Configuring Volume Groups and Logical Volumes

Andrew Mallett
LINUX AUTHOR AND TRAINER

@theurbanpenguin www.theurbanpenguin.com

Overview

Options for physical volumes

Setting the volume group PE size

Creating logical volumes and filesystems

Resizing logical volumes online

Creating volume groups, logical volumes and filesystems with Ansible

Physical Volumes Options

We can skip the separate definition of our physical volumes unless we need to control the meta-data storage. Physical volumes will be defined with the volume group.

```
$ sudo vgcreate vg1 -s 8m /dev/sdb1 /dev/sdb2
```

Volume Group Physical Extent Size

The default PE size is 4M and affects all physical volumes in a volume group. The setting was more important in LVM1 when there was a limitation of 65k extents within a volume group, this is not the case with LVM2. Setting a larger PE size now has more of an affect when striping data with RAID in LVM.

```
$ sudo vgextend vg1 /dev/sdb3 /dev/sdb4
```

Extending the Volume Group Size

Adding physical volumes to the volume group increases the volume group size.

```
$ sudo lvcreate -n lv1 -L 100M vg1

$ sudo lvcreate -n lv2 -l 12 vg1
```

Creating Logical Volumes

When defining logical volumes we can use -L to specify the desired size that will be adjusted to the nearest value matching multiples of the PE size.

Using -l we can specify the number of extents to use, 12 extents would be 96M where an 8 M extent size is set.

```
$ dmsetup ls

$ dmsetup info

$ ls -l /dev/vg1/lv1 /dev/mapper/vg1-lv1

$ lsblk /dev/sdb

$ sudo mkfs.ext4 /dev/vg1/lv1

$ sudo mkdir /data ; sudo mount /dev/vg1/lv1 /data
```

Device Mapper Devices

Logical volumes are created as /dev/dm-<number> devices, for ease we have symbolic links that we can use to refer to the volumes. You can use either the link /dev/mapper/vg1-lv1 or /dev/vg1/lv1.

```
$ sudo lvextend -l +12 -r vg1/lv1
```

Extending LV Size

Logical volumes can be increased in size whilst they are in use. If the filesystem is EXT4 or XFS we can also support the extension of the filesystem size whilst online. Here we add another 12 extents.

Demo

In this first demonstration we will create the volume group and then extend it

Demo

Now that we have the volume group we can create logical volumes and retrieve information about the devices created

Demo

Will will now create a filesystem on the logical volume and mount it. With the volume mounted we will extend the size of the logical volume and the filesystem

```
$ ansible-doc -l | grep vg

$ ansible-doc lvg

- name: Create vg1
  lvg:
    vg: vg1
    pvs: /dev/sdb1, /dev/sdb2
    pesize: 8
```

Volume Groups Using Ansible

The module documentation is useful when working with Ansible

```
$ ansible -b all -m lvg -a "vg=vg1 pvs=/dev/sdb1,/dev/sdb2 pesize=8"

$ ansible -b all -m lvol -a "vg=vg1 lv=lv1 size=100m"

$ ansible -b all -m file -a " path=/data state=directory"

$ ansible -b all -m filesystem -a "type=ext4 dev=/dev/vg1/lv1"

$ ansible -b all -m mount \
  -a "path=/data src=/dev/vg1/lv1 fstype=ext4 state=mounted"
```

Configuration Using Ad-Hoc Commands

If we do not want to create the Playbook we can create the setup entirely with ad-hoc commands

Demo

For the managed devices we will replicate the setup using Ansible

Overview

The command vgcreate can also create the physical volumes if not pre-defined

Setting the PE size is less important in LVM2

Logical volumes can be defined by size or number of PEs

We can extend logical volumes online along with the filesystem using the option -r

Ansible can configure all this for us

VG and LV

site.yml

```
- name: Create VG
  lvg:
        vg: vg1
        pvs: /dev/sdb1,/dev/sdb2
        pesize: 8

- name: Create LV
  lvol:
        vg: vg1
        lv: lv1
        size: 96m
```

Mount Point and Format

site.yml

```yaml
- name: create dir
  file:
      path: /data
      state: directory

- name: format
  filesystem:
      type: ext4
      device: /dev/vg1/lv1
```

Mount FS

site.yml

```yaml
- name: mount
  mount:
        path: /data
        src: /dev/vg1/lv1
        fstype: ext4
        state: mounted
```

Creating Thinly Provisioned Volumes