Руководство по установке REDROOM

Оглавление

Требования к инфраструктуре

Введение

Как и любое другое программное решение, установка и запуск платформы Redroom требует выполнения определённых требований к инфраструктуре.

Общие требования к инфраструктуре

- Redroom может работать только при рабочей инфраструктуре DNS. В составе Redroom есть внутренний сервер DNS, который должен быть доступен из корпоративного DNS как отдельная зона, также он может работать самостоятельно. Использование IP в качестве адресов подключения не поддерживается.
- Redroom не поддерживает запуск и функционирование через незашифрованные соединения, так как по умолчанию использует протокол HTTP/2. Поэтому в продуктивных средах нужно использовать сертификаты валидных центров сертификации (общедоступных, частных или внутренних) для корректной работы протокола TLS. Использование самоподписанных сертификатов возможно только при тестовых инсталляциях.
- Каждый кластер LXD должен располагаться в своей подсети с соответствующим диапазоном IP-адресов. При взаимодействии между кластерами LXD должна использоваться сетевая маршрутизация. В отдельных случаях допускается использование единой сети для части или всех кластеров LXD, однако для этого должны быть веские технические и организационные причины.
- Каждый кластер LXD должен использовать валидный уникальный кластерный сертификат TLS, так как этот сертификат используется как средство аутентификации системы управления Redroom к этим кластерам.

Требования к аппаратному обеспечению

Общие требования к аппаратному обеспечению

- Все узлы, добавленные в один кластер, должны иметь одинаковую конфигурацию по всем основным ресурсам (CPU, RAM, количество дисков, сетевых интерфейсов и так далее). Включение узлов с различной конфигурацией в одном кластере официально не поддерживается, однако такую конфигурацию допустимо использовать для тестовых инсталляций.
- Поддерживается архитектура процессоров x86_64.
- Во всех типах кластеров LXD должно быть как минимум 3 узла для обеспечения сохранности кворума и надежности.
- В качестве дисков должны быть использованы устройства SSD с контроллером SATA или NVMe корпоративного/серверного уровня (в зависимости от требуемых скоростных характеристик и цен на конечные комплектующие). Использование дисков HDD возможно для систем резервного копирования и иных “холодных” данных:
  - В SSD должна иметься возможность аварийного сброса буферизированных данных при отключении питания. Обычно для этих целей в SSD используют специальные конденсаторы.
  - Для очень критичных данных необходимо использовать SSD с TLC-чипами памяти и BER не менее 10^-9.
  - Для обычных данных или данных с достаточным количество реплик достаточно использовать SSD с MLC-чипами и BER не менее 10^-7. Такие же диски рекомендуются использовать для дисков с операционной системой.

Требования к компонентам узлов серверной части:

- CPU:
  - Intel Xeon или AMD Epyc для стандартных конфигураций;
  - Для RWP-приложений при определенных ситуациях может быть рекомендована конфигурация с процессорами Intel Xeon W или AMD Threadripper (обычная версия или Pro) для поддержки высокой частоты ядер.
- GPU:
  - Для поддержки 3D-ускорения рекомендуется использование карт AMD или Nvidia Consumer-уровня.

- Для поддержки 3D-ускорения и вычислений CUDA/OpenCL рекомендуется использовать профессиональные карты AMD и Nvidia.
- Для поддержки кодирования видеопотока используйте профессиональные карты Nvidia с поддержкой функций NVENC.

Требования к компонентам узлов клиентской части:

- CPU:
  - Intel Atom/Celeron или AMD Embedded на Zen.
  - Рекомендуется использование процессоров с поддержкой AES-NI.
- GPU:
  - Встроенные решения Intel Graphics 600+, AMD Vega 3+ или Nvidia 730+.
  - Рекомендуется поддержка аппаратного декодирования видеопотока. На данный момент такая поддержка имеется для Intel Graphics и AMD.

Минимальные требования к аппаратным характеристикам серверной части, на которой функционирует системой управления программным комплексом:

- Количество узлов: не менее 3 серверов;
- Архитектура CPU: x86_64;
- Количество физических ядер CPU: не менее 8;
- Частота физических ядер CPU: не менее 2.0 Ггц;
- Объем оперативной памяти: не менее 16 ГБ;
- Частота оперативной памяти: не ниже DDR4 2666 МГц ECC;
- Количество системных дисков: не менее 1;
- Объем системного диска: не менее 120 ГБ;
- Тип системного диска: SSD;
- Объем диска для хранения данных контейнерной виртуализации: не менее 120 ГБ;
- Тип диска для хранения данных контейнерной виртуализации: SSD;

- Возможно использование RAID1/10, однако не рекомендуется использование RAID5/6 по причине низкой производительности записи в случае использования баз данных.
- Количество сетевых интерфейсов: не менее 2:
  - mgmt - сеть, предназначенная для общения между сервисами Redroom, для работы с кластерами LXD и соединения с клиентами Redroom (но не RWP). Скорость этого интерфейса должна составлять не менее 1Gbit/s.
  - storage - сеть, предназначенная для доступа к системе хранения Ceph. Скорость этого интерфейса должна составлять не менее 10Gbit/s.
  - Допускается использование большего количества интерфейсов, например, для обеспечения отказоустойчивости сети или для дальнейшего разделения трафика.
- Сетевой интерфейс для контура управления: не менее 1 Гбит/сек;
- Сетевой интерфейс для контура передачи данных: не менее 10 Гбит/сек;
- Сетевой интерфейс удаленного администрирования серверным оборудованием: требуется;
- Клавиатура: требуется для локального администрирования;
- Мышь: требуется для локального администрирования;
- Монитор: требуется для локального администрирования.

Минимальные требования к аппаратным характеристикам серверной части, на которой функционирует гипервизор системной контейнерной виртуализации с удаленными графическими приложениями:

- Количество узлов: не менее 3 серверов;
- Архитектура CPU: x86_64;
- Количество сокетов CPU: не менее 2;
- Количество физических ядер CPU: не менее 8;
- Частота физических ядер CPU: не менее 2.2 Ггц;
- Объем оперативной памяти: не менее 64 ГБ;
- Частота оперативной памяти: не ниже DDR4 2666 МГц ECC;
- Количество системных дисков: не менее 1;
- Объем системного диска: не менее 120 ГБ;
- Тип системного диска: SSD;

- Объем диска для хранения данных контейнерной виртуализации: не менее 1000 ГБ;
- Тип диска для хранения данных контейнерной виртуализации: SSD;
- Для повышения отказоустойчивости допустимо использование RAID1/10, однако следует избегать RAID5/6 из-за низкой производительности записи при случайном доступе.
- mgmt - сеть, предназначенная для сервисов LXD кластера, через которые Redroom может отправлять им команды на выполнение:
  - Интерфейс mgmt также должен быть использован для загрузки ОС по протоколу PXE;
  - Скорость интерфейса должна составлять не менее 1Gbit;
  - Количество интерфейсов: не менее 1;
- appnet - сеть, предназначенная для доставки приложений через RWP до клиентов Redroom:
  - Скорость интерфейса должна составлять не менее 10Gbit;
  - Этот интерфейс должен поддерживать использование Jumbo Frame до 9000 байтов;
  - Количество интерфейсов: не менее 1;
- storage - сеть, предназначенная для доступа к системе хранения Ceph:
  - Скорость интерфейса должна составлять не менее 10Gbit;
  - Этот интерфейс должен поддерживать использование Jumbo Frame до 9000 байтов;
  - Количество интерфейсов: не менее 1;
- Допускается использование единого сетевого интерфейса для appnet и storage сетей;
- Сетевой интерфейс удаленного администрирования серверным оборудованием: требуется;
- Графический ускоритель: встроенный или дискретный;
- Клавиатура: требуется для локального администрирования;
- Мышь: требуется для локального администрирования;
- Монитор: требуется для локального администрирования.

Минимальные требования к аппаратным характеристикам серверной части, на которой функционирует программно-определяемая система хранения пользовательских данных (если планируется отдельный серверный контур хранения):

- На данный момент поддерживается только система хранения Ceph (блочный и файловый вариант хранения данных). Поддержка протоколов FibreChannel, iSCSI и сетевых файловых систем будет добавлена в будущих версиях.
- Количество узлов: не менее 3 серверов, Ceph должен хранить как минимум три копии пользовательских данных;
- Архитектура CPU: x86_64;
- Количество физических ядер CPU: не менее 8;
- Частота физических ядер CPU: не менее 2.2 Ггц;
- Объем оперативной памяти: не менее 32 ГБ;
- Частота оперативной памяти: не ниже DDR4 2666 МГц ECC;
- Количество системных дисков: не менее 1;
- Объем системного диска: не менее 120 ГБ;
- Тип системного диска: SSD;
- Количество дисков для хранения пользовательской информации: не менее 3;
- Объем диска для хранения пользовательской информации: не менее 500 ГБ;
- Тип диска для хранения пользовательской информации: SSD;
- Сайзинг системы хранения по количеству объема должен проводиться по расчетам необходимого количества пространства для хранения данных пользователей с учетом репликации и запаса на свободное место (20% от общей емкости системы хранения). К примеру, при наличии 100 пользователей и размером отдельного пользовательского каталога в 50GiB для каждого и при уровне репликации 3 требуется 100*50*3+(100*50*3*0.2)=18TiB. При использовании дисков размером в 500GiB получится 18/0.5=36 дисков.
- Сайзинг системы хранения по производительности выполняется индивидуально в зависимости от требований для каждого инстанса.
- Узлы системы хранения должны содержать как минимум три сетевых интерфейса:
- mgmt - сеть, предназначенная для взаимодействия между мониторами Ceph и службами Ceph OSD, запущенные в узлах хранения.
    - Интерфейс mgmt также должен быть использован для загрузки ОС по протоколу PXE;
    - Скорость интерфейса должна составлять не менее 1Gbit/s;
    - Количество интерфейсов: не менее 1;
- storage - сеть, предназначенная для доступа к данным прикрепленных пользовательских каталогов сервисами RWP с запущенными приложениями:
    - Скорость интерфейса должна составлять не менее 10Gbit/s;
    - Этот интерфейс должен поддерживать использование Jumbo Frame (до 9000 байтов);
    - Количество интерфейсов: не менее 1;
- storage-internal - сеть, предназначенная для внутренней репликации между узлами хранения;

    - Скорость интерфейса должна составлять величину, которая вычисляется путём умножения скорости интерфейса сети storage и количества узлов, куда необходимо отправить реплики данных (это число всегда равно количеству реплик минус 1). Если число реплик 3, а скорость интерфейса storage составляет 10Gbit/s, то минимальная скорость интерфейса storage-internal должна составлять не менее 10*(3-1)=20Gbit/s;
    - Этот интерфейс должен поддерживать использование Jumbo Frame (до 9000 байтов);
    - Количество интерфейсов: не менее 1;
- сетевой интерфейс удаленного администрирования серверным оборудованием: требуется;
- графический ускоритель: встроенный или дискретный;
- клавиатура: требуется для локального администрирования;
- мышь: требуется для локального администрирования;
- монитор: требуется для локального администрирования.

Минимальные требования к аппаратным характеристикам серверной части, на которой функционирует программно-определяемая система хранения резервных копий пользовательских данных:

- Для резервного копирования обязательно использование отдельной системы хранения Ceph. Ни в коем случае не храните данные резервного копирования в той же системе хранения, где хранятся сами данные пользователей.
- Резервные копии должны копироваться в файловую систему CephFS.
- Количество узлов: не менее 3 серверов;
- Архитектура CPU: x86_64;
- Количество физических ядер CPU: не менее 8;
- Частота физических ядер CPU: не менее 2.0 Ггц;
- Объем оперативной памяти: не менее 32 ГБ;
- Частота оперативной памяти: не ниже DDR4 2666 МГц ECC;
- Объем системного диска: не менее 120 ГБ;
- Тип системного диска: SSD;
- Количество дисков для хранения резервных копий пользовательской информации: не менее 8;

- Объем диска для хранения резервных копий пользовательской информации: не менее 1000 ГБ;
- Тип диска для хранения пользовательской информации: HDD;
- Количество сетевых интерфейсов: не менее 3;
- Сетевой интерфейс для контура управления: не менее 1 Гбит/сек;
- Сетевой интерфейс для контура передачи данных: не менее 10 Гбит/сек;
- Сетевой интерфейс для контура репликации данных: не менее 20 Гбит/сек;
- Сетевой интерфейс удаленного администрирования серверным оборудованием: требуется;
- Графический ускоритель: встроенный или дискретный;
- Клавиатура: требуется для локального администрирования;
- Мышь: требуется для локального администрирования;
- Монитор: требуется для локального администрирования.

Минимальные требования к аппаратным характеристикам клиентской части, на которой функционирует программное обеспечение для доступа в систему:

- Архитектура CPU: x86_64, ARMv8;
- Количество физических ядер CPU: не менее 2;
- Частота физических ядер CPU: не менее 1.0 Ггц;
- Объем оперативной памяти: не менее 2 ГБ;
- Частота оперативной памяти: не ниже DDR3 1333 МГц;
- Объем системного диска: не менее 32 ГБ;
- Тип системного диска: HDD;
- Количество проводных сетевых интерфейсов: не менее 1;
- Сетевой интерфейс для контура передачи данных: не менее 1 Гбит/сек;
- Количество беспроводных сетевых интерфейсов: не менее 1;
- Беспроводной сетевой интерфейс для контура передачи данных: не ниже WiFi 802.11n 2.4 ГГц;
- Графический ускоритель: встроенный или дискретный;
- Клавиатура: требуется;
- Мышь: требуется;
- Монитор: требуется.

Требования к программному обеспечению и протоколам

Серверная часть

- Операционная система
  - Ubuntu 20.04+/22.04+ - полная поддержка;
  - Alt Linux 9, 10 - планируется.
- Версия ядра: 5.x+ (рекомендуется 5.10+)
  - Для Ubuntu рекомендуется использовать generic-вариант ядра.
- Версия LXD: 4.19+
- Версия Ceph: 16.2.x+

Клиентская часть

- Операционная система
  - Ubuntu 20.04+/22.04+ - полная поддержка
  - Alt Linux 9, 10 - планируется.
- Версия ядра: 5.x (рекомендуется 5.10+)
  - Для Ubuntu рекомендуется использовать lowlatency-вариант ядра.
- Версия LXD: 4.19+
- Графическое окружение:
  - Для стандартной установки: KDE 5.18.x+
  - Для маломощных машин: LXQt 0.12.x+

Установка кластера LXD

Введение

Перед установкой компонентов управления Redroom вначале необходимо подготовить кластеры LXD:

- кластер LXD с компонентами управления Redroom и все сервисов для работы с инфраструктурой (например, сервисы менеджера Ceph). Основным отличием этого кластера является наличие проекта LXD с названием manager, где будут запущены все управляющие компоненты Redroom.
- кластер или кластеры LXD, которые будут использоваться в качестве вычислительных кластеров.

Установка сервиса

Установка LXD для всех типов кластеров выполняются одинаково. LXD в Ubuntu доступен через пакетный менеджер Snap (в Ubuntu Server пакет LXD уже будет установлен). Это можно узнать по команде показа списка доступных приложений Snap:

```
snap list | grep lxd
```

В ответ вы получите примерно следующий вывод:

```
lxd 5.0.1 21858 latest/stable canonical* -
```

Убедитесь, что версия LXD (второй столбец) равна 5.0.x (любая версия из ветки 5.0).

Запуск кластера для управляющих компонентов

Требования к кластеру

- Кластер управления должен содержать хотя бы три узла.
- Узел кластера должен содержать как минимум 50 ГБ свободного места для контейнеров управления.
- Должен иметься хотя бы один сетевой интерфейс, на базе которого можно создать управляющий интерфейс типа bridge.

Инициализация кластера

Основной командой инициализации кластера является выполнение команды

```
lxd init
```

Эта команда работает в интерактивном режиме: сервис спросит про данные хранения, настроит сетевую часть кластера и прочие настройки. Инициализацию нужно выполнить на всех узлах кластера.

Запуск инициализации на первом узле

Выберите один из узлов будущего кластера и запустите выше предложенную команду.

```
lxd init
```

Запустится интерактивная сессия, которая настроит сервис LXD в узле. Список вопросов следующий:

- Would you like to use LXD clustering? (yes/no) [default=no]
  - Параметр определяет, в каком режиме сервис LXD будет запущен.
  - Ответ: yes
- What IP address or DNS name should be used to reach this node? [default=node1.rr.local]
  - Указанный адрес узла будет использован для подключения к сервису LXD другими инстансами кластера LXD.
  - Ответ: укажите DNS или IP-имя узла в сети mgmt.
- Are you joining an existing cluster? (yes/no) [default=no]
  - Параметр определяет, подключиться ли настраиваемому сервису LXD в уже существующий кластер.
  - Ответ: no
- What name should be used to identify this node in the cluster? [default=dxl-zero]
  - Указывает имя сервиса LXD в кластере.
  - Ответ: укажите имя узла LXD в кластере. По умолчанию берется hostname.
- Setup password authentication on the cluster? (yes/no) [default=no]
  - Включает парольную аутентификацию к кластеру LXD.
  - Ответ: yes (в последующем этот параметр будет выключен).
- Trust password for new clients
  - Параметр отвечает за указание пароля для аутентификации.
  - Ответ: укажите пароль.
- Again
  - Параметр отвечает за повторный набор пароля для аутентификации.
  - Ответ: повторите указанный ранее пароль.
- Do you want to configure a new local storage pool? (yes/no) [default=yes]
  - Параметр, отвечающий за создание локального хранилища для контейнера.
  - Ответ: yes

- Name of the storage backend to use (dir, lvm, zfs, btrfs) [default=zfs]
  - Параметр типа локального хранилища.
  - Ответ: btrfs
- Create a new BTRFS pool? (yes/no) [default=yes]
  - Параметр создание нового пула BTRFS.
  - Ответ: yes
- Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]
  - Параметр использования имеющегося блочного дискового устройства.
  - Ответ: yes
- Path to the existing block device
  - Параметр пути до блочного устройства.
  - Ответ: укажите полный путь до блочного устройства, расположенный в /dev.
- Do you want to configure a new remote storage pool? (yes/no) [default=no]
  - Параметр добавления удаленного хранилища в кластер.
  - Ответ: no
- Would you like to connect to a MAAS server? (yes/no) [default=no]
  - Параметр интеграции с сервисом MAAS.
  - Ответ: no
- Would you like to configure LXD to use an existing bridge or host interface? (yes/no) [default=no]
  - Параметр настройки уже существующего бридж-интерфейса и его добавления в кластер.
  - Ответ: yes.
- Name of the existing bridge or host interface
  - Параметр имени существующего бридж-интерфейса.
  - Ответ: укажите имя бридж-интерфейса.
- Would you like stale cached images to be updated automatically? (yes/no) [default=yes]
  - Параметр обновления закэшированных образов в автоматическом режиме.
  - Ответ: yes
- Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]
  - Параметр предоставления файла preseed с данными кластера
  - Ответ: no

На этом инициализация первого узла закончится.

Инициализация остальных узлов

Для включения остальных узлов в кластер нужно пройти два больших шага:

- В первом узле нужно сгенерировать токен для аутентификации в вычислительном кластере.
- В остальных узлах нужно пройти шаги инициализации.

Генерация токена аутентификации

В первом узле запустите команду добавления узла в кластер:

```
lxc cluster add $ИМЯ_УЗЛА
```

Вместо имени узла нужно указать hostname добавляемого узла. Для каждого узла токен необходимо генерировать отдельно.

Токен будет выглядеть примерно так:

```
eyJzZXJ2ZXJfbmFtZSI6ImR4bC1vbmUiLCJmaW5nZXJwcmludCI6ImVkZDk5NGY2NG
JlMTg3OGFiMmI4YjJkNDBjYzc4NDFlMTczMGU1YmMxZDBhMWYwYjAwNGMyNDA1YTQz
MGM1OTAiLCJhZGRyZXNzZXMiOlsiMTAuMjM2LjY0LjI0Nzo4NDQzIl0sInNlY3JldC
I6IjFhNzE0Y2IwYjNiMGU2NDM2NzkyYjU5MDE1ZGRkM2Y3MWJhZDdjODdlZjU1NTdl
MjNjMTExZDZjN2I4YmVjOTgifQ==
```

Добавление узла в кластер

Для добавления узла в кластер необходимо провести инициализацию сервиса вычислений.

В добавляемом узле запустите команду:

```
lxd init
```

Список вопросов интерактивного режима:

- Would you like to use LXD clustering? (yes/no) [default=no]
  - Параметр определяет, в каком режиме сервис LXD будет запущен.
  - Ответ: yes
- What IP address or DNS name should be used to reach this node? [default=node2.rr.local]
  - Указанный адрес узла будет использован для подключения к сервису LXD другими инстансами кластера LXD.
  - Ответ: укажите DNS или IP-имя узла в сети mgmt.
- Are you joining an existing cluster? (yes/no) [default=no]
  - Параметр определяет, подключиться ли настраиваемому сервису LXD в уже существующий кластер.
  - Ответ: yes
- Do you have a join token? (yes/no/[token]) [default=no]:
  - Параметр использования токена аутентификации.
  - Ответ: укажите токен в поле ввода
- All existing data is lost when joining a cluster, continue? (yes/no) [default=no]

- Параметр очистки сервиса виртуализации перед добавлением кластера.
- Ответ: yes.

– Choose "source" property for storage pool "local"
  - Параметр определения пути для локального пула хранения.
  - Ответ: укажите блочное дисковое устройство.

– Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]
  - Параметр предоставления файла preseed с данными кластера
  - Ответ: no

Через некоторое время интерактивный режим закончится.

Проверка добавления узла в кластер

После добавления узла проверьте, что он действительно успешно добавился в кластер. Для этого выполните команду

```
lxc cluster list
```

Вы увидите примерно такой вывод:

```
+----------+-----------------------------+------------------+-----
---------+----------------+-------------+--------+----------------
---+

|   NAME   |            URL              |      ROLES       | AR-
CHITECTURE | FAILURE DOMAIN | DESCRIPTION | STATE  |      MESSAGE
|

+----------+-----------------------------+------------------+-----
---------+----------------+-------------+--------+----------------
---+

| dxl-one  | https://node2.rr.local:8443 | database-standby |
x86_64       | default        |             | ONLINE | Fully oper-
ational |

+----------+-----------------------------+------------------+-----
---------+----------------+-------------+--------+----------------
---+

| dxl-zero | https://node1.rr.local:8443 | database-leader  |
x86_64       | default        |             | ONLINE | Fully oper-
ational |

|          |                             | database         |
|               |             |        |                  |
```

```
+----------+------------------------------+------------------+-----
---------+----------------+-------------+--------+----------------
---+
```

Инициализация всего кластера закончится после добавления всех узлов.

Установка Redroom Manager

Введение

Redroom Manager — это приложение, предназначенное для управления инфраструктурой по предоставлению удалённых графических приложений. Экосистема Redroom сама по себе содержит достаточно большое количество компонентов, и Manager берет на себя функцию “дирижера”, регистрируя эти компоненты в базе данных и позволяя взаимодействовать между собой и конечными пользователями.

Эта статья расскажет про установку Manager и его первичный запуск.

Установка компонентов Manager

Как и все остальные модули, Redroom Manager поставляется в виде готового образа для запуска в качестве контейнера LXD. Отдельно Manager не распространяется. Поэтому первым шагом необходима настройка хостовой системы для запуска контейнеров.

Настройка хостовой системы

На данный момент в качестве хостовой системы используется ОС Ubuntu 22.04.

Для хостовой системы жестких требований нет, но нужно обратить внимание на следующее:

- Минимальная версия LXD: 5.0
- Для сети у вас должен быть настроен bridge (для обычного варианта сетевой настройки) или иметь отдельный физический интерфейс (для SDN).
- Нужно как минимум три физических узла с одинаковой хостовой системой.

Также хостовая система должна быть настроена в соответствие с этой статьёй.

Добавление службы образов Manager

DevBand официально предоставляет свою службу образов для Manager, которая выполняет роль репозитория стабильных версий Manager. Для получения доступа к образам необходимо следующее:

- Служба образов DevBand запущена с использованием TLS-аутентификации. Часть образов распространяются без TLS-аутентификации.

- Адрес сервиса образа должен быть добавлен в remote сервисов LXD в управляющих узлах:

```
lxc remote add redroom-manager-1.0 https://images.devband.ru:8443
--protocol=lxd
```

- Для получения доступа к публичным образам, к команде выше добавьте параметр --public.

После добавления remote при команде получения образов появятся образы Manager:

```
lxc image list
```

Запуск и настройка контейнеров Manager

После получения доступа к образам можно приступить к запуску Manager.

Вначале необходимо создать первый контейнер, в котором первоначально будет запущен Manager. Для этого в первом узле выполните:

```
lxc launch redroom-manager-1.0/stable redroom-manager-node1
```

По умолчанию ни одна служба, связанная с менеджером, внутри контейнера не запустится, так как их необходимо предварительно настроить.

Сам контейнер содержит следующие сервисы:

- База данных MariaDB, предназначенный для хранения состояния платформы.
- Сама служба redroom-manager, состоящий из фреймворка Django и ASGI-сервера HyperCorn.
- Nginx в качестве https proxy-сервера к ASGI-серверу.
- Балансировщик нагрузки haproxy.
- Менеджер виртуального IP-адреса (VIP) keepalived.

Контейнер содержит и прочие вспомогательные службы, их описание можно найти в описании к архитектуре платформы Redroom.

Пока дальнейшие команды должны выполняться в одном из контейнеров.

База данных MariaDB

Перед созданием базы данных крайне рекомендуем использовать отдельное устройство для хранения данных баз данных MariaDB.

Вначале в пуле хранения default во всех управляющих узлах нужно создать диск минимальным размером в 10 GiB. Для этого нужно выполнить (default здесь - имя пула хранения данных, настроенный при первом запуске):

```
lxc storage volume create default redroom-manager-db
```

Этот диск потом нужно добавить в конфигурацию контейнера LXD:

```
lxc config device add redroom-manager-node1 redroom-manager-db
disk source=default:redroom-manager-db path=/var/lib/mysql
```

Затем зайдите внутрь контейнера:

```
lxc shell redroom-manager-node1
```

Теперь откройте файл /etc/mysql/conf.d/99-cluster.cnf. Он будет иметь следующее содержимое:

```
[mysqld]

max_connections = 8192

binlog_format = ROW

default-storage-engine = innodb

innodb_autoinc_lock_mode = 2

bind-address = 127.0.0.1

innodb_flush_log_at_trx_commit = 0

wsrep_slave_threads = 1

sync_binlog = 0

gtid_domain_id = $DIFFERENT_GTID_DOMAIN_ID



# WREP

wsrep_on = OFF

wsrep_provider = /usr/lib/galera/libgalera_smm.so

wsrep_cluster_address = gcomm://$FIRST_NODE,$SEC-
OND_NODE,$THIRD_NODE
```

```
wsrep_cluster_name = redroom

wsrep_sst_auth = redroom:redroom

wsrep_gtid_mode = ON

wsrep_gtid_domain_id = $SAME_WSREP_GTID_DOMAIN_ID

log_slave_updates = ON

wsrep_sst_method = rsync

wsrep_node_address = $CURRENT_NODE_ADDR

wsrep_node_name = $CURRENT_NODE_NAME
```

Из того, что следует поменять:

- **gtid_domain_id** - *уникальный* для каждого узла идентификатор узла GTID, обычное число от 1 и выше.
- **wsrep_on** - включает кластер репликации Galera для MariaDB, нужно указать “ON”.
- **wsrep_cluster_address** - все адреса кластеров Galera.
- **wsrep_sst_auth** - указывает параметры аутентификации узлов Galera. Можно менять на любые значения, главное, чтобы они были одинаковы для всех узлов.
- **wsrep_gtid_domain_id** - *одинаковый* для всех узлов идентификатор GTID кластера.
- **wsrep_node_address** - определяет IP-адрес узла кластера. Требуется указать IP, полученный контейнером.
- **wsrep_node_name** - указывает на имя узла кластера. Лучше указать доменное имя узла.

Пример настройки конфигурации:

```
[mysqld]

max_connections = 8192

binlog_format = ROW

default-storage-engine = innodb

innodb_autoinc_lock_mode = 2

bind-address = 127.0.0.1

innodb_flush_log_at_trx_commit = 0

wsrep_slave_threads = 1
```

```
sync_binlog = 0

gtid_domain_id = 11



# WREP

wsrep_on = ON

wsrep_provider = /usr/lib/galera/libgalera_smm.so

wsrep_cluster_address =
gcomm://zero.redrum.loc,one.redrum.loc,two.redrum.loc

wsrep_cluster_name = redroom

wsrep_sst_auth = redroom:redroom

wsrep_gtid_mode = ON

wsrep_gtid_domain_id = 42

log_slave_updates = ON

wsrep_sst_method = rsync

wsrep_node_address = 10.236.64.126

wsrep_node_name = dxl-zero.redroom.local
```

Далее создайте начальные базы данных:

```
mariadb-install-db
```

После этой настройки запустите новый кластера Galera:

```
galera_new_cluster
```

Если все ОК, то команда молча завершит свою работу. Статус работы базы можно проверить через systemd:

```
systemctl status mariadb
```

Не лишним будет выполнить скрипт активации безопасных настроек СУБД:

```
mariadb-secure-installation
```

Настройка службы Redroom Manager

Вначале для Manager создайте базу данных в MariaDB. Для этого запустите следующие команды:

```
mariadb -e "CREATE DATABASE redroom;"

mariadb -e "GRANT ALL PRIVILEGES ON redroom.* TO 'redroom'@'localhost' IDENTIFIED BY 'your_password'";
```

Далее нужно настроить файл /etc/redroom/db.conf. Изначально он выглядит так:

```
[client] database = redroom user = redroom password = redroom

default-character-set = utf8
```

В ней смените на пароль, который был указан при предоставлении прав внутри СУБД:

```
[client] database = redroom user = redroom password = your_password

default-character-set = utf8
```

Перейдите в каталог с Manager:

```
cd /srv/redroom
```

Вначале выполните миграцию базы:

```
python3 manage.py migrate
```

Соберите статичные данные:

```
python3 manage.py collectstatic
```

Наконец, создайте суперпользователя:

```
python3 manage.py createsuperuser
```

Настройка службы завершена, запустите его:

```
systemctl start redroom-manager
```

Так же поместите службу в список автозапуска:

```
systemctl enable redroom-manager
```

Настройка прокси-сервера Nginx

Для лучшей производительности и безопасности мы в платформе используем промежуточный до самого Manager прокси-сервер на базе Nginx. Его по умолчанию не нужно настраивать, его просто нужно запустить:

```
systemctl start nginx
```

И поместить в автозапуск:

```
systemctl enable nginx
```

Отметим лишь, что конфигурация для Manager находится по пути /etc/nginx/conf.d/redroom.conf.

Настройка балансировщика нагрузка HAProxy

Для настройки балансировщика нагрузки требуется изменить файл /etc/haproxy/haproxy.conf. По умолчанию он выглядит так:

```
global

  maxconn 81920

  tune.ssl.default-dh-param 2048

  pidfile /run/haproxy.pid

  stats socket /run/haproxy.sock level admin

  ssl-default-bind-ciphersuites
TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY130
5_SHA256

  ssl-default-bind-options no-sslv3 no-tlsv10 no-tlsv11 no-tlsv12
no-tls-tickets



defaults

  mode http

  option http-use-htx

  option forwardfor

  timeout connect 5s
```

```
  timeout client 5s

  timeout server 5s



frontend rr_nginx_frontend

  mode http

  bind $VIP_ADDRESS:443 ssl crt /etc/redroom/tls/main.pem alpn
h2,http/1.1

  http-request add-header X-Forwarded-Proto "https"

  default_backend rr_nginx_backend



backend rr_nginx_backend

  mode http

  server node1 $FIRST_SERVER_DNS_NAME:8443 check send-proxy-v2 ssl
verify none

  server node2 $SECOND_SERVER_DNS_NAME:8443 check send-proxy-v2
ssl verify none

  server node3 $THIRD_SERVER_DNS_NAME:8443 check send-proxy-v2 ssl
verify none



listen stats

  bind $IP_STATS:8080

  stats enable

  stats uri /status/

  stats realm "Redroom HAPRoxy Stats"

  stats auth $NAME:$PASSWORD
```

Здесь нужно поменять:

- в frontend - bind нужно поменять адрес привязки $VIP_ADDRESS на виртуальный IP-адрес, выбранный для платформы. Необходимо использовать DNS-имя с резолвингом на этот адрес.
- Далее в backend - server нужно указать DNS-имена всех экземпляров Manager.
- В listen stats - bind нужно указать DNS-имя сервера, где запущен HAProxy.

После настройки запустите службу HAProxy:

```
systemctl start haproxy
```

Так же добавьте сервис в автозапуск:

```
systemctl enable haproxy
```

Настройка службы keepalived

Последняя основная служба, требующая настройки — это служба виртуального IP-адреса keepalived. По умолчанию его конфигурация выглядит так:

```
global_defs {
  router_id $ROUTER_ID
}

vrrp_instance $INSTANCE_NAME {
  state BACKUP
  priority 100
  interface eth0                        # Network card
  virtual_router_id $VIRTUAL_ROUTER_ID
  advert_int 1
  authentication {
    auth_type PASS
    auth_pass $PASSWORD
```

```
  }


  virtual_ipaddress {
    $VIP_ADDRESS/32                 # The VIP address
  }
}
```

Тут нужно сменить следующие части:

- router_id – ID роутера keepalived в виде строки текста. Должен быть одинаковый во всех узлах кластера и не совпадать с ID других кластеров keepalived.
- virtual_router_id – числовой ID роутера. Тоже должен быть одинаковым для всех узлов кластера и не совпадать с ID других.
- auth_pass $PASSWORD – укажите пароль для взаимодействия между узлами keepalived. Должен быть одинаковым во всех узлах.
- $VIP_ADDRESS – здесь необходимо указать виртуальный IP-адрес. /32 всегда должен иметься в адресе.

Запустите службу keepalived:

```
systemctl start keepalived
```

Добавьте службу в автозагрузку:

```
systemctl enable keepalived
```

На этом настройка первого узла кластера завершается.

Проверьте вход в веб-панель Redroom. Для этого в своем веб-браузере загрузите адрес

```
https://VIP_ADDRESS/admin
```

Вы должны получить страницу приглашения на вход. Наберите данные суперпользователя и зайдите в основную страницу администрирования платформы.

Настройка остальных узлов кластера

Их настройка состоит в копировании полученных в первом узле конфигурационных файлов, небольших изменений в этих копиях и запуска самих служб.

Настройка MariaDB

Вначале не забудьте, как и в случае первого узла, создать отдельный диск для базы данных.

Далее скопируйте содержимое файла /etc/mysql/conf.d/99-cluster.cnf с первого узла, и измените адрес узла в конфигурации:

```
wsrep_node_address = $CURRENT_NODE_ADDR

wsrep_node_name = $CURRENT_NODE_NAME
```

После этого запустите службу MariaDB:

```
systemctl start mariadb
```

После чего добавьте службу в автозапуск:

```
systemctl enable mariadb
```

Настройка Redroom Manager

Здесь нужно скопировать полученный файл /etc/redroom/db.conf с первого узла без изменений.

Затем нужно запустить службу Manager:

```
systemctl start redroom-manager
```

И поместить в автозагрузку:

```
systemctl enable redroom-manager
```

Настройка прокси-сервера Nginx

Дополнительная настройка Nginx на остальных инстансах так же не требуется, просто запустите сам прокси-сервер:

```
systemctl start nginx
```

И поместите его в автозагрузку:

```
systemctl enable nginx
```

Настройка балансировщика нагрузка HAProxy

Тут так же необходимо взять конфигурацию HAProxy с первого узла и поместить на оставшиеся по пути /etc/haproxy/haproxy.conf. В ней нужно изменить лишь DNS-имя контейнера в listen stats, предназначенный для показа статистики.

После чего нужно запустить HAProxy:

```
systemctl start haproxy
```

И поместить в автозагрузку:

```
systemctl enable haproxy
```

Настройка службы keepalived

Здесь так же нужно взять полученный конфигурационный файл с первого узла, после чего запустите службу keepalived:

```
systemctl start keepalived
```

И поместить службу в автозагрузку:

```
systemctl enable keepalived
```

На этом основная настройка Manager заканчивается. При добавлении новых контейнеров Manager в кластер повторите все те же шаги, что были указаны в этом разделе.

Настройка сервиса Redroom SDN

Введение

Redroom по умолчанию использует функции виртуализации сетей гипервизора с помощью сервиса программно-определяемых сетей, основанные на технологиях проекта Open Virtual Network.

Настройка сервиса делится на две части:

- Запуск центрального части сервиса SDN, который называется ovn-central. Данный сервис доступен в официальном репозитории DevBand Images по алиасу “rr-net”.
- Настройка вычислительных кластеров для доступа к базе данных SDN и построению виртуальных сетей.

Краткая информация о SDN

Базовая архитектура решения представлена здесь.

Комментарий по компонентам:

- Redroom Manager — это веб-сервис платформы. Этот компонент инициализирует основные команды по управлению сетями.
- Manager Cluster — это вычислительный кластер, который содержит основные компоненты системы управления платформой, в том числе Manager.
- OVN Northbound DB (OVN NB) — это база данных по данным северных интерфейсов, предоставляемый сервисом OVN.
- ovn-northd — это сетевой сервис OVN, предоставляющий механизм доступа к OVN NB.
- OVN Southbound DB (OVN SB) — это база данных по данным южных интерфейсов, предоставляемый сервисом OVN.
- Compute Cluster — это вычислительный кластер для запуска конечных контейнеров.
- ovn-controller — это компонент OVN, который предоставляет виртуальному контроллеру доступ к OVN SB.

- ovs-vswitchd и ovsdb-server - это компоненты виртуального контроллера Open vSwitch.
- Containers - конечные пользовательские контейнеры.

Нужно отметить некоторые моменты:

- OVN NB/SB запускаются в Manager Cluster в виде обычного общего контейнера с использованием штатного образа.
- В продуктивной среде экземпляров OVN NB/SB должно быть как минимум три.
- В качестве адресов узлов кластера OVN NB/SB по умолчанию должны быть использованы имена DNS, так как в конфигурации сервиса OVN внутри контейнера нужно указывать адреса всех экземпляров кластера и они должны быть статичны. IP-адреса использовать допустимо, однако необходимо, что экземпляры кластера OVN всегда имели одни и те же адреса.
- Один кластер OVN NB/SB можно использовать для нескольких вычислительных кластеров. При этом разрешается использование подсетей с пересекающими диапазонами, так как разделение сетей происходит по тегам сетевых туннелей Geneve.
- В каждом узле вычислительного кластера должен быть отдельный неуправляемый интерфейс (физический или типа мост) для доступа к внешним сетям и другим виртуальным сетям OVN (так называемый uplink-интерфейс, на схеме не указано).
- Вычислительный кластер должен иметь доступ до OVN NB по сети management.

Требования к оборудованию и настройке

Центральный сервис OVN

- OVN с NB/SB особых требований к оборудованию не возлагает. Сам дистрибутив предоставляется в виде преднастроенного образа контейнера.
- Настоятельно рекомендуется производить периодическое создание снимков и резервных копий приложением Recovery, так как информация о сетях в NB/SB является критичной.

- Основное требование: публичные порты OVN NB и SB должны доступны через сеть management.

Вычислительный узел с компонентами OVN

- Для нормального функционирования компонентов OVN вычислительного узла требуется как минимум 2 интерфейса.
    - Первый интерфейс предназначен для доступа к сети management, через который вычислительный кластер и виртуальный коммутатор сможет взаимодействовать с центральным сервисом OVN.
    - Второй интерфейс требуется для реализации так называемого ненастроенного uplink-интерфейса, позволяющий OVN маршрутизировать виртуальные сети между собой, а также предоставляющие доступ к внешним сетям, в частности, к Интернету.
    - Uplink-интерфейс не может использоваться для иных целей, например, для сети системы хранения.

Версии ПО:

- OVN не должен иметь версию ниже 22.03.

Установка центральной части

Создание контейнеров

Установка базы данных OVN NB/SB сводится к получению образа и указанию адресов экземпляров кластера OVN Central.

Запустите три контейнера на базе образа devband:rr-network в управляющем кластере Manager:

```
lxc launch devband:rr-net rr-nb-sb-0 lxc launch devband:rr-net rr-nb-sb-1

lxc launch devband:rr-net rr-nb-sb-2
```

После запуска контейнеры получат записи во внутренней зоне DNS, вид их записей зависят от настроек инсталляции.

Настройка первого контейнера

По умолчанию сервис ovn-central не запустится, он требует дополнительно настройки. Зайдите в оболочку первого контейнера:

```
lxc shell rr-nb-sb-0
```

Откройте файл /etc/default/ovn-central и измените его с помощью следующего шаблона:

```
OVN_CTL_OPTS= \

   --db-nb-addr=<MGMT_ADDR_C1> \

   --db-nb-create-insecure-remote=yes \

   --db-sb-addr=<MGMT_ADDR_C1> \

   --db-sb-create-insecure-remote=yes \

   --db-nb-cluster-local-addr=<CLSTR_ADDR_C1> \

   --db-sb-cluster-local-addr=<CLSTR_ADDR_c2> \

   --ovn-northd-nb-
db=tcp:<MGMT_ADDR_C1>:6641,tcp:<MGMT_ADDR_C2>:6641,tcp:<MGMT_ADDR_
C3>:6641 \

   --ovn-northd-sb-
db=tcp:<MGMT_ADDR_C1>:6642,tcp:<MGMT_ADDR_C2>:6642,tcp:<MGMT_ADDR_
C3>:6642
```

Здесь требуется указать:

- db-nb-addr — это публичный адрес базы данных NB. Нужно указать DNS-имя или постоянный IP-адрес с адресом в сети management.
- db-sb-addr — это публичный адрес базы данных SB. Нужно указать DNS-имя с адресом в сети management.
- db-nb-cluster-local-addr — это внутренний адрес репликации базы NB. Может иметь адрес в сети management и совпадать с db-nb-addr, но рекомендуется использовать адрес в отдельной изолированной сети.
- db-sb-cluster-local-addr — это внутренний адрес репликации базы NB. Может иметь адрес в сети management и совпадать с db-sb-addr, но рекомендуется использовать адрес в отдельной изолированной сети.

- ovn-northd-nb-db — это адреса всех экземпляров кластеров OVN NB. Адреса в этой опции должны совпадать с тем, что указано в db-nb-addr для соответствующего экземпляра.
- ovn-northd-sb-db — это адреса всех экземпляров кластеров OVN SB. Адреса в этой опции должны совпадать с тем, что указано в db-sb-addr для соответствующего экземпляра.

После указания этих параметров запустите сервис ovn-central:

```
systemctl enable ovn-central

systemctl start ovn-central
```

Проверьте, что сервис ovn-central успешно запустился:

```
systemctl status ovn-central
```

Настройка последующих контейнеров

Настройка оставшихся контейнеров почти ничем не отличается от настройки первого. Главное изменение — это указание первого узла как источника репликации. Конфигурация должны выглядеть так:

```
OVN_CTL_OPTS=" \

    --db-nb-addr=<local> \

     --db-nb-cluster-remote-addr=<server_1> \

     --db-nb-create-insecure-remote=yes \

     --db-sb-addr=<local> \

     --db-sb-cluster-remote-addr=<server_1> \

     --db-sb-create-insecure-remote=yes \

     --db-nb-cluster-local-addr=<local> \

     --db-sb-cluster-local-addr=<local> \

     --ovn-northd-nb-
db=tcp:<server_1>:6641,tcp:<server_2>:6641,tcp:<server_3>:6641 \

     --ovn-northd-sb-
db=tcp:<server_1>:6642,tcp:<server_2>:6642,tcp:<server_3>:6642"
```

- В параметрах db-nb-addr и db-sb-addr необходимо указать свои адреса из сети management, которые предоставлены контейнерам.
- В параметрах db-nb-cluster-local-addr и db-sb-cluster-local-addr необходимо указать свои адреса из изолированной сети для кластеризации OVN Central или указать адреса в сети management, соответствующие параметрам db-nb-addr и db-sb-addr соответственно.

После сохранения настроек в каждом из оставшихся контейнеров нужно запустить сервис onv-central:

```
systemctl enable ovn-central

systemctl start ovn-central
```

На этом настройка центрального сервиса OVN завершена.

Некоторые советы

- Как уже был сказано выше, желательно для кластеризации данных экземпляров OVN использовать свою внутреннюю изолированную сеть. Для этого просто необходимо создать полностью виртуальную сеть и добавить его в контейнеры OVN в качестве второй сети.
- OVN NB и OVN SB содержит всю виртуальную топологию сетей и критически важно сохранить в случае сбоев. Поэтому рекомендуется периодически создавать снимки и, реже, резервные копии контейнеров OVN.

Настройка кластера LXD

Настройка кластера LXD сводится к настройке виртуального коммутатора и добавления uplink-интерфейса.

Поддерживается как запуск OVN при создании нового кластера, так и переход кластера LXD на OVN (без сохранения сетей).

Установка компонентов OVN

Перед настройкой OVN в список репозиториев узлов должен быть добавлен репозиторий Cloud Archive Yoga для Ubuntu Focal (для OVN в Ubuntu Jammy используются штатные пакеты). Для этого нужно во всех узлах кластера LXD создать файл

```
/etc/apt/sources.list.d/cloudarchive-yoga.list:

echo 'deb http://ubuntu-cloud.archive.canonical.com/ubuntu focal-
updates/yoga main' > /etc/apt/sources.list.d/cloudarchive-
yoga.list
```

Обновите метаданные репозиториев:

```
apt update
```

После чего установите пакет ovn-host:

```
apt install -y ovn-host
```

Запуск компонентов OVN

В каждом кластере необходимо запустить сервис ovn-host:

```
systemctl enable ovn-host

systemctl enable ovn-host
```

Убедитесь, что сервис успешно запустился:

```
systemctl status ovn-host
```

Настройка виртуального коммутатора

Сам виртуальный коммутатор вручную настраивать не нужно, достаточно указать адреса OVN SB:

```
ovs-vsctl set open_vswitch . external_ids:ovn-re-
mote=tcp:<MGMT_ADDR_C1>:6642,tcp:<MGMT_ADDR_C2>:6642,tcp:<MGMT_ADD
R_C3>:6642 \

ovs-vsctl set open_vswitch . external_ids:ovn-encap-type=geneve

ovs-vsctl set open_vswitch . external_ids:ovn-encap-
ip=<MGMT_ADDR_LOCAL>
```

Вместо MGMT_ADDR_LOCAL укажите локальный адрес узла в сети management.

Создание uplink-интерфейса

Для маршрутизации сетевых пакетов между различными сетями OVN (как между собой, если не используются network peers, так и с внешними

сетями) требуется так называемый uplink-интерфейс. Этот интерфейс не должен никак настраиваться операционной системой и передан кластеру LXD как есть. В качестве uplink-интерфейса может быть передано физическое устройство, bond, а также мост-интерфейс (bridge).

Перед добавлением выбранного интерфейса в uplink убедитесь, что этот интерфейс на всех узлах кластера не содержит IPv4- или IPv6-адрес. Иначе при попытке добавить сеть OVN на базе этого uplink-интерфейса вы получите ошибку:

```
Error: Cannot start network as uplink network interface "$inter-
face_name" has one or more IP addresses configured on it
```

Вначале во всех узлах кластера LXD нужно добавить данные uplink-интерфейса:

```
 lxc network create UPLINK --type=physical parent=<uplink_inter-
face> --target=<machine_name_1>

 lxc network create UPLINK --type=physical parent=<uplink_inter-
face> --target=<machine_name_2>

 lxc network create UPLINK --type=physical parent=<uplink_inter-
face> --target=<machine_name_3>

 lxc network create UPLINK --type=physical parent=<uplink_inter-
face> --target=<machine_name_4>
```

Заметьте, что для каждой машины можно указать uplink-интерфейс с различными именами, однако настоятельно рекомендуется унифицировать имена интерфейсов в ОС узлов перед добавлением. При добавлении этих данных сеть UPLINK будет в статусе Pending.

Далее необходимо инициализировать сеть UPLINK:

```
lxc network create UPLINK --type=physical \

   ipv4.ovn.ranges=IPv4START-IPv4END \

   ipv6.ovn.ranges=IPv6START-IPv6END \

   ipv4.gateway=IPv4_GW_ADDR \

   ipv6.gateway=IPv6_GW_ADDR \

   dns.nameservers=DNS_ADDR
```

Нужно отметить следующие моменты:

- ipv{4,6}.ovn.ranges - диапазон IP-адресов для виртуальной сети OVN. Эти адреса используются для адресации точек соединения сетевых туннелей.
  - Пример: 10.5.1.2-10.5.1.251
- ipv{4,6}.gateway - это шлюз, который будет использован адресами в ovn.ranges для маршрутизации в другие сети.
  - Пример: 10.5.1.1/24
- dns.nameservers - это список DNS-серверов, можно указать до трёх через запятую:
  - Пример: 10.3.1.2,8.8.8.8

Данная команда фактически создаст сеть UPLINK, его статус изменится на Created.

Настройка доступа до OVN NB

Кластер LXD должен иметь доступ до OVN NB, что позволит создавать записи о создаваемых сетях в базе данных OVN, которые дальше через OVN SB будут применяться на конечных виртуальных коммутаторах.

Для этого необходимо запустить следующую команду на любом узле кластера LXD):

```
lxc config set network.ovn.northbound_connection
tcp:<MGMT_ADDR_C1>:6641,tcp:<MGMT_ADDR_C2>:6641,tcp:<MGMT_ADDR_C3>
:6641
```

Система управления Ceph

Введение

Репозиторий образов DevBand содержит контейнеры для предустановленными компонентами системы управления Ceph. В этой статье описана установка и настройка части управления и добавление блочных устройств в Ceph.

Добавление контейнера с компонентами управления Ceph

Инициализация первого контейнера

В одном из кластеров управления запустите контейнер на базе образа с меткой rr-stor-mgr:

```
lxc launch redroom-manager-1.0/rr-stor-mgr ceph-node-1
```

Зайдите в окружение контейнера:

```
lxc shell ceph-node-1
```

Откройте файл /etc/ceph/ceph.conf. Он должен выглядеть примерно так:

```
[global]

fsid =

mon initial members =

mon host =

public network =

cluster network =

cephx_require_signatures = cephx

cephx_cluster_require_signatures = cephx

cephx_sign_messages = cephx

min_alloc_size = 16384
```

```
osd journal size = 1024

osd pool default size = 3

osd pool default min size = 2

osd crush chooseleaf type = 1



# https://docs.ceph.com/en/latest/security/CVE-2021-20288/

auth_allow_insecure_global_id_reclaim = false
```

В пустые строки нужно ввести данные:

- fsid. Уникальный идентификатор кластера. Укажите комбинацию UUID4.
- mon initial members. Список имен узлов кластера. Должно совпадать с именами контейнеров. Для примера здесь принято, что три контейнера управления Ceph будут иметь имена ceph-node-1, ceph-node-2 и ceph-node-3.
- mon host. IP-адрес монитора. Необходимо указать адрес интерфейса контейнера (в сети mgmt).
- public network. Определение публичной подсети в формате CIDR. Необходимо указать подсеть сети mgmt.
- cluster network. Определение подсети репликации данных Ceph в формате CIDR. Для компонентов управления этот параметр игнорируется, но в информационных целях укажите подсеть репликации данных. По умолчанию она равна 169.254.0.0/16.

После указания всех параметров перезапустите сервис ceph-mon:

```
systemctl restart ceph-mon@ceph-node-1
```

После запуска монитора проверьте статус установки:

```
ceph -s
```

Ответ будет выглядеть примерно так:

```
  cluster:

    id:     8a8cd0a4-1680-43f6-a084-fd411a122d16

    health: HEALTH_OK
```

```
  services:
    mon: 1 daemons, quorum ceph-node-1 (age 6s)
    mgr: no daemons active
    osd: 0 osds: 0 up, 0 in

  data:
    pools:   0 pools, 0 pgs
    objects: 0 objects, 0 B
    usage:   0 B used, 0 B / 0 B avail
    pgs:
```

Инициализация остальных контейнеров

После инициализации первого контейнера необходимо поднять оставшиеся. Каждый инстанс компонентов управления нужно поднимать по очереди.

Вначале запустите второй контейнер:

```
lxc launch redroom-manager-1.0/rr-stor-mgr ceph-node-2
```

Зайдите в окружение контейнера:

```
lxc shell ceph-node-2
```

Откройте файл /etc/ceph/ceph.conf. Он должен выглядеть примерно так:

```
[global]
fsid =
mon initial members =
mon host =
public network =
cluster network =
```

```
cephx_require_signatures = cephx

cephx_cluster_require_signatures = cephx

cephx_sign_messages = cephx

min_alloc_size = 16384

osd journal size = 1024

osd pool default size = 3

osd pool default min size = 2

osd crush chooseleaf type = 1


# https://docs.ceph.com/en/latest/security/CVE-2021-20288/

auth_allow_insecure_global_id_reclaim = false
```

Конфигурация должна совпадать с тем, что было указано в первом контейнере, кроме:

- mon_host. В этом параметре нужно указать свой IP-адрес контейнера.

После запуска монитора проверьте статус установки:

```
ceph -s
```

Ответ будет выглядеть примерно так:

```
[global]

fsid =

mon initial members =

mon host =

public network =

cluster network =

cephx_require_signatures = cephx
```

```
cephx_cluster_require_signatures = cephx

cephx_sign_messages = cephx

min_alloc_size = 16384

osd journal size = 1024

osd pool default size = 3

osd pool default min size = 2

osd crush chooseleaf type = 1


# https://docs.ceph.com/en/latest/security/CVE-2021-20288/

auth_allow_insecure_global_id_reclaim = false
```

Повторите эти шаги для третьего контейнера.

Активация сервиса mgr

Для полноценной работы компонентов управления необходима активация сервиса mgr.

Создайте каталог для mgr в первом контейнере:

```
mkdir -p /var/lib/ceph/mgr/ceph-mgr-ceph-node-1
```

Внутрь этого каталога добавьте ключ mgr:

```
ceph auth get mgr."ceph-node-1" -o /var/lib/ceph/mgr/ceph-mgr-ceph-node-1/keyring
```

После этого запустите сервис mgr:

```
systemctl status ceph-mgr@ceph-node-1
```

Проверьте статус сервиса mgr. Запустите команду статуса кластера с фильтром строк, связанный со статусов сервиса mgr:

```
ceph -s | grep 'mgr:'
```

Вывод будет выглядеть примерно так:

```
mgr: ceph-node-1(active, since 2m)
```

Готово, сервис запущен.

Запустите сервис mgr и в остальных узлах, они будут работать в режиме standby. Их настройка идентична.

Добавление диска в кластер

После настройки компонентов управления Ceph необходимо добавить блочные устройства.

Для этого в физическом узле с дисками необходимо установить компоненты Ceph OSD:

```
apt -y install ceph-osd
```

Откройте файл/etc/ceph/ceph.confи скопируйте конфигурацию из контейнера мониторов как есть. Сохраните файл.

Отформатируйте дисковое устройство:

```
ceph-volume raw prepare --bluestore --data /dev/mapper/system-ceph
```

Произведите активацию диска:

```
/usr/sbin/ceph-volume activate --osd-id 0
```

Где 0 — это порядковый номер диска. Для каждого добавленного диска ID нужно увеличивать на 1.

После активации диска в статусе кластера должна обновиться информация о доступных устройствах OSD:

```
  cluster:

    id:     8a8cd0a4-1680-43f6-a084-fd411a122d16

    health: HEALTH_OK



  services:

    mon: 2 daemons, quorum ceph-node-1,ceph-node-2,ceph-node-3
(age 6s)

    mgr: ceph-node-1(active, since 2m), standbys: ceph-node-2,
ceph-node-3

    osd: 1 osds: 1 up, 1 in



  data:
```

```
    pools:   0 pools, 0 pgs

    objects: 0 objects, 0 B

    usage:   0 B used, 0 B / 0 B avail

    pgs:
```

Предоставление доступа к системе хранения Ceph вычислительным узлам

Для предоставления доступа к пулам Ceph с вычислительных узлов необходимо выполнить следующие шаги:

```
apt -y install ceph-common
```

Откройте файл /etc/ceph/ceph.conf и перенесите конфигурацию Ceph с контейнера компонентов управления.

Так же с контейнеров управления скопируйте ключ по пути /etc/ceph/ceph.client.admin.keyring и по тому же пути поместите в вычилистельных узлах.