Курс «Администратор Linux»

Linux HA. iSCSI, GFS2

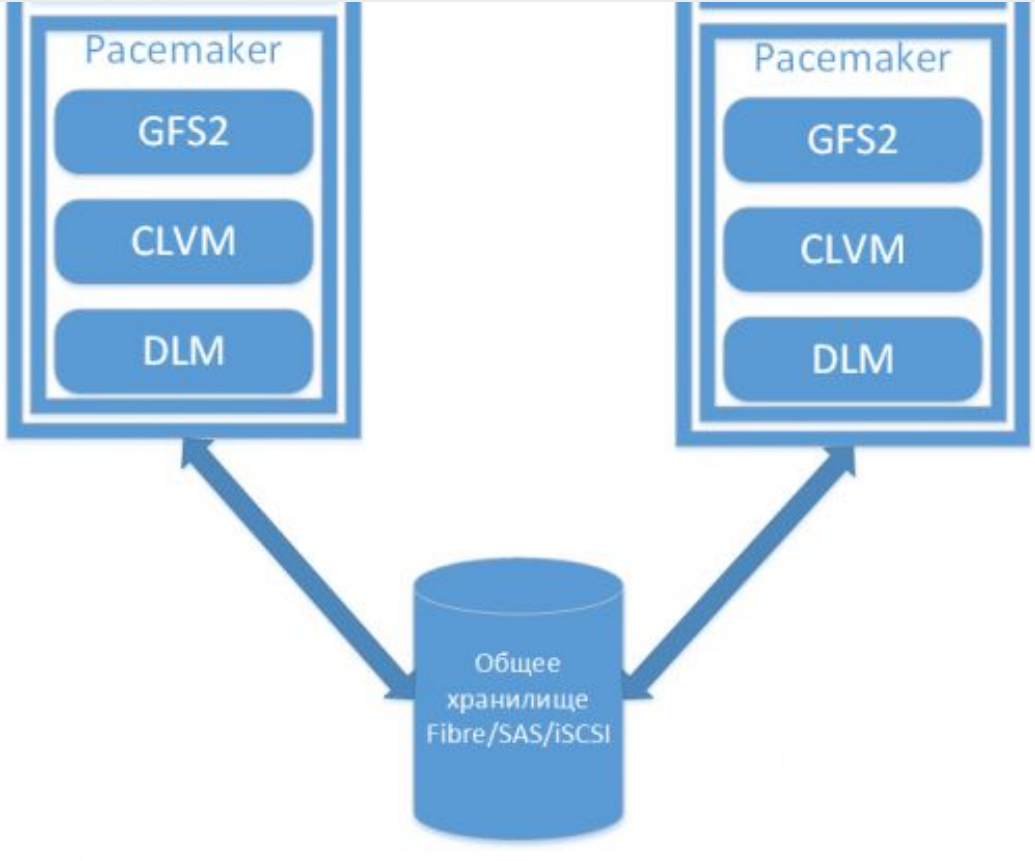
Занятие # 35

Алексей Цыкунов

Сегодня разберем ...

- **Как настроить iSCSI target**
- **Как настроить iSCSI initiator**
- **Как настроить multipath**
- **Как настроить GFS2**

Кластер


Pacemaker
GFS2
CLVM
DLM
Pacemaker
GFS2
CLVM
DLM
Общее
хранилище
Fibre/SAS/iSCSI

Терминология iSCSI

- **initiator** — устанавливает соединение (клиент)
- **target** — тот, кто предоставляет блочное устройство (сервер) Портал — IP и порт таргета
- **IQN** — iqn.yyyy-mm.naming-authority:unique .
- **LUN** (Logical Unit Number) — номер объекта внутри цели(target). Ближайшим аналогом является раздел диска или отдельный том.
- **Enterprise Unique Identifier (EUI)** - eui.0123456789ABCDEF.

LIO: Backstores

- **FILEIO -** расшаривает файл как образ диска
- **BLOCK -** позволяет расшаривать диски
- **PSCSI -** позволяет расшраивать любые SCSI устройства, предпочтительнее чем BLOCK
- **RAMDISK -** шаринг памяти как SCSI

LIO: таргеты

- **TPGs (Target Portal Groups)** - поддержка нескольких конфигурация для одного таргета
- **Portals -** адрес:порт
- **LUNS (Logical Unit Number)**
- **ACLS -** позволяет разные конфигурации в зависимости от инициатора
- **AUTHENTICATION**

LIO: настройка таргета

```
targetcli /backstores/block create disk01 /dev/sdb
targetcli /iscsi create iqn.2018-09.ru.otus:storage.target00
targetcli /iscsi/iqn.2018-09.ru.otus:storage.target00/tpg1/portals create 0.0.0.0
targetcli /iscsi/iqn.2018-09.ru.otus:storage.target00/tpg1/luns create /backstores/block/disk01 lun=1
targetcli /iscsi/iqn.2018-09.ru.otus:storage.target00/tpg1/luns ls lun1
targetcli /iscsi/iqn.2018-09.ru.otus:storage.target00/tpg1 set attribute authentication=0
targetcli /iscsi/iqn.2018-09.ru.otus:storage.target00/tpg1 set auth userid=otus
targetcli /iscsi/iqn.2018-09.ru.otus:storage.target00/tpg1 set auth password=otus
```

Настройка инициатора: discovery

```
iscsiadm -m discovery -t st -p 192.168.7.153
iscsiadm -m discovery -t st -p 192.168.8.153
iscsiadm -m node
```

Настройка инициатора

- /etc/iscsi/iscsid.conf
- /etc/iscsi/initiatorname.iscsi
  - InitiatorName=iqn.1994-05.com.redhat:edf53f9a25
  - Данный инициатор нужно прописать в ACL в таргете
- systemctl enable iscsi
- systemctl start iscsi

Настройка инициатора: подключение

login

iscsiadm -m node -l -T iqn.2018-09.ru.otus:storage.target00

unlogin

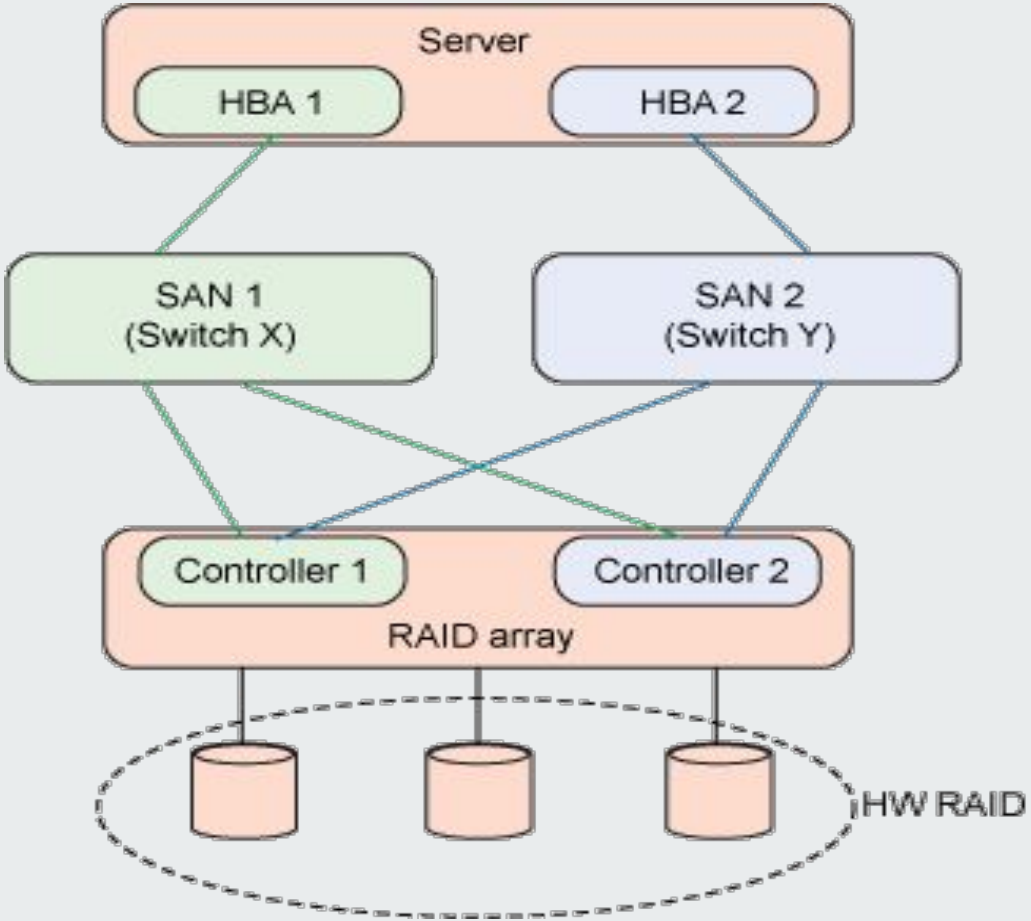
iscsiadm -m node -u -T iqn.2018-09.ru.otus:storage.target00

Информация по сессии

iscsiadm -m session -P3

iscsiadm -m node -T iqn.2018-09.ru.otus:storage.target00

Multipath


Server
HBA 1
HBA 2
SAN 1
(Switch X)
SAN 2
(Switch Y)
Controller 1
Controller 2
RAID array
HW RAID

Multipath

Multipath I/O — технология, позволяющая задействовать нескольких контроллеров или шин для доступа к одному устройству хранения данных.

Device-Mapper Multipath (DM Multipath, множественное связывание устройств) — технология (модуль ядра) для выделения мета-устройства связанного с дисками по нескольким путям

Multipath: multipath.conf

```
defaults {
        user_friendly_names yes
        find_multipaths yes
        path_selector               "round-robin 0"
        path_grouping_policy        failover
        failback                    immediate
}
blacklist {
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]"
}
multipaths {
   multipath {
        wwid                    360014058f29dd52c67e4d25bdd1001ad
        alias                   otusDsk
   }
}
```

Multipath

модуль ядра

dm_multipath

управление конфигом

mpathconf --enable --with_multipathd y

сервис

systemctl start multipathd

команда управления

multipath -ll

Multipath

модуль ядра

dm_multipath

управление конфигом

mpathconf --enable --with_multipathd y

сервис

systemctl start multipathd

команда управления

multipath -ll

Терминология

- **DLM (Distributed lock manager):** сервисы ради которых строится HA-кластер
- **CLVM (clustered LVM):** Кластерное управление логическими томам

GFS2: опции форматирования

- **Размер ФС** — 100 TB для 64-битных систем
  - Рекомендация: чем меньше тем лучше
- **Размер блока -** предпочтительный - 4к
- **Кол-во журналов** - 1 журнал под точку монтирования
- **Размер журнала** - по дефолту -128M
- **Размер и кол-во ресурсных групп (RG)** - по дефолту от 32M до 2G
  - маленькие и много - долгий поиск свободного места
  - большие и мало - конкуренция за доступ к RG

GFS2: рекомендации

- **В системе должно быть свободное место**
  - Иначе фрагментация и долгий поиск
- **Желательно чтоб каждая нода оперировала своими файлами**
  - При начале работы с файлом первая нода поставившая блокировку - считается мастером
  - Все остальные - спрашивают у мастера
- **RedHat** не рекомендует больше 16 узлов

GFS2: создаем ресурсы DLM и CLVM

```
# pcs property set stonith-enabled=false
# pcs property set no-quorum-policy=freeze
# pcs resource create dlm systemd:dlm op monitor interval=30s on-fail=ignore
clone interleave=true ordered=true
# pcs resource create clvmd ocf:heartbeat:clvm op monitor interval=30s
on-fail=ignore clone interleave=true ordered=true
# pcs constraint order start dlm-clone then clvmd-clone
# pcs status resources
```

GFS2: создаем ФС

```
# pvcreate /dev/mapper/mpatha
# vgcreate -Ay -cy cluster_vg /dev/mapper/maptha
# lvcreate -L900M -n cluster_lv cluster_vg

# mkfs.gfs2 -j2 -p lock_dlm -t otusha:gfs2 /dev/cluster_vg/cluster_lv
```

GFS2: включаем ресурс в кластер

```
# pcs resource create clusterfs Filesystem \
> device="/dev/cluster_vg/cluster_lv" directory="/mnt/gfs2" \
> fstype="gfs2" "options=noatime" op monitor interval=10s \
> on-fail=ignore clone interleave=true

# pcs constraint order start clvmd-clone then clusterfs-clone
# pcs constraint colocation add clusterfs-clone with clvmd-clone
```

GFS2: утилиты управления

```
# gfs2_edit -p journals /dev/cluster_vg/cluster_lv

# cat /sys/kernel/debug/gfs2/clustername:file_system_name/glocks

# tunegfs2 -l device
# tunegfs2 -o lockproto=lock_dlm device
# tunegfs2 -U uuid device

# dlm_tool -n ls
# dlm_tool fence_ack <nodeid>
```

GFS2: Glocks

- У каждого узла есть свой кэш
- Целостность кешей между узлами обеспечивают **GLOCKS**
  - Посредством DLM
- **1 glock per inode**
- При разделяемой блокировке (DLM:PR) кеш на данные под glock может быть на нескольких нодах
- При эксклюзивной блокировке(DLM:EX) только у одной ноды могут быть данные по этому файлу в кэше

GFS2: Glocks

- Если другая нода хочет установить DLM:EX
  - Посылается request ноде(ам) на сброс блокировки
  - Для DLM:PR - это инвалидация кеша
  - Для DLM:EX - это сброс буферов, журнализация и фиксация изменений и только затем инвалидация кеша

Спасибо за внимание

Алексей Цыкунов