redhat.

New RHEL 7.5 features: VDO, USBGuard, NBDE and AIDE

RHUG Q1.2018

Marc Skinner
Principal Solutions Architect

3/21/2018

RHEL7.5beta :: New Features

Storage
- Virtual Data Optimizer (VDO)

Security
- NBDE
- USBGuard
- AIDE


VDO

VDO :: Overview

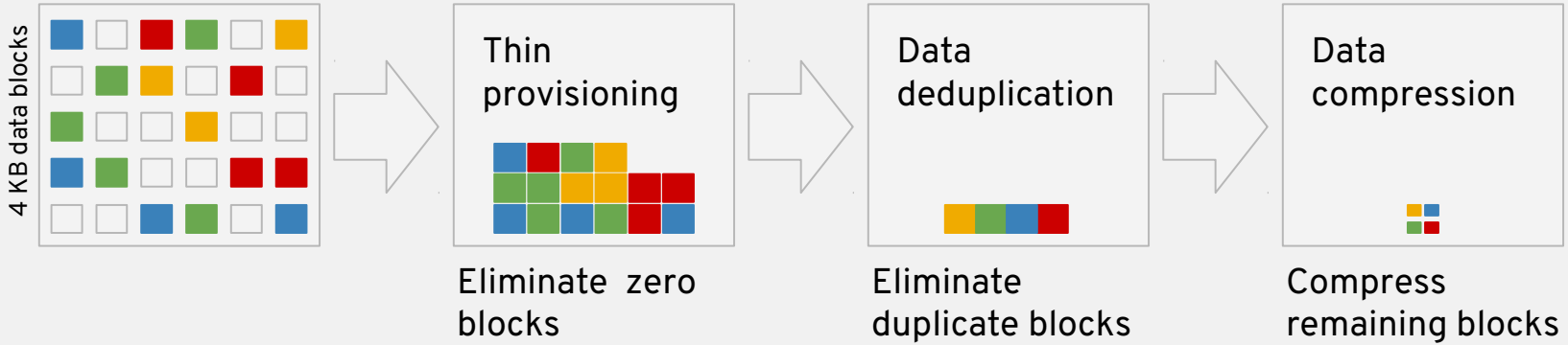
Virtual Data Optimizer (VDO)

- Permabit acquired in July, 2017
- Data reduction
  - Data compression
  - Data deduplication
  - Device mapper module
  - 4 KB granularity
  - Thin provisioning
  - Zero block elimination

Red Hat Enterprise Linux 7.5

How does Virtual Data Optimizer work?

VDO data reduction processing



- Operates inline
- Works at the block level, with the file system of your choice
- Supports up to **256 TB physical / 4 PB logical storage**

redhat.

VDO :: Requirements

- Memory requirements: 370 MB + 268 MB per 1 TB of physical storage managed
- Under VDO
  - Block device
  - DM-Multipath
  - DM-Crypt
  - Software RAID (MDRaid/LVM)
- On top of VDO
  - LVM cache
  - LVM Logical Volumes
  - LVM snapshots
  - LVM Thin Provisioning

VDO :: Installation

Included in base RHEL 7.5

```
# yum install vdo kmod-kvdo
```

Configuration steps:

```
[root@beast]# vdo create --name=vdo1 --device=/dev/sdb --vdoLogicalSize=1T
Creating VDO vdo1
Starting VDO vdo1
Starting compression on VDO vdo1
VDO instance 1 volume is ready at /dev/mapper/vdo1
```

VDO :: Installation

Physically this is a 512Gb drive
Logically it is 1TB

```
[root@beast ~]# lsblk /dev/sdb
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sdb      8:16   0  477G  0 disk
└─vdo1 253:0    0    1T  0 vdo
```

VDO :: Installation

Make sure to pass the nodiscard at mkfs time option.
XFS = -K
Ext4 = -E nodiscard

```
[root@beast ~]# mkfs.xfs -K /dev/mapper/vdo1
```

Mount file system:

```
[root@beast ~]# mkdir -m 1777 /vdo1
[root@beast /]# mount /dev/mapper/vdo1 /vdo1

[root@beast /]# df -h /vdo1
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vdo1  1.0T   33M  1.0T    1%        /vdo1
```

VDO :: Installation

/etc/fstab configuration

XFS:
/dev/mapper/vdo_name /mnt/vdo_name xfs defaults,**x-systemd.requires=vdo.service** 0 0

Ext4:
/dev/mapper/vdo_name /mnt/vdo_name ext4 defaults,**x-systemd.requires=vdo.service** 0 0

VDO :: Installation

systemd mount unit file

/etc/systemd/system/mnt-vdo_name.mount

```
[Unit]
Description = VDO unit file to mount file system
name = vdo_name.mount
Requires = vdo.service
After = multi-user.target
Conflicts = umount.target

[Mount]
What = /dev/mapper/vdo_name
Where = /mnt/vdo_name
Type = xfs

[Install]
WantedBy = multi-user.target
```

VDO :: Discards

Discard Support

Choose a discard method

Batch or Online

- Batch = use “fstrim /mnt” from command line or cron
- Online = pass ‘-o discard’ during mount command or in fstab

VDO :: Write Modes

Write modes : SYNC vs ASYNC

- Default is sync – guarantees data is written to persistent storage
- Change to async when your persistent storage is backed by volatile write back cache

How to change online:
vdo changeWritePolicy --writePolicy=sync_or_async –name=vdo_name

During creation:
--writePolicy=sync or --writePolicy=async option

VDO :: Changing Features

```
[root@beast ~]# vdo status --name=vdo1 | grep enable
    Activate: enabled
    Compression: enabled
    Deduplication: enabled
```

You can enable/disable options

- vdo disableCompression
- vdo enableCompression
- vdo disableDeduplication
- vdo enableDeduplication

VDO :: Growing

```
[root@beast ~]# vdo growLogical --name=vdo1 --vdoLogicalSize=2T
```

You can grow both logical and physical:

- vdo growLogical
- vdo growPhysical

Don’t forget to grow file system after!

XFS:

```
[root@beast /]# xfs_growfs /dev/mapper/vdo1
```

Ext4:

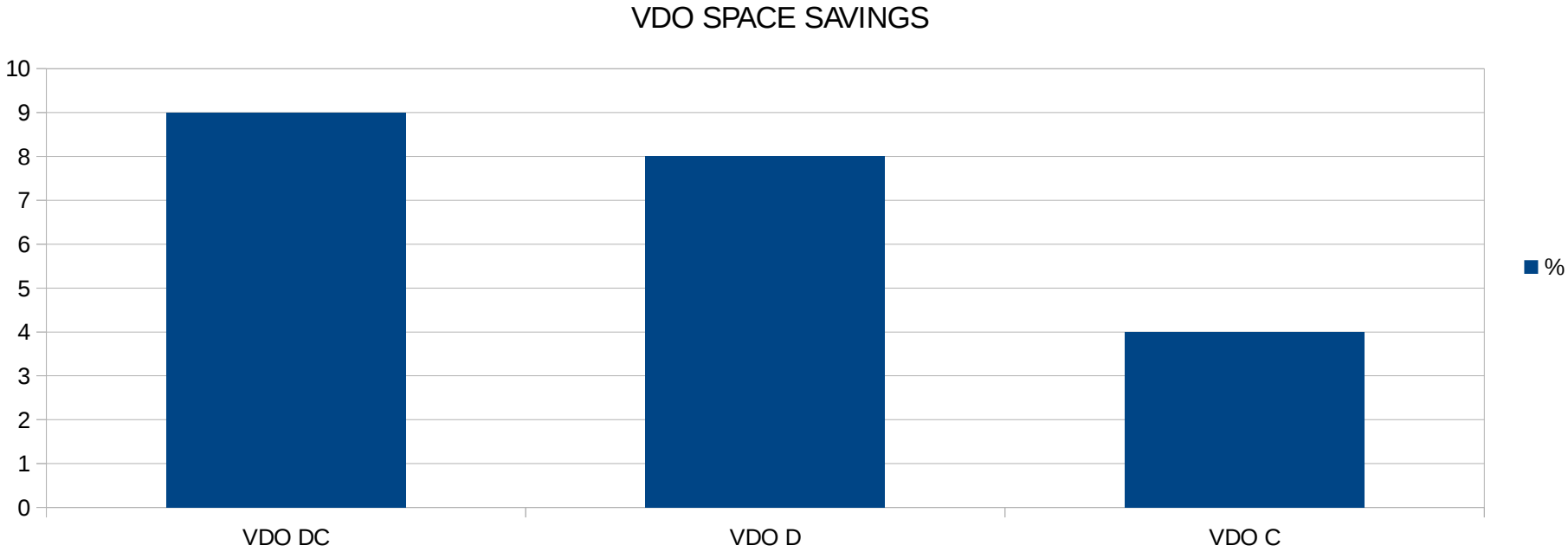
```
[root@beast /]# resize2fs /dev/mapper/vdo2
```

VDO :: Monitoring

```
[root@beast ~]# vdostats
Device                    1K-blocks       Used          Available  Use% Space saving%
/dev/mapper/vdo3    1953514584  1720733384  232781200  88%         4%
/dev/mapper/vdo2    1953514584  1656238584  297276000  84%         8%
/dev/mapper/vdo1    1953514584  1637262208  316252376  83%         9%

[root@beast ~]# vdostats --si
Device                    Size    Used   Available  Use%    Space saving%
/dev/mapper/vdo3        2.0T    1.8T     238.4G     88%         4%
/dev/mapper/vdo2        2.0T    1.7T     304.4G     84%         8%
/dev/mapper/vdo1        2.0T    1.7T     323.8G     83%         9%
```

VDO :: Comparison



* Data set = 1.7TB of ISOs, docs, mp3, mp4, KVM Images *


USBGuard

USBGuard :: Overview

- Framework to protect against intrusive USB devices
- Leverages basic white/black listing capabilities
- Runs as a service to enforce configuration

USBGuard :: Installation

#yum install usbguard

Create initial policy:

[root@beast /]# usbguard generate-policy
allow id 413c:2003 serial "" name "Dell USB Keyboard" hash "3eEGsGE566EIofQwRf06ElNoPRynZIa/09c3uyy4TTY=" parent-hash "kubOH4vQHJd4zkFzCv1sCQG996gY2Syo4t78mponbz0=" via-port "5-3" with-interface 03:01:01

allow id 045e:00d1 serial "" name "Microsoft Optical Mouse with Tilt Wheel" hash "6YtdWilS3nccohW1/gto2o+HjZ016We+FAJU32Pzeow=" parent-hash "kubOH4vQHJd4zkFzCv1sCQG996gY2Syo4t78mponbz0=" via-port "5-4" with-interface 03:01:02

USBGuard :: Installation

Create initial policy:

```
[root@beast /]# usbguard generate-policy > /etc/usbguard/rules.conf
[root@beast /]# systemctl start usbguard.service
[root@beast /]# systemctl enable usbguard.service
```

List USB devices recognized by USBGuard:

```
[root@beast usbguard]# usbguard list-devices | grep Dell
7: allow id 413c:2003 serial "" name "Dell USB Keyboard" hash
"3eEGsGE566EIofQwRf06EINoPRynZIa/09c3uyy4TTY=" parent-hash
"kubOH4vQHJd4zkFzCv1sCQG996gY2Syo4t78mponbz0=" via-port "5-3" with-interface
03:01:01
```

USBGuard :: Allow

Allow device:

```
[root@beast usbguard]# usbguard list-devices | grep Dell
7: block id 413c:2003 serial "" name "Dell USB Keyboard" hash
"3eEGsGE566EIofQwRf06EINoPRynZIa/09c3uyy4TTY=" parent-hash
"kubOH4vQHJd4zkFzCv1sCQG996gY2Syo4t78mponbz0=" via-port "5-3" with-interface
03:01:01

[root@beast usbguard]# usbguard allow-device 7

[root@beast usbguard]# usbguard list-devices | grep Dell
7: allow id 413c:2003 serial "" name "Dell USB Keyboard" hash
"3eEGsGE566EIofQwRf06EINoPRynZIa/09c3uyy4TTY=" parent-hash
"kubOH4vQHJd4zkFzCv1sCQG996gY2Syo4t78mponbz0=" via-port "5-3" with-interface
03:01:01
```

USBGuard :: Block

Block device:

```
[root@beast usbguard]# usbguard list-devices | grep -i mouse
8: allow id 045e:00d1 serial "" name "Microsoft Optical Mouse with Tilt Wheel" hash
"6YtdWilS3nccohW1/gto2o+HjZ016We+FAJU32Pzeow=" parent-hash
"kubOH4vQHJd4zkFzCv1sCQG996gY2Syo4t78mponbz0=" via-port "5-4" with-interface
03:01:02

[root@beast usbguard]# usbguard block-device 8

[root@beast usbguard]# usbguard list-devices | grep -i mouse
8: block id 045e:00d1 serial "" name "Microsoft Optical Mouse with Tilt Wheel" hash
"6YtdWilS3nccohW1/gto2o+HjZ016We+FAJU32Pzeow=" parent-hash
"kubOH4vQHJd4zkFzCv1sCQG996gY2Syo4t78mponbz0=" via-port "5-4" with-interface
03:01:02
```

USBGuard :: Reject

Reject device:

#usbguard reject-device 9

What is difference between block vs reject?

Block = do not talk to device for now
Reject = ignore device as if it wasn’t plugged in

USBGuard :: More Rules

Add more rules to your rules.conf file

Allow USB mass storage devices and block everything else:

allow with-interface equals { 08:*:* }

Now, all USB Storage is blocked, unless allowed:

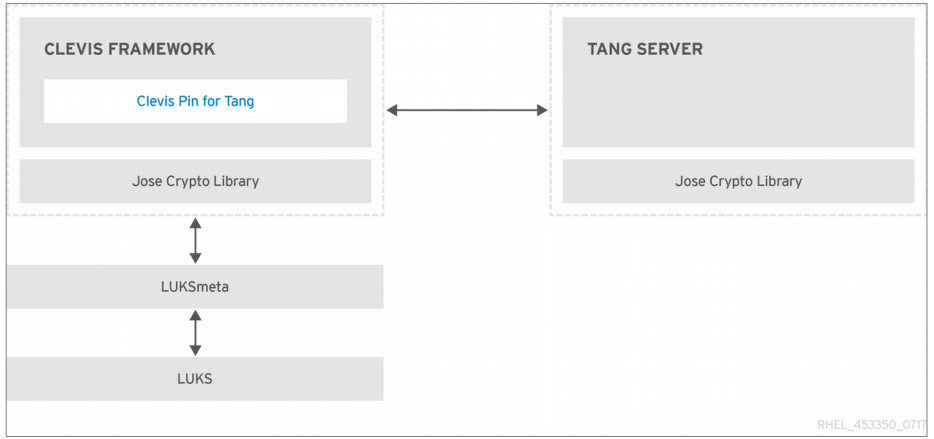
block id 0781:5575 serial "20052243710EDF505221" name "Cruzer Glide" hash "RID8xP5D82KeiyMRtwi/ucBuP7BBHgqlrl+5M/sH65M=" parent-hash "WLLK8WnDcOZsmC13ldzWRmahoyPD7Y+TLn1uJ4TB2GU=" via-port "1-4" with-interface 08:06:50


NBDE

NBDE :: Overview

Network-Bound Disk Encryption (NBDE)
Linux Unified Key Setup-on-disk-format (LUKS)

- Enables LUKS disk encryption of root/non-root disks without requiring manual passphrase
- CLEVIS Framework
- TANG Server



NBDE :: TANG - Installation

On pair of RHEL servers – need HA!

```
# yum -y install tang
[root@tang1 ~]# systemctl enable tangd.socket –now
[root@tang2 ~]# systemctl enable tangd.socket –now

[root@tang1 ~]# firewall-cmd --zone=public --add-port=80/tcp –permanent
[root@tang1 ~]# firewall-cmd --reload
```

TANG listens on port TCP 80

Can override port in systemd unit file: tangd.socket

NBDE :: LUKS - Installation

Create LUKS Volume:

[root@beast ~]# cryptsetup --verify-passphrase luksFormat /dev/md0

WARNING!
========
This will overwrite data on /dev/md0 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase: **redhat2018**
Verify passphrase: **redhat2018**

NBDE :: LUKS - Installation

Open encrypted device:

[root@beast ~]# cryptsetup luksOpen /dev/md0 secret
Enter passphrase for /dev/md0: **redhat2018**

Look at new device:
[root@beast ~]# ls /dev/mapper/secret
/dev/mapper/secret

Format encrypted device:
[root@beast ~]# mkfs.xfs /dev/mapper/secret

NBDE :: LUKS - Installation

Mount encrypted device:

```
[root@beast ~]# mount /dev/mapper/secret /SECRET

[root@beast ~]# df /SECRET
Filesystem       1K-blocks  Used Available Use% Mounted on
/dev/mapper/secret 499727972 32944 499695028   1% /SECRET
```

Tell LUKS to open device on boot – which will prompt for passphrase

```
[root@beast ~]# echo secret /dev/md0 none _netdev >> /etc/crypttab
```

Update fstab before reboot!

```
/dev/mapper/secret       /SECRET            xfs       _netdev      1   2
```

NBDE :: CLEVIS - Installation

```
# yum install clevis clevis-luks clevis-dracut

TEST - Download TANG server advertisements:

[root@beast ~]# curl -f http://tang1.i.skinnerlabs.com/adv > adv1.jws
[root@beast ~]# curl -f http://tang2.i.skinnerlabs.com/adv > adv2.jws
```

NBDE :: CLEVIS - Installation

BIND LUKS to TANG Server: TANG1

```
[root@beast ~]# clevis bind luks -d /dev/md0 tang '{"url":"http://tang1.i.skinnerlabs.com"}'
The advertisement contains the following signing keys:

Mdbv_aFzqDpRR9_L-O-ByY-a9B8

Do you wish to trust these keys? [ynYN] Y
You are about to initialize a LUKS device for metadata storage.
Attempting to initialize it may result in data loss if data was
already written into the LUKS header gap in a different format.
A backup is advised before initialization is performed.

Do you wish to initialize /dev/md0? [yn] y
Enter existing LUKS password:  redhat2018
```

NBDE :: CLEVIS - Installation

BIND LUKS to TANG Server: TANG2

[root@beast ~]# clevis bind luks -d /dev/md0 tang '{"url":"http://tang2.i.skinnerlabs.com"}'
The advertisement contains the following signing keys:

kFH77GdVfZ11CRfEQ5U47w3jGfQ

Do you wish to trust these keys? [ynYN] Y
Enter existing LUKS password: **redhat2018**

NBDE :: CLEVIS - Installation

Confirm both TANG servers have keys registered to device:

```
[root@beast ~]# yum -y install luksmeta
[root@beast ~]# luksmeta show -d /dev/md0
0   active empty
1   active cb6e8904-81ff-40da-a84a-07ab9ab5715e
2   active cb6e8904-81ff-40da-a84a-07ab9ab5715e
3 inactive empty
4 inactive empty
5 inactive empty
6 inactive empty
7 inactive empty
```

NBDE :: CLEVIS - Installation

Boot Unlocking

1 – ROOT Volumes need to update initramfs:
[root@beast ~]# dracut -f

*** NON DHCP PROD ENVIRONMENTS!!! ***
[root@beast ~]# dracut -f --kernel-cmdline "ip=192.168.33.225 netmask=255.255.255.0 gateway=192.168.33.1 nameserver=192.168.33.45"

2 – NON-ROOT Volumes need helper app:

[root@beast ~]# systemctl enable clevis-luks-askpass.path

NBDE :: CLEVIS - Installation

TANG Logs:

[root@tang2 log]# tail -f messages
Mar 19 12:53:32 tang2 systemd: Created slice system-tangd.slice.
Mar 19 12:53:32 tang2 systemd: Starting system-tangd.slice.
Mar 19 12:53:32 tang2 systemd: Started Tang Server (**192.168.33.225:42312**).
Mar 19 12:53:32 tang2 systemd: Starting Tang Server (**192.168.33.225:42312**)...
Mar 19 12:53:32 tang2 tangd: **192.168.33.225 POST** /rec/9mbH8oDHppKTzwmD_b8EsfbZXFI => 200 (src/tangd.c:168)

AIDE

AIDE :: Overview

Advanced Intrusion Detection Environment (AIDE)

- Creates a database of file/directories to monitor
- Ensures integrity of monitored files/directories
- Detects changes

AIDE :: Installation

```
# yum install aide

Initialize database based on paths in /etc/aide.conf

[root@beast /]# aide --init

AIDE, version 0.15.1

### AIDE database at /var/lib/aide/aide.db.new.gz initialized.
```

AIDE :: Installation

```
[root@beast /]# mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz

[root@beast /]# aide --check

AIDE, version 0.15.1

### All files match AIDE database. Looks okay!
```

AIDE :: Configuration

Configure /etc/aide.conf

PATH	RULE	ACTUAL
/boot/	CONTENT_EX	Watch directory, use sha256, extended content
/root/..*	PERMS	Watch ACLs only
/opt/	CONTENT	Watch directory, use sha256 and file type
!/etc/.*~		Ignore backup files

redhat.

AIDE :: Detection

```
[root@beast etc]# aide --check
AIDE 0.15.1 found differences between database and filesystem!!
Start timestamp: 2018-03-08 13:52:21

Summary:
  Total number of files:       156467
  Added files:                 0
  Removed files:               0
  Changed files:               1
---------------------------------------------------
Changed files:
---------------------------------------------------
changed: /etc/resolv.conf
---------------------------------------------------
Detailed information about changes:
---------------------------------------------------

File: /etc/resolv.conf
 Size    : 125                       , 100
 SHA256  : 39tnhzhuiTsbKLHb/fV58GztrS8O74/T , IdsL4CvZWTFfyMeIeCIKhDXrNr0YqUfS
```

AIDE :: Update

```
[root@beast etc]# aide --update

[root@beast etc]# aide --check

AIDE, version 0.15.1

### All files match AIDE database. Looks okay!
```

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos