Virtual Data Optimizer (VDO)

Unit objectives

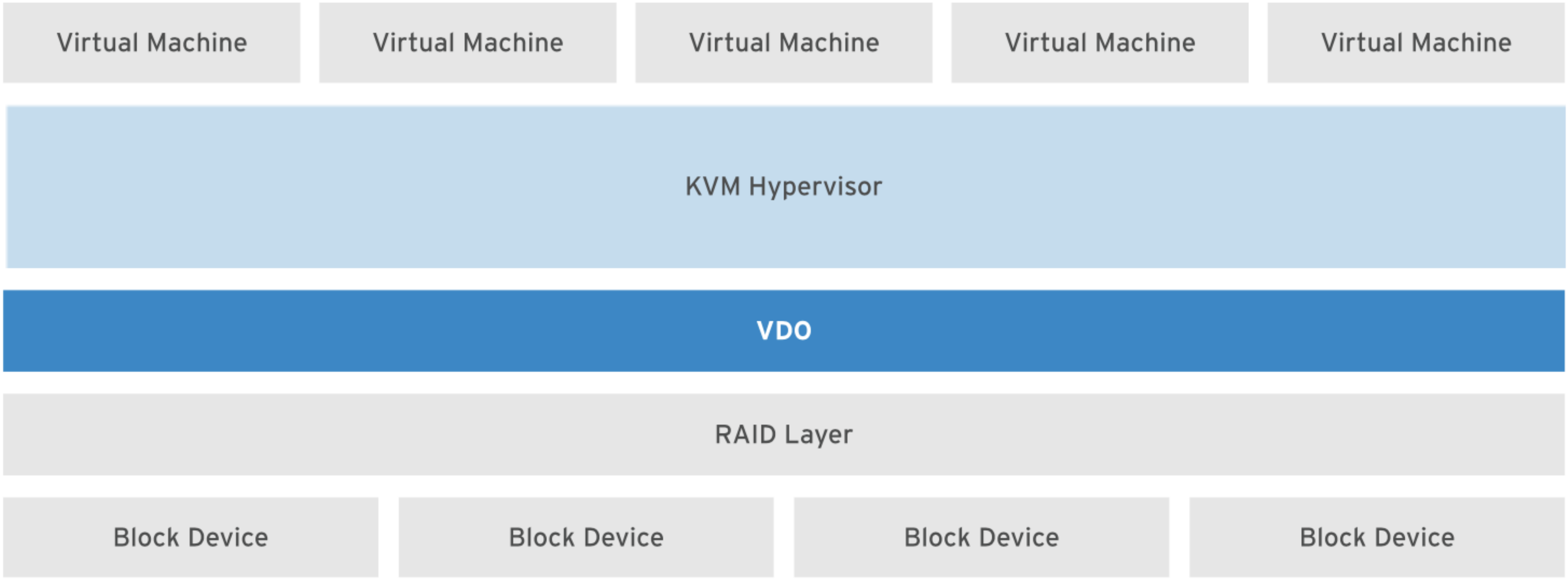
After completing this unit, you should be able to:

- Understand Virtual Data Optimizer
- Implement Virtual Data Optimizer

Understand Virtual Data Optimer

- RHELv8 includes VDO driver
- Implement data efficiency on block devices
- Device Mapper to reduce disk space usage
- Minimizes replication of data
- Increased data throughput
- Cost saving
- Two key function in kernel
  - uds: Deduplication
  - kvdo module : Compress

VDO-based virtual machines



Virtual Machine

Virtual Machine

Virtual Machine

Virtual Machine

Virtual Machine

KVM Hypervisor

VDO

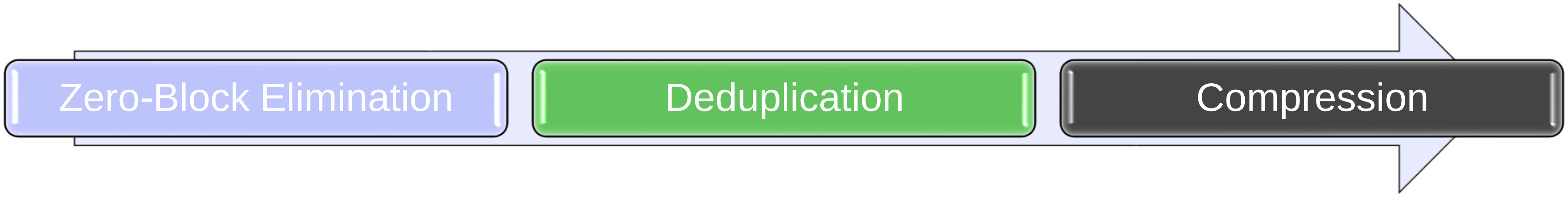
RAID Layer

Block Device

Block Device

Block Device

Block Device



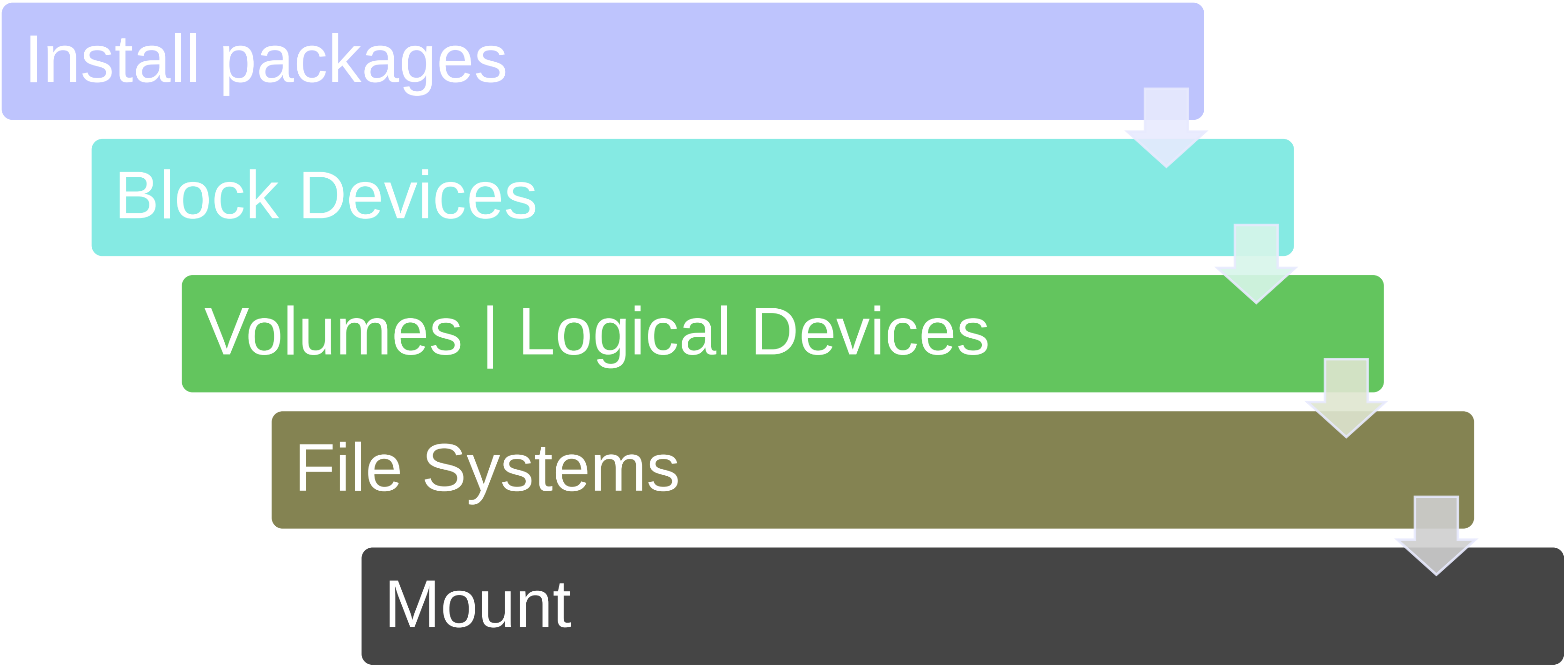
Zero-Block Elimination

Deduplication

Compression

Implementing Virtual Data Optimizer

- VDO volumes can be thin provisioned
- By default 1:1 ratio
- Logical size > physical size = TP
- Keep monitor volume consumption



Install packages

Block Devices

Volumes | Logical Devices

File Systems

Mount

Enable VDO and create some volumes

- Enabling VDO

```
# dnf –y install vdo kmod-kvdo
```

Create a VDO volume

```
# vdo create --name=vdo1 --device=/dev/vdc --vdoLogicalSize=100G
# vdo create --name=vdo2 --device=/dev/vdd --vdoLogicalSize=100G
```

List VDO volumes

```
# vdo list
```

Verify compression and deduplication is enabled

```
# vdo status -a | egrep –i "vdo|compression|deduplication"
# udevadm settle
```

Create FS and mount the VDO volume

- Create FS

```
# mkfs –K /dev/mapper/vdo1
# mkfs.xfs /dev/mapper/vdo2
```

- Mount the FS

```
# mkdir /mnt/vdo1 /mnt/vdo2
# mount /dev/mapper/vdo1 /mnt/vdo1
# mount /dev/mapper/vdo2 /mnt/vdo2
```

- Verify

```
# vdostats --human-readable --si
Device              Size     Used    Available Use%    Space saving%
/dev/mapper/vdo2    21.5G    4.3G    17.2G     20%     N/A
/dev/mapper/vdo1    21.5G    4.3G    17.2G     20%     99%
```

Test the deduplication and compression

- Copy /boot/vmlinuz-*

```
# mkdir /mnt/vdo1/copy1 /mnt/vdo1/copy2 /mnt/vdo1/copy3
# cp /boot/initramfs-* /mnt/vdo1/copy1
# cp /boot/initramfs- * /mnt/vdo1/copy2
# cp /boot/initramfs- * /mnt/vdo1/copy3
```

- Verify

```
# vdostats --human-readable --si
Device                 Size      Used Available Use% Space saving%
/dev/mapper/vdo2       21.5G     4.3G     17.2G  20%          N/A
/dev/mapper/vdo1       21.5G     4.3G     17.2G  20%          99%
```

Test the deduplication and compression

- Verify

```
# vdostats --human-readable --si
Device                  Size      Used Available Use% Space saving%
/dev/mapper/vdo2          21.5G      4.3G     17.2G  20%          N/A
/dev/mapper/vdo1          21.5G      4.3G     17.2G  20%          99%
```

- Copy /boot/vmlinuz-*

```
# mkdir /mnt/vdo1/copy1 /mnt/vdo1/copy2 /mnt/vdo1/copy3
# cp /boot/initramfs-* /mnt/vdo1/copy1
# cp /boot/initramfs- * /mnt/vdo1/copy2
# cp /boot/initramfs- * /mnt/vdo1/copy3
# vdostats --human-readable --si
```

Enabling/Disable Compressions

- To Check

```
# vdo status –n vdo1 | grep –i compress
```

- To disable compression

```
# vdo disableCompression –n vdo1
```

- To re-enable compression

```
# vdo enableCompression –n vdo1
```

Enabling/Disable Deduplication

- To Check

```
# vdo status –n vdo1 | grep –i deduplication
```

- To disable compression

```
# vdo disableDeduplication –n vdo1
```

- To re-enable compression

```
# vdo enableDeduplication –n vdo1
```

Remove vdo

- Unmount first!

- Remove individual volume

```
# vdo remove –n vdo1
```

- or remove all volumes

```
# vdo remove –a
```

- Confirm

```
# vdo list
```

Checkpoint

1. Users are complaining that each time they access files in the server, it lags. You suspected its due to VDO volumes. What should you do to speed up user's uploading for now?
   a) systemctl stop vdo; reboot
   b) systemctl stop --now vdo
   c) vdo disableCompression -a
   d) vdo enableCompression -a

2. Which statements are true about # vdo create --name=vdo1 --device=/dev/sdc --vdoLogicalSize=120G? PS: /dev/sdc is SAS-connected SSD drive with 20G in size.
   a) The vdo command create a VDO logical volume with compression and deduplication enabled
   b) The vdo command create a VDO logical volume with 6:1 efficiency ratio
   c) The vdo command create a VDO logical volume with 1:6 efficiency ratio
   d) The vdo command create a VDO logical volume with compression and deduplication disabled

3. How do you disable compression on vdo2?
   a) sysemctl disableCompression vdo2
   b) systemctl disableCompression -n vdo2
   c) vdo disableCompression vdo2
   d) vdo disableCompression -n vdo2

4. True or False: VDO can be enabled on RHEL v8 onward only

Checkpoint

1. Users are complaining that each time they access files in the server, it lags. You suspected its due to VDO volumes. What should you do to speed up user's uploading for now?
   a) systemctl stop vdo; reboot
   b) systemctl stop --now vdo
   c) vdo disableCompression -a
   d) vdo enableCompression -a

2. Which statements are true about # vdo create --name=vdo1 --device=/dev/sdc --vdoLogicalSize=120G? PS: /dev/sdc is SAS-connected SSD drive with 20G in size.
   a) The vdo command create a VDO logical volume with compression and deduplication enabled
   b) The vdo command create a VDO logical volume with 6:1 efficiency ratio
   c) The vdo command create a VDO logical volume with 1:6 efficiency ratio
   d) The vdo command create a VDO logical volume with compression and deduplication disabled

3. How do you disable compression on vdo2?
   a) sysemctl disableCompression vdo2
   b) systemctl disableCompression -n vdo2
   c) vdo disableCompression vdo2
   d) vdo disableCompression -n vdo2

4. True or False: VDO can be enabled on RHEL v8 onward only

Unit summary

Having completed this unit, you should be able to:

- Understand Virtual Data Optimizer
- Implement Virtual Data Optimizer