

РУКОВОДСТВО  
ПО УСТАНОВКЕ И АДМИНИСТРИРОВАНИЮ  
ПО BITBLAZE KFS

**BITBLAZE**

Версия 1.5

2022

# Содержание

|   |           |
|---|-----------|
| <b>ВВЕДЕНИЕ</b>   | <b>3</b>  |
| Назначение документа  | 3         |
| Требования к квалификации специалиста, выполняющего установку | 3         |
| Необходимые условия для установки ПО Bitblaze KFS             | 3         |
| <b>ЭЛЕМЕНТЫ АРХИТЕКТУРЫ</b>                                   | <b>4</b>  |
| <b>УСТАНОВКА СЕРВЕРНОЙ ЧАСТИ ПО BITBLAZE KFS</b>              | <b>5</b>  |
| Подготовка системы  | 5         |
| Установка из репозитория                                      | 5         |
| Установка из локальных установочных пакетов                   | 7         |
| Проверка установки  | 8         |
| <b>УСТАНОВКА КЛИЕНТСКОЙ ЧАСТИ ПО BITBLAZE KFS</b>             | <b>9</b>  |
| Подготовка системы  | 9         |
| Установка   | 10        |
| Проверка монтирования   | 11        |
| <b>УПРАВЛЕНИЕ ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ</b>                    | <b>12</b> |
| <b>ЕСЛИ НЕ ПОЛУЧАЕТСЯ</b>                                     | <b>23</b> |

# Введение

## Назначение документа

Документ «Руководство по установке и администрированию ПО Bitblaze KFS» предназначен для информирования технических специалистов о функциях и параметрах конфигурирования компонент распределенной файловой системы Bitblaze KFS.

Моноширинным шрифтом выделены фрагменты редактируемых файлов, команды и параметры команд: например, `heartbeat_period = <heartbeat_period>`. Угловыми скобками отмечены поля, в которых необходимо указать требуемые значения: например, `<ip>` — требуется указать конкретный ip-адрес.

## Требования к квалификации специалиста, выполняющего установку

Специалист, выполняющий установку и администрирование ПО Bitblaze KFS, должен:

- уметь устанавливать ОС ALT Linux (ОС «Эльбрус»),
- уметь устанавливать и удалять программы (пакеты) в Linux,
- знать базовые команды ОС Linux,
- знать основы настройки локальной сети в ОС Linux,
- уметь устанавливать компоненты ПО файловой системы KFS.

## Необходимые условия для установки ПО Bitblaze KFS

1. Должна быть предустановлена операционная система ALT Linux или «Эльбрус» с настроенным сетевым интерфейсом, известным IP-адресом, а также с наличием не менее 30 Гигабайт свободного места на системном диске.
2. Для установки из репозитория должен быть разрешен доступ к сети Интернет.
3. Необходимы реквизиты учетной записи суперпользователя данной машины (пользователь root).
4. Необходимы данные об учетной записи и репозитории, полученные от инженеров компании ООО «Промобит».

# ЭЛЕМЕНТЫ АРХИТЕКТУРЫ

ПО Bitblaze KFS — управляющее ПО, которое предназначено для развертывания на серверах хранения распределённой многоузловой масштабируемой отказоустойчивой кластерной системы хранения данных (СХД). Распределенная файловая система KFS позволяет использовать носители данных, расположенные на серверных узлах системы, как цельное пространство хранения с единой файловой иерархией. Компоненты ПО Bitblaze KFS функционируют на аппаратном обеспечении серверной и клиентской частей системы и взаимодействуют в локальной сети Ethernet.

ПО Bitblaze KFS включает программные компоненты, или модули, реализующие различные роли при управлении функционированием кластера. Другими словами, роли — модели поведения управляющих программных процессов. Для удобства в тексте данного Руководства теми же ролями будут обозначаться программные процессы в некластерных конфигурациях.

При развертывании ПО каждый компонент устанавливается в соответствии с его ролью. Можно выделить следующие основные модули:

kfs-master — модуль обработки метаданных, роль **fsmaster**;

kfs-chunkserver — модуль доступа к данным, роль **fschunk**;

kfs-mount — модуль клиентского доступа, роль **client**.

Основу серверной части системы составляют серверы метаданных и серверы хранения фрагментов данных. Первые управляют всей организацией файловой системы и называются главными. Различают активный сервер метаданных — master server, управляющий функционированием всей системы, и резервные, называемые shadow master. Серверы хранения фрагментов данных, называемые также chunk-серверами, содержат пользовательские данные.

В общем случае для построения отказоустойчивого кластера система должна включать не менее трех серверов метаданных, в которых роли активного и резервных серверов определяются автоматически. Любой из резервных серверов при необходимости может стать активным главным сервером метаданных.

Основные компоненты аппаратного обеспечения системы и соответствующие им программные модули ПО Bitblaze KFS представлены на рис. 1.

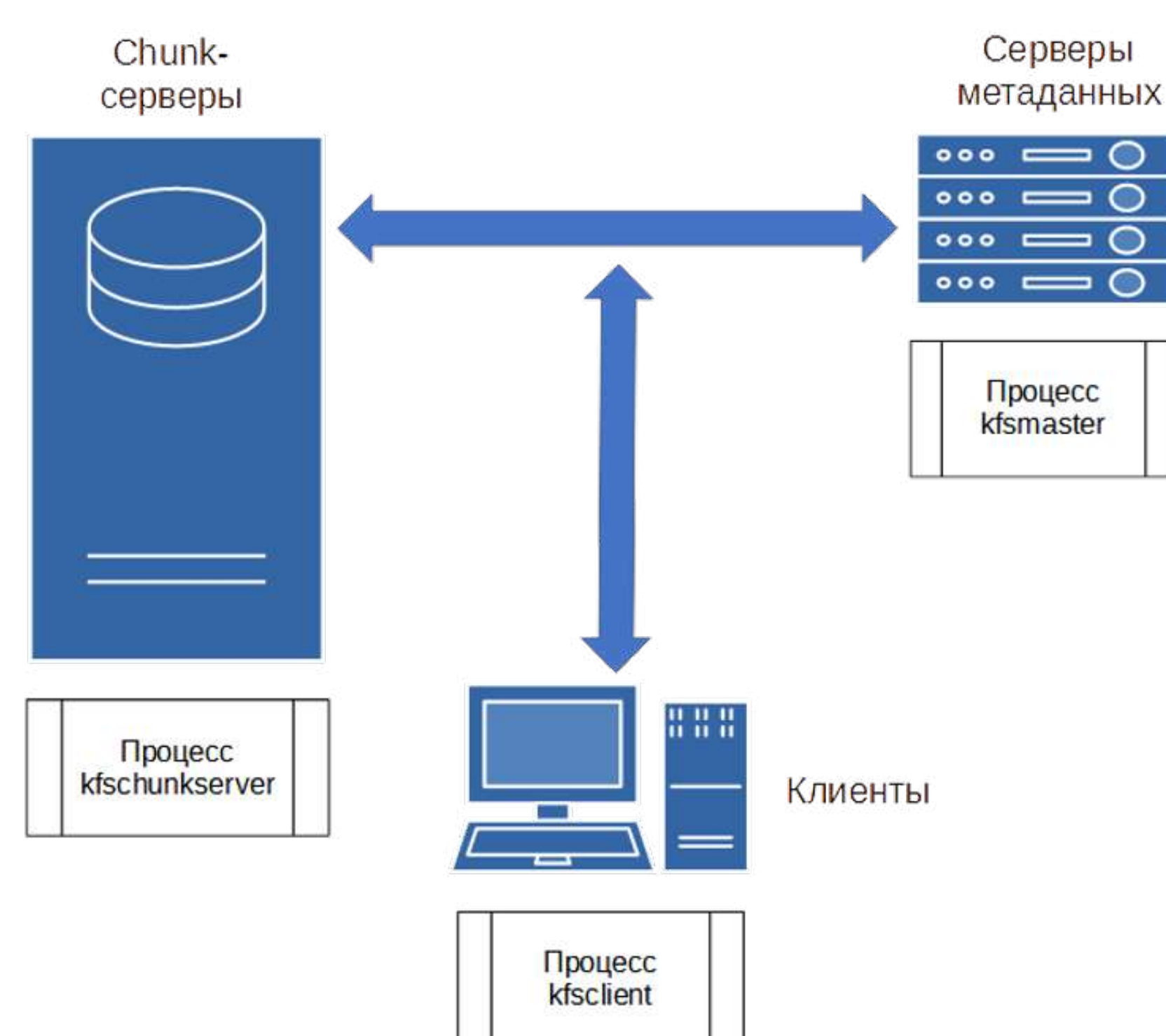


Рисунок 1 — Элементы архитектуры

# УСТАНОВКА СЕРВЕРНОЙ ЧАСТИ ПО VITBLAZE KFS

## Подготовка системы

1. Войти в систему под пользователем root.
2. Для обновления репозиториев выполнить команду

```
apt-get update
```

## Установка из репозитория

1. Создать файл /etc/apt/sources.list.d/promobit.list и добавить в него полученные от инженеров ООО «Промобит» репозитории. Пример строк с репозиториями (названия могут отличаться):

```
rpm http://<имя_пользователя:пароль>@repos.promobit.ru/kfs/p9/branch/ <arch> kfs  
rpm http://<имя_пользователя:пароль>@repos.promobit.ru/kfs/p9/branch/ noarch kfs
```

2. Выполнить обновление

```
apt-get update
```

3. Установить пакет promobit-kfs-deploy для развертывания серверных компонент ПО

```
apt-get install promobit-kfs-deploy
```

4. Перейти в каталог /opt/promobit/kfs-deploy/group\_vars/, сделать копию файла all.yml.example и назвать ее all.yml.
5. Отредактировать файл all.yml:

- 5.1.1 При установке из репозитория параметр pbit\_install\_local\_packages принимает значение no:

```
pbit_install_local_packages: no
```

- 5.1.2. При установке из репозитория отредактировать параметры подключения в соответствии с данными, полученными от инженеров компании ООО «Промобит».

- 5.1.3. Задать размер блока KFS (по умолчанию 65536):

```
pbit_kfs_block_size: <размер блока>
```

#### 5.1.4. Задать желаемую точку монтирования на клиентской части системы: (по умолчанию точка монтирования /mnt/kfs)

```
pbit_kfs_mount_point: <точка монтирования>
```

ПРИМЕЧАНИЕ — синтаксис YAML зависит от пробелов.

### 6. Выполнить программу интерактивного формирования описи системы

```
promobit-make-inventory
```

ПРИМЕЧАНИЕ — при выполнении `promobit-make-inventory` также необходимо сделать описание клиентской части системы для автоматического развертывания и конфигурирования ПО.

### 7. Выполнить программу развертывания ПО

```
promobit-kfs-deploy
```

### 8. Убедиться, что все сервисы запущены

#### 8.1. На хостах с ролью **fsmaster** ввести команду

```
systemctl status kfs-master
```

#### 8.2 На хостах с ролью **fschunk** ввести команду

```
systemctl status kfs-chunkserver
```

ПРИМЕЧАНИЕ — в ОС «Эльбрус» данные команды отличаются:

```
service kfs-master status  
и
```

```
service kfs-chunkserver status  
соответственно.
```

#### 8.3. Убедиться, что интересующие процессы в состоянии `running`

### 9 Если на каких-либо хостах процесс не запустился, перезапустить его командой

```
systemctl restart <имя_процесса>
```

ПРИМЕЧАНИЕ — в ОС «Эльбрус» данная команда:

```
service <имя_процесса> restart
```

В качестве имени процесса выбираются `kfs-master` или `kfs-chunkserver` в соответствии с описью системы.

10. Если на каких-либо хостах в соответствии с п. 9 перезапускался процесс `kfs-chunkserver`, перейти в каталог `/opt/promobit/kfs-deploy/` и запустить роль добавления дисков командой

```
ansible-playbook -i /root/.kfs-inventory.yml /opt/promobit/kfs-deploy/add_disks.yml
```

## Установка из локальных установочных пакетов

1. Скопировать и установить полученный от инженеров ООО «Промобит» базовый установочный пакет

```
promobit-kfs-deploy<номер версии>.noarch.rpm
```

ПРИМЕЧАНИЕ — название пакета может отличаться

2. Скопировать в каталог `/opt/promobit/kfs-deploy/packages/` полученные от инженеров ООО «Промобит» установочные пакеты в соответствии с используемой вычислительной платформой. Примеры названий:

```
promobit-kfs-<номер версии>.e2kv4.rpm  
promobit-kfs-mount-<номер версии>.e2kv4.rpm
```

ПРИМЕЧАНИЕ — названия пакетов могут отличаться

3. Перейти в каталог `/opt/promobit/kfs-deploy/group_vars/`, сделать копию файла `all.yml.example` и назвать ее `all.yml`.

4. Отредактировать файл `all.yml`:

- 4.1.1. При установке из локальных установочных пакетов параметр `pbit_install_local_packages` принимает значение `yes`:

```
pbit_install_local_packages: yes
```

- 4.1.2. Задать размер блока KFS (по умолчанию 65536):

```
pbit_kfs_block_size: <размер блока>
```

- 4.1.3. Задать желаемую точку монтирования на клиентской части системы:  
(по умолчанию точка монтирования `/mnt/kfs`)

```
pbit_kfs_mount_point: <точка монтирования>
```

ПРИМЕЧАНИЕ — синтаксис YAML зависит от пробелов.

Далее выполнять установку согласно вышеприведенным пп. 6–11 (см. Установка из репозитория).

## Проверка установки

1. На хостах с ролью **fsmaster** ввести команду

```
systemctl status kfs-master
```

2. На хостах с ролью **fchunk**, ввести команду

```
systemctl status kfs-chunkserver
```

ПРИМЕЧАНИЕ — в ОС «Эльбрус» данные команды отличаются:

```
service kfs-master status
```

и

```
service kfs-chunkserver status
```

соответственно.

3. Убедиться, что что интересующие процессы в состоянии `running`
4. При установке в режиме отказоустойчивого кластера убедиться, что выбран лидер и остальные мастер-серверы к нему подключены:

```
kfs-admin metadataserver-status <ip> <port>
```

Интересующий сервер метаданных должен находиться или в состоянии `running` (для случая, когда значение параметра `personality: master`), или в состоянии `connected` (для случая, когда значение параметра `personality: shadow`).

ПРИМЕЧАНИЕ — в качестве `<ip>` и `<port>` необходимо указать ip-адрес интересующего хоста и порт (по умолчанию для сервера метаданных значение `<port> = 9421`).

ПРИМЕЧАНИЕ — подробнее о `kfs-admin` см. **4.4 Утилита `kfs-admin`**.

5. Если проверка завершилась неудачно, можно удалить установку командой

```
promobit-kfs-remove
```

и повторить процесс развертывания ПО серверной части системы.



# УСТАНОВКА КЛИЕНТСКОЙ ЧАСТИ ПО BITBLAZE KFS

При использовании операционной системы ОС Альт Сервер установка клиентской части ПО и монтирование файловой системы KFS выполняется автоматически при развертывании и конфигурировании компонент серверной части.

Для проверки монтирования файловой системы KFS на компьютерах клиентской части выполнить команду `mount` и проанализировать вывод. Для упрощения вывода можно выполнить команду:

```
mount | grep <точка монтирования>
```

или команду:

```
mount | grep 9421
```

Варианты ожидаемого вывода:

```
kfsmaster:9421 on /mnt/kfs type fuse.kfs (rw,nosuid,nodev,relatime,user_id=0,group_id=0,default_permissions,allow_other)
```

или:

```
<masterhost ip-address>:9421 on /mnt/kfs type fuse.kfs (rw,nosuid,nodev,relatime, user_id=0,group_id=0,default_permissions,allow_other)
```

**ВНИМАНИЕ** — Выполнение приведенных далее команд в обычном случае (когда клиентская часть устанавливается вместе с серверной) **не требуется**.

## Подготовка системы

1. Войти в систему под пользователем `root`.
2. Для обновления репозитория выполнить команду

```
apt-get update
```

**ПРИМЕЧАНИЕ** — для редактирования текстовых файлов по ходу дальнейшего выполнения установки ПО Bitblaze KFS необходим текстовый редактор, который, как правило, предустановлен в системе.

3. Убедиться в наличии в системе текстового редактора. При его отсутствии необходимо выполнить установку. Например, установить текстовый редактор «nano», выполнив команду

```
apt-get install nano
```

## Установка

### 1. При установке из репозитория:

#### 1.1. Создать файл `/etc/apt/sources.list.d/promobit.list` и добавить в него полученные от инженеров ООО «Промобит» репозитории (названия могут отличаться):

```
rpm http://<имя_пользователя:пароль>@repos.promobit.ru/kfs/p9/branch/ <arch> kfs
rpm http://<имя_пользователя:пароль>@repos.promobit.ru/kfs/p9/branch/ noarch kfs
```

#### 1.2. Выполнить обновление

```
apt-get update
```

### 2. Установить пакет или (вариант 1) локально:

```
apt-get install <название установочного пакета>
```

ПРИМЕЧАНИЕ — локальный установочный пакет необходимо получить предварительно от инженеров ООО «Промобит» в соответствии с используемой вычислительной платформой; пример названия:  
`promobit-kfs-mount-<номер версии>.e2kv4.rpm`.

### или (вариант 2) из репозитория:

```
apt-get install promobit-kfs-mount
```

ПРИМЕЧАНИЕ — возможны два варианта:

- 1) в серверной части системы 1 хост,
- 2) в серверной части системы более двух хостов.

### 3. Если в серверной части системы 1 хост, выполнить пункты 3.1 – 3.3, пункты 4.1 – 4.3 не выполнять.

#### 3.1. В файле `/opt/promobit/kfs/cfg/kfs-mount.cfg` указать путь монтирования, например `/mnt/kfs/`

#### 3.2 В этом же файле задать параметр `MFSBLOCKSIZE = <значение>`. Для этого необходимо воспользоваться файлом `group_vars/all.yml` (серверная часть системы), где требуемое число можно взять из параметра `pbit_kfs_block_size`: Например, в файле `group_vars/all.yml` задано

```
pbit_kfs_block_size: 65536
```

Тогда в файле `/opt/promobit/kfs/cfg/kfs-mount.cfg` тоже надо задать параметр

```
MFSBLOCKSIZE=65536
```

### 3.3. В файле `kfs-mount.cfg` добавить ip-адрес хоста с KFS-ролью **fsmaster**

```
mfsmaster=<ip>
```

4. Если в серверной части системы более двух хостов, то выполнить следующие действия.

4.1. Сделать копию файла `/opt/promobit/kfs/cfg/kfs-chunkserver.toml` в серверной части системы.

4.2. Переименовать в `kfs-mount.toml` копию файла, сделанную в п.4.1.

4.3. Скопировать файл `kfs-mount.toml` на каждую клиентскую машину в каталог `/opt/promobit/kfs/cfg/`.

5. Для монтирования файловой системы KFS выполнить команду

```
kfs-mount-tool <точка монтирования>
```

ПРИМЕЧАНИЕ — каталог для монтирования файловой системы должен быть предварительно создан.

## Проверка монтирования

Для проверки монтирования файловой системы KFS на компьютерах клиентской части выполнить команду `mount` и проанализировать вывод. Для упрощения вывода можно выполнить команду

```
mount | grep <точка монтирования>
```

или команду

```
mount | grep 9421
```

## Варианты ожидаемого вывода

```
kfsmaster:9421 on /mnt/kfs type fuse.kfs (rw,nosuid,nodev,relatime,user_id=0,group_id=0,default_permissions,allow_other)
```

или

```
<masterhost ip-address>:9421 on /mnt/kfs type fuse.kfs (rw,nosuid,nodev,relatime, user_id=0,group_id=0,default_permissions,allow_other)
```

# УПРАВЛЕНИЕ ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ

Возможны варианты применения ПО Bitblaze KFS в СХД в соответствии с различиями в составе аппаратного обеспечения.

Конфигурационные файлы `.cfg` и `.toml` находятся в каталоге `/opt/promobit/kfs/cfg/`

**ВНИМАНИЕ** — В данной версии ПО конфигурационные файлы серверной части создаются автоматически при развертывании и не требуют редактирования. Внесение изменений в заданные значения параметров может привести к нарушениям в работе системы. При необходимости специальной настройки параметров функционирования необходимо обратиться в службу технической поддержки.

**ВНИМАНИЕ** — При необходимости дополнительного конфигурирования системы после развертывания ее компонент рекомендуется вносить требуемые правки в шаблоны соответствующих конфигурационных файлов и запускать на выполнение роли установки `ansible`.

## 4.1. Развертывание отказоустойчивого кластера

### 4.1.1 Описание ролей и порядок запуска процессов в отказоустойчивом кластере

В составе отказоустойчивого кластера могут быть выделены следующие роли и соответствующие им процессы:

**fsmaster** — роль сервера метаданных, процесс `kfs-master`;

**fschunk** — роль `chunk`-сервера, или сервера фрагментов, процесс `kfs-chunkserver`;

**client** — роль для клиентской части системы.

Роли `fsmaster` и `fschunk` настраиваются автоматизированно в процессе развертывания ПО серверной части.

Для функционирования отказоустойчивого кластера необходимо не менее трех хостов, на которых выполняются процессы с ролью `fsmaster`. Система автоматически выберет среди них лидера.

### 4.1.2. Запуск, рестарт и остановка процессов.

Процесс `kfs-chunkserver` реализует управление сервером фрагментов данных, процесс `kfs-master` — управление сервером метаданных. Синтаксис управляющих команд для процессов следующий: `<имя_процесса> [действие]`, где в качестве действий может быть указано:

```
start
stop
restart.
```

По умолчанию в качестве действия выбирается `restart`. Без дополнительных параметров команда `<имя_процесса>` запускает процесс, снимая с выполнения ранее запущенный процесс с тем же именем, если есть файл блокировки.

### 4.1.3 Настройка отказоустойчивого кластера

Для настройки отказоустойчивого кластера применяются конфигурационные файлы `.toml`. Эти конфигурационные файлы создаются автоматически при развертывании ПО и не требуют редактирования вручную.

**ВНИМАНИЕ** — В данной версии ПО конфигурационные файлы серверной части создаются автоматически при развертывании и не требуют редактирования. Внесение изменений в заданные значения параметров может привести к нарушениям в работе системы. При необходимости специальной настройки параметров функционирования необходимо обратиться в службу технической поддержки.

Для всех ролей, предусмотренных в каждой отдельно взятой конфигурации кластера, должны быть созданы `.toml`-конфигурации:

```
kfs-master.toml;  
kfs-chunkserver.toml;  
kfs-mount.toml;
```

**ПРИМЕЧАНИЕ** — данные конфигурационные файлы различаются только в части ID, а в остальном одинаковы для всех ролей.

#### Конфигурационный файл `.toml`.

В секции `[cluster.leader]`:

`heartbeat_period = <heartbeat_period>` — длительность интервала внутрисетевого обмена служебными пакетами между серверами в системе, измеряется в миллисекундах. Выполняется с целью получения информации о состоянии лидера — управляющего сервера: работоспособное или отказ. Значение данного параметра должно быть намного меньше, чем значение параметра `election_timeout_min`.

`election_timeout_min = <election_timeout_min>` — нижнее значение интервала времени до начала голосования, измеряется в миллисекундах.

`election_timeout_max = <election_timeout_max>` — верхнее значение интервала времени до начала голосования, измеряется в миллисекундах.

В случае, когда управляющий сервер считается отказавшим, среди оставшихся работоспособными серверов метаданных происходит голосование с целью выяснения, кто из них становится лидером взамен отказавшего. Время начала голосования определяется случайным образом на интервале `[election_timeout_min, election_timeout_max]`. Чем больше данный интервал, тем выше вероятность того, что не произойдет одновременная подача голосов от работоспособных серверов метаданных.

`quorum = 1` — наименьшее количество ролей с правом голоса в кластере (параметр необходим для голосования за нового лидера, т. е. управляющего сервера метаданных, при выходе предыдущего лидера из строя).

`follower_quorum = 1` — наименьшее количество ролей **fsmaster** с правом голоса в кластере.

В секции `[cluster.local]`:

`id = <id>` — id узла, на котором находится данный конфигурационный файл. Данный параметр должен принимать уникальные значения для узлов с ролями **fsmaster**. Для всех остальных ролей допустимо задавать `id = 0`.

Таким образом, в рассматриваемом случае id может выбираться в соответствии с таблицей 1.

| Роль в кластере | id         |
|-----------------|------------|
| fsmaster        | Уникальный |
| fschunk         | 0          |
| client          | 0          |

Таблица 1.

В секции `[cluster.nodes]`:

`id = <id>` — id роли в кластере;

`matocl_ip = <"ip">` — ip-адрес узла с данной ролью.

**ПРИМЕЧАНИЕ** — возможны ситуации, когда на одном хосте установлены несколько разных ролей. Тогда необходимо дополнительно задать значение портов:

`matocl_port = <port>`. Также может понадобиться задание ip-адресов и портов для связи роли `fsmaster` с другими ролями: параметры `matocs_ip`, `matocs_port`. Также может понадобиться задание ip-адресов и портов для связи роли **fsmaster** с другими ролями: параметры `matocs_ip`, `matocs_port`, `matoml_ip`, `matoml_port`.

**ПРИМЕЧАНИЕ** — для обозначения параметров используются следующие акронимы: `matocl` — от `Master_TO_Client`, `matocs` — от `Master_TO_ChunkServer`.

**ПРИМЕЧАНИЕ** — данная секция обязательна для всех ролей и одинакова во всех `.toml`-файлах.

## 4.2 Настройка параметров стандартной репликации

Серверы метаданных используют конфигурационный файл `kfsgoals.cfg`, в котором заданы схемы репликации для файлов и каталогов. Схема репликации предназначена для того, чтобы определить количество серверов для хранения и порядок распределения по ним реплицированных элементов данных.

Файл `kfsgoals.cfg` нужен для того, чтобы при необходимости переопределить схемы репликации, заданные по умолчанию.

**ВНИМАНИЕ** — При необходимости дополнительного конфигурирования системы после развертывания ее компонент рекомендуется вносить требуемые правки в шаблоны соответствующих конфигурационных файлов и запускать на выполнение роли установки `ansible`.

При первоначальном конфигурировании системы файл `kfsgoals.cfg` содержит следующие схемы репликации:

```
1 1 : _
2 2 : _ _
3 3 : _ _ _
4 4 : _ _ _ _
5 5 : _ _ _ _ _
```

Синтаксис описания схемы:

```
id имя : метка1 метка2 ...
```

В приведенных обозначениях:

`id` — уникальный числовой признак схемы;

`имя` — уникальный признак схемы, который можно задать в нечисловом виде;

`метка1 метка2 ...` — перечень меток chunk-серверов.

Каждая метка chunk-сервера, указанная в перечне, означает, что будет сделана одна копия на chunk-сервере, помеченном этой меткой. Специальный символ «\_» обозначает копию на любом chunk-сервере в системе. Система не делает копии фрагментов файлов на одном и том же сервере.

ПРИМЕЧАНИЕ — метки chunk-серверов назначаются в соответствии с обозначениями хостов, заданными в части «all: hosts:» в файле `inventory.yml` при развертывании ПО (по умолчанию: `host1, host2, ...`).

При выполнении развертывания системы по умолчанию устанавливается схема репликации с `id=1`, то есть делается одна копия на любом chunk-сервере.

Возможно конфигурирование до 40 схем. Каждому файлу в системе назначен один из идентификаторов, и репликация этого файла выполняется в соответствии со схемой, связанной с данным идентификатором.

ПРИМЕЧАНИЕ — вновь определенная схема повлияет на репликацию всех файлов, которым назначен соответствующий `id`.

Примеры схем репликации:

```
6 four_copies_anywhere : _ _ _ _
7 one_copy_in_host1_and_a_copy_anywhere : host1 _
8 three_copies_in_three_hosts : host1 host2 host3
```

### 4.3 Настройка клиентской части

Для монтирования файловой системы KFS применяется команда `kfs-mount-tool` с указанием точки монтирования:

```
kfs-mount-tool <точка монтирования>
```

ПРИМЕЧАНИЕ — каталог для монтирования файловой системы должен быть предварительно создан.

Основные параметры содержатся в конфигурационном файле: `kfs-mount.cfg`. Данный файл может содержать следующие параметры:

`mfsmaster=<HOST-ip>` — `ip`-адрес хоста с ролью **fsmaster** (по умолчанию `kfsmaster`);

`mfsport=<HOST-port>` — порт хоста с ролью **fsmaster** (по умолчанию 9421);  
`password=<password>` — пароль для доступа к **kfsmaster**;  
`/mnt/kfs` — точка монтирования (строка должна начинаться символом «/»);  
`fusebigwrites=<YES или NO>` — поддержка нестандартного размера блока FUSE на запись.  
По умолчанию размер блока 4 КиБ.  
`MFSBLOCKSIZE=65536` — размер блока KFS.

ПРИМЕЧАНИЕ — размер блока 64КиБ это значение по умолчанию.

`kfs_inode_cache_delay_ms=0` — задержка перед отправкой данных на chunk-сервер при записи, измеряется в миллисекундах. Варьирование этого параметра может способствовать увеличению производительности при выполнении последовательной записи для больших блоков. По умолчанию значение параметра равно 0, означающее отсутствие задержки перед отправкой. Рекомендуемый диапазон значений для данного параметра от 0 до 300.

`kfs_inode_cache_blocks_number=1` — количество кэшируемых блоков перед отправкой на chunk-сервер. Варьирование этого параметра может способствовать увеличению производительности при выполнении последовательной записи для больших блоков. По умолчанию значение параметра равно 1. Рекомендуемый диапазон значений от 1 до 20. Накопление блоков перед записью происходит в оперативной памяти, необходимый объем которой определяется через установленное значение размера блока KFS:

```
RAM_CACHE = kfs_inode_cache_blocks_number * MFSBLOCKSIZE.
```

#### 4.4 Утилита **kfs-admin**

Для контроля состояния и администрирования системы может применяться утилита **kfs-admin**.

**ВНИМАНИЕ** — Внесение изменений в заданные значения параметров может привести к нарушениям в работе системы. При необходимости специальной настройки параметров функционирования необходимо обратиться в службу технической поддержки.

Формат вызова:

```
kfs-admin [команда] [опции] [аргументы]
```

ПРИМЕЧАНИЕ — Далее по тексту `<master ip>` и `<master port>` — ip-адрес и порт хоста с соответствующей ролью: **fsmaster** или **fschunk**.

#### Список возможных команд

```
chunks-health <master ip> <master port>
```

вывод отчета о состоянии фрагментов файлов в системе. По умолчанию будут показаны все отчеты



В частях таблицы «Chunks replication state» И «Chunks deletion state» числа в столбцах показывают количество фрагментов с количеством копий, помеченных для репликации или удаления соответственно.

### Доступны опции:

|   |  |
|---|--|
| <code>&lt;&lt;--availability&gt;&gt;</code>                             | отчет о доступности фрагментов   |
| <code>&lt;&lt;--replication&gt;&gt;</code>                              | отчет о количестве фрагментов, для которых необходима репликация,  |
| <code>&lt;&lt;--deletion&gt;&gt;</code>                                 | отчет о количестве фрагментов, которые необходимо удалить  |
| <code>info &lt;master ip&gt; &lt;master port&gt;</code>                 | вывод сводной информации о системе   |
| <code>iolimits-status &lt;master ip&gt; &lt;master port&gt;</code>      | вывод настроек глобальных ограничений ввода-вывода   |
| <code>list-chunkservers &lt;master ip&gt; &lt;master port&gt;</code>    | вывод данных обо всех chunk-серверах в системе   |
| <code>list-defective-files &lt;master ip&gt; &lt;master port&gt;</code> | вывод пути к файлам, для которых в системе есть недоступные фрагменты или фрагменты с неопределенной схемой репликации |

### Доступны опции:

|   |   |
|---|---|
| <code>&lt;&lt;--unavailable&gt;&gt;</code>                    | отчет о фрагментах с флагом «недоступен»                                  |
| <code>&lt;&lt;--undergoal&gt;&gt;</code>                      | отчет о количестве фрагментов, для которых не определена схема репликации |
| <code>&lt;&lt;--limit&gt;&gt;</code>                          | ограничение количества отображаемых файлов                                |
| <code>list-disks &lt;master ip&gt; &lt;master port&gt;</code> | вывод данных обо всех chunk-серверах и дисках в системе                   |

### Доступны опции:

|  |   |
|--|---|
| <code>&lt;&lt;--verbose&gt;&gt;</code>                         | более подробный отчет со статистикой операций         |
| <code>list-goals &lt;master ip&gt; &lt;master port&gt;</code>  | вывод списка схем репликации, определенных в системе. |
| <code>list-mounts &lt;master ip&gt; &lt;master port&gt;</code> | вывод данных обо всех точках монтирования в системе   |

|  |   |
|--|---|
| <code>metadataserver-status &lt;master ip&gt; &lt;master port&gt;</code>               | вывод сведений о состоянии серверов метаданных  |
| <code>ready-chunkservers-count &lt;master ip&gt; &lt;master port&gt;</code>            | вывод количества chunk-серверов, на которые можно производить запись.   |
| <code>list-metadataservers &lt;master ip&gt; &lt;master port&gt;</code>                | вывод списка активных серверов метаданных.  |
| <code>stop-master-without-saving-metadata &lt;master ip&gt; &lt;master port&gt;</code> | команда останавливает главный сервер метаданных без сохранения метаданных в файл <code>metadata.mfs</code> . Применяется для быстрой миграции сервера метаданных.   |
| <code>reload-config &lt;master ip&gt; &lt;master port&gt;</code>                       | запрос на выполнение повторной загрузки параметров, заданных в конфигурационных файлах  |
| <code>save-metadata &lt;metadataserver ip&gt; &lt;metadataserver port&gt;</code>       | запрос на сохранение текущего состояния метаданных в файл <code>metadata.mfs</code> . С опцией <code>-async</code> не выполняется, если процесс не может быть запущен, например, по причине того, что процесс уже выполняется.<br>Без этой опции не выполняется или если процесс не может быть запущен, или если он завершился с ошибкой (то есть файл метаданных не был создан). |
| <code>list-tasks &lt;master ip&gt; &lt;master port&gt;</code>                          | вывод задач, выполняемых в данное время на главном сервере метаданных, и соответствующих им <code>id</code> для применения команды <code>stop-task</code>   |
| <code>stop-tasks &lt;master ip&gt; &lt;master port&gt; &lt;task id&gt;</code>          | остановить выполнение задачи с указанным <code>id</code> .  |

**ПРИМЕЧАНИЕ** — команда `promote-shadow <shadow ip> <shadow port>` в данной версии ПО не поддерживается.

## 4.5 Утилита kfs-tools

Для контроля состояния и администрирования системы можно применять утилиту kfs-tools.

**ВНИМАНИЕ** — Внесение изменений в заданные значения параметров может привести к нарушениям в работе системы. При необходимости специальной настройки параметров функционирования необходимо обратиться в службу технической поддержки.

Формат вызова:

```
kfs-tools [команда] [опции] [параметр] [объект]
```

Опция `-r` включает так называемый «рекурсивный» режим, применяемый, например, для смены признаков для директории и вложенных в нее объектов (к признакам относятся схемы репликации, время удаления из корзины и др.).

Данная опция необходима по следующим причинам.

Когда в файловой системе KFS создается новый объект, он получает признаки от родительского каталога. Но если, например, для некоторого каталога поменять признаки, это не отразится на вложенных объектах. При необходимости распространить новые признаки вглубь директории применяется опция `-r`.

Для выбора формата отображаемых чисел применяются следующие опции:

- `-n` — полноформатный вывод значений (без десятичных или двоичных приставок);
- `-h` — вывод значений с двоичными приставками;
- `-H` — вывод значений с десятичными приставками;

**ПРИМЕЧАНИЕ** — команды `repquota`, `setquota`, `makesnapshot`, `appendchunks` в данной версии ПО не поддерживаются.

### Список возможных команд.

```
getgoal [-r] [-n или -h или -H] объект_KFS
```

возвращает схему репликации для указанного объекта

```
setgoal [-r] [-n или -h или -H] GOAL_NAME объект_KFS
```

устанавливает схему репликации GOAL\_NAME для указанного объекта;

**ПРИМЕЧАНИЕ** — устаревшие команды «`rgetgoal [-n или -h или -H] объект_KFS`», «`rsetgoal [-n или -h или -H] GOAL_NAME объект_KFS`», аналогичны вышеприведенным с опцией `-r`.

```
gettrashtime [-r] [-n или -h или -H] объект_KFS
```

возвращает для указанного объекта время его хранения в корзине до момента окончательного удаления

```
settrashtime [-r] [-n или -h или -H] время_в_секундах  
+ или -] объект_KFS
```

устанавливает для указанного объекта время его хранения в корзине до момента окончательного удаления

**ПРИМЕЧАНИЕ** — если добавлен символ «+», то текущее время хранения увеличивается до заданной величины для объектов, значение времени хранения которых было меньше заданного значения (для остальных — не меняется), а если добавлен символ «-», то текущее время хранения уменьшается до заданной величины для объектов, значение времени хранения которых было больше заданного значения (для остальных — не меняется).

ПРИМЕЧАНИЕ — устаревшие команды «`rgettrashedtime`», «`rsettrashedtime`» аналогичны вышеприведенным с опцией `-r`.

`geteattr [-r] [-n или -h или -H] объект_KFS`

возвращает дополнительные признаки для указанного объекта

`seteattr [-r] [-n или -h или -H] -f признак [-f признак ...] объект_KFS`

устанавливает дополнительные признаки для указанного объекта

`deleattr [-r] [-n или -h или -H] -f признак [-f признак ...] объект_KFS`

удаляет дополнительные признаки для указанного объекта

ПРИМЕЧАНИЕ — к дополнительным признакам относятся: `poowner` (означает, что данный объект принадлежит текущему пользователю), `poattrcache` (стандартные атрибуты файлов не будут храниться в кэше ядра), `poentrycache` (не допускать, чтобы записи директорий кэшировались в ядре).

`checkfile файл_KFS...`

проверяет и выводит количество фрагментов и количество копий фрагментов заданного(ных) файла(ов), в том числе файлов в корзине

`fileinfo файл_KFS...`

вывод сводной информации о заданном объекте

`filerepair [-n или -h или -H] файл_KFS...`

восстанавливает поврежденные файлы, делая их частично читаемыми

ПРИМЕЧАНИЕ — в случае потери фрагмента производится заполнение недостающей части нулями, а в случае несовпадения версий фрагментов будет установлена наиболее высокая из тех, что есть на серверах фрагментов. Во втором случае несовпадение содержимого может произойти в фрагментах с одинаковыми версиями. Поэтому рекомендуется делать копию и удалять оригинальный файл после «восстановления».

`rremove [-l] объект_KFS...`

рекурсивное удаление объектов (файлов и директорий)

ПРИМЕЧАНИЕ — опция «`-l`» отключает установленные тайм-ауты для данной команды (по умолчанию для тайм-аута установлено значение 60 секунд).

## 4.6 Удаление файлов

Функциональность файловой системы Bitblaze KFS включает настраиваемую «Корзину», в которую отправляются удаленные пользователями файлы. Точнее, это специальная точка доступа к метаданным.

После удаления файлы продолжают храниться в системе и могут быть окончательно удалены по решению администратора или по прошествии заданного времени нахождения в «Корзине». По умолчанию время до окончательного удаления файлов равно 86400 секунд, то есть 24 часа. Для доступа к корзине необходимо на всех нодах кластера с ролью **fsmaster** выполнить следующие действия.

**ВНИМАНИЕ** — При необходимости дополнительного конфигурирования системы после развертывания ее компонент рекомендуется вносить требуемые правки в шаблоны соответствующих конфигурационных файлов и запускать на выполнение роли установки ansible.

1. Отредактировать конфигурационный файл `kfsexports.cfg`, добавив ip-адреса хостов клиентской части, где может быть смонтирована точка доступа к метаданным системы.

Пример содержимого файла `kfsexports.cfg`:

```
<ip-адрес> . rw,alldirs,maproot=0
<ip-адрес> / rw,alldirs,maproot=0
```

2. Перезапустить процессы `kfs-master` командой

```
systemctl restart kfs-master
```

**ПРИМЕЧАНИЕ** — в ОС «Эльбрус» данные команды отличаются:

```
service kfs-master restart
и
service kfs-chunkserver restart
соответственно.
```

3. На соответствующих хостах клиентской части выполнить команду монтирования с указанием каталога монтирования. Пример команды:

```
kfs-mount -e /opt/promobit/kfs/cfg/kfs-mount.toml /mnt/kfs-meta/ -o mfsmeta
```

**ПРИМЕЧАНИЕ** — В данном примере команды монтирования `/mnt/kfs-meta/` — заданный каталог монтирования метаданных (если он не существует, его необходимо создать).

В результате монтирования в системе появятся каталоги следующего вида:

```
├reserved
├trash
└undel
```

Директория `trash` («Корзина») — место хранения временно удаленных файлов, которые еще можно восстановить.

В системе может быть восстановлен любой файл, атрибут `“trashtime”` которого больше нуля. Для этого достаточно переместить его из директории `trash` в директорию `undel`. Перемещенный файл появится не в директории `undel`, а в директории, откуда он был ранее удален.

Для окончательного удаления файла достаточно удалить его из директории `trash`. При этом свободное пространство на дисках появится не мгновенно, а через некоторый промежуток времени, необходимый для удаления фрагментов данных.

ПРИМЕЧАНИЕ — имена файлов представлены композитной строкой, содержащей:

1) inode файла и список его директорий (`path`),

2) 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 (пользователями). Такие файлы помещаются в эту директорию до момента освобождения дескриптора.

Время до окончательного удаления файлов из «Корзины» можно переопределить при помощи опций `gettrashtime` и `settrashtime` утилиты `kfs-tools` (подробнее см. **4.5 Утилита `kfs-tools`**).

# ЕСЛИ НЕ ПОЛУЧАЕТСЯ

## Свяжитесь со службой технической поддержки

Адрес электронной почты: [help@bitblaze.ru](mailto:help@bitblaze.ru)

Интернет-сайт: [bitblaze.ru/support/](https://bitblaze.ru/support/)

Техническая поддержка программного обеспечения Bitblaze KFS включает следующий набор услуг:

- Предоставление обновлений программного обеспечения по мере выхода новых релизов.
- Консультация ИТ-специалистов заказчика по работе.
- Помощь в устранении инцидентов и обновлении ПО в удаленном режиме.

**Стандартная базовая поддержка предоставляется в рабочее время.**

**Премиум-поддержка предоставляется круглосуточно, в том числе в выходные дни.**

### **ООО «Промобит»**

Телефон компании: **+7 (3812) 36-11-11**

Адрес электронной почты: **info@bitblaze.ru**

Почтовый адрес: **644024, г. Омск, пр-т К. Маркса, д. 3/1.**

ПРИМЕЧАНИЕ — Все гарантии, касающиеся товаров и услуг, реализуемых ООО «Промобит, изложены в формулировках прямых гарантий, сопровождающих соответствующие товары и услуги. Никакая информация, приведенная в данном документе, не должна рассматриваться как дополнительная гарантия.

ООО «Промобит» не несет ответственности за технические или редакторские ошибки либо пропуски в данном документе.