



ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование

Меня хорошо видно && слышно?

Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

НЕ ЗАБЫТЬ ВКЛЮЧИТЬ
ЗАПИСЬ!!!

Файловая система ZFS

Правила вебинара

- Активно участвуем: выполняем задания, отвечаем на вопросы
- Если возникли сложности задаем вопрос в чат
- На вопросы постараюсь отвечать сразу, но возможны паузы

После занятия вы сможете

1. Перечислить технологии которые используются в ZFS
2. Создать pool и файловую систему ZFS
3. Выбрать вариант pool по скорости и по избыточности
4. Выполнять базовые действия с файловой системой

Зачем вам это уметь

ВАШ ВАРИАНТ?

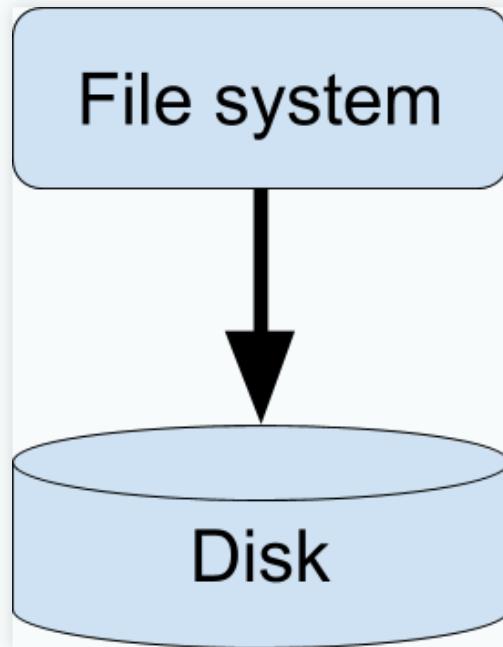
Зачем вам это уметь

МОЙ ВАРИАНТ

1. Добавить инструментов в арсенале файловых систем
2. Повысите отказоустойчивость дисковой подсистемы
3. Меньше волнения при замене вылетевшего диска

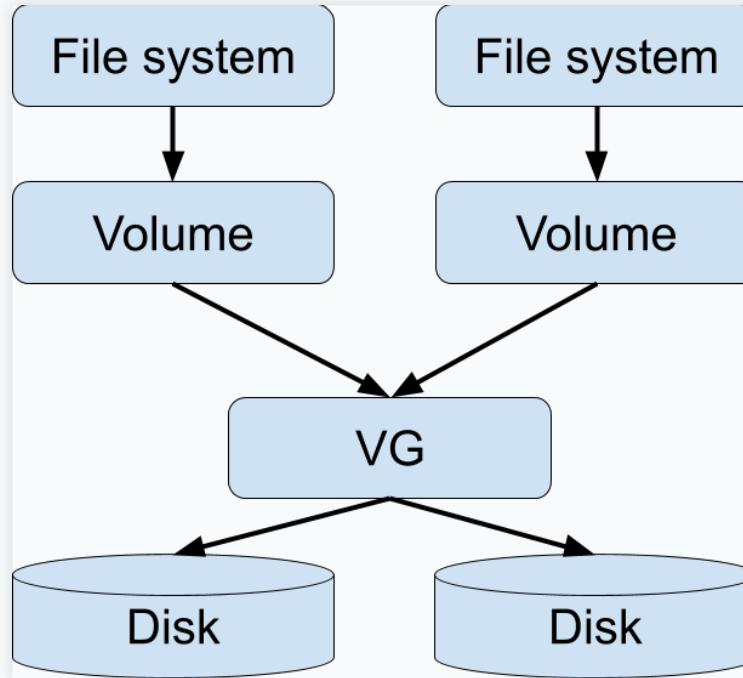
Модель хранения данных

- ФС -> одно физическое устройство



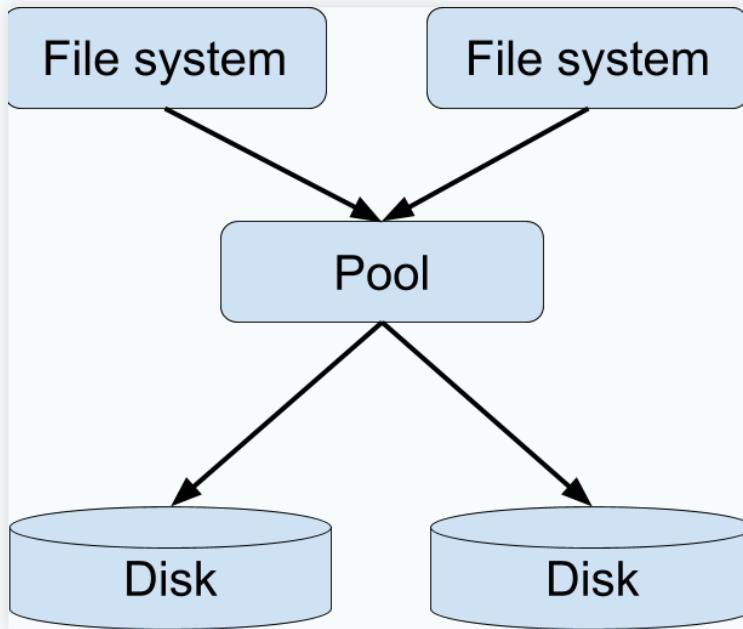
Модель хранения данных LVM

- ФС -> диспетчер томов (LVM) -> физические устройства



Модель хранения данных ZFS

- ФС -> pool of disks -> нескольких физических устройств



Компоненты ZFS

ZFS POSIX Layer

ZFS Volume
Emulator

Data Management Unit (DMU)

Storage Pool Allocator (SPA)

SPA Storage Pool Allocator

Пул устройств хранения данных

- описывает физические характеристики хранения
 - размещение устройств
 - избыточность данных
- выступает в качестве хранилища данных

DMU Data management unit

- предоставляет транзакционную модель поверх SPA
 - транзация это серия действий которые записываются на диск группой
- клиенты DMU работают с объектами, наборами объектов, транзакциями
 - ZFS POSIX Layer
 - ZFS Volume Emulator

ZFS vs RAID, LVM, ext4

Перечислить команды чтобы создать файловую систему ext4/LVM/RAID ? Сколько команд?

ZFS vs RAID, LVM, ext4

```
mdadm -C /dev/md0 -l 0 -n 4 /dev/sde /dev/sdf /dev/sdg /dev/sdh  
pvcreate /dev/md0  
vgcreate /dev/md0 tank  
lvcreate -l 100%FREE -n videos tank  
mkfs.ext4 /dev/tank/videos  
mkdir -p /tank/videos  
mount -t ext4 /dev/tank/videos /tank/videos
```

СКОЛЬКО НУЖНО КОМАНД ЧТОБЫ СОЗДАТЬ ФАЙЛОВУЮ СИСТЕМУ?

ZFS vs RAID, LVM, ext4

```
zpool create tank mirror sde sdf mirror sdg sdh  
zfs create tank/test2
```

СКОЛЬКО НУЖНО КОМАНД ЧТОБЫ СОЗДАТЬ ФАЙЛОВУЮ СИСТЕМУ?

Целостность данных (data integrity)

- CoW copy-on-write
 - никогда не перезаписывает данные
 - состояния на диске всегда корректные
 - **следствие:** нет fsck
- транзакционность (DMU)
 - связанные действия обрабатываются целиком
 - **следствие:** не нужен журнал
- Контрольные суммы (checksums, Merkle tree)
- избыточность данных
 - raidz
 - ditto blocks. metadata copy (2 default. 1 per disk)

Избыточность RAIDz

Динамический размер страйпа

RAID-5			
A1	A2	A3	Ap
B1	B2	Bp	B3
C1	Cp	C2	C3
Dp	D1	D2	D3
E1	E2	E3	Ep
F1	F2	Fp	F3
G1	Gp	G2	G3
Hp	H1	H2	H3

RAID-Z1			
A1	A2	A3	Ap
A4	A5	A6	Ap'
B1	B2	Bp	C1
C2	C3	Cp	D1
Dp	E1	E2	E3
Ep	E4	Ep'	F1
Fp	G1	G2	G3
Gp	H1	H2	Hp

Кеширование чтения

- Adaptive Replacement Cache (ARC)
 - кэш в памяти (занимает до 50% от доступной RAM)
- Layer-2 Adaptive Replacement Cache » (L2ARC)
 - кэш на диске (быстрые диски SSDs или SAS 15k)
 - гибридный кэш большого размера
- Кэширует то что недавно читалось и часто читается
 - увеличение производительности чтения

Кеширование записи

- ZFS Intent Log (ZIL)
 - журнал записей (используется для sync записи для хранения транзакций)
 - обеспечивает целостность данных
- Separate Intent Log » (SLOG)
 - рекомендуется размещать на быстрых дисках SSDs или SAS 15k
 - используется только для синхронных операций записи
 - увеличивает скорость записи

Время практики.
Установка ZFS.
Добавляем pool

Установка ZFS on Linux

- Какие ОС поддерживаются?
 - FreeBSD, illumos, Linux
- Какие дистрибутивы Linux поддерживаются?
 - RHEL/CentOS ... ?
 - Открыть сайт zfsonlinux.org
- Репозиторий демо-окружения
 - <https://github.com/nixuser/zfs>

Установка в RHEL based OS

kABI-tracking kmod

- собранные бинарные пакеты
- подходит в большинстве случаев

DKMS

- сборка из исходников при обновлении
- для собственной сборки ядра kernel

Шаги установки в RHEL based OS

```
dnf install -y \
https://zfsonlinux.org/epel/zfs-release-2-2$(rpm --eval "%{dist}")
dnf config-manager --disable zfs
dnf config-manager --enable zfs-kmod
dnf install -y zfs
modprobe zfs
```

Создание пула устройств хранения данных ZFS

- Определим диски
- Выберем тип репликации

```
man zpoolconcepts  
zpool create poolmirror mirror sdb sdc
```

```
echo disk{1..6} | xargs -n 1 fallocate -l 500M  
zpool create stripe $PWD/disk[1-5]  
zpool create mir mirror $PWD/disk[1-5]  
zpool create raid raidz1 $PWD/disk[1-3]  
zpool create raid raidz2 $PWD/disk[1-2]  
zpool create raid raidz3 $PWD/disk[1-3] # less disks  
zpool create raid raidz3 $PWD/disk[1-5]
```

Скорость от типа vdev

- RAID-0 (fastest)
- RAID-1
- RAIDZ-1
- RAIDZ-2
- RAIDZ-3 (slowest)

```
zpool create hybrid mirror $PWD/disk[12] mirror $PWD/disk[34]
zpool create hybrid2 raidz2 $PWD/disk[1-3]  raidz2 $PWD/disk[4-6]
```

Размер сектора. Параметр ashift

- Устанавливается при создание vdev
- Значение ashift это степень двойки
 - $2^9 = 512$ байт
 - $2^{12} = 4,096$ байт - это рекомендуемое значение
 - 2^{13} устанавливается для дисков SSD с размером сектора 8К

```
zpool create -o ashift=12 tank mirror sda sdb
```

<https://openzfs.github.io/openzfs-docs/Performance%20and%20Tuning/Workload%20Tuning.html>

Создание pool с SLOG и L2LARC

- навязчивое повторение что такое SLOG ? и ARC? L2ARC?
Создать сразу

```
zpool create storage mirror sdb sdc \
    log mirror nvme0n1 nvme0n2 \
    cache nvme0n3 nvme0n4
```

Добавить позже

```
zpool create storage mirror sdb sdc
zpool add storage cache nvme0n3 nvme0n4 # L2ARC
zpool add storage log mirror nvme0n1 nvme0n2 # SLOG
```

Время создавать файловую систему

Создание dataset

Файловая система создается поверх пула (в терминологии ZFS это **dataset**)

```
zfs create storage/userdir  
zfs create storage/data
```

ФС Может быть вложенной

```
zfs create storage/data/video  
zfs create storage/data/music
```

Параметры монтирования

Что примонтировано?

```
mount  
zfs get mounted
```

По умолчанию файловая система монтируется в директорию пула. Можно поменять через параметр: `mountpoint`

```
zfs set mountpoint=/home/testuser storage/data/music
```

директория должна быть пустой

Дисковые квоты (Quotas)

По умолчанию файловая система (dataset) займет все предоставленное место

```
dd if=/dev/urandom of=/home/testuser/file bs=1M  
df -h  
zfs list
```

Квоты устанавливают лимит на количество данных в ФС

```
zfs get quota  
zfs set quota=500M storage/data/media
```

Резерв (Reservations)

Гарантирует доступное место файловой системе

```
zfs list # before
zfs set reservation=5G storage/data/video
zfs list # after
```

Параметр размер блока

```
zfs create storage/data/movies
zfs create storage/data/torrents
zfs set recordsize=1M storage/data/movies
zfs set recordsize=64K storage/data/torrents
```

Наследование параметров

```
zfs set checksum=sha256 storage/data/movies
zfs get checksum
zfs set checksum=skein storage/data
zfs get checksum
```

Параметры кэширования

```
zfs set primarycache={all|metadata|none} # ARC  
zfs set secondarycache={all|metadata|none} # L2ARC
```

Дедупликация vs Сжатие

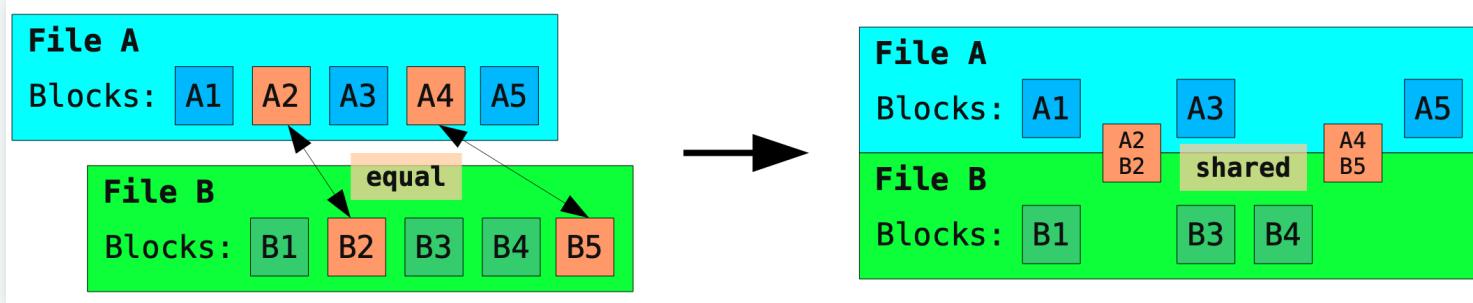
Сжатие

- ZFS может сжимать данные при записи на файловую систему и разжимать их при чтении
- только новые данные сжимаются. Старые не перепаковываются

Получим файлы для проверки и установки сжатие

```
zfs create storage/src  
zfs create storage/src/compressed  
zfs set compression=on storage/src/compressed  
zfs get compression,compressratio
```

Дедупликация



Дедупликация vs Сжатие

```
zfs create storage/src/dedup
zfs get dedup storage/src/dedup
zfs set dedup=on storage/src/dedup
dd if=/dev/sda of=file bs=1M count=10
cp file /storage/src/dedup/file1
cp file /storage/src/dedup/file2
cp file /storage/src/dedup/file3
zpool list
```

А другие ФС так могут?

Пример переноса дисков между хостами

Перед переносом диски нужно отключить

```
zpool export storage  
zpool status
```

На соседнем хосте (пример из файлов поэтому -d)

```
zpool import -d ${PWD}/zpoolexport/  
zpool import -d ${PWD}/zpoolexport/ storage
```

Пример работы со snapshot

Создать снимок

```
zfs snapshot storage/data/music@snap001
```

Список снимков

```
zfs list -t snapshot
```

Удалить снимок

```
zfs destroy storage/data/music@snap001
```

Восстанавливаем файлы из snapshot

Заполняем данными

```
zfs create storage/text  
cp War_and_Peace.txt /storage/text/
```

Создаем снимок

```
zfs snapshot storage/text@copy001  
rm /storage/text/War_and_Peace.txt
```

Восстанавливаем файлы из снимка

```
zfs rollback storage/text@copy001  
ls /storage/text
```

Пример переноса snapshots между хостами

Хост номер 1

```
zfs send storage/text@copy001 > snapshot
```

Хост номер 2

```
zfs receive storage/data/text2 < snapshot
```

Восстановление

Диск вышел из строя

```
dd if=/dev/zero of=/dev/sdb  
zpool status
```

Запустим процедуру проверки

```
zpool scrub  
zpool status
```

Заменим сбойный диск

```
zpool replace storage sdb sdd
```

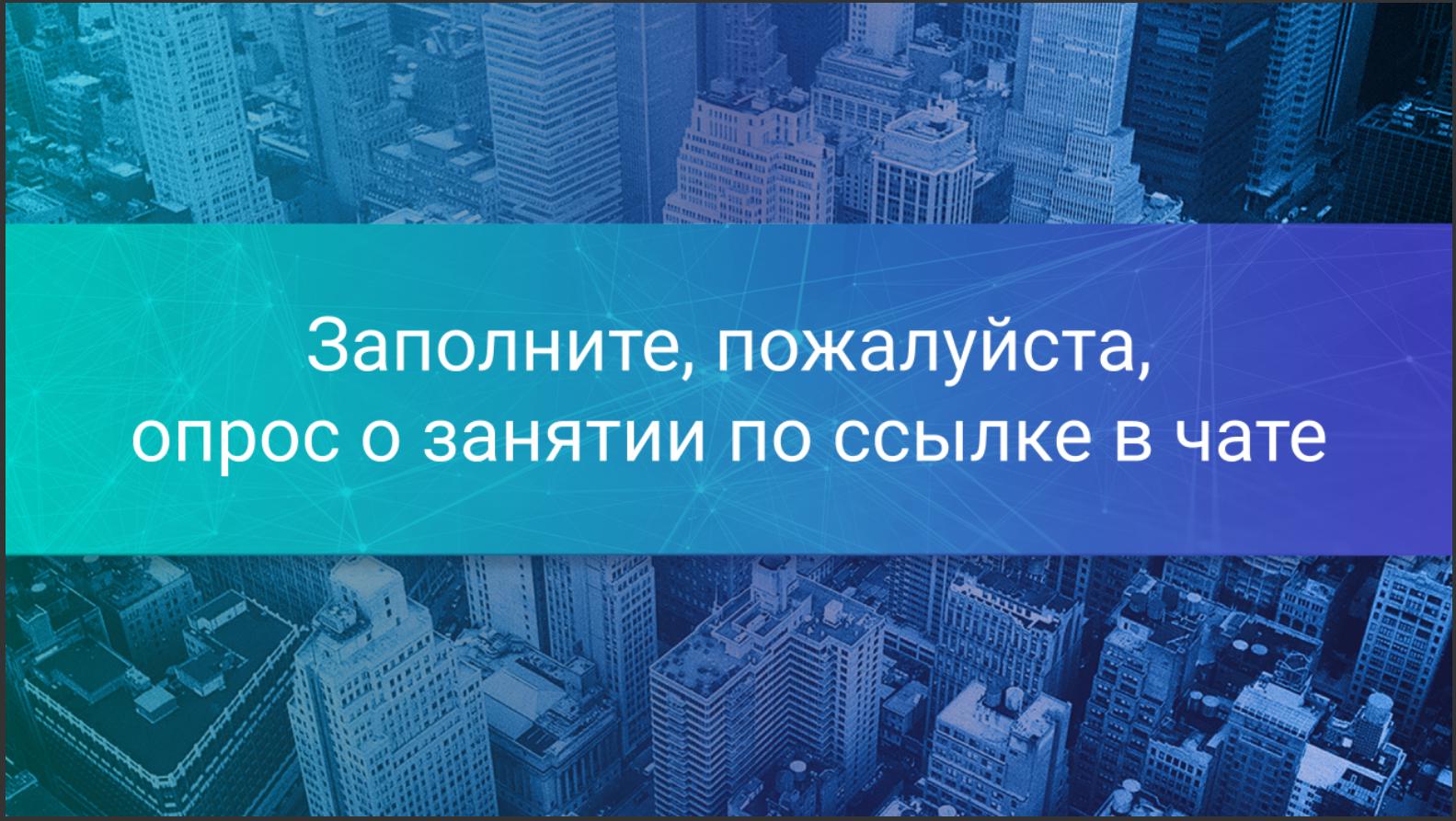
Рефлексия



Отметьте 3 пункта, которые вам запомнились с вебинара



Что вы будете применять в работе из сегодняшнего вебинара?



Заполните, пожалуйста,
опрос о занятии по ссылке в чате

