分布式场景之 NFS 存储高可用部署

1 准备事项

1.1 简介

NFS（Network File System）是一种网络文件系统，可以让不同的计算机系统之间共享文件。JumpServer 是一种用于进行安全管理计算机的堡垒机工具，通过配置 NFS，可以实现让各个节点共享 JumpServer 上的录像文件，从而实现多个节点同时查看录像的需求。

1.2 准备资源

角色	系统	IP
Master	centos 7.6	10.1.11.39
Slave	centos 7.6	10.1.11.67
VIP	/	10.1.11.68
JumpServer	v3.10.7	10.1.11.31

2 部署步骤

2.1 环境准备

在Master和Slave都执行

```
#
[root@jumpserver-nfs ~]# mkdir /data

#
[root@jumpserver-nfs ~]# firewall-cmd --permanent --add-port=2049/tcp
[root@jumpserver-nfs ~]# firewall-cmd --permanent --add-port=111/tcp
[root@jumpserver-nfs ~]# firewall-cmd --permanent --add-port=20048/tcp
[root@jumpserver-nfs ~]# firewall-cmd --permanent --add-port=4045/tcp
[root@jumpserver-nfs ~]# firewall-cmd --reload

#
[root@jumpserver-nfs ~]# firewall-cmd --list-ports

#  selinux
[root@jumpserver-nfs ~]# setenforce 0
[root@jumpserver-nfs ~]# sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
```

2.2 安装 NFS

在Master和Slave安装 NFS 组件

```
#  NFS
[root@jumpserver-nfs ~]# yum -y install nfs-utils rpcbind
[root@jumpserver-nfs ~]# systemctl start rpcbind && systemctl enable rpcbind
[root@jumpserver-nfs ~]# systemctl start nfs && systemctl enable nfs
[root@jumpserver-nfs ~]# lsmod | grep nfs
nfsd                   351321  13
auth_rpcgss            59415  1 nfsd
nfs_acl                12837  1 nfsd
lockd                  98048  1 nfsd
grace                  13515  2 nfsd,lockd
sunrpc                 358425  15 nfsd,auth_rpcgss,lockd,nfs_acl

#  JumpServer
[root@jumpserver-nfs ~]# cat /etc/exports
/data 10.1.11.31/32(rw,async,no_root_squash)
/data 10.1.11.67/32(rw,async,no_root_squash)
/data 10.1.11.39/32(rw,async,no_root_squash)

#
rw
async
no_root_squash root  root

[root@jumpserver-nfs ~]# exportfs -rav
exporting 10.1.11.31/32:/data
exporting 10.1.11.67/32:/data
exporting 10.1.11.39/32:/data
```

2.3 测试挂载

在 JumpServer 上测试挂载

```
# JumpServer  NFS
[root@jumpserver ~]# yum -y install nfs-utils

# Master
[root@jumpserver ~]# mount -t nfs 10.1.11.39:/data /data/jumpserver/core/data
[root@jumpserver ~]# df -Th
10.1.11.39:/data      nfs4       100G  1G   98G  1% /data/jumpserver/core/data

[root@jumpserver ~]# umount  /data/jumpserver/core/data

# Master
[root@jumpserver ~]# mount -t nfs 10.1.11.67:/data /data/jumpserver/core/data
[root@jumpserver ~]# df -Th
10.1.11.67:/data      nfs4       100G  1G   98G  1% /data/jumpserver/core/data

[root@jumpserver ~]# umount  /data/jumpserver/core/data
```

出现上述结果表示挂载可用

2.4 使用 rsync 工具配置 NFS 同步

Master 节点配置同步Slave节点

```
#  rsync
[root@jumpserver-nfs ~]# yum -y install rsync
[root@jumpserver-nfs ~]# cat /etc/rsyncd.conf
uid = root                      #  ID
gid = root                      #  ID
port = 873                      #  rsync
pid file = /var/rsyncd.pid      #  rsync  PID
log file = /var/log/rsyncd.log  #  rsync
use chroot = no                 #  chroot
max connections = 200           #
read only = false               #
list = false                    #
fake super = yes                # “fake super”
ignore errors                   #

[data]
path = /data                    #
auth users = appuser        #
secrets file = /etc/rsync_salve.pass  #
hosts allow = 10.1.11.67        # Slave IP, Master

#
[root@jumpserver-nfs ~]# echo 'appuser:Change@2024' > /etc/rsync_salve.pass
[root@jumpserver-nfs ~]# chmod 600 /etc/rsync_salve.pass   #  600

#
[root@jumpserver-nfs ~]# chown -R root:root /data/

#
[root@jumpserver-nfs ~]# rsync --daemon --config=/etc/rsyncd.conf
```

在Slave上面测试

```
[root@jumpserver-nfs ~]# yum -y install rsync
[root@jumpserver-nfs ~]# chown -R root:root /data/
[root@jumpserver-nfs ~]# echo "Change@2024" > /etc/rsync.pass     #
[root@jumpserver-nfs ~]# chmod 600 /etc/rsync.pass                #
[root@jumpserver-nfs ~]# touch /data/test01.txt                   #
[root@jumpserver-nfs ~]# rsync -arv /data/ appuser@10.1.11.39::data --password-file=/etc/rsync.pass       #

#Master
[root@jumpserver-nfs ~]# ls /data/
test01.txt
```

配置Master自动同步

```
[root@jumpserver-nfs ~]# cd /usr/local/
[root@jumpserver-nfs ~]# wget https://dl.qiyuesuo.com/private/nfs/sersync2.5.4_64bit_binary_stable_final.tar.gz
[root@jumpserver-nfs ~]# tar xvf sersync2.5.4_64bit_binary_stable_final.tar.gz
[root@jumpserver-nfs ~]# mv GNU-Linux-x86/ sersync
[root@jumpserver-nfs ~]# cd sersync/

#
[root@jumpserver-nfs ~]# sed -ri 's#<delete start="true"/>#<delete start="false"/>#g' confxml.xml
[root@jumpserver-nfs ~]# sed -ri '24s#<localpath watch="/opt/tongbu">#<localpath watch="/data">#g' confxml.xml
[root@jumpserver-nfs ~]# sed -ri '25s#<remote ip="127.0.0.1" name="tongbu1"/>#<remote ip="10.1.11.39" name="
data"/>#g' confxml.xml
[root@jumpserver-nfs ~]# sed -ri '30s#<commonParams params="-artuz"/>#<commonParams params="-az"/>#g' confxml.
xml
[root@jumpserver-nfs ~]# sed -ri '31s#<auth start="false" users="root" passwordfile="/etc/rsync.pas"/>#<auth
start="true" users="appuser" passwordfile="/etc/rsync.pass"/>#g' confxml.xml
[root@jumpserver-nfs ~]# sed -ri '33s#<timeout start="false" time="100"/><!-- timeout=100 -->#<timeout start="
true" time="100"/><!-- timeout=100 -->#g' confxml.xml

# Sersync
[root@jumpserver-nfs ~]# /usr/local/sersync/sersync2 -dro /usr/local/sersync/confxml.xml

#  master /data
[root@jumpserver-nfs ~]# touch test02.txt

#  salve  /data
[root@jumpserver-nfs ~]# ls /data
test01.txt   test02.txt
```

Slave 节点配置同步Master节点

```
#  rsync
[root@jumpserver-nfs ~]# yum -y install rsync
[root@jumpserver-nfs ~]# cat /etc/rsyncd.conf
uid = root                      #  ID
gid = root                      #  ID
port = 873                      #  rsync
pid file = /var/rsyncd.pid      #  rsync  PID
log file = /var/log/rsyncd.log  #  rsync
use chroot = no                 #  chroot
max connections = 200           #
read only = false               #
list = false                    #
fake super = yes                # “fake super”
ignore errors                   #

[data]
path = /data                    #
auth users = appuser        #
secrets file = /etc/rsync_salve.pass  #
hosts allow = 10.1.11.39        # Master IP, Slave

#
[root@jumpserver-nfs ~]# echo 'appuser:Change@2024' > /etc/rsync_salve.pass
[root@jumpserver-nfs ~]# chmod 600 /etc/rsync_salve.pass   #  600

#
[root@jumpserver-nfs ~]# chown -R root:root /data/

#
[root@jumpserver-nfs ~]# rsync --daemon --config=/etc/rsyncd.conf
```

在Master上面测试

```
[root@jumpserver-nfs ~]# yum -y install rsync
[root@jumpserver-nfs ~]# chown -R root:root /data/
[root@jumpserver-nfs ~]# echo "Change@2024" > /etc/rsync.pass     #
[root@jumpserver-nfs ~]# chmod 600 /etc/rsync.pass                 #
[root@jumpserver-nfs ~]# touch /data/test03.txt                   #
[root@jumpserver-nfs ~]# rsync -arv /data/ appuser@10.1.11.67::data --password-file=/etc/rsync.pass       #
PSrsync -avzP --bwlimit=1000 --delete  --password-file=/etc/rsync.pass appuser@10.1.11.67::data /data/

#Master
[root@jumpserver-nfs ~]# ls /data/
test01.txt test02.txt test03.txt
```

配置Slave自动同步

```
[root@jumpserver-nfs ~]# cd /usr/local/
[root@jumpserver-nfs ~]# wget https://dl.qiyuesuo.com/private/nfs/sersync2.5.4_64bit_binary_stable_final.tar.gz
[root@jumpserver-nfs ~]# tar xvf sersync2.5.4_64bit_binary_stable_final.tar.gz
[root@jumpserver-nfs ~]# mv GNU-Linux-x86/ sersync
[root@jumpserver-nfs ~]# cd sersync/

#
[root@jumpserver-nfs ~]# sed -ri 's#<delete start="true"/>#<delete start="false"/>#g' confxml.xml
[root@jumpserver-nfs ~]# sed -ri '24s#<localpath watch="/opt/tongbu">#<localpath watch="/data">#g' confxml.xml
[root@jumpserver-nfs ~]# sed -ri '25s#<remote ip="127.0.0.1" name="tongbu1"/>#<remote ip="10.1.11.67" name="
data"/>#g' confxml.xml
[root@jumpserver-nfs ~]# sed -ri '30s#<commonParams params="-artuz"/>#<commonParams params="-az"/>#g' confxml.
xml
[root@jumpserver-nfs ~]# sed -ri '31s#<auth start="false" users="root" passwordfile="/etc/rsync.pas"/>#<auth
start="true" users="appuser" passwordfile="/etc/rsync.pass"/>#g' confxml.xml
[root@jumpserver-nfs ~]# sed -ri '33s#<timeout start="false" time="100"/><!-- timeout=100 -->#<timeout start="
true" time="100"/><!-- timeout=100 -->#g' confxml.xml

# Sersync
[root@jumpserver-nfs ~]# /usr/local/sersync/sersync2 -dro /usr/local/sersync/confxml.xml

#  master /data
[root@jumpserver-nfs ~]# touch test04.txt

#  salve  /data
[root@jumpserver-nfs ~]# ls /data
test01.txt   test02.txt   test03.txt   test04.txt
```

2.5 使用 Keepalived 实现高可用

在Master上部署 keepalived 工具

```
[root@jumpserver-nfs ~]# yum -y install keepalived
[root@jumpserver-nfs ~]# vim /etc/keepalived/keepalived.conf
#  eth0  ip addr
#  10.1.11.68  ip  ip
! Configuration File for keepalived

global_defs {
   router_id Master
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 150
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass Calong@2015
    }
    virtual_ipaddress {
      10.1.11.68
    }
}

#
systemctl start  keepalived.service && systemctl enable keepalived.service
```

在Slave上部署 keepalived 工具

```
[root@jumpserver-nfs ~]# yum -y install keepalived
[root@jumpserver-nfs ~]# vim /etc/keepalived/keepalived.conf
#  eth0  ip addr
#  10.1.11.68  ip  ip
! Configuration File for keepalived

global_defs {
   router_id Slave
}

vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 51
    priority 120
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass Calong@2015
    }
    virtual_ipaddress {
      10.1.11.68
    }
}

#
[root@jumpserver-nfs ~]# systemctl start  keepalived.service && systemctl enable keepalived.service
```

设置 keepalived 脚本实现漂移

```
# Master nfs
[root@jumpserver-nfs ~]# vim /usr/local/sbin/check_nfs.sh
#!/bin/sh
#
step=1 #60
for (( i = 0; i < 60; i=(i+step) )); do
  ###nfs
  /sbin/service nfs status &>/dev/null
  if [ $? -ne 0 ];then
    ###
    /sbin/service nfs restart
    /sbin/service nfs status &>/dev/null
    if [ $? -ne 0 ];then
       #  keepalived
       systemctl stop keepalived.service
    fi
  fi
  sleep $step
done

# SlaveMaster
[root@jumpserver-nfs ~]# vim /usr/local/sbin/check_mount.sh
#!/bin/sh
#
step=1 #60
for (( i = 0; i < 60; i=(i+step) )); do
  mount=`df -Th|grep /data/jumpserver/core/data`
  if [ $mount = "" ];then
     umount /data/jumpserver/core/data
     mount -t nfs 10.1.11.68:/data /data/jumpserver/core/data
  fi
  sleep $step
done

#
[root@jumpserver-nfs ~]# chmod +x /usr/local/sbin/check_mount.sh
[root@jumpserver-nfs ~]# crontab -e
#
* * * * *  /usr/local/sbin/check_nfs.sh &> /dev/null
```

3 结语

确保 NFS（Network File System）的高可用性对于 JumpServer 的录像文件保存存在重要意义，不过本身高可用非 NFS 官方实现，存在一定风险，请谨慎使用。