



/// Общество с ограниченной ответственностью «Звезда»
ООО «Звезда»
125124, г. Москва, ул. 3-я Ямского Поля, д. 32
ОГРН 1217700119702
ИНН 7730263051/КПП 771401001
Тел.: 8 (495) 668-86-00

ОКПД2 62

ОКС 35.080

УТВЕРЖДАЮ

Генеральный директор

ООО «Звезда»

Мироненко Р.В.

«20» мая 2023 г.



ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

**ВСТРАИВАЕМОЙ ГИПЕРКОНВЕРГЕНТНОЙ ВИРТУАЛИЗАЦИИ,
ВКЛЮЧАЮЩЕЕ ГИПЕРВИЗОР 1ГО ТИПА И СИСТЕМУ УПРАВЛЕНИЯ
И МОНИТОРИНГА ГИПЕРКОНВЕРГЕНТНЫМ ВЫЧИСЛИТЕЛЬНЫМ
КЛАСТЕРОМ**

«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «ЗВЕЗДА» АЛЬКОР»

Руководство по эксплуатации ПО

RU.ВНРЯ.00003-01 ЭД

Оглавление

Введение.....	5
Терминология.....	6
Кластер.....	6
Узел.....	6
Экземпляр (VM).....	7
Шаблон диска.....	7
iAllocator.....	8
Концепция «Первичный» и «вторичный».....	9
Теги.....	9
Задания и опкоды.....	9
Начальные требования.....	9
Управление экземпляром.....	10
Добавление экземпляра.....	10
Регулярные операции с экземпляром.....	11
Удаление.....	11
Запуск/выключение.....	11
Запрос экземпляров.....	11
Изменение оперативной памяти экземпляра.....	12
Экспор/импорт.....	12
Импорт сторонних экземпляров.....	13
Выбор ядра экземпляра.....	13
Xen PVM.....	13
KVM.....	14
Функции высокой доступности экземпляра.....	14
Изменение основного узла.....	14
Отказ экземпляра.....	14
Живая миграция экземпляра.....	15
Перемещение экземпляра (офлайн).....	15
Дисковые операции.....	16
Подготовка к дисковым операциям.....	16
Восстановление избыточности для экземпляров на основе DRBD.....	16

Повторное создание дисков для избыточных экземпляров.....	17
Преобразование типа диска экземпляра.....	17
Экземпляры отладки.....	18
Доступ к дискам экземпляра.....	18
Доступ к консоли экземпляра.....	19
Другие операции с экземпляром.....	19
Перезагрузить.....	19
Отладка определений ОС экземпляра.....	19
Перемещение экземпляра.....	19
Управление сетью.....	20
Практика работы с alkor-network.....	20
Внешние компоненты.....	21
snf-сеть.....	22
nfdhcpd.....	22
Известные недостатки.....	22
Будущая работа.....	23
Узловые операции.....	23
Добавить/прочитатьДобавить.....	23
Изменение роли узла.....	23
Отказ главного узла.....	23
Переключение между другими ролями.....	23
Эвакуация узлов.....	24
Преобразование основного экземпляра.....	24
Эвакуация вторичного экземпляра.....	24
Удаление.....	25
Изменения сети репликации.....	25
Обработка хранения.....	25
Логические тома.....	26
Обобщенная обработка хранилища.....	26
Использование команд хранилища.....	27
Кластерные операции.....	27
Стандартные операции.....	27
Команды глобального узла.....	28
Проверка кластера.....	28

Перераспределение конфигурации.....	29
Переименование кластера.....	29
Очередь операций.....	30
Автоматическое обслуживание узла.....	31
Полное удаление кластера.....	31
Замена ключей SSH и SSL.....	32

Введение

Звезда Алькор — это программное обеспечение для управления кластером виртуализации. Вы должны быть системным администратором, знакомым с принципами работы операционных систем и средами виртуализации перед его использованием.

Различные компоненты Звезда Алькор имеют справочные страницы и интерактивную справку. Однако это руководство поможет вам ознакомиться с системой, объяснив наиболее распространенные операции, сгруппированные по назначению.

Терминология

В этом разделе представлено небольшое введение в терминологию Звезда Алькор, которое может оказаться полезным при чтении остальной части документа.

Кластер

Набор машин (узлов), которые взаимодействуют друг с другом, чтобы предложить согласованную, высокодоступную услугу виртуализации в рамках одного административного домена.

Узел

Физическая машина, входящая в состав кластера. Узлы — это базовая инфраструктура кластера, и они не должны быть отказоустойчивыми, чтобы обеспечить высокую доступность экземпляров.

Узел можно добавлять и удалять (если на нем нет экземпляров) по желанию из кластера. В кластере высокой доступности и только с экземплярами высокой доступности потеря любого отдельного узла не приведет к потере дисковых данных для любого экземпляра; конечно, сбой узла вызовет сбой его первичных экземпляров.

Узел, принадлежащий кластеру, может в данный момент времени находиться в одной из следующих ролей:

- главный узел, который является узлом, с которого управляется кластер
- главный узел-кандидат, только узлы в этой роли имеют полную конфигурацию и знания кластера, и только главные кандидаты могут стать главным узлом
- обычный узел, в котором большинство узлов находится в больших кластерах (> 20 узлов)
- истощенный узел, узлы в этом состоянии работают нормально, но не могут получать новые экземпляры; намерение состоит в том, что узлы в этой роли имеют некоторые проблемы, и они эвакуируются для ремонта оборудования.
- автономный узел, в котором есть запись в конфигурации кластера об узле, но демоны на главном узле не будут разговаривать с этим узлом; любые экземпляры, объявленные как имеющие автономный узел в качестве основного или вторичного, будут помечены как ошибка в операции проверки кластера.

В зависимости от роли каждый узел будет запускать набор демонов:

- демон `alkor-noded`, управляющий манипуляциями с аппаратными ресурсами этого узла; он работает на всех узлах, которые находятся в кластере
- демон `alkor-confd`, который работает на всех главных кандидатах и отвечает на запросы о конфигурации кластера.
- демон `alkor-wconfd`, который работает на главном узле и позволяет управлять кластером
- демон `alkor-luxid`, который работает на главном узле и отвечает на запросы, связанные с текущим состоянием кластера, и планирует задания кластера

- демон `alkor-rapi`, который работает на главном узле и предлагает API на основе HTTP для кластера
- демон `alkor-kvmd`, который запускается на всех узлах с поддержкой KVM, когда параметр кластера `user_shutdown` включен, и определяет, был ли экземпляр отключен пользователем или администратором.
- демон `alkor-mond`, который работает на всех узлах и собирает информацию о кластере
- демон `alkor-metad`, который запускается по запросу и предоставляет информацию о кластере сценариям или экземплярам установки ОС.

Помимо роли узла, есть и другие флаги узла, влияющие на его поведение:

- флаг `master_sarable` указывает, может ли узел когда-либо стать главным кандидатом; установка этого параметра на «нет» означает, что автоматическое продвижение никогда не сделает этот узел главным кандидатом; этот флаг может быть полезен для удаленного узла, который запускает только локальные экземпляры, и сделать его главным нецелесообразно из-за сетевых или других ограничений.
- флаг `vm_sarable` указывает, может ли узел размещать экземпляры или нет; например, можно использовать узел, не поддерживающий `vm_sarable`, просто как главный кандидат для резервного копирования конфигурации; установка этого флага в `no` запрещает размещение экземпляров этого узла, деактивирует гипервизор и связанные с ним проверки (например, проверки моста, проверка LVM и т. д.), а также удаляет его из вычислений емкости кластера.

Экземпляр (VM)

Виртуальная машина, работающая в кластере. Это может быть отказоустойчивый, высокодоступный объект.

Экземпляр VM имеет различные параметры, которые подразделяются на три категории: параметры, связанные с гипервизором (называемые `hparams`), общие параметры (называемые `berparams`) и параметры сетевой карты (называемые `nicparams`). Все эти параметры можно изменить либо на уровне экземпляра, либо с помощью значений по умолчанию на уровне кластера.

Шаблон диска

Существует несколько вариантов хранилища, предоставляемого экземпляру; в то время как экземпляр видит один и тот же виртуальный диск во всех случаях, конфигурация на уровне узла различается между ними.

Вы можете выбрать один из нескольких шаблонов дисков:

Бездисковый - в экземпляре нет дисков. Используется только для операционных систем специального назначения или для тестирования.

Файл - Экземпляр будет использовать простые файлы для своих дисков. Избыточность не предусмотрена, и это несколько сложнее настроить для высокой производительности.

Общий файл - Экземпляр будет использовать простые файлы в качестве серверной части, но Алькор предполагает, что эти файлы будут доступны и автоматически синхронизированы на всех узлах. Это позволяет выполнять

динамическую миграцию и аварийное переключение экземпляров с помощью этого метода.

Простой - Экземпляр будет использовать устройства LVM в качестве серверной части для своих дисков. Избыточность не предусмотрена.

DRBD - Между локальным узлом и удаленным устанавливается зеркало, которое необходимо указать вторым значением опции `-node`. Используйте этот параметр, чтобы получить экземпляр с высокой доступностью, который может быть переключен на удаленный узел в случае сбоя основного узла.

Примечание. Звезда Алькор не поддерживает стекированные устройства DRBD: установка стека DRBD не является полностью симметричной и поэтому не работает с динамической миграцией.

RBD - Экземпляр будет использовать тома внутри кластера RADOS в качестве серверной части для своих дисков. Он будет получать к ним доступ с помощью блочного устройства RADOS (RBD).

Gluster - Экземпляр будет использовать том Gluster для хранения экземпляра. Образы дисков будут храниться в каталоге `alkor/` верхнего уровня тома. Этот каталог будет создан автоматически для вас.

EXT - Экземпляр будет использовать внешний поставщик хранилища.

Примечание. Дисковые шаблоны, отмеченные звездочкой, требуют от Звезда Алькор доступа к файловой системе.

Пути по умолчанию, используемые Звезда Алькор:

Шаблон диска	Путь по умолчанию
Файл	<code>/srv/ganeti/file-storage</code>
Общий файл	<code>/srv/ganeti/shared-file-storage</code>
Gluster	<code>/var/run/ganeti/gluster</code>

Эти пути можно изменить во время инициализации `alkor`-кластера.

iAllocator

Платформа для использования внешних (предоставленных пользователем) сценариев для вычисления размещения экземпляров на узлах кластера. Это устраняет необходимость вручную указывать узлы при добавлении экземпляра, перемещении экземпляра, эвакуации узла и т. д.

Чтобы Звезда Алькор мог использовать эти скрипты, они должны быть размещены в каталоге `iallocator` (обычно это `lib/alkor/iallocators` с префиксом установки, например, `/usr/local`).

Концепция «Первичный» и «вторичный»

Экземпляр имеет первичный и, в зависимости от конфигурации диска, также может иметь вторичный узел. Экземпляр всегда работает на первичном узле и использует только его вторичный узел для репликации диска.

Точно так же термин первичный и вторичный экземпляры, когда речь идет об узле, относится к набору экземпляров, в которых данный узел является первичным или, соответственно, вторичным.

Теги

Теги — это короткие строки, которые можно прикрепить либо к самому кластеру, либо к узлам или экземплярам. Они полезны как очень упрощенное хранилище информации для помощи в администрировании кластера, например, путем прикрепления информации о владельце к каждому экземпляру после его создания:

```
$ alkor-instance add ... instance1
$ alkor-instance add-tags instance1 owner:user2
```

А затем, перечислив каждый экземпляр и его теги, эту информацию можно использовать для связи с пользователями каждого экземпляра.

Задания и опкоды

Хотя конечный пользователь не работает с этими сущностями напрямую, но ему полезно знать, что базовая операция кластера (например, запуск экземпляра) внутренне представлена в Звезда Алькор, как OpCode, (аббревиатура от кода операции). Эти коды операций выполняются как часть задания. Коды операций в одном задании обрабатываются Звезда Алькор последовательно, но разные задания будут обрабатываться (в зависимости от доступности ресурсов) параллельно. Они будут выполняться не в порядке отправки, а в зависимости от наличия ресурсов, блокировок и приоритета. Более раннее задание может ожидать блокировки, в то время как более новое задание не требует никаких блокировок и может выполняться сразу. Операции, требующие определенного порядка, должны быть отправлены как одно задание, или клиент должен отправлять одно задание за раз и ждать его завершения, прежде чем продолжить.

Например, выключение всего кластера можно выполнить, выполнив команду `alkor-instance shutdown --all`, которая отправит для каждого экземпляра отдельное задание, содержащее опкод «shutdown instance».

Начальные требования

Вам необходимо установить и настроить кластер Звезда Алькор, прежде чем вы попробуете любую из команд, описанных в этом документе. Пожалуйста, следуйте инструкциям по установке Звезда Алькор, чтобы узнать, как это сделать.

Управление экземпляром

Добавление экземпляра

Операция добавления может показаться сложной из-за множества параметров, которые она принимает, но как только вы поймете (несколько) обязательных параметров и возможностей настройки, вы увидите, что это простая операция.

Для операции добавления требуется как минимум пять параметров:

- ОС для экземпляра
- шаблон диска
- количество дисков и размер
- спецификация узла или, альтернативно, аллокатор для использования
- и, наконец, имя экземпляра

ОС для экземпляра должна быть видна в выводе команды `alkor-os list` и указывает, какую гостевую ОС следует установить на экземпляре.

Шаблон диска указывает, какое хранилище использовать в качестве серверной части для (виртуальных) дисков, представленных экземпляру; обратите внимание, что для экземпляров с несколькими виртуальными дисками все они должны быть одного типа.

Узлы, на которых будет работать экземпляр, могут быть заданы либо вручную с помощью параметра `-n`, либо автоматически вычислены Звезда Алькор, если вы установили какой-либо скрипт `iallocator`.

С учетом вышеуказанных параметров команда выглядит так:

```
$ alkor-instance add \
  -n TARGET_NODE:SECONDARY_NODE \
  -o OS_TYPE \
  -t DISK_TEMPLATE -s DISK_SIZE \
  INSTANCE_NAME
```

Имя экземпляра должно быть разрешимым (например, существовать в DNS) и обычно указывает на адрес в той же подсети, что и сам кластер.

Приведенная выше команда имеет минимально необходимые параметры; другие варианты, которые вы можете указать, включают, среди прочего:

Максимальный/минимальный размер памяти (`-B maxmem`, `-B minmem`) (`-B memogu` может использоваться для указания только одного размера)

Количество виртуальных процессоров (`-B vcpus`)

Аргументы для сетевых адаптеров экземпляра; по умолчанию создается экземпляр с одной сетевой картой. IP-адрес и/или мост сетевой карты можно изменить с помощью `--net 0:ip=IP,link=BRIDGE`

Например, если вы хотите создать высокодоступный экземпляр с одним диском объемом 50 ГБ и размером памяти по умолчанию, с первичным узлом `node1` и вторичным узлом `node3`, используйте следующую команду:

```
$ alkor-instance add -n node1:node3 -o debootstrap -t drbd -s 50G \
```

`instance1`

Существует также команда для пакетного создания экземпляров из файла спецификации, см. операцию пакетного создания на странице руководства `alkor-instance`.

Регулярные операции с экземпляром

Удаление

Удалить экземпляр даже проще, чем создать его. Эта операция необратима и уничтожает все содержимое вашего экземпляра. Используйте с осторожностью:

```
$ alkor-instance remove INSTANCE_NAME
```

Запуск/выключение

Экземпляры автоматически запускаются во время создания экземпляра. Чтобы вручную запустить тот, который в данный момент остановлен, вы можете запустить:

```
$ alkor-экземпляр startup INSTANCE_NAME
```

Звезда Алькор запустит экземпляр с максимальной памятью экземпляра. Если памяти недостаточно, Звезда Алькор будет использовать всю доступную память вплоть до минимальной памяти экземпляра. Если даже этот объем памяти не свободен, Ганети откажется запускать экземпляр.

Обратите внимание, что это не будет работать, когда экземпляр находится в постоянно остановленном автономном состоянии. В этом случае вам сначала нужно будет вернуть его в онлайн-режим, выполнив:

```
$ alkor-instance modify --online INSTANCE_NAME
```

Команда для остановки запущенного экземпляра:

```
$ alkor-instance shutdown INSTANCE_NAME
```

Если вы хотите завершить работу экземпляра на постоянной основе, чтобы он не требовал динамически выделяемых ресурсов (памяти и `vsrus`), после завершения работы экземпляра выполните следующее:

```
$ alkor-instance modify --offline INSTANCE_NAME
```

Предупреждение Не используйте команды `Xen` или `KVM` напрямую для остановки экземпляров. Если вы запустите, например, `xm shutdown` или `xm destroy` на экземпляре, Звезда Алькор автоматически перезапустит его (через команду Звезда Алькор-`watcher(8)`, которая запускается через `cron`).

Экземпляры также могут быть отключены пользователем из самого экземпляра, и в этом случае они будут помечены соответствующим образом, и Звезда `alkor-watcher(8)` не перезапустит их. Подробности смотрите в `alkor-cluster(8)`.

Запрос экземпляров

Есть два способа получить информацию об экземплярах: список экземпляров, который выводит табличный вывод, содержащий заданный набор полей о каждом экземпляре, и запрос подробной информации о наборе экземпляров.

Команда для просмотра всех настроенных экземпляров и их статуса:

```
$ alkor-instance list
```

Команда может возвращать пользовательский набор информации при использовании параметра -o (как всегда, смотрите подробную спецификацию на странице руководства). Каждый экземпляр будет представлен в виде строки, что упрощает анализ этого вывода с помощью обычных утилит оболочки (grep, sed и т. д.).

Чтобы получить более подробную информацию об экземпляре, вы можете запустить:

```
$ alkor-instance info INSTANCE
```

который даст многострочный блок информации об экземпляре, его аппаратных ресурсах (особенно о его дисках и их статусе избыточности) и т. д. Это сложнее анализировать и дороже, чем операция списка, но возвращает гораздо более подробную информацию.

Изменение оперативной памяти экземпляра

Звезда Алькор всегда будет следить за тем, чтобы экземпляр имел значение между его максимальным и минимальным объемом памяти, доступной в качестве памяти времени выполнения. Начиная с версии 2.6 Звезда Алькор будет выбирать размер, отличный от максимального, только при запуске, сбое или переносе экземпляра на узле с объемом доступной памяти меньше максимального. Он не будет изменять размер других экземпляров, чтобы освободить место для экземпляра.

Если вы обнаружите, что вам нужно больше памяти на узле, размер любого экземпляра можно изменить вручную без простоя с помощью команды:

```
$ alkor-instance modify -m ИМЯ_ЭКЗЕМПЛЯРА РАЗМЕРА
```

Эту же команду можно использовать для увеличения памяти, доступной на экземпляре, при условии, что на его узле доступно достаточно свободной памяти и указанный размер не превышает максимальный объем памяти, который был у экземпляра при первой загрузке (экземпляр не сможет увидеть новую память выше максимального, указанного для гипервизора во время загрузки, если ему необходимо увеличить объем памяти в дальнейшем). перезагрузка становится необходимой).

Экспорт/импорт

Вы можете создать моментальный снимок диска экземпляра и его конфигурации Звезда Алькор, а затем сделать резервную копию или импортировать его в другой кластер. Способ экспорта экземпляра:

```
$ alkor-backup export -n TARGET_NODE INSTANCE_NAME
```

Целевым узлом может быть любой узел в кластере с достаточным пространством в каталоге /srv/alkor для хранения образа экземпляра. Используйте параметр --noshutdown для моментального снимка экземпляра без его перезагрузки. Обратите внимание, что Звезда Алькор хранит только один снимок для экземпляра — любой предыдущий снимок того же экземпляра, существующий на уровне кластера в /srv/alkor, будет удален этой операцией: если вы хотите сохранить их, вам нужно переместить их из Звезда Алькор. каталог экспорта.

Импорт экземпляра аналогичен созданию нового, но дополнительно необходимо указать расположение снимка. Команда:

```
$ alkor-backup import -n TARGET_NODE \
  --src-node=УЗЕЛ --src-dir=КАТАЛОГ ЭКЗЕМПЛЯР_ИМЯ
```

По умолчанию параметры будут считываться из информации об экспорте, но вы, конечно, можете передать их через командную строку — здесь также поддерживается большинство опций, доступных для команды `alkor-instance add`.

Импорт сторонних экземпляров

Существует возможность импортировать внешний экземпляр, данные диска которого уже хранятся в виде томов LVM, не выполняя их копирование: режим адаптации диска.

Для этого убедитесь, что исходный неуправляемый экземпляр остановлен, затем создайте экземпляр Звезда Алькор обычным способом, за исключением того, что вместо передачи информации о диске вы указываете текущие тома:

```
$ alkor-instance add -t plain -n HOME_NODE ... \
  --disk 0:adopt=lv_name[,vg=vg_name] INSTANCE_NAME
```

Это возьмет на себя заданные логические тома, переименует их в стандарт Звезда Алькор (на основе UUID) и, не устанавливая на них ОС, запустит непосредственно экземпляр. Если вы настроите гипервизор аналогично неуправляемой конфигурации, которая была у экземпляра, переход должен быть бесшовным для экземпляра. Для более чем одного диска просто передайте другой параметр диска (например, `--disk 1:adopt=...`).

Выбор ядра экземпляра

Ядро, которое экземпляры используют для загрузки, может исходить либо от узла, либо от самих экземпляров, в зависимости от настройки.

Xen PVM

С Xen PVM есть три варианта.

Во-первых, вы можете использовать ядро из ноды, установив параметры гипервизора как таковые:

- `kernel_path` к действительному файлу на узле (и, соответственно, `initrd_path`)
- Для `kernel_args` необязательно установлено допустимое значение параметра Linux (например, `ro`)
- `root_path` к допустимому параметру (например, `/dev/xvda1`)
- `bootloader_path` и `bootloader_args` для очистки

В качестве альтернативы вы можете делегировать управление ядром экземплярам и использовать либо `rvgrub`, либо устаревший `rugrub`. Для этого вы должны установить ядра и `initrds` в экземпляр и создать действительный файл конфигурации GRUB v1.

Для `rvgrub` (новое в версии 2.4.2) нужно установить:

- `kernel_path` для указания на загрузчик `rvgrub`, присутствующий на узле (например, `/usr/lib/xen/boot/pv-grub-x86_32.gz`)

- `kernel_args` на путь к конфигурационному файлу GRUB относительно экземпляра (например, `(hd0,0)/grub/menu.lst`)
 - `root_path` должен быть пустым
- `bootloader_path` и `bootloader_args` для очистки

Хотя `pygrub` устарел, вот как вы можете его настроить:

- `bootloader_path` к двоичному файлу `pygrub` (например, `/usr/bin/pygrub`)
- остальные настройки не важны

Дополнительную информацию можно найти на вики-страницах Xen для `pvgrub` и `pygrub`.

KVM

Для KVM также можно загрузить ядро любым способом.

Для загрузки ядер с ноды нужно установить:

- `kernel_path` в допустимое значение
- `initrd_path` необязательно устанавливается, если вы используете `initrd`
- `kernel_args` необязательно устанавливается в допустимое значение (например, `ro`)

Если вместо этого вы хотите, чтобы экземпляр загружался со своего диска (и выполнял свой загрузчик), просто установите для параметра `kernel_path` пустую строку, а все остальные будут игнорироваться.

Функции высокой доступности экземпляра

Примечание. Этот раздел относится только к многоузловым кластерам.

Изменение основного узла

Существует три способа обмена первичными и вторичными узлами экземпляра; правильный выбор зависит от того, как экземпляр был создан, и от состояния его текущего основного узла. Информацию об изменении вторичного узла см. в разделе Восстановление избыточности для экземпляров на основе DRBD. Обратите внимание, что можно изменить только первичный узел на вторичный и наоборот; прямая замена основного узла третьим узлом при сохранении текущего вторичного узла невозможна за один шаг, только с помощью нескольких операций, как подробно описано в разделе Перемещение экземпляра.

Отказ экземпляра

Если экземпляр построен в режиме высокой доступности, вы можете в любой момент переключиться на его вторичный узел, даже если первичный по каким-то причинам вышел из строя и больше не работает. Сделать это очень просто, на главном узле вы можете просто запустить:

```
$ alkor-instance failover INSTANCE_NAME
```

Этоэто. После выполнения команды вторичный узел становится первичным, и наоборот.

Экземпляр будет запущен с объемом памяти между его значением `maxmem` и `minmem`, в зависимости от свободной памяти на его целевом узле, или операция завершится ошибкой, если это невозможно. Дополнительные сведения см. в разделе `Запуск/выключение`.

Если шаблон диска экземпляра имеет тип `gbd`, то вы можете явно указать целевой узел (который может быть любым узлом) или указать подключаемый модуль `iallocator`. Если вы опустите оба параметра, для определения целевого узла будет использоваться иаллокатор по умолчанию:

```
$ alkor-instance failover -n TARGET_NODE INSTANCE_NAME
```

Живая миграция экземпляра

Если экземпляр построен в режиме высокой доступности, в настоящее время он работает и оба его узла работают нормально, вы можете перенести его на дополнительный узел без простоев. На мастер-узле нужно запустить:

```
$ alkor-instance migrate INSTANCE_NAME
```

Текущая нагрузка на экземпляр и объем его памяти будут влиять на продолжительность миграции. В любом случае, как для гипервизоров KVM, так и для гипервизоров Xen миграция будет прозрачна для инстанса.

Если на целевом узле меньше памяти, чем текущая оперативная память экземпляра, но, по крайней мере, минимальная доступная память экземпляра, Звезда Алькор автоматически уменьшит оперативную память экземпляра перед его миграцией, если только не будет передан параметр `--no-runtime-changes`, и в этом случае на целевом узле должна быть свободна как минимум текущая рабочая память экземпляра.

Если шаблон диска экземпляра имеет тип `gbd`, то вы можете явно указать целевой узел (который может быть любым узлом) или указать подключаемый модуль `iallocator`. Если вы опустите оба параметра, для определения целевого узла будет использоваться иаллокатор по умолчанию:

```
$ alkor-instance migrate -n TARGET_NODE INSTANCE_NAME
```

Перемещение экземпляра (офлайн)

Если экземпляр не был создан как зеркальный, то единственный способ изменить его первичный узел — выполнить команду перемещения:

```
$ alkor-instance move -n NEW_NODE INSTANCE
```

Для этого есть несколько предпосылок:

- экземпляр должен быть остановлен
- его текущий первичный узел должен быть в сети и работоспособен
- на дисках экземпляра не должно быть ошибок

Поскольку эта операция фактически копирует данные со старого узла на новый узел, ожидайте, что она будет пропорциональна размеру дисков экземпляра и скорости как системы ввода-вывода узлов, так и их сети.

Дисковые операции

Отказы дисков являются распространенной причиной ошибок при развертывании любого сервера. Звезда Алькор предлагает защиту от сбоя одного узла, если ваши экземпляры были созданы в режиме высокой доступности, а также предлагает способы восстановления избыточности после сбоя.

Подготовка к дисковым операциям

Важно отметить, что для того, чтобы Звезда Алькор мог выполнять любые дисковые операции, машины Linux, на которых работает Звезда Алькор, должны быть согласованы; для LVM это означает, что команды LVM не должны возвращать ошибки; обычно после полного отказа диска любая команда LVM прерывается с ошибкой, похожей на:

```
$ vgs
```

```
/dev/sdb1: ошибка чтения после 0 из 4096 в 0: ошибка ввода/вывода
```

```
/dev/sdb1: ошибка чтения после 0 из 4096 по адресу 750153695232: ошибка ввода/вывода
```

```
/dev/sdb1: ошибка чтения после 0 из 4096 в 0: ошибка ввода/вывода
```

```
Не удалось найти устройство с uuid «t30jmN-4Rcf-Fr5e-CURS-pawt-z0jU-m1Tgej».
```

```
Не удалось найти все физические тома для группы томов xenvg.
```

Перед восстановлением работоспособного состояния дисков экземпляра необходимо исправить группу томов, используемую Звезда Алькор, чтобы мы могли фактически создавать логические тома и управлять ими. Обычно это делается в несколько этапов:

во-первых, если диск полностью исчез и команды LVM завершаются с «Не удалось найти устройство с uuid...», вам нужно выполнить команду:

```
$ vgreduce --removemissing VOLUME_GROUP
```

после приведенной выше команды команды LVM должны выполняться нормально (предупреждения нормальны, но команды не завершатся полностью).

если неисправный диск все еще виден в выводе команды `pvs`, вам необходимо деактивировать его из выделения, выполнив:

```
$ pvs -x n /dev/ДИСК
```

На этом этапе группа томов должна быть согласованной, и любые поврежденные физические тома больше не должны быть доступны для выделения.

Восстановление избыточности для экземпляров на основе DRBD

Экземпляр DRBD имеет два узла, и на одном из них произошел сбой хранилища. В зависимости от того, какой узел (первичный или вторичный) вышел из строя, у вас есть три варианта:

если хранилище на первичном узле вышло из строя, нужно пересоздать на нем диски

если хранилище на вторичном узле отказало, вы можете либо пересоздать диски на нем, либо изменить вторичное и воссоздать избыточность на новом вторичном узле.

Конечно, в любой момент можно принудительно пересоздать диски, хотя все уже хорошо.

Во всех трех случаях можно использовать операцию замены дисков:

```
# пересоздаем диски на первичном узле
$ alkor-instance replace-disks -p INSTANCE_NAME

# пересоздаем диски на текущем вторичном
$ alkor-instance replace-disks -s INSTANCE_NAME

# изменить вторичный узел, через ручную спецификацию
$ alkor-instance replace-disks -n NODE INSTANCE_NAME

# изменить вторичный узел с помощью скрипта iallocator
$ alkor-instance replace-disks -I SCRIPT INSTANCE_NAME

# начиная с Звезда Алькор 2.1: автоматически исправлять первичный или
вторичный узел
$ alkor-instance replace-disks -a INSTANCE_NAME
```

Поскольку процесс включает в себя копирование всех данных с рабочего узла на целевой узел, это займет некоторое время, в зависимости от размера диска экземпляра, системы ввода-вывода узла и скорости сети. Но это (за исключением любого прерывания сети) полностью прозрачно для экземпляра.

Повторное создание дисков для неизбыточных экземпляров

Для неизбыточных экземпляров нет копии (кроме резервных копий) для повторного создания дисков. Но можно, по крайней мере, пересоздать пустые диски, после чего можно запустить переустановку с помощью команды `recreate-disks`:

```
$ alkor-instance recreate-disks INSTANCE
```

Обратите внимание, что это не удастся, если диски уже существуют. Экземпляр может быть назначен новым узлам автоматически, указав `iallocator` с помощью опции `--iallocator`.

Преобразование типа диска экземпляра

Можно преобразовать между неизбыточным экземпляром простого типа (хранилище LVM) и избыточным `drbd` с помощью команды модификации `alkor-instance`:

```
# начать с неизбыточного экземпляра
$ alkor-instance add -t plain ... ЭКЗЕМПЛЯР

# позже преобразовать его в избыточный
$ alkor-instance stop INSTANCE
```

```
$ alkor- instance modify -t drbd -n НОВЫЙ_ВТОРИЧНЫЙ_ЭКЗЕМПЛЯР
```

```
$ alkor- instance start INSTANCE
```

```
# и преобразовать его обратно
```

```
$ alkor- instance stop INSTANCE
```

```
$ alkor-instance modify -t простой INSTANCE
```

```
$ alkor- instance start INSTANCE
```

Преобразование должно выполняться, когда экземпляр остановлен, а преобразование из обычного шаблона в шаблон drbd представляет небольшой риск, особенно если экземпляр имеет несколько дисков и/или если один узел выходит из строя во время процедуры преобразования). Таким образом, рекомендуется (как всегда) убедиться, что время простоя для ручного восстановления приемлемо и что у экземпляра есть актуальные резервные копии.

Экземпляры отладки

Доступ к дискам экземпляра

С основного узла экземпляра вы можете получить доступ к его диску. Никогда не подключайте базовый логический том вручную к отказоустойчивому экземпляру, иначе репликация нарушится, и ваши данные будут несогласованными. Правильный способ получить доступ к дискам экземпляра — запустить (как обычно, на главном узле) команду:

```
$ alkor-instance activate-disks INSTANCE
```

А затем на основном узле экземпляра получите доступ к созданному устройству. Например, вы можете монтировать данные диски, затем редактировать файлы в файловой системе и т. д.

Обратите внимание, что с разбитыми на разделы дисками (в отличие от файловых систем всего диска) вам потребуется использовать такой инструмент, как kpartx(8):

```
# на узле1
```

```
$ alkor-instance активировать диски экземпляра1
```

```
узел3:диск/0:...
```

```
$ ssh-узел3
```

```
# на узле 3
```

```
$ kpartx -l /dev/...
```

```
$ kpartx -a /dev/...
```

```
$ смонтировать /dev/mapper/... /mnt/
```

```
# редактируем файлы под mnt по желанию
```

```
$ сумма /млн/
```

```
$ kpartx -d /dev/...
```

```
$ выход
```

```
# вернуться к узлу 1
```

После того, как вы закончите, вы можете деактивировать их с помощью команды `deactivate-disks`, которая работает точно так же:

```
$ alkor-instance deactivate-disks INSTANCE
```

Обратите внимание, что если какой-либо процесс, запущенный вами, все еще использует диски, приведенная выше команда приведет к ошибке, и вы должны выполнить очистку и убедиться, что указанная выше команда успешно выполняется, прежде чем запускать экземпляр, иначе экземпляр будет поврежден.

Доступ к консоли экземпляра

Команда для доступа к консоли работающего экземпляра:

```
$ alkor- instance консоли INSTANCE_NAME
```

Используйте консоль как обычно, а затем введите `^]`, когда закончите, чтобы выйти.

Другие операции с экземпляром

Перезагрузить

Существует команда-оболочка для перезагрузки экземпляров:

```
$ alkor- instance reboot экземпляр2
```

По умолчанию это эквивалентно выключению и последующему запуску экземпляра, но принимает параметры для выполнения мягкой перезагрузки (через гипервизор), жесткой перезагрузки (отключение гипервизора и затем запуск) или полной перезагрузки (по умолчанию, который также отменяет конфигурацию, а затем снова настраивает диски экземпляра).

Отладка определений ОС экземпляра

Если у вас есть какие-либо проблемы с операционными системами экземпляра, команда для просмотра полного состояния всех ваших узлов:

```
$ alkor-os diagnose
```

Перемещение экземпляра

Хотя невозможно переместить экземпляр из узлов (A, B) в узлы (C, D) за один ход, это можно сделать за несколько шагов:

```
# экземпляр расположен на A, B
```

```
$ alkor-instance replace-disks -n nodeC instance1
```

```
# экземпляр перемещен из (A, B) в (A, C)
```

```
# теперь мы меняем первичный/вторичный узлы
```

```
$ alkor- instance миграции экземпляр1
```

```
# экземпляр живет на (C, A)
```

```
# затем мы можем изменить A на D с помощью:
```

```
$ alkor-instance replace-disks -n nodeD instance1
```

Что приводит его к окончательной конфигурации (C, D). Обратите внимание, что нам нужно было выполнить две операции замены дисков (две копии экземплярных дисков), потому что нам нужно было избавиться от обоих исходных узлов (A и B).

Управление сетью

Звезда Алькор использовал для описания сетевых адаптеров экземпляра с IP-адресом, MAC-адресом, каналом подключения и режимом. У него было три основных недостатка:

- не было простого способа назначить экземпляру уникальный IP
- информация о сети (подсеть, шлюз, домен и т. д.) недоступна на целевом узле (`kvm-ifup`, хуки и т. д.)
- следует явно передавать информацию L2 (режим и ссылку) каждому сетевому адаптеру

Кроме того, не было простого способа получить текущий обзор сети (какие экземпляры находятся в одной сети L2 или L3, какие IP-адреса зарезервированы и т. д.).

Для всего вышеперечисленного требуется внешний инструмент управления, который имеет общий вид и предоставляет Ганети соответствующую информацию.

`alkor-network` стремится поддерживать большую часть этой функциональности внутри Звезда Алькор и абстрагировать сеть как отдельный объект. В настоящее время сеть Звезда Алькор обеспечивает следующее:

- Единый пул IPv4, подсеть и шлюз
- Информация о подключении для каждой группы узлов (режим, ссылка)
- Префикс MAC для каждой сетевой карты внутри сети
- Префикс IPv6/шлюз, связанный с этой сетью
- Теги

Управление пулом IP-адресов обеспечивает уникальность IP-адресов внутри этой сети. Пользователь может пройти `ip=pool,network=test` и будет:

1. Получить первый доступный IP в пуле
2. Наследовать режим подключения и ссылку сетевых параметров сети
3. NIC получит префикс MAC сети
4. Вся информация, связанная с сетью, будет доступна как переменные среды в сценариях и ловушках `kvm-ifup`, чтобы они могли динамически управлять всеми настройками, связанными с сетью, на хосте.

Практика работы с `alkor-network`

Для создания сети выполните:

```
# alkor-network add --network=192.0.2.0/24 --gateway=192.0.2.1 test
```

Пожалуйста, просмотрите все другие доступные параметры (-add-reserved-ips, -mac-prefix, -network6, -gateway6, -tags).

В настоящее время информация IPv6 не используется самой Звезда Алькор. Он экспортируется только в сценарии конфигурации NIC и ловушки через переменные среды.

Чтобы сделать эту сеть доступной в группе узлов, вы должны указать режим подключения и ссылку во время подключения:

```
# alkor-network connect --nic-parameters mode=bridged,link=br100 test default nodegroup1
```

Чтобы добавить сетевой адаптер внутри этой сети:

```
# alkor-instance изменить --net -1:add,ip=pool,network=test inst1
```

Это позволит сетевой карте получить уникальный IP-адрес внутри этой сети и унаследовать сетевые параметры группы узлов (bridged, br100). IP здесь необязателен. Если сетевой адаптер отсутствует, он просто получит информацию L2.

Чтобы переместить существующую сетевую карту из сети в другую и удалить ее IP-адрес:

```
# alkor-instance изменить --net -1:ip=none,network=test1 inst1
```

Это освободит старый IP-адрес из старого пула IP-адресов, и сетевая карта унаследует новые параметры nicparams.

На вышеуказанные действия есть дополнительная опция - no-conflicts-check. Это не проверяет конфликтующие настройки. Конкретно:

1. При добавлении сети IP-адреса узлов и мастера не проверяются.
2. При подключении сети к группе узлов IP-адреса экземпляров внутри этой группы узлов не проверяются, находятся ли они в подсети или нет.
3. При явном указании IP-адреса без передачи сети Звезда Алькор не будет проверять, включен ли этот IP-адрес в какую-либо доступную сеть в группе узлов.

Внешние компоненты

Все вышеупомянутые шаги обеспечивают настройку сетевой карты с точки зрения Звезда Алькор. Конечно, это не имеет никакого отношения к тому, как экземпляр в конечном итоге получит желаемое подключение (IPv4, IPv6, маршруты по умолчанию, информация DNS и т. д.) и где будет разрешен IP-адрес. Эта функциональность управляется внешними компонентами.

Предположим, что виртуальной машине потребуется получить динамический IP-адрес через DHCP, получить адрес SLAAC и использовать DHCPv6 для другой информации о конфигурации (в случае, если RFC-6106 не поддерживается клиентом, например, Windows). Это означает, что необходимы следующие внешние службы:

1. DHCP-сервер
2. Маршрутизатор IPv6, отправляющий объявления маршрутизатора

3. Сервер DHCPv6, экспортирующий информацию DNS
4. Динамический DNS-сервер

Эти компоненты должны быть настроены динамически и для каждой сетевой карты. Это можно сделать с помощью пользовательских сценариев и ловушек `kvm-ifup`.

snf-сеть

Пакет `snf-network` [1,3] включает в себя пользовательские скрипты, обеспечивающие вышеупомянутую функциональность. `kvm-vif-bridge` и `vif-custom` — это альтернатива `kvm-ifup` и `vif-Звезда Алькор`, которые учитывают всю экспортируемую сетевую информацию. Их действия зависят от сетевых тегов. Конкретно:

`dns`: обновит внешний сервер DDNS (`nsupdate` на сервере привязки)

`ip-less-routed`: настроит маршруты, правила и прокси-ARP. Эта настройка предполагает уже существующую таблицу маршрутизации вместе с некоторой локальной конфигурацией и обеспечивает подключение к экземплярам через внешний шлюз/маршрутизатор, не требуя, чтобы узлы имели IP-адрес внутри этой сети.

`private-filter`: установит правила `eatables`, чтобы обеспечить изоляцию L2 на общем мосту. Только пакеты с одинаковым префиксом MAC будут пересылаться на соответствующий виртуальный интерфейс.

`nfdhcpd`: обновит внешний DHCP-сервер

nfdhcpd¶

`snf-network` работает с `nfdhcpd` [2,3]: DHCP-сервером пользовательского пространства на основе NFQUEUE. В настоящее время `nfdhcpd` отвечает на запросы BOOTP/DHCP, исходящие от крана или моста. Кроме того, в случае настройки с маршрутизацией он обеспечивает конфигурацию без сохранения состояния, отвечая на запросы маршрутизатора и соседей вместе с запросами DHCPv6 для параметров DNS. Его БД динамически обновляется с помощью текстовых файлов внутри локального каталога с помощью `inotify` (`snf-network` просто добавляет Файл привязки C со всей необходимой информацией, если найден соответствующий сетевой тег). Тем не менее, нам нужно манипулировать всеми этими пакетами и отправлять их в соответствующую NFQUEUE.

Известные недостатки

В настоящее время известны следующие слабые места проектирования и реализации `alkor-network`:

- Невозможно определить сеть без пула IP-адресов
- Пул определяет размер сети
- Зарезервированные IP-адреса должны быть определены явно (неудобно для большого диапазона)
- Невозможно определить сеть только для IPv6

Будущая работа

Любые предстоящие исправления должны быть нацелены на:

- Отдельные L2, L3, IPv6, информация о пуле IP
- Поддержка набора пулов IP-адресов для каждой сети
- Заставить IP/сеть в объекте NIC принимать список записей
- Внедрить внешние сценарии для конфигурации узла (динамически создавать/удалять мосты/маршруты при подключении/отключении сети)

Узловые операции

Доступно гораздо меньше операций с узлами, чем для экземпляров, но они одинаково важны для поддержания работоспособности кластера.

Добавить/прочитатьДобавить

В любой момент можно расширить кластер еще одним узлом, используя операцию добавления узла:

```
$ alkor-node add NEW_NODE
```

Если в кластере определена сеть репликации, вам необходимо передать параметр `-s REPLICATION_IP` в эту опцию.

Вариант этой команды можно использовать для перенастройки узла, если его конфигурация Звезда Алькор нарушена, например, если он был переустановлен по ошибке:

```
$ alkor-node add --readd EXISTING_NODE
```

Это приведет к повторной инициализации узла, как если бы он был недавно добавлен, но с сохранением его существующей конфигурации в кластере (основной/дополнительный IP-адрес и т. д.), другими словами, вам не нужно будет использовать здесь `-s`.

Изменение роли узла

Узел может играть разные роли, как описано в разделе терминологии Звезда Алькор. Повышение роли узла до главной роли является особым, в то время как другие роли обрабатываются с помощью одной команды.

Отказ главного узла

Если вы хотите повысить роль другого узла до роли мастера (по какой-либо причине), запустите на любом другом узле-кандидате команду:

```
$ alkor-cluster master-failover
```

и узел, на котором вы его запустили, теперь является новым мастером. Если вы попытаетесь запустить это на узле, не являющемся главным кандидатом, вы получите сообщение об ошибке, указывающее, какие узлы действительны.

Переключение между другими ролями

Для выбора новой роли можно использовать команду модификации `alkor-node`:

```
# изменить на главного кандидата
```

```

$ alkor-node modify -C да УЗЕЛ
# изменить статус на слив
$ alkor-node modify -D да УЗЕЛ
# изменить статус в автономном режиме
$ alkor-node modify -O да УЗЕЛ
# перейти в обычный режим (сбросить все флаги)
$ alkor-node modify -O нет -D нет -C нет УЗЕЛ

```

Обратите внимание, что для кластера требуется, чтобы в любой момент времени определенное количество узлов являлось главными кандидатами, поэтому переход от главного кандидата к другим ролям может завершиться ошибкой. Рекомендуется либо принудительно выполнить операцию (с помощью параметра `--force`), либо сначала изменить количество мастер-кандидатов в кластере — см. Стандартные операции.

Эвакуация узлов

Существует два этапа перемещения экземпляров с узла:

- перемещение первичных экземпляров (фактически преобразование их во вторичные экземпляры)
- перемещение вторичных экземпляров (включая любые экземпляры, преобразованные на шаге выше)

Преобразование основного экземпляра

На этом шаге вы можете использовать команды перемещения отдельных экземпляров (как показано в разделе «Изменение основного узла») или групповые версии для каждого узла; это:

```

$ alkor-node migrate УЗЕЛ
$ alkor-node evacuate -s УЗЕЛ

```

Обратите внимание, что команда экземпляра «переместить» в настоящее время не имеет эквивалента узла.

Обе эти команды или эквивалентная команда для каждого экземпляра сделают этот узел вторичным узлом для соответствующих экземпляров, тогда как их текущий вторичный узел станет первичным. Обратите внимание, что невозможно за один шаг изменить первичный узел на другой узел в качестве основного, сохраняя при этом тот же вторичный узел.

Эвакуация вторичного экземпляра

Для эвакуации вторичных экземпляров предусмотрена команда `alkor-node evacuate` со следующим синтаксисом:

```

$ alkor-node evacuate -I IALLOCATOR_SCRIPT NODE
$ alkor-node evacuate -n DESTINATION_NODE NODE

```


Первая версия будет вычислять новый вторичный экземпляр для каждого экземпляра по очереди, используя заданный скрипт `iallocator`, тогда как вторая просто переместит все экземпляры в `DESTINATION_NODE`.

Удаление

Когда у узла больше нет экземпляров (ни основных, ни дополнительных), его легко удалить из кластера:

```
$ alkor-node remove NODE_NAME
```

Это деконфигурирует узел, остановит на нем демоны Звезда Алькор и, надеюсь, оставит его таким, каким он был до присоединения к кластеру.

Изменения сети репликации

Команда `alkor-node modify -s` может использоваться для изменения вторичного IP-адреса узла. Эта операция может быть выполнена только в том случае, если:

- На целевом узле нет активных экземпляров
- Новый целевой IP-адрес доступен с вторичного IP-адреса мастера.

Также эта операция не позволит изменить узел с односетевое (один и тот же первичный и вторичный IP-адрес) на многосетевой (отдельная сеть репликации) или наоборот, за исключением случаев, когда:

- Целевой узел — мачтаер узел и `-force` переданы.
- Целевой кластер является односетевым, а новый первичный IP-адрес является изменением на однодомный для определенного узла.
- Целевой кластер является многосетевым, и новый первичный IP-адрес является изменением на многосетевой для определенного узла.

Например, чтобы сделать односетевое преобразование в многосетевое:

```
$ alkor-node изменить --force -s SECONDARY_IP MASTER_NAME
```

```
$ alkor-node изменить -s SECONDARY_IP NODE1_NAME
```

```
$ alkor-node изменить -s SECONDARY_IP NODE2_NAME
```

```
$ alkor-node изменить -s SECONDARY_IP NODE3_NAME
```

...

Те же команды можно использовать для многосетевого и односетевое подключения, за исключением того, что в этом случае вторичные IP-адреса должны быть такими же, как первичные для каждого узла.

Обработка хранения

При использовании LVM (автономно или с DRBD) отладка и исправление в случае ошибок может стать утомительным. Более того, даже файловое хранилище может стать сложным для ручной обработки на многих хостах. Звезда Алькор предоставляет несколько команд для автоматизации.

Логические тома

Это команда, специфичная для обработки LVM. Это позволяет перечислить логические тома на данном узле или на всех узлах и их связь с экземплярами с помощью команды Volumes:

```
$ alkor-node volumes
```

Узел PhysDev VG Имя Размер Экземпляра

```
node1 /dev/sdb1 xenvg e61fbc97-....disk0 512M instance17
node1 /dev/sdb1 xenvg ebd1a7d1-....disk0 512M instance19
node2 /dev/sdb1 xenvg 0af08a3d-....disk0 512M instance20
node2 /dev/sdb1 xenvg cc012285-....disk0 512M instance16
node2 /dev/sdb1 xenvg f0fac192-....disk0 512M instance18
```

Приведенная выше команда сопоставляет каждый логический том с группой томов и базовым физическим томом и (возможно) с экземпляром.

Обобщенная обработка хранилища

Новое в версии 2.1.

В Звезда Алькор реализована структура хранения, которая пытается абстрагироваться от обработки типа хранилища, используемого кластером.

Во-первых, это список серверных хранилищ и их ситуация с пространством:

```
$alkor-node list-storage
```

Node	Type	Name	Size	Used	Free	Allocatable
node1	lvm-vg	xenvg	3.6T	0M	3.6T	Y
node2	lvm-vg	xenvg	3.6T	0M	3.6T	Y
node3	lvm-vg	xenvg	3.6T	2.0G	3.6T	Y

По умолчанию перечисляются физические тома LVM. Также можно перечислить группы томов LVM:

```
$ alkor-node list-storage -t lvm-vg
```

Тип узла	Имя	Размер	Используемый	Свободный	Распределяемый	
узел 1	lvm-vg	xenvg	3.6T	0M	3.6T	Y
node2	lvm-vg	xenvg	3.6T	0M	3.6T	Да
node3	lvm-vg	xenvg	3.6T	2.0G	3.6T	Y

Далее следует восстановление единиц хранения, которое в настоящее время реализовано только для групп томов и эквивалентно `vgreduce --removemissing:`

```
$ alkor-node repair-storage node2 lvm-vg xenvg
```

Воскресенье, 25 октября, 22:21:45 2009 Восстановление устройства хранения "xenvg" на узле 2 ...

И наконец, изменение свойств тома, которое (опять же) реализовано только для физических томов LVM и позволяет переключать выделяемое значение:

```
$ alkor-node mod-storage --allocatable=no node2 lvm-pv /dev/sdb1
```

Использование команд хранилища

Все эти команды нужны при восстановлении узла после сбоя диска:

- во-первых, нам нужно восстановиться после полного сбоя LVM (из-за отсутствия диска), выполнив команду `repair-storage`
- во-вторых, нам нужно изменить распределение на любом частично сломанном диске (т.е. LVM все еще видит его, но у него есть плохие блоки), запустив `mod_storage`
- затем мы можем эвакуировать экземпляры по мере необходимости

Кластерные операции

Помимо команды инициализации кластера (которая подробно описана в учебном документе по установке Звезда Алькор) и основной команды аварийного переключения, которая описана в разделе «Управление узлами», есть несколько других доступных кластерных операций.

Стандартные операции

Одна из немногих команд, которую можно запустить на любом узле (не только на ведущем), — это команда `getmaster`:

```
# на узле2
```

```
$ alkor-cluster getmaster
```

```
node1.example.com
```

Можно запросить и изменить глобальные параметры кластера с помощью команд `info` и `mod`:

```
$ alkor-cluster info
Cluster name: cluster.example.com
Cluster UUID: 07805e6f-f0af-4310-95f1-572862ee939c
Creation time: 2009-09-25 05:04:15
Modification time: 2009-10-18 22:11:47
Master node: node1.example.com
Architecture (this node): 64bit (x86_64)
...
Tags: foo
Default hypervisor: xen-pvm
Enabled hypervisors: xen-pvm
Hypervisor parameters:
- xen-pvm:
  root_path: /dev/sda1
...
Cluster parameters:
- candidate pool size: 10
```

```
...
Default instance parameters:
- default:
  memory: 128
```

```
...
Default nic parameters:
- default:
  link: xen-br0
```

Там различные параметры выше могут быть изменены с помощью команд модификации следующим образом:

- параметры гипервизора можно изменить с помощью `modify -H xen-pvm:root_path=...` и т. д. для других гипервизоров/ключей/значений
- «параметры экземпляра по умолчанию» можно изменить с помощью синтаксиса `Modify -B параметр=значение...`
- параметры кластера можно изменить с помощью отдельных опций команды модификации (например, `--candidate-pool-size` и т. д.)

Подробный список опций смотрите на справочной странице `alkor-cluster(8)`.

Версию кластера можно получить с помощью команды `version`:

```
$ alkor-cluster version
Software version: 2.1.0
Internode protocol: 20
Configuration format: 2010000
OS api version: 15
Export interface: 0
```

Это не очень полезно при отладке Звезда Алькор.

Команды глобального узла

Предусмотрены две команды для репликации файлов на все узлы кластера и для запуска команд на всех узлах:

```
$ alkor-cluster copyfile /path/to/file
$ alkor-cluster command ls -l /path/to/file
```

Это простые оболочки над `scp/ssh`, а более расширенное использование можно получить с помощью `dsh(1)` и подобных команд. Но они полезны, например, для обновления сценария ОС с главного узла.

Проверка кластера

Есть три команды, которые относятся к глобальным проверкам кластера. Первый — это проверка, которая дает обзор состояния кластера, выделяя любые проблемы. При нормальной работе эта команда не должна возвращать сообщения об ОШИБКЕ:

\$ alkor-cluster verify

```

Sun Oct 25 23:08:58 2009 * Verifying global settings
Sun Oct 25 23:08:58 2009 * Gathering data (2 nodes)
Sun Oct 25 23:09:00 2009 * Verifying node status
Sun Oct 25 23:09:00 2009 * Verifying instance status
Sun Oct 25 23:09:00 2009 * Verifying orphan volumes
Sun Oct 25 23:09:00 2009 * Verifying remaining instances
Sun Oct 25 23:09:00 2009 * Verifying N+1 Memory redundancy
Sun Oct 25 23:09:00 2009 * Other Notes
Sun Oct 25 23:09:00 2009 - NOTICE: 5 non-redundant instance(s) found.
Sun Oct 25 23:09:00 2009 * Hooks Results

```

Вторая команда — `verify-disks`, которая проверяет, что диски экземпляра имеют правильный статус на основе желаемого состояния экземпляра (работает/не работает):

```
$ alkor-cluster verify-disks
```

Обратите внимание, что эта команда не будет отображать вывод, если диски исправны.

Последняя команда используется для исправления любых расхождений в записанном Звезда Алькор размере диска и фактическом размере диска (информация о размере диска необходима для правильной активации и роста дисков на основе DRBD):

\$ alkor-cluster repair-disk-sizes

```

Sun Oct 25 23:13:16 2009 - INFO: Disk 0 of instance instance1 has mismatched
size, correcting: recorded 512, actual 2048
Sun Oct 25 23:13:17 2009 - WARNING: Invalid result from node node4, ignoring
node results

```

Выше показан один экземпляр с неправильным размером диска и узел, который вернул неверные данные, поэтому мы проигнорировали все первичные экземпляры этого узла.

Перераспределение конфигурации

Если команда `verify` жалуется на несоответствие файлов между главным и другими узлами из-за проблем с некоторыми узлами или если вы вручную изменили файлы конфигурации, вы можете принудительно отправить основную конфигурацию на все остальные узлы с помощью команды `redist-conf`:

```
$ alkor-cluster redist-conf
```

Эта команда будет молчать, если нет проблем с отправкой обновлений на другие узлы.

Переименование кластера

Можно переименовать кластер или изменить его IP-адрес с помощью команды `rename`. Если изменился только IP-адрес, вам нужно передать текущее имя, и Звезда Алькор поймет, что его IP-адрес изменился:

```
$ alkor-cluster rename cluster.example.com
This will rename the cluster to 'cluster.example.com'. If
you are connected over the network to the cluster name, the operation
is very dangerous as the IP address will be removed from the node and
the change may not go through. Continue?
y/[n]/?: y
Failure: prerequisites not met for this operation:
Neither the name nor the IP address of the cluster has changed
```

В приведенном выше выводе ни одно из значений не изменилось с момента инициализации кластера, поэтому операция не завершена.

Очередь операций

Выполнение очереди заданий в Звезда Алькор 2.0 и выше можно проверить, приостановить и возобновить с помощью команды очереди:

```
$ alkor-cluster queue info
The drain flag is unset
$ alkor-cluster queue drain
$ alkor-instance stop instance1
Failed to submit job for instance1: Job queue is drained, refusing job
$ alkor-cluster queue info
The drain flag is set
$ alkor-cluster queue undrain
```

Это наиболее полезно, если у вас есть активный кластер и вам нужно обновить программное обеспечение Звезда Алькор или просто перезапустить программное обеспечение на любом узле:

1. приостановить очередь через утечку очереди
2. подождите, пока больше не будет запущенных заданий через `alkor-job list`
3. перезапустите мастер или другой узел или обновите программное обеспечение
4. возобновить очередь через очередь `undrain`

Обратите внимание, что эта команда сохраняет только локальный файл флагов, и если вы переключитесь на мастер, это не повлияет на новый мастер.

Управление наблюдателем

Звезда Алькор-watcher(8) — это программа, обычно запускаемая через `cron`, которая заботится об операциях обслуживания кластера (перезапуск отключенных экземпляров, активация отключенных DRBD-дисков и т. д.). Однако при обслуживании и устранении неполадок это может помешать; отключение его с помощью комментирования задания `cron` не так уж хорошо, так как об этом можно забыть. Таким образом, есть несколько команд для автоматического управления наблюдателем: пауза, информация и продолжение:

```
$ alkor-cluster watcher info
The watcher is not paused.
$ alkor-cluster watcher pause 1h
The watcher is paused until Mon Oct 26 00:30:37 2009.
$ alkor-cluster watcher info
The watcher is paused until Mon Oct 26 00:30:37 2009.
```

```

$ alkor-watcher -d
2009-10-25 23:30:47,984: pid=28867 alkor-watcher:486 DEBUG Pause has been
set, exiting
$ alkor-cluster watcher continue
The watcher is no longer paused.
$ alkor-watcher -d
2009-10-25 23:31:04,789: pid=28976 alkor-watcher:345 DEBUG Archived 0 jobs,
left 0
2009-10-25 23:31:05,884: pid=28976 alkor-watcher:280 DEBUG Got data from
cluster, writing instance status file
2009-10-25 23:31:06,061: pid=28976 alkor-watcher:150 DEBUG Data didn't
change, just touching status file
$ alkor-cluster watcher info
The watcher is not paused.

```

Обратите внимание, что эта команда сохраняет только локальный файл флагов, и если вы переключитесь на мастер, это не повлияет на новый мастер.

Автоматическое обслуживание узла

Если параметр кластера `serve_node_health` включен (см. man-страницу для `alkor-cluster`, подкоманды `init` и `modify`), то автоматически произойдет следующее:

- наблюдатель выключит все экземпляры, работающие на автономных узлах
- наблюдатель деактивирует любые устройства DRBD на автономных узлах

В будущем планируется больше действий, поэтому включайте этот параметр только в том случае, если узлы полностью выделены для Ганети; в противном случае возможна потеря данных из-за действий автоматического обслуживания.

Полное удаление кластера

Обычный метод очистки кластера — запустить `alkor-cluster destroy`, однако, если установка Звезда Алькор каким-либо образом нарушена, она не запустится.

В таком случае можно удалить вручную большинство, если не все следы установки кластера, выполнив следующие действия на всех узлах:

1. Выключите все экземпляры. Это зависит от используемого метода виртуализации (Xen, KVM и т. д.):
 1. Xen: запустите `xm list` и `xm destroy` на всех экземплярах, отличных от домена 0.
 2. KVM: убить все процессы KVM
 3. chroot: убить все процессы под точками монтирования chroot
2. При использовании DRBD отключите все второстепенные DRBD (которые в настоящее время больше не должны использоваться экземплярами); на каждом узле запустите `drbdsetup /dev/drbdN down` для каждого активного второстепенного DRBD.
3. Если вы используете LVM, очистите группу томов Звезда Алькор; если только Звезда Алькор создал логические тома (и вы, например, не делитесь группой томов с ОС), то просто запустив `lvremove -f хенvg` (замените «хенvg» на имя вашей группы томов), вы должны выполнить необходимую очистку.

4. При использовании файлового хранилища рекурсивно удалите все файлы и каталоги в вашем каталоге хранилища файлов: `rm -rf /srv/Звезда Алькор/file-storage/*` заменив путь на правильный путь для вашего кластера.
5. Остановите демонов Звезда Алькор (`/etc/init.d/Звезда Алькор stop`) и убейте всех оставшихся в живых (`pgrep Звезда Алькор` и `rkill Звезда Алькор`).
6. Удалите каталог состояния Звезда Алькор (`rm -rf /var/lib/Звезда Алькор/*`), заменив путь правильным путем для вашей установки.
7. При использовании RBD запустите `rbdm unmap /dev/rbdN`, чтобы отменить сопоставление дисков RBD. Затем удалите образы дисков RBD, используемые Звезда Алькор, идентифицированные их UUID (`rbdm rm uuid.rbd.diskN`).

На главном узле удалите кластер из `master-netdev` (обычно это `xen-br0` для режима моста, иначе `eth0` или аналогичный), запустив `ip a del $clusterip/32 dev xen-br0` (используйте правильный IP-адрес кластера и сетевое устройство). имя).

На этом этапе машины готовы к созданию кластера; если вы хотите полностью удалить Звезда Алькор, вам также необходимо отменить некоторые изменения SSH и каталоги журналов:

- `rm -rf /var/log/alkor/srv/alkor` (замените правильными путями)
- удалить из `/root/.ssh` ключи, добавленные Звезда Алькор (проверьте файлы `author_keys` и `id_dsa`)
- повторно сгенерировать SSH-ключи хоста (проверьте сценарии запуска OpenSSH)
- удалить ПО

В противном случае, если вы планируете воссоздать кластер, вы можете просто перезапустить `alkor-cluster init`.

Замена ключей SSH и SSL

Звезда Алькор использует ключи SSL и SSH и активно изменяет ключи SSH на узлах. В результате, чтобы заменить эти ключи, необходимо выполнить несколько дополнительных шагов: Замена ключей кластера.