КИБЕРПРОТЕКТ

КИБЕР Бэкап

Версия 18.6

Установка и настройка отказоустойчивых кластеров

Редакция: 30.06.2026

Авторское право

Содержание

1 Общие сведения

1.1 Описание

В этом руководстве приведены описание типовой архитектуры и пример настройки двухузлового отказоустойчивого кластера для программного обеспечения резервного копирования и восстановления данных Кибер Бэкап в операционной системе Ред ОС 7.3 Муром. В качестве программного обеспечения для поддержания целостности кластера и выполнения сигнальных функций используется Corosync, а для управления ресурсами кластера – Pacemaker.

1.2 Назначение документа и целевая аудитория

Данный документ предназначен для специалистов, которые выполняют внедрение и эксплуатацию сред резервного копирования и восстановления данных на основе программного обеспечения Кибер Бэкап. При написании документа не учитываются все возможные варианты кластеризации, опции кластерных ресурсов, способы обеспечения целостности кластера, доступные в программном стеке Pacemaker/Corosync, ввиду их огромного разнообразия и зависимости от используемого оборудования в конкретной инсталляции. Документ описывает общие принципы настройки кластера Pacemaker/Corosync, перечень и рекомендуемые настройки ресурсов кластера для Кибер Бэкап, пример настройки типового двухузлового кластера на физических серверах с выделенной виртуальной машиной-свидетелем.

Данный документ не является руководством по программному обеспечению Pacemaker/Corosync и не заменяет техническую документацию по указанным продуктам.

2 Архитектура

Подход, используемый в данном руководстве, может быть применен как при построении кластеров высокой доступности, расположенных целиком на одной площадке, так и при построении катастрофоустойчивых кластеров, распределенных по нескольким центрам обработки данных.

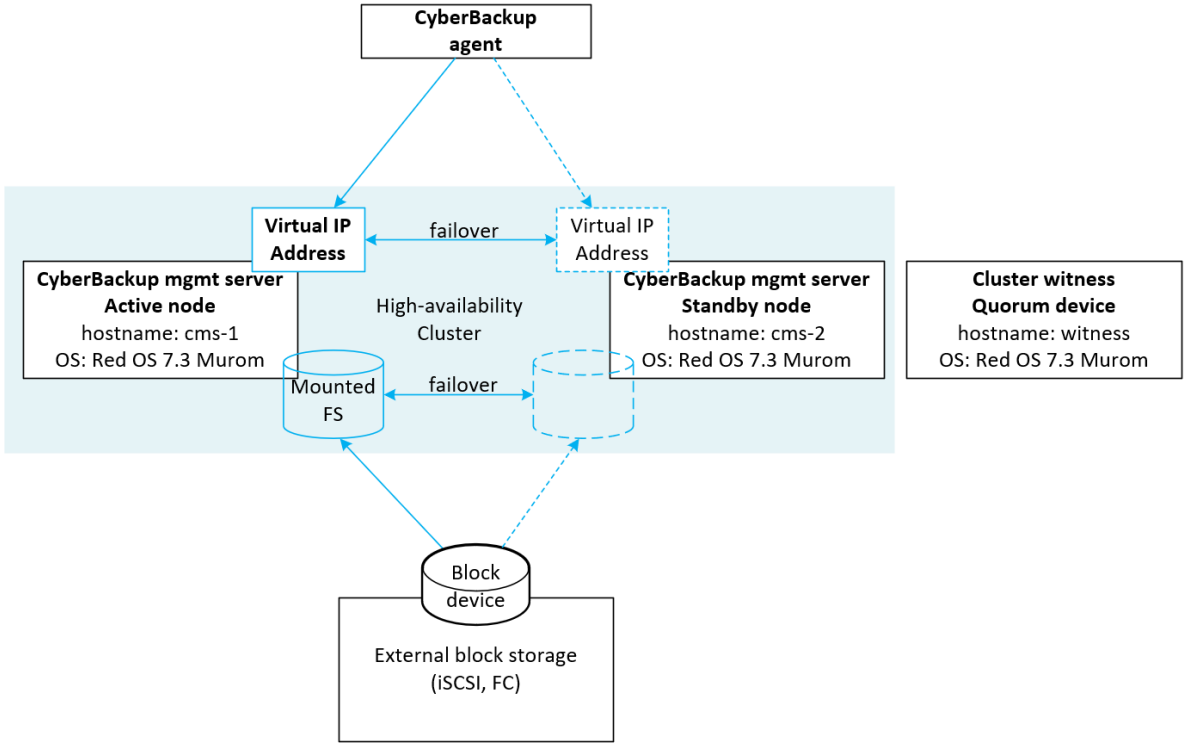
2.1 Компоненты кластера

В данном руководстве описан способ кластеризации Кибер Бэкап, включающий задействование следующих компонентов:

- Два аппаратных сервера с выделенными IPMI-интерфейсами;
- Внешняя блочная система хранения, доступная по протоколу iSCSI, или локальная блочная система хранения данных, доступная для репликации;
- Выделенный сервер, предпочтительно виртуализованный для кворумного устройства (так называемый "свидетель" или "арбитр").

Кластер Кибер Бэкап является классическим отказоустойчивым кластером с общими разделяемыми дисками и виртуальным IP-адресом сервиса. Целостность кластера обеспечивается механизмами Stonith с использованием функционала IPMI и защитой от split brain при помощи внешнего арбитра.

Схема компонентов кластера:

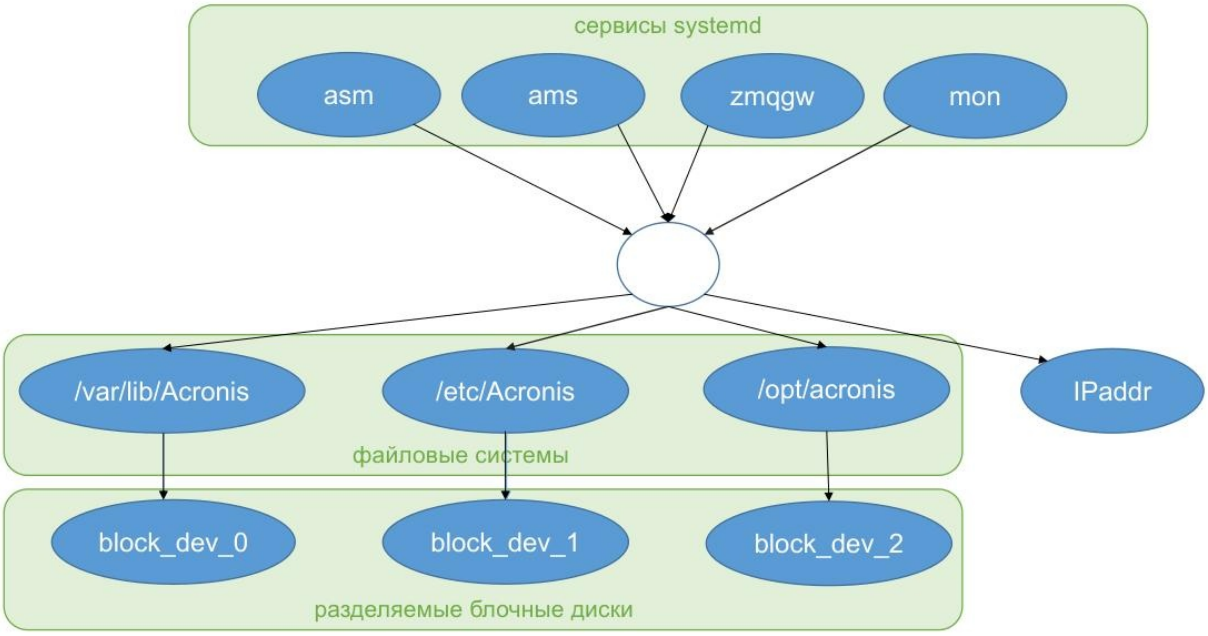


2.2 Ресурсы кластера

Для функционирования кластера необходимо настроить следующие типы кластерных ресурсов:

- Разделяемые блочные дисковые устройства;
- Перемещаемые файловые системы;
- Перемещаемый (виртуальный) IP-адрес, также называемый vIP (Virtual IP);
- Службы Кибер Бэкап;
- IPMI-ресурсы.

Дерево зависимостей ресурсов кластера изображено на следующем рисунке.



3 Развертывание и настройка

3.1 Настройка сети

3.1.1 Требования к организации сети

Узлам кластера, IPMI-интерфейсам и внешнему арбитру должны быть назначены статические адреса IPv4. Адреса узлов кластера и перемещаемый IP-адрес должны находиться в одном сегменте сети. Адреса внешнего арбитра и IPMI-интерфейсов могут находиться в отдельных сегментах сети, доступных с использованием сетевой маршрутизации, но также должна быть обеспечена сетевая связность со стороны узлов кластера.

Тип	ОС	Имя хоста	IP-адрес	IPMI-интерфейс	Виртуальный IP-адрес	Описание
Узел кластера	РЕД ОС 7.3.3 x64	cms-1	10.77.44.58/22	10.77.0.58/22	10.77.44.57/22	Первый узел кластера для сервера управления Кибер Бэкап. Вариант установки: "сервер минимальный".
	РЕД ОС 7.3.3 x64	cms-2	10.77.44.59/22	10.77.0.59/22		Второй узел кластера для сервера управления Кибер Бэкап. Вариант установки: "сервер минимальный".
Арбитр	РЕД ОС 7.3.3 x64	witness	10.77.44.61/22	–	–	Арбитр с кворумным устройством для предотвращения split brain. Вариант установки: "сервер минимальный". Допустимо использование виртуальной машины.

3.1.2 Настройка сетевых интерфейсов

На всех узлах кластера и арбитре настройте статические сетевые адреса, шлюз по умолчанию и адреса серверов имен (файл /etc/sysconfig/network-scripts/ifcfg-ens3):

```
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
NAME=ens3
DEVICE=ens3
ONBOOT=yes
IPADDR=10.77.44.58
PREFIX=22
GATEWAY=10.77.44.1
DNS1=10.77.29.101
DNS2=10.77.29.10
```

В файл /etc/hosts на узлах кластера и арбитре добавьте записи с остальными компонентами кластера, чтобы целостность кластера не зависела от стабильности работы внешних DNS-серверов:

```
10.77.44.57 cms     # Virtual (floating) cluster IP address
10.77.44.58 cms-1   # Node 1 static address
10.77.44.59 cms-2   # Node 2 static address
10.77.44.61 witness # Cluster witness with quorum device
```

Перезагрузите серверы и проверьте:

- Наличие статических адресов на интерфейсах;
- Наличие маршрута по умолчанию;
- Сетевую связность между всеми компонентами кластера.

```
[root@cms-1 network-scripts]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 56:6f:93:3c:00:64 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.77.44.58/22 brd 10.77.47.255 scope global noprefixroute ens3
       valid_lft forever preferred_lft forever
    inet6 fe80::546f:93ff:fe3c:64/64 scope link
       valid_lft forever preferred_lft forever
```

```
[root@cms-1 ~]# ip route
default via 10.77.44.1 dev ens3 proto static metric 100
10.77.44.0/22 dev ens3 proto kernel scope link src 10.77.44.58 metric 100
```

3.2 Подготовка узлов кластера

3.2.1 Установка кластерного программного обеспечения

Установите программное обеспечение для кластера, используя следующую таблицу.

Тип	Имя хоста	Команды
Узел кластера	cms-1	# dnf update # dnf install pcs pacemaker corosync corosync-qdevice lsyncd # dnf install iscsi-initiator-utils
Узел кластера	cms-2	# dnf update # dnf install pcs pacemaker corosync corosync-qdevice lsyncd # dnf install iscsi-initiator-utils
Арбитр	witness	# dnf update # dnf install pcs corosync-qnetd

3.2.2 Отключение SELinux

Для отключения SELinux на узлах кластера и арбитре выполните следующую команду:

```
setenforce 0
```

В конфигурационном файле /etc/selinux/config измените значение **enforcing** на **disabled**:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3.2.3 Настройка доступа по SSH-ключам

Для root настройте доступ между узлами кластера по SSH-ключам:

```
[root@cms-1 ~]# ssh-copy-id root@cms-2

[root@cms-2 ~]# ssh-copy-id root@cms-1
```

3.2.4 Настройка lsyncd

На первом узле кластера (cms-1) создайте конфигурационные файлы для автоматической двунаправленной синхронизации следующих системных файлов:

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/security/acronisagent.conf

Конфигурационный файл lsyncd (/etc/lsyncd.conf):

```
settings {
   logfile = "/var/log/lsyncd.log"
}

sync {
   default.rsyncssh,
   source = "/etc",
   host = "cms-2",
   targetdir = "/etc",
   delete = 'false',
   rsync = {
      update = "true",
      whole_file = "true",
      _extra = {
         "--include-from=/etc/lsyncd.list",
         "--exclude=*"
      }
   }
}

sync {
   default.rsyncssh,
   source = "/etc/security",
   host = "cms-2",
   targetdir = "/etc/security",
   delete = 'false',
   rsync = {
      update = "true",
```

```
        whole_file = "true",
        _extra = {
           "--include=acronisagent.conf",
           "--exclude=*"
        }
     }
}
```

Создайте список файлов для синхронизации (/etc/lsyncd.list):

```
passwd
shadow
group
```

Перезапустите сервис lsyncd:

```
systemctl enable lsyncd --now
```

Выполните аналогичную процедуру на втором узле кластера (cms-2). В конфигурационном файле /etc/lsyncd.conf необходимо изменить имя хоста на cms-1.

Далее описывается настройка кластера для двух способов его практической реализации:

- через подключение и настройку блочных iSCSI-устройств хранения (см. раздел "Настройка кластера для iSCSI-устройств" (стр. 11));
- через подключение и настройку локальных блочных устройств хранения с использованием drbd (см. раздел "Настройка кластера для локальных устройств" (стр. 20)).

3.3 Настройка кластера для iSCSI-устройств

3.3.1 Настройка кластера

3.3.1.1 Пользователь hacluster

На узлах кластера и арбитре задайте пароль для пользователя **hacluster**:

```
passwd hacluster
```

3.3.1.2 Кластерные службы

На узлах кластера добавьте запуск кластерных служб при перезагрузке:

```
systemctl enable pcsd pacemaker corosync
```

3.3.1.3 Инициализация кластера

На всех узлах кластера и на свидетеле запустите pscd:

```
systemctl start pcsd
```

На одном из узлов кластера выполните команды по инициализации кластера:

```
pcs host auth cms-1 cms-2 -u hacluster # Запросит пароль hacluster
pcs cluster setup CyberBackup cms-1 cms-2
pcs cluster enable --all
pcs cluster start --all
```

где cms-1 и cms-2 – имена узлов кластера; CyberBackup – имя кластера.

3.3.1.4 Настройка арбитра

Инициализируйте на арбитре кворумное устройство и запустите pcsd:

```
[root@witness ~]# pcs qdevice setup model net --enable --start
Quorum device 'net' initialized
quorum device enabled
Starting quorum device...
quorum device started

[root@witness ~]# systemctl enable pcsd --now
Created symlink /etc/systemd/system/multi-user.target.wants/pcsd.service →
/usr/lib/systemd/system/pcsd.service.
```

Активируйте и запустите corosync-qdevice на каждом узле кластера:

```
[root@cms-1 ~]# systemctl enable corosync-qdevice
Created symlink /etc/systemd/system/multi-user.target.wants/corosyncqdevice.
service → /usr/lib/systemd/system/corosync-qdevice.service.
```

Для инициализации и настройки кворумного устройства на одном из узлов кластера выполните следующие команды:

```
[root@cms-1 ~]# pcs host auth witness
Username: hacluster
Password:
witness: Authorized

[root@cms-1 ~]# pcs quorum device add model net host=witness algorithm=ffsplit
Setting up qdevice certificates on nodes...
cms-1: Succeeded
cms-2: Succeeded
Enabling corosync-qdevice...
cms-1: corosync-qdevice enabled
cms-2: corosync-qdevice enabled
Sending updated corosync.conf to nodes...
cms-1: Succeeded
cms-2: Succeeded
```

```
cms-1: Corosync configuration reloaded
Starting corosync-qdevice...
cms-2: corosync-qdevice started
cms-1: corosync-qdevice started

[root@cms-1 ~]# pcs quorum status
Quorum information
------------------
Date:             Thu Nov 16 21:50:29 2023
Quorum provider:  corosync_votequorum
Nodes:            2
Node ID:          1
Ring ID:          1.6a
Quorate:          Yes

Votequorum information
----------------------
Expected votes:   3
Highest expected: 3
Total votes:      3
Quorum:           2
Flags:            Quorate Qdevice

Membership information
----------------------
Nodeid      Votes    Qdevice      Name
1          1    A,V,NMW     cms-1 (local)
2          1    A,V,NMW     cms-2
0          1    Qdevice
```

3.3.1.5 Настройка механизма stonith / fencing

Для обеспечения целостности разделяемых данных в кластере существует механизм, называемый stonith или fencing. В случае сетевой недоступности неисправного узла кластера он всё еще может выполнять доступ к разделяемым данным, поэтому уцелевшая часть кластера должна иметь механизм для предотвращения доступа узла кластера, не отвечающего на запросы. Чаще всего это реализуется с помощью аппаратных средств, например, перезагрузка узла при помощи интерфейса управления IPMI, временного прерывания подачи питания через управляемые устройства распределения питания PDU (Power Distribution Unit), отключение портов узла на LAN/SAN-коммутаторах и т. п. Возможно использование одновременно нескольких fencing-методов.

Внимание
Промышленная эксплуатация кластера без настроенного механизма stonith / fencing недопустима и ведет к риску порчи данных.

Ниже приведен вариант настройки fencing с использованием IPMI-совместимых модулей управления физического сервера.

1. Включите механизм fencing:

```
pcs property set stonith-enabled=true
```

2. Установите на узлы кластера пакет с fencing-агентом:

```
dnf install fence-agents-ipmilan
```

3. Для каждого узла кластера создайте fencing-ресурс:

```
pcs stonith create cms1_ipmi fence_ipmilan pcmk_host_list="cms-1" \
    ip=10.77.0.58 username=testuser password=acd123 privlvl=operator \
    lanplus=1 op monitor interval=60s

pcs stonith create cms2_ipmi fence_ipmilan pcmk_host_list="cms-2" \
    ip=10.77.0.59 username=testuser password=acd123 privlvl=operator \
    lanplus=1 op monitor interval=60s
```

   Все параметры, доступные при создании fencing-ресурса, можно уточнить следующей командой:

```
pcs stonith describe fence_ipmilan
```

4. Создайте правила, запрещающие fencing-ресурсу для отключения узла запускаться на этом же самом узле:

```
pcs constraint location cms1_ipmi avoids cms-1
pcs constraint location cms2_ipmi avoids cms-2
```

5. Работа механизма fencing должна быть протестирована как отдельно, так и при тестировании отработки кластером сценариев отказа. Ручное тестирование интеграции возможно с использованием следующей команды, которая должна перезагрузить сервер cms-1 с использованием IPMI-интерфейса:

```
[root@cms-2 ~]# stonith_admin --reboot cms-1
```

   В тестовых кластерах, которые не содержат данных, представляющих ценность, и используются только с целью проверки совместимости или отработки какой-либо гипотезы, допустимо отключение механизма fencing. Такие кластеры могут быть целиком реализованы внутри виртуальных машин и не иметь работающего механизма fencing. В таком случае его следует отключить:

```
pcs property set stonith-enabled=false
```

3.3.1.6 Монтирование блочных iSCSI-устройств

Выполните подключение блочных iSCSI-устройств на одном из узлов кластера. Сканируйте доступные из системы устройства с именем хоста iscsi-tgt.

```
[root@cms-1 ~]# iscsiadm --mode discovery --type st --portal iscsi-tgt --login
10.77.44.62:3260,1 iqn.2003-01.org.linux-iscsi.iscsitgt.
x8664:sn.bfbc1ebb624c
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.iscsitgt.
x8664:sn.bfbc1ebb624c, portal: 10.77.44.62,3260]
Login to [iface: default, target: iqn.2003-01.org.linux-iscsi.iscsitgt.
x8664:sn.bfbc1ebb624c, portal: 10.77.44.62,3260] successful.
```

Проверьте наличие активных iSCSI-сессий до целевого устройства и отключите автоматическое подключение iSCSI-клиентом дисков с внешней системы хранения:

```
[root@cms-2 ~]# iscsiadm --mode session
tcp: [1] 10.77.44.62:3260,1 iqn.2003-01.org.linux-iscsi.iscsitgt.
x8664:sn.bfbc1ebb624c (non-flash)

[root@cms-2 ~]# iscsiadm --mode node --portal iscsi-tgt --op update \
    --name node.startup --value manual
```

Проверьте доступность новых блочных устройств:

```
[root@cms-1 ~]# lsblk
NAME    MAJ:MIN  RM  SIZE  RO  TYPE  MOUNTPOINTS
...
sdc       8:32     0   5G     0   disk
sdd       8:48     0   100G   0   disk
sde       8:64     0   20G    0   disk
...
```

Отформатируйте новые диски в ext4 и смонтируйте файловые системы:

```
[root@cms-1 ~]# mkfs.ext4 -m 0 -L etc /dev/sdc
[root@cms-1 ~]# mkfs.ext4 -m 0 -L var /dev/sdd
[root@cms-1 ~]# mkfs.ext4 -m 0 -L opt /dev/sde

[root@cms-1 ~]# mkdir -p /etc/Acronis /opt/acronis /var/lib/Acronis
[root@cms-2 ~]# mkdir -p /etc/Acronis /opt/acronis /var/lib/Acronis

[root@cms-1 ~]# mount /dev/sdc /etc/Acronis
[root@cms-1 ~]# mount /dev/sdd /var/lib/Acronis
[root@cms-1 ~]# mount /dev/sde /opt/acronis

[root@cms-1 ~]# lsblk -f
NAME              FSTYPE    FSVER    LABEL    UUID
FSAVAIL    FSUSE%  MOUNTPOINTS
...
sdc               ext4 1.0 etc cdbcd761-01fc-4f13-ad48-
cdcee09a6dd2 4,8G 0% /etc/Acronis
sdd               ext4 1.0 var eda7bfbb-edd8-4af3-b866-
1bdfa26879d6 97,7G 0% /var/lib/Acronis
sde               ext4 1.0 opt 1fa314b9-765f-460a-b0ef-
```

```
9f3c56c0dd81 19,5G 0% /opt/acronis
...
```

3.3.1.7 Кластерные ресурсы и файловые системы

На одном из узлов кластера создайте кластерные ресурсы:

```
pcs resource create rsc_shared_disks ocf:heartbeat:iscsi udev="no" \
    portal=10.77.44.62 target=iqn.2003-01.org.linux-iscsi.iscsi-tgt.x8664:sn.bfbc1ebb624c \
    op monitor interval=60s timeout=30s

pcs resource create rsc_fs_etc ocf:heartbeat:Filesystem \
    device="/dev/disk/by-uuid/cdbcd761-01fc-4f13-ad48-cdcee09a6dd2" \
    directory="/etc/Acronis" fstype="ext4" options="lazytime" \
    force_unmount="true" \
    op monitor interval=60 timeout=60 OCF_CHECK_LEVEL=20 \
    meta migration-threshold=2 failure-timeout=1d

pcs resource create rsc_fs_var ocf:heartbeat:Filesystem \
    device="/dev/disk/by-uuid/eda7bfbb-edd8-4af3-b866-1bdfa26879d6" \
    directory="/var/lib/Acronis/" fstype="ext4" options="lazytime" \
    force_unmount="true" \
    op monitor interval=60 timeout=60 OCF_CHECK_LEVEL=20 \
    meta migration-threshold=2 failure-timeout=1d

pcs resource create rsc_fs_opt ocf:heartbeat:Filesystem \
    device="/dev/disk/by-uuid/1fa314b9-765f-460a-b0ef-9f3c56c0dd81" \
    directory="/opt/acronis" fstype="ext4" options="lazytime" \
    force_unmount="true" \
    op monitor interval=60 timeout=60 OCF_CHECK_LEVEL=20 \
    meta migration-threshold=2 failure-timeout=1d

pcs constraint colocation set rsc_fs_etc rsc_fs_var rsc_fs_opt \
    sequential=false set rsc_shared_disks setoptions id=col_fs

pcs constraint order set rsc_shared_disks set rsc_fs_etc rsc_fs_var \
    rsc_fs_opt sequential=false require-all=true setoptions id=ord_fs
```

3.3.1.8 Установка Кибер Бэкап

Установите Кибер Бэкап на узле с примонтированными файловыми системами стандартным образом (см. раздел см. раздел "Установка" Руководства пользователя Кибер Бэкап). После успешной установки сервера управления Кибер Бэкап необходимо установить его также на второй узел. Для этого остановите службы Кибер Бэкап и переместите файловые системы на второй узел:

```
[root@cms-1 ~]# systemctl stop acronis_ams
[root@cms-1 ~]# systemctl stop acronis_asm
[root@cms-1 ~]# systemctl stop acronis_zmqgw
[root@cms-1 ~]# systemctl stop acronis_monitoring_service
```

```
[root@cms-1 ~]# pcs resource move rsc_shared_disks --autodelete

[root@cms-1 ~]# pcs status
Cluster name: CyberBackup
Cluster Summary:
    * Stack: corosync (Pacemaker is running)
    * Current DC: cms-1 (version 2.1.6-1.el7-6fdc9deea29) - partition with quorum
    * Last updated: Wed Nov  1 09:58:45 2023 on cms-1
    * Last change:  Wed Nov  1 09:58:43 2023 by root via crm_resource on cms-1
    * 2 nodes configured
    * 9 resource instances configured

Node List:
    * Online: [ cms-1 cms-2 ]

Full List of Resources:
    * rsc_shared_disks    (ocf::custom:iscsi):    Started cms-2
    * rsc_fs_etc  (ocf::heartbeat:Filesystem):    Started cms-2
    * rsc_vip (ocf::heartbeat:IPaddr2):    Started cms-1
    * rsc_fs_var  (ocf::heartbeat:Filesystem):    Started cms-2
    * rsc_fs_opt  (ocf::heartbeat:Filesystem):    Started cms-2

Daemon Status:
    corosync: active/enabled
    pacemaker: active/enabled
    pcsd: active/enabled
```

3.3.1.9 Кластерные ресурсы и службы Кибер Бэкап

Создайте кластерные ресурсы для служб Кибер Бэкап на одном из узлов кластера.

Внимание

В параметре **portal** ресурса rsc_vip обязательно укажите IP-адрес, а не доменное имя.

```
pcs resource create rsc_vip ocf:heartbeat:IPaddr2 \
    ip="10.77.44.57" cidr_netmask="22" \
    op monitor interval="30"

pcs resource create rsc_ams systemd:acronis_ams \
    op start interval=0 timeout=60 \
    op stop interval=0 timeout=60 \
    op monitor interval=10 \
    meta target-role=Started \
    migration-threshold=2 failure-timeout=1d


pcs resource create rsc_asm systemd:acronis_asm \
    op start interval=0 timeout=60 \
    op stop interval=0 timeout=60 \
    op monitor interval=10 \
```

```
    meta target-role=Started \
    migration-threshold=2 failure-timeout=1d


pcs resource create rsc_mon systemd:acronis_monitoring_service \
    op start interval=0 timeout=60 \
    op stop interval=0 timeout=60 \
    op monitor interval=10 \
    meta target-role=Started \
    migration-threshold=2 failure-timeout=1d


pcs resource create rsc_zmqgw systemd:acronis_zmqgw \
    op start interval=0 timeout=60 \
    op stop interval=0 timeout=60 \
    op monitor interval=10 \
    meta target-role=Started \
    migration-threshold=2 failure-timeout=1d

pcs constraint colocation set rsc_ams rsc_asm rsc_mon rsc_zmqgw \
        sequential=false set rsc_fs_etc rsc_fs_var rsc_fs_opt setoptions \
        id=col_services

pcs constraint order set rsc_fs_etc rsc_fs_var rsc_fs_opt sequential=false \
        require-all=true set rsc_ams rsc_asm rsc_mon rsc_zmqgw sequential=false \
        setoptions id=ord_services

pcs constraint colocation add rsc_vip with rsc_ams id=col_vip


[root@cms-2 ~]# pcs resource cleanup

[root@cms-2 ~]# pcs  status
Cluster name: CyberBackup
Cluster Summary:
    * Stack: corosync (Pacemaker is running)
    * Current DC: cms-1 (version 2.1.6-1.el7-6fdc9deea29) - partition with quorum
    * Last updated: Wed Nov  1 10:04:32 2023 on cms-2
    * Last change:  Wed Nov  1 09:58:43 2023 by root via crm_resource on cms-1
    * 2 nodes configured
    * 9 resource instances configured

Node List:
    * Online: [ cms-1 cms-2 ]

Full List of Resources:
    * rsc_shared_disks    (ocf::custom:iscsi):    Started cms-2
    * rsc_fs_etc  (ocf::heartbeat:Filesystem):    Started cms-2
    * rsc_vip (ocf::heartbeat:IPaddr2):    Started cms-2
    * rsc_fs_var  (ocf::heartbeat:Filesystem):    Started cms-2
    * rsc_fs_opt  (ocf::heartbeat:Filesystem):    Started cms-2
    * rsc_ams (systemd:acronis_ams):   Started cms-2
```

```
      * rsc_asm (systemd:acronis_asm):   Started cms-2
      * rsc_mon (systemd:acronis_monitoring_service):    Started cms-2
      * rsc_zmqgw   (systemd:acronis_zmqgw):     Started cms-2

  Daemon Status:
      corosync: active/enabled
      pacemaker: active/enabled
      pcsd: active/enabled
```

Убедитесь, что авторизация в веб-консоли по публичному адресу или его имени http://cms:9877 проходит успешно.

3.3.2 Подключение агентов

Для корректной работы клиентов после переключения активного узла кластера необходимо, чтобы регистрация агентов была выполнена с использованием плавающего кластерного адреса сервера управления. Данный адрес соответствует адресу, указанному в кластерном ресурсе rsc_vip.

Если агент ранее был зарегистрирован на некластеризованном сервере управления, то необходимо выполнить его перерегистрацию с явным указанием кластерного адреса. Пример перерегистрации агента из клиентской Linux-системы с использованием одноразового токена:

```
root@cbclient:~# /usr/lib/Acronis/RegisterAgentTool/RegisterAgent --
operation unregister
Success.
200
null

root@cbclient:~# /usr/lib/Acronis/RegisterAgentTool/RegisterAgent --
operation register --address cms --token 8D77-CEDC-433B
Success.
200
null
```

В этом примере:

- cms – кластерное доменное имя или vIP-адрес сервера управления;
- 8D77-CEDC-433B – токен, полученный в консоли управления Кибер Бэкап для регистрации агента.

Подробное описание дополнительных возможностей и вариантов регистрации агентов см. в отдельной статье базы знаний.

3.4 Настройка кластера для локальных устройств

3.4.1 Настройка кластера

3.4.1.1 Пользователь hacluster

На узлах кластера и арбитре задайте пароль для пользователя **hacluster**:

```
passwd hacluster
```

3.4.1.2 Кластерные службы

На узлах кластера добавьте запуск кластерных служб при перезагрузке:

```
systemctl enable pcsd pacemaker corosync
```

3.4.1.3 Подключение блочных устройств

На узлах кластера и арбитре установите kmod с помощью команды:

```
dnf install drbd-kmod
```

Перезагрузите машины:

```
reboot
```

На узлах кластера и арбитре создайте файл /etc/drbd.d/ha.res с содержимым, как в следующем примере:

```
resource "ha" {
  volume 0 {
    device "/dev/drbd0";
    disk "/dev/sdb";
    meta-disk internal;
  }
  volume 1 {
    device "/dev/drbd1";
    disk "/dev/sdc";
    meta-disk internal;
  }
  volume 2 {
    device "/dev/drbd2";
    disk "/dev/sdd";
    meta-disk internal;
  }
  options {
```

```
    on-no-quorum suspend-io;
    quorum majority;
  }
  on "node1" {
    address 192.168.12.88:7788;
    node-id 0;
  }
  on "node2" {
    address 192.168.12.132:7788;
    node-id 1;
  }
  on "arbiter" {
    volume 0 {
      disk none;
    }
    volume 1 {
      disk none;
    }
    volume 2 {
      disk none;
    }
    address 192.168.12.151:7788;
    node-id 2;
  }
  connection-mesh {
    hosts "node1" "node2" "arbiter";
  }
}
```

Примечание

В этих данных вместо node1, node2, arbiter нужно указать имена хостов; в значении address нужно указать IP-адреса этих хостов.

В файле /etc/drbd.d/ha.res на узлах кластера и арбитре внесите следующие строки:

```
common {
    net {
        protocol C;
```

Далее на узлах кластера выполните следующие команды:

```
wipefs -a -f /dev/sdb
wipefs -a -f /dev/sdc
wipefs -a -f /dev/sdd

modprobe drbd
drbdadm down ha
drbdadm --force create-md ha
drbdadm up ha
drbdadm status ha
```

На первом узле кластера выполните следующие команды:

```
drbdadm primary --force ha
mkfs.ext4 -F /dev/drbd0
mkfs.ext4 -F /dev/drbd1
mkfs.ext4 -F /dev/drbd2
drbdadm secondary ha

drbdadm primary --force ha
```

Проверьте статусы узлов:

```
[root@node1 ~]# drbdadm status
ha role:Primary
  volume:0 disk:UpToDate
  volume:1 disk:UpToDate
  volume:2 disk:UpToDate
  node2 role:Secondary
    volume:0 peer-disk:UpToDate
    volume:1 peer-disk:UpToDate
    volume:2 peer-disk:UpToDate
   arbiter role:Secondary
    volume:0 peer-disk:Diskless
    volume:1 peer-disk:Diskless
    volume:2 peer-disk:Diskless
```

Убедитесь, что статусы соответствуют назначенным. Если арбитр имеет другой статус, попробуйте выполнить следующие команды:

```
drbdadm down ha
drbdadm up ha
```

Проверьте статус дисков, используя команду:

```
mount | grep drbd
```

После этого на первом узле кластера запустите службы:

```
systemctl enable pcsd pacemaker
systemctl enable --now drbd
```

3.4.1.4 Инициализация кластера

На всех узлах кластера и на свидетеле запустите pscd:

```
systemctl start pcsd
```

На одном из узлов кластера выполните команды по инициализации кластера:

```
pcs host auth cms-1 cms-2 -u hacluster # Запросит пароль hacluster
pcs cluster setup CyberBackup cms-1 cms-2
pcs cluster enable --all
pcs cluster start --all
```

где cms-1 и cms-2 – имена узлов кластера; CyberBackup – имя кластера.

Создайте необходимые каталоги на узлах кластера с помощью команды:

```
mkdir -p /etc/Acronis /opt/acronis /var/lib/Acronis
```

3.4.1.5 Настройка арбитра

Инициализируйте на арбитре кворумное устройство и запустите pcsd:

```
[root@witness ~]# pcs qdevice setup model net --enable --start
Quorum device 'net' initialized
quorum device enabled
Starting quorum device...
quorum device started

[root@witness ~]# systemctl enable pcsd --now
Created symlink /etc/systemd/system/multi-user.target.wants/pcsd.service →
/usr/lib/systemd/system/pcsd.service.
```

Активируйте и запустите corosync-qdevice на каждом узле кластера:

```
[root@cms-1 ~]# systemctl enable corosync-qdevice
Created symlink /etc/systemd/system/multi-user.target.wants/corosyncqdevice.
service → /usr/lib/systemd/system/corosync-qdevice.service.
```

Для инициализации и настройки кворумного устройства на одном из узлов кластера выполните следующие команды:

```
[root@cms-1 ~]# pcs host auth witness
Username: hacluster
Password:
witness: Authorized

[root@cms-1 ~]# pcs quorum device add model net host=witness algorithm=ffsplit
Setting up qdevice certificates on nodes...
cms-1: Succeeded
cms-2: Succeeded
Enabling corosync-qdevice...
cms-1: corosync-qdevice enabled
cms-2: corosync-qdevice enabled
Sending updated corosync.conf to nodes...
cms-1: Succeeded
cms-2: Succeeded
```

```
cms-1: Corosync configuration reloaded
Starting corosync-qdevice...
cms-2: corosync-qdevice started
cms-1: corosync-qdevice started

[root@cms-1 ~]# pcs quorum status
Quorum information
------------------
Date:             Thu Nov 16 21:50:29 2023
Quorum provider:  corosync_votequorum
Nodes:            2
Node ID:          1
Ring ID:          1.6a
Quorate:          Yes

Votequorum information
----------------------
Expected votes:   3
Highest expected: 3
Total votes:      3
Quorum:           2
Flags:            Quorate Qdevice

Membership information
----------------------
Nodeid      Votes    Qdevice      Name
1          1      A,V,NMW     cms-1 (local)
2          1      A,V,NMW     cms-2
0          1      Qdevice
```

3.4.1.6 Настройка механизма stonith / fencing

Для обеспечения целостности разделяемых данных в кластере существует механизм, называемый stonith или fencing. В случае сетевой недоступности неисправного узла кластера он всё еще может выполнять доступ к разделяемым данным, поэтому уцелевшая часть кластера должна иметь механизм для предотвращения доступа узла кластера, не отвечающего на запросы. Чаще всего это реализуется с помощью аппаратных средств, например, перезагрузка узла при помощи интерфейса управления IPMI, временного прерывания подачи питания через управляемые устройства распределения питания PDU (Power Distribution Unit), отключение портов узла на LAN/SAN-коммутаторах и т. п. Возможно использование одновременно нескольких fencing-методов.

Внимание
Промышленная эксплуатация кластера без настроенного механизма stonith / fencing недопустима и ведет к риску порчи данных.

Ниже приведен вариант настройки fencing с использованием IPMI-совместимых модулей управления физического сервера.

1. Включите механизм fencing:

```
pcs property set stonith-enabled=true
```

2. Установите на узлы кластера пакет с fencing-агентом:

```
dnf install fence-agents-ipmilan
```

3. Для каждого узла кластера создайте fencing-ресурс:

```
pcs stonith create cms1_ipmi fence_ipmilan pcmk_host_list="cms-1" \
    ip=10.77.0.58 username=testuser password=acd123 privlvl=operator \
    lanplus=1 op monitor interval=60s

pcs stonith create cms2_ipmi fence_ipmilan pcmk_host_list="cms-2" \
    ip=10.77.0.59 username=testuser password=acd123 privlvl=operator \
    lanplus=1 op monitor interval=60s
```

   Все параметры, доступные при создании fencing-ресурса, можно уточнить следующей командой:

```
pcs stonith describe fence_ipmilan
```

4. Создайте правила, запрещающие fencing-ресурсу для отключения узла запускаться на этом же самом узле:

```
pcs constraint location cms1_ipmi avoids cms-1
pcs constraint location cms2_ipmi avoids cms-2
```

5. Работа механизма fencing должна быть протестирована как отдельно, так и при тестировании отработки кластером сценариев отказа. Ручное тестирование интеграции возможно с использованием следующей команды, которая должна перезагрузить сервер cms-1 с использованием IPMI-интерфейса:

```
[root@cms-2 ~]# stonith_admin --reboot cms-1
```

   В тестовых кластерах, которые не содержат данных, представляющих ценность, и используются только с целью проверки совместимости или отработки какой-либо гипотезы, допустимо отключение механизма fencing. Такие кластеры могут быть целиком реализованы внутри виртуальных машин и не иметь работающего механизма fencing. В таком случае его следует отключить:

```
pcs property set stonith-enabled=false
```

3.4.1.7 Кластерные ресурсы и файловые системы

На одном из узлов кластера создайте кластерные ресурсы:

```
pcs resource create drbd_ha_cb ocf:linbit:drbd drbd_resource=ha op monitor interval="60s"
pcs resource promotable drbd_ha_cb meta promoted-max=1 meta promoted-node-max=1 meta
clone-max=2 meta clone-node-max=1 meta notify=true
pcs resource create fs_etc ocf:heartbeat:Filesystem device=/dev/drbd0 directory="/etc/Acronis"
fstype=ext4
pcs resource create fs_var ocf:heartbeat:Filesystem device=/dev/drbd1 directory="/var/lib/Acronis/"
fstype=ext4
pcs resource create fs_opt ocf:heartbeat:Filesystem device=/dev/drbd2 directory="/opt/acronis"
fstype=ext4
pcs constraint colocation set fs_etc fs_var fs_opt sequential=false set drbd_ha_cb-clone
role=Promoted
pcs constraint order set drbd_ha_cb-clone action=promote set fs_etc fs_var fs_opt sequential=false
require-all=true setoptions id=ord_fs
pcs resource cleanup
```

3.4.1.8 Установка Кибер Бэкап

Установите Кибер Бэкап на узле с примонтированными файловыми системами стандартным образом (см. раздел см. раздел "Установка" Руководства пользователя Кибер Бэкап). После успешной установки сервера управления Кибер Бэкап необходимо установить его также на второй узел. Для этого остановите службы Кибер Бэкап и переместите файловые системы на второй узел.

Для остановки служб Кибер Бэкап выполните:

```
systemctl stop acronis_ams acronis_asm acronis_zmqgw acronis_monitoring_service
systemctl disable acronis_ams acronis_asm acronis_zmqgw acronis_monitoring_service
```

Для переноса ресурсов кластера на второй узел выполните:

```
pcs resource move drbd_ha_cb node2
```

где node2 — имя переносимого ресурса.

3.4.1.9 Кластерные ресурсы и службы Кибер Бэкап

Создайте кластерные ресурсы для служб Кибер Бэкап на втором узле кластера.

Выполните команду:

```
pcs resource create cb_vip ocf:heartbeat:IPaddr2 ip=192.168.12.250 cidr_netmask=24 op monitor
interval=20s timeout=20s
```

где для параметра ip нужно указать виртуальный IP-адрес, который будет использоваться для авторизации в Кибер Бэкап.

Далее выполните следующие команды:

```
pcs resource create rsc_ams systemd:acronis_ams op monitor interval=10s
pcs resource create rsc_asm systemd:acronis_asm op monitor interval=10s
```

```
pcs resource create rsc_mon systemd:acronis_monitoring_service op monitor interval=10s
pcs resource create rsc_zmqgw systemd:acronis_zmqgw op monitor interval=10s
pcs constraint colocation set rsc_ams rsc_asm rsc_mon rsc_zmqgw sequential=false set fs_etc fs_
var fs_opt
pcs constraint order set fs_etc fs_var fs_opt sequential=false require-all=true set rsc_ams rsc_asm
rsc_mon rsc_zmqgw sequential=false require-all=true
pcs constraint colocation add cb_vip with rsc_ams
pcs resource cleanup
```

Убедитесь, что авторизация в веб-консоли по виртуальному IP-адресу http://cms:9877 проходит успешно.

3.4.2 Подключение агентов

Для корректной работы клиентов после переключения активного узла кластера необходимо, чтобы регистрация агентов была выполнена с использованием плавающего кластерного адреса сервера управления. Данный адрес соответствует адресу, указанному в кластерном ресурсе rsc_vip.

Если агент ранее был зарегистрирован на некластеризованном сервере управления, то необходимо выполнить его перерегистрацию с явным указанием кластерного адреса. Пример перерегистрации агента из клиентской Linux-системы с использованием одноразового токена:

```
root@cbclient:~# /usr/lib/Acronis/RegisterAgentTool/RegisterAgent --
operation unregister
Success.
200
null

root@cbclient:~# /usr/lib/Acronis/RegisterAgentTool/RegisterAgent --
operation register --address cms --token 8D77-CEDC-433B
Success.
200
null
```

В этом примере:

- cms – кластерное доменное имя или vIP-адрес сервера управления;
- 8D77-CEDC-433B – токен, полученный в консоли управления Кибер Бэкап для регистрации агента.

Подробное описание дополнительных возможностей и вариантов регистрации агентов см. в отдельной статье базы знаний.

Указатель