



# Онлайн-образование



# Меня хорошо видно && слышно?

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



НЕ ЗАБЫТЬ ВКЛЮЧИТЬ  
ЗАПИСЬ!!!

# Linux High Availability. Distributed Replicated Block Device (DRBD).

# Цель занятия

- Познакомиться с технологией Distributed Replicated Block Device (DRBD)

# План занятия

- DRBD - концепция и сфера применения;
- Организация блочного хранилища;
- Построение HA NFS-хранилища поверх DRBD.

# DRBD

DRBD - ПО для построения системы хранения данных путём зеркалирования содержимого блочных устройств между несколькими хостами.

- в реальном режиме времени;
- прозрачно для приложений;
- синхронно или асинхронно, на выбор.

# DRBD

Поставляется в виде модуля для ядра и комплекта управляющих утилит. Представляет собой драйвер виртуального блочного устройства.

- работает быстро;
- безразличен к тому, что находится "выше" (файловые системы, приложения и т.п.).

```
yum install https://www.elrepo.org/elrepo-release-7.el7.elrepo.no
yum install kmod-drbd90 drbd90-utils
```

# DRBD. Ресурсы и роли

Ресурс - описание реплицируемого набора данных.

- имя ресурса;
- список томов для репликации (данные и мета-данные);
- виртуальное блочное устройство DRBD;
- описание канала связи между хостами.

Single-primary mode. Каждому ресурсу назначается роль из двух возможных вариантов:

- Primary (вручную, только один на каждый ресурс);
- Secondary.

Dual-primary возможен, но только при определённых сценариях с Distributed Lock Manager (к примеру, с файловыми системами GFS/GFS2 или OCFS2) и при 100% соблюдении некоторых условий. В DRBD 9.0.x сломан, о чём разработчики сообщают в документации.

# DRBD. Протоколы репликации

## **Protocol A - асинхронный.**

Данные записаны локально и помещены в буфер TCP. Могут быть недоставлены последние обновления на Secondary. Часто используется для репликации на большие расстояния в сочетании с DRBD Proxy (кеш и сжатие).

## **Protocol B - полусинхронный.**

Данные записаны локально и получено подтверждение о получении. Может быть рассогласованность при одновременной потере питания на всех узлах.

## **Protocol C - синхронный.**

Данные записаны локально и получено подтверждение о записи на удалённом узле.

# DRBD. Синхронизация

При первом запуске, добавлении устройства или потере связи нужно произвести **синхронизацию**.

- линейный процесс, а не в порядке изменения блоков;
- скоростью можно управлять и даже динамически приостанавливать;
- доступна оптимизация с помощью хеширования;
- доступна проверка на целостность;

# DRBD. Разное

- DRBD 9 поддерживает до 16 узлов. В DRBD 8.3+ было меньше, но поддерживалось стекирование.
- Сеть в DRBD 9 строится по принципу "все со всеми" (connection-mesh).
- В качестве транспортного протокола возможно использовать TCP over IPv4/IPv6 и RDMA.
- Есть варианты автоматического решения проблемы split-brain.
- Автоматически догоняет реплики со статусом Outdated до UpToDate.
- Поддержка disk flush, TRIM. Обработка I/O errors блочного устройства.
- Есть своя простая поддержка кворума и фенсинга на скриптах (без pacemaker).

# DRBD. Конфигурация и управление

```
man drbd.conf  
man drbdadm  
lsmod | grep drbd  
modinfo drbd  
semanage permissive -a drbd_t  
systemctl status drbd.service  
firewall-cmd --zone=public --permanent --add-port=7788/tcp  
vi /etc/drbd.conf  
vi /etc/drbd.d/global_common.conf  
vi /etc/hosts  
vi /etc/drbd.d/demo.res
```

# DRBD. Конфигурация и управление

```
resource demo {
    protocol C;
    volume 0 { device /dev/drbd0; disk /dev/sdb; meta-disk internal; }
    volume 1 { device /dev/drbd1; disk /dev/sdc; meta-disk internal; }
    on drbd-node01 {
        node-id 0;
        address 192.168.11.101:7788; }
    on drbd-node02 {
        node-id 1;
        address 192.168.11.102:7788; }
    on drbd-node03 {
        node-id 2;
        address 192.168.11.103:7788;
        volume 1 { disk none; } }
    connection-mesh {
        hosts drbd-node01 drbd-node02 drbd-node03;
        net {
            use-rle yes;
            csums-alg sha1;
            data-integrity-alg sha1; } } }
```

# DRBD. Конфигурация и управление

```
drbdadm create-md demo
drbdadm up demo
drbdadm status demo
drbdadm primary demo --force
drbdadm secondary demo
drbdadm down demo
```

# DRBD и Pacemaker

Ресурсный агент ocf:linbit:drbd будет заниматься назначением Primary.

```
pcs resource create DRBD ocf:linbit:drbd \
    drbd_resource=demo op monitor interval=10s

pcs resource master DRBDClone DRBD notify=true \
    master-max=1 master-node-max=1 clone-node-max=1

pcs resource create DRBDMount Filesystem fstype=xfs \
    device=/dev/drbd0 directory=/mnt

pcs constraint colocation add DRBDMount with DRBDClone
    INFINITY with-rsc-role=Master

pcs constraint order promote DRBDClone then start DRBDMount
```

# DRBD, Pacemaker и NFS

Логика работы кластера высокодоступного файлового хранилища:

- Выбираем DRBD Primary на одном из узлов;
- Монтируем файловую систему на выбранном узле;
- Перемещаем туда же "плавающий IP-адрес";
- Запускаем там сервисы NFS;
- Экспортируем примонтированный каталог в NFS.

В случае проблем: DRBD Primary автоматически переезжает на другой узел, а кластер производит "огораживание" сбояного узла.

# DRBD, Pacemaker и NFS. Пример

```
pcs resource create DRBD ocf:linbit:drbd drbd_resource=demo
pcs resource master DRBDClone DRBD \
    master-max=1 master-node-max=1 clone-node-max=1 notify=true
pcs resource create DRBDMount Filesystem device=/dev/drbd0 directory=/mnt fstype=xfs
pcs resource create NFSAddress ocf:heartbeat:IPaddr2 ip=192.168.11.111 cidr_netmask=24
pcs resource create NFSDaemon systemd:nfs-server
pcs resource clone NFSDaemon
pcs resource create NFSEexport ocf:heartbeat:exportfs \
    fsid=0 directory=/mnt options=rw clientspec=192.168.11.0/24
pcs resource clone NFSEexport
pcs constraint colocation add DRBDMount with DRBDClone INFINITY with-rsc-role=Master
pcs constraint colocation add NFSAddress with DRBDClone INFINITY with-rsc-role=Master
pcs constraint colocation add NFSDaemon-clone with DRBDClone INFINITY with-rsc-role=Master
pcs constraint colocation add NFSEexport-clone with DRBDClone INFINITY with-rsc-role=Master
pcs constraint order promote DRBDClone then start DRBDMount
pcs constraint order start DRBDMount then start NFSAddress
pcs constraint order start NFSAddress then start NFSDaemon-clone
pcs constraint order start NFSDaemon-clone then start NFSEexport-clone
```

# 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

