LVM

Problem with standard partitions

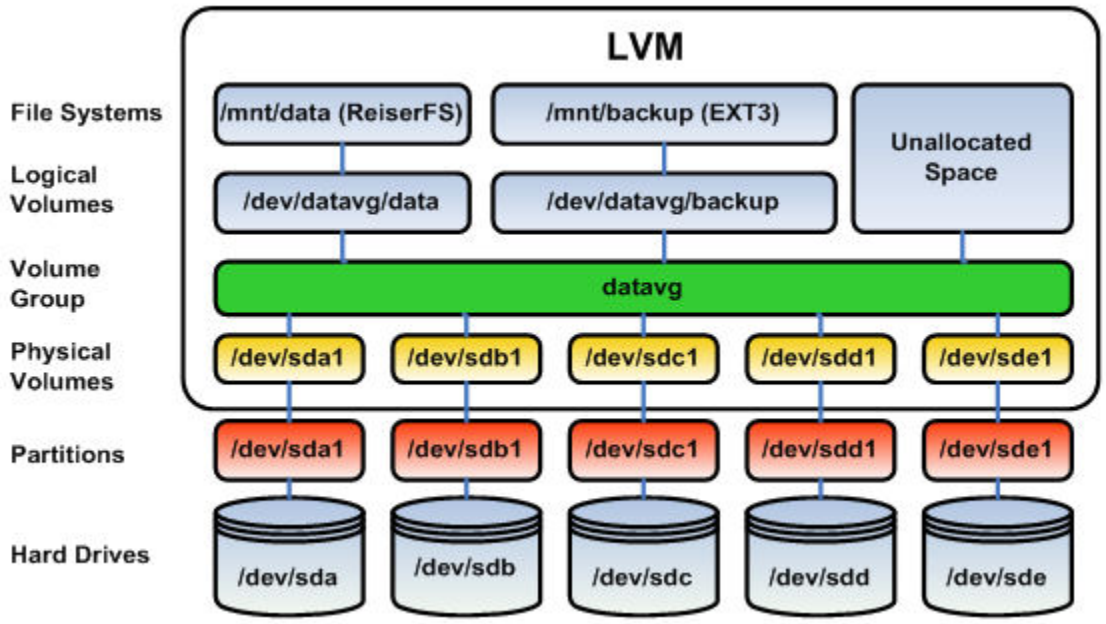
There are some problems when working with the harddisks and standard partitions. Consider a system with a small and a large hard disk drive, partitioned.

Suppose, the first disk(/ dev/sda) is partition in two, the second disk (/dev/sdb) has two partitions and some empty space.

In the example above, consider the option when you want to enlarge the space available for /mnt/backup. What can you do? The solution will always force you to unmount the file system, take a backup of the data, remove and recreate partitions, and then restore the data and remount the file system.

Solution with LVM

Using LVM will create a virtual layer between the mounted fle systems and the hardware devices. This virtual layer will allow for an administrator to enlarge a mounted file system in use. When LVM is properly used, then there is no need to unmount the file system to enlarge it.



Resizing Root disk using shell provisioner

Disk resizing through vagrantfile requires the vagrant-disk resize plugin

1. List the vagrant plugins installed with the below command

   vagrant plugin list

2. Install vagrant-disksize plugin with the below command

   vagrant plugin install vagrant-disksize

Vagrantfile for disk resizing on centos8

Save the below as Vagrantfile

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
unless Vagrant.has_plugin?("vagrant-disksize")
 raise 'vagrant-disksize is not installed!'
end
Vagrant.configure("2") do |config|
 config.vm.box = "generic/centos8"

 #config.vm.network "private_network", ip: "192.168.56.2"
  #config.vm.synced_folder "../CentOS", "/vagrant"

  config.vm.provider "virtualbox" do |vb|
    vb.memory = "1024"
    vb.cpus = 2
  end
  #Expand default 128GB disk to 140GB
  config.disksize.size = '140GB'
  config.vm.provision "shell", path: "resizeDisk.sh", keep_color: "true"
end
```

https://github.com/srtimsina/DevOps/tree/master/vagrantDisk

Shell script to resize the disk

```
#!/bin/bash

echo "Expanding disk...."

ROOR_DISK_TO_EXPAND="/dev/sda"
sudo fdisk $ROOR_DISK_TO_EXPAND <<EOF
p
d
n
p
1
2048

w
EOF
/usr/sbin/partprobe
/usr/sbin/xfs_growfs -d /
df -h /
echo "******************************************************"
echo "Disk seems to be expanded"
echo "Run vagrant ssh to login the vm and check disk status"
```

Run ***vagrant up***

Vagrant file for disk resizing on ubuntu

Save the below as Vagrantfile

```
unless Vagrant.has_plugin?("vagrant-disksize")
   raise 'vagrant-disksize is not installed!'
 end
Vagrant.configure("2") do |config|
   config.vm.hostname = "UbuntuVM"
   config.vm.box = "bento/ubuntu-18.04"
   config.vm.provider "virtualbox" do |vb|
     vb.name = "UbuntuLab"
     vb.memory = 1024
     vb.cpus = 1
   end
     config.vm.define :"UbuntuVM" do |t|
```

```
  end
  # Expand disk from 64GB to 84GB
  config.disksize.size = "84GB"
  config.vm.provision "shell", path: "resizeDisk.sh", keep_color: "true"
end
```

Save the below as resizeDisk.sh

```
#!/bin/bash

echo "Expanding disk..."
ROOR_DISK_TO_EXPAND="/dev/sda"
ROOT_DISK_DEVICE_PART="/dev/sda1"
LV_PATH=`sudo lvdisplay -c | sed -n 1p | awk -F ":" '{print $1;}'`
FS_PATH=`df / | sed -n 2p | awk '{print $1;}'`
ROOT_FS_SIZE=`wr`
echo "The root file system (/) has a size of $ROOT_FS_SIZE"
echo "> Increasing disk size of $ROOR_DISK_TO_EXPAND to available maximum"
sudo fdisk $ROOR_DISK_TO_EXPAND <<EOF
d
n
p
1
2048

no
w
EOF
sudo pvresize $ROOT_DISK_DEVICE_PART
sudo lvextend -l +100%FREE $LV_PATH
sudo resize2fs -p $FS_PATH
ROOT_FS_SIZE=`df -h / | sed -n 2p | awk '{print $2;}'`
echo "The root file system (/) has a size of $ROOT_FS_SIZE"
echo "**************************************************"
echo "Disk seems to be expanded"
echo "Run vagrant ssh to login the vm and check disk status"
```

Run ***vagrant up***

Ref:
https://www.learnlinux.tv/linux-logical-volume-manager-lvm-deep-dive-tutorial/