Menu

Хранение

Распределённое хранилище Ceph

Введение

Обзор возможностей

Сравнение решений для хранения

Установка

Архитектура

Техническая архитектура

Основные понятия

Руководства

Как сделать

MinIO Object Storage

Введение

Установка

Предварительные требования

Процедура

Связанная информация

Архитектура

Основные компоненты:

Архитектура развертывания:

Масштабирование с несколькими пулами:

Заключение:

Основные понятия

Руководства

Как сделать

Локальное хранилище TopoLVM

Введение

Установка

Предварительные требования

Процедура

Руководства

Как сделать

Menu

Распределённое хранилище Ceph

Введение

Введение

Обзор возможностей

Сравнение решений для хранения

Установка

Создание кластера стандартного типа

Предварительные требования

Важные замечания

Порядок действий

Связанные операции

Создание Stretch Type кластера

Терминология

Типовая схема развертывания

Ограничения и требования

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 компонентов на конкретных узлах

Обновление конфигурации развертывания компонентов

Перезапуск компонентов хранилища

Добавление устройств/классов устройств

Добавление классов устройств

Добавление устройств

Статус жесткого диска

Мониторинг и оповещения

Мониторинг

Оповещения

Как сделать

Настройка выделенного кластера для распределённого хранилища

Архитектура

Требования к инфраструктуре

Процедура

Последующие действия

Очистка распределённого хранилища

Меры предосторожности

Процедура

Восстановление после сбоев

Обновление параметров оптимизации

Процедура

Создание пользователя ceph object store

Предварительные условия

Процедура

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 этого кластера, а **внешнее хранилище** — Ceph-хранилище вне этого кластера.

Содержание

Обзор возможностей

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

- **Профессиональная эксплуатация**: Предлагает 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**: Распределённые и многократные механизмы репликации обеспечивают безопасность и надёжность данных; простое и надёжное автоматизированное управление поддерживает онлайн-расширение ресурсов хранения.
- **Отличная производительность**: Обеспечивает 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; конфигурация может быть выполнена на бизнес-кластере, что экономит затраты.

Доступ к внешнему хранилищу

Вариант 1: Доступ к ресурсам распределённого хранилища других бизнес-кластеров внутри платформы для обеспечения изоляции хранения и бизнеса, что облегчает управление и обслуживание.

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

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

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

Menu

Установка

Создание кластера стандартного типа

Предварительные требования

Важные замечания

Порядок действий

Связанные операции

Создание Stretch Type кластера

Терминология

Типовая схема развертывания

Ограничения и требования

Предварительные условия

Процедура

Связанные операции

Создание кластера стандартного типа

Кластер стандартного типа — это наиболее типичный способ развертывания хранилища Ceph. 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

- **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 минимум 3 узла.
- На каждом узле должен быть как минимум 1 пустой жесткий диск или 1 неформатированный раздел жесткого диска.
- Рекомендуемый объем доступного жесткого диска — более 50 ГБ.
- Если вы используете присоединённый 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`, чтобы обеспечить успешное развертывание распределённого 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 распределённое хранилище Ceph на вашем кастомном Kubernetes-кластере с Containerd в качестве runtime, также требуется установить параметр `LimitNOFILE` в файле `/etc/systemd/system/containerd.service` на всех узлах кластера в значение `1048576`.

Важные замечания

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

2 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 к следующему шагу означает успешное развертывание Operator.
- Если развертывание не удалось, следуйте подсказке на интерфейсе **Clean Up Deployed Information and Retry** и повторите развертывание Operator; чтобы вернуться на страницу выбора распределённого хранилища, нажмите **Application Store**, сначала удалите ресурсы уже развернутого **rook-operator**, затем удалите сам **rook-operator**.

3 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	Классы устройств — это группировки жестких дисков; вы можете настраивать классы устройств в соответствии с вашими потребностями хранения, распределяя разный контент по дискам с разной производительностью. • **Default Device Class**: Платформа автоматически классифицирует типы жестких дисков в узлах кластера, например, создавая классы устройств с именами `hdd`, `ssd`, `nvme`. • **Custom Device Class**: Настройка имени 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 `hdd`, `ssd`, `nvme`.
Device Class - Storage Devices	Выберите **Blank Hard Disk** или **Unformatted Hard Disk Partition** на узлах.

Параметр	Описание
	• Если переключатель "Open All Blank Devices" включен: все пустые устройства на узле будут добавлены в класс устройств; • Если переключатель выключен: вручную введите имена пустых устройств на узле, например, `sda`.
Snapshot	При включении поддерживается создание снимков PVC и использование снимков для настройки новых PVC для быстрого резервного копирования и восстановления бизнес-данных. Если при создании хранилища снимки не были включены, их можно включить при необходимости в разделе **Operations** на странице деталей кластера хранения. **Примечание**: Перед использованием убедитесь, что для текущего кластера развернуты плагины volume snapshot.
Monitoring Alarm	При включении обеспечивается встроенный сбор метрик мониторинга и возможности оповещений, см. Monitoring and Alarming. **Примечание**: Если не включать сейчас, потребуется использовать альтернативные решения для мониторинга и оповещений хранилища, например, вручную настраивать панели мониторинга и стратегии оповещений в центре эксплуатации и обслуживания.

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

Параметр	Описание
Network Configuration	• **Host Network**: Кластер хранения будет использовать сеть хоста, необходимо заполнить соответствующие параметры оптимизации сети в колонке параметров оптимизации, например, настроить подсети `public` и `cluster`. Если оставить

Параметр	Описание
	пустым, будет использована подсеть хоста по умолчанию. **Примечание**: Использование сети хоста может нести риски безопасности из-за передачи данных в незашифрованном (открытом) виде через порты хоста. Обратитесь в службу поддержки платформы для получения решения по шифрованной передаче. • **Container Network**: Кластер хранения будет использовать контейнерную сеть; можно создать подсети в управлении сетью и назначить их пространству имён `rook-ceph`. Если оставить пустым, будет использована подсеть по умолчанию. **Примечание**: IPv6 не поддерживается. При использовании контейнерной сети доступ к хранилищу возможен только внутри кластера. Отказы или перезапуски Pod Ceph CSI могут привести к прерыванию сервиса.
Optimization Parameters	Поддерживается заполнение параметров в формате конфигурационного файла Ceph; система перезапишет параметры по умолчанию на основе предоставленного содержимого. **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 фиксированной точки развертывания: MON, MGR, MDS, RGW.

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

4 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	• Файловое хранилище: обеспечивает безопасные, надежные и масштабируемые услуги совместного файлового хранения. Подходит для совместного использования файлов, 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 снижается эффективность использования хранилища. Обычно устанавливается значение 3, что удовлетворяет большинству потребностей.
Device Class	Единообразно классифицируйте типы для одного типа устройств или дисков с одинаковой бизнес-логикой, выбирая из классов устройств, добавленных на предыдущем шаге. • При выборе класса устройств данные будут храниться в выбранном классе.

Параметр	Описание
	• Если класс устройств не выбран, данные будут случайным образом распределены по всем устройствам пула хранения.

Если это объектное хранилище, необходимо также настроить следующие параметры:

Параметр	Описание
Region	Укажите регион, в котором расположен пул хранения.
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 к следующему шагу означает успешное развертывание пула хранения.
- Если развертывание не удалось, следуйте подсказкам интерфейса для проверки основных компонентов, 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 распределённого 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, Ceph всё равно может поддерживать доступность.

Содержание

Терминология

Термин	Объяснение
Quorum Availability Zone	Обычно располагается в отдельной зоне, не несущей основные рабочие нагрузки, сосредоточена на поддержании согласованности кластера и используется преимущественно для принятия арбитражных решений при сбоях в основном дата-центре или сетевых разрывах.
Data Availability Zone	Основная зона в кластере Ceph, где фактически хранится и обрабатывается данные, несущая операционные нагрузки и задачи хранения данных, вместе с зоной кворума формирующая полноценную высокодоступную систему хранения.

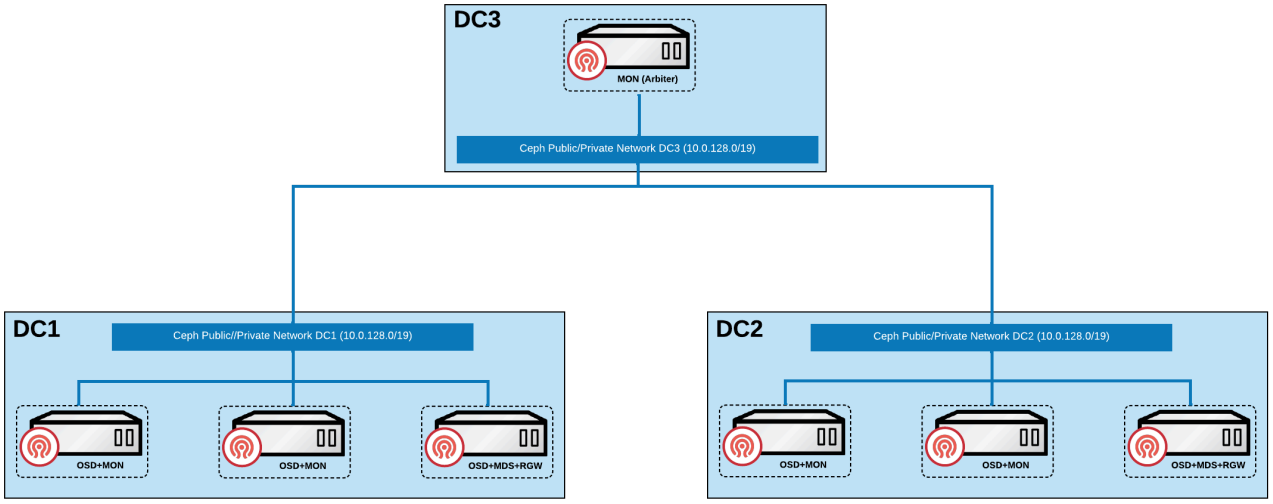
Типовая схема развертывания

Ниже приведена типовая схема развертывания stretch кластера 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 определит эту зону как находящуюся в состоянии сбоя.
- В зоне quorum availability достаточно развернуть 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 две реплики, 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 проводить арбитраж между двумя зонами data availability на основе предустановленных политик, выбирая зону с лучшим состоянием для продолжения предоставления сервисов в качестве основной зоны данных.

Ограничения и требования

- **Ограничения по пулам хранения**: Пулы с кодированием с исправлением ошибок (erasure-coded) не поддерживаются, для защиты данных можно использовать только механизм реплик.
- **Ограничения по классификации устройств**: Функциональность device class не поддерживается, стратификация хранения по характеристикам устройств

невозможна.

- **Ограничения по региональному развертыванию**: Поддерживается только две зоны data availability; не допускается более двух таких зон.
- **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 задержкам в сети**: RTT (время туда-обратно) 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 не менее 5 узлов между одной зоной quorum availability и двумя зонами 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 не менее 4 вычислительных узлов по 2 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. Зайдите 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, чтобы классифицировать их в указанную зону доступности. Здесь вместо <zone> укажите имя зоны доступности.

2 Создание сервиса хранения

В этом документе описаны только параметры, отличающиеся от стандартных кластеров; по остальным параметрам смотрите Create Standard Type Cluster.

Создание кластера

Параметр	Описание
Cluster Type	Выберите **Stretch**.
Quorum Availability Zone	Выберите имя зоны quorum availability.
Data Availability Zone	Выберите имена зон доступности и укажите узлы.

Создание пула хранения

Параметр	Описание
Number of Replicas	По умолчанию 4.

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

Связанные операции

Создание стандартного кластера

Подробности смотрите в Create Standard Type Cluster.

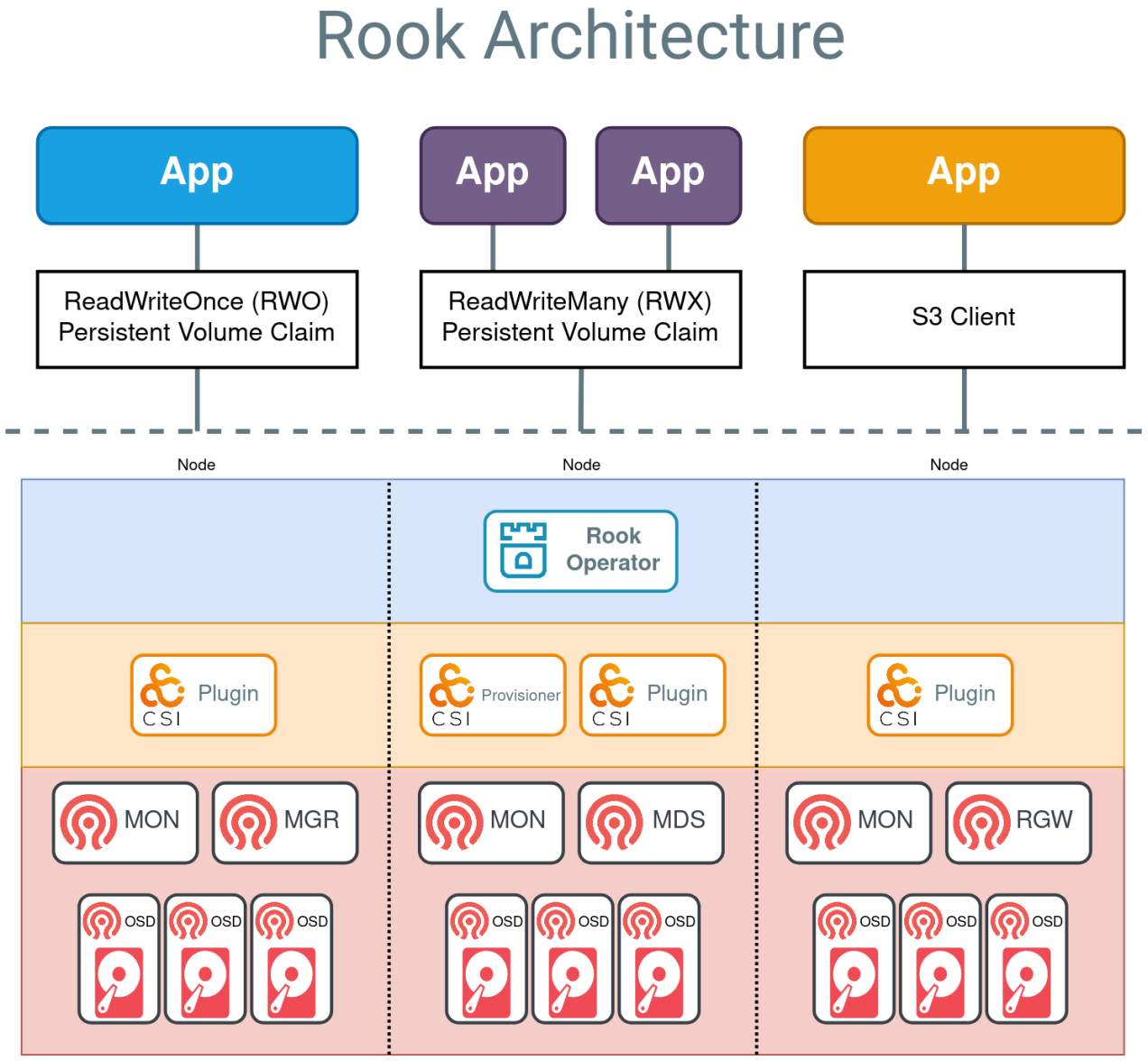
Очистка распределённого хранилища

Подробности смотрите в Cleanup Distributed Storage.

Архитектура

Содержание

Техническая архитектура



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 активно 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 S3 клиента.

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 создание 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 создания хранилища.
- K8s вызывает provisioner Ceph-CSI RBD для создания образа Ceph RBD.
- kubelet вызывает CSI плагин RBD для монтирования тома в приложении.
- Том становится 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 создания хранилища.
- K8s вызывает provisioner Ceph-CSI CephFS для создания подтома CephFS.
- kubelet вызывает CSI плагин CephFS для монтирования тома в приложении.
- Том становится доступен для чтения и записи.
- Том ReadWriteMany может быть смонтирован на нескольких узлах для использования приложением.

Объектное хранилище S3

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

- (Оранжевое) приложение создаёт BucketClaim для запроса бакета.
- Драйвер Ceph COSI создаёт бакет Ceph RGW.
- Драйвер Ceph COSI 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 S3 клиента.

Menu

Основные концепции

Основные концепции

Rook Operator

Ceph CSI

Функции модулей Ceph

Основные концепции

Содержание

Rook Operator

Оператор Rook — это простой контейнер, который содержит всё необходимое для инициализации и мониторинга кластера хранения. Оператор запускает и контролирует поды мониторов Ceph, демоны Ceph OSD для предоставления хранилища RADOS, а также запускает и управляет другими демонами Ceph. 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. Мониторы Ceph будут запускаться или переключаться при необходимости, а также вноситься другие корректировки по мере роста или уменьшения кластера. Оператор также отслеживает изменения желаемого состояния, указанные в пользовательских ресурсах Ceph (CR), и применяет эти изменения.

Rook автоматически настраивает драйвер Ceph-CSI для монтирования хранилища к вашим подам. Образ rook/ceph включает все необходимые инструменты для управления кластером.

Ceph CSI

Плагины Ceph CSI реализуют интерфейс между CSI-совместимым оркестратором контейнеров (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 выполняет роль лидера 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 хранилища (OSD) хранят фактические пользовательские данные. Каждый OSD обычно привязан к одному физическому диску. OSD обрабатывают запросы на чтение и 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 прямые сервисы данных.

Модуль	Функция
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 виде объектов через RADOS, независимо от типа данных. Слой RADOS обеспечивает согласованность и надёжность данных через репликацию, обнаружение и восстановление сбоев, а также восстановление данных между узлами кластера.
LIBRADOS	Librados — это метод, упрощающий доступ к RADOS. В настоящее время он поддерживает языки программирования PHP, Ruby, Java, Python, C и C++. Он предоставляет RADOS — локальный интерфейс к кластеру 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.
RBD	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.
CephFS	CephFS предоставляет распределённую файловую систему, совместимую с POSIX, любого размера. Она зависит от Ceph MDS для отслеживания иерархии файлов, то есть метаданных.

Menu

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 компонентов на конкретных узлах

Обновление конфигурации развертывания компонентов

Перезапуск компонентов хранилища

Добавление устройств/классов устройств

Добавление классов устройств

Добавление устройств

Статус жесткого диска

Мониторинг и оповещения

Доступ к сервисам хранения

Доступ к сервисам хранения поддерживает два метода интеграции: во-первых, интеграция распределённых ресурсов хранения из других бизнес-кластеров внутри платформы для обеспечения изоляции хранения и бизнеса, что упрощает управление и обслуживание; во-вторых, подключение внешних Ceph-ресурсов хранения для использования распределённого хранения.

Содержание

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 имя storage pool для последующей интеграции.
- Вне платформы создано внешнее Ceph-хранилище (версия ≥ 14.2.3) с созданным storage pool. Запишите имя 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 — внешнее Ceph, необходимо получить данные аутентификации с помощью следующих команд.

Параметр	Способ получения
FSID	`ceph fsid`
Информация о MON-компонентах	`ceph mon dump` Должна быть в формате {name= IP}, например *a=192.168.100.100:6789* .

Параметр	Способ получения
Админский ключ	`ceph auth get-key client.admin`
Storage Pool	• Файловое хранилище: используйте команду `ceph fs ls` для получения значения `name`. • Блочное хранилище: `ceph osd dump` | `grep "application rbd"` | `awk '{print $3}'`
Data Storage Pool	(только для файлового хранилища) используйте команду `ceph fs ls` для получения значения `data pools`.

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 — `stable`.
Installation Mode	`Cluster`: все пространства имён в кластере используют один экземпляр Operator для создания и управления, что

Параметр	Рекомендуемая конфигурация
	снижает использование ресурсов.
Installation Place	Выберите `Recommended`, Namespace поддерживается только **acp-storage**.
Upgrade Strategy	`Manual`: 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	При включении поддерживается создание снимков PVC и использование снимков для настройки новых PVC для быстрого резервного копирования и восстановления бизнес-данных. Если при доступе к хранилищу снимки не были включены, их можно включить позже в разделе **Operations** на странице деталей кластера хранения по мере необходимости. **Примечание**: перед использованием убедитесь, что для текущего кластера развернут плагин volume snapshot.
Network Configuration	• **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

Параметр	Описание
	подсеть в управлении сетью и назначить её пространству имён `rook-ceph`. Если оставить пустым, будет использоваться подсеть по умолчанию.
Other Parameters	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	В зависимости от типа созданного выше storage pool, по умолчанию будет соответствующий класс хранения: • Файловое хранилище: CephFS File Storage • Блочное хранилище: CephRBD Block Storage
Reclaim Policy	Политика освобождения для persistent volumes. • Delete: при удалении persistent volume claim будет также удалён связанный persistent volume. • Retain: даже при удалении persistent volume claim связанный persistent volume сохраняется.
Project Allocation	Проекты, которые могут использовать этот тип хранилища. Если в данный момент нет проектов, требующих этот тип хранилища, можно не выделять проекты сейчас и обновить позже.

5. Дождитесь примерно 1-5 минут для успешной интеграции.

Последующие действия

- Создание классов хранения: CephFS File Storage, CephRBD Block Storage
- Разработчики, использующие указанные классы хранения для создания persistent volume claims, могут расширять функционал с помощью снимков томов и масштабирования.

Примечание: если необходимо поддерживать storage pool, 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 различные бизнес-требования.

Содержание

Создание Storage Pool

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

Совет: В рамках одного кластера хранения допускается только один файловый 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	Выберите тип хранилища, который ещё не развернут. - File Storage: Обеспечивает безопасные, надёжные и масштабируемые услуги общего файлового хранилища. 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.д. - Object Storage: Предоставляет стандартный интерфейс S3, подходит для big data, резервного архивирования, облачных сервисов хранения и т.д.
Replica Count	• **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	• **Для кластера типа Standard**: Выберите уже добавленный класс устройств в созданном storage pool.
	  • При выборе класса устройств данные будут храниться в выбранном классе.
	  • Если класс устройств не выбран, данные будут случайным образом распределены по всем устройствам в storage pool.

Параметр	Описание
	• **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 ресурсных инстансов для object storage.

5. Нажмите **Create**.

Удаление Storage Pool

Если определённый тип хранилища больше не нужен, storage pool можно удалить после отвязки его 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 на ⋮ 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 на ⋮ рядом с object storage pool и выберите **View Address**.

Развертывание компонентов на конкретных узлах

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

Содержание

Обновление конфигурации развертывания компонентов

Меры предосторожности

- Обновление конфигурации вызовет автоматическую пересборку экземпляров компонентов системой, что может повлиять на доступ сервисов к системе хранения. 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 требованиями бизнеса и разверните компоненты на указанных узлах. Количество узлов должно быть не менее трёх для обеспечения минимальной доступности. Компоненты, для которых применима настройка фиксированного развертывания, включают MON, MGR, MDS, RGW.
5. Нажмите **Update**, и компоненты начнут планироваться на указанные узлы.

Перезапуск компонентов хранилища

При удалении развернутых компонентов хранилища система автоматически переназначит и повторно развернёт компоненты на узлах согласно текущей стратегии 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

Объединяйте классификацию устройств одного типа или жестких дисков с одинаковой бизнес-логикой в узлах кластера, настраивайте классы устройств в соответствии с требованиями к хранению и распределяйте различное содержимое хранения по разным типам дисков.

Примечания

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

Процедура

1. Войдите под пользователем **Administrator**.
2. В левой навигационной панели выберите **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: `hdd`, `ssd`, `nvme`.
Storage Devices	Выберите **Blank Disks** или **Unformatted Disk Partitions** в узле. • Если переключатель для всех пустых устройств включен: все пустые устройства в узле будут добавлены в этот класс устройств; • Если переключатель для всех пустых устройств выключен: вручную введите имена пустых устройств в узле, например, `sda`.

Добавление устройств

Отображайте 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 под пользователем **Administrator**.
2. В левой навигационной панели выберите **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
Node Type	Выберите тип узла, в котором находится жесткий диск, который вы хотите добавить как устройство хранения. **Compute Node**: Узел, в котором еще не добавлены устройства хранения. **Storage Node**: Узел, в котором уже добавлены устройства хранения.
Add Type	Выберите способ добавления жестких дисков в качестве устройств хранения. **All Empty Disks**: Добавить все неразмеченные смонтированные диски в узле как устройства хранения. **Specified Disks**: Добавить выбранные диски в узле как устройства хранения, включая пустые диски или уже размеченные смонтированные диски. Если тип узла — **Storage Node**, можно выбрать только **Specified Disks**.
Specified Disks	Если выбран тип добавления **Specified Disks**, введите имена всех жестких дисков, которые необходимо добавить как устройства хранения, например, `sda`, `sdb`. После ввода каждого имени нажмите Enter для подтверждения. **Примечание**: Рекомендуется использовать весь жесткий диск в качестве устройства хранения, а не отдельные разделы на диске.

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 по сбору метрик мониторинга и уведомлению 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 оптимизации производительности, а комплексный механизм оповещений поможет обеспечить стабильную работу системы хранения.

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

Содержание

Мониторинг

Платформа автоматически собирает распространённые 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, storage pool и OSD. Кроме того, можно мониторить задержки чтения и записи конкретно 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. Предустановленные 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 кластер хранения находится в состоянии `Warning`, это означает, что сработало оповещение, и связанная аномалия может привести к сбою. Пожалуйста, оперативно проверьте детали в разделе **Real-time Alerts** и выявите и устраните неисправность на основе причины.
- Если кластер хранения находится в состоянии `Failure`, это указывает на то, что кластер хранения не может нормально функционировать. Необходимо немедленно локализовать проблему и провести устранение неисправности.

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

Уровень оповещения	Значение
Disaster	Ресурс, соответствующий правилу оповещения, вышел из строя, что вызвало прерывание работы платформы, потерю данных и значительное воздействие.
Severe	Ресурс, соответствующий правилу оповещения, имеет известные проблемы, которые могут привести к сбоям функций платформы и повлиять на нормальную работу сервисов.
Warning	Ресурс, соответствующий правилу оповещения, подвергается операционным рискам, которые могут повлиять на нормальную работу сервисов при отсутствии своевременных действий.

Анализ инцидентов

В разделе **Alert History** фиксируются все сработавшие оповещения, которые больше не требуют действий. При проведении анализа инцидентов с использованием истории оповещений для эффективного подведения итогов рекомендуется ответить на следующие вопросы.

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

Menu

Как сделать

Настройка выделенного кластера для распределённого хранилища

Очистка распределённого хранилища

Восстановление после сбоев

Восстановление после сбоев блочного хранилища

Терминология

Настройка резервного копирования

Переключение при сбое (Failover)

Восстановление после аварий в объектном хранилище

Терминология

Предварительные требования

Процедуры

Переключение при сбое (Failover)

Связанные операции

Обновление параметров оптимизации

Обновление параметров оптимизации

Процедура

Создание пользователя ceph object store

Создание пользователя ceph object store

Предварительные условия

Процедура

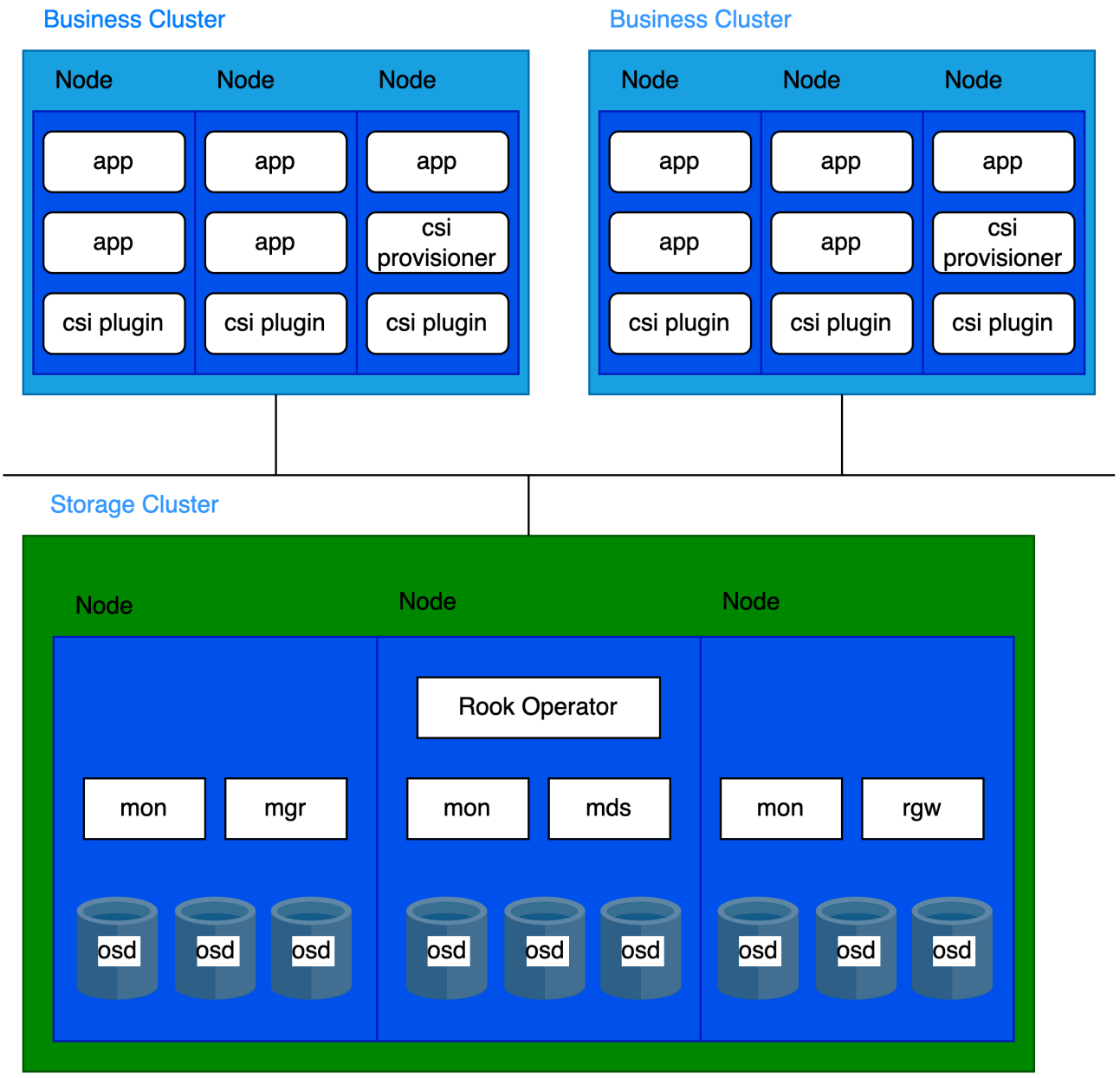
Настройка выделенного кластера для распределённого хранилища

Развёртывание выделенного кластера подразумевает использование отдельного кластера для развертывания 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 других бизнес-нагрузок. Такой раздельный подход к развёртыванию является рекомендуемой лучшей практикой для распределённого хранилища платформы.

Содержание

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 развёртывания 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 развёртывать 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 развёртыванием ёмкость хранилища должна быть спланирована в соответствии с конкретными бизнес-требованиями. По умолчанию система распределённого хранения использует стратегию избыточности с 3 репликами. Следовательно, полезная ёмкость рассчитывается как общая сырая ёмкость хранения (со всех устройств) делённая на 3.

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

Размер устройства хранения(D)	**Устройств хранения на узел(M)**	**Общая ёмкость(D*M*N)**	**Полезная ёмкость(D*M*N/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 потребление. 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 двух порогах:

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

Всегда своевременно 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, как и существующие узлы.

Требования к сети

Распределённое хранилище должно использовать **HostNetwork**.

Сетевая изоляция

Сеть делится на два типа:

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

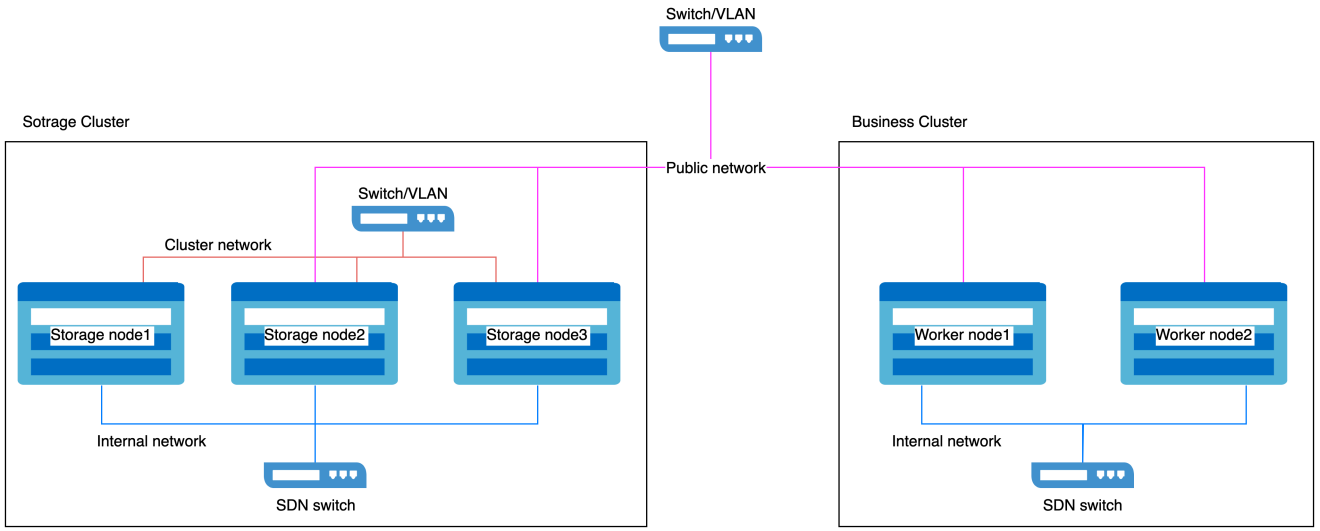
Для обеспечения качества сервиса и стабильности производительности:

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 хранилища.

Пример конфигурации сетевой изоляции



Требования к скорости сетевых интерфейсов

1. **Узлы хранения**

   - Для **публичной сети** и **кластерной сети** требуются сетевые интерфейсы 10GbE или выше.

2. **Узлы бизнес-кластеров**

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

Процедура

1 Развёртывание Operator

1. Перейдите в раздел **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 развёртывание не удалось, следуйте подсказке на интерфейсе **Clean Up Deployed Information and Retry** и повторно разверните Operator; чтобы 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, чтобы раскрыть

Параметры:

- **public network cidr**: CIDR публичной сети хранилища (например, `10.0.1.0/24`).
- **cluster network cidr**: CIDR кластерной сети хранилища (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 измениться после перезагрузки.

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, чтобы раскрыть

Создание блочного пула

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

▶ Нажмите, чтобы раскрыть

Создание объектного пула

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

▶ Нажмите, чтобы раскрыть

Последующие действия

Когда другие кластеры нуждаются в использовании сервиса распределённого хранения, 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

Перед очисткой rook-ceph убедитесь, что все ресурсы PVC и PV, использующие хранилище Ceph, были 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 хранения 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 область пула хранения покажет **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, включив 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 результаты, выполните следующую команду, 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 результаты, выполните следующую команду, 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, связанные с 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 бизнес-кластера, 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 начала очистки введите yes.

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

Menu

Восстановление после сбоев

Восстановление после сбоев файлового хранилища

Терминология

Настройка резервного копирования

Переключение при сбое

Восстановление после сбоев блочного хранилища

Терминология

Настройка резервного копирования

Переключение при сбое (Failover)

Восстановление после аварий в объектном хранилище

Терминология

Предварительные требования

Процедуры

Переключение при сбое (Failover)

Связанные операции

Восстановление после сбоев файлового хранилища

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

WARNING

- CephFS 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.
- CephFS Mirror обеспечивает только резервное копирование данных нижележащего хранилища и не может выполнять резервное копирование ресурсов Kubernetes. Для резервного копирования PVC и PV ресурсов используйте функцию платформы **Backup and Restore** совместно.

Содержание

Терминология

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

Настройка резервного копирования

Предварительные требования

- Подготовьте два кластера, подходящих для развертывания Alauda Build Rook-Ceph: Primary и Secondary, убедитесь, 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 Получение 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 Создание 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. Подробности см. в официальной документации ↗.
- `<retention-policy>`: Политика хранения снимков. Подробности см. в официальной документации ↗.

5 Развертывание Mirror Daemon в Primary кластере

Mirror Daemon непрерывно отслеживает изменения данных в пуле файлового хранилища (с включённым Mirror). Он периодически создаёт снимки и передаёт их дельты по сети в Secondary кластер.

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 можно напрямую продолжить использование CephFS в Secondary кластере.

Предварительные требования

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 первичный-резервный, обеспечивая быстрое переключение на резервный кластер при сбое первичного.

WARNING

- RBD Mirror выполняет инкрементальную синхронизацию на основе снимков, с интервалом создания снимков по умолчанию — один раз в час (настраивается). Дифференциальные данные между первичным 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 платформы **Backup and Restore**.

Содержание

Терминология

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

Настройка резервного копирования

Предварительные требования

- Подготовьте два кластера с возможностью развертывания Alauda Build Rook-Ceph: Primary и Secondary, с сетевым соединением между ними.
- Оба кластера должны работать на одной версии платформы (v3.12 или выше).
- Создайте распределённые сервисы хранения в кластерах Primary и Secondary.
- Создайте блочные пулы хранения с **одинаковыми именами** в обоих кластерах.

- Убедитесь, что следующие три образа загружены в приватный репозиторий образов платформы:
  - `quay.io/csiaddons/k8s-controller:v0.5.0`
  - `quay.io/csiaddons/k8s-sidecar:v0.8.0`
  - `quay.io/brancz/kube-rbac-proxy:v0.8.0`

Процедуры

1 Включение зеркалирования для блочного пула Primary кластера

Выполните следующую команду на управляющем узле Primary кластера:

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 блочного пула хранения.

2 Получение Peer Token

Этот токен служит ключевым учётным данным для установления зеркальных соединений между кластерами.

Выполните следующую команду на управляющем узле Primary кластера:

Command

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

Output

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

Параметры:

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

3 Создание секрета Peer Token в Secondary кластере

Выполните следующую команду на управляющем узле Secondary кластера:

Command

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

Output

```
secret/rbd-primary-site-secret created
```

Параметры:

- `<token>`: токен, полученный на Шаге 2.
- `<block-pool-name>`: имя блочного пула хранения.

4 Включение зеркалирования для блочного пула Secondary кластера

Выполните следующую команду на управляющем узле Secondary кластера:

Command

```
kubectl -n rook-ceph patch cephblockpool <block-pool-name> --type merge -p \
'{
  "spec": {
    "mirroring": {
      "enabled": true,
      "mode": "image",
      "peers": {
        "secretNames": [
          "rbd-primary-site-secret"
        ]
      }
    }
  }
}'
```

Output

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

Параметры:

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

5 Развёртывание демона зеркалирования в Secondary кластере

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

Выполните следующую команду на управляющем узле Secondary 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 Проверка статуса зеркалирования

Выполните следующую команду на управляющем узле Secondary кластера:

Command

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

Output

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

Параметры:

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

7 Включение репликационного sidecar

Эта функция обеспечивает эффективную репликацию и синхронизацию данных без прерывания работы основного приложения, повышая надёжность и доступность системы.

1. Развёртывание csiaddons-controller

Выполните следующие команды на управляющих узлах обоих кластеров — Primary и Secondary:

▶ Нажмите для просмотра

Параметры:

- `<registry>`: адрес реестра платформы.

1. Включение csi sidecar

Выполните следующие команды на управляющих узлах обоих кластеров — Primary и Secondary:

```
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"
  }
]'
```

8 Создание VolumeReplicationClass

Выполните следующие команды на управляющих узлах обоих кластеров — Primary и Secondary:

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 каждые 1 час).

9 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 необходимо включить зеркалирование.

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

Переключение при сбое (Failover)

При сбое Primary кластера необходимо переключить отношения первичный-резервный для RBD-образа.

Предварительные требования

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

Процедуры

Создание VolumeReplication

Выполните следующую команду на управляющем узле Secondary кластера:

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

1. `<vr-name>`: имя VolumeReplication.
2. `<namespace>`: namespace PVC.
3. `<mirror-pvc-name>`: имя PVC.

Примечание После создания RBD-образ в Secondary кластере становится первичным и доступен для записи.

Восстановление после аварий в объектном хранилище

Функция Ceph RGW Multi-Site представляет собой механизм асинхронной репликации данных между кластерами, предназначенный для синхронизации данных объектного хранилища между географически распределёнными кластерами Ceph, обеспечивая высокую доступность (HA) и возможности аварийного восстановления (DR).

Содержание

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	• Realm: Наивысший уровень логической группировки 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 группировка внутри Realm, содержащая несколько Zones. ZoneGroups обеспечивают синхронизацию и репликацию данных между Zones, обычно в пределах одного географического региона. • Zone: Логическая группировка 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 Primary и Secondary кластерах не развернуто объектное хранилище Ceph.
- Обратитесь к документации Create Storage Service для развертывания Operator и создания кластеров. Не создавайте пулы объектного хранилища через мастер после

создания кластера. Вместо этого используйте CLI-инструменты для настройки, как описано ниже.

Процедуры

Данное руководство описывает решение для синхронизации между двумя Zones в одном ZoneGroup.

1 Создание объектного хранилища в Primary кластере

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

Выполните следующие команды на управляющем узле Primary кластера:

Команда

```
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: <zonegroup-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: <zonegroup-name>
  metadataPool:
    failureDomain: host
    replicated:
      size: 3
      requireSafeReplicaSize: true
  dataPool:
    failureDomain: host
    replicated:
      size: 3
      requireSafeReplicaSize: true
    parameters:
      compression_mode: none
  preservePoolsOnDelete: false

---
cat << EOF | kubectl apply -f -
apiVersion: ceph.rook.io/v1
kind: CephObjectStore
```

```
metadata:
  name: <object-store-name>
  namespace: rook-ceph
spec:
  gateway:
    port: 7480
    instances: 2
  zone:
    name: <zone-name>
EOF
```

Вывод

```
cephobjectrealm.ceph.rook.io/<realm-name> создан
cephobjectzonegroup.ceph.rook.io/<zonegroup-name> создан
cephobjectzone.ceph.rook.io/<zone-name> создан
cephobjectstore.ceph.rook.io/<object-store-name> создан
```

Параметры :

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

2 Настройка внешнего доступа для Primary Zone

1. Получите UID ObjectStore {#uid}

```
kubectl -n rook-ceph get cephobjectstore <object-store-name> -o
jsonpath='{.metadata.uid}'
```

Параметры

- `<object-store-name>` : Имя шлюза, настроенное в Шаге 1.

2. Создайте Service для внешнего доступа

```
cat << EOF | kubectl apply -f -
apiVersion: v1
kind: Service
metadata:
  name: rook-ceph-rgw-<object-store-name>-external
  namespace: rook-ceph
  labels:
    app: rook-ceph-rgw
    rook_cluster: rook-ceph
    rook_object_store: <object-store-name>
  ownerReferences:
    - apiVersion: ceph.rook.io/v1
      kind: CephObjectStore
      name: <object-store-name>
      uid: <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: <object-store-name>
  sessionAffinity: None
  type: NodePort
EOF
```

Параметры:

- `<object-store-name>`: Имя шлюза, настроенное здесь.
- `<object-store-uid>`: UID, полученный здесь.

3. Добавьте внешние конечные точки в CephObjectZone.

```
kubectl -n rook-ceph patch cephobjectzone <primary-zone-name> --type merge -p
'{"spec":{"customEndpoints":["<external-endpoint>"]}}'
```

Параметры:

- `<zone-name>`: Имя Primary Zone, настроенное здесь.
- `<external-endpoint>`: Внешний адрес, полученный из Primary кластера.

3 Получение `access-key` и `secret-key`

```
kubectl -n rook-ceph get secrets <realm-name>-keys -o yaml | grep access-key
kubectl -n rook-ceph get secrets <realm-name>-keys -o yaml | grep secret-key
```

Параметры:

- `<realm-name>`: Имя Realm, настроенное здесь.

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:

```
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: <new-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: CephObjectStore
metadata:
  name: <secondary-object-store-name>
  namespace: rook-ceph
spec:
  gateway:
    port: 7480
    instances: 2
  zone:
    name: <secondary-zone-name>
EOF
```

Параметры:

- `<access-key>` : AK, полученный здесь.
- `<secret-key>` : SK, полученный здесь.
- `<realm-endpoint>` : Внешний адрес, полученный из Primary кластера.
- `<realm-name>` : Realm.
- `<zone-group-name>` : ZoneGroup.
- `<secondary-zone-name>` : Имя Secondary Zone.
- `<secondary-object-store-name>` : Имя шлюза Secondary.

5 Настройка внешнего доступа для Secondary Zone

1. Получите UID шлюза Secondary {#uids}

```
kubectl -n rook-ceph get cephobjectstore <secondary-object-store-name> -o
jsonpath='{.metadata.uid}'
```

Параметры:

- `<secondary-object-store-name>`: Имя шлюза в Secondary кластере.

2. Создайте Service для внешнего доступа

```
cat << EOF | kubectl apply -f -
apiVersion: v1
kind: Service
metadata:
  name: rook-ceph-rgw-<object-store-name>-external
  namespace: rook-ceph
  labels:
    app: rook-ceph-rgw
    rook_cluster: rook-ceph
    rook_object_store: <object-store-name>
  ownerReferences:
    - apiVersion: ceph.rook.io/v1
      kind: CephObjectStore
      name: <object-store-name>
      uid: <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: <object-store-name>
  sessionAffinity: None
  type: NodePort
EOF
```

Параметры:

- `<secondary-object-store-name>`: Шлюз Secondary.
- `<secondary-object-store-uid>`: UID шлюза Secondary.

3. Добавьте внешние конечные точки в Secondary CephObjectZone

```
kubectl -n rook-ceph patch cephobjectzone <secondary-zone-name> --type merge -p
'{"spec":{"customEndpoints":["<external-endpoint>"]}}'
```

Параметры:

- `<secondary-zone-name>`: Имя Secondary Zone.
- `<secondary-zone-external-endpoint>`: Внешний адрес, полученный из Secondary кластера.

Переключение при сбое (Failover)

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

Процедуры

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

```
radosgw-admin zone modify --rgw-realm=<realm-name> --rgw-zonegroup=<zone-group-name> --
rgw-zone=<secondary-zone-name> --master
```

Параметры

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

Связанные операции

Получение внешнего адреса

1. Зайдите в **Administrator**.
2. В левой навигационной панели выберите **Storage Management** > **Distributed Storage**.
3. На вкладке **Cluster Information** прокрутите вниз до области **Storage Pool**, нажмите на ⋮ рядом с пулом объектного хранилища и выберите **View Address**.

Обновление параметров оптимизации

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

Содержание

Процедура

1. Сначала обновите параметры оптимизации хранения в Configmap с именем `rook-config-override-user`, заменив 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 с tools, отредактируйте ClusterServiceVersion (CSV) в пространстве имён 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 object store через определения пользовательских ресурсов (CRD).

Содержание

Предварительные условия

- Пул объектного хранилища был 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: -1
    maxSize: -1
    maxObjects: -1
  capabilities:
    user: "*"
    bucket: "*"
EOF
```

Параметры

Параметры	Описание
name	Имя создаваемого пользователя object store.
namespace	Пространство имён, в котором создаётся пользователь object store.
displayName	Отображаемое имя.
clusterNamespace	Пространство имён, где находятся родительские ресурсы `CephCluster` и `CephObjectStore`. Если не указано, пользователь должен находиться в том же пространстве имён, что и кластер и object store. Для включения этой функции в `CephObjectStore` параметр `allowUsersInNamespaces` должен включать пространство имён этого пользователя.
ObjectStore	Object store, в котором будет создан пользователь. Соответствует имени пула объектного хранилища.

Параметры	Описание
quotas	**необязательно** Здесь можно задать ограничения квот для пользователя. • maxBuckets: Максимальное количество бакетов для пользователя. Значение `-1` означает отсутствие ограничений. • 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 object store. Для изменения прав пользователя его необходимо удалить и создать заново. Подробнее см. в документации Ceph ↗. Поддерживаются права `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 пространств имён, либо указано "*" для разрешения всех пространств имён. Это полезно для приложений, которым необходимо создавать учётные данные object store 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
```

Menu

MinIO Object Storage

Введение

Введение

Установка

Установка

Предварительные требования

Процедура

Связанная информация

Архитектура

Архитектура

Основные компоненты:

Архитектура развертывания:

Масштабирование с несколькими пулами:

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 условия

Шаги выполнения

Связанные операции

Menu

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 проектирования 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 скорости чтения/записи могут достигать до 183 ГБ/с и 171 ГБ/с соответственно.
- **Масштабируемость**: Можно создавать несколько небольших и средних кластеров, легко управляемых, с поддержкой объединения нескольких кластеров в сверхбольшой пул ресурсов между дата-центрами, вместо прямого использования крупномасштабного централизованно управляемого распределенного кластера.
- **Облачно-нативность**: Соответствует всем нативным архитектурам и процессам построения облачных вычислений, включает новейшие технологии и концепции облачных вычислений, что делает объектное хранение 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

- 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, 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 — `stable`.
Installation Mode	`Cluster`: Все пространства имён в кластере используют один экземпляр Operator для создания и управления, что снижает использование ресурсов.
Installation Place	Выберите `Recommended`, Namespace поддерживается только **acp-storage**.
Upgrade Strategy	`Manual`: При появлении новой версии в Operator Hub требуется ручное подтверждение для обновления

Параметр	Рекомендуемая конфигурация
	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 базовую информацию.

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

2. В разделе **Resource Configuration** настройте характеристики 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 CPU по умолчанию — 2 ядра, запрос и лимит памяти — 4 Gi.
Medium scale	Предназначен для корпоративных приложений с хранением 1 000 000 объектов и поддержкой 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 потребностью в хранении 10 000 000 объектов и обработке 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 параметры настройки для профессиональных пользователей с особыми требованиями, обеспечивая точное соответствие масштаба сервиса и требований к производительности. Примечание: при настройке пользовательских характеристик убедитесь, что: • Запрос CPU больше 100 m. • Запрос памяти не менее 2 Gi. • Лимиты CPU и памяти не меньше соответствующих запросов.

3. В разделе **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.

Параметр	Описание
	Однако слишком большое количество экземпляров может привести к следующим проблемам: • Увеличение потребления ресурсов. • Если на одном узле размещено несколько экземпляров, сбой узла может привести к одновременному отключению нескольких экземпляров, снижая общую надёжность кластера. **Примечание**: • Минимальное количество экземпляров — 4. • Если количество экземпляров больше 16, введённое значение должно быть кратно 8. • При добавлении дополнительных пулов хранения количество экземпляров не должно быть меньше, чем в первом пуле хранения.
Single Storage Volume	Ёмкость одного тома PVC. Каждый сервис хранения управляет одним томом. После ввода ёмкости одного тома платформа автоматически рассчитает ёмкость пула хранения и другую информацию, которую можно просмотреть в **Storage Pool Overview**.
Underlying Storage	Базовое хранилище, используемое кластером MinIO. Выберите класс хранилища, созданный в текущем кластере. Рекомендуется TopoLVM.
Storage Nodes	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.
Storage Pool Overview	Для конкретных параметров и формул расчёта см. Storage Pool Overview.

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

Параметр	Описание
External Access	При включении поддерживается доступ к MinIO из других кластеров; при отключении — доступ только внутри кластера.
Protocol	Поддерживает HTTP и HTTPS; при выборе HTTPS необходимо ввести **Domain** и импортировать **Public Key** и **Private Key** сертификата домена. **Примечание**: • 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 домену 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 домену требуется вручную настроить сопоставление между полученным IP и введённым доменом при создании кластера.
Access Method	• **NodePort**: Открывает фиксированный порт на каждом хосте вычислительного узла для внешнего доступа к сервису. При настройке доступа по домену рекомендуется использовать VIP для разрешения доменного имени, чтобы обеспечить высокую доступность. • **LoadBalancer**: Использует балансировщик нагрузки для перенаправления трафика на бэкенд-сервисы. Перед использованием убедитесь, что в текущем кластере

Параметр	Описание
	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 развернутая информация о кластере будет очищена, 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 endpoint>]:<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-адрес, полученный на шаге 1, для `minio endpoint`.
- Введите **Access Key** и **Secret Key**, созданные при создании кластера, для `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 можно 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 загружается 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

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

Обзор пула хранения

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

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 надёжности данных и масштабируемости в Kubernetes.

Содержание

Основные компоненты:

- **MinIO Operator**: Управляет развертыванием и обновлением кластеров MinIO.
- **MinIO Peer**: Настраивает и управляет функцией репликации сайта 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 инстансов (pod) и количество томов (дисков) на каждый инстанс. Каждый сервер MinIO управляется через StatefulSet, что обеспечивает стабильные идентификаторы и постоянное хранилище. MinIO объединяет все диски в один или несколько erasure set и применяет стирающее кодирование для обеспечения отказоустойчивости.

Масштабирование с несколькими пулами:

Кластеры MinIO могут масштабироваться за счёт добавления дополнительных серверных пулов, каждый из которых имеет собственный erasure set. Хотя это увеличивает ёмкость хранения, такая архитектура усложняет обслуживание кластера и снижает общую надёжность. Сбой в любом серверном пуле может привести к недоступности всего кластера MinIO, 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 архитектуру пулов хранения, настраивать параметры стирающего кодирования и внедрять стратегии высокой доступности для гарантии целостности данных и стабильности работы в Kubernetes.

Menu

Основные понятия

Основные концепции

Menu

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 разделены на 12 фрагментов данных и 4 контрольных фрагмента, что позволяет системе восстанавливать данные даже при отказе 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 формируют 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.

Menu

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 различные 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

Добавление пула хранения вызовет кратковременное прерывание работы сервиса MinIO, но после этого он автоматически восстановится до нормального состояния.

Процедура

1. Перейдите в раздел **Administrator**.
2. В левой навигационной панели выберите **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 хранилища, созданный в текущем кластере, рекомендуется использовать TopoLVM.
Storage Nodes	Выберите узлы хранения, необходимые для кластера MinIO. Рекомендуется использовать от 4 до 16 узлов хранения; платформа развернёт по 1 сервису хранения для каждого выбранного узла. **Примечание**: при использовании 3 узлов хранения для обеспечения надёжности будет развернуто по 2 сервиса хранения на каждый узел.
Single Storage Volume	Вместимость одного тома хранения PVC. Каждый сервис хранения управляет 1 томом хранения, и после ввода вместимости одного тома платформа автоматически рассчитает ёмкость пула хранения и другую информацию, которую можно просмотреть в разделе **Storage Pool Overview**.

5. Нажмите **Confirm**.

Мониторинг и оповещения

Система объектного хранения 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 в режиме реального времени помогает оптимизировать производительность и принимать операционные решения, а автоматические оповещения обеспечивают стабильность и надежность системы хранения.

Содержание

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 хранилища

Этот раздел предоставляет общий обзор состояния системы хранения, статуса сервисов и использования сырой емкости. Если статус хранилища аномален, 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

Контролируйте шаблоны доступа, включая общее количество запросов и количество неудачных запросов. Эти данные помогают анализировать нагрузку на хранилище и обнаруживать аномалии, которые могут указывать на сбои сервисов или риски 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 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 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, что помогает 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 нарушению бизнес-процессов.
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 превентивные меры для предотвращения повторения?
3. Был ли зафиксирован всплеск оповещений в определённый период? Был ли он вызван операционной проблемой или внешним фактором? Следует ли скорректировать стратегию реагирования?

Постоянный анализ шаблонов оповещений и совершенствование стратегий мониторинга позволяет повысить устойчивость системы, минимизировать время

простоя и обеспечить бесперебойную работу хранилища.

Menu

Как сделать

Восстановление данных после аварий

Применимые сценарии

Терминология

Предварительные условия

Шаги выполнения

Связанные операции

Восстановление данных после аварий

MinIO поддерживает создание центра аварийного восстановления через удалённое резервное копирование данных или активное-активное развертывание, чтобы обеспечить сохранность и целостность исходных данных в случае аварии, тем самым гарантируя безопасность и надёжность данных.

Содержание

Применимые сценарии

- **Горячее резервное копирование**: Имеется два дата-центра в одном городе или в разных локациях — один основной и один резервный. Данные в реальном времени реплицируются с основного кластера на резервный, чтобы обеспечить согласованность данных. При аварии в основном кластере трафик бизнеса может быть бесшовно переключён на резервный кластер для обеспечения непрерывности работы.
- **Активное-активное на уровне города**: В архитектуре активное-активное на уровне города (мультикластер) имеется два дата-центра, расположенных в разных кластерах. Оба дата-центра активны и могут одновременно принимать бизнес-

трафик. При аварии в одном дата-центре бизнес может продолжать работу 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 в этот кластер для обработки.
- **Целевой кластер**: Кластер, который получает репликацию данных, миграцию или переключение. Обычно находится в резервном или ожидательном состоянии, ожидая получения данных от основного кластера или переключения бизнес-трафика на себя. 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 условия

- И основной, и целевой кластеры должны иметь включённый доступ к внешней сети. Для конкретных методов настройки обратитесь к разделу 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 оба HTTP или оба HTTPS.
- При использовании протокола HTTPS оба кластера должны настроить DNS-разрешение для себя и друг друга.
- При использовании HTTPS рекомендуется, чтобы оба кластера 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 выполнения

1. Войдите в **Administrator**.
2. В левой навигационной панели 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 http:// или https://.
Access Key	Access Key ID 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 запросов.

5. Нажмите **Add**.

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

Параметр	Описание
Cluster Status	Статус целевого кластера, включая **Healthy**, **Abnormal** или **Unknown**.

Параметр	Описание
Buckets	Количество бакетов, ожидающих синхронизации, и уже синхронизированных. • В сценариях горячего резервного копирования ожидающая синхронизация — это количество бакетов, которые основной кластер должен синхронизировать с целевым. • В сценариях активное-активное на уровне города ожидающая синхронизация — это общее количество бакетов, которые необходимо синхронизировать между основным и целевым кластерами.
Objects	Количество объектов, неудачно синхронизированных в бакете. **Примечание**: Это число приведено для справки, так как MinIO синхронизирует связанные конфигурационные файлы во время синхронизации.
Network Traffic Rate	Скорость входящего и исходящего сетевого трафика основного кластера. • В сценариях горячего резервного копирования скорость входящего трафика всегда равна 0. • В сценариях активное-активное на уровне города имеются данные по входящему и 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.

Menu

Локальное хранилище TopoLVM

Введение

Установка

Руководства

Мониторинг и оповещения

Мониторинг

Оповещения

Как сделать

Резервное копирование и восстановление PVC файловой системы TopoLVM с помощью Velero

Предварительные требования

Ограничения

Процедура

Введение

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

Основные особенности и преимущества:

- **Управление локальными томами**: TopoLVM 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.
- **Осведомлённость о топологии**: TopoLVM способен распознавать топологию кластера Kubernetes (например, узлы, зоны доступности), что позволяет автоматически выделять тома хранения на том же узле, где запланированы Pods, дополнительно оптимизируя производительность.
- **Динамическое выделение томов**: TopoLVM поддерживает динамическое создание, удаление и изменение размера томов хранения без ручного вмешательства, значительно упрощая операции и снижая сложность.
- **Глубокая интеграция с Kubernetes**: В качестве плагина CSI TopoLVM бесшовно интегрируется с API управления хранилищем Kubernetes, позволяя пользователям управлять локальными томами напрямую через стандартные ресурсы Kubernetes, такие как PersistentVolumeClaims.

В итоге, TopoLVM решает распространённые проблемы, связанные с использованием локального хранилища в Kubernetes, такие как ручное управление, отсутствие учёта 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

- Пакет lvm2 должен быть установлен на каждом узле кластера хранения. 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 — `stable`.
Installation Mode	`Cluster`: Все пространства имён в кластере используют один экземпляр Operator для создания и управления, что снижает использование ресурсов.
Installation Place	Выберите `Recommended`, Namespace поддерживается только **acp-storage**.
Upgrade Strategy	`Manual`: 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 устранения проблем. Затем нажмите **Clean Up** и повторно разверните Operator.

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

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

   - 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 хранилища. Должно быть уникальным в пределах текущего кластера.
Display Name	Имя, помогающее идентифицировать или фильтровать, например, описание класса хранилища на русском языке.
Device Class	Класс устройств — способ категоризации устройств хранения в TopoLVM. Каждый класс устройств соответствует набору устройств с одинаковыми характеристиками. При отсутствии особых требований можно использовать класс устройств **Auto-Allocated** из кластера.
File System	- **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 экстентами и поддерживающая обработку больших файлов. Максимальная ёмкость файловой системы — 1 EiB, максимальный размер файла — 16 TiB.
Recycling Policy	Политика переработки для постоянных томов. - Delete: При удалении persistent volume claim связанный persistent volume также удаляется. - Retain: Даже при удалении persistent volume claim связанный persistent volume сохраняется.

Параметр	Описание
Access Mode	ReadWriteOnce (RWO): Может быть смонтирован одним узлом в режиме чтения-записи.
Allocation Project	Этот тип persistent volume claim можно создавать только в определённых проектах. Если проект временно не назначен, его можно **Обновить** позже.

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

Menu

Руководства

Управление устройствами

Предварительные требования

Добавление устройств

Мониторинг и оповещения

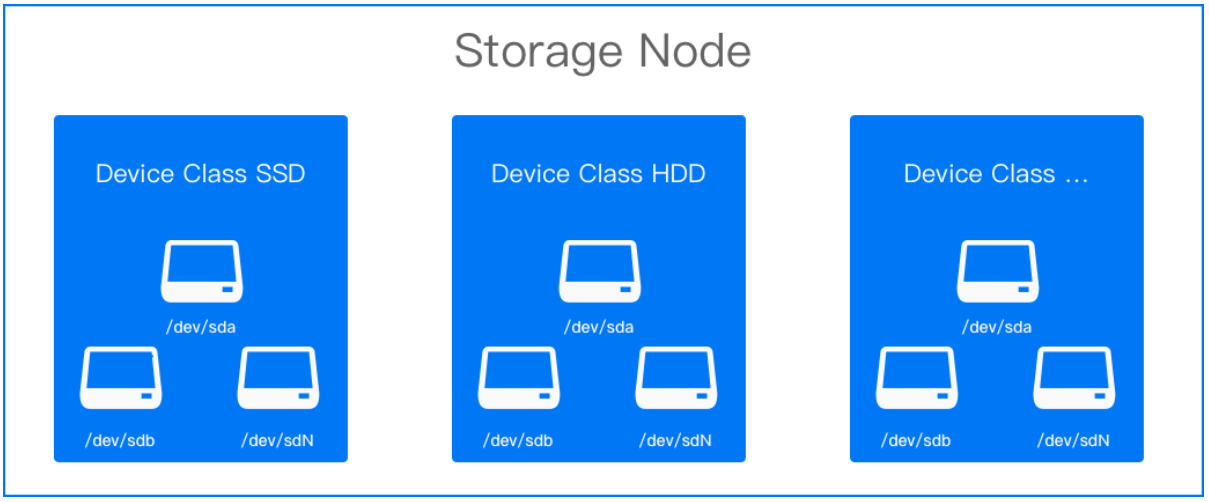
Мониторинг

Оповещения

Управление устройствами

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

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



Содержание

Предварительные требования

- При создании кластера локального хранилища должен быть добавлен как минимум 1 Класс устройств (deviceClasses.classes), включая устройства в этом классе.
- На узле должен присутствовать как минимум 1 голый диск.

Добавление устройств

1. Перейдите в раздел **Администратор**.
2. В левой навигационной панели нажмите **Управление хранилищем** > **Локальное хранилище**.
3. На вкладке **Детали** нажмите **Добавить узел хранения**.
4. Настройте соответствующие параметры согласно приведённым ниже инструкциям.

Параметр	Описание
Узел хранения	Узел, на котором имеется как минимум 1 голый диск.
Класс устройств	Каждый класс устройств соответствует группе устройств хранения с одинаковыми характеристиками; рекомендуется называть его в соответствии с типом дисков, например, *hdd*, *ssd*.
Устройство хранения	Например, */dev/sda*. 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 **Добавить**.

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

6. Перейдите на вкладку **Устройства хранения** и нажмите **Добавить устройство хранения**.

7. Добавьте устройства согласно подсказкам на интерфейсе.

8. Нажмите **Добавить**.

Мониторинг и оповещения

Локальное хранилище предоставляет готовые возможности по сбору метрик мониторинга и 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 могут использоваться для поддержки принятия решений при операционных проверках или настройке производительности, а комплексный механизм оповещений поможет обеспечить стабильную работу системы хранения.

Содержание

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 избыточным объявлением.

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

Оповещения

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

Настройка уведомлений

Для своевременного получения оповещений необходимо настроить политики уведомлений в центре операций. Уведомления могут отправляться по электронной почте, SMS или другими способами соответствующим сотрудникам, что обеспечивает

оперативное реагирование для устранения проблем или предотвращения сбоев. Пользователи могут получить доступ к настройкам политик уведомлений непосредственно 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 диагностики 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 типом устройства.
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 серьёзными 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 анализе после инцидента следует рассмотреть следующие вопросы:

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

Menu

Как сделать

Резервное копирование и восстановление PVC файловой системы TopoLVM с помощью Velero

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 применяется специально к PVC файловой системы TopoLVM.

Содержание

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 Ceph 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 резервного копирования

1. Убедитесь, что доступно S3-совместимое хранилище.
2. Перейдите в **Administrator** > **Cluster Management** > **Backup and Restore** > **Backup Repository**.
3. Создайте репозиторий резервного копирования, 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 файловой системы PVC требует pod, который монтирует PVC для импорта данных; без pod PVC останется в состоянии Pending. Для сложных приложений приостановите работу приложения и подключите 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 новое резервное копирование:

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

```yaml
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. Выполните операцию восстановления.

4. После восстановления проверьте:

- Совпадение имен PVC с исходным кластером.
- Целостность и доступность данных приложений в PVC.