

Shortly after becoming the Linux system administrator for Dunder Mifflin, you receive a few ‘friendly’ e-mails from ‘management’ complaining that the internal file server’s disk is full. Instead of being able to jump to work on it, ‘management’ pulls you into a 2 hour meeting to discuss which project management approach should be used to handle the disaster, who will manage the project, etc., etc.

After putting up with the meeting for a total of 15 minutes, you silently step out, grab a spare 5 GB SATA hard drive, and plug it into the server. You then find some [online resources](#) (this one being somewhat [detailed](#)) leading you through the process of setting up logical volumes on the new disk because you recall a buddy mentioning that this was a good practice.

Once you finish setting up logical volumes, you’ll need to make a file system (mkfs) on each one, mount the filesystems, copy the old contents to the new filesystems, then modify /etc/fstab to roughly match the table below. Please note that you’ll need to use the appropriate device names for your specific virtual machine. The sizes are also approximate and need not match exactly.

Device	Mount Point	Size	Type	Mount Options
/dev/centos/root	/	4 GB	xfs	defaults
/dev/sda1	/boot	500 MB	xfs	defaults
/dev/centos/home	/home	4 GB	xfs	nodev
/dev/centos/tmp	/tmp	1 GB	xfs	nodev,nosuid,noexec
/dev/centos/swap	None	512 MB	swap	

Why Logical Volumes?

Logical volumes mitigate many of the issues commonly faced with partitioning; e.g., the very limited number of partitions. The logical volume manager supports up to 2^{32} logical volumes. Logical volumes not only alleviate some of the physical limitations of partitioning, but also give advanced features such as:

- the ability to easily resize logical volumes (a typically painful and error-prone process with partitions)
- the ability to take a snapshot of a logical volume and revert to it at any time (impossible with partitions alone)
- the ability to move an entire logical volume from one physical disk to another (also impossible with partitions alone)

These are just some of the reasons to prefer logical volumes to partitions. Major downsides are the added complexity of setup and lack of redundancy. This is currently being addressed by the development of a kernel-based filesystem called btrfs. btrfs wraps the functionality of RAID and Logical Volumes directly into the filesystem itself.

Dunder Mifflin Network Topology & Configuration

Please note that some of this information may not make much sense at this point. We'll be using this network throughout the semester so please keep this as a reference and gloss over some of the details for now.

D-M has the following CentOS Linux servers:

1. A gateway (machine A) handles dhcp, ip_forwarding, and network/port address translation (nat/pat).
2. A file server (machine E) stores company files centrally.
3. An http server (machine B) hosts the corporate website.
4. An ftp server (machine C) updates prices and accepts batch orders.
5. A dns server (machine D) maps between domain names and IP addresses.

You will be given a unique number this semester. You need to substitute that number for each N in an IP address.

All your production machines have the same default root password of saclass. You are strongly encouraged to change this password to something more secure immediately.

The ssh (secure shell) service is running on all production machines so you can ssh in after connecting to the vpn.

Most of your production machines can be accessed from the vpn. The sole exception is machine E. To reach machine E, first ssh from the vpn into machine A. From machine A you can then ssh (pivot) into machine E.

Prerequisites

Please finish this week's reading before trying this lab. You are strongly encouraged to try this on your own virtual machine before trying it on the production machines.

You will also need to install some software packages on your machine. Run the following command to ensure that you have everything needed before proceeding:

```
# yum install parted ...
```

Submission Requirements

1. Please submit your notes on this lab. At a minimum they should list the commands you ran to add the storage, copy the existing contents, etc.

2. Your internal file server (machine E) should have two logical volumes named `lv_home` (4GB) and `lv_tmp` (1GB). They should be contained in one volume group named `vg_new`. That volume group should live on one physical volume `/dev/sdb` on the new hard drive that has been added to the production virtual machine for you.
3. Any existing contents of `/home/`, and `/tmp/` should be moved to the logical volumes on the new disk and should match the structure of the old disk.

Hints

Below is a list of steps you may take to successfully add storage. Feel free to use it as a basis for your lab notes.

1. Create a new volume group (`vgcreate`).
2. Create the logical volumes (`lvcreate`).
3. Make two xfs filesystems (`mkfs`).
4. Mount the new filesystems to temporary mount points (`mount`).
5. Copy the existing contents from the old disk to the new filesystems (`cp -r`).
6. Mount the new logical volumes where they belong and update `/etc/fstab` so they are remounted there on reboot.

At this point, have you actually freed up the disk blocks in the original filesystem? If not, how can you free this no-longer-used storage in the root filesystem?