Данный документ применим к версиям **DRBD 8.4.x**

**Первоначальное подключение DRBD**

1. Текущая базовая конфигурация DRBD находится в […..](http://svn.edss.ee/sys/cmdb/techpool/drbd/apps/)
2. Отмонтировать диск с данными если он подмонтирован
3. Необходимо открыть порты 7701:7709, используя SuSEFirewall
4. Для защиты от ошибок необходимо сменить shared-secret в global\_common.conf
5. Для каждого подключаемого ресурса создать его описание в /etc/drbd.d/{RESOURCE}.res
6. Выложить systemd-скрипты, выполнить systemctl daemon-reload
7. Для создания диска с метаданными выполнить #drbdadm create-md storage
8. Выполнить на мастере
   1. drbdadm up all
   2. drbdadm primary --force all
9. Выполнить на слейве
   1. drbdadm up all
10. Дождаться окончания синхронизации
11. Опустить ресурс с обоих сторон #drbdadm down all
12. Выполнить на мастере   
    #systemctl enable replica-master\_storage  
    #systemctl start replica-master\_storage
13. Выполнить на слейве   
    #systemctl enable replica-slave\_storage  
    #systemctl start replica-slave\_storage
14. При смене ролей выполнять для перемещаемых ролей «systemctl disable»

**Запуск сервиса DRBD**

Не во всех операционных системах модуль ядра drbd загружается автоматически с системой. Чтобы исправить положение, необходимо выполнить ряд команд:

* modprobe drbd;
* echo «drbd» >> /etc/modprobe.d/drbd.conf

Главными критериями, которые напрямую влияют на скорость восстановления работы сервиса DRBD, являются:

* Используемый протокол режима репликации;
* Скорость ресинхронизации;
* Поддержка технологии Jumbo Frame серверами и связующим оборудованием;
* Параметр скорости ресинхронизации в конфигурации DRBD.
* Задержка и пропускная способность соединения между серверами;
* Скорость работы дисковых подсистем ведущего и ведомого серверов;
* Факторы, влияющие на производительность сетевого стека в системах виртуализации.

**Протоколы репликации данных**

**Протокол «C»** (синхронная репликация) — Является самым медленным и самым надёжным протоколом репликации, при котором запись считается завершенной, если она произведена на ведущем сервере и отреплицирована на ведомом.

**Потокол «B»** (полу-синхронная репликация) — Запись считается завершённой, если данные записаны на ведущий сервер и в буферный кэш ведомого.

**Потокол «A»** (асинхронная репликация) — Запись считается завершённой, если данные записаны на ведущий сервер и в локальный буфер TCP для отправки.

Результаты тестов выявили серьёзные задержки чтения/записи ведущего сервера при обрыве связи или нестабильном сетевом канале с ведомым сервером в режиме протокола «C». Исходя из этого, скорость восстановления и быстродействия решения зависит от чередования протоколов и/или выполнения необходимых команд, в зависимости от сложившейся ситуации.

**Обработчики**

Обработчики в DRBD используются в основном для автоматической попытки восстановления работоспособности системы и оповещения ответственного за эту систему лица о возможных или происшедших событиях. Важнейшей частью обработчиков является моментальное реагирование в критических для системы ситуациях, путём запуска пользовательских скриптов, привязанным к определённым событиям.

**Возможные ситуации и их решения**

**Разрыв линка: Ситуация №1**

Условия:

* Используемый протокол репликации: C;
* Скорость линии связи между серверами: 1 Gbit/s;
* Поддержка Jumbo Frame на серверах и промежуточном оборудовании: MTU 9000;
* Проблема: обрыв линии связи между серверами.

Текущий статус:

* Ведущий и ведомый серверы перешли в режим ожидания соединения WFConnection;
* Ведущий сервер продолжает работу, изменяя или сохраняя вновь поступившие данные, стараясь при этом соединиться с ведомым сервером.
* Замедление быстродействия подсистемы I/O DRBD, вследствие использования протокола «C».

Шаги для исправления:

* Изменение протокола репликации ведущего сервера с «С» на «A», вследствие чего скорость операций над локальными данными текущего сервера возрастёт.
* При восстановлении соединения между серверами, изменение протокола ведущего сервера на «С».

Конечный статус:

* Происходит сравнение контрольных сумм блоков данных ведущего и ведомого серверов;
* При обнаружении изменённых блоков производится автоматическая синхронизация;
* Система продолжает работать в штатном режиме.

**Отказ дисковой подсистемы: Ситуация №1**

Условия:

* Используемый протокол репликации: C;
* Скорость линии связи между серверами: 1 Gbit/s;
* Поддержка Jumbo Frame на серверах и промежуточном оборудовании: MTU 9000;
* Проблема: отказ дисковой подсистемы ведомого сервера.

Текущий статус:

* Режим дисковой подсистемы ведомого сервера изменился на Diskless.

Шаги для исправления (отдельный диск с мета-данными):

* Заменить испорченный носитель новым.
* Выполнить создание новых мета-данных *(drbdadm create-md <resource>)*
* Присоединить новый диск к DRBD *(drbdadm attach <resource>)*

Конечный статус:

* Режим дисковой подсистемы изменился на Inconsistent;
* Статус ведомого сервера принял SyncTarget; ведущего — SyncSource;
* Началась полная синхронизация с ведущего на ведомый сервер.

**Отказ дисковой подсистемы: Ситуация №2**

Условия:

* Используемый протокол репликации: C;
* Скорость линии связи между серверами: 1 Gbit/s;
* Поддержка Jumbo Frame на серверах и промежуточном оборудовании: MTU 9000;
* Проблема: отказ дисковой подсистемы ведущего сервера.

Текущий статус:

* Режим дисковой подсистемы ведущего сервера изменился на Diskless.

Шаги для исправления (отдельный диск с мета-данными):

* Заменить испорченный носитель новым.
* Выполнить создание новых мета-данных *(drbdadm create-md <resource>)*
* Присоединить новый диск к DRBD *(drbdadm**attach <resource>)*
* Сменить статус нового носителя *(drbdadm invalidate <resource>)*
* Указать роль ведущего сервера *(drbdadm primary <resource>)*

Конечный статус:

* Режим дисковой подсистемы изменился на Inconsistent;
* Статус ведущего сервера принял SyncTarget; ведомого — SyncSource;
* Началась полная синхронизация с ведомого на ведущий сервер.

Система DRBD имеет несколько ключевых параметров, для предотвращения ошибочных операций человека и самой системы в целом. Тем не менее, риск выхода системы из строя, всё же существует.

**Человеческий фактор: Ситуация №1**

Условия:

* Проблема: Во время потери соединения между ведущим и ведомым серверами, последний изменил свою роль в системе и стал ведущим.

Текущий статус:

* Ведущий и ведомый серверы перешли в режим ожидания соединения WFConnection;
* Ведущий сервер продолжает работу, изменяя или сохраняя вновь поступившие данные, стараясь при этом соединиться с ведомым сервером;

С этого момента существуют два варианта развития событий, исходя из условий:

Условия:

* Ведомый сервер был назначен ведущим. Блочное устройство было смонтировано в системе, но информация на носителе не менялась.
* Ведомый сервер был назначен ведущим. Блочное устройство было смонтировано в системе, информация на носителе изменилась.

Шаги для исправления (Условие №1):

* Размонтировать блочное устройство *(umount /dev/drbdX)*;
* Назначить серверу роль ведомого *(drbdadm secondary <resource>)*.

Конечный статус:

* Ведущий сервер отмечает ситуацию split-brain и, выбрав подходящий обработчик, пытается автоматически решить проблему;
* Ведущий сервер сравнивает контрольные суммы данных на локальном и ведомом серверах. Если они совпадают, система продолжает работу в штатном режиме.

Шаги для исправления (Условие №2):

* Размонтировать блочное устройство *(umount /dev/drbdX)*;
* Назначить серверу роль ведомого *(drbdadm secondary <resource>);*
* Выполнить валидацию носителя *(drbdadm invalidate <resource>).*

Конечный статус:

* Ведущий сервер отмечает ситуацию split-brain и, выбрав подходящий обработчик, пытается автоматически решить проблему;
* Статус ведущего сервера принял SyncTarget; ведомого — SyncSource;
* Выполняется полная синхронизация с ведущего сервера на ведомый.