

КИБЕРПРОТЕКТ



КИБЕР Инфраструктура

Версия 6.0

Руководство по интеграции сервиса «хранилище как услуга»

Редакция: 05.07.2024

Заявление об авторских правах

Все права защищены.

Все остальные упоминаемые товарные знаки могут быть зарегистрированными товарными знаками соответствующих владельцев.

Распространение существенно измененных версий данного руководства запрещено без явного разрешения владельца авторских прав.

Распространение настоящих или переработанных материалов, входящих в данное руководство, в виде печатного издания (книги) запрещено без письменного разрешения их владельца.

ДОКУМЕНТАЦИЯ ПОСТАВЛЯЕТСЯ «КАК ЕСТЬ». НЕ СУЩЕСТВУЕТ НИКАКИХ ЯВНО ВЫРАЖЕННЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ОБЯЗАТЕЛЬСТВ, ПОДТВЕРЖДЕНИЙ ИЛИ ГАРАНТИЙ, В ТОМ ЧИСЛЕ И СВЯЗАННЫХ С ТОВАРНОСТЬЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЛИ ПРИГОДНОСТЬЮ ЕГО ДЛЯ ИСПОЛЬЗОВАНИЯ В ОПРЕДЕЛЕННЫХ ЦЕЛЯХ, НАСКОЛЬКО ТАКАЯ ОГРАНИЧЕННОСТЬ ОТВЕТСТВЕННОСТИ ДОПУСКАЕТСЯ ЗАКОНОМ.

С ПО или Услугой может быть предоставлен исходный код сторонних производителей. Лицензии этих сторонних производителей подробно описаны в файле `license.txt`, находящемся в корневом каталоге установки.

Содержание

1 Введение	6
1.1 Об этом руководстве	6
1.2 О решении WHMCS	6
1.3 О кластерах S3	6
1.4 Методы интеграции	6
2 Интеграция через интерфейс командной строки	8
2.1 Управление пользователями S3 через интерфейс командной строки	8
2.1.1 Добавление пользователей S3 через интерфейс командной строки	9
2.1.2 Вывод списка пользователей S3 через интерфейс командной строки	9
2.1.3 Запрос информации о пользователе S3 через интерфейс командной строки	10
2.1.4 Отключение и включение пользователей S3 через интерфейс командной строки	10
2.1.5 Удаление пользователей S3 через интерфейс командной строки	10
2.1.6 Создание пар ключей доступа для пользователя S3 через интерфейс командной строки	11
2.1.7 Отзыв пар ключей доступа для пользователя S3 через интерфейс командной строки	11
2.2 Управление учетными записями S3 через интерфейс командной строки	11
2.2.1 Создание учетных записей S3 через интерфейс командной строки	11
2.2.2 Вывод списка учетных записей S3 через интерфейс командной строки	12
2.2.3 Запрос информации об учетной записи S3 через интерфейс командной строки	12
2.2.4 Удаление учетных записей S3 через интерфейс командной строки	13
2.2.5 Создание пар ключей доступа для учетной записи S3 через интерфейс командной строки	13
2.2.6 Отзыв пар ключей доступа для учетной записи S3 через интерфейс командной строки	13
2.3 Управление лимитами для пользователей и корзин S3 через интерфейс командной строки	13
2.3.1 Установка числа операций в секунду для пользователей через интерфейс командной строки	14
2.3.2 Установка пропускной способности в секунду для пользователей через интерфейс командной строки	14
2.3.3 Запрос лимитов пользователя через интерфейс командной строки	14
2.3.4 Удаление лимитов пользователя через интерфейс командной строки	15
2.3.5 Установка числа операций в секунду для корзин через интерфейс командной строки	15
2.3.6 Установка пропускной способности в секунду для корзин через интерфейс командной строки	15
2.3.7 Запрос лимитов корзины через интерфейс командной строки	16
2.3.8 Удаление лимитов корзины через интерфейс командной строки	16
3 Интеграция через REST API	17

3.1 Требования для интеграции через REST API	17
3.1.1 Подготовка среды	17
3.1.2 Включение статистики	18
3.2 Управление пользователями S3 и вывод списка корзин через REST API	19
3.2.1 Создание пользователей S3 через REST API	19
3.2.2 Вывод списка пользователей S3 через REST API	19
3.2.3 Запрос пользователей S3 через REST API	20
3.2.4 Отключение и включение пользователей S3 через REST API	21
3.2.5 Удаление пользователей и корзин S3 через REST API	21
3.2.6 Создание ключей доступа для пользователя S3 через REST API	22
3.2.7 Отзыв ключей доступа для пользователя S3 через REST API	22
3.2.8 Вывод списка корзин пользователей через REST API	22
3.3 Управление учетными записями S3 через REST API	23
3.3.1 Создание учетных записей S3 через REST API	23
3.3.2 Удаление учетных записей S3 через REST API	24
3.3.3 Создание ключей доступа для учетной записи S3 через REST API	24
3.3.4 Отзыв ключей доступа для учетной записи S3 через REST API	24
3.4 Управление лимитами для пользователей и корзин S3 через REST API	24
3.4.1 Установка числа операций в секунду для пользователей через REST API	25
3.4.2 Установка пропускной способности в секунду для пользователей через REST API	25
3.4.3 Запрос лимитов пользователя через REST API	25
3.4.4 Удаление лимитов пользователя через REST API	25
3.4.5 Установка числа операций в секунду для корзин через REST API	26
3.4.6 Установка пропускной способности в секунду для корзин через REST API	26
3.4.7 Запрос лимитов корзины через REST API	26
3.4.8 Удаление лимитов корзины через REST API	26
3.5 Управление квотами для пользователей и корзин S3 через REST API	27
3.5.1 Установка квот для пользователей через REST API	27
3.5.2 Запрос квот для пользователя через REST API	27
3.5.3 Удаление квот для пользователей через REST API	27
3.5.4 Установка квот для корзин через REST API	28
3.5.5 Запрос квот для корзины через REST API	28
3.5.6 Удаление квот для корзины через REST API	28
3.5.7 Установка квот по умолчанию через REST API	28
3.5.8 Запрос квот по умолчанию через REST API	28
3.5.9 Удаление квот по умолчанию через REST API	29
3.6 Получение статистики использования через REST API	29

3.6.1 Вывод списка объектов статистики через REST API	29
3.6.2 Запрос объектов статистики через REST API	30
3.6.3 Удаление объектов статистики через REST API	31
4 Интеграция с WHMCS	32
4.1 Требования	32
4.1.1 Настройка	32
4.1.2 Файлы include	33
4.1.3 Статистика	36
4.2 Управление пользователями S3 в WHMCS	37
4.2.1 Создание пользователей S3 в WHMCS	37
4.2.2 Вывод списка пользователей S3 в WHMCS	38
4.2.3 Запрос пользователей S3 в WHMCS	39
4.2.4 Отключение пользователей S3 в WHMCS	40
4.2.5 Включение пользователей S3 в WHMCS	41
4.2.6 Удаление пользователей S3 в WHMCS	42
4.2.7 Создание ключей доступа S3 в WHMCS	43
4.2.8 Отзыв ключей доступа S3 в WHMCS	44
4.3 Управление лимитами для пользователей и корзин S3 в WHMCS	45
4.3.1 Установка лимитов пользователя в WHMCS	46
4.3.2 Запрос лимитов пользователя в WHMCS	47
4.3.3 Удаление лимитов пользователя в WHMCS	48
4.3.4 Установка лимитов корзины в WHMCS	49
4.3.5 Запрос лимитов корзины в WHMCS	50
4.3.6 Удаление лимитов корзины в WHMCS	51
4.4 Получение статистики использования в WHMCS	52
4.4.1 Вывод списка объектов статистики в WHMCS	52
4.4.2 Запрос объектов статистики в WHMCS	53
4.4.3 Удаление объектов статистики в WHMCS	54

1 Введение

1.1 Об этом руководстве

Этот документ поможет интегрировать сервисы, совместимые с Amazon S3, в вашу систему биллинга и распределения ресурсов WHMCS. Руководство предназначено главным образом для разработчиков, у которых уже есть рабочие кластеры хранилища с правильно настроенными шлюзами и ролями на базе Amazon S3.

В этом документе приведены примеры интеграции кластеров S3 продукта Кибер Инфраструктура через интерфейс командной строки или REST API, а также через WHMCS. Используя это руководство в качестве отправной точки, вы сможете создавать базовые решения «хранилище как услуга» на основе продукта Кибер Инфраструктура.

1.2 О решении WHMCS

WHMCS – это универсальная платформа для автоматизации хостинга, которая предоставляет возможности управления клиентами, распределения сервисов, биллинга и поддержки. Она обеспечивает все функции, от регистрации клиентов до их удаления. Функциональные возможности расширяются с помощью дополнительных модулей, расширений и крючков, которые выполняют сторонний код при определенных событиях. Дополнительную информацию о WHMCS можно найти на сайте <https://www.whmcs.com/>.

1.3 О кластерах S3

Кибер Инфраструктура позволяет экспортировать дисковое пространство кластера для клиентов в форме S3-подобного хранилища на основе объектов.

Кибер Инфраструктура реализует API-интерфейс, подобный интерфейсу Amazon S3, который является одним из самых распространенных API-интерфейсов объектного хранилища. Конечные пользователи могут работать с продуктом Кибер Инфраструктура так же, как они работают с Amazon S3. Можно использовать привычные приложения для S3 и продолжать работу с ними после миграции данных из Amazon S3 в решение Кибер Инфраструктура.

Дополнительные сведения о кластерах S3 можно найти в руководстве администратора.

1.4 Методы интеграции

Кибер Инфраструктура предоставляет REST API оркестрации, а также REST API, совместимый с Amazon S3.

С помощью API оркестрации можно управлять пользователями и корзинами, настраивать лимиты пользователей и корзин, а также собирать статистику использования. API оркестрации можно использовать посредством единого инструмента командной строки, поставляемого с продуктом Кибер Инфраструктура.

REST API, совместимый с Amazon S3, также позволяет управлять пользователями и корзинами, настраивать лимиты пользователей и корзин, а также собирать статистику использования. Модель пользователя и политики доступа соответствуют модели и политикам Amazon S3.

2 Интеграция через интерфейс командной строки

В этой главе объясняется, как использовать интерфейс командной строки для подготовки, включения, отключения и удаления пользователей и учетных записей S3, а также для установки лимитов пользователей и корзин в целях биллинга.

2.1 Управление пользователями S3 через интерфейс командной строки

Концепция пользователя S3 – одна из базовых идей хранилища объектных данных 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 Кибер Инфраструктура, соответствуют модели пользователей и политикам доступа Amazon S3.

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

Пользователями можно управлять с помощью инструмента `ostor-s3-admin`.

Чтобы сделать это через интерфейс командной строки, необходимо знать идентификатор тома, на котором они находятся. Его можно получить 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 подразумевается, что команды `ostor-s3-admin` выполняются администратором хранилища объектов, они не содержат проверок аутентификации или авторизации.

2.1.1 Добавление пользователей S3 через интерфейс командной строки

Можно создать случайный уникальный идентификатор пользователя S3 и пару ключей доступа (идентификатор ключа доступа S3, секретный ключ доступа S3) с помощью команды `ostor-s3-admin create-user`. Потребуется указать адрес электронной почты пользователя, например:

```
# ostor-s3-admin create-user -e user@email.com -V 01000000000000002
UserEmail:user@email.com
UserId:a49e12a226bd760f
KeyPair[0]:S3AccessKeyId:a49e12a226bd760fGHQ7
KeyPair[0]:S3SecretAccessKey:HSDu2DA00JNGjnRcAhLKfhrvlymzOVdLPsCK2dcq
Flags:none
```

Идентификатор пользователя S3 представляет собой строку из 16 символов в шестнадцатеричном формате. Созданная пара ключей доступа используется для подписи запросов к хранилищу объектов S3 в соответствии со схемой аутентификации Amazon S3 Signature Version 2.

2.1.2 Вывод списка пользователей S3 через интерфейс командной строки

Вывести список всех пользователей объектного хранилища S3 можно с помощью команды `ostor-s3-admin query-users`. Информация по каждому пользователю может занимать одну или несколько последовательных строк в таблице. Дополнительные строки используются для вывода пар ключей доступа S3, связанных с пользователем. Если у пользователя нет активных пар ключей, вместо них отображается знак минус, например:

```
# ostor-s3-admin query-users -V 01000000000000002
S3 USER ID   S3 ACCESS KEY ID   S3 SECRET ACCESS KEY   S3 USER EMAIL
bf0b3b15eb7c9019 bf0b3b15eb7c9019136Y   *** user2@abc.com
d866d9d114cc3d20 d866d9d114cc3d20G456   *** user1@abc.com
                d866d9d114cc3d20D8EW   ***
e86d1c19e616455   -                       - user3@abc.com
```

Чтобы вывести список в формате XML, используйте аргумент `-X`; для вывода секретных ключей используйте аргумент `-a`, например:

```
# ostor-s3-admin query-users -V 01000000000000002 -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>a49e12a226
bd760fGHQ7</S3AccessKeyId><S3SecretAccessKey>HSDu2DA00JNGjnRcAhLKfhrvlymzOVdLP
sCK2dcq</S3SecretAccessKey></KeyPair></Keys></User><User><Id>d7c53fc1f931661f</Id><E
mail>user@email.com</Email><Keys><OwnerId>0000000000000000</OwnerId><KeyPair><S3A
ccessKeyId>d7c53fc1f931661fZLIV</S3AccessKeyId><S3SecretAccessKey>JL7gt1OH873zR0Fzv
```

```
8Oh9ZuA6JtCVnkgV7IET6ET</S3SecretAccessKey></KeyPair></Keys></User></Users></QueryUsersResult>
```

2.1.3 Запрос информации о пользователе S3 через интерфейс командной строки

Чтобы отобразить информацию об указанном пользователе, используйте команду `ostor-s3-admin query-user-info`. Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-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=83tTsNAuuRyoBBqhxFqHAC60dhKHtTCckQe54zu
```

2.1.4 Отключение и включение пользователей S3 через интерфейс командной строки

Отключить пользователя можно с помощью команды `ostor-s3-admin disable-user`. Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-i), например:

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

Чтобы включить отключенного пользователя, используйте команду `ostor-s3-admin enable-user`. Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-i), например:

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

2.1.5 Удаление пользователей S3 через интерфейс командной строки

Удалить существующих пользователей объектного хранилища можно с помощью команды `ostor-s3-admin delete-user`. Пользователей, являющихся владельцами корзин, нельзя удалить, поэтому сначала удалите корзины пользователя. Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-i), например:

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

2.1.6 Создание пар ключей доступа для пользователя S3 через интерфейс командной строки

Создать новую пару ключей доступа для указанного пользователя можно с помощью команды `ostor-s3-admin gen-access-key`. У каждого пользователя может быть максимум две активные пары ключей доступа (как и в Amazon Web Services). Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-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
```

Рекомендуется периодически отзывать старые пары ключей доступа и формировать новые.

2.1.7 Отзыв пар ключей доступа для пользователя S3 через интерфейс командной строки

Отозвать указанную пару ключей доступа для указанного пользователя можно с помощью команды `ostor-s3-admin revoke-access-key`. Необходимо указать ключ доступа в паре ключей, которую следует удалить, а также адрес электронной почты пользователя либо его идентификатор S3, 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
```

2.2 Управление учетными записями S3 через интерфейс командной строки

В этом разделе описывается, как управлять учетными записями S3 через интерфейс командной строки. Учетная запись S3 (проект) представляет собой изолированный контейнер корзины пользователя S3 с заданными лимитами использования. Учетная запись принадлежит одному пользователю, при этом у пользователя может быть несколько учетных записей.

2.2.1 Создание учетных записей S3 через интерфейс командной строки

Можно создать учетную запись пользователя S3 и пару ключей доступа (идентификатор ключа доступа S3, секретный ключ доступа S3) с помощью команды `ostor-s3-admin create-account`. Потребуется указать адрес электронной почты пользователя и имя учетной записи, например:

```
# ostor-s3-admin create-account -V 0100000000000002 -n account -e user@email.com
UserEmail:user@email.com
UserId:b09693b73b3c7686
AccountName:account
Flags:none
KeyPair[0]:S3AccessKeyId:b09693b73b3c7686ESY0
KeyPair[0]:S3SecretAccessKey:VxGZc12KKECUe1361IWrvXSVvZOdZAsfg4pL4M7T
```

Первые 16 символов созданного идентификатора ключа доступа соответствуют идентификатору пользователя.

2.2.2 Вывод списка учетных записей S3 через интерфейс командной строки

Вывести список всех учетных записей пользователя S3 можно с помощью команды `ostor-s3-admin list-user-accounts`. Информация по каждой учетной записи может занимать одну или несколько последовательных строк в таблице. Дополнительные строки используются для вывода пар ключей доступа S3, связанных с учетной записью. Если у учетной записи нет активных пар ключей, они не отображаются. Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-i), например:

```
# ostor-s3-admin list-user-accounts -V 0100000000000002 -e user@email.com
NAME  USER_ID  EMAIL  S3AccessKeyId  S3SecretAccessKey
account1 b09693b73b3c7686 user@email.com b09693b73b3c768613NV ***
account2 b09693b73b3c7686 user@email.com b09693b73b3c7686LCZ5 ***
          b09693b73b3c76866NI2 ***
account3 b09693b73b3c7686 user@email.com
```

2.2.3 Запрос информации об учетной записи S3 через интерфейс командной строки

Чтобы отобразить информацию об указанной учетной записи, используйте команду `ostor-s3-admin query-account-info`. Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-i) и имя учетной записи, например:

```
# ostor-s3-admin query-account-info -V 0100000000000002 -n account -e user@email.com
UserEmail:user@email.com
UserId:b09693b73b3c7686
AccountName:account
Flags:none
KeyPair[0]:S3AccessKeyId:b09693b73b3c7686ESY0
KeyPair[0]:S3SecretAccessKey:VxGZc12KKECUe1361IWrvXSVvZOdZAsfg4pL4M7T
```

2.2.4 Удаление учетных записей S3 через интерфейс командной строки

Удалить существующие учетные записи S3 можно с помощью команды `ostor-s3-admin delete-account`. Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-i) и имя учетной записи, например:

```
# ostor-s3-admin delete-account -V 0100000000000002 -n account -e user@email.com
```

2.2.5 Создание пар ключей доступа для учетной записи S3 через интерфейс командной строки

Создать новую пару ключей доступа для указанной учетной записи пользователя S3 можно с помощью команды `ostor-s3-admin gen-access-key`. У каждой учетной записи может быть максимум две активные пары ключей доступа. Необходимо указать адрес электронной почты пользователя (-e) либо его идентификатор S3 (-i) и имя учетной записи, например:

```
# ostor-s3-admin gen-access-key -V 0100000000000002 -n account -e user@email.com  
S3AccessKeyId:b09693b73b3c7686WCF3  
S3SecretAccessKey:GMzOPHJ66qbnxjrjFMHrN2fnxjkGf67u9OXec3PQ
```

Рекомендуется периодически отзывать старые пары ключей доступа и формировать новые.

2.2.6 Отзыв пар ключей доступа для учетной записи S3 через интерфейс командной строки

Отозвать указанную пару ключей доступа для указанной учетной записи пользователя S3 можно с помощью команды `ostor-s3-admin revoke-access-key`. Необходимо указать ключ доступа в паре ключей, которую следует удалить, а также адрес электронной почты пользователя (-e) либо его идентификатор S3 (-i), например:

```
# ostor-s3-admin revoke-access-key -e user@email.com -k de86d1c19e616455YIPU -V  
0100000000000002
```

2.3 Управление лимитами для пользователей и корзин S3 через интерфейс командной строки

В этом разделе описываются лимиты, которые можно задать для пользователей и корзин через интерфейс командной строки. Лимиты можно применять в соответствии с определенными параметрами, которые могут входить в ваш сервисный план.

2.3.1 Установка числа операций в секунду для пользователей через интерфейс командной строки

Скорость операций можно ограничить с помощью команды `set-limits` и следующих параметров: `-e` с указанием адреса электронной почты, `-t ops` с указанием типа лимита и `-L default=, get=, put=, list=` или `delete=` с указанием ключа лимита.

```
# ostor-s3-admin set-limits -e client@example.com -t ops -L get=3600
ops:default=0.00ops/s
ops:get=3600.00ops/s
ops:put=0.00ops/s
ops:list=0.00ops/s
ops:delete=0.00ops/s
bandwidth:out=0kbs/s
```

2.3.2 Установка пропускной способности в секунду для пользователей через интерфейс командной строки

Исходящую пропускную способность ответа можно ограничить с помощью команды `set-limits` и следующих параметров: `-e` с указанием адреса электронной почты, `-t bandwidth` с указанием типа лимита и `-L out=` с указанием ключа лимита.

```
# ostor-s3-admin set-limits -e client@example.com -t bandwidth -L out=100
ops:default=0.00ops/s
ops:get=3600.00ops/s
ops:put=0.00ops/s
ops:list=0.00ops/s
ops:delete=0.00ops/s
bandwidth:out=100kbs/s
```

2.3.3 Запрос лимитов пользователя через интерфейс командной строки

Текущие лимиты можно отобразить с помощью команды `query-limits` и параметра `-e` с указанием адреса электронной почты.

```
# ostor-s3-admin query-limits -e client@example.com
ops:default=0.00ops/s
ops:get=3600.00ops/s
ops:put=0.00ops/s
ops:list=0.00ops/s
ops:delete=0.00ops/s
bandwidth:out=100kbs/s
```

2.3.4 Удаление лимитов пользователя через интерфейс командной строки

Текущие лимиты можно удалить с помощью команды `rm-limits` и параметра `-e` с указанием адреса электронной почты:

```
# ostor-s3-admin rm-limits -e client@example.com
ops:default=0.00ops/s
ops:get=0.00ops/s
ops:put=0.00ops/s
ops:list=0.00ops/s
ops:delete=0.00ops/s
bandwidth:out=0kbs/s
```

2.3.5 Установка числа операций в секунду для корзин через интерфейс командной строки

Скорость операций можно ограничить с помощью команды `set-limits` и следующих параметров: `-b` с указанием имени корзины, `-t ops` с указанием типа лимита и `-L default=, get=, put=, list=` или `delete=` с указанием ключа лимита.

```
# ostor-s3-admin set-limits -b example -t ops -L get=3600
ops:default=0.00ops/s
ops:get=3600.00ops/s
ops:put=0.00ops/s
ops:list=0.00ops/s
ops:delete=0.00ops/s
bandwidth:out=0kbs/s
```

2.3.6 Установка пропускной способности в секунду для корзин через интерфейс командной строки

Исходящую пропускную способность ответа можно ограничить с помощью команды `set-limits` и следующих параметров: `-b` с указанием имени корзины, `-t bandwidth` с указанием типа лимита и `-L out=` с указанием ключа лимита.

```
# ostor-s3-admin set-limits -b example -t bandwidth -L out=100
ops:default=0.00ops/s
ops:get=3600.00ops/s
ops:put=0.00ops/s
ops:list=0.00ops/s
ops:delete=0.00ops/s
bandwidth:out=100kbs/s
```

2.3.7 Запрос лимитов корзины через интерфейс командной строки

Текущие лимиты можно отобразить с помощью команды `query-limits` и параметра `-b` с указанием имени корзины.

```
# ostor-s3-admin query-limits -b example
ops:default=0.00ops/s
ops:get=3600.00ops/s
ops:put=0.00ops/s
ops:list=0.00ops/s
ops:delete=0.00ops/s
bandwidth:out=100kbs/s
```

2.3.8 Удаление лимитов корзины через интерфейс командной строки

Текущие лимиты можно удалить с помощью команды `rm-limits` и параметра `-b` с указанием имени корзины.

```
# ostor-s3-admin rm-limits -b example
ops:default=0.00ops/s
ops:get=0.00ops/s
ops:put=0.00ops/s
ops:list=0.00ops/s
ops:delete=0.00ops/s
bandwidth:out=0kbs/s
```


3 Интеграция через REST API

В этой главе объясняется, как выполнять подготовку, включение, отключение и удаление пользователей и учетных записей S3, а также задавать лимиты пользователей и корзин в целях биллинга.

Приведенные примеры представляют собой команды Bash, с помощью которых можно отправлять запросы к REST API кластера S3 через cURL и OpenSSL. Ответы выдаются в формате JSON и могут подвергаться дальнейшей обработке с помощью таких инструментов, как `json_pp` или `json_reformat`.

Примечание

Замените `http://s3.example.com` в примерах фактическим URL-адресом вашего шлюза S3.

3.1 Требования для интеграции через REST API

Любой запрос на операцию или управление должен пройти проверку подлинности посредством подписанного запроса через Signature Version 2 или 4 протокола Amazon S3 от соответствующего системного пользователя S3. Для проверки подлинности API-запросов необходимо создать системного пользователя. Сначала получите идентификатор тома с помощью команды `ostor-ctl get-config`, например:

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

Затем создайте системного пользователя на любом сервере хранилища в кластере с помощью команды `ostor-s3-admin create-user -S -e <email>`, например:

```
# ostor-s3-admin create-user -S -e user@example.com -V 0100000000000002
UserEmail:user@example.com
UserId:a14040e0b2ef8b28
KeyPair[0]:S3AccessKeyId:a14040e0b2ef8b28FZZ8
KeyPair[0]:S3SecretAccessKey:dbwTnQTW602aAAAdq8DQVFzB6yrTCFTNiGB8C8RFA
Flags:system
```

С этим пользователем можно выполнять аутентификацию последующих API-запросов для управления кластером S3. Можно создать несколько системных учетных записей для разных типов операций управления.

3.1.1 Подготовка среды

В примерах этой главы cURL используется для проверки подлинности, а также выполнения запросов GET, PUT, POST и DELETE в Bash-скриптах. Чтобы упростить отправку запросов, можно

создать следующий скрипт `~/s3_environment`, заменив `s3_key` на `S3AccessKeyId` и `s3_secret` на `S3SecretAccessKey` системного пользователя.

```
# S3 login variables.
s3_key="a14040e0b2ef8b28FZZ8"
s3_secret="dbwTnQTW602aAAdq8DQVFzB6yrTCFTNiGB8C8RFA"

# Sign S3 requests and run curl.
function s3_curl() {

    # Parse command line.
    [-z "${2}" ] && {
        echo "Usage: ${FUNCNAME[0]} <request_type> <s3_url>"
        return 1
    }

    # Prepare a signature.
    s3_url="${2%/*}"
    s3_host="${s3_url##*/}"
    s3_query="${2##*/}"
    s3_date="$(date -R)"

    # Generate a signature.
    s3_signature="$(echo -en "${1}\n\n${s3_date}\n${s3_query}%&*" | \
        openssl sha1 -hmac ${s3_secret} -binary | base64)"

    # Make the request.
    curl -H "Host: ${s3_host}" \
        -H "Accept: */*" \
        -H "Date: ${s3_date}" \
        -H "Authorization: AWS ${s3_key}:${s3_signature}" \
        -X "${1}" \
        "${s3_url}/${s3_query}"
}
```

Загрузите скрипт в среду по умолчанию, чтобы сделать доступной функцию `s3_curl`.

```
# source ~/s3_environment
```

После загрузки скрипта можно делать запросы S3 с использованием `s3_curl`.

3.1.2 Включение статистики

Сбор статистики должен быть включен на шлюзе S3. Шлюз S3 будет сохранять статистику в виде обычных объектов хранилища. На каждом сервере хранилища S3 создайте файл `/var/lib/ostor/local/gw.conf` со следующим содержимым:

```
# Enable usage statistics collection.
S3_GW_COLLECT_STAT=1
```

Перезапустите сервис хранилища S3, чтобы применить изменения конфигурации. Выполните следующую команду на всех серверах хранилища S3:

```
# systemctl restart ostor-agentd.service
```

3.2 Управление пользователями S3 и вывод списка корзин через REST API

В этом разделе описывается, как управлять пользователями через REST API в сценарии поставщика услуг. Новые клиенты будут подписываться на сервис во время покупки в вашем онлайн-магазине, и для них потребуется создать пользователей в кластере S3.

3.2.1 Создание пользователей S3 через REST API

Пользователя S3 можно создать, отправив запрос PUT к сервису ostor-users вместе с адресом электронной почты.

```
# s3_curl PUT "http://s3.example.com/?ostor-users&emailAddress=user@example.com"
{
  "UserEmail": "user@example.com",
  "UserId": "ca55631f9f3d59dc",
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "ca55631f9f3d59dcDF4M",
      "AWSSecretAccessKey": "QCbj17BzeepyvUAdJeFNFYW9fCzbq0uFal6e5pGm"
    }
  ]
}
```

3.2.2 Вывод списка пользователей S3 через REST API

Информацию обо всех пользователях можно вывести, отправив запрос GET к сервису ostor-users. В дополнительных строках могут выводиться пары ключей доступа S3, связанные с каждым пользователем, например:

```
# s3_curl GET "http://s3.example.com/?ostor-users"
[
  {
    "UserEmail": "user@example.com",
    "UserId": "a14040e0b2ef8b28",
    "State": "enabled",
    "OwnerId": "0000000000000000"
  },
  {
    "UserEmail": "user@example.com",
    "UserId": "ca55631f9f3d59dc",

```

```
"State": "enabled",
"OwnerId": "0000000000000000"
}
]
```

3.2.3 Запрос пользователей S3 через REST API

Информацию о пользователе, его статус и все учетные записи можно отобразить, отправив запрос GET к сервису `ostor-users` вместе с адресом электронной почты пользователя.

```
# s3_curl GET "http://s3.example.com/?ostor-users&emailAddress=user@email.com"
{
  "UserEmail": "user@email.com",
  "UserId": "b09693b73b3c7686",
  "State": "enabled",
  "OwnerId": "0000000000000000",
  "Flags": [],
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "b09693b73b3c7686FIGH",
      "AWSSecretAccessKey": "jO2p4JBN1tWc4FEGxwZ8qW2jPCJBYp8RJ4KgBcZP"
    }
  ],
  "AccountCount": "3",
  "Accounts": [
    {
      "Name": "account1",
      "AWSAccessKeys": [
        {
          "AWSAccessKeyId": "b09693b73b3c768613NV",
          "AWSSecretAccessKey": "CBUpFmnpUGlXskTivgDQu4qjYksWpceGZeH6Qyct"
        }
      ]
    },
    {
      "Name": "account2",
      "AWSAccessKeys": [
        {
          "AWSAccessKeyId": "b09693b73b3c7686LCZ5",
          "AWSSecretAccessKey": "xLpUDFJMFM05rR9acAbUDplrPqIO6fneKNFjEB5c"
        },
        {
          "AWSAccessKeyId": "b09693b73b3c76866NI2",
          "AWSSecretAccessKey": "ajowU8pWSGW5ZJhA7AR9OjTrt11HmHPCJsMd247W"
        }
      ]
    },
    {
      "Name": "account3",
      "AWSAccessKeys": []
    }
  ]
}
```

```
]
}
```

3.2.4 Отключение и включение пользователей S3 через REST API

Пользователя можно отключить (по умолчанию пользователи включены), отправив запрос POST к сервису `ostor-users` вместе с адресом электронной почты пользователя и параметром `disable`.

```
# s3_curl POST "http://s3.example.com/?ostor-users&emailAddress=user@example.com&disable"
```

Ранее отключенного пользователя можно включить, отправив запрос POST к сервису `ostor-users` вместе с адресом электронной почты пользователя и параметром `enable`.

```
# s3_curl POST "http://s3.example.com/?ostor-users&emailAddress=user@example.com&enable"
```

3.2.5 Удаление пользователей и корзин S3 через REST API

Существующего пользователя можно удалить, отправив запрос DELETE к сервису `ostor-users` вместе с адресом электронной почты пользователя.

```
# s3_curl DELETE "http://s3.example.com/?ostor-users&emailAddress=user@example.com"
```

Пользователей, являющихся владельцами корзин, нельзя удалить, пока не будут удалены их корзины. Чтобы вывести список корзин пользователя, отправьте запрос GET к сервису `ostor-buckets` вместе с адресом электронной почты пользователя.

```
# s3_curl GET "http://s3.example.com/?ostor-buckets&emailAddress=user@example.com"
{
  "Buckets": [
    {
      "size": {
        "current": 12288,
        "h_integral": 7360512,
        "hmax": 12288,
        "last_ts": 424241
      },
      "epoch": 0,
      "owner_id": "ba7eba06129464c5",
      "name": "bucketname",
      "creation_date": "2018-05-25T17:12:00.000Z"
    }
  ]
}
```

Затем можно удалить корзины по имени.

```
# s3_curl DELETE "http://s3.example.com/bucketname"
```

3.2.6 Создание ключей доступа для пользователя S3 через REST API

Новые или дополнительные пары ключей доступа для указанного пользователя можно создать, отправив запрос POST к сервису ostor-users с адресом электронной почты пользователя и параметром genKey.

```
# s3_curl POST "http://s3.example.com/?ostor-users&emailAddress=user@example.com&genKey"
{
  "UserEmail": "user@example.com",
  "UserId": "ca55631f9f3d59dc",
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "ca55631f9f3d59dcZMDX",
      "AWSSecretAccessKey": "ffWvnOcNiH0jkQod4huv51BMYBuSWs4zRLFVwd4d"
    }
  ]
}
```

3.2.7 Отзыв ключей доступа для пользователя S3 через REST API

Указанную пару ключей доступа для указанного пользователя можно отозвать, отправив запрос POST к сервису ostor-users с адресом электронной почты пользователя и ключом доступа в паре ключей.

```
# s3_curl POST "http://s3.example.com/?ostor-
users&emailAddress=user@example.com&revokeKey=ca55631f9f3d59dcZMDX"
```

3.2.8 Вывод списка корзин пользователей через REST API

Вывести список всех корзин в S3 можно с помощью сервиса ostor-buckets.

```
# s3_curl GET "http://s3.example.com/?ostor-buckets"
{
  "Buckets": [
    {
      "size": {
        "current": 12288,
        "h_integral": 7360512,
        "hmax": 12288,
        "last_ts": 424241
      },
      "epoch": 0,
      "owner_id": "ba7eba06129464c5",
      "name": "bucket1",
      "creation_date": "2018-05-25T17:12:00.000Z"
    },
  ],
}
```

```

{
  "size": {
    "current": 46700160,
    "h_integral": 28160196480,
    "hmax": 46700160,
    "last_ts": 424237
  },
  "epoch": 0,
  "owner_id": "ccbec013d9fd3918",
  "name": "bucket2",
  "creation_date": "2018-05-25T13:51:55.000Z"
},
{
  "size": {
    "current": 12288,
    "h_integral": 8036352,
    "hmax": 12288,
    "last_ts": 424186
  },
  "epoch": 0,
  "owner_id": "9d80d59edbe2862a",
  "name": "bucket3",
  "creation_date": "2018-05-23T10:30:49.000Z"
}
}]

```

3.3 Управление учетными записями S3 через REST API

В этом разделе описывается, как управлять учетными записями S3 через REST API в сценарии поставщика услуг. Учетная запись S3 (проект) представляет собой изолированный контейнер корзины пользователя S3 с заданными лимитами использования. Учетная запись принадлежит одному пользователю, при этом у пользователя может быть несколько учетных записей.

3.3.1 Создание учетных записей S3 через REST API

Учетную запись для пользователя S3 можно создать, отправив запрос PUT к сервису `ostor-accounts` с адресом электронной почты пользователя и именем учетной записи.

```

# s3_curl POST "http://s3.example.com/?ostor-accounts&emailAddress=user@email.com&accountName=account"
{
  "Name": "account",
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "bc6265392b818465V8NZ",
      "AWSSecretAccessKey": "KcPbLkqOrbtLXCn3UgqAz3CEurRXtikrT9ZeDVFfa"
    }
  ]
}

```

3.3.2 Удаление учетных записей S3 через REST API

Существующую учетную запись пользователя S3 можно удалить, отправив запрос DELETE к сервису ostor-accounts с адресом электронной почты пользователя и именем учетной записи.

```
# s3_curl DELETE "http://s3.example.com/?ostor-accounts&emailAddress=user@example.com&accountName=account"
```

3.3.3 Создание ключей доступа для учетной записи S3 через REST API

Новые или дополнительные пары ключей доступа для указанной учетной записи пользователя S3 можно создать, отправив запрос POST к сервису ostor-users с адресом электронной почты пользователя, именем учетной записи и параметром genKey.

```
# s3_curl POST "http://s3.example.com/?ostor-users&emailAddress=user@email.com&accountName=account&genKey"
{
  "UserEmail": "user@email.com",
  "UserId": "b09693b73b3c7686",
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "b09693b73b3c7686Z8BU",
      "AWSSecretAccessKey": "m8PgWFLXPeJVSwojCE3DxWDoRk80g7CMYB7xK3Hd"
    }
  ]
}
```

3.3.4 Отзыв ключей доступа для учетной записи S3 через REST API

Указанную пару ключей доступа для указанной учетной записи пользователя S3 можно отозвать, отправив запрос POST к сервису ostor-users с адресом электронной почты пользователя, именем учетной записи и ключом доступа в паре ключей.

```
# s3_curl POST "http://s3.example.com/?ostor-users&emailAddress=user@example.com&accountName=account&revokeKey=ca55631f9f3d59dcZMDX"
```

3.4 Управление лимитами для пользователей и корзин S3 через REST API

В этом разделе описываются лимиты, которые можно задать для пользователей и корзин через REST API. Лимиты можно применять в соответствии с определенными параметрами, которые могут входить в ваш сервисный план.

3.4.1 Установка числа операций в секунду для пользователей через REST API

Скорость операций можно ограничить с помощью сервиса ostor-limits и следующих параметров: emailAddress с указанием адреса электронной почты, ops с указанием типа лимита и default=, get=, put=, list= или delete= с указанием значения лимита.

```
# s3_curl PUT "http://s3.example.com/?ostor-limits&emailAddress=client@example.com&limit-type=ops&limit-resource=get&limit-value=3600"
```

3.4.2 Установка пропускной способности в секунду для пользователей через REST API

Исходящую пропускную способность ответа можно ограничить с помощью сервиса ostor-limits и следующих параметров: emailAddress с указанием адреса электронной почты, bandwidth с указанием типа лимита и out= с указанием значения лимита.

```
# s3_curl PUT "http://s3.example.com/?ostor-limits&emailAddress=client@example.com&limit-type=bandwidth&limit-resource=out&limit-value=100"
```

3.4.3 Запрос лимитов пользователя через REST API

Текущие лимиты можно отобразить с помощью сервиса ostor-limits и параметра emailAddress с указанием адреса электронной почты.

```
# s3_curl GET "http://s3.example.com/?ostor-limits&emailAddress=client@example.com"
{
  "ops:default": "0.00",
  "ops:get": "3600.00",
  "ops:put": "0.00",
  "ops:list": "0.00",
  "ops:delete": "0.00",
  "bandwidth:out": "100"
}
```

3.4.4 Удаление лимитов пользователя через REST API

Текущие лимиты можно удалить с помощью сервиса ostor-limits и параметра emailAddress с указанием адреса электронной почты.

```
# s3_curl DELETE "http://s3.example.com/?ostor-limits&emailAddress=client@example.com"
```

3.4.5 Установка числа операций в секунду для корзин через REST API

Скорость операций можно ограничить с помощью сервиса ostor-limits и следующих параметров: bucket с указанием имени корзины, ops с указанием типа лимита и default=, get=, put=, list= или delete= с указанием значения лимита.

```
# s3_curl PUT "http://s3.example.com/?ostor-limits&bucket=client&limit-type=ops&limit-resource=get&limit-value=3600"
```

3.4.6 Установка пропускной способности в секунду для корзин через REST API

Исходящую пропускную способность ответа можно ограничить с помощью сервиса ostor-limits и следующих параметров: bucket с указанием имени корзины, bandwidth с указанием типа лимита и out= с указанием значения лимита.

```
# s3_curl PUT "http://s3.example.com/?ostor-limits&bucket=client&limit-type=bandwidth&limit-resource=out&limit-value=100"
```

3.4.7 Запрос лимитов корзины через REST API

Текущие лимиты можно отобразить с помощью сервиса ostor-limits и параметра bucket с указанием имени корзины.

```
# s3_curl GET "http://s3.example.com/?ostor-limits&bucket=client"
{
  "ops:default": "0.00",
  "ops:get": "3600.00",
  "ops:put": "0.00",
  "ops:list": "0.00",
  "ops:delete": "0.00",
  "bandwidth:out": "100"
}
```

3.4.8 Удаление лимитов корзины через REST API

Текущие лимиты можно удалить с помощью сервиса ostor-limits и параметра bucket с указанием имени корзины.

```
# s3_curl DELETE "http://s3.example.com/?ostor-limits&bucket=client"
```

3.5 Управление квотами для пользователей и корзин S3 через REST API

В этом разделе описываются квоты, которые можно задать для пользователей и корзин S3 через REST API. Эти квоты ограничивают использование хранилища объектов для отдельных пользователей или корзин. Можно применять их к пользователям или корзинам по отдельности либо задать квоты по умолчанию, которые будут применяться ко всем пользователям и корзинам S3.

Перед установкой квот на использование хранилища S3 выполните команду `ostor-ctl add-acc`, указав идентификатор тома и путь к хранилищу.

```
# ostor-ctl add-acc -V 0100000000000002 -s /mnt/vstorage/vols/ostor
```

3.5.1 Установка квот для пользователей через REST API

Можно ограничить использование хранилища для отдельных пользователей с помощью сервиса `ostor-quotas` и параметров `emailAddress`, указав адрес электронной почты, и `quota-size`, указав лимит использования в гигабайтах.

```
# s3_curl PUT "http://s3.example.com/?ostor-quotas&emailAddress=user@example.com&quota-size=1024"
```

3.5.2 Запрос квот для пользователя через REST API

Текущие квоты для отдельных пользователей можно отобразить с помощью сервиса `ostor-quotas` и параметра `emailAddress` с указанием адреса электронной почты.

```
# s3_curl GET "http://s3.example.com/?ostor-quotas&emailAddress=user@example.com"
{
  "version": "1",
  "type": "0",
  "size": "1024"
}
```

3.5.3 Удаление квот для пользователей через REST API

Текущие квоты для отдельных пользователей можно удалить с помощью сервиса `ostor-quotas` и параметра `emailAddress` с указанием адреса электронной почты.

```
# s3_curl DELETE "http://s3.example.com/?ostor-quotas&emailAddress=user@example.com"
```

3.5.4 Установка квот для корзин через REST API

Можно ограничить использование хранилища для отдельных корзин с помощью сервиса `ostor-quotas` и параметров `bucket`, указав имя корзины, и `quota-size`, указав лимит использования в гигабайтах.

```
# s3_curl PUT "http://s3.example.com/?ostor-quotas&bucket=bucket1&quota-size=256"
```

3.5.5 Запрос квот для корзины через REST API

Текущие квоты для отдельных корзин можно отобразить с помощью сервиса `ostor-quotas` и параметра `bucket` с указанием имени корзины.

```
# s3_curl GET "http://s3.example.com/?ostor-quotas&bucket=bucket1"
{
  "version": "1",
  "type": "1",
  "size": "256"
}
```

3.5.6 Удаление квот для корзины через REST API

Текущие квоты для отдельных корзин можно удалить с помощью сервиса `ostor-quotas` и параметра `bucket` с указанием имени корзины.

```
# s3_curl DELETE "http://s3.example.com/?ostor-quotas&bucket=bucket1"
```

3.5.7 Установка квот по умолчанию через REST API

Можно ограничить использование хранилища по умолчанию для всех пользователей и корзин с помощью сервиса `ostor-quotas` и параметров `default`, указав `user` для пользователей или `bucket` для корзин, и `quota-size`, указав лимит использования в гигабайтах.

```
# s3_curl PUT "http://s3.example.com/?ostor-quotas&default=user&quota-size=1024"
```

```
# s3_curl PUT "http://s3.example.com/?ostor-quotas&default=bucket&quota-size=256"
```

3.5.8 Запрос квот по умолчанию через REST API

Текущие квоты по умолчанию для всех пользователей или корзин можно отобразить с помощью сервиса `ostor-quotas` и параметра `default`, указав `user` для пользователей или `bucket` для корзин.

```
# s3_curl GET "http://s3.example.com/?ostor-quotas&default=user"
{
  "version": "1",
  "type": "0",
  "size": "1024"
}
```

```
# s3_curl GET "http://s3.example.com/?ostor-quotas&default=bucket"
{
  "version": "1",
  "type": "1",
  "size": "256"
}
```

3.5.9 Удаление квот по умолчанию через REST API

Текущие квоты по умолчанию для всех пользователей или корзин можно удалить с помощью сервиса `ostor-quotas` и параметра `default`, указав `user` для пользователей или `bucket` для корзин.

```
# s3_curl DELETE "http://s3.example.com/?ostor-quotas&default=user"
```

```
# s3_curl DELETE "http://s3.example.com/?ostor-quotas&default=bucket"
```

3.6 Получение статистики использования через REST API

В этом разделе описывается, как получить через REST API статистику использования для биллинга или других целей.

Примечание

Удалите объекты статистики после сбора необходимых данных.

3.6.1 Вывод списка объектов статистики через REST API

Вывести список всех доступных объектов статистики можно с помощью сервиса `ostor-usage` без параметров. Выходные данные содержат только объекты статистики, которые не были удалены, например:

```
# s3_curl GET "http://s3.example.com/?ostor-usage"
{
  "nr_items": 7,
  "truncated": false,
  "items": [
    "s3-usage-80000000000000065-2017-02-01T16:31:54.000Z-1800",
  ]
}
```

```

"s3-usage-80000000000000067-2017-02-01T16:30:51.000Z-1800",
"s3-usage-80000000000000068-2017-02-01T16:27:25.000Z-1800",
"s3-usage-80000000000000069-2017-02-01T16:27:24.000Z-1800",
"s3-usage-80000000000000069-2017-02-01T16:31:07.000Z-1800",
"s3-usage-8000000000000006a-2017-02-01T16:27:24.000Z-1800",
"s3-usage-8000000000000006a-2017-02-01T16:31:08.000Z-1800"
]
}

```

3.6.2 Запрос объектов статистики через REST API

Статистику использования можно отобразить с помощью сервиса `ostor-usage` и параметра `obj` с указанием объекта статистики. Выходные данные включают корзины, к которым осуществлялся доступ, идентификатор пользователя и счетчики, например:

```

# s3_curl GET "http://s3.example.com/?ostor-usage&obj=s3-usage-80000000000000065-2017-02-01T16:31:54.000Z-1800"
{
  "fmt_version": 1,
  "service_id": 80000000000000065,
  "start_ts": 1485966714,
  "period": 1390,
  "nr_items": 1,
  "items": [
    {
      "key": {
        "bucket": "client",
        "epoch": 98309,
        "user_id": "b81d6c5f895a8c86",
        "tag": ""
      },
      "counters": {
        "ops": {
          "put": 1,
          "get": 3,
          "list": 0,
          "other": 0
        },
        "net_io": {
          "uploaded": 41258,
          "downloaded": 45511311
        }
      }
    }
  ]
}

```

3.6.3 Удаление объектов статистики через REST API

Существующие объекты статистики можно удалить с помощью сервиса `ostor-usage` и параметра `obj` с указанием объекта статистики.

```
# s3_curl DELETE "http://s3.example.com/?ostor-usage&obj=s3-usage-8000000000000065-2017-02-01T16:31:54.000Z-1800"
```

4 Интеграция с WHMCS

В этой главе объясняется, как выполнять подготовку, включение, отключение и удаление пользователей S3, а также задавать лимиты пользователей и корзин в целях биллинга.

Приведенные примеры представляют собой скрипты PHP, с помощью которых можно отправлять запросы к REST API кластера S3 через cURL и OpenSSL.

Примечание

Замените `http://s3.example.com` в примерах фактическим URL-адресом вашего шлюза S3, а `http://whmcs.example.com` – фактическим URL-адресом портала WHMCS.

4.1 Требования

Любой запрос на операцию или управление должен пройти проверку подлинности посредством подписанного запроса через Signature Version 2 или 4 протокола Amazon S3 от соответствующего системного пользователя S3. Для проверки подлинности API-запросов необходимо создать системного пользователя. Сначала получите идентификатор тома с помощью команды `ostor-ctl get-config`, например:

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

Затем создайте системного пользователя на любом сервере хранилища в кластере с помощью команды `ostor-s3-admin create-user -S -e <email>`, например:

```
# ostor-s3-admin create-user -S -e user@example.com -V 0100000000000002
UserEmail:user@example.com
UserId:a14040e0b2ef8b28
KeyPair[0]:S3AccessKeyId:a14040e0b2ef8b28FZZ8
KeyPair[0]:S3SecretAccessKey:dbwTnQTW602aAAdq8DQVFzB6yrTCFTNiGB8C8RFA
Flags:system
```

С этим пользователем можно выполнять аутентификацию последующих API-запросов для управления кластером S3. Можно создать несколько системных учетных записей для разных типов операций управления.

4.1.1 Настройка

Дополнительно необходимо создать каталоги продукта Кибер Инфраструктура для изменения функций по умолчанию.

Перейдите в корневой каталог документов сервера WHMCS (например, `/srv/http`) и создайте в нем следующие каталоги:

- whmcs/includes/staas_scripts
- whmcs/admin/staas_scripts

Перейдите в каталог whmcs/includes/staas_scripts.

Первый файл, который необходимо создать, содержит конфигурацию S3. Создайте файл конфигурации S3_getConfig.php со следующим содержимым, заменяя переменные следующим образом:

- s3_key на ваш идентификатор S3AccessKeyId,
- s3_secret на ваш ключ S3SecretAccessKey,
- s3_gateway на настроенный адрес вашего шлюза S3,
- whmcs_username на имя пользователя администратора WHMCS.

```
<?php

// Return array with default configuration.
if (!function_exists('S3_getConfig')) {
    function S3_getConfig() {

        // s3 login.
        $vars['s3_key'] = "939e2ac6916b57082P9O";
        $vars['s3_secret'] = "tVYF3kZD9zcTtl6q6QDTHaZKM2nuq4xVcl8ikJpd";

        // s3 gateway.
        $vars['s3_gateway'] = "http://s3.example.com";

        // whmcs login.
        $vars['whmcs_username'] = "admin";

        // Return config array.
        return $vars;
    }
}

?>
```

4.1.2 Файлы include

Общие функции, необходимые для API-операций, предоставляются в отдельных PHP-файлах include. Первый файл возвращает информацию о клиенте (например, адрес электронной почты), которая требуется дальнейшим API-запросам управления пользователями S3 для различных операций. Создайте файл S3_getClient.php со следующим содержимым:

```
<?php

// API request to get whmcs client information.
if (!function_exists('S3_getClient')) {
```

```

function S3_getClient($userid, $whmcs_username) {

    // Get client details for user email.
    $command = 'GetClientsDetails';
    $data = array(
        'clientid' => $userid,
    );
    $results = localAPI($command, $data, $whmcs_username);

    // Return client information.
    return $results;
}
}

?>

```

Следующий файл добавляет к клиенту в WHMCS примечания с парами ключей доступа S3 при каждом создании нового пользователя или пары ключей. Создайте файл S3_addClientNote.php со следующим содержимым:

```

<?php

// API request to add note to client in whmcs.
if (!function_exists('S3_addClientNote')) {
    function S3_addClientNote(
        $userid,
        $whmcs_username,
        $s3_client_userid,
        $s3_client_key,
        $s3_client_secret
    ) {

        // Add note only for non-empty users.
        if (!empty($s3_client_userid)) {

            // Add note with the s3 access key and s3 secret.
            $command = 'AddClientNote';
            $data = array(
                'userid' => $userid,
                'notes' =>
                    "UserId: " . $s3_client_userid . "\n" .
                    "AWSAccessKeyId: " . $s3_client_key . "\n" .
                    "AWSSecretAccessKey: " . $s3_client_secret,
            );
            localAPI($command, $data, $whmcs_username);
        }
    }
}

?>

```

Следующий файл удаляет из клиента в WHMCS примечания с парами ключей доступа S3 при каждом удалении пользователя или пары ключей. Создайте файл S3_delClientNote.php со следующим содержимым:

```
<?php

// whmcs database access.
use WHMCS\Database\Capsule;

// API request to remove note from client in whmcs.
if (!function_exists('S3_delClientNote')) {
    function S3_delClientNote(
        $userid,
        $whmcs_username,
        $s3_client_userid,
        $s3_client_key
    ) {

        // Delete notes in database.
        $db = Capsule::connection()->getPdo();
        $db->exec('
            DELETE FROM
            tblnotes
            WHERE
            userid = ' . $userid . '
            AND
            note LIKE "%' . $s3_client_userid . '%"
            AND
            note LIKE "%' . $s3_client_key . '%"
        ');
    }
}

?>
```

Последний файл представляет собой библиотеку cURL для отправки запросов GET, PUT, POST и DELETE. Создайте файл S3_requestCurl.php со следующим содержимым:

```
<?php

// API request to s3 gateway.
if (!function_exists('S3_requestCurl')) {
    function S3_requestCurl($s3_key, $s3_secret, $s3_gateway, $s3_query, $method) {

        // Prepare signature.
        $s3_host = parse_url($s3_gateway, PHP_URL_HOST);
        $s3_date = date(DATE_RFC2822);

        // Generate signature.
        $s3_signature = hash_hmac('sha1', $method . "\n\n\n" . $s3_date . "\n" .
            current(explode('&', $s3_query)), $s3_secret, true);
    }
}

?>
```

```

$s3_signature = base64_encode($s3_signature);

// Curl init.
$s3_curl = curl_init($s3_gateway . $s3_query);

// Curl options.
switch ($method) {
    case "PUT":
        curl_setopt($s3_curl, CURLOPT_PUT, 1);
        break;
    case "POST":
        curl_setopt($s3_curl, CURLOPT_POST, 1);
        break;
    case "DELETE":
        curl_setopt($s3_curl, CURLOPT_CUSTOMREQUEST, "DELETE");
        break;
}
curl_setopt($s3_curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($s3_curl, CURLOPT_URL, $s3_gateway . $s3_query);
curl_setopt($s3_curl, CURLOPT_HTTPHEADER, array(
    'Host: ' . $s3_host,
    'Date: ' . $s3_date,
    'Authorization: AWS ' . $s3_key . ':' . $s3_signature,
    'Content-Type:',
    'Expect:',
));

// Call.
$response = curl_exec($s3_curl);
$response = json_decode($response, true);

// Curl deinit.
curl_close($s3_curl);

// Return response.
return $response;
}
}
?>

```

4.1.3 Статистика

Сбор статистики должен быть включен на шлюзе S3. Шлюз S3 будет сохранять статистику в виде обычных объектов хранилища. На каждом сервере хранилища S3 создайте файл `/var/lib/ostor/local/gw.conf` со следующим содержимым:

```

# Enable usage statistics collection.
S3_GW_COLLECT_STAT=1

```

Перезапустите сервис хранилища S3, чтобы применить изменения конфигурации. Выполните следующую команду на всех серверах хранилища S3:

```
# systemctl restart ostor-agentd.service
```

Теперь можно выполнить вход в WHMCS. Дополнительные ссылки и параметры управления S3 будут отображаться в разделе **Client Profile** (Профиль клиента).

4.2 Управление пользователями S3 в WHMCS

В этом разделе описывается, как управлять пользователями через WHMCS в сценарии поставщика услуг. Новые клиенты будут подписываться на сервис во время покупки в вашем онлайн-магазине, и для них потребуется создать пользователей в кластере S3.

Создавайте все файлы, упомянутые далее, в каталоге `whmcs/admin/staas_scripts`.

4.2.1 Создание пользователей S3 в WHMCS

Создать пользователя можно с помощью сервиса `ostor-users` и параметра `emailAddress` с указанием адреса электронной почты пользователя. WHMCS создаст пользователя в кластере S3 при нажатии ссылки **Create User** (Создать пользователя). Создайте файл `S3_createUser.php` со следующим содержимым:

```
<?php

// Load configuration and libraries.
require('../includes/staas_scripts/S3_addClientNote.php');
require('../includes/staas_scripts/S3_getClient.php');
require('../includes/staas_scripts/S3_getConfig.php');
require('../includes/staas_scripts/S3_requestCurl.php');
require('../init.php');

// Create s3 user.
function S3_createUser($userid) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get whmcs user email.
    $s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

    // Create s3 user.
    $s3_client = S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-users&emailAddress=" . $s3_whmcs['email'],
        "PUT"
    );
}
```

```

// Add note with the s3 access key and s3 secret.
S3_addClientNote(
    $s3_whmcs['userid'],
    $s3_config['whmcs_username'],
    $s3_client['UserId'],
    $s3_client['AWSAccessKeys']['0']['AWSAccessKeyId'],
    $s3_client['AWSAccessKeys']['0']['AWSSecretAccessKey']
);

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_createUser($_GET['userid']);

?>

```

4.2.2 Вывод списка пользователей S3 в WHMCS

Информацию обо всех пользователях можно вывести с помощью сервиса `ostor-users`. В дополнительных строках могут выводиться пары ключей доступа S3, связанные с пользователем. WHMCS выведет информацию о пользователях, полученную из кластера S3, при нажатии ссылки **List Users (on/off)** (Список пользователей (вкл./выкл.)). Создайте файл `S3_listUsers.php` со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// List s3 users.
function S3_listUsers() {

    // Hide now.
    if ($_SESSION['s3_list_users'] == 1) {

        // Hide.
        $_SESSION['s3_list_users'] = 0;

        // Redirect back.
        header('Location: ' . $_SERVER['HTTP_REFERER']);

        // Return immediately.
        return;
    }
}

```

```

// Load configuration.
$s3_config = s3_getConfig();

// Get s3 users.
$s3_client = S3_requestCurl(
    $s3_config['s3_key'],
    $s3_config['s3_secret'],
    $s3_config['s3_gateway'],
    "?ostor-users",
    "GET"
);

// Store s3 result.
$_SESSION['s3_list_users'] = 1;
$_SESSION['s3_list'] = $s3_client;

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_listUsers();

?>

```

4.2.3 Запрос пользователей S3 в WHMCS

Информацию о пользователе S3 и его статус можно отобразить с помощью сервиса `ostor-users` и параметра `emailAddress` с указанием адреса электронной почты пользователя. WHMCS отобразит информацию о пользователе, полученную из кластера S3, при нажатии ссылки **Query User (on/off)** (Запрос пользователя (вкл./выкл.)). Создайте файл `S3_queryUser.php` со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_getClient.php');
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// Query s3 user.
function S3_queryUser($userid) {

    // Hide now.
    if ($_SESSION['s3_query_user'] == 1) {

        // Hide.
        $_SESSION['s3_query_user'] = 0;

        // Redirect back.
    }
}

```

```

    header('Location: ' . $_SERVER['HTTP_REFERER']);

    // Return immediately.
    return;
}

// Load configuration.
$s3_config = s3_getConfig();

// Get whmcs user email.
$s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

// Get s3 user id.
$s3_client = S3_requestCurl(
    $s3_config['s3_key'],
    $s3_config['s3_secret'],
    $s3_config['s3_gateway'],
    "?ostor-users&emailAddress=" . $s3_whmcs['email'],
    "GET"
);

// Store s3 result.
$_SESSION['s3_query_user'] = 1;
$_SESSION['s3_userid'] = $s3_client['UserId'];
$_SESSION['s3_aws_access_keys'] = $s3_client['AWSAccessKeys'];

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_queryUser($_GET['userid']);

?>

```

4.2.4 Отключение пользователей S3 в WHMCS

Отключить пользователя можно с помощью сервиса `ostor-users` и параметра `emailAddress` с указанием адреса электронной почты пользователя. WHMCS отключит доступ к кластеру S3 для чтения и записи при нажатии ссылки **Disable User** (Отключить пользователя). Создайте файл `S3_disableUser.php` со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/taas_scripts/S3_getClient.php');
require('../../includes/taas_scripts/S3_getConfig.php');
require('../../includes/taas_scripts/S3_requestCurl.php');
require('../../init.php');

```



```

// Disable user.
function S3_disableUser($userid) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get whmcs user email.
    $s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

    // Disable user.
    $s3_client = S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-users&emailAddress=" . $s3_whmcs['email'] . "&disable",
        "POST"
    );

    // Redirect back.
    header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_disableUser($_GET['userid']);

?>

```

4.2.5 Включение пользователей S3 в WHMCS

Ранее отключенного пользователя можно включить с помощью сервиса `ostor-users` и параметра `emailAddress` с указанием адреса электронной почты пользователя. WHMCS включит доступ к кластеру S3 для чтения и записи при нажатии ссылки **Enable User** (Включить пользователя).

Создайте файл `S3_enableUser.php` со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../includes/staas_scripts/S3_getClient.php');
require('../includes/staas_scripts/S3_getConfig.php');
require('../includes/staas_scripts/S3_requestCurl.php');
require('../init.php');

// Enable user.
function S3_enableUser($userid) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get whmcs user email.
    $s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

```

```

// Enable user.
$s3_client = S3_requestCurl(
    $s3_config['s3_key'],
    $s3_config['s3_secret'],
    $s3_config['s3_gateway'],
    "?ostor-users&emailAddress=" . $s3_whmcs['email'] . "&enable",
    "POST"
);

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_enableUser($_GET['userid']);

?>

```

4.2.6 Удаление пользователей S3 в WHMCS

Удалить пользователя можно с помощью сервиса `ostor-users` и параметра `emailAddress` с указанием адреса электронной почты пользователя. WHMCS удалит пользователя из кластера S3 при нажатии ссылки **Delete User** (Удалить пользователя). Создайте файл `S3_deleteUser.php` со следующим содержанием:

```

<?php

// Load configuration and libraries.
require('../includes/staas_scripts/S3_delClientNote.php');
require('../includes/staas_scripts/S3_getClient.php');
require('../includes/staas_scripts/S3_getConfig.php');
require('../includes/staas_scripts/S3_requestCurl.php');
require('../init.php');

// Delete s3 user.
function S3_deleteUser($userid) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get whmcs user email.
    $s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

    // Get s3 user id.
    $s3_client = S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-users&emailAddress=" . $s3_whmcs['email'],

```

```

    "GET"
);

// Delete s3 user.
S3_requestCurl(
    $s3_config['s3_key'],
    $s3_config['s3_secret'],
    $s3_config['s3_gateway'],
    "?ostor-users&emailAddress=" . $s3_whmcs['email'],
    "DELETE"
);

// Delete note with the s3 access key and s3 secret.
S3_delClientNote(
    $s3_whmcs['userid'],
    $s3_config['whmcs_username'],
    $s3_client['UserId'],
    ""
);

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_deleteUser($_GET['userid']);

?>

```

4.2.7 Создание ключей доступа S3 в WHMCS

Создать новые или дополнительные пары ключей доступа можно с помощью сервиса ostor-users и следующих параметров: emailAddress с указанием адреса электронной почты пользователя, genKey. WHMCS сформирует новую пару ключей при нажатии ссылки **Generate Access Key** (Сформировать ключ доступа). Создайте файл S3_generateAccessKey.php со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_addClientNote.php');
require('../../includes/staas_scripts/S3_getClient.php');
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// Generate s3 access key pair.
function S3_generateAccessKey($userid) {

    // Load configuration.

```

```

$s3_config = s3_getConfig();

// Get whmcs user email.
$s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

// Generate s3 key pair.
$s3_client = S3_requestCurl(
    $s3_config['s3_key'],
    $s3_config['s3_secret'],
    $s3_config['s3_gateway'],
    "?ostor-users&emailAddress=" . $s3_whmcs['email'] . "&genKey",
    "POST"
);

// Add note with the s3 access key and s3 secret.
S3_addClientNote(
    $s3_whmcs['userid'],
    $s3_config['whmcs_username'],
    $s3_client['UserId'],
    $s3_client['AWSAccessKeys']['0']['AWSAccessKeyId'],
    $s3_client['AWSAccessKeys']['0']['AWSSecretAccessKey']
);

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_generateAccessKey($_GET['userid']);

?>

```

4.2.8 Отзыв ключей доступа S3 в WHMCS

Отозвать указанную пару ключей доступа для указанного пользователя можно с помощью сервиса `ostor-users` и следующих параметров: `emailAddress` с указанием адреса электронной почты пользователя, `revokeKey` с указанием ключа доступа в паре ключей. WHMCS удалит пару ключей при нажатии ссылки **Revoke Access Key** (Отозвать ключ доступа). Создайте файл `S3_revokeAccessKey.php` со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../includes/staas_scripts/S3_delClientNote.php');
require('../includes/staas_scripts/S3_getClient.php');
require('../includes/staas_scripts/S3_getConfig.php');
require('../includes/staas_scripts/S3_requestCurl.php');
require('../init.php');

// Revoke s3 access key pair.

```

```

function S3_revokeAccessKey($userid) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get whmcs user email.
    $s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

    // Get first s3 access key.
    $s3_client = S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-users&emailAddress=" . $s3_whmcs['email'],
        "GET"
    );

    // Revoke s3 access key.
    S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-users&emailAddress=" . $s3_whmcs['email'] .
        "&revokeKey=" . $s3_client['AWSAccessKeys'][0]['AWSAccessKeyId'],
        "POST"
    );

    // Delete note with the s3 access key and s3 secret.
    S3_delClientNote(
        $s3_whmcs['userid'],
        $s3_config['whmcs_username'],
        $s3_client['UserId'],
        $s3_client['AWSAccessKeys'][0]['AWSAccessKeyId']
    );

    // Redirect back.
    header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_revokeAccessKey($_GET['userid']);

?>

```

4.3 Управление лимитами для пользователей и корзин S3 в WHMCS

В этом разделе описываются лимиты, которые можно задать для пользователей и корзин в WHMCS. Лимиты можно применять в соответствии с определенными параметрами, которые могут входить в ваш сервисный план.

4.3.1 Установка лимитов пользователя в WHMCS

Скорость операций можно ограничить с помощью сервиса ostor-limits и следующих параметров: emailAddress с указанием адреса электронной почты, default=, get=, put=, list= или delete= с указанием значения лимита.

Аналогичным образом можно ограничить исходящую пропускную способность ответа с помощью следующих параметров: emailAddress с указанием адреса электронной почты, out= с указанием значения лимита. WHMCS настроит лимиты пользователя в кластере S3 при нажатии кнопки **Set** (Установить). Создайте файл S3_setLimitsForUser.php со следующим содержимым:

```
<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_getClient.php');
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// Set s3 user limits.
function S3_setLimitsForUser($vars) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get whmcs user email.
    $s3_whmcs = S3_getClient($vars['userid'], $s3_config['whmcs_username']);

    // Set only if value specified.
    if (!empty($vars['ops-value'])) {

        // Set s3 bucket limits (ops).
        S3_requestCurl(
            $s3_config['s3_key'],
            $s3_config['s3_secret'],
            $s3_config['s3_gateway'],
            "?ostor-limits&emailAddress=" . $s3_whmcs['email'] .
            "&limit-type=ops&limit-resource=" . $vars['ops-name'] .
            '&limit-value=' . $vars['ops-value'],
            "PUT"
        );
    }

    // Set only if value specified.
    if (!empty($vars['bandwidth-value'])) {

        // Set s3 bucket limits (bandwidth).
        S3_requestCurl(
            $s3_config['s3_key'],
            $s3_config['s3_secret'],
```

```

    $s3_config['s3_gateway'],
    "?ostor-limits&emailAddress=" . $s3_whmcs['email'] .
    "&limit-type=bandwidth&limit-resource=" . $vars['bandwidth-name'] .
    '&limit-value=' . $vars['bandwidth-value'],
    "PUT"
);
}

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_setLimitsForUser($_GET);

?>

```

4.3.2 Запрос лимитов пользователя в WHMCS

Текущие лимиты можно отобразить с помощью сервиса ostor-limits и параметра emailAddress с указанием адреса электронной почты. WHMCS отобразит лимиты пользователя в кластере S3 при нажатии кнопки **Get** (Получить). Создайте файл S3_getLimitsForUser.php со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_getClient.php');
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// Get s3 user limits.
function S3_getLimitsForUser($userid) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get whmcs user email.
    $s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

    // Get s3 user limits.
    $s3_client = S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-limits&emailAddress=" . $s3_whmcs['email'],
        "GET"
    );
}

```

```

// Store s3 result.
$_SESSION['s3_limits_user'] = $s3_client;

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_getLimitsForUser($_GET['userid']);

?>

```

4.3.3 Удаление лимитов пользователя в WHMCS

Текущие лимиты можно удалить с помощью сервиса ostor-limits и параметра emailAddress с указанием адреса электронной почты. WHMCS удалит лимиты пользователя из кластера S3 при нажатии кнопки **Delete** (Удалить). Создайте файл S3_deleteLimitsForUser.php со следующим содержанием:

```

<?php

// Load configuration and libraries.
require('../includes/staas_scripts/S3_getClient.php');
require('../includes/staas_scripts/S3_getConfig.php');
require('../includes/staas_scripts/S3_requestCurl.php');
require('../init.php');

// Delete s3 user limits.
function S3_getLimitsForUser($userid) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get whmcs user email.
    $s3_whmcs = S3_getClient($userid, $s3_config['whmcs_username']);

    // Delete s3 user limits.
    S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-limits&emailAddress=" . $s3_whmcs['email'],
        "DELETE"
    );

    // Clear array.
    $_SESSION['s3_limits_user'] = null;

    // Redirect back.
    header('Location: ' . $_SERVER['HTTP_REFERER']);
}

```



```

}

// Call function.
S3_getLimitsForUser($_GET['userid']);

?>

```

4.3.4 Установка лимитов корзины в WHMCS

Скорость операций можно ограничить с помощью сервиса ostor-limits и следующих параметров: bucket с указанием имени корзины, default=, get=, put=, list= или delete= с указанием значения лимита.

Аналогичным образом можно ограничить исходящую пропускную способность ответа с помощью сервиса ostor-limits и следующих параметров: bucket с указанием имени корзины, out= с указанием значения лимита. WHMCS настроит лимиты корзины в кластере S3 при нажатии кнопки **Set** (Установить). Создайте файл S3_setLimitsForBucket.php со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// Set s3 bucket limits.
function S3_setLimitsForBucket($vars) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Set only if value specified.
    if (!empty($vars['ops-value'])) {

        // Set s3 bucket limits (ops).
        S3_requestCurl(
            $s3_config['s3_key'],
            $s3_config['s3_secret'],
            $s3_config['s3_gateway'],
            "?ostor-limits&bucket=" . $vars['bucket'] .
            "&limit-type=ops&limit-resource=" . $vars['ops-name'] .
            '&limit-value=' . $vars['ops-value'],
            "PUT"
        );
    }

    // Set only if value specified.
    if (!empty($vars['bandwidth-value'])) {

        // Set s3 bucket limits (bandwidth).
    }
}

```

```

S3_requestCurl(
    $s3_config['s3_key'],
    $s3_config['s3_secret'],
    $s3_config['s3_gateway'],
    "?ostor-limits&bucket=" . $vars['bucket'] .
    "&limit-type=bandwidth&limit-resource=" . $vars['bandwidth-name'] .
    '&limit-value=' . $vars['bandwidth-value'],
    "PUT"
);
}

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_setLimitsForBucket($_GET);

?>

```

4.3.5 Запрос лимитов корзины в WHMCS

Текущие лимиты можно отобразить с помощью сервиса `ostor-limits` и параметра `bucket` с указанием имени корзины. WHMCS отобразит лимиты корзины в кластере S3 при нажатии кнопки **Get** (Получить). Создайте файл `S3_getLimitsForBucket.php` со следующим содержанием:

```

<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// Get s3 bucket limits.
function S3_getLimitsForBucket($bucket) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get s3 user limits.
    $s3_client = S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-limits&bucket=" . $bucket,
        "GET"
    );

    // Store s3 result.
    $_SESSION['s3_limits_bucket'] = $s3_client;
}

```

```

$_SESSION['s3_bucket'] = $bucket;

// Redirect back.
header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_getLimitsForBucket($_GET['bucket']);

?>

```

4.3.6 Удаление лимитов корзины в WHMCS

Текущие лимиты можно удалить с помощью сервиса `ostor-limits` и параметра `bucket` с указанием имени корзины. WHMCS удалит лимиты корзины из кластера S3 при нажатии кнопки **Delete** (Удалить). Создайте файл `S3_deleteLimitsForBucket.php` со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// Delete s3 bucket limits.
function S3_deleteLimitsForBucket($bucket) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Delete s3 bucket limits.
    S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-limits&bucket=" . $bucket,
        "DELETE"
    );

    // Clear array.
    $_SESSION['s3_limits_bucket'] = null;

    // Redirect back.
    header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_deleteLimitsForBucket($_GET['bucket']);

?>

```

4.4 Получение статистики использования в WHMCS

В этом разделе описывается, как получить в WHMCS статистику использования для биллинга или других целей.

Примечание

Удалите объекты статистики после сбора необходимых данных.

4.4.1 Вывод списка объектов статистики в WHMCS

Вывести список всех доступных объектов статистики можно с помощью сервиса `ostor-usage` без параметров. Выходные данные содержат только объекты статистики, которые не были удалены. WHMCS выведет список доступных объектов статистики в кластере S3 при нажатии ссылки **List statistics objects (on/off)** (Список объектов статистики (вкл./выкл.)). Создайте файл `S3_listStatsObjects.php` со следующим содержимым:

```
<?php

// Load configuration and libraries.
require('../includes/staas_scripts/S3_getConfig.php');
require('../includes/staas_scripts/S3_requestCurl.php');
require('../init.php');

// List s3 statistics objects.
function S3_listStatsObjects() {

    // Hide now.
    if ($_SESSION['s3_stat_objects'] == 1) {

        // Hide.
        $_SESSION['s3_stat_objects'] = 0;

        // Redirect back.
        header('Location: ' . $_SERVER['HTTP_REFERER']);

        // Return immediately.
        return;
    }

    // Load configuration.
    $s3_config = s3_getConfig();

    // Get s3 statistics objects.
    $s3_client = S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-usage",
```

```

    "GET"
  );

  // Store s3 result.
  $_SESSION['s3_stat_objects'] = 1;
  $_SESSION['s3_stat'] = $s3_client;

  // Redirect back.
  header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_listStatsObjects();

?>

```

4.4.2 Запрос объектов статистики в WHMCS

Статистику использования можно отобразить с помощью сервиса ostor-usage и параметра obj с указанием объекта статистики. WHMCS отобразит корзины, к которым осуществлялся доступ, идентификатор пользователя и счетчики при нажатии кнопки **Get** (Получить). Создайте файл S3_getStatsForObject.php со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/taas_scripts/S3_getConfig.php');
require('../../includes/taas_scripts/S3_requestCurl.php');
require('../../init.php');

// Get s3 statistics object.
function S3_getStatsObjects($object) {

  // Load configuration.
  $s3_config = s3_getConfig();

  // Get s3 statistics object.
  $s3_client = S3_requestCurl(
    $s3_config['s3_key'],
    $s3_config['s3_secret'],
    $s3_config['s3_gateway'],
    "?ostor-usage&obj=" . $object,
    "GET"
  );

  // Store s3 result.
  $_SESSION['s3_object_statistic'] = $s3_client;
  $_SESSION['s3_object'] = $object;

  // Redirect back.

```

```

header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_getStatsObjects($_GET['object']);

?>

```

4.4.3 Удаление объектов статистики в WHMCS

Существующие объекты статистики можно удалить с помощью сервиса ostor-usage и параметра obj с указанием объекта статистики. WHMCS удалит объект статистики из кластера S3 при нажатии кнопки **Delete** (Удалить). Создайте файл S3_deleteStatsForObject.php со следующим содержимым:

```

<?php

// Load configuration and libraries.
require('../../includes/staas_scripts/S3_getConfig.php');
require('../../includes/staas_scripts/S3_requestCurl.php');
require('../../init.php');

// Delete s3 statistics object.
function S3_deleteStatsForObject($object) {

    // Load configuration.
    $s3_config = s3_getConfig();

    // Delete s3 statistics object.
    S3_requestCurl(
        $s3_config['s3_key'],
        $s3_config['s3_secret'],
        $s3_config['s3_gateway'],
        "?ostor-usage&obj=" . $object,
        "DELETE"
    );

    // Clear array.
    $_SESSION['s3_limits_bucket'] = null;

    // Redirect back.
    header('Location: ' . $_SERVER['HTTP_REFERER']);
}

// Call function.
S3_deleteStatsForObject($_GET['object']);

?>

```