WHITE PAPER

BEST PRACTICE GUIDE
ПО УСТАНОВКЕ И НАСТРОЙКЕ TATLIN.OBJECT

YADRO TATLIN

О WHITE PAPERS

Данный документ описывает архитектурные принципы и подходы, а также рекомендации по их применению при построении корпоративных решений на основе продуктов производства компании YADRO.

Документ предназначен для широкой аудитории, но также может включать материалы, требующие специальной профессиональной подготовки, ориентированные на системных архитекторов, администраторов, сервисных инженеров и других IT-специалистов, участвующих в планировании, проектировании и внедрении комплексных решений.

! Представленная информация основана на результатах исследований, а также экспертизе специалистов YADRO. Документ носит не предписывающий, а информационный характер, целью которого является информирование получателей документа о предмете произведенного исследования и о его результатах. При этом конкретные результаты и эффективность конечных решений зависят от множества факторов, которые могут различаться в зависимости от особенностей инфраструктуры и условий эксплуатации. Ни при каких обстоятельствах YADRO не несет ответственность за последствия применения получателем документа рекомендаций, инструкций, подходов и решений, указанных в документе.

Запрещается копировать, воспроизводить, изменять, публиковать, передавать, распространять, продавать информацию (часть информации), содержащуюся в документе, или создавать производные работы на основе содержания документа без предварительного письменного разрешения YADRO.

Любая информация или утверждения относительно возможностей, емкости, производительности или других характеристик продукции, указанной в документе, предоставляются исключительно на условиях «как есть» и приведена исключительно в целях описания произведенного исследования. YADRO оставляет за собой право вносить изменения/обновлять информация в документе без уведомления.

«YADRO» является зарегистрированным товарным знаком ООО «КНС ГРУПП». Другие названия продуктов и компаний, упомянутые здесь, могут быть товарными знаками или торговыми наименованиями их соответствующих владельцев.

Актуальные версии документации публикуются на информационном ресурсе YADRO по адресу service.yadro.com.

ОГЛАВЛЕНИЕ

1. ОБ ЭКСПЕРТАХ ... 4
2. ВВЕДЕНИЕ ... 5
3. ПЛАНИРОВАНИЕ РАЗМЕЩЕНИЯ И УСТАНОВКА СХД TATLIN.OBJECT ... 6
   - 3.1. Атрибуты узла хранения ... 6
   - 3.2. Размещение узлов по площадкам. Планирование сетевого подключения ... 7
   - 3.3. Варианты размещения узлов СХД ... 7
4. ПОЛИТИКИ ХРАНЕНИЯ ... 12
5. ЭНДПОИНТЫ (БАЗОВЫЕ URL), НЕЙМСПЕЙСЫ, TLS-СЕРТИФИКАТЫ ... 23
6. НАСТРОЙКА БАЛАНСИРОВКИ НАГРУЗКИ ... 25
   - 6.1. Варианты схем подключения ... 27
7. ПРИЛОЖЕНИЕ 1: ПОЛУЧЕНИЕ ИНФОРМАЦИИ О СОСТОЯНИИ СХД (УТИЛИТА FROSTFS-CLI) ... 36
   - 7.1. Вывод карты сети ... 36
   - 7.2. Вывод политики бакета (контейнера) ... 36
   - 7.3. Вывод списка контейнерных узлов ... 37
   - 7.4. Соотнесение s3-объекта и OID ... 38
   - 7.5. Вывод списка узлов, где расположена копия объекта ... 39
   - 7.6. Проверка наличия объекта на контейнерном узле ... 40
   - 7.7. Получение S3-имени объекта ... 40
8. ЗАКЛЮЧЕНИЕ ... 41
9. ДОПОЛНИТЕЛЬНЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ ... 41

ОБ ЭКСПЕРТАХ

Андрей Баранов — специалист с более чем 25-летним опытом работе в сфере ИТ. Имеет опыт работы в Motorola, EMC, Dell, YADRO, где участвовал в разработке ПО, внедрении и поддержке инфраструктурных IT решений.

Последние 10 лет специализируется в области виртуализации ИТ инфраструктуры, а так же в scale-out системах хранения данных (прежде всего объектных и файловых), включая отечественных производителей (YADRO).

В компании YADRO Андрей занимает позицию ведущего специалиста отдела мультивендорной сервисной экспертизы.
Андрей активно участвует в разработке и тестировании проприетарных продуктов YADRO семейства TATLIN и обеспечивает глубокую интеграцию продуктов TATLIN в инфраструктуру заказчиков.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

ВВЕДЕНИЕ

TATLIN.OBJECT — объектная система хранения данных, которая поставляется в виде программно-аппаратного комплекса с поддержкой протоколов S3, HTTP(S), gRPC. В основе системы лежит эффективный децентрализованный распределенный движок хранения данных, спроектированный для работы в сложных условиях нестабильной глобальной сети, отказов дисков и других узлов. Управление системой происходит централизованно, независимо от количества узлов хранения. Каждый узел хранения использует все доступное локальное дисковое пространство для данных и их индексации. Избыточность и защита обеспечивается на уровне всей сети. Часть узлов помимо хранения данных занимается мониторингом сети хранения и поддержанием актуального списка доступных узлов. Эта информация хранится в реплицируемой на все узлы специализированной базе данных. Таким образом, в системе нет единой точки отказа или центрального источника информации, ограничивающего производительность системы. Это позволяет размещать узлы в разных ЦОД в удаленных регионах без дополнительных операционных издержек.

Данный документ описывает подходы к оптимальному планированию размещения блоков объектной системы хранения данных TATLIN.OBJECT, примеры конфигурации одной СХД с несколькими кластерами, плюсы и минусы применения различных политик хранения для управления размещением данных на узлах СХД, дает информацию по дополнительным разделам документации, например, сетевое планирование и сервисное обслуживание.

Отдельные разделы посвящены особенностям присвоения эндпоинтов для неймспейсов, рекомендациям по выбору балансировщика нагрузки и настройке для различных схем подключения. В приложении к документу также описаны команды для получения информации о состоянии СХД с помощью утилиты frostfs-cli.

Документ предназначен для архитекторов, ИТ-специалистов и администраторов, ответственных за проектирование, внедрение и поддержку объектной системы хранения данных TATLIN.OBJECT.

ВЕРСИИ ИСПОЛЬЗУЕМОГО ПО	
TATLIN.OBJECT	1.6, 1.7

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

ПЛАНИРОВАНИЕ РАЗМЕЩЕНИЯ И УСТАНОВКА СХД TATLIN.OBJECT

Атрибуты узла хранения

Сеть Хранения Данных (СХД) TATLIN.OBJECT состоит из набора узлов, которым при начальной настройке назначаются атрибуты. Атрибуты [1] применяются для логической группировки узлов:

АТРИБУТ	ОПИСАНИЕ	ПРИМЕР ЗНАЧЕНИЯ

`ClusterName`	Имя кластера (группы узлов, которым принадлежит узел)	`Cluster1`
		`Cluster2`
		`Cluster3`

`UN-LOCODE`	LOCODE (код города, в котором расположен узел)	`RU MOW`
		`RU LED`
		`RU GOJ`

`Node`	IP-адрес управления узла хранения	`10.64.75.123`
`Role`	Роль узла в обработке блокчейн-транзакций	`alphabet / storage`

Атрибут `ClusterName` позволяет группировать узлы по:

- территориальному признаку (разные серверные комнаты, ЦОД, разные города);
- доменам отказа (разные стойки, линии питания);
- аппаратным признакам (поколение СХД, дисковая конфигурация узла);
- другим значимым критериям (например, по сетевым сегментам).

Атрибут `UN-LOCODE` используется для идентификации города, в котором установлен узел.
Список кодов `LOCODE` можно посмотреть по ссылке: https://service.unece.org/trade/locode/ru.htm.
На основе значений `UN-LOCODE` автоматически создаются производные атрибуты (страна, континент и т. д.).

Атрибут `Role` определяет роль узла в системе:

- Алфавитный узел — участвует в подписании новых блоков блокчейна Neo (содержащих конфигурацию СХД).
- Неалфавитный узел — обеспечивает хранение и обработку данных, но не участвует в формировании блоков.

Все узлы СХД (вне зависимости от роли) используются для хранения и обработки пользовательских данных (объектов и метаданных).
Алфавитные узлы должны располагаться в рамках одной геоплощадки (один ЦОД).
Местоположение геоплощадки для алфавитных узлов выбирается при инициализации СХД и не может быть изменено в текущей версии СХД.

Атрибуты (включая `Role`) используются в политиках хранения для:

- Выбора узлов, на которых будут храниться копии объектов (векторы хранения).
- Определения количества копий для каждого вектора хранения.

Подробнее о политиках хранения — в конце главы.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Размещение узлов по площадкам. Планирование сетевого подключения

Сетевые настройки узла задаются при его инициализации. В версиях ТО 1.5–1.8 изменение сетевых настроек узла возможно только через обращение в сервис YADRO. Изменение IP-адресации сети Internal не поддерживается. Изменение остальных сетевых настроек разрешено.

Схемы сетевого подключения узлов СХД представлены в документах «Требования к площадке» и «Установка и обновление на узлах TATLIN.OBJECT».

Для работы СХД необходимо три типа сетевого подключения (трафика): Management, Data, Internal. Каждый тип подключения использует выделенные сетевые интерфейсы [2] и требует назначения IP-адресов в своем широковещательном домене. Не допускается использование одного сетевого сегмента для разных типов трафика.

В рамках одной геоплощадки для каждого типа сетевого подключения рекомендуется использовать сеть уровня L2 (IP-адреса узлов должны находиться в одной подсети). Допускается использование маршрутизируемой сети уровня L3 между узлами разных геоплощадок (кластеров).

Сеть Internal
Используется для работы СХД и взаимодействия между узлами СХД. Для работы СХД требуется сетевая связность между всеми ее узлами по сети Internal.

Сеть Management
Используется для администрирования СХД (веб-интерфейс, CAPI) и взаимодействия между ее узлами при обновлении на новую версию TATLIN.OBJECT. Через сеть Management СХД взаимодействует с инфраструктурными сервисами заказчика (DNS, NTP, AD, LDAP, Keycloak, VictoriaMetrics, rsyslog).

Сеть Data
Используется для работы с СХД по объектным протоколам (GRPC, S3, IAM). Для работы СХД необходима сетевая связность между клиентами данных и узлами СХД на TCP-портах 8080, 443.

Варианты размещения узлов СХД

При первичной инициализации узлы TATLIN.OBJECT могут:

- создать новую СХД;
- подключиться к существующей СХД (существующему либо новому кластеру).

Выбор фиксируется при установке и не может быть изменен без переустановки узлов с полной потерей данных.

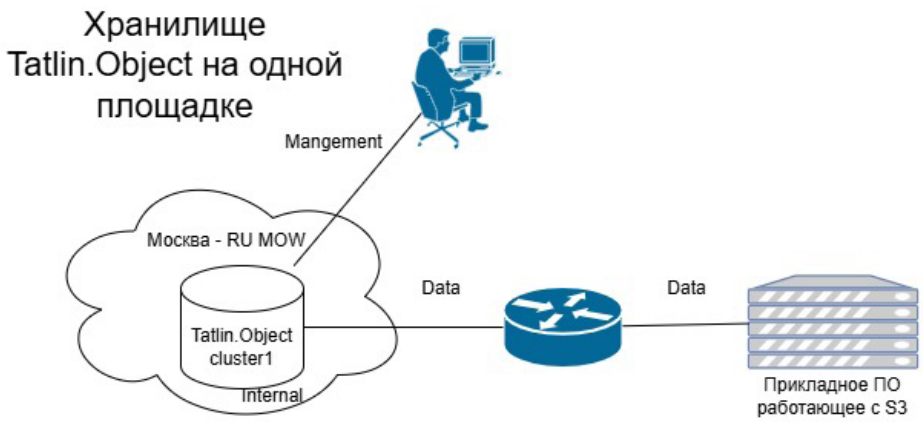
> ! Узлы из разных СХД не имеют возможности взаимодействовать друг с другом по сети Internal. Политика хранения применяется только в пределах одной СХД.

Далее представлены примеры конфигурации одной СХД с несколькими кластерами.

Локальная конфигурация: одна геоплощадка, один кластер, одна СХД

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Локальная конфигурация: одна геоплощадка, один кластер, одна СХД



Геоплощадка в Москве

```
UN-LOCODE=RU MOW
ClusterName=cluster1
Role = alphabet
```

Минимальная конфигурация включает четыре узла. Алфавитные узлы расположены на одной площадке.

Интерфейсы `Internal`:

- режим порта на свиче — `access port`;
- подключены в отдельный VLAN (уровень L2[3]) без фильтрации трафика по сети Internal;
- находятся в одной IP-подсети.

Интерфейсы `Data`:

- режим порта на свиче — `access port`;
- подключены в отдельный VLAN (уровень L2 [3]);
- находятся в одной IP-подсети.

Доступ к сети Data со стороны прикладного ПО обеспечивается через маршрутизируемую сеть (уровень L3).
Должны быть открыты порты: 443/TCP (HTTPS и S3), 8080/TCP (gRPC frostfs).

Интерфейсы `Management`:

- режим порта на свиче — `access port`;
- подключены в отдельном VLAN (уровень L2 [3]);
- находятся в одной IP-подсети.

Доступ к сети Management обеспечивается через маршрутизируемую сеть.
Список необходимых открытых TCP-портов указан в документе «Требования к площадке».

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Геораспределенная конфигурация: две геоплощадки, два кластера, одна СХД



Геоплощадка в Москве

```
UN-LOCODE=RU MOW
ClusterName=cluster1
Role = alphabet
```

Геоплощадка в Нижнем Новгороде

```
UN-LOCODE=RU GOJ
ClusterName=cluster2
Role = storage
```

Минимальная конфигурация включает четыре узла. Алфавитные узлы расположены на одной площадке (Москва).

Интерфейсы `Internal`, `Data`, `Management` на каждой из площадок подключаются по схеме локальной конфигурации: одна геоплощадка, один кластер, одна СХД. IP-адресация внутри площадки осуществляется на уровне L2 [3].

Интерфейсы узлов `Internal` могут взаимодействовать на уровне L2/L3.
Должна быть обеспечена полная сетевая связность между узлами на разных площадках.
Список необходимых открытых TCP-портов указан в документе «Требования к площадке».

Доступ к сети `Data` со стороны прикладного ПО обеспечивается через маршрутизируемую сеть (уровень L3).
Должны быть открыты порты: 443/TCP (HTTPS и S3), 8080/TCP (gRPC frostfs).

Если доступ к сети `Data` разрешен, клиенты работают напрямую с узлами всех площадок, используя сеть `Data`.
Если доступ к сети `Data` между площадками запрещен (как в примере выше), S3-запрос к узлу СХД, расположенному на геоплощадке в Москве, может быть перенаправлен по внутренней сети на узел, расположенный на геоплощадке в Нижнем Новгороде. После обработки результат будет передан обратно на узел, расположенный на геоплощадке в Москве, используя сеть `Internal`.

Доступ к сети `Management` обеспечивается через маршрутизируемую сеть.
Список необходимых открытых TCP-портов указан в документе «Требования к площадке».

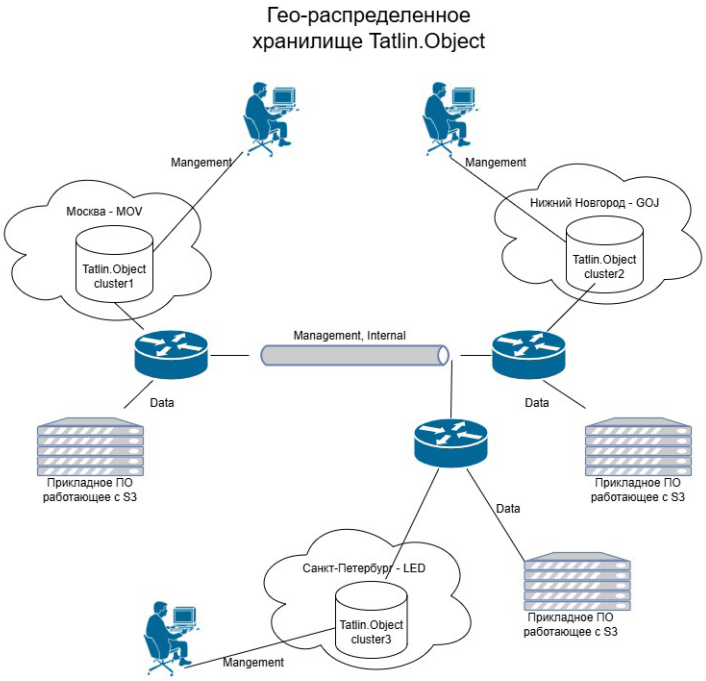
Обязательные требования к доступности:

- Все узлы должны иметь доступ к сетевым сервисам (кроме LDAP/AD).
- Управляющий терминал должен иметь подключение ко всем узлам всех площадок.

Рекомендуется открыть порт TCP/22 (SSH), чтобы разрешить удаленный доступ между площадками и упростить сервисное обслуживание.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Георазпределенная конфигурация: три геоплощадки, три кластера, одна СХД



Геоплощадка в Москве

```
UN-LOCODE=RU MOW
ClusterName=cluster1
Role = alphabet
```

Геоплощадка в Нижнем Новгороде

```
UN-LOCODE=RU GOJ
ClusterName=cluster2
Role = storage
```

Геоплощадка в Санкт-Петербурге

```
UN-LOCODE=RU LED
ClusterName=cluster3
Role = storage
```

Минимальная конфигурация включает четыре узла. Алфавитные узлы расположены на одной площадке (Москва).
Интерфейсы `Internal`, `Data`, `Management` на каждой из площадок подключаются по схеме локальной конфигурации: одна геоплощадка, один кластер, одна СХД. IP-адресация внутри площадки осуществляется на уровне L2[3].

Интерфейсы узлов `Internal` могут взаимодействовать на уровне L2/L3[4].
Должна быть обеспечена полная сетевая связность между узлами на разных площадках.
Список необходимых открытых TCP-портов указан в документе «Требования к площадке».

Доступ к сети `Data` со стороны прикладного ПО обеспечивается через маршрутизируемую сеть (уровень L3).
Должны быть открыты порты: 443/TCP (HTTPS и S3), 8080/TCP (gRPC frostfs).

Если доступ к сети `Data` разрешен, клиенты работают напрямую с узлами всех площадок, используя сеть `Data`.
Если доступ к сети `Data` между площадками запрещен (как в примере выше), S3-запрос к узлу СХД, расположенному на геоплощадке в Нижнем Новгороде, может быть перенаправлен по внутренней сети на узел, расположенный на геоплощадке в Москве или Санкт-Петербурге.
После обработки результат будет передан обратно на узел, расположенный на геоплощадке в Нижнем Новгороде, используя сеть `Internal`.

Доступ к сети `Management` обеспечивается через маршрутизируемую сеть.
Список необходимых открытых TCP-портов указан в документе «Требования к площадке».

Обязательные требования к доступности:

1. Все узлы должны иметь доступ к сетевым сервисам (кроме LDAP/AD).
2. Управляющий терминал должен иметь подключение ко всем узлам всех площадок.

Рекомендуется открыть порт TCP/22 (SSH), чтобы разрешить удаленный доступ между площадками и упростить сервисное обслуживание.

Масштабирование СХД на большее число площадок осуществляется аналогичным образом.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Планирование расширения СХД

Основные принципы:

- Масштабирование осуществляется добавлением узлов на существующие площадки.
- Минимальный шаг расширения — один узел.

Требование к новым узлам: IP-адресация в тех же L2-сегментах, что и существующие узлы площадки.
Для заказа дополнительного оборудования обратитесь в отдел продаж YADRO.

Примечания

[1] В версии 1.7.2 (лето 2025 года) запланировано обновление, которое добавит возможность создавать и изменять дополнительные пользовательские атрибуты узлов (например, номер стойки, домен отказа, инвентарный номер и другие).

[2] В минимальной схеме сетевого подключения data- и internal-трафик передаются через один физический интерфейс с использованием тегированных VLAN. Такая схема рекомендуется для демонстрационных и тестовых инсталляций. Для промышленной эксплуатации, в целях повышения производительности и отказоустойчивости, рекомендуется использовать схемы с физически выделенными интерфейсами.

[3] В рамках одной площадки передача одного типа трафика в разных VLAN (L3-сетях) технически возможна, но не является рекомендуемой практикой.

[4] Поддержка L3-маршрутизации между тремя и более площадками поддерживается с релиза 1.7.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

ПОЛИТИКИ ХРАНЕНИЯ

Размещение данных на узлах СХД управляется политиками хранения[1].
Политика хранения определяет:

- на каких узлах могут храниться копии объекта (вектора);
- сколько копий должно быть создано для каждого вектора.

Политика хранения задается при создании бакета/контейнера и не может быть изменена в дальнейшем[2].
Узлы, на которых могут размещаться копии объекта (вектора), определяются при создании бакета[3] и называются контейнерными узлами.

Политика хранения может содержать несколько векторов (например, для размещения копий в разных ЦОД).

Расчет размера вектора

> **!** Подробное описание синтаксиса политик хранения смотрите в документе «Руководство разработчика» (раздел «Синтаксис политик хранения»), доступном на сервисном портале.

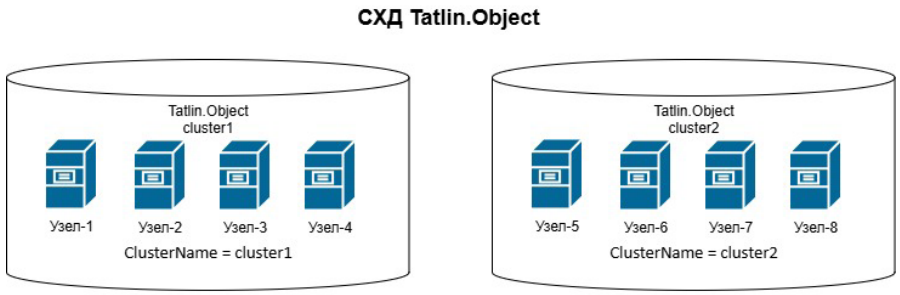
Размер вектора вычисляется по формуле:

CBF × **SELECT**, где:

- **SELECT** — минимальное количество узлов в векторе.
  Если в карте сети меньше узлов, запись в вектор блокируется.
- **CBF** (Capacity Balancing Factor) — множитель, задающий максимально возможное количество узлов в векторе (**SELECT** × **CBF**).
  Если в карте сети меньше узлов, чем **SELECT** × **CBF**, но больше, чем **SELECT**, используются все доступные узлы.
  По умолчанию **CBF** = 3.

Пример работы политик хранения

Рассмотрим СХД TATLIN.OBJECT из 8 узлов: 2 кластера по 4 узла в каждом.



Вектор описывается выражением:

```
CBF 8 SELECT 2 FROM *
```

Расчет размера вектора:

- Минимальный размер вектора: **SELECT** = 2 узла.
- Максимальный размер вектора: **SELECT** × **CBF** = 16 узлов.
- Поскольку правило FROM * применяется ко всем узлам, а в системе всего 8 узлов, эффективный размер вектора — от 8 узлов.

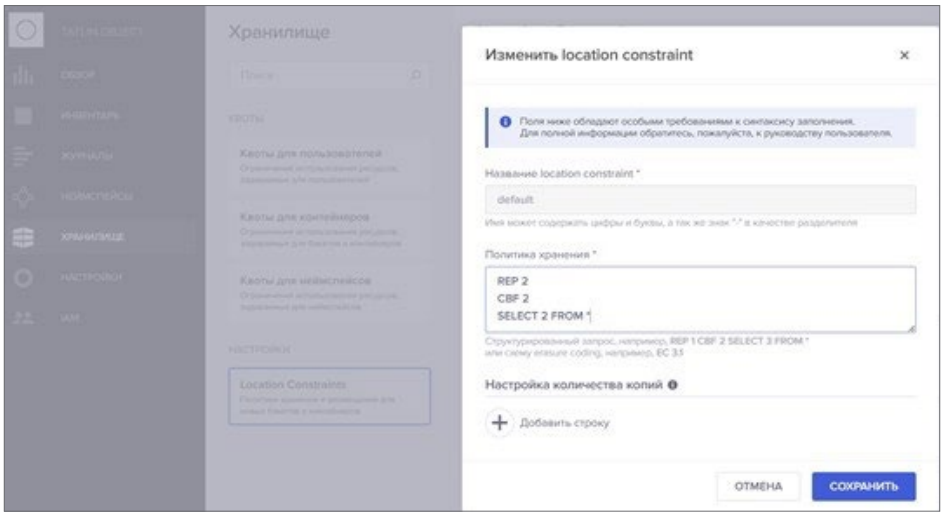
Если добавить 4 узла, размер вектора станет 12 узлов.
Если отключить 2 узла, размер вектора уменьшится до 6 узлов.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Политика хранения `default` (локальная репликация)

Политика `default`[4] описывается выражением:

```
REP 2 CBF2 SELECT 2 FROM *
```

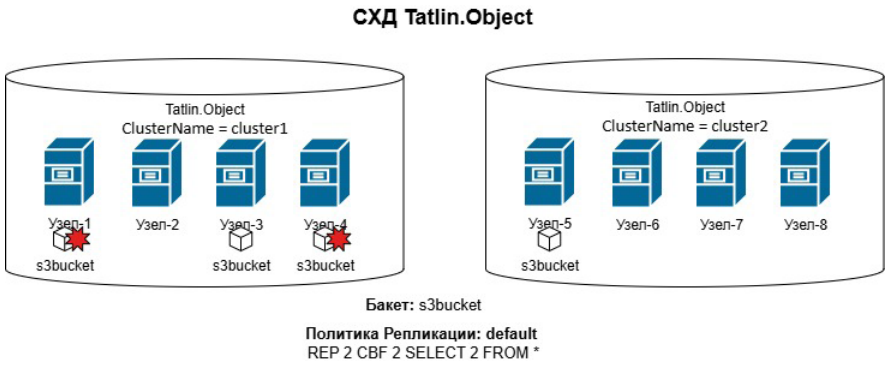


При создании бакета `s3bucket` с политикой `default`:

- Система выбирает 4 узла (по формуле **SELECT** × **CBF** = 2 × 2), на которых будут храниться копии объектов.
- Узлы выбираются произвольно, так как не заданы дополнительные правила фильтрации.

Пример

```
aws s3api --no-verify-ssl create-bucket --bucket s3bucket \
          --endpoint-url https://10.78.12.13 \
          --create-bucket-configuration LocationConstraint=default
```



Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

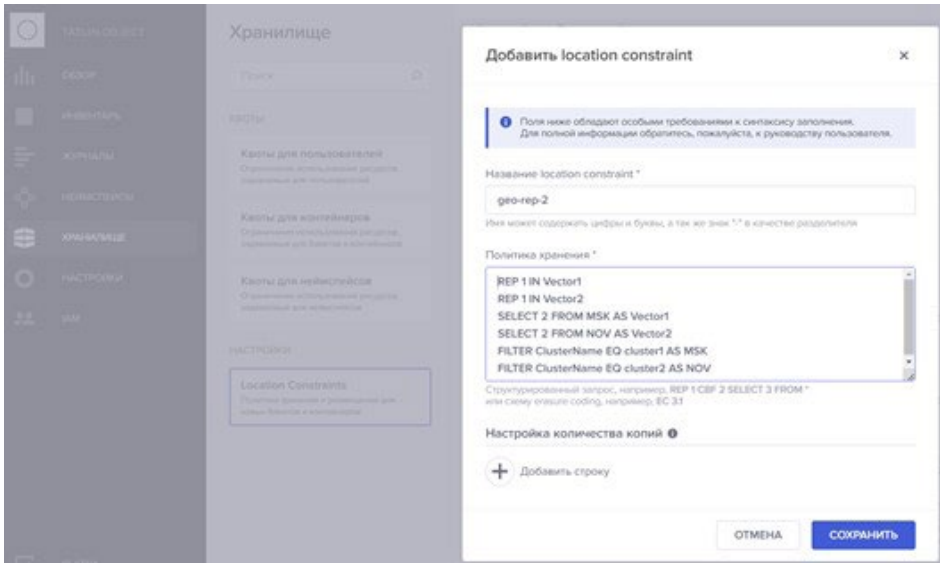
В данном случае файл `/tmp/s3object.txt` сохранен как `https://10.78.12.13/s3bucket/s3obj.txt` с репликами на узлах 1 и 4.
При следующей записи могут быть выбраны другие узлы (например, 3 и 8).
СХД автоматически распределяет данные между контейнерными узлами, поддерживая баланс их заполненности.

Преимущества и недостатки политики хранения `default`

- Преимущество — **простота и универсальность**.
  Политика хорошо работает на СХД небольшого размера без георазпределения.
- Недостаток — **отсутствие учета доменов отказа**.
  Обе копии объекта (REP 2) могут попасть в один домен отказа.
  При отказе REP-узлов часть данных станет недоступна для чтения (даже если размер вектора = 90, а REP = 2).
  Запись остается работоспособной, если доступно ≥ **SELECT** узлов.
- Недостаток — **ограничение емкости бакета**.
  Если все контейнерные узлы заполнены на 100 %, запись в бакет остановится, даже если на других узлах есть свободное место.

Политика хранения geo-rep-2 (синхронная межкластерная репликация - две площадки)
Политика `geo-rep-2` описывается выражением:

```
REP 1 IN Vector1
REP 1 IN Vector2
SELECT 2 FROM MSK AS Vector1
SELECT 2 FROM NOV AS Vector2
FILTER ClusterName EQ cluster1 AS MSK
FILTER ClusterName EQ cluster2 AS NOV
```



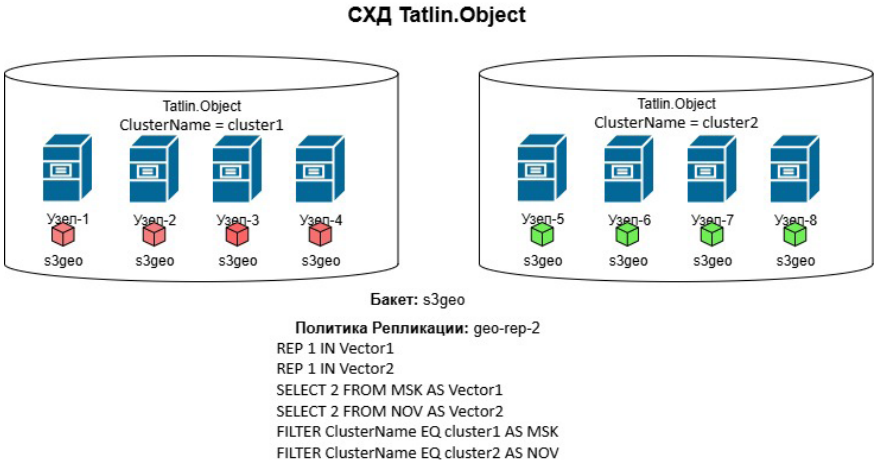
При создании бакета `s3bucket` с политикой `geo-rep-2`:

- Система формирует два вектора хранения (Vector1 и Vector2) по одному на каждый кластер.
- Для каждого вектора выбирается от 2 до 6 узлов (**SELECT** × **CBF**, где **CBF** по умолчанию = 3). Фактический размер каждого вектора ограничен 4 узлами (количество узлов в кластере).

Пример

```
aws s3api --no-verify-ssl create-bucket --bucket s3geo \
          --endpoint-url https://10.78.12.13 \
          --create-bucket-configuration LocationConstraint=geo-rep-2
```

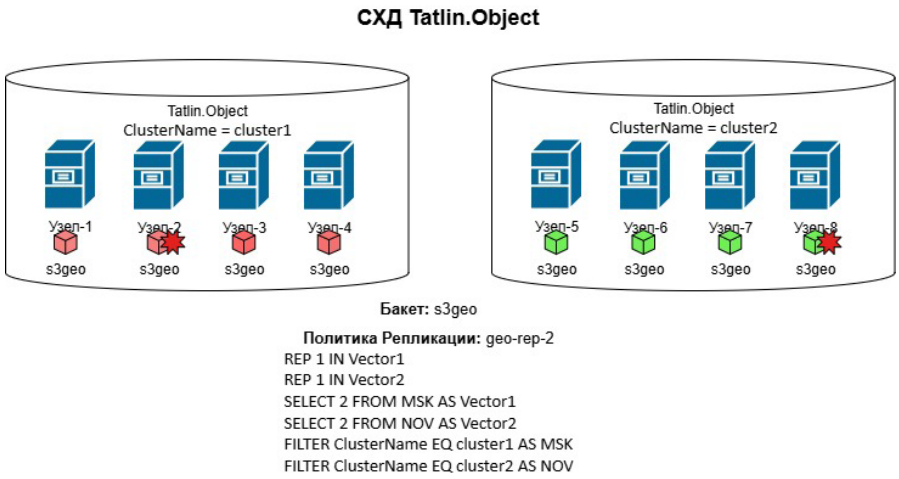

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11



При каждой записи нового объекта система создает по одной копии объекта в каждом векторе (Vector1 и Vector2).

Пример

```
aws s3api --no-verify-ssl put-object --bucket s3geo \
          --key s3obj.txt --body /tmp/s3object.txt \
          --endpoint-url https://10.78.12.13
```



В данном случае файл `/tmp/s3object.txt` сохранен как `https://10.78.12.13/s3geo/s3obj.txt` с репликами на узлах 2 (Vector1) и 8 (Vector2).
При следующей записи могут быть выбраны другие узлы из тех же векторов.
СХД автоматически распределяет данные между контейнерными узлами, поддерживая баланс их заполненности.

Преимущества и недостатки политики хранения `geo-rep-2`

- Преимущество — **отказоустойчивость**.
  Копии распределены по разным кластерам.
  Поддержка асинхронной репликации (опционально).
  Сохранность данных гарантирована при отказе целой площадки и одновременном отказе одного узла на резервной площадке.
- Недостаток — **ограничения конфигурации**.
  Поддерживаются только существующие кластеры.
  Невозможно добавить новые кластеры в политику.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

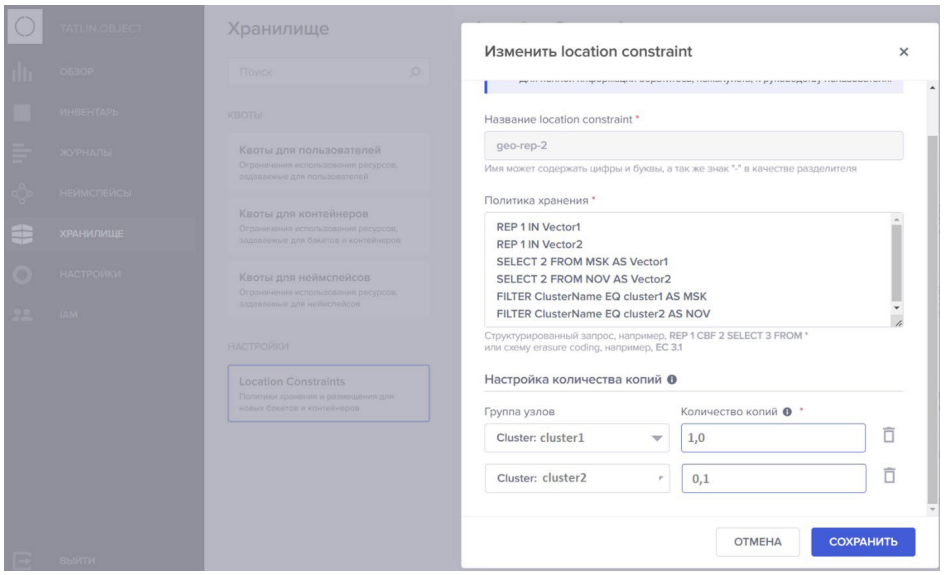
- Недостаток — **доступность записи**.
  Запись блокируется при отказе одной площадки. Решение: активация асинхронной репликации.
- Недостаток — **ограничение емкости**.
  Запись прекращается при заполнении любого вектора. Неиспользуемое пространство в других кластерах остается недоступным.
- Недостаток — **технологические ограничения**.
  Несовместимость с Erasure Coding (EC, избыточное кодирование).

Политика хранения async-rep-2 (асинхронная межкластерная репликация - две площадки)

Настройка количества копий для асинхронной репликации задается отдельными группами для каждого кластера, где указывается количество синхронных (1) и асинхронных (0) копий для каждого вектора.

Политика `async-rep-2` описывается выражением:

```
REP 1 IN Vector1
REP 1 IN Vector2
SELECT 2 FROM MSK AS Vector1
SELECT 2 FROM NOV AS Vector2
FILTER ClusterName EQ cluster1 AS MSK
FILTER ClusterName EQ cluster2 AS NOV
```



Количество синхронных копий:
- Задается параметром REP для каждого вектора.
- В примере: по одной синхронной копии на кластер.

Механизм репликации при записи на любой узел:
- `cluster1` синхронно (первое правило REP) — сохраняет одну копию объекта в Vector1 (локальный кластер).
- `cluster1` асинхронно (второе правило REP) — реплицирует одну копию объекта в Vector2 (фоновый процесс).
- `cluster2` синхронно (первое правило REP) — сохраняет одну копию объекта в Vector2 (локальный кластер).
- `cluster2` асинхронно (второе правило REP) — реплицирует одну копию объекта в Vector1 (фоновый процесс).

Особенности:
- СХД автоматически поддерживает требуемое количество копий.
- Настройки REP действуют независимо для каждого кластера.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Преимущества и недостатки политики хранения `async-rep-2`

- Преимущество — **отказоустойчивость**.
  Настройки REP действуют независимо для каждого кластера. СХД автоматически поддерживает требуемое количество копий.
  Поддержка асинхронной репликации (опционально). Выполняется фоновым процессом.
  Сохранность данных гарантирована при отказе целой площадки и одновременном отказе одного узла на резервной площадке.
- Преимущество — **производительность**.
  Ускорение операций записи по сравнению с синхронной репликацией.
  Снижение зависимости от скорости межкластерного канала.
- Недостаток — **ограничения конфигурации**.
  Поддерживаются только существующие кластеры.
  Невозможно добавить новые кластеры в политику.
- Недостаток — **ограничение емкости**.
  Запись прекращается при заполнении любого вектора. Неиспользуемое пространство в других кластерах остается недоступным.
- Недостаток — **технологические ограничения**.
  Несовместимость с Erasure Coding.
  Время завершения репликации не гарантировано (от часов до дней).

Политика хранения для трех и более площадок

Политики хранения для трех и более площадок создаются с помощью добавления векторов.
Настройка числа копий для асинхронной репликации будет содержать группы для каждого кластера и перечисление числа копий для каждого из векторов.

Политика `geo-rep-3` (для трех площадок) описывается выражением:

```
REP 1 IN Vector1
REP 1 IN Vector2
REP 1 IN Vector3
SELECT 2 FROM MSK AS Vector1
SELECT 2 FROM NOV AS Vector2
SELECT 2 FROM SPB AS Vector3
FILTER ClusterName EQ cluster1 AS MSK
FILTER ClusterName EQ cluster2 AS NOV
FILTER ClusterName EQ cluster3 AS SPB
```

Настройка количества копий:

```
Группы узлов              Количество копий
Cluster: cluster1         1,0,0
Cluster: cluster2         0,1,0
Cluster: cluster3         0,0,1
```

Политика `geo-rep-4` (для четырех площадок) описывается выражением:

```
REP 1 IN Vector1
REP 1 IN Vector2
REP 1 IN Vector3
REP 1 IN Vector4
SELECT 2 FROM MSK AS Vector1
SELECT 2 FROM NOV AS Vector2
SELECT 2 FROM SPB AS Vector3
SELECT 2 FROM DUB AS Vector4
FILTER ClusterName EQ cluster1 AS MSK
FILTER ClusterName EQ cluster2 AS NOV
FILTER ClusterName EQ cluster3 AS SPB
FILTER ClusterName EQ cluster4 AS DUB
```


Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Настройка количества копий:

```
Группы узлов                Количество копий
Cluster: cluster1           1,0,0,0
Cluster: cluster2           0,1,0,0
Cluster: cluster3           0,0,1,0
Cluster: cluster4           0,0,0,1
```

Политика хранения для СХД с большим числом узлов на одной площадке

Политика хранения позволяет достичь целей:

1. Обеспечить достаточный размер вектора (максимальный требуемый размер бакета).
2. Оптимально распределить нагрузку между узлами СХД при интенсивной записи (в рамках вектора).
3. Разместить данные в разных доменах отказа (гарантировать доступность данных при одновременном отказе всех узлов одного домена).

Размер бакета оценивается в тебибайтах (ТиБ) или в количестве объектов заданного размера. Эта оценка зависит от уровня избыточности, определяемого политикой хранения (**REP** или **EC**).

Цель 1 — Оценка размера вектора (защита на основе фактора REP)

ПРИМЕР РАСЧЕТА ЕМКОСТИ («СЫРАЯ» ЕМКОСТЬ УЗЛА: 168 ТиБ)

Число узлов в кластере СХД	Параметры, задающие число узлов в векторе		Число узлов в векторе	Максимально возможный объем вектора (бакета) в зависимости от фактора репликации (в тебибайтах)				
	SELECT	CBF	SELECT × CBF	REP 2	REP 3	REP 4	REP 5	REP 6
16	2	3	6	504	336	252	202	168
32	4	3	12	1 008	672	504	403	336
48	5	3	15	1 260	840	630	504	420
64	6	3	18	1 512	1 008	756	605	504
80	7	3	21	1 764	1 176	882	706	588
96	8	3	24	2 016	1 344	1 008	806	672
100	10	3	30	2 520	1 680	1 260	1 008	840

ПРИМЕР РАСЧЕТА ЕМКОСТИ ДЛЯ ОБЪЕКТОВ РАЗМЕРОМ 1 МЕБИБАЙТ (МиБ)[5]

Число узлов в кластере СХД	Параметры, задающие число узлов в векторе		Число узлов в векторе	Максимально возможное число объектов размером 1 Миб в векторе (бакете) в зависимости от фактора репликации				
	SELECT	CBF	SELECT × CBF	REP 2	REP 3	REP 4	REP 5	REP 6
16	2	3	6	504 000 000	336 000 000	252 000 000	201 600 000	168 000 000
32	4	3	12	1 008 000 000	672 000 000	504 000 000	403 200 000	336 000 000
48	5	3	15	1 260 000 000	840 000 000	630 000 000	504 000 000	420 000 000
64	6	3	18	1 512 000 000	1 008 000 000	756 000 000	604 800 000	504 000 000
80	7	3	21	1 764 000 000	1 176 000 000	882 000 000	705 600 000	588 000 000
96	8	3	24	2 016 000 000	1 344 000 000	1 008 000 000	806 400 000	672 000 000
100	10	3	30	2 520 000 000	1 680 000 000	1 260 000 000	1 008 000 000	840 000 000

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

ПРИМЕР РАСЧЕТА ЕМКОСТИ ДЛЯ ОБЪЕКТОВ РАЗМЕРОМ 100 МиБ[5]								
Число узлов в кластере СХД	Параметры, задающие число узлов в векторе		Число узлов в векторе	Максимально возможное число объектов размером 100 Миб в векторе (бакете) в зависимости от фактора репликации				
	SELECT	CBF	SELECT × CBF	REP 2	REP 3	REP 4	REP 5	REP 6
16	2	3	6	5 040 000	3 360 000	2 520 000	2 016 000	1 680 000
32	4	3	12	10 080 000	6 720 000	5 040 000	4 032 000	3 360 000
48	5	3	15	12 600 000	8 400 000	6 300 000	5 040 000	4 200 000
64	6	3	18	15 120 000	10 080 000	7 560 000	6 048 000	5 040 000
80	7	3	21	17 640 000	11 760 000	8 820 000	7 056 000	5 880 000
96	8	3	24	20 160 000	13 440 000	10 080 000	8 064 000	6 720 000
100	10	3	30	25 200 000	16 800 000	12 600 000	10 080 000	8 400 000

Рекомендуется включать в вектор приблизительно треть узлов кластера, чтобы минимизировать перекрытие узлов при записи в разные бакеты[6]. Если требуется включить в вектор более трети узлов, увеличить параметр **CBF**.

!

Увеличение размера вектора может быть необходимо в случаях:

- необходимость повысить производительность за счет распараллеливания записи на большее число узлов;
- хранение объектов большого размера;
- наличие нескольких крупных бакетов, использующих одни и те же узлы СХД.

ПРИМЕР РАСЧЕТА ЕМКОСТИ ДЛЯ ОБЪЕКТОВ РАЗМЕРОМ 10 ГИБИБАЙТ (ГиБ)								
Число узлов в кластере СХД	Параметры, задающие число узлов в векторе		Число узлов в векторе	Максимально возможное число объектов размером 10 Гиб в векторе (бакете) в зависимости от фактора репликации				
	SELECT	CBF	SELECT × CBF	REP 2	REP 3	REP 4	REP 5	REP 6
16	2	3	6	50 400	33 600	25 200	20 160	16 800
32	4	3	12	100 800	67 200	50 400	40 320	33 600
48	5	3	15	126 000	84 000	63 000	50 400	42 000
64	6	3	18	151 200	100 800	75 600	60 480	50 400
80	7	3	21	176 400	117 600	88 200	70 560	58 800
96	8	3	24	201 600	134 400	100 800	80 640	67 200
100	10	3	30	252 000	168 000	126 000	100 800	84 000

Цель 1 — Оценка размера вектора (защита на основе EC)

Политика EC задает количество разделенных и избыточных частей объекта, которые можно скомбинировать в изначальный объект.
Структура запроса команды EC:

```
EC X.Y
```

X — количество частей разделенного объекта с данными. Максимальное допустимое количество частей — 12.
Y — количество избыточных частей. Максимальное допустимое количество частей — 4.
X + **Y** — общее количество разделенных и избыточных частей. Размер вектора (**SELECT** × **CBF**) может превышать это количество.
Все части **X** и **Y** разделенного объекта имеют один размер, примерно в **X** раз меньше исходного объекта.
Сумма **X**+**Y** в политике хранения должна быть меньше размера вектора (**CBF** × **SELECT**). Иначе отказ любого узла приведет к остановке записи в СХД.

Увеличение значений **X** и **Y** может быть необходимо в случаях:

- необходимость повысить производительность за счет распараллеливания записи на большее число узлов;
- хранение объектов большого размера;
- наличие нескольких крупных бакетов, использующих одни и те же узлы СХД.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Ограничения политики EC:

- Может содержать только один вектор.
- Георепликация невозможна.
- Распределение копий по доменам отказа также не поддерживается.

Рекомендации:

- Для приложений, чувствительных к операциям чтения, листинга и поиска, размер вектора не должен превышать 8–9 узлов.
- Политики **EC X.Y** целесообразны в крупных СХД, где важно снизить стоимость хранения без ущерба для надежности.
- **EC** особенно рекомендуется для архивного хранения (холодные данные с низкими требованиями к скорости чтения).

Примеры:

ПРИМЕР РАСЧЕТА ЕМКОСТИ («СЫРАЯ» ЕМКОСТЬ УЗЛА: 168 ТиБ)

Число узлов в кластере СХД	Параметры, задающие число узлов в векторе		Число узлов в векторе	Максимально возможный объем вектора (бакета) в зависимости от параметров EC (в Тебибайтах)			
	SELECT	CBF	SELECT × CBF	EC 3.1	EC 6.2	EC 9.3	EC 12.4
16	2	3	6	756			
32	4	3	12	1 512	1 512	1 512	1 512
48	5	3	15	1 890	1 890	1 890	1 890
64	6	3	18	2 268	2 268	2 268	2 268
80	7	3	21	2 646	2 646	2 646	2 646
96	8	3	24	3 024	3 024	3 024	3 024
100	10	3	30	3 780	3 780	3 780	3 780

ПРИМЕР РАСЧЕТА ЕМКОСТИ ДЛЯ ОБЪЕКТОВ РАЗМЕРОМ 1 МБ

Число узлов в кластере СХД	Параметры, задающие число узлов в векторе		Число узлов в векторе	Максимально возможное число объектов размером 1 Миб в векторе (бакете) в зависимости от параметров EC			
	SELECT	CBF	SELECT × CBF	EC 3.1	EC 6.2	EC 9.3	EC 12.4
16	2	3	6	756 000 000			
32	4	3	12	1 512 000 000	1 512 000 000		
48	5	3	15	1 890 000 000	1 890 000 000	1 890 000 000	
64	6	3	18	2 268 000 000	2 268 000 000	2 268 000 000	2 268 000 000
80	7	3	21	2 646 000 000	2 646 000 000	2 646 000 000	2 646 000 000
96	8	3	24	3 024 000 000	3 024 000 000	3 024 000 000	3 024 000 000
100	10	3	30	3 780 000 000	3 780 000 000	3 780 000 000	3 780 000 000

ПРИМЕР РАСЧЕТА ЕМКОСТИ ДЛЯ ОБЪЕКТОВ РАЗМЕРОМ 100 МБ

Число узлов в кластере СХД	Параметры, задающие число узлов в векторе		Число узлов в векторе	Максимально возможное число объектов размером 100 Миб в векторе (бакете) в зависимости от параметров EC			
	SELECT	CBF	SELECT × CBF	EC 3.1	EC 6.2	EC 9.3	EC 12.4
16	2	3	6	7 560 000			
32	4	3	12	15 120 000	15 120 000		
48	5	3	15	18 900 000	18 900 000	18 900 000	
64	6	3	18	22 680 000	22 680 000	22 680 000	22 680 000
80	7	3	21	26 460 000	26 460 000	26 460 000	26 460 000
96	8	3	24	30 240 000	30 240 000	30 240 000	30 240 000
100	10	3	30	37 800 000	37 800 000	37 800 000	37 800 000

: 2026-11-25

Версия: 1.12:0:0

! Рекомендуется включать в вектор приблизительно половину узлов кластера, чтобы минимизировать перекрытие узлов при записи в разные бакеты [6].

ПРИМЕР РАСЧЕТА ЕМКОСТИ ДЛЯ ОБЪЕКТОВ РАЗМЕРОМ 10 ГиБ

Число узлов в кластере СХД	Параметры, задающие число узлов в векторе		Число узлов в векторе	Максимально возможное число объектов размером 10 Гиб в векторе (бакете) в зависимости от параметров EC			
	SELECT	CBF	SELECT × CBF	EC 3.1	EC 6.2	EC 9.3	EC 12.4
16	2	3	6	75 600			
32	4	3	12	151 200	151 200		
48	5	3	15	189 000	189 000	189 000	
64	6	3	18	226 800	226 800	226 800	226 800
80	7	3	21	264 600	264 600	264 600	264 600
96	8	3	24	302 400	302 400	302 400	302 400
100	10	3	30	378 000	378 000	378 000	378 000

Цель 2 — Оптимальное распределение нагрузки между узлами СХД

Система равномерно распределяет копии объектов по всем контейнерным узлам при записи в бакет.
Репликатор выбирает наиболее свободный шард узла для размещения копии объекта.
Если один узел хранит данные нескольких бакетов, репликатор распределяет копии по разным шардам, что позволяет узлу эффективно обслуживать несколько высоконагруженных бакетов одновременно.

Рекомендации по настройке :

- Для интенсивной записи — включите в вектор достаточное количество узлов, чтобы обеспечить параллельную запись с учетом фактора репликации. При добавлении узлов в вектор производительность записи масштабируется линейно.
- Для операций чтения и поиска — ограничьте размер вектора 8-9 узлами. Большее количество ухудшит производительность листинга и поиска. Избегайте необоснованного увеличения вектора "на всякий случай".

Цель 3 — Распределение данных по доменам отказа

Требования к организации доменов отказа:

- Разделите СХД на независимые кластеры в рамках одной площадки[7].
- Используйте физическое разделение: разные стойки, машинные залы.
- Обеспечьте независимые линии питания и сетевые коммутаторы для каждого домена.
- Для СХД в составе от 32 узлов создавайте минимум 2 домена отказа.
- Для критичных систем увеличивайте количество доменов отказа.

Пример политики хранения согласно разделу «Политика хранения geo-rep-2 (синхронная межкластерная репликация)».
Число копий в каждом кластере СХД можно увеличить в соответствии с требованиями по защите данных.
За размер вектора можно взять рекомендации из раздела «Оценка размера вектора».

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Примечания:

[1] Настройка политики хранения через S3 API
Задавайте политику хранения через параметр LocationConstraint.
Пример создания бакета с политикой rep3:

```
aws s3api create-bucket \
  --create-bucket-configuration 'LocationConstraint=rep3'
```

В веб-интерфейсе политики отображаются как именованные Location Constraints.

[2] Планы развития
В релизе ТО 1.8 (конец 2025) запланирована реализация возможности редактирования политик хранения для существующих бакетов. Это позволит:

- Изменять состав вектора хранения (масштабирование бакетов).
- Изменять параметры гео-репликации (удаление / добавление площадок гео репликации).

Текущие ограничения:

- Невозможно изменить политику хранения для существующих бакетов.

[3] Условия изменения вектора хранения
Состав узлов в векторе изменяется только при изменении карты сети хранения:

- Добавлении новых узлов в кластер.
- Выводе узлов из эксплуатации.
- Выключении либо сетевой изоляции узлов.

Автоматический пересчет происходит при изменении топологии.

[4] Управление политиками
Подробное описание политик хранения смотрите в документе «Руководство администратора» (раздел «Управление политиками хранения и репликациями в S3»), доступном на сервисном портале.

[5] Лимиты TATLIN.OBJECT версии 1.6, 1.7
Максимальное количество объектов:

- На бакет: 100 000 000
- На СХД: 1 000 000 000

Актуальные лимиты уточняйте в документе "Уведомление об обновлении", доступном на сервисном портале.

[6] Особые бизнес-кейсы
Для специализированных задач допускаются нестандартные конфигурации, требующие индивидуального расчета нагрузки. Пример: приложение, работающее с объектами большого размера - в вектор входят все узлы.

Пример для 16 узлов:

```
REP 6 CBF 2 SELECT 8 FROM *
```

Пример для 9 узлов:

```
EC 6.2 CBF 3 SELECT 3 FROM *
```


Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

ЭНДПОИНТЫ (БАЗОВЫЕ URL), НЕЙМСПЕЙСЫ, TLS-СЕРТИФИКАТЫ

Эндпоинт (или базовый URL) — полное доменное имя (FQDN), по которому клиенты обращаются к сервисам узла TATLIN.OBJECT.

ТИПЫ ЭНДПОИНТОВ	
`https://s3.mydomain.com`	Доступ к сервису S3
`https://iam.mydomain.com`	Доступ к сервису IAM
`https://web.mydomain.com`	Доступ к сервису HTTP-gateway
`https://capi.nodename.mydomain.com`	Доступ к сервису CAPI (управление СХД)

Механизм работы:

1. **Получение HTTP-запроса:**
   Узел принимает входящий HTTPS-запрос. Устанавливается безопасное соединение и проверяется валидность TLS-сертификата.
2. **Анализ заголовка Host**:
   Узел извлекает значение поля `Host` в заголовке. Значение является ключевым для дальнейшей маршрутизации запроса.
3. **Сопоставление с эндпоинтами**:
   Узел проводит проверку соответствия значения `Host` с настроенными эндпоинтами СХД. Значение должно соответствовать одному из зарегистрированных в системе FQDN.
4. **Маршрутизация запроса**:
   На основании результатов сопоставления запрос направляется в соответствующий сервисный модуль: `s3.*`, `http.*`, `iam.*`, `capi.*`

Пример:

```
HTTP-запрос:   HTTPS GET /documents/report.pdf
Заголовок:     Host: s3.mydomain.com
Сопоставление: s3.* → Сервис S3
Сохранение:    /documents/report.pdf
```

Настройка эндпоинтов

Корневой неймспейс: настраивается при первоначальной настройке СХД[1].
Пользовательские неймспейсы: задаются при создании неймспейса.
Начиная с версии 1.7 пользователь может в разделе "Неймспейсы" в WEBUI отредактировать (удалить, изменить, задать дополнительные) эндпоинты любых неймспейсов.

TLS-сертификаты
По умолчанию СХД генерирует самоподписанные сертификаты для каждого неймспейса[2], [3].

Порядок замены самоподписанного сертификата на внешний:

1. Получите список DNS-записей из текущего сертификата:

```
openssl s_client -connect <endpoint>:443 </dev/null 2>/dev/null \
| openssl x509 -noout -text | grep DNS: | sed "s/DNS:\|IP Address://
DNS: iivanov-nodel.object.set.yadro.com
DNS: web.s3.iivanov.object.set.yadro.com
DNS: web.s3.rosa.iivanov.object.set.yadro.com
DNS: d0.nodel_web.s3.iivanov.object.set.yadro.com
DNS: nodel.iam.rosa.iivanov.object.set.yadro.com
```

2. Включите все FQDN из оригинального сертификата в секцию `ALT_NAME` создаваемого внешнего сертификата.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

СТАНДАРТНЫЕ ЭНДПОИНТЫ ПОЛЬЗОВАТЕЛЬСКИХ НЕЙМСПЕЙСОВ	
`iam.<NAMESPACE_NAME>.<DOMAIN_NAME>`	Сервис IAM
`s3.<NAMESPACE_NAME>.<DOMAIN_NAME>`	Сервис S3
`web.<NAMESPACE_NAME>.<DOMAIN_NAME>`[5]	Сервис WEB (альтернативный доступ к объектам записанным по gRPC)

<NAMESPACE_NAME> — Имя неймспейса. Например, `ns1`.
<DOMAIN_NAME> — Полное доменное имя без короткого имени узла (задается при установке СХД).
Для узла `node1.mydomain.com` это `mydomain.com`.

Пример эндпоинтов для узла `node1.mydomain.com` и неймспейса `ns1`:
`https://iam.ns1.mydomain.com`, `https://s3.ns1.mydomain.com`, `https://web.ns1.mydomain.com`

Пользовательские домены
При редактировании эндпоинтов неймспейса можно задать любое уникальное доменное имя (например, `mycloud.com`).
При этом рекомендуется использовать префиксы обозначающие протокол:

ПОЛЬЗОВАТЕЛЬСКИЕ ДОМЕНЫ	
`iam.<CUSTOM_DOMAIN_NAME>`	Сервис IAM
`s3.<CUSTOM_DOMAIN_NAME>`	Сервис S3
`web.<CUSTOM_DOMAIN_NAME>`	Сервис WEB (альтернативный доступ к объектам записанным по gRPC)

Пример эндпоинтов для домена `mycloud.com`:
`https://iam.mycloud.com`, `https://s3.mycloud.com`, `https://web.mycloud.com`

При добавлении эндпоинта `s3.mycloud.com` под капотом в настройках СХД добавляется поддержка эндпоинта `*.s3.mycloud.com`.
Это необходимо для поддержки FQDN-based адресации бакетов, а так же GUBN[4].

> **!** Эндпоинты неймспейсов должны быть уникальными. Задание идентичных эндпоинтов разным неймспейсам - это ошибка конфигурации.

Примечания:

[1] Эндпоинты корневого неймспейса настраиваются сервисной службой YADRO при первоначальной настройке СХД.

[2] После изменения эндпоинтов неймспейса самоподписанный сертификат пересоздается автоматически.

[3] Самоподписанные сертификаты автоматически обновляются ежегодно. Корневой сертификат CA раз в 10 лет.

[4] С версии 1.7 добавлена поддержка GUBN (Global Unique Bucket Naming, Управление глобально-уникальными именами бакетов).
GUBN обеспечивает возможность доступа по URL вида:
`https://<BUCKET_NAME>.<GLOBAL_S3_ENDPOINT>`
<BUCKET_NAME> — Имя целевого бакета. Например, `my-bucket-1`.
<GLOBAL_S3_ENDPOINT> — Глобальный S3-эндпоинт, общий для нескольких неймейсов. Например, `storage.example.com`.
Пример: `https://my-bucket-1.storage.example.com`

[5] С версии 1.7 формат WEB-эндпоинтов изменился с `web.s3.<DOMAIN_NAME>` на `web.<DOMAIN_NAME>`.

НАСТРОЙКА БАЛАНСИРОВКИ НАГРУЗКИ

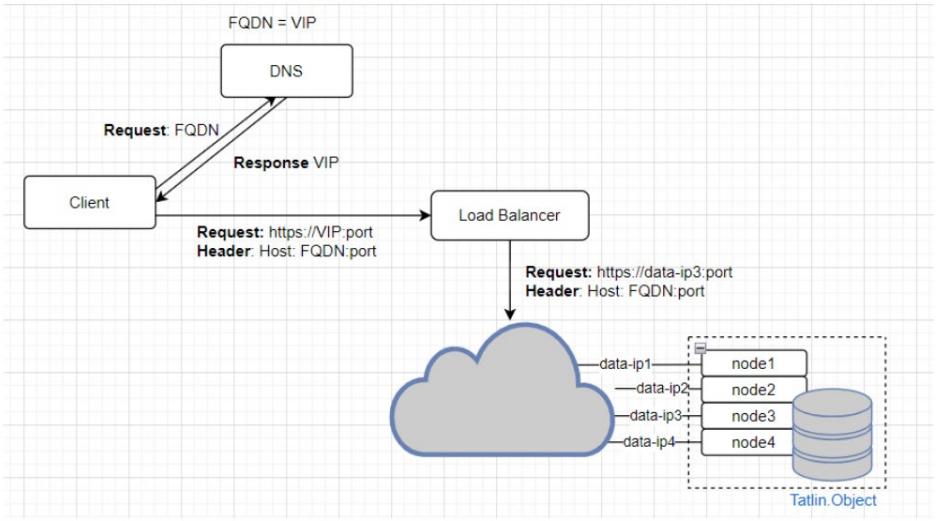
> **!**
>
> TATLIN.OBJECT не включает балансировщики нагрузки и инструменты управления кластерным VIP.
> Представленные схемы — возможные варианты реализации (не входят в комплект поставки СХД).
>
> Подробные рекомендации по настройке внешнего балансировщика нагрузки[1] смотрите в документе «Руководство администратора» (раздел «Рекомендации по настройке внешнего балансировщика нагрузки»), доступном на сервисном портале.

Балансировщик нагрузки:

- Перенаправляет пользовательские запросы на узлы СХД по выбранному алгоритму.
- Проверяет состояние узлов (health checks).
- Исключает узел из списка доступных для распределения запросов, если узел перестает отвечать.
- Распределяет трафик между работоспособными узлами.

Схема работы первого типа (виртуальный IP)

Схема работы внешнего балансировщика:



1. **Настройка DNS**
   В DNS назначается соответствие виртуального IP (VIP) балансировщика нагрузки для доменного имени data-эндпоинта (конечная точка данных).
   DNS-сервер сохраняет соответствие между FQDN и VIP.
   Пример дата-эндпоинта: `s3.mydomain.com`
   Пример VIP: `192.168.1.100`
2. **Отправка запроса клиентом**
   Клиент отправляет запрос, используя доменное имя data-эндпоинта.
   Пример запроса:

```
GET https://s3.mydomain.com/data/file.txt
```

3. **Преобразование DNS**
   DNS-сервер преобразует имя data-эндпоинта в VIP балансировщика и возвращает клиенту.
   Пример преобразования: `s3.mydomain.com` — `192.168.1.100`
4. **Обработка запроса балансировщиком**
   Балансировщик принимает запрос на VIP.
   Балансировщик выбирает доступный узел СХД по алгоритму балансировки.
   Балансировщик перенаправляет запрос на реальный IP-адрес выбранного узла.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

5. **Обработка запроса узлом СХД**
   Узел СХД получает запрос и выполняет необходимые операции (чтение/запись данных).
   Узел СХД формирует ответ, содержащий запрошенные данные или результат операции.
6. **Возврат ответа клиенту**
   Вариант 1: Напрямую к клиенту — узел СХД отправляет ответ клиенту напрямую, минуя балансировщик.
   Вариант 2: Через балансировщик — узел СХД отправляет ответ на балансировщик, затем балансировщик перенаправляет ответ клиенту.

Примеры балансировщиков нагрузки: HAProxy, NGINX, F5 BIG-IP (L4/L7).

Схема работы второго типа (DNS-балансировщик)

1. **Настройка DNS**
   В конфигурации DNS-сервера балансировщику делегируется управление NS зоной в которой расположены доменные записи data-эндпоинтов.
   DNS-сервер перенаправляет запросы для этой зоны на DNS-балансировщик.
2. **Отправка запроса клиентом**
   Клиент отправляет запрос, используя доменное имя data-эндпоинта.
   Пример запроса:

```
GET https://s3.mydomain.com/data/file.txt
```

3. **Преобразование DNS**
   DNS-балансировщик проверяет доступность узлов СХД.
   DNS-балансировщик выбирает подходящий реальный IP-адрес узла (data-IP) и возвращает его клиенту.
   Пример преобразования: `s3.mydomain.com` — `10.0.1.10`
4. **Возврат ответа клиенту**
   Узел СХД отправляет ответ клиенту напрямую, минуя DNS-балансировщик.

Примеры DNS-балансировщиков нагрузки: PowerDNS/DNSdist, F5 BigIP DNS.

СРАВНЕНИЕ СХЕМ РАБОТЫ

Аспект	VIP-балансировка	DNS-балансировка
Маршрут ответа	Через балансировщик / напрямую	Только напрямую
Исполнитель балансировки	VIP Балансировщик	DNS балансировщик
Возврат IP	Виртуальный (VIP)	Реальный (узла СХД)

Для реализации отказоустойчивых схем **ACTIVE-PASSIVE** и **ACTIVE-PASSIVE-STANDBY**, где требуется переключение VIP между балансировщиками на разных площадках, применяются технологии кластерных IP-адресов.
Примеры решений для организации такого переключения: Keepalived, Pacemaker.

Для географически распределенных систем с требованиями локализации трафика (когда клиенты должны подключаться к ближайшей площадке) предпочтительнее использовать DNS-балансировку.

DNS-балансировки предпочтительна для:
- Локализации трафика (клиенты подключаются к ближайшей площадке).
- Гибкого управления маршрутизацией запросов.
- Устранения зависимости от виртуальных IP-адресов.
- Устранения узких мест в сетевой инфраструктуре (при использовании VIP балансировщика весь трафик идет через балансировщик).

Балансировка локальной СХД (одна площадка):
- Настройка выполняется согласно документу **«Руководство администратора»**.
- Как правило, применяется балансировщик первого типа (VIP-балансировка).
- Используется единый эндпоинт для всех узлов кластера.
- Распределение запросов осуществляется: по очереди (round-robin) или по количеству соединений.

Особенности работы с геораспределенной СХД (несколько площадок):

- **Локализация трафика**
  Клиенты должны подключаться к узлам своей локальной площадки для чтения и записи данных[2].
- **Управление репликацией**
  Определяется необходимое количество копий данных на каждой площадке.
- **Режим синхронизации**
  Определяется, должен ли клиент ждать подтверждения записи от удаленных площадок или только от локальной.
- **Балансировка нагрузки**
  Организуется распределение клиентских запросов между площадками.
- **Отказоустойчивость**
  Определяются правила переключения запросов на резервные площадки при недоступности локального кластера.

Варианты схем подключения

Для всех конфигураций действуют следующие настройки:

- Каждая площадка имеет уникальные идентификаторы `ClusterName` и `UN-LOCODE`.
- Применяется политика хранения, обеспечивающая необходимое количество копий данных.
- Запросы распределяются через систему балансировки нагрузки.

Конфигурация ACTIVE-ACTIVE для двух площадок
TATLIN.OBJECT является **ACTIVE-ACTIVE** системой.
С архитектурной точки зрения, **ACTIVE-ACTIVE** означает, что узлы СХД развернуты на обеих площадках и поддерживают операции чтения, записи и управления.
Репликация данных: синхронная или асинхронная (с обязательным сохранением хотя бы одной синхронной копии).
Копии данных поддерживаются на всех площадках одновременно.
Механизм отказоустойчивости: перенаправление запросов на другие узлы при недоступности локального кластера.
Конфигурация **ACTIVE-ACTIVE** для трех и более площадок реализуется по аналогичному принципу.
Прикладное ПО должно поддерживать работу с данной схемой (сценарий одновременного изменения объекта на двух площадках — в зависимости от настроек версионирования бакета сохранится либо последнее изменение либо все версии объекта в порядке их создания).

Конфигурация ACTIVE-PASSIVE для двух площадок
С архитектурной точки зрения, **ACTIVE-PASSIVE**[3] означает, что узлы СХД развернуты на обеих площадках, но используются по-разному:
Активная площадка (**ACTIVE**) — обрабатывает все запросы по протоколу доступа к данным.
Пассивная площадка (**PASSIVE**) — работает в режиме горячего резерва и активируется только при отказе основной.
Репликация данных: асинхронная (с возможностью сохранения синхронной копии в **ACTIVE** и асинхронных копий в **PASSIVE**).
Механизм переключения: при недоступности **ACTIVE** запросы перенаправляются на **PASSIVE**.

Конфигурация ACTIVE-PASSIVE-STANDBY для трех площадок
С архитектурной точки зрения, **ACTIVE-PASSIVE-STANDBY** означает, что узлы СХД развернуты на всех площадках, но используются по-разному:
Активная площадка (**ACTIVE**) — обрабатывает все запросы по протоколу доступа к данным.
Пассивная площадка (**PASSIVE**) — работает в режиме горячего резерва и активируется только при отказе основной.
Резервная площадка (**STANDBY**) — используется как холодный резерв или для аварийного восстановления данных.
Репликация данных: асинхронная (с возможностью сохранения синхронной копии в **ACTIVE** и асинхронных копий в **PASSIVE**/**STANDBY**).

Механизм переключения:
При недоступности **ACTIVE** запросы перенаправляются на **PASSIVE**.
STANDBY активируется вручную (или автоматически, в зависимости от политики) только при потере данных на **ACTIVE**/**PASSIVE**.

Если данные утеряны на **ACTIVE**/**PASSIVE**, возможны два сценария:
Автоматическое восстановление: сервисная служба YADRO заменяет поврежденные узлы, и данные реплицируются с **STANDBY**.
Ручное переключение: балансировщик настраивается на перевод **STANDBY** в режим **ACTIVE**.

Балансировка трафика между площадками (примеры реализации)
Для реализации отказоустойчивости комбинируются следующие подходы:

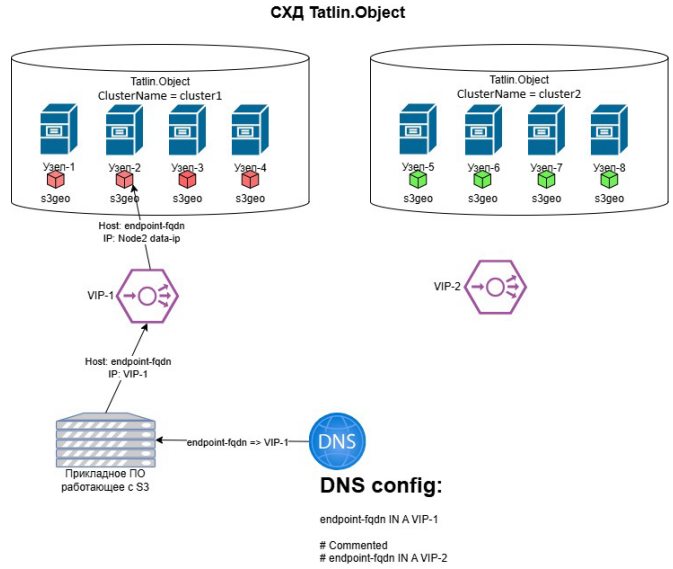
1. **DNS-переключение**
   Доменное имя эндпоинта вручную или автоматически переключается между VIP разных балансировщиков.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

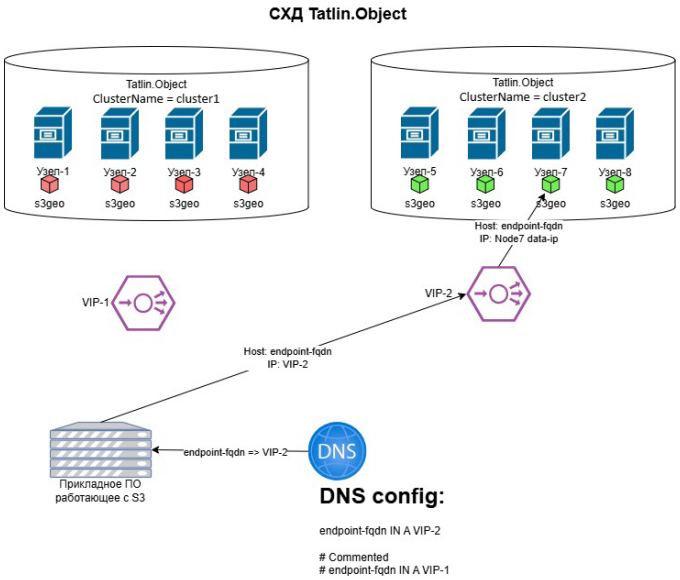
2. **Использование балансировщика с двумя пулами адресов (например, NGINX)**
   Поддерживается основной и резервный пулы адресов для одного VIP.
   Применяется при необходимости переключения между двумя площадками.
3. **Кластерный IP (VIP с управлением через Pacemaker или Keepalived)**
   Виртуальный IP переключается между несколькими балансировщиками.
   Используется для переключения между двумя и более площадками.

Пример 1: Ручное переключение через DNS на вторую площадку (ACTIVE-STANDBY):

```
Cluster1 – ACTIVE
Cluster2 - STANDBY
```




```
Cluster1 – STANDBY
Cluster2 - ACTIVE
```



Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Пример 2: Автоматическое переключение через кластерный VIP (ACTIVE-PASSIVE)[5]:

Конфигурация:

- Активная площадка (ACTIVE): два балансировщика нагрузки с экземплярами Pacemaker[6], управляющими кластерным VIP.
- Пассивная площадка (PASSIVE): один балансировщик нагрузки с экземпляром Pacemaker.

Конфигурация требует три экземпляра Pacemaker для корректного управления кластерным VIP после включения/выключения площадки.

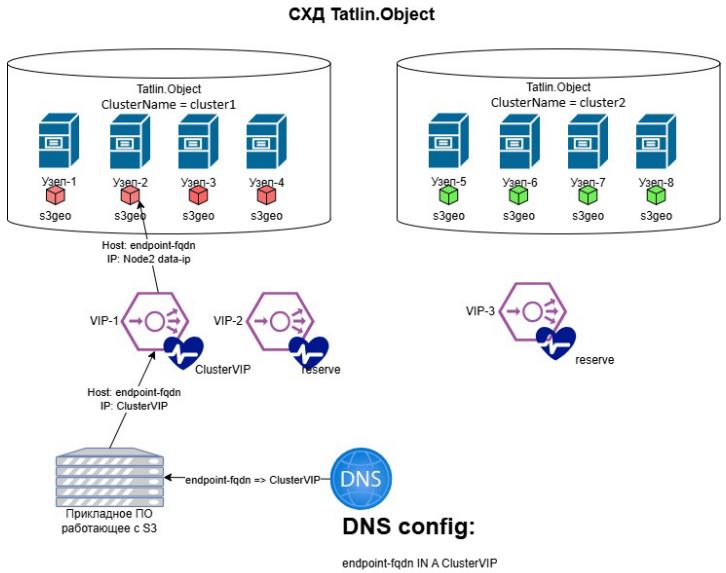
Pacemaker отслеживает:

- Доступность узлов СХД на своей площадке.
- Состояние локального балансировщика нагрузки.
- Состояние других экземпляров Pacemaker.

При недоступности локальных узлов СХД или балансировщика Pacemaker передает управление кластерным VIP на другую площадку[7].

Штатный режим:

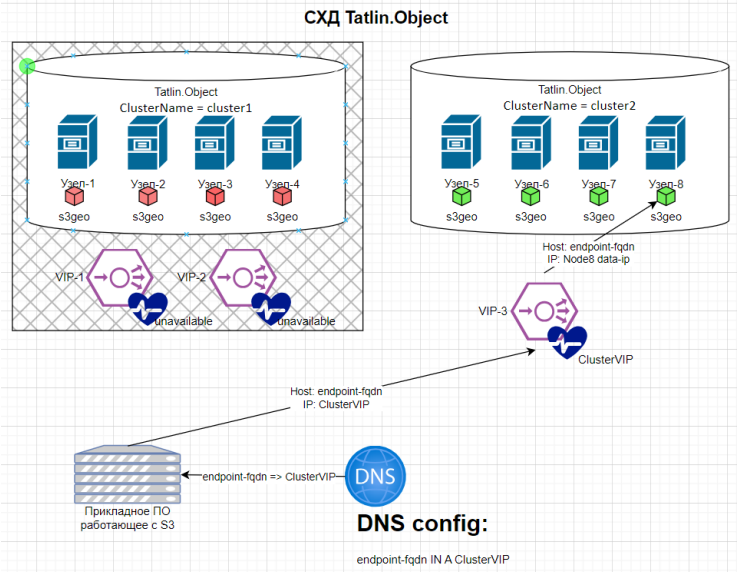
- VIP активен на основной площадке (ACTIVE).
- Основной балансировщик обрабатывает клиентские запросы.
- Трафик поступает на активную площадку и направляется на `cluster1`.
- Резервная площадка (PASSIVE) находится в режиме горячего резерва.



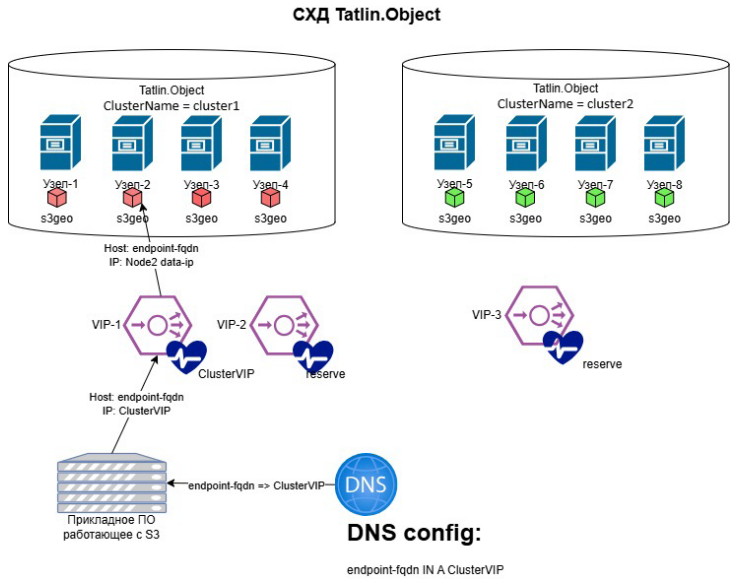
Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Аварийный сценарий: полный отказ активной площадки

Условие: на активной площадке недоступны СХД и оба балансировщика
Действия системы: Pacemaker переключает VIP на резервную площадку. Все клиентские подключения перенаправляются.
Трафик поступает на `cluster2`.



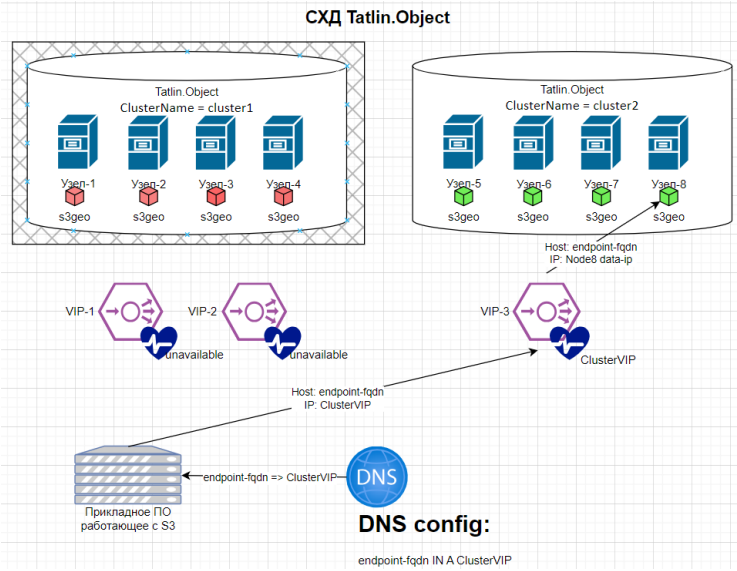
В случае восстановления активной площадки: Pacemaker возвращает VIP на основную площадку.
Трафик перенаправляется обратно с `cluster2` на `cluster1`.



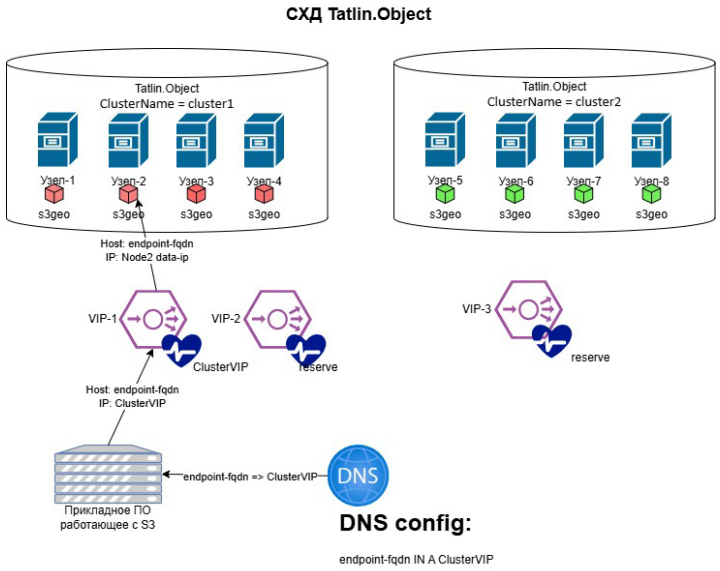
Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Аварийный сценарий: частичный отказ активной площадки

Условие: на активной площадке недоступна СХД `cluster1` , оба балансировщика доступны.
Действия системы: Pacemaker переключает VIP на резервную площадку. Все клиентские подключения автоматически перенаправляются на `cluster2`.



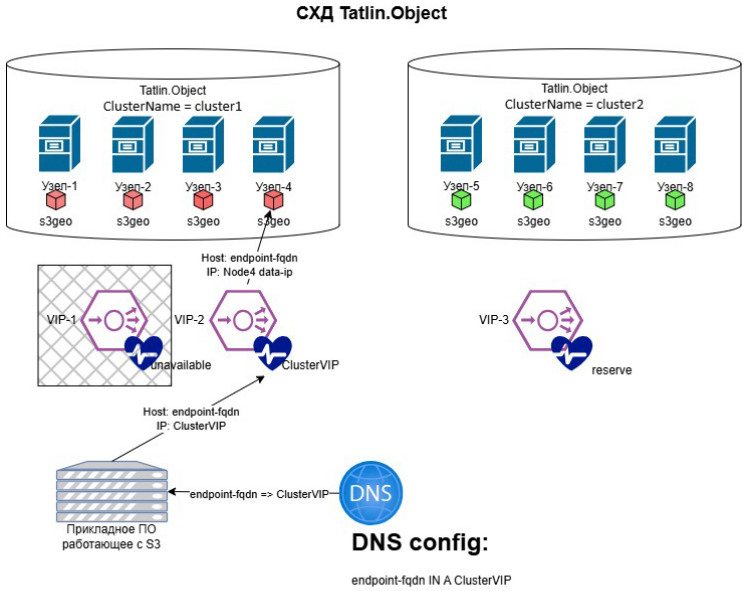
В случае восстановления активной площадки: Pacemaker возвращает VIP на основную площадку.
Трафик перенаправляется обратно с `cluster2` на `cluster1`.



Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Аварийный сценарий: частичный отказ активной площадки

Условие: на активной площадке недоступен основной балансировщик, СХД и второй балансировщик доступны.
Действия системы: Pacemaker переключает VIP на резервный балансировщик активной площадки. Трафик продолжает поступать на `cluster1`.
В случае восстановления активной площадки: Pacemaker возвращает VIP на основной балансировщик (или VIP остается на резервном в зависимости от установленной политики).



Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Пример 3: Автоматическое переключение через кластерный VIP (ACTIVE-ACTIVE):

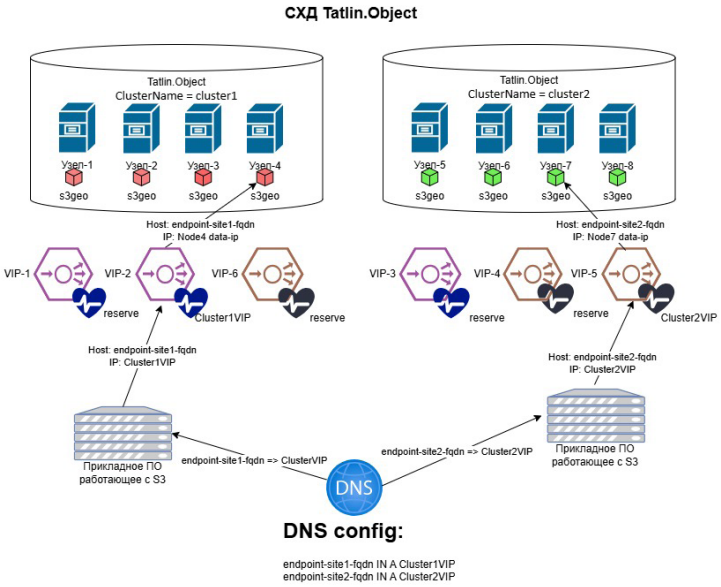
В режиме **ACTIVE-ACTIVE** клиенты могут обращаться к любой из площадок, но система оптимизирует трафик, предпочитая использовать узлы СХД на локальной площадке клиента.

Способы реализации

1. **Geo-DNS**[8]
   DNS возвращает VIP локального балансировщика на основе IP-адреса клиента.
   Для каждой площадки регистрируются уникальные эндпоинты, привязанные к локальному VIP.
2. **Симметричная схема ACTIVE-PASSIVE**
   Каждая площадка имеет: собственный набор эндпоинтов; независимый кластерный VIP (`Cluster1VIP`, `Cluster2VIP`); полную копию инфраструктуры балансировки (2 балансировщика и резерв).
   Реализует схему автоматического переключения аналогично **ACTIVE-PASSIVE**, но для каждого эндпоинта отдельно.

Штатный режим:

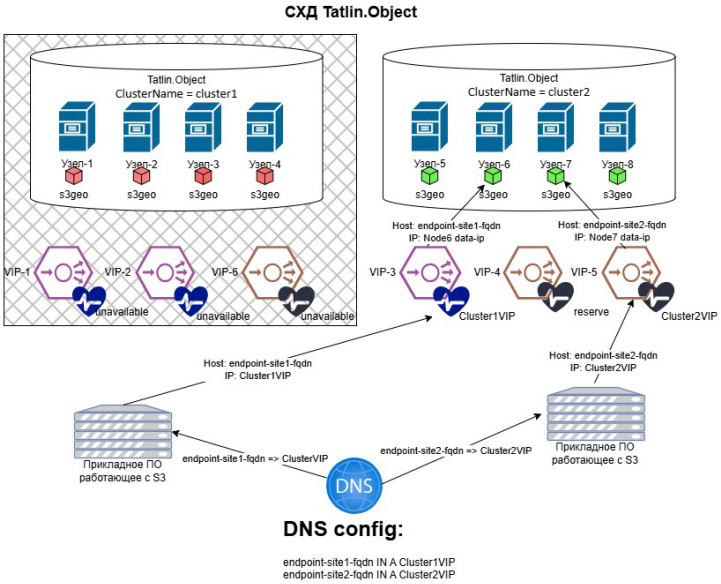
- **Локальные подключения**
  Клиенты используют эндпоинт своей площадки.
  Запросы направляются через локальный кластерный VIP на узлы СХД той же площадки.
- **Балансировка**
  Каждая площадка обслуживает своих клиентов независимо.
  Pacemaker управляет VIP в пределах своей площадки.



Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Аварийный сценарий: полный отказ активной площадки

Действия системы: клиенты проблемной площадки перенаправляются на рабочую площадку.
Рабочая площадка принимает дополнительную нагрузку через свой кластерный VIP.
В случае восстановления активной площадки: трафик возвращается на локальные узлы.
Частичные отказы (СХД или балансировщика) на одной площадке обрабатываются по тому же принципу, что и в схеме **ACTIVE-PASSIVE**:
VIP переключается на резервные компоненты в рамках текущей площадки или на соседнюю площадку при полной недоступности локальных ресурсов.



Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Примечания:

[1] Требования к балансировке:

- Обязательна поддержка Session Persistence (сохранение сессий клиентов, например, через привязку по IP).
- Обязательно использование TCP-протокола для балансировки, включая поддержку SSL-терминации.
- Обязательна проверка доступности узлов через HEAD-запросы.
- Не допускается модификация заголовков запросов для S3-трафика.
- Не допускается перенаправление портов).

[2] Локализация подключений реализуется через:

- Использование разных эндпоинтов для узлов разных площадок.
- GeoDNS (географическую привязку DNS-записей к VIP ближайшей площадки)
- примеры реализации: Механизм GeoIP в DNS Windows Server.

[3] Возможные варианты реализации ACTIVE-PASSIVE:

- Keepalived (пример).
- HAProxy с конфигурацией primary/backup.
- Встроенные механизмы коммерческих балансировщиков.

[4] Интеграционные особенности:

- TATLIN.OBJECT не включает балансировщики нагрузки и инструменты управления кластерным VIP.
- Представленные схемы — возможные варианты реализации (не входят в комплект поставки СХД).

[5] Альтернативные решения:

- Некоторые балансировщики (например NGINX) поддерживают:
- Основной и резервный пулы адресов.
- Отказоустойчивость без использования кластерного VIP.

[6] Коммерческие решения могут включать встроенное управление кластерным VIP и позволяют отказаться от использования Pacemaker.

[7] Критерии доступности:

- Определяются конкретной задачей.
- Могут учитывать недоступность N узлов на площадке.

[8] Умный DNS автоматически перенаправляет клиентов на резервную площадку при отказе основной.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

ПРИЛОЖЕНИЕ 1: ПОЛУЧЕНИЕ ИНФОРМАЦИИ О СОСТОЯНИИ СХД (УТИЛИТА FROSTFS-CLI)

Вывод карты сети

Карта сети отображает:

- Состояние узлов. Только узлы в статусе `ONLINE` участвуют в работе СХД.
- Номер эпохи. Например, `Epoch: 1827`, где 1827 — дискретное время (увеличивается на 1 каждые 10 минут, если счетчик не изменяется более 10 минут — недоступны алфавитные узлы СХД).
- Атрибуты узлов (`ClusterName`, `UN-LOCODE`, `role`), используемые в политиках хранения.

Выполните команду:

```
# frostfs-cli netmap snapshot -r <DATA-IP>:8080 -g
```

<DATA-IP> — IP адрес `data` интерфейса СХД. Является частью эндпоинта GRPC. Например, `10.78.128.166:8080`

Пример:

```
root@abaranov-node1[abaranov]:~# frostfs-cli netmap snapshot -r 10.78.128.166:8080 -g
Epoch: 1827
Node 1: 0248bbce8cb6f7367f9d405b0d6a5f99fa8368740bcdf01973f01a63182617164d ONLINE /ip4/192.168.198.166/tcp/8080 /ip4/192.168.199.166/tcp/8080
        ClusterName: abaranov
        Continent: Europe
        Country: Russia
        CountryCode: RU
        ExternalAddr: /ip4/10.78.128.166/tcp/8080,/ip4/10.78.129.166/tcp/8080,/ip4/10.78.128.166/tcp/8081/tls,/ip4/10.78.129.166/tcp/8081/tls,/
        Location: Moskva
        Node: 10.78.132.166
        SubDiv: Moskva
        SubDivCode: MOW
        UN-LOCODE: RU MOW
        role: alphabet
Node 2: 02977e3dbea58e4b8d8a2814260a201cf288b05e2a9fc7aba7be2c7357bb87ca88 ONLINE /ip4/192.168.198.167/tcp/8080 /ip4/192.168.199.167/tcp/8080
        ClusterName: abaranov
        Continent: Europe
        Country: Finland
        CountryCode: FI
        ExternalAddr: /ip4/10.78.128.167/tcp/8080,/ip4/10.78.129.167/tcp/8080,/ip4/10.78.128.167/tcp/8081/tls,/ip4/10.78.129.167/tcp/8081/tls,/
        Location: Helsinki (Helsingfors)
        Node: 10.78.132.167
        SubDiv: Uusimaa
        SubDivCode: 18
        UN-LOCODE: FI HEL
        role: alphabet
Node 3: 0297f7aa7dd5b93351dcd01dccc14d7b84d72e81d7c43cd2149e8a111998e8fa62 ONLINE /ip4/192.168.198.168/tcp/8080 /ip4/192.168.199.168/tcp/8080
        ClusterName: abaranov
        Continent: Europe
        Country: Sweden
        CountryCode: SE
        ExternalAddr: /ip4/10.78.128.168/tcp/8080,/ip4/10.78.129.168/tcp/8080,/ip4/10.78.128.168/tcp/8081/tls,/ip4/10.78.129.168/tcp/8081/tls,/
        Location: Stockholm
        Node: 10.78.132.168
        SubDiv: Stockholms l□n
        SubDivCode: AB
        UN-LOCODE: SE STO
        role: alphabet
Node 4: 02d281ebc0c7c5269e7d659b682589dfc56ebfa2f77e41492f877c616903502787 ONLINE /ip4/192.168.198.169/tcp/8080 /ip4/192.168.199.169/tcp/8080
        ClusterName: abaranov
        Continent: Europe
        Country: Russia
        CountryCode: RU
        ExternalAddr: /ip4/10.78.128.169/tcp/8080,/ip4/10.78.129.169/tcp/8080,/ip4/10.78.128.169/tcp/8081/tls,/ip4/10.78.129.169/tcp/8081/tls,/
        Location: Saint Petersburg (ex Leningrad)
        Node: 10.78.132.169
        SubDiv: Sankt-Peterburg
        SubDivCode: SPE
        UN-LOCODE: RU LED
        role: alphabet
```

Вывод политики бакета (контейнера)

1. Для получения CID контейнера выполните команду:

```
# aws s3api head-bucket --debug --bucket <BUCKET_NAME> --no-verify-ssl
--endpoint-url <S3_ENDPOINT> 2>&1 | grep 'X-Container-Id'
```

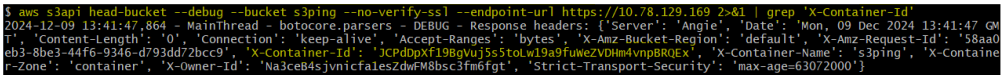
<BUCKET_NAME> — имя целевого бакета. Например, `s3ping`.
<S3_ENDPOINT> — адрес S3 эндпоинта, Например `https://10.78.128.166`

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Пример:

```
# aws s3api head-bucket --debug --bucket s3ping --no-verify-ssl --endpoint-url
https://10.78.129.169 2>&1 | grep 'X-Container-Id'
```

Согласно примеру, узел с IP-адресом `10.78.129.169` сообщает, что бакет `s3ping` соответствует gRPC контейнеру `JCPdDpXf19BgVuj5s5toLw19a9fuWeZVDHm4vnpBRQEx`.

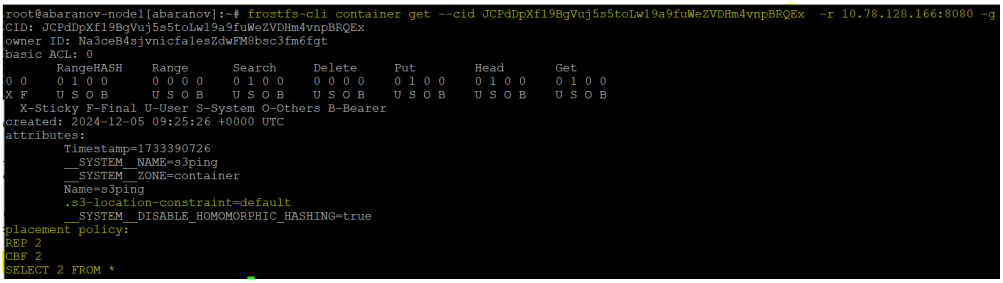


3. Для просмотра политики хранения выполните команду:

```
# frostfs-cli container get --cid <CONTAINER_ID> -r <DATA-IP>:8080 -g
```

<CONTAINER_ID> — уникальный идентификатор контейнера. Отображается при создании контейнера в поле вывода `container ID`. Например, `4fNnsGhgwKqpphYLZCrHDaUAtJmwGM8jVf7EPER1Hyqw`.
<DATA-IP> — IP адрес data интерфейса СХД. Является частью эндпоинта GRPC. Например, `10.78.128.166:8080`

Согласно примеру, узел с IP-адресом `10.78.129.169` сообщает, что бакет `s3ping` создан с политикой хранения `REP 2 CBF 2 SELECT 2 FROM *`.



Вывод списка контейнерных узлов

Выполните команду:

```
# frostfs-cli container nodes --cid <CONTAINER_ID> -r <DATA-IP>:8080 -g
```

<CONTAINER_ID> — уникальный идентификатор контейнера. Отображается при создании контейнера в поле вывода `container ID`. Например, `4fNnsGhgwKqpphYLZCrHDaUAtJmwGM8jVf7EPER1Hyqw`.
<DATA-IP> — IP адрес `data` интерфейса СХД. Является частью эндпоинта GRPC. Например, `10.78.128.166:8080`

Команда выводит:

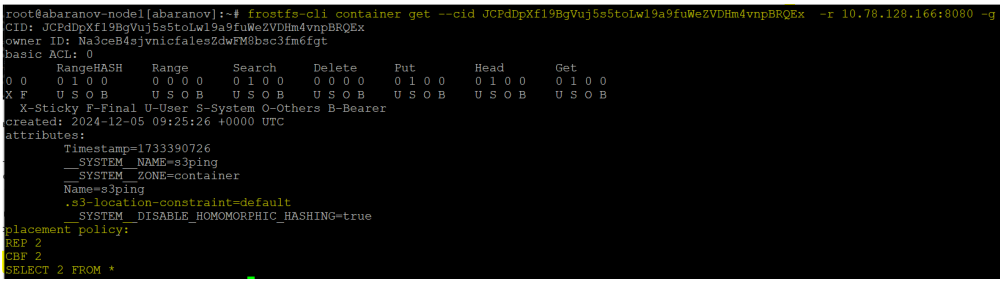
- Список узлов хранения.
- Фактор репликации.

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Пример:

```
# frostfs-cli container nodes --cid 4fNnsGhgwKqpphYLZCrHDaUAtJmwGM8jVf7EPER1Hyqw -r
https://10.78.128.166:8080 -g
```

Согласно примеру, узел с IP-адресом `10.78.129.168` сообщает, что бакет `s3ping` хранится с фактором репликации `REP 2` на 4 узлах системы.



Соотнесение s3-объекта и OID

Выполните команду:

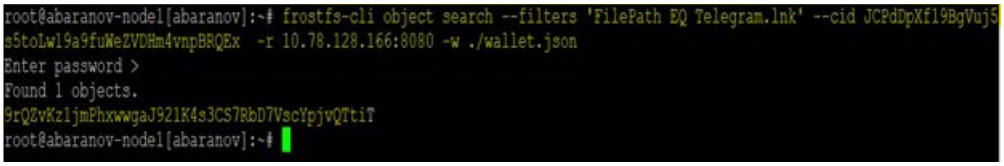
```
# frostfs-cli object search --filters 'FilePath EQ <OBJECT_NAME>' --cid <CONTAINER_ID> -r
<DATA-IP>:8080 -w <WALLET_PATH>
```

<OBJECT_NAME> — имя объекта в бакете. Например, `my-object-1`.
<CONTAINER_ID> — уникальный идентификатор контейнера. Отображается при создании контейнера в поле вывода `container ID`. Например, `4fNnsGhgwKqpphYLZCrHDaUAtJmwGM8jVf7EPER1Hyqw`.
<WALLET_PATH> — путь к файлу `wallet.json` с секретным ключом.
<DATA-IP> — IP адрес data интерфейса СХД. Является частью эндпоинта GRPC. Например, `10.78.128.166:8080`

Пример:

```
# frostfs-cli object search --filters 'FilePath EQ Telegram.lnk' --cid 9rQZvKz1jmPhxwwgaJ921K
4s3CS7RbD7VscYpjvQTtiT -r https://10.78.128.166:8080 -w ./wallet.json
```

Согласно примеру, узел с IP-адресом `10.78.129.166` сообщает, что бакет `s3ping` хранит S3-объект `Telegram.lnk` в gRPC-объекте с OID `9rQZvKz1jmPhxwwgaJ921K4s3CS7RbD7VscYpjvQTtiT`.



Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

Вывод списка узлов, где расположена копия объекта

Выполните команду:

```
# frostfs-cli object nodes --oid <OBJECT_ID> --cid <CONTAINER_ID> -r <DATA-IP>:8080 -w
<WALLET_PATH>
```

<CONTAINER_ID> — уникальный идентификатор контейнера. Отображается при создании контейнера в поле вывода container ID. Например, `4fNnsGhgwKqpphYLZCrHDaUAtJmwGM8jVf7EPER1Hyqw`.
<OBJECT_ID> — уникальный идентификатор объекта. Вместе с **<CONTAINER_ID>** однозначно определяет объект. Например, `E22dge15NLeznzuZw6Yn566bmJWNy1czaiy6bArit5xh`.
<WALLET_PATH> — путь к файлу `wallet.json` с секретным ключом.
<DATA-IP> — IP адрес data интерфейса СХД. Является частью эндпоинта GRPC. Например, `10.78.128.166:8080`

Пример:

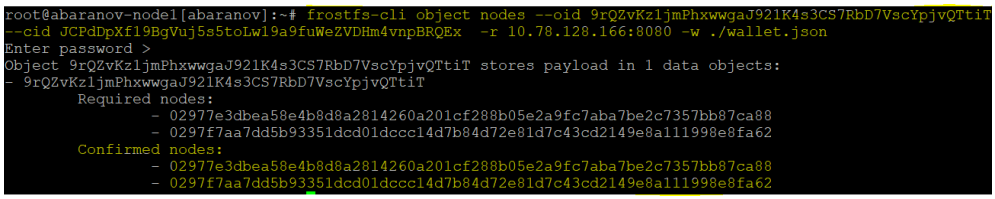
```
# frostfs-cli object search --filters 'FilePath EQ Telegram.lnk' --cid 9rQZvKz1jmPhxwwgaJ921K
4s3CS7RbD7VscYpjvQTtiT -r https://10.78.128.166:8080 -w ./wallet.json
```

Команда выводит:

- Узлы хранения реплик.
- Их идентификаторы, которые можно сопоставить с их атрибутами в выводе команд `frostfs-cli container nodes` и `frostfs-clinetmap snapshot`.

Согласно примеру, узел с IP-адресом `10.78.129.166` сообщает, что бакет `s3ping` хранит объект `9rQZvKz1jmPhxwwgaJ921K4s3CS7RbD7VscYpjvQTtiT` с фактором репликации `REP 2` на двух узлах системы с идентификаторами:

- `02977e3dbea58e4b8d8a2814260a201cf288b05e2a9fc7aba7be2c7357bb87ca88`
- `0297f7aa7dd5b93351dcd01dccc14d7b84d72e81d7c43cd2149e8a111998e8fa62`



Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11

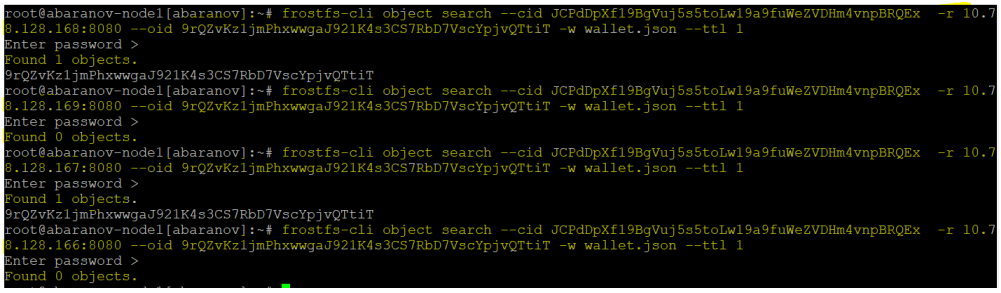
Проверка наличия объекта на контейнерном узле

Выполните команду:

```
# frostfs-cli object search --oid <OBJECT_ID> --cid <CONTAINER_ID> -r <DATA-IP>:8080 -w
<WALLET_PATH> --ttl 1
```

Пример:

```
# frostfs-cli object search --oid E22dge15NLeznzuZw6Yn566bmJWNy1czaiy6bArit5xh --cid 4fNnsGh
gwKqpphYLZCrHDaUAtJmwGM8jVf7EPER1Hyqw -r https://10.78.128.166:8080 -w /etc/frostfs/storage/
wallet.json --ttl 1
```



Получение S3-имени объекта

Имя S3-объекта хранится в атрибуте `FilePath` заголовка объекта.

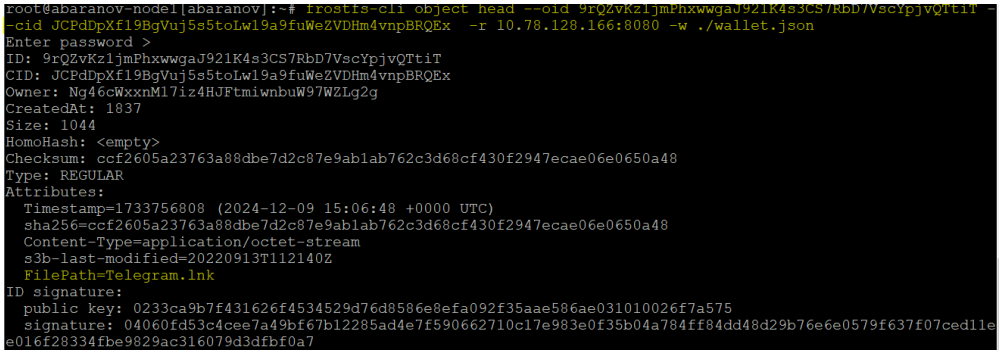
Выполните команду:

```
# frostfs-cli object head --oid <OBJECT_ID> --cid <CONTAINER_ID> -r <DATA-IP>:8080 -w
<WALLET_PATH>
```

Пример:

```
# frostfs-cli object head --oid E22dge15NLeznzuZw6Yn566bmJWNy1czaiy6bArit5xh --cid 4fNnsGhg
wKqpphYLZCrHDaUAtJmwGM8jVf7EPER1Hyqw -r https://10.78.128.166:8080 -w /etc/frostfs/storage/
wallet.json
```

Согласно примеру, узел с IP-адресом `10.78.129.166` сообщает, что бакет `s3ping` содержит в хедере объекта `9rQZvKz1jmPhxwwgaJ921K4s3CS7RbD7VscYpjvQTtiT` значение атрибута `FilePath = Telegram.lnk`:



ЗАКЛЮЧЕНИЕ

TATLIN.OBJECT зарекомендовал себя как ведущее в отрасли масштабируемое программно-аппаратное решение для хранения объектных типов данных. Система идеально подходит как самостоятельное решение для специализированных задач отделов разработки, хранения архивов, резервных копий, так и для частной облачной инфраструктуры. TATLIN.OBJECT обеспечивает надежное хранилище с возможностью геораспределения узлов хранения. TATLIN.OBJECT разработан на основе базового набора принципов проектирования объектной системы хранения корпоративного класса — масштабирование практически неограниченной емкости, консолидации нескольких хранилищ с уменьшенными расходами на управление эксплуатацией, постоянной доступностью данных и высочайшими уровнями надежности данных. Система гарантирует защиту данных в долгосрочной перспективе с сохранением экономической эффективности.

TATLIN.OBJECT предоставляет комплексное решение программно-определяемого хранилища (SDS) на стандартных отраслевых серверных платформах Yadro, что дает возможность полностью соответствовать поставленным принципам при разработке системы.

Дополнительную информацию для TATLIN.OBJECT можно найти на веб-сайте **service.yadro.com**.

ДОПОЛНИТЕЛЬНЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ

Дополнительные документы можно получить через раздел «Документы» портала service.yadro.com, выбрав TATLIN.OBJECT в списке продуктов и версию ПО.

1. Требования к площадке
2. Руководство администратора

Свяжитесь с нами:
support@yadro.com
www.yadro.com
Москва, ул. Рочдельская, 15, стр.13
+7 800 777-06-11