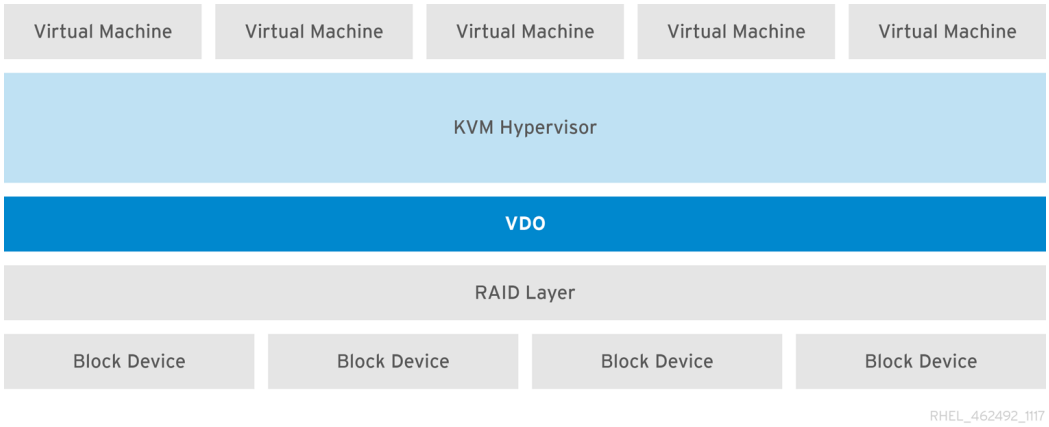
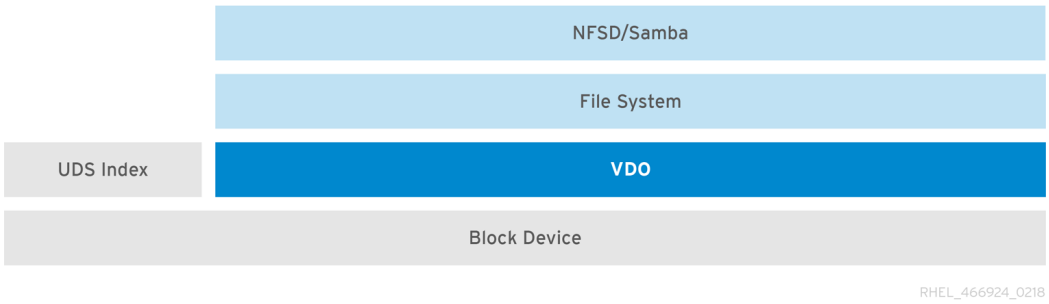
RHEL환경에서 VDO 사용하기

VDO 소개

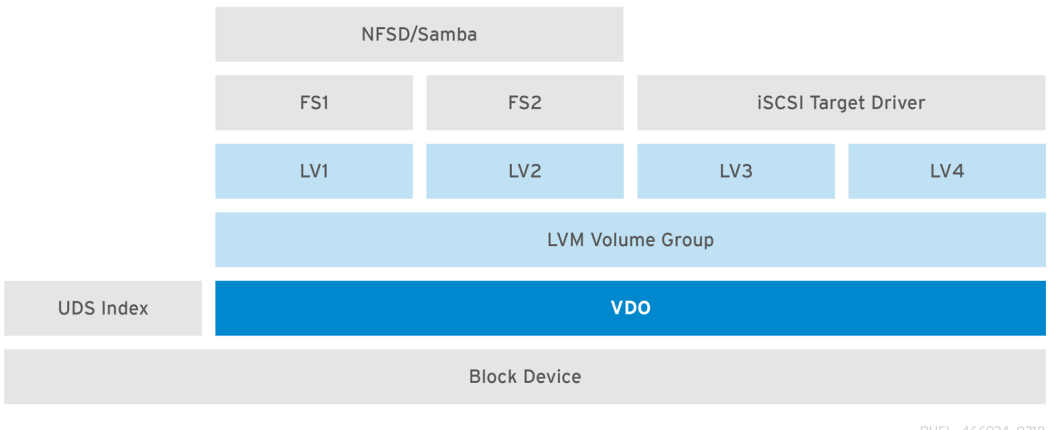
1. VDO; Virtual Data Optimizer 기술은 스토리지의 활용을 증가시키기 위해 데이터 중복제거, 압축 기능을 제공하는 스토리지 기술
2. 활용 목적에 따른 아키텍쳐
    1. VM호스트 기반의 아키텍쳐



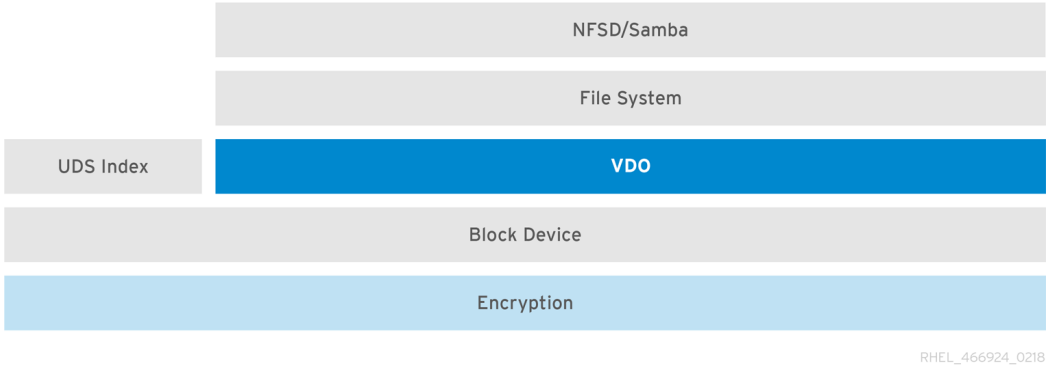
    블록 디바이스 상단에 VDO 디스크 생성 후 하이퍼바이져를 통해 VM image 저장
    2. NFS 활용한 VDO 아키텍쳐



    VDO디스크 생성 후 파일시스템 생성. NFS나 samba로 export 수행
    3. VDO기반의 LVM구성



    VDO디스크 생성 후 LV생성 후 파일시스템 생성. 필요시 NFS / samba로 export 수행
    4. 암호화 적용시 VDO구성



암호화 적용시 VDO하단에 암호화 알고리즘이 적용되어야 하기 때문에 중복제거의 큰 효과는 없음

3. 스토리지 스택별 VDO 구성
    1. VDO 아래 stack에 배치할 component
        1. Software RAID / DM Multi path
    2. VDO 상단 stack
        1. LVM cache / snapshot / thin provisioning
    3. unsupport stack
        1. VDO 상단에 다른 볼륨 구성, LVM 상단에 구성되는 VDO,
4. 장점
    1. 공간활용도 증가
        1. VM호스트 머신으로 운영시 물리/논리의 용량 비율은 약 1:10 비율로 활용 가능
           (1TB디스크로 중복제거와 압축 기능을 화용해 10TB까지 사용 가능)
        2. Ceph기반의 경우 1:3 비율로 용량 활용 가능
5. 단점
    1. 암호화 / 중복제거 적용시 I/O 성능 저하

시스템 요구사항

1. 메모리
    1. VDO모듈 (실제로 점유하는 메모리 용량은 i + ii + iii)
        1. 38MB의 고정 메모리 할당 + 블록 맵 크기 150MB 이상 필요
        2. logical size 1TB당 1.6MB 할당
        3. Physical size 1TB 당 268MB 할당
    2. UDS Size
        1. 기본 250MB 필요
        2. 인덱스 유형별 메모리

인덱스 유형	중복제거용 용량	비 고
Dense	1GB메모리당 당 1TB디스크	일반적으로 4TB 물리디스크에 1GB 인덱스 구성도 충분
Sparse	1GB메모리당 당 10TB디스크	권장 모드, 일반적으로 40TB 물리디스크에 1GB 인덱스 구성도 가능

2. 스토리지
    1. physical size는 최대 256TB 까지 구성 가능
    2. VDO 메타데이터와 UDS 인덱스 구성용 스토리지 구성
    3. 메타 데이터는 스토리지 4GB당 1MB를 저장하고, slabe당 1MB씩 추가로 저장
    4. 일반적으로 UDS 인덱스 구성시 dense는 1GB 메모리가 17GB스토리지를 사용, sparse는 170GB 스토리지를 사용

컴포넌트 리스트

1. kvdo : VDO를 사용하기 위한 커널 모듈
2. uds : VDO기반에서 인덱싱 후 중복데이터를 분석하는 커널 모듈, 새로운 데이터 저장시 이전에 저장된 데이터와 동일한지 검색 후 동일한 데이

터를 두번 저장하지 않도록 참조 구성
3. cli : 스토리지 구성 및 관리 수행

스토리지 용량 관리

1. physical size : VDO를 사용하기 위한 블록장치, 메타데이터 크기를 뺀 값만큼의 용량을 사용
2. logical size : VDO볼륨으로 셍성된 크기, 일반적으로는 physical size보다 크게 설정하면 되나, 기본값은 1:1 비율로 생성, logical 최대 사이즈는 4PB까지 사용가능
3. slab size : VDO볼륨에서 여러개 slab으로 분할 관리. 기본 slab크기는 2GB씩 최대 8192개의 Slab이 생성될 수 있음
   physical size별 권장 slab 크기

Physical Size	Slab Size
10 ~ 99GB	1GB
100GB ~ 1TB	2GB
2 ~ 256TB	32GB

slab사이즈는 vdo 생성시 --vdoSlabSize={{ size }} 옵션으로 생성 가능

VDO 구축

1. VDO 모듈 설치

```
$ yum install vdo kmod-kvdo
$  lsmod  | grep vdo
kvdo               577536  0
uds                253952  1 kvdo
dm_mod             151552  12 kvdo,dm_thin_pool,dm_bufio
```

2. VDO 볼륨 생성

```
$ vdo create --name= {{ VDO 이름 }} --device={{ 디스크 경로 }} --vdoLogicalSize= {{ logical siz }}
```

3. 파일시스템 생성
   #xfs로 생성시

```
$ mkfs.xfs -K /dev/mapper/{{ VDO 이름 }}
```

#ext4로 생성시

```
# mkfs.ext4 -E nodiscard /dev/mapper/{{ VDO 이름 }}
```

4. /etc/fstab에 파일시스템 마운트
   # xfs로 마운트시

```
$ /dev/mapper/{{ VDO 이름 }} {{ 마운트 경로 }} xfs defaults,x-systemd.device-timeout=0,x-systemd.requires=vdo.service 0 0
```

#ext4로 마운트시

```
$ /dev/mapper/{{ VDO 이름 }} {{ 마운트 경로 }} ext4 defaults,x-systemd.device-timeout=0,x-systemd.requires=vdo.service 0 0
```

VDO 운영

1. VDO 볼륨 관리

```
$ vdo start --name= {{ VDO 이름 }} or vdo start --all
$ vdo stop --name= {{ VDO 이름 }} or vdo stop --all
```

2. VDO 볼륨 활성화
   #볼륨 활성화

```
$ vdo active --name {{ VDO 이름 }} or vdo active --all
```

   \\#볼륨 비활성화

```
$ vdo deactivate --name {{ VDO 이름 }} or vdo deactive ~-~-all
```

   기본적으로 OS부팅시 VDO볼륨 활성화 수행 (vdo create 시 ~-~-activate=disabled 옵션을 추가하면 자동 활성화 안함)
3. vdo볼륨 제거

```
$> vdo remove --name {{ VDO 이름 }} or vdo remove --all
```

4. VDO에서 사용하지 않는 블록 삭제

```
$ systemctl enable --now fstrim.timer
```

5. VDO 사용현황 확인

```
$ vdostats ~-~-human-readable
\\Device                1K-blocks  Used   Available  Use%  Space saving%
/dev/mapper/node1osd1   926.5G     21.0G  905.5G     2%    73%
/dev/mapper/node1osd2   926.5G     28.2G  898.3G     3%    64%
```

6. vdo 볼륨 크기 증가

```
$> vdo growLogical --name={{ VDO 이름 }} --vdoLogicalSize= {{ 변경할 크기 }}
```

reference

1. *https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/vdo-quick-start*
2. *https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/deduplicating_and_compressing_storage/deploying-vdo_deduplicating-and-compressing-storage*

Revision #3
★Created 7 June 2022 14:43:56 by artop0420
Updated 24 December 2023 02:30:29 by artop0420