Proxmox VE

- Configurazione Synology DS1817 come QDevice per il quorum

Configurazione Synology DS1817 come QDevice per il quorum

Obiettivo

Configurare un NAS **Synology DS1817** come voto esterno per il quorum di un cluster **Proxmox VE a 2 nodi**, usando `corosync-qnetd`.

Schema finale:

```
proxmox01      1 voto
proxmox02      1 voto
Synology NAS   1 voto esterno QDevice

Totale voti:   3
Quorum:        2
```

Il Synology **non diventa un nodo Proxmox**. Serve solo come server `corosync-qnetd`.

Scenario di esempio

Host	IP
`proxmox01`	`192.168.0.10`
`proxmox02`	`192.168.0.11`
`Synology DS1817`	`192.168.0.50`

Cluster Proxmox già esistente:

```
proxmox01 + proxmox02
```

Nota importante

Questa guida è pensata per **Synology DS1817 non Plus**, quindi modello ARM.

Non useremo:

```
VM Debian
Docker
Raspberry
Debian chroot
```

La soluzione funzionante è:

```
Synology DS1817
└── Entware
    ├── corosync
    ├── corosync-qdevice
    ├── nss-utils
    └── openssh-sftp-server
```

---

1. Abilitare SSH sul Synology

Da DSM:

```
Pannello di controllo
→ Terminale e SNMP
→ Abilita servizio SSH
```

Poi collegarsi al NAS:

```
ssh tuo_utente@192.168.0.50
sudo -i
```

---

2. Installare Entware sul Synology

Creare la directory persistente:

```
mkdir -p /volume1/@Entware/opt
rm -rf /opt
mkdir -p /opt
mount -o bind /volume1/@Entware/opt /opt
```

Installare Entware per ARMv7:

```
wget -O - https://bin.entware.net/armv7sf-k3.2/installer/generic.sh | /bin/sh
```

Aggiornare `opkg`:

```
/opt/bin/opkg update
```

Installare alcuni strumenti base:

```
/opt/bin/opkg install bash wget tar gzip xz binutils
```

Verificare che `opkg` funzioni:

```
/opt/bin/opkg update
```

---

3. Installare i pacchetti necessari

Sul Synology:

```
/opt/bin/opkg update
/opt/bin/opkg install corosync corosync-qdevice nss-utils openssh-sftp-server
```

Verificare i binari:

```
find /opt -type f | grep -E 'corosync-qnetd|qnetd|qdevice|certutil|sftp-server'
```

Output atteso simile:

```
/opt/bin/corosync-qnetd
/opt/bin/corosync-qnetd-tool
/opt/bin/corosync-qnetd-certutil
/opt/bin/certutil
/opt/lib/sftp-server
```

---

4. Creare i wrapper per Proxmox

Proxmox, durante il comando `pvecm qdevice setup`, si aspetta alcuni binari in percorsi standard come `/usr/bin` e `/usr/sbin`.

Creare i wrapper:

```
cat > /usr/sbin/corosync-qnetd <<'EOF'
#!/bin/sh
exec /opt/bin/corosync-qnetd "$@"
EOF

cat > /usr/bin/corosync-qnetd-tool <<'EOF'
#!/bin/sh
exec /opt/bin/corosync-qnetd-tool "$@"
EOF

cat > /usr/bin/corosync-qnetd-certutil <<'EOF'
#!/bin/sh
exec /opt/bin/corosync-qnetd-certutil "$@"
EOF

cat > /usr/bin/certutil <<'EOF'
#!/bin/sh
exec /opt/bin/certutil "$@"
EOF

chmod +x /usr/sbin/corosync-qnetd
chmod +x /usr/bin/corosync-qnetd-tool
chmod +x /usr/bin/corosync-qnetd-certutil
chmod +x /usr/bin/certutil
```

Verificare:

```
which corosync-qnetd
which corosync-qnetd-tool
which corosync-qnetd-certutil
which certutil
```

---

5. Sistemare le directory di Corosync

Creare la directory di configurazione:

```
mkdir -p /etc/corosync/qnetd
chmod 700 /etc/corosync
chmod 700 /etc/corosync/qnetd
```

Sul Synology, Entware usa `/opt/etc/corosync`, mentre Proxmox cerca `/etc/corosync`.

La struttura funzionante è:

```
/etc/corosync = directory reale
/opt/etc/corosync -> /etc/corosync
```

Creare il symlink:

```
rm -rf /opt/etc/corosync
ln -s /etc/corosync /opt/etc/corosync
```

Verificare:

```
ls -ld /etc/corosync /opt/etc/corosync
```

Output atteso:

```
drwx------ ... /etc/corosync
lrwxrwxrwx ... /opt/etc/corosync -> /etc/corosync
```

---

6. Sistemare SFTP/SCP sul Synology

`pvecm qdevice setup` usa `scp` per copiare certificati e file necessari.

Sul DS1817 può comparire questo errore:

```
subsystem request failed on channel 0
scp: Connection closed
```

Per risolvere, usare il server SFTP installato tramite Entware.

Modificare il file SSH:

```
vi /etc/ssh/sshd_config
```

Cercare la riga `Subsystem sftp`.

Commentare eventuali righe esistenti, per esempio:

```
#Subsystem       sftp    internal-sftp -f DAEMON -u 000
```

Aggiungere questa riga:

```
Subsystem       sftp    /opt/lib/sftp-server
```

Verificare che ci sia una sola riga attiva:

```
grep -n "^[^#].*Subsystem.*sftp" /etc/ssh/sshd_config
```

Riavviare SSH:

```
synosystemctl restart sshd
```

Se non funziona, disabilitare e riabilitare SSH da DSM:

```
Pannello di controllo
→ Terminale e SNMP
→ Disabilita SSH
→ Applica
```

```
→ Abilita SSH
→ Applica
```

---

7. Abilitare accesso root SSH via chiave

Da `proxmox01`, controllare se esiste già la chiave SSH root:

```
ls -l /root/.ssh/id_rsa.pub
```

Se non esiste, crearla:

```
ssh-keygen -t rsa -b 4096 -f /root/.ssh/id_rsa
```

Mostrare la chiave pubblica:

```
cat /root/.ssh/id_rsa.pub
```

Sul Synology, da root:

```
mkdir -p /root/.ssh
chmod 700 /root/.ssh
vi /root/.ssh/authorized_keys
```

Incollare dentro `authorized_keys` la chiave pubblica di `proxmox01`.

Poi sistemare i permessi:

```
chmod 600 /root/.ssh/authorized_keys
chown -R root:root /root/.ssh
```

Nel file SSH:

```
vi /etc/ssh/sshd_config
```

assicurarsi che siano presenti:

```
PermitRootLogin prohibit-password
PubkeyAuthentication yes
```

Riavviare SSH:

```
synosystemctl restart sshd
```

Test da `proxmox01`:

```
ssh root@192.168.0.50 'hostname'
```

Output atteso:

```
synology
```

Test SCP:

```
scp root@192.168.0.50:/etc/hosts /tmp/synology-hosts-test
```

Se questo comando funziona, si può proseguire.

---

8. Rimuovere eventuale vecchio QDevice

Se era già presente un QDevice precedente, ad esempio un Raspberry Pi, rimuoverlo da Proxmox.

Su `proxmox01`:

```
pvecm qdevice remove
```

Verificare:

```
pvecm status
```

Output temporaneo atteso:

```
Expected votes: 2
Total votes:    2
```

```
Quorum:         2
Quorate:        Yes
```

---

9. Pulire eventuale database qnetd precedente

Sul Synology:

```
rm -rf /etc/corosync/qnetd/nssdb
rm -rf /opt/etc/corosync/qnetd/nssdb
```

---

10. Configurare il Synology come QDevice

Da `proxmox01`:

```
pvecm qdevice setup 192.168.0.50
```

Al termine, verificare:

```
pvecm status
```

Output atteso:

```
Expected votes:   3
Total votes:      3
Quorum:           2
Flags:            Quorate Qdevice
```

Membership attesa:

```
Nodeid      Votes    Qdevice Name
0x00000001      1    A,V,NMW 192.168.0.10
```

```
0x00000002      1    A,V,NMW 192.168.0.11
0x00000000      1            Qdevice
```

---

11. Avviare qnetd sul Synology

Sul Synology:

```
nohup /opt/bin/corosync-qnetd -f > /var/log/corosync-qnetd.log 2>&1 &
```

Verificare:

```
ps | grep '[c]orosync-qnetd'
netstat -tlnp | grep 5403
```

Output atteso:

```
tcp6  0  0 :::5403  :::*  LISTEN  .../corosync-qnetd
```

Controllare lo stato:

```
/opt/bin/corosync-qnetd-tool -l
```

Output atteso simile:

```
Cluster "casa":
    Algorithm: Fifty-Fifty split (KAP Tie-breaker)
    Tie-breaker: Node with lowest node ID
    Node ID 1:
        Vote: ACK
    Node ID 2:
        Vote: No change (ACK)
```

---

12. Creare avvio automatico al boot del Synology

Creare lo script:

```
cat > /usr/local/bin/start-qnetd.sh <<'EOF'
#!/bin/sh

mkdir -p /opt

if ! mountpoint -q /opt; then
    mount -o bind /volume1/@Entware/opt /opt
fi

if ! ps | grep '[c]orosync-qnetd' >/dev/null 2>&1; then
    nohup /opt/bin/corosync-qnetd -f > /var/log/corosync-qnetd.log 2>&1 &
fi
EOF

chmod +x /usr/local/bin/start-qnetd.sh
```

Testare lo script:

```
pkill corosync-qnetd
/usr/local/bin/start-qnetd.sh
ps | grep '[c]orosync-qnetd'
netstat -tlnp | grep 5403
```

Poi da DSM:

```
Pannello di controllo
→ Utilità di pianificazione
→ Crea
→ Attività attivata
```

Impostare:

```
Utente: root
Evento: Avvio
```

Script:

```
/usr/local/bin/start-qnetd.sh
```

---

13. Verifiche finali

Su Proxmox:

```
pvecm status
```

Output corretto:

```
Expected votes: 3
Total votes:    3
Quorum:         2
Flags:          Quorate Qdevice
```

Su Synology:

```
ps | grep '[c]orosync-qnetd'
netstat -tlnp | grep 5403
/opt/bin/corosync-qnetd-tool -l
```

Errori comuni e soluzioni

Errore: `certutil: command not found`

Errore tipico:

```
/opt/bin/corosync-qnetd-certutil: line 143: certutil: command not found
```

Soluzione:

```
/opt/bin/opkg install nss-utils
```

Poi creare il wrapper:

```
cat > /usr/bin/certutil <<'EOF'
#!/bin/sh
exec /opt/bin/certutil "$@"
```

```
EOF

chmod +x /usr/bin/certutil
```

Errore: `subsystem request failed on channel 0`

Errore tipico durante `scp`:

```
subsystem request failed on channel 0
scp: Connection closed
```

Soluzione:

```
/opt/bin/opkg install openssh-sftp-server
```

Poi in `/etc/ssh/sshd_config`:

```
Subsystem       sftp    /opt/lib/sftp-server
```

Riavviare SSH:

```
synosystemctl restart sshd
```

Errore: QDevice configurato ma voto non attivo

Su Proxmox può apparire:

```
Qdevice (votes 1)
Total votes: 2
```

E sul Synology:

```
/opt/bin/corosync-qnetd-tool -l
```

può dare:

```
Can't connect to qnetd socket
```

Significa che `qnetd` non è avviato.

Avviarlo:

```
nohup /opt/bin/corosync-qnetd -f > /var/log/corosync-qnetd.log 2>&1 &
```

Poi verificare:

```
pvecm status
```

Risultato finale

La configurazione corretta lato Proxmox deve essere:

```
Expected votes: 3
Total votes:    3
Quorum:         2
Flags:          Quorate Qdevice
```

La configurazione corretta lato Synology deve mostrare:

```
ps | grep '[c]orosync-qnetd'
netstat -tlnp | grep 5403
/opt/bin/corosync-qnetd-tool -l
```

Con `qnetd` attivo e i due nodi Proxmox collegati, il Synology DS1817 sta facendo correttamente da **voto esterno quorum** per Proxmox HA.