ООО «Промобит»

644024 г.Омск, пр-т. К. Маркса, д. 3 корп. 1
Тел.: +7 (3812) 36-11-11
www.bitblaze.ru
info@.bitblaze.ru

Bitblaze KFS

РУКОВОДСТВО АДМИНИСТРАТОРА

Омск
2019

Содержание

1. Требования к квалификации системного администратора

Для установки и обслуживания распределенной файловой системы Bitblaze KFS системный администратор должен обладать опытом работы с операционными системами семейства Linux, опытом работы с современным аппаратным обеспечением, знанием сетевого стека протоколов TCP/IP, умением читать и понимать скрипты автоматизации задач (bash), опытом работы со средами виртуализации Proxmox VE, VirualBox, Vmware.

2. Первоначальное конфигурирование системы

Прежде всего, скажем о некоторых важных правилах:

- Главный сервер метаданных (master-server) лучше всего работает на выделенном компьютере, предпочтительно оснащенном SSD-диском.
- Сервер хранения блоков данных (chunk-server) лучше всего работает при наличии как минимум одного выделенного диска.
- Не устанавливайте сервер журналирования метаданных (metalogger) на той же машине, что и главный сервер, это не сделает ваши метаданные более безопасными. С другой стороны, сервера журналирования метаданных не являются обязательными, и вполне возможно установить сервер журналирования метаданных вместе с сервером хранения блоков данных.
- Чтобы повысить безопасность своих данных, вам необходимо использовать резервные («теневые») главные серверы метаданных (shadow masters servers).

На каждом сервере необходимо выполнить некоторые основные настройки, прежде чем мы перейдем к настройке конфигурации Bitblaze KFS.

- настроить параметры сети
- настроить параметры ядра системы

3. Основные настройки операционной системы для Bitblaze KFS

Файл / etc / hosts

Поскольку Bitblaze KFS - это сетевая файловая система, настройка сети очень важна. Особенно важно правильно настроить службу разрешения имен, которая помогает компьютерам в сети находить друг друга. Существует два основных способа, с помощью которых компьютеры находят друг друга по имени: статические записи в файле /etc/hosts или система DNS.

Для первоначальной настройки мы настоятельно рекомендуем использовать файл /etc/hosts для разрешения имен между компонентами кластера Bitblaze KFS. Этот способ, во-первых, проще, чем использование системы DNS, а во-вторых, он предотвратит потерю данных кластером в случае, если кто-то изменит настройки в

службе DNS. Это требует дополнительной работы от администратора кластера Bitblaze KFS, чтобы поддерживать файлы /etc/hosts в актуальном состоянии, но защитит вас от сбоя хранилища и потери данных в случае, аварии в системе DNS.

Структура файла /etc/hosts очень проста:

```
<ip-address>  <hostname>  <alias>
```

В реальной установке Bitblaze KFS один из таких файлов может выглядеть так:

```
127.0.0.1          localhost
192.168.84.137     kfsmaster
192.168.84.138     shadowmaster
192.168.84.20      chunkserver1  metalogger
192.168.84.21      chunkserver2
192.168.84.22      chunkserver3
```

Файловые системы для главного сервера метаданных Bitblaze KFS

В силу скорости и стабильности файловых систем XFS и ZFS мы рекомендуем использовать именно их на серверах Bitblaze KFS, предназначенных для промышленной эксплуатации.

Главный сервер метаданных хранит свои записи в памяти, но при этом часто сохраняет их на диске, и поэтому в данном случае диск должен быть очень быстрым, однако его объем не столь важен. Обычного 250 Gb SSD диска вполне достаточно. Использование файловой системы XFS в силу ее быстроты здесь предпочтительнее. Использование аппаратного контроллера RAID здесь, напротив, не рекомендуется.

4. Настройка главного сервера метаданных Bitblaze KFS

Главный сервер метаданных исключительно важен для работы всего Bitblaze KFS. Он хранит всю мета-информацию о каждом файле, каждом фрагменте данных и каждом срезе (slice) в режиме EC. В любой момент времени он знает, что хранится в кластере Bitblaze KFS и как его найти. Кроме того, он отвечает за устранение последствий выхода из строя дисков и серверов блоков данных в целом.

База данных метаданных

Чтобы главный сервер метаданных работал правильно, нужно сначала снабдить его именем файла, в котором он будет хранить свою базу данных метаданных. Имя такого файла по умолчанию (и которое можно изменить в файле kfsmaster.cfg) таково:

```
/var/lib/kfs/metadata.kfs
```

Файл kfsmaster.cfg

Файл kfsmaster.cfg содержит много настроек для полного использования возможностей системы Bitblaze KFS, которые будут рассмотрены в отдельном руководстве по его настройке. Для базовой настройки важны следующие вещи: текущая «индивидуальность» (роль) данного экземпляра сервера метаданных. Для начала работы с Bitblaze KFS допустимыми значениями являются master и shadow.

```
PERSONALITY = master
```

означает, что этот экземпляр сервера метаданных выступает в качестве основного сервера метаданных, управляющего всеми изменениями метаданных файловой системы.

```
PERSONALITY = shadow
```

означает, что этот экземпляр сервера метаданных выступает в качестве резервного сервера метаданных, готового к немедленной работе в качестве основного сервера метаданных в случае сбоя текущего основного сервера.

Роль сервера метаданных может быть изменена в любой момент, если сервер меняет роль с **shadow** на **master**. Изменение роли в другом направлении запрещено.

Параметр, определяющий место хранения метаданных и файлов блокировок:

```
## Location for metadata files.
## (Default: /var/lib/kfs)

DATA_PATH = /var/lib/kfs
```

Остальные многочисленные настройки параметров в файле kfsmaster.cfg будут рассмотрены отдельно.

5. Конфигурирование резервного сервера метаданных (shadow master)

Резервный сервер метаданных (shadow master) настроен почти так же, как и главный сервер метаданных (master). Поскольку предполагается, что он берет на себя функциональные возможности главного сервера метаданных в случае его сбоя, он должен синхронизировать свою базу метаданных и, кроме того, иметь все настройки, аналогичные настройкам главного сервера метаданных.

Настройки, характерные для резервного сервера метаданных:

В файле kfsmaster.cfg установите **роль** сервера метаданных:

```
# Роль сервера метаданных
PERSONALITY = shadow
```

А также адрес главного сервера метаданных:


```
# Адрес хоста на котором работает главный сервер метаданных (master)
# Обязателен для настройки резервного сервера метаданных (shadow master)
MASTER_HOST = 192.168.84.137
```

Файлы kfsexports.cfg, kfsgoals.cfg и kfstopology.cfg резервного сервера метаданных должны быть синхронизированы (то есть быть одинаковыми) с главным сервером метаданных.

6. Настройка серверов фрагментов данных

Сервера фрагментов данных просты в настройке. Обычно, если в ваших файлах **/etc/hosts** адрес главного сервера метаданных настроен правильно, и вам не требуется маркировка сервера фрагментов данных, то файл kfschunkserver.cfg может остаться без изменений. Если вам требуется фиксировано настроить связь с главным сервером метаданных по конкретному адресу, измените соответствующим образом следующую строку:

```
MASTER_HOST = 192.168.84.137
```

чтобы зафиксировать связь с главным сервером метаданных по фиксированному адресу 192.168.84.137.

Далее нужно указать, где именно сервер хранения данных будет хранить фактические данные. Это делается в файле kfshdd.cfg. Вы указываете директории с их полным путем, по одному в каждой строке этого файла конфигурации.

```
# использовать каталог  '/ mnt/hd1' с параметрами по умолчанию
/mnt/hd1

# использовать каталог '/mnt/hd2' с целью реплицировать все данные из него
* /mnt/hd2
```

В настройках всегда предполагается, что такая директория является выделенным устройством, то есть это отдельный HDD, RAIDset или SSD, и именно на этом предположении основывается расчет пространства.

После такой настройки сервер хранения фрагментов данных готов к работе в составе системы Bitblaze KFS.

Чтобы удалить директорию из списка используемых системой Bitblaze KFS директорий, просто добавьте символ * («звездочка») в начало соответствующей строки в файле kfshdd.cfg, например:

```
# использовать каталог '/mnt/hd2' с целью реплицировать все данные из него
* /mnt/hd2
```


Система Bitblaze KFS начнет реплицировать все данные из него в другое место. Как только вы увидите (например, в веб-интерфейсе), что все данные были благополучно скопированы, вы можете обновить файл конфигурации, удалив эту строку, а затем удалить связанное с ним физическое устройство из вашего сервера хранения фрагментов данных.

7. Настройка сервера журналирования метаданных

Сервер журналирования метаданных (metalogger) используется для аварийного восстановления метаданных в случае сбоя главного и теневого серверов метаданных. База метаданных может быть построена заново на основе информации в таких журналах. Настройка сервера журналирования метаданных проста. По сути, вам не нужно ничего настраивать, если ваш файл **/etc/hosts** правильно настроен, в противном случае вам нужно прямо установить следующие параметры в файле kfsmetalogger.cfg:

```
# адрес главного хоста Bitblaze KFS для подключения (по умолчанию kfsmaster)
MASTER_HOST=192.168.84.137
```

а также

```
# номер главного порта Bitblaze KFS для подключения (по умолчанию 9419)
MASTER_PORT=9419
```

и сервер журналирования метаданных (metalogger) готов к работе.

8. Соединение клиентов с распределенной файловой системой Bitblaze KFS

Чтобы клиент мог воспользоваться возможностями сетевой распределенной файловой системой Bitblaze KFS, необходимо установить соответствующее программное обеспечение на клиентский компьютер. В случае ALT Linux клиента, необходимо установить пакет **kfs-client-1.0.0-alt2.x86_64.rpm**, загруженный с сайта компании ООО «Промобит». Если вся система Bitblaze KFS устанавливается в ознакомительных и демонстрационных целях на одном компьютере, реальном или виртуальном, то клиентское программное обеспечение устанавливается в числе других программных компонентов устанавливаемой системы Bitblaze KFS. В рамках данного краткого руководства будем считать, что клиентское ПО уже установлено.

Далее следует создать так называемую точку монтирования

```
$ mkdir /mnt/kfs_test
```

Далее монтируем сетевую распределенную файловую систему Bitblaze KFS с опциями "по умолчанию", указанными в файле kfsmount.cfg. Этот файл создается в директории /etc/kfs при установке программного обеспечения Bitblaze KFS с

параметрами "по умолчанию", которые вы впоследствии можете настроить в соответствии с вашими требованиями.

```
[root@192 kfs_setup]# kfsmount  /mnt/kfs_test
```

Базовый набор серверов системы Bitblaze KFS в этот момент уже должен быть запущен. Процедура запуска серверов Bitblaze KFS описана в разделе «Запуск и остановка системы Bitblaze KFS» в документе «Инструкция по установке системы Bitblaze KFS в операционной системе ALT Linux». Следуйте описанной там процедуре запуска серверов, и **убедитесь в её успешности, перед тем как переходить к монтированию директории доступа** к Bitblaze KFS.

Проверить результаты работы утилиты kfsmount можно следующим образом:

```
[root@192 kfs_setup]#  mount
```

Вывод этой команды может быть, например, таким:

```
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
tracefs on /sys/kernel/debug/tracing type tracefs (rw,relatime)
/srv/kfs_chunk1 on /mnt/kfs_chunk1 type ext4 (rw,relatime,data=ordered)
/srv/kfs_chunk2 on /mnt/kfs_chunk2 type ext4 (rw,relatime,data=ordered)
/srv/kfs_chunk3 on /mnt/kfs_chunk3 type ext4 (rw,relatime,data=ordered)
/srv/kfs_chunk4 on /mnt/kfs_chunk4 type ext4 (rw,relatime,data=ordered)
kfsmaster:9421 on /mnt/kfs_test type fuse.mfs
        (rw,nosuid,nodev,relatime,user_id=0,group_id=0,default_permissions,allow_other)
```

Здесь подчеркнуто сообщение утилиты mount о том, что точка монтирования /mnt/kfs_test соединена с сервисом kfsmaster по порту 9421. Наличие такого сообщения в выводе утилиты mount говорит об успешности процесса монтирования Bitblaze KFS.

На этом процесс соединения клиента с сетевой распределенной файловой системой Bitblaze KFS в его простейшем варианте можно считать выполненным.

9. **Проверка общей** работоспособности установленного клиентского соединения с распределенной **файловой системой Bitblaze KFS**

Самым простым и доступным способом проверки общей работоспособности вновь установленной системы Bitblaze KFS может быть выполнение обычных операций записи-чтения файлов в директорию монтирования, например, в /mnt/kfs_test, естественно, предварительно смонтированную утилитой kfsmount. Перед этим обязательно проверьте успешность такого монтирования командой

```
[root@192 kfs_setup]#  mount
```


как описано в разделе «Соединение клиентов с распределенной файловой системой Bitblaze KFS» данного документа.

Более полная процедура проверки работоспособности установленной системы Bitblaze KFS с помощью программы **fio**. описана в разделе «Проверка общей работоспособности системы Bitblaze KFS»

10. Custom Goals

Каждый из серверов метаданных (kfs-master, kfs-shadow) имеет массив из 40 (сорок) записей, содержащий т.н. «цели» (goals).

Цель – это некая настройка (свойство) для (каждого, любого) файла или директории, монтированной с помощью kfsmount. Цель указывает, как именно будет храниться конкретный файл или все содержащиеся в директории файлы. В совокупности файлы и директории, хранящиеся в KFS, будем называть «объекты KFS». Цель, указанная для директории, распространяются на все содержащиеся в ней объекты KFS (файлы, вложенные директории).

Каждый элемент массива целей имеет идентификатор (id), по которому этот массив индексируется.

Такой массив (отдельный, индивидуальный(!)), на каждом сервере метаданных, как на master, так и на shadow server, выглядит по-умолчанию вот так:

ID	Name	Goal (формат и образец записи в файле goals.cfg
1	1	1: $std _
2	2	2: $std {_ _}
3	3	3: $std {_ _ _}
4	4	4: $std {_ _ _ _}
5	5	5: $std {_ _ _ _ _}
6	6	6: $std {_ _ _ _ _}
7	7	7: $std {_ _ _ _ _}
8	8	8: $std {_ _ _ _ _}
9	9	9: $std {_ _ _ _ _}
10	10	10: $std {_ _ _ _ _}
11	11	11: $std {_ _ _ _ _}
. . .	. .	. . : $std {_ _ _ _ _}
39	39	39: $std {_ _ _ _ _}
40	40	40: $std {_ _ _ _ _}

То, что эта подсистема структурно реализована в виде отдельных массивов (и отдельных файлов) на разных masters (на разных машинах), диктует необходимость поддерживать синхронность этих файлов, т.е строго одинаковое их содержимое. Это особенно важно в случае HA кластера, в котором переключение серверов может произойти в любой момент времени.

Символы «_» - это метки серверов метаданных. Они указываются в файлах конфигурации **kfschunkserver.cfg** в виде строки вида: LABEL = _

Т.е. по умолчанию метки всех chunk серверов имеют значение «_». Это задает некую «одинаковость» этих серверов с точки зрения балансировки распределения чанков между ними. Параметр CHUNKS_REBALANCING_BETWEEN_LABELS = 0 в файле **kfsmaster.cfg** определяет возможность балансировки нагрузки (перемещения фрагментом/блоков данных) между серверами с различными метками (значение = 1). Значение 0 разрешает перемещение блоков данных только между серверами с одинаковыми метками.

Существует возможность работы с goals посредством kfs-admin. Показаны различные форматы указания IP мастера. Операция выполнится только если адресоваться к master server. Shadow master на нее не ответит. Т.е. вывести можно только актуальный список целей с активного в данный момент master-server.

Вывести список целей для актуального мастера

```
# kfs-admin list-goals  localhost  9421
# kfs-admin list-goals  192.168.70.5  9421
# kfs-admin list-goals  kfsmaster  9421
Утилита kfs (ставится пакетом client) производит следующие манипуляции:
# kfs  getgoal  [-r] [-n|-h|-H]  'OBJECT'...
# kfs  setgoal  [-r] [-n|-h|-H]  'OBJECT'...
```

OBJECT – это файл, директория или «удаленный файл» (trash file).

Примеры (выполняем на клиенте!):

```
# kfs getgoal /mnt/kfs_storage        // запрашиваем goal на нашу директорию монтирования
/mnt/kfs_storage: 1
# kfs getgoal /mnt/kfs_storage/client10G_test1/fio.1      // запрашиваем goal на один файл
# kfs getgoal -r /mnt/kfs_storage/client10G_test1  // на директорию, рекурсивно
```

Установим цель 2 (сохранять в 2-х блоках данных (chunks)) для директории client10G_test1

```
# kfs setgoal 2 /mnt/kfs_storage/client10G_test1
```

Посмотрим что получилось:

```
# kfs getgoal -r /mnt/kfs_storage/client10G_test1
/mnt/kfs_storage/client10G_test1:
 files with goal            2 :         1
```



```
directories with goal   1 :         64
```

Т.е. директория имеет цель 2, расположенные в ней файлы имеют цель 1. Но новые файлы в этой директории будут уже иметь цель 2.

11. Работа с «корзиной» (Trash)

KFS позволяет смонтировать специальную точку доступа к метаданным, в частности, к удаленным файлам. Это делается с помощью команды kfsmount (kfsmount3) на компьютере-клиенте, например:

```
# kfsmount /mnt/kfs-meta -o mfsmeta
```

Предварительно в локальной файловой системе «клиента» должна быть создана директория /mnt/kfs-meta, либо с другим именем, и тогда команду монтирования придется соответственно изменить.

В результате успешного монтирования в директории /mnt/kfs-meta появятся директории

```
/mnt/kfs-meta
|- reserved
|- trash
   |- undel
```

Директория **trash** – это хранилище (временно) удаленных файлов, которые еще можно восстановить. Любой файл, атрибут “trashtime” которого больше 0 (нуля), может быть восстановлен. Для этого достаточно перенести его из директории **trash** в директорию **undel**. (Примечение: при таком «перемещении» файл появится не в директории **undel**, а уже в своей исходной директории.

Для окончательного удаления файла, напротив, достаточно удалить его из директории **trash**. Имена файлов представлены композитной строкой, содержащей inode файла, список его директорий (path) и собственно первоначальным именем файла. Это обеспечивает уникальность имен файлов в одной служебной директории.

Директория **reserved** предназначена для временного хранения файлов, удаленных одним пользователем, но дескриптор которых еще используется другим(и) пользователем. Такие файлы помещаются в эту директорию до момента освобождения дескриптора.

12. Установка времени хранения удаленных объектов

По умолчанию время хранения удаленных объектов trashtime в trash составляет 24 часа (86400 секунд). Если мы установим значение trashtime 0, то такие объекты (файлы и директории) будут удаляться немедленно. Если нужно установить время

хранения, например, равное 2 минутам, то на клиенте необходимо выполнить следующую команду:

```
# kfs settrashtime 120 -r /mnt/kfs_storage/a_tests
```

Ключ –r указывает на рекурсивное изменение атрибута trashtime применительно к вложенным файлам и директориям.

Запросить установленное время хранения можно командой

```
# kfs gettrashtime  -r  /mnt/kfs_storage/client10G_test1
/mnt/kfs_storage/client10G_test1:
  files with trashtime                60 :          65
  directories with trashtime          60 :           1
```

13. Как запустить cgi-server

WEB сервер kfs-cgiserver, обслживающий запросы WEB-интерфейса ПО KFS, необходимо (и логично) установить на одном или нескольких серверах типа kfs-master. Тогда в случае необходимости можно будет легко запустить другой экземпляр kfs-cgiserver для непрерывного мониторинга работы кластера KFS.

Запуск kfs-cgiserver производитс следующей командой (на машине, где он установлен):

```
# /usr/sbin/kfs-cgiserver -H 192.168.84.55 -P 9425  -v
```

Ключ –H позволяет указать адрес хоста, где расположен активный kfs-master.

Ключ –P позволяет указать порт этого хоста.

Ключ –v (verbose) бывает полезным при отладке кластера, позволяя видеть в ограниченных пределах обмен kfs-cgiserver и его WEB-клиента.

Для доступа к серверу со стороны клиента, наберите в любом современном браузере в его адресной строке команду, например, http://192.168.84.55:9425/kfs.cgi

14. Управление процессом старта/останова системы хранения данных KFS

Для старта системы хранения данных KFS в конфигурации с одним узлом управления, то есть с одним сервером метаданных (master server), используется команда kfsmaster:

1. Старт сервера метаданных:

```
# kfsmaster  start
```

2. Проверка статуса сервера метаданных:


```
# kfsmaster  test
```

3. Останов сервера метаданных:

```
# kfsmaster  stop
```

Для старта и управления системой хранения KFS в составе отказоустойчивого кластера (HA-режим, High Availablity Mode) используется команда, а точнее скрипт, написанный на языке оболочки BASH: kfsmaster-ha (kfsmaster-ha.sh в одной из первых версий), управляемый следующими параметрами:

1. Старт кластера:

```
# kfsmaster-ha  start
```

2. Проверка статуса системы или вывод списка запущенных модулей (процессов):

```
# kfsmaster-ha  status
# kfsmaster-ha  test
```

**для вывода более детальной информации о процессах:

```
# kfsmaster-ha  allinfo
```

3. Останов кластера:

```
# kfsmaster-ha  stop
```

Примечание:

В общем случае, старт сервера мастера посредством **kfsmaster-ha** сопровождается запуском дополнительного сервиса мониторинга кластера — **kfs-uraft** в режиме демона, и авто-выбором роли сервера управления метаданными: ведущий (master) или ведомый (shadow). Именно демон **kfs-uraft** следит за текущим состоянием и управляет ролью основного сервера кластера. В случае выхода его из строя или отказа, корректность переключения ролей серверов метаданных кластера будет нарушена, в связи с чем обычно происходит его автоматический перезапуск даже в случае останова мастер-роли.

Кроме этого, в случае наличия в кластере всего двух серверов метаданных с ролью ведущий (master) или ведомый (shadow), для корректного функционирования алгоритма переключения роли мастера, в случае отказа одного из них, требуется наличие запущенного сервиса мониторинга кластера (kfs-uraft) на еще одном из узлов кластера, в режиме URAFT_ELECTOR_MODE = 1 (этот параметр указывается в конфигурационном файле сервиса kfs-uraft). Для этого может быть использован, например, один из серверов хранения блоков данных (chunks, kfschunkserver).

