Настройка SDS Ceph в распределенной системе oVirt 4.1

(Часть №1. NFS-Ganesha)

Всем доброго времени суток. Статья будет посвящена распределенной системе виртуализации oVirt, разработанной компанией Red Hat и ее связки с (SDS) хранилищем Ceph. Статья предназначена специалистам, умеющим как минимум выполнять развертывание системы oVirt и иметь представление о системе хранения Ceph на базовом уровне.

В статье мы не будем рассматривать причины необходимости данной связки, т.к. у каждого эти причины могут быть разными. Больше к этому вопросу возвращаться не будем.

При подготовке данной статьи много информации, конфигурационных файлов, и советов были предоставлены Алексеем Костиным. За это огромное ему человеческое спасибо. Помощь оказана огромная, без его участия процесс затянулся на продолжительное время.

И так приступим. Предположим, что у нас имеются уже три сервера на которых есть накопитель (лучше RAID) под саму систему и свободные диски для использования в Ceph. Система oVirt и Ceph будут развернуты на одних и тех - же серверах. В продуктиве рекомендуется разделить хранилище Ceph с серверами системы виртуализации oVirt, так же Ceph требует сеть не ниже 10G с ее "правильными настройками", но это тема уже для другой статьи.

И так, назовем сервера и присвоим им IP адреса следующим образом:

```
h31.testnet 192.168.120.31
h32.testnet 192.168.120.32
h33.testnet 192.168.120.33
```

Имена серверов должны быть внесены в DNS, можно дополнительно, на случай падения DNS, внести имена в файлы /etc/hosts на все сервера. Я все действия буду проводить внутри развернутого oVirt с включенным режимом Nested, который позволяет использовать вложенную виртуализацию, но это не столь важно.

Запасаемся кофе или чаем, кто к чему привык и начнем.

ЭТАП №1

(Подготовка серверов для развертывания oVirt 4.1)

Подготовим наши сервера к развертыванию системы oVirt. Я на серверах oVirt всегда выключаю iptables, что бы не загружать ресурсы самого сервера и не добавить лишнего Latency. Все необходимые правила вынесены на аппаратные маршрутизаторы кластера L3 уровня. Так же поступаю и с SELinux. Тут каждый выбирает политику сам, я в данном вопросе ничего советовать не стану.

Отключаем iptables.

(запускаем на всех серверах):

```
systemctl stop firewalld  && systemctl disable firewalld
systemctl stop iptables  && systemctl disable iptables
```

Отключаем SELinux.

(запускаем на всех серверах):

```
sed -i 's/^SELINUX=.*/SELINUX=disabled/g' /etc/selinux/config
```

Синхронизируем время серверов, это очень важно:

(запускаем на всех серверах):

```
yum -y install chrony && service chronyd restart
```

Установим пакеты для развертывания системы oVirt.

(запускаем на всех серверах):

```
yum -y install epel-release yum-utils
yum -y install https://resources.ovirt.org/pub/ovirt-4.1/rpm/el7/noarch/ovirt-release41-4.1.9-1.el7.centos.noarch.rpm
```

Официально в CentOS 7 уже не поддерживается oVirt 4.1, поэтому отключаем отсутствующие репозитарии, иначе получим ошибку при установке пакетов.

(запускаем на всех серверах):

```
    yum-config-manager --disable ovirt-4.1-centos-gluster38
    yum-config-manager --disable ovirt-centos-ovirt41
```

Устанавливаем пакет с необходимым для развертывания oVirt 4.1, остальные зависимости подтянутся сами.

(запускаем на всех серверах):

```
yum -y install ovirt-hosted-engine-setup
```

Все сервера будущего кластера готовы к развертыванию системы виртуализации oVirt 4.1. Теперь нужно подготовить наши сервера для работы с Ceph.

ЭТАП №2

(Подготовка серверов для работы с Ceph)

При написании данной статьи, сборка всех вспомогательных компонентов производилась с версией Ceph 12.2.8 (Luminous), поэтому устанавливаем на сервера именно эту версию Ceph.

(запускаем на всех серверах):

```
yum -y install https://download.ceph.com/rpm-luminous/el7/noarch/ceph-release-1-1.el7.noarch.rpm
yum -y install ceph
```

Один из серверов (h31.testnet) будет использован для деплоя (развертывания) системы Ceph, по этому устанавливаем на него ceph-deploy, с помощью которого будем производить развертывание хранилища.

(запускаем на сервере h31.testnet):

```
yum -y install ceph-deploy
```

Все наши сервера готовы к развертыванию. Запустим обновление и перезапустим их:

(запускаем на всех серверах):

```
yum -y update && reboot
```

ЭТАП №3

(Планирование дисковой подсистемы)

Так, как мы уже решили, что наш кластер будет работать полностью на распределенном хранилище Ceph, обойдемся без локальных NFS, GlusterFS и т.д. Будем полностью хранить все в Ceph.

Сам oVirt не умеет напрямую использовать Ceph ни в каком его виде. А вот qemu, в котором у нас живут все виртуалки может использовать Ceph RBD напрямую, без абстракций и прослоек. Хранилище Ceph в oVirt на данный момент поддерживается только в связке oVirt -> Cinder(openstack компонента) -> Ceph и никак по другому. Про интеграцию с Cinder можно почитать на офф. сайте.

Теперь очень хорошо подумаем и примем решение, как нам обеспечить oVirt хранилищем для “важного” механизма - HostedEngine, которое он понимает из коробки и при этом у нас будут выполнятся условия отказоустойчивости, а данные будут находиться в Ceph.

Сам HostedEngine не умеет запускаться из хранилища Ceph, даже через посредника Cinder, по этому на данном этапе нам Cinder не подходит. В решении данной задачи нам поможет демон NFS-Ganesha, который умеет использовать CephFS (файловое хранилище в Ceph,не путаем с RBD), и отдавать его по NFS V4, который в свою очередь уже отлично понимает oVirt.

Но вы спросите, а где отказоустойчивость ? А вот как раз NFS-Ganesha нам это и предоставит. NFS-Ganesha мы будем использовать не на одном хосте и монтировать его удаленно, а на всех, тем самым лишая схему единой точки отказа, а монтировать том будем через localhost. Немного поясню, т.к. у нас NFS-Ganesha будет запущена на всех серверах кластера и будет работает с Ceph напрямую, то где - бы мы не смонтировали том таким образом "mount.nfs localhost:/ceph ....", он будет доступен, вне зависимости жив - ли соседний сервер с NFS-Ganesha или нет. Немного упрощенно – у нас получается что то вроде для каждого сервера свой NFS, находящийся на этом – же сервере.

HostedEngine хранилищем мы обеспечили, но виртуальным машинам отдавать образы дисков по NFS просто не рационально, потому как Ceph->Ganesha->NFS будет работать на много медленнее, чем отдать виртуальным машинам Ceph RBD напрямую(графики будут позже), вот тут нам уже поможет прослойка с Cinder.

Подведем итог и спланируем хранилища. Что нам нужно от кластера Ceph ? Нам нужно следующее:

```
CephFS (MDS) - файловое хранилище, с которым будет работать NFS-Ganesha
Ceph RBD     - блочное хранилище, которое мы предоставим нашим
               виртуалкам через прослойку - Cinder
```

Какие тома у нас будут в итоге:

```
NFS-Ganesha:
    hed -   том для хранения данных HostedEngine
    dta -   создадим в NFS хранилище ( на всякий случай и для тестов скорости) для
            виртуалок
Ceph RBD:
    volumes - пул, где будут хранится «быстрые» диски виртуальных машин.
```

ЭТАП №4

(Подготовка CephFS)

Подготовим CephFS для дальнейшего использования, как NFS хранилище. Базовое развертывание кластера ceph я описывать не буду, т.к. статья станет огромной и не читабельной. Предположим, что мы развернули кластер ceph, и теперь нужно в нем создать файловое хранилище:

Создаем пулы (PG рассчитываем для своего кол-ва OSD и с расчетом, что будет создан еще один пул для RBD).

(запускаем на любой из admin нод кластера Ceph):

```
ceph osd pool create cephfs_data 32
ceph osd pool application enable cephfs_data cephfs
ceph osd pool create cephfs_metadata 32
ceph osd pool application enable cephfs_metadata cephfs
ceph osd pool create nfs-ganesha 8
ceph osd pool application enable nfs-ganesha cephfs
```

Создадим саму CephFS.

(запускаем на любой из admin нод кластера Ceph):

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

Разворачиваем три демона MDS (1-Active, 2-Standby)

(запускаем на deploy ноде кластера Ceph):

```
ceph-deploy mds create h31 h32 h33
```

Создаем пользователя 'cephfs' для доступа в файловое хранилище и сохраняем ключи.

(запускаем на любой из admin нод кластера Ceph):

```
ceph auth get-or-create client.cephfs mon 'allow r' mds 'allow rw' osd 'allow rw pool=cephfs_data allow rw pool=nfs-ganesha' -o /e
ceph auth get-key client.cephfs | tee /etc/ceph/client.cephfs.key
```

Ключи для пользователя 'cephfs' нам нужны на всех серверах, копируем их.

(запускаем на том - же сервере, что и предыдущее действие):

```
for i in h31 h32 h33;do scp /etc/ceph/*cephfs* root@$i:/etc/ceph/; done
```

ЭТАП №6

(Подготовка NFS-Ganesha)

Подготовим, запустим и проверим NFS-Ganesha на всех серверах нашего кластера. Так как CentOS 7 отсутствует NFS-Ganesha с поддержкой ceph, соберем ее сами или можно взять с моего репозитария.

(запускаем на всех серверах):

```
yum -y install ftp://lantaris.ru/pub/CentOS/7/ltrepo-contrib-1.0-0.el7.noarch.rpm
yum -y install nfs-ganesha nfs-ganesha-ceph nfs-ganesha-utils
```

Настраиваем NFS-Ganesha.

(запускаем на всех серверах):

```
CEPHFS_KEY=`cat /etc/ceph/client.cephfs.key|head -1` && cat << EOF >/etc/ganesha/ganesha.conf
NFS_CORE_PARAM
{
    Enable_NLM = false;
    Enable_RQUOTA = false;
    Protocols = 4;
}
NFSv4
{
    Delegations = true;
    RecoveryBackend = rados_ng;
}
CACHEINODE {
    Dir_Max = 1;
    Dir_Chunk = 0;
    Cache_FDs = false;
    NParts = 1;
    Cache_Size = 1;
}
EXPORT
{
    Export_ID=100;
    Protocols = 4;
    Transports = TCP;
    Path = /;
    Pseudo = /ceph;
    Access_Type = RW;
    Attr_Expiration_Time = 0;
    Squash = No_root_squash;
    FSAL {
        Name = CEPH;
         User_Id = "cephfs";
```

```
        Secret_Access_Key = "$CEPHFS_KEY";
    }
}
CEPH
{
    Ceph_Conf = /etc/ceph/ceph.conf;
}
RADOS_KV
{
    Ceph_Conf = /etc/ceph/ceph.conf;
}
RADOS_URLS
{
    Ceph_Conf = /etc/ceph/ceph.conf;
    UserId = "cephfs";
}
EOF
```

Запускаем демон nfs-ganesha для контроля смотрим лог файл (/var/log/ganesha/ganesha.log).

(запускаем на всех серверах):

```
systemctl enable nfs-ganesha.service && systemctl start nfs-ganesha.service
```

Попробуем смонтировать том

(желательно проверить на всех серверах с NFS-Ganesha):

```
mkdir -p /mnt/ceph
mount.nfs localhost:/ceph /mnt/ceph
```

Запустим 'df -h' или 'mount |grep nfs4' проконтролируем, что том смонтирован и все сделано правильно.

Теперь нужно создать каталоги и назначить права для HostedEngine и Data хранилищ.

(запускаем на том - же сервере, что и предыдущее действие):

```
mkdir -p /mnt/ceph/hed
mkdir -p /mnt/ceph/dta
chown -R 36.36 /mnt/ceph
```

NFS хранилище у нас готово, можно приступать к развертыванию oVirt.

ЭТАП №7

(Развертывание oVirt 4.1)

Как было сказано в начале статьи, человек читающий данную статью, уже имеет базовые знания системы виртуализации oVirt и уметь выполнять ее развертывание.

Производим развертывание oVirt 4.1. Уточню пару моментов. При развертывании HostedEngine, нужно обязательно указать для него хранилище через localhost.

Когда при развертывании HostedEngine появится вопрос о типе вводим: - тип 'nfs4' - путь 'localhost:/ceph/hed'.

После развертывания HostedEngine, заходим в его Web интерфейс и добавляем дата домен:

```
- тип 'NFS4',
- путь 'localhost:/ceph/dta'
```

Дожидаемся когда кластер развернется и на этом этапе мы получили работоспособный oVirt, который полностью работает на базе отказоустойчивого NFS поверх Ceph кластера.

В связи с большим количеством информации, как заставить теперь наш кластер работать с Ceph RBD, я опишу в следующей статье.

...спасибо за уделенное внимание... удачи...

Использованные в статье материалы:

Официальный сайт Openstack

Документация по NFS-Ganesha

Официальный сайт oVirt

Огромная помощь Алексея Костина