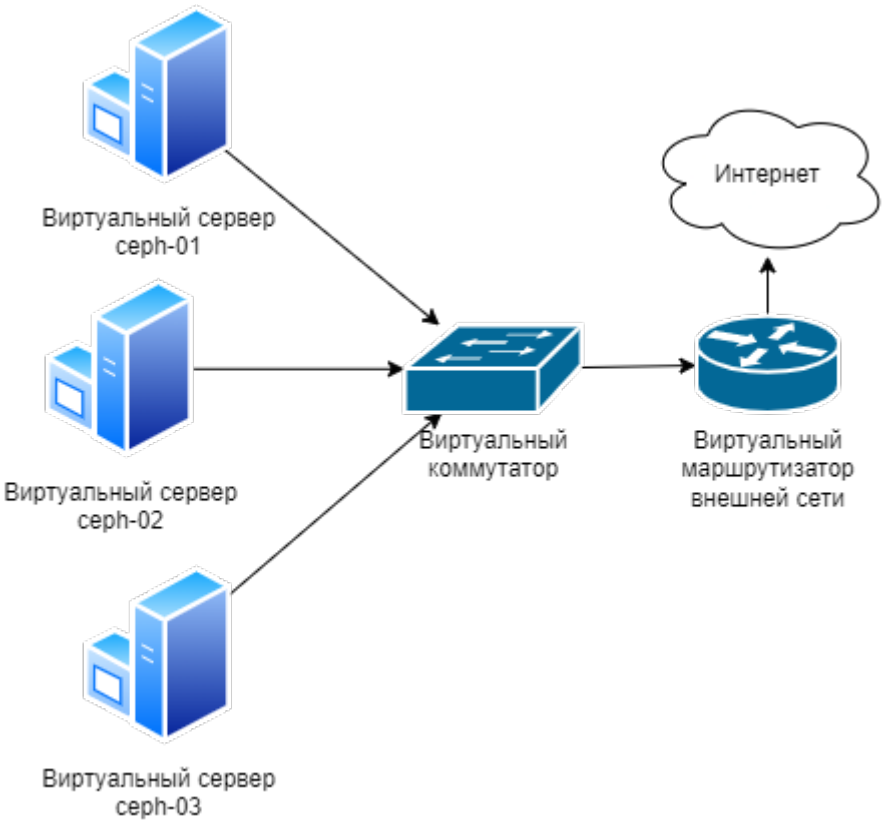
Лабораторная работа 2. Установка клиентской ВМ и настройка динамической миграции на базе Corosync/Pacemaker

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
```

Версия #19
Баев Артем Олегович создал 21 октября 2022 10:26:55
Баев Артем Олегович обновил 3 ноября 2022 15:59:22