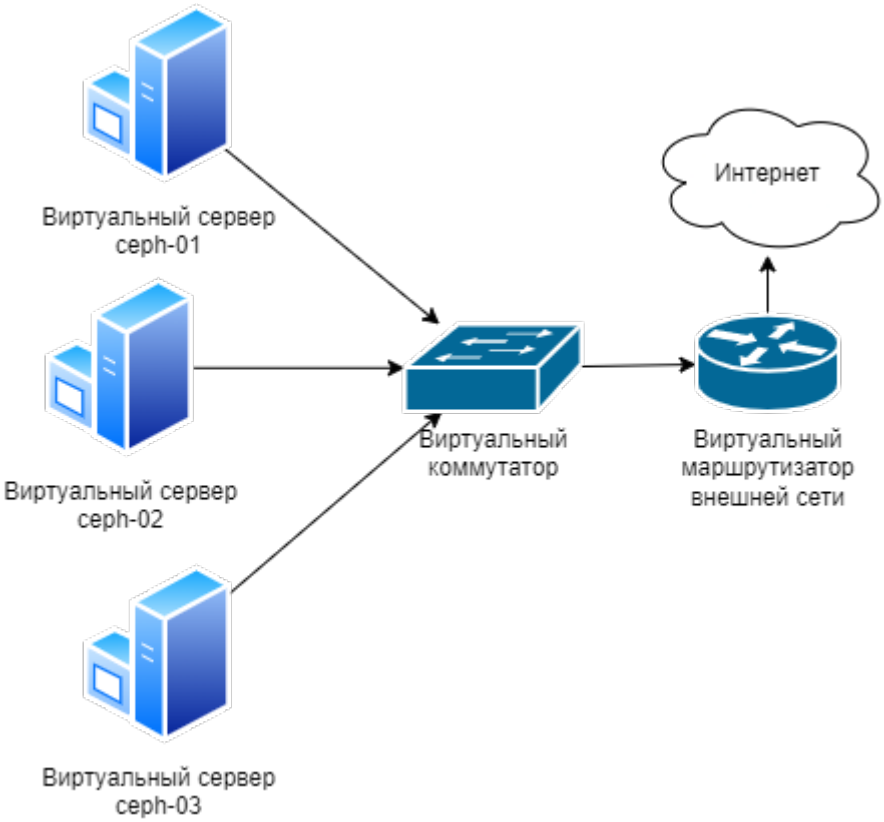
ПВСС - подходы к виртуализации сетевых функций

- Лабораторная работа 1. Запуск кластера Ceph
- Лабораторная работа 2. Установка клиентской ВМ и настройка динамической миграции на базе Corosync/Pacemaker
- Лабораторная работа 3. Ceph FS, Ceph Dashboard

Лабораторная работа 1. Запуск кластера Ceph

Схема виртуального лабораторного стенда



Для работы с облачной платформой необходимо прочитать инструкцию.

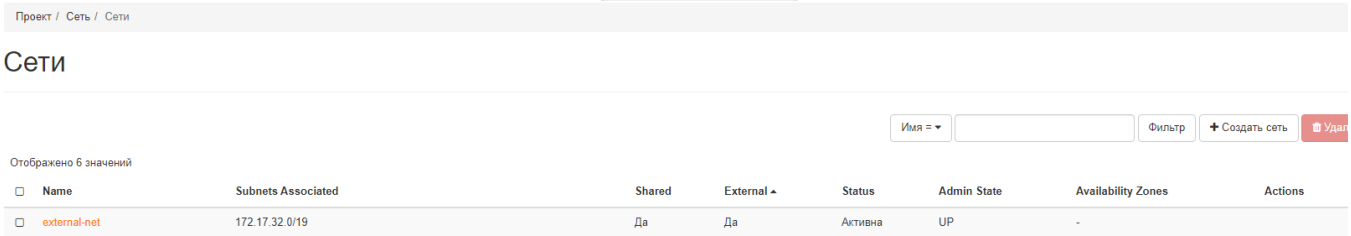
Переключиться на проект [GROUP]:[team]-lab:sandbox.

Подготовка системы

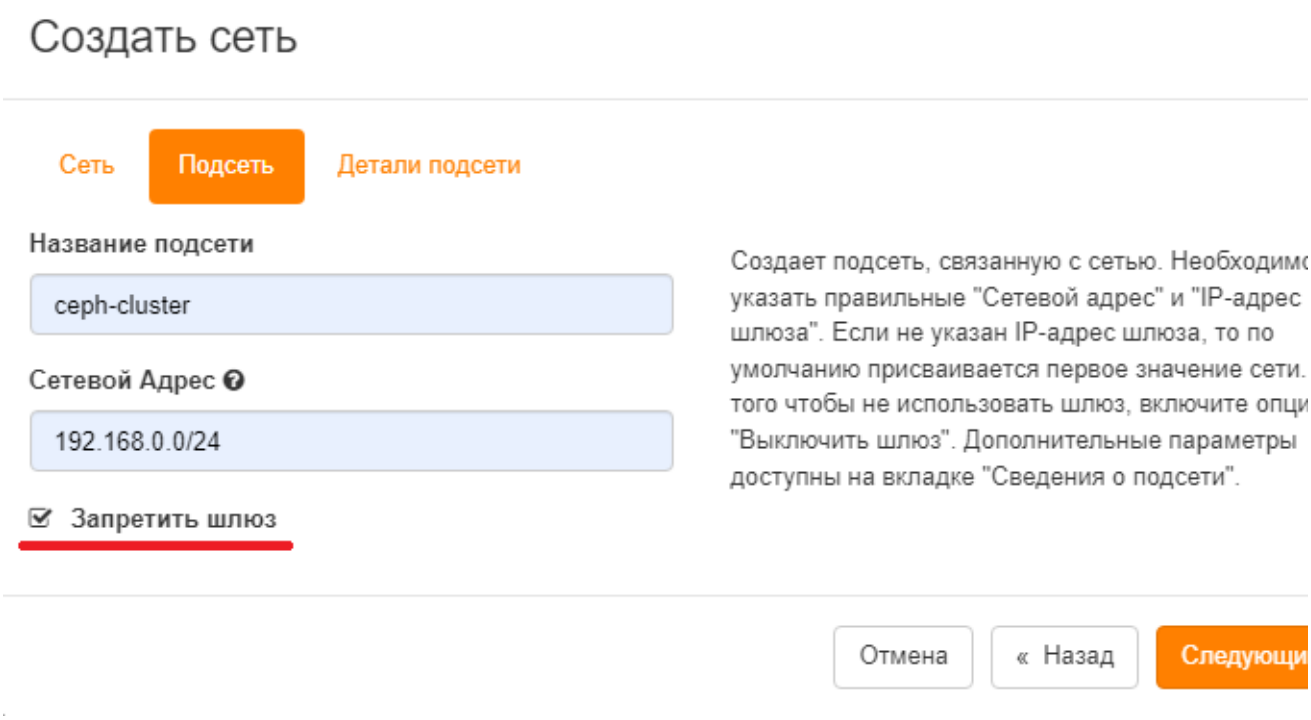
Запуск сети

Создание и настройка маршрутизатора

1. Необходимо перейти в раздел `Сеть`, в пункте `Сети`, у вас должна быть доступна сеть external-net, вторую сеть необходимо создать самостоятельно. Название для сети выбираем произвольно, например `my-internal-net`.



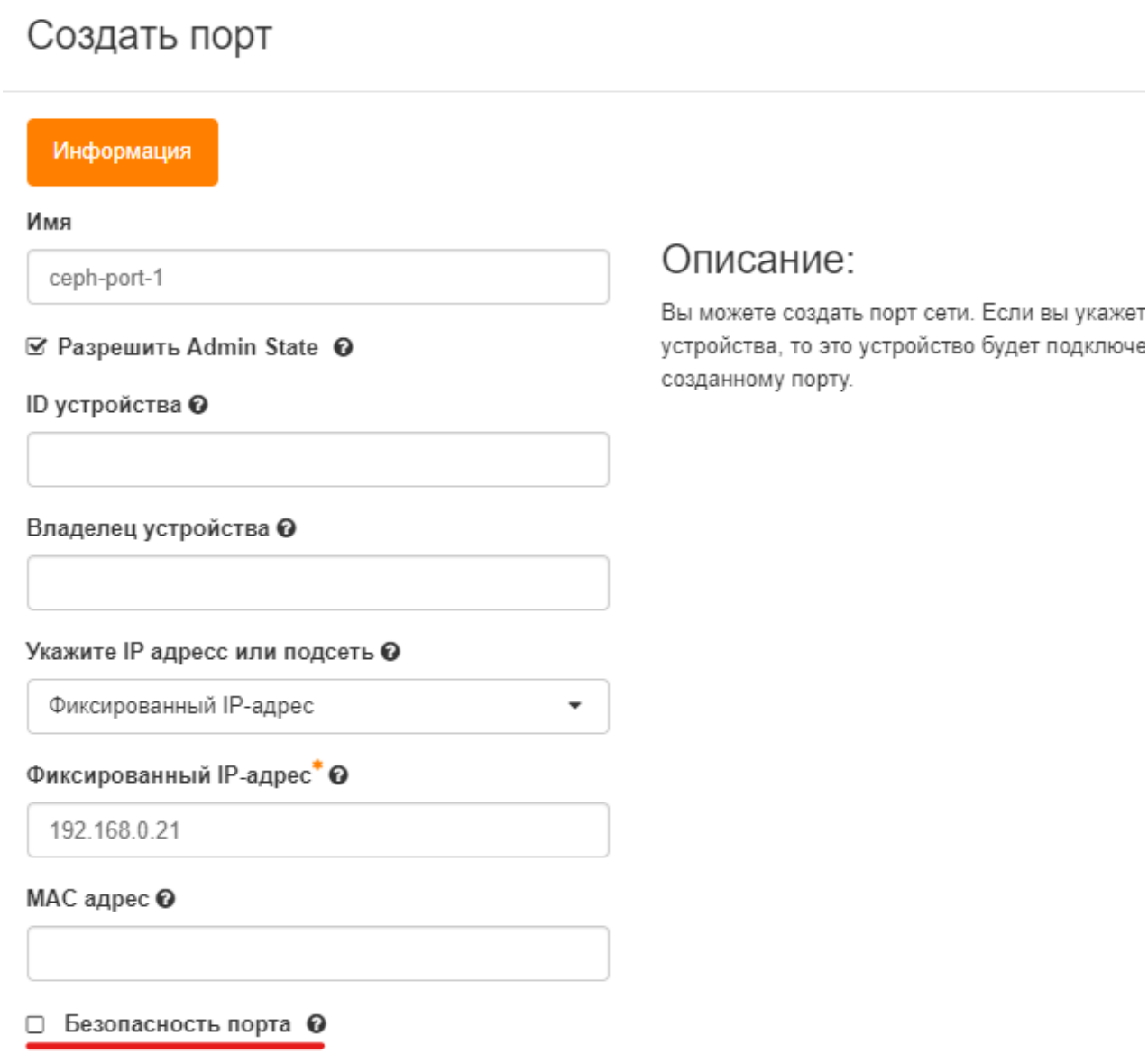
2. Создаем подсеть в этой сети аналогично скриншоту ниже:
   - В Деталях подсети отключаем DHCP



Создание и настройка портов для инстансов

1. Заходим в созданную ранее сеть -> `Порты`:
   - Нажимаем `Создать порт`
   - Заполняем поля формы:
     - Имя: ceph-port-1
     - IP адрес или подсеть: Фиксированный IP-адрес
     - Фиксированный IP-адрес: 192.168.0.21

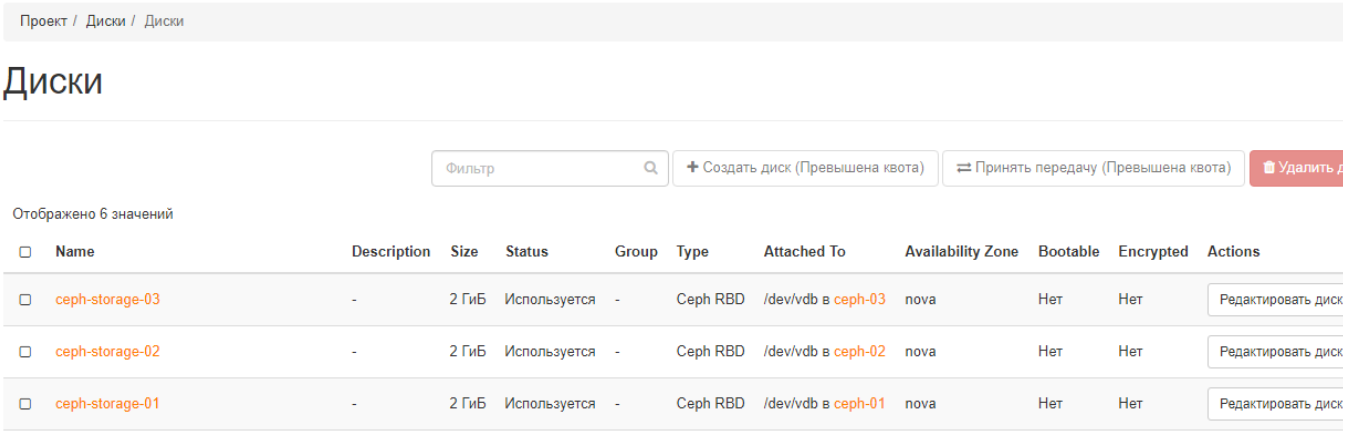
- Выключить Безопасность порта



- Аналогично создаём порты ceph-port-2 и ceph-port-3 с адресами 192.168.0.22 и 192.168.0.23 соответственно

Создание инстансов

1. Запустить 3 инстанса с характеристиками:
   - *Имена инстансов*: **ceph-01**, **ceph-02**, **ceph-03**;
   - *Источник*: Образ **Ubuntu-server-20.04**;
   - *Тип инстанса*: **small**;
   - *Сети*: **external_net**;
   - *Сетевые порты*: **port-ceph-1**, **port-ceph-2**, **port-ceph-3** для каждого инстанса соответственно.
2. Создать 3 диска по 2 Гб, подключить по одному диску на каждый инстанс. Список подключенных дисков может иметь следующий вид:



3. На каждом инстансе добавить записи в файл /etc/hosts:

```
192.168.0.21 ceph-01
192.168.0.22 ceph-02
192.168.0.23 ceph-03

<ext-ceph-01 ip> ext-ceph-01
<ext-ceph-02 ip> ext-ceph-02
<ext-ceph-03 ip> ext-ceph-03
```

4. Создать пару ключей на ceph-01 для соединения между инстансами и скопировать публичный ключ ceph-01 на каждый инстанс:

```
ssh-keygen
ssh-copy-id ceph-01
ssh-copy-id ceph-02
ssh-copy-id ceph-03
```

Запуск кластера Ceph

Установка Ceph

На данном этапе необходимо установить пакет `ceph` на каждый узел.
Для этого нужно предварительно добавить ключ безопасности репозитория в список доверенных и добавить репозиторий для скачивания указанных пакетов.

1. Для этого на каждом узле выполнить следующие команды:

```
sudo apt update
sudo apt install -y ca-certificates
wget -q -O- 'https://download.ceph.com/keys/release.asc' | sudo apt-key add -
sudo apt-add-repository 'deb https://download.ceph.com/debian-quincy/ focal main'
sudo apt update && sudo apt install ceph ceph-mds -y
```

- Либо использовать скрипт:

Скрипт для запуска на ceph-01

```
#!/bin/bash


for NODE in ceph-01 ceph-02 ceph-03
do
  ssh $NODE \
    "sudo apt update; \
    sudo apt install -y ca-certificates; \
    wget -q -O- 'https://download.ceph.com/keys/release.asc' | sudo apt-key add -; \
    sudo apt-add-repository 'deb https://download.ceph.com/debian-quincy/ focal main'; \
    sudo apt update && sudo apt install ceph ceph-mds -y"
done
```

2. Проверить статус установки можно с помощью команды:

```
sudo ceph -v
```

Запуск демонов ceph-mon, ceph-mgr, ceph-osd, ceph-mds

Подготовка к запуску демонов

На узле ceph-01 необходимо выполнить следующие действия:

1. Получить уникальный идентификатор для кластера (можно с помощью команды `uuidgen`):
   `ad56ab6d-7f7a-4ee4-8b02-9c7ef1ddb438` - используется в качетсве примера
2. Создать файл конфигурации для кластера по пути `/etc/ceph/ceph.conf`. Необходимо вставить свои значения параметров: fsid, адреса интерфейсов внешней сети.
   `vi /etc/ceph/ceph.conf`:

```
[global]
  fsid = ad56ab6d-7f7a-4ee4-8b02-9c7ef1ddb438
  mon initial members = ceph-01, ceph-02, ceph-03
  mon host = ext-ceph-01, ext-ceph-02, ext-ceph-03
  cluster network = 192.168.0.0/24
  public network = 172.17.32.0/19
  auth cluster required = cephx
  auth service required = cephx
  auth client required = cephx
```

```
osd pool default size = 3
osd pool default min size = 2
osd pool default pg num = 64
osd pool default pgp num = 64
osd crush chooseleaf type = 1
```

- Разберём параметры конфигурации:
  - `fsid` - идентификатор кластера
  - `mon initial members` - имена узлов стартовых мониторов
  - `mon host` - адреса стартовых мониторов
  - `cluster network` - сеть кластера в формате CIDR, в которой будут передаваться heartbeat-сигналы, реплики данных между демонами OSD
  - `public network` - внешняя сеть для взаимодействия с клиентами кластера
  - `auth cluster/service/client required` - включение авторизации между демонами кластера и подключаемымиклиентами
  - `osd pool default size` - количество реплик по умолчанию
  - `osd pool default min size` - минимальное количество записанных реплик для объектов, чтобы подтвердить операцию ввода-вывода клиенту
  - `osd pool default pg num` - количество групп размещения по умолчанию для пула
  - `osd pool default pgp num` - количество групп размещения по умолчанию для размещения пула. PG и PGP должны быть равны
  - `osd crush chooseleaf type` - максимальный уровень размещения кластера (при значении больше 1 имеется возможность создания групп узлов, таким образом можно разделить иерархию узлов на стойки, ЦОДы и другие физически разнесённые группы)

3. Создать связку ключей для своего кластера и сгенерировать секретный ключ монитора. Также создать связку ключей администратора, `client.admin` пользователя и добавить пользователя в связку ключей. Создать связку ключей `bootstrap-osd`, `client.bootstrap-osd` пользователя и добавить пользователя в связку ключей:

```
sudo ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow *'
sudo ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring \
 --gen-key -n client.admin --cap mon 'allow *' --cap osd 'allow *' \
 --cap mds 'allow *' --cap mgr 'allow *'
sudo ceph-authtool --create-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring \
 --gen-key -n client.bootstrap-osd --cap mon 'profile bootstrap-osd' \
 --cap mgr 'allow r'
```

4. Добавить сгенерированные ключи в связку `ceph.mon.keyring`:

```
sudo ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring
sudo ceph-authtool /tmp/ceph.mon.keyring --import-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring
```

5. Сгенерировать карту мониторов, используя имена узлов, внешние адреса (записи из файла `/etc/hosts` здесь не работают) и FSID. Сохранить как /tmp/monmap:

```
monmaptool --create --fsid ad56ab6d-7f7a-4ee4-8b02-9c7ef1ddb438 \
 --add ceph-01 <ceph-01 ext-ip> --add ceph-02 <ceph-02 ext-ip> \
 --add ceph-03 <ceph-03 ext-ip> /tmp/monmap
```

Запуск ceph-mon

1. Необходимо скопировать файлы `/etc/ceph/ceph.conf`, `/etc/ceph/ceph.client.admin.keyring`, `/tmp/monmap`, `/tmp/ceph.mon.keyring`, `/var/lib/ceph/bootstrap-osd/ceph.keyring` с ceph-01 на ceph-02 и ceph-03.
   - Примеры выполнения данного пункта:

Пошаговые действия с описанием

1. Создадим папку `/tmp/ceph_files` и скопируем в неё все необходимые файлы:

```
mkdir /tmp/ceph_files
sudo cp /etc/ceph/ceph.conf /tmp/ceph_files/ceph.conf
sudo cp /etc/ceph/ceph.client.admin.keyring
/tmp/ceph_files/ceph.client.admin.keyring
sudo cp /tmp/monmap /tmp/ceph_files/monmap
sudo cp /tmp/ceph.mon.keyring /tmp/ceph_files/ceph.mon.keyring
sudo cp /var/lib/ceph/bootstrap-osd/ceph.keyring /tmp/ceph_files/ceph.keyring
```

2. Далее добавим общие права на чтение для всех файлов в папке и отправим эти файлы на ceph-02 и ceph-03. Удалим созданную ранее папку:

```
sudo chmod 644 /tmp/ceph_files/*
scp -r /tmp/ceph_files/ ceph-02:~
scp -r /tmp/ceph_files ceph-03:~
sudo rm -rf /tmp/ceph_files
```

3. На ceph-02 и ceph-03 выполняем следующие команды (возвращаем файлам их исходные права, копируем их по нужным путям и удаляем полученные файлы:

```
cd ~/ceph_files
sudo chmod 600 ceph.client.admin.keyring ceph.mon.keyring ceph.keyring
sudo cp ceph.conf /etc/ceph/ceph.conf
```

```
sudo cp ceph.client.admin.keyring /etc/ceph/ceph.client.admin.keyring
sudo cp monmap /tmp/monmap
sudo cp ceph.mon.keyring /tmp/ceph.mon.keyring
sudo cp ceph.keyring /var/lib/ceph/bootstrap-osd/ceph.keyring
cd ..
rm -rf ceph_files
```

Скрипт для запуска на ceph-01

```
#!/bin/bash

mkdir /tmp/ceph_files
sudo cp /etc/ceph/ceph.conf /tmp/ceph_files/ceph.conf
sudo cp /etc/ceph/ceph.client.admin.keyring /tmp/ceph_files/ceph.client.admin.keyring
sudo cp /tmp/monmap /tmp/ceph_files/monmap
sudo cp /tmp/ceph.mon.keyring /tmp/ceph_files/ceph.mon.keyring
sudo cp /var/lib/ceph/bootstrap-osd/ceph.keyring /tmp/ceph_files/ceph.keyring

sudo chmod 644 /tmp/ceph_files/*
scp -r /tmp/ceph_files/ ceph-02:~
scp -r /tmp/ceph_files ceph-03:~
sudo rm -rf /tmp/ceph_files

for NODE in ceph-02 ceph-03
do
  ssh $NODE \
    "cd ~/ceph_files; \
    sudo chmod 600 ceph.client.admin.keyring ceph.mon.keyring ceph.keyring; \
    sudo cp ceph.conf /etc/ceph/ceph.conf; \
    sudo cp ceph.client.admin.keyring /etc/ceph/ceph.client.admin.keyring; \
    sudo cp monmap /tmp/monmap; \
    sudo cp ceph.mon.keyring /tmp/ceph.mon.keyring; \
    sudo cp ceph.keyring /var/lib/ceph/bootstrap-osd/ceph.keyring; \
    cd ..; \
    rm -rf ceph_files"
done
```

2. Далее создадим каталог для данных мониторов на узлах. Заполним демоны монитора картой мониторов и набором ключей. Это можно сделать с помощью следующих команд на каждом хосте:

```
sudo chown ceph:ceph /tmp/ceph.mon.keyring
sudo -u ceph mkdir /var/lib/ceph/mon/ceph-$HOSTNAME
sudo -u ceph ceph-mon --mkfs -i $HOSTNAME \
 --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring
sudo systemctl enable ceph-mon@$HOSTNAME --now
```

   - Либо использовать скрипт:

Скрипт для запуска на ceph-01

```
#!/bin/bash

for NODE in ceph-01 ceph-02 ceph-03
do
  ssh $NODE \
  "sudo chown ceph:ceph /tmp/ceph.mon.keyring; \
  sudo -u ceph mkdir /var/lib/ceph/mon/ceph-$NODE; \
  sudo -u ceph ceph-mon --mkfs -i $NODE \
   --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring; \
  sudo systemctl enable ceph-mon@$NODE --now"
done
```

3. Включим протокол `MESSENGER V2`, отключим небезопасное использование клиентами глобальных идентификаторов и установим вывод предупреждений о малом свободном пространстве на основном диске хоста при остакте в 10% (иначе без этих параметров кластер может иметь статус HEALTH_WARN). Также установим параметр, позволяющий удалять пулы, в значение `true`:

```
sudo ceph mon enable-msgr2
sudo ceph config set mon auth_allow_insecure_global_id_reclaim false
sudo ceph config set mon mon_data_avail_warn 10
sudo ceph config set mon mon_allow_pool_delete true
```

4. Так как ещё не добавлено никаких устройств хранения, то в выводе состояния нас интересует только образовавшийся кворум мониторов:

```
sudo ceph -s
```

Запуск ceph-mgr

На каждом узле, где вы запускаете демон ceph-mon, вы также должны настроить демон ceph-mgr. Сначала создадим ключ аутентификации для демона, поместим этот ключ в путь `/var/lib/ceph/mgr/ceph-$HOSTNAME/keyring` и запустим демон ceph-mgr.

1. Для этого небоходимо выполнить данные действия на каждом хосте:

```
sudo ceph auth get-or-create mgr.$HOSTNAME mon 'allow profile mgr' osd 'allow *' mds 'allow *'
sudo ceph auth get-or-create mgr.$HOSTNAME > /tmp/mgr.admin.keyring
sudo -u ceph mkdir /var/lib/ceph/mgr/ceph-$HOSTNAME
sudo cp /tmp/mgr.admin.keyring /var/lib/ceph/mgr/ceph-$HOSTNAME/keyring
sudo chown -R ceph. /var/lib/ceph/mgr/ceph-$HOSTNAME
sudo systemctl enable --now ceph-mgr@$HOSTNAME
```

   - Либо использовать скрипт:

Скрипт для запуска на ceph-01

```
#!/bin/bash


for NODE in ceph-01 ceph-02 ceph-03
do
  MGR_PATH=/var/lib/ceph/mgr/ceph-$NODE;
  ssh $NODE \
   "sudo ceph auth get-or-create mgr.$NODE \
    mon 'allow profile mgr' osd 'allow *' mds 'allow *'; \
   sudo ceph auth get-or-create mgr.$NODE > /tmp/mgr.admin.keyring; \
   sudo -u ceph mkdir $MGR_PATH; \
   sudo cp /tmp/mgr.admin.keyring $MGR_PATH/keyring; \
   sudo chown -R ceph. $MGR_PATH; \
   sudo systemctl enable --now ceph-mgr@$NODE"
done
```

2. На данном этапе мы можем увидеть статус запуска демонов менеджеров:

```
sudo ceph -s
```

Запуск ceph-osd

Ceph предоставляет `ceph-volume` утилиту, которая может подготовить логический том, диск или раздел для использования с Ceph. Утилита `ceph-volume` создает идентификатор OSD путем увеличения индекса. Кроме того, `ceph-volume` добавит новый OSD в карту CRUSH хоста.

1. На каждом хосте выполнить:

```
sudo ceph-volume lvm create --data /dev/vdb
```

- Либо использовать скрипт:

Скрипт для запуска на ceph-01

```
#!/bin/bash


for NODE in ceph-01 ceph-02 ceph-03
do
    ssh $NODE "sudo ceph-volume lvm create --data /dev/vdb"
done
```

2. На данном этапе кластер должен иметь статус HEALTH_OK:

```
sudo ceph -s
```

3. Также подключенные OSD можно получить с помощью команды:

```
sudo ceph osd tree
```

Запуск услуг кластера

Создание пула и блочного устройства

1. Создадим пул `rbd` и инициализируем его:

```
sudo ceph osd pool create rbd 32
sudo rbd pool init rbd
```

2. Получить список созданных пулов с помощью команды:

```
sudo ceph osd pool ls
```

3. Создать блочное устройство в новом пуле:

```
sudo rbd create --size 10G --pool rbd rbd01
```

Проверка

1. Подключаем созданное устройство к ceph-01:

```
sudo rbd map rbd01
```

2. Получить список подлючённых устройств можно с помощью команды:

```
sudo rbd showmapped
```

3. Установим файловую систему на подключенное устройство и примонтируем его к каталогу `/mnt/ceph_rbd`:

```
sudo mkfs.xfs /dev/rbd0
sudo mkdir /mnt/ceph_rbd
sudo mount /dev/rbd0 /mnt/ceph_rbd/
```

4. На ceph-01 создать файл с произвольным текстом и поместить его в каталог `/mnt/ceph_rbd`. Затем отмонтировать блочное устройство:

```
echo "my text for ceph rbd" | tee /mnt/ceph_rbd/rbd.txt
sudo umount /dev/rbd0
sudo rbd unmap /dev/rbd/rbd/rbd01
```

5. На ceph-02 подключить использованное ранее блочное устройство и примонтировать его к каталогу `/mnt/ceph_rbd`:

```
sudo rbd map rbd01
sudo mkdir /mnt/ceph_rbd
sudo mount /dev/rbd0 /mnt/ceph_rbd/
```

6. Проверить содержимое подключенного блочного устройства:

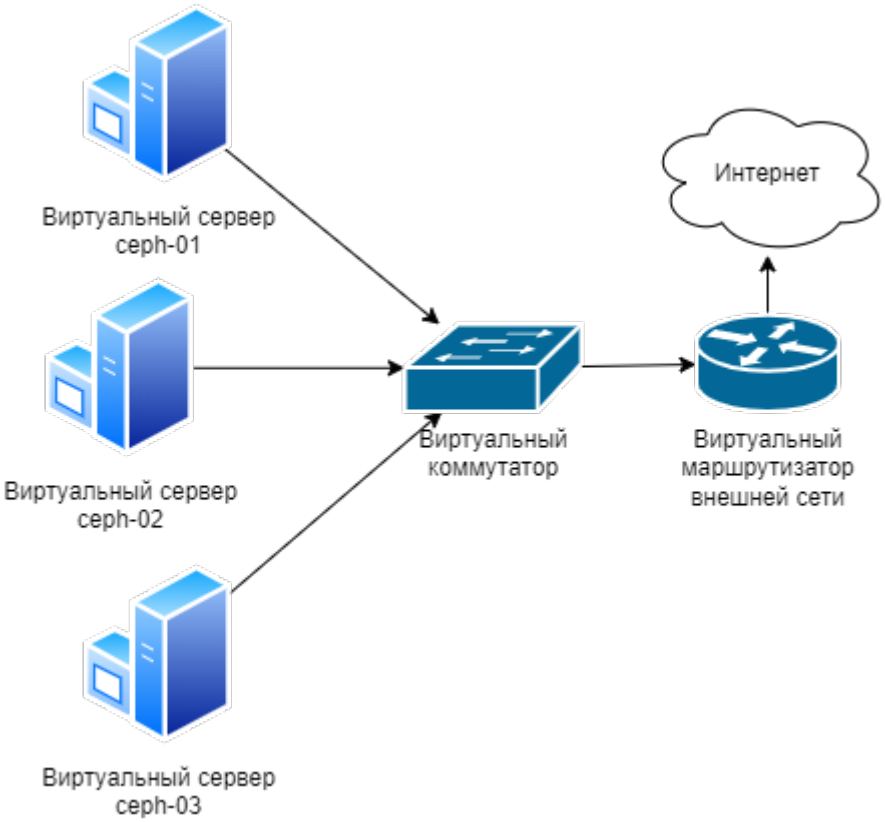
```
ls /mnt/ceph_rbd/
cat /mnt/ceph_rbd/rbd.txt
```

7. Отмонтировать блочное устройство и удалить его, удалить пул:

```
sudo umount /dev/rbd0
sudo rbd unmap /dev/rbd/rbd/rbd01
sudo rbd rm rbd01 -p rbd
sudo ceph osd pool delete rbd rbd --yes-i-really-really-mean-it
```

Лабораторная работа 2. Установка клиентской ВМ и настройка динамической миграции на базе Corosync/Pacemaker

Схема виртуального лабораторного стенда



Запуск ВМ с использованием Ceph RBD

Интеграция Ceph и libvirt

Установка libvirt

```
sudo apt install -y qemu-kvm virtinst libvirt-clients libvirt-daemon-system
```

Добавление пула Ceph в libvirt

1. Во-первых, нам нужно создать пул Ceph OSD специально для использования хранилища kvm, qemu, libvirt:

```
sudo ceph osd pool create libvirt-pool 64 64
sudo rbd pool init libvirt-pool
```

2. Во-вторых, нам нужен пользователь Ceph для манипулирования только что созданным пулом.

```
sudo ceph auth get-or-create "client.libvirt" mon "profile rbd" osd "profile rbd pool=libvirt-pool"
```

> **Выполнить пункты 3-6 на каждом узле**

3. Нам нужно добавить файл с секретом libvirt для аутентификации.
`uuidgen` -> `e6ca4cff-bf4f-4444-9089-8bd1304b3500` - для libvirt секрета, использовать один для всех хостов
`sudo vi /tmp/libvirt-secret.xml`:

```
<secret ephemeral='no' private='no'>
  <uuid>e6ca4cff-bf4f-4444-9089-8bd1304b3500</uuid>
  <usage type='ceph'>
    <name>client.libvirt secret</name>
  </usage>
</secret>
```

4. Встраиваем ключ авторизации пользователя Ceph в файл с секретом с указанным uuid.

```
sudo virsh secret-define --file "/tmp/libvirt-secret.xml"
sudo rm -f "/tmp/libvirt-secret.xml"
sudo virsh secret-set-value --secret "e6ca4cff-bf4f-4444-9089-8bd1304b3500" \
 --base64 "$(sudo ceph auth get-key client.libvirt)"
```

5. Нам также необходимо определить пул хранения RBD в libvirt. Сначала нам нужно создать файл определения пула хранения (вставить свои адреса внешних интерфейсов в теги **host**):
`vi /tmp/libvirt-rbd-pool.xml`:

```
<pool type="rbd">
  <name>libvirt-pool</name>
  <source>
    <name>libvirt-pool</name>
    <host name='ext-ceph-01' port='6789' />
    <host name='ext-ceph-02' port='6789' />
    <host name='ext-ceph-03' port='6789' />
    <auth username='libvirt' type='ceph'>
      <secret uuid='e6ca4cff-bf4f-4444-9089-8bd1304b3500'/>
    </auth>
  </source>
</pool>
```

6. Теперь мы можем определить и запустить пул.

```
sudo virsh pool-define "/tmp/libvirt-rbd-pool.xml"
sudo rm -f "/tmp/libvirt-rbd-pool.xml"
sudo virsh pool-autostart "libvirt-pool"
sudo virsh pool-start "libvirt-pool"
```

- Проверить доступность созданного пула: `sudo virsh pool-list`

Запуск ВМ

1. Скачать образ `cirros` и скопировать его по пути `/tmp/cirros.img`

```
wget http://download.cirros-cloud.net/0.5.1/cirros-0.5.1-x86_64-disk.img
sudo cp cirros-0.5.1-x86_64-disk.img /tmp/cirros.img
```

2. Создать Ceph RBD для виртуальной машины:

```
sudo qemu-img convert -p -t none /tmp/cirros.img rbd:libvirt-pool/cirros
sudo virsh pool-refresh libvirt-pool
```

- Проверить доступность созданного образа: `sudo virsh vol-list libvirt-pool`

3. Создать ВМ:

```
sudo virt-install --name Test-VM --graphics none \
  --vcpus 1 --memory 128 --disk "vol=libvirt-pool/cirros" \
  --import --autostart
```

4. Проверить работоспособность запущенной ВМ:

```
sudo virsh list --all
sudo virsh console Test-VM
```

Настройка динамической миграции

Установка Pacemaker/Corosync

- На всех узлах нужно установить требуемые пакеты и запустить демон `pcsd`:

```
sudo apt install pacemaker corosync pcs resource-agents
sudo systemctl enable --now pcsd
```

Запуск кластера

1. Задать пароль пользователя `hacluster` на всех узлах (в примере используется пароль `password`):

```
sudo passwd hacluster
```

2. Отредактировать раздел `nodelist` файла `/etc/corosync/corosync.conf` (на одном из узлов):

```
nodelist {
  node {
        name: ceph-01
        nodeid: 1
        ring0_addr: ceph-01
  }
  node {
        name: ceph-02
        nodeid: 2
        ring0_addr: ceph-02
  }
  node {
        name: ceph-03
        nodeid: 3
        ring0_addr: ceph-03
  }
}
```

3. С помощью pcs создать кластер (на одном из узлов):

```
sudo pcs host auth ceph-01 ceph-02 ceph-03 -u hacluster -p password
sudo pcs cluster setup newcluster ceph-01 ceph-02 ceph-03 --force
sudo pcs cluster start --all
```

4. Отключить fencing (в рамках работы он не рассматривается):

```
sudo pcs property set stonith-enabled=false
```

5. Включить автозапуск сервисов на всех трех машинах:

```
sudo systemctl enable pacemaker corosync
```

6. Просмотреть информацию о кластере и кворуме:

```
sudo pcs status
sudo corosync-quorumtool
```

Создание ресурса

1. Создать дамп созданной ВМ и затем удалить её:

```
sudo virsh dumpxml Test-VM > vm_conf.xml
sudo virsh undefine Test-VM
sudo virsh destroy Test-VM
```

2. Скопировать дамп с ceph-01 на ceph-02 и ceph-03:

```
scp vm_conf.xml ceph-02:~
scp vm_conf.xml ceph-03:~
```

3. На ceph-01, ceph-02 и ceph-03 переместить файл в /etc/pacemaker/

```
sudo mv vm_conf.xml /etc/pacemaker/
sudo chown hacluster:haclient /etc/pacemaker/vm_conf.xml
```

4. Теперь добавить сам ресурс:

```
sudo pcs resource create test-vm VirtualDomain \
config="/etc/pacemaker/vm_conf.xml" \
migration_transport=tcp meta allow-migrate=true
```

5. Просмотреть список добавленных ресурсов:

```
sudo pcs status
sudo pcs resource config test-vm
```

6. Проверить список виртуальных машин на узле, на котором запустился ресурс:

```
sudo virsh list --all
```

7. Проверить, что ресурс успешно запустился.

Настройка миграции

> **Выполнить действия на всех узлах**

1. Необходимо отредактировать файл `/etc/libvirt/libvirtd.conf`:

```
listen_tls = 0
listen_tcp = 1
auth_tcp = "none"
```

2. Отредактировать файл `/etc/default/libvirtd`:

```
libvirtd_opts="--config /etc/libvirt/libvirtd.conf"
```

3. Запустить сокет `libvirt-tcp`:

```
sudo systemctl stop libvirtd && sudo systemctl start libvirtd-tcp.socket
```

Миграция ресурса

1. Нужно переместить ресурс на `ceph-02`:

```
sudo pcs resource move test-vm ceph-02
```

2. На ceph-02 посмотреть статус кластера, и проверить список запущенных гостевых машин можно следующими командами:

```
sudo pcs status
sudo virsh list --all
```

3. Команда move добавляет ресурсу правило, заставляющее его запускаться только на указанном узле. Для того, чтобы очистить все добавленные ограничения - clear:

```
sudo pcs resource clear test-vm
```

Лабораторная работа 3. Ceph FS, Ceph Dashboard

Использование файловой системы Ceph

Запуск ceph-mds

Чтобы услуга Ceph FS работала для клиентов, необходимо запустить демон сервера метаданных (MDS). Для этого создадим папку для демона, пользователя Ceph и запустим службу ceph-mds.

1. Запустить демоны MDS на хостах ceph-02, ceph-03:

```
sudo mkdir -p /var/lib/ceph/mds/ceph-$HOSTNAME
sudo ceph-authtool --create-keyring /var/lib/ceph/mds/ceph-$HOSTNAME/keyring \
 --gen-key -n mds.$HOSTNAME
sudo chown -R ceph. /var/lib/ceph/mds/ceph-$HOSTNAME
sudo ceph auth add mds.$HOSTNAME osd "allow rwx" mds "allow" mon "allow profile mds" \
 -i /var/lib/ceph/mds/ceph-$HOSTNAME/keyring
sudo systemctl enable --now ceph-mds@$HOSTNAME
```

- Либо использовать скрипт:

Скрипт для запуска на ceph-01

```
#!/bin/bash


for NODE in ceph-02 ceph-03
do
  MDS_PATH=/var/lib/ceph/mds/ceph-$NODE;
  ssh $NODE \
    "sudo mkdir -p $MDS_PATH; \
    sudo ceph-authtool --create-keyring $MDS_PATH/keyring \
     --gen-key -n mds.$NODE; \
    sudo chown -R ceph. $MDS_PATH; \
    sudo ceph auth add mds.$NODE osd 'allow rwx' mds 'allow' mon 'allow profile mds' \
```

```
    -i $MDS_PATH/keyring; \
    sudo systemctl enable --now ceph-mds@$NODE"
done
```

Создание файловой системы

На любом узле кластера необходимо выполнить следующие команды:

1. Чтобы запустить файловую систему ceph, нужно создать два пула: пул для данных и пул для метаданных:

   ```
   sudo ceph osd pool create cephfs_data 64
   sudo ceph osd pool create cephfs_metadata 64
   ```

2. Создадим файловую систему:

   ```
   sudo ceph fs new cephfs cephfs_metadata cephfs_data
   ```

   С помощью данных команд можно просмотреть статус файловой системы:

   ```
   sudo ceph fs ls
   sudo ceph mds stat
   sudo ceph fs status cephfs
   ```

3. Создадим пользователя для подключения к файловой системе и запишем ключ пользователя в отдельный файл:

   ```
   sudo ceph fs authorize cephfs client.fsclient / rw -o /etc/ceph/ceph.client.fsclient.keyring
   sudo ceph auth get-or-create-key client.fsclient -o /etc/ceph/fsclient.secret
   ```

4. Необходимо скопировать файлы `/etc/ceph/fsclient.secret` и `/etc/ceph/ceph.client.fsclient.keyring` на остальные узлы.

Проверка

1. На каждом узле кластера необходимо выполнить следующие команды:

   ```
   sudo mkdir /mnt/cephfs
   sudo mount -t ceph fsclient@.cephfs=/ /mnt/cephfs -o secretfile=/etc/ceph/fsclient.secret
   ```

2. Проверить статус монтирования:

   ```
   df -hT | grep ceph
   ```

3. На ceph-01 создать файл с произвольным текстом и поместить его в каталог `/mnt/cephfs`:

```
echo "my text for ceph fs" | tee /mnt/cephfs/test.txt
```

4. Проверить содержание файла на других узлах:

```
ls /mnt/cephfs/
cat /mnt/cephfs/test.txt
```

Динамическая миграция ВМ с использованием Ceph FS

1. Поместим образ `cirros` в созданную файловую систему:

```
sudo cp /tmp/cirros.img /mnt/cephfs/cirros.img
```

2. Запустим ВМ:

```
sudo virt-install --name cirros --graphics none \
 --vcpus 1 --memory 128 --disk "vol=libvirt-pool/cirros" \
 --import --autostart
```

3. Проверим работоспособность запущенной ВМ:

```
sudo virsh list --all
sudo virsh console cirros
```

4. Дальнейшие действия аналогичны пунктам "Создание ресурса", "Миграция ресурса" лабораторной работы 2.

Запуск Ceph Dashboard

1. В работающем кластере Ceph панель мониторинга Ceph активируется с помощью:

```
sudo ceph mgr module enable dashboard
```

2. Сгенерируем и установим самоподписанный SSL сертификат:

```
sudo ceph dashboard create-self-signed-cert
```

3. Чтобы иметь возможность войти в систему, вам необходимо создать учетную запись пользователя и связать ее хотя бы с одной ролью. Ceph предоставляет набор предопределенных системных ролей, которые вы можете использовать. Чтобы создать пользователя `admin` с паролем `password` и ролью администратора, можно воспользоваться следующими командами:

```
echo "password" >> mypass
sudo ceph dashboard ac-user-create admin -i mypass --force-password administrator
```

```
rm mypass
```

4. В Openstack в группе безопасности добавить правило для порта 8443/tcp.
5. С помощью команды `sudo ceph mgr services` на одном из узлов кластера можно получить ссылки доступных сервисов. Пример вывода:

```
{
   "dashboard": "https://172.17.5.232:8443/"
}
```