

S3 в частной инсталляции

Руководство пользователя

Оглавление

Термины и сокращения	6
1. Введение	7
1.1. Область применения	7
1.2. Краткое описание возможностей	7
1.3. Требования к квалификации	7
1.4. Перечень эксплуатационной документации	8
2. Инструменты для работы с бакетами и объектами S3	9
2.1. S3 CLI	9
2.2. Файловые менеджеры для S3	9
3. Создание, настройка и работа с бакетами S3	11
3.1. Правила именования бакетов	11
3.1.1. Примеры названий бакетов	11
3.2. Создание бакета	12
3.2.1. Создание бакета при помощи S3 CLI	12
3.2.2. Создание бакета при помощи файлового менеджера	12
3.3. Просмотр свойств бакета S3	14
3.3.1. Просмотр свойств бакета при помощи файлового менеджера	14
3.4. Очистка бакета	16
3.4.1. Очистка бакета при помощи S3 CLI	16
3.5. Удаление бакета	16
3.5.1. Удаление бакета при помощи S3 CLI	17
3.5.2. Удаление бакета при помощи файлового менеджера	17
3.6. Ограничения и лимиты для бакетов	18
3.6.1. Ограничения	18
3.6.2. Лимиты	18
4. Выгрузка, загрузка и работа с объектами в S3	20
4.1. Создание имен (ключей) объектов	20
4.1.1. Рекомендации по созданию имен (ключей) объекта	21
4.2. Загрузка объектов	22
4.2.1. Загрузка объекта в бакет при помощи AWS CLI	23
4.2.2. Загрузка объекта в бакет при помощи файлового менеджера S3	23
4.3. Загрузка объектов с помощью многокомпонентной загрузки	26
4.3.1. Процесс многокомпонентной загрузки	27
4.3.2. Загрузка объекта с помощью многокомпонентной загрузки с помощью AWS CLI ..	28
4.4. Копирование объектов	29

4.5. Скачивание объекта	29
4.5.1. Скачивание объекта при помощи AWS CLI	29
4.5.2. Скачивание объекта при помощи файлового менеджера	30
4.6. Удаление объектов	32
4.6.1. Удаление объектов при помощи AWS CLI	32
4.6.2. Удаление объектов при помощи файлового менеджера	32
4.7. Организация, перечисление и работа с объектами	34
4.8. Использование предварительно подписанных URL-адресов	35
4.8.1. Ограничение возможностей предварительно подписанного URL	35
4.8.2. Создание предварительно подписанного URL-адреса при помощи AWS CLI	36
4.8.3. Создание предварительно подписанного URL-адреса при помощи файлового менеджера	36
5. API	40
5.1. Операции с бакетами	40
5.1.1. CreateBucket	40
5.1.2. ListBuckets	41
5.1.3. HeadBucket	42
5.1.4. ListObjects	42
5.1.5. DeleteBucket	44
5.2. Операции с объектами	44
5.2.1. Get	44
5.2.2. Upload	45
5.2.3. Copy	46
5.2.4. HeadObject	47
5.2.5. Delete	48
5.2.6. DeleteMultipleObjects	49
5.3. Списки управления доступом (ACL)	50
5.3.1. Get Bucket ACL	50
5.3.2. Put Bucket ACL	51
5.3.3. Get Object ACL	52
5.3.4. Put Object ACL	53
5.4. Составная загрузка	54
5.4.1. Initiate Multipart Upload	55
5.4.2. Upload Part	56
5.4.3. Complete Multipart Upload	57
5.4.4. Abort Multipart Upload	58
5.4.5. List Parts	58

5.5. Управление жизненным циклом	60
5.5.1. Общий вид конфигурации	60
5.5.2. Get Bucket Lifecycle Configuration	60
5.5.3. Configure Bucket Lifecycle	61
5.5.4. Delete Bucket Lifecycle	62
5.6. Совместное использование ресурсов между разными источниками	63
5.6.1. Общий вид XML конфигурации CORS	63
5.6.2. Get Bucket CORS	63
5.6.3. Set Bucket CORS	64
5.6.4. Delete Bucket CORS	66
5.7. Описание типов данных	66
5.7.1. AbortIncompleteMultipartUpload	66
5.7.2. AccessControlPolicy	66
5.7.3. Bucket	67
5.7.4. CompletedMultipartUpload	67
5.7.5. CompletedPart	67
5.7.6. CompleteMultipartUploadResult	67
5.7.7. Contents	67
5.7.8. CopyObjectResult	68
5.7.9. CORSConfiguration	68
5.7.10. CORSRule	68
5.7.11. CreateBucketConfiguration	68
5.7.12. CreatePrefixKeyResult	69
5.7.13. Delete	69
5.7.14. DeletedObject	69
5.7.15. DeletePrefixKeyResult	69
5.7.16. DeleteResult	69
5.7.17. Error	70
5.7.18. Grant	70
5.7.19. Grantee	70
5.7.20. InitiateMultipartUploadResult	70
5.7.21. LifecycleConfiguration	71
5.7.22. LifecycleExpiration	71
5.7.23. LifecycleRule	71
5.7.24. LifecycleRuleFilter	71
5.7.25. ListAllMyBucketsResult	72
5.7.26. ListBucketResult	72

5.7.27. ListPartsResult	72
5.7.28. ListPrefixKeysResult	73
5.7.29. Object	73
5.7.30. ObjectIdentifier	74
5.7.31. Owner	74
5.7.32. Part	74
5.7.33. Transition	74

Термины и сокращения

В настоящем документе используются сокращения, перечисленные в таблице ниже.

Сокращение	Расшифровка
БД	База данных
ВМ	Виртуальная машина
ОС	Операционная система
ПИКД	Подсистема идентификации и контроля доступа
ПО	Программное обеспечение
СУБД	Система управления базами данных
ACL	Access Control List, список управления доступом, который определяет, кто или что может получать доступ к объекту, и какие именно операции разрешено или запрещено выполнять субъекту
API	Application Programming Interface - Аппаратно-программный интерфейс. Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой
JavaScript	Мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили
JSON	JavaScript Object Notation – текстовый формат обмена данными, основанный на JavaScript
S3	Simple Storage Service. Сервис хранения объектов
URL	Uniform Resource Locator, унифицированный указатель ресурса – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса. Используется как стандарт записи ссылок на объекты в Интернете

1. Введение

1.1. Область применения

Настоящее руководство предназначено для пользователей S3 в частной инсталляции (далее по тексту – S3, Система, Объектное хранилище).

Область применения настоящего документа распространяется на все подсистемы, модули и разделы S3 в частной инсталляции.

1.2. Краткое описание возможностей

S3 в частной инсталляции – это совместимый с протоколом S3 сервис для организации безопасного, надежного и масштабируемого объектного хранилища данных, устанавливаемый на аппаратные мощности, предоставляемые в собственном в центре обработки данных (далее – ЦОД) Заказчика (On premise).

S3 решает проблему разрозненного хранения различного типа объектов, а также решает задачу стабильной скорости раздачи любых типов объектов независимо от числа одновременных обращений. В качестве объектов могут выступать: мультимедиа, контент сайтов, логи транзакций, электронные документы, большие данные и резервные копии и архивы.

1.3. Требования к квалификации

Пользователи выполняют в рамках работы с Объектным хранилищем следующие действия:

- Создание и удаление бакетов и объектов в Объектном хранилище;
- Загрузка и выгрузка объектов из бакетов Объектного хранилища.

К пользователям S3 предъявляются следующие требования:

- Прохождение обучения от компании ООО «ВК ЦИФРОВЫЕ ТЕХНОЛОГИИ» по использованию компонентов, разработанных вендором;
- Навыки работы с ПК, в том числе способность самостоятельно осуществлять:
 - Установку, и запуск S3 совместимого файлового менеджера;
 - Установку и запуск AWS CLI;
 - Выполнение операций с файлами средствами операционной системы (создание, копирование, переименование, удаление, перемещение и т.п.).

1.4. Перечень эксплуатационной документации

Перечень эксплуатационной документации, с которой необходимо ознакомиться:

- S3 в частной инсталляции. Пояснительная записка.
- S3 в частной инсталляции. Руководство пользователя.

2. Инструменты для работы с бакетами и объектами S3

2.1. S3 CLI

S3 CLI или интерфейс командной строки S3 – это единый инструмент для управления сервисами S3, основанный на пакете инструментов AWS S3. Загрузив всего одно средство, возможно контролировать множество сервисов S3 из командной строки и автоматизировать их с помощью скриптов.

В интерфейсе командной строки S3 представлен новый набор простых команд для эффективного получения и отправки файлов в S3.

При минимальной настройке интерфейс командной строки S3 позволяет запускать команды из командной строки в программе терминала:

Оболочки Linux – общие программы оболочки, такие как `bash`, `zsh`, и `tcsh` для выполнения команд в Linux или macOS. Командная строка Windows – в Windows запускаются команды из командной строки Windows или в PowerShell. Amazon S3 CLI доступен в двух версиях, и информация в этом руководстве применима к обеим версиям, если не указано иное.

Версия 2.x – текущая общедоступная версия интерфейса командной строки S3, предназначенная для использования в производственных средах. Версия 1.x – предыдущая версия интерфейса командной строки AWS, доступная для обеспечения обратной совместимости.

2.2. Файловые менеджеры для S3

Графические инструменты, такие как файловые менеджеры, позволяют просмотреть объекты в бакете, имитируя иерархическую структуру каталогов. Помимо этого, графические файловые менеджеры позволяют выполнять все необходимые операции с бакетами и объектами.

Для работы с Объектным хранилищем могут использоваться следующие приложения:

- Диск-О.
- Forklift.
- Cloudberry.
- MSP360 Explorer.
- Cyberduck.
- S3browser.
- S3FS-Fuse.

- WinSCP.

3. Создание, настройка и работа с бакетами S3

Чтобы хранить данные в Объектном хранилище, вы работаете с ресурсами, известными как бакеты и объекты. Бакет представляет собой контейнер для объектов. Объект представляет собой файл и любые метаданные, описывающие этот файл.

Чтобы сохранить объект в Объектном хранилище, вы создаете бакет, а затем загружаете объект в этот бакет. Когда объект находится в бакете вы можете открыть его, скачать или переместить. В случае, если объект или бакет более не нужны, вы можете их удалить.

3.1. Правила именования бакетов

Для именования бакетов S3 применяются следующие правила:

- Имена бакетов должны содержать от 3 до 63 символов.
- Имена бакетов могут состоять только из строчных букв, цифр, точек «.» и дефисов «-».
- Имена бакетов должны начинаться и заканчиваться буквой или цифрой.
- Имена бакетов не должны быть отформатированы как IP-адрес (например, 192.168.5.4).
- Имена бакетов не должны начинаться с префикса «xn--».
- Имена бакетов не должны заканчиваться суффиксом «-s3alias». Этот суффикс зарезервирован для псевдонимов точек доступа.
- Имена бакетов внутри проекта должны быть уникальными.
- Для лучшей совместимости мы рекомендуем избегать использования точек «.» в именах бакетов. Если вы включите точки в имя сегмента, вы не сможете использовать адресацию в стиле виртуального хоста через HTTPS, если не выполните собственную проверку сертификата. Это связано с тем, что сертификаты безопасности, используемые для виртуального хостинга бакетов, не работают для бакетов с точками в их именах.

Это ограничение не влияет на сегменты, используемые для статического хостинга веб-сайтов, поскольку статический хостинг веб-сайтов доступен только через HTTP.

3.1.1. Примеры названий бакетов

Следующие примеры имен бакетов допустимы и соответствуют рекомендуемым правилам именования:

- `docexamplebucket1`.
- `log-delivery-march-2020`.

- `my-hosted-content`.

Следующие примеры имен бакетов недопустимы:

- `doc_example_bucket` (содержит символы подчеркивания).
- `DocExampleBucket` (содержит прописные буквы).
- `doc-example-bucket-` (заканчивается дефисом).

3.2. Создание бакета

Чтобы загрузить данные в S3, необходимо сначала создать бакет.

Аккаунт, который создает бакет, получает полные права доступа к бакету и объектам. В бакет может быть загружено любое количество объектов.

Для создания бакета можно использовать Файловый менеджер или S3 CLI.

3.2.1. Создание бакета при помощи S3 CLI

Следующая команда создаст бакет с именем `new-bkt` в вашем проекте:

```
aws s3 mb s3://new-bkt --endpoint-url http://hb.your-domain.com
```

3.2.2. Создание бакета при помощи файлового менеджера

ПРИМЕЧАНИЕ

Здесь и далее мы используем файловый менеджер «CloudBerry Explorer for Amazon S3». Если вы используете другой файловый менеджер, то интерфейс и названия его элементов могут отличаться от используемых в данной инструкции.

Для создания нового бакета выполните следующие действия:

1. На панели файлового менеджера, в которой открыт ваш проект, нажмите кнопку «New Bucket»:

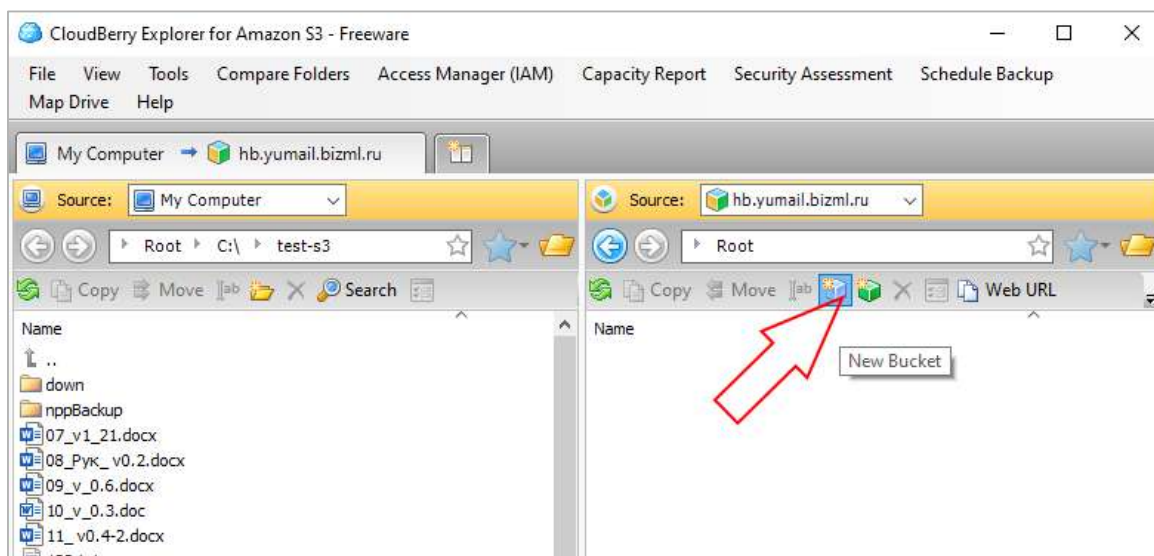


Рисунок 1 – Создание бакета. Кнопка «New Bucket»

2. Затем укажите имя нового бакета (например, `new-bkt-1`) и нажмите кнопку «Ок»:

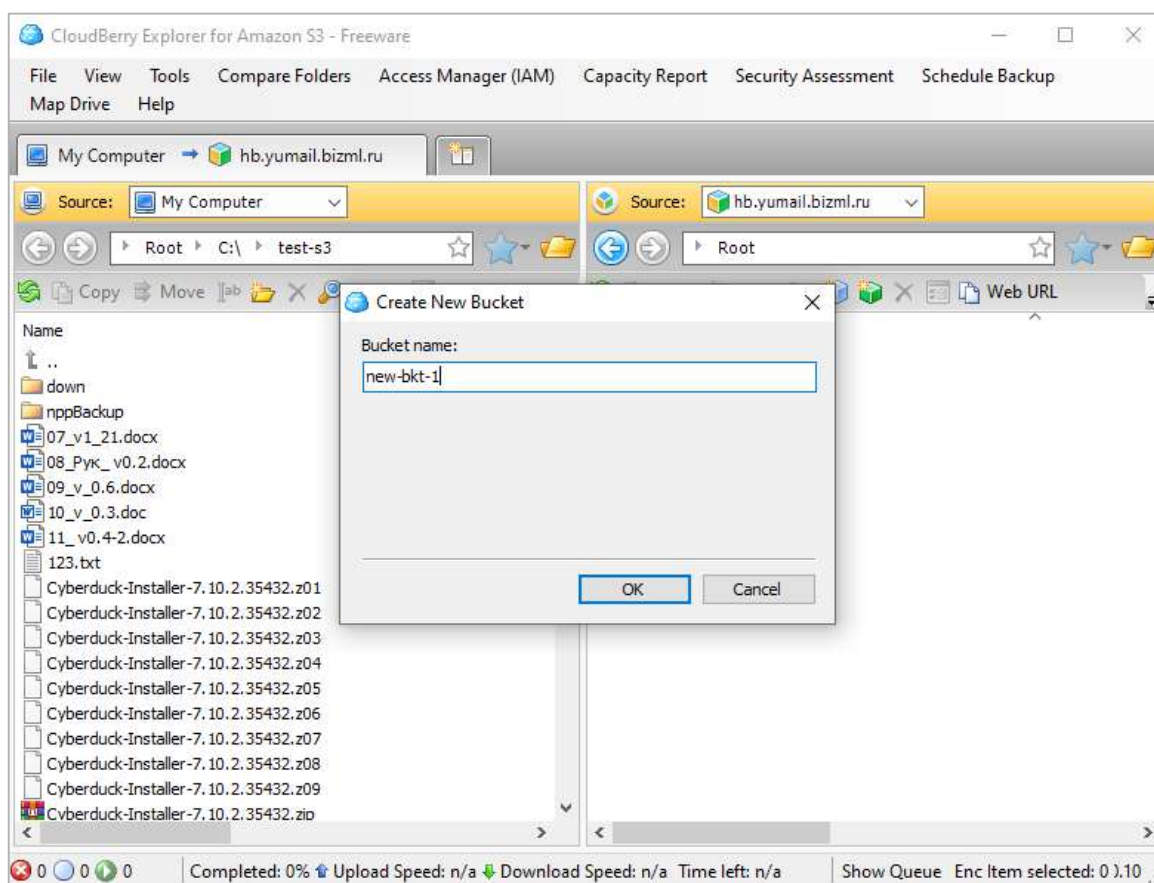


Рисунок 2 – Указание имени нового бакета

3. Готово, бакет создан и отображается на панели вашего проекта:

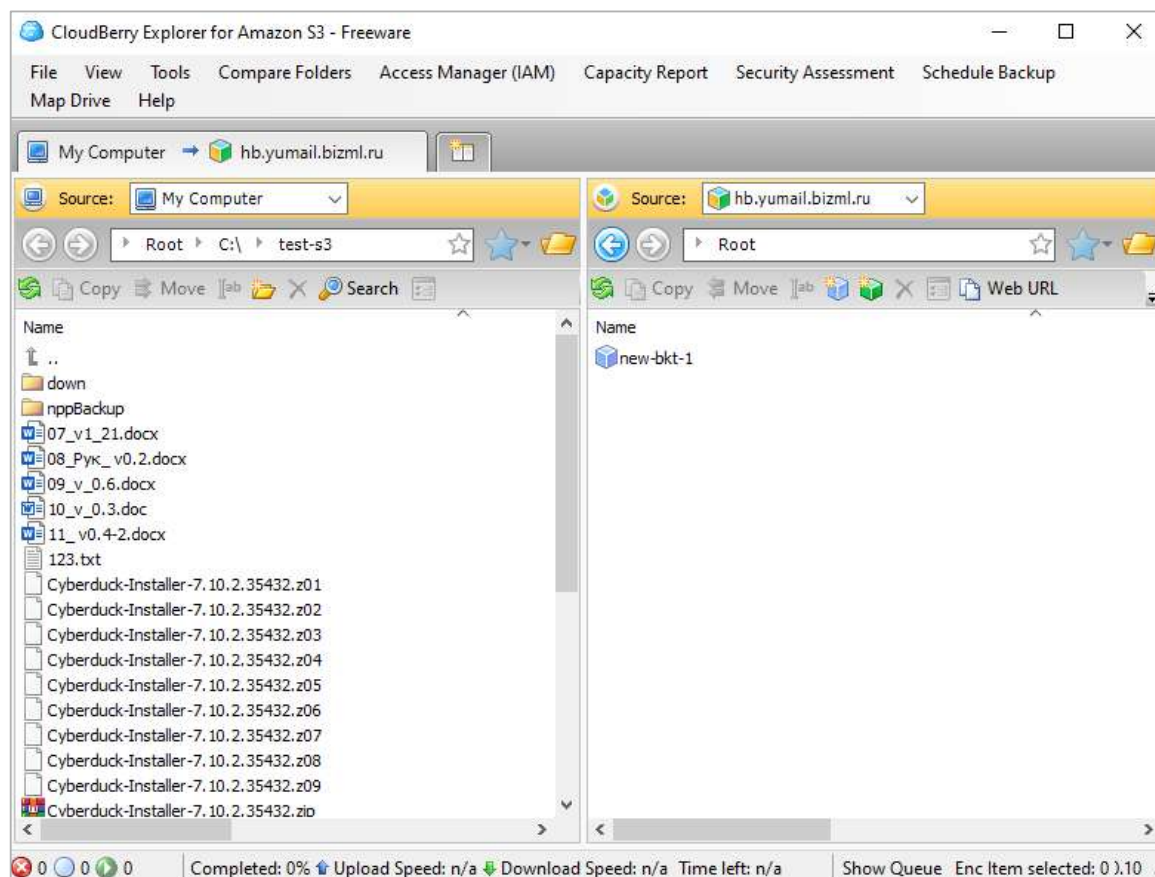


Рисунок 3 – Проект с новым бакетом

3.3. Просмотр свойств бакета S3

Вы можете просматривать и настраивать свойства бакетов S3, включая параметры управления версиями, теги, шифрование по умолчанию, ведение журнала, уведомления и многое другое.

3.3.1. Просмотр свойств бакета при помощи файлового менеджера

ПРИМЕЧАНИЕ

Здесь и далее мы используем файловый менеджер «CloudBerry Explorer for Amazon S3». Если вы используете другой файловый менеджер, то интерфейс и названия его элементов могут отличаться от используемых в данной инструкции.

Для просмотра свойств бакета выполните следующие действия:

1. На панели файлового менеджера, в которой открыт ваш проект выберите нужный бакет и вызовите контекстное меню правой кнопкой мыши. В контекстном меню выберите пункт «Properties»:

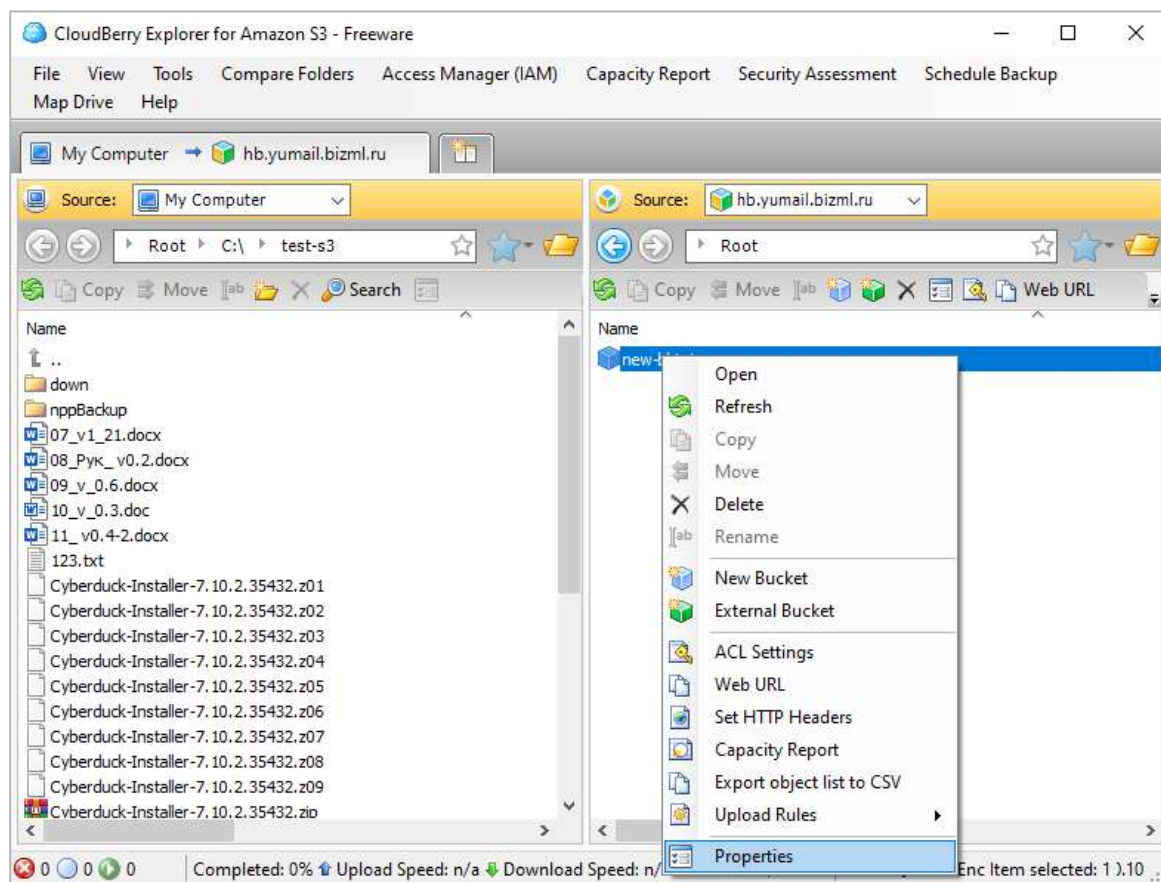


Рисунок 4 – Выбор пункта меню «Properties»

2. В открывшемся окне доступны для просмотра свойства бакета:

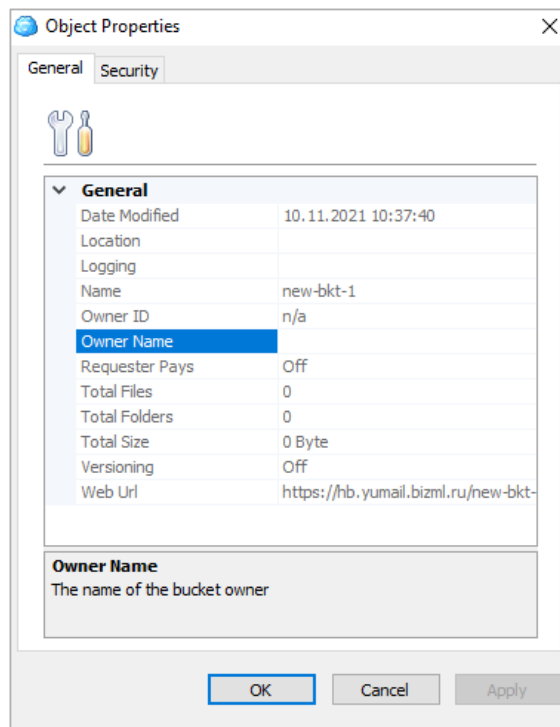


Рисунок 5 – Свойства бакета

3.4. Очистка бакета

При выполнении очистки бакета удаляются все объекты, но сам бакет остается. Операцию очистки бакета нельзя отменить, все объекты будут уничтожены безвозвратно. Когда вы очищаете бакет, для которого включено управление версиями объектов, все версии всех объектов в бакете удаляются.

Выполнить очистку бакета можно при помощи S3 CLI.

3.4.1. Очистка бакета при помощи S3 CLI

Следующая команда полностью очистит бакет с именем `new-bkt` в вашем проекте, удалив все файлы:

```
aws s3 rm s3://new-bkt --recursive --endpoint-url http://hb.your-domain.com
```

3.5. Удаление бакета

Вы можете удалить бакет в своем проекте. Удалять можно как пустые, так и содержащие объекты бакеты. Вы можете удалить все бакеты своего проекта. Операцию удаления бакета нельзя отменить, все объекты будут уничтожены безвозвратно.

Для удаления бакета можно использовать Файловый менеджер или S3 CLI.

3.5.1. Удаление бакета при помощи S3 CLI

Следующая команда удалит бакет с именем `new-bkt` в вашем проекте:

```
aws s3 rb s3://new-bkt --force --endpoint-url http://hb.your-domain.com
```

3.5.2. Удаление бакета при помощи файлового менеджера

ПРИМЕЧАНИЕ

Здесь и далее мы используем файловый менеджер «CloudBerry Explorer for Amazon S3». Если вы используете другой файловый менеджер, то интерфейс и названия его элементов могут отличаться от используемых в данной инструкции.

1. На панели файлового менеджера в которой открыт ваш проект выберите нужный бакет и вызовите контекстное меню правой кнопкой мыши. В контекстном меню выберите пункт «Delete»:

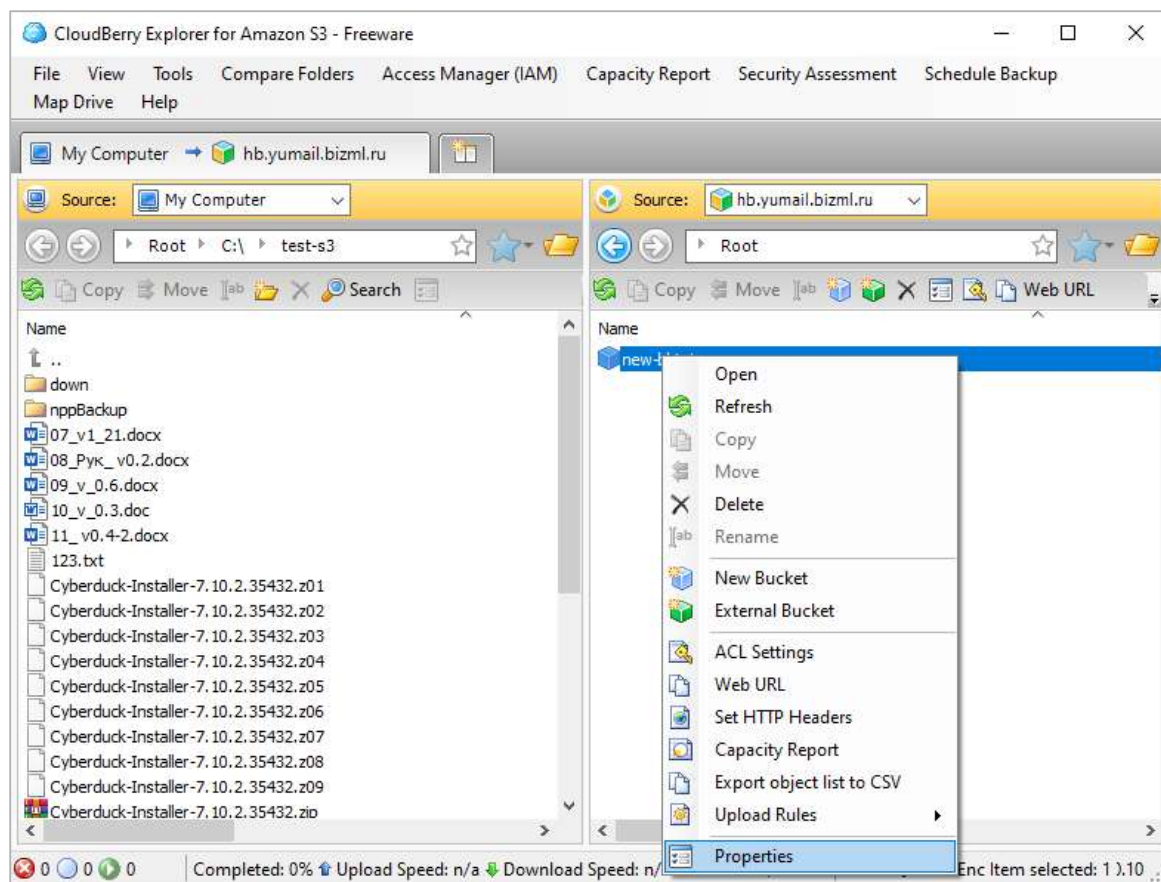


Рисунок 6 – Выбор пункта меню «Delete»

2. Затем подтвердите удаление в диалоговом окне нажав кнопку «Yes»:

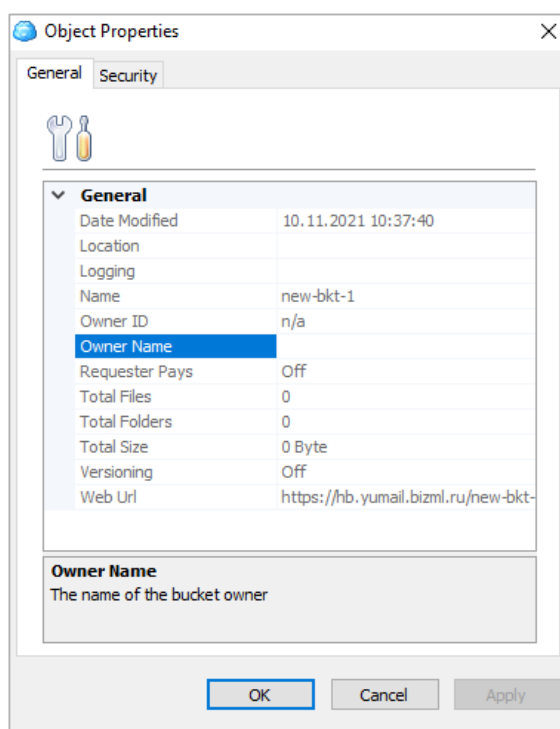


Рисунок 7 – Подтверждение удаления бакета

3.6. Ограничения и лимиты для бакетов

3.6.1. Ограничения

- Невозможно создать бакет, который будет вложен в другой бакет. Но внутри бакетов можно создавать каталоги для организации оптимальной системы хранения файлов;
- Протокол S3 не поддерживает переименование и перемещение объектов. Переименование может быть произведено только копированием содержимого объекта в новый объект с нужным именем. В файловых менеджерах с графическим интерфейсом все действия для переименования файла делаются автоматически.

3.6.2. Лимиты

Лимиты – технические ограничения, распространяющиеся на всю платформу S3. Лимиты представлены в таблице 1.

Таблица 1 — Лимиты Объектного хранилища

Параметр	Количество / объем
Бакеты	25

Параметр	Количество / объем
Пользователи	25
Максимальное количество файлов в бакете	Не ограничено
Максимальный объем всех файлов в бакете	Не ограничен
Размер файла	32 Гб (для объектов от 32 Гб до 320 ТБ необходимо использовать составную загрузку)
Минимальный размер объекта	0 байт
Размер бакета	Не ограничен
Количество запросов в сутки	Листинг: 10,000,000 Остальные: 10,000,000
Количество запросов в секунду	Листинг: 15 Остальные: 500

4. Выгрузка, загрузка и работа с объектами в S3

4.1. Создание имен (ключей) объектов

Имя объекта (или ключ объекта) однозначно идентифицирует объект в бакете S3.

При создании объекта вы указываете имя объекта, которое однозначно идентифицирует объект в бакете. Например, в панели файлового менеджера, когда вы открываете бакет, появляется список объектов в вашем бакете. Эти имена являются ключами объекта. Имя объекта или ключ – это последовательность символов Юникода, длина которой в кодировке UTF-8 составляет не более 1024 байтов.

Модель данных S3 представляет собой плоскую структуру: вы создаете бакет, в котором хранятся объекты. Нет иерархии вложенных бакетов. Однако вы можете создать логическую иерархию используя концепцию папок. Для этого к именам объектов нужно добавить префиксы и разделители.

Предположим, что в вашем бакете `new-bkt` есть четыре объекта со следующими ключами объектов:

- `Development/Projects.xls`.
- `Finance/statement1.pdf`.
- `Private/taxdocument.pdf`.
- `s3-dg.pdf`.

S3 использует префиксы имени (`Development/`, `Finance/` и `Private/`) и разделитель `/` для представления структуры папок. Имя `s3-dg.pdf` не имеет префикса, поэтому этот объект размещается на корневом уровне бакета. Если вы откроете папку `Development/`, вы увидите в ней объект `Projects.xlsx`.

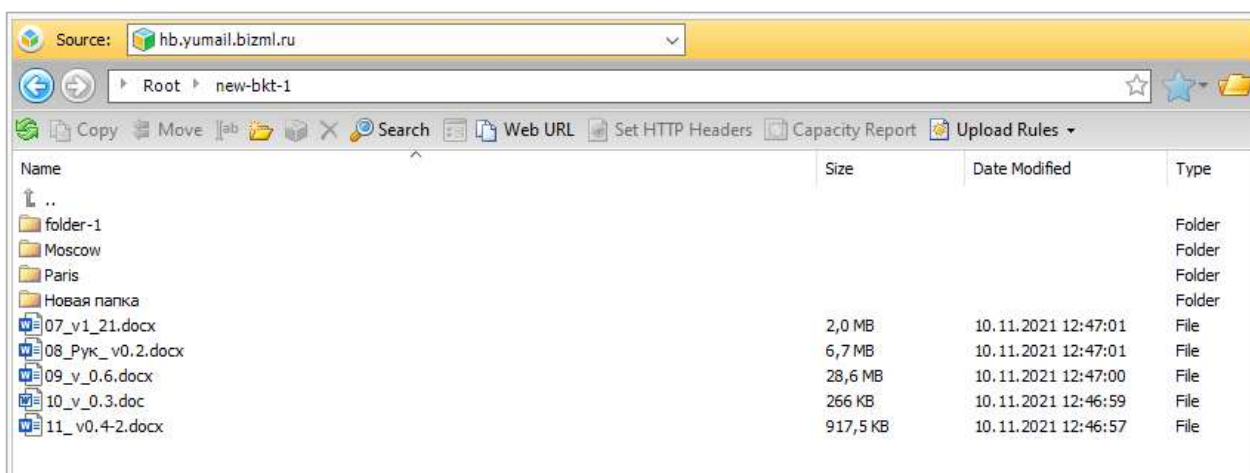


Рисунок 8 – Иерархия папок в бакете

4.1.1. Рекомендации по созданию имен (ключей) объекта

В имени (ключе) объекта можно использовать любой символ UTF-8. Однако использование определенных символов в именах объектов может вызвать проблемы с некоторыми приложениями и протоколами. Следующие ниже рекомендации помогут вам в максимальной степени соответствовать требованиям DNS, веб-безопасным символам, синтаксическим анализаторам XML и другим API.

Допустимые наборы символов для использования в именах объектов представлены в таблице 2:

Таблица 2 — Допустимые наборы символов для использования в именах объектов

Буквенно-цифровые символы	• 0-9 • a-z • A-Z
Специальные символы	• Прямой слеш «/». • Восклицательный знак «!». • Дефис «-». • Подчеркивание «_». • Точка «.». • Звездочка «*». • Одиночная кавычка «'». • Открывающая скобка «(». • Закрывающая скобка «)»

Ниже приведены примеры допустимых имен (ключей) объекта:

- `4my-organization.`
- `my.great_photos-2020/jan/myvacation.jpg.`
- `videos/2020/birthday/video1.wmv.`

Следующие символы нельзя использовать в имени (ключе) объекта

- Обратный слеш «\».
- Левая фигурная скобка «{».
- Правая фигурная скобка «}».
- Непечатаемые символы ASCII (128–255 десятичных символов).
- Крышка «^».
- Знак процента «%».
- Обратный апостроф «'».
- Правая квадратная скобка «]».

- Левая квадратная скобка «[».
- Кавычки «"».
- Символ «Больше» «>».
- Символ «Меньше» «<».
- Тильда «~».
- Решетка «#».
- Вертикальная черта «|».

Следующие символы не рекомендуется использовать в имени (ключе) объекта, т.к. они могут потребовать дополнительной обработки кода и, вероятно, должны быть закодированы в URL-адресе или обозначены как HEX. Некоторые из них являются непечатаемыми символами, которые ваш браузер может не обрабатывать, что также требует специальной обработки:

- Амперсанд «&».
- Доллар «\$».
- Диапазон символов ASCII 00–1F шестнадцатеричный (0–31 десятичный) и 7F (127 десятичный).
- «Собака» «@».
- Равно «=».
- Точка с запятой «;».
- Двоеточие «:».
- Плюс «+».
- Пробел – в некоторых случаях могут быть потеряны значимые последовательности пробелов (особенно при использовании нескольких пробелов).
- Запятая «,».
- Вопросительный знак «?».

4.2. Загрузка объектов

Когда вы загружаете файл в S3, он сохраняется как объект S3. Объекты состоят из данных файла и метаданных, описывающих объект. В бакете может быть неограниченное количество объектов. Для того чтобы загрузить объекты в бакет, у вашей учетной записи должны быть разрешения на запись для этого бакета.

В бакет S3 Вы можете загружать файлы любого типа – изображения, резервные копии, данные, фильмы и т.п.

В зависимости от размера загружаемых данных в S3 возможны следующие варианты загрузки:

1. Объект загружается при помощи SDK, REST API или AWS CLI. За одну операцию PUT вы можете загрузить один объект размером до 5 ГБ.
2. Объект загружается при помощи файлового менеджера S3. С помощью файлового менеджера S3 вы можете загрузить один объект размером до 160 ГБ.
3. Загрузка объекта по частям с помощью SDK, REST API или AWS CLI. Используя API многокомпонентной загрузки, вы можете загрузить один большой объект размером до 320 ТБ.

API-интерфейс многокомпонентной загрузки разработан для улучшения загрузки больших объектов. Вы можете загружать объект по частям. Эти части объекта можно загружать независимо, в любом порядке и параллельно. Вы можете использовать составную загрузку для объектов размером от 5 МБ до 320 ТБ.

4.2.1. Загрузка объекта в бакет при помощи AWS CLI

Для загрузки объекта в бакет выполните в AWS CLI следующую команду:

```
aws s3 cp "C:\folder_name\file_name.file_extension" s3://new-bkt --endpoint-url  
https://hb.bizmrg.com
```

Здесь имя загружаемого файла указано в нотации MS Windows.

4.2.2. Загрузка объекта в бакет при помощи файлового менеджера S3

ПРИМЕЧАНИЕ

Здесь и далее мы используем файловый менеджер «CloudBerry Explorer for Amazon S3». Если вы используете другой файловый менеджер, то интерфейс и названия его элементов могут отличаться от используемых в данной инструкции.

На левой панели вашего файлового менеджера откройте папки вашей операционной системы. На правой панели файлового менеджера откройте S3 и перейдите в нужный бакет:

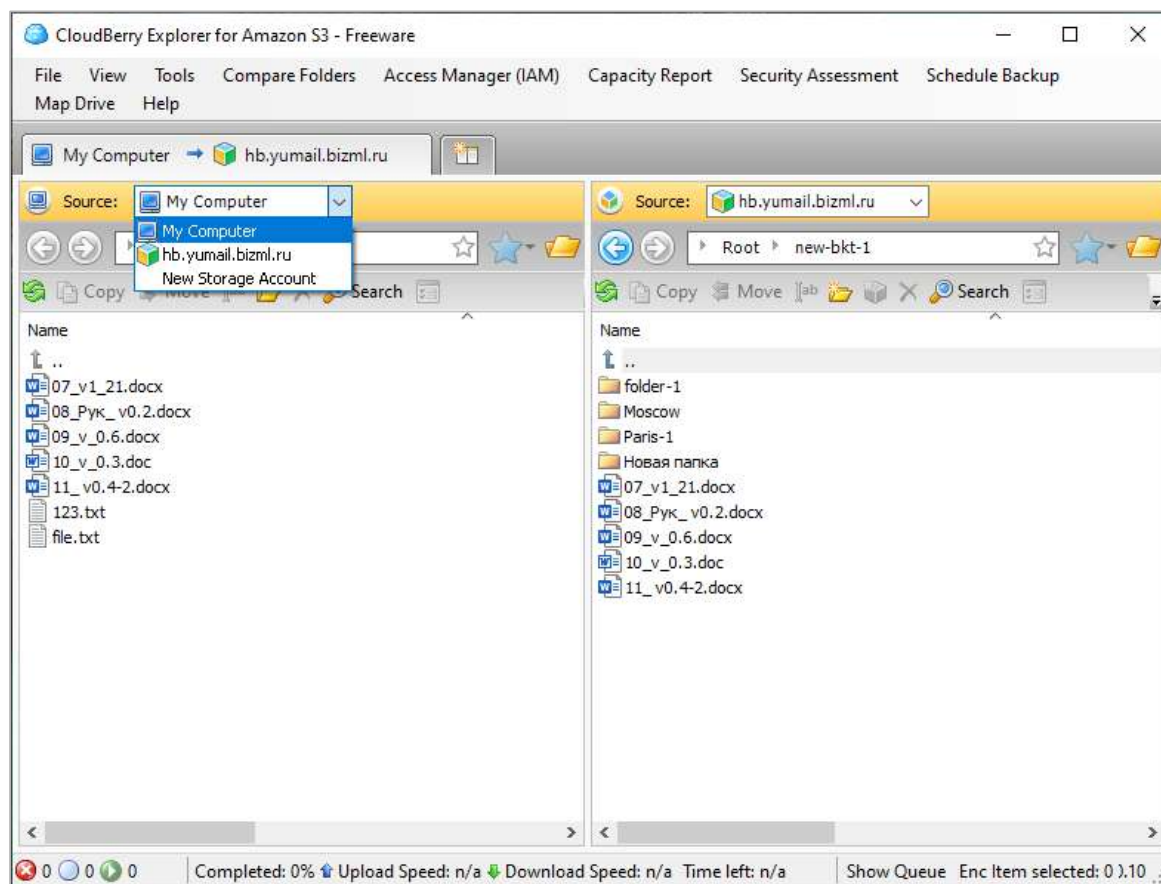


Рисунок 9 – Панели файлового менеджера

На левой панели файлового менеджера найдите файл, который вы планируете загрузить в бакет и с помощью мыши перетащите его на правую панель. Таким образом, в бакете будет создана копия файла из вашей ОС:

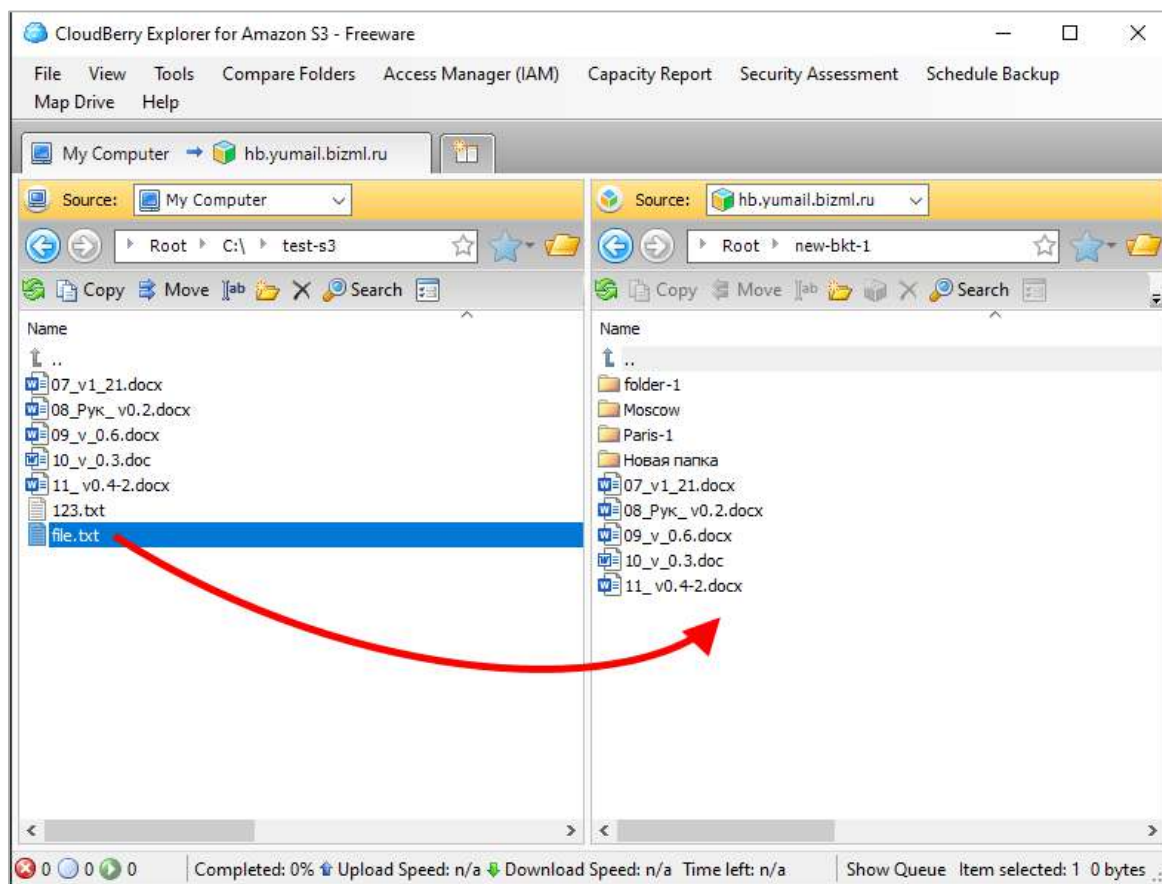


Рисунок 10 – Перетаскивание файла

Или используйте кнопки в левой панели файлового менеджера «Сору» или «Move». Для этого выделите нужный файл или группу файлов и нажмите соответствующую кнопку. Функция «Сору» создает копию файла в бакете, функция «Move» перемещает файл в бакет:

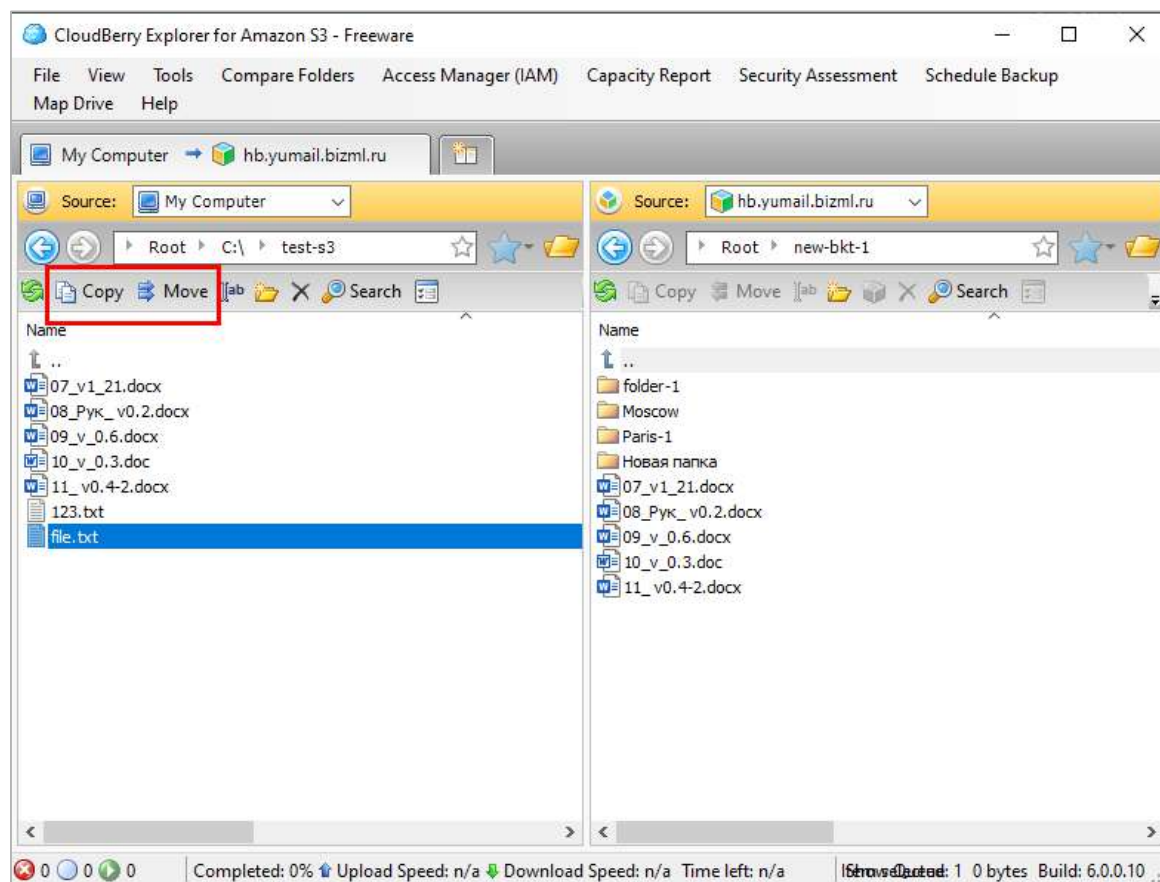


Рисунок 11 – Кнопки «Copy» и «Move»

4.3. Загрузка объектов с помощью многокомпонентной загрузки

Многокомпонентная загрузка позволяет загружать один объект как набор частей. Каждая часть представляет собой непрерывную часть данных объекта. Вы можете загружать эти части объекта независимо и в любом порядке. Если передача какой-либо части не удалась, вы можете повторно передать эту часть, не затрагивая другие части. После того как все части вашего объекта загружены, S3 собирает эти части и создает объект, когда размер вашего объекта достигает 100 МБ,

Использование составной загрузки дает следующие преимущества:

- Повышенная пропускная способность – вы можете загружать части файлов параллельно, чтобы повысить пропускную способность.
- Быстрое восстановление при любых сетевых проблемах – меньший размер частей файлов сводит к минимуму влияние перезапуска неудачной загрузки из-за сетевой ошибки.
- Приостановить и возобновить загрузку объекта – вы можете загружать части объекта с течением времени. После того как вы иницилируете многокомпонентную загрузку, она может длиться бесконечно, т.е. вы должны явно завершить или остановить составную загрузку.

- Начать загрузку до того, как будет известен окончательный размер объекта – вы можете загружать объект по мере его создания.

Мы рекомендуем использовать многокомпонентную загрузку следующими способами:

- Если вы загружаете большие объекты по стабильной сети с высокой пропускной способностью, используйте многостраничную загрузку, чтобы максимально использовать доступную пропускную способность путем параллельной загрузки частей объекта для многопоточной производительности.
- Если вы загружаете через неоднородную сеть, используйте многостраничную загрузку, чтобы повысить устойчивость к сетевым ошибкам, избегая перезапусков загрузки. При использовании многокомпонентной загрузки вам необходимо повторить загрузку только тех частей, которые были прерваны во время загрузки. Вам не нужно перезапускать загрузку объекта с самого начала.

4.3.1. Процесс многокомпонентной загрузки

Многокомпонентная загрузка – это трёхэтапный процесс: вы инициируете загрузку, затем загружаете части объекта и после того, как все части будут загружены, вы завершаете составную загрузку. После получения полного запроса на загрузку нескольких частей S3 конструирует объект из загруженных частей, и затем можно получить доступ к объекту так же, как и к любому другому объекту в вашей корзине.

Вы можете перечислить все выполняемые вами составные загрузки или получить список частей, которые вы загрузили для конкретной составной загрузки.

1. Инициирование многокомпонентной загрузки.

Когда вы отправляете запрос на инициирование составной загрузки, S3 возвращает ответ с идентификатором загрузки, который является уникальным идентификатором для вашей составной загрузки. Этот идентификатор загрузки должен использоваться каждый раз, когда вы загружаете части файлов, перечисляете детали, завершаете или останавливаете загрузку. Если вы хотите предоставить какие-либо метаданные, описывающие загружаемый объект, вы должны указать их в запросе, чтобы начать многокомпонентную загрузку.

2. Загрузка частей файла.

При загрузке частей файла, помимо идентификатора загрузки, необходимо указать номер части. Вы можете выбрать любой номер части от 1 до 10 000. Номер части однозначно определяет часть и ее положение в загружаемом объекте. Номер части, который вы выбираете, необязательно должен быть последовательным (например, это может быть 1, 5 и 14). Если вы загружаете новую часть файла, используя тот же номер части, что и ранее загруженная часть, часть перезаписывается.

Каждый раз, когда вы загружаете часть, S3 возвращает заголовок ETag в своем ответе. Для каждой загрузки части файла вы должны записать номер части и значение ETag. Вы должны включить эти значения в последующий запрос, чтобы завершить составную загрузку.

3. Завершение многокомпонентной загрузки.

Когда вы завершаете многокомпонентную загрузку, S3 создает объект, объединяя части в порядке возрастания на основе номера части файла. Если какие-либо метаданные объекта были предоставлены в запросе на инициирование многокомпонентной загрузки, S3 связывает эти метаданные с объектом. После успешного выполнения запроса части файла больше не существуют.

Ваш полный запрос на многокомпонентную загрузку должен включать идентификатор загрузки и список номеров частей файла и соответствующих значений ETag. Ответ S3 включает ETag, который однозначно идентифицирует объединенные данные объекта.

При желании вы можете остановить многокомпонентную загрузку. После остановки многокомпонентной загрузки вы не сможете снова загрузить какую-либо часть с этим идентификатором загрузки. При отмене многокомпонентной загрузки все ранее загруженные части файлов удаляются. Если загрузка какой-либо части была в процессе, она все равно может быть успешной или неудачной даже после того, как вы остановите многокомпонентную загрузку. Чтобы освободить все хранилище, используемое всеми частями файлов, вы должны остановить многокомпонентную загрузку только после завершения загрузки всех частей.

4.3.2. Загрузка объекта с помощью многокомпонентной загрузки с помощью AWS CLI

1. Старт составной загрузки:

```
aws s3api create-multipart-upload --bucket new-bkt --key 'multipart/01' --endpoint-url https://hb.bizmrg.com
```

2. Независимая загрузка частей объекта в произвольном порядке:

```
aws s3api upload-part --bucket new-bkt --key 'multipart/01' --part-number 1 --body part01 --upload-id "3xN3EmBZahFtG7X5HBuhWxrmoKXD3NDahniuAxbMVBQi7dHbLVZcaKUGChNyoMPfQrUPQbnA8dQoGrdz2douu fWnaUKfDbMNFcxZcYDXHwdZNN" --endpoint-url https://hb.bizmrg.com
```

3. Окончание составной загрузки:

```
aws s3api complete-multipart-upload --multipart-upload file://mpustruct --bucket new-bkt --key 'multipart/01' --upload-id 3xN3EmBZahFtG7X5HBuhWxrmoKXD3NDahniuAxbMVBQi7dHbLVZcaKUGChNyoMPfQrUPQbnA8dQoGrdz2douu fWnaUKfDbMNFcxZcYDXHwdZNN --endpoint-url https://hb.bizmrg.com
```

4. Отмена составной загрузки:

```
aws s3api abort-multipart-upload --bucket new-bkt --key multipart/01 --upload-id
3xN3EmBZahFtG7X5HBuhWxrmoKXD3NDahniuAxbMVBQi7dHbLVZcaKUGChNyOMPfQrUPQbnA8dQoGrdz2douu
fWnaUKfDbMNFcxZcYDXHwdZNN --endpoint-url https://hb.bizmrg.com
```

4.4. Копирование объектов

Операция копирования создает копию объекта, который уже хранится в S3.

Вы можете создать копию своего объекта размером до 32 ГБ за одну атомарную операцию. Для копирования объекта, размер которого превышает 32 ГБ, необходимо использовать API многокомпонентной загрузки.

Используя операцию Копирования, вы можете:

1. Создавать дополнительные копии объектов.
2. Переименовывать объекты, копируя их и удаляя исходные.
3. Изменить метаданные объекта.

У каждого объекта S3 есть метаданные. Это набор пар имя-значение. Вы можете установить метаданные объекта во время его загрузки. После загрузки объекта вы не можете изменять метаданные объекта. Единственный способ изменить метаданные объекта - сделать копию объекта и установить метаданные. В операции копирования вы устанавливаете тот же объект, что и источник, и цель.

Пользователи могут управлять некоторыми метаданными системы, такими как конфигурация класса хранилища, используемого для объекта, и настраивают шифрование на стороне сервера. При копировании объекта также копируются управляемые пользователем системные метаданные и определенные пользователем метаданные. S3 сбрасывает метаданные, управляемые системой. Например, при копировании объекта S3 сбрасывает дату создания скопированного объекта. Вам не нужно устанавливать какие-либо из этих значений в вашем запросе на копирование.

4.5. Скачивание объекта

4.5.1. Скачивание объекта при помощи AWS CLI

Чтобы скачать объект из бакета выполните следующую команду:

```
aws s3 cp "s3://new-bkt/file.name" C:\folder_name --endpoint-url http://hb.bizmrg.com
```

4.5.2. Скачивание объекта при помощи файлового менеджера

ПРИМЕЧАНИЕ

Здесь и далее мы используем файловый менеджер «CloudBerry Explorer for Amazon S3». Если вы используете другой файловый менеджер, то интерфейс и названия его элементов могут отличаться от используемых в данной инструкции.

На левой панели вашего файлового менеджера откройте папку в вашей операционной системе, в которую вы планируете загрузить файл. На правой панели файлового менеджера откройте S3 и перейдите в нужный бакет:

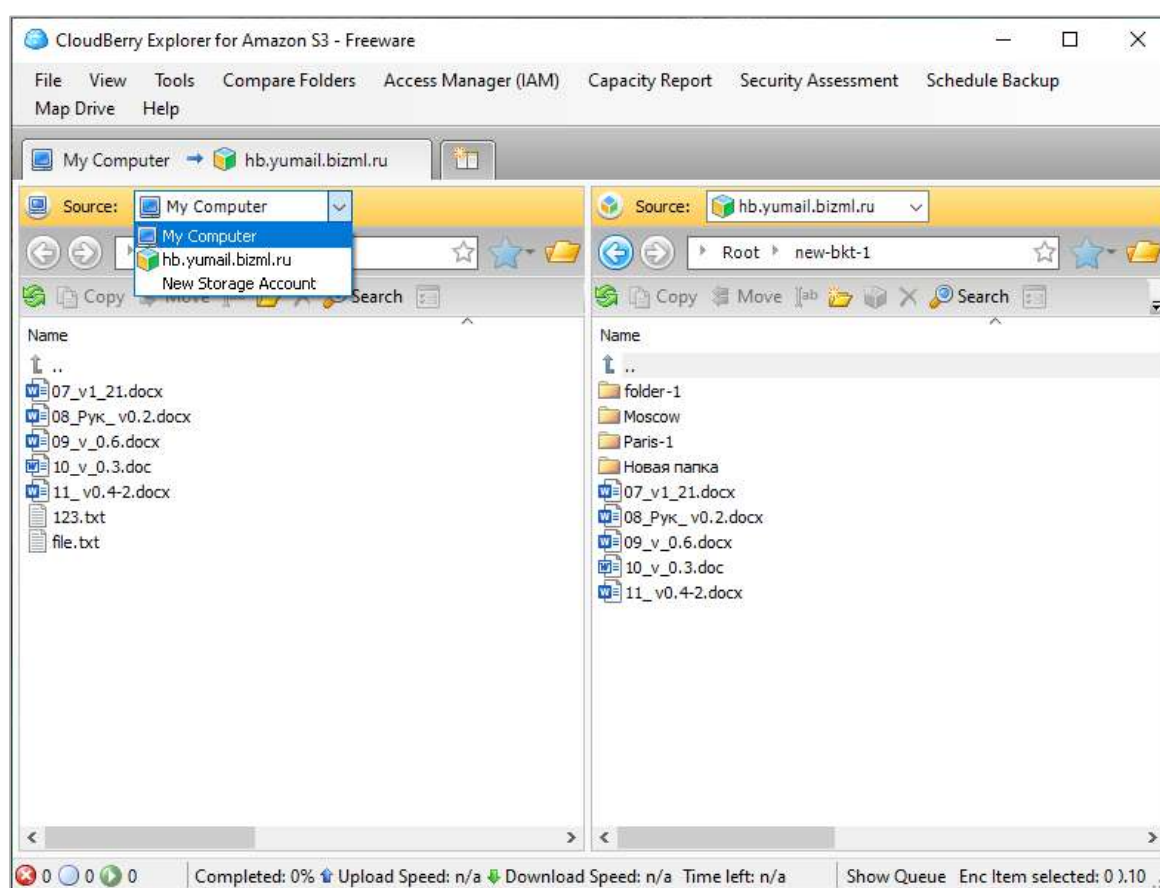


Рисунок 12 – Панели файлового менеджера

На правой панели файлового менеджера найдите файл, который вы планируете скачать из бакета и с помощью мыши перетащите его на левую панель. Таким образом вы загрузите копию файла из бакета:

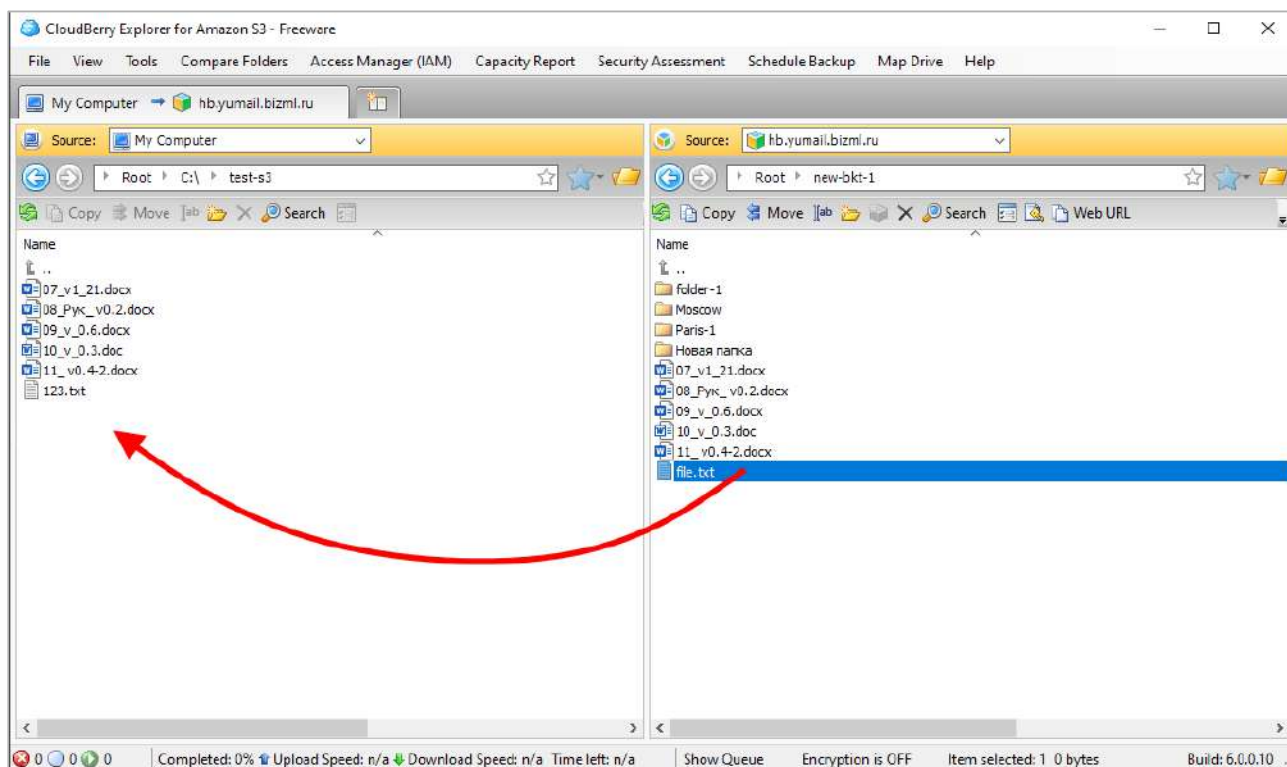


Рисунок 13 – Перетаскивание файла

Или используйте кнопки в правой панели файлового менеджера «Сору» или «Move». Для этого выделите нужный файл или группу файлов и нажмите соответствующую кнопку. Функция «Сору» загружает копию файла из бакета, функция «Move» перемещает файл из бакета. Исходный файл в бакете удаляется:

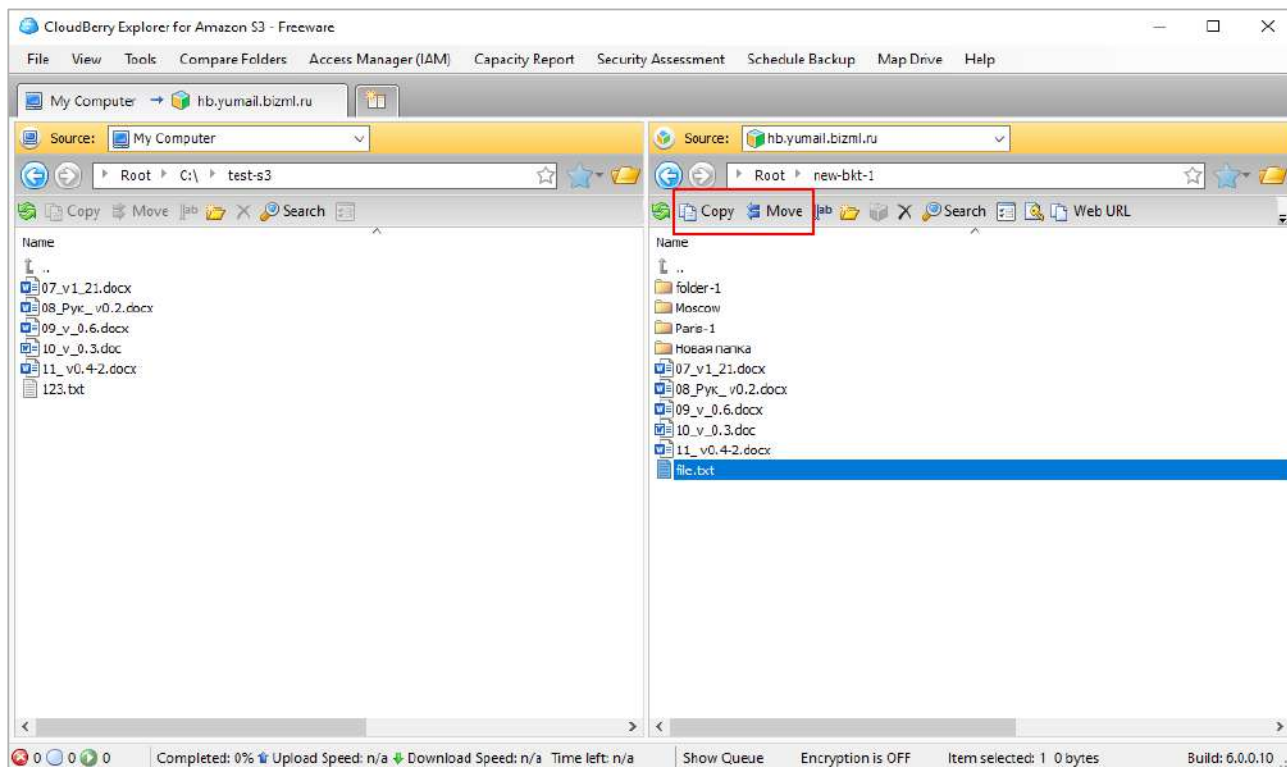


Рисунок 14 – Кнопки «Сору» и «Move»

4.6. Удаление объектов

4.6.1. Удаление объектов при помощи AWS CLI

Чтобы удалить объект из бакета выполните следующую команду:

```
aws s3 rm "s3://new-bkt/file.name" --endpoint-url https://hb.bizmrg.com
```

4.6.2. Удаление объектов при помощи файлового менеджера

ПРИМЕЧАНИЕ

Здесь и далее мы используем файловый менеджер «CloudBerry Explorer for Amazon S3». Если вы используете другой файловый менеджер, то интерфейс и названия его элементов могут отличаться от используемых в данной инструкции.

1. На правой панели файлового менеджера откройте S3 и перейдите в нужный бакет. Найдите файл, который вы планируете удалить и вызовите для него контекстное меню правой кнопкой мыши:

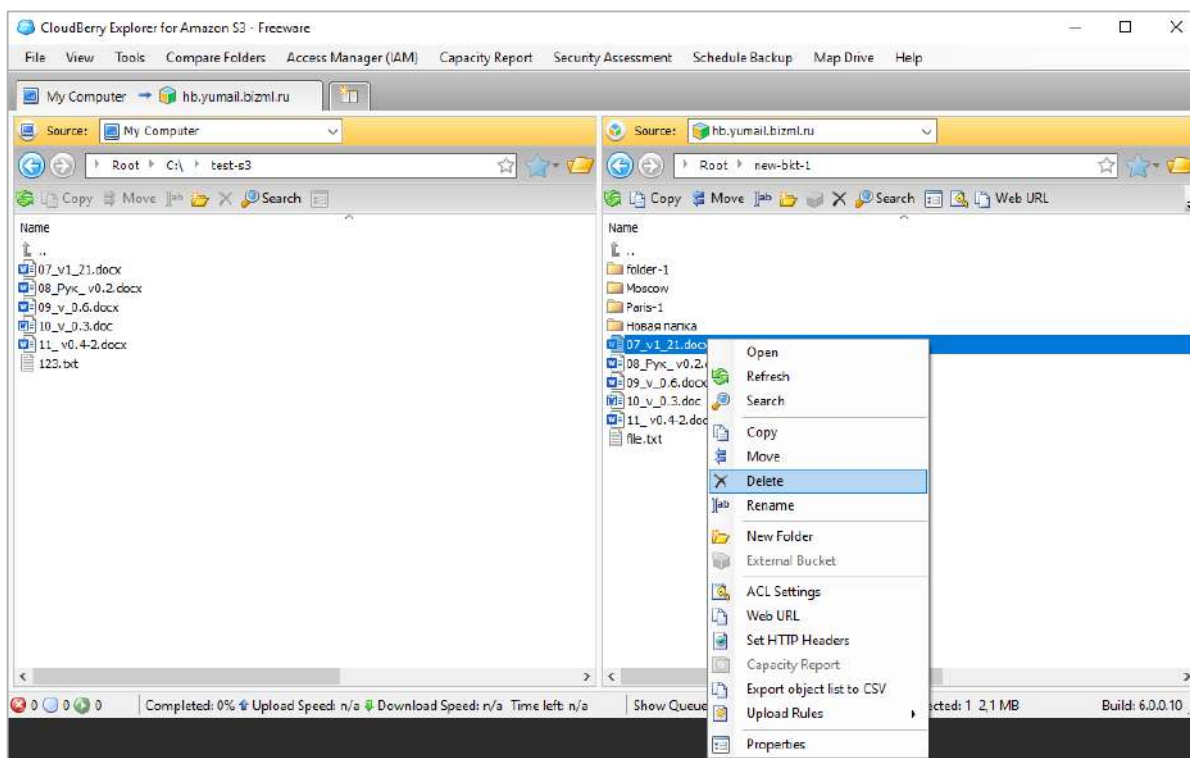


Рисунок 15 – Удаление файла с помощью контекстного меню

Или, выберите нужный файл и нажмите кнопку «Delete» на панели инструментов файлового менеджера:

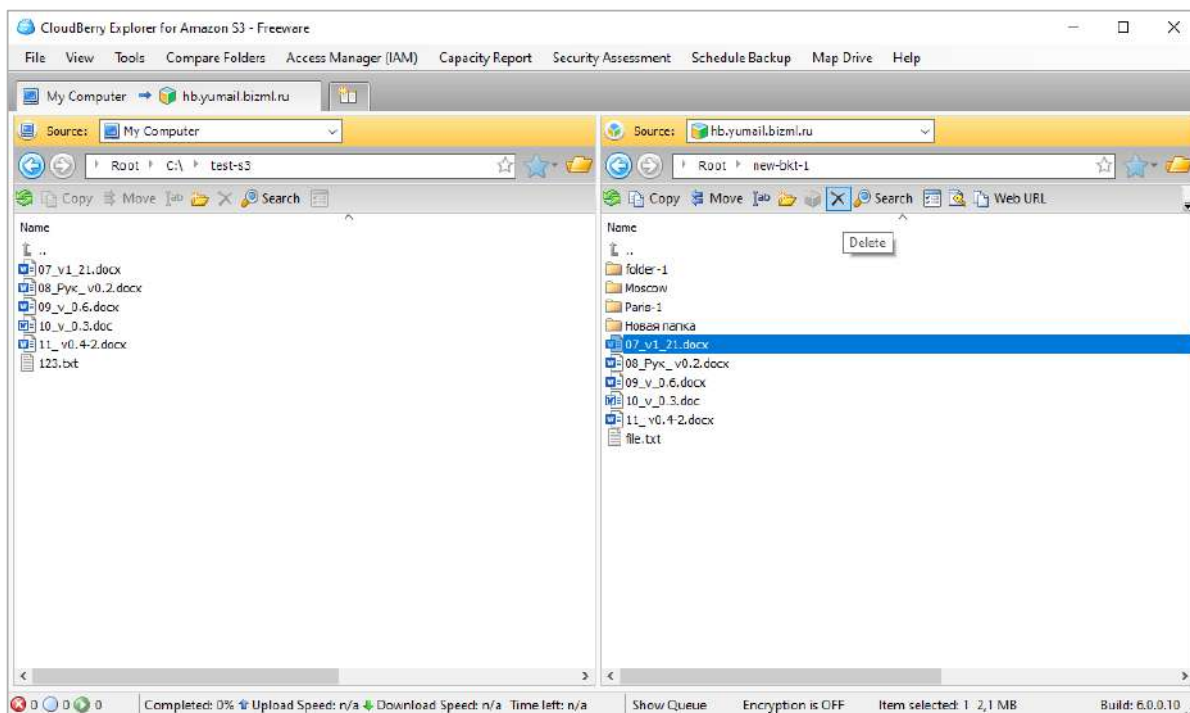


Рисунок 16 – Удаление файла с помощью панели инструментов файлового менеджера

2. В открывшемся диалоговом окне можно подтвердить или отменить удаление файла:

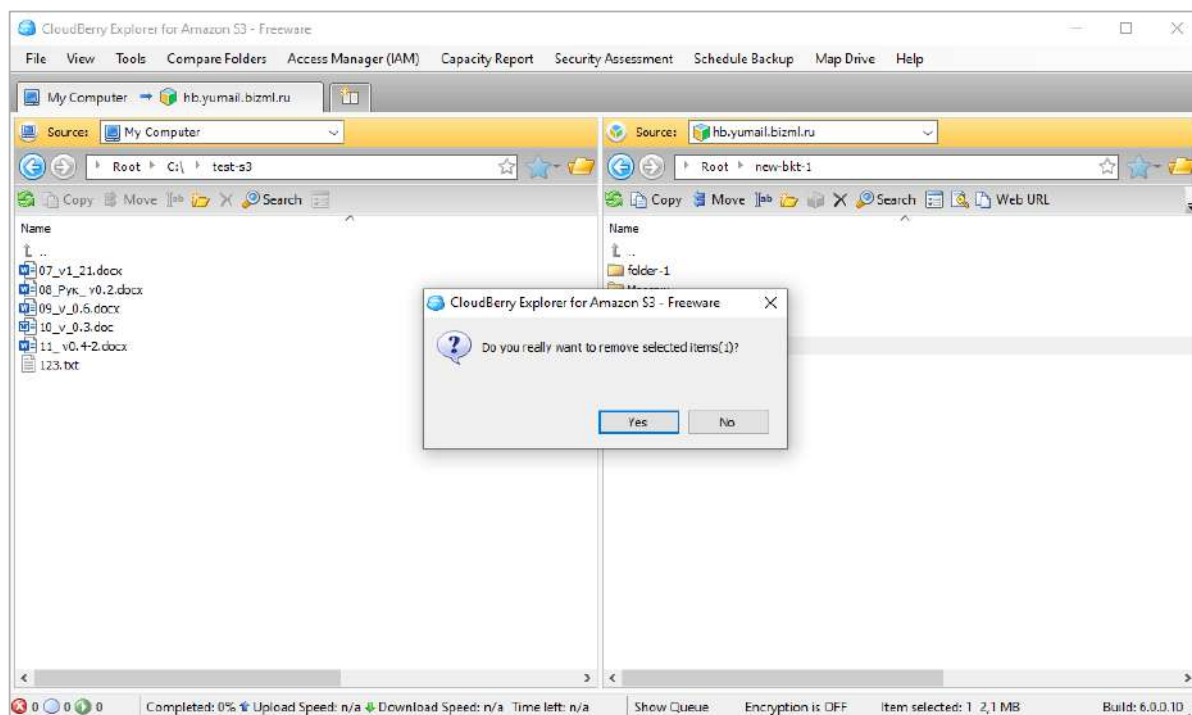


Рисунок 17 – Подтверждение удаления файла

4.7. Организация, перечисление и работа с объектами

В S3 вы можете использовать префиксы для организации хранилища. Префикс – это логическая группа объектов в бакете. Значение префикса похоже на имя каталога, которое позволяет хранить аналогичные данные в одном каталоге в бакете. При программной загрузке объектов вы можете использовать префиксы для организации данных.

На панели файлового менеджера для S3 префиксы называются папками. Вы можете просмотреть все свои объекты и папки на панели файлового менеджера для S3, перейдя в бакет. Вы можете просмотреть информацию о каждом объекте, включая свойства объекта.

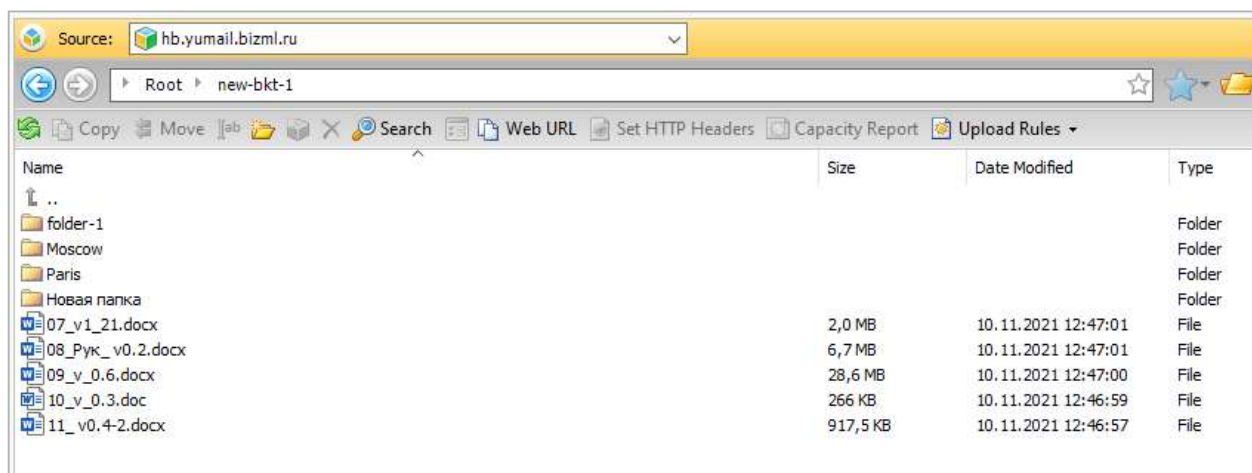


Рисунок 18 – Папки на панели файлового менеджера

В S3 бакеты и объекты являются основными ресурсами. Объекты хранятся в бакетах. S3 имеет плоскую структуру, а не иерархию, как в файловой системе. Однако для упрощения организации S3 поддерживает концепцию папки как средства группировки объектов. Это достигается за счет использования общего префикса имени для объектов, то есть объекты имеют имена, начинающиеся с общей строки. Имена объектов также называются ключами.

Например, на панели файлового менеджера, вы можете создать папку с именем `photos` и сохранить в ней объект названный `myphoto.jpg`. На уровне хранилища этот объект сохраняется с ключом `photos/myphoto.jpg`, где `photos/` - префикс.

У вас могут быть папки внутри папок. Вы можете загружать и копировать объекты прямо в папку. Папки можно создавать, удалять, публиковать и переименовывать. Объекты можно копировать из одной папки в другую.

4.8. Использование предварительно подписанных URL-адресов

По умолчанию все объекты и корзины являются приватными. Вы можете использовать предварительно подписанный URL-адрес, чтобы при желании обмениваться объектами или разрешить вашим клиентам/пользователям скачивать или загружать объекты в бакеты без учетных данных или разрешений безопасности S3.

4.8.1. Ограничение возможностей предварительно подписанного URL

Вы можете использовать предварительно подписанный URL-адрес для создания URL-адреса, который можно использовать для доступа к вашим бакетам S3. Когда вы создаете предварительно подписанный URL-адрес, вы связываете его с определенным действием. Вы можете поделиться URL-адресом, и любой, у кого есть доступ к нему, может выполнить действие, встроенное в URL-адрес, как если бы он был исходным пользователем. URL-адрес перестанет работать, когда истечет срок его

действия. Возможности URL-адреса ограничены разрешениями пользователя, создавшего предварительно подписанный URL-адрес.

По сути, заранее подписанные URL-адреса – это токен-носитель, который предоставляет доступ любому пользователю, независимо от наличия учетных данных для авторизации в S3.

4.8.2. Создание предварительно подписанного URL-адреса при помощи AWS CLI

Для создания предварительно подписанного URL-адрес выполните команду при помощи AWS CLI:

```
aws s3 presign "s3://new-bkt/file-name" --endpoint-url https://hb.bizmrg.com
```

Для того чтобы создать предварительно подписанный URL-адрес с ограниченным сроком действия выполните команду при помощи AWS CLI:

```
aws s3 presign "s3://new-bkt/file-name" --endpoint-url https://hb.bizmrg.com --expires-in 60
```

4.8.3. Создание предварительно подписанного URL-адреса при помощи файлового менеджера

ПРИМЕЧАНИЕ

Здесь и далее мы используем файловый менеджер «CloudBerry Explorer for Amazon S3». Если вы используете другой файловый менеджер, то интерфейс и названия его элементов могут отличаться от используемых в данной инструкции.

1. Выберите в бакете нужный файл и вызовите для него контекстное меню правой кнопкой мыши:

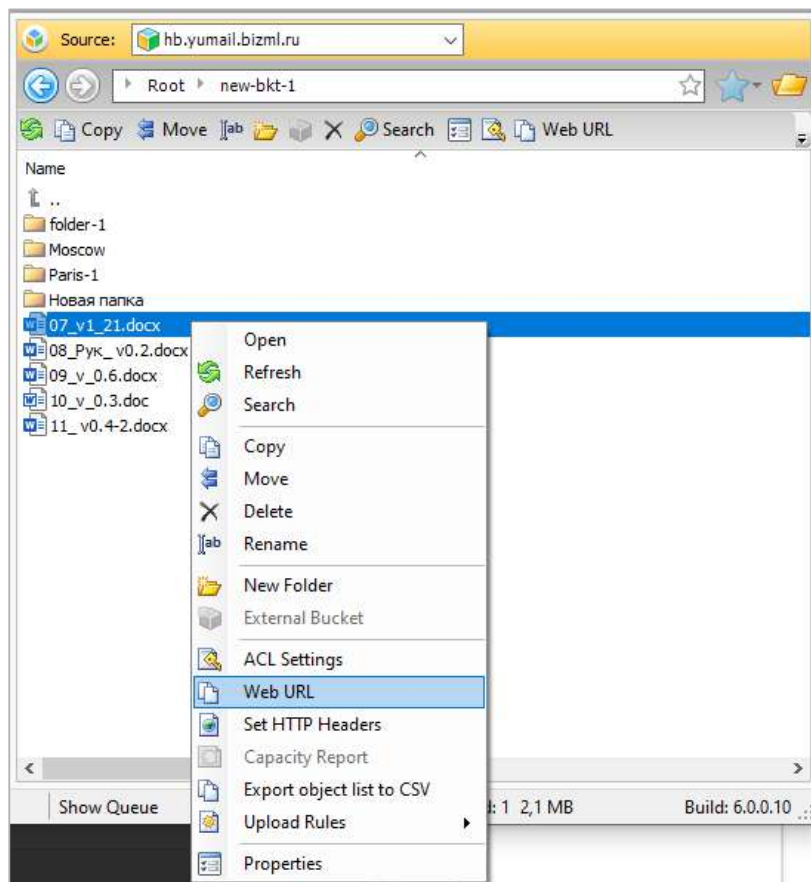


Рисунок 19 – Web URL в контекстном меню

Или, выберите нужный файл и нажмите кнопку «Web URL» на панели инструментов файлового менеджера:

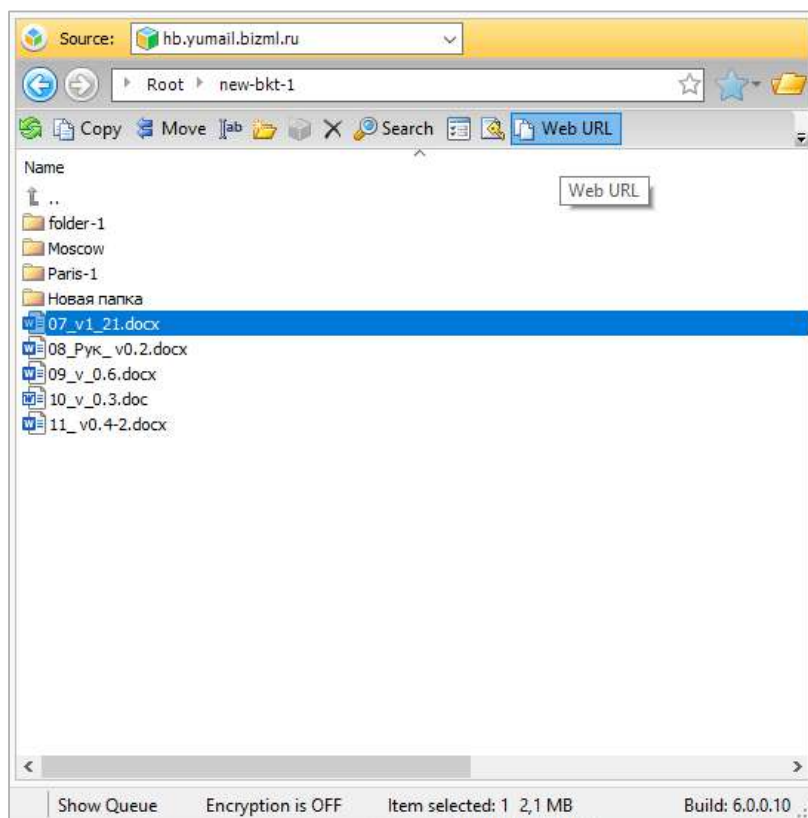


Рисунок 20 – «Web URL» на панели инструментов файлового менеджера

2. Далее в открывшемся диалогом окне задайте срок окончания действия URL и нажмите кнопку «Generate»:

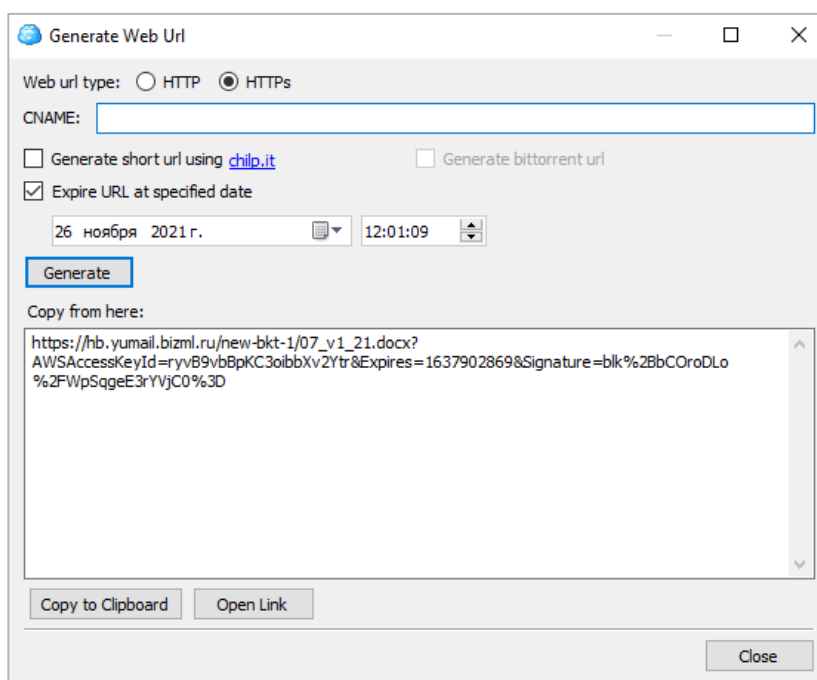


Рисунок 21 – Создание предварительно подписанного URL

5. API

Объектное хранилище предоставляет RESTful XML API для программного управления хранимыми данными с помощью стандартных HTTP-запросов. API-интерфейс совместим с API-интерфейсом Amazon AWS S3, что позволяет взаимодействовать с сервисом, используя уже известные инструменты.

Основной протокол доступа к объектному хранилищу – это протокол, созданный Amazon Web Services для сервиса AWS S3 (Simple Storage Service). S3 API — набор команд, которые «понимает» Объектное хранилище и выполняет в ответ некие действия.

5.1. Операции с бакетами

Операции с бакетами:

- CreateBucket – Создать бакет.
- ListBuckets – Получить список бакетов.
- HeadBucket – Проверка наличия бакета и доступа к нему.
- ListObjects – Получить список объектов бакета.
- DeleteBucket – Удалить бакет.

5.1.1. CreateBucket

Операция **PUT** создает новый бакет. Для создания бакета необходимо пройти регистрацию в Объектном хранилище и получить ключ доступа для авторизации запросов. Создание бакетов с помощью анонимных запросов невозможно. Инициатор запроса создания бакета автоматически становится его владельцем.

При использовании операции по созданию бакета можно указать проекты или группы, которым необходимо предоставить определенные права на бакет. Существует два способа предоставления прав при помощи заголовков запроса:

- Указание готового ACL в запросе с помощью заголовка **x-amz-acl** в запросе.
- Указание явным образом права доступа с помощью заголовков **x-amz-grant-read**, **x-amz-grant-write**, **x-amz-grant-read-acp**, **x-amz-grant-write-acp**, **x-amz-grant-full-control** в запросе.

Возможно использование готового ACL или указать права доступа явным образом, но использовать оба способа одновременно невозможно.

Пример запроса

```
PUT / HTTP/1.1
Host: my-test-bucket1.bizmrg.com
x-amz-acl: public-read
x-amz-content-sha256: c6f1fc479f5f690c443b73a258aacc06ddad09eca0b001e9640ff2cd56fe5710
x-amz-date: 20200831T173143Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-acl;x-amz-content-sha256;x-amz-
date,Signature=6cab03bef74a80a0441ab7fd33c829a2cdb46bba07e82da518cdb78ac238fda5

<CreateBucketConfiguration>
  <LocationConstraint>ru-msk</LocationConstraint>
</CreateBucketConfiguration>
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 17:31:43 GMT
Content-Length: 0
Content-Type: text/plain; charset=utf-8
Connection: close
```

5.1.2. ListBuckets

Операция **GET** вернет список существующих бакетов в проекте.

Пример запроса

```
GET / HTTP/1.1
Host: hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T183940Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=245e867a6a653b7b88cbb71a734daf2cbb4ba927d9aa5fdce57c85ab4f2b40b
```

Пример ответа

```
HTTP/1.1 200 OK
x-amz-request-id: tx000000000000002ba2427-0059651b6d-1268c-ru-mska
Date: Mon, 31 Aug 2020 17:31:43 GMT
Content-Length: 525
Content-Type: text/plain
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
```

```
<Owner>
  <ID>6174283</ID>
  <DisplayName>6174283</DisplayName>
</Owner>
<Buckets>
  <Bucket>
    <Name>static-images</Name>
    <CreationDate>2020-08-31T18:37:48.157Z</CreationDate>
  </Bucket>
  <Bucket>
    <Name>log-files</Name>
    <CreationDate>2020-08-31T18:37:48.157Z</CreationDate>
  </Bucket>
</Buckets>
</ListAllMyBucketsResult>
```

5.1.3. HeadBucket

Операция **HEAD** используется для определения существования бакета и наличия разрешения доступа к нему. Операция возвращает HTTP статус 200, если бакет существует с разрешением доступа к нему. Иначе операция вернёт либо HTTP ошибку 404, либо 403.

Пример запроса

```
HEAD / HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20170714T185156Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
```

Пример ответа

```
HTTP/1.1 200 OK
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzWAlQEjLbTMTagKdBX2z7I1/jGhDeJ3j6s80
x-amz-request-id: 32FE2CEB32F5EE25
x-amz-bucket-region: ru-msk
Date: Mon, 31 Aug 2020 21:34:56 GMT
Connection: close
```

5.1.4. ListObjects

Операция **GET** возвращает некоторые или все (до 1000) объекты в бакете. Можно использовать параметры запроса в качестве критериев выборки, чтобы отфильтровать объекты в бакете.

Для успешного выполнения операции, необходимо обладать правами на чтение бакета **READ**.

Пример запроса

```
GET / HTTP/1.1
Host: my-test-bucket1.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T172613Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=delbf8931e315c0576edb81a7d8be98874e847548fc70682f6c646e1cfd9177a
```

Пример ответа

```
HTTP/1.1 200 OK
x-amz-request-id: tx00000000000000029ac87-0059690330-8dla-ru-mska
Date: Mon, 31 Aug 2020 17:31:43 GMT
Content-Length: 858
Content-Type: application/xml
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>static-images</Name>
  <Prefix/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>example.txt</Key>
    <LastModified>2020-08-31T18:40:46.777Z</LastModified>
    <ETag>"b3a92f49e7ae64acbf6b3e76f2040f5e"</ETag>
    <Size>14</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>6174283</ID>
      <DisplayName>6174283</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>sammy.png</Key>
    <LastModified>2020-08-31T17:44:03.597Z</LastModified>
    <ETag>"fb08934ef619f205f272b0adfd6c018c"</ETag>
    <Size>35369</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>6174283</ID>
      <DisplayName>6174283</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
```

5.1.5. DeleteBucket

Операция **DELETE** удаляет бакет, указанный в запросе. Успех выполнения операции будет обозначен HTTP статусом 204.

Если получен HTTP статус 409, то удаление невозможно, так как в бакете есть объекты. Необходимо удалить все объекты из бакета, а затем попробовать снова удалить бакет.

Пример запроса

```
DELETE / HTTP/1.1
Host: my-test-bucket1.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20170710T181321Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=b0558a25e794bcd1ca9b620b4318bb8eb62ddb34e2b9c1921034bc5acd597b
```

Пример ответа

```
HTTP/1.1 204 No Content
Date: Mon, 31 Aug 2020 18:13:21 GMT
Connection: close
```

5.2. Операции с объектами

Операции с Объектами:

- Get – Скачать объект из бакета.
- Upload – Загрузить объект в бакет.
- Copy – Скопировать объект, находящийся в бакете.
- HeadObject – Получить информацию об объекте.
- Delete – Удалить объект.
- DeleteMultipleObjects – Удалить группу объектов по списку.

5.2.1. Get

Операция **GET** извлекает объект из бакета. Для выполнения операции необходимо обладать правами **READ** на объект.

Предоставив доступ **READ** анонимному пользователю, можно возвращать объект без использования заголовка авторизации.

У бакета нет иерархии каталогов, как в стандартной файловой системе. Тем не менее можно создать логическую иерархию, используя имена ключей объектов, которые подразумевают структуру папок. Например, можно назвать объект не **sample.jpg**, а **photos/2020/August/01.jpg**.

Чтобы извлечь объект с такой логической иерархией, следует указать в операции **GET** полное имя объекта.

Пример запроса

```
GET /example.txt HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T190539Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20170710/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=e912de79c88f07832558244bd867c3d834584c7f8b3d8efe4d0f0ba60b7a1dcb
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 19:05:39 GMT
x-amz-request-id: tx00000000000000279f46e-005963d003-1268c-ru-mska
Content-Type: text/plain
Content-Length: 14
Accept-Ranges: bytes
Last-Modified: Mon, 31 Aug 2020 19:05:09 GMT
Etag: "b3a92f49e7ae64acbf6b3e76f2040f5e"
Connection: close
```

5.2.2. Upload

Операция **PUT** добавляет объект в бакет. Для выполнения операции необходимо обладать правами **WRITE** на бакет.

Объектное хранилище добавляет объекты частично, если размер объекта превышает 50МБ. Если получен ответ об успешном выполнении, значит объект добавлен в бакет целиком.

При выполнении одновременных операций **PUT** и при наличии одинаковых загружаемых объектов, Объектное хранилище перезаписывает всё, кроме последнего записанного объекта.

Чтобы предотвратить повреждение данных при прохождении сети, рекомендуется использовать заголовок **Content-MD5**, благодаря которому происходит сверка объекта с предоставленным

значением MD5 и, в случае несовпадения, возвращает ошибку. Кроме того, можно вычислить MD5, помещая объект в бакет, и сравнить возвращаемый **ETag** с вычисленным значением MD5.

Пример запроса

```
PUT /example.txt HTTP/1.1
Content-Length: 14
Content-Type: text/plain
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: 003f0e5fe338b17be8be93fec537764ce199ac50f4e50f2685a753c4cc781747
x-amz-date: 20200831T194605Z
x-amz-meta-s3cmd-
attrs:uid:1000/gname:asb/uname:asb/gid:1000/mode:33204/mtime:1499727909/ctime:1499727909/
md5:fb08934ef619f205f272b0adfd6c018c/ctime:1499713540
x-amz-storage-class: STANDARD
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=content-length;content-type;host;x-amz-content-
sha256;x-amz-date;x-amz-meta-s3cmd-attrs;x-amz-storage-
class,Signature=a9a9e16da23e0b37ae8362824de77d66bba2edd702ee5f291f6ecbb9ebac6013
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 19:46:06 GMT
x-amz-request-id: tx0000000000000027bd57c-005963d97e-1268c-ru-mska
Content-Length: 0
Accept-Ranges: bytes
Last-Modified: Mon, 31 Aug 2020 19:05:09 GMT
Etag: "fb08934ef619f205f272b0adfd6c018c"
Connection: close
```

5.2.3. Copy

Операция **PUT** может создать копию объекта, которая уже хранится в бакете. Операция копирования идентична последовательному выполнению **GET** и **PUT**. Добавление заголовка **x-amz-copy-source** в запрос приводит к тому, что операция **PUT** копирует исходный объект в целевой бакет.

При копировании объекта можно сохранить большую часть метаданных (по умолчанию) или указать новые метаданные. Для записи новых метаданных при копировании объекта необходимо включить в запрос заголовок **x-amz-metadata-directive: REPLACE**. Однако ACL не копируется – для пользователя, отправляющего запрос, новый объект будет частный.

Все запросы на копирование должны пройти проверку подлинности и не могут содержать текст сообщения. Кроме того, необходимо обладать доступом **READ** к исходному объекту и доступ **WRITE** к целевому бакету.

Запрос на копирование может вернуть ошибку в двух случаях. Это может произойти, когда получен запрос на копирование, либо когда происходит копирование объектов. Если ошибка возникает до начала операции копирования, будет получена стандартная ошибка. Если ошибка возникает во время операции копирования, ошибка будет встроена в ответ 200 OK. Это означает, что ответ 200 OK может содержать сообщение как об успехе, так и об ошибке.

Пример запроса

```
PUT /copied-example.txt HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-copy-source: /static-images/example.txt
x-amz-date: 20200831T202253Z
x-amz-metadata-directive: COPY
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20170710/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-copy-source;x-amz-
date;x-amz-metadata-directive;x-amz-storage-
class,Signature=0cb03470dd80bdd41a4b8fb06c1800b27a5059b61b0303fe589578835531c877
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 20:22:54 GMT
x-amz-request-id: tx0000000000000027d8430-005963e21d-1268c-ru-mska
Content-Length: 183
Connection: close

<CopyObjectResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <LastModified>2020-08-31T20:22:54.167Z</LastModified>
  <ETag>7967bfe102f83fb5fc7e5a02bf05e8fc</ETag>
</CopyObjectResult>
```

5.2.4. HeadObject

Операция **HEAD** извлекает метаданные из объекта без возвращения самого объекта. Данная операция используется только в том случае, если нужны только метаданные объекта. Для того чтобы воспользоваться операцией **HEAD**, необходимо обладать правом **READ** на объект.

Для запроса операции **HEAD** указываются те же параметры, что и для операции **GET** для объекта. Ответ идентичен ответу **GET**, за исключением отсутствия тела ответа.

Если запрашиваемый объект не существует, то возвращаемая ошибка зависит от того, есть ли у учетной записи дополнительное разрешение **s3:ListBucket**.

- Если есть разрешение **s3:ListBucket** на бакете, то возвращается HTTP ошибка 404.
- Если нет, то возвращается HTTP ошибка 403.

Пример запроса

```
HEAD /example.txt HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T185156Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20170710/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 18:51:58 GMT
x-amz-request-id: tx0000000000000002ff1c9-00596912ce-6441-ru-mska
Content-Type: text/plain
Content-Length: 14
Accept-Ranges: bytes
Last-Modified: Mon, 31 Aug 2020 18:40:46 GMT
Etag: "b3a92f49e7ae64acbf6b3e76f2040f5e"
Connection: close
```

5.2.5. Delete

Операции **DELETE** удаляет объект, указанный в запросе.

Пример запроса

```
DELETE /sammy.png HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T194408Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20170710/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=c2a46b21e2e8589dfbfa54382030bbef8108b2504a9f1d8aaba70fb0e1c46522
```

Пример ответа

```
HTTP/1.1 204 No Content
Date: Mon, 31 Aug 2020 19:44:09 GMT
x-amz-request-id: tx00000000000000027bbc48-005963d908-1268c-ru-mska
Connection: close
```

5.2.6. DeleteMultipleObjects

Удаляет объекты по списку ключей, переданному в запросе. Список на удаление может содержать не более 1000 ключей. Отсутствие объекта не является ошибкой – если объекта не существует, то в ответе он будет отмечен как удаленный.

Заголовки `Content-MD5` и `Content-Length` обязательны, а список ключей на удаление передаётся в XML формате.

Пример запроса

```
DELETE /sammy.png HTTP/1.1
Host: [my-test-bucket1.hb.bizmrg.com] (/my-test-bucket1.hb.bizmrg.com)
Content-Length: 176
Content-MD5: 44c4dcd3d2f3645544a366ae481342fa
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T194408Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20170710/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=c2a46b21e2e8589dfbfa54382030bbef8108b2504a9f1d8aaba70fb0e1c46522

<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Object>
    <Key>picture.png</Key>
  </Object>
  <Object>
    <Key>picture2.jpg</Key>
  </Object>
</Delete>
```

Пример ответа

```
HTTP/1.1 204 No Content
Date: Mon, 31 Aug 2020 19:44:09 GMT
x-amz-request-id: tx0000000000000027bbc48-005963d908-1268c-ru-mska
Connection: close

<DeleteResult>
  <Deleted>
    <Key>picture.png</Key>
  </Deleted>
  <Error>
    <Key>some/another/key.txt</Key>
    <Code>TextErrorCode</Code>
    <Message>Describing message</Message>
  </Error>
</DeleteResult>
```

5.3. Списки управления доступом (ACL)

ACL (Access Control List) позволяет контролировать, какие операции разрешены каким пользователям. ACL может стоять как и на уровне всего бакета, так и на уровне конкретного объекта. Установить и прочесть ACL можно через приведенные методы ниже.

Операции с ACL:

- Get Bucket ACL – Получить ACL для бакета.
- Put Bucket ACL – Установить ACL у бакета.
- Get Object ACL – Получить ACL для объекта.
- Put Object ACL – Установить ACL у объекта.

Общая XML структура конфигурации ACL:

```
<AccessControlPolicy>
  <Owner>
    <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
    <DisplayName>test.help@mcs.mail.ru</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="CanonicalUser">
        <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
        <DisplayName>MCS_UserName</DisplayName>
      </Grantee>
      <Permission>WRITE</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

5.3.1. Get Bucket ACL

Получение ACL для бакета. Для получения необходимо иметь у бакета право **READ_ACP**.

Если право **READ_ACP** предоставлено анонимному пользователю, то можно получать ACL у бакета без использования заголовка авторизации.

Пример запроса

```
GET /?acl HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T174434Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
```

```
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-date,Signature=71dfa4666fb740d40d05307a29321c65cc620cdb17e8a9cb83d4f0e1b1b9d236
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 17:44:35 GMT
x-amz-request-id: tx00000000000002764fa6-005963bd03-1268c-ru-mska
Content-Type: application/xml
Content-Length: 848
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
    <DisplayName>MCS_UserName</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
        <DisplayName>MCS_Username</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

5.3.2. Put Bucket ACL

Установка ACL для бакета. Для установки ACL необходимо иметь у бакета право `WRITE_ACP`.

Есть два способа установки ACL:

- В теле запроса;
- В заголовках запроса.

Невозможно использовать для способа одновременно в одном запросе.

Пример запроса

```
PUT /?acl HTTP/1.1
Content-Type: application/xml
Content-Length: 675
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: 724483e3830b19d6960345c484fb7904b26e8f2fb34a6c002fa779353b68c8d8
x-amz-date: 20200831T183709Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-
date,Signature=1cf3f7771a4086375e5b6597026db6d55d84fbc86e3c3a86ec420ea9123e3163

<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 18:37:10 GMT
x-amz-request-id: tx00000000000000278ac49-005963c956-1268c-ru-mska
Content-Type: application/xml
Content-Length: 0
Connection: close
```

5.3.3. Get Object ACL

Получение ACL для объекта. Для получения необходимо иметь у объекта право **READ_ACP**.

Пример запроса

```
GET /sammy.png?acl HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T191224Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=950e133849cd19d626291fd2937d927957cf3e97a36707d30d51a9b61ac08a8e
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 19:12:24 GMT
x-amz-request-id: tx000000000000027a42dc-005963d198-1268c-ru-mska
Content-Type: application/xml
Content-Length: 848
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
    <DisplayName>MCS_UserName</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
        <DisplayName>MCS_UserName</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

5.3.4. Put Object ACL

Установка ACL для объекта. Для установки ACL необходимо иметь у объекта право WRITE_ACP.

Есть два способа установки ACL:

- В теле запроса;

- В заголовках запроса.

Невозможно использовать для способа одновременно в одном запросе.

Пример запроса

```
PUT /sammy.png?acl HTTP/1.1
Content-Type: application/xml
Content-Length: 443
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256:c0bd9ba784be78d4f38bbc1e3b0da2de2e7a8f4ee259b3b840369cf00a78dad2
x-amz-date:20200831T192142Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-
date,Signature=dfeeb2386f76b29097adadb35ac15f7d5f244f18cc95f082b0ac6d14ced48b10

<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 19:21:42 GMT
x-amz-request-id: tx0000000000000027aafc9-005963d3c6-1268c-ru-mska
Content-Type: application/xml
Content-Length: 0
Connection: close
```

5.4. Составная загрузка

Составная или многокомпонентная загрузка позволяет сохранять объекты в Объектном хранилище по частям. Это может пригодиться при загрузке или копировании больших объектов. Рекомендуем использовать составную загрузку для объектов от 100 МБ.

Составная загрузка состоит из следующих шагов:

- Initiate Multipart Upload – Инициализация загрузки.

- Upload Part – Загрузка объекта по частям.
- Complete Multipart Upload – Завершение загрузки путем объединения ранее; загруженных. частей
- Abort Multipart Upload – Прерывание загрузки.
- List Parts – Возврат списка загруженных частей.

5.4.1. Initiate Multipart Upload

Операция инициализирует многокомпонентную загрузку и возвращает идентификатор загрузки. Идентификатор загрузки используется для объединения всех частей одной многокомпонентной загрузки. Необходимо указывать этот идентификатор загрузки в каждом из последующих запросов на загрузку части.

Если настроено правило жизненного цикла на прерывание неоконченных многокомпонентных загрузок, то загрузка должна завершиться в течение количества дней, указанном в конфигурации жизненного цикла бакета. В противном случае для незавершенной многокомпонентной загрузки становится доступной операция прерывания, и сервис прерывает многокомпонентную загрузку.

ПРИМЕЧАНИЕ

После инициализации многокомпонентной загрузки и собственно загрузки одной или нескольких частей необходимо завершить или прервать многокомпонентную загрузку, чтобы приостановить списание оплаты за хранение загруженных частей. Только после завершения или прерывания многокомпонентной загрузки Объектное хранилище освобождает место, предоставляемое частям в хранилище, и прекращает производить списание оплаты за хранение этих частей.

Пример запроса

```
POST /multipart-file.tar.gz?uploads HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20200831T174652Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=e356966fff5a3d49d19c0e44e0fdb294964384a58061d3e60dfd1a4a5b605ad
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 286
Content-Type: application/xml
Date: Mon, 31 Aug 2020 17:46:53 GMT
x-amz-request-id: tx00000000000000ab66a13-005991e20d-66a8-ru-mska
```

```
Connection: close
```

```
<?xml version="1.0" encoding="UTF-8"?>
<InitiateMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>static-images</Bucket>
  <Key>multipart-file.tar.gz</Key>
  <UploadId>2~iCw_lDY8VoBhoRrIJbPMrUqnE3Z-3Qh</UploadId>
</InitiateMultipartUploadResult>
```

5.4.2. Upload Part

Операция загружает часть многокомпонентной загрузки. Для выполнения этой операции необходимо предоставить данные из части в запросе. Для загрузки части из существующего объекта используется операция загрузки части (копии).

Часть может иметь любой номер от 1 до 10 000 включительно. Номер части однозначно определяет часть и ее положение в создаваемом объекте. Если загружается новую часть, с назначенным ей номером, используемым для одной из существующих частей, то существующая часть будет перезаписана. Размер каждой части, кроме последней, должен составлять как минимум 5 МБ. Последняя часть многокомпонентной загрузки не имеет ограничения по размеру.

Чтобы удостовериться в том, что данные не искажаются при их передаче по сети, следует указать заголовок Content-MD5 в запросе загрузки части. Объектное хранилище сверяет данные из частей с предоставленным значением MD5-хеша и при их несовпадении возвращает ошибку.

Пример запроса

```
PUT /multipart-file.tar.gz?partNumber=1&uploadId=2~iCw_lDY8VoBhoRrIJbPMrUqnE3Z-3Qh
HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
Content-Length: 5242880
x-amz-content-sha256: 6ab0931e613fd0cd39af0fa4733edb6ad942df0ad33e057713d44df7195b6ecf
x-amz-date: 20200831T184459Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=2e4adca0a3164af8717f340ca874b4f3036dd0294b590a70f2147427221ca1e8
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Content-Type: application/xml
Date: Mon, 31 Aug 2020 18:45:01 GMT
Etag: "d8d3ed3a4de016917a814a2cf5acad3c"
x-amz-request-id: tx00000000000000ab85dab-005991efac-66a8-ru-mska
Connection: close
```

5.4.3. Complete Multipart Upload

Операция завершает многокомпонентную загрузку путем объединения ранее загруженных частей. После получения данного запроса Объектное хранилище объединяет все загруженные части в возрастающем порядке по номеру частей, создавая новый объект. В запросе на завершение многокомпонентной загрузки необходимо предоставить список частей. Для каждой части из списка следует предоставить номер части и значение заголовка **ETag**, возвращаемое после загрузки этой части.

Обработка запроса на завершение многокомпонентной загрузки может занять несколько минут. После начала обработки запроса, Объектное хранилище отправляет заголовок ответа HTTP, содержащий ответ «200 OK». Во время обработки запроса периодически отправляются пробельные символы для предотвращения превышения лимита времени ожидания. Так как ошибка может возникать в запросе после отправления первоначального ответа «200 OK», необходимо проверять тело ответа для определения успешности исполнения запроса.

Пример запроса

```
POST /multipart-file.tar.gz?uploadId=2~iCw_1DY8VoBhoRrIJbPMrUqnE3Z-3Qh HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
Content-Length: 358
x-amz-content-sha256: 6ab0931e613fd0cd39af0fa4733edb6ad942df0ad33e057713d44df7195b6ecf
x-amz-date: 20200831T184459Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=2e4adca0a3164af8717f340ca874b4f3036dd0294b590a70f2147427221ca1e8

<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"d8d3ed3a4de016917a814a2cf5acad3c"</ETag>
  </Part>
  <Part>
    <PartNumber>2</PartNumber>
    <ETag>"adf5feafc0fe4632008d5cb30beb1c49"</ETag>
  </Part>
  <Part>
    <PartNumber>3</PartNumber>
    <ETag>"363f6bb50866541d78e5f6f626592263"</ETag>
  </Part>
</CompleteMultipartUpload>
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 336
Content-Type: application/xml
Date: Mon, 31 Aug 2020 18:45:01 GMT
```

```
x-amz-request-id: tx00000000000000ab962c8-005991f6fe-66a8-ru-mska
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Location>my-test-bucket1.hb.bizmrg.com</Location>
  <Bucket>my-test-bucket1</Bucket>
  <Key>multipart-file.tar.gz</Key>
  <ETag>f935869350d7cbfcdd219df3f377531b-3</ETag>
</CompleteMultipartUploadResult>
```

5.4.4. Abort Multipart Upload

Операция прерывает многокомпонентную загрузку. После прерывания многокомпонентной загрузки невозможно загрузить дополнительные части, используя идентификатор загрузки прерванной многокомпонентной загрузки. Место, выделенное для хранения ранее загруженных частей, будет освобождено. При этом если происходит загрузка каких-либо частей, то такая операция может завершиться или прерваться. В результате этого может появиться необходимость многократного прерывания многокомпонентной загрузки для полного высвобождения пространства, занимаемого всеми частями.

Пример запроса

```
DELETE /multipart-file.tar.gz?uploadId=2~iCw_lDY8VoBhoRrIJbPMrUqnE3Z-3Qh HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
Content-Length: 5242880
x-amz-content-sha256:e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20200831T202611Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=9e13d3f9e71ca5fb034fe66e92c60e30b3az3e177573702dd11d2b541358bf92
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 18:45:01 GMT
x-amz-request-id: tx00000000000000abbaefe-0059920764-66a8-ru-mska
Connection: close
```

5.4.5. List Parts

Операция возвращает список частей, загруженных по определенной многокомпонентной загрузке. Операция должна включать в себя идентификатор загрузки, который получен после отправки запроса на инициализацию многокомпонентной загрузки. Такой запрос возвращает не более 1000 загруженных частей. Можно ограничить количество возвращаемых частей, указав параметр запроса `max-parts`. Если многокомпонентная загрузка состоит из более, чем 1000 частей, то ответ возвращает

элемент `NextPartNumberMarker` и поле `IsTruncated` со значением `true`. В последующих запросах на просмотр списка частей можно включать параметр строки запроса `part-number-marker`, устанавливая для него значение поля `NextPartNumberMarker` из предыдущего ответа.

Пример запроса

```
GET /?uploads&delimiter=Delimiter&encoding-type=EncodingType&key-marker=KeyMarker&max-uploads=MaxUploads&prefix=Prefix&upload-id-marker=UploadIdMarker HTTP/1.1

PUT /multipart-file.tar.gz?partNumber=1&uploadId=2~iCw_lDY8VoBhoRrIJbPMrUqnE3Z-3Qh
HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
Content-Length: 5242880
x-amz-content-sha256: 6ab0931e613fd0cd39af0fa4733edb6ad942df0ad33e057713d44df7195b6ecf
x-amz-date: 20200831T184459Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=2e4adca0a3164af8717f340ca874b4f3036dd0294b590a70f2147427221cale8
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 980
Content-Type: application/xml
Date: Mon, 31 Aug 2020 18:45:01 GMT
Etag: "d8d3ed3a4de016917a814a2cf5acad3c"
x-amz-request-id: tx00000000000000ab85dab-005991efac-66a8-ru-mska
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<ListPartsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>my-test-bucket1</Bucket>
  <Key>multipart-file.tar.gz</Key>
  <UploadId>2~iCw_lDY8VoBhoRrIJbPMrUqnE3Z-3Qh</UploadId>
  <StorageClass>STANDARD</StorageClass>
  <PartNumberMarker>0</PartNumberMarker>
  <NextPartNumberMarker>1</NextPartNumberMarker>
  <MaxParts>1000</MaxParts>
  <IsTruncated>>false</IsTruncated>
  <Owner>
    <ID>eab55955-ebdb-4f18-a94d-f3558ff150da</ID>
    <DisplayName>MCS_UserName</DisplayName>
  </Owner>
  <Part>
    <LastModified>2017-08-14T18:45:01.601Z</LastModified>
    <PartNumber>1</PartNumber>
    <ETag>"d8d3ed3a4de016917a814a2cf5acad3c"</ETag>
    <Size>5242880</Size>
  </Part>
  <Part>
    <LastModified>2017-08-14T18:45:01.601Z</LastModified>
    <PartNumber>2</PartNumber>
```

```

    <ETag>"adf5feafc0fe4632008d5cb30beblc49"</ETag>
    <Size>5242880</Size>
  </Part>
</ListPartsResult>

```

5.5. Управление жизненным циклом

Все доступные методы для работы с жизненным циклом объектов:

- Get Bucket Lifecycle Configuration – Просмотр конфигурации жизненного цикла в бакете.
- Configure Bucket Lifecycle – Настройка правил конфигурации жизненного цикла.
- Delete Bucket Lifecycle – Удаление конфигурации жизненного цикла.

5.5.1. Общий вид конфигурации

```

<LifecycleConfiguration>
  <Rule>
    <ID>Описание правила</ID>
    <Status>{Enabled|Disabled}</Status>
    <Filter>
      <Prefix>префикс_ключа</Prefix>
    </Filter>
    <Transition>
      <StorageClass>Идентификатор класса хранилища</StorageClass>
      <!-- <Date> или <Days> -->
    </Transition>
    ...
    <Expiration>
      <!-- <Date> или <Days> -->
    </Expiration>
    ...
  </Rule>
  <Rule>
    ...
  </Rule>
</LifecycleConfiguration>

```

5.5.2. Get Bucket Lifecycle Configuration

В конфигурации жизненного цикла для бакета можно указывать правило жизненного цикла с помощью префикса имени ключа объекта, одного или нескольких тегов объекта или сочетания обоих этих параметров. Ответ содержит элемент фильтра, которым можно воспользоваться для изменения параметров фильтра и выбора поднабора объектов, к которым должно быть применимо правило.

Операция **GET** возвращает информацию по конфигурации жизненного цикла, установленной для бакета.

Пример запроса

```
GET /?lifecycle HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T001757Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=e92e48fb16dad3d9d332460adde86493b8930262d9385e002b0408e17a2781f4
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 17:44:35 GMT
x-amz-request-id: tx000000000000000023935-005a613936-fcf92-ru-mska
Content-Type: application/xml
Content-Length: 488
Connection: close

<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <ID>Expire old logs</ID>
    <Prefix>logs</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>90</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Remove uncompleted uploads</ID>
    <Status>Enabled</Status>
    <Prefix/>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>1</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

5.5.3. Configure Bucket Lifecycle

В конфигурации жизненного цикла для бакета можно указывать правило жизненного цикла с помощью префикса имени ключа объекта.

Операция **PUT** создает новую конфигурацию жизненного цикла для бакета или замещает существующую конфигурацию жизненного цикла.

Пример запроса

```
PUT /?lifecycle HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
Content-Length: 488
Content-Type: application/xml
x-amz-content-sha256: 34850007f92ec3331486b48fd7db15f48315fe73c4a9b135e6d9fd629276c1e7
x-amz-date: 20200831T000345Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=content-md5;content-type;host;x-amz-content-sha256;x-
amz-date,Signature=fc07a541c2acdbf7527eba358afa0a6d460c9bfec539dd29dfa6b5b854aae109

<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <ID>Expire old logs</ID>
    <Prefix>logs</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>90</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Remove uncompleted uploads</ID>
    <Status>Enabled</Status>
    <Prefix/>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>1</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 17:31:43 GMT
x-amz-request-id: tx00000000000000010ad2b-005a6135e2-f647d-ru-mska
Content-Length: 0
Content-Type: application/xml
Connection: close
```

5.5.4. Delete Bucket Lifecycle

Операция **DELETE** удаляет конфигурацию жизненного цикла из указанного бакета. Удаляются все правила конфигурации жизненного цикла из подресурса жизненного цикла, связанного с бакетом, что позволяет исключать срок действия из объектов. Как следствие, сервис больше не будет автоматически удалять объекты согласно правилам, содержащимся в удаленной конфигурации жизненного цикла.

Пример запроса

```
DELETE /?lifecycle HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T204101Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=376fe41764fe6493a33160b36055d8f617b92f9337bce0cf91bc9c5b1e7482b2
```

Пример ответа

```
HTTP/1.1 204 No Content
Date: Mon, 31 Aug 2020 18:13:21 GMT
Connection: close
```

5.6. Совместное использование ресурсов между разными источниками

Сервис объектное хранилище Объектного хранилища позволяет управлять конфигурацией CORS бакета. Для загрузки конфигурации CORS необходимо сформировать XML-документ.

Перечень доступных методов для конфигураций CORS бакета:

- Get Bucket CORS – Получить список конфигураций CORS для бакета.
- Set Bucket CORS – Установить конфигурацию CORS.
- Delete Bucket CORS – Удалить конфигурацию CORS.

5.6.1. Общий вид XML конфигурации CORS

```
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>URL</AllowedOrigin>
    <AllowedMethod>HTTP_Method</AllowedMethod>
    <AllowedHeader>Header_Name</AllowedHeader>
    ...
  </CORSRule>
  ...
</CORSConfiguration>
```

5.6.2. Get Bucket CORS

Операция **GET** возвращает информацию по конфигурации CORS, установленной для бакета.

Для того чтобы воспользоваться данной операцией, необходимо обладать правами на запись WRITE_ACP. Владелец бакета имеет это разрешение по умолчанию и может предоставлять его другим пользователям.

Пример запроса

```
GET /?cors HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T185319Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=f7d7879992a9f3a06ddacd59e53ac318e99b2ed6230692b30099739e34469a91
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 18:53:20 GMT
x-amz-request-id: tx00000000000000279651f-005963cd20-1268c-ru-mska
Content-Type: application/xml
Content-Length: 430
Connection: close

<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <CORSRule>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedOrigin>http://example.com</AllowedOrigin>
    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedOrigin>*</AllowedOrigin>
  </CORSRule>
</CORSConfiguration>
```

5.6.3. Set Bucket CORS

Операция **PUT** устанавливает конфигурацию CORS для бакета. Если конфигурация уже существует, то она переписывается.

Для того чтобы воспользоваться данной операцией, необходимо обладать правами на запись WRITE_ACP.

Можно установить эту конфигурацию на бакете так, что он сможет обслуживать запросы Cross-origin. Например, можно предоставить доступ запросу из источника <http://www.example.com> к бакету my.example.bucket.com, используя функционал браузера XMLHttpRequest.

Для разрешения совместного использования ресурсов между разными источниками (CORS) на бакете необходимо добавить подресурс CORS на бакет. Подресурс CORS является XML-документом, в котором настраиваются правила, определяющие источники и методы HTTP, которые можно использовать в вашем бакете. Максимальный размер документа - 64 КБ. Например, у конфигурации CORS на бакете могут быть установлены два нижеследующих правила:

- Первое правило CORSRule разрешает использовать Cross-origin-запросы **PUT**, **POST** и **DELETE** из источников <https://www.example.com>. Данное правило также разрешает использовать все заголовки в предварительном запросе **OPTIONS** с помощью заголовка **Access-Control-Request-Headers**. Следовательно, в качестве ответа на любой предварительный запрос OPTIONS, сервис возвращает любой запрошенный заголовок.
- Второе правило разрешает использовать Cross-origin-запросы GET со всех источников. Подстановочный знак «*» указывает на возможность использования любого источника.

Пример запроса

```
PUT /?cors HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
Content-Length: 374
Content-Type: application/xml
x-amz-content-sha256: 745320970930725bd18820ec990f7334960f0a47358be189e77504cc094be77e
x-amz-date: 20200831T185043Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=content-md5;content-type;host;x-amz-content-sha256;x-
amz-date,Signature=f52b2bfb6ec975c86cadd2e51a6ee9842c6151b737e46ce90a3cb3cc0d0dea97

<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://example.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
  </CORSRule>
</CORSConfiguration>
```

Пример ответа

```
HTTP/1.1 200 OK
Date: Mon, 31 Aug 2020 18:50:44 GMT
x-amz-request-id: tx0000000000000027946fc-005963cc84-1268c-ru-mska
Content-Type: application/xml
Content-Length: 0
Connection: close
```

5.6.4. Delete Bucket CORS

Операция **DELETE** удаляет информацию по конфигурации CORS, установленной для бакета.

Для того чтобы воспользоваться данной операцией, необходимо обладать правами на запись WRITE_ACP.

Пример запроса

```
DELETE /static-images?cors HTTP/1.1
Host: my-test-bucket1.hb.bizmrg.com
x-amz-content-sha256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20200831T182537Z
Authorization: AWS4-HMAC-SHA256 Credential=II5JDQBAN3JYM4DNEB6C/20200831/ru-
msk/s3/aws4_request,SignedHeaders=content-md5;content-type;host;x-amz-content-sha256;x-
amz-date,Signature=e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
```

Пример ответа

```
HTTP/1.1 204 No Content
Date: Mon, 31 Aug 2020 18:25:38 GMT
x-amz-request-id: tx000000000000002fae1f-0059690ca2-6441-ru-mska
Connection: close
```

5.7. Описание типов данных

5.7.1. AbortIncompleteMultipartUpload

Имя	Описание	Тип	Обязательное
DaysAfterInitiation	Количество дней, по истечении которых удаляется незавершенная составная загрузка	Integer	Нет

5.7.2. AccessControlPolicy

Имя	Описание	Тип	Обязательное
Owner	Владелец бакета	Owner	Нет
AccessControlList	Array of Grant	String	Нет

5.7.3. Bucket

Имя	Описание	Тип	Обязательное
Name	Наименование бакета	String	Нет
CreationDate	Дата создания. Эта дата может измениться при внесении изменений, например, при изменении политики бакета	Timestamp	Нет

5.7.4. CompletedMultipartUpload

Имя	Описание	Тип	Обязательное
Part	Массив CompletedPart	Array of CompletedPart	Нет

5.7.5. CompletedPart

Имя	Описание	Тип	Обязательное
ETag	Тег объекта, возвращенный при загрузке части	String	Нет
PartNumber	Номер составной загрузки	Integer	Нет

5.7.6. CompleteMultipartUploadResult

Имя	Описание	Тип	Обязательное
Bucket	Наименование бакета, содержащего загружаемый объект	String	Нет
ETag	Тег объекта, возвращенный при загрузке части	String	Нет
Key	Ключ объекта	String	Нет
Location	URI, идентифицирующий загружаемый объект	String	Нет

5.7.7. Contents

Имя	Описание	Тип	Обязательное
Contents	Данные об объекте	Objects	Нет

5.7.8. CopyObjectResult

Имя	Описание	Тип	Обязательное
LastModified	Дата последнего изменения объекта	String	Нет
ETag	Возвращает ETag нового объекта. ETag отражает только изменения содержимого объекта, но не его метаданных	String	Нет

5.7.9. CORSConfiguration

Имя	Описание	Тип	Обязательное
CORSRule	Набор источников и методов доступ между которыми необходимо разрешить	Array of CORSRule	Нет

5.7.10. CORSRule

Имя	Описание	Тип	Обязательное
AllowedHeaders	Заголовки, указанные в заголовке <code>Access-Control-Request-Headers</code> . Эти заголовки разрешены в предварительном запросе <code>OPTIONS</code> . В ответ на любой предварительный запрос <code>OPTIONS</code> Объектное хранилище возвращает все разрешенные запрошенные заголовки	Array of strings	Нет
AllowedMethods	HTTP-метод, который разрешается выполнять источнику. Допустимые значения: <code>GET</code> , <code>PUT</code> , <code>HEAD</code> , <code>POST</code> и <code>DELETE</code>	Array of strings	Нет
AllowedOrigins	Один или несколько источников, из которых вы хотите предоставить клиентам доступ к бакету	Array of strings	Нет
ExposeHeaders	Один или несколько заголовков в ответе, к которым вы хотите предоставить клиентам доступ из своих приложений (например, из объекта JavaScript XMLHttpRequest)	Array of strings	Нет
ID	Уникальный идентификатор правила	Array of strings	Нет
MaxAgeSeconds	Время в секундах, в течение которого ваш браузер должен кэшировать предварительный ответ для указанного ресурса	Integer	Нет

5.7.11. CreateBucketConfiguration

Имя	Описание	Тип	Обязательное
LocationConstraint	Регион, в котором создается бакет. Если регион не указан, он создается в регионе по умолчанию	String	Нет

5.7.12. CreatePrefixKeyResult

Имя	Описание	Тип	Обязательное
BucketName	Наименование бакета	String	Нет
Prefix	Префикс пути, который будет доступен по данным ключам	String	Нет
UserName	Пользователь, которому принадлежат ключи	String	Нет
SecretKey	Секретный ключ, возвращается только при создании, получить ключ впоследствии невозможно	String	Нет
AccessKey	Публичный ключ	String	Нет

5.7.13. Delete

Имя	Описание	Тип	Обязательное
Objects	Объекты для удаления	Array of ObjectIdentifier	Да

5.7.14. DeletedObject

Имя	Описание	Тип	Обязательное
Key	Ключ удаленного объекта	String	Нет

5.7.15. DeletePrefixKeyResult

Имя	Описание	Тип	Обязательное
Prefix	Префикс пути, который будет доступен по данным ключам	String	Нет
UserName	Пользователь, которому принадлежат ключи	String	Нет

5.7.16. DeleteResult

Имя	Описание	Тип	Обязательное
Deleted	Контейнер для успешно удаленных объектов	Array of DeletedObject	Да
Error	Контейнер для описания объекта, который не удалось удалить	Array of Error	Нет

5.7.17. Error

Имя	Описание	Тип	Обязательное
Code	Код ошибки	String	Нет
Key	Ключ объекта	String	Нет
Message	Сообщение об ошибке	String	Нет

5.7.18. Grant

Имя	Описание	Тип	Обязательное
Grantee	Лицо, которому предоставлены разрешения	Grantee	Нет
Permission	Тип разрешения, предоставленное лицу	Одно из значений FULL_CONT ROL / WRITE / WRITE_ACP / READ / READ_ACP	Нет

5.7.19. Grantee

Имя	Описание	Тип	Обязательное
ID	ID лица, которому предоставлены разрешения	String	Нет
DisplayName	Отображаемое имя лица, которому предоставлены разрешения	String	Нет

5.7.20. InitiateMultipartUploadResult

Имя	Описание	Тип	Обязательное
Bucket	Наименование бакета, в который инициирована составная загрузка	String	Да
Key	Ключ объекта, для которого инициирована составная загрузка	String	Да
UploadId	ID инициированной составной загрузки	String	Да

5.7.21. LifecycleConfiguration

Имя	Описание	Тип	Обязательное
Rule	Массив LifecycleRule	Array of LifecycleRule	Нет

5.7.22. LifecycleExpiration

Имя	Описание	Тип	Обязательное
Date	Дата, в которую объект должен быть удален	Timestamp	Нет
Days	Время жизни объекта в днях	Integer	Нет

5.7.23. LifecycleRule

Имя	Описание	Тип	Обязательное
ID	Уникальный идентификатор правила	String	Нет
Status	Если Enabled правило применяется в данный момент, если Disabled правило в настоящее время не применяется.	String	Нет
Filter	Контейнер для фильтра правила жизненного цикла	LifecycleRuleFilter	Нет
Transition	Указывает, когда объект переходит к указанному классу хранилища	Array of Transition	Нет
Expiration	Указывает срок жизни объекта	LifecycleExpiration	Нет
AbortIncompleteMultipartUpload	Указывает количество дней с начала незавершенной многокомпонентной загрузки, по истечении которых Объектное хранилище безвозвратно удаляет все части загрузки	AbortIncompleteMultipartUpload	Нет

5.7.24. LifecycleRuleFilter

Имя	Описание	Тип	Обязательное
Prefix	Префикс, идентифицирующий один или несколько объектов, к которым применяется правило	String	Нет

5.7.25. ListAllMyBucketsResult

Имя	Описание	Тип	Обязательное
Owner	Информация о владельце	Owner	Нет
Buckets	Список бакетов	Array of Bucket	Нет

5.7.26. ListBucketResult

Имя	Описание	Тип	Обязательное
IsTruncated	Указывает, были ли возвращены (true) или нет (false) все результаты. Все результаты могут не быть возвращены, если количество результатов превышает указанное <code>MaxKeys</code>	Boolean	Нет
MaxKeys	Максимальное количество объектов, возвращаемых в теле объекта	String	Нет
Name	Наименование бакета	String	Нет
Prefix	Ключи, начинающиеся с указанного префикса	String	Нет
Contents	Метаданные о каждом возвращенном объекте	Array of Contents	Нет

5.7.27. ListPartsResult

Имя	Описание	Тип	Обязательное
Bucket	Наименование бакета, в который инициирована составная загрузка	String	Нет
IsTruncated	Указатель усечения списка возвращаемых частей	Boolean	Нет
Key	Ключ объекта	String	Нет
MaxParts	Максимальное количество частей, которые были разрешены в ответе	Integer	Нет
NextPartNumberMarker	Указатель последней части составной загрузки в ответе для усеченного списка возвращаемых частей	Integer	Нет
Owner	Владелец объекта	Owner	Нет
Part	Массив частей составной загрузки	Array of Part	Нет
PartNumberMarker	Указатель последней части составной загрузки в ответе для усеченного списка возвращаемых частей	Integer	Нет

Имя	Описание	Тип	Обязательное
StorageClass	Класс хранения, заполняется значением STANDARD	String	Нет
UploadId	Идентификатор загрузки, идентифицирующий составную загрузку, части которой указаны в списке	String	Нет

5.7.28. ListPrefixKeysResult

Имя	Описание	Тип	Обязательное
BucketName	Наименование бакета	String	Нет
IsTruncated	Указывает, были ли возвращены (true) или нет (false) все результаты	Boolean	Нет
Marker	Имя или часть имени пользователя с которого начнется листинг	String	Нет
NamePrefix	Префикс по именам пользователей	String	Нет
MaxKeys	Максимальное количество элементов в листинге	String	Нет
Contents	Блок, содержащий пользователя	Array of Contents	Нет
UserName	Пользователь, которому принадлежат ключи	String	Нет
Prefix	Префикс, по которому доступны данные для этого пользователя	String	Нет

5.7.29. Object

Имя	Описание	Тип	Обязательное
ETag	Хэш объекта. ETag отражает только изменения содержимого объекта, но не его метаданных	String	Нет
Key	Имя объекта. Используется как ключ для извлечения объекта	String	Нет
LastModified	Дата создания объекта	Timestamp	Нет
Owner	Владелец объекта	Owner	Нет
Size	Размер объекта в байтах	Integer	Нет
StorageClass	Класс хранилища, используемый для хранения объекта. В текущей реализации принимает значение STANDARD	String	Нет

5.7.30. ObjectIdentifier

Имя	Описание	Тип	Обязательное
Key	Ключ объекта	String	Да

5.7.31. Owner

Имя	Описание	Тип	Обязательное
DisplayName	Отображаемое имя владельца	String	Нет
ID	ID владельца	String	Нет

5.7.32. Part

Имя	Описание	Тип	Обязательное
ETag	Тег объекта, возвращаемый при загрузке части объекта	String	Нет
LastModified	Дата и время загрузки части объекта	Timestamp	Нет
PartNumber	Номер составной загрузки	Integer	Нет
Size	Размер загруженной части составной загрузки	Integer	Нет

5.7.33. Transition

Имя	Описание	Тип	Обязательное
Date	Указывает, когда объекты переводятся в указанный класс хранения	Timestamp	Нет
Days	Указывает количество дней после создания, когда объекты переводятся в указанный класс хранения	Integer	Нет
StorageClass	Идентификатор класса хранилища	String	Нет