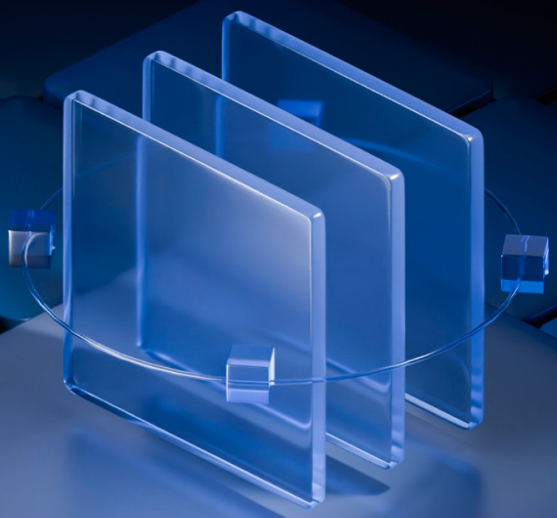
КИБЕРПРОТЕКТ

КИБЕР Хранилище

Версия 6.7



Руководство администратора по командной строке

Редакция: 15.07.2026

Оглавление

Введение

Данный раздел содержит основные сведения об этом документе и продукте Кибер Хранилище.

О данном руководстве

Данное руководство предназначено для продукта Кибер Хранилище, установленного отдельно от продукта Кибер Инфраструктура.

Сведения от том, как управлять продуктом Кибер Хранилище, установленным как часть продукта Кибер Инфраструктура, см. в Руководстве администратора Кибер Инфраструктуры.

О продукте Кибер Хранилище

Кибер Хранилище — это решение, позволяющее быстро и легко преобразовать недорогое оборудование для хранения данных и сетевое оборудование в защищенное хранилище корпоративного уровня, такое как SAN (Storage Area Network) или NAS (Network Attached Storage).

Решение Кибер Хранилище оптимизировано для хранения больших объемов данных и обеспечивает избыточность, высокую доступность и самовосстановление данных. Используя Кибер Хранилище, вы можете безопасно хранить различные данные, такие как, например, диски виртуальных машин, и предоставлять к ним доступ различными способами.

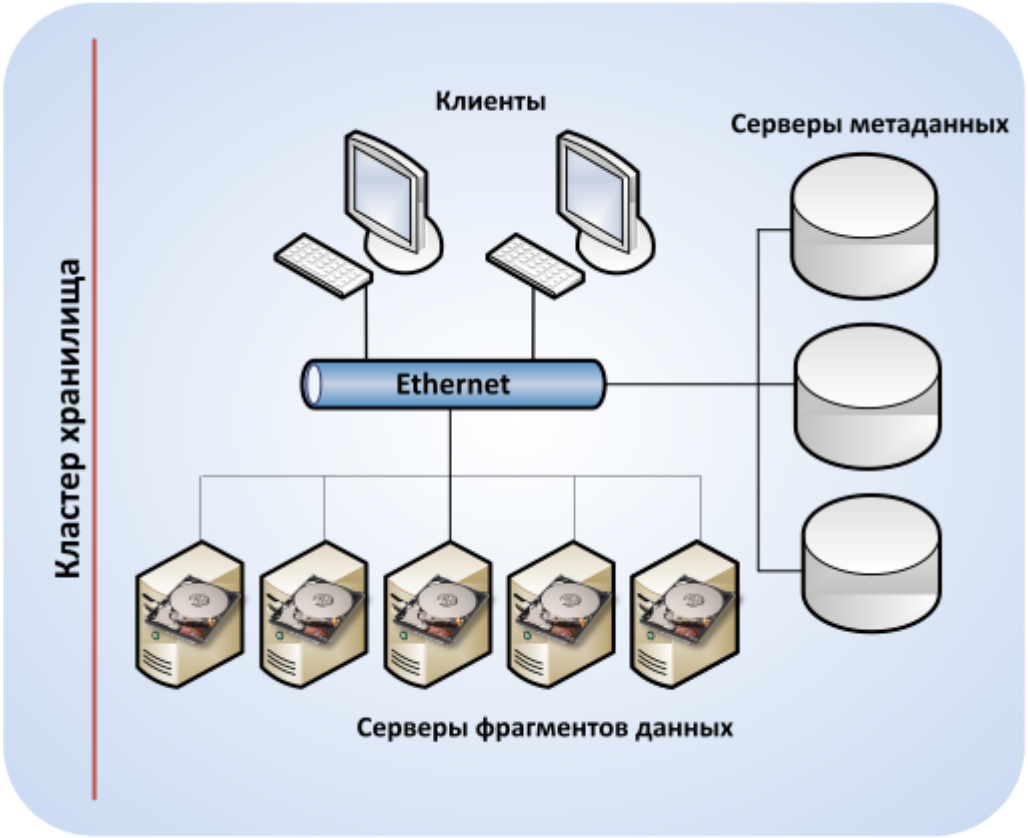
Планирование инфраструктуры для кластера хранилища

Чтобы спланировать инфраструктуру для кластера хранилища, вам необходимо определить конфигурацию каждого сервера кластера и спланировать сети хранилища.

Сведения в этом разделе помогут вам решить эти задачи.

Обзор архитектуры кластера хранилища

Перед началом процесса развертывания вы должны иметь четкое представление об инфраструктуре кластера хранилища. Типичный кластер хранилища показан ниже.



Основным компонентом продукта Кибер Хранилище является кластер хранилища. Кластер хранилища — это группа физических компьютеров, подключенных к одной сети Ethernet и выполняющих следующие роли:

- серверы фрагментов данных (CS),
- серверы метаданных (MDS),

- клиентские компьютеры (клиенты).

Все данные в кластере, включая образы дисков виртуальных машин, хранятся в виде фрагментов фиксированного размера на серверах фрагментов данных. Кластер автоматически реплицирует фрагменты и распределяет их между доступными серверами фрагментов данных для обеспечения высокой доступности.

Для отслеживания фрагментов данных и их реплик кластер хранит их метаданные на серверах метаданных (MDS). Один из серверов метаданных является главным. Он отслеживает всю активность кластера и поддерживает метаданные в актуальном состоянии.

Клиенты манипулируют данными, хранящимися в кластере, отправляя файловые запросы различного типа, которые, например, изменяют существующие файлы или создают новые.

- **Серверы фрагментов данных (CS)**. Серверы фрагментов данных хранят все данные в виде фрагментов фиксированного размера и предоставляют к ним доступ. Все фрагменты данных реплицируются, и их реплики хранятся на разных серверах для обеспечения высокой доступности. Если один из серверов фрагментов данных выйдет из строя, другие серверы фрагментов данных продолжат предоставлять фрагменты данных, которые хранились на вышедшем из строя сервере.
- **Серверы метаданных (MDS)**. Серверы метаданных хранят метаданные о серверах фрагментов данных и контролируют, как файлы с данными разбиваются на фрагменты и где эти фрагменты размещаются. Серверы метаданных также обеспечивают наличие в кластере достаточного количества реплик фрагментов данных и хранят глобальный журнал всех важных событий, происходящих в кластере.

  Чтобы обеспечить высокую доступность кластера хранилища, необходимо установить несколько серверов метаданных в кластере. В этом случае, если один сервер метаданных выйдет из строя, другой сервер метаданных продолжит контролировать кластер.

  > **Примечание**
  >
  > Серверы метаданных занимаются только обработкой метаданных и обычно не участвуют ни в каких операциях чтения/записи, связанных с фрагментами данных.

- **Клиенты**. Клиенты — это компьютеры, которые используют дисковое пространство кластера. Например, на клиенте могут быть запущены виртуальные машины, данные которых хранятся в кластере.

Обратите внимание на следующее:

- Любой компьютер в кластере можно настроить на выполнение роли сервера метаданных, сервера фрагментов данных или клиента. Можно также назначить две или все три роли одному и тому же

компьютеру. Например, можно настроить компьютер на роль клиента и запустить на нем виртуальные машины. Если вы хотите, чтобы этот компьютер в то же время предоставлял свое локальное дисковое пространство кластеру, вы можете настроить его как сервер фрагментов данных.

- Хотя дисковое пространство кластера хранилища можно монтировать как файловую систему, эта файловая система не является совместимой с POSIX и не имеет некоторых возможностей POSIX, таких как списки контроля доступа (ACL), пользователи и группы, жесткие ссылки и некоторые другие.

Конфигурации кластера хранилища

В этом разделе представлены сведения о двух конфигурациях кластера хранилища:

- *Минимальная конфигурация*. Вы можете использовать минимальную конфигурацию для оценки возможностей продукта Кибер Хранилище. Однако эту конфигурацию не рекомендуется использовать в производственных средах.
- *Рекомендуемая конфигурация*. Рекомендуемую конфигурацию кластера хранилища можно использовать в производственной среде в исходном виде или адаптировать ее под свои нужды.

Минимальная конфигурация

Ниже приведена минимальная конфигурация оборудования для развертывания кластера хранилища.

Роль сервера	**Количество серверов**
Сервер метаданных	1 (может использоваться совместно с серверами фрагментов данных и клиентами)
Сервер фрагментов данных	1 (может использоваться совместно с серверами метаданных и клиентами)
Клиент	1 (может использоваться совместно с серверами метаданных и серверами фрагментов данных)
Общее количество серверов	1 (совместное использование) 3 (раздельное использование)

Графически минимальная конфигурация может быть представлена следующим образом:

Чтобы кластер хранилища функционировал, в нем должны быть как минимум один сервер метаданных, один сервер фрагментов данных и один клиент. Минимальная конфигурация имеет два основных ограничения:

- Кластер имеет один сервер метаданных, который представляет собой единую точку отказа. Если сервер метаданных выйдет из строя, весь кластер станет неработоспособным.
- Кластер имеет один сервер фрагментов данных, который может хранить только одну реплику фрагмента данных. Если сервер фрагментов данных выйдет из строя, кластер приостановит все операции с фрагментами до тех пор, пока в него не будет добавлен новый сервер фрагментов данных.

Рекомендуемая конфигурация

В таблице ниже приведены две рекомендуемые конфигурации для развертывания кластеров хранилища.

Сервер метаданных	Сервер фрагментов данных	Общее количество серверов
3 (могут использоваться совместно с серверами фрагментов данных и клиентами)	5-9 (могут использоваться совместно с серверами метаданных и клиентами)	5 или больше (в зависимости от количества клиентов и серверов фрагментов данных, а также от того, совмещают ли они роли)

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Сервер метаданных	Сервер фрагментов данных	Общее количество серверов
5 (могут использоваться совместно с серверами фрагментов данных и клиентами)	10 и больше (могут использоваться совместно с серверами метаданных и клиентами)	5 или больше (в зависимости от количества клиентов и серверов фрагментов данных, а также от того, совмещают ли они роли)
Клиенты	1 или больше Вы можете включить в кластер любое количество клиентов. Серверы, выступающие в роли клиентов, можно совместно использовать с серверами фрагментов данных и метаданных. Например, можно иметь 5 физических серверов и настроить каждый из них на одновременную работу в качестве сервера метаданных, сервера фрагментов данных и клиента.	

Хотя новые кластеры по умолчанию настроены так, что для каждого фрагмента данных создается по 1 реплике, для обеспечения высокой доступности данных необходимо настроить кластер так, чтобы создавалось не менее 3 реплик для каждого фрагмента данных.

В целом, рекомендуется создавать кластеры хранилища, состоящие из не менее 9 серверов. Кластеры меньшего размера будут работать так же хорошо, но не обеспечат значительных преимуществ в производительности по сравнению с хранилищем с прямым подключением (DAS) или улучшенного времени восстановления.

Обратите внимание на следующее:

- Для больших кластеров очень важно настроить правильные области отказа, чтобы повысить доступность данных. Дополнительные сведения см. в разделе *Конфигурация областей отказа*.
- В маленьких и средних кластерах серверы метаданных потребляют мало ресурсов и не требуют установки на выделенные серверы.
- Маленький кластер — это 3-5 серверов, средний кластер — от 6 до 15-20 серверов, а большой кластер — 15-20 и больше серверов.
- Время на всех серверах кластера должно быть синхронизировано по протоколу NTP. Это облегчит службе поддержки работу с журналами кластера (например, в случаях миграций, аварийных переключений и т. д.).

Требования к оборудованию

Перед созданием кластера хранилища убедитесь, что у вас есть в наличии все необходимое оборудование.

Также рекомендуем:

- ознакомиться с разделом *Использование дисков SSD*, чтобы узнать, как можно повысить производительность кластера за счет использования твердотельных накопителей для журналирования записи и кэширования данных, а также сколько SSD-накопителей может понадобиться в зависимости от количества жестких дисков в вашем кластере;
- ознакомиться с разделом *Приложение A. Устранение неполадок*, чтобы узнать о распространенных аппаратных проблемах и неправильных конфигурациях, которые могут повлиять на производительность кластера и привести к несогласованности и повреждению данных.

Общие требования

Каждой службе (MDS, CS или клиент) требуется 1,5 ГБ свободного места в корневом разделе (`/`) для журналов. Например, для запуска 1 службы метаданных, 1 клиента и 12 служб фрагментов данных на сервере вам потребуется 21 ГБ свободного места в корневом разделе.

Службы метаданных

- 1 ядро ЦП
- 1 ГБ ОЗУ на 100 ТБ данных в кластере
- 3 ГБ дискового пространства на 100 ТБ данных в кластере
- 1 или больше Ethernet-адаптеров (1 Гбит/с или быстрее)

> **Примечание**
>
> Рекомендуется размещать журнал службы метаданных на SSD-диске, выделенном или общем с кэшами служб фрагментов данных и клиентов, или, по крайней мере, на выделенном жестком диске, который не используется службами фрагментов данных.

Службы фрагментов данных

- 1/8 ядра ЦП (например, 1 ядро ЦП на 8 служб фрагментов данных)
- 1 ГБ ОЗУ
- 100 ГБ или больше свободного дискового пространства
- 1 или больше адаптеров Ethernet (1 Гбит/с или быстрее)

Обратите внимание на следующее:

- Сведения об использовании локального RAID-массива с кластером хранилища см. в разделе *Возможные конфигурации дисковых накопителей*.

- Использование общего массива JBOD на нескольких серверах, на которых запущены службы фрагментов данных, может привести к возникновению единой точки отказа и сделать кластер недоступным, если все реплики данных будут размещены и сохранены на неисправном массиве JBOD. Дополнительные сведения см. в разделе *Конфигурация областей отказа*.
- Для больших кластеров очень важно настроить соответствующие области отказа, чтобы повысить доступность данных. Дополнительные сведения см. в разделе *Конфигурация областей отказа*.
- Не используйте для служб фрагментов данных диски, которые уже используются (например, для пространства подкачки или операционной системы). Совместное использование дисков службами фрагментов данных и другими источниками нагрузки на диски приведет к значительной потере производительности и высоким задержкам операций ввода-вывода.

Клиенты

- 1 ядро ЦП в расчете на 30 000 операций ввода-вывода в секунду
- 1 ГБ ОЗУ
- 1 или больше Ethernet-адаптеров (1 Гбит/с или быстрее)

В следующей таблице указана максимальная производительность сети, которую может получить клиент кластера хранилища с указанным сетевым интерфейсом. Для клиентов рекомендуется использовать сетевое оборудование, поддерживающее обмен данными со скоростью 10 Гбит/с между любыми двумя серверами кластера, и минимизировать сетевые задержки, особенно если используются SSD-диски.

Сетевой интерфейс для сети хранилища	**1 Гбит/с**	**2 x 1 Гбит/с**	**3 x 1 Гбит/с**	**10 Гбит/с**	**2 x 10 Гбит/с**
Максимальная пропускная способность всего сервера	100 МБ/с	~175 МБ/с	~250 МБ/с	1 ГБ/с	1,75 ГБ/с
Максимальная пропускная способность ВМ (репликация)	100 МБ/с	100 МБ/с	100 МБ/с	1 ГБ/с	1 ГБ/с
Максимальная пропускная способность ВМ (помехоустойчивое кодирование)	70 МБ/с	~130 МБ/с	~180 МБ/с	700 МБ/с	1,3 ГБ/с

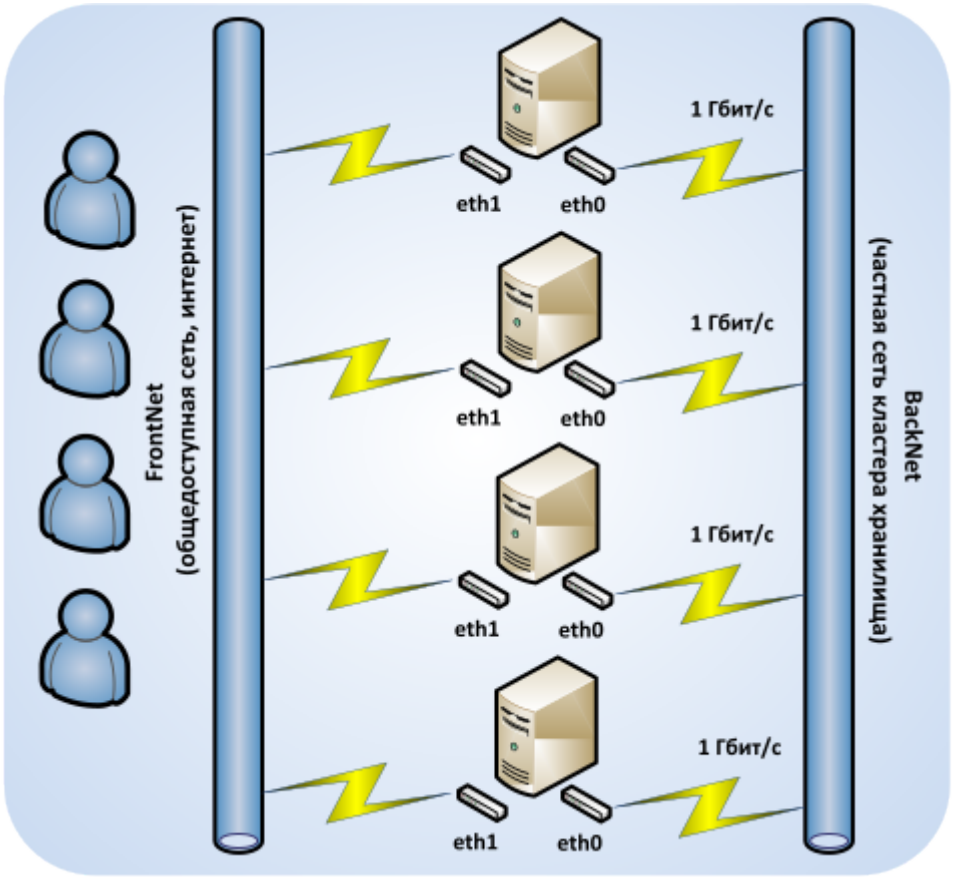
Требования к сети

При планировании сети убедитесь, что она:

- работает на скорости 1 Гбит/с или быстрее (подробности см. в разделе *Использование сетей 1 GbE и 10 GbE*),
- имеет неблокирующие Ethernet-коммутаторы.

Для пользовательского и кластерного трафика следует использовать отдельные сети и Ethernet-адаптеры. Это предотвратит возможное снижение производительности ввода-вывода в кластере из-за внешнего трафика. Кроме того, если кластер доступен из публичных сетей (например, из интернета), он может стать мишенью для атак типа ”отказ в обслуживании” и вся подсистема ввода-вывода кластера может выйти из строя.

На рисунке ниже показан пример конфигурации сети для кластера хранилища.



В этой конфигурации сети:

- BackNet — это частная сеть, используемая исключительно для соединения и взаимодействия серверов в кластере и являющаяся недоступной из публичной сети. Все серверы в кластере

подключены к этой сети через одну из своих сетевых карт.

- FrontNet — это общедоступная сеть, которую клиенты используют для доступа к своим данным, хранящимся в кластере.

Обратите внимание на следующее:

- Сетевые коммутаторы являются очень распространенной точкой отказа, поэтому для повышения доступности данных очень важно настроить соответствующие области отказа. Дополнительные сведения см. в разделе *Конфигурация областей отказа*.
- Дополнительные сведения о сетях кластера хранилища (в частности, сведения о том, как привязать службы фрагментов данных к определенным IP-адресам) см. в разделе *Защита связи между серверами кластера*.

Развертывание серверов хранения на базе РЕД ОС

Установка РЕД ОС в данном разделе не рассматривается. До начала выполнения работ необходимо получить архив `cyber-storage-DDD-NNN-BBBB.tgz` и разместить его в домашней директории пользователя `root`.

Подготовка узла хранения

Обновление ядра ОС

Проверка версии ядра ОС

```
uname -r
```

Если версия ядра меньше 6.1.52-1.el7.3.x86_64, необходимо его обновить (см. инструкцию ниже). Сертификация ОС после обновления не пострадает (подробнее см. в документации РЕД ОС).

Обновление ядра ОС

```
dnf update -y
dnf install redos-kernels6-release -y
dnf makecache
dnf update -y
reboot
```

После перезагрузки проверьте версию используемого ядра ОС (команда приведена выше).

Прочие операции по настройке

Извлечение содержимого архива

```
cd /root
tar -xzf cyber-storage-DDD-NNN-BBBB.tgz
```

Примечание

`cyber-storage-DDD-NNN-BBBB.tgz` — название полученного архива.

Выключение SELinux

```
sed -i ”s/SELINUX=enforcing/SELINUX=permissive/” /etc/selinux/config
setenforce 0
```

Установка пакетов

Для установки пакетов необходимо выполнить скрипт установщика, который располагается в папке `scripts`:

```
sh scripts/install-package.sh
```

Удаление пакетов

Для удаления пакетов необходимо выполнить команду:

```
uninstall-cyberstorage
```

Приведение файла конфигурации nginx.conf к требуемому виду

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log crit;
pid /run/nginx.pid;
worker_rlimit_nofile 884736;# 65536;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;
events {
    use epoll;
    multi_accept on;
    worker_connections 8192;
}

http {
    log_format  main    '$remote_addr - [$time_local] ”$request” '
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main buffer=20m;

    sendfile on;
    tcp_nopush on;
    keepalive_timeout 65;
    types_hash_max_size 4096;
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    server_names_hash_max_size 2048;
    server_names_hash_bucket_size 1024;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html
    # for more information.
    include /etc/nginx/conf.d/*.conf;
    server {
        listen 80;
        listen [::]:80;
        server_name _;
        root /usr/share/nginx/html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;
        error_page 404 /404.html;
        location = /404.html {
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }
}
```

Настройка ядра Linux

```
cat > /etc/sysctl.d/85-mhd.o.conf << EOF
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_timestamps = 1
kernel.pid_max = 4194303
fs.file-max = 9223372036854775807
vm.swappiness = 1
vm.vfs_cache_pressure = 100
vm.min_free_kbytes = 1000000
net.core.rmem_max = 268435456
net.core.wmem_max = 268435456
net.core.rmem_default = 67108864
net.core.wmem_default = 67108864
net.core.netdev_budget = 1200
net.core.optmem_max = 134217728
net.core.somaxconn = 65535
net.core.netdev_max_backlog = 250000
net.ipv4.tcp_rmem = 67108864 134217728 268435456
net.ipv4.tcp_wmem = 67108864 134217728 268435456
net.ipv4.tcp_low_latency = 1
net.ipv4.tcp_adv_win_scale = 1
net.ipv4.tcp_max_syn_backlog = 30000
net.ipv4.tcp_max_tw_buckets = 2000000
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_fin_timeout = 5
net.ipv4.udp_rmem_min = 8192
net.ipv4.udp_wmem_min = 8192
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.tcp_mtu_probing = 1
fs.aio-max-nr = 1048576
EOF
```

Применение настроек ядра Linux

```
sysctl --system
```

Настройка сервиса sshd

```
cat > /etc/ssh/sshd_config.d/45-cyberprotect.conf << EOF
MACs hmac-sha2-512,hmac-sha2-256,umac-128@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256-
↪etm@openssh.com,umac-128-etm@openssh.com
EOF
```

Применение настроек сервиса sshd

```
systemctl restart sshd
```

Включение автоматического запуска сервиса tuned и его запуск

```
systemctl enable --now tuned
```

Создание файлов профилей производительности

```
mkdir -p /etc/tuned/profiles/vstorage-random-io
cp -f vstorage-random-io.conf /etc/tuned/profiles/vstorage-random-io/tuned.conf
mkdir -p /etc/tuned/profiles/vstorage-sequential-io
cp -f vstorage-sequential-io.conf /etc/tuned/profiles/vstorage-sequential-io/tuned.conf
cp -f recommend.conf /etc/tuned/recommend.conf
```

Выбор профиля производительности

```
tuned-adm profile vstorage-random-io
systemctl enable --now adaptivemmd
```

Сведения о профилях производительности см. в документации РЕД ОС.

Выключение сетевого экрана

```
systemctl disable --now firewalld
systemctl disable --now iptables
iptables -F
```

Настройка сетевых интерфейсов

Пример создания сетевого объединения

```
nmcli con add type bond con-name <имя создаваемого сетевого объединения, например bond0> \
ifname <имя создаваемого сетевого объединения, например bond0> mode 802.3ad \
ip4 <IP-адрес сетевого объединения (если предусмотрен), например 192.168.10.11/24> mtu 9000

nmcli con mod id <имя создаваемого сетевого объединения, например bond0> \
bond.options mode=802.3ad,miimon=300,lacp_rate=fast,downdelay=300,updelay=300

nmcli con add type bond-slave ifname <имя сетевого интерфейса, добавляемого в сетевое объединение>
↪ \
con-name <имя сетевого интерфейса, добавляемого в сетевое объединение> \
master <имя создаваемого сетевого объединения, например bond0>

nmcli con add type bond-slave ifname <имя сетевого интерфейса, добавляемого в сетевое объединение>
↪ \
con-name <имя сетевого интерфейса, добавляемого в сетевое объединение> \
master <имя создаваемого сетевого объединения, например bond0>
```

Пример создания VLAN-интерфейса

```
nmcli con add type vlan con-name <имя создаваемого VLAN-интерфейса, например bond0.12> \
dev <имя интерфейса, поверх которого создается VLAN-интерфейс, например bond0> \
id <ID создаваемого VLAN-интерфейса, например 12> ip4 <IP-адрес VLAN-интерфейса, например 192.168.
↪100.11/24>
```

Подготовка диска

Проверка поддержки NVMe-диском пространств имен (NVMe namespaces)

```
nvme id-ctrl /dev/nvme<номер устройства> | grep ^nn
```

В случае если значение меньше 4, увеличение количества пространств имен не требуется.

Увеличение количества пространств имен

```
/usr/libexec/vstorage/split_nvme_ns 4 4096
```

Скрипт подготовки дисков

```
#!/bin/bash
# На вход скрипт ожидает список, разделенный символами пробела, например ”sda sdb sdc nvme1n1¬
↪nvme1n2”.
list=($1)

mklab() {
    parted -s /dev/${1} mklabel gpt
}

mkpart() {
    parted -a optimal /dev/${1} mkpart primary 0% 100%
}

mfs() {
    echo 'y' | mkfs.ext4 /dev/${1}1
}

for disk in ${list[*]}; do
    echo ”${disk}”
    if [ -f /sys/block/${disk}/queue/rotational ]; then
        mklab ${disk}
        mkpart ${disk}
        mfs ${disk}
        diskid=`blkid -o value -s UUID /dev/${disk}1`
        ssd=$(cat /sys/block/${disk}/queue/rotational)
        if [ $ssd != 1 ]; then
            echo ”UUID=${diskid} /vstorage/${diskid} ext4 defaults,noatime,lazytime,discard,
↪nofail 1 0” >> /etc/fstab
        else
            echo ”UUID=${diskid} /vstorage/${diskid} ext4 defaults,noatime,lazytime,nofail 1 0” >>
↪ /etc/fstab
        fi
        mkdir -p /vstorage/${diskid}
    else
        echo ”Error in parameter ${disk}”
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    fi
done


mount -a
```

Очистка диска и создание раздела с файловой системой

```
parted -s /dev/<имя диска> mklabel gpt
parted -a optimal /dev/<имя диска> mkpart primary 0% 100%
echo 'y' | mkfs.ext4 /dev/<имя созданного раздела>
```

Создание каталога монтирования для диска

```
mkdir -p /vstorage/<UUID созданного раздела>
# Для SSD-дисков с ролью "Кэш"
mkdir -p /vstorage/w-cache/<UUID созданного раздела>
```

Формирование записи в файл /etc/fstab для диска

```
# Для SSD-дисков
echo "UUID=$(blkid -o value -s UUID /dev/<имя созданного раздела>) /vstorage/<UUID созданного¬
↪раздела>  ext4    defaults,noatime,lazytime,discard,nofail 1 0" >> /etc/fstab
# Для SSD-дисков с ролью "Кэш"
echo "UUID=$(blkid -o value -s UUID /dev/<имя созданного раздела>) /vstorage/w-cache/<UUID¬
↪созданного раздела>  ext4    defaults,noatime,lazytime,discard,nofail 1 0" >> /etc/fstab
# Для HDD-дисков
echo "UUID=$(blkid -o value -s UUID /dev/<имя созданного раздела>) /vstorage/<UUID созданного¬
↪раздела>  ext4    defaults,noatime,lazytime,nofail 1 0" >> /etc/fstab
```

Монтирование дисков

```
mount -a
```

Развертывание кластера хранения

Первый сервер (мастер)

Создание первого сервиса MDS

```
export CLNAME=<имя кластера>
vstorage -c ${CLNAME} make-mds --init-cluster -a <IP-адрес узла из диапазона адресов сети¬
↪хранения> -r /vstorage/<UUID раздела>/${CLNAME}/ -p
systemctl enable --now vstorage-mdsd.target
```

Пароль, указанный при создании первого сервиса MDS, в дальнейшем будет использоваться при добавлении узлов в кластер.

Перед назначением роли диск необходимо подготовить (см. раздел *Подготовка диска*).

Расчет необходимой емкости диска с ролью ”Кэш”

Роль ”Кэш” используется для повышения производительности записи на HDD-диски путем переноса журналов записи на более быстрое устройство SSD, в том числе NVMe.

Сведения о расчете размера журнала операций записи см. в разделе *Расчет размера журнала записи*.

Создание сервиса CS

```
export CLNAME=<имя кластера>
# Операция повторяется для всех дисков с этой ролью.
echo '<пароль кластера>' | vstorage -c ${CLNAME} auth-node -P
# Для HDD-дисков
vstorage -c ${CLNAME} make-cs -r /vstorage/<UUID раздела>/data -t <уровень хранения для этого¬
↪диска (tier)>
# Для HDD-дисков с журналом записи (кэш для операций записи)
vstorage -c ${CLNAME} make-cs -r /vstorage/<UUID раздела>/data -j /vstorage/w-cache/<UUID раздела¬
↪диска с ролью ”Кэш”>/<UUID раздела> -s <размер журнала записи CS в МБ> -t <уровень хранения для¬
↪этого диска (tier)>
# Для SSD-дисков
vstorage -c ${CLNAME} make-cs -r /vstorage/<UUID раздела>/data -T ssd -t <уровень хранения для¬
↪этого диска (tier)>
```

Запуск сервисов CS и монтирование кластера хранения

```
systemctl enable --now vstorage-csd.target
mkdir -p /mnt/vstorage
echo '# Точка монтирования кластера' >> /etc/fstab
echo ”vstorage://${CLNAME} /mnt/vstorage fuse.vstorage  rw,nosuid,nodev,_netdev 0 0” >> /etc/fstab
mount -a
```

Проверка состояния кластера хранения

Для проверки состояния кластера хранения можно использовать следующие команды:

- `vstorage -c <имя кластера> top` — обновляемый вывод статуса кластера хранения;
- `vstorage -c <имя кластера> stat` — однократный вывод статуса кластера хранения.

При необходимости можно просмотреть список журналов (см. раздел *Просмотр списка журналов служб фрагментов данных*).

Добавление узла в кластер высокой доступности (High Availability)

```
export CLNAME=<имя кластера>
systemctl enable shaman.target shaman@${CLNAME}.service
systemctl daemon-reload
systemctl restart shaman.target shaman@${CLNAME}.service
shaman join --cluster ${CLNAME}
shaman set-roles S3
```

Последующие серверы

На момент подключения нового узла необходимо обеспечить:

- однотипные настройки сетевых интерфейсов, включая параметр ”MTU”;
- сетевую достижимость между мастер-узлом и подключаемым узлом по сети хранения;
- корректные настройки сетевого экрана.

Перед назначением роли диск необходимо подготовить (см. раздел *Подготовка диска*).

Для проверки доступности созданного кластера следует выполнить команду `vstorage discover`, которая выведет список достижимых кластеров хранения.

Добавление узла в кластер

```
echo '<пароль кластера>' | vstorage -c <имя кластера> auth-node -P
```

Создание сервиса MDS

```
export CLNAME=<имя кластера>
echo ”<пароль кластера>” | vstorage -c ${CLNAME} make-mds -a <IP-адрес узла из диапазона адресов¬
↪сети хранения> -r /vstorage/<UUID раздела>/${CLNAME}/
systemctl enable --now vstorage-mdsd.target
```

Расчет необходимой емкости диска с ролью ”Кэш”

Роль ”Кэш” используется для повышения производительности записи на HDD-диски путем переноса журналов записи на более быстрое устройство SSD, в том числе NVMe.

Сведения о расчете размера журнала операций записи см. в разделе *Расчет размера журнала записи*.

Создание сервиса CS

```
export CLNAME=<имя кластера>
# Операция повторяется для всех дисков с этой ролью.
# Для HDD-дисков
vstorage -c ${CLNAME} make-cs -r /vstorage/<UUID раздела>/data -t <уровень хранения для этого¬
↪диска (tier)>
# Для HDD-дисков с журналом записи (кэш для операций записи)
vstorage -c ${CLNAME} make-cs -r /vstorage/<UUID раздела>/data -j /vstorage/w-cache/<UUID раздела¬
↪диска с ролью ”Кэш”>/<UUID раздела> -s <размер журнала записи CS в МБ> -t <уровень хранения для¬
↪этого диска (tier)>
# Для SSD-дисков
vstorage -c ${CLNAME} make-cs -r /vstorage/<UUID раздела>/data -T ssd -t <уровень хранения для¬
↪этого диска (tier)>
```

Запуск сервисов CS и монтирование кластера хранения

```
systemctl enable --now vstorage-csd.target
export CLNAME=<имя кластера>
mkdir -p /mnt/vstorage
echo '# Точка монтирования кластера' >> /etc/fstab
echo ”vstorage://${CLNAME}    /mnt/vstorage fuse.vstorage rw,nosuid,nodev,_netdev 0 0” >> /etc/
↪fstab
mount -a
```

Проверка состояния кластера хранения

Для проверки состояния кластера хранения можно использовать следующие команды:

- `vstorage -c <имя кластера> top` — обновляемый вывод статуса кластера хранения;
- `vstorage -c <имя кластера> stat` — однократный вывод статуса кластера хранения.

При необходимости можно просмотреть список журналов (см. раздел *Просмотр списка журналов служб фрагментов данных*).

Добавление узла в кластер высокой доступности (High Availability)

```
export CLNAME=<имя кластера>
systemctl enable shaman.target shaman@${CLNAME}.service
systemctl restart shaman.target shaman@${CLNAME}.service
shaman join --cluster ${CLNAME}
shaman set-roles S3
```

Настройка сервиса синхронизации времени

```
nano /etc/chrony.conf
# Перечисляем доступные серверы NTP.
# server 0.ru.pool.ntp.org iburst
systemctl restart chronyd.service
# Проверяем статус синхронизации времени.
timedatectl
               Local time: Вт 2024-02-06 18:17:43 MSK
           Universal time: Вт 2024-02-06 15:17:43 UTC
                 RTC time: Вт 2024-02-06 15:17:43
                Time zone: Europe/Moscow (MSK, +0300)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

Управление параметрами кластера

В этом разделе объясняется, что такое параметры кластера и как их настраивать с помощью утилиты **vstorage**.

Обзор параметров кластера

Параметры кластера определяют то, как кластер хранилища создает и размещает реплики фрагментов данных, а также то, как он ими управляет. Все параметры можно разделить на три основные группы: параметры репликации, параметры помехоустойчивого кодирования, параметры размещения.

В таблице ниже кратко описаны некоторые параметры кластера. Дополнительные сведения о параметрах и их настройке см. в следующих разделах.

Параметр	Описание
Параметры репликации	
Нормальное количество реплик	Количество реплик, создаваемых для фрагмента данных (от 1 до 15). Рекомендуемое значение — 3.
Минимальное количество реплик	Минимальное количество реплик для фрагмента данных (от 1 до 15). Рекомендуемое значение — 2.
Параметры размещения	
Область отказа	Политика размещения реплик (`host` или `disk`). Значение по умолчанию — `host`.
Уровень хранения данных	Уровни хранения данных (от 0 до 3). Значение по умолчанию — 0.

Настройка параметров репликации

Параметры репликации кластера определяют следующее:

- Нормальное количество реплик фрагмента данных. При создании нового фрагмента данных кластер хранилища автоматически реплицирует его до тех пор, пока не будет достигнуто нормальное количество реплик.
- Минимальное количество реплик фрагмента данных (необязательно). В течение жизненного цикла фрагмента данных количество его реплик может меняться. Если большое количество серверов фрагментов данных выходит из строя, оно может упасть ниже заданного минимума. В этом случае все операции записи в затронутые реплики приостанавливаются до тех пор, пока их количество не достигнет минимального значения.

Чтобы просмотреть текущие параметры репликации, примененные к кластеру, вы можете использовать команду `vstorage get-attr`. Например, если ваш кластер смонтирован в каталог `/vstorage/stor1`, вы можете выполнить следующую команду:

```
# vstorage get-attr /vstorage/stor1
connected to MDS#1
File: '/vstorage/stor1'
Attributes:
  <...>
  replicas=1:1
  <...>
```

Как показано в выводе команды, нормальное количество и минимальное количество реплик фрагментов данных равны 1.

Изначально любой кластер настроен так, чтобы иметь только одну реплику на каждый фрагмент данных, что достаточно для оценки функциональности продукта Кибер Хранилище с использованием

только одного сервера. Однако в промышленной среде для обеспечения высокой доступности данных рекомендуется настроить кластер так, чтобы:

- у каждого фрагмента данных было не менее трех реплик,
- минимальное количество реплик было равно двум.

Для конфигурации такого типа необходимо, чтобы в кластере было как минимум три сервера фрагментов данных.

Для того чтобы настроить текущие параметры репликации таким образом, чтобы они применялись сразу ко всем данным в кластере, вы можете запустить команду `vstorage set-attr` для каталога, в который кластер смонтирован. Например, чтобы задать рекомендуемые значения параметров репликации для кластера `stor1`, смонтированного в каталог `/vstorage/stor1`, сделайте нормальное количество реплик равным 3:

```
# vstorage set-attr -R /vstorage/stor1 replicas=3
```

Минимальное количество реплик автоматически будет сделано равным 2.

> **Примечание**
>
> Для получения сведений о том, как рассчитывается минимальное количество реплик, см. документацию утилиты **vstorage-set-attr**.

Наряду с настройкой параметров репликации сразу для всех данных кластера вы также можете настраивать их только для определенных каталогов или файлов, например:

```
# vstorage set-attr -R /vstorage/stor1/private/MyCT replicas=3
```

Настройка параметров помехоустойчивого кодирования

В качестве лучшей альтернативы репликации продукт Кибер Хранилище может использовать помехоустойчивое кодирование для обеспечения избыточности данных. При использовании помехоустойчивого кодирования Кибер Хранилище разбивает входящий поток данных на части определенного размера, затем разбивает каждую часть на определенное количество (M) блоков размером 1 МБ и создает определенное количество (N) блоков четности для обеспечения избыточности. Все блоки распределяются по M+N серверам фрагментов данных, то есть по одному блоку на сервер. В кластере хранилища блоки хранятся в обычных фрагментах данных по 256 МБ, но такие фрагменты данных не реплицируются, поскольку избыточность уже достигнута. Кластер может выдержать отказ любых N серверов фрагментов данных без потери данных.

Значения M и N составляют имена режимов избыточности на основе помехоустойчивого кодирования. Например, в режиме 5+2 входящий поток данных разбивается на части по 5 МБ, каждая часть разделяется на 5 блоков по 1 МБ, добавляются 2 блока четности по 1 МБ для обеспечения избыточности. Кроме того, если N равно 2, то данные кодируются с использованием схемы RAID 6, а если N больше 2, то применяются коды Рида — Соломона.

Рекомендуется использовать следующие режимы избыточности на основе помехоустойчивого кодирования (M+N):

- 1+0,
- 1+2,
- 3+2,
- 5+2,
- 7+2,
- 17+3.

Помехоустойчивое кодирование настраивается для каталогов, например:

```
# vstorage set-attr -R /vstorage/stor1 encoding=5+2
```

После того как помехоустойчивое кодирование включено, режим избыточности нельзя переключить обратно на репликацию. Тем не менее, вы можете переключать режимы помехоустойчивого кодирования для одного и того же каталога.

Общая рекомендация заключается в том, чтобы в кластере всегда было по крайней мере на один сервер больше, чем требуется для выбранного режима избыточности. Например, кластер, использующий репликацию с тремя репликами, должен состоять из четырех серверов, а кластер, использующий помехоустойчивое кодирование 7+2, должен состоять из десяти серверов. Такая конфигурация кластера обладает следующими преимуществами:

- Кластер будет более защищен от нескольких одновременных отказов серверов. В противном случае в кластере с одним неработоспособным сервером возможна потеря данных даже при отказе одного из дисков на каком-либо из работоспособных серверов.
- Вы сможете выполнять обслуживание серверов кластера, которое может потребоваться для восстановления вышедшего из строя сервера или, например, для установки обновлений программного обеспечения.
- В большинстве случаев кластер будет иметь достаточно серверов для самовосстановления. В кластере без запасного сервера на всех серверах кластера размещено по одной реплике фрагмента данных для обеспечения избыточности. Если один или два сервера выйдут из строя,

данные не будут утрачены, но работоспособность кластера деградирует, а кластер начнет самовосстановление только после того, как неисправные серверы будут заменены. Во время выполнения самовосстановления кластер может быть подвержен дополнительным сбоям до тех пор, пока все его серверы не станут работоспособными.

- Вы сможете заменить и обновить сервер кластера, не добавляя новый сервер в кластер. Корректное освобождение сервера возможно только в том случае, если остальные серверы в кластере соответствуют настроенной схеме обеспечения избыточности. При необходимости вы можете принудительно освободить сервер без переноса данных, но это приведет к ухудшению производительности кластера и запуску его самовосстановления.

Конфигурация областей отказа

Область отказа — это набор служб, которые могут отказать взаимосвязанным образом. Вследствие возможности такого рода отказов критически важно распределять реплики данных по разным областям отказа. Примеры областей отказа:

- Отдельный диск (наименьшая возможная область отказа). По этой причине Кибер Хранилище никогда не размещает более одной реплики данных на диск службы фрагментов данных (CS).
- Отдельный сервер, на котором запущено несколько служб фрагментов данных (CS). В случае отказа такого рода сервера (например, из-за отключения электроэнергии или отключения от сети) все службы фрагментов данных одновременно становятся недоступными. По этой причине продукт Кибер Хранилище по умолчанию настроен таким образом, чтобы гарантировать, что на одном сервере никогда не будет храниться более одной реплики данных (см. раздел *Назначение областей отказа* далее).

Полный перечень областей отказа приведен в разделе *Назначение областей отказа*.

Топология области отказа

Каждому служебному компоненту кластера хранилища назначены топологические сведения. Топологические пути, состоящие из автоматически генерируемых идентификаторов `host_ID.cs_ID`, определяют логическое дерево физических расположений компонентов.

- `host_ID` — уникальный, генерируемый случайным образом идентификатор сервера. Этот идентификатор генерируется во время установки и размещается в файле `/etc/vstorage/host_id`.
- `cs_ID` — уникальный идентификатор службы фрагментов данных. Этот идентификатор генерируется при создании службы CS.

Примечание

Чтобы просмотреть сведения о текущей топологии служебных компонентов и доступное дисковое пространство в каждом расположении, запустите команду `vstorage top` и нажмите `w`.

Назначение областей отказа

Основываясь на уровнях иерархии, описанных выше, вы можете использовать команду `vstorage set-attr`, чтобы назначать области отказа для подходящего размещения реплик:

```
# vstorage -c <cluster_name> set-attr -R -p / failure-domain=<disk|host|rack|row|room>
```

Где:

- `disk` означает, что только одна реплика фрагмента данных может быть размещена на диске службы фрагментов данных.
- `host` означает, что только одна реплика фрагмента данных может быть размешена на сервере, на котором запущены службы фрагментов данных. Это значение используется по умолчанию.
- `rack` означает, что только одна реплика фрагмента данных может быть размещена в серверной стойке.
- `row` означает, что только одна реплика фрагмента данных может быть размещена в ряду серверных стоек.
- `room` означает, что только одна реплика фрагмента данных может быть размещена в серверной комнате.

Рекомендуется использовать одинаковую конфигурацию для всех данных кластера, так как это упрощает управление и снижает риск возникновения ошибок.

Рекомендации по использованию областей отказа

Важно

Не используйте область отказа `disk` и не размещайте журналы на дисках SSD одновременно. В этом случае несколько реплик могут быть размещены на дисках служб фрагментов данных, хранящих свои журналы на одном и том же диске SSD. При отказе этого диска SSD все реплики, зависимые от журналов на нем, будут утрачены. Как следствие ваши данные могут быть потеряны.

- Для обеспечения гибкости работы механизмов размещения и повторной балансировки рекомендуется иметь в промышленной среде как минимум пять областей отказа. Зарезервируйте

достаточный объем дискового пространства в каждой области отказа для того, чтобы при сбое некоторой области ее данные могли быть восстановлены в работоспособных областях.

- Рекомендуется использовать как минимум три реплики.
- Если большая область отказывает и становится недоступной, Кибер Хранилище по умолчанию не выполняет восстановление данных, потому что время на репликацию значительного объема данных может быть больше, чем то время, за которое область восстановится. Это поведение управляется посредством глобального параметра `mds.wd.max_offline_cs_hosts` (настраивается с помощью утилиты **`vstorage-config`**), контролирующего, какое количество неработоспособных серверов следует считать обычным отказом и автоматически выполнять восстановление.
- В зависимости от значения глобального параметра `mds.alloc.strict_failure_domain` (настраивается с помощью утилиты **`vstorage-config`**) политика областей отказа может быть либо строгой (используется по умолчанию), либо рекомендательной. Настоятельно рекомендуется не изменять этот параметр.

Использование уровней хранения данных

В этом разделе описаны уровни хранения данных, используемые в продукте Кибер Хранилище, и предоставлены сведения о том, как их настраивать и выполнять их мониторинг.

Об уровнях хранения данных

Уровни хранения данных представляют собой способ организации дискового пространства. Вы можете использовать их для хранения различных категорий данных на разных серверах фрагментов данных. Например, вы можете использовать высокоскоростные твердотельные накопители для хранения данных, для которых критична производительность, а не для кэширования операций кластера.

Настройка уровней хранения данных

Чтобы назначить дисковому пространству уровень хранения данных, сделайте следующее:

1. Назначьте всем серверам фрагментов данных с дисками SSD один и тот же уровень хранения. Вы можете это сделать при установке сервера фрагментов данных.

   > **Примечание**
   >
   > Сведения о рекомендуемых дисках SSD см. в разделе *Использование дисков SSD*.

2. Назначьте часто используемым каталогам и файлам уровень хранения 1 с помощью команды `vstorage set-attr`, например:

```
# vstorage set-attr -R /vstorage/stor1/private/MyCT tier=1
```

Эта команда рекурсивно назначает каталогу `/vstorage/stor1/private/MyCT` и его содержимому уровень хранения 1.

При назначении дисковому пространству уровней хранения принимайте во внимание, что более быстрым дискам следует назначать более высокие уровни хранения. Например, вы можете использовать уровень хранения 0 для резервных копий и других холодных данных (CS без журналов на дисках SSD), уровень хранения 1 для виртуальных машин — большой объем холодных данных с частыми операциями случайной записи (CS с журналами на дисках SSD), уровень хранения 2 для горячих данных (CS с журналами на дисках SSD), журналов, кэшей, отдельных дисков виртуальных машин и т. п.

Эта рекомендация относится к тому, как кластер хранилища работает с дисковым пространством. Если на некотором уровне хранения заканчивается свободное место, кластер хранилища попытается временно использовать дисковое пространство более низкого уровня хранения. Если вы добавите еще дискового пространства для исходного уровня хранения, данные, временно хранящиеся на других уровнях хранения, будут перемещены на тот уровень хранения, где они должны были быть изначально.

Например, если вы пытаетесь записать данные на уровень хранения 2 и он заполнен, кластер хранилища попытается записать эти данные на уровень хранения 1, а затем на уровень хранения 0. Если вы потом добавите еще дискового пространства на уровень хранения 2, упомянутые выше данные, хранящиеся в данный момент на уровнях хранения 1 или 0, будут перемещены обратно на уровень хранения 2, где они должны были храниться изначально.

Автоматическая балансировка данных

Для обеспечения максимальной производительности ввода-вывода сервисов фрагментов данных в кластере хранилища данных выполняется автоматическая балансировка нагрузки на эти сервисы: каждые 60 секунд горячие фрагменты данных перемещаются с горячих сервисов фрагментов данных на более холодные.

Сервис фрагментов данных считается горячим, если средняя длина его очереди запросов превышает среднюю длину очереди запросов во всем кластере на 40 % или больше. Фрагмент данных считается горячим, если к нему часто обращаются.

Насколько сервис фрагментов данных является горячим, можно оценить с помощью команд `vstorage top` и `vstorage stat`. Например:

```
...
 IO QDEPTH: 0.1 aver, 1.0 max; 1 out of 1 hot CS balanced     46 sec ago
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
...
 CSID STATUS     SPACE  AVAIL  REPLICAS   UNIQUE IOWAIT IOLAT(ms) QDEPTH HOST      BUILD_VERSION
 1025 active     1007.3 156.8G     7142        0    10%     1/117    0.3 10.31.240.167 6.0.11-10
 1026 active     1007.3 156.8G     7267        0    11%     0/225    0.1 10.31.240.167 6.0.11-10
 1027 active     1007.3 156.8G     7151        0     2%      0/10    0.1 10.31.240.167 6.0.11-10
 1028 active     1007.3 156.8G     7285        0    13%     1/141    1.0 10.31.240.167 6.0.11-10
...
```

В выводе этих команд колонка `QDEPTH` содержит среднюю длину очереди запросов для каждого сервиса фрагментов данных за последние 5 секунд, а строка `IO QDEPTH` содержит среднюю длину очереди запросов и максимальную длину очереди запросов во всем кластере за последние 60 секунд.

Мониторинг уровней хранения данных

Вы можете просмотреть сведения о дисковом пространстве каждого уровня хранения данных, запустив команду `vstorage -c <cluster_name> top` и нажав клавишу `v`. Типичный вывод может выглядеть следующим образом:

```
Cluster 'tiers': healthy
Space: [OK] allocatable 3.3TB of 3.5TB, 3.5TB total, 3.5TB free
 tier 0: allocatable 1.6TB of 1.7TB, 1.7TB total, 1.7TB free
 tier 1: allocatable 866GB of 912GB, 912GB total, 912GB free
 tier 3: allocatable 866GB of 912GB, 912GB total, 912GB free
MDS nodes: 1 of 1, epoch uptime:  3 min
CS nodes:  4 of 4 (4 avail, 0 inactive, 0 offline)
Replication:  1 norm,  1 limit
Chunks: [OK] 0 healthy,  0 standby,  0 degraded,  0 urgent,
             0 blocked,  0 pending,  0 offline,  0 replicating,
             0 overcommitted,  0 deleting,  0 void
FS:  0B in 1 files, 0 inodes,  0 file maps,  0 chunks,  0 chunk
IO:       read     0B/s (  0ops/s), write     0B/s (  0ops/s)
IO total: read       0B (    0ops), write       0B (    0ops)
Repl IO:  read     0B/s, write:     0B/s
Sync rate:   0ops/s, datasync rate:   0ops/s
```

Изменение сети кластера хранилища

Перед переносом вашего кластера хранилища в новую сеть примите во внимание следующее:

- Изменение сети кластера приводит к кратковременному простою в течение периода, когда более половины серверов MDS недоступны.
- Настоятельно рекомендуется создать резервную копию всех репозиториев служб MDS перед изменением сети кластера.

Чтобы изменить сеть кластера, выполните следующие действия на каждом сервере кластера, на котором запущена служба MDS:

1. Остановите службу MDS:

```
# systemctl stop vstorage-mdsd.target
```

2. Укажите новые IP-адреса для всех серверов метаданных в кластере с помощью команды `vstorage configure-mds -r <MDS_repo> -n <MDS_ID@new_IP_address>[:port] [-n ...]`, где:

   - `<MDS_repo>` — путь к репозиторию службы MDS на текущем сервере;
   - каждая пара `<MDS_ID@new_IP_address>` содержит идентификатор службы MDS и ее новый IP-адрес.

   Например, для кластера, включающего в себя 5 серверов метаданных, запустите:

```
# vstorage -c stor1 configure-mds -r /vstorage/stor1-cs1/mds/data -n 1@10.10.20.1 \
-n 2@10.10.20.2 -n 3@10.10.20.3 -n 4@10.10.20.4 -n 5@10.10.20.5
```

   Обратите внимание на следующее:

   - Вы можете узнать идентификатор текущего сервиса MDS и путь к его репозиторию с помощью команды `vstorage list-services -M`.
   - Если вы не укажете порт, по умолчанию будет использован порт 2510.

3. Запустите службу MDS:

```
# systemctl start vstorage-mdsd.target
```

Включение и выключение серверов кластера хранилища

Чтобы полностью выключить кластер хранилища, сделайте следующее:

1. Остановите службы `shaman.target` и `shaman@<cluster_name>.service` и выключите их автоматический запуск, например:

```
# systemctl stop shaman.target shaman@stor1.service
# systemctl disable shaman.target shaman@stor1.service
```

2. Выключите всех клиентов кластера. Для этого выполните следующее на каждом компьютере-клиенте:

   1. С помощью утилиты **`lsof`** убедитесь, что файлы, размещенные в кластере, не являются

открытыми. Например, для кластера `stor1` выполните:

```
# lsof +D /vstorage/stor1
```

2. Отсоедините (размонтируйте) файловую систему кластера, использовав команду `umount`. Например, если кластер смонтирован в каталог `/vstorage/stor1` на компьютере-клиенте, вы можете отсоединить его следующим образом:

```
# umount /vstorage/stor1
```

3. Выключите автоматическое монтирование кластера, удалив соответствующую строку из файла `/etc/fstab`.

3. Остановите службы метаданных и службы фрагментов данных и выключите их автоматический запуск:

```
# systemctl stop vstorage-csd.target vstorage-mdsd.target
# systemctl disable vstorage-csd.target vstorage-mdsd.target
```

4. Выключите серверы кластера.

Для включения кластера хранилища сделайте следующее:

1. Включите серверы кластера.
2. Запустите службы метаданных и службы фрагментов данных и включите их автоматический запуск:

```
# systemctl start vstorage-mdsd.target vstorage-csd.target
# systemctl enable vstorage-mdsd.target vstorage-csd.target
```

3. Включите автоматическое монтирование кластера, добавив соответствующую строку в файл `/etc/fstab` и смонтировав файловую систему кластера с помощью команды `mount`.
4. Запустите службы `shaman.target` и `shaman@<cluster_name>.service` и включите их автоматический запуск, например:

```
# systemctl start shaman.target shaman@stor1.service
# systemctl enable shaman.target shaman@stor1.service
```

Удаление серверов фрагментов данных

Если вам необходимо удалить сервер фрагментов данных, например, чтобы выполнить на нем некоторые задачи обслуживания, сделайте следующее:

1. Удостоверьтесь, что после удаления сервера будут выполнены эти условия:
   - В кластере есть достаточное количество серверов фрагментов данных, чтобы обеспечить хранение требуемого числа реплик фрагментов данных (т. е. количество серверов должно быть не меньше, чем текущее значение параметра репликации).
   - Серверы фрагментов данных должны обладать достаточным объемом дискового пространства, чтобы обеспечить хранение фрагментов данных. Инструкции о том, как получить сведения о дисковом пространстве, см. в разделе *Мониторинг серверов фрагментов данных*.
2. Узнайте порядковый номер сервера фрагментов данных, который вы хотите удалить, выполнив следующую команду на одном из серверов кластера:

   ```
   # vstorage -c stor1 top
   ```

   Будут выведены подробные сведения о кластере `stor1`. Найдите секцию с информацией о серверах фрагментов данных, например:

   ```
   ...
    CSID  STATUS     SPACE   FREE REPLICAS IOWAIT IOLAT(ms) QDEPTH HOST
    1025  active     105GB   88GB       40     0%       0/0    0.0 10.30.17.38
    1026  active     105GB   88GB       40     0%       0/0    0.0 10.30.18.40
    1027  active     105GB   99GB       40     0%       0/0    0.0 10.30.21.30
    1028  active     107GB  101GB       40     0%       0/0    0.0 10.30.16.38
   ...
   ```

   В колонке **CSID** отображаются порядковые номера серверов фрагментов данных. В выводе выше показано, что четыре сервера фрагментов данных настроены для кластера `stor1`. Их порядковые номера — 1025, 1026, 1027 и 1028.
3. Удалите сервер фрагментов данных из кластера. Например, чтобы удалить сервер фрагментов данных с порядковым номером 1028 из кластера `stor1`, выполните следующую команду:

   ```
   # vstorage -c stor1 rm-cs --wait 1028
   ```

При запуске команды без параметра `--wait` произойдет немедленный возврат в командную оболочку, а в кластере в фоновом режиме будут выполнены репликация фрагментов данных с удаляемого сервера

на другие серверы фрагментов данных и удаление целевого сервера. При указании этого параметра возврат в командную оболочку не произойдет до тех пор, пока не будут завершены репликация фрагментов данных и удаление целевого сервера (репликация может занять значительное время).

Обратите внимание на следующее:

- Если диск службы CS станет недоступным для ОС и служба CS не будет обладать доступом к своему хранилищу, фрагменты данных, которые необходимо реплицировать, будут недоступны, а удаление сервера фрагментов данных не будет выполнено. В этом случае вам будет нужно принудительно удалить сервер фрагментов данных из кластера с помощью параметра `-f`.

  > **⚠ Предупреждение**
  >
  > Делайте так только в том случае, если сервер фрагментов данных безвозвратно утрачен. Никогда не удаляйте работоспособный сервер фрагментов данных с использованием параметра `-f`.

- Запущенную операцию удаления можно отменить с помощью команды `vstorage -c stor1 rm-cs --cancel 1028`. Это может пригодиться, например, если вы указали неверный порядковый номер сервера фрагментов данных для удаления.

Чтобы вернуть удаленный сервер фрагментов данных обратно в кластер, установите его повторно, как описано в разделе *Последующие серверы*.

Экспорт данных кластера хранилища

Обычный способ использования дискового пространства кластера хранилища — хранение данных виртуальных машин, обращения к которым происходят на серверах, где установлены клиентские компоненты и смонтирован кластер.

Другой способ доступа к данным кластера хранилища — это экспортировать эти данные с помощью протокола S3. Подробнее см. далее в этом разделе.

Доступ к кластеру хранилища через S3

Продукт Кибер Хранилище может предоставлять доступ к данным через API, совместимый с Amazon S3, что позволяет поставщикам услуг:

- запускать службы на основе S3 в их собственных кластерах хранилища,
- продавать клиентам хранилище S3 как услугу.

Поддержка S3 расширяет возможности продукта Кибер Хранилище и требует наличия работающего кластера хранилища.

О хранилище объектов

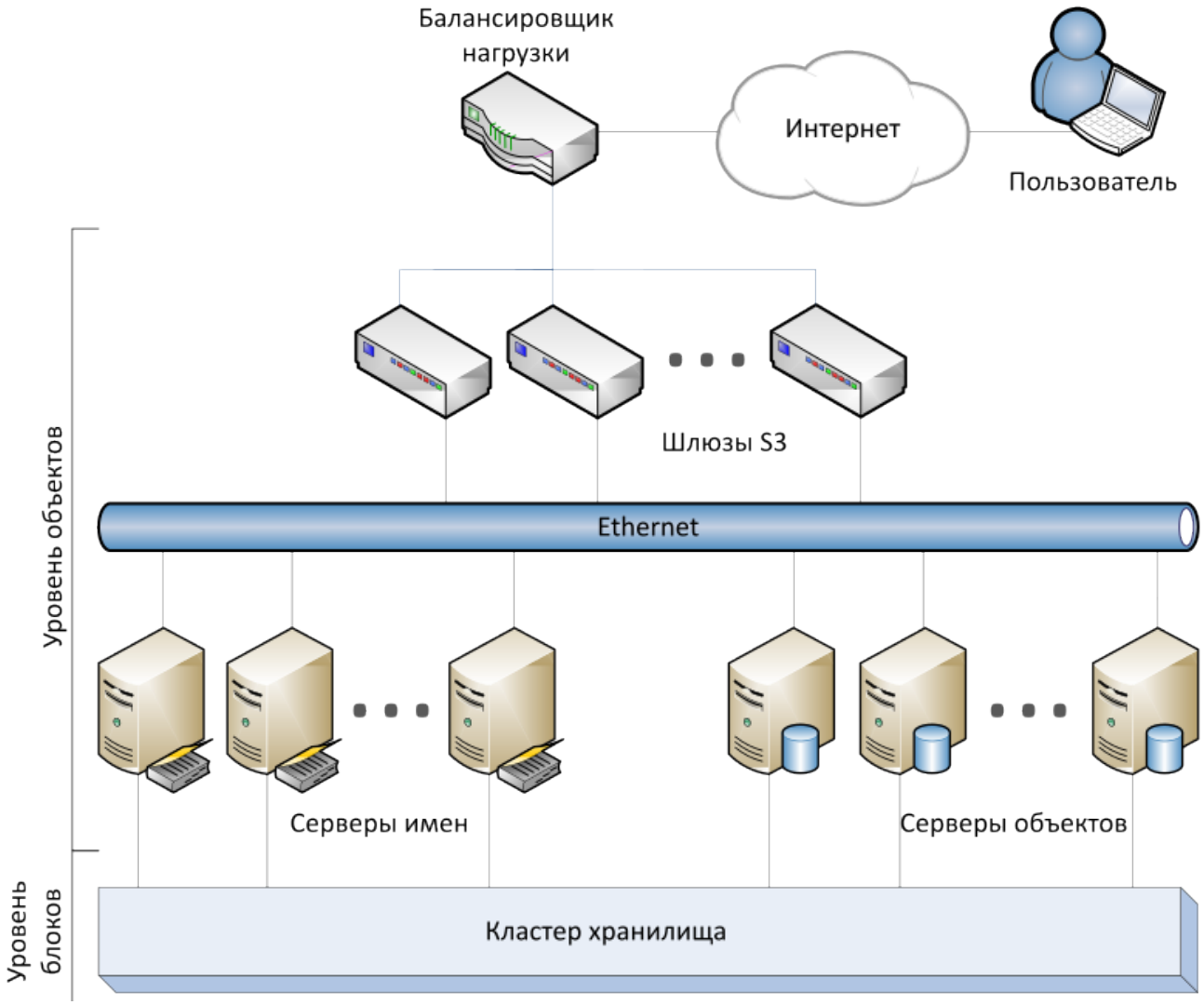
Хранилище объектов представляет собой архитектуру хранения данных, которая позволяет управлять данными в виде объектов (как в хранилище данных типа ”ключ-значение”) в противоположность файлам в файловых системах или блокам в блочном хранилище. За исключением данных, каждый объект обладает именем (т. е. полным путем к объекту), которое его описывает, а также уникальным идентификатором, позволяющим найти объект в хранилище. Хранилище объектных данных оптимизировано для хранения миллиардов объектов, в частности для хранения данных приложений, хостинга статического веб-контента, онлайн-сервисов хранения данных, ”больших данных” и резервных копий. Все эти варианты использования доступны с помощью хранилища объектов благодаря сочетанию очень высокой масштабируемости с доступностью и согласованностью данных.

По сравнению с другими типами хранилищ ключевое отличие хранилища объектов в том, что части объекта нельзя изменить, поэтому при изменении объекта вместо этого формируется его новая версия. Такой подход крайне важен для поддержания доступности и согласованности данных. Прежде всего, изменение объекта как единого целого устраняет проблему конфликтов. То есть объект с самой недавней меткой времени считается текущей версией и является таковой. В результате объекты всегда согласованы, то есть их состояние является актуальным и соответствующим.

Другим отличием хранилища объектов является согласованность в конечном счете. Согласованность в конечном счете не гарантирует, что операции чтения будут возвращать новое состояние во всех центрах обработки данных (ЦОД) после выполнения записи в одном из них. Читатели могут наблюдать старое состояние в одном из ЦОД в течение неопределенного периода времени, пока запись не будет распространена на все ЦОДы. Это очень важно для обеспечения доступности хранилища, так как географически удаленные ЦОДы могут не иметь возможности выполнять обновление данных синхронно (например, из-за сетевых проблем), а само обновление также может быть медленным из-за того, что ожидание подтверждений от всех реплик данных на больших расстояниях может занимать сотни миллисекунд. Поэтому согласованность в конечном счете помогает скрывать задержки связи при операциях записи ценой вероятного старого состояния, наблюдаемого читателями. Однако во многих случаях это вполне допустимо.

Инфраструктура хранилища объектов

Инфраструктура хранилища объектов состоит из следующих элементов: серверы объектов, серверы имен, шлюзы S3 и внутреннее хранилище блочного уровня.



- Сервер объектов (OS) хранит фактические данные объектов (содержимое), полученные от шлюза S3. Данные хранятся в блочном хранилище со встроенной высокой доступностью.
- Сервер имен хранит метаданные объектов, полученные от шлюза S3. Метаданные включают в себя имя объекта, размер, ACL (список управления доступом), расположение, имя владельца и т. д. Сервер имен (NS) также хранит данные в блочном хранилище со встроенной высокой доступностью.
- Шлюз S3 (GW) — это прокси-сервер между службами хранилища объектов и конечными пользователями. Он получает и обрабатывает запросы протокола Amazon S3, а также использует веб-сервер `nginx` для внешних подключений. Шлюз S3 отвечает за аутентификацию пользователей S3 и проверку списков управления доступом. Шлюз S3 не хранит никаких данных (то есть он не имеет состояний).
- Внутреннее хранилище блочного уровня — это блочное хранилище с высокой доступностью служб и данных.

Обзор хранилища объектов

С точки зрения хранилища объектов S3, файл — это объект. Серверы хранилища объектов хранят каждый объект, загруженный через API S3, как пару сущностей:

- Имена объектов и связанные с объектами метаданные (хранение обеспечивают службы NS). Имя объекта в хранилище определяется на основе параметров запроса и свойств корзины следующим образом:
  - Если для корзины выключено управление версиями, имя объекта в хранилище содержит имя корзины и имя объекта, взятые из запроса S3.
  - Если для корзины включено управление версиями, имя объекта также содержит список версий объекта.
- Данные объекта (хранение обеспечивают службы OS). Часть имени объекта, относящаяся к каталогам, определяет службу NS для его хранения, а полное имя объекта определяет службу OS для хранения данных объекта.

Взаимодействие между хранилищем объектов S3 и кластером хранилища

Для каждого из серверов кластера хранилища объектов S3 необходимо наличие работающего кластера хранилища. Кластер хранилища обеспечивает совместное использование данных, высокую согласованность данных, достаточную производительность случайных операций ввода-вывода и высокую доступность служб хранения. С точки зрения кластера хранилища, данные S3 представляют собой набор файлов (см. раздел *Сервер объектов*), которые никак не интерпретируются на уровне файловой системы кластера хранилища.

Передача по частям

Имя передачи по частям (multipart upload) задается с помощью схемы, аналогичной схеме для имен объектов, но соответствующий ей объект содержит таблицу вместо содержимого файла. Таблица содержит индексные номера частей и их смещения в файле. Это позволяет загружать части файла параллельно (рекомендуется для больших файлов). Максимальное количество частей — 10 000.

Компоненты хранилища объектов

В этом разделе вы познакомитесь с компонентами хранилища S3, такими как шлюзы, серверы объектов и серверы имен, а также узнаете об инструментах управления и служебных корзинах.

Шлюз S3

Шлюз S3 выполняет следующие функции:

- Получает запросы S3 от веб-сервера (через nginx и FastCGI).
- Анализирует пакеты S3 и проверяет запросы S3 (проверяет поля запроса и XML-данные из тела запроса).
- Выполняет аутентификацию пользователей S3.
- Проверяет права доступа к корзинам и объектам, используя списки управления доступом (ACL).
- Собирает статистику о количестве различных запросов и объеме переданных и полученных данных.
- Определяет пути к серверам имен (NS) и серверам объектов (OS), хранящим метаданные и данные объекта.
- Запрашивает имена и связанные метаданные у серверов имен (NS).
- Получает ссылки на объекты, хранящиеся на серверах объектов (OS), запрашивая имена у серверов имен (NS).
- Кэширует метаданные и списки управления доступом объектов S3, полученные от серверов имен (NS), а также данные, необходимые для аутентификации пользователей, также хранящиеся на серверах имен (NS).
- Выступает в качестве прокси-сервера, когда клиенты записывают и считывают данные объектов на серверы объектов (OS) и с них. Во время операций чтения и записи передаются только запрошенные данные. Например, если пользователь запрашивает чтение 10 МБ из объекта размером 1 ТБ, из сервера объектов (OS) будут прочитаны только указанные 10 МБ.

Шлюз S3 состоит из синтаксического анализатора входящих запросов, зависимых от типа асинхронных

обработчиков этих запросов и асинхронного обработчика прерванных запросов, требующих завершения (сложных операций, таких как создание или удаление корзины). Шлюз S3 не хранит данные о своем состоянии в долговременной памяти. Вместо этого он хранит все данные, необходимые для хранилища объектов S3, в самом объектном хранилище (то есть на серверах имен и серверах объектов).

Сервер имен

Сервер имен выполняет следующие функции:

- хранение имен и метаданных объектов;
- предоставление API для вставки, удаления, перечисления имен объектов, а также для изменения метаданных объектов.

Сервер имен состоит из данных (т. е. метаданных объектов), журнала изменений объектов, асинхронного сборщика мусора и асинхронных обработчиков входящих запросов от различных системных компонентов.

Данные хранятся в виде B-дерева, где имени объекта соответствуют метаданные этого объекта. Метаданные объекта состоят их трех частей: сведения об объекте, определяемые пользователем заголовки (необязательно) и список управления доступом (ACL) объекта. Файлы хранятся в соответствующем каталоге кластера хранилища.

Сервер имен отвечает за часть пространства имен кластера хранилища объектов. Каждый экземпляр службы NS является процессом в пространстве пользователя, который работает параллельно с другими процессами и может использовать вплоть до одного ядра ЦП. Оптимальное количество служб NS в расчете на один сервер — это 4-10 служб. При создании кластера хранилища объектов рекомендуется начать с создания 10 экземпляров на сервер, чтобы упростить масштабирование в будущем. Если на вашем сервере есть ядра ЦП, которые не используются другими службами хранения, вы можете создать дополнительные службы NS, чтобы использовать эти ядра ЦП.

Сервер объектов

Сервер объектов выполняет следующие функции:

- хранение данных объектов в пулах (контейнерах данных);
- предоставление API для создания, чтения (включая частичное чтение), записи и удаления объектов.

Сервер объектов состоит из следующего:

- сведения о блоках объекта, хранящихся на данном сервере;
- контейнеры, в которых хранятся данные объекта;

- асинхронный сборщик мусора, освобождающий секции контейнеров после операций удаления объектов.

Блоки данных объектов хранятся в пулах. В хранилище используется 12 пулов с блоками размером, равным степеням 2 (от 4 КБ до 8 МБ). Пул — это обычный файл в блочном хранилище, состоящий из блоков (областей) фиксированного размера. Другими словами, каждый пул — это очень большой файл, предназначенный для хранения объектов определенного размера: первый пул — для объектов размером 4 КБ, второй — для объектов размером 8 КБ и т. д.

Каждый пул состоит из блока с системной информацией и областей данных фиксированного размера. У каждой области есть битовая маска ”свободно”/”грязно”. Данные области хранятся в одном и том же файле, что и B-дерево объекта. Это обеспечивает атомарность при выделении и отмене выделения блока. Каждый блок в области содержит заголовок и данные объекта. В заголовке хранится идентификатор объекта, которому принадлежат данные. Идентификатор необходим для алгоритма дефрагментации на уровне пула, который не имеет доступа к B-дереву объекта. Пул для хранения объекта выбирается в зависимости от размера объекта.

Например, объект размером 30 КБ будет помещен в пул для объектов размером 32 КБ и займет один блок размером 32 КБ. Объект размером 129 КБ будет разделен на одну часть размером 128 КБ и одну часть размером 1 КБ. Первая будет помещена в пул для объектов размером 128 КБ, а вторая — в пул для объектов размером 4 КБ. Накладные расходы могут показаться значительными в случае небольших объектов, поскольку даже объект размером 1 байт займет блок размером 4 КБ. Кроме того, на сервере имен (NS) будет храниться около 4 КБ метаданных в расчете на один объект. Однако такой подход позволяет достичь максимальной производительности, устранить фрагментацию свободного пространства и обеспечить гарантированную производительность вставки объектов. Более того, чем больше объект, тем меньше заметны накладные расходы. Наконец, при удалении объекта его блок пула помечается свободным и может быть использован для хранения новых объектов.

Многокомпонентные объекты хранятся в виде частей (каждая часть сама по себе является объектом), которые могут храниться на разных серверах объектов.

Инструменты управления

Хранилище объектов снабжено следующими инструментами:

- Утилита `ostor-ctl` предназначена для настройки компонентов хранилища.
- Утилита `ostor-s3-admin` предназначена для управления пользователями. Она позволяет создавать, редактировать и удалять пользователей S3, а также управлять ключами доступа пользователей (т. е. создавать и удалять ключи доступа — пары, состоящие из идентификаторов ключей доступа и секретных ключей доступа).

Служебная корзина

Служебная корзина хранит служебные и временные данные, необходимые для хранилища объектов. Эта корзина доступна только администратору S3, в то время как системному администратору потребуются ключи доступа, созданные с помощью утилиты `ostor-s3-admin`.

Обмен данными

В хранилище объектов продукта Кибер Хранилище у каждой службы есть 64-битный уникальный идентификатор. В то же время у каждого объекта есть уникальное имя. Часть имени объекта, относящаяся к каталогам, определяет сервер имен для хранения метаданных объекта, а полное имя объекта — сервер объектов для хранения данных объекта. Списки серверов имен и объектов хранятся в кластере хранилища в каталоге, предназначенном для данных хранилища объектов, и доступны всем, кто обладает доступом к кластеру хранилища. Этот каталог содержит подкаталоги, которые соответствуют службам, размещенным на серверах имен и объектов. Имена подкаталогов соответствуют шестнадцатеричным представлениям идентификаторов служб. В подкаталоге каждой службы есть файл с идентификатором сервера, на котором эта служба запущена. Таким образом, с помощью шлюза системный компонент, обладающий доступом к кластеру хранилища, может получить идентификатор службы, определить ее сервер и отправить ей запрос.

Шлюз S3 обеспечивает обмен данными между следующими компонентами:

- Клиенты (через веб-сервер). Шлюз получает запросы S3 от пользователей и отвечает на них.
- Серверы имен. Шлюз создает, удаляет, изменяет имена, соответствующие корзинам или объектам, проверяет их существование и запрашивает наборы имен списков корзин.
- Серверы объектов. Шлюз отправляет серверам имен и объектов запросы на изменение данных.

Кэширование данных

Чтобы обеспечить эффективное использование данных в хранилище объектов, все шлюзы, серверы имен и серверы объектов кэшируют данные, которые они хранят. Серверы имен и серверы объектов кэшируют B-деревья.

Шлюзы хранят и кэшируют следующие данные, полученные от служб имен (NS):

- Списки пар, состоящих из идентификаторов пользователей и их адресов электронной почты.
- Данные, необходимые для аутентификации пользователей: идентификаторы ключей доступа и секретные ключи доступа. Более подробную информацию об их семантике см. в документации Amazon S3.
- Метаданные и списки управления доступом (ACL) корзин. Метаданные содержат их Unix-время, текущий идентификатор версии и передачи службе NS для проверки их актуальности.

Операции над объектами

Этот раздел знакомит вас с операциями, которые обрабатывает хранилище объектов: запросы операций; операции создания, чтения, удаления.

Запросы операций

Чтобы создать, удалить, прочитать объект или изменить его данные, объектное хранилище S3 должно сначала запросить одну из этих операций, а затем выполнить ее. Общий процесс запроса и выполнения операции состоит из следующего:

1. Запрос учетных данных пользователя. Они будут храниться на сервере имен в определенном формате (см. раздел *Служебная корзина*). Для получения данных (идентификатора, адреса электронной почты, ключей доступа) на соответствующий сервер имен отправляется запрос с кодом операции поиска.
2. Аутентификация пользователя.
3. Запрос метаданных корзины и объекта. Для их получения на сервер имен, хранящий имена объектов и корзин, отправляется еще один запрос с кодом операции поиска.
4. Проверка прав доступа пользователя к корзинам и объектам.
5. Выполнение запрошенной операции с объектом: создание, изменение данных, чтение данных или удаление.

Операция создания

Для создания объекта шлюз отправляет следующие запросы:

1. Запрос с кодом операции охраны к серверу имен. Он создает охранника с таймером, который проверит через фиксированный промежуток времени, действительно ли объект с данными был создан. Если это не так, операция создания завершится ошибкой, и охранник запросит сервер объектов удалить данные объекта, если некоторая их часть была записана. После этого охранник удаляется.
2. Запрос с кодом операции создания к серверу объектов с последующими сообщениями фиксированного размера, содержащими данные объекта. Последнее сообщение содержит флаг окончания данных.
3. Еще один запрос с кодом операции создания к серверу имен. Сервер имен проверяет, существует ли соответствующий охранник, и, если его нет, операция завершается ошибкой. В противном случае сервер создает имя и отправляет шлюзу подтверждение об успешном создании.

Операция чтения

Для исполнения запроса на чтение шлюз определяет идентификатор соответствующего сервера имен на основе имени каталога объекта и идентификатор соответствующего сервера объектов на основе полного имени объекта. Для выполнения операции чтения шлюз отправляет следующие запросы:

1. Запрос с кодом операции чтения на соответствующий сервер имен. Ответ на него содержит ссылку на объект.
2. Запрос к соответствующему серверу объектов с кодом операции чтения и ссылкой на объект, полученной от сервера имен.

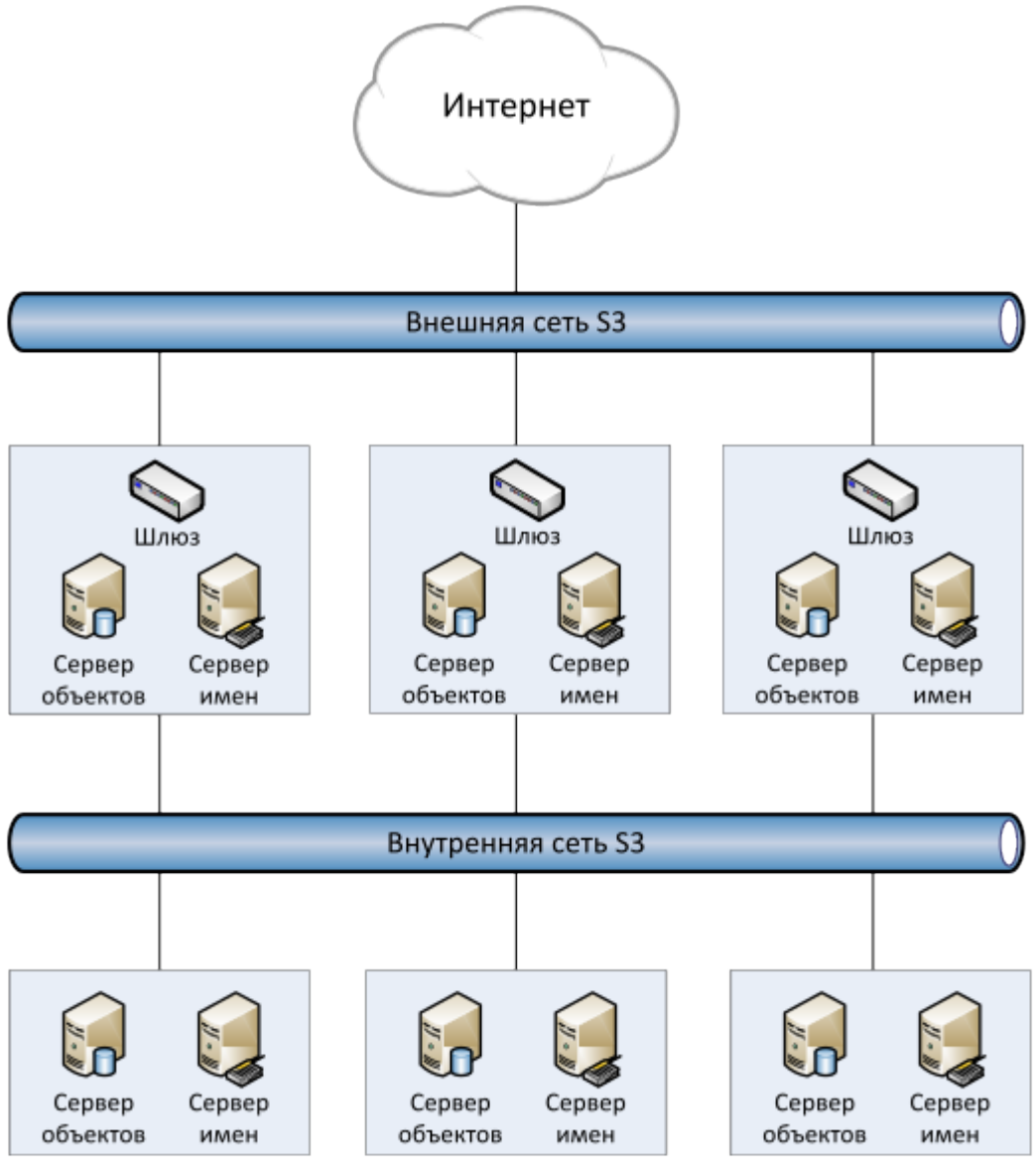
Для исполнения запроса сервер объектов передает шлюзу сообщения фиксированного размера с данными объекта. Последнее сообщение содержит флаг конца данных.

Операция удаления

Чтобы удалить объект (и его имя) из хранилища объектов, шлюз определяет идентификатор соответствующего сервера имен на основе имени каталога объекта и отправляет на этот сервер запрос с кодом операции удаления. В свою очередь, сервер имен удаляет имя из своих данных и отправляет ответ. Через некоторое время сборщик мусора удаляет соответствующий объект из хранилища объектов.

Развертывание хранилища объектов

Этот раздел описывает процесс развертывания хранилища объектов поверх существующего кластера хранилища. В результате вы создадите систему, подобную той, что изображена на рисунке. Обратите внимание, что необязательно использовать все серверы кластера хранилища для служб хранилища объектов. Выбор должен основываться на рабочей нагрузке и конфигурации оборудования.



Для установки служб хранилища объектов, сделайте следующее:

1. Спланируйте сеть S3. Как и кластеру хранилища, кластеру хранилища объектов нужно две сети:

    - Внутренняя сеть, в которой будут взаимодействовать серверы имен (NS), серверы объектов (OS) и шлюзы (GW). Эти службы будут генерировать трафик, аналогичный по объему общему (входящему и исходящему) пользовательскому трафику S3. Если объем трафика служб будет небольшим, то имеет смысл использовать одну и ту же внутреннюю сеть как для хранилища объектов, так и для основного хранилища. Однако если вы ожидаете, что трафик хранилища объектов будет конкурировать с трафиком основного хранилища, имеет смысл сделать так, чтобы трафик S3 проходил через сеть для пользовательских данных (т. е. сеть центра обработки данных). Как только вы выберете сеть для трафика S3, вы определите, какие IP-адреса можно использовать при добавлении серверов кластера.
    - Внешняя (открытая) сеть, через которую конечные пользователи будут осуществлять доступ к

хранилищу объектов. Стандартные порты HTTP и HTTPS должны быть открыты в этой сети.

Кластер хранилища объектов почти полностью независим от базового блочного хранилища. Серверы объектов и имен хранят свои данные в кластере хранилища. Поэтому службы OS и NS зависят от клиента кластера хранилища (vstorage-mount) и могут работать только при смонтированном кластере хранилища. В отличие от них, шлюз является службой без состояний, у которой нет данных. Поэтому он не зависит от клиента кластера хранилища и теоретически может быть запущен даже на тех серверах, где кластер хранилища не смонтирован. Однако для простоты мы рекомендуем создавать шлюзы на серверах со службами OS и NS.

Серверы объектов и имен также используют стандартные средства обеспечения высокой доступности (т. е. службу shaman). Службы OS и NS подписаны на события отказоустойчивого кластера. Однако компоненты кластера хранилища объектов не могут управляться (т. е. отслеживаться и перемещаться между серверами) с помощью shaman. Вместо этого за управление компонентами кластера хранилища объектов отвечает служба конфигурации S3, которая подписана на события отказоустойчивого кластера и получает от shaman уведомления о том, что серверы работоспособны и могут запускать службы. По этой причине компоненты кластера хранилища объектов не отображаются в выводе команды `shaman top`.

Службы шлюзов, которые не имеют состояний, никогда не перемещаются, и их высокая доступность не управляется кластером хранилища. Вместо этого при необходимости создается новая служба шлюза.

2. Убедитесь, что каждый сервер, на котором будут запущены службы OS и NS, добавлен в отказоустойчивый кластер. Вы можете добавить серверы в отказоустойчивый кластер с помощью команды `shaman join`.

3. Установите пакет `vstorage-ostor` на каждом сервере кластера хранилища объектов.

```
# yum install vstorage-ostor
```

4. Создайте кластерную конфигурацию на одном из серверов кластера хранилища объектов, где будут запущены службы хранилища объектов. Рекомендуется создать 10 служб NS и 10 служб OS на каждом сервере. Например, если вы собираетесь использовать пять серверов, вам нужно будет 50 служб NS и 50 служб OS. Выполните следующую команду на первом сервере кластера хранилища объектов:

```
# ostor-ctl create -r /var/lib/ostor/configuration -n <IP_addr>
```

В этой команде `<IP_addr>` — это IP-адрес сервера (принадлежит диапазону адресов внутренней сети S3), который будет использоваться службой конфигурации.

Вам будет предложено ввести и подтвердить пароль для нового кластера хранилища объектов (он

может быть таким же, как пароль кластера хранилища). Этот пароль понадобится для добавления новых серверов.

Служба конфигурации будет хранить кластерную конфигурацию локально в каталоге `/var/lib/ostor/configuration`. Кроме того, `<IP_addr>` будет сохранен в файле `/<storage_mount>/<ostor_dir>/control/name` (`<ostor_dir>` — это каталог в кластере хранилища с файлами служб хранилища объектов). Если первая служба конфигурации откажет, а команда `ostor-ctl get-config` перестанет работать, в файле `/<storage_mount>/<ostor_dir>/control/name` замените этот IP-адрес на IP-адрес сервера с работоспособной службой конфигурации (создается на шаге 6).

5. Запустите службу конфигурации.

```
# systemctl start ostor-cfgd.service
# systemctl enable ostor-cfgd.service
```

6. Добавьте еще как минимум две службы конфигурации для обеспечения избыточности (всего их должно быть не менее трех). Служба конфигурации нужна только для добавления серверов в кластер хранилища объектов и их удаления из него и не влияет на работу служб хранилища объектов и их высокую доступность. Поэтому отказ службы конфигурации не является критическим для кластера хранилища объектов. Тем не менее, это все равно нежелательно и мы рекомендуем создать несколько служб конфигурации, чтобы хотя бы одна из них всегда была в рабочем состоянии.

   Чтобы добавить еще одну службу конфигурации, выполните следующие команды на сервере, где будут запущены службы хранилища объектов. Повторите, чтобы создать необходимое количество служб конфигурации.

```
# ostor-ctl join -n <remote_IP_addr> -a <local_IP_addr>
# systemctl start ostor-cfgd.service
# systemctl enable ostor-cfgd.service
```

   В этих командах `<remote_IP_addr>` — это `<IP_addr>` из шага 4.

   Каждая добавленная служба конфигурации будет хранить кластерную конфигурацию локально в каталоге `/var/lib/ostor/configuration`.

7. Инициализируйте новое хранилище объектов на первом сервере. каталог `<ostor_dir>` будет создан в корневом каталоге вашего кластера хранилища.

```
# ostor-ctl init-storage -n <IP_addr> -s <cluster_mount_point>
```

   Вам нужно будет предоставить IP-адрес и пароль хранилища объектов, указанные на шаге 4.

8. Добавьте в DNS публичные IP-адреса серверов, на которых будут работать службы шлюзов (GW). Вы можете настроить DNS, чтобы обеспечить доступ к хранилищу объектов по DNS-имени, а также чтобы оконечная точка S3 получала запросы REST API в стиле виртуального хостинга с URI вида http://bucketname.s3.example.com/objectname.

   После конфигурации DNS удостоверьтесь, что преобразование DNS-имени вашей оконечной точки S3 работает на компьютерах-клиентах.

   > **Примечание**
   >
   > Только к корзинам с DNS-совместимыми именами можно обращаться с помощью запросов в стиле виртуального хостинга. Подробные сведения см. в разделе *Политики именования корзин и ключей объектов*.

   Ниже приведен пример файла конфигурации DNS-зоны для DNS-сервера BIND.

```
;$Id$
$TTL 1h  @  IN   SOA    ns.example.com. s3.example.com. (
                        2013052112      ; serial
                        1h      ; refresh
                        30m     ; retry
                        7d      ; expiration
                        1h )    ; minimum
               NS       ns.example.com.
$ORIGIN s3.example.com
h1 IN   A 10.29.1.95
        A 10.29.0.142
        A 10.29.0.137
* IN    CNAME    @
```

   Эта конфигурация предписывает DNS перенаправлять все запросы с URI http://s3.example.com и его поддоменов (http://*.s3.example.com/*) на одну из оконечных точек, перечисленных в записи ресурса `h1` (10.29.1.95, 10.29.0.142 или 10.29.0.137), циклическим (round-robin) способом.

9. Добавьте серверы, где будут запущены службы хранилища объектов, в кластерную конфигурацию.

   > **Примечание**
   >
   > Добавление серверов в существующие кластеры осуществляется аналогичным образом (см. шаги 8-12).

Чтобы это сделать, запустите команду `ostor-ctl add-host` на каждом таком сервере:

```
# ostor-ctl add-host -r <cluster_mount_point>/<ostor_dir> --hostname <name> --roles OBJ
```

Вам нужно будет предоставить пароль, указанный на шаге 4.

> **Примечание**
>
> Если вы хотите, чтобы служба агента хранилища объектов использовала внутренний IP-адрес, добавьте параметр `-H <internal_IP_address>` в команду выше.

10. Создайте том S3 и необходимое количество служб NS и OS:

```
# ostor-ctl add-vol --type OBJ -s <cluster_mount_point> --os-count <OS_num> \
--ns-count <NS_num> --acc-count 1 --vstorage-attr ”failure-domain=host tier=0 replicas=3”
```

В этой команде:

- `<NS_num>` и `<OS_num>` задают количества служб NS и OS.
- Параметры `failure-domain=host`, `tier=0`, `replicas=3` задают область отказа, уровень хранения данных и режим избыточности для тома (подробности см. в разделе *Обзор параметров кластера*).

Эта команда выведет идентификатор созданного тома. Вам он понадобится на следующем шаге.

11. Создайте экземпляры службы шлюза на выбранных серверах, у которых есть доступ к Интернету и внешние IP-адреса. Рекомендуется создавать 4 службы шлюза на одном сервере.

> **Примечание**
>
> В целях безопасности убедитесь, что только `nginx` может получить доступ к внешней сети и что службы шлюза используют только внутренние IP-адреса.

```
# ostor-ctl add-s3gw -a <internal_IP_address>:<port> -V <volume_ID>
```

В этой команде:

- `<internal_IP_address>` — внутренний IP-адрес сервера со службами шлюза.
- `<port>` (обязательный) — незанятый порт, уникальный для каждого экземпляра службы шлюза (GW) на этом сервере.

- `<volume_ID>` — идентификатор тома, который вы создали на предыдущем шаге (его также можно получить с помощью команды `ostor-ctl get-config`).

Например:

```
# ostor-ctl add-s3gw -a 127.0.0.1:9001 -V 0100000000000001
# ostor-ctl add-s3gw -a 127.0.0.1:9002 -V 0100000000000001
# ostor-ctl add-s3gw -a 127.0.0.1:9003 -V 0100000000000001
# ostor-ctl add-s3gw -a 127.0.0.1:9004 -V 0100000000000001
```

12. Запустите службу агента хранилища объектов на каждом сервере кластера хранилища объектов, добавленном в кластерную конфигурацию.

```
# systemctl start ostor-agentd.service
# systemctl enable ostor-agentd.service
```

13. Убедитесь, что службы NS и OS назначены серверам.

    По умолчанию агенты попытаются назначить службы NS и OS серверам автоматически, использовав циклический метод (round-robin). Однако необходимо назначение вручную, если в кластерную конфигурацию был добавлен новый сервер или существующая конфигурация не является оптимальной (подробности см. в разделе *Назначение служб S3 серверам вручную*).

    Вы можете проверить текущее назначение служб с помощью команды `ostor-ctl agent-status`, например:

```
# ostor-ctl agent-status
TYPE     SVC_ID               STATUS          UPTIME  HOST_ID            ADDRS
S3GW     8000000000000009     ACTIVE             527  fcbf5602197245da   127.0.0.1:9090
S3GW     8000000000000008     ACTIVE             536  4f0038db65274507   127.0.0.1:9090
S3GW     8000000000000007     ACTIVE             572  958e982fcc794e58   127.0.0.1:9090
OS       1000000000000005     ACTIVE             452  4f0038db65274507   10.30.29.124:39746
OS       1000000000000004     ACTIVE             647  fcbf5602197245da   10.30.27.69:56363
OS       1000000000000003     ACTIVE             452  4f0038db65274507   10.30.29.124:52831
NS       0800000000000002     ACTIVE             647  fcbf5602197245da   10.30.27.69:56463
NS       0800000000000001     ACTIVE             452  4f0038db65274507   10.30.29.124:53044
NS       0800000000000000     ACTIVE             647  fcbf5602197245da   10.30.27.69:37876
```

14. На каждом сервере с экземплярами службы шлюза установите веб-сервер `nginx`, который будет обслуживать запросы S3 от конечных пользователей:

```
# yum install nginx
```

Используйте утилиту **ostor-configure-nginx**, чтобы настроить nginx для обслуживания запросов S3. Есть два параметра конфигурации:

- create создает новую конфигурацию nginx.
- update обновляет существующую конфигурацию nginx. Используйте этот параметр после изменения конфигурации служб шлюза (GW).

Для первоначальной конфигурации nginx используйте параметр create. Например, для HTTP выполните:

```
# ostor-configure-nginx create -D s3.mydomain.com -p 80
```

В этой команде s3.mydomain.com — это домен оконечной точки S3, а 80 — это порт, который nginx будет использовать.

Чтобы настроить HTTP с сертификатом SSL для домена оконечной точки S3, а также для его поддоменов, укажите сертификат и ключ, например:

```
# ostor-configure-nginx create -D s3.mydomain.com -p 443 --ssl --ssl-cert file.cert --ssl-
↪key file.key
```

Будет создан конфигурационный файл /etc/nginx/conf.d/ostor-s3.conf. Перенаправление запросов локальным экземплярам служб шлюза (GW) с использованием FastCGI будет осуществляться согласно настройкам в этом файле.

15. Запустите nginx:

```
# systemctl start nginx.service
# systemctl enable nginx.service
```

16. Добавьте серверы, которые находятся в отказоустойчивом кластере, но не запускайте на них никаких служб хранилища объектов. То есть удостоверьтесь, что все серверы, которые добавлены в отказоустойчивый кластер, также добавлены в кластер хранилища объектов. Это необходимо для правильной работы механизмов обеспечения высокой доступности.

```
# ostor-ctl add-host -n <IP_addr>
# systemctl start ostor-agentd.service
# systemctl enable ostor-agentd.service
```

Хранилище объектов развернуто. Теперь вы можете добавлять пользователей S3 с помощью утилиты **ostor-s3-admin**, как описано в разделе *Создание пользователей S3*.

Для проверки успешности установки или для мониторинга состояния хранилища объектов используйте

команду `ostor-ctl get-config`, например:

```
# ostor-ctl get-config
07-08-15 11:58:45.470 Use configuration service 'ostor'
SVC_ID             TYPE  URI
8000000000000006   S3GW  svc://1039c0dc90d64607/?address=127.0.0.1:9000
0800000000000000     NS  vstorage://cluster1/ostor/services/0800000000000000
1000000000000001     OS  vstorage://cluster1/ostor/services/1000000000000001
1000000000000002     OS  vstorage://cluster1/ostor/services/1000000000000002
1000000000000003     OS  vstorage://cluster1/ostor/services/1000000000000003
1000000000000004     OS  vstorage://cluster1/ostor/services/1000000000000004
8000000000000009   S3GW  svc://7a1789d20d9f4490/?address=127.0.0.1:9000
800000000000000c   S3GW  svc://7a1789d20d9f4490/?address=127.0.0.1:9090
```

Рекомендации по включению службы политики корзин (ACC)

1. Создайте конфигурационный файл `<cluster_mount_point>/<ostor_dir>/<volume_ID>/control/lifecycle` со следующим содержимым:

   ```
   QUERY_INTERVAL_SECONDS=21600
   ```

2. Создайте системного пользователя:

   ```
   ostor-s3-admin create-user -e lifecycle -S -V <volume_ID>
   ```

Изменения вступят в силу в течение 15 минут.

Назначение служб S3 серверам вручную

При развертывании хранилища объектов вы можете вручную назначать службы серверам с помощью команды `ostor-ctl bind`. Вам нужно будет указать идентификатор целевого сервера и от одного до нескольких идентификаторов служб, чтобы назначить их этому серверу. Например:

```
# ostor-ctl bind -H 4f0038db65274507 -S 0800000000000001 -S 1000000000000003 -S 1000000000000005
```

Эта команда назначает службы с идентификаторами `800000000000001`, `1000000000000003` и `1000000000000005` серверу с идентификатором `4f0038db65274507`.

Служба может быть назначена серверу, который подключен к общему хранилищу, в котором хранятся данные этой службы. То есть имя кластера хранилища в URI-идентификаторе службы должно совпадать с именем кластера хранилища в URI-идентификаторе сервера.

Например, в конфигурации с двумя общими хранилищами `stor1` и `stor2` (см. ниже) службы с

идентификаторами URI, начинающимися с `vstorage://stor1`, могут быть назначены только серверам `host510` и `host511`, тогда как службы с идентификаторам URI, начинающимися с `vstorage://stor2`, могут быть назначены только серверам `host512` и `host513`.

```
# ostor-ctl get-config
SVC_ID             TYPE  URI
0800000000000000     NS  vstorage://stor1/s3-data/services/0800000000000000
0800000000000001     NS  vstorage://stor1/s3-data/services/0800000000000001
0800000000000002     NS  vstorage://stor2/s3-data/services/0800000000000002
1000000000000003     OS  vstorage://stor1/s3-data/services/1000000000000003
1000000000000004     OS  vstorage://stor2/s3-data/services/1000000000000004
1000000000000005     OS  vstorage://stor1/s3-data/services/1000000000000005
HOST_ID            HOSTNAME      URI
0fcbf5602197245da  host510:2530  vstorage://stor1/s3-data
4f0038db65274507   host511:2530  vstorage://stor1/s3-data
958e982fcc794e58   host512:2530  vstorage://stor2/s3-data
953e976abc773451   host513:2530  vstorage://stor2/s3-data
```

Управление пользователями S3

Концепция пользователя S3 — одна из базовых идей хранилища объектов наряду с концепциями объекта и корзины (контейнера для хранения объектов). В протоколе Amazon S3 используется модель разрешений на основе списков контроля доступа (ACL), где каждой корзине и каждому объекту назначается список ACL, в котором указаны все пользователи с доступом к данному ресурсу и тип доступа (чтение, запись, чтение ACL, запись ACL). Список пользователей включает в себя владельца сущности, который назначается каждому объекту и корзине при создании. Владелец сущности имеет дополнительные права по сравнению с другими пользователями. Например, только владелец корзины может ее удалить.

Модель пользователей и политики доступа, реализованные в продукте Кибер Хранилище, соответствуют модели пользователей и политикам доступа Amazon S3.

Сценарии управления пользователями в продукте Кибер Хранилище большей частью основаны на управлении пользователями в Amazon Web Services и включают следующие операции: создание, запрос и удаление пользователей, а также формирование и отзыв ключей доступа пользователей.

Вы можете управлять пользователями с помощью утилиты **`ostor-s3-admin`**. Чтобы делать это, вам будет нужно получить идентификатор тома пользователей. Вы можете получить его с помощью команды `ostor-ctl get-config`, например:

```
# ostor-ctl get-config -n 10.94.97.195
VOL_ID             TYPE     STATE
0100000000000002   OBJ      READY
...
```

Примечание

Поскольку предполагается, что команды `ostor-s3-admin` будут запускаться администраторами хранилища объектов, эти команды не включают в себя никаких проверок подлинности или прав доступа.

Создание пользователей S3

Вы можете создать пользователя S3 с уникальным идентификатором и ключом доступа, состоящим из идентификатора ключа доступа и секретного ключа доступа, использовав команду `ostor-s3-admin create-user`. При использовании команды вам нужно указать адрес электронной почты пользователя. Например:

```
# ostor-s3-admin create-user -e user@email.com -V 0100000000000002
UserEmail:user@email.com
UserId:a49e12a226bd760f
KeyPair[0]:S3AccessKeyId:a49e12a226bd760fGHQ7
KeyPair[0]:S3SecretAccessKey:HSDu2DA00JNGjnRcAhLKfhrvlymzOVdLPsCK2dcq
Flags:none
```

Идентификатор пользователя S3 — это 16-разрядное шестнадцатеричное число в виде строки. Сгенерированный ключ доступа используется для создания подписи в запросах к хранилищу объектов согласно схеме аутентификации Amazon S3 Signature Version 2.

Просмотр пользователей S3

Вы можете просматривать всех пользователей хранилища объектов с помощью команды `ostor-s3-admin query-users`. Сведения о каждом пользователе могут занимать в таблице от одной до нескольких строк. Дополнительные строки используются, чтобы перечислить ключи доступа пользователя, состоящие из идентификаторов ключей доступа и секретных ключей доступа. Если у пользователя нет ключей доступа, то в соответствующих ячейках таблицы отображается знак минуса (-). Например:

```
# ostor-s3-admin query-users -V 0100000000000002
      S3 USER ID      S3 ACCESS KEY ID              S3 SECRET ACCESS KEY  S3 USER EMAIL
bf0b3b15eb7c9019  bf0b3b15eb7c9019I36Y                                ***  user2@abc.com
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
d866d9d114cc3d20  d866d9d114cc3d20G456                          ***  user1@abc.com
                  d866d9d114cc3d20D8EW                          ***
e86d1c19e616455                     -                            -  user3@abc.com
```

Чтобы вывести список пользователей в формате XML, используйте параметр `-X`. Чтобы вывести секретные ключи доступа, используйте параметр `-a`. Например:

```
# ostor-s3-admin query-users -V 0100000000000002 -a -X
<?xml version=”1.0” encoding=”UTF-8”?><QueryUsersResult><Users><User><Id>a49e12a226bd760f</Id>
↪<Email>user@email.com</Email><Keys><OwnerId>0000000000000000</OwnerId><KeyPair><S3AccessKeyId>
↪a49e12a226bd760fGHQ7</S3AccessKeyId><S3SecretAccessKey>HSDu2DA00JNGjnRcAhLKfhrvlymzOVdLPsCK2dcq
↪</S3SecretAccessKey></KeyPair></Keys></User><User><Id>d7c53fc1f931661f</Id><Email>user@email.com
↪</Email><Keys><OwnerId>0000000000000000</OwnerId><KeyPair><S3AccessKeyId>d7c53fc1f931661fZLIV</
↪S3AccessKeyId><S3SecretAccessKey>JL7gt1OH873zR0Fzv8Oh9ZuA6JtCVnkgV7lET6ET</S3SecretAccessKey></
↪KeyPair></Keys></User></Users></QueryUsersResult>
```

Просмотр сведений о пользователе S3

Чтобы вывести сведения о пользователе S3, используйте команду `ostor-s3-admin query-user-info`. При запуске команды вам нужно указать адрес электронной почты пользователя (`-e`) или его идентификатор (`-i`). Например:

```
# ostor-s3-admin query-user-info -e user@email.com -V 0100000000000002
Query user: user id=d866d9d114cc3d20, user email=user@email.com
Key pair[0]: access key id=d866d9d114cc3d20G456,
secret access key=5EAne6PLL1jxprouRqq8hmfONMfgrJcOwbowCoTt
Key pair[1]: access key id=d866d9d114cc3d20D8EW,
secret access key=83tTsNAuuRyoBBqhxMFqHAC60dhKHtTCCkQe54zu
```

Отключение пользователей S3

Вы можете отключить пользователя S3 с помощью команды `ostor-s3-admin disable-user`. При запуске команды вам нужно указать адрес электронной почты пользователя (`-e`) или его идентификатор (`-i`). Например:

```
# ostor-s3-admin disable-user -e user@email.com -V 0100000000000002
```

Удаление пользователей S3

Вы можете удалить пользователя S3 с помощью команды `ostor-s3-admin delete-user`. Пользователи, которые являются владельцами корзин, не могут быть удалены, поэтому вначале удалите их корзины. При запуске команды вам нужно указать адрес электронной почты пользователя (`-e`) или его идентификатор (`-i`). Например:

```
# ostor-s3-admin delete-user -i bf0b3b15eb7c9019 -V 0100000000000002
Deleted user: user id=bf0b3b15eb7c9019
```

Генерация ключей доступа для пользователей S3

Вы можете сгенерировать новый ключ доступа для пользователя S3 с помощью команды `ostor-s3-admin gen-access-key`. Как и в AWS, у одного пользователя S3 может быть до двух активных ключей доступа. При запуске команды вам нужно указать адрес электронной почты пользователя (`-e`) или его идентификатор (`-i`). Например:

```
# ostor-s3-admin gen-access-key -e user@email.com -V 0100000000000002
Generate access key: user id=d866d9d114cc3d20, access key id=d866d9d114cc3d20D8EW,
secret access key=83tTsNAuuRyoBBqhxMFqHAC60dhKHtTCCkQe54zu
```

Примечание

Рекомендуется периодически отзывать старые ключи доступа и генерировать новые.

Отзыв ключей доступа пользователей

Вы можете отозвать ключ доступа пользователя с помощью команды `ostor-s3-admin revoke-access-key`. Вам нужно указать идентификатор ключа доступа, который вы хотите отозвать, а также идентификатор или адрес электронной почты пользователя, которому этот ключ принадлежит, например:

```
# ostor-s3-admin revoke-access-key -e user@email.com -k de86d1c19e616455YIPU -V 0100000000000002
Revoke access key: user id=de86d1c19e616455, access key id=de86d1c19e616455YIPU
```

Примечание

Рекомендуется периодически отзывать старые ключи доступа и генерировать новые.

Управление корзинами хранилища объектов

Все объекты в хранилище, подобном Amazon S3, хранятся в контейнерах, называемых корзинами. В качестве адреса для обращения корзины имеют имена, уникальные для данного хранилища объектов, поэтому пользователь S3 не сможет создать корзину с таким же именем, как у другой корзины в том же хранилище объектов. Корзины используются, чтобы:

- группировать объекты и изолировать их от объектов в других корзинах;
- обеспечивать механизмы управления ACL для объектов в корзинах;
- задавать политики доступа для корзин (например, для включения управления версиями объектов в корзине).

Вы можете управлять корзинами с помощью утилиты **`ostor-s3-admin`**, API сервиса S3, а также посредством сторонних браузеров S3, таких как CyberDuck или DragonDisk. Утилита `ostor-s3-admin` должна использоваться администраторами хранилища объектов, поэтому при ее использовании не выполняется аутентификация или авторизация. Рекомендуется в первую очередь использовать стандартные команды API сервиса Amazon S3.

Чтобы управлять корзинами из командной строки, вам нужно знать идентификатор тома, на котором размещены эти корзины. Вы можете получить его с помощью команды `ostor-ctl get-config`, например:

```
# ostor-ctl get-config -n 10.94.97.195
VOL_ID             TYPE     STATE
0100000000000002   OBJ      READY
<...>
```

Важно

Операции изменения и удаления корзины принудительно выполняются в объектном хранилище. Эти команды не являются частью стандартного API S3 и могут нарушить интеграцию с внешними системами биллинга и учета. Используйте их только по уважительной причине и тогда, когда вы знаете, что делаете.

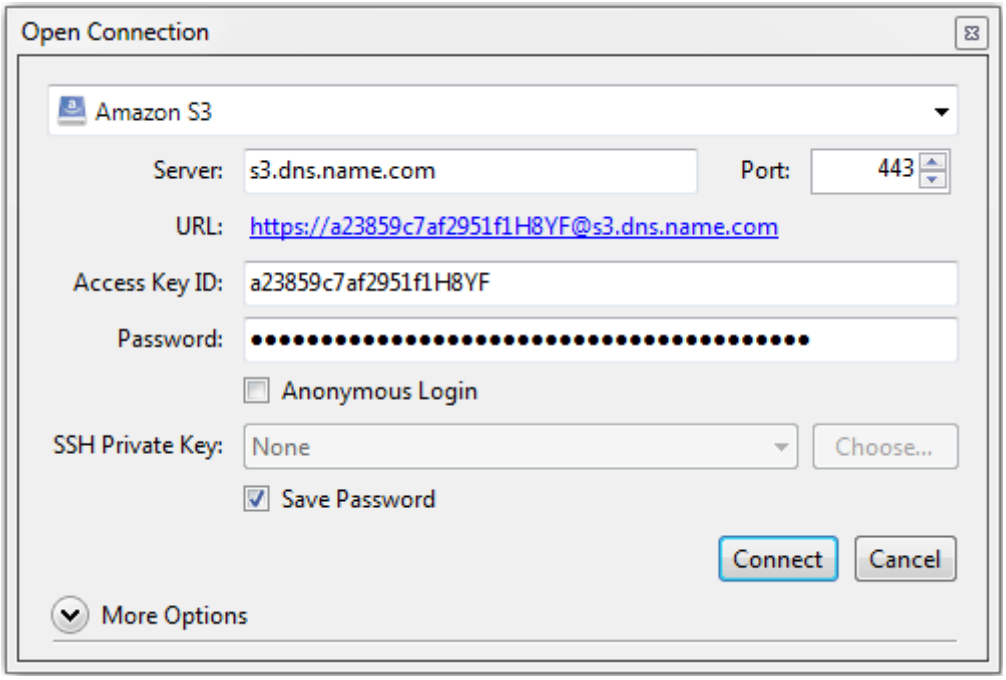
Управление корзинами с помощью CyberDuck

В этом разделе рассказывается об управлении корзинами с помощью приложения CyberDuck.

Создание корзин

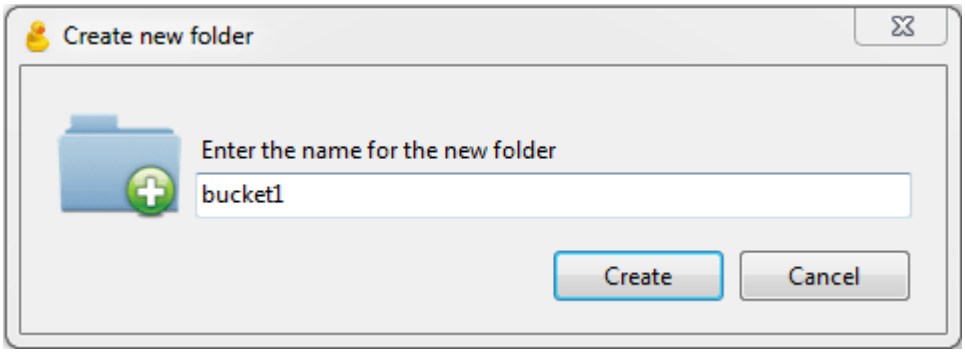
Чтобы создать корзину S3 с помощью CyberDuck, сделайте следующее:

1. Нажмите **Open Connection**.
2. Укажите следующие параметры:
   - Внешнее DNS-имя оконечной точки S3, которое вы указали при создании кластера хранилища объектов.
   - Идентификатор ключа доступа и секретный ключ доступа пользователя S3 (см. раздел *Создание пользователей S3*).



   По умолчанию подключение устанавливается через протокол HTTPS. Чтобы использовать CyberDuck поверх HTTP, необходимо установить специальный профиль S3.

3. После того как подключение будет установлено, выберите **File** > **New Folder**.



4. Укажите имя для новой корзины, а затем нажмите **Create**.

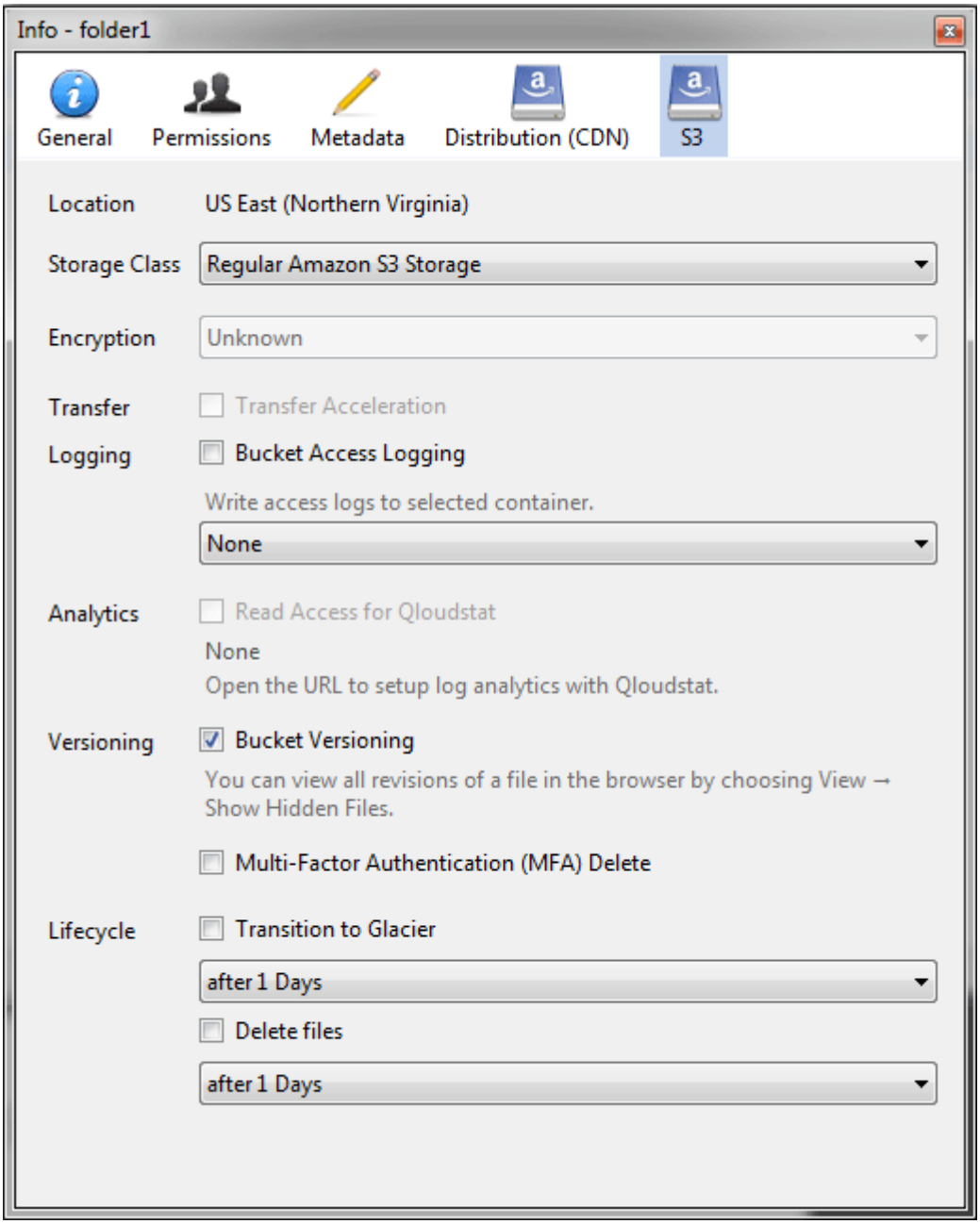
> **Примечание**
>
> Рекомендуется использовать имена корзин, соответствующие соглашениям об именовании DNS. Подробные сведения об именовании корзин см. в разделе *Политики именования корзин и ключей объектов*.

Новая корзина появится в CyberDuck, а вы сможете управлять ей и загружать в нее файлы.

Управление версиями объектов

Управление версиями позволяет поддерживать несколько вариантов одного объекта в одной и той же корзине. С его помощью можно хранить, извлекать и восстанавливать любую версию любого объекта, хранящегося в вашей корзине S3. С помощью управления версиями можно легко выполнять восстановление как после непреднамеренных действий пользователей, так и после сбоев приложений. Дополнительные сведения об управлении версиями объектов см. в документации Amazon.

Управление версиями объектов по умолчанию отключено. В CyberDuck его можно включить в свойствах корзины, например:



Просмотр содержимого корзины

Вы можете просмотреть содержимое корзины с помощью браузера. Чтобы это сделать, перейдите по адресу, состоящему из внешнего DNS-имени оконечной точки S3, которое вы указали при создании кластера хранилища объектов, и имени корзины, например: `mys3storage.example.com/mybucket`.

> **Примечание**
>
> Вы также можете скопировать адрес корзины, щелкнув на ней правой кнопкой мыши и затем выбрав **Copy URL**.

Управление корзинами с помощью командной строки

В этом разделе рассказывается об управлении корзинами с помощью интерфейса командной строки.

Просмотр списка корзин

Вы можете просмотреть список всех корзин хранилища объектов с помощью команды `ostor-s3-admin list-all-buckets`. Для каждой корзины эта команда выведет владельца, дату создания, статус управления версиями, общий размер (размер всех объектов, хранящихся в корзине, плюс размер всех незавершенных передач по частям для данной корзины). Например:

```
# ostor-s3-admin list-all-buckets -V 0100000000000002
Total 3 buckets
BUCKET                OWNER             CREATION_DATE  VERSIONING     TOTAL SIZE, BYTES
bucket1    968d1a79968d1a79  2015-08-18T09:32:35.000Z        none                  1024
bucket2    968d1a79968d1a79  2015-08-18T09:18:20.000Z     enabled                     0
bucket3    968d1a79968d1a79  2015-08-18T09:22:15.000Z   suspended               1024000
```

Чтобы вывести этот список в формате XML, используйте параметр `-X`, например:

```
# ostor-s3-admin list-all-buckets -X
<?xml version=”1.0” encoding=”UTF-8”?><ListBucketsResult><Buckets><Bucket><Name>bucker2</Name>
↪<Owner>d7c53fc1f931661f</Owner><CreationDate>2017-04-03T17:11:44.000Z</CreationDate><Versioning>
↪none</Versioning><Notary>off</Notary><TotalSize>0</TotalSize></Bucket><Bucket><Name>bucket1</
↪Name><Owner>d7c53fc1f931661f</Owner><CreationDate>2017-04-03T17:11:33.000Z</CreationDate>
↪<Versioning>none</Versioning><Notary>off</Notary><TotalSize>0</TotalSize></Bucket></Buckets></
↪ListBucketsResult>
```

Чтобы вывести принадлежащие указанному пользователю S3 корзины, используйте параметр `-i`, например:

```
# ostor-s3-admin list-all-buckets -i d7c53fc1f931661f
BUCKET   OWNER             CREATION_DATE             VERSIONING  TOTAL_SIZE NOTARY NOTARY_PROVIDER
bucker2  d7c53fc1f931661f  2017-04-03T17:11:44.000Z  none        0          off    0
```

Получение сведений о корзине

Вы можете получить метаданные и список управления доступом (ACL) корзины с помощью команды `ostor-s3-admin query-bucket-info`. Например, чтобы это сделать для корзины `bucket1`, выполните:

```
# ostor-s3-admin query-bucket-info -b bucket1 -V 0100000000000002
BUCKET   OWNER             CREATION_DATE             VERSIONING  TOTAL_SIZE
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
bucket1  d339edcf885eeafc  2017-12-21T12:42:46.000Z  none        0

ACL: d339edcf885eeafc: FULL_CONTROL
```

Смена владельца корзины

Вы можете передать владение корзиной указанному пользователю с помощью команды `ostor-s3-admin change-bucket-owner`. Например, чтобы сделать пользователя с идентификатором `bf0b3b15eb7c9019` владельцем корзины `bucket1`, выполните:

```
# ostor-s3-admin change-bucket-owner -b bucket1 -i bf0b3b15eb7c9019 -V 0100000000000002
Changed owner of the bucket bucket1. New owner bf0b3b15eb7c9019
```

Удаление корзин

Вы можете удалить корзину с помощью команды `ostor-s3-admin delete-bucket`. Удаление корзины приведет к удалению всех находящихся в ней объектов и их версий, а также к удалению всех незавершенных передач по частям для данной корзины. Например:

```
# ostor-s3-admin delete-bucket -b bucket1 -V 0100000000000002
```

Установка нового пароля хранилища объектов

Чтобы установить новый пароль хранилища объектов, выполните следующие действия:

1. Получите список серверов, на которых установлена служба конфигурации хранилища объектов:

   ```
   # ostor-ctl get-config
   <...>
   CFGD_ID                    ADDR       IS MASTER
   3          192.168.194.254:2532              no
   2           192.168.194.46:2532              no
   1           192.168.193.52:2532             yes
   <...>
   ```

2. Выполните следующие шаги на каждом сервере из списка:

   1. Остановите службу конфигурации:

      ```
      # systemctl stop ostor-cfgd.service
      ```

2. Запустите команду, приведенную ниже, и введите новый пароль.

```
# ostor-ctl passwd -r /var/lib/ostor/configuration
```

3. Запустите службу конфигурации:

```
# systemctl start ostor-cfgd.service
```

Рекомендации по использованию хранилища объектов

Этот раздел содержит рекомендации по использованию различных возможностей хранилища объектов. Эти рекомендации призваны помочь вам включить дополнительные функции или улучшить удобство использования или производительность хранилища объектов.

Политики именования корзин и ключей объектов

Рекомендуется использовать имена корзин, соответствующие соглашениям об именовании DNS:

- длиной от 3 до 63 символов;
- должны начинаться и заканчиваться буквой нижнего регистра или цифрой;
- могут содержать буквы нижнего регистра, цифры, точки (.), дефисы (-) и символы подчеркивания (_);
- могут представлять собой ряд допустимых частей имен (описанных ранее), разделенных точками.

Ключ объекта может быть строкой из любых символов в кодировке UTF-8 длиной до 1024 байт.

Улучшение производительности операций PUT

Хранилище объектов поддерживает передачу объектов размером до 5 ГБ на один запрос PUT. Производительность передачи можно улучшить, разбив большие объекты на фрагменты и передавая их параллельно с помощью API передачи по частям. Такой способ позволит разделить нагрузку между несколькими серверами объектов (OS).

Рекомендуется использовать передачу по частям для объектов размером больше 5 МБ.

Приложения

Этот раздел содержит справочные сведения о хранилище объектов.

Приложение A. Поддерживаемые операции REST Amazon S3

Следующие операции REST Amazon S3 в настоящее время поддерживаются реализацией протокола Amazon S3 в продукте Кибер Хранилище:

Поддерживаемые операции с сервисами:

- GET Service.

Поддерживаемые операции с корзинами:

- DELETE Bucket,
- GET Bucket (List Objects),
- GET Bucket ACL,
- GET Bucket Location,
- GET Bucket Object Versions,
- GET Bucket Versioning,
- HEAD Bucket,
- List Multipart Uploads,
- PUT Bucket,
- PUT Bucket ACL,
- PUT Bucket Versioning.

Поддерживаемые операции с объектами:

- DELETE Object,
- DELETE Multiple Objects,
- GET Object,
- GET Object ACL,
- HEAD Object,
- POST Object,
- PUT Object,
- PUT Object - Copy,
- PUT Object ACL,
- Initiate Multipart Upload,

- Upload Part,
- Complete Multipart Upload,
- Abort Multipart Upload,
- List Parts,
- Upload Part Copy.

Примечание

Полный список операций см. в документации по REST API Amazon S3.

Приложение Б. Поддерживаемые заголовки запросов Amazon S3

Следующие заголовки запросов REST Amazon S3 в настоящее время поддерживаются используемой в продукте Кибер Хранилище реализацией протокола Amazon S3:

- Authorization,
- Content-Length,
- Content-MD5,
- Content-Type,
- Date,
- Host,
- x-amz-acl,
- x-amz-algorithm,
- x-amz-bucket-object-lock-enabled,
- x-amz-bypass-governance-retention,
- x-amz-content-sha256,
- x-amz-copy-source,
- x-amz-credential,
- x-amz-date,
- x-amz-delete-marker,
- x-amz-geo-access-key,

- x-amz-geo-access-secret,
- x-amz-geo-endpoint,
- x-amz-geo-region,
- x-amz-grant-full-control,
- x-amz-grant-read,
- x-amz-grant-read-acp,
- x-amz-grant-write,
- x-amz-grant-write-acp,
- x-amz-meta-<name> (поддерживаемые символы в имени ключа: строчные и заглавные буквы латинского алфавита, цифры, знак минуса),
- x-amz-metadata-directive,
- x-amz-object-lock-legal-hold,
- x-amz-object-lock-mode,
- x-amz-object-lock-retain-until-date,
- x-amz-object-map,
- x-amz-s3-restore-version,
- x-amz-set-modification-date,
- x-amz-signature,
- x-amz-storage-class,
- x-amz-tagging,
- x-amz-version-id.

Приложение B. Поддерживаемые схемы проверки подлинности

Реализация протокола Amazon S3 в продукте Кибер Хранилище поддерживает следующие схемы проверки подлинности:

- Signature Version 2 (Подпись версии 2),
- Signature Version 4 (Подпись версии 4).

Приложение Г. Возможности Amazon S3, поддерживаемые политиками корзин

Реализация политик корзин Amazon S3 в продукте Кибер Хранилище поддерживает следующие действия, ключи условий и операции сравнения:

Действия с объектами:

- s3:AbortMultipartUpload,
- s3:DeleteObject,
- s3:DeleteObjectTagging,
- s3:DeleteObjectVersion,
- s3:DeleteObjectVersionTagging,
- s3:GetObject,
- s3:GetObject,
- s3:GetObjectAcl,
- s3:GetObjectTagging,
- s3:GetObjectTorrent,
- s3:GetObjectVersion,
- s3:GetObjectVersionAcl,
- s3:GetObjectVersionTagging,
- s3:ListMultipartUploadParts,
- s3:PutObject,
- s3:PutObjectAcl,
- s3:PutObjectTagging,
- s3:PutObjectVersionAcl,
- s3:PutObjectVersionTagging,
- s3:RestoreObject.

Действия с корзинами:

- s3:CreateBucket,
- s3:DeleteBucket,

- s3:ListBucket,
- s3:ListBucketMultipartUploads,
- s3:ListBucketVersions.

Действия с подресурсами корзин:

- s3:DeleteBucketPolicy,
- s3:DeleteBucketWebsite,
- s3:GetBucketAcl,
- s3:GetBucketCORS,
- s3:GetBucketLocation,
- s3:GetBucketLogging,
- s3:GetBucketNotification,
- s3:GetBucketPolicy,
- s3:GetBucketRequestPayment,
- s3:GetBucketTagging,
- s3:GetBucketVersioning,
- s3:GetBucketWebsite,
- s3:GetLifecycleConfiguration,
- s3:GetReplicationConfiguration,
- s3:PutBucketAcl,
- s3:PutBucketCORS,
- s3:PutBucketLogging,
- s3:PutBucketNotification,
- s3:PutBucketPolicy,
- s3:PutBucketRequestPayment,
- s3:PutBucketTagging,
- s3:PutBucketVersioning,
- s3:PutBucketWebsite,

- s3:PutLifecycleConfiguration,
- s3:PutReplicationConfiguration.

Ключи условий:

- s3:x-amz-storage-class,
- s3:x-amz-acl,
- s3:x-amz-grant-full-control,
- s3:x-amz-grant-read,
- s3:x-amz-grant-read-acp,
- s3:x-amz-grant-write,
- s3:x-amz-grant-write-acp,
- s3:x-amz-copy-source,
- s3:TlsVersion,
- s3:x-amz-content-sha256,
- s3:signatureversion,
- s3:signatureAge,
- s3:authType,
- s3:x-amz-website-redirect-location,
- s3:object-lock-mode,
- s3:object-lock-retain-until-date,
- s3:object-lock-legal-hold,
- s3:object-lock-remaining-retention-days,
- s3:prefix,
- s3:versionid,
- s3:max-keys,
- s3:locationconstraint,
- aws:SourceIp.

Операции сравнения:

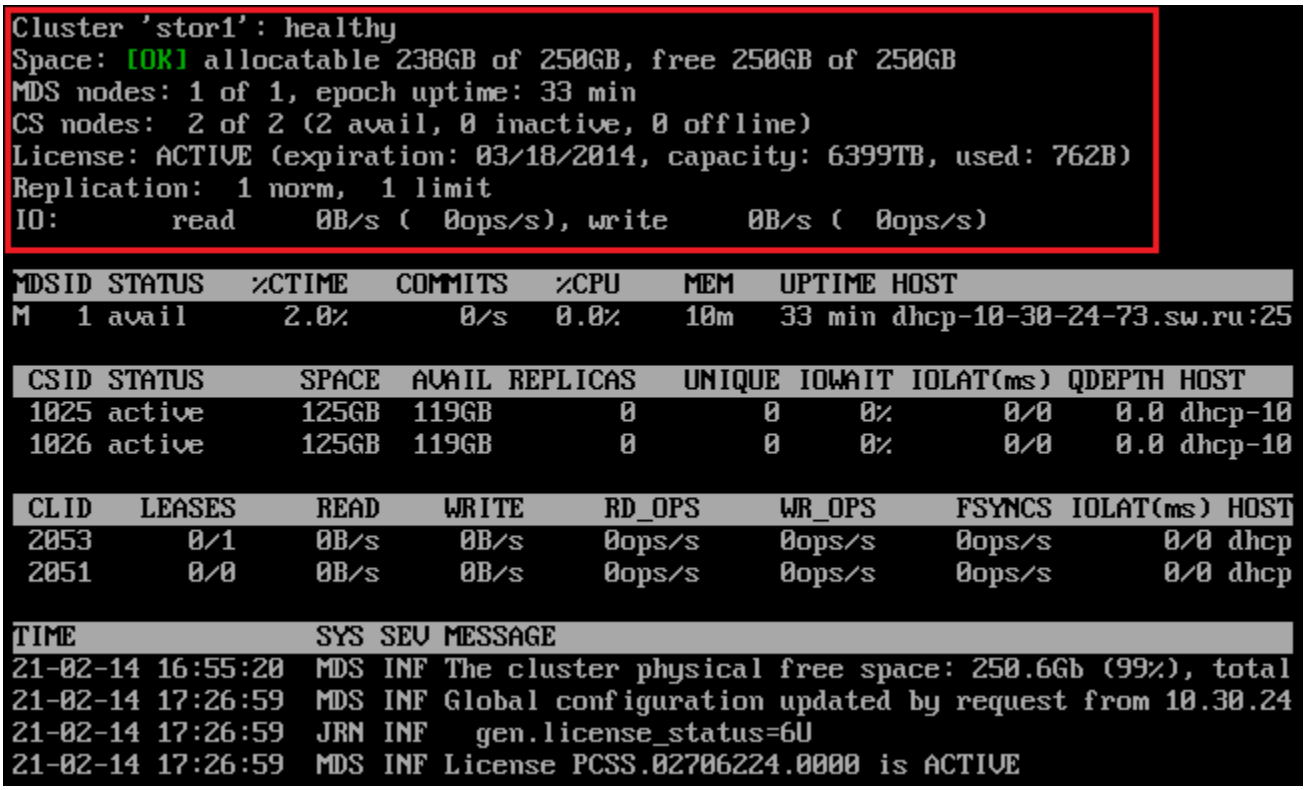
- StringNotEquals,
- StringEqualsIgnoreCase,
- StringNotEqualsIgnoreCase,
- StringLike,
- StringNotLike,
- NumericEquals,
- NumericNotEquals,
- NumericLessThan,
- NumericLessThanEquals,
- NumericGreaterThan,
- NumericGreaterThanEquals,
- DateEquals,
- DateNotEquals,
- DateLessThan,
- DateLessThanEquals,
- DateGreaterThan,
- DateGreaterThanEquals,
- BinaryEquals,
- IpAddress,
- NotIpAddress.

Мониторинг кластеров хранилища

Мониторинг кластера хранилища очень важен, поскольку он позволяет вам проверять состояние и работоспособность всех компонентов кластера и реагировать так, как это необходимо. Этот раздел объясняет, как выполнять мониторинг вашего кластера хранилища.

Мониторинг основных параметров кластера

Посредством мониторинга основных параметров вы можете получать подробные сведения обо всех компонентах кластера хранилища, его общем состоянии и работоспособности. Для вывода этой информации используйте команду `vstorage -c <cluster_name> top`, например:



Приведенная выше команда показывает подробные сведения о кластере хранилища `stor1`. Основные параметры (выделены красным) объяснены в таблице ниже.

Параметр	Описание
Cluster	Общее состояние кластера: • `healthy`: Все серверы фрагментов данных в этом кластере активны. • `unknown`: Недостаточно информации о состоянии этого кластера (например, потому, что главный сервер метаданных был выбран только некоторое время назад). • `degraded`: Часть серверов фрагментов данных в кластере неактивна. • `failure`: В кластере слишком много неактивных серверов фрагментов данных; автоматическая репликация отключена. • `SMART warning`: У одного или нескольких физических дисков, подключенных к серверам кластера, близится аппаратный отказ. Подробности см. в разделе *Мониторинг физических дисков*.
Space	Количество дискового пространства в кластере: • `free`: Свободное дисковое пространство в кластере. • `allocatable`: Объем логического дискового пространства, доступного для клиентов. Доступное для выделения дисковое пространство рассчитывается на основе текущих параметров репликации и объема свободного дискового пространства на серверах фрагментов данных. Объем логического дискового пространства также может ограничиваться лицензией. Дополнительные сведения про мониторинг и анализ использования дискового пространства в кластерах хранилища см. в разделе *Общие сведения об использовании дискового пространства*.
MDS nodes	Количество активных серверов метаданных по сравнению с общим числом серверов метаданных, настроенных для кластера.
epoch time	Время, прошедшее с момента выбора главного сервера метаданных.

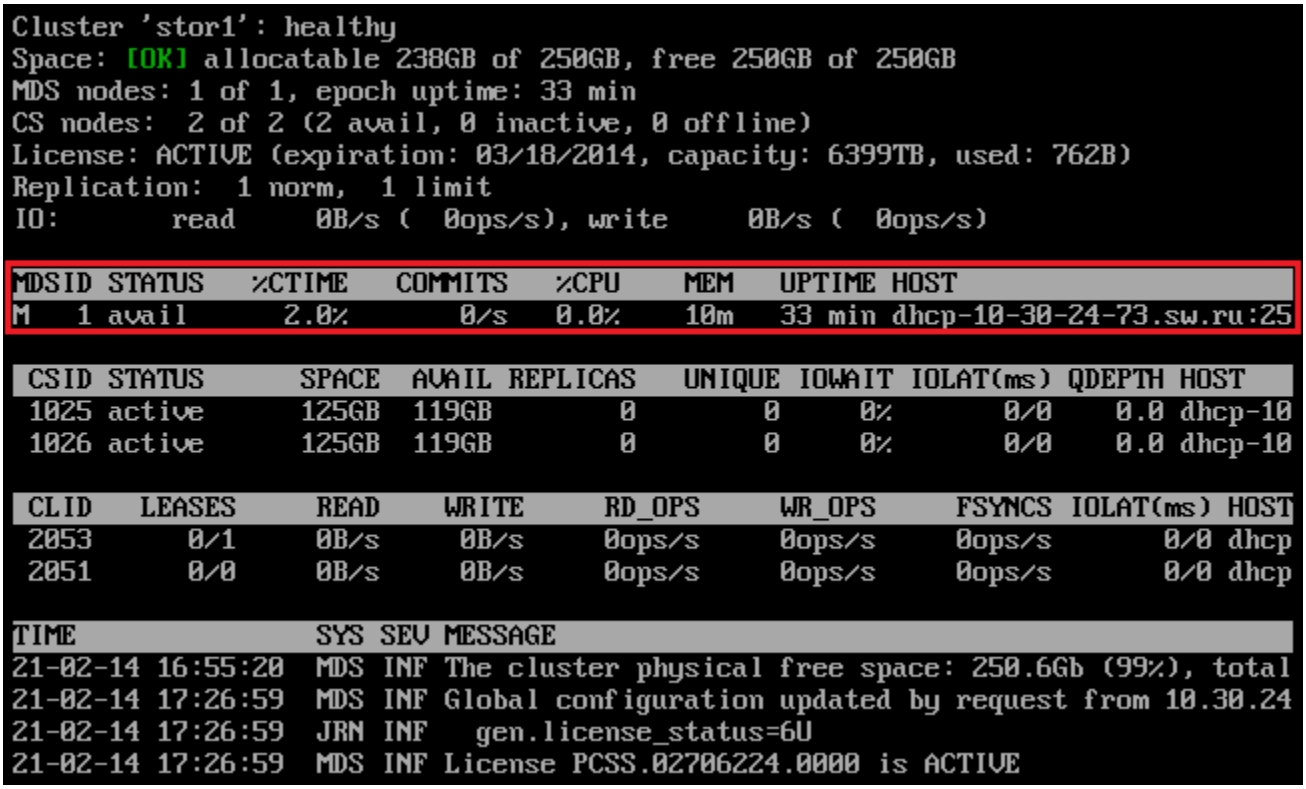
(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Параметр	Описание
CS nodes	Количество активных серверов фрагментов данных по сравнению с общим числом серверов фрагментов данных, настроенных для кластера. В скобках отображается дополнительная информация об этих серверах фрагментов данных: • `avail.`: Активные серверы фрагментов данных, которые в настоящее время запущены и работают в кластере. • `inactive`: Неактивные серверы фрагментов данных, которые в настоящее время недоступны. Сервер фрагментов данных помечается как неактивный в течение первых 5 минут его неактивности. • `offline`: Отключенные серверы фрагментов данных, которые были неактивны в течение более 5 минут. Сервер фрагментов данных меняет свое состояние на отключенный, если он неактивен больше 5 минут. После изменения состояния сервера на `offline` кластер начинает репликацию данных, чтобы восстановить фрагменты, которые хранились на отключенном сервере фрагментов данных.
License	Номер ключа, под которым лицензия зарегистрирована на сервере проверки подлинности ключей (Key Authentication server), и состояние лицензии.
Replication	Параметры репликации. Нормальное количество реплик фрагментов и предельное число, ниже которого фрагмент блокируется до момента, когда он будет восстановлен.
IO	Активность дискового ввода-вывода в кластере: • Скорость операций ввода-вывода (чтения и записи) в байтах в секунду. • Количество операций ввода-вывода (чтения и записи) в секунду.

Мониторинг серверов метаданных

Серверы метаданных (MDS) являются критически важным компонентом любого кластера хранилища, поэтому контроль состояния и работоспособности серверов метаданных — это крайне важная задача. Для мониторинга серверов метаданных используйте команду `vstorage -c <cluster_name> top`, например:

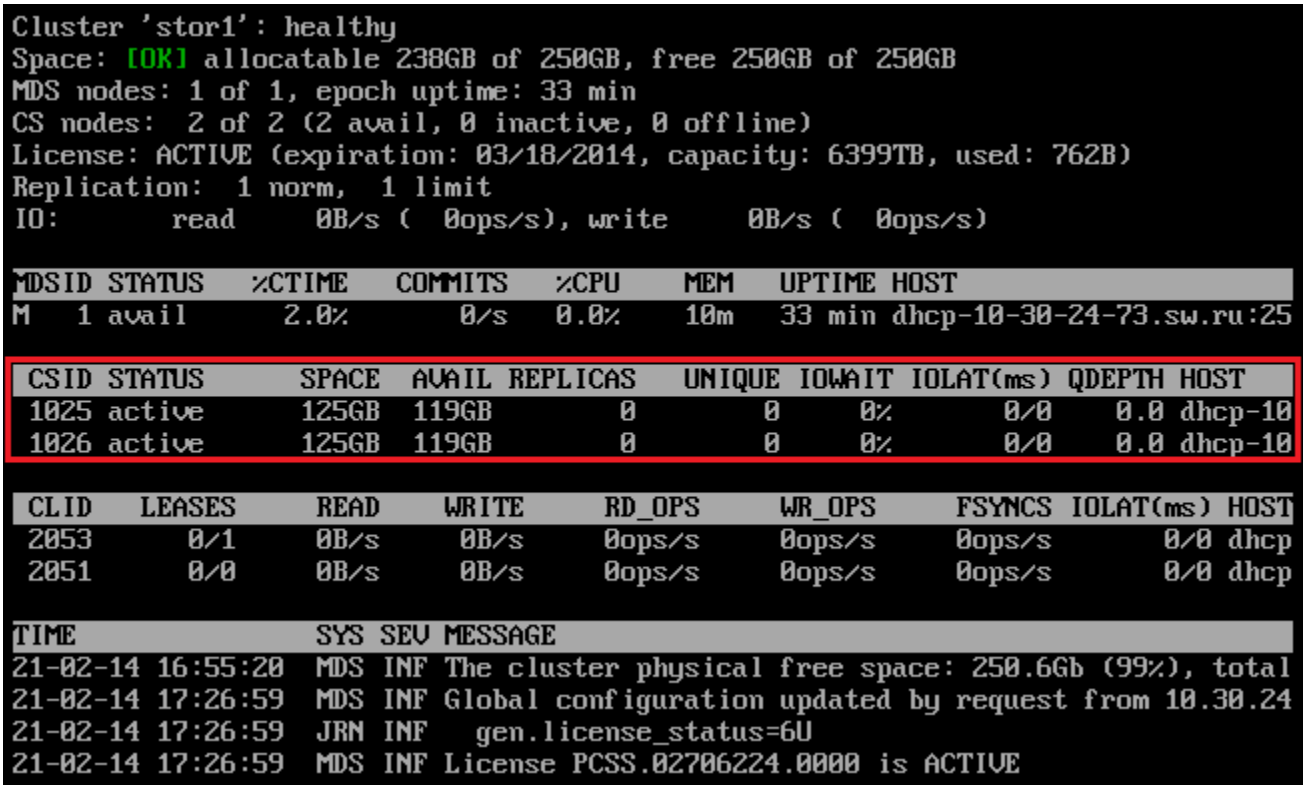


Приведенная выше команда показывает подробные сведения о кластере хранилища `stor1`. Параметры серверов метаданных для мониторинга (выделены красным) объяснены в таблице ниже.

Параметр	Описание
MDSID	Идентификатор сервера метаданных. Если перед идентификатором добавлена буква M, она указывает, что этот сервер является главным (master) сервером метаданных.
STATUS	Статус сервера метаданных.
%CTIME	Суммарное время, в течение которого сервер метаданных осуществлял запись в локальный журнал.
COMMITS	Интенсивность фиксаций в локальный журнал.
%CPU	Время активности сервера метаданных.
MEM	Объем физической памяти, используемой сервером метаданных.
UPTIME	Время, прошедшее с момента последнего запуска сервера метаданных.
HOST	Имя хоста или IP-адрес сервера метаданных.

Мониторинг серверов фрагментов данных

Посредством мониторинга серверов фрагментов данных (CS) вы можете отслеживать доступное дисковое пространство кластера хранилища. Для мониторинга серверов фрагментов данных используйте команду `vstorage -c <cluster_name> top`, например:



Приведенная выше команда показывает подробные сведения о кластере хранилища `stor1`. Параметры серверов фрагментов данных для мониторинга (выделены красным) объяснены в таблице ниже.

> **Примечание**
>
> Параметры отображаются группами. Для переключения между группами параметров нажимайте клавишу `i` на клавиатуре.

Параметр	Описание
CSID	Идентификатор сервера фрагментов данных.
STATUS	Статус сервера фрагментов данных: • `active`: Сервер фрагментов данных запущен и работает. • `inactive`: Сервер фрагментов данных временно недоступен. Сервер фрагментов данных помечается как неактивный в течение первых 5 минут неактивности. • `offline`: Сервер фрагментов данных неактивен в течение более чем 5 минут. Когда сервер фрагментов данных оказывается недоступен, кластер начинает репликацию данных, чтобы восстановить те фрагменты, которые хранились на затронутом сервере фрагментов данных. • `dropped`: Сервер фрагментов данных был удален администратором.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Параметр	Описание
SPACE	Общий объем дискового пространства на сервере фрагментов данных.
AVAIL	Объем доступного дискового пространства на сервере фрагментов данных.
REPLICAS	Количество реплик, сохраненных на сервере фрагментов данных.
IOWAIT	Процент времени, затраченный на ожидание выполнения операций ввода-вывода.
IOLAT	Среднее/максимальное время в миллисекундах, которое требовалось клиенту для выполнения одной операции ввода-вывода за последние 20 секунд.
QDEPTH	Средняя глубина очереди ввода-вывода на сервере фрагментов данных.
HOST	Имя хоста или IP-адрес сервера фрагментов данных.
SWAIT	Процент времени, затраченного на ожидание выполнения операций синхронизации данных.
SLAT	Средняя/максимальная задержка операций синхронизации.
RMW	Количество последовательностей ”чтение-модификация-запись”, возникающих из-за невыровненного ввода-вывода. Могут появляться при использовании старых гостевых ОС из-за невыровненных разделов. Последовательности такого рода снижают производительность.
JRMW	Количество последовательностей ”чтение-модификация-запись”, обработанных с помощью SSD-журнала. Если журнал не настроен, учитывается невыровненный ввод-вывод. В этом случае производительность будет зависеть от размера сектора физического жесткого диска.
READ	Текущая скорость чтения в байтах в секунду.
WRITE	Текущая скорость записи в байтах в секунду.
RD_OPS	Текущая скорость чтения в операциях в секунду.
WR_OPS	Текущая скорость записи в операциях в секунду.
MAP_OPS	Количество операций сопоставления за секунду.
SYNC	Количество операций синхронизации за секунду.
DATASYNC	Количество операций синхронизации данных за секунду.
HEALTHY	Количество фрагментов данных в состоянии `healthy`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.
DEGRDED	Количество фрагментов данных в состоянии `degraded`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.
URGENT	Количество фрагментов данных в состоянии `urgent`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.
BLOCKED	Количество фрагментов данных в состоянии `blocked`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Параметр	Описание
OFFLINE	Количество фрагментов данных в состоянии `offline`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.
REPL	Количество фрагментов данных в состоянии `replicating`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.
OVERCMT	Количество фрагментов данных в состоянии `overcommitted`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.
DELETNG	Количество фрагментов данных в состоянии `deleting`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.
VOID	Количество фрагментов данных в состоянии `void`, реплики которых размещены на сервере фрагментов данных. Подробные сведения см. в разделе *Состояния фрагментов данных*.
TIER	Уровень хранения данных, назначенный серверу фрагментов данных.
COST	Стоимость размещения фрагмента данных на сервере фрагментов данных.
ERR	Статус ошибки сервера фрагментов данных. Если значение не `None`, это означает, что сервер фрагментов данных не используется для размещения фрагментов данных в данный момент.
LAST_ERR	Предыдущий статус локальной ошибки сервера фрагментов данных и время, прошедшее с момента его наблюдения.
LAST_LINK_ERR	Предыдущий статус ошибки связи сервера фрагментов данных и время, прошедшее с момента его наблюдения.
JRN_FULL	Процент данных SSD-журнала, ожидающих сохранения на жесткий диск (чем меньше, тем лучше). 100 % означает переполнение.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Параметр	Описание
FLAGS	Для активных серверов фрагментов данных могут отображаться следующие флаги: • `J`: Сервер фрагментов данных использует журнал записи. • `C`: На сервере фрагментов данных включено вычисление контрольной суммы. Расчет контрольных сумм позволяет вам получать уведомление, если третья сторона изменит данные на диске. • `D`: Прямой ввод-вывод — нормальное состояние сервера фрагментов данных без журнала записи. • `c`: Журнал записи сервера фрагментов данных пуст: нет никаких данных, ожидающих фиксации с твердотельного накопителя журналирования записи на жесткий диск в месте расположения сервера фрагментов данных. Флаги, которые могут отображаться для неработоспособных серверов фрагментов данных, описаны в разделе *Неисправные серверы фрагментов данных*.
RETRANS	Общее количество повторных передач на хосте.
LAT_MAX	Максимальная задержка посреди всех серверов фрагментов данных на хосте.
LOCATION	Идентификатор месторасположения.
HOSTID	Идентификатор хоста.

Общие сведения об использовании дискового пространства

Обычно сведения о том, как используется дисковое пространство в вашем кластере хранилища, можно получить с помощью команды `vstorage top`. Эта команда выводит следующую информацию о дисках: общий объем, свободный объем и подлежащий выделению объем, например:

```
# vstorage -c stor1 top
connected to MDS#1
Cluster 'stor1': healthy
Space: [OK] allocatable 180GB of 200GB, free 1.6TB of 1.7TB
<...>
```

В выходных данных этой команды:

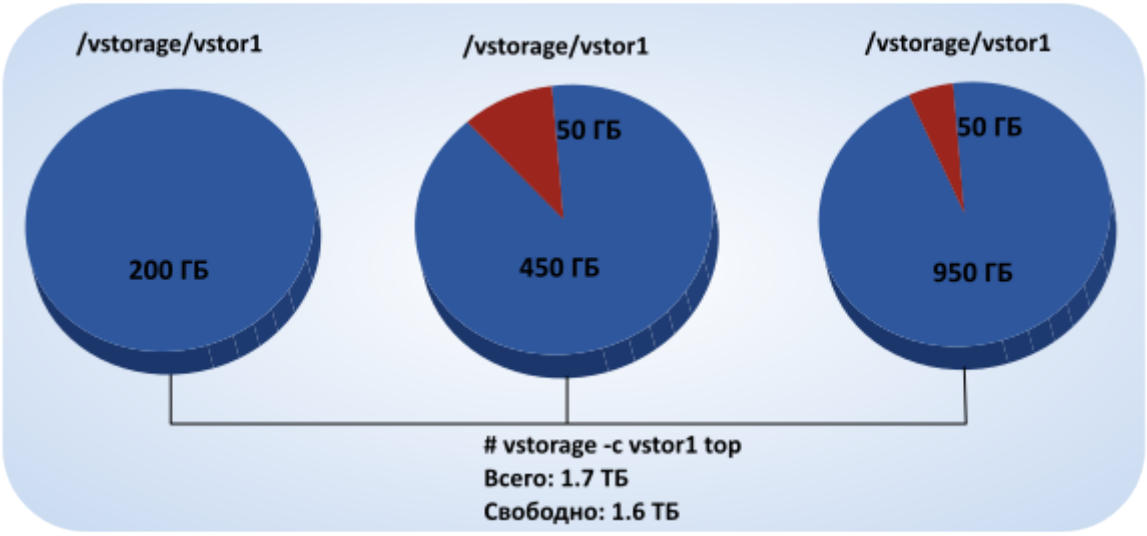
- 1,7TB — это общий объем дисков в кластере `stor1`. Общий объем дисков рассчитывается на основе всего используемого и свободного пространства на всех разделах в кластере. Используемое дисковое пространство включает пространство, занятое всеми фрагментами данных и их репликами, а также пространство, занятое любыми другими файлами, которые хранятся на разделах кластера.

  Допустим, что имеется раздел объемом в 100 ГБ и что 20 ГБ на этом разделе занято теми или

иными файлами. В случае если настроить на этом разделе сервер фрагментов данных, к кластеру будет добавлено 100 ГБ общего дискового пространства, хотя лишь 80 ГБ из этого пространства будет свободно и доступно для сохранения фрагментов данных.

- 1,6TB — это свободное дисковое пространство в кластере `stor1`. Свободное дисковое пространство рассчитывается посредством вычета объема дискового пространства, занятого фрагментами данных и любыми другими файлами на разделах кластера, из объема общего дискового пространства.

  Например, если объем свободного пространства составляет 1,6 ТБ, а общий объем дискового пространства равен 1,7 ТБ, это означает, что около 100 ГБ на разделах кластера уже занято теми или иными файлами.



- `allocatable 180GB of 200GB` (подлежат выделению 180 ГБ из 200 ГБ) — это объем свободного дискового пространства, который может использоваться для сохранения фрагментов данных. Более подробные сведения см. в разделе *Общие сведения о распределяемом дисковом пространстве*.

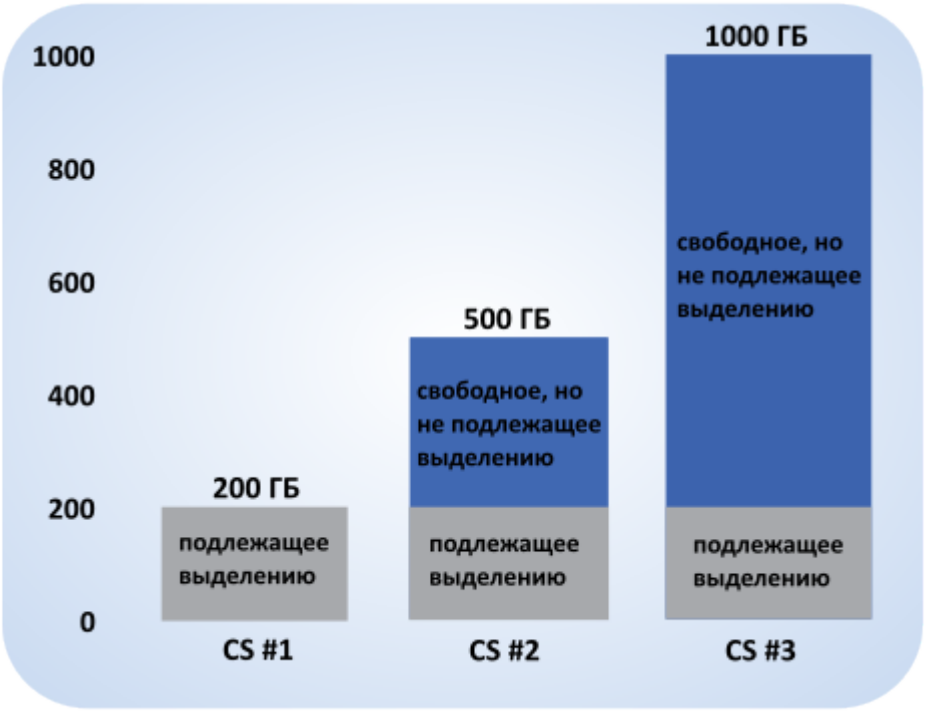
Общие сведения о распределяемом дисковом пространстве

Контролируя данные о дисковом пространстве в кластере хранилища, следует также обращать внимание на пространство, о котором утилита **`vstorage top`** сообщает как о подлежащем выделению (*allocatable*). Подлежащее выделению пространство — это объем дискового пространства, которое свободно и может использоваться для сохранения пользовательских данных. Когда это пространство закончится, записывать в кластер данные станет невозможно.

Расчет подлежащего выделению дискового пространства показан в следующем примере.

- Кластер содержит три сервера фрагментов данных. На первом сервере имеется 200 ГБ дискового пространства, на втором — 500 ГБ, а на третьем — 1 ТБ.

- В кластере используется коэффициент репликации по умолчанию 3, а это означает, что у каждого фрагмента данных должно быть три реплики, которые должны храниться на трех различных серверах фрагментов данных.



В этом примере доступное дисковое пространство составляет 200 ГБ, то есть равно объему дискового пространства наименьшего из серверов фрагментов данных.

```
# vstorage -c stor1 top
connected to MDS#1
Cluster 'stor1': healthy
Space: [OK] allocatable 180GB of 200GB, free 1.6TB of 1.7TB
<...>
```

В этой конфигурации кластера одна реплика каждого из фрагментов данных должна сохраняться на каждом из серверов. Таким образом, когда доступное дисковое пространство на наименьшем сервере фрагментов данных (200 ГБ) исчерпается, в кластере нельзя будет создавать новые фрагменты, пока в кластер не будет добавлен новый сервер фрагментов данных или не будет уменьшен коэффициент репликации.

Если изменить коэффициент репликации на 2, то команда `vstorage top` сообщит о доступном дисковом пространстве в 700 ГБ.

```
# vstorage set-attr -R /vstorage/stor1 replicas=2:1
# vstorage -c stor1 top
connected to MDS#1
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
Cluster 'stor1': healthy
Space: [OK] allocatable 680GB of 700GB, free 1.6TB of 1.7TB
<...>
```

Доступное дисковое пространство увеличилось, поскольку теперь для каждого фрагмента данных создаются только две реплики и можно создавать новые фрагменты, даже когда на наименьшем из серверов фрагментов данных закончится место (в этом случае реплики будут сохраняться на одном из больших серверов фрагментов данных).

Примечание

Доступное для выделения дисковое пространство также может ограничиваться лицензией.

Просмотр пространства, занятого фрагментами данных

Чтобы посмотреть общий объем дискового пространства, занимаемого всеми пользовательскими данными в кластере хранилища, выполните команду `vstorage top` и нажмите клавишу `V` на клавиатуре. После этого вывод команды будет выглядеть следующим образом:

```
# vstorage -c stor1 top
Cluster 'stor1': healthy
Space: [OK] allocatable 180GB of 200GB, free 1.6TB of 1.7TB
MDS nodes: 1 of 1, epoch uptime: 2d  4h
CS nodes:  3 of 3 (3 avail, 0 inactive, 0 offline)
Replication:  2 norm,  1 limit,  4 max
Chunks: [OK] 38 (100%) healthy,  0 (0%) degraded,  0 (0%) urgent,
             0 (0%) blocked,  0 (0%) offline,  0 (0%) replicating,
             0 (0%) overcommitted,  0 (0%) deleting,  0 (0%) void
FS:  1GB in 51 files, 51 inodes,  23 file maps,  38 chunks,  76 chunk replicas
<...>
```

Примечание

В поле **FS** отображается размер всех пользовательских данных в кластере хранилища без учета реплик.

Состояния фрагментов данных

В нижеприведенной таблице перечислены все возможные состояния, в которых могут находиться фрагменты данных.

Состояние	Описание
healthy	Количество и процент фрагментов данных, у которых достаточно активных реплик. Это нормальное состояние фрагментов.
replicating	Количество и процент фрагментов данных, для которых осуществляется репликация. Операции записи в эти фрагменты заморожены до момента окончания репликации.
offline	Количество и процент фрагментов данных, все реплики которых находятся в отключенном состоянии. Такие фрагменты полностью недоступны для кластера, невозможно их реплицировать, считывать или записывать в них данные. Все запросы к фрагменту, находящемуся в состоянии `offline`, замораживаются до тех пор, пока служба фрагментов данных (CS), хранящая реплику соответствующего фрагмента, не станет активной. Во избежание потери данных следует как можно быстрее вернуть службы фрагментов данных, находящиеся в состоянии `offline`, в подключенное состояние.
void	Количество и процент фрагментов данных, которые были выделены, но еще ни разу не использовались. Такие фрагменты не содержат данных. Наличие в кластере хранилища некоторого числа пустых фрагментов — это нормальная ситуация.
pending	Количество и процент фрагментов данных, которые необходимо реплицировать немедленно. Для выполнения запроса от клиента на запись во фрагмент у этого фрагмента должно быть не менее заданного минимального числа реплик. Если их меньше, фрагмент блокируется и выполнить запрос невозможно. Поскольку заблокированные фрагменты необходимо реплицировать как можно быстрее, кластер помещает их в отдельную, высокоприоритетную очередь репликации и сообщает о них как об ожидающих репликации.
blocked	Количество и процент фрагментов данных, у которых число активных реплик меньше заданного минимального количества. Запросы на запись в заблокированный фрагмент замораживаются до тех пор, пока у него не будет по крайней мере заданного минимального количества реплик. В то же время запросы на чтение из заблокированных фрагментов выполняются, поскольку у них еще есть активные реплики. Заблокированные фрагменты имеют более высокий приоритет репликации, чем деградировавшие. Наличие заблокированных фрагментов в кластере повышает риск потери данных, поэтому следует отложить все техническое обслуживание на рабочих серверах кластера и как можно быстрее вернуть недоступные серверы фрагментов данных в рабочее состояние.

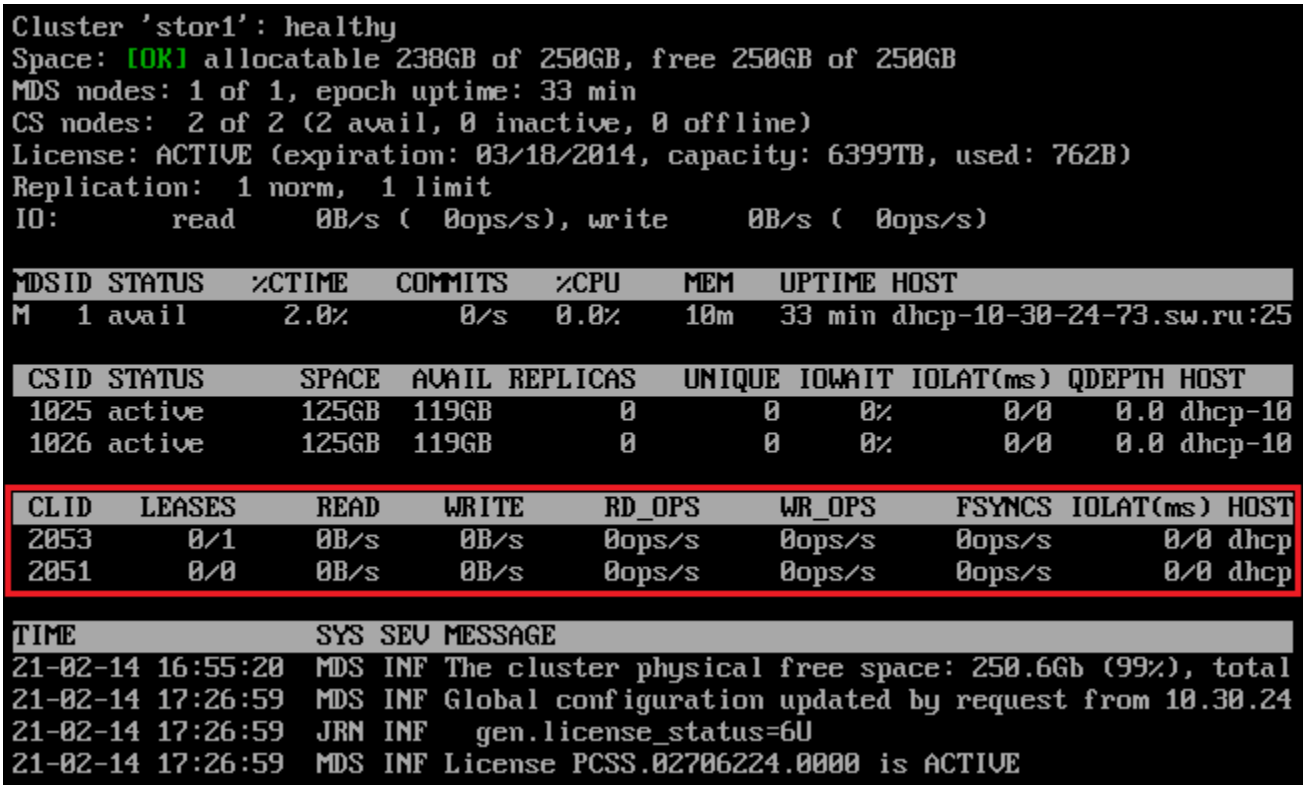
(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Состояние	Описание
degraded	Количество и процент фрагментов данных с небольшим числом активных реплик, но не меньше установленного минимума. Для таких фрагментов данных возможны и чтение, и запись в них. Однако в последнем случае деградировавший фрагмент данных становится срочным.
urgent	Количество и процент фрагментов данных, которые деградировали и имеют неидентичные реплики. Реплики деградировавшего фрагмента могут оказаться неидентичными, если некоторые из них окажутся недоступными во время операции записи. В таком случае часть реплик будет содержать новые данные, в то время как в других будут по-прежнему содержаться старые данные. Такие реплики удаляются кластером хранилища как можно быстрее. Срочные фрагменты не влияют на целостность информации, так как актуальные данные все равно хранятся не менее чем в заданном минимальном числе реплик.
standby	Количество и процент фрагментов данных, у которых одна или более реплик находятся в резервном состоянии. Реплика отмечается как резервная, если она была неактивна не более 5 минут.
overcommitted	Количество и процент фрагментов данных, у которых количество реплик больше нормального. Обычно такие фрагменты возникают после снижения нормального числа реплик или удаления большого объема данных. Со временем излишние реплики удаляются, однако во время репликации этот процесс может происходить медленнее.
deleting	Количество и процент фрагментов данных, которые должны быть удалены.

Мониторинг клиентов

Посредством мониторинга клиентов можно контролировать состояние и работоспособность серверов, используемых для доступа к данным кластера хранилища. Для мониторинга клиентов используйте команду `vstorage -c <cluster_name> top`, например:



Приведенная выше команда выводит подробные сведения о кластере `stor1`. Параметры мониторинга для клиентов (выделены красным) объяснены в таблице ниже.

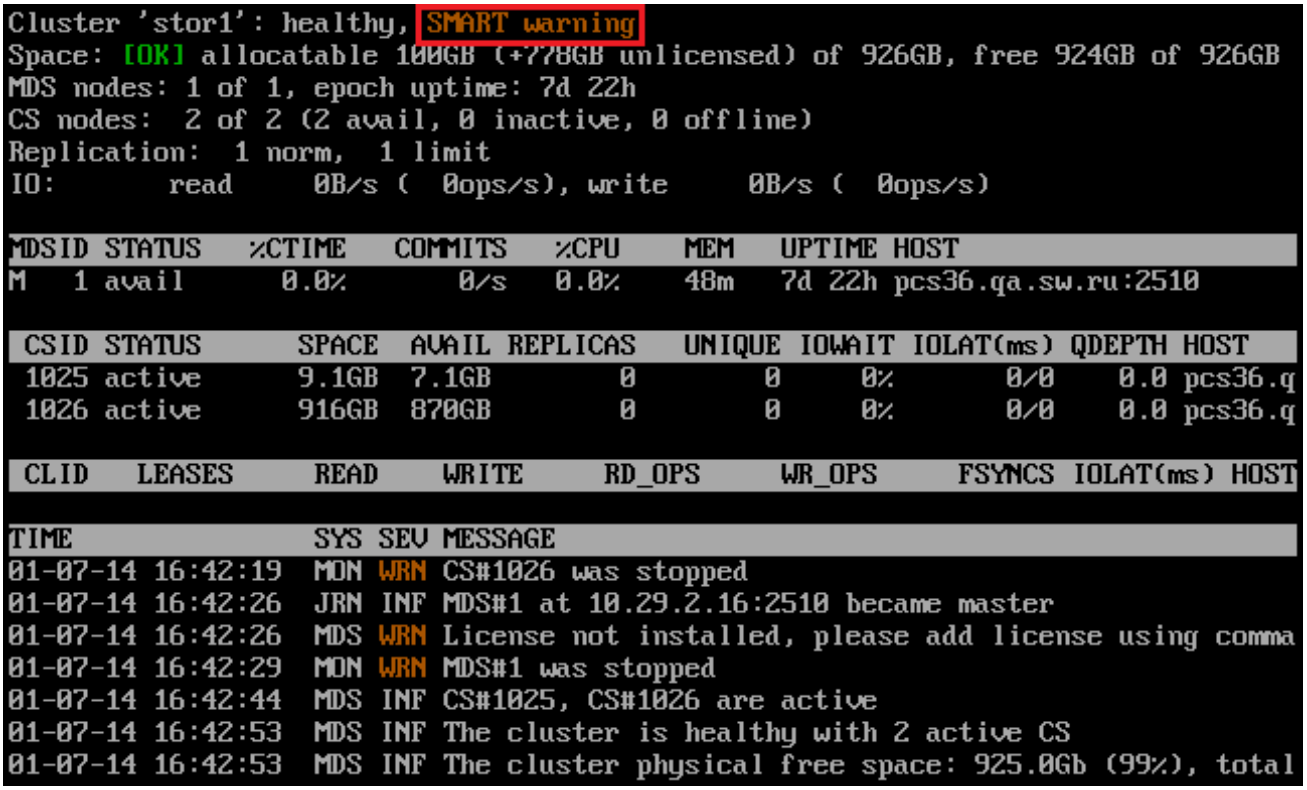
Параметр	Описание
CLID	Идентификатор клиента.
LEASES	Среднее количество файлов, открытых клиентом для чтения/записи и все еще не закрытых, за последние 20 секунд.
READ	Средняя скорость в байтах в секунду, с которой клиент считывает данные, за последние 20 секунд.
WRITE	Средняя скорость в байтах в секунду, с которой клиент записывает данные, за последние 20 секунд.
RD_OPS	Среднее число операций чтения, выполняемых клиентом за секунду, в течение последних 20 секунд.
WR_OPS	Среднее число операций записи, выполняемых клиентом за секунду, в течение последних 20 секунд.
FSYNCS	Среднее число операций синхронизации, выполняемых клиентом за секунду, в течение последних 20 секунд.
IOLAT	Среднее/максимальное время в миллисекундах, которое требовалось клиенту для выполнения одной операции ввода-вывода, за последние 20 секунд.
HOST	Имя хоста или IP-адрес клиента.

Мониторинг физических дисков

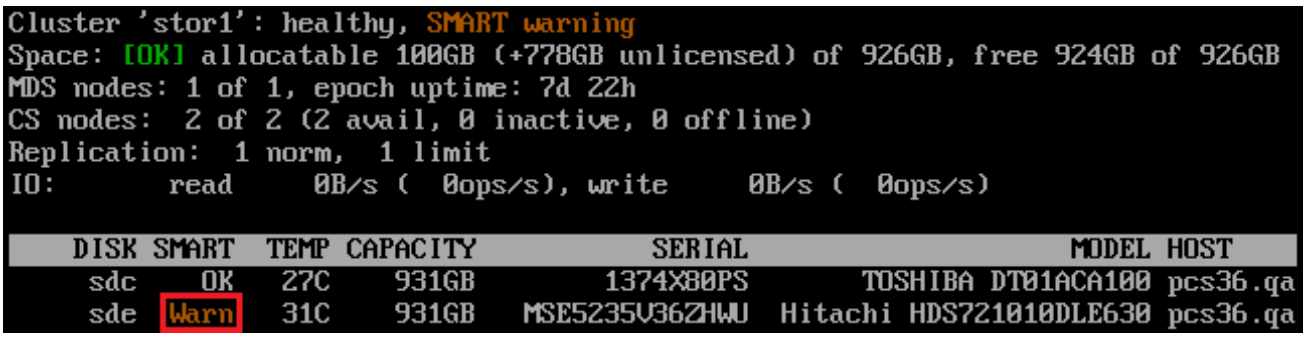
Состояние S.M.A.R.T. физических дисков отслеживается с помощью инструмента `smartctl`, устанавливаемого вместе с продуктом Кибер Хранилище. Инструмент запускается каждые 5 минут. Инструмент `smartctl` опрашивает все физические диски, подсоединенные к серверам в кластере, в том числе твердотельные накопители кэширования и журналирования, и передает полученные результаты на сервер метаданных.

> **Примечание**
>
> Чтобы инструмент `smartctl` работал, необходимо включить поддержку S.M.A.R.T. в BIOS соответствующего сервера.

Результаты опроса накопителей за последние 10 минут можно просмотреть в выходных данных команды `vstorage top`, например:



Если в главной таблице отображается сообщение `SMART warning` (предупреждение SMART), это означает, что для одного из физических дисков близок аппаратный отказ по данным S.M.A.R.T. Нажмите клавишу `d`, чтобы переключиться на таблицу дисков и просмотреть более подробные сведения, например:



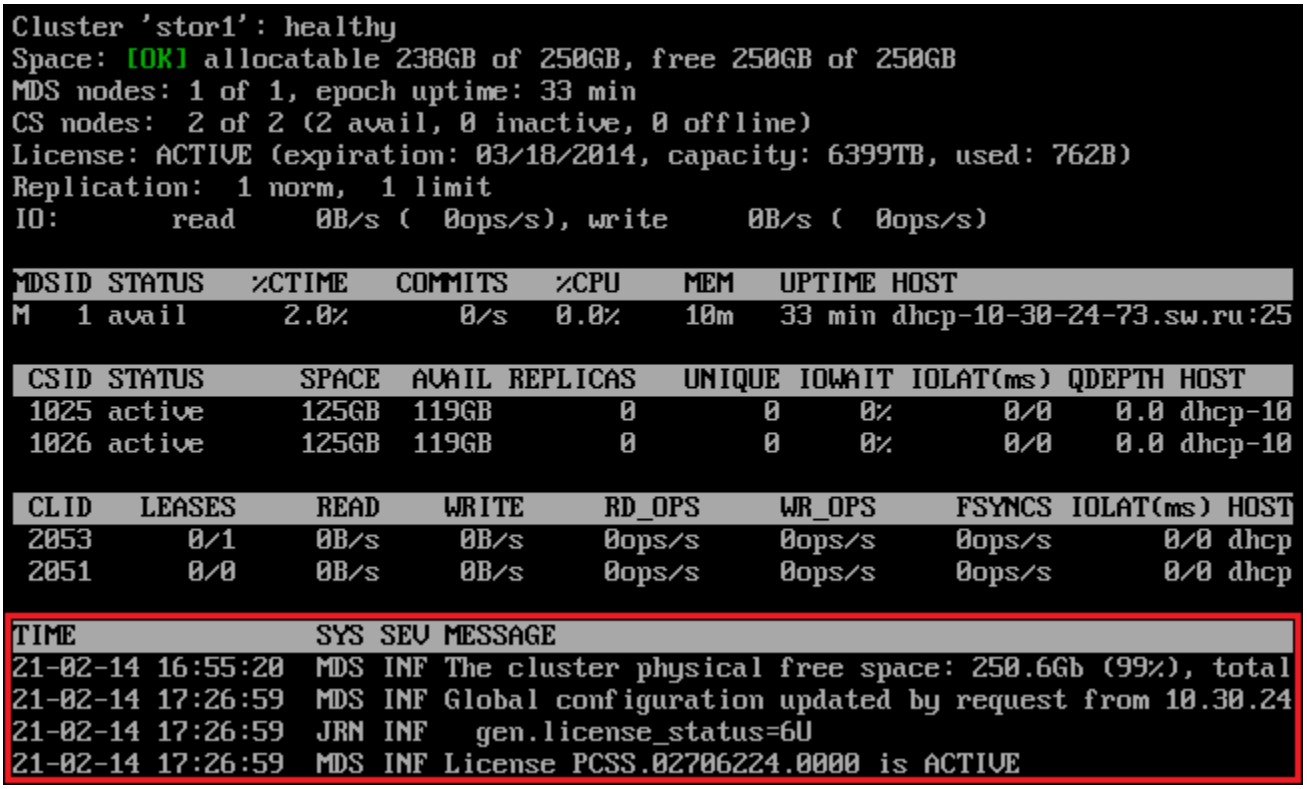
В таблице дисков отображаются следующие параметры:

Параметр	Описание
DISK	Имя диска, назначенное операционной системой.
SMART	Состояние S.M.A.R.T. диска: • `OK`: Диск исправен. • `Warn`: Диск в состоянии, близком к аппаратному отказу. Близость к аппаратному отказу означает, что по меньшей мере один из следующих счетчиков S.M.A.R.T. не равен нулю: • Reallocated Sector Count (Количество переназначенных секторов), • Reallocated Event Count (Количество событий переназначения), • Current Pending Sector Count (Текущее количество секторов, являющихся кандидатами на замену), • Offline Uncorrectable (Отключено без возможности исправления).
TEMP	Температура диска в градусах Цельсия.
CAPACITY	Емкость диска.
SERIAL	Серийный номер диска.
MODEL	Модель диска.
HOST	Имя хоста диска.

Чтобы отключить мониторинг S.M.A.R.T. дисков, остановите и выключите соответствующие службу и таймер systemd: `vstorage-send-diskinfo.service` и `vstorage-send-diskinfo.timer`.

Мониторинг журналов событий

Для мониторинга важных событий, происходящих в кластере хранилища, вы можете использовать команду `vstorage -c <cluster_name> top`, например:



Приведенная выше команда отображает последние события, которые произошли в кластере `stor1`. Сведения о событиях (выделены красным) отображаются в таблице со следующими столбцами:

Столбец	Описание
TIME	Время события.
SYS	Компонент кластера, в котором произошло событие (например, MDS для сервера метаданных или JRN для локального журнала).
SEV	Уровень серьезности события.
MESSAGE	Описание события.

Полный список событий можно получить с помощью команды `vstorage -c <cluster_name> get-event`, например:

```
# vstorage -c stor1 get-event
connected to MDS#4
2024-12-03 23:21:46.520 MDS WRN: Initial settings are 1 data replica by default. Please change to¬
↪more reliable 2 or 3 copies.
2024-12-03 23:21:48.962 JRN INF: MDS#1 at 192.168.40.11:2512 became master
2024-12-03 23:21:48.962 MDS WRN: License not installed, please add license using command load-
↪license of vstorage.
2024-12-03 23:21:48.962 JRN INF:   mds.alloc.strict_tier=1U
2024-12-03 23:21:51.497 MON INF: MDS#1 was started
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
2024-12-03 23:21:57.702 MON INF: CS#1025 was started
2024-12-03 23:21:58.668 MDS INF: New CS#1025 at 192.168.40.11:36456 (0.0.0.ea6dc0f7c69a4975),¬
↪tier=0
2024-12-03 23:22:02.740 MON INF: CS#1026 was started
<...>
```

Знакомство с основными событиями

В следующей таблице перечислены основные события, отображаемые при запуске утилиты **`vstorage top`**.

Таблица 5.8: Основные события

Событие	Серьезность	Описание
MDS#<N> (<addr>:<port>) lags behind for more than 1000 rounds (Сервер MDS № <N> отстает в течение более 1000 циклов)	JRN err (Ошибка журнала)	Создается главным сервером MDS, когда тот обнаруживает устаревание/отставание сервера MDS № <N>. Это сообщение может указывать на то, что какой-то сервер MDS работает очень медленно и не успевает обрабатывать операции.
MDS#<N> (<addr>:<port>) didn't accept commits for *M* sec (Сервер MDS № <N> не принимал фиксации в течение M сек.)	JRN err (Ошибка журнала)	Создается главным сервером MDS, если сервер MDS № <N> не принимал фиксаций в течение *M* сек. Такой сервер MDS № <N> помечается как устаревший. Это сообщение может указывать, что на сервере MDS № <N> возникли проблемы со службой MDS. Такая проблема может быть критической, и ее следует решить как можно скорее.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Событие	Серьезность	Описание
MDS#<N> (<addr>:<port>) state is outdated and will do a full resync (Состояние сервера MDS № <N> устарело, будет произведена полная повторная синхронизация)	JRN err (Ошибка журнала)	Создается главным сервером MDS, если на сервере MDS № <N> будет проведена полная повторная синхронизация. Такой сервер MDS № <N> помечается как устаревший. Это сообщение может указывать, что определенный сервер MDS работал слишком медленно или соединение с ним нарушилось на такое продолжительное время, что он уже не поддерживает актуального состояния метаданных и его необходимо синхронизировать повторно. Такая проблема может быть критической, и ее следует решить как можно скорее.
MDS#<N> at <addr>:<port> became master (Сервер MDS № <N> стал главным сервером)	JRN info (Информация в журнале)	Создается каждый раз, когда в кластере выбирается новый главный сервер MDS. Частые изменения главного сервера MDS могут указывать на проблемы с сетевыми подключениями и негативно влиять на работу кластеров.
The cluster is healthy with *N* active CS (Кластер работоспособен с N активных CS)	MDS info (Информация MDS)	Создается в случае, если состояние кластера меняется на работоспособное, и при выборе нового главного сервера MDS. Это сообщение указывает, что все серверы фрагментов в кластере активны и количество реплик соответствует установленным в кластере требованиям.
The cluster is degraded with *N* active, *M* inactive, *K* offline CS (Кластер деградировал: N активных серверов фрагментов, M неактивных, K отключены)	MDS warn (Предупреждение MDS)	Создается, когда состояние кластера меняется на деградировавшее, а также при выборе нового главного сервера MDS. Это сообщение указывает, что некоторые серверы фрагментов в кластере либо неактивны, то есть не посылают каких-либо сообщений о регистрации, либо отключены, то есть были неактивны дольше, чем задано в параметре `mds.wd.offline_tout` (по умолчанию 5 минут).

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Событие	Серьезность	Описание
The cluster failed with *N* active, *M* inactive, *K* offline CS (mds.wd.max_offline_cs=<n>) (Произошел отказ кластера: N активных серверов фрагментов данных, M неактивных, K отключены)	MDS err (Ошибка MDS)	Создается, когда состояние кластера меняется на отказ кластера, а также при выборе нового главного сервера MDS. Это сообщение указывает, что количество отключенных серверов фрагментов данных превышает число, заданное в параметре `mds.wd.max_offline_cs` (по умолчанию равно 2). В случае отказа кластера планирование автоматической репликации прекращается. Поэтому администратору кластера необходимо принять меры: либо восстановить отказавшие серверы кластера, либо увеличить значение `mds.wd.max_offline_cs`. При установке значения 0 этого параметра перехода в режим отказа не происходит никогда.
The cluster is filled up to <N>% (Кластер заполнен на <N> %)	MDS info/warn (Информация/предупреждение MDS)	Уведомляет о текущем использовании пространства в кластере. Если израсходовано 80 % и более дискового пространства, формируется предупреждение. Важно иметь некоторый резервный объем дискового пространства для реплик данных на случай отказа одного из серверов фрагментов данных.
Replication started, N chunks are queued (Репликация начата, в очередь помещено N фрагментов)	MDS info (Информация MDS)	Создается, когда кластер начинает автоматическую репликацию данных для восстановления отсутствующих реплик.
Replication completed (Репликация завершена)	MDS info (Информация MDS)	Создается, когда кластер завершает автоматическую репликацию данных.
CS#<N> has reported hard error on *path* (CS № <N> сообщил об аппаратной ошибке по пути *path*)	MDS warn (Предупреждение MDS)	Создается, когда CS № <N> обнаруживает повреждение данных на диске. Рекомендуется проверить оборудование на предмет ошибок и заменить диски с поврежденными данными как можно скорее.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Событие	Серьезность	Описание
CS#<N> has not registered during the last *T* sec and is marked as inactive/offline (CS № <N> не проходил регистрацию в течение последних T секунд и помечен как неактивный/отключенный)	MDS warn (Предупреждение MDS)	Создается, когда CS № <N> был недоступен в течение некоторого времени. Через 5 минут его состояние меняется на отключенный (offline), при этом запускается автоматическая репликация данных для восстановления реплик, хранившихся на сервере фрагментов данных, который оказался отключен.
Failed to allocate *N* replicas for '*path*' by request from <addr>:<port> - *K* out of *M* chunks servers are available (Не удалось распределить N реплик для пути 'path' по запросу от <addr>:<port> — K из M серверов фрагментов данных недоступны)	MDS warn (Предупреждение MDS)	Создается, когда кластер не может выделить пространство для реплик фрагментов, например в случае, если на нем исчерпалось дисковое пространство.
Failed to allocate *N* replicas for '*path*' by request from <addr>:<port> since only *K* chunk servers are registered (Не удалось распределить N реплик для пути 'path' по запросу от <addr>:<port>, поскольку зарегистрировано лишь K серверов фрагментов данных)	MDS warn (Предупреждение MDS)	Создается, когда кластер не может выделить пространство для реплик фрагментов, потому что в кластере зарегистрировано слишком мало серверов фрагментов данных.

Мониторинг параметров репликации

Настраивая параметры репликации, учитывайте, что новые параметры не вступают в силу немедленно. Например, при увеличении параметра количества реплик по умолчанию для фрагментов данных его отработка может занять некоторое время в зависимости от нового значения параметра и числа фрагментов данных в кластере.

Чтобы убедиться, что новые параметры репликации были успешно применены в кластере, выполните следующие действия:

1. Выполните команду `vstorage -c <cluster_name> top`.
2. Нажмите клавишу v, чтобы отобразить дополнительные сведения о кластере. Типичный вывод

этой команды может выглядеть следующим образом:

```
# vstorage -c stor1 top
connected to MDS#1
Cluster 'stor1': healthy
Space: [OK] allocatable 200GB of 211GB, free 211GB of 211GB
MDS nodes: 1 of 1, epoch uptime: 2h 21m
CS nodes:  2 of 2 (2 avail, 0 inactive, 0 offline)
License: PCSS.02444715.0000 is ACTIVE, 6399TB capacity
Replication:  3 norm,  2 limit
Chunks: [OK] 431 (100%) healthy,  0 (0%) degraded,  0 (0%) urgent,
             0 (0%) blocked,  0 (0%) offline,  0 (0%) replicating,
             0 (0%) overcommitted,  0 (0%) deleting,  0 (0%) void
<...>
```

3. Проверьте значение в поле `Chunks` (Фрагменты), учитывая следующие моменты:

   - При уменьшении параметров числа реплик обращайте внимание на фрагменты в состоянии фиксации с избытком (`overcommitted`) и в состоянии удаления (`deleting`). После завершения процесса репликации в выходных данных команды не должны отображаться фрагменты ни в одном из этих состояний.
   - При увеличении параметров количества реплик обращайте внимание на фрагменты в заблокированном состоянии (`blocked`) и подлежащие срочной обработке (`urgent`). После завершения процесса репликации в выходных данных команды не должны отображаться фрагменты ни в одном из этих состояний. Кроме того, пока процесс продолжает выполняться, значение параметра работоспособности (`healthy`) будет ниже 100 %.

> **Примечание**
>
> Дополнительные сведения о возможных состояниях фрагментов см. в разделе *Состояния фрагментов данных*.

Управление безопасностью кластера хранилища

В этом разделе описаны ситуации, которые могут повлиять на безопасность вашего кластера.

Вопросы безопасности

В этом разделе описаны ограничения безопасности, о которых следует помнить при развертывании кластера хранилища.

Перехват трафика

Продукт Кибер Хранилище не защищает вас от перехвата трафика. Любой, кто имеет доступ к вашей сети, может перехватывать и анализировать данные, отправляемые и получаемые по вашей сети.

Сведения о том, как обеспечить безопасность данных, см. в разделе *Защита связи между серверами кластера*.

Отсутствие пользователей и групп

Продукт Кибер Хранилище не использует концепцию пользователей и групп, в рамках которой определенным пользователям и группам предоставляется доступ к определенным частям кластера. Кто угодно, обладающий правом доступа к кластеру, может получить доступ ко всем его данным.

Незашифрованные данные на дисках

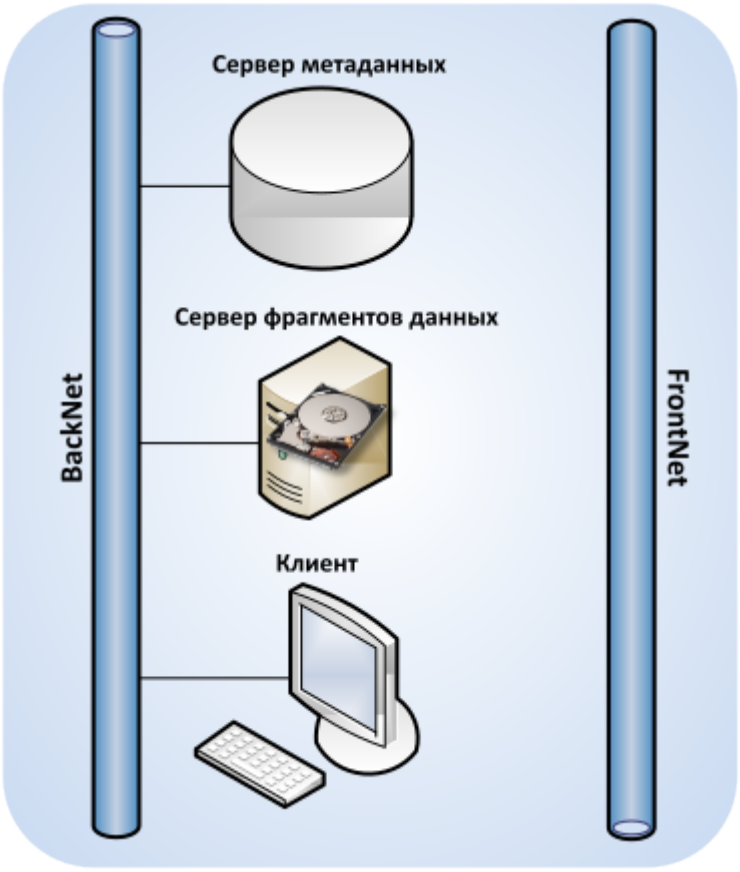
Продукт Кибер Хранилище не шифрует данные, хранящиеся в кластере. Злоумышленники могут сразу же увидеть данные, как только получат доступ к физическому диску.

Защита связи между серверами кластера

Кластер хранилища состоит из серверов трех типов:

- серверы метаданных (MDS),
- серверы фрагментов данных (CS),
- клиенты.

Во время работы кластера серверы взаимодействуют друг с другом. Чтобы защитить их связь, необходимо разместить все серверы в изолированной частной сети — BackNet. На рисунке ниже показан пример конфигурации кластера, в которой все серверы расположены в сети BackNet.



Для развертывания такой конфигурации выполните следующие шаги:

1. Создайте кластер, установив сервер метаданных и указав один из его IP-адресов:

   ```
   # vstorage -c Cluster-Name make-mds -I -a MDS-IP-Address -r Journal-Directory -p
   ```

   Указанный IP-адрес будет использоваться для связи между серверами метаданных, а также для связи с остальными серверами кластера.

2. Установите сервер фрагментов данных:

   ```
   # vstorage -c Cluster-Name make-cs -r CS-Directory
   ```

   После создания сервер фрагментов данных подключается к серверу метаданных и привязывается к тому IP-адресу, который он использует для установления соединения. Если сервер фрагментов данных имеет несколько сетевых карт, вы можете ему явно назначить IP-адрес определенной сетевой карты, чтобы все соединения между сервером фрагментов данных и серверами метаданных осуществлялись через этот IP-адрес.

   Чтобы привязать сервер фрагментов данных к пользовательскому IP-адресу, укажите параметр `-a` в команде `vstorage make-cs` при создании сервера фрагментов данных:

```
# vstorage make-cs -r CS-Directory -a Custom-IP-Address
```

> **Примечание**
>
> Пользовательский IP-адрес должен принадлежать диапазону адресов сети BackNet, чтобы не скомпрометировать безопасность кластера.

3. Смонтируйте кластер на клиенте:

```
# vstorage-mount -c Cluster-Name Mount-Directory
```

После того как кластер смонтирован, клиент подключается к серверам метаданных и фрагментов данных по их IP-адресам.

В этом примере конфигурации обеспечивается высокий уровень безопасности связи между серверами, поскольку сервер метаданных, сервер фрагментов данных и клиент находятся в изолированной сети BackNet и не могут быть скомпрометированы.

Порты, используемые продуктом Кибер Хранилище

В этой секции перечислены порты, которые должны быть открыты на серверах, входящих в состав кластера хранилища.

Серверы метаданных (MDS)

Следующие порты должны быть открыты на сервере метаданных:

- **Локальные порты**: 2510 для входящих подключений с других серверов метаданных и 2511 для входящих подключений с серверов фрагментов данных и клиентов.
- **Удаленные порты**: 2510 для исходящих подключений к другими серверам метаданных.

По умолчанию продукт Кибер Хранилище использует порт 2510 для связи между серверами метаданных и порт 2511 для входящих подключений с серверов фрагментов данных и клиентов. Вы можете переопределить порты по умолчанию при создании серверов метаданных:

1. Зарезервируйте два незанятых последовательных порта.

   Эти порты должны быть одинаковыми на всех серверах метаданных, которые вы планируете настроить в кластере.

2. Выполните команду `vstorage make-mds`, чтобы создать сервер метаданных, и укажите наименьший порт после IP-адреса сервера.

Например, чтобы настроить серверы метаданных в кластере `stor1` так, чтобы они использовали порты 4110 и 4111, вы можете выполнить следующую команду:

```
# vstorage -c stor1 make-mds -I -a 10.30.100.101:4110 -r /vstorage/stor1-mds -p
```

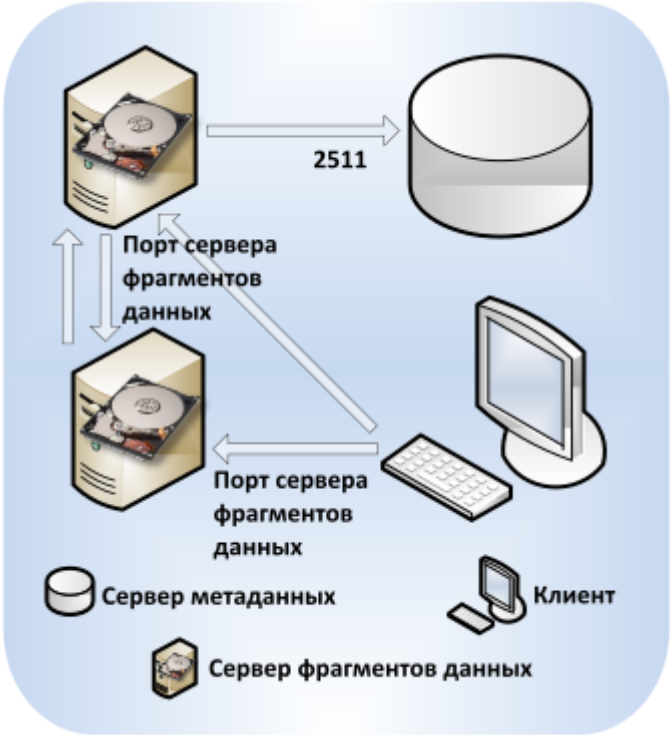
Если вы решили использовать пользовательские порты 4110 и 4111, сделайте следующее:

- На каждом сервере метаданных с пользовательскими портами откройте порты 4110 и 4111 для входящего трафика и порт 4110 для исходящего трафика.
- На всех серверах фрагментов данных и клиентах кластера откройте порт 4111 для исходящего трафика.

Серверы фрагментов данных (CS)

Следующие порты должны быть открыты на сервере фрагментов данных:

- **Локальные порты**: произвольно выбранный порт для входящих подключений с клиентов и других серверов фрагментов данных.
- **Удаленные порты**: 2511 для исходящих подключений к серверам метаданных и произвольно выбранный порт для исходящих подключений к другим серверам фрагментов данных.



Для службы управления сервером фрагментов данных необходимы:

- порт для связи с серверами метаданных (либо порт по умолчанию 2511, либо пользовательский порт);
- порт для связи с серверами фрагментов данных и клиентами.

По умолчанию эта служба использует любой доступный порт. Вы можете вручную переопределить этот порт при создании сервера фрагментов данных, указав параметр `-a` в команде `vstorage make-cs`. Например, чтобы настроить службу управления так, чтобы она использовала порт 3000, выполните следующую команду:

```
# vstorage make-cs -r /vstorage/stor1-cs -a 132.132.1.101:3000
```

Как только вы зададите пользовательский порт, откройте его для исходящего трафика на всех клиентах и других серверах фрагментов данных кластера.

Клиенты

Следующие порты должны быть открыты на клиентах:

- **Локальные порты**: нет.
- **Удаленные порты**: 2511 для исходящих подключений к серверам метаданных и порт, используемый серверами фрагментов данных, для исходящих подключений к серверам фрагментов данных.

По умолчанию продукт Кибер Хранилище автоматически открывает на клиенте следующие удаленные порты:

- порт по умолчанию 2511 для связи с серверами фрагментов данных;
- порт, используемый серверами фрагментов данных, для связи с серверами фрагментов данных.

Если вы задали пользовательские порты для связи с серверами метаданных и серверами фрагментов данных, откройте эти порты на клиенте для исходящего трафика. Например, если вы настроили порт 4111 на серверах метаданных и порт 3000 на серверах фрагментов данных, откройте удаленные порты 4111 и 3000 на клиенте.

Аутентификация на основе пароля

Продукт Кибер Хранилище использует аутентификацию на основе пароля для повышения безопасности кластеров хранилища. Аутентификация с использованием пароля является обязательной, то есть вы должны пройти этап аутентификации, прежде чем сможете добавить новый сервер в кластер.

Аутентификация на основе пароля работает следующим образом:

1. Вы задаете пароль для аутентификации при создании первого сервера метаданных (MDS) в кластере. Указанный вами пароль шифруется и сохраняется в файле `/etc/vstorage/clusters/stor1/auth_digest.key` на сервере.
2. Вы добавляете новые серверы метаданных, серверы фрагментов данных или клиентов в кластер и используете команду `vstorage auth-node` для их аутентификации. При аутентификации используется пароль, заданный при создании первого сервера метаданных.
3. Кластер хранилища сравнивает предоставленный пароль с паролем, хранящимся на первом сервере метаданных, и, если пароли совпадают, успешно аутентифицирует сервер.

Для каждого физического сервера аутентификация выполняется один раз. После аутентификации сервера в кластере (например, когда вы настраиваете его как сервер метаданных) на нем создается файл `/etc/vstorage/clusters/stor1/auth_digest.key`. Когда вы настраиваете этот сервер в качестве другого компонента кластера (например, в качестве сервера фрагментов данных), кластер проверяет наличие файла `auth_digest.key` и не требует повторной аутентификации сервера.

Повышение производительности кластера хранилища

В этом разделе описаны рекомендуемые конфигурации кластеров хранилища и способы их настройки для достижения максимальной производительности.

Также см. раздел *Приложение А. Устранение неполадок*, где описаны общие проблемы, которые могут повлиять на производительность кластера.

Проверка функций сброса данных на диски

Перед созданием кластера рекомендуется проверить, что все устройства хранения данных (жесткие диски, твердотельные накопители, RAID-массивы и т. д.), которые вы планируете включить в кластер, могут успешно сбрасывать данные на диск при незапланированном отключении питания сервера. Это поможет вам обнаружить устройства, которые могут потерять данные, хранящиеся в их кэше, в случае отключения питания.

Продукт Кибер Хранилище поставляется с утилитой `vstorage-hwflush-check`, которая проверяет, как устройство хранения сбрасывает данные на диск в аварийной ситуации. Утилита может работать в режиме клиента или сервера.

- **Клиент**. Клиент непрерывно записывает блоки данных на устройство хранения. После записи блока данных клиент увеличивает значение специального счетчика и отправляет его на сервер для сохранения.
- **Сервер**. Сервер отслеживает значения счетчика, получаемые от клиента, и всегда знает следующее значение. Если на сервер приходит меньшее значение счетчика, чем уже существующее (например, когда из-за сбоя питания устройство хранения не сбросило кэшированные данные на диск), то сервер сообщает об ошибке.

Чтобы убедиться, что устройство хранения успешно сбрасывает данные на диск при сбое питания, выполните следующую процедуру.

Запуск утилиты в качестве сервера:

1. Установите утилиту **`vstorage-hwflush-check`** на компьютер, который будет использоваться для запуска утилиты в качестве сервера. Эта утилита входит в пакет `vstorage-ctl` и может быть установлена с помощью следующей команды:

```
# yum install vstorage-ctl
```

2. Запустите утилиту **`vstorage-hwflush-check`** в качестве сервера:

```
# vstorage-hwflush-check -l
```

Запуск утилиты в качестве клиента:

1. На компьютере, устройство хранения которого вы хотите проверить, установите утилиту **`vstorage-hwflush-check`**:

```
# yum install vstorage-ctl
```

2. Запустите утилиту **`vstorage-hwflush-check`** в качестве клиента, например:

```
# vstorage-hwflush-check -s vstorage1.example.com -d /vstorage/stor1-ssd/test -t 50
```

   В этой команде:

   - `-s vstorage1.example.com` — имя компьютера, где утилита **`vstorage-hwflush-check`** запущена в качестве сервера.
   - `-d /vstorage/stor1-ssd/test` — каталог для тестирования сброса данных. Во время выполнения клиент создает в этом каталоге файл, в который записывает блоки данных.
   - `-t 50` — количество потоков для записи клиентом данных на диск. У каждого потока есть собственный файл и счетчик. Можно увеличить количество потоков (до 200), чтобы протестировать систему в более сложных условиях. Также можно указать другие параметры при запуске клиента. Дополнительные сведения о доступных параметрах см. на справочной странице `vstorage-hwflush-check`.

3. Подождите как минимум 10–15 секунд, отключите питание компьютера, на котором утилита запущена в качестве клиента, а затем снова включите.

> **Примечание**
>
> Кнопка **Reset** не отключает питание, поэтому вам нужно нажать кнопку **Power** или отсоединить шнур, чтобы выключить компьютер.

4. Запустите утилиту в качестве клиента еще раз, выполнив ту же самую команду, которую вы использовали для первого запуска утилиты:

```
# vstorage-hwflush-check -s vstorage1.example.com -d /vstorage/stor1-ssd/test -t 50
```

После запуска клиент прочитает все ранее записанные данные, определит версию данных на диске и перезапустит тестирование с последнего действительного значения счетчика. Затем он отправит это значение на сервер, а сервер сравнит его с последним, полученным ранее. Будут выведены данные вида:

```
id<N>:<counter_on_disk> -> <counter_on_server>
```

В зависимости от выведенных данных возможны следующие варианты:

- Если значение счетчика на диске меньше значения на сервере, то устройству хранения не удалось сбросить данные на диск. Это устройство лучше не использовать в производственной среде, особенно для служб CS или журналов, поскольку вы рискуете потерять данные.
- Если значение счетчика на диске больше значения на сервере, то устройство хранения сбросило данные на диск, но клиенту не удалось сообщить об этом серверу. Возможно, что скорость сети недостаточна либо устройство хранения работает слишком быстро для заданного количества потоков и следует увеличить их количество. Это устройство хранения можно использовать в производственной среде.
- Если значения счетчиков равны, то устройство хранения сбросило данные на диск и клиент сообщил об этом серверу. Это устройство хранения можно использовать в производственной среде.

На всякий случай повторите процедуру несколько раз. Проверив первое устройство хранения, выполните проверку всех устройств, которые планируется использовать в кластере. Необходимо протестировать все устройства: твердотельные накопители, используемые для журналов служб CS, диски, используемые для хранения данных служб MDS, а также диски, используемые для хранения данных служб CS.

Возможные конфигурации дисковых накопителей

Если к серверам, которые вы планируете включить в кластер хранилища, подключены несколько дисков, вы можете выбрать одну из следующих конфигураций:

1. **(Рекомендуется для конфигураций с твердотельными накопителями) Конфигурация без использования локальных RAID-массивов.**

   Служба фрагментов данных устанавливается для каждого жесткого диска, а каждый фрагмент данных имеет две или более реплики. Такая конфигурация обеспечивает защиту от отказов дисков, а репликация добавляет надежность, близкую к схеме RAID 1 (зеркалирование без

контроля четности или чередования). Настоятельно рекомендуется использовать твердотельные накопители для хранения журналов служб фрагментов данных, так как это уменьшает задержки фиксации записи (например, для баз данных).

2. **(Рекомендуется для конфигураций со всеми типами накопителей) Конфигурация с отдельными дисками, подключенными к аппаратному контроллеру RAID (прямое подключение, без использования RAID-массивов уровня 0, 1, 10, 5 или 6).**

   Эта настоятельно рекомендуемая конфигурация похожа на предыдущую, но работает гораздо быстрее. В ней отдельные диски подключены к аппаратному RAID-контроллеру с кэшем типа write-back и блоком резервного питания (BBU). Кэш RAID-контроллера значительно повышает скорость случайных операций ввода-вывода, а также производительность баз данных. Использование твердотельных накопителей еще больше оптимизирует случайный ввод-вывод, особенно операции чтения.

3. **(Не рекомендуется) Конфигурация с локальными RAID-массивами уровня 0 для фрагментов данных с двумя или более репликами.**

   Такая конфигурация снижает надежность кластера, поскольку отказ одного диска приводит к отказу всего массива RAID 0, что вынуждает кластер реплицировать больше данных при каждом таком сбое. Однако это можно считать незначительной проблемой, поскольку кластеры хранилища выполняют параллельное восстановление с нескольких серверов, что гораздо быстрее, чем восстановление традиционного RAID 1.

   Использование твердотельных накопителей для кэширования и журналирования еще больше повысит общую производительность кластера и обеспечит вычисление контрольных сумм и проверку данных для повышения надежности кластера.

4. **(Не рекомендуется) Конфигурация с локальными RAID-массивами уровня 1 для фрагментов данных с одной или несколькими репликами.**

   Конфигурация с одной репликой на каждый фрагмент данных не обеспечивает высокую доступность кластера в случае отказа нескольких серверов. Кроме того, такие конфигурации не дают экономии дискового пространства, поскольку они эквивалентны зеркалированию в кластере, а локальный RAID-массив просто удваивает коэффициент репликации данных и экономит сетевой трафик кластера.

5. **(Категорически не рекомендуется) Конфигурация с локальными RAID-массивами уровня 1, 5 или 6 для фрагментов данных с двумя или более репликами.**

   Избегайте использования для кластера хранилища массивов RAID, обеспечивающих избыточность данных (например, RAID 1, 5 или 6). В этом случае одна операция записи может затронуть значительное количество жестких дисков, что приведет к очень низкой производительности. Например, для трех реплик и RAID 5 на серверах с 5 жесткими дисками (на

каждом) одна операция записи может привести к 15 операциям ввода-вывода.

Проверка производительности

При тестировании производительности кластера хранилища, управляемого с помощью продукта Кибер Хранилище, и сравнении с аналогичными продуктами учтите следующее:

- Необходимо сравнивать конфигурации с аналогичными уровнями избыточности. Например, некорректно сравнивать производительность кластера с двумя или тремя репликами в расчете на один фрагмент данных с одним сервером, который не обеспечивает какой-либо избыточности данных наподобие RAID 1, 10 или 5.
- Примите во внимание использование интерфейсов файловой системы. Помните, что монтирование кластера хранилища с помощью интерфейса FUSE обеспечивает удобное представление данных кластера, но не является оптимальным с точки зрения производительности. Из-за этого следует выполнять тесты производительности внутри виртуальных машин.
- Помните, что значение коэффициента репликации данных влияет на производительность кластера: кластеры с двумя репликами работают немного быстрее, чем кластеры с тремя репликами.

Использование сетей 1 GbE и 10 GbE

Сеть 1 GbE может обеспечить передачу данных со скоростью 110-120 МБ/с, что сопоставимо с производительностью одного диска при последовательном вводе-выводе. Поскольку несколько дисков на одном сервере могут обеспечить более высокую пропускную способность, чем один канал сети 1 GbE, сеть может стать узким местом.

Однако в реальных приложениях и виртуализированных средах последовательный ввод-вывод встречается нечасто (в основном при резервном копировании), а большинство операций ввода-вывода носят случайный характер. Поэтому типичная пропускная способность диска обычно гораздо ниже (около 10-20 МБ/с) согласно статистике, накопленной на сотнях серверов ряда крупных хостинговых компаний.

Исходя из этих двух наблюдений, мы рекомендуем использовать одну из следующих сетевых конфигураций (или лучше):

- Один канал с пропускной способностью 1 Гбит/с на каждые два жестких диска сервера. Даже если на сервере установлено один или два диска, для большей надежности все равно рекомендуется использовать два сетевых адаптера (см. раздел *Настройка объединения сетевых интерфейсов*).
- Один канал с пропускной способностью 10 Гбит/с на один сервер для достижения максимальной

производительности.

В таблице ниже показано, как эти рекомендации могут применяться к серверу с 1-6 жесткими дисками:

Количество жестких дисков	Количество каналов 1 GbE	Количество каналов 10 GbE
1	1 (2 для высокой доступности)	1 (2 для высокой доступности)
2	1 (2 для высокой доступности)	1 (2 для высокой доступности)
3	2	1 (2 для высокой доступности)
4	2	1 (2 для высокой доступности)
5	3	1 (2 для высокой доступности)
6	3	1 (2 для высокой доступности)

Обратите внимание на следующее:

- Для достижения максимальной производительности последовательного ввода-вывода рекомендуется использовать один канал со скоростью 1 Гбит/с в расчете на один жесткий диск или один канал со скоростью 10 Гбит/с в расчете на один сервер.
- Не рекомендуется устанавливать для сетевых адаптеров со скоростью 1 Гбит/с нестандартные значения MTU (например, 9000-байтные jumbo-кадры). Такие параметры требуют дополнительной настройки коммутаторов и часто приводят к пользовательским ошибкам. Сетевые адаптеры со скоростью 10 Гбит/с, напротив, следует настроить на использование jumbo-кадров для достижения максимальной производительности.
- Для достижения максимальной эффективности используйте режим сетевого объединения `balance-xor` с политикой хэширования `layer3+4`. Если вы хотите использовать режим сетевого объединения `802.3ad`, также настройте ваш коммутатор для использования политики хэширования `layer3+4`.

Настройка объединения сетевых интерфейсов

Объединение нескольких сетевых интерфейсов дает следующие преимущества:

1. Высокая доступность сети. В случае отказа одного из интерфейсов трафик будет автоматически перенаправлен через один или несколько работающих интерфейсов.
2. Повышенная производительность сети. Например, два объединенных интерфейса Gigabit Ethernet обеспечат пропускную способность около 1,7 Гбит/с (200 МБ/с). Для сервера необходимое количество объединяемых сетевых интерфейсов может зависеть от количества дисков. Например, жесткий диск может выдавать данные со скоростью до 1 Гбит/с.

Чтобы настроить объединение сетевых интерфейсов, сделайте следующее:

1. Создайте файл `/etc/modprobe.d/bonding.conf`, содержащий следующую строку:

```
alias bond0 bonding
```

2. Создайте файл `/etc/sysconfig/network-scripts/ifcfg-bond0`, содержащий следующие строки:

```
DEVICE=bond0
ONBOOT=yes
BOOTPROTO=none
IPV6INIT=no
USERCTL=no
BONDING_OPTS=”mode=balance-xor xmit_hash_policy=layer3+4 miimon=300 downdelay=300 \
updelay=300”
NAME=”Storage net0”
NM_CONTROLLED=yes
IPADDR=xxx.xxx.xxx.xxx
PREFIX=24
```

   Удостоверьтесь, что параметрам `IPADDR` и `PREFIX` назначены правильные значения.

   Режим `balance-xor` является рекомендуемым, потому что он обеспечивает как отказоустойчивость, так и лучшую производительность. Дополнительные сведения см. в документах, приведенных ниже.

3. Удостоверьтесь, что конфигурационный файл каждого сетевого интерфейса, который вы хотите использовать в объединении сетевых интерфейсов (например, `/etc/sysconfig/network-scripts/ifcfg-eth0`), содержит строки, показанные в этом примере:

```
DEVICE=”eth0”
BOOTPROTO=none
NM_CONTROLLED=”yes”
ONBOOT=”yes”
TYPE=”Ethernet”
HWADDR=xx:xx:xx:xx:xx:xx
MASTER=bond0
SLAVE=yes
USERCTL=no
```

4. Активируйте сетевой интерфейс `bond0`:

```
# ifup bond0
```

5. Используйте вывод команды `dmesg`, чтобы проверить, что сетевой интерфейс `bond0` и его подчиненные сетевые интерфейсы активны и подключены.

Примечание

Подробные сведения об объединениях сетевых интерфейсов см. в документации ОС Red Hat Enterprise Linux и в статье о настройке объединений сетевых интерфейсов в ОС Linux.

Использование дисков SSD

Продукт Кибер Хранилище поддерживает диски SSD, на которых создана файловая система ext4. Функцию TRIM можно включить при монтировании при необходимости.

Примечание

Сценарий использования продуктом Кибер Хранилище накопителя SSD не запрашивает выполнения команд TRIM. Кроме того, современные накопители, такие как Intel SSD DC S3700, вообще не нуждаются в команде TRIM.

Наряду с конфигурациями All-Flash, в которых диски SSD используются для хранения фрагментов данных, продукт Кибер Хранилище также поддерживает гибридные кластеры, в которых диски SSD используются для журналирования записи. Диск SSD можно подключить к серверу фрагментов данных, использующего жесткие диски для хранения фрагментов данных, и настроить для хранения журналов записи, повысив производительность операций записи в кластере до двух и более раз.

В следующих разделах представлены подробные сведения о настройке дисков SSD для журналирования записи и кэширования данных.

Обратите внимание на следующее:

- Не все твердотельные накопители подчиняются семантике сброса и фиксируют данные в соответствии с протоколом. Это может привести к произвольной потере или повреждению данных в случае сбоя питания. Всегда проверяйте твердотельные накопители с помощью утилиты **`vstorage-hwflush-check`** (дополнительные сведения см. в разделе *Проверка функций сброса данных на диски*).
- Рекомендуется использовать накопители Intel SSD DC S3700. Однако допустимо использование Samsung SM1625, Intel SSD 710, Kingston SSDNow E или любого другого SSD-накопителя с поддержкой защиты данных при отключении питания. Некоторые названия данной технологии: Enhanced Power Loss Data Protection (Intel), Cache Power Protection (Samsung), Power-Failure Support (Kingston), Complete Power Fail Protection (OCZ). Дополнительные сведения см. в разделе *Диски SSD не сбрасывают данные из кэша*.

Расчет размера журнала записи

Использование SSD-накопителей для хранения журналов записи поможет сократить задержки записи, тем самым повысив общую производительность кластера хранилища. При наличии нескольких служб фрагментов данных (CS) на одном сервере, для каждой службы CS создайте отдельный журнал на SSD-накопителе.

Чтобы определить размер каждого журнала службы CS, размещаемого на SSD-накопителе, следуйте приведенным ниже рекомендациям.

1. Узнайте, сколько жестких дисков может обслуживать твердотельный накопитель, исходя из *требований к оборудованию*. Также можно использовать данную формулу:

   ```
   SSD_SSWS * 0.8 / HDD_SSWS
   ```

   В этой формуле:

   - `SSD_SSWS` — это устойчивая скорость последовательной записи на диск SSD.
   - `HDD_SSWS` — это устойчивая скорость последовательной записи на жесткий диск (при условии, что в соответствии с рекомендациями используются идентичные модели жестких дисков).
   - `0,8` — приблизительный процент погрешности.

   Примечание

   Устойчивая скорость последовательной записи (SSWS) — это средняя скорость последовательной записи, измеренная в течение 60 секунд.

2. Для полноценной работы SSD-накопителя 20 % его объема должны быть свободными и использоваться для обычной работы, хранения контрольных сумм и метаданных при необходимости. Оставляйте 1 ГБ свободного объема на диске SSD для контрольных сумм на каждый 1 ТБ объема на жестком диске. Избегайте полного заполнения SSD-накопителя, иначе его производительность ухудшится.
3. Разделите оставшиеся 80 % объема диска SSD на допустимое количество жестких дисков.

Например, диск SSD на 512 ГБ со скоростью SSWS 1 500 МБ/с сможет обслуживать 1 500 * 0,8 / 150 = 8 жестких дисков со скоростью SSWS 150 МБ/с. А размер журнала для каждого жесткого диска будет (512 - 512 * 0,2) / 8 = 51 ГБ.

Примечание

Удостоверьтесь, что ваше оборудование соответствует *требованиям к оборудованию*.

В следующей таблице приведены несколько примеров моделей твердотельных накопителей и количество жестких дисков, которые они могут обслуживать:

Тип твердотельного накопителя	Количество твердотельных накопителей
Твердотельный накопитель Intel серии 320, твердотельный накопитель Intel серии 710, корпоративная серия Kingston SSDNow E или другие модели твердотельных накопителей SATA 3 Гбит/с, обеспечивающие последовательную запись произвольных данных со скоростью 150–200 МБ/с.	1 накопитель SSD на 3 жестких диска
Твердотельные накопители Intel серии DC S3700, корпоративные серии Samsung SM1625 или другие модели твердотельных накопителей SATA 6 Гбит/с, обеспечивающие последовательную запись произвольных данных со скоростью не менее 300 МБ/с.	1 накопитель SSD на 5-6 жестких дисков

Установка сервера фрагментов данных с журналом на SSD-диске

Чтобы установить сервер фрагментов данных (CS), который хранит журнал на диске SSD, сделайте следующее:

1. Войдите на компьютер, который вы хотите настроить в качестве сервера фрагментов данных, как администратор или как пользователь с административными привилегиями. К компьютеру должны быть подсоединены как минимум один жесткий диск (HDD) и один твердотельный накопитель (SSD).
2. Установите следующие пакеты RPM: `vstorage-ctl`, `vstorage-libs-shared` и `vstorage-chunk-server`. Эти пакеты можно установить с помощью следующей команды:

   ```
   # yum install vstorage-chunk-server
   ```

3. Выполните аутентификацию сервера в кластере, если это еще не сделано:

   ```
   # vstorage -c stor1 auth-node
   ```

4. При необходимости подготовьте диск SSD:

   1. Запустите утилиту, указав `--ssd` и имя физического диска в качестве параметров. Например:

      ```
      # /usr/libexec/vstorage/prepare_vstorage_drive /dev/sdb --ssd
      ALL data on /dev/sdb will be completely destroyed. Are you sure to continue? [y]
      ```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
y
Zeroing out beginning and end of /dev/sdb...
Partitioning /dev/sdb...
Waiting for kernel...
Formatting /dev/sdb1 partition...
Done!
```

> **Примечание**
>
> Утилита не принимает имя раздела в качестве параметра.

2. Узнайте UUID раздела:

```
# ls -al /dev/disk/by-uuid/ | grep sdb1
lrwxrwxrwx 1 root root  10 Jun 19 02:41 f3fbcbb8-4224-4a6a-89ed-3c55bbc073e0 -> ../../
↪sdb1
```

3. Добавьте новый раздел в файл `/etc/fstab`, использовав его UUID, и укажите параметры монтирования `noatime`, `nofail` и `lazytime`. Например:

```
UUID=f3fbcbb8-4224-4a6a-89ed-3c55bbc073e0 /vstorage/stor1-ssd1   ext4 \
   defaults,nofail,lazytime,noatime 0 0
```

4. Смонтируйте раздел в каталог `/vstorage/<cluster>-ssd<N>`, где `<cluster>` — это имя кластера, а `<N>` — это первый свободный порядковый номер для SSD-диска.

> **Примечание**
>
> Если каталог `/vstorage/<cluster>-ssd<N>` не существует, создайте его.

5. Создайте конфигурацию сервера фрагментов данных, его репозиторий и журнал, например:

```
# vstorage -c stor1 make-cs -r /vstorage/stor1-cs -j /ssd/stor1/cs1 -s 30720 -T ssd
```

Эта команда:

1. Создает каталог `/vstorage/stor1-cs` на жестком диске вашего компьютера и настраивает его для хранения фрагментов данных.
2. Настраивает ваш компьютер в качестве сервера фрагментов данных и добавляет его в

кластер хранилища `stor1`.

3. Создает журнал в каталоге `/ssd/stor1/cs1` на диске SSD и выделяет ему дисковое пространство размером 30 ГБ.
4. Настраивает журнал для работы на диске SSD, чтобы получить повышенную производительности записи и чтения (наилучшие результаты заметны при случайной записи в кластерах, использующих только твердотельные накопители).

> **Примечание**
>
> При выборе каталога для журнала и определении его размера выделите необходимое пространство для журнала и убедитесь, что на каждый 1 ТБ пространства на жестком диске приходится 1 ГБ пространства на диске SSD для хранения контрольных сумм.

6. Запустите службу управления сервером фрагментов данных `vstorage-csd` и настройте ее автоматический запуск при загрузке сервера:

```
# systemctl start vstorage-csd.target
# systemctl enable vstorage-csd.target
```

Управление журналами служб фрагментов данных в работающих кластерах хранилища

> **Примечание**
>
> Чтобы просмотреть пути к репозиториям служб фрагментов данных, используйте команду `vstorage -c <cluster> list-services -C`.

Просмотр списка журналов служб фрагментов данных

Чтобы просмотреть список журналов, используйте следующий Bash-скрипт:

```
for i in $(find /vstorage/*/data -name journal); do
    echo ”$i $(du -sh $(readlink ${i}) | awk '{print $2,$1}')”
done
```

В выводе скрипта содержатся следующие сведения о каждом журнале:

- UUID дискового раздела службы фрагментов данных, к которой относится журнал;
- UUID дискового раздела, на котором размещен журнал;

- размер журнала.

Например:

```
<...>
/vstorage/d06a3c00-7235-4e49-8f3e-e09e2b489c62/data/control/journal /vstorage/w-cache/673c252c-
↪b9a5-4024-b19e-3508e3a1e77b/d06a3c00-7235-4e49-8f3e-e09e2b489c62 31G
<...>
```

Добавление журналов служб фрагментов данных

Чтобы добавить новый журнал к службе фрагментов данных, используйте команду `vstorage configure-cs -a -s`. Например, чтобы добавить журнал размером 2 048 МБ к службе фрагментов данных `CS#1`, разместите его в некотором каталоге на диске SSD, смонтированном в каталог `/ssd`, и настройте этот журнал для работы на дисках SSD:

```
# vstorage -c stor1 configure-cs -r <CS_repository_path> \
-a /ssd/stor1-cs1-journal -s 2048 -T ssd
```

Перезапустите службу фрагментов данных, чтобы применить изменения, вызванные параметром `-T`:

```
# systemctl status vstorage-csd* | grep -m1 ”stor1-cs1”
vstorage-csd.stor1.1027.service - vstorage-csd(/vstorage/stor1-cs1)
# systemctl restart vstorage-csd.stor1.1027.service
```

Удаление журналов служб фрагментов данных

Чтобы удалить журнал службы фрагментов данных, используйте команду `vstorage configure-cs -d`, например:

```
# vstorage -c stor1 configure-cs -r <CS_repository_path> -d
```

Перемещение журналов служб фрагментов данных

Чтобы изменить каталог журнала службы фрагментов данных, выполните следующее, использовав приведенные ранее команды:

1. Удалите существующий журнал.
2. Создайте новый журнал нужного размера в требуемом каталоге.

Изменение размера журналов служб фрагментов данных

Чтобы изменить размер журнала службы фрагментов данных, используйте команду `vstorage configure-cs -s`. Например, чтобы сделать размер журнала равным 4 096 МБ, выполните:

```
# vstorage -c stor1 configure-cs -r <CS_repository_path> -s 4096
```

Выключение расчета контрольных сумм

Используя расчет контрольных сумм, вы можете обеспечить лучшую надежность и целостность всех данных в вашем кластере хранилища. Когда расчет контрольных сумм включен, продукт Кибер Хранилище генерирует контрольные суммы каждый раз, когда изменяются данные в кластере. При последующем считывании этих данных контрольная сумма вычисляется еще раз и сравнивается с уже существующим значением.

Расчет контрольных сумм включается автоматически при создании служб фрагментов данных, использующих журналирование. Вы можете выключить эту функцию при необходимости, использовав параметр `-S` при установке службы фрагментов данных, например:

```
# vstorage -c stor1 make-cs -r /vstorage/stor1-cs -j /ssd/stor1/cs1 -s 30720 -S
```

Настройка проверки данных

Кластер хранилища автоматически проверяет фрагменты данных на долговечность и тестирует их содержимое на читабельность и корректность. По умолчанию кластер настроен на проверку двух фрагментов в минуту на каждом сервере фрагментов данных в кластере. При необходимости вы можете изменить это число с помощью утилиты **`vstorage`**, например:

```
# vstorage -c stor1 set-config mds.wd.verify_chunks=3
```

Эта команда делает число фрагментов для проверки на каждом сервере фрагментов данных в кластере `stor1` равным 3.

Улучшение производительности жестких дисков большой емкости

В отличие от старых жестких дисков с секторами размером 512 байт многие современные жесткие диски (емкостью 3 ТБ и более) используют физические секторы размером 4 КБ. В некоторых случаях это может значительно снизить производительность системы (в 3-4 раза) из-за дополнительных циклов ”чтение-модификация-запись” (RMW), необходимых для выравнивания исходного запроса на запись.

Почему это происходит? Когда операционная система выдает невыровненный запрос на запись, жесткий диск должен выровнять начало и конец запроса по границам 4 КБ. Для этого жесткий диск считывает головной и хвостовой диапазоны запроса, чтобы определить точное количество секторов для модификации. Например, при запросе на запись блока размером 4 КБ со смещением 2 КБ жесткий диск считывает диапазоны 0-2 КБ и 6-8 КБ, чтобы изменить весь диапазон данных 0-8 КБ.

Типичными причинами низкой производительности жестких дисков с секторами размером 4 КБ являются:

1. Файловая система ОС сервера не выровнена по границе 4 КБ. Команда `vstorage make-cs` пытается заранее обнаружить и сообщить администратору о подобных проблемах, но имейте в виду, что утилита **`fdisk`** не рекомендуется для разметки жестких дисков. Вместо нее следует использовать утилиту **`parted`**.
2. Невыровненные операции записи (например, 1 КБ), выполняемые гостевой ОС. Многие устаревшие операционные системы, такие как Microsoft Windows XP, Microsoft Windows Server 2003 или Red Hat Enterprise Linux 5.x, по умолчанию имеют невыровненные разделы и, как следствие, производят невыровненные операции записи, которые являются довольно медленными как в кластере хранилища, так и на реальных жестких дисках с секторами размером 4 КБ. Если вы планируете использовать такие устаревшие операционные системы, обратите внимание на следующее:
    - Использование жестких дисков меньшего размера с секторами размером 512 байт или использование SSD-дисков для хранения журналов служб CS в некоторой степени снижает остроту проблемы.
    - Необходимо правильное выравнивание разделов ОС.

Вы можете проверить наличие невыровненных операций записи в кластере следующим образом:

1. Выполните команду `vstorage top` или `vstorage stat`, например:

```
# vstorage -c stor1 top
```

2. Нажмите `i`, чтобы в выводе команды были отображены столбцы **RMW** и **JRMW** в разделе служб CS.
3. Проверьте значение счетчиков **RMW** или **JRMW**, которые объяснены ниже.
    - Когда журналы размещены на дисках SSD, счетчик **RMW** показывает количество запросов, которые приводят к циклам ”чтение-модификация-запись”, тогда как счетчик **JRMW** показывает количество циклов ”чтение-модификация-запись”, влияние которых удалось смягчить за счет использования журналов на дисках SSD.
    - Когда журналы не размещены на дисках SSD, счетчик **RMW** показывает количество

невыровненных запросов, которые потенциально могут привести к циклам ”чтение-модификация-запись” на рассматриваемом жестком диске.

Отключение автоматической миграции данных между уровнями

Если на каком-либо уровне хранения заканчивается свободное место, кластер хранилища попытается временно использовать пространство более низких уровней вплоть до низшего. Если заполнен и низший уровень, кластер хранилища попытается задействовать более высокий уровень. Если позже добавить пространства на исходный уровень, то данные, временно хранящиеся в другом месте, будут снова перемещены на него.

Смешивать рабочие нагрузки разных уровней не рекомендуется, так как это может снизить производительность кластера. Чтобы предотвратить это, можно отключить автоматическую миграцию данных между уровнями, предварительно убедившись, что на каждом уровне достаточно свободного места. Для отключения выполните следующую команду на любом сервере кластера:

```
# vstorage -c cluster1 set-config mds.alloc.strict_tier=1
```

Справочная информация

Устранение неполадок

В этом разделе описаны распространенные проблемы, с которыми вы можете столкнуться при работе с кластерами хранилища, и способы их устранения. Основным инструментом, используемым для устранения неполадок кластера хранилища и обнаружения аппаратных сбоев, является утилита **`vstorage top`**.

Отправка отчетов о неполадках в службу технической поддержки

Если в кластере возникла проблема, которую вы не можете решить самостоятельно, можно воспользоваться командой `vstorage make-report` для составления подробного отчета о кластере. Затем вы можете отправить отчет в службу технической поддержки, которая внимательно изучит проблему и сделает все возможное, чтобы решить ее как можно быстрее.

Чтобы составить отчет, выполните следующие действия:

1. Настройте для пользователя `root` беспарольный SSH-доступ с сервера, на котором вы планируется выполнить команду `vstorage make-report`, на все серверы, участвующие в кластере.

   Самый простой способ сделать это — создать SSH-ключ с помощью утилиты **`ssh-keygen`** и использовать утилиту **`ssh-copy-id`**, чтобы настроить все серверы на доверие к этому ключу. Подробности см. в документации утилит **`ssh-keygen`** и `ssh-copy-id`.

2. Выполните команду `vstorage make-report`, чтобы составить отчет:

   ```
   # vstorage -c stor1 make-report
   The report is created and saved to vstorage-report-20121023-90630.tgz
   ```

   Команда собирает связанную с кластером информацию на всех серверах, участвующих в кластере, и сохраняет ее в файл в вашем текущем рабочем каталоге. Точное имя файла можно узнать, проверив вывод команды `vstorage make-report` (`vstorage-report-20121023-90630.tgz` в примере выше).

   При необходимости вы можете сохранить отчет в файл с заданным именем и поместить его в нужное место. Для этого передайте команде параметр `-f` и укажите желаемое имя файла (с расширением `.tgz`) и местоположение, например:

```
# vstorage -c stor1 make-report -f /home/reportSTOR1.tgz
```

Как только отчет будет подготовлен, отправьте его в службу технической поддержки.

> **Примечание**
>
> Отчет содержит информацию, связанную только с настройками и конфигурацией вашего кластера. Он не содержит никакой частной информации.

Нехватка дискового пространства

Когда в кластере хранилища остается очень мало свободного дискового пространства, критически важно как можно скорее увеличить его за счет добавления новых серверов фрагментов данных или удаления некоторых данных. Как только 95 % дискового пространства кластера оказывается занято, выделение новых фрагментов данных становится невозможным, а такого рода запросы блокируются до тех пор, пока кластер не сможет удовлетворить спрос. В результате пользовательский ввод-вывод также блокируется.

> **Примечание**
>
> Настоятельно рекомендуется держать свободным не менее 10 % дискового пространства для восстановления в случае сбоев серверов. Также следует контролировать использование дискового пространства, например, с помощью команды `vstorage top` или `vstorage get-event` (подробные сведения см. в разделе *Мониторинг кластеров хранилища*).

Симптомы

1. Заблокированный ввод-вывод или неотвечающая точка монтирования, сообщения в выводе команды `dmesg` о заблокированном вводе-выводе, ”замороженные” виртуальные машины.
2. Команды `vstorage top` и `vstorage get-event` выводят сообщения об ошибке наподобие ”Failed to allocate X replicas at tier Y since only Z chunk servers are available for allocation”.

Решения

1. Удалите все ненужные данные, чтобы освободить дисковое пространство.

   > **Примечание**

> Дополнительный эффект, который поначалу может удивить, заключается в том, что, как только очереди ввода-вывода в ядре переполнятся заблокированными данными, точка монтирования на клиентской машине может вообще перестать отвечать и не сможет обслуживать запросы, даже такие, как получение списка файлов. В этом случае можно создать дополнительную точку монтирования для получения списка, чтения и удаления ненужных данных.

2. Добавьте дополнительные службы фрагментов данных для неиспользуемых дисков.

Если вышеперечисленные решения невозможны, можно воспользоваться одним из следующих *временных* обходных путей:

1. Понизить коэффициент репликации для некоторых наименее важных пользовательских данных (см. раздел *Настройка параметров репликации*). Не забудьте потом вернуть изначальное значение.

2. Уменьшить резерв выделения. Например, для кластера `stor1`:

   ```
   # vstorage -c stor1 set-config mds.alloc.fill_margin=2
   ```

   В этой команде параметр `mds.alloc.fill_margin` — процент зарезервированного для рабочих нужд служб CS дискового пространства (значение по умолчанию — 5 %). Не забудьте потом вернуть изначальное значение.

Низкая производительность записи

Известно, что некоторые сетевые адаптеры, такие как, например, RTL8111/8168B, не могут обеспечить полную пропускную способность, полнодуплексный сетевой трафик. Это может привести к низкой производительности записи.

Поэтому перед развертыванием кластера хранилища настоятельно рекомендуется проверить сети на поддержку полного дуплекса. Вы можете использовать утилиту **`netperf`** для одновременной генерации входящего и исходящего трафика. Например, в сетях 1 GbE должно постоянно генерироваться около 2 Гбит/с общего трафика (1 Гбит/с для входящего и 1 Гбит/с для исходящего трафика).

Низкая производительность дискового ввода-вывода

В большинстве версий BIOS устройства SATA по умолчанию настроены для работы в режиме IDE. В этом режиме ваши серверы работают с устройствами SATA медленнее, чем в режиме AHCI, что может повлиять на производительность кластера, сделав ее до двух раз меньше, чем ожидалось. Проверить, какой режим используется для жесткого диска, можно с помощью команды `hdparm`, например:

```
# hdparm -i /dev/sda
...
 PIO modes:  pio0 pio1 pio2 pio3 *pio4
 DMA modes:  mdma0 mdma1 mdma2
 UDMA modes: udma0 udma1 udma2 udma3 udma4 udma5 udma6
```

Звездочка перед `pio4` в поле **PIO modes** указывает на то, что жесткий диск `/dev/sda` в настоящее время работает в устаревшем режиме IDE.

Чтобы решить эту проблему и увеличить производительность кластера, включите для жесткого диска режим AHCI в настройках BIOS.

Аппаратный RAID-контроллер и дисковые кэши записи

Важно, чтобы все жесткие диски подчинялись команде FLUSH и записывали свои кэши до завершения команды. К сожалению, не все аппаратные RAID-контроллеры и диски выполняют эту команду, что может привести к несогласованности данных и повреждению файловой системы при незапланированном отключении питания.

Некоторые RAID-контроллеры 3ware не отключают кэши записи на диски и не отправляют дискам команды ”flush”. В результате файловая система иногда может быть повреждена, даже если в самом RAID-контроллере есть батарея. Чтобы решить эту проблему, отключите кэш записи на всех дисках в RAID-массиве.

Кроме того, перед развертыванием кластера хранилища убедитесь, что ваша конфигурация тщательно протестирована на согласованность. Сведения о том, как это сделать, см. в разделе *Диски SSD не сбрасывают данные из кэша*.

Диски SSD не сбрасывают данные из кэша

Многие SSD-накопители настольного класса могут игнорировать команды сброса данных из кэша и вводить в заблуждение операционные системы, сообщая им, что данные были записаны, в то время как на самом деле этого не было. Примерами таких накопителей являются OCZ Vertex 3, Intel X25-E и Intel X-25-M G2, которые, как известно, небезопасны при фиксации данных. Такие диски не следует использовать для баз данных, и они могут легко повредить файловую систему при незапланированном отключении питания.

SSD-накопители Intel третьего поколения (серий S3700 и 710) не имеют таких проблем, поскольку оснащены конденсатором, обеспечивающим резервное питание для очистки кэша диска при отключении питания.

Используйте SSD-накопители с осторожностью и только в том случае, если вы уверены, что они

относятся к серверному классу и подчиняются правилам сброса данных кэша на диск. Для получения дополнительных сведений по этому вопросу см. статью о PostgreSQL.

Кластер хранилища не может создать достаточное количество реплик

Иногда кластер хранилища может не создавать фрагменты данных в необходимом количестве, даже если в нем присутствует достаточное количество серверов фрагментов данных.

Это может произойти, если вы создаете новые серверы фрагментов данных путем создания копий существующего сервера фрагментов данных (например, вы устанавливаете сервер фрагментов данных на виртуальной машине, а затем создаете копии этой машины). В этом случае все скопированные серверы фрагментов данных будут иметь одинаковый UUID, то есть UUID исходного сервера. Кластер будет считать, что все службы фрагментов данных находятся на исходном сервере, и не сможет выделять новые фрагменты данных.

Чтобы решить эту проблему, сгенерируйте новый UUID для клонированного сервера фрагментов данных, выполнив на нем следующую команду:

```
# /usr/bin/uuidgen -r | tr '-' ' ' | awk '{print $1$2$3}' > /etc/vstorage/host_id
```

Для получения дополнительной информации об утилите **uuidgen** см. ее документацию.

Неисправные серверы фрагментов данных

В случае неисправности сервера фрагментов данных (CS) необходимо определить причину ее возникновения и выбрать правильный способ решения проблемы.

Выполните следующие действия:

1. Запустите команду `vstorage top`, например:

   ```
   # vstorage -c stor1 top
   ```

2. Нажмите `i`, чтобы перейти к столбцу `FLAGS` в разделе служб фрагментов данных, и найдите флаги неисправной службы фрагментов данных.

3. Найдите отображаемые флаги в таблице ниже, чтобы идентифицировать причину сбоя и найти способ устранить проблему.

Флаг	Проблема	Что делать
H	Ошибка ввода-вывода. Неисправен диск службы фрагментов данных.	Проверьте диск на наличие ошибок. Если диск неисправен, замените его и создайте службу фрагментов данных повторно, как описано в разделе *Замена диска службы фрагментов данных*. В противном случае обратитесь в службу технической поддержки.
h	Несоответствие контрольной суммы фрагмента данных. Либо фрагмент данных поврежден, либо неисправен диск, на котором хранится этот фрагмент данных.	Проверьте диск на наличие ошибок. Если диск неисправен, замените его и создайте службу фрагментов данных повторно, как описано в разделе *Замена диска службы фрагментов данных*. В противном случае обратитесь в службу технической поддержки.
S	Недоступен журнал службы фрагментов данных, расположенный на диске SSD. Либо поврежден журнал, либо неисправен диск SSD.	Проверьте диск SSD, на котором расположен журнал, на наличие ошибок. Если диск неисправен, замените его, как описано в разделе *Выход из строя SSD-диска с журналами записи*.
R	При запуске службы фрагментов данных обнаружено, что путь к репозиторию фрагментов данных является недействительным. Диск службы фрагментов данных не подключен или не смонтирован.	Убедитесь, что диск подключен и смонтирован. Удостоверьтесь, что у диска есть правильная запись в файле `/etc/fstab`.
T	Истекло время ожидания запроса ввода-вывода. Диск недоступен по некоторой причине, но не обязательно сломан.	Убедитесь, что диск подключен. Проверьте в выводе команды `dmesg` сообщения об истечении времени ожидания запросов ввода-вывода, чтобы узнать, почему диск был недоступен.

Замена диска службы фрагментов данных

Чтобы заменить вышедший из строя жесткий диск или твердотельный накопитель, используемый службой фрагментов данных, сделайте следующее:

1. Удалите неработоспособную службу фрагментов данных из кластера, как описано в разделе *Удаление серверов фрагментов данных*.

> **Примечание**
>
> Не отсоединяйте вышедший из строя диск до тех пор, пока вы не удалите соответствующую

службу фрагментов данных.

2. Замените неисправный диск на новый.
3. Подготовьте диск, как описано в разделе *Подготовка диска*.
4. Создайте новую службу фрагментов данных, как описано в разделе *Последующие серверы*.

Выход из строя SSD-диска с журналами записи

Если вышел из строя диск SSD, используемый для хранения журналов записи, откажут все службы фрагментов данных, журналы которых размещены на этом диске. Кластер хранилища продолжит работу и создаст новые реплики, чтобы компенсировать утраченные. Если вам нужно разместить журналы записи на новом диске SSD, сделайте следующее:

1. Удалите неработоспособные службы фрагментов данных, как описано в разделе *Удаление серверов фрагментов данных*.
2. Подготовьте диск SSD, как описано в разделе *Подготовка диска*.
3. Создайте новые службы фрагментов данных, которые будут хранить журналы записи на новом диске SSD, как описано в разделе *Использование дисков SSD*.

Часто задаваемые вопросы

В этом разделе перечислены наиболее часто задаваемые вопросы о кластерах хранилища.

Общие

- **Нужно ли приобретать дополнительное оборудование для хранения данных для данного продукта?**

  Нет. Продукт Кибер Хранилище избавляет от необходимости иметь внешние устройства хранения данных, обычно используемые в SAN, преобразуя локально подключенные хранилища нескольких серверов в общее хранилище.

- **Каковы требования данного продукта к оборудованию?**

  Продукт Кибер Хранилище не требует специального оборудования и может работать на обычных компьютерах с традиционными дисками SATA и сетями 1 GbE. Однако некоторые жесткие диски и RAID-контроллеры игнорируют команду FLUSH для имитации лучшей производительности и не должны использоваться в кластерах, так как это может привести к повреждению файловой системы или журнала. Это особенно актуально для RAID-контроллеров и SSD-дисков. Чтобы

убедиться, что вы используете надежное оборудование, обратитесь к руководству по эксплуатации жесткого диска.

Дополнительные сведения см. в разделе *Требования к оборудованию*.

- **Сколько нужно серверов для запуска кластера хранилища?**

  Для начала использования продукта Кибер Хранилище вам потребуется только один физический сервер. Однако для обеспечения высокой доступности данных рекомендуется иметь не менее трех реплик на каждый фрагмент данных. Для этого вам потребуется не менее трех серверов (и не менее пяти в итоге) в кластере. Подробные сведения см. в разделах *Конфигурации кластера хранилища* и *Настройка параметров репликации*.

Масштабируемость и производительность

- **Сколько серверов можно добавить в кластер хранилища?**

  Строгих ограничений на количество серверов, которые можно включить в кластер хранилища, не существует. Однако рекомендуется ограничить количество серверов в кластере одной серверной стойкой, чтобы избежать возможного снижения производительности из-за обмена данными между серверами, расположенными в разных серверных стойках.

- **Как много дискового пространства может быть у кластера хранилища?**

  Кластер хранилища может поддерживать эффективное доступное дисковое пространство объемом до 8 ПБ, что в сумме составляет физическое дисковое пространство объемом 24 ПБ при сохранении 3 реплик для каждого фрагмента данных.

- **Можно ли добавлять серверы в существующий кластер хранилища?**

  Да, вы можете динамически добавлять серверы в кластер хранилища и удалять их из него, чтобы увеличить его емкость или отключить серверы для обслуживания.

- **Какова ожидаемая производительность кластера хранилища?**

  Производительность зависит от скорости сети и жестких дисков, используемых в кластере. В целом производительность должна быть такой же, как у локально подключенного хранилища, или даже выше. Вы также можете использовать SSD-кэширование для повышения производительности оборудования, добавив SSD-диски в кластер для кэширования. Дополнительные сведения см. в разделе *Использование дисков SSD*.

- **Какой производительности следует ожидать при использовании 1-гигабитного Ethernet?**

  Максимальная скорость сети 1 GbE близка к скорости одного жесткого диска. Однако в большинстве рабочих ситуаций преобладает случайный ввод-вывод, и сеть обычно не является узким местом. Исследования крупных поставщиков услуг доказали, что средняя

производительность ввода-вывода редко превышает 20 МБ/с из-за бессистемности. Виртуализация сама по себе вносит дополнительную бессистемность, поскольку несколько независимых сред осуществляют ввод-вывод одновременно. Тем не менее, 10-гигабитный Ethernet часто обеспечивает более высокую производительность и рекомендуется для использования в производстве.

- **Повысится ли общая производительность кластера, если в него добавить новые серверы фрагментов данных?**

  Да. Поскольку данные распределяются между всеми жесткими дисками в кластере, приложения, выполняющие произвольный ввод-вывод, испытывают увеличение количества операций ввода-вывода в секунду, когда в кластер добавляется больше дисков. Даже одна клиентская машина может получить заметные преимущества за счет увеличения количества серверов фрагментов данных и достичь производительности, значительно превышающей производительность традиционного локально подключенного хранилища.

- **Зависит ли производительность от количества реплик фрагментов данных?**

  Каждая дополнительная реплика снижает производительность записи примерно на 10 %, но в то же время она может повысить производительность чтения, поскольку кластер хранилища имеет больше возможностей для выбора более быстрого сервера.

Доступность

- **Как кластер хранилища защищает мои данные?**

  Кластер хранилища защищает данные от потери и временной недоступности путем создания копий данных (реплик) и хранения их на разных серверах. Чтобы обеспечить дополнительную надежность, вы можете настроить в продукте поддержку контрольных сумм пользовательских данных и их проверку при необходимости.

- **Что произойдет, если откажет диск или сервер станет недоступен?**

  Кластер хранилища автоматически восстанавливается из деградированного состояния до указанного уровня избыточности путем репликации данных на действующие серверы. Во время процесса восстановления пользователи могут осуществлять доступ к своим данным.

- **Как быстро кластер хранилища восстанавливается из деградированного состояния?**

  Поскольку кластер хранилища восстанавливается из деградированного состояния, используя все доступные жесткие диски, процесс восстановления происходит гораздо быстрее, чем в традиционных локально подключенных RAID-массивах. Это значительно повышает надежность системы хранения, так как вероятность потери единственной оставшейся копии данных в период восстановления очень мала.

- **Могу ли я изменить настройки избыточности ”на лету”?**

  Да, в любой момент вы можете изменить количество копий данных, а кластер хранилища применит новые настройки, создав новые копии или удалив ненужные. Более подробные сведения о настройке параметров репликации см. в разделе *Настройка параметров репликации*.

- **Нужно ли мне по-прежнему использовать локальные RAID-массивы?**

  Нет, кластер хранилища обеспечивает такую же встроенную избыточность данных, как и массив RAID 1 с несколькими копиями. Однако для повышения производительности последовательного ввода-вывода можно использовать локальный массив RAID 0, экспортированный в ваш кластер. Дополнительные сведения об использовании RAID-массивов см. в разделе *Возможные конфигурации дисковых накопителей*.

- **Имеет ли кластер хранилища уровни избыточности, аналогичные RAID 5?**

  Нет. Чтобы создать надежную программную систему RAID 5, необходимо использовать специальные аппаратные возможности, такие как, например, батареи резервного питания. В будущем продукт Кибер Хранилище может быть усовершенствован для обеспечения избыточности уровня RAID 5 для доступных только для чтения данных, таких как резервные копии.

- **Каково рекомендуемое количество копий данных?**

  Рекомендуется настроить кластер хранилища для поддержки 2 или 3 копий, что позволит вашему кластеру пережить одновременную потерю 1 или 2 жестких дисков.

Эксплуатация кластера хранилища

- **Как узнать, что новые параметры репликации были успешно применены к кластеру?**

  Чтобы проверить, завершен ли процесс репликации, запустите команду `vstorage top`, нажмите клавишу `v` на клавиатуре и проверьте сведения в поле **Chunks**:

  - При уменьшении параметров репликации в выводе не должно быть фрагментов данных в состоянии `overcommitted` или `deleting`.
  - При увеличении параметров репликации в выводе не должно быть фрагментов данных в состоянии `blocked` или `urgent`. Кроме того, общее состояние работоспособности кластера должно составлять 100 %.

  Подробные сведения см. в разделе *Мониторинг параметров репликации*.

- **Как выключить кластер?**

  Чтобы выключить кластер хранилища, сделайте следующее:

  1. Остановите и отключите все необходимые службы кластера.

2. Остановите все запущенные виртуальные машины.
3. Остановите всех клиентов.
4. Остановите все серверы фрагментов данных.
5. Остановите все серверы метаданных.

Подробные сведения см. в разделе *Включение и выключение серверов кластера хранилища*.

- **С помощью какого инструмента можно отслеживать состояние и работоспособность кластера?**

  Состояние и работоспособность кластера можно отслеживать с помощью команды `vstorage top`. Подробнее см. в разделе *Мониторинг кластеров хранилища*.

  Чтобы посмотреть общий объем дискового пространства, занимаемого всеми пользовательскими данными в кластере, запустите команду `vstorage top`, нажмите клавишу `V` на клавиатуре и найдите в выводе поле **FS**. Поле **FS** показывает, сколько дискового пространства используется всеми пользовательскими данными в кластере и в скольких файлах эти данные хранятся. Подробные сведения см. в разделе *Общие сведения об использовании дискового пространства*.

- **Почему “vmstat“/“top“ и “vstorage stat“ показывают разное время ожидания ввода-вывода?**

  Утилиты `vstorage stat` и `vmstat`/`top` используют разные методы вычисления процента времени ЦП, потраченного на ожидание дискового ввода-вывода (**wa%** в `top`, **wa** в `vmstat` и **IOWAIT** в `vstorage stat`). Утилиты `vmstat` и `top` помечают простаивающий процессор как ожидающий только в том случае, если на нем обрабатывается запрос ввода-вывода, в то время как утилита **`vstorage stat`** помечает все простаивающие процессоры как ожидающие, независимо от количества запросов ввода-вывода, ожидающих ввода-вывода. В результате утилита **`vstorage stat`** может сообщать гораздо более высокие значения. Например, в системе с 4 ЦП и одним потоком, выполняющим ввод-вывод, `vstorage stat` покажет более 90 % времени IOWAIT, в то время как `vmstat` и `top` покажут не более 25 % времени IO.

- **Как нумерация уровней хранения данных влияет на работу кластера хранилища?**

  При назначении дисковому пространству уровней хранения принимайте во внимание, что более быстрым дискам следует назначать более высокие уровни хранения. Например, вы можете использовать уровень хранения 0 для резервных копий и других холодных данных (CS без журналов на дисках SSD), уровень хранения 1 для виртуальных машин — большой объем холодных данных с частыми операциями случайной записи (CS с журналами на дисках SSD), уровень хранения 2 для горячих данных (CS с журналами на дисках SSD), журналов, кэшей, отдельных дисков виртуальных машин и т. п.

  Эта рекомендация относится к тому, как кластер хранилища работает с дисковым пространством.

Если на некотором уровне хранения заканчивается свободное место, кластер хранилища попытается временно использовать дисковое пространство более низкого уровня хранения. Если вы добавите еще дискового пространства для исходного уровня хранения, данные, временно хранящиеся на других уровнях хранения, будут перемещены на тот уровень хранения, где они должны были быть изначально.

Например, если вы пытаетесь записать данные на уровень хранения 2 и он заполнен, кластер хранилища попытается записать эти данные на уровень хранения 1, а затем на уровень хранения 0. Если вы потом добавите еще дискового пространства на уровень хранения 2, упомянутые выше данные, хранящиеся в данный момент на уровнях хранения 1 или 0, будут перемещены обратно на уровень хранения 2, где они должны были храниться изначально.

Порты, используемые кластером хранилища

В таблице приведены номера сетевых портов, используемых продуктом Кибер Хранилище.

Порт	Описание
Службы метаданных (MDS)	
2510	(IO) Используется для связи между службами метаданных.
2511	(IO) Используется для связи со службами фрагментов данных и клиентами.
Службы фрагментов данных (CS)	
2511	(O) Используется для связи со службами метаданных.
`<random_port>`	(I) Используется для связи с клиентами. Служба управления фрагментами данных автоматически привязывается к любому незанятому порту. При необходимости вы можете вручную назначить этой службе отдельный порт.
Клиенты кластера	
2511	(O) Используется для связи со службами метаданных.
`<random_port>`	(O) Используется для связи со службами фрагментов данных. Служба управления клиентскими компонентами автоматически привязывается к любому незанятому порту. При необходимости вы можете вручную назначить этой службе отдельный порт.