

목차

PVE9 local repository 구성 3

레포지토리 복제 3

레포지토리 구성 5

레포지토리 사용 9

만료일 무시 설정 10

참조링크 10

PVE9 local repository 구성

인터넷이 안되는 환경이나 특수한 상황에서 PVE를 구성해야 할 경우 레포지토리를 접근하기 위해 로컬에 레포지토리 미러를 구성하는 방법에 대해 설명한다.

레포지토리 복제

먼저 인터넷환경의 pve9 서버를 하나 설치한다. 그리고 레포지토리 미러링을 위한 디스크를 추가로 구성하도록 한다. 기본 디스크는 OS와 local storage로 등록되기 때문에 추가 디스크를 할당하는것이 좋다.

로컬 미러를 구성할 경로를 /repo라고 가정한다.

먼저 미러를 위한 proxmox-offline-mirror 툴을 설치한다.

```
$ apt install proxmox-offline-mirror
```

이후 미러를 하기 위한 설정을 진행한다.

```
proxmox-repo:~# proxmox-offline-mirror setup  
Loaded existing config.
```

Existing config entries:

```
mirror 'pve_trixie_no-subscription'  
mirror 'debian_trixie_backports'  
mirror 'debian_trixie_security'  
mirror 'debian_trixie_main'  
mirror 'debian_trixie_updates'  
mirror 'ceph_squid_trixie'
```

Select Action:

- 0) Add new mirror entry
- 1) Add new medium entry
- 2) Add new subscription key
- 3) Quit

Add new mirror entry 로 미러할 레포지토리를 선택한다. medium과 subscription은 없으므로 생략한다.

- pve_trixie_no-subscription
- debian_trixie_backports
- debian_trixie_security
- debian_trixie_main
- debian_trixie_updates
- ceph_squid_trixie

위의 6개 레포지토리를 선택했다면 저장하고 나와서 스냅샷을 생성하고 미러링을 수행한다.

```
# 스냅샷 생성 및 미러링 시작
$ proxmox-offline-mirror mirror snapshot create <MIRROR_NAME>

# 미러링 완료 후 정리
$ proxmox-offline-mirror mirror gc
```

하지만 미러링이 동작하게 되면 오랜시간이 걸리거나 중간에 오류로 중단되거나 할 경우가 있는데 일일이 명령어를 재시도 하기보단 아래 스크립트를 이용하도록 한다.

mirror.sh

```
#!/bin/bash
# 미러 ID 목록
MIRRORS=(
    pve_trixie_no-subscription
    ceph_squid_trixie
    debian_trixie_main
    debian_trixie_security
    debian_trixie_updates
    debian_trixie_backports
)
BASE_DIR="/data/proxmox-offline-mirror/mirrors"

# 각 미러에 대해 스냅샷 생성
for MIRROR in "${MIRRORS[@]}"; do
    if [ ! -d "$BASE_DIR/$MIRROR" ]; then
        echo "Dir not exists : $MIRROR creating..."
        mkdir -p "$BASE_DIR/$MIRROR"
    fi

    echo "Creating snapshot for $MIRROR..."
    proxmox-offline-mirror mirror snapshot create "$MIRROR"
    if [ $? -eq 0 ]; then
        echo "Snapshot for $MIRROR created successfully."
    else
        echo "Error creating snapshot for $MIRROR."
    fi
done

# 가비지 컬렉션 실행 (스냅샷에 존재하지 않는 패키지 제거)
echo "Running garbage collection..."
proxmox-offline-mirror mirror gc
echo "Garbage collection completed."
```

미러링을 시작하면 한번에 완료되는것이 아니라 네트워크 상태나 레포지토리 서버의 상태에 따라 중간에 실패하거나 끊기는 경우가 있다. 그럴경우 스크립트를 다시 재실행하면 된다. 물론 스냅샷은 실행할

때마다 생성되겠지만 어차피 패키지 파일은 동일한 파일이 있는경우 생략하기 때문에 상관이 없다.

2025-11-21일 현재 모든 레포지토리의 동기화 후 용량은 대략 아래와 같다.

- 570M ceph_squid_trixie
- 14G debian_trixie_backports
- 96G debian_trixie_main
- 2.9G debian_trixie_security
- 19M debian_trixie_updates
- 8.8G pve_trixie_no-subscription

레포지토리 구성

로컬 미러를 동기화 완료하면 아래와 같은 디렉토리 구성이 보이게 된다.

```
root@pve:/data# cd proxmox-offline-mirror/mirrors/
root@pve:/data/proxmox-offline-mirror/mirrors# ll
total 4
drwxr-xr-x 9 root root 4096 Nov 21 09:44 ./
drwxr-xr-x 3 root root  21 Nov 20 23:08 ../
drwxr-xr-x 3 root root  34 Nov 21 10:49 ceph_squid_trixie/
drwxr-xr-x 3 root root  34 Nov 21 10:52 debian_trixie_backports/
drwxr-xr-x 3 root root  34 Nov 21 10:52 debian_trixie_main/
drwxr-xr-x 3 root root  34 Nov 21 10:52 debian_trixie_security/
drwxr-xr-x 3 root root  34 Nov 21 10:52 debian_trixie_updates/
drwxr-xr-x 4 root root  47 Nov 20 23:11 .pool/
drwxr-xr-x 3 root root  34 Nov 21 10:49 pve_trixie_no-subscription/
root@pve:/data/proxmox-offline-mirror/mirrors# tree -L 3
```

```
.
├── ceph_squid_trixie
│   ├── 2025-10-16T01:52:19Z
│   │   └── dists
│   ├── 2025-10-16T02:13:46Z
│   │   └── dists
│   ├── 2025-10-16T03:01:59Z
│   │   └── dists
│   ├── 2025-10-16T04:40:44Z
│   │   └── dists
│   └── 2025-10-16T06:12:13Z
│       └── dists
├── debian_trixie_main
│   ├── 2025-10-16T01:54:37Z.tmp
│   │   ├── dists
│   │   └── pool
│   ├── 2025-10-16T02:13:47Z.tmp
│   │   ├── dists
│   │   └── pool
│   └── 2025-10-16T03:02:02Z.tmp
│       └── dists
```

```
├── pool
│   ├── 2025-10-16T04:40:47Z
│   │   ├── dists
│   │   └── pool
│   └── 2025-10-16T06:12:16Z
│       ├── dists
│       └── pool
├── debian_trixie_security
│   ├── 2025-10-16T01:31:59Z
│   │   ├── dists
│   │   └── pool
│   ├── 2025-10-16T01:58:16Z
│   │   ├── dists
│   │   └── pool
│   ├── 2025-10-16T02:22:15Z
│   │   ├── dists
│   │   └── pool
│   ├── 2025-10-16T04:38:51Z
│   │   ├── dists
│   │   └── pool
│   ├── 2025-10-16T06:11:35Z
│   │   ├── dists
│   │   └── pool
│   └── 2025-10-16T06:16:05Z
│       ├── dists
│       └── pool
├── debian_trixie_updates
│   ├── 2025-10-16T01:58:29Z
│   │   ├── dists
│   │   └── pool
│   ├── 2025-10-16T02:22:28Z
│   │   ├── dists
│   │   └── pool
│   ├── 2025-10-16T04:39:05Z
│   │   ├── dists
│   │   └── pool
│   ├── 2025-10-16T06:11:50Z
│   │   ├── dists
│   │   └── pool
│   └── 2025-10-16T06:16:20Z
│       ├── dists
│       └── pool
└── pve_trixie_no-subscription
    ├── 2025-10-16T01:40:20Z.tmp
    │   └── dists
    ├── 2025-10-16T02:04:53Z
    │   └── dists
    ├── 2025-10-16T03:01:43Z
    │   └── dists
    └── 2025-10-16T04:40:27Z
```

```
└─ dists
└─ 2025-10-16T06:11:55Z
└─ dists
```

74 directories, 0 files

위와같이 미러링을 시도할때마다 스냅샷이 날짜시간 형태로 계속해서 만들어진다. 그렇다고 해서 동일한 패키지 파일을 계속 받아두는 방식은 아니고 **.pool** 디렉토리에 패키지 원본파일이 저장되어있고 스냅샷 디렉토리에는 하드링크로 존재하는 형태로 구성된다.

만약 스냅샷을 모두 삭제하고 싶다면 아래 스크립트를 이용하도록 한다. **snap_clear.sh**

```
#!/bin/bash
MIRRORS=(
    pve_trixie_no-subscription
    ceph_squid_trixie
    debian_trixie_main
    debian_trixie_security
    debian_trixie_updates
    debian_trixie_backports
)

# 각 미러에 대해 모든 스냅샷 제거
for MIRROR in "${MIRRORS[@]"; do
    echo "Deleting snapshot for $MIRROR..."
    proxmox-offline-mirror mirror snapshot list $MIRROR --output-format json
    | jq -r ".[\"$MIRROR\"][]" | while read SNAPSHOT; do
        echo "Deleting snapshot: $SNAPSHOT"
        proxmox-offline-mirror mirror snapshot remove $MIRROR "$SNAPSHOT"
    done
done
```

실행후 반드시 다시한번 미러링을 실시하여 스냅샷을 최소 한개 만들어놔야지만 **.pool** 에 받아놓은 패키지 데이터가 gc로 인해 삭제되지 않는다.

이렇게 미러는 구성되어 있지만 우리는 일반적인 레포지토리 형태로 사용하고 싶기 때문에 아래 스크립트를 이용하여 레포지토리를 복제하도록 한다.

copy-mirror.sh

```
#!/bin/bash
MIRROR_BASE=/data/repo
ORIG_BASE=/data/proxmox-offline-mirror/mirrors

# directory check
if [ ! -d "$MIRROR_BASE" ]; then
    echo "Dir not exists : $MIRROR_BASE creating..."
    mkdir -p "$MIRROR_BASE"
fi
```

```
for MIRROR in ceph_squid_trixie debian_trixie_main debian_trixie_security
debian_trixie_updates debian_trixie_backports pve_trixie_no-subscription; do
    LATEST=$(ls -d $ORIG_BASE/$MIRROR/*Z | sort -r | head -n 1)
    echo "LATEST=$LATEST"
    if [ -n "$LATEST" ]; then
        rm -rf $MIRROR_BASE/$MIRROR/
        rsync -a --copy-links $LATEST/ $MIRROR_BASE/$MIRROR/
        echo "Converted $MIRROR to actual files from $LATEST."
    fi
done
```

위 코드는 스냅샷형태로 되어있는 레포지토리를 일반적인 디렉토리 형태로 복제하는 스크립트이다. 예제에서는 /repo 디렉토리에 복사하도록 되어있다.

위 스크립트를 실행하게 되면 /repo 디렉토리에 아래와 같이 최신 스냅샷 기준으로 레포지토리가 복제된다.

```
proxmox-repo:~# cd /repo/
proxmox-repo:/repo# ls -al
total 28
drwxr-xr-x  7 root root 4096 Oct 17 12:18 ./
drwxr-xr-x 19 root root 4096 Oct 16 15:17 ../
drwxr-xr-x  3 root root 4096 Oct 16 15:12 ceph_squid_trixie/
drwxr-xr-x  4 root root 4096 Oct 16 15:12 debian_trixie_main/
drwxr-xr-x  4 root root 4096 Oct 16 15:16 debian_trixie_security/
drwxr-xr-x  4 root root 4096 Oct 16 15:16 debian_trixie_updates/
drwxr-xr-x  3 root root 4096 Oct 16 15:11 pve_trixie_no-subscription/
proxmox-repo:/repo# tree -L3
```

```
.
├── ceph_squid_trixie
│   └── dists
│       └── trixie
├── debian_trixie_main
│   ├── dists
│   │   └── trixie
│   └── pool
│       ├── contrib
│       └── main
├── debian_trixie_security
│   ├── dists
│   │   └── trixie-security
│   └── pool
│       └── updates
├── debian_trixie_updates
│   ├── dists
│   │   └── trixie-updates
│   └── pool
│       └── main
└── pve_trixie_no-subscription
```



```
└─ dists
  └─ trixie
```

23 directories, 0 files

레포지토리 사용

위와같이 최종적으로 만들어진 레포지토리를 다른 서버에서 사용하기 위해 레포지토리 소스를 아래처럼 등록한다.

local.sources 파일 형태로 사용할 경우

```
#####
# PVE9 local repository sources type
# 2025-10-16
#####
Types: deb
URIs: file:///repo/ceph_squid_trixie
Suites: trixie
Components: no-subscription
Trusted: yes

Types: deb
URIs: file:///repo/debian_trixie_main
Suites: trixie
Components: main contrib
Trusted: yes

Types: deb
URIs: file:///repo/debian_trixie_security
Suites: trixie-security
Components: main
Trusted: yes

Types: deb
URIs: file:///repo/debian_trixie_updates
Suites: trixie-updates
Components: main
Trusted: yes

Types: deb
URIs: file:///repo/pve_trixie_no-subscription
Suites: trixie
Components: pve-no-subscription
Trusted: yes
```

local.list 형태로 사용할 경우

```
#####  
# PVE9 local repository list type  
# 2025-10-16  
#####  
deb [trusted=yes] file:///repo/ceph_squid_trixie trixie no-subscription  
deb [trusted=yes] file:///repo/debian_trixie_main trixie main contrib  
deb [trusted=yes] file:///repo/debian_trixie_security trixie-security main  
deb [trusted=yes] file:///repo/debian_trixie_updates trixie-updates main  
deb [trusted=yes] file:///repo/pve_trixie_no-subscription trixie pve-no-subscription
```

만료일 무시 설정

이렇게 만들어진 로컬 레포지토리를 영구적으로 사용시 레포지토리가 3주이상 동기화가 되지 않는 경우 만료일 오류가 발생한다.

```
Error: Release file for file:/data/repo/debian_trixie_security/dists/trixie-security/InRelease is expired (invalid since 21d 21h 19min 4s). Updates for this repository will not be applied.  
Error: Release file for file:/data/repo/debian_trixie_updates/dists/trixie-updates/InRelease is expired (invalid since 21d 22h 14min 21s). Updates for this repository will not be applied.
```

물론 레포지토리를 최신으로 업데이트해주면 되지만 실질적으로 오프라인 환경에서 사용하는 경우 그냥 만료일을 무시하도록 설정해주면 된다.

/etc/apt/apt.conf.d/99local-repo-no-valid-until 파일을 아래와 같이 생성한다.

```
// 로컬/오프라인 미러는 Release 파일 만료 검사 완전히 비활성화  
Acquire::Check-Valid-Until "false";  
Acquire::Check-Date "false";  
  
// 추가로 Release 파일의 날짜 검사도 완전히 끄기 (더 강력함)  
Acquire::AllowInsecureRepositories "true";  
Acquire::AllowDowngradeToInsecureRepositories "true";  
  
// 신뢰할 수 있는 로컬 미러이므로 서명 검증도 선택적으로 무시 (필요시)  
APT::Get::AllowUnauthenticated "true";
```

참조링크

- <https://pom.proxmox.com/>

From:
<https://atl.kr/dokuwiki/> - AllThatLinux!

Permanent link:
https://atl.kr/dokuwiki/doku.php/pve9_local_repository_%EA%B5%AC%EC%84%B1?rev=1763691233

Last update: **2025/11/21 02:13**

