

Хранение данных

Распределённое хранилище Ceph

Введение

Обзор возможностей

Сравнение решений для хранения

Установка

Основные понятия

Как сделать

Архитектура

Техническая архитектура

Руководство

MinIO Object Storage

Введение

Основные понятия

Руководства

Установка

Предварительные требования

Как сделать

Архитектура

Основные компоненты

настройка

операции

Заключение:

Локальное хранилище ToroLVM

[Введение](#)

[Установка](#)

[Руководство](#)

Предварительные требования

[Как сделать](#)

Распределённое хранилище Ceph

Введение

Введение

Обзор возможностей

Сравнение решений для хранения

Установка

Создание кластера стандартн

Предварительные требования

Важные моменты

Порядок действий

Связанные операции

Создание Stretch Type Кластера

Терминология

Типовая схема развертывания

Ограничения и лимиты

Предварительные условия

Процедура

Связанные операции

Архитектура

Архитектура

Техническая архитектура

Основные понятия

Основные концепции

Rook Operator

Ceph CSI

Функции модулей Ceph

Руководства

Доступ к сервисам хранения

Предварительные требования

Процедура

Последующие действия

Управление Storage Pools

Создание Storage Pool

Удаление Storage Pool

Просмотр адресов Object Storage Pool

Развертывание

Обновление к

Перезапуск ко

Добавление устройств/классов

Добавление классов устройств

Добавление устройств

Статус жесткого диска

Мониторинг и оповещения

Мониторинг

Оповещения

Как сделать

Настройка выделенного класса

Архитектура

Требования к инфраструктуре

Процедура

Последующие действия

Очистка распределённого хранилища

Меры предосторожности

Процедура

Восстановление

Обновление

Процедура

Создание пользователя Ceph Object Store

Предварительные требования

Процедура

Установка и настройка

Предварительные требования

Установка квот

Установка квот

Установка квот

Проверка квот

Введение

Alauda Build of Rook-Ceph — это гиперконвергентное решение для хранения данных, предоставляемое платформой внутри кластера. Основанное на открытом решении Rook + Ceph, распределённое хранилище обеспечивает автоматическое управление, автоматическое масштабирование и автоматическое восстановление, удовлетворяя потребности малых и средних приложений в блочном, файловом и объектном хранении.

NOTE

В данном документе **распределённое хранилище** означает Ceph-хранилище внутри данного кластера, а **внешнее хранилище** — Ceph-хранилище вне этого кластера.

Содержание

[Обзор возможностей](#)

Сравнение решений для хранения

Создание кластера хранения

Доступ к внешнему хранилищу

Обзор возможностей

- **Простое развертывание:** Предоставляет графические сервисы автоматического развертывания и управления кластерами хранения; поддерживает как интегрированный, так и отдельный режим развертывания вычислений и хранения.

- **Профессиональные операции:** Обеспечивает функции создания снимков постоянных томов и клонирования новых томов; визуальный мониторинг ёмкости, производительности и состояния компонентов; оснащён встроенными политиками оповещений для удовлетворения большинства сценариев эксплуатации хранилища.
- **Безопасность и надёжность:** Распределённый механизм с мульти-репликами гарантирует безопасность и надёжность данных; простое и надёжное автоматизированное управление поддерживает онлайн-расширение ресурсов хранения.
- **Отличная производительность:** Предоставляет эластичные и высокопроизводительные сервисы хранения; поддерживает развертывание гибридных дисковых устройств для повышения производительности и эффективности системы хранения.

Сравнение решений для хранения

Платформа поддерживает два типа решений для хранения; вы можете выбрать одно из них.

Создание кластера хранения

Требование	Преимущества
Вы можете выбрать создание либо кластера стандартного типа , либо кластера расширенного типа	Нет необходимости в дополнительной подготовке решения для хранения; настройка может быть выполнена на бизнес-кластере, что экономит затраты.

Доступ к внешнему хранилищу

Вариант 1: Доступ к распределённым ресурсам хранения других бизнес-кластеров внутри платформы для обеспечения изоляции хранения и бизнеса, что упрощает управление и сопровождение.

Вариант 2: Интеграция внешних Serp-ресурсов хранения в качестве распределённого хранилища.

Требование (выберите одно)	Преимущества
Вариант 1: Распределённое хранилище уже развернуто в других бизнес-кластерах.	<p>Позволяет полноценно использовать ресурсы хранения между кластерами и избегать влияния изменений в бизнесе. Обеспечивает безопасность и стабильность данных при снижении сложности эксплуатации.</p> <p>Примечание: Если хранилище для доступа — распределённое хранилище с других платформ, например, основная/резервная платформа в среде аварийного восстановления, используйте метод интеграции внешнего Serp.</p>
Вариант 2: Внешнее Serp-хранилище вне платформы, версия $\geq 14.2.3$.	<p>По сравнению с прямым созданием класса хранения, этот метод удобнее для использования интерфейса платформы для создания снимков томов, масштабирования и других функций.</p>

Примечание: Если необходимо поддерживать пул хранения, устройства хранения и другие настройки внешнего хранилища, операции должны выполняться в интерфейсе управления кластера хранения.

Установка

Создание кластера стандартн

Предварительные требования

Важные моменты

Порядок действий

Связанные операции

Создание Stretch Type Кластера

Терминология

Типовая схема развертывания

Ограничения и лимиты

Предварительные условия

Процедура

Связанные операции

Создание кластера стандартного типа

Кластер стандартного типа — это наиболее типичный способ развертывания хранилища Serf. Он распределяет реплики данных по жестким дискам на разных хостах, обеспечивая, что при отказе одного хоста копии данных на других хостах смогут поддерживать доступность сервиса.

Содержание

[Предварительные требования](#)

- Подготовка пакетов

- Подготовка инфраструктуры

- Важные моменты

- Порядок действий

 - Развертывание Alauda Container Platform Storage Essentials

 - (Опционально) Развертывание Alauda Build of LocalStorage

 - Развертывание Operator

 - Создание кластера

 - Создание пула хранения

- Связанные операции

 - Создание кластера типа Stretch

 - Очистка распределенного хранилища

Предварительные требования

Подготовка пакетов

- Скачайте установочный пакет **Alauda Container Platform Storage Essentials**, соответствующий архитектуре вашей платформы.
- Загрузите установочный пакет **Alauda Container Platform Storage Essentials** с помощью механизма Upload Packages.
- Скачайте установочный пакет **Alauda Build of Rook-Ceph**, соответствующий архитектуре вашей платформы.
- Загрузите установочный пакет **Alauda Build of Rook-Ceph** с помощью механизма Upload Packages.

Подготовка инфраструктуры

- Для кластера хранения требуется не менее 3 узлов.
 - Каждый узел должен иметь как минимум 1 пустой жесткий диск или 1 неформатированный раздел жесткого диска.
 - Рекомендуемый объем доступного дискового пространства — более 50 ГБ.
 - Если вы используете присоединенный Kubernetes кластер с Containerd в качестве компонента runtime, убедитесь, что параметр `LimitNOFILE` в файле `/etc/systemd/system/containerd.service` на всех узлах кластера установлен в значение `1048576` для успешного развертывания распределенного хранилища. Инструкции по настройке см. в разделе [Modifying Containerd Configuration Information](#).
- Примечание:** При обновлении с версий ранее v3.10.2 до текущей версии, если необходимо развернуть распределенное хранилище Ceph на вашем кастомном Kubernetes кластере с Containerd в качестве runtime, также необходимо установить параметр `LimitNOFILE` в файле `/etc/systemd/system/containerd.service` на всех узлах кластера в значение `1048576`.

Важные моменты

Создание сервиса хранения и Доступ к сервису хранения поддерживают выбор только одного метода.

Порядок действий

1 Развертывание Alauda Container Platform Storage Essentials

1. Войдите в систему, перейдите на страницу **Administrator**.
2. Нажмите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите **Alauda Container Platform Storage Essentials**, нажмите **Install** и перейдите на страницу **Install Alauda Container Platform Storage Essentials**.

Параметры конфигурации:

Параметр	Рекомендуемая конфигурация
Channel	По умолчанию канал <code>stable</code> .
Installation Mode	<code>Cluster</code> : Все пространства имен в кластере используют один экземпляр Operator для создания и управления, что снижает использование ресурсов.
Installation Place	Выберите <code>Recommended</code> , Namespace поддерживается только <code>acp-storage</code> .
Upgrade Strategy	<code>Manual</code> : При появлении новой версии в Operator Hub требуется ручное подтверждение для обновления Operator до последней версии.

2 (Опционально) Развертывание Alauda Build of LocalStorage

При использовании метода `Selection Device` для добавления устройств хранения в кластер Serp необходимо развернуть `Alauda Build of LocalStorage Operator`. Этот Operator автоматически обнаруживает все жесткие диски на каждом узле Kubernetes кластера и собирает полную информацию об устройствах, упрощая процесс интеграции хранилища.

1. Войдите в систему, перейдите на страницу **Administrator**.
2. Нажмите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите **Alauda Build of LocalStorage**, нажмите **Install** и перейдите на страницу **Install Alauda Build of LocalStorage**.

Параметры конфигурации:

Параметр	Рекомендуемая конфигурация
Channel	По умолчанию канал <code>stable</code> .
Installation Mode	<code>Cluster</code> : Все пространства имен в кластере используют один экземпляр Operator для создания и управления, что снижает использование ресурсов.
Installation Place	Выберите <code>Recommended</code> , Namespace поддерживается только <code>acp-storage</code> .
Upgrade Strategy	<code>Manual</code> : При появлении новой версии в Operator Hub требуется ручное подтверждение для обновления Operator до последней версии.

3

Развертывание Operator

1. Перейдите в раздел **Administrator**.
2. В левой боковой панели нажмите **Storage Management > Distributed Storage**.
3. Нажмите **Configure Now**.
4. На странице мастера **Deploy Operator** нажмите кнопку **Deploy Operator** в правом нижнем углу.
 - Автоматический переход страницы к следующему шагу означает успешное развертывание Operator.
 - Если развертывание не удалось, следуйте подсказке интерфейса **Clean Up Deployed Information and Retry** и повторите развертывание Operator; чтобы вернуться на страницу выбора распределенного хранилища, нажмите

Application Store, сначала удалите ресурсы уже развернутого **rook-operator**, затем удалите сам **rook-operator**.

4

Создание кластера

1. На странице мастера **Create Cluster** настройте соответствующие параметры и нажмите кнопку **Create Cluster** в правом нижнем углу.

Параметр	Описание
Cluster Type	Выберите Standard .
Device Class Type	<p>Классы устройств — это группировки жестких дисков; вы можете настраивать классы устройств в соответствии с вашими требованиями к хранению, распределяя разные типы данных по дискам с разной производительностью.</p> <ul style="list-style-type: none"> • Default Device Class: Платформа автоматически классифицирует типы жестких дисков на узлах кластера, например, создавая классы устройств с именами <code>hdd</code>, <code>ssd</code>, <code>nvme</code>. • Custom Device Class: Позволяет задать имя класса устройств для конкретных комбинаций дисков на узле; поддерживается добавление нескольких классов устройств. Один жесткий диск может принадлежать только одному классу устройств.
Device Class - Name	Имя класса устройств. При выборе Custom Device Class имя класса не может быть <code>hdd</code> , <code>ssd</code> , <code>nvme</code> .
Device Class - Storage Devices	<p>Для добавления устройств хранения в класс устройств можно выбрать методы <code>Selection Device</code> и <code>Input Device</code>:</p> <ul style="list-style-type: none"> • Selection Device: Выбор из доступных устройств хранения. Устройство считается доступным, если выполняются следующие условия:

Параметр	Описание
	<ul style="list-style-type: none"> • Тип устройства — диск или mpath • Файловая система не обнаружена (fsType пустой) • Емкость превышает 10 ГиБ <p>Устройства типа rbd, nbd и dm-* не отображаются в списке доступных для выбора.</p> <p>Примечание: Требуется предварительное развертывание Alauda Build of LocalStorage Operator.</p> <ul style="list-style-type: none"> • Input Device: Ручной ввод имен пустых устройств на узле, например, <code>sda</code>. <p>Примечание: Для оптимальной производительности и управления настоятельно рекомендуется использовать необработанные диски, а не отдельные разделы на диске.</p>
Snapshot	<p>При включении поддерживается создание снимков PVC и использование снимков для настройки новых PVC для быстрого резервного копирования и восстановления данных. Если снимки не были включены при создании хранилища, их можно включить позже в разделе Operations на странице деталей кластера хранения.</p> <p>Примечание: Перед использованием убедитесь, что для текущего кластера развернуты плагины volume snapshot.</p>
Monitoring Alarm	<p>При включении обеспечивается готовый сбор метрик мониторинга и возможности оповещений, см. Monitoring and Alarming.</p> <p>Примечание: Если не включать сейчас, потребуется использовать альтернативные решения для мониторинга и оповещений хранилища, например, вручную настраивать</p>

Параметр	Описание
	панели мониторинга и стратегии оповещений в центре эксплуатации.

2. Нажмите **Advanced Configuration** для расширенной настройки компонентов.

Параметр	Описание
Network Configuration	<ul style="list-style-type: none"> • Host Network: Кластер хранения будет использовать сеть хоста, необходимо заполнить соответствующие параметры оптимизации сети в колонке параметров оптимизации, например, указать подсети <code>public</code> и <code>cluster</code>. Если оставить пустым, будет использована подсеть хоста по умолчанию. <p>Примечание: Использование сети хоста может представлять угрозу безопасности из-за передачи данных в незашифрованном виде через порты хоста. Обратитесь в службу поддержки платформы для получения решения по шифрованию передачи.</p> <ul style="list-style-type: none"> • Container Network: Кластер хранения будет использовать контейнерную сеть; можно создать подсети в управлении сетью и назначить их пространству имен <code>rook-ceph</code>. Если оставить пустым, будет использована подсеть по умолчанию. <p>Примечание: IPv6 не поддерживается. При использовании контейнерной сети доступ к хранилищу возможен только внутри кластера. Сбои или перезапуски Pod Ceph CSI могут привести к прерыванию сервиса.</p>
Optimization Parameters	Поддерживается заполнение параметров в формате конфигурационного файла Ceph; система перезапишет параметры по умолчанию на основе предоставленного содержимого.

Параметр	Описание
	Примечание: После первого заполнения или изменения параметров инициализации необходимо нажать на параметры инициализации; успешная инициализация обязательна для создания кластера.
Component Fixed-point Deployment	Можно развернуть компоненты на указанных узлах; требуется минимум три узла для обеспечения минимальной доступности. Компоненты, поддерживающие настройку фиксированного размещения: MON, MGR, MDS, RGW.

- Автоматический переход страницы к следующему шагу означает успешное развертывание кластера Ceph.
- Если создание не удалось, можно нажать очистку **Created Information or Retry** для автоматической очистки ресурсов и повторного создания кластера, либо вручную очистить ресурсы согласно документации [Distributed Storage Service Resource Cleanup](#).

5

Создание пула хранения

1. На странице мастера **Create Storage Pool** настройте соответствующие параметры и нажмите кнопку **Create Storage Pool** в правом нижнем углу.

Параметр	Описание
Storage Type	<ul style="list-style-type: none"> Файловое хранилище: обеспечивает безопасные, надежные и масштабируемые услуги совместного файлового хранения. Подходит для совместного доступа к файлам, резервного копирования и т.д. Блочное хранилище: обеспечивает высокую производительность IOPS и низкую задержку. Подходит для баз данных, виртуализации и т.д. Объектное хранилище: предоставляет стандартный интерфейс S3, подходит для больших данных, резервного архивирования, облачного хранения и т.д.
Replica Count	Чем больше количество реплик, тем выше избыточность и безопасность данных, однако уменьшается эффективность использования хранилища. Обычно устанавливается значение 3, что удовлетворяет большинству потребностей.
Device Class	<p>Единообразная классификация устройств одного типа или дисков с одинаковой бизнес-логикой, выбирается из классов устройств, добавленных на предыдущем шаге.</p> <ul style="list-style-type: none"> При выборе класса устройств данные будут храниться в выбранном классе. Если класс устройств не выбран, данные будут случайным образом распределены по всем устройствам в пуле хранения.

Для объектного хранилища необходимо также настроить следующие параметры:

Параметр	Описание
Region	Укажите регион, в котором расположен пул хранения.

Параметр	Описание
Gateway Type	По умолчанию S3, изменить нельзя.
Internal Port	Укажите порт для внутреннего доступа в кластере.
External Access	Включение/отключение внешнего доступа создаст/удалит сервис типа Nodeport.
Instance Count	Количество экземпляров ресурсов для объектного хранилища.

- Автоматический переход страницы к следующему шагу означает успешное развертывание пула хранения.
- Если развертывание не удалось, проверьте основные компоненты согласно подсказкам интерфейса, затем нажмите **Clean Up Created Information and Retry** для повторного создания пула хранения.

2. Нажмите **Create Storage Pool**. Во вкладке **Details** можно просмотреть информацию о созданном пуле хранения.

Связанные операции

Создание кластера типа Stretch

Подробности см. в разделе [Create Stretch Type Cluster](#).

Очистка распределенного хранилища

Подробности см. в разделе [Cleanup Distributed Storage](#).

Создание Stretch Type Кластера

Stretch кластер может распространяться на две географически разнесённые локации, обеспечивая возможности аварийного восстановления для инфраструктуры хранения. В случае катастрофы, когда одна из зон доступности из двух полностью недоступна, Serp всё равно может поддерживать доступность.

⚠️ ВАЖНО: Дисклеймер для Alpha-функции

В этом документе описывается функция, находящаяся в **Alpha-стадии**, предоставляемая исключительно для ранней технической проверки и оценки. Как поставщик функции, мы не даём никаких гарантий по её стабильности, надёжности или целостности данных. Настраивая и используя эту функцию, вы принимаете следующие технические ограничения:

- **Не для использования в продакшене:** Функция не прошла всестороннюю системную проверку. Использование в продуктивных средах несёт высокий риск сбоев процессов или повреждения данных.
- **Отсутствие гарантии SLA:** Функция не входит в стандартные соглашения об уровне обслуживания (SLA). Мы не гарантируем сроки реакции технической поддержки и не предоставляем экстренные хотфиксы.
- **Изменения с нарушением обратной совместимости и устаревание:** API, схемы конфигурации и основная логика обработки могут изменяться с нарушением обратной совместимости в будущих релизах. Функция может быть полностью удалена без предварительного уведомления.
- **Отсутствие плавного пути обновления:** Мы не предоставляем скрипты обновления или инструменты миграции данных между версиями. Обновление обычно требует полного удаления существующих ресурсов и новой установки. Любые потери состояния или данных ложатся на ответственность пользователя.

Содержание

Терминология

Типовая схема развертывания

Описание компонентов

Объяснение аварийного восстановления

Ограничения и лимиты

Предварительные условия

Процедура

Тегирование узлов

Создание сервиса хранения

Связанные операции

Создание стандартного типа кластера

Очистка распределённого хранилища

Терминология

Термин	Объяснение
Quorum Availability Zone	Обычно располагается в отдельной зоне, не несущей основные рабочие нагрузки, ориентирована на поддержание согласованности кластера и используется преимущественно для принятия арбитражных решений при сбоях в основном дата-центре или сетевых подразделениях.
Data Availability Zone	Основная зона в кластере Serph, где фактически хранятся и обрабатываются данные, несущая операционные нагрузки и задачи хранения, вместе с зоной кворума формирует полноценную систему высокодоступного хранения.

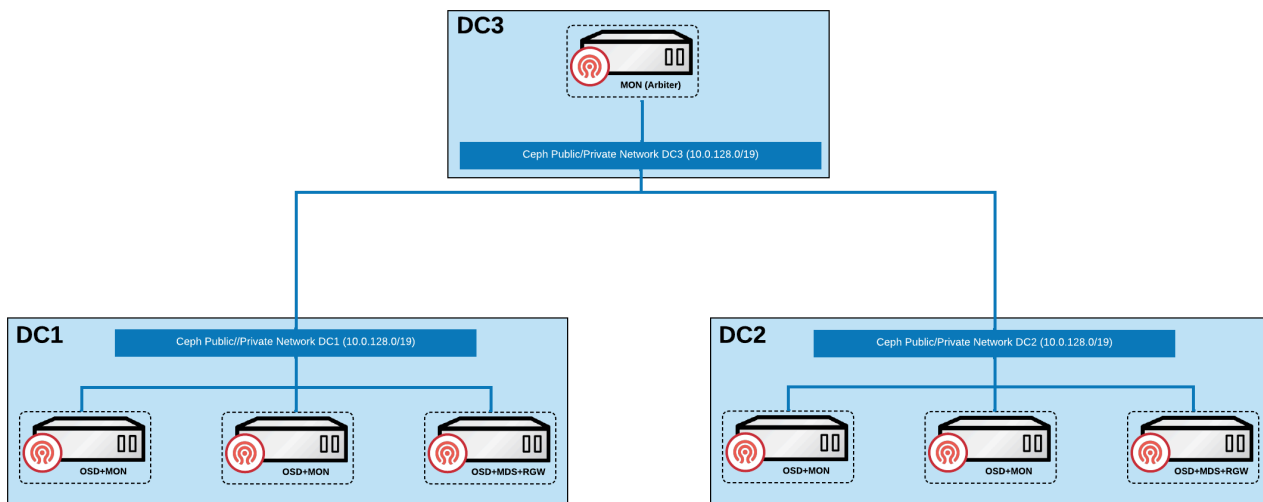
Типовая схема развертывания

Ниже приведена типовая схема развертывания stretch кластеров с описанием компонентов и принципов аварийного восстановления.

Описание компонентов

Узлы необходимо распределить по трём зонам доступности: две зоны data availability и одна зона quorum availability.

- В обеих зонах data availability необходимо полностью развернуть все основные компоненты Ceph (MON, OSD, MGR, MDS, RGW), при этом в каждой зоне data availability должно быть настроено по два экземпляра MON для обеспечения высокой доступности. Если оба MON в одной зоне data availability становятся недоступны, система считает эту зону вышедшей из строя.
- В зоне quorum availability требуется развернуть только один экземпляр MON, который служит узлом для принятия арбитражных решений.



Объяснение аварийного восстановления

- При полном отказе зоны data availability Ceph кластер автоматически переходит в деградированное состояние и генерирует уведомление об аварии. Система изменит минимальное количество реплик в пуле хранения (`min_size`) с дефолтного значения 2 на 1. Поскольку другая зона data availability по-прежнему поддерживает двойное дублирование, кластер остаётся доступным. После восстановления отказавшей зоны система автоматически выполнит синхронизацию данных и вернётся в здоровое состояние; если восстановить зону не удаётся, рекомендуется заменить её на новую зону data availability.

- При разрыве сетевого соединения между двумя зонами data availability, но при сохранении нормального подключения к зоне quorum availability, зона quorum будет арбитражно выбирать между двумя зонами data availability согласно заданным политикам, выбирая зону с лучшим состоянием для продолжения обслуживания в качестве основной зоны данных.

Ограничения и лимиты

- **Ограничения по пулам хранения:** Пулы с кодированием с удалением (erasure-coded) не поддерживаются, допускается только механизм репликации для защиты данных.
- **Ограничения по классификации устройств:** Функциональность device class не поддерживается, стратификация хранения по характеристикам устройств невозможна.
- **Ограничения по региональному развертыванию:** Поддерживаются только две зоны data availability; более двух зон data availability быть не может.
- **Требования к балансировке данных:** Вес OSD в двух зонах data availability должен строго совпадать для обеспечения сбалансированного распределения данных.
- **Требования к носителям хранения:** Разрешена только конфигурация All-Flash OSD, что минимизирует время восстановления после восстановления соединения и максимально снижает риск потери данных.
- **Требования к сетевой задержке:** RTT (время кругового обхода) между двумя зонами data availability не должен превышать 10 мс, а зона quorum availability должна соответствовать требованиям по задержкам спецификации ETCD для обеспечения надёжности арбитражного механизма.

Предварительные условия

Заранее классифицируйте все или часть узлов кластера по трём зонам доступности следующим образом:

- Обеспечьте распределение минимум 5 узлов по одной зоне quorum availability и двум зонам data availability. В зоне quorum availability должен быть минимум один узел,

который может быть виртуальной машиной или облачным хостом.

- Обеспечьте наличие как минимум одного Master-узла (контрольного узла) в одной из трёх зон доступности.
- Обеспечьте равномерное распределение минимум 4 вычислительных узлов по двум зонам data availability, при этом в каждой зоне data availability должно быть не менее 2 вычислительных узлов.
- По возможности обеспечьте одинаковое количество узлов и конфигурации дисков в двух зонах data availability.

Процедура

1 Тегирование узлов

1. Войдите в **Administrator**.
2. В левой навигационной панели выберите **Cluster Management > Cluster**.
3. Нажмите на имя нужного кластера, чтобы перейти на страницу обзора кластера.
4. Перейдите на вкладку **Nodes**.
5. В соответствии с планом из раздела [Предварительные условия](#) добавьте метку `topology.kubernetes.io/zone=<zone>` этим узлам для классификации в указанную зону доступности. Вместо `<zone>` укажите имя зоны доступности.

2 Создание сервиса хранения

В этом документе описываются только параметры, отличающиеся от стандартного типа кластера; по остальным параметрам смотрите [Создание стандартного типа кластера](#).

Создание кластера

Параметр	Описание
Cluster Type	Выберите Stretch .
Quorum Availability Zone	Выберите имя зоны quorum availability.

Параметр	Описание
Data Availability Zone	Выберите имена зон доступности и укажите узлы.

Создание пула хранения

Параметр	Описание
Number of Replicas	По умолчанию 4.
Number of Instances	При типе хранения Object Storage для обеспечения доступности минимальное количество экземпляров — 2, максимальное — 5.

Связанные операции

Создание стандартного типа кластера

Подробности смотрите в [Создание стандартного типа кластера](#).

Очистка распределённого хранилища

Подробности смотрите в [Очистка распределённого хранилища](#).

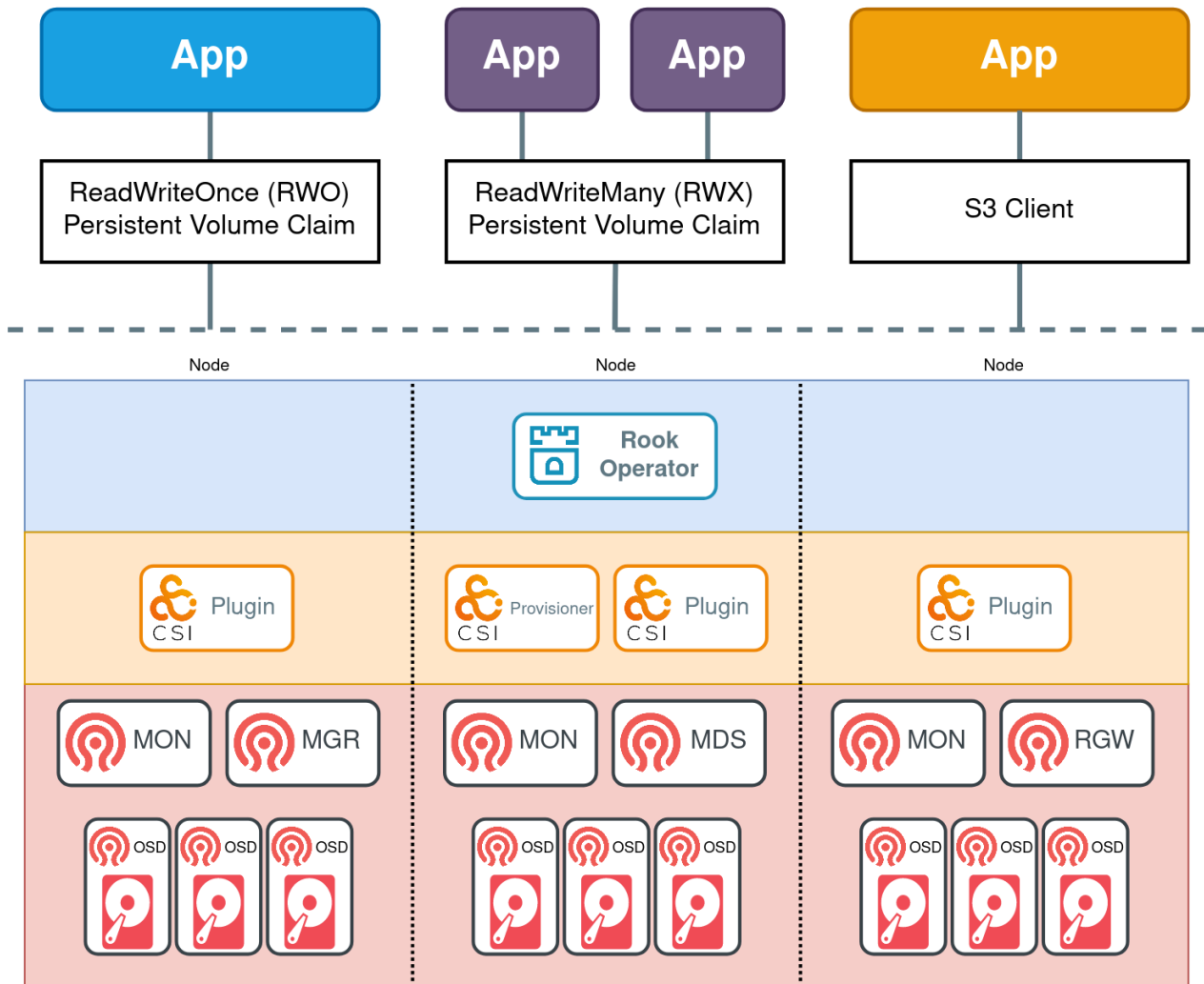
Архитектура

Содержание

| [Техническая архитектура](#)

Техническая архитектура

Rook Architecture



На рисунке выше показаны примерные приложения для трёх поддерживаемых типов хранилищ:

- Блочное хранилище представлено синим приложением, к которому подключён том с режимом ReadWriteOnce (RWO). Приложение может читать и записывать данные в том RWO, при этом Ceph управляет вводом-выводом.
- Общая файловая система представлена двумя фиолетовыми приложениями, которые совместно используют том с режимом ReadWriteMany (RWX). Оба приложения могут одновременно читать и записывать данные в том. Ceph обеспечивает безопасную защиту данных для нескольких писателей с помощью демона MDS.

- Объектное хранилище представлено оранжевым приложением, которое может читать и записывать данные в бакет с помощью стандартного клиента S3.

Ниже пунктирной линии на диаграмме компоненты разделены на три категории:

- **Rook operator** (синий слой): оператор автоматизирует конфигурацию Ceph
- **CSI plugins and provisioners** (оранжевый слой): драйвер Ceph-CSI обеспечивает provision и монтирование томов
- **Ceph daemons** (красный слой): демоны Ceph запускают основную архитектуру хранилища. Подробнее о каждом демоне см. в Глоссарии.

Блочное хранилище

На диаграмме выше процесс создания приложения с томом RWO следующий:

- (синее) приложение создаёт PVC для запроса хранилища.
- PVC определяет класс хранения Ceph RBD (sc) для provision хранилища.
- K8s вызывает provisioner Ceph-CSI RBD для создания образа Ceph RBD.
- kubelet вызывает CSI RBD volume plugin для монтирования тома в приложении.
- Том теперь доступен для чтения и записи.
- Том ReadWriteOnce может быть смонтирован только на одном узле одновременно.

Общая файловая система

На диаграмме выше процесс создания приложений с томом RWX следующий:

- (фиолетовое) приложение создаёт PVC для запроса хранилища.
- PVC определяет класс хранения CephFS (sc) для provision хранилища.
- K8s вызывает provisioner Ceph-CSI CephFS для создания подтома CephFS.
- kubelet вызывает CSI CephFS volume plugin для монтирования тома в приложении.
- Том теперь доступен для чтения и записи.
- Том ReadWriteMany может быть смонтирован на нескольких узлах для использования вашим приложением.

Объектное хранилище S3

На диаграмме выше процесс создания приложения с доступом к бакету S3 следующий:

- (оранжевое) приложение создаёт BucketClaim для запроса бакета.

- Ceph COSI Driver создаёт бакет Ceph RGW.
- Ceph COSI Driver создаёт секрет с учётными данными для доступа к бакету.
- Приложение получает учётные данные из секрета.
- Приложение теперь может читать и записывать данные в бакет с помощью клиента S3.

Основные концепции

Основные концепции

Rook Operator

Ceph CSI

Функции модулей Ceph

Основные концепции

Содержание

[Rook Operator](#)

Сeph CSI

Функции модулей Сeph

Rook Operator

Оператор Rook — это простой контейнер, который содержит всё необходимое для инициализации и мониторинга кластера хранения. Оператор запускает и контролирует поды Сeph monitor, демоны Сeph OSD для обеспечения хранения RADOS, а также запускает и управляет другими демонами Сeph. Оператор управляет CRD для пулов, объектных хранилищ (S3/Swift) и файловых систем, инициализируя поды и другие ресурсы, необходимые для работы сервисов.

Оператор следит за демонами хранения, чтобы обеспечить здоровье кластера. Сeph mons запускаются или переключаются при необходимости, а также вносятся другие корректировки по мере роста или уменьшения кластера. Оператор также отслеживает изменения желаемого состояния, указанные в пользовательских ресурсах Сeph (CR), и применяет эти изменения.

Rook автоматически настраивает драйвер Сeph-CSI для монтирования хранилища в ваши поды. Образ rook/ceph включает все необходимые инструменты для управления кластером.

Ceph CSI

Плагины Ceph CSI реализуют интерфейс между CSI-совместимым контейнерным оркестратором (CO) и кластерами Ceph. Они обеспечивают динамическое выделение томов Ceph и их подключение к рабочим нагрузкам.

Функции модулей Ceph

Модуль	Функция
MON	Монитор (MON) — самый важный компонент в кластере Ceph. Он управляет кластером Ceph и поддерживает статус всего кластера. MON обеспечивает синхронизацию связанных компонентов кластера одновременно. Он функционирует как лидер кластера и отвечает за сбор, обновление и публикацию информации о кластере.
MGR	Менеджер (MGR) — это система мониторинга, которая обеспечивает сбор, хранение, анализ (включая оповещения) и визуализацию данных. Он делает определённые параметры кластера доступными для внешних систем.
OSD	Демоны объектного хранения (OSD) хранят фактические пользовательские данные. Каждый OSD обычно привязан к одному физическому диску. OSD обрабатывают запросы на чтение/запись от клиентов.
MDS	Сервер метаданных Ceph (MDS) отслеживает иерархию файлов и хранит метаданные, используемые только для CephFS. RBD и RGW не требуют метаданных. MDS не предоставляет клиентам прямых сервисов данных.
RGW	Шлюз RADOS (RGW) — это объектный шлюз Ceph, который предоставляет RESTful API, совместимые с S3 и Swift. RGW также поддерживает мультиарендность и сервис идентификации OpenStack (Keystone).
RADOS	Надёжное автономное распределённое объектное хранилище (RADOS) — это ядро кластера хранения Ceph. Всё в Ceph хранится в RADOS в виде объектов независимо от типа данных. Слой RADOS обеспечивает согласованность и надёжность данных через репликацию, обнаружение и восстановление сбоев, а также восстановление данных между узлами кластера.

Модуль	Функция
LIBRADOS	<p>Librados — это метод, упрощающий доступ к RADOS. В настоящее время поддерживаются языки программирования PHP, Ruby, Java, Python, C и C++. Он предоставляет локальный интерфейс RADOS для кластера хранения Ceph и является базовым компонентом других сервисов, таких как блочное устройство RADOS (RBD) и шлюз RADOS (RGW). Кроме того, он предоставляет интерфейс Portable Operating System Interface (POSIX) для файловой системы Ceph (CephFS). API Librados можно использовать для прямого доступа к RADOS, что позволяет разработчикам создавать собственные интерфейсы для доступа к хранилищу кластера Ceph.</p>
RBD	<p>Блочное устройство RADOS (RBD) — это блочное устройство Ceph, которое предоставляет блочное хранилище для внешних систем. Его можно отображать, форматировать и монтировать как диск на сервере.</p>
CephFS	<p>CephFS предоставляет распределённую файловую систему, совместимую с POSIX, любого размера. Она зависит от Ceph MDS для отслеживания иерархии файлов, то есть метаданных.</p>

Руководства

Доступ к сервисам хранения

Предварительные требования

Процедура

Последующие действия

Управление Storage Pools

Создание Storage Pool

Удаление Storage Pool

Просмотр адресов Object Storage Pool

Развертывание

Обновление кс

Перезапуск ко

Добавление устройств/классс

Добавление классов устройств

Добавление устройств

Статус жесткого диска

Мониторинг и оповещения

Мониторинг

Оповещения

Доступ к сервисам хранения

Доступ к сервисам хранения поддерживает два метода интеграции: первый — интеграция распределённых ресурсов хранения из других бизнес-кластеров внутри платформы для обеспечения изоляции хранения и бизнеса, что облегчает управление и обслуживание; второй — подключение внешних ресурсов S3 для использования распределённого хранения.

Содержание

Предварительные требования

- Подготовка пакета

- Подготовка хранилища

- Открытие портов

- Получение данных аутентификации (внешний S3)

Процедура

- Развертывание Alauda Container Platform Storage Essentials

- Доступ к хранилищу

- Последующие действия

Предварительные требования

Подготовка пакета

- **Скачайте** установочный пакет **Alauda Container Platform Storage Essentials**, соответствующий архитектуре вашей платформы.
- **Загрузите** установочный пакет **Alauda Container Platform Storage Essentials** с помощью механизма Upload Packages.
- **Скачайте** установочный пакет **Alauda Build of Rook-Ceph**, соответствующий архитектуре вашей платформы.
- **Загрузите** установочный пакет **Alauda Build of Rook-Ceph** с помощью механизма Upload Packages.

Подготовка хранилища

Выберите один из вариантов:

- В других бизнес-кластерах уже развернуто распределённое хранилище и создан storage pool. Запишите название storage pool для последующей интеграции.
- Вне платформы создано внешнее хранилище Ceph (версия $\geq 14.2.3$) с созданным storage pool. Запишите название storage pool для последующей интеграции.

Открытие портов

IP назначения	Порты назначения	IP источника	Порт источника
IP узла Ceph	3300, 6789, 6800-7300, 7480	IP всех узлов бизнес-кластера	любой

Получение данных аутентификации (внешний Ceph)

Если подготовленное хранилище — внешний Ceph, необходимо получить данные аутентификации с помощью следующих команд.

Параметр	Способ получения
FSID	<code>ceph fsid</code>

Параметр	Способ получения
Информация о компоненте MON	<pre>ceph mon dump</pre> <p>Должна быть в формате {name= IP}, например <code>a=192.168.100.100:6789</code>.</p>
Административный ключ	<pre>ceph auth get-key client.admin</pre>
Storage Pool	<ul style="list-style-type: none"> Файловое хранилище: используйте команду <code>ceph fs ls</code> для получения значения <code>name</code>. Блочное хранилище: <code>ceph osd dump grep "application rbd" awk '{print \$3}'</code>
<i>Storage Pool для данных</i>	<p>(только для файлового хранилища) используйте команду <code>ceph fs ls</code> для получения значения <code>data pools</code>.</p>

Процедура

Примечание: В следующих шагах в качестве примера рассматривается **доступ к внешнему Ceph-хранилищу**, операции для доступа к распределённому хранилищу аналогичны.

1 Развертывание Alauda Container Platform Storage Essentials

1. Войдите в систему, перейдите на страницу **Administrator**.
2. Нажмите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите **Alauda Container Platform Storage Essentials**, нажмите **Install** и перейдите на страницу **Install Alauda Container Platform Storage Essentials**.
Параметры конфигурации:

Параметр	Рекомендуемая конфигурация
Channel	Канал по умолчанию — <code>stable</code> .
Installation Mode	<code>Cluster</code> : Все пространства имён в кластере используют один экземпляр Operator для создания и управления, что снижает использование ресурсов.
Installation Place	Выберите <code>Recommended</code> , Namespace поддерживается только <code>acp-storage</code> .
Upgrade Strategy	<code>Manual</code> : При появлении новой версии в Operator Hub требуется ручное подтверждение для обновления Operator до последней версии.

2

Доступ к хранилищу

1. В левой навигационной панели нажмите **Storage Management > Distributed Storage**.
2. Нажмите **Access Storage**.
3. На странице мастера **Access Configuration** выберите **External Ceph**.

Параметр	Описание
Snapshot	<p>При включении поддерживается создание снимков PVC и использование снимков для настройки новых PVC, что обеспечивает быстрое резервное копирование и восстановление бизнес-данных.</p> <p>Если при доступе к хранилищу снимки не были включены, их можно включить позже в разделе Operations на странице деталей кластера хранения по мере необходимости.</p> <p>Примечание: Перед использованием убедитесь, что для текущего кластера развернут плагин volume snapshot.</p>

Параметр	Описание
Network Configuration	<ul style="list-style-type: none"> • Host Network: Вычислительные компоненты в этом кластере будут обращаться к кластеру хранения через host network. • Container Network: Вычислительные компоненты в этом кластере будут обращаться к кластеру хранения через container network. Можно создать подсеть в управлении сетью и назначить её пространству имён <code>rook-ceph</code>. Если оставить пустым, будет использоваться подсеть по умолчанию.
Другие параметры	Заполните параметры аутентификации для внешнего Ceph, полученные на этапе предварительных требований.

4. На странице мастера **Create Storage Class** завершите конфигурацию и нажмите **Access**.

Параметр	Описание
Type	<p>В зависимости от типа созданного выше storage pool, по умолчанию будет соответствующий класс хранения:</p> <ul style="list-style-type: none"> • Файловое хранилище: CephFS File Storage • Блочное хранилище: CephRBD Block Storage
Reclaim Policy	<p>Политика восстановления для persistent volumes.</p> <ul style="list-style-type: none"> • Delete: при удалении persistent volume claim будет также удалён связанный persistent volume. • Retain: даже при удалении persistent volume claim связанный persistent volume сохраняется.

Параметр	Описание
Project	Проекты, которые могут использовать этот тип хранилища.
Allocation	Если в данный момент нет проектов, требующих этот тип хранилища, можно не выделять проекты сейчас и обновить их позже.

5. Дождитесь успешной интеграции, это займет примерно 1-5 минут.

Последующие действия

- Создание классов хранения: [CephFS File Storage](#), [CephRBD Block Storage](#)
- Разработчики, использующие вышеуказанные классы хранения для создания persistent volume claims, могут расширить функционал с помощью снимков томов и масштабирования.

Примечание: Если необходимо поддерживать storage pools, конфигурации устройств хранения и т.п. для внешнего хранилища, операции должны выполняться в управляющей платформе кластера хранения.

Управление Storage Pools

Storage pool — это логический раздел, используемый для хранения данных. Один кластер хранения поддерживает одновременное использование различных типов storage pools, таких как файловое хранилище и блочное хранилище, чтобы удовлетворить разные бизнес-требования.

Содержание

[Создание Storage Pool](#)

Порядок действий

[Удаление Storage Pool](#)

Порядок действий

[Просмотр адресов Object Storage Pool](#)

Порядок действий

Создание Storage Pool

Помимо storage pools, созданных при настройке распределённого хранилища, вы можете создавать дополнительные типы storage pools.

Совет: В рамках одного кластера хранения допускается только один файловый storage pool и один объектный storage pool, при этом можно создать до восьми блочных storage pools.

Порядок действий

1. Перейдите в **Administrator**.
2. В левой навигационной панели выберите **Storage Management > Distributed Storage**.
3. На вкладке **Cluster Information** прокрутите вниз до области **Storage Pool** и нажмите **Create Storage Pool**.
4. Настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Storage Type	<p>Выберите тип хранилища, который ещё не развернут.</p> <ul style="list-style-type: none"> - File Storage: Обеспечивает безопасные, надёжные и масштабируемые услуги совместного файлового хранения. Подходит для совместного использования файлов, резервного копирования данных и т.д. - Block Storage: Обеспечивает хранилище с высокой производительностью IOPS и низкой задержкой. Подходит для баз данных, виртуализации и т.п. - Object Storage: Обеспечивает хранилище с интерфейсом S3, подходит для больших данных, архивного резервного копирования, облачных сервисов и т.д.
Replica Count	<ul style="list-style-type: none"> • Для кластера типа Standard: Большое количество реплик повышает отказоустойчивость и безопасность данных, но снижает эффективность использования хранилища. Обычно достаточно значения 3. • Для кластера типа Extended: Значение реплик по умолчанию — 4, изменить нельзя.
Device Class	<ul style="list-style-type: none"> • Для кластера типа Standard: Выберите уже добавленный класс устройств в созданном storage pool.

Параметр	Описание
	<ul style="list-style-type: none"> • При выборе класса устройств данные будут храниться в выбранном классе устройств. • Если класс устройств не выбран, данные будут случайным образом распределены по всем устройствам в storage pool. • Для кластера типа Extended: Добавление класса устройств не поддерживается.

Если выбран тип object storage, можно также настроить следующие параметры:

Параметр	Описание
Region	Укажите регион, в котором расположен storage pool.
Gateway Type	По умолчанию S3, изменить нельзя.
Internal Port	Укажите порт для внутреннего доступа к кластеру.
External Access	Включение/отключение внешнего доступа создаст/удалит сервис типа NodePort.
Instance Count	Количество экземпляров ресурсов для object storage.


5. Нажмите **Create**.

Удаление Storage Pool

Если определённый тип хранилища больше не нужен, storage pool можно удалить после отвязки от storage class.

Порядок действий

1. Перейдите в **Administrator**.

2. В левой навигационной панели выберите **Storage Management > Distributed Storage**.
3. На вкладке **Cluster Information** прокрутите вниз до области **Storage Pool**, нажмите  рядом с нужным storage pool и выберите **Delete**.
4. Ознакомьтесь с информацией в подсказке и введите имя storage pool.
5. Нажмите **Delete**.

Просмотр адресов Object Storage Pool

После создания object storage pool вы можете просмотреть адреса внутреннего и внешнего доступа к storage pool.

Порядок действий

1. Перейдите в **Administrator**.
2. В левой навигационной панели выберите **Storage Management > Distributed Storage**.
3. На вкладке **Cluster Information** прокрутите вниз до области **Storage Pool**, нажмите  рядом с object storage pool и выберите **View Address**.

Развертывание компонентов на конкретных узлах

После создания распределённого хранилища вы можете просматривать и изменять место развертывания компонентов, что облегчает расширение и обслуживание хранилища.

Содержание

[Обновление конфигурации развертывания компонентов](#)

Меры предосторожности

Порядок действий

Перезапуск компонентов хранилища

Порядок действий

Обновление конфигурации развертывания компонентов

Меры предосторожности

- Обновление конфигурации вызовет автоматическую пересборку экземпляров компонентов системой, что может повлиять на доступ к сервисам хранилища. Рекомендуется выполнять обновление в непиковое время.

- При типе кластера **Extend** функция фиксированного развертывания компонентов не поддерживается.

Порядок действий

1. Перейдите в раздел **Administrator**.
2. В левой навигационной панели нажмите **Storage Management > Distributed Storage**.
3. На вкладке **Storage Components** нажмите **Component Deployment Configuration**.
4. Включите/отключите переключатель **Fixed Deployment** в соответствии с бизнес-требованиями и разверните компоненты на указанных узлах. Количество узлов должно быть не менее трёх для обеспечения минимальной доступности. Компоненты, для которых доступна настройка фиксированного развертывания, включают MON, MGR, MDS, RGW.
5. Нажмите **Update**, после чего компоненты начнут планироваться на указанные узлы.

Перезапуск компонентов хранилища

При удалении развернутых компонентов хранилища система автоматически переназначит и повторно развернёт компоненты на узлах в соответствии с текущей стратегией развертывания компонентов.

Порядок действий

1. Перейдите в раздел **Administrator**.
2. В левой навигационной панели нажмите **Storage Management > Distributed Storage**.
3. На вкладке **Storage Components** нажмите **:** рядом с названием компонента **> Delete**.

Добавление устройств/классов устройств

Содержание

[Добавление классов устройств](#)

Примечания

Процедура

Добавление устройств

Процедура

Статус жесткого диска

Добавление классов устройств

Объедините классификацию устройств одного типа или жестких дисков с одинаковой бизнес-логикой в узлах кластера, настройте классы устройств в соответствии с требованиями хранения и распределите различные содержимые хранилища по разным типам дисков.

Примечания

Добавление классов устройств не поддерживается, если тип кластера — **Extend**.

Процедура

1. Войдите в **Administrator**.
2. В левой навигационной панели нажмите **Storage Management > Distributed Storage**.
3. Перейдите на вкладку **Device Classes**.
4. Нажмите **Add Device Class** и настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Name	Имя класса устройств. Следующие имена нельзя использовать для класса устройств: <code>hdd</code> , <code>ssd</code> , <code>nvme</code> .
Storage Devices	<p>Для добавления устройств хранения в класс устройств можно выбрать методы <code>Selection Device</code> и <code>Input Device</code>:</p> <ul style="list-style-type: none"> • Selection Device: Выберите из доступных устройств хранения. Устройство считается доступным, если оно соответствует следующим критериям: <ul style="list-style-type: none"> • Тип устройства — <code>disk</code> или <code>mpath</code> • Файловая система не обнаружена (<code>fsType</code> пустой) • Емкость превышает 10 GiB <p>Устройства, такие как <code>rbd</code>, <code>nbd</code> и <code>dm-*</code>, не будут отображаться в списке доступных для выбора устройств.</p> <p>Примечание: Требуется предварительное развертывание <code>Alauda Build LocalStorage Operator</code>.</p> • Input Device: Вручную введите имена пустых устройств на узле, например <code>sda</code>. <p>Примечание: Для оптимальной производительности и управления настоятельно рекомендуется использовать</p>

Параметр	Описание
	необработанные диски в качестве устройств хранения вместо отдельных разделов на диске.

Добавление устройств

Отобразите доступные жесткие диски как устройства хранения для использования и управления.

Примечание: После добавления жестких дисков в качестве устройств хранения обновление или удаление через интерфейс не поддерживается.

Процедура

1. Войдите в **Administrator**.
2. В левой навигационной панели нажмите **Storage Management > Distributed Storage**.
3. Перейдите на вкладку **Device Classes**.
4. Справа от класса устройств нажмите **Add Device** и настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Specified Disks	<p>Для добавления устройств хранения в класс устройств можно выбрать методы <code>Selection Device</code> и <code>Input Device</code> :</p> <ul style="list-style-type: none"> • Selection Device: Выберите из доступных устройств хранения. Устройство считается доступным, если оно соответствует следующим критериям: <ul style="list-style-type: none"> • Тип устройства — disk или mpath • Файловая система не обнаружена (fsType пустой)

Параметр	Описание
	<ul style="list-style-type: none">• Емкость превышает 10 GiB <p>Устройства, такие как rbd, nbd и dm-*, не будут отображаться в списке доступных для выбора устройств.</p> <p>Примечание: Требуется предварительное развертывание Alauda Build LocalStorage Operator.</p> <ul style="list-style-type: none">• Input Device: Вручную введите имена пустых устройств на узле, например <code>sda</code> . <p>Примечание: Для оптимальной производительности и управления настоятельно рекомендуется использовать необработанные диски в качестве устройств хранения вместо отдельных разделов на диске.</p>

5. Нажмите **Add**.

Статус жесткого диска

- **Normal:** Соответствующий статус устройства хранения — IN+UP.
- **Abnormal:** Соответствующий статус устройства хранения — IN+DOWN.
- **Offline:** Соответствующий статус устройства хранения — OUT+UP.
- **Fault:** Соответствующий статус устройства хранения — OUT+DOWN.

Мониторинг и оповещения

Распределённое хранилище предоставляет встроенные возможности сбора метрик мониторинга и уведомлений об оповещениях. После включения функций мониторинга и оповещений вы можете отслеживать и получать оповещения по таким аспектам, как кластер хранения, производительность хранилища и компоненты хранилища, с поддержкой настройки стратегий уведомлений.

Интуитивно представленные данные мониторинга могут использоваться для поддержки принятия решений при проведении инспекций эксплуатации и обслуживания или настройке производительности, а комплексный механизм оповещений и уведомлений поможет обеспечить стабильную работу системы хранения.

Совет: Если функции мониторинга и оповещений не были включены при создании распределённого хранилища, вам потребуется найти альтернативные решения для мониторинга и оповещений хранилища. Например, вручную настроить панели мониторинга и стратегии оповещений в центре эксплуатации и обслуживания.

Содержание

Мониторинг

- Обзор хранилища

- Мониторинг производительности

- Мониторинг компонентов

Оповещения

- Настройка уведомлений

- Обработка оповещений

- Анализ неисправностей

Мониторинг

Платформа автоматически собирает общие метрики мониторинга для распределённого хранилища, такие как производительность чтения и записи, использование CPU и памяти. В разделе **Storage Management > Distributed Storage** на вкладке **Monitoring** вы можете просматривать данные мониторинга в реальном времени по этим метрикам.

Обзор хранилища

Отслеживайте состояние здоровья хранилища, использование физической ёмкости и количество активных компонентов OSD/MON. В случае аномального состояния хранилища вы можете проверить причину оповещения.

Мониторинг производительности

Отслеживайте пропускную способность чтения и записи, а также IOPS чтения и записи с трёх уровней: кластер, пул хранения и OSD. Кроме того, можно мониторить задержки чтения и записи специально для OSD.

Мониторинг компонентов

Отслеживайте использование CPU и памяти таких компонентов, как MON и OSD.

Оповещения

В платформе включён набор стандартных стратегий оповещений. Как только ресурс становится аномальным или данные мониторинга достигают состояния предупреждения, оповещения автоматически срабатывают. Предусмотренные стратегии достаточно для типичных операционных задач, таких как оповещения о состоянии компонентов и кластера, оповещения о ёмкости устройств и оповещения по пользовательским данным.

Настройка уведомлений

Для своевременного получения оповещений рекомендуется настроить стратегии уведомлений в центре эксплуатации и обслуживания: отправлять информацию об оповещениях по электронной почте, SMS и другим каналам соответствующим сотрудникам, напоминая им принять необходимые меры для устранения проблем или предотвращения сбоев. Нажмите **Alert Configuration**, чтобы перейти в центр эксплуатации и обслуживания для завершения настройки, см. [Create Alert Strategies](#).

Обработка оповещений

- Если мониторинг кластера хранения показывает состояние **Warning**, это означает, что сработало оповещение, и связанная аномалия может привести к сбою. Пожалуйста, своевременно проверьте детали в разделе **Real-time Alerts** и выявите и устраните неисправность на основе причины.
- Если мониторинг кластера хранения показывает состояние **Failure**, это указывает на то, что кластер хранения не может работать нормально. Немедленно локализируйте проблему и приступайте к устранению неисправности.

В таблице ниже приведены значения уровней оповещений, используемых в предустановленных стратегиях, которые могут служить вам ориентиром при выработке принципов обработки оповещений.

Уровень оповещения	Значение
Disaster	Ресурс, соответствующий правилу оповещения, вышел из строя, что вызывает прерывание работы платформы, потерю данных и значительное влияние.
Severe	Ресурс, соответствующий правилу оповещения, имеет известные проблемы, которые могут привести к сбоям функций платформы и повлиять на нормальную работу.
Warning	Ресурс, соответствующий правилу оповещения, сталкивается с операционными рисками, которые могут повлиять на нормальную работу сервисов при отсутствии действий.

Анализ неисправностей

В разделе **Alert History** фиксируются все оповещения, которые были сработаны и больше не требуют действий. При проведении анализа неисправностей с использованием истории оповещений для эффективного подведения итогов вам может потребоваться ответить на следующие вопросы.

- Каковы были конкретные аномальные условия во время инцидента.
- Есть ли закономерность в повторяющемся оповещении, можно ли предотвратить его появление в следующий раз.
- Показывает ли временная шкала всплеск оповещений в определённый период; был ли он вызван форс-мажором или операционной ошибкой, требуется ли корректировка плана эксплуатации.

Как сделать

Настройка выделенного кластера для распределённого хранилища

Настройка выделенного кластера для распределённого хранилища

Архитектура

Требования к инфраструктуре

Процедура

Последующие действия

Очистка распределённого хранилища

Очистка распределённого хранилища

Меры предосторожности

Процедура

Восстановление после сбоев

Восстановление после сбоев

Терминология

Конфигурация резервного копирования

Переключение при сбое

Восстановление после сбоев

Терминология

Конфигурация резервного копирования

Плановая миграция

Восстановление после сбоев

Восстановление после сбоев

Терминология

Предварительные требования

Архитектура

Процедуры

Переключение

Возврат на работу

Обновление параметров оптимизации

Обновление параметров оптимизации

Процедура

Создание пользователя Ceph Object Store

Создание пользователя Ceph Object Store

Предварительные требования

Процедура

Установка квот для Storage Pool

Установка квот для Storage Pool

Предварительные требования

Установка квоты пула для File Storage Pool

Установка квоты пула для Block Storage Pool

Установка квоты пула для Object Storage Pool

Проверка квоты пула через Ceph Terminal

Настройка выделенного кластера для распределённого хранилища

Развёртывание выделенного кластера подразумевает использование отдельного кластера для развёртывания распределённого хранилища платформы, при этом другие бизнес-кластеры внутри платформы получают доступ и используют предоставляемые им сервисы хранения через интеграцию.

Для обеспечения производительности и стабильности распределённого хранилища платформы в выделенном кластере развёртываются только основные компоненты платформы и компоненты распределённого хранилища, избегая совместного размещения с другими бизнес-нагрузками. Такой отдельный подход к развёртыванию является рекомендуемой лучшей практикой для распределённого хранилища платформы.

Содержание

Архитектура

- Требования к инфраструктуре

 - Требования к платформе

 - Требования к кластеру

 - Требования к ресурсам

 - Требования к устройствам хранения

 - Требования к типу устройств хранения

 - Планирование ёмкости

 - Мониторинг ёмкости и масштабирование

 - Требования к сети

 - Сетевая изоляция

Требования к скорости сетевых интерфейсов

Процедура

Развертывание Operator

Создание кластера serh

Создание пулов хранения

Создание файлового пула

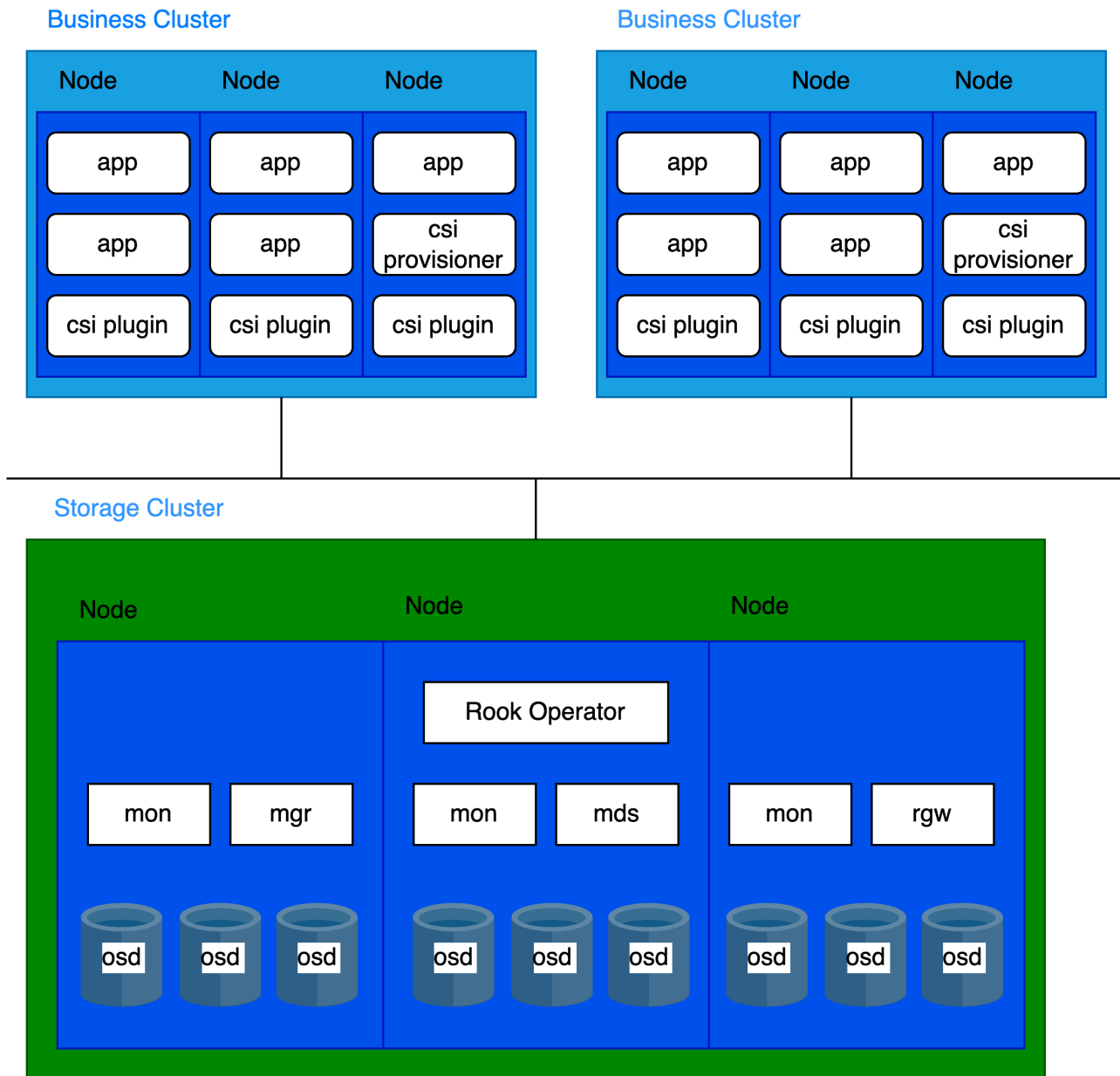
Создание блочного пула

Создание объектного пула

Последующие действия

Архитектура

Архитектура разделения хранения и вычислений



Требования к инфраструктуре

Требования к платформе

Поддерживается в версии 3.18 и выше.

Требования к кластеру

Рекомендуется использовать bare-metal кластеры в качестве выделенных кластеров хранения.

Требования к ресурсам

Пожалуйста, ознакомьтесь с [Основными понятиями](#) для компонентов развертывания распределённого хранилища.

Каждый компонент имеет свои требования к CPU и памяти. Рекомендуемые конфигурации приведены ниже:

Процесс	CPU	Память
MON	2c	3Gi
MGR	3c	4Gi
MDS	3c	8Gi
RGW	2c	4Gi
OSD	4c	8Gi

В кластере обычно запускаются:

- 3 MON
- 2 MGR
- несколько OSD
- 2 MDS (если используется CephFS)
- 2 RGW (если используется CephObjectStorage)

Исходя из распределения компонентов, применимы следующие рекомендации по ресурсам на узел:

CPU	Память
16c + (4c * OSD на узел)	20Gi + (8Gi * OSD на узел)

Требования к устройствам хранения

Рекомендуется развертывать не более 12 устройств хранения на узел. Это помогает ограничить время восстановления после сбоя узла.

Требования к типу устройств хранения

Рекомендуется использовать корпоративные SSD ёмкостью 10TiB или меньше на устройство, при этом все диски должны быть одинакового размера и типа.

Планирование ёмкости

Перед развертыванием ёмкость хранилища должна быть спланирована в соответствии с конкретными бизнес-требованиями. По умолчанию распределённая система хранения использует стратегию избыточности с 3 репликами. Следовательно, полезная ёмкость рассчитывается как общая сырая ёмкость хранения (суммарно по всем устройствам) делённая на 3.

Пример для 30(N) узлов (число реплик = 3), сценарий полезной ёмкости:

Размер устройства хранения(D)	Устройств на узел(M)	Общая ёмкость(DMN)	Полезная ёмкость(DMN/3)
0.5 TiB	3	45 TiB	15 TiB
2 TiB	6	360 TiB	120 TiB
4 TiB	9	1080 TiB	360 TiB

Мониторинг ёмкости и масштабирование

1. Проактивное планирование ёмкости

Всегда следите, чтобы полезная ёмкость хранилища превышала текущие потребности. Если хранилище полностью исчерпано, восстановление требует ручного вмешательства и не может быть решено простым удалением или миграцией данных.

2. Оповещения о ёмкости

Кластер генерирует оповещения при достижении двух порогов:

- **80% использования** ("почти заполнено"): проактивно **освободите место** или масштабируйте кластер.
- **95% использования** ("заполнено"): хранилище полностью исчерпано, стандартные команды не могут освободить место. Немедленно обратитесь в службу поддержки платформы.

Всегда оперативно реагируйте на оповещения и регулярно контролируйте использование хранилища, чтобы избежать простоев.

3. Рекомендации по масштабированию

- **Избегайте:** добавления устройств хранения к существующим узлам.
- **Рекомендуется:** масштабирование путём добавления новых узлов хранения.
- **Требование:** новые узлы должны использовать устройства хранения идентичного размера, типа и количества с существующими узлами.

Требования к сети

Распределённое хранилище должно использовать **HostNetwork**.

Сетевая изоляция

Сеть делится на два типа:

- **Публичная сеть:** используется для взаимодействия клиентов с компонентами хранения (например, I/O запросы).
- **Кластерная сеть:** выделена для репликации данных между репликами и балансировки данных (например, восстановление).

Для обеспечения качества сервиса и стабильности производительности:

1. Для выделенных кластеров хранения:

Зарезервируйте два сетевых интерфейса на каждом хосте:

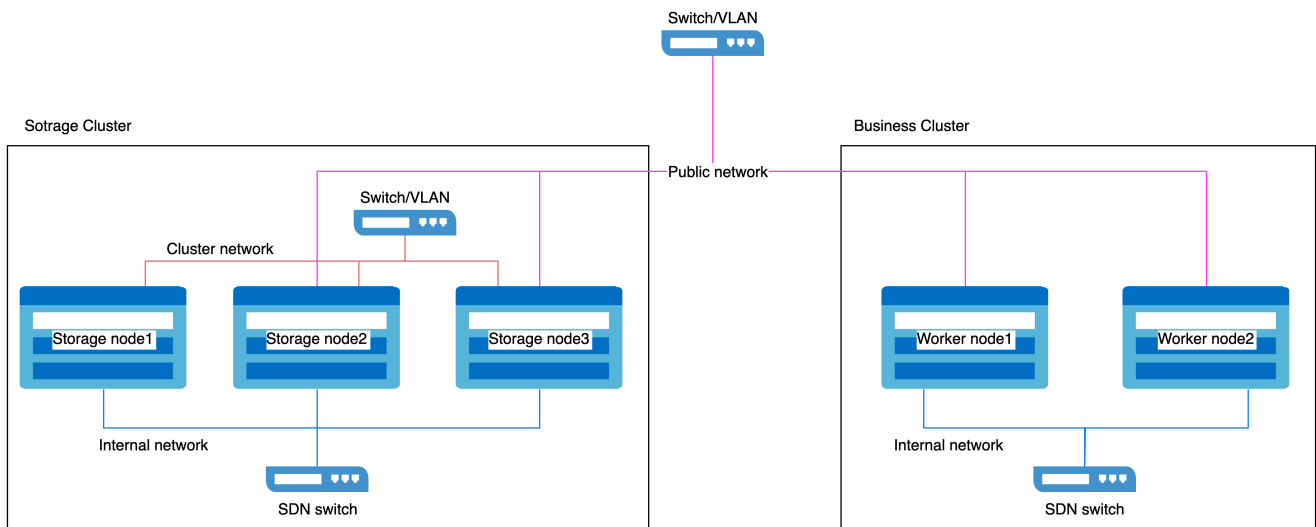
- Публичная сеть: для связи клиентов и компонентов.
- Кластерная сеть: для внутреннего трафика репликации и балансировки.

2. Для бизнес-кластеров:

Зарезервируйте один сетевой интерфейс на каждом хосте для доступа к публичной

сети хранения.

Пример конфигурации сетевой изоляции



Требования к скорости сетевых интерфейсов

1. Узлы хранения

- Сетевые интерфейсы для **Публичной сети** и **Кластерной сети** должны быть 10GbE или выше.

2. Узлы бизнес-кластеров

- Сетевой интерфейс, используемый для доступа к публичной сети хранения, должен быть 10GbE или выше.

Процедура

1 Развертывание Operator

1. Зайдите в **Administrator**.
2. В левой боковой панели нажмите **Storage Management > Distributed Storage**.
3. Нажмите **Create Now**.
4. На странице мастера **Deploy Operator** нажмите кнопку **Deploy Operator** в правом нижнем углу.

- Если страница автоматически перейдёт к следующему шагу, значит Operator успешно развернут.
- Если развертывание не удалось, следуйте подсказке на интерфейсе **Clean Up Deployed Information and Retry** и повторно разверните Operator; если хотите вернуться на страницу выбора распределённого хранилища, нажмите **Application Store**, сначала удалите ресурсы уже развернутого **rook-operator**, затем удалите сам **rook-operator**.

2

Создание кластера ceph

Выполните команды на **контрольном узле** кластера хранения.

- ▶ Нажмите для просмотра

Параметры:

- **public network cidr**: CIDR публичной сети хранения (например, `10.0.1.0/24`).
- **cluster network cidr**: CIDR кластерной сети хранения (например, `10.0.2.0/24`).
- **storage devices**: Укажите устройства хранения, которые будут использоваться распределённым хранилищем.

Пример форматирования:

```

nodes:
- name: storage-node-01
  devices:
  - name: /dev/disk/by-id/wwn-0x5000cca01dd27d60
  useAllDevices: false
- name: storage-node-02
  devices:
  - name: sdb
  - name: sdc
  useAllDevices: false
- name: storage-node-03
  devices:
  - name: sdb
  - name: sdc
  useAllDevices: false

```

Совет

Используется World Wide Name (WWN) диска для стабильного именования, что позволяет избежать зависимости от нестабильных путей устройств, таких как `sdb`, которые могут изменяться после перезагрузок.

3

Создание пулов хранения

Доступны три типа пулов хранения. Выберите и создайте подходящие в соответствии с вашими бизнес-требованиями.

Создание файлового пула

Выполните команды на **контрольном узле** кластера хранения.

- ▶ Нажмите для просмотра

Создание блочного пула

Выполните команды на **контрольном узле** кластера хранения.

- ▶ Нажмите для просмотра

Создание объектного пула

Выполните команды на **контрольном узле** кластера хранения.

- ▶ Нажмите для просмотра

Последующие действия

Когда другие кластеры нуждаются в использовании сервиса распределённого хранилища, руководствуйтесь следующими рекомендациями.

[Доступ к сервисам хранения](#)

Очистка распределённого хранилища

Если необходимо удалить кластер rook-ceph и развернуть новый, следует последовательно очистить ресурсы, связанные с сервисом распределённого хранилища, согласно данной инструкции.

Содержание

Меры предосторожности

Процедура

- Удаление VolumeSnapshotClasses

- Удаление StorageClasses

- Удаление Storage Pools

- Удаление ceph-cluster

- Удаление rook-operator

- Выполнение скрипта очистки

 - Скрипт очистки

 - Меры предосторожности

 - Процедура

Меры предосторожности

Перед очисткой rook-ceph убедитесь, что все ресурсы PVC и PV, использующие хранилище Ceph, удалены.

Процедура

1 Удаление VolumeSnapshotClasses

1. Удалите VolumeSnapshotClasses.

```
kubectl delete VolumeSnapshotClass csi-cephfs-snapshotclass csi-rbd-snapshotclass
```

2. Проверьте, что VolumeSnapshotClasses удалены.

```
kubectl get VolumeSnapshotClass | grep csi-cephfs-snapshotclass  
kubectl get VolumeSnapshotClass | grep csi-rbd-snapshotclass
```

Отсутствие вывода означает, что очистка завершена.

2 Удаление StorageClasses

1. Перейдите в раздел **Administrator**.
2. В левой навигационной панели выберите **Storage Management > Storage Classes**.
3. Нажмите **: > Delete** и удалите все StorageClasses, использующие решения Ceph.

3 Удаление Storage Pools

Этот шаг выполняется после завершения предыдущего.

1. Перейдите в раздел **Administrator**.
2. В левой навигационной панели выберите **Storage Management > Distributed Storage**.
3. В области **Storage Pool Area** нажмите **: > Delete** и удалите все пулы хранения по одному. Когда в области пула хранения отображается сообщение **No Storage Pools**, это означает успешное удаление.
4. (Опционально) Если режим кластера — **Extended**, выполните следующую команду на Master-узле кластера для удаления созданных встроенных пулов

хранения.

```
kubectl -n rook-ceph delete cephblockpool -l cpaas.io/builtin=true
```

Ответ:

```
cephblockpool.ceph.rook.io "builtin-mgr" deleted
```

4

Удаление ceph-cluster

Этот шаг выполняется после завершения предыдущего.

1. Обновите ceph-cluster и включите политику очистки.

```
kubectl -n rook-ceph patch cephcluster ceph-cluster --type merge -p '{"spec":{"cleanupPolicy":{"confirmation":"yes-really-destroy-data"}}}'
```

2. Удалите ceph-cluster.

```
kubectl delete cephcluster ceph-cluster -n rook-ceph
```

3. Удалите задания, выполняющие очистку.

```
kubectl delete jobs --all -n rook-ceph
```

4. Проверьте, что очистка ceph-cluster завершена.

```
kubectl get cephcluster -n rook-ceph | grep ceph-cluster
```

Отсутствие вывода означает, что очистка завершена.

5

Удаление rook-operator

Этот шаг выполняется после завершения предыдущего.

1. Удалите rook-operator.

```
kubectl -n rook-ceph delete subscriptions.operators.coreos.com rook-operator
```

2. Проверьте, что очистка rook-operator завершена.

```
kubectl get subscriptions.operators.coreos.com -n rook-ceph | grep rook-operator
```

Отсутствие вывода означает, что очистка завершена.

3. Проверьте, что все ConfigMaps удалены.

```
kubectl get configmap -n rook-ceph
```

Отсутствие вывода означает, что очистка завершена. Если есть вывод, выполните следующую команду для очистки, заменив `<configmap>` на фактическое имя.

```
kubectl -n rook-ceph patch configmap <configmap> --type merge -p '{"metadata":{"finalizers": []}}'
```

4. Проверьте, что все Secrets удалены.

```
kubectl get secret -n rook-ceph
```

Отсутствие вывода означает, что очистка завершена. Если есть вывод, выполните следующую команду для очистки, заменив `<secret>` на фактическое имя.

```
kubectl -n rook-ceph patch secrets <secret> --type merge -p '{"metadata":{"finalizers": []}}'
```

5. Проверьте, что очистка rook-ceph завершена.

```
kubectl get all -n rook-ceph
```

Отсутствие вывода означает, что очистка завершена.

6

Выполнение скрипта очистки

После выполнения вышеуказанных шагов Kubernetes и ресурсы Ceph будут очищены. Далее необходимо удалить остатки rook-ceph на хосте.

Скрипт очистки

Содержимое скрипта `clean-rook.sh`:

► Нажмите для просмотра

Меры предосторожности

Скрипт очистки зависит от команды `sgdisk`, поэтому убедитесь, что она установлена перед запуском скрипта.

- Команда установки для Ubuntu: `sudo apt install gdisk`
- Команда установки для RedHat или CentOS: `sudo yum install gdisk`

Процедура

1. Запустите скрипт очистки `clean-rook.sh` на каждой машине в бизнес-кластере, где развернуто распределённое хранилище.

```
sh clean-rook.sh /dev/[device_name]
```

Пример: `sh clean-rook.sh /dev/vdb`

При выполнении будет запрошено подтверждение очистки устройства. Для подтверждения введите `yes`.

2. Используйте команду `lsblk -f` для проверки информации о разделах. Если в столбце `FSTYPE` вывод пустой, очистка завершена.

Восстановление после сбоев

Восстановление после сбоев

Терминология

Конфигурация резервного копирования

Переключение при сбое

Восстановление после сбоев

Терминология

Конфигурация резервного копирования

Плановая миграция

Восстановление после сбоев

Восстановление после сбоев

Терминология

Предварительные

Архитектура

Процедуры

Переключение

Возврат на основную

Восстановление после сбоев файлового хранилища

SephFS Mirror — это функция файловой системы Seph, предназначенная для асинхронной репликации данных между разными кластерами Seph, обеспечивая тем самым аварийное восстановление между кластерами. Основная её задача — синхронизация данных в режиме `primary-backup`, что гарантирует быструю передачу управления на резервный кластер в случае сбоя основного.

WARNING

- SephFS Mirror выполняет инкрементальную синхронизацию на основе снимков (snapshots), при этом интервал создания снимков по умолчанию установлен на один раз в час (настраивается). Дифференциальные данные между основным и резервным кластерами обычно составляют объём данных, записанных за один цикл снимка.
- SephFS Mirror обеспечивает только резервное копирование данных базового хранилища и не может выполнять резервное копирование ресурсов Kubernetes. Для резервного копирования PVC и PV ресурсов используйте функцию платформы **Backup and Restore**.

Содержание

Терминология

Конфигурация резервного копирования

Предварительные условия

Процедура

Включите Mirror для пула файлового хранилища в Secondary кластере

Получите Peer Token

Создайте Peer Secret в Primary кластере

Включите Mirror для пула файлового хранилища в Primary кластере

Разверните Mirror Daemon в Primary кластере

Переключение при сбое

Предварительные условия

Терминология

Термин	Объяснение
Primary Cluster	Кластер, который в данный момент предоставляет услуги хранения.
Secondary Cluster	Резервный кластер.

Конфигурация резервного копирования

Предварительные условия

- Подготовьте два кластера, подходящих для развертывания Alauda Build Rook-Ceph, а именно Primary и Secondary, убедившись, что сети между кластерами взаимосвязаны.
- Версии платформы, используемые в обоих кластерах (v3.12 и выше), должны совпадать.
- [Создайте распределённый сервис хранения](#) в обоих кластерах — Primary и Secondary.
- Создайте пулы файлового хранилища с **одинаковыми именами** в обоих кластерах.

Процедура

1

Включите Mirror для пула файлового хранилища в Secondary кластере

Выполните следующие команды на управляющем узле Secondary кластера:

Command Line

```
kubectl -n rook-ceph patch cephfilesystem <fs-name> \
--type merge -p '{"spec":{"mirroring":{"enabled": true}}}'
```

Output

```
cephfilesystem.ceph.rook.io/<fs-name> patched
```

Параметры:

- `<fs-name>` : Имя пула файлового хранилища.

2

Получите Peer Token

Этот токен является ключевым учётным данным для установления зеркального соединения между двумя кластерами.

Выполните следующие команды на управляющем узле Secondary кластера:

Command

```
kubectl get secret -n rook-ceph \
$(kubectl -n rook-ceph get cephfilesystem <fs-name> -o jsonpath
='{.status.info.fsMirrorBootstrapPeerSecretName}') \
-o jsonpath='{.data.token}' | base64 -d
```

Output

```
# Из-за наличия конфиденциальной информации вывод сокращён.
eyJmc2lkIjogImMyYjAyNmMzLTA3ZGQtNDA3Z...
```

Параметры:

- `<fs-name>` : Имя пула файлового хранилища.

3

Создайте Peer Secret в Primary кластере

После получения Peer Token из Secondary кластера необходимо создать Peer Secret в Primary кластере.

Выполните следующие команды на управляющем узле Primary кластера:

Command

```
kubectl -n rook-ceph create secret generic fs-secondary-site-secret \
  --from-literal=token=<token> \
  --from-literal=pool=<fs-name>
```

Output

```
secret/fs-secondary-site-secret created
```

Параметры:

- `<token>` : Токен, полученный на [шаге 2](#).
- `<fs-name>` : Имя пула файлового хранилища.

4

Включите Mirror для пула файлового хранилища в Primary кластере

Выполните следующие команды на управляющем узле Primary кластера:

Command

```
kubectl -n rook-ceph patch cephfilesystem <fs-name> --type merge
-p \
'{
  "spec": {
    "mirroring": {
      "enabled": true,
      "peers": {
        "secretNames": [
          "fs-secondary-site-secret"
        ]
      },
    },
    "snapshotSchedules": [
      {
        "path": "/",
        "interval": "<schedule-interval>"
      }
    ],
    "snapshotRetention": [
      {
        "path": "/",
        "duration": "<retention-policy>"
      }
    ]
  }
}'
```

Sample

```
kubectl -n rook-ceph patch cephfilesystem cephfs --type merge -p \
'{
  "spec": {
    "mirroring": {
      "enabled": true,
      "peers": {
        "secretNames": [
          "fs-secondary-site-secret"
        ]
      },
      "snapshotSchedules": [
        {
          "path": "/",
          "interval": "1h"
        }
      ],
      "snapshotRetention": [
        {
          "path": "/",
          "duration": "h 1"
        }
      ]
    }
  }
}'
```

Output

```
cephfilesystem.ceph.rook.io/<fs-name> patched
```

Параметры:

- `<fs-name>` : Имя пула файлового хранилища.
- `<schedule-interval>` : Интервал выполнения снимков. Подробнее см. в [официальной документации](#) ↗.
- `<retention-policy>` : Политика хранения снимков. Подробнее см. в [официальной документации](#) ↗.

5

Разверните Mirror Демон в Primary кластере

Mirror Демон непрерывно отслеживает изменения данных в пуле файлового хранилища (с включённым Mirror). Он периодически создаёт снимки и отправляет разницы снимков в Secondary кластер по сети.

Выполните следующие команды на управляющем узле Primary кластера:

Command

```
cat << EOF | kubectl apply -f -
apiVersion: ceph.rook.io/v1
kind: CephFilesystemMirror
metadata:
  name: cephfs-mirror
  namespace: rook-ceph
spec:
  placement:
    tolerations:
      - key: NoSchedule
        operator: Exists
  resources:
    limits:
      cpu: "500m"
      memory: "1Gi"
    requests:
      cpu: "500m"
      memory: "1Gi"
  priorityClassName: system-node-critical
EOF
```

Output

```
cephfilesystemmirror.ceph.rook.io/cephfs-mirror created
```

Переключение при сбое

В случае сбоя Primary кластера вы можете напрямую продолжить использование CephFS в Secondary кластере.

Предварительные условия

Ресурсы Kubernetes Primary кластера были забэкаплены и восстановлены в Secondary кластер, включая PVC, PV и рабочие нагрузки приложений.

Восстановление после сбоев блочного хранилища

RBD Mirror — это функция Ceph Block Storage (RBD), которая обеспечивает асинхронную репликацию данных между разными кластерами Ceph, предоставляя межкластерное восстановление после сбоев (Disaster Recovery, DR). Основная функция — синхронизация данных в режиме primary-backup, обеспечивающая быстрое переключение на резервный кластер при сбое основного.

WARNING

- RBD Mirror выполняет инкрементальную синхронизацию на основе снимков (snapshots) с интервалом по умолчанию один раз в час (настраивается). Дифференциальные данные между основным и резервным кластерами обычно соответствуют записям за один цикл снимка.
- RBD Mirror обеспечивает только резервное копирование данных на уровне хранилища и не занимается резервным копированием ресурсов Kubernetes. Для резервного копирования PVC и PV используйте функцию **Backup and Restore** платформы.

Содержание

Терминология

Конфигурация резервного копирования

Предварительные требования

Сетевые требования

Процедуры

Инициализация пиров (`Primary <-> Secondary`)

Настройка окружения для репликации томов

Включение зеркалирования для PVC

Плановая миграция

Перемещение (Relocation)

Предварительные требования

Процедуры

Восстановление после сбоев

Переключение (Failover) при внезапном отключении

Предварительные требования

Процедуры

Возврат (Failback) после восстановления

Предварительные требования

Процедуры

Терминология

Термин	Объяснение
Primary Cluster	Кластер, который в данный момент предоставляет услуги хранения.
Secondary Cluster	Резервный кластер, используемый для бэкапа.

Конфигурация резервного копирования

Предварительные требования

- Подготовьте два кластера, способных развернуть Alauda Build Rook-Ceph: Primary и Secondary.
 - Оба кластера должны работать на одной версии платформы (v3.12 или выше).
-

- [Создайте распределённые сервисы хранения](#) в кластерах Primary и Secondary.
- Создайте пулы блочного хранилища с **одинаковыми именами** в обоих кластерах.
- Убедитесь, что следующие два образа загружены в приватный репозиторий образов платформы:
 - `quay.io/csiaddons/k8s-controller:v0.12.0` -> `<registry>/csiaddons/k8s-controller:v0.12.0`
 - `quay.io/csiaddons/k8s-sidecar:v0.12.0` -> `<registry>/csiaddons/k8s-sidecar:v0.12.0`

Сетевые требования

Синхронизация между Primary и Secondary кластерами выполняется через Public сеть Serp. Поэтому межсайтовая Public сеть должна соответствовать следующим требованиям:

- Каждый кластер должен иметь выделенный сегмент Public сети. Public сети Primary и Secondary должны быть полностью маршрутизируемы и доступны друг для друга.
- Постоянная загрузка пропускной способности Public сети не должна превышать 60%, чтобы обеспечить запас для пиковых нагрузок репликации и сценариев переключения.
- Время отклика (RTT) между двумя сайтами должно быть менее 30 мс.
- Потеря пакетов между двумя Public сетями должна быть ниже 0.05% для стабильной и предсказуемой производительности репликации.

Процедуры

Инициализация пиров (`Primary <-> Secondary`)

1. **Включите зеркалирование для пула блочного хранилища Primary кластера**
Выполните следующую команду на управляющих узлах обоих кластеров:

Command

```
kubectl -n rook-ceph patch cephblockpool <block-pool-name> \
  --type merge -p '{"spec":{"mirroring":{"enabled":true,"mode":"image"}}}'
```

Output

```
cephblockpool.ceph.rook.io/<block-pool-name> patched
```

Параметры:

- `<block-pool-name>`: имя пула блочного хранилища.

2. Получение токена пира

Этот токен является ключевым для установления зеркальных соединений между кластерами.

Выполните следующую команду на управляющих узлах обоих кластеров:

Command

```
kubectl get secret -n rook-ceph \
  $(kubectl get cephblockpool.ceph.rook.io <block-pool-name> -
  n rook-ceph -o jsonpath='{.status.info.rbdMirrorBootstrapPeers
  ecretName}') \
  -o jsonpath='{.data.token}' | base64 -d
```

Output

```
# Вывод сокращён из-за чувствительной информации
eyJmc2lkIjoiMjc2N2I3ZmEtY2YwYi00N...
```

3. Создайте секрет с токеном пира в пировом кластере

Выполните следующую команду на управляющем узле каждого кластера:

Command

```
kubectl -n rook-ceph create secret generic rbd-peer-site-secret \
  --from-literal=token=<token> \
  --from-literal=pool=<block-pool-name>
```

Output

```
secret/rbd-peer-site-secret created
```

Параметры:

- `<token>` : токен, полученный на [Share 2](#).
В Primary кластере укажите токен, полученный из Secondary кластера.
В Secondary кластере укажите токен, полученный из Primary кластера.
- `<block-pool-name>` : имя пула блочного хранилища.

4. Примените патч к секрету пира для пула блочного хранилища

Выполните следующую команду на управляющем узле каждого кластера:

Command

```
kubectl -n rook-ceph patch cephblockpool <block-pool-name> --type merge -p '{
  "spec": {
    "mirroring": {
      "peers": {
        "secretNames": [
          "rbd-peer-site-secret"
        ]
      }
    }
  }
}'
```

Output

```
cephblockpool.ceph.rook.io/<block-pool-name> patched
```

Параметры:

- `<block-pool-name>` : имя пула блочного хранилища.

5. Разверните демон зеркалирования

Этот демон отвечает за мониторинг и управление процессами синхронизации RBD mirror, включая синхронизацию данных и обработку ошибок.

Выполните следующую команду на управляющем узле каждого кластера:

Command

```
cat << EOF | kubectl apply -f -
apiVersion: ceph.rook.io/v1
kind: CephRBDMirror
metadata:
  name: rbd-mirror
  namespace: rook-ceph
spec:
  count: 1
EOF
```

Output

```
cephrbdmirror.ceph.rook.io/rbd-mirror created
```

6. Проверьте статус зеркалирования

Выполните следующую команду на управляющем узле каждого кластера:

Command

```
kubectl get cephblockpools.ceph.rook.io <block-pool-name> -n r
ook-ceph -o jsonpath='{.status.mirroringStatus.summary}'
```

Output

```
# Все статусы "OK" означают нормальную работу
{"daemon_health":"OK","health":"OK","image_health":"OK","state
s":{}}
```

Параметры:

- `<block-pool-name>` : имя пула блочного хранилища.

Настройка окружения для репликации томов

Эта функция позволяет эффективно реплицировать и синхронизировать данные без прерывания работы основных приложений, повышая надёжность и доступность системы.

1. Установите контроллер CsiAddons

Выполните следующие команды на управляющих узлах обоих кластеров:

```
kubectl create -f https://raw.githubusercontent.com/csi-addons/kub
ernetes-csi-addons/v0.12.0/deploy/controller/crds.yaml
kubectl create -f https://raw.githubusercontent.com/csi-addons/kub
ernetes-csi-addons/v0.12.0/deploy/controller/setup-controller.yaml
kubectl create -f https://raw.githubusercontent.com/csi-addons/kub
ernetes-csi-addons/v0.12.0/deploy/controller/rbac.yaml
kubectl create -f https://raw.githubusercontent.com/csi-addons/kub
ernetes-csi-addons/v0.12.0/deploy/controller/csi-addons-config.yam
l

kubectl -n csi-addons-system set image deployment/csi-addons-contr
oller-manager manager=<registry>/csiaddons/k8s-controller:v0.12.0
```

Параметры:

- `<registry>` : адрес реестра платформы.

2. Включите sidecar CsiAddons

Выполните следующие команды на управляющих узлах обоих кластеров:

```
kubectl patch cm rook-ceph-operator-config -n rook-ceph --type json --patch \
'[
  {
    "op": "add",
    "path": "/data/CSI_ENABLE_OMAP_GENERATOR",
    "value": "true"
  },
  {
    "op": "add",
    "path": "/data/CSI_ENABLE_CSIADDONS",
    "value": "true"
  },
  {
    "op": "add",
    "path": "/data/ROOK_CSIADDONS_IMAGE",
    "value": "<registry>/csiaddons/k8s-sidecar:v0.12.0"
  }
]
```

Дождитесь успешного перезапуска всех csi-подов:

```
kubectl get po -n rook-ceph -w | grep csi
```

3. Создайте VolumeReplicationClass

Выполните следующие команды на управляющих узлах обоих кластеров:

Command

```
cat << EOF | kubectl apply -f -
apiVersion: replication.storage.openshift.io/v1alpha1
kind: VolumeReplicationClass
metadata:
  name: rbd-volumereplicationclass
spec:
  provisioner: rook-ceph.rbd.csi.ceph.com
  parameters:
    mirroringMode: snapshot
    schedulingInterval: "<scheduling-interval>" 1
    replication.storage.openshift.io/replication-secret-name:
rook-csi-rbd-provisioner
    replication.storage.openshift.io/replication-secret-namespace: rook-ceph
EOF
```

Output

```
volumereplicationclass.replication.storage.openshift.io/rbd-volumereplicationclass created
```

1 `<scheduling-interval>` : Интервал планирования (например, `schedulingInterval: "1h"` означает выполнение каждый час).

Включение зеркалирования для PVC

Выполните следующую команду на управляющем узле Primary кластера:

Command

```

cat << EOF | kubectl apply -f -
apiVersion: replication.storage.openshift.io/v1alpha1
kind: VolumeReplication
metadata:
  name: <vr-name> ①
  namespace: <namespace> ②
spec:
  autoResync: false
  volumeReplicationClass: rbd-volumereplicationclass
  replicationState: primary
  dataSource:
    apiGroup: ""
    kind: PersistentVolumeClaim
    name: <pvc-name> ③
EOF

```

Output

```

volumereplication.replication.storage.openshift.io/<mirror-pvc-name> created

```

① `<vr-name>` : имя объекта VolumeReplication, рекомендуется совпадать с именем PVC.

② `<namespace>` : namespace, к которому принадлежит VolumeReplication, должен совпадать с namespace PVC.

③ `<pvc-name>` : имя PVC, для которого нужно включить зеркалирование.

Примечание

После включения RBD-образ в Secondary кластере становится доступен только для чтения.

Плановая миграция

Сценарии использования: обслуживание дата-центра, обновление технологий, предотвращение сбоев и т.п.

Перемещение (Relocation)

Операция Relocation — это процесс переключения производства на резервный объект (обычно ваш сайт восстановления) или обратно.

Для перемещения необходимо прекратить доступ к образу на основном сайте. Затем образ должен быть сделан primary на вторичном кластере, чтобы доступ к нему возобновился там.

Предварительные требования

- Ресурсы Kubernetes основного кластера были сохранены и восстановлены в Secondary кластере, включая PVC, PV, рабочие нагрузки приложений и т.п.

Процедуры

Следуйте шагам для плановой миграции нагрузки с Primary кластера на Secondary:

1. Остановите все поды приложений, использующие зеркалируемые PVC на Primary кластере.
2. Обновите VolumeReplication для всех PVC с включённым зеркалированием на Primary кластере.
Установите `spec.replicationState` в значение `secondary`.
3. Создайте VolumeReplication для всех PVC с зеркалированием на Secondary.

Example

```

cat << EOF | kubectl apply -f -
apiVersion: replication.storage.openshift.io/v1alpha1
kind: VolumeReplication
metadata:
  name: <vr-name> 1
  namespace: <namespace> 2
spec:
  autoResync: false
  volumeReplicationClass: rbd-volumereplicationclass
  replicationState: primary
dataSource:
  apiGroup: ""
  kind: PersistentVolumeClaim
  name: <pvc-name> 3
EOF

```

1 `<vr-name>` : имя объекта VolumeReplication, рекомендуется совпадать с именем PVC.

2 `<namespace>` : namespace, к которому принадлежит VolumeReplication, должен совпадать с namespace PVC.

3 `<pvc-name>` : имя PVC, для которого нужно включить зеркалирование.

4. Проверьте статус VolumeReplication CR, чтобы убедиться, что образ помечен как `primary` на вторичном сайте.
5. После того как образ станет `primary`, PVC готов к использованию. Можно масштабировать приложения для работы с этим PVC.

Восстановление после сбоев

Сценарии использования: природные катастрофы, отключения питания, сбои систем и аварии и т.п.

Переключение (Failover) при внезапном отключении

В случае аварийного восстановления создайте VolumeReplication CR на Secondary сайте.

Поскольку связь с Primary сайтом потеряна, оператор автоматически отправляет GRPC-запрос драйверу для принудительного назначения dataSource как `primary` на Secondary сайте.

Предварительные требования

- Ресурсы Kubernetes основного кластера были сохранены и восстановлены в Secondary кластере, включая PVC, PV, рабочие нагрузки приложений и т.п.

Процедуры

1. Создайте VolumeReplication для всех PVC с включённым зеркалированием на Secondary.

Example

```
cat << EOF | kubectl apply -f -
apiVersion: replication.storage.openshift.io/v1alpha1
kind: VolumeReplication
metadata:
  name: <vr-name> 1
  namespace: <namespace> 2
spec:
  autoResync: false
  volumeReplicationClass: rbd-volumereplicationclass
  replicationState: primary
  dataSource:
    apiGroup: ""
    kind: PersistentVolumeClaim
    name: <pvc-name> 3
EOF
```

1 `<vr-name>` : имя объекта VolumeReplication, рекомендуется совпадать с именем PVC.

2 `<namespace>` : namespace, к которому принадлежит VolumeReplication, должен совпадать с namespace PVC.

3 `<pvc-name>` : имя PVC, для которого нужно включить зеркалирование.

2. Проверьте статус VolumeReplication CR, чтобы убедиться, что образ помечен как `primary` на вторичном сайте.
3. После того как образ станет `primary`, PVC готов к использованию. Можно масштабировать приложения для работы с этим PVC.

Возврат (Failback) после восстановления

После восстановления сбойного кластера на основном сайте и при необходимости возврата с Secondary сайта выполните следующие шаги:

Предварительные требования

- Ресурсы Kubernetes основного кластера были сохранены и восстановлены в Secondary кластере, включая PVC, PV, рабочие нагрузки приложений и т.п.

Процедуры

1. Остановите работающие приложения (если есть) на основном сайте. Убедитесь, что все используемые рабочей нагрузкой постоянные тома больше не используются на Primary кластере.
2. Обновите VolumeReplication CR, изменив replicationState с primary на secondary на основном сайте.
3. Остановите приложения на вторичном сайте.
4. Обновите VolumeReplication CR, изменив replicationState с `primary` на `secondary` на вторичном сайте.
5. На основном сайте проверьте, что статус VolumeReplication показывает, что том готов к использованию.
6. После того как том помечен как готовый к использованию, измените replicationState с secondary на primary на основном сайте.
7. Снова масштабируйте приложения на основном сайте.

Восстановление после сбоев в объектном хранилище

Функция Ceph RGW Multi-Site — это механизм асинхронной репликации данных между кластерами, предназначенный для синхронизации данных объектного хранилища между географически распределёнными кластерами Ceph, обеспечивающий высокую доступность (HA) и восстановление после сбоев (DR).

Содержание

Терминология

Предварительные требования

Архитектура

Процедуры

Создание объектного хранилища в Primary Cluster

Настройка внешнего доступа для Primary Zone

Получение `access-key` и `secret-key`

Создание Secondary Zone и настройка синхронизации Realm

Настройка внешнего доступа для Secondary Zone

Проверка статуса синхронизации Ceph Object Storage

Переключение на резервный кластер (Failover)

Процедуры

Возврат на основной кластер (Failback)

Процедуры

Терминология

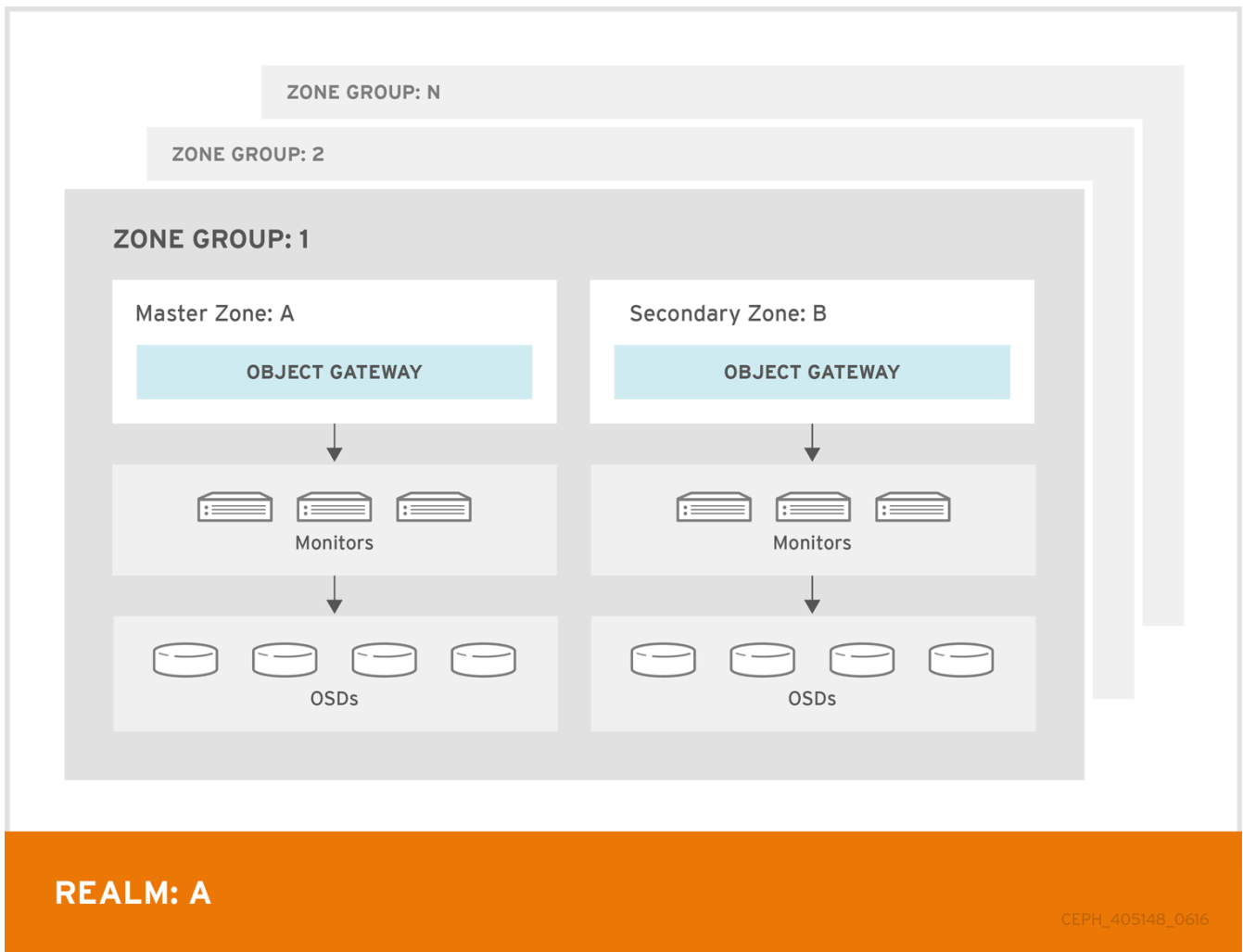
Термин	Объяснение
Primary Cluster	Кластер, который в данный момент предоставляет услуги хранения.
Secondary Cluster	Резервный кластер, используемый для целей резервного копирования.
Realm, ZoneGroup, Zone	<ul style="list-style-type: none"> • Realm: Самая верхняя логическая группа в объектном хранилище Ceph. Представляет собой полное пространство имён объектного хранилища, обычно используется для мультисайтовой репликации и синхронизации. Realm может охватывать разные географические локации или дата-центры. • ZoneGroup: Логическая группа внутри Realm, содержащая несколько Zones. ZoneGroups обеспечивают синхронизацию и репликацию данных между Zones, обычно в пределах одного географического региона. • Zone: Логическая группа внутри ZoneGroup, которая физически хранит данные. Каждая Zone управляет и хранит объекты независимо и может иметь собственные конфигурации пулов данных и метаданных.

Предварительные требования

- Подготовьте два кластера для развертывания Rook-Ceph (Primary и Secondary) с сетевым соединением между ними.
- Оба кластера должны использовать одну и ту же версию платформы (v3.12 или новее).
- Убедитесь, что на кластерах Primary и Secondary не развернуто объектное хранилище Ceph.
- Обратитесь к документации [Create Storage Service](#) для развертывания Operator и создания кластеров. После создания кластера не создавайте пулы объектного

хранилища через мастер. Вместо этого используйте CLI-инструменты для настройки, как описано ниже.

Архитектура



Процедуры

В этом руководстве описано решение для синхронизации между двумя Zones в одном ZoneGroup.

1

Создание объектного хранилища в Primary Cluster

На этом шаге создаются Realm, ZoneGroup, Primary Zone и ресурсы шлюза Primary Zone.

Выполните следующие команды на управляющем узле Primary кластера:

1. Установите параметры

```
export REALM_NAME=<realm-name>
export ZONE_GROUP_NAME=<zonegroup-name>
export PRIMARY_ZONE_NAME=<primary-zone-name>
export PRIMARY_OBJECT_STORE_NAME=<primary-object-store-name>
```

Описание параметров:

- `<realm-name>` : имя Realm.
- `<zonegroup-name>` : имя ZoneGroup.
- `<primary-zone-name>` : имя Primary Zone.
- `<primary-object-store-name>` : имя шлюза.

2. Создайте объектное хранилище

Command

```
cat << EOF | kubectl apply -f -
---
apiVersion: ceph.rook.io/v1
kind: CephObjectRealm
metadata:
  name: $REALM_NAME
  namespace: rook-ceph

---
apiVersion: ceph.rook.io/v1
kind: CephObjectZoneGroup
metadata:
  name: $ZONE_GROUP_NAME
  namespace: rook-ceph
spec:
  realm: $REALM_NAME

---
apiVersion: ceph.rook.io/v1
kind: CephObjectZone
metadata:
  name: $PRIMARY_ZONE_NAME
  namespace: rook-ceph
spec:
  zoneGroup: $ZONE_GROUP_NAME
  metadataPool:
    failureDomain: host
    replicated:
      size: 3
      requireSafeReplicaSize: true
  dataPool:
    failureDomain: host
    replicated:
      size: 3
      requireSafeReplicaSize: true
    parameters:
      compression_mode: none
  preservePoolsOnDelete: false

---
apiVersion: ceph.rook.io/v1
kind: CephBlockPool
metadata:
```

```
labels:
  cpaas.io/builtin: "true"
name: builtin-rgw-root
namespace: rook-ceph
spec:
  name: .rgw.root
  application: rgw
  enableCrushUpdates: true
  failureDomain: host
  replicated:
    size: 3
  parameters:
    pg_num: "8"

---
apiVersion: ceph.rook.io/v1
kind: CephObjectStore
metadata:
  name: $PRIMARY_OBJECT_STORE_NAME
  namespace: rook-ceph
spec:
  gateway:
    port: 7480
    instances: 2
  zone:
    name: $PRIMARY_ZONE_NAME
EOF
```

Output

```
cephobjectrealm.ceph.rook.io/<realm-name> created
cephobjectzonegroup.ceph.rook.io/<zonegroup-name> created
cephobjectzone.ceph.rook.io/<primary-zone-name> created
cephobjectstore.ceph.rook.io/<primary-object-store-name> creat
ed
```

2

Настройка внешнего доступа для Primary Zone

1. Получите UID ObjectStore

```
export PRIMARY_OBJECT_STORE_UID=$(kubectl -n rook-ceph get cephobj  
ectstore $PRIMARY_OBJECT_STORE_NAME -o jsonpath='{.metadata.uid}')
```

2. Создайте Service для внешнего доступа

```
cat << EOF | kubectl apply -f -  
apiVersion: v1  
kind: Service  
metadata:  
  name: rook-ceph-rgw-$PRIMARY_OBJECT_STORE_NAME-external  
  namespace: rook-ceph  
  labels:  
    app: rook-ceph-rgw  
    rook_cluster: rook-ceph  
    rook_object_store: $PRIMARY_OBJECT_STORE_NAME  
ownerReferences:  
  - apiVersion: ceph.rook.io/v1  
    kind: CephObjectStore  
    name: $PRIMARY_OBJECT_STORE_NAME  
    uid: $PRIMARY_OBJECT_STORE_UID  
spec:  
  ports:  
    - name: rgw  
      port: 7480  
      targetPort: 7480  
      protocol: TCP  
  selector:  
    app: rook-ceph-rgw  
    rook_cluster: rook-ceph  
    rook_object_store: $PRIMARY_OBJECT_STORE_NAME  
  sessionAffinity: None  
  type: NodePort  
EOF
```

3. Добавьте внешние конечные точки в CephObjectZone.

```

IP=$(kubectl get nodes -l 'node-role.kubernetes.io/control-plane'
-o jsonpath='{.items[0].status.addresses[?(@.type=="InternalIP")].
address}' | cut -d ' ' -f1 | tr -d '\n')
PORT=$(kubectl -n rook-ceph get svc rook-ceph-rgw-$PRIMARY_OBJECT_
STORE_NAME-external -o jsonpath='{.spec.ports[0].nodePort}')
ENDPOINT=http://$IP:$PORT
kubectl -n rook-ceph patch cephobjectzone $PRIMARY_ZONE_NAME --typ
e merge -p "{\"spec\":{\"customEndpoints\":[\"$ENDPOINT\"]}}"

```

3

Получение `access-key` и `secret-key`

```

kubectl -n rook-ceph get secrets $REALM_NAME-keys -o jsonpath='{.dat
a.access-key}'
kubectl -n rook-ceph get secrets $REALM_NAME-keys -o jsonpath='{.dat
a.secret-key}'

```

4

Создание Secondary Zone и настройка синхронизации Realm

В этом разделе описывается создание Secondary Zone и настройка синхронизации путём получения информации Realm из Primary кластера.

Выполните следующие команды на управляющем узле Secondary кластера:

1. Установите параметры

```

export REALM_NAME=<realm-name>
export ZONE_GROUP_NAME=<zonegroup-name>

export REALM_ENDPOINT=<realm-endpoint>
export ACCESS_KEY=<access-key>
export SECRET_KEY=<secret-key>

export SECONDARY_ZONE_NAME=<secondary-zone-name>
export SECONDARY_OBJECT_STORE_NAME=<secondary-object-store-name>

```

Описание параметров:

- `<realm-name>` : [Имя Realm](#).
- `<zone-group-name>` : [Имя ZoneGroup](#).
- `<realm-endpoint>` : [Внешний адрес](#), полученный из Primary кластера.
- `<access-key>` : АК, полученный [здесь](#).
- `<secret-key>` : СК, полученный [здесь](#).
- `<secondary-zone-name>` : имя Secondary Zone.
- `<secondary-object-store-name>` : имя Secondary шлюза.

2. Создайте Secondary Zone и настройте синхронизацию Realm



```
cat << EOF | kubectl apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: $REALM_NAME-keys
  namespace: rook-ceph
data:
  access-key: $ACCESS_KEY
  secret-key: $SECRET_KEY

---
apiVersion: ceph.rook.io/v1
kind: CephObjectRealm
metadata:
  name: $REALM_NAME
  namespace: rook-ceph
spec:
  pull:
    endpoint: $REALM_ENDPOINT

---
apiVersion: ceph.rook.io/v1
kind: CephObjectZoneGroup
metadata:
  name: $ZONE_GROUP_NAME
  namespace: rook-ceph
spec:
  realm: $REALM_NAME

---
apiVersion: ceph.rook.io/v1
kind: CephObjectZone
metadata:
  name: $SECONDARY_ZONE_NAME
  namespace: rook-ceph
spec:
  zoneGroup: $ZONE_GROUP_NAME
  metadataPool:
    failureDomain: host
    replicated:
      size: 3
      requireSafeReplicaSize: true
  dataPool:
```

```
failureDomain: host
replicated:
  size: 3
  requireSafeReplicaSize: true
preservePoolsOnDelete: false

---
apiVersion: ceph.rook.io/v1
kind: CephBlockPool
metadata:
  labels:
    cpaas.io/builtin: "true"
  name: builtin-rgw-root
  namespace: rook-ceph
spec:
  name: .rgw.root
  application: rgw
  enableCrushUpdates: true
  failureDomain: host
  replicated:
    size: 3
  parameters:
    pg_num: "8"

---
apiVersion: ceph.rook.io/v1
kind: CephObjectStore
metadata:
  name: $SECONDARY_OBJECT_STORE_NAME
  namespace: rook-ceph
spec:
  gateway:
    port: 7480
    instances: 2
  zone:
    name: $SECONDARY_ZONE_NAME
EOF
```

5

Настройка внешнего доступа для Secondary Zone

1. Получите UID Secondary шлюза

```
export SECONDARY_OBJECT_STORE_UID=$(kubectl -n rook-ceph get cephobjectstore $SECONDARY_OBJECT_STORE_NAME -o jsonpath='{.metadata.uid}')
```

2. Создайте Service для внешнего доступа

```
cat << EOF | kubectl apply -f -
apiVersion: v1
kind: Service
metadata:
  name: rook-ceph-rgw-$SECONDARY_OBJECT_STORE_NAME-external
  namespace: rook-ceph
  labels:
    app: rook-ceph-rgw
    rook_cluster: rook-ceph
    rook_object_store: $SECONDARY_OBJECT_STORE_NAME
ownerReferences:
  - apiVersion: ceph.rook.io/v1
    kind: CephObjectStore
    name: $SECONDARY_OBJECT_STORE_NAME
    uid: $SECONDARY_OBJECT_STORE_UID
spec:
  ports:
    - name: rgw
      port: 7480
      targetPort: 7480
      protocol: TCP
  selector:
    app: rook-ceph-rgw
    rook_cluster: rook-ceph
    rook_object_store: $SECONDARY_OBJECT_STORE_NAME
  sessionAffinity: None
  type: NodePort
EOF
```

3. Добавьте внешние конечные точки в Secondary CephObjectZone

```

IP=$(kubectl get nodes -l 'node-role.kubernetes.io/control-plane'
-o jsonpath='{.items[0].status.addresses[?(@.type=="InternalIP")].
address}' | cut -d ' ' -f1 | tr -d '\n')
PORT=$(kubectl -n rook-ceph get svc rook-ceph-rgw-$SECONDARY_OBJECT_STORE_NAME-external -o jsonpath='{.spec.ports[0].nodePort}')
ENDPOINT=http://$IP:$PORT
kubectl -n rook-ceph patch cephobjectzone $SECONDARY_ZONE_NAME --type merge -p "{\"spec\":{\"customEndpoints\":[\"$ENDPOINT\"]}}"

```

6

Проверка статуса синхронизации Ceph Object Storage

Выполните следующие команды в pod `rook-ceph-tools` Secondary кластера

```

# войти в pod rook-ceph-tools
kubectl -n rook-ceph exec -it $(kubectl -n rook-ceph get po -l app=rook-ceph-tools -o jsonpath='{range .items[*]}{@.metadata.name}') -- bash

radosgw-admin sync status

```

Пример вывода

```
realm 962d6b80-b218-4fc8-8198-e498fab4e9d (realm-primary)
zonegroup 9de3acd7-0b01-4a04-ac84-1421c6103a16 (zonegroup-primary)
zone 7b3ce7f5-7090-46f6-afb1-d1bb156053da (zone-secondary)
current time 2025-12-04T06:27:15Z
zonegroup features enabled: resharding
                        disabled: compress-encrypted
metadata sync syncing
                        full sync: 0/64 shards
                        incremental sync: 64/64 shards
                        metadata is caught up with master
data sync source: 6319ca70-964e-47be-8b96-7c5bf5a128b1 (zone-primary)
                        syncing
                        full sync: 0/128 shards
                        incremental sync: 128/128 shards
                        data is caught up with source
```

`metadata is caught up with master` и `data is caught up with source`

означают, что статус синхронизации в норме.

Переключение на резервный кластер (Failover)

При сбое Primary кластера необходимо повысить статус Secondary Zone до Primary Zone. После переключения шлюз Secondary Zone сможет продолжать предоставлять услуги объектного хранилища.

Процедуры

Выполните следующие команды в pod `rook-ceph-tools` Secondary кластера

```
# войти в pod rook-ceph-tools
kubectl -n rook-ceph exec -it $(kubectl -n rook-ceph get po -l app=rook-ceph-tools -o jsonpath='{range .items[*]}{@.metadata.name}') -- bash

# Сделать восстановленную зону главной и зоной по умолчанию
radosgw-admin zone modify --rgw-realm=<realm-name> --rgw-zonegroup=<zone-group-name> --rgw-zone=<secondary-zone-name> --master

# Обновить период, чтобы изменения вступили в силу
radosgw-admin period update --commit --rgw-realm=<realm-name> --rgw-zonegroup=<zone-group-name>
```

Параметры

- `<realm-name>` : имя Realm.
- `<zone-group-name>` : имя Secondary Zone Group.
- `<secondary-zone-name>` : имя Secondary Zone.

Возврат на основной кластер (Failback)

После восстановления сбойного кластера на основном сайте и при необходимости возврата с резервного сайта выполните следующие шаги.

Процедуры

Выполните следующие команды в pod `rook-ceph-tools` Primary кластера

```

# войти в pod rook-ceph-tools
kubectl -n rook-ceph exec -it $(kubectl -n rook-ceph get po -l app=rook-ceph-tools -o jsonpath='{range .items[*]}{@.metadata.name}') -- bash

# проверить статус синхронизации, дождаться синхронизации с резервного са
йта
radosgw-admin sync status

#           realm 962d6b80-b218-4fc8-8198-e498fab4e9d (realm-primary)
#           zonegroup 9de3acd7-0b01-4a04-ac84-1421c6103a16 (zonegroup-primar
y)
#           zone 6319ca70-964e-47be-8b96-7c5bf5a128b1 (zone-primary)
#           current time 2025-12-04T07:18:26Z
# zonegroup features enabled: resharding
#           disabled: compress-encrypted
# metadata sync syncing
#           full sync: 0/64 shards
#           incremental sync: 64/64 shards
#           metadata is caught up with master
#           data sync source: 7b3ce7f5-7090-46f6-afb1-d1bb156053da (zone-seco
ndary)
#           syncing
#           full sync: 0/128 shards
#           incremental sync: 128/128 shards
#           data is caught up with source

# Сделать восстановленную зону главной и зоной по умолчанию
radosgw-admin zone modify --rgw-realm=<realm-name> --rgw-zonegroup=<zone-
group-name> --rgw-zone=<primary-zone-name> --master

# Обновить период, чтобы изменения вступили в силу
radosgw-admin period update --commit --rgw-realm=<realm-name> --rgw-zoneg
roup=<zone-group-name>

```

Параметры

- `<realm-name>` : имя Realm.
- `<zone-group-name>` : имя Zone Group.
- `<primary-zone-name>` : имя Primary Zone.

Обновление параметров оптимизации

Платформа поддерживает заполнение параметров оптимизации в формате конфигурационного файла Serp при создании кластера хранения, но не предоставляет способ их изменения через интерфейс после создания. Необходимо вручную обновить их согласно следующим шагам.

Содержание

[Процедура](#)

Процедура

1. Сначала обновите параметры оптимизации хранения в Configmap с именем `rook-config-override-user`, замените поле `.data.config` и установите значение поля `.metadata.annotations[rook.cpaas.io/need-sync]` в `true`. Например:

```
apiVersion: v1
data:
  config: |
    [global]
    mon_memory_target=1073741824
    mds_cache_memory_limit=2147483648
    osd_memory_target=4147483648
kind: ConfigMap
metadata:
  annotations:
    cpaas.io/creator: admin
    cpaas.io/updated-at: "2022-03-01T12:24:04Z"
    rook.cpaas.io/need-sync: "true"
    rook.cpaas.io/sync-status: synced
  creationTimestamp: "2022-03-01T12:24:04Z"
  finalizers:
  - rook.cpaas.io/config-merge
  name: rook-config-override-user
  namespace: default
  resourceVersion: "38816864"
  uid: ce3a8f3e-6453-4bdd-bff0-e16cf7d5d5fa
```

2. Выполните команду `ceph tell [mon|osd|mgr|mds|rgw].* config set [key] [value]` в Pod с rook-ceph-tools для применения конфигурации в реальном времени.
3. Чтобы запустить Pod tools, отредактируйте ClusterServiceVersion (CSV) в пространстве имен rook-ceph и установите значение replicas для rook-ceph-tools в разделе Deployments равным 1.

Создание пользователя Ceph Object Store

Мы поддерживаем создание и настройку пользователей объектного хранилища через определения пользовательских ресурсов (CRD).

Содержание

[Предварительные требования](#)

Процедура

Создание пользователя

Разрешение создания пользователя в других пространствах имён

Получение информации о пользователе

Предварительные требования

- Пул объектного хранилища создан

Процедура

1 Создание пользователя

Выполните команды на **контрольном узле** кластера.

```

cat << EOF | kubectl apply -f -
apiVersion: ceph.rook.io/v1
kind: CephObjectStoreUser
metadata:
  name: <name>
  namespace: <namespace>
spec:
  store: <ObjectStore>
  displayName: <displayName>
  clusterNamespace: <clusterNamespace>
  quotas:
    maxBuckets: 100
    maxSize: -1
    maxObjects: -1
  capabilities:
    user: "*"
    bucket: "*"
EOF

```

Параметры

Параметры	Описание
name	Имя создаваемого пользователя объектного хранилища.
namespace	Пространство имён, в котором создаётся пользователь объектного хранилища.
displayName	Отображаемое имя.
clusterNamespace	Пространство имён, где находятся родительские <code>CephCluster</code> и <code>CephObjectStore</code> . Если не указано, пользователь должен находиться в том же пространстве имён, что и кластер и объектное хранилище. Для использования этой функции в <code>CephObjectStore</code> параметр <code>allowUsersInNamespaces</code> должен включать пространство имён этого пользователя.
ObjectStore	Объектное хранилище, в котором будет создан пользователь. Соответствует имени пула объектного хранилища.

Параметры**Описание****quotas****необязательно**

Задаёт ограничения квот для пользователя.

- `maxBuckets`: Максимальное количество бакетов для пользователя. По умолчанию `100`.
- `maxSize`: Максимальный общий размер всех объектов во всех бакетах пользователя. Значение `-1` означает отсутствие ограничений.
- `maxObjects`: Максимальное количество объектов во всех бакетах пользователя. Значение `-1` означает отсутствие ограничений.

capabilities**необязательно**

Ceph позволяет назначать пользователям дополнительные права. Эта настройка может использоваться только при создании пользователя объектного хранилища. Для изменения прав пользователя его необходимо удалить и создать заново. Подробнее см. в [документации Ceph](#).

Поддерживается добавление прав `read`, `write`, `read,write` или `*` для следующих ресурсов:

- `user`
- `buckets`
- `usage`
- `metadata`
- `zone`
- `roles`
- `info`
- `amz-cache`
- `bilog`
- `mdlog`

Параметры**Описание**

- datalog
- user-policy
- odic-provider
- ratelimit

2

Разрешение создания пользователя в других пространствах имён

Если CephObjectStoreUser создаётся в пространстве имён, отличном от пространства имён кластера Rook, это пространство имён должно быть добавлено в список разрешённых, либо указать "*" для разрешения всех пространств имён. Это полезно для приложений, которым необходимо создавать учётные данные объектного хранилища в собственном пространстве имён.

Выполните команды на **контрольном узле** кластера.

```
kubectl -n rook-ceph patch cephobjectstore <ObjectStore> --type merge
-p '{"spec":{"allowUsersInNamespaces":["*"]}}'
```

3

Получение информации о пользователе

Выполните команды на **контрольном узле** кластера.

```
user_secret=$(kubectl -n <namespace> get cephobjectstoreuser <user-name>
-o jsonpath='{.status.info.secretName}')
```

```
# ACCESS_KEY
```

```
kubectl -n <namespace> get secret $user_secret -o jsonpath='{.data.AccessKey}' | base64 --decode
```

```
# SECRET_KEY
```

```
kubectl -n <namespace> get secret $user_secret -o jsonpath='{.data.SecretKey}' | base64 --decode
```


Установка квот для Storage Pool

Квота пула — это логический лимит размера данных, применяемый на уровне пула хранения Ceph.

Она контролирует, сколько логических данных может быть записано пулом, фактическое физическое пространство будет равно **квота * репликация пула**.

Содержание

[Предварительные требования](#)

Установка квоты пула для File Storage Pool

Установка квоты пула для Block Storage Pool

Установка квоты пула для Object Storage Pool

Проверка квоты пула через Ceph Terminal

Предварительные требования

- Установлен кластер Ceph
- Созданы пулы Ceph

Установка квоты пула для File Storage Pool

Выполните команды на **контрольном узле** кластера.

Command

```

SIZE="<size>"
POOL_NAME="<fs-pool-name>"

kubectl patch cephfilesystem $POOL_NAME -n rook-ceph --type=json \
  -p "[{"op": "add", "path": "/spec/dataPools/0/quotas/maxLength", "value": "$SIZE"}]"

```

Example

```

SIZE="100Gi"
POOL_NAME="cephfs"

kubectl patch cephfilesystem $POOL_NAME -n rook-ceph --type=json \
  -p "[{"op": "add", "path": "/spec/dataPools/0/quotas/maxLength", "value": "$SIZE"}]"

```

Параметры

Параметры	Описание
size	квота в байтах в виде строки с суффиксами количества (например, "10Gi")
fs-pool-name	Имя File Storage Pool.

Установка квоты пула для Block Storage Pool

Выполните команды на **контрольном узле** кластера.

Command

```
SIZE="<size>"  
POOL_NAME="<block-pool-name>"  
  
kubectl patch cephblockpool $POOL_NAME -n rook-ceph --type=json \  
  -p "[{\\"op\\":\\"add\\",\\"path\\":\\"/spec/quotas/maxLength\\",\\"value\  
  \":\\"$SIZE\\"}]"
```

Example

```
SIZE="100Gi"  
POOL_NAME="rbd"  
  
kubectl patch cephblockpool $POOL_NAME -n rook-ceph --type=json \  
  -p "[{\\"op\\":\\"add\\",\\"path\\":\\"/spec/quotas/maxLength\\",\\"value\  
  \":\\"$SIZE\\"}]"
```

Параметры

Параметры	Описание
size	квота в байтах в виде строки с суффиксами количества (например, "10Gi")
block-pool-name	Имя Block Storage Pool.

Установка квоты пула для Object Storage Pool

Выполните команды на **контрольном узле** кластера.

Command

```
SIZE="<size>"
POOL_NAME="<object-pool-name>"

kubectl patch cephobjectstore $POOL_NAME -n rook-ceph --type=json \
  -p "[{"op": "add", "path": "/spec/dataPool/quotas/maxLength
  ", "value": "$SIZE"}]"
```

Example

```
SIZE="100Gi"
POOL_NAME="object"

kubectl patch cephobjectstore $POOL_NAME -n rook-ceph --type=json \
  -p "[{"op": "add", "path": "/spec/dataPool/quotas/maxLength
  ", "value": "$SIZE"}]"
```

Параметры

Параметры	Описание
size	квота в байтах в виде строки с суффиксами количества (например, "10Gi")
object-pool-name	Имя Object Storage Pool.

Проверка квоты пула через Ceph Terminal

```
ceph osd pool ls detail | grep max_bytes
```

```
pool 3 'cephfs-data0' replicated size 3 min_size 2 crush_rule 4 object_hash rjenkins pg_num 32 pgp_num 32 autoscale_mode on last_change 100 lfor 0/0/56 flags hashspool max_bytes 107374182400 stripe_width 0 application cephfs read_balance_score 1.31
```

```
pool 4 'rbd' replicated size 3 min_size 2 crush_rule 5 object_hash rjenkins pg_num 32 pgp_num 32 autoscale_mode on last_change 117 lfor 0/0/111 flags hashspool,selfmanaged_snaps max_bytes 107374182400 stripe_width 0 application rbd read_balance_score 1.31
```

```
pool 12 'object.rgw.buckets.data' replicated size 3 min_size 2 crush_rule 13 object_hash rjenkins pg_num 32 pgp_num 32 autoscale_mode on last_change 304 lfor 0/0/168 flags hashspool max_bytes 107374182400 stripe_width 0 compression_mode none application rgw read_balance_score 1.59
```

MinIO Object Storage

Введение

[Введение](#)

Установка

[Установка](#)

[Предварительные требования](#)

[Процедура](#)

[Связанная информация](#)

Архитектура

[Архитектура](#)

[Основные компоненты:](#)

[Архитектура развертывания:](#)

[Масштабирование с помощью нескольких пулов:](#)

Заключение:

Основные понятия

[Основные концепции](#)

Руководства

[Добавление пула хранения](#)

Примечания

Процедура

[Мониторинг и оповещения](#)

Мониторинг

Оповещения

Как сделать

[Восстановление данных после аварий](#)

Применимые сценарии

Терминология

Предварительные требования

Шаги операции

Связанные операции

Введение

Alauda Container Platform (ACP) Object Storage с MinIO — это сервис объектного хранения, лицензированный под Apache License v2.0. Он совместим с интерфейсом облачного хранилища Amazon S3, что делает его особенно подходящим для хранения больших объемов неструктурированных данных, таких как изображения, видео, файлы журналов, резервные копии и образы контейнеров/виртуальных машин. Размер объекта может варьироваться от нескольких КБ до максимума в 5 ТБ.

Основные преимущества следующие:

- **Простота:** Минимализм — главный принцип дизайна MinIO, обеспечивающий функциональность «из коробки». Простота снижает вероятность ошибок, увеличивает время безотказной работы и повышает надежность, а также улучшает производительность.
- **Высокая производительность:** MinIO является мировым лидером в области объектного хранения. На стандартном оборудовании скорость чтения/записи может достигать до 183 ГБ/с и 171 ГБ/с соответственно.
- **Масштабируемость:** Можно создавать несколько небольших и средних кластеров, которые легко управляются, поддерживая агрегацию нескольких кластеров в сверхбольшой пул ресурсов между дата-центрами, вместо прямого использования крупномасштабного централизованного распределенного кластера.
- **Облачная нативность:** Соответствует всем нативным архитектурам и процессам построения облачных вычислений, включает новейшие технологии и концепции облачных вычислений, делая объектное хранилище более удобным для Kubernetes.

Установка

Alauda Container Platform (ACP) Object Storage с MinIO — это сервис объектного хранения, основанный на протоколе с открытым исходным кодом Apache License v2.0. Он совместим с интерфейсом облачного хранилища Amazon S3 и идеально подходит для хранения больших объемов неструктурированных данных, таких как изображения, видео, файлы журналов, резервные копии и образы контейнеров/виртуальных машин. Размер объекта может быть любым — от нескольких килобайт до максимума в 5 терабайт.

Содержание

[Предварительные требования](#)

Процедура

- Развертывание Alauda Container Platform Storage Essentials

- Развертывание Operator

- Создание кластера

- Создание Bucket

- Загрузка/скачивание файлов

Связанная информация

- Таблица соответствия коэффициента избыточности

- Обзор пула хранения

Предварительные требования

- MinIO строится на базовом хранилище, поэтому убедитесь, что в текущем кластере создан класс хранилища. Рекомендуется TopoLVM.
- **Скачайте** установочный пакет **Alauda Container Platform Storage Essentials**, соответствующий архитектуре вашей платформы.
- **Загрузите** установочный пакет **Alauda Container Platform Storage Essentials** с помощью механизма Upload Packages.
- **Скачайте** установочный пакет **Alauda Container Platform (ACP) Object Storage with MinIO**, соответствующий архитектуре вашей платформы.
- **Загрузите** установочный пакет **Alauda Container Platform (ACP) Object Storage with MinIO** с помощью механизма Upload Packages.

Процедура

1

Развертывание Alauda Container Platform Storage Essentials

1. Войдите в систему и перейдите на страницу **Administrator**.
2. Нажмите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите **Alauda Container Platform Storage Essentials**, нажмите **Install** и перейдите на страницу **Install Alauda Container Platform Storage Essentials**.

Параметры конфигурации:

Параметр	Рекомендуемая конфигурация
Channel	Канал по умолчанию — <code>stable</code> .
Installation Mode	<code>Cluster</code> : Все пространства имён в кластере используют один экземпляр Operator для создания и управления, что снижает использование ресурсов.
Installation Place	Выберите <code>Recommended</code> , Namespace поддерживается только <code>acp-storage</code> .

Параметр	Рекомендуемая конфигурация
Upgrade Strategy	Manual : При появлении новой версии в Operator Hub требуется ручное подтверждение для обновления Operator до последней версии.

2 Развертывание Operator

1. В левой навигационной панели нажмите **Storage > Object Storage**.
2. Нажмите **Configure Now**.
3. На странице мастера **Deploy MinIO Operator** нажмите внизу справа **Deploy Operator**.
 - Если страница автоматически перейдёт к следующему шагу, значит развертывание Operator прошло успешно.
 - Если развертывание не удалось, следуйте подсказкам интерфейса для **Clean Up Deployed Information and Retry** и повторно разверните Operator.

3 Создание кластера

1. На странице мастера **Create Cluster** настройте основную информацию.

Параметр	Описание
Access Key	Идентификатор ключа доступа. Уникальный идентификатор, связанный с приватным ключом доступа; используется вместе с access key ID для шифрования и подписания запросов.
Secret Key	Приватный ключ доступа, используемый вместе с access key ID для шифрования и подписания запросов, идентификации отправителя и предотвращения подделки запросов.

2. В области **Resource Configuration** настройте спецификации согласно следующим инструкциям.

Параметр	Описание
Small scale	Подходит для обработки до 100 000 объектов, поддерживает не более 50 одновременных обращений в тестовых средах или сценариях резервного копирования. Запрос и лимит ресурсов CPU по умолчанию — 2 ядра, запрос и лимит памяти — 4 Gi.
Medium scale	Предназначен для корпоративных приложений, требующих хранения 1 000 000 объектов и обработки до 200 одновременных запросов. Запрос и лимит CPU по умолчанию — 4 ядра, запрос и лимит памяти — 8 Gi.
Large scale	Предназначен для групп пользователей с потребностями хранения 10 000 000 объектов и обработки до 500 одновременных запросов, подходит для сценариев с высокой нагрузкой. Запрос и лимит CPU по умолчанию — 8 ядер, запрос и лимит памяти — 16 Gi.
Custom	Предлагает гибкие настройки для профессиональных пользователей с особыми требованиями, обеспечивая точное соответствие масштаба сервиса и требований к производительности. Примечание: при настройке пользовательских спецификаций убедитесь, что: <ul style="list-style-type: none"> • Запрос CPU больше 100 m. • Запрос памяти не меньше 2 Gi. • Лимиты CPU и памяти не меньше соответствующих запросов.

3. В области **Storage Pool** настройте соответствующую информацию согласно следующим инструкциям.

Параметр	Описание
Instance Number	Увеличение количества экземпляров в кластере MinIO значительно повышает производительность и надёжность системы, обеспечивая высокую доступность данных. Однако

Параметр	Описание
	<p>слишком большое число экземпляров может привести к следующим проблемам:</p> <ul style="list-style-type: none"> • Увеличение потребления ресурсов. • Если на одном узле размещено несколько экземпляров, сбой узла может привести к одновременному отключению нескольких экземпляров, снижая общую надёжность кластера. <p>Примечание:</p> <ul style="list-style-type: none"> • Минимальное количество экземпляров — 4. • Если количество экземпляров больше 16, значение должно быть кратно 8. • При добавлении дополнительных пулов хранения количество экземпляров не должно быть меньше, чем в первом пуле хранения.
Single Storage Volume	<p>Вместимость одного тома PVC. Каждый сервис хранения управляет одним томом. После ввода вместимости одного тома платформа автоматически рассчитает ёмкость пула хранения и другую информацию, которую можно просмотреть в разделе Storage Pool Overview.</p>
Underlying Storage	<p>Базовое хранилище, используемое кластером MinIO. Выберите класс хранилища, созданный в текущем кластере. Рекомендуется TopoLVM.</p>
Storage Nodes	<p>Выберите узлы хранения, необходимые для кластера MinIO. Рекомендуется использовать от 4 до 16 узлов хранения. Платформа развернёт по одному сервису хранения на каждом выбранном узле.</p>
Storage Pool Overview	<p>Для конкретных параметров и формул расчёта смотрите Storage Pool Overview.</p>

4. В области **Access Configuration** настройте соответствующую информацию согласно следующим инструкциям.

Параметр	Описание
External Access	<p>При включении поддерживается доступ к MinIO из других кластеров; при отключении — доступ возможен только внутри кластера.</p>
Protocol	<p>Поддерживает HTTP и HTTPS; при выборе HTTPS необходимо ввести Domain и импортировать Public Key и Private Key сертификата домена.</p> <p>Примечание:</p> <ul style="list-style-type: none"> • При протоколе HTTP поды внутри кластера могут обращаться к MinIO напрямую по полученному IP или доменному имени без настройки сопоставления IP и домена; узлы внутри кластера могут обращаться к MinIO напрямую по IP, а для доступа по доменному имени требуется ручная настройка сопоставления IP и домена; внешний доступ возможен напрямую по IP. • При протоколе HTTPS доступ к MinIO по IP невозможен как внутри, так и снаружи кластера. Для нормального доступа по доменному имени требуется ручная настройка сопоставления между полученным IP и доменом, введённым при создании кластера.
Access Method	<ul style="list-style-type: none"> • NodePort: Открывает фиксированный порт на каждом хосте вычислительного узла для внешнего доступа к сервису. При настройке доступа по доменному имени рекомендуется использовать VIP для разрешения домена, чтобы обеспечить высокую доступность. • LoadBalancer: Использует балансировщик нагрузки для перенаправления трафика на бэкенд-сервисы. Перед использованием убедитесь, что в текущем кластере развернут плагин MetalLB и в пуле внешних адресов есть доступные IP.

5. Нажмите внизу справа **Create Cluster**.

- Если страница автоматически перейдёт к **Cluster Details**, значит создание кластера прошло успешно.
- Если кластер остаётся в процессе создания, можно нажать **Cancel**. После отмены развернутые данные кластера будут очищены, и вы сможете вернуться на страницу создания кластера для повторного создания.

4

Создание Bucket

Войдите на управляющий узел кластера и используйте команду для создания bucket.

1. На странице сведений о кластере перейдите на вкладку **Access Method**, чтобы посмотреть адрес доступа MinIO, или выполните следующую команду для запроса.

```
kubectl get svc -n <tenant ns> minio | grep -w minio | awk '{print $3}'
```

Примечание:

- Замените `tenant ns` на фактическое пространство имён `minio-system`.
- Пример: `kubectl get svc -n minio-system minio | grep -w minio | awk '{print $3}'`

2. Получите команду mc.

```
wget https://dl.min.io/client/mc/release/linux-amd64/mc -O /bin/mc
&& chmod a+x /bin/mc
```

3. Настройте псевдоним кластера MinIO.

- IPv4:

```
mc --insecure alias set <minio cluster alias> http://<minio endpoint>:<port> <accessKey> <secretKey>
```

- IPv6:

```
mc --insecure alias set <minio cluster alias> http://[<minio end point>]:<port> <accessKey> <secretKey>
```

- Доменное имя:

```
mc --insecure alias set <minio cluster alias> http://<domain name>:<port> <accessKey> <secretKey>
mc --insecure alias set <minio cluster alias> https://<domain name>:<port> <accessKey> <secretKey>
```

Примечание:

- Введите IP, полученный на шаге 1, вместо `minio endpoint`.
- Введите **Access Key** и **Secret Key**, созданные при создании кластера, вместо `accessKey` и `secretKey`.

- Примеры конфигурации:

- IPv4: `mc --insecure alias set myminio http://12.4.121.250:80 07Apples@ 07Apples@`

- IPv6: `mc --insecure alias set myminio http://[2004::192:168:143:117]:80 07Apples@ 07Apples@`

- Доменное имя: `mc --insecure alias set myminio http://test.minio.alauda:80 07Apples@ 07Apples@` или `mc --insecure alias set myminio https://test.minio.alauda:443 07Apples@ 07Apples@`

4. Создайте bucket.

```
mc --insecure mb <minio cluster alias>/<bucket name>
```

5

Загрузка/скачивание файлов

После создания bucket вы можете использовать командную строку для загрузки файлов в bucket или скачивания существующих файлов из bucket.

1. Создайте файл для тестовой загрузки. Этот шаг можно пропустить, если загружаете существующий файл.

```
touch <file name>
```

2. Загрузите файлы в bucket.

```
mc --insecure cp <file name> <minio cluster alias>/<bucket name>
```

3. Просмотрите файлы в bucket, чтобы подтвердить успешную загрузку.

```
mc --insecure ls <minio cluster alias>/<bucket name>
```

4. Удалите загруженные файлы.

```
mc --insecure rm <minio cluster alias>/<bucket name>/<file name>
```

Связанная информация

Таблица соответствия коэффициента избыточности

Примечание: При добавлении дополнительных пулов хранения коэффициент избыточности рассчитывается на основе количества экземпляров в первом пуле хранения.

Количество экземпляров	Коэффициент избыточности
4 - 5	2
6 - 7	3
>= 8	4

Обзор пула хранения

Параметр обзора пула хранения	Формула расчёта
<p>Доступная ёмкость</p>	<p>При Instance Number ≤ 16, Доступная ёмкость = Вместимость одного тома \times (Количество экземпляров - Коэффициент избыточности).</p>
<p>При количестве экземпляров > 16, Доступная ёмкость = Вместимость одного тома \times (Количество экземпляров - $4 \times$ (Количество экземпляров + 15) / 16). Результат выражения "$4 \times$ (Количество экземпляров + 15) / 16" округляется вниз.</p>	
<p>Общая ёмкость</p>	<p>Общая ёмкость = Количество экземпляров \times Вместимость одного тома</p>
<p>Количество отказоустойчивых сервисов хранения</p>	<p>При Instance Number $> 2 \times$ Коэффициент избыточности, Количество отказоустойчивых сервисов хранения = Коэффициент избыточности.</p>
<p>При Instance Number = $2 \times$ Коэффициент избыточности, количество отказоустойчивых сервисов хранения = Коэффициент избыточности - 1</p>	

Архитектура

Alauda Container Platform (ACP) Object Storage с MinIO — это высокопроизводительная распределённая система объектного хранения, разработанная для облачно-нативных сред. Она использует стирающее кодирование, распределённые пулы хранения и механизмы высокой доступности для обеспечения долговечности данных и масштабируемости в Kubernetes.

Содержание

Основные компоненты:

Архитектура развертывания:

Масштабирование с помощью нескольких пулов:

Заключение:

Основные компоненты:

- **MinIO Operator:** Управляет развертыванием и обновлением кластеров MinIO.
- **MinIO Peer:** Настраивает и управляет функциональностью репликации сайтов MinIO.
- **MinIO Pool:** Основной компонент MinIO, отвечающий за обработку запросов объектного хранения. Каждый пул соответствует StatefulSet и предоставляет ресурсы хранения.

Архитектура развертывания:

Для развертывания MinIO в Kubernetes необходимо определить MinIO tenant, указав количество серверных экземпляров (pod) и количество томов (дисков) на каждый экземпляр. Каждый сервер MinIO управляется через StatefulSet, что обеспечивает стабильные идентификаторы и постоянное хранилище. MinIO агрегирует все диски в один или несколько стирающих наборов и применяет стирающее кодирование для обеспечения отказоустойчивости.

Масштабирование с помощью нескольких пулов:

Кластеры MinIO могут масштабироваться путём добавления дополнительных серверных пулов, каждый из которых имеет свой стирающий набор. Хотя это увеличивает ёмкость хранения, такая архитектура усложняет обслуживание кластера и снижает общую надёжность. Сбой в любом серверном пуле может сделать весь кластер MinIO недоступным, даже если другие пулы продолжают работать.

Заключение:

MinIO — это высокомасштабируемое облачно-нативное решение для объектного хранения, которое обеспечивает баланс между производительностью и надёжностью. При проектировании кластера MinIO важно тщательно продумывать пулы хранения, настраивать параметры стирающего кодирования и реализовывать стратегии высокой доступности для обеспечения целостности данных и стабильности работы в Kubernetes.

ОСНОВНЫЕ ПОНЯТИЯ

[Основные концепции](#)

Основные концепции

- **Erasure Coding (EC):** MinIO использует кодирование с удалением (Reed-Solomon erasure coding) для разбиения объектов на фрагменты данных и контрольные фрагменты (parity shards), распределяя их по нескольким дискам для обеспечения отказоустойчивости. Например, в конфигурации с 16 дисками данные могут быть разбиты на 12 фрагментов данных и 4 контрольных фрагмента, что позволяет системе восстанавливать данные даже при выходе из строя до 4 дисков.
- **Server Pools & Erasure Sets:** Server Pools MinIO — это логические объединения ресурсов хранения, где каждый пул состоит из нескольких узлов, совместно использующих возможности хранения и вычислений. Внутри пула диски автоматически организуются в один или несколько **Erasure Sets**.
 - **Распределение данных:** При сохранении объекта он разбивается на фрагменты данных и контрольные фрагменты, которые распределяются по разным дискам внутри erasure set.
 - **Модель избыточности:** Erasure sets представляют собой базовую единицу избыточности данных, обеспечивая устойчивость на основе настроенного соотношения фрагментов данных и контрольных фрагментов.
 - **Масштабируемость:** Один пул хранения MinIO может содержать несколько erasure sets, и новые данные всегда записываются в erasure set с наибольшей доступной емкостью.

Руководства

Добавление пула хранения

Примечания

Процедура

Мониторинг и оповещения

Мониторинг

Оповещения

Добавление пула хранения

Пул хранения — это логический раздел, используемый для хранения данных. В одном и том же кластере хранения могут одновременно использоваться разные типы базового хранилища для удовлетворения различных бизнес-требований.

Помимо пулов хранения, созданных при настройке объектного хранилища, вы также можете добавить дополнительные пулы хранения.

Содержание

[Примечания](#)

[Процедура](#)

Примечания

Добавление пула хранения вызовет кратковременное прерывание работы сервиса MinIO, но после этого он автоматически восстановится в нормальное состояние.

Процедура

1. Перейдите в раздел **Administrator**.
2. В левой навигационной панели нажмите **Storage Management > Object Storage**.
3. На вкладке **Cluster Information** прокрутите вниз до раздела **Storage Pool** и нажмите **Add Storage Pool**.

4. Настройте соответствующие параметры согласно приведённым ниже инструкциям.

Параметр	Описание
Underlying Storage	Базовое хранилище, используемое кластером MinIO. Пожалуйста, выберите существующий класс хранения, созданный в текущем кластере, рекомендуется использовать ToroLVM.
Storage Nodes	Выберите узлы хранения, необходимые для кластера MinIO. Рекомендуется использовать от 4 до 16 узлов хранения; платформа развернёт по 1 сервису хранения на каждый выбранный узел. Примечание: при использовании 3 узлов хранения для обеспечения надёжности на каждый узел будет развернуто по 2 сервиса хранения.
Single Storage Volume	Вместимость одного тома хранения PVC. Каждый сервис хранения управляет 1 томом хранения, и после ввода вместимости одного тома платформа автоматически рассчитывает ёмкость пула хранения и другую информацию, которую можно просмотреть в Storage Pool Overview .

5. Нажмите **Confirm**.

Мониторинг и оповещения

Система объектного хранения данных оснащена встроенными возможностями мониторинга и оповещений, охватывающими кластеры хранения, состояние сервисов и использование ресурсов. Также поддерживаются настраиваемые политики уведомлений для информирования вашей операционной команды. Информация мониторинга в реальном времени помогает в оптимизации производительности и принятии операционных решений, а автоматические оповещения обеспечивают стабильность и надежность вашей системы хранения.

Содержание

Мониторинг

- Обзор хранения

- Мониторинг кластера

- Мониторинг объектов

Оповещения

- Настройка уведомлений

- Обработка оповещений

- Анализ после инцидента

Мониторинг

По умолчанию платформа собирает ключевые метрики по кластерам хранения и состоянию сервисов. Вы можете получить доступ к данным мониторинга в реальном

времени в разделе **Storage Management > Object Storage > Monitoring**.

Обзор хранения

Этот раздел предоставляет общий обзор состояния системы хранения, статуса сервисов и использования сырой емкости. Если состояние хранения ненормальное, детали оповещения укажут на первопричину, что поможет эффективно диагностировать и устранять проблемы.

Мониторинг кластера

Отслеживайте использование сырой емкости и тенденции производительности ввода-вывода по всему кластеру хранения. Это помогает выявлять узкие места, оптимизировать распределение ресурсов и обеспечивать бесперебойную работу с данными.

Мониторинг объектов

Отслеживайте модели доступа, включая общее количество запросов и количество неудачных запросов. Эти данные помогают анализировать нагрузку на хранилище и выявлять аномалии, которые могут свидетельствовать о сбоях сервисов или угрозах безопасности.

Оповещения

Платформа поставляется с преднастроенными политиками оповещений для обнаружения аномалий и отправки уведомлений при достижении заданных порогов. Встроенные правила охватывают ключевые области, такие как состояние компонентов, использование емкости и целостность пользовательских данных.

Настройка уведомлений

Для обеспечения своевременного реагирования настройте политики уведомлений в **Operations Center**. Оповещения могут отправляться по электронной почте, SMS или

другим каналам, чтобы информировать ответственных сотрудников. Тонко настройте параметры в соответствии с рабочим процессом реагирования вашей организации.

Обработка оповещений

- **Кластер в состоянии "Alert"**: Сработало предупреждение, и стабильность системы может быть под угрозой. Проверьте раздел **Live Alerts** для получения подробностей, определите первопричину и примите корректирующие меры.
- **Кластер в состоянии "Failure"**: Кластер хранения перестал работать. Требуется немедленное вмешательство для восстановления доступности сервиса.

Платформа классифицирует оповещения по уровням серьезности, помогая командам приоритизировать реагирование на инциденты:

Уровень серьезности	Описание
Critical	Сбой системы, влияющий на бизнес-процессы или приводящий к потере данных. Требуется немедленное действие.
Major	Известная проблема, которая может привести к сбоям в функциональности и нарушить бизнес-процессы.
Warning	Потенциальный риск, который при отсутствии реакции может повлиять на производительность или доступность.

Анализ после инцидента

Журнал **Alert History** содержит все прошлые инциденты, предоставляя ценные данные для анализа и улучшения системы. При рассмотрении прошлых оповещений учитывайте следующее:

1. Каковы были точные симптомы во время инцидента?
2. Повторяются ли определённые оповещения со временем? Можно ли принять проактивные меры для предотвращения повторения?
3. Был ли определённый временной интервал с резким увеличением количества оповещений? Был ли он вызван операционной проблемой или внешним фактором?

Нужно ли скорректировать стратегию реагирования?

Постоянный анализ шаблонов оповещений и совершенствование стратегий мониторинга позволяют повысить устойчивость системы, минимизировать время простоя и обеспечить бесперебойную работу хранилища.

Как сделать

Восстановление данных после аварий

Применимые сценарии

Терминология

Предварительные требования

Шаги операции

Связанные операции

Восстановление данных после аварий

MinIO поддерживает создание центра аварийного восстановления через удалённое резервное копирование данных или активное-активное развертывание, чтобы обеспечить сохранность исходных данных при возникновении аварийных ситуаций, тем самым гарантируя безопасность и надёжность данных.

Содержание

[Применимые сценарии](#)

Терминология

Предварительные требования

Шаги операции

Связанные операции

Применимые сценарии

- **Горячее резервное копирование:** Два дата-центра находятся в одном городе или в разных местах, один из них основной, другой — резервный. Данные в реальном времени реплицируются с основного кластера на резервный, чтобы обеспечить согласованность данных. При аварии в основном кластере бизнес-трафик может быть бесшовно переключён на резервный кластер для обеспечения непрерывности работы.
- **Активное-активное на уровне города:** В архитектуре активное-активное на уровне города (мультикластер) два дата-центра расположены в разных кластерах. Оба дата-

центра активны и могут одновременно принимать бизнес-трафик. При аварии в одном дата-центре бизнес продолжает работать без прерывания в другом.

Терминология

- **Основной кластер:** Кластер, который в данный момент активен и обрабатывает бизнес-запросы. Является источником данных или инициатором операций. В основном кластере данные создаются, изменяются или обновляются, и бизнес-трафик сначала направляется именно в этот кластер для обработки.
- **Целевой кластер:** Кластер, который принимает репликацию данных, миграцию или переключение. Обычно находится в резервном или ожидательном состоянии, готовый принять данные с основного кластера или взять на себя бизнес-трафик. При сбое основного кластера или необходимости переключения целевой кластер получает копии данных с основного или принимает бизнес-трафик для обеспечения непрерывности работы. В сценарии активное-активное оба кластера могут выступать в роли целевого друг для друга.

Предварительные требования

- И основной, и целевой кластер должны иметь включённый доступ к внешней сети. Для конкретных методов настройки см. [Create Object Storage](#).
- Основной кластер должен использовать метод доступа **LoadBalancer**, а целевой кластер рекомендуется поддерживать функциональность **балансировки нагрузки**.
- Основной и целевой кластеры должны использовать одинаковый протокол доступа, то есть либо оба HTTP, либо оба HTTPS.
- При использовании протокола HTTPS оба кластера должны настроить DNS-разрешение для себя и друг друга.
- При использовании HTTPS рекомендуется, чтобы и основной, и целевой кластеры использовали сертификаты, подписанные CA, для обеспечения безопасного и доверенного соединения; если используются самоподписанные сертификаты, обе стороны должны импортировать и доверять сертификатам друг друга для успешного установления защищённого HTTPS-соединения.

Шаги операции

1. Войдите в **Administrator**.
2. В левой панели навигации нажмите **Storage Management > Object Storage**.
3. На вкладке **Data Disaster Recovery** нажмите **Add Target Cluster**.
4. Настройте соответствующие параметры целевого кластера согласно следующим инструкциям.

Параметр	Описание
Access Address	Внешний адрес доступа целевого кластера, начинающийся с <code>http://</code> или <code>https://</code> .
Access Key	Идентификатор Access Key для целевого кластера. Уникальный идентификатор, связанный с приватным ключом доступа; используется вместе с приватным ключом для шифрования запросов.
Secret Key	Приватный ключ доступа, используемый вместе с Access Key ID для шифрования запросов, идентификации отправителя и предотвращения изменения запроса.

5. Нажмите **Add**.

- После успешного добавления вы сможете просматривать статус целевого кластера и статус синхронизации между кластерами.

Параметр	Описание
Cluster Status	Статус целевого кластера, включая Healthy , Abnormal или Unknown .
Buckets	Количество бакетов, ожидающих синхронизации, и уже синхронизированных. <ul style="list-style-type: none"> • В сценариях горячего резервного копирования ожидающая синхронизация — это количество бакетов, которые основной кластер должен синхронизировать с целевым.

Параметр	Описание
	<ul style="list-style-type: none"> В сценариях активное-активное на уровне города ожидающая синхронизация — это общее количество бакетов, которые необходимо синхронизировать между основным и целевым кластерами.
Objects	<p>Количество объектов, неудачно синхронизированных в бакете.</p> <p>Примечание: Это число приведено для справки, так как MinIO синхронизирует связанные конфигурации файлов во время синхронизации.</p>
Network Traffic Rate	<p>Скорость входящего и исходящего сетевого трафика основного кластера.</p> <ul style="list-style-type: none"> В сценариях горячего резервного копирования скорость входящего трафика всегда равна 0. В сценариях активное-активное на уровне города данные есть как по входящему, так и по исходящему трафику.

- Если добавление целевого кластера не удалось, вы можете нажать **Re-add**, чтобы очистить информацию о кластере и вернуться на страницу добавления целевого кластера для повторного добавления.

Связанные операции

Когда необходимость в аварийном восстановлении отпадёт, вы можете нажать **Remove Target Cluster**. Удаление целевого кластера не удаляет уже синхронизированные данные; если в данный момент происходит синхронизация данных, она будет прервана.

Локальное хранилище TopoLVM

Введение

[Введение](#)

Установка

[Установка](#)

[Предварительные требования](#)

[Процедура](#)

Руководства

[Управление устройствами](#)

[Предварительные требования](#)

[Добавление устройств](#)

[Мониторинг и оповещения](#)

[Мониторинг](#)

[Оповещения](#)

Как сделать

Резервное копирование и восстановление TopoLVM с помощью Velero

Предварительные требования

Ограничения

Процедура

Настройка полосатых логических томов

Предварительные требования

Процедура

Введение

ТopoLVM — это плагин Container Storage Interface (CSI), разработанный специально для Kubernetes, предназначенный для эффективного и удобного управления локальными томами хранения.

Основные функции и преимущества:

- **Управление локальными томами:** ТopoLVM ориентирован на управление локальными устройствами хранения (такими как диски и SSD) на узлах Kubernetes. По сравнению с традиционным сетевым хранилищем, локальные тома обеспечивают меньшую задержку и более высокую производительность.
- **Осведомленность о топологии:** ТopoLVM способен распознавать топологию кластера Kubernetes (например, узлы, зоны доступности), что позволяет автоматически выделять тома хранения на том же узле в соответствии с фактическим местоположением планирования Pod, дополнительно оптимизируя производительность.
- **Динамическое выделение томов:** ТopoLVM поддерживает динамическое создание, удаление и изменение размера томов хранения без ручного вмешательства, значительно упрощая операции и снижая сложность.
- **Глубокая интеграция с Kubernetes:** Как плагин CSI, ТopoLVM бесшовно интегрируется с API управления хранилищем Kubernetes, позволяя пользователям управлять локальными томами напрямую через стандартные объекты ресурсов Kubernetes, такие как PersistentVolumeClaims.

В итоге, ТopoLVM решает распространённые проблемы, связанные с использованием локального хранилища в Kubernetes, такие как ручное управление, отсутствие осведомленности о топологии и недостаточные возможности динамического выделения.

Он предоставляет более эффективное и удобное решение для приложений, требующих высокопроизводительного локального хранилища, таких как базы данных и кэши.

Установка

Local storage — это программно-определяемое локальное хранилище на сервере, которое обеспечивает простую, удобную в обслуживании и высокопроизводительную возможность локального хранения данных. Основанное на решении сообщества TopoLVM, оно реализует оркестрацию управления постоянными томами локального хранилища через системный подход LVM.

Содержание

Предварительные требования

Процедура

Развертывание Alauda Container Platform Storage Essentials

Развертывание хранилища

Предварительные требования

- На каждом узле кластера хранения должен быть установлен пакет `lvm2`. Если он не установлен, выполните команду `yum install -y lvm2` на узле.
- **Скачайте** установочный пакет **Alauda Container Platform Storage Essentials**, соответствующий архитектуре вашей платформы.
- **Загрузите** установочный пакет **Alauda Container Platform Storage Essentials** с помощью механизма Upload Packages.
- **Скачайте** установочный пакет **Alauda Build of TopoLVM**, соответствующий архитектуре вашей платформы.

- Загрузите установочный пакет **Alauda Build of TopoLVM** с помощью механизма Upload Packages.

Процедура

1 Развертывание Alauda Container Platform Storage Essentials

1. Войдите в систему, перейдите на страницу **Administrator**.
2. Нажмите **Marketplace > OperatorHub**, чтобы перейти на страницу **OperatorHub**.
3. Найдите **Alauda Container Platform Storage Essentials**, нажмите **Install** и перейдите на страницу **Install Alauda Container Platform Storage Essentials**.

Параметры конфигурации:

Параметр	Рекомендуемая конфигурация
Channel	Канал по умолчанию — <code>stable</code> .
Installation Mode	<code>Cluster</code> : Все пространства имён в кластере используют один экземпляр Operator для создания и управления, что снижает использование ресурсов.
Installation Place	Выберите <code>Recommended</code> , Namespace поддерживается только <code>acp-storage</code> .
Upgrade Strategy	<code>Manual</code> : При наличии новой версии в Operator Hub требуется ручное подтверждение для обновления Operator до последней версии.

2 Развертывание хранилища

1. Перейдите в **Administrator**.
2. В левой навигационной панели нажмите **Storage Management > Local Storage**.
3. Нажмите **Configure Now**.

4. На странице мастера **Install Operator** нажмите **Start Deployment**.

- Если страница автоматически переходит к следующему шагу, это означает успешное развертывание Operator.
- Если развертывание не удалось, следуйте подсказкам интерфейса для решения проблемы. Затем нажмите **Clean Up** и повторно разверните Operator.

5. На странице мастера **Create Cluster** добавьте устройства.

Параметр	Описание
Select Node	Узел с минимум одним неразмеченным диском.
Device Class	Каждый класс устройств соответствует набору устройств хранения с одинаковыми характеристиками. Рекомендуется указывать название, исходя из типа диска, например <i>hdd</i> , <i>ssd</i> .
Device Type	Поддерживаются только типы дисков.
Storage Device	Например, <i>/dev/sda</i> . Если дисков несколько, их можно добавлять по одному.
Snapshot	<p>При включении поддерживается создание снимков PVC и использование их для настройки новых PVC для быстрого резервного копирования и восстановления данных.</p> <p>Если при создании хранилища снимок не был включён, его можно активировать при необходимости в разделе Operations на странице деталей кластера хранения.</p> <p>Примечание: Перед использованием убедитесь, что для текущего кластера развернут Volume Snapshot Plugin.</p>

- Нажмите Далее. Если страница автоматически переходит к следующему шагу, это означает успешное развертывание кластера.
- Если создание не удалось, следуйте подсказкам интерфейса и своевременно очистите ресурсы.

6. На странице мастера **Create Storage Class** настройте соответствующие параметры.

Параметр	Описание
Name	Имя класса хранения. Должно быть уникальным в пределах текущего кластера.
Display Name	Имя, помогающее идентифицировать или фильтровать, например, описание класса хранения на русском языке.
Device Class	Класс устройств — способ категоризации устройств хранения в ToroLVM. Каждый класс устройств соответствует набору устройств с одинаковыми характеристиками. При отсутствии особых требований можно использовать класс устройств Auto-Allocated из кластера.
File System	<ul style="list-style-type: none"> - XFS — высокопроизводительная журналируемая файловая система, хорошо справляющаяся с параллельными I/O нагрузками, поддерживающая обработку больших файлов и обеспечивающая плавную передачу данных. - EXT4 — журналируемая файловая система в Linux, использующая методы хранения с экстендами и поддерживающая обработку больших файлов. Максимальная ёмкость файловой системы — 1 EiB, максимальный размер файла — 16 TiB.
Recycling Policy	<p>Политика переработки для постоянных томов.</p> <ul style="list-style-type: none"> - Delete: При удалении persistent volume claim связанный persistent volume также удаляется. - Retain: Даже при удалении persistent volume claim связанный persistent volume сохраняется.
Access Mode	ReadWriteOnce (RWO): Может быть смонтирован одним узлом в режиме чтения-записи.

Параметр	Описание
Allocation	Этот тип persistent volume claim может быть создан только в определённых проектах.
Project	Если проект временно не назначен, его можно Обновить позже.

7. Нажмите **Next** и дождитесь завершения создания ресурсов.

Руководства

Управление устройствами

Предварительные требования

Добавление устройств

Мониторинг и оповещения

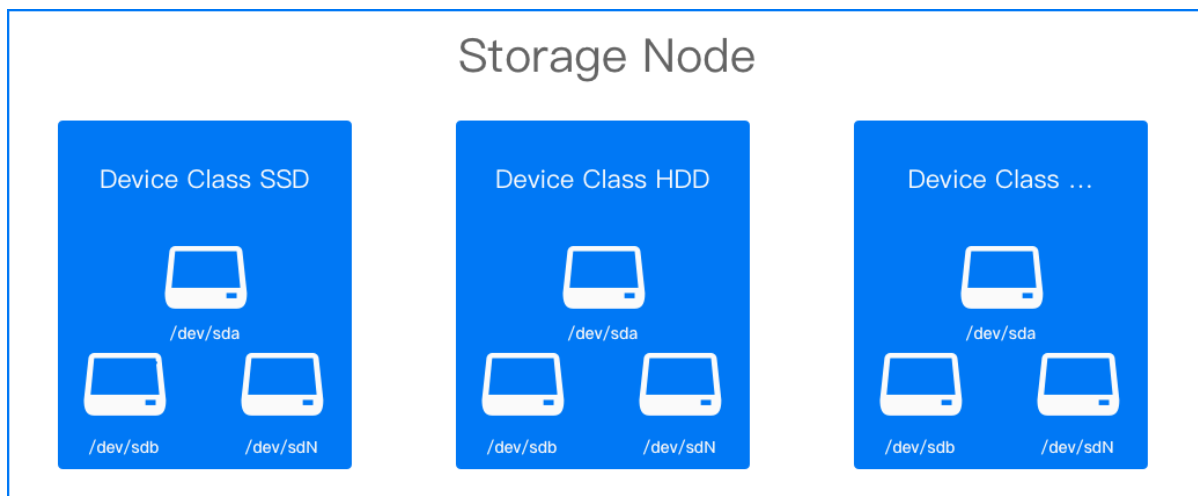
Мониторинг

Оповещения

Управление устройствами

Будь то для первоначального развертывания или расширения ресурсов, необходимо сопоставить доступные диски на узле с устройствами хранения для использования и управления.

Устройства хранения с похожими характеристиками обычно используются централизованно, и такие устройства классифицируются в локальном хранилище под **Device Classes**. Использование device classes эквивалентно прямому использованию дисков, обеспечивая нулевые потери и высокую производительность, а также снижая осведомленность приложений и зависимость от конкретных устройств.



Содержание

[Предварительные требования](#)

[Добавление устройств](#)

Предварительные требования

- При создании кластера локального хранилища должен быть добавлен как минимум 1 [Device Class \(deviceClasses.classes\)](#), включая устройства в этом device class.
- На узле должен присутствовать как минимум 1 голый диск.

Добавление устройств

1. Перейдите в раздел **Administrator**.
2. В левой навигационной панели нажмите **Storage Management > Local Storage**.
3. На вкладке **Details** нажмите **Add Storage Node**.
4. Настройте соответствующие параметры согласно приведённым ниже инструкциям.

Параметр	Описание
Storage Node	Узел, на котором имеется как минимум 1 голый диск.
Device Class	Каждый device class соответствует группе устройств хранения с одинаковыми характеристиками; рекомендуется называть его в соответствии с типом дисков, например, <i>hdd</i> , <i>ssd</i> .
Storage Device	Например, <i>/dev/sda</i> . Если дисков несколько, их можно добавлять по одному. Примечание: Устройство хранения должно быть целым жёстким диском, а не разделом на диске, так как это вызовет ошибки.

5. Нажмите **Add**.

Примечание: Если статус device class отображается как `Unavailable` из-за отсутствия добавленных устройств, можно продолжить выполнение следующих операций.

6. Перейдите на вкладку **Storage Devices** и нажмите **Add Storage Device**.
7. Добавьте устройства согласно подсказкам на интерфейсе.

8. Нажмите **Add**.

Мониторинг и оповещения

Локальное хранилище предоставляет готовые возможности для сбора метрик мониторинга и настройки оповещений. После включения компонента мониторинга платформы можно настроить мониторинг и оповещения на основе кластеров хранения, производительности и ёмкости хранилища с поддержкой настройки политик уведомлений.

Интуитивно представленные данные мониторинга можно использовать для поддержки принятия решений при операционных проверках или настройке производительности, а комплексный механизм оповещений поможет обеспечить стабильную работу системы хранения.

Содержание

Мониторинг

- Мониторинг производительности

- Мониторинг ёмкости

Оповещения

- Настройка уведомлений

- Обработка оповещений

- Анализ после инцидента

Мониторинг

Мониторинг производительности

По умолчанию платформа собирает часто используемые метрики мониторинга производительности, такие как пропускная способность чтения и записи, IOPS и задержки для локального хранилища. Данные мониторинга в реальном времени по этим метрикам можно просмотреть на вкладке **Monitoring** страницы **Local Storage** в разделе **Storage Management**. Платформа визуально отображает эти метрики с помощью графиков и диаграмм, что позволяет администраторам чётко наблюдать текущую производительность хранилища и быстро выявлять потенциальные проблемы.

Мониторинг ёмкости

Поскольку локальное хранилище может использовать только локально доступные ресурсы хранения на узлах, пользователи должны убедиться в наличии достаточного объёма свободной ёмкости на узлах перед объявлением локального хранилища, чтобы избежать проблем, вызванных чрезмерным объявлением.

Для помощи в этом платформа предоставляет подробный мониторинг ёмкости в разделе **Details** локального хранилища, классифицированный по типам устройств. Пользователи могут проверить доступное пространство для хранения, чётко отображаемое в числовом и графическом форматах. Если у какого-либо типа устройств наблюдается недостаток доступной ёмкости, следует освободить место или добавить дополнительные дисковые устройства перед использованием локального хранилища.

Оповещения

Платформа включает набор предустановленных политик оповещений. Если ресурсы становятся аномальными или данные мониторинга достигают порога предупреждения, оповещения автоматически срабатывают. Преднастроенные политики оповещений эффективно покрывают распространённые операционные потребности, включая оповещения о состоянии здоровья кластера и ёмкости по типам устройств.

Настройка уведомлений

Для своевременного получения оповещений следует настроить политики уведомлений в центре операций. Уведомления могут отправляться по электронной почте, SMS или другими способами соответствующим сотрудникам, что побуждает к немедленному реагированию для устранения проблем или предотвращения сбоев. Пользователи могут получить доступ к настройкам политик уведомлений напрямую из интерфейса центра операций. Подробные инструкции по настройке оповещений доступны в документации [Creating Alert Policies].

Обработка оповещений

- Если состояние здоровья кластера хранения изменяется на **Alert**, администраторы должны немедленно провести расследование. Раздел **Details** предоставляет информацию для устранения и решения этих проблем. Распространённые причины включают аномалии в службах узлов или проблемы с конкретными типами устройств.

Пункт проверки	Соответствующее состояние	Причина
Состояние здоровья	Alert	Вызвано аномалиями в службах узлов или проблемами с типами устройств.
Состояние службы	Unknown	Узел находится в состоянии notready , возможно из-за сбоев сети или отключения питания.
Состояние типа устройства	Unavailable	Используемый диск может не быть raw-диском или отсутствовать.

- Оповещения в реальном времени, срабатывающие на вкладке **Alert**, требуют оперативного внимания, даже если текущее состояние кластера хранения отображается как **Healthy**. Быстрая реакция предотвращает развитие более серьёзных проблем. В следующей таблице приведены уровни оповещений и их значение:

Уровень оповещения	Значение
Critical	Указывает на серьёзные проблемы, вызывающие перебои в работе платформы или потерю данных с тяжёлыми последствиями.
Major	Известные проблемы, которые могут повлиять на функциональность платформы и нормальное ведение бизнеса.
Warning	Существует риск операционных проблем; требуется своевременное вмешательство, чтобы избежать влияния на нормальную работу.

Анализ после инцидента

В журнале **Alert History** фиксируются все ранее сработавшие оповещения, которые больше не требуют немедленных действий. При проведении анализа после инцидента следует учитывать следующее:

- Какие конкретные аномалии наблюдались во время инцидента?
- Есть ли повторяющиеся шаблоны определённых оповещений? Как можно проактивно предотвратить их в будущем?
- Был ли всплеск оповещений в определённые периоды, связанный с внешними факторами или операционными инцидентами? Следует ли скорректировать операционные стратегии соответственно?

Как сделать

[Резервное копирование и восстановление ToroLVM с помощью Velero](#)

Предварительные требования

Ограничения

Процедура

[Настройка полосатых логических томов](#)

Предварительные требования

Процедура

Резервное копирование и восстановление PVC файловой системы TopoLVM с помощью Velero

Velero позволяет выполнять резервное копирование и восстановление Persistent Volume Claims (PVC) и Persistent Volumes (PV) для файловых систем TopoLVM. Эта функциональность интегрирована в платформу.

Данное руководство применимо исключительно к PVC файловой системы TopoLVM.

Содержание

[Предварительные требования](#)

Ограничения

Процедура

Шаг 1: Настройка репозитория резервных копий

Шаг 2: Выполнение резервного копирования

Шаг 3: Восстановление кластера

Предварительные требования

1. Разверните «Alauda Container Platform Data Backup for Velero» через Marketplace/Cluster Plugins.

2. Настройте S3-совместимое хранилище для `BackupStorageLocation` Velero.
Используйте предоставленное платформой объектное хранилище S3 или MinIO.

Ограничения

1. В S3-хранилище должно быть достаточно свободного места для хранения всех данных PV из целевого кластера.
2. При восстановлении квота namespace и класс хранения должны поддерживать общий объем всех PVC.

Процедура

Шаг 1: Настройка репозитория резервных копий

1. Убедитесь, что доступно S3-совместимое хранилище.
2. Перейдите в **Administrator > Cluster Management > Backup and Restore > Backup Repository**.
3. Создайте репозиторий резервных копий, используя учетные данные объектного хранилища.

Шаг 2: Выполнение резервного копирования

1. Пометьте PVC и связанные с ними pod'ы, которые необходимо сохранить:
Velero для восстановления PVC файловой системы требует pod, который монтирует PVC для импорта данных; без pod PVC останется в состоянии Pending. Для сложных приложений приостановите работу приложения и присоедините PVC к легковесному pod (например, Nginx) для резервного копирования/восстановления, затем после восстановления верните исходную конфигурацию приложения.

```
kubectl label pvc -n <namespace> <pvc-name> velero-backup=true
kubectl label pod -n <namespace> <pod-name> velero-backup=true
```

2. Перейдите в **Backup and Restore** и создайте новую резервную копию:

- Выберите **Backup Kubernetes Resources and PVC Data Volumes**.
- Укажите namespaces, содержащие данные для резервного копирования.
- Настройте резервное копирование с параметрами:

```
apiVersion: velero.io/v1
kind: Schedule
metadata:
  name: <backup-name>
  namespace: cpaas-system
  annotations:
    cpaas.io/description: ''
spec:
  template:
    includedNamespaces:
      - <namespace>
    includedResources:
      - '*'
    labelSelector:
      matchLabels:
        velero-backup: 'true'
    excludedNamespaces: []
    excludedResources: []
    defaultVolumesToFsBackup: true
    storageLocation: default
    ttl: 720h
    schedule: '@every 876000h'
    skipImmediately: false
status:
  phase: Enabled
```

3. После завершения резервного копирования проверьте данные в S3-бакете (например, MinIO):

```
mc ls <minio-alias>/<bucket-name>/<backup-path>/<namespace>/
```

Пример вывода:

```
[2025-03-14 00:18:33 CST] 155B STANDARD config
[2025-03-14 09:04:56 CST] 0B data/
[2025-03-14 09:04:56 CST] 0B index/
[2025-03-14 09:04:56 CST] 0B keys/
[2025-03-14 09:04:56 CST] 0B snapshots/
```

Шаг 3: Восстановление кластера

1. В целевом кластере настройте тот же S3-бакет, который использовался для резервного копирования. Velero автоматически обнаружит существующую резервную копию.
2. Перейдите в **Backup and Restore** и создайте задачу восстановления:
 - Выберите namespace(ы) для восстановления.
 - В расширенных настройках при необходимости сопоставьте исходный namespace с целевым.
3. Запустите операцию восстановления.
4. После восстановления проверьте:
 - Совпадение имен PVC с исходным кластером.
 - Целостность и доступность данных приложений в PVC.

Настройка полосатых логических томов

Когда вы записываете данные в логический том LVM, файловая система распределяет данные по базовым физическим томам. Вы можете контролировать способ записи данных на физические тома, создав полосатый логический том. Для больших последовательных операций чтения и записи это может повысить эффективность ввода-вывода данных.

Полосатость улучшает производительность за счёт записи данных на заранее определённое количество физических томов поочерёдно. При полосатости ввод-вывод может выполняться параллельно. В некоторых случаях это может привести к почти линейному увеличению производительности с каждым дополнительным физическим томом в полосе.

ТороLVM достигает этого, задавая параметр `lvcreate-option-class` в `StorageClass`.

Содержание

Предварительные требования

Процедура

Использование параметра `lvcreate-option-class` по умолчанию

Создание пользовательского `lvcreate-option-class` (необязательно)

Предварительные требования

- Класс устройств должен содержать **не менее двух устройств** на одном узле.

Процедура

1 Использование параметра `lvcreate-option-class` по умолчанию

1. Перейдите в раздел **Administrator**.
2. В левой боковой панели выберите **Storage Management > Storage Classes**.
3. Нажмите **Create Storage Class**.
4. Выберите **Block Storage**.
5. Выберите **TopoLVM**, затем нажмите **Next**.
6. Настройте необходимые параметры класса хранения.
7. Переключитесь в **YAML view** и добавьте параметр `topolvm.io/lvcreate-option-class`:

```
parameters:  
  topolvm.io/lvcreate-option-class: striped-default
```

NOTE

`striped-default` — встроенный `lvcreate-option-class` в Alauda Container Platform. Он автоматически добавляет `--stripes=2` и `--stripesize=64` к команде `lvcreate` при создании логических томов драйвером TopoLVM CSI.

2 Создание пользовательского `lvcreate-option-class` (необязательно)

Если встроенный класс `striped-default` не соответствует вашим требованиям, вы можете определить собственный `LVCreateOptionClass`:

```
cat << EOF | kubectl apply -f -
apiVersion: topolvm.cybozu.com/v2
kind: LVCreateOption
metadata:
  name: custom
  namespace: nativestor-system
spec:
  options:
  - name: striped-custom
    type: striped
    options:
    - --stripes=3
    - --stripesize=64
EOF
```

Этот манифест создаёт пользовательский LVCreateOptionClass с именем striped-custom, который задаёт 3 полосы и размер полосы 64 КиБ. После применения вы можете сослаться на него в вашем StorageClass с помощью

```
topolvm.io/lvcreate-option-class: striped-custom .
```