen*. Now, as Linux knows the timezone of this location, use this in the command to set the local timezone:

```
sudo timedatectl set-timezone Europe/Copenhagen
```

To list the adjusted setting, run again:

```
timedatectl
```

That will now list the information like this:

```
               Local time: Fri 2021-11-19 11:15:51 CET
           Universal time: Fri 2021-11-19 10:15:51 UTC
                 RTC time: Fri 2021-11-19 10:15:51
                Time zone: CET (CET, +0100)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

Veeam Linux user account

To access the repository on the Linux server, a strictly limited user account will be used by *Veeam Backup & Replication*. This is because Veeam has to store the password for the account and, though stored encrypted, it would represent a potential security risk, if the account had administrator rights.

We will name the use *veeamuser*. To create the user, call these two commands:

```
sudo useradd veeamuser --create-home -s /bin/bash
sudo passwd veeamuser
```

You will be prompted for entering the password for the user and to retype this.

Finally, you'll see:

```
passwd: password updated successfully
```

XFS storage

The first physical disk is not intended to be used for anything else than the system. Thus, the second physical disk will be allocated to serve the Veeam backup repository.

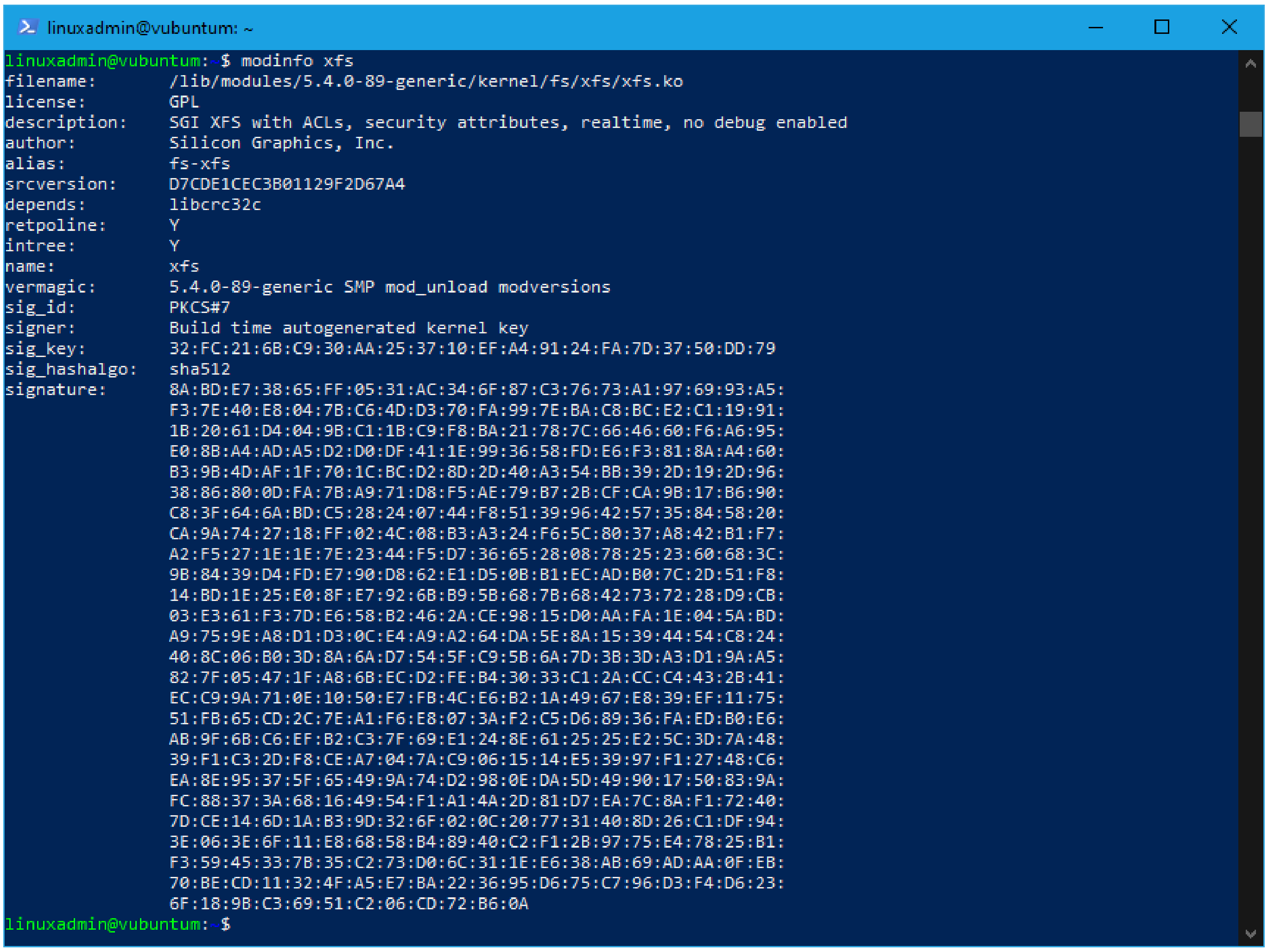
For this, it must be formatted using the XFS file system (read about XFS above).

First, verify that XFS is installed. It will be with Ubuntu, but if you use another distribution, it may not.

Use this command:

```
modinfo xfs
```

It will return an output like this:



linuxadmin@vubuntum: ~

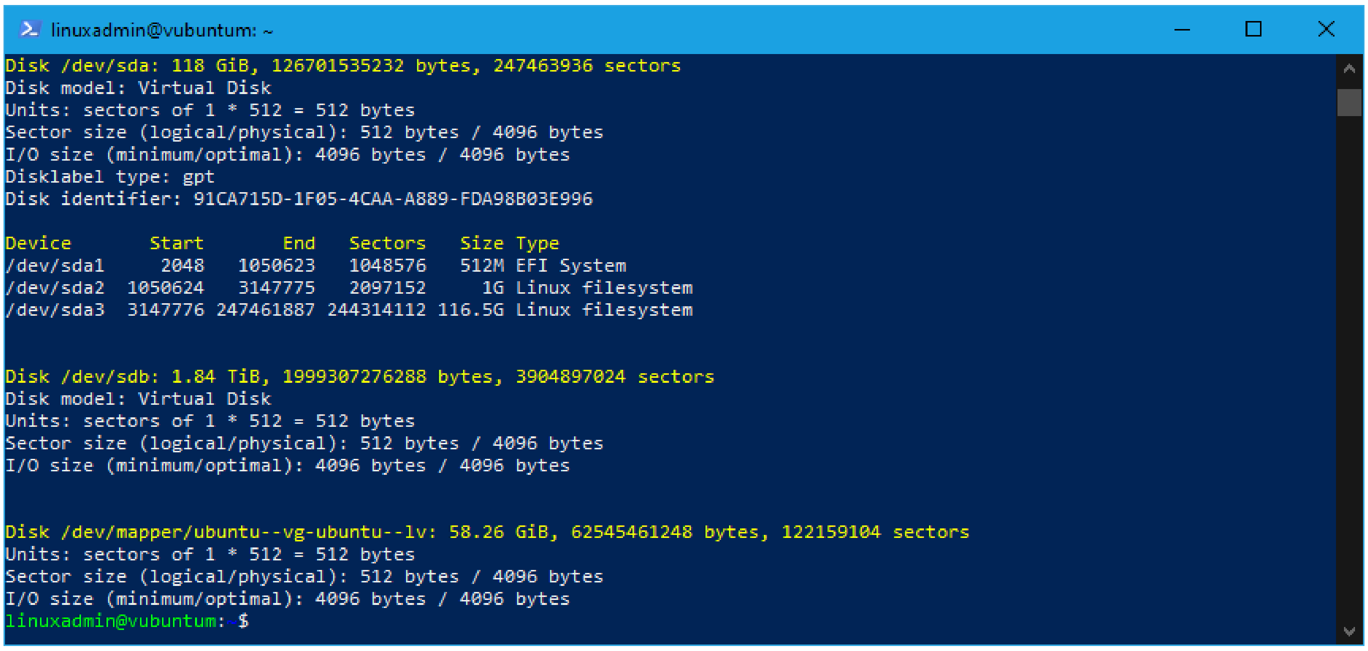
```
linuxadmin@vubuntum:~$ modinfo xfs
filename:       /lib/modules/5.4.0-89-generic/kernel/fs/xfs/xfs.ko
license:        GPL
description:    SGI XFS with ACLs, security attributes, realtime, no debug enabled
author:         Silicon Graphics, Inc.
alias:          fs-xfs
srcversion:     D7CDE1CEC3B01129F2D67A4
depends:        libcrc32c
retpoline:      Y
intree:         Y
name:           xfs
vermagic:       5.4.0-89-generic SMP mod_unload modversions
sig_id:         PKCS#7
signer:         Build time autogenerated kernel key
sig_key:        32:FC:21:6B:C9:30:AA:25:37:10:EF:A4:91:24:FA:7D:37:50:DD:79
sig_hashalgo:   sha512
signature:      8A:BD:E7:38:65:FF:05:31:AC:34:6F:87:C3:76:73:A1:97:69:93:A5:
                F3:7E:40:E8:04:7B:C6:4D:D3:70:FA:99:7E:BA:C8:BC:E2:C1:19:91:
                1B:20:61:D4:04:9B:C1:1B:C9:F8:BA:21:78:7C:66:46:60:F6:A6:95:
                E0:8B:A4:AD:A5:D2:D0:DF:41:1E:99:36:58:FD:E6:F3:81:8A:A4:60:
                B3:9B:4D:AF:1F:70:1C:BC:D2:8D:2D:40:A3:54:BB:39:2D:19:2D:96:
                38:86:80:0D:FA:7B:A9:71:D8:F5:AE:79:B7:2B:CF:CA:9B:17:B6:90:
                C8:3F:64:6A:BD:C5:28:24:07:44:F8:51:39:96:42:57:35:84:58:20:
                CA:9A:74:27:18:FF:02:4C:08:B3:A3:24:F6:5C:80:37:A8:42:B1:F7:
                A2:F5:27:1E:1E:7E:23:44:F5:D7:36:65:28:08:78:25:23:60:68:3C:
                9B:84:39:D4:FD:E7:90:D8:62:E1:D5:0B:B1:EC:AD:B0:7C:2D:51:F8:
                14:BD:1E:25:E0:8F:E7:92:6B:B9:5B:68:7B:68:42:73:72:28:D9:CB:
                03:E3:61:F3:7D:E6:58:B2:46:2A:CE:98:15:D0:AA:FA:1E:04:5A:BD:
                A9:75:9E:A8:D1:D3:0C:E4:A9:A2:64:DA:5E:8A:15:39:44:54:C8:24:
                40:8C:06:B0:3D:8A:6A:D7:54:5F:C9:5B:6A:7D:3B:3D:A3:D1:9A:A5:
                82:7F:05:47:1F:A8:6B:EC:D2:FE:B4:30:33:C1:2A:CC:C4:43:2B:41:
                EC:C9:9A:71:0E:10:50:E7:FB:4C:E6:B2:1A:49:67:E8:39:EF:11:75:
                51:FB:65:CD:2C:7E:A1:F6:E8:07:3A:F2:C5:D6:89:36:FA:ED:B0:E6:
                AB:9F:6B:C6:EF:B2:C3:7F:69:E1:24:8E:61:25:25:E2:5C:3D:7A:48:
                39:F1:C3:2D:F8:CE:A7:04:7A:C9:06:15:14:E5:39:97:F1:27:48:C6:
                EA:8E:95:37:5F:65:49:9A:74:D2:98:0E:DA:5D:49:90:17:50:83:9A:
                FC:88:37:3A:68:16:49:54:F1:A1:4A:2D:81:D7:EA:7C:8A:F1:72:40:
                7D:CE:14:6D:1A:B3:9D:32:6F:02:0C:20:77:31:40:8D:26:C1:DF:94:
                3E:06:3E:6F:11:E8:68:58:B4:89:40:C2:F1:2B:97:75:E4:78:25:B1:
                F3:59:45:33:7B:35:C2:73:D0:6C:31:1E:E6:38:AB:69:AD:AA:0F:EB:
                70:BE:CD:11:32:4F:A5:E7:BA:22:36:95:D6:75:C7:96:D3:F4:D6:23:
                6F:18:9B:C3:69:51:C2:06:CD:72:B6:0A
linuxadmin@vubuntum:~$
```

> If XFS is not installed, install it with these commands:
>
> ```
> sudo apt-get install xfsprogs
> sudo modprobe -v xfs
> ```

Next, list the *disk devices* installed in the machine:

```
sudo fdisk -l
```

Note: The screenshot is from a virtual machine, thus *Disk model* is reported at *Virtual Disk*. For a physical machine, it will be the true disk model number.



```
Disk /dev/sda: 118 GiB, 126701535232 bytes, 247463936 sectors
Disk model: Virtual Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 91CA715D-1F05-4CAA-A889-FDA98B03E996

Device       Start       End   Sectors   Size Type
/dev/sda1     2048   1050623   1048576   512M EFI System
/dev/sda2  1050624   3147775   2097152     1G Linux filesystem
/dev/sda3  3147776 247461887 244314112 116.5G Linux filesystem


Disk /dev/sdb: 1.84 TiB, 1999307276288 bytes, 3904897024 sectors
Disk model: Virtual Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes


Disk /dev/mapper/ubuntu--vg-ubuntu--lv: 58.26 GiB, 62545461248 bytes, 122159104 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
linuxadmin@vubuntum:~$
```

> Harddisks in Linux are named like **/dev/sdx** where **x** is **a** for the first disk, **b** for the second, etc.
>
> Partitions of a disk are named like **/dev/sdxy** where **y** is **1** for the first partition, **2** for the second, etc.
>
> You can also have *logical volumes* (lv). This will, however, not be explained here.

Locate at the bottom of the list the device name of the backup disk. Here it is **/dev/sdb**:

```
Disk /dev/sdb: 1.84 TiB, 1999307276288 bytes, 3904897024 sectors
Disk model: ST2000LM015
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

Write this down or memorise it - that device name shall be used in the following.

Now it is time to format the backup drive using XFS. The basic command is simple:

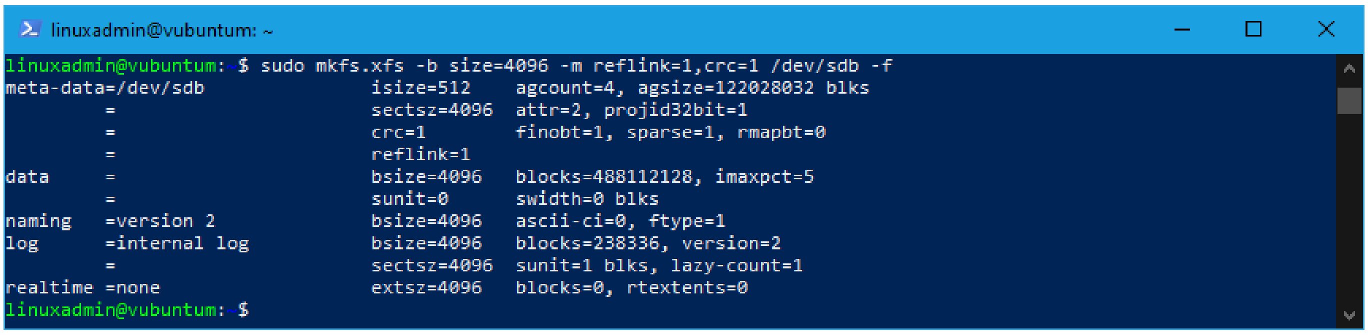
```
sudo mkfs.xfs /dev/sdx
```

However, we need to format the partition with the parameters required by Veeam to leverage *Fast-Clone* technology: *reflink*. Also, *CRC* must be enabled.

Thus, the full command (here for device /dev/sdb) will be:

```
sudo mkfs.xfs -b size=4096 -m reflink=1,crc=1 /dev/sdb -f
```

The output will be similar to:



```
linuxadmin@vubuntum:~$ sudo mkfs.xfs -b size=4096 -m reflink=1,crc=1 /dev/sdb -f
meta-data=/dev/sdb               isize=512    agcount=4, agsize=122028032 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=488112128, imaxpct=5
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=238336, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
linuxadmin@vubuntum:~$
```

We now have a disk formatted with XFS. To expose it to Veeam, it must also be *mounted*. In Windows, it is like bringing a disk online and assigning it a volume.

For a start, we must know the *disk id*. It can be read with this command:

```
sudo blkid /dev/sdb
```

The output will show the id, a GUID string:

```
/dev/sdb: UUID="c84e532c-f519-40de-94f9-21adfcd3cf57c" TYPE="xfs"
```

This must be listed in a config file, */etc/fstab*, to be mounted automatically at boot.

The name of the mount point is not fixed, but we will use the name, that Veeam expects by default:

```
/mnt/veeam
```

This can be specified in two ways:

Manually by calling this command to load the file with the *nano* editor:

```
sudo nano /etc/fstab
```

This will - if you are old enough - bring back memories of Quick Basic, but the positive thing is, that the shortcut keys to interact with the editor are displayed at the bottom.

The two lines to append in the editor will be these (of course, your GUID retrieved above should be used).

The first is a comment line, the second the actual configuration entry (Note: In total, two lines only. No line break in the second line):

```
# /mnt/veeam was on /dev/sdb
/dev/disk/by-uuid/c84e532c-f519-40de-94f9-21adfcd3cf57 /mnt/veeam xfs defaults 0 0
```

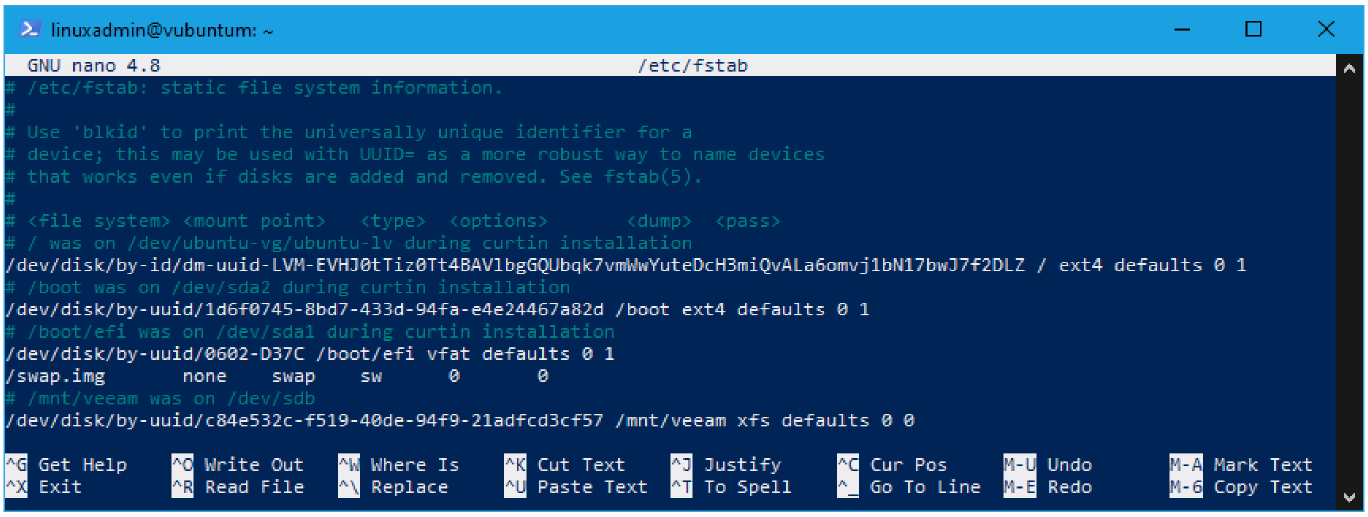
From the command line by calling these two commands that pass the two strings to the utility *tee* (that is similar to *edlin* of DOS) which each will append one line to the */etc/fstab* file (Again: Two lines in total, no line breaks):

```
echo '# /mnt/veeam was on /dev/sdb' | sudo tee -a /etc/fstab
echo '/dev/disk/by-uuid/c84e532c-f519-40de-94f9-21adfcd3cf57 /mnt/veeam xfs
defaults 0 0' \ | sudo tee -a /etc/fstab
```

You may confirm the result by this command:

```
sudo nano /etc/fstab
```

and you'll see:



```
linuxadmin@vubuntum: ~
  GNU nano 4.8                                 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-EVHJ0tTiz0Tt4BAV1bgGQUbqk7vmWwYuteDcH3miQvALa6omvj1bN17bwJ7f2DLZ / ext4 defaults 0 1
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/1d6f0745-8bd7-433d-94fa-e4e24467a82d /boot ext4 defaults 0 1
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/0602-D37C /boot/efi vfat defaults 0 1
/swap.img       none    swap    sw      0       0
# /mnt/veeam was on /dev/sdb
/dev/disk/by-uuid/c84e532c-f519-40de-94f9-21adfcd3cf57 /mnt/veeam xfs defaults 0 0

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text
```

We are now ready to create the mounting point. It is a two-step process.

Step one is to establish the name of the mounting point using the command *mkdir*. This may confuse you, as *mkdir* doesn't just *make a directory* ready to use:

```
sudo mkdir /mnt/veeam
```

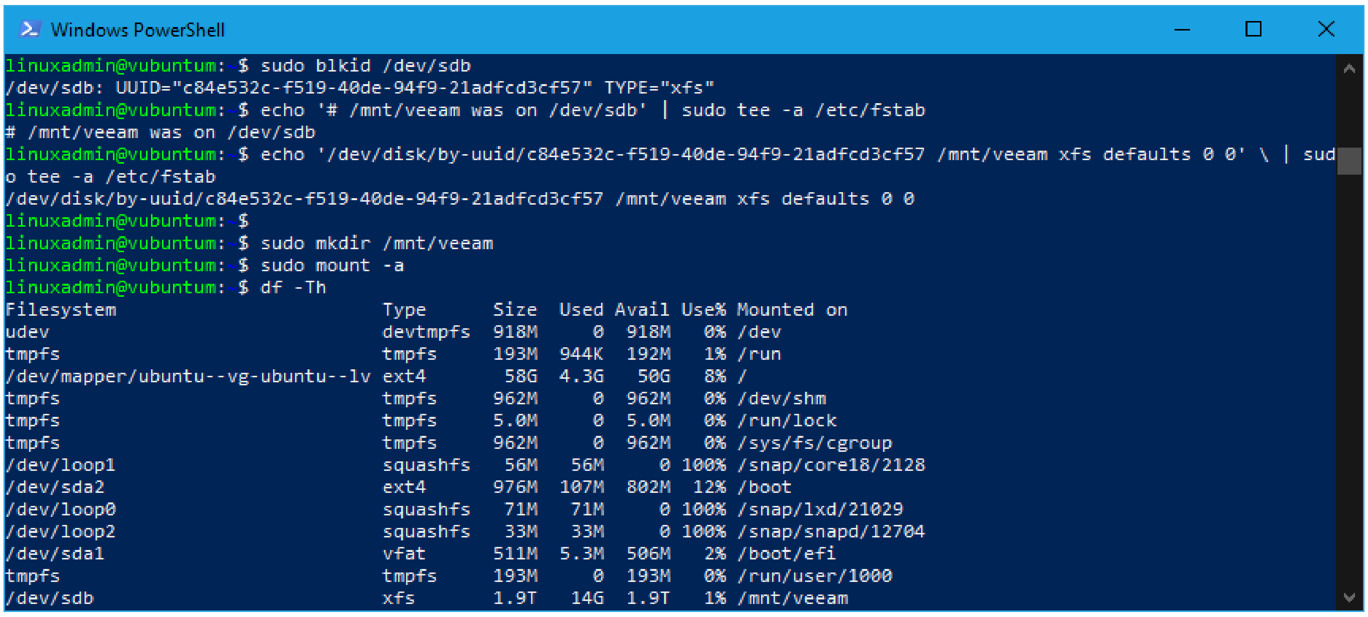
Step two is to mount the drives as specified in the */etc/fstab* configuration file:

```
sudo mount -a
```

Finally, confirm that we are set, by listing the devices with this command:

```
df -Th
```

The last commands and outputs will appear as here:



```
linuxadmin@vubuntum:~$ sudo blkid /dev/sdb
/dev/sdb: UUID="c84e532c-f519-40de-94f9-21adfcd3cf57" TYPE="xfs"
linuxadmin@vubuntum:~$ echo '# /mnt/veeam was on /dev/sdb' | sudo tee -a /etc/fstab
# /mnt/veeam was on /dev/sdb
linuxadmin@vubuntum:~$ echo '/dev/disk/by-uuid/c84e532c-f519-40de-94f9-21adfcd3cf57 /mnt/veeam xfs defaults 0 0' \ | sud
o tee -a /etc/fstab
/dev/disk/by-uuid/c84e532c-f519-40de-94f9-21adfcd3cf57 /mnt/veeam xfs defaults 0 0
linuxadmin@vubuntum:~$
linuxadmin@vubuntum:~$ sudo mkdir /mnt/veeam
linuxadmin@vubuntum:~$ sudo mount -a
linuxadmin@vubuntum:~$ df -Th
Filesystem                        Type      Size  Used Avail Use% Mounted on
udev                              devtmpfs  918M     0  918M   0% /dev
tmpfs                             tmpfs     193M  944K  192M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv ext4       58G  4.3G   50G   8% /
tmpfs                             tmpfs     962M     0  962M   0% /dev/shm
tmpfs                             tmpfs     5.0M     0  5.0M   0% /run/lock
tmpfs                             tmpfs     962M     0  962M   0% /sys/fs/cgroup
/dev/loop1                        squashfs   56M   56M     0 100% /snap/core18/2128
/dev/sda2                         ext4      976M  107M  802M  12% /boot
/dev/loop0                        squashfs   71M   71M     0 100% /snap/lxd/21029
/dev/loop2                        squashfs   33M   33M     0 100% /snap/snapd/12704
/dev/sda1                         vfat      511M  5.3M  506M   2% /boot/efi
tmpfs                             tmpfs     193M     0  193M   0% /run/user/1000
/dev/sdb                          xfs       1.9T   14G  1.9T   1% /mnt/veeam
```

You will notice our XFS drive listed having the *Type* of xfs and the mounting point for Veeam.

Permissions

The last task is to set the very limited permissions to the backup drive.

First, assign our veeam user account as the *owner* of the mounting point for the drive:

```
sudo chown -R veeamuser:veeamuser /mnt/veeam/
```

Next, assign the

- *owner* (**u** in the command line) *read*, *write*, and *execute* rights
- *groups* and *other users* (**g** and **o** in the command line) *no rights*

by calling this command:

```
sudo chmod u+rwx,go-rwx /mnt/veeam
```

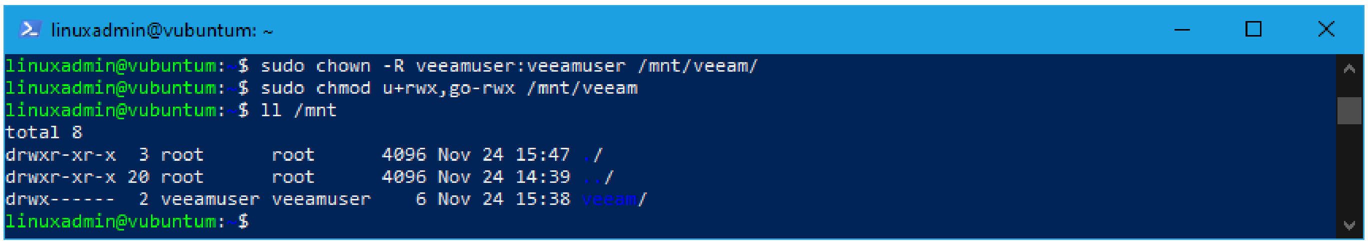
Confirm the settings:

```
ll /mnt
```

Here, it is the last line where the tripled dashes, **---**, indicate no rights:

```
drwx------  2 veeamuser veeamuser    6 Nov 12 13:36 veeam/
```

here:



```
linuxadmin@vubuntum:~$ sudo chown -R veeamuser:veeamuser /mnt/veeam/
linuxadmin@vubuntum:~$ sudo chmod u+rwx,go-rwx /mnt/veeam
linuxadmin@vubuntum:~$ ll /mnt
total 8
drwxr-xr-x  3 root      root      4096 Nov 24 15:47 ./
drwxr-xr-x 20 root      root      4096 Nov 24 14:39 ../
drwx------  2 veeamuser veeamuser    6 Nov 24 15:38 veeam/
linuxadmin@vubuntum:~$
```

Conclusion

The server is now running Ubuntu Server ready for the next step - installation of the immutable backup repository - which is explained in detail in **Part 4** of this series:

Build an immutable backup repository for Veeam Backup & Replication. Part 4