Menu

Хранение

Распределённое хранилище Ceph

Введение

Обзор возможностей

Сравнение решений для хранения

Установка

Архитектура

Техническая архитектура

Основные понятия

Руководства

Как сделать

Разрешения

MinIO Object Storage

Введение

Установка

Предварительные требования

Развертывание оператора

Создание кластера

Создание бакета

Загрузка/скачивание файлов

Связанная информация

Архитектура

Основные компоненты:

Архитектура развертывания:

Масштабирование с несколькими пулами:

Заключение:

Основные понятия

Руководства

Как сделать

Локальное хранилище TopoLVM

Введение

Установка

Требования

Шаги

Руководства

Menu

Распределённое хранилище Ceph

Введение

Введение

Обзор возможностей

Сравнение решений для хранения

Установка

Создание кластера стандартного типа

Предварительные требования

Меры предосторожности

Процедура

Связанные операции

Создание Stretch Type Кластера

Терминология

Типовая схема развертывания

Ограничения и лимиты

Предварительные условия

Процедура

Связанные операции

Архитектура

Архитектура

Техническая архитектура

Основные понятия

Основные концепции

Rook Operator

Ceph CSI

Функции модулей Ceph

Руководства

Доступ к сервисам хранения

Предварительные требования

Процедура

Последующие действия

Управление Storage Pools

Создание Storage Pool

Удаление Storage Pool

Просмотр адресов Object Storage Pool

Развертывание 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

Разрешения

Разрешения

Введение

Alauda Container Platform (ACP) Storage с Ceph — это гиперконвергентное решение для хранения данных, предоставляемое платформой внутри кластера. Основанное на open-source решении Rook + Ceph, распределённое хранилище обеспечивает автоматическое управление, автоматическое масштабирование и автоматическое восстановление, удовлетворяя потребности малых и средних приложений в блочном, файловом и объектном хранении.

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

Содержание

Обзор возможностей

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

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

Сравнение решений для хранения

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

Создание кластера хранения

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

Доступ к внешнему хранилищу

Вариант 1: Использовать распределённые ресурсы хранения других бизнес-кластеров внутри платформы для обеспечения изоляции хранения и бизнеса, что упрощает управление и обслуживание.

Вариант 2: Интегрировать внешние Ceph-ресурсы хранения как распределённое хранилище.

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, к которому осуществляется доступ, является распределённым хранилищем с других платформ, например, основной/резервной платформой 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 конфигурации внешнего хранилища, операции должны выполняться в интерфейсе управления кластера хранения.

Menu

Установка

Создание кластера стандартного типа

Предварительные требования

Меры предосторожности

Процедура

Связанные операции

Создание Stretch Type Кластера

Терминология

Типовая схема развертывания

Ограничения и лимиты

Предварительные условия

Процедура

Связанные операции

Создание кластера стандартного типа

Кластер стандартного типа — это наиболее типичный способ развертывания хранилища Ceph. Он распределяет реплики данных по жестким дискам на разных хостах, обеспечивая, что в случае отказа одного хоста копии данных на других хостах смогут сохранить доступность сервиса.

Содержание

Предварительные требования

- В кластере хранения должно быть не менее 3 узлов.
- Каждый узел должен иметь как минимум 1 пустой жесткий диск или 1 неотформатированный раздел жесткого диска.
- Рекомендуемый объем доступного жесткого диска должен быть больше 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`.

Меры предосторожности

Создание сервисов хранения и **Доступ к сервисам хранения** поддерживают выбор только одного метода.

Процедура

1 Развертывание Operator

1. Перейдите в раздел **Platform Management**.
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.
   - Если страница автоматически перейдет к следующему шагу, это означает успешное развертывание Operator.
   - Если развертывание 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

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 device class в соответствии с вашими потребностями хранения, распределяя разный контент по дискам с разной производительностью. • **Default Device Class**: Платформа автоматически классифицирует типы жестких дисков на узлах кластера. Например, создаются device class с именами `hdd`, `ssd`, `nvme`. • **Custom Device Class**: Настройте имя device class для конкретных комбинаций дисков на узле; поддерживается добавление нескольких device class. Один и тот же жесткий диск может принадлежать только одному device class.
Device Class - Name	Имя device class. При выборе **Custom Device Class** имя device class не может быть `hdd`, `ssd`, `nvme`.
Device Class - Storage Devices	Выберите **Blank Hard Disk** или **Unformatted Hard Disk Partition** на узлах. • Если переключатель "Open All Blank Devices" включен: все пустые устройства на узле будут добавлены в device class; • Если переключатель "Open All Blank Devices" выключен: вручную введите имена пустых устройств

Параметр	Описание
	на узле, например `sda`.
Snapshot	При включении поддерживается создание снимков PVC и использование снимков для конфигурации новых PVC для быстрого резервного копирования и восстановления 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 предоставляются готовые возможности сбора метрик мониторинга и оповещений, см. Monitoring and Alarming. **Примечание**: Если не включить сейчас, потребуется искать альтернативные решения для мониторинга и оповещений хранилища, например, вручную настраивать дашборды 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, указать подсети `public` и `cluster`. Если оставить пустым, будет использована подсеть хоста по умолчанию. **Примечание**: Использование сети хоста может представлять угрозу безопасности из-за передачи

Параметр	Описание
	данных в незашифрованном (открытом) виде через порты хоста. Обратитесь в службу поддержки платформы для получения решения по шифрованию передачи. • **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. Сбои 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, доступные для настройки фиксированного развертывания: MON, MGR, MDS, RGW.

- Если страница автоматически перейдет к следующему шагу, это означает успешное развертывание кластера Ceph.
- Если создание не удалось, можно нажать очистку **Created Information or Retry** для автоматической очистки ресурсов и повторного создания кластера,

либо вручную очистить ресурсы согласно документации Distributed Storage Service Resource Cleanup.

3 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	- File Storage: обеспечивает безопасные, надежные и масштабируемые услуги совместного файлового хранения. Подходит для совместного использования файлов, резервного копирования данных и т.д. - Block Storage: предоставляет хранилище с высокой производительностью IOPS и низкой задержкой. Подходит для баз данных, виртуализации и т.д. - Object Storage: предоставляет хранилище с интерфейсом 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 device class, добавленных на предыдущем шаге. • При выборе device class данные будут храниться в выбранном device class. • Если device class 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.

- Если страница автоматически перейдет к следующему шагу, это 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**. На вкладке **Details** можно просмотреть информацию о созданном пуле хранения.

Связанные операции

Создание кластера расширенного типа

Подробности см. в разделе Create Extended 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 данных. В случае катастрофы, когда одна из зон доступности в двух зонах полностью недоступна, Ceph всё равно может сохранять доступность.

Содержание

Терминология

Термин	Объяснение
Quorum Availability Zone	Обычно располагается в отдельной зоне, которая не несёт основных рабочих нагрузок, 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 хранятся и обрабатываются данные, несущая операционные нагрузки и задачи хранения данных, вместе с зоной кворума формирующая полноценную систему хранения с высокой доступностью.

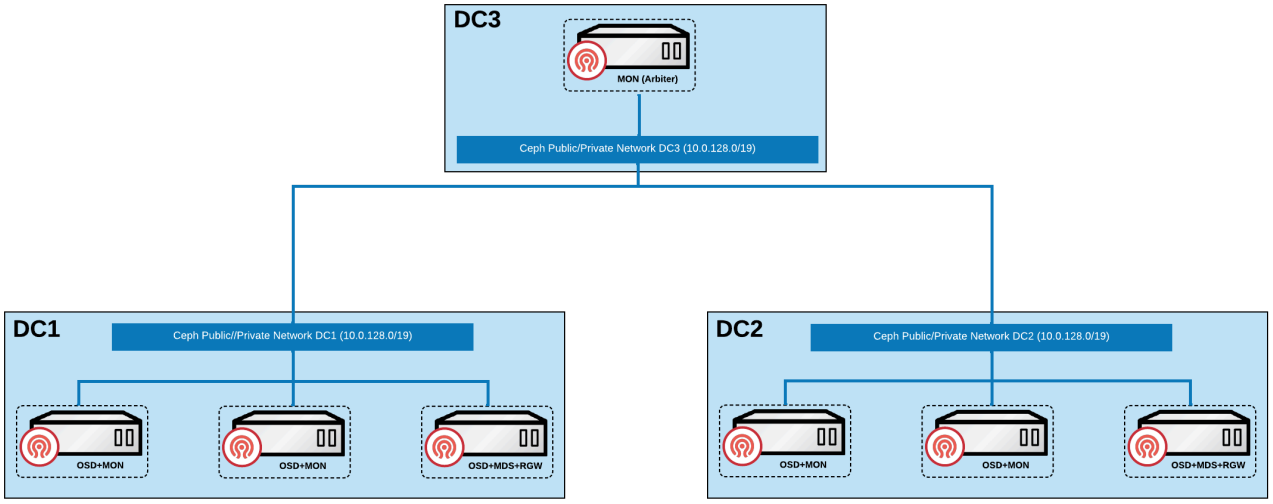
Типовая схема развертывания

Ниже приведено типовое решение по развертыванию stretch кластера, а также описание компонентов и принципы аварийного восстановления.

Описание компонентов

Узлы необходимо распределить по трём зонам доступности, включая две зоны доступности данных и одну зону кворума.

- В обеих зонах доступности данных необходимо полностью развернуть все основные компоненты Ceph (MON, OSD, MGR, MDS, RGW), при этом в каждой зоне доступности данных должно быть настроено по два экземпляра MON для обеспечения высокой доступности. Если оба экземпляра MON в одной зоне доступности данных становятся недоступны, система определяет эту зону как находящуюся в состоянии сбоя.
- В зоне кворума требуется развернуть только один экземпляр MON, который служит узлом для принятия арбитражных решений.



Объяснение аварийного восстановления

- При полном отказе зоны доступности данных Ceph кластер автоматически переходит в деградированное состояние и генерирует уведомление об аварии. Система изменит минимальное количество реплик в пуле хранения (min_size) с значения по умолчанию 2 на 1. Поскольку другая зона доступности данных сохраняет двойное дублирование, кластер остаётся доступным. После восстановления отказавшей зоны система автоматически выполнит синхронизацию данных и вернётся в здоровое состояние; если сбой не удаётся устранить, рекомендуется заменить её на новую зону доступности данных.
- При разрыве сетевого соединения между двумя зонами доступности данных, но при сохранении нормального подключения к зоне кворума, зона кворума на основе заранее заданных политик проведёт арбитраж между двумя зонами данных, выбрав ту, которая находится в лучшем состоянии, для продолжения предоставления сервисов в качестве основной зоны данных.

Ограничения и лимиты

- **Ограничения по пулам хранения**: Пулы хранения с erasure coding не поддерживаются, допускается только механизм репликации для защиты данных.
- **Ограничения по классификации устройств**: Функциональность device class не поддерживается, хранение не может быть стратифицировано по характеристикам устройств.

- **Ограничения по региональному развертыванию**: Поддерживаются только две зоны доступности данных; не допускается более двух зон доступности данных.
- **Требования к балансировке данных**: Вес OSD в двух зонах доступности данных должен строго совпадать для обеспечения сбалансированного распределения данных.
- **Требования к носителям хранения**: Разрешена только конфигурация All-Flash OSD, что минимизирует время восстановления после восстановления соединения и максимально снижает риск потери данных.
- **Требования к сетевой задержке**: RTT (время кругового обхода) между двумя зонами доступности данных не должен превышать 10 мс, а зона кворума должна соответствовать требованиям по задержкам спецификации ETCD для обеспечения надёжности арбитражного механизма.

Предварительные условия

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

- Обеспечить распределение не менее 5 узлов между одной зоной кворума и двумя зонами доступности данных. При этом в зоне кворума должен быть минимум один узел, который может быть виртуальной машиной или облачным хостом.
- Обеспечить наличие как минимум одного Master узла (контрольного узла) в одной из трёх зон доступности.
- Обеспечить равномерное распределение не менее 4 вычислительных узлов по 2 зонам доступности данных, при этом в каждой зоне доступности данных должно быть минимум по 2 вычислительных узла.
- По возможности обеспечить одинаковое количество узлов и конфигурацию дисков в двух зонах доступности данных.

Процедура

1 Тегирование узлов

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Cluster Management** > **Cluster**.
3. Кликните по названию соответствующего кластера, чтобы перейти на страницу обзора кластера.
4. Переключитесь на вкладку **Nodes**.
5. В соответствии с планированием в разделе Prerequisites добавьте метку `topology.kubernetes.io/zone=<zone>` этим узлам для классификации их в указанную зону доступности. Здесь вместо <zone> укажите имя зоны доступности.

2 Создание сервисов хранения

В данном документе описаны только параметры, отличающиеся от стандартных кластеров; по остальным параметрам обращайтесь к Create Standard Type Cluster.

Создание кластера

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

Создание пула хранения

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

Связанные операции

Создание стандартного типа кластера

Подробности см. в Create Standard Type Cluster.

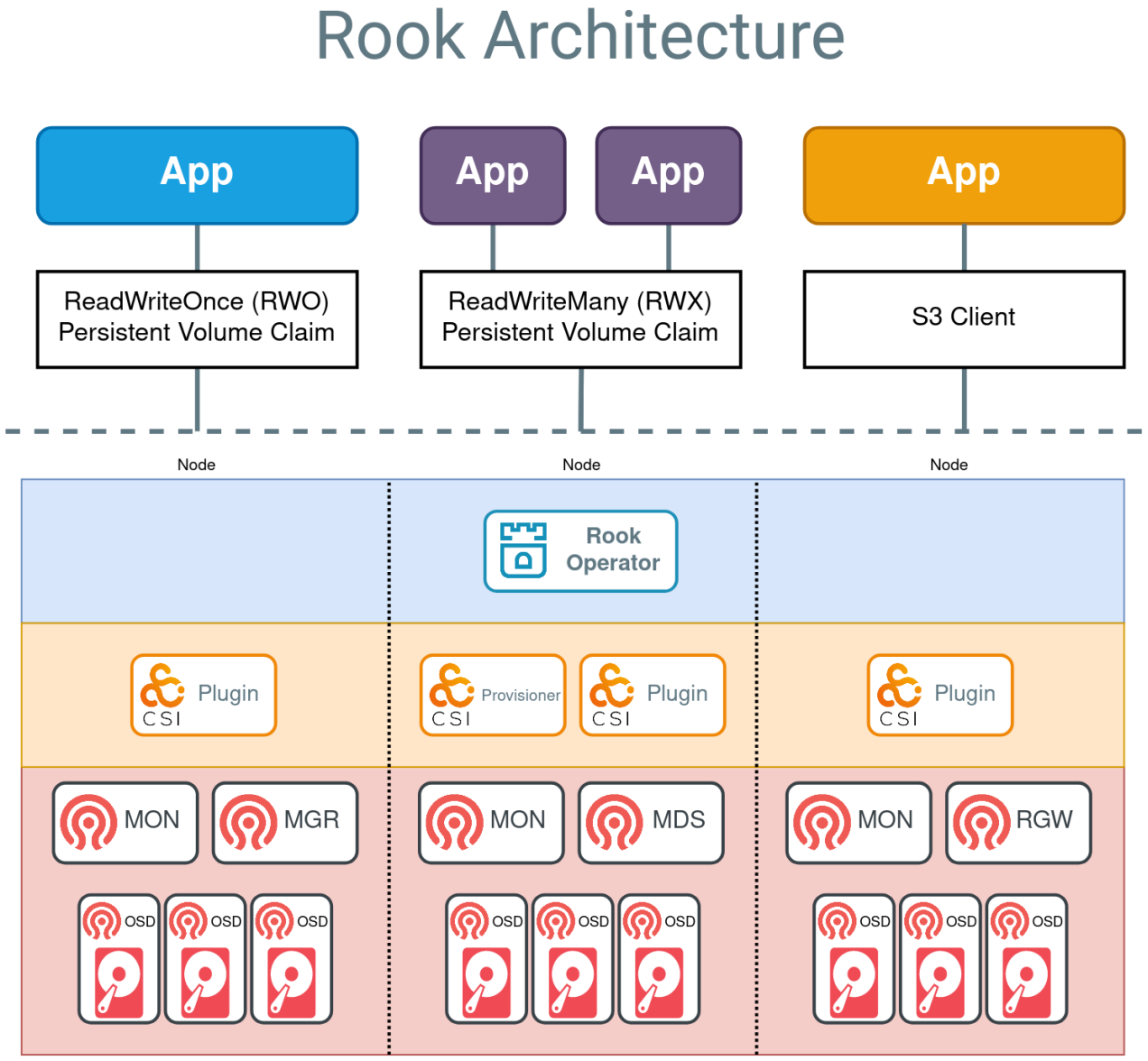
Очистка распределённого хранилища

Подробности см. 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 способа интеграции: во-первых, интеграция распределённых ресурсов хранения из других бизнес-кластеров внутри платформы для обеспечения изоляции хранения и бизнеса, что облегчает управление и сопровождение; во-вторых, подключение внешних ресурсов хранения Ceph для использования распределённого хранения.

Содержание

Предварительные требования

Подготовка хранилища

Выберите один из вариантов:

- В других бизнес-кластерах развернуто распределённое хранилище, создан пул хранения. Запишите имя пула хранения для последующего использования при интеграции.

- Вне платформы создано внешнее хранилище Ceph (версия ≥ 14.2.3) с пулом хранения. Запишите имя пула хранения для последующего использования при 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 компоненте MON	`ceph mon dump` Должна быть в формате {name= IP}, например *a=192.168.100.100:6789* .
Административный ключ	`ceph auth get-key client.admin`
Пул хранения	• Файловое хранилище: используйте команду `ceph fs ls` для получения значения `name` . • Блочное хранилище: `ceph osd dump` | `grep "application rbd"` | `awk '{print $3}'`
Пул хранения данных	(требуется 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 хранилищу Ceph**, операции для доступа к распределённому хранилищу аналогичны.

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 снимки не были включены при доступе к хранилищу, 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 подсеть в управлении сетью и назначить её пространству имён `rook-ceph`. Если оставить пустым, будет использована подсеть по умолчанию.

Параметр	Описание
Other Parameters	Заполните параметры аутентификации для внешнего Ceph, полученные на этапе предварительных требований.

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

Параметр	Описание
Type	В зависимости от типа созданного пула хранения, по умолчанию будет соответствующий класс хранения: • Файловое хранилище: CephFS File Storage • Блочное хранилище: CephRBD Block Storage
Reclaim Policy	Политика возврата для постоянных томов. • Delete: при удалении запроса на постоянный том, связанный постоянный том также будет удалён. • Retain: даже при удалении запроса на постоянный том, связанный постоянный том останется.
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 их позже.

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

Последующие действия

- Создание классов хранения: CephFS File Storage, CephRBD Block Storage

- Разработчики, использующие указанные классы хранения для создания запросов на постоянные тома, могут расширять функциональность с помощью снимков томов и масштабирования.

Примечание: Если необходимо выполнять обслуживание пулов хранения, конфигураций устройств хранения и т.п. для внешнего хранилища, операции должны выполняться в управляющей платформе кластера хранения.

Управление Storage Pools

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

Содержание

Создание Storage Pool

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

Совет: В рамках одного кластера хранения допускается наличие только одного файлового и одного объектного storage pool, при этом можно создать до восьми блочных storage pools.

Порядок действий

> 1. Перейдите в **Platform Management**.

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 совместного файлового хранения. Подходит для совместного использования файлов, резервного копирования данных и т.д. - Block Storage: Обеспечивает хранилище с высокой производительностью IOPS и низкой задержкой. Подходит для баз данных, виртуализации и т.д. - Object Storage: Предоставляет стандартный интерфейс S3, подходит для big data, резервного архивирования, облачных сервисов хранения и др.
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 device class в рамках созданного storage pool.
	  • При выборе device class данные будут храниться в выбранном классе устройств.
	  • Если device class не выбран, данные будут случайным образом распределены по всем устройствам в storage

Параметр	Описание
	pool. • **Для кластера типа Extended**: Добавление device class 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 Service 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 **Platform Management**.
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 предупреждением 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 **Platform Management**.
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 место 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 к сервисам хранилища. Рекомендуется выполнять обновление в непиковое время.
- При типе кластера **Extend** функция фиксированного развертывания компонентов не поддерживается.

Порядок действий

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

Перезапуск компонентов хранилища

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

Порядок действий

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Storage Management** > **Distributed Storage**.
3. Во вкладке **Storage Components** нажмите ⋮ рядом с именем компонента > **Delete**.

Добавление устройств/классов устройств

Содержание

Добавление классов устройств

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

Примечания

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

Процедура

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Storage Management** > **Distributed Storage**.

3. Нажмите на вкладку **Device Classes**.
4. Нажмите **Add Device Class** и настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Name	Имя класса устройства. Следующие имена нельзя использовать для класса устройства: `hdd`, `ssd`, `nvme`.
Storage Devices	Выберите **Blank Disks** или **Unformatted Disk Partitions** в узле. • Если переключатель для всех пустых устройств включен: добавляются все пустые устройства узла в этот класс устройств; • Если переключатель для всех пустых устройств выключен: вручную введите имена пустых устройств узла, например, `sda`.

Добавление устройств

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

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

Процедура

1. Перейдите в **Platform Management**.
2. В левой навигационной панели нажмите **Storage Management** > **Distributed Storage**.
3. Нажмите на вкладку **Device Classes**.

4. Справа от класса устройства 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 в качестве устройства хранения. **Compute Node**: Узел, в котором 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**: Выбрать для добавления всех неразмеченных монтированных дисков узла в качестве устройств хранения. **Specified Disks**: Выбрать для добавления некоторых дисков узла в качестве устройств хранения, включая пустые диски или уже размеченные монтированные 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 нужно добавить в качестве устройств хранения, например, `sda`, `sdb`. После ввода каждого имени жесткого диска нажмите Enter для подтверждения. **Примечание**: Рекомендуется использовать весь жесткий диск в качестве устройства хранения, а не разделы на жестком 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

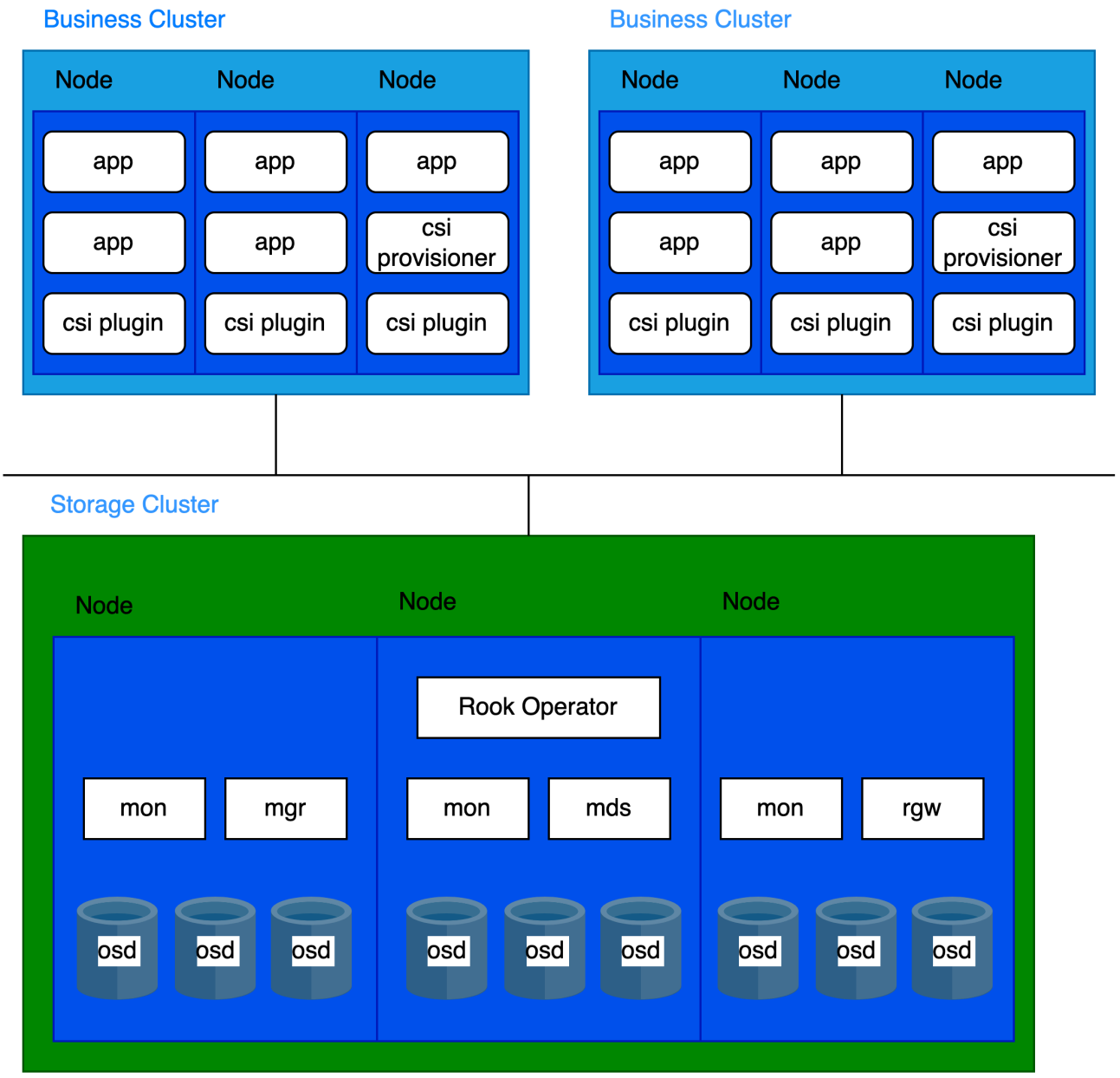
Настройка выделенного кластера для распределённого хранилища

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 следующие:

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 с ёмкостью 10TiB или меньше на устройство и обеспечить одинаковый размер и тип всех дисков.

Планирование ёмкости

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

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

Размер устройства хранения(D)	**Устройств на узел(M)**	**Общая ёмкость(D*M*N)**	**Доступная 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 расширение

1. **Проактивное планирование ёмкости**

   Всегда следите, чтобы доступная ёмкость хранения 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 расширьте кластер.
   - **95% использования** («заполнено»): хранилище полностью исчерпано, стандартные команды не могут освободить место. Немедленно обратитесь в службу поддержки платформы.

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

3. **Рекомендации по масштабированию**

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

Требования к сети

Распределённое хранилище должно использовать **HostNetwork**.

Сетевая изоляция

Сеть делится на два типа:

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

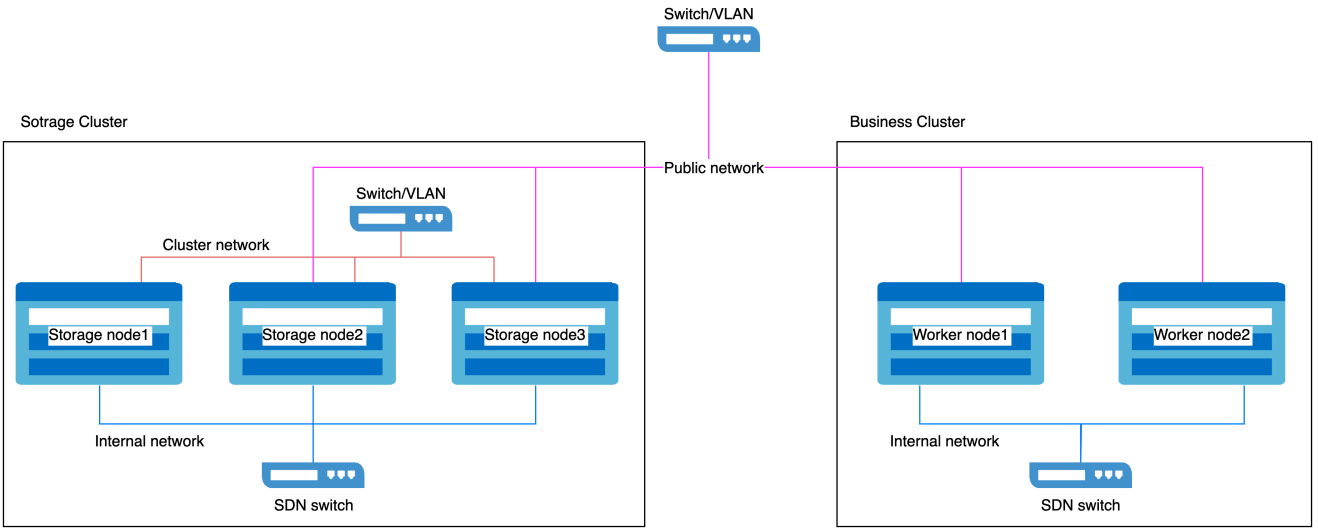
Для обеспечения качества сервиса и стабильности производительности:

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 **Управление платформой**.
2. В левой боковой панели нажмите **Управление хранилищем** > **Распределённое хранилище**.
3. Нажмите **Создать сейчас**.
4. На странице мастера **Развёртывание Operator** нажмите кнопку **Deploy Operator** в правом нижнем углу.
   - Если страница автоматически перейдёт к следующему шагу, это означает успешное развёртывание Operator.

- Если развёртывание не удалось, следуйте подсказке на интерфейсе **Очистить информацию о развёртывании и повторить попытку**, и повторно разверните Operator; если хотите вернуться на страницу выбора распределённого хранилища, нажмите **Application Store**, сначала удалите ресурсы уже развернутого **rook-operator**, затем удалите сам **rook-operator**.

2 Создание кластера ceph

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

▶ Нажмите, чтобы посмотреть

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

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

3 Создание пулов хранения

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

Создание файлового пула

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

▶ Нажмите, чтобы посмотреть

Создание блочного пула

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

▶ Нажмите, чтобы посмотреть

Создание объектного пула

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

▶ Нажмите, чтобы посмотреть

Последующие действия

Когда другие кластеры нуждаются в использовании сервиса распределённого хранения, следуйте следующим рекомендациям.

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 **Platform Management**.
2. В левой навигационной панели выберите **Storage Management** > **Storage Classes**.
3. Нажмите ⋮ > **Delete** и удалите все StorageClasses, использующие решения хранилища Ceph.

3 Удаление Storage Pools

Этот шаг выполняется после завершения предыдущего.

1. Перейдите в **Platform Management**.
2. В левой навигационной панели выберите **Storage Management** > **Distributed Storage**.
3. В разделе **Storage Pool Area** нажмите ⋮ > **Delete** и удалите все пулы хранилища по одному. Когда в области пула хранилища отображается сообщение **No Storage Pools**, это означает успешное удаление.

4. (Опционально) Если режим кластера — **Extended**, выполните следующую команду на Master-узле кластера для удаления созданных встроенных пулов хранилища.

```
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 для очистки, заменив `<configmap>` на фактическое имя.

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

4. Проверьте, что все Secrets удалены.

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

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

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

5. Проверьте, что очистка rook-ceph завершена.

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

Отсутствие вывода означает, что очистка завершена.

6 Выполнение скрипта очистки

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

Скрипт очистки

Содержимое скрипта clean-rook.sh:

▶ Нажмите для просмотра

Меры предосторожности

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

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

Процедура

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

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

Пример: `sh clean-rook.sh /dev/vdb`

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

2. Используйте команду `lsblk -f` для проверки информации о разделах. Если в столбце `FSTYPE` вывод отсутствует, 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 возможность быстрого переключения 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 создания снимка.
- CephFS Mirror обеспечивает только резервное копирование данных базового хранилища и не может выполнять резервное копирование ресурсов Kubernetes. Для резервного копирования PVC и PV ресурсов используйте функцию платформы **Backup and Restore** совместно.

Содержание

Терминология

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

Настройка резервного копирования

Предварительные требования

- Подготовьте два кластера, подходящих для развертывания Alauda Container Platform (ACP) Storage с Ceph: Primary cluster и Secondary cluster, обеспечив сетевое взаимодействие между ними.
- 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.

Порядок действий

1 Включение зеркалирования для пула файлового хранилища во Secondary cluster

Выполните следующие команды на управляющем узле Secondary cluster:

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 cluster:

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 cluster

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

Выполните следующие команды на управляющем узле Primary cluster:

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 Включение зеркалирования для пула файлового хранилища в Primary cluster

Выполните следующие команды на управляющем узле Primary cluster:

Command

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

Sample

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

Output

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

Параметры:

- `<fs-name>`: имя пула файлового хранилища.
- `<schedule-interval>`: интервал выполнения снимков. Подробнее см. в официальной документации↗.
- `<retention-policy>`: политика хранения снимков. Подробнее см. в официальной документации↗.

5 Развёртывание Mirror Daemon в Primary cluster

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

Выполните следующие команды на управляющем узле Primary cluster:

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 cluster вы можете напрямую продолжить использование CephFS в Secondary cluster.

Предварительные требования

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

Восстановление после сбоев блочного хранилища

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

WARNING

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

Содержание

Терминология

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

Настройка резервного копирования

Предварительные условия

- Подготовьте два кластера, способных развернуть Alauda Container Platform (ACP) Storage с 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 обоих 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 является 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 нужно 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 primary-backup для RBD-образа.

Предварительные условия

- Ресурсы Kubernetes Primary кластера были забэкаплены и восстановлены в Secondary кластер, 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 primary 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: Высший 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 зон (Zones). ZoneGroups обеспечивают синхронизацию и репликацию данных между зонами, обычно в пределах одного географического региона. • Zone: Логическая группировка внутри ZoneGroup, которая физически хранит данные. Каждая зона 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 конфигурации, как описано ниже.

Процедуры

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

1 Создание объектного хранилища в Primary кластере

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

Выполните следующие команды на управляющем узле Primary кластера:

Command

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

---
apiVersion: ceph.rook.io/v1
kind: CephObjectZoneGroup
metadata:
  name: <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
```

Output

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

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 объекта 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 описано, как создать Secondary Zone и настроить синхронизацию, подтягивая информацию 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 **Platform Management**.
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.

Menu

Разрешения

Функция	Действие	Platform Administrator	Platform auditors	Project Manager	Namesp Administ
builtinstorage `acp-builtinstorage`	Просмотр	✓	✓	✓	✓
	Создать	✓	×	×	×
	Обновить	✓	×	×	×
	Удалить	✓	×	×	×

Menu

MinIO Object Storage

Введение

Установка

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 аварии

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 по лицензии 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. Объектный файл может иметь любой размер — 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 использовать TopoLVM.

Развертывание оператора

1. В левой навигационной панели нажмите **Storage** > **Object Storage**.
2. Нажмите **Configure Now**.
3. На странице мастера **Deploy MinIO Operator** нажмите в правом нижнем углу **Deploy Operator**.
   - Если страница автоматически перейдет к следующему шагу, это означает успешное развертывание оператора.
   - Если развертывание не удалось, следуйте подсказкам интерфейса для **Clean Up Deployed Information and Retry** и повторно разверните оператора.

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 совместно 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 подключений в тестовых средах или

Параметр	Описание
	сценариях резервного копирования. Запрос и лимит ресурсов CPU по умолчанию установлены на 2 ядра, запрос и лимит памяти — 4 Gi.
Medium scale	Предназначен для корпоративных приложений, требующих хранения 1 000 000 объектов и способных обрабатывать до 200 одновременных запросов. Запрос и лимит CPU по умолчанию — 4 ядра, запрос и лимит памяти — 8 Gi.
Large scale	Предназначен для групп пользователей с потребностями хранения 10 000 000 объектов и обработки до 500 одновременных запросов, подходит для сценариев с высокой нагрузкой. Запрос и лимит CPU по умолчанию — 8 ядер, запрос и лимит памяти — 16 Gi.
Custom	Предлагает гибкие параметры настройки для профессиональных пользователей с особыми требованиями, обеспечивая точное соответствие масштаба сервиса и требований к производительности. Примечание: при настройке пользовательских характеристик необходимо убедиться, что:

- Запрос CPU больше 100 м.
- Запрос памяти не менее 2 Gi.
- Лимиты CPU и памяти не меньше 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 надежность системы, обеспечивая высокую доступность данных. Однако слишком большое количество экземпляров может привести к следующим проблемам:

- Повышенное потребление ресурсов.

- Если на одном узле размещено несколько экземпляров, сбой узла может привести к одновременному отключению нескольких экземпляров, что снижает общую надежность кластера. **Примечание**:
- Минимальное количество экземпляров — 4.
- Если количество экземпляров больше 16, введённое значение должно быть кратно 8.
- При добавлении дополнительных пулов хранения количество экземпляров должно быть не меньше, чем в первом пуле хранения. | | **Single Storage Volume** | Вместимость одного тома PVC хранения. Каждый сервис хранения управляет одним томом. После ввода вместимости одного тома платформа автоматически рассчитает емкость пула хранения и другую информацию, которую можно просмотреть в разделе **Storage Pool Overview**. | | **Underlying Storage** | Базовое хранилище, используемое кластером MinIO. Пожалуйста, 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 развернет по одному сервису хранения на каждом выбранном узле. | | **Storage Pool Overview** | Для конкретных параметров и формул расчета обратитесь к 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 возможен только внутри кластера.
Protocol	Поддерживает HTTP и HTTPS; при выборе HTTPS необходимо ввести **Domain** и импортировать **Public Key** и **Private Key** сертификата доменного имени.
Note:	

- При протоколе доступа HTTP поды внутри кластера могут обращаться к MinIO напрямую по полученному IP или доменному имени без настройки сопоставления IP и домена; узлы внутри кластера могут обращаться к MinIO напрямую по IP, а для

доступа по доменному имени требуется ручная настройка сопоставления IP и домена; внешний доступ возможен напрямую по IP.

- При протоколе HTTPS доступ к MinIO по IP невозможен как внутри, так и вне кластера. Для нормального доступа по доменному имени 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 доменному имени рекомендуется использовать VIP для разрешения домена, чтобы обеспечить высокую доступность.
- **LoadBalancer**: Использует балансировщик нагрузки для перенаправления трафика на бэкенд-сервисы. Перед использованием убедитесь, что в текущем кластере развернут плагин MetalLB и в пуле внешних адресов есть доступные IP. |

5. Нажмите в правом нижнем углу **Create Cluster**.

- Если страница автоматически перейдет к **Cluster Details**, это означает успешное создание кластера.
- Если кластер остается в процессе создания, можно нажать **Cancel**. После отмены развернутые данные кластера будут очищены, и вы сможете вернуться на страницу создания кластера для повторного создания.

Создание бакета

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

Порядок действий

1. На странице сведений о кластере перейдите на вкладку **Access Method**, чтобы просмотреть 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 настройки:

- 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 бакет.

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

Загрузка/скачивание файлов

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

Порядок действий

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

```
touch <file name>
```

2. Загрузите файлы в бакет.

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

3. Просмотрите файлы в бакете, 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 необходимо рассчитывать на основе количества экземпляров в первом пуле хранения.

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

Обзор пула хранения

Параметр обзора пула хранения	Формула расчёта
Доступная ёмкость	Если Instance Number ≤ 16, Доступная ёмкость = Вместимость одного тома × (Количество экземпляров - Коэффициент избыточности).
Если количество экземпляров > 16, Доступная ёмкость = Вместимость одного тома × (Количество экземпляров - 4 × (Количество экземпляров + 15) / 16)). Результат выражения	

Параметр обзора пула хранения	Формула расчёта
"4 × (Количество экземпляров + 15) / 16" округляется вниз.	
Общая ёмкость	Общая ёмкость = Количество экземпляров × Вместимость одного тома
Количество отказоустойчивых сервисов хранения	Если Количество экземпляров > 2 × Коэффициент избыточности, Количество отказоустойчивых сервисов = Коэффициент избыточности.
Если Количество экземпляров = 2 × Коэффициент избыточности, количество отказоустойчивых сервисов = Коэффициент избыточности - 1	

Архитектура

Alauda Container Platform (ACP) Object Storage с MinIO — это высокопроизводительная распределённая система объектного хранения, разработанная для облачных 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 сайтов MinIO.
- **MinIO Pool**: Основной компонент MinIO, отвечающий за обработку запросов объектного хранения. Каждый пул соответствует StatefulSet и предоставляет ресурсы хранения.

Архитектура развертывания:

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

Масштабирование с несколькими пулами:

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

Заключение:

MinIO — это высокомасштабируемое облачное объектное хранилище, которое балансирует производительность и надёжность. При проектировании кластера MinIO важно тщательно продумывать пулы хранения, настраивать параметры стирающего кодирования и реализовывать стратегии высокой доступности для обеспечения целостности данных и стабильности работы в Kubernetes.

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 в нормальное состояние.

Процедура

1. Перейдите в **Platform Management**.
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 хранения, 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 мониторинга в реальном времени помогает оптимизировать производительность и принимать операционные решения, а автоматические оповещения обеспечивают стабильность и надежность вашей 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 состояние хранилища ненормальное, детали оповещения укажут на первопричину, что поможет эффективно диагностировать и устранять проблемы.

Мониторинг кластера

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

Мониторинг объектов

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

Оповещения

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

Настройка уведомлений

Для обеспечения своевременного реагирования настройте политики уведомлений в **Operations Center**. Оповещения могут отправляться по электронной почте, SMS или другим каналам, чтобы уведомлять ответственных сотрудников. Тонко настройте параметры в соответствии с рабочим процессом реагирования вашей организации.

Обработка оповещений

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

Платформа классифицирует оповещения по уровням серьезности, 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 определённый временной интервал с резким увеличением оповещений? Был ли он вызван операционной проблемой или внешним фактором? Следует ли скорректировать стратегию реагирования?

Постоянно анализируя паттерны оповещений и совершенствуя стратегии мониторинга, команды могут повысить устойчивость системы, минимизировать время простоя и

обеспечить бесперебойную работу хранилища.

Menu

Как сделать

Восстановление данных после аварии

Применимые сценарии

Терминология

Предварительные требования

Шаги операции

Связанные операции

Восстановление данных после аварии

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**: Два дата-центра находятся в одном городе или в разных местах, один из них основной, другой — резервный. Данные в реальном времени реплицируются с основного кластера на резервный, чтобы обеспечить согласованность данных. При возникновении аварии в основном кластере трафик бизнеса может быть бесшовно переключён на резервный кластер для обеспечения непрерывности работы.
- **Городской уровень Active-Active**: В архитектуре активного активного уровня города (мультикластерной) два дата-центра расположены в разных кластерах. Оба дата-центра активны и могут одновременно принимать бизнес-трафик. При аварии в одном дата-центре бизнес продолжает работать без прерывания в другом.

Терминология

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

Предварительные требования

- И основной кластер, и целевой кластер должны иметь включённый доступ к внешней сети. Для конкретных методов настройки см. Create Object Storage.
- Основной кластер должен использовать метод доступа **LoadBalancer**, а целевой кластер рекомендуется поддерживать функциональность **балансировки нагрузки**.
- Основной и целевой кластеры должны использовать одинаковый протокол доступа, то есть 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 и основной, и целевой кластеры использовали сертификаты, подписанные CA, для обеспечения безопасного и доверенного соединения; если используются самоподписанные сертификаты, обе стороны должны импортировать и доверять сертификатам друг друга для успешного установления защищённого HTTPS-соединения.

Шаги операции

1. Перейдите в **Управление платформой**.
2. В левой навигационной панели выберите **Управление хранилищем** > **Объектное хранилище**.
3. На вкладке **Восстановление данных после аварии** нажмите **Добавить целевой кластер**.
4. Настройте соответствующие параметры целевого кластера согласно следующим инструкциям.

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

5. Нажмите **Добавить**.
   - После успешного добавления вы сможете просмотреть статус целевого кластера и статус синхронизации между кластерами.

Параметр	Описание
Статус кластера	Статус целевого кластера, включая **Healthy**, **Abnormal** или **Unknown**.
Buckets	Количество бакетов, ожидающих синхронизации, и уже синхронизированных.

Параметр	Описание
	• В сценариях горячего резервного копирования ожидающая синхронизация — это количество бакетов, которые основной кластер должен синхронизировать с целевым кластером. • В сценариях городского уровня active-active ожидающая синхронизация — это общее количество бакетов, которые необходимо синхронизировать между основным и целевым кластерами.
Objects	Количество объектов, не синхронизированных в бакете. **Примечание**: Это число приведено для справки, так как MinIO синхронизирует связанные конфигурации файлов во время синхронизации.
Скорость сетевого трафика	Скорость входящего и исходящего сетевого трафика основного кластера. • В сценариях горячего резервного копирования скорость входящего трафика всегда равна 0. • В сценариях городского уровня active-active данные есть как по входящему, так и по исходящему трафику.

- Если добавление целевого кластера не удалось, вы можете нажать **Повторно добавить**, чтобы очистить информацию о кластере и вернуться на страницу добавления целевого кластера, где можно повторно добавить целевой кластер.

Связанные операции

Когда необходимость в аварийном восстановлении отпадёт, вы можете нажать **Удалить целевой кластер**. 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

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, обеспечивающее простую, удобную в обслуживании и высокопроизводительную возможность локального хранения данных. Основанное на решении TopoLVM из сообщества, оно реализует оркестрацию управления постоянными томами локального хранилища через системный подход LVM.

Содержание

Требования

Пакет lvm2 должен быть установлен на каждом узле кластера хранения. Если он не установлен, выполните команду `yum install -y lvm2` на узле.

Шаги

1. Перейдите в **Platform Management**.
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, что развертывание Operator прошло успешно.
- Если развертывание не удалось, следуйте подсказкам интерфейса для устранения проблемы. Затем нажмите **Clean Up** и повторно разверните Operator.

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

Параметр	**Описание**
Select Node	Узел с минимум одним неразмеченным диском.
Device Class	Каждый класс устройств соответствует набору устройств хранения с одинаковыми характеристиками. Рекомендуется указывать имя, исходя из 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 этих снимков для настройки новых PVC, что обеспечивает быструю резервную копию и восстановление бизнес-данных. Если при создании хранилища снимки не были включены, их можно активировать 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 хранения. Должно быть уникальным в рамках текущего кластера.
Display Name	Имя, помогающее идентифицировать или фильтровать, например, описание класса хранения на русском языке.
Device Class	Класс устройств — способ категоризации устройств хранения в TopoLVM. Каждый класс устройств соответствует набору устройств с одинаковыми характеристиками. Если нет 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 емкость 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.
Access Mode	ReadWriteOnce (RWO): может быть смонтирован одним узлом в режиме чтения и записи.
Allocation Project	Этот тип persistent volume claim может быть создан только в определённых проектах. Если проект не назначен временно, его можно **Обновить** позже.

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

Menu

Руководства

Управление устройствами

Предварительные требования

Добавление устройств

Мониторинг и оповещения

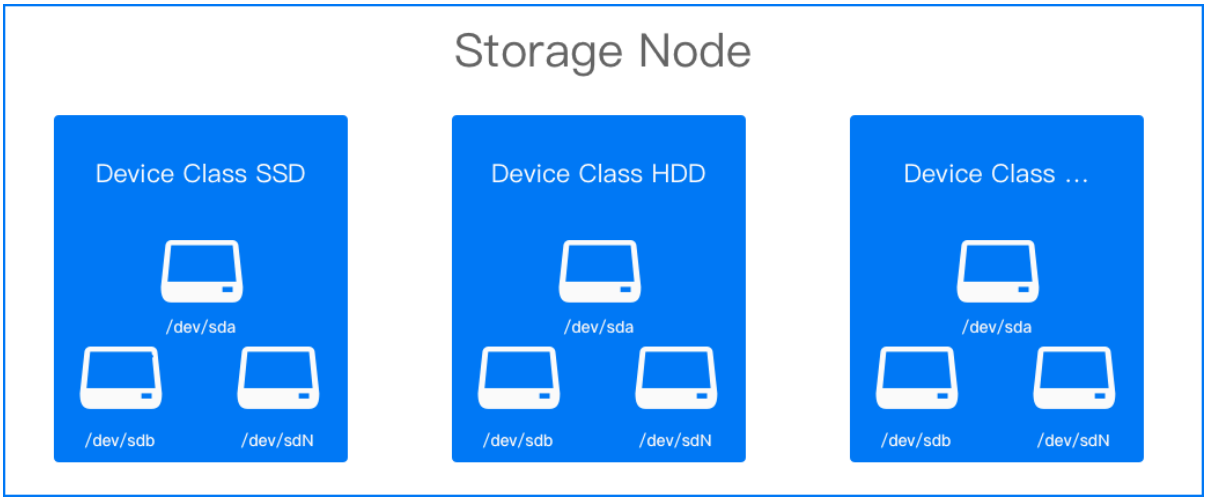
Мониторинг

Оповещения

Управление устройствами

Будь то для первоначального развертывания или расширения ресурсов, необходимо сопоставить доступные диски на узле с устройствами хранения для 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 классе устройств.
- На узле должен присутствовать как минимум 1 голый диск.

Добавление устройств

1. Перейдите в **Управление платформой**.
2. В левой навигационной панели выберите **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 `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 интерфейса.

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?