목차

Debian계열 최신 레포지토리 미러 구성 3
작업 흐름 3
미러 구성 스크립트 3

Last update: 2025/07/08 14:28	debian 계열_최신_레포지토리_미러_구성	https://atl.kr/dokuwiki/doku.php/debian%EA%B3%84%EC%97%B4_%EC%B5%9C%EC%8B%A0_%EB%A0%88%ED%8F%AC%EC%A7%80%ED%86%A0%EB%A6%AC_%EB%AF%B8%EB%9F%AC_%EA%B5%AC%EC%84%B1

Debian계열 최신 레포지토리 미러 구성

Debian/Ubuntu 계열의 apt/deb 기반 레포지토리는 RHEL계열의 rpm레포지토리처럼 reposync를 이용하여 최신버전의 패키지로만 레포지토리 미러를 구성할 수 없다.

하지만 aptly 툴을 이용하면 이와 유사하게 레포지토리를 구성할 수 있다.

aptly를 사용하면 원격 리포지토리에서 현재 날짜 기준으로 각 패키지의 가장 최신 버전만을 가져와 로컬 미러를 구성할 수 있습니다. debmirror와 달리 aptly는 **미러링 → 스냅샷 생성 → 필터링 → 게시**의 단계별 접근 방식을 통해 이를 구현합니다.

작업 흐름

- 미러 생성 (Mirror Create): 원격 리포지토리의 정보를 aptly에 등록합니다.
- 미러 업데이트 (Mirror Update): 원격 리포지토리의 패키지 정보를 로컬 데이터베이스로 다운로드합니다. (이때 실제 .deb 파일이 모두 다운로드되는 것은 아닙니다.)
- 스냅샷 생성 (Snapshot Create): 업데이트된 미러의 현재 상태를 스냅샷으로 저장합니다.
- 스냅샷 병합 및 필터링 (Snapshot Merge & Filter): 여러 스냅샷을 하나로 합치면서 -latest 플래그를 사용해 각 패키지의 최신 버전만 남깁니다.
- 스냅샷 게시 (Publish Snapshot): 필터링이 완료된 최종 스냅샷을 apt가 사용할 수 있는 실제 리포지토리로 발행합니다.

미러 구성 스크립트

아래는 레포지토리 미러 구성 스크립트 예제이다. create-mirror.sh

```
#!/bin/bash

# --- 변수 설정 ---
export GNUPGHOME=~/.gnupg
export APTLY_CONFIG=~/.aptly.conf

# 미러링할 리포지토리 정보
REPO_LIST=(
    "debian-main|http://ftp.debian.org/debian/|bookworm|main
contrib|/usr/share/keyrings/debian-archive-keyring.gpg"
    "debian-updates|http://ftp.debian.org/debian/|bookworm-updates|main
contrib|/usr/share/keyrings/debian-archive-keyring.gpg"
    "debian-
security|http://security.debian.org/debian-security/|bookworm-security|main
contrib|/usr/share/keyrings/debian-archive-keyring.gpg"
    "proxmox-
ve|http://download.proxmox.com/debian/pve|bookworm|pve-no-subscription|/etc/
apt/trusted.gpg.d/proxmox-release-bookworm.gpg"
    "proxmox-
ceph|http://download.proxmox.com/debian/ceph-squid|bookworm|no-subscription|
```

```
/etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg"
)

# 공통 설정
ARCH="amd64"
FINAL_REPO_NAME="pve-local"
TIMESTAMP=$(date +%Y%m%d-%H%M)
SNAPSHOT_PREFIX="snap-${TIMESTAMP}"
declare -a SNAPSHOT_LIST_TO_MERGE

# 오류 발생 시 스크립트 중단
set -e

echo "### [단계 1/7] GPG 공개 키 가져오기 ###"
for repo_info in "${REPO_LIST[@]}"; do
    IFS='|' read -r name url dists comps key_path <<< "$repo_info"
    if [ -f "$key_path" ]; then
        echo ">> '${name}' 리포지토리의 키를 가져옵니다..."
        gpg --no-default-keyring --keyring "$key_path" --export | gpg --no-
default-keyring --keyring ${GNUPGHOME}/trustedkeys.gpg --import
    else
        echo "!! 경고: 키 파일 '$key_path'을 찾을 수 없습니다."
    fi
done


echo -e "\n### [단계 2/7] 모든 미러 생성 (존재하지 않을 경우) ###"
for repo_info in "${REPO_LIST[@]}"; do
    IFS='|' read -r name url dists comps key_path <<< "$repo_info"
  # 미러가 존재하는지 확인
    if ! aptly mirror list -raw | grep -q "^${name}$"; then
        echo ">> '${name}' 미러를 새로 생성합니다..."
        CREATE_ARGS=(-filter-with-deps -architectures=${ARCH})
        if [[ "$name" == "debian-security" ]]; then
            CREATE_ARGS+=(-force-components)
        fi
        CREATE_ARGS+=("${name}" "${url}" "${dists}")
        CREATE_ARGS+=($comps)
        aptly mirror create "${CREATE_ARGS[@]}"
    else
        echo ">> '${name}' 미러는 이미 존재합니다."
    fi
done


echo -e "\n### [단계 3/7] 모든 미러 업데이트 (패키지 목록 다운로드) ###"
for repo_info in "${REPO_LIST[@]}"; do
    IFS='|' read -r name url dists comps key_path <<< "$repo_info"
    echo ">> '${name}' 미러를 업데이트합니다..."
    aptly mirror update "${name}"
```

```
done


echo -e "\n### [단계 4/7] 각 미러로부터 스냅샷 생성 ###"
for repo_info in "${REPO_LIST[@]}"; do
    IFS='|' read -r name url dists comps key_path <<< "$repo_info"
    snapshot_name="${SNAPSHOT_PREFIX}-${name}"
    echo ">> '${name}' 미러로부터 '${snapshot_name}' 스냅샷을 생성합니다..."
    aptly snapshot create "${snapshot_name}" from mirror "${name}"
    SNAPSHOT_LIST_TO_MERGE+=("${snapshot_name}")
done


echo -e "\n### [단계 5/7] 모든 스냅샷을 하나로 병합 (최신 버전 유지) ###"
final_snapshot_name="${FINAL_REPO_NAME}-${TIMESTAMP}"
echo ">> 모든 스냅샷을 '${final_snapshot_name}'으로 병합합니다 (-latest 적용)..."
aptly snapshot merge -latest "${final_snapshot_name}"
${SNAPSHOT_LIST_TO_MERGE[@]}


echo -e "\n### [단계 6/7] 최종 리포지토리 게시 또는 업데이트 ###"
if aptly publish list | grep -q " ${FINAL_REPO_NAME} "; then
    echo ">> 기존 '${FINAL_REPO_NAME}' 리포지토리를 업데이트합니다..."
    #aptly publish switch -force-overwrite -passphrase=""
"${FINAL_REPO_NAME}" "${final_snapshot_name}"
    aptly publish switch -force-overwrite -skip-signing "${FINAL_REPO_NAME}"
"${final_snapshot_name}"

else
    echo ">> 새 '${FINAL_REPO_NAME}' 리포지토리를 게시합니다..."
    #aptly publish snapshot -distribution="${FINAL_REPO_NAME}" -
passphrase="" "${final_snapshot_name}"
    aptly publish snapshot -distribution="${FINAL_REPO_NAME}" -skip-signing
"${final_snapshot_name}"
fi


echo -e "\n### [단계 7/7] 오래된 스냅샷 정리 (공간 확보) ###"
aptly snapshot list -sort=time | grep "snap-" | head -n -$((${#REPO_LIST[@]}
* 5)) | awk '{print $1}' | xargs -r aptly snapshot drop


echo -e "\n ### 작업 완료 ###"
echo "로컬 리포지토리가 성공적으로 생성/업데이트되었습니다."
echo "클라이언트의 sources.list에 아래 주소를 추가하여 사용하세요:"
echo "deb http://<YOUR_SERVER_IP>:8080/${FINAL_REPO_NAME} ${FINAL_REPO_NAME}
main"
```

clean-mirror.sh

```
#!/bin/bash

echo "### Aptly 전체 데이터 삭제를 시작합니다. ###"

# 오류가 발생해도 계속 진행
set +e

# --- 1. 게시된 리포지토리 삭제 ---
echo -e "\n>> 1. 게시된 모든 리포지토리를 삭제합니다..."
PUBLISHED_REPOS=$(aptly publish list -raw)
if [ -n "$PUBLISHED_REPOS" ]; then
    for repo in $PUBLISHED_REPOS; do
        echo "   - Deleting published repo: $repo"
        aptly publish drop "$repo"
    done
else
    echo "   - 게시된 리포지토리가 없습니다."
fi

# --- 2. 모든 스냅샷 삭제 ---
echo -e "\n>> 2. 모든 스냅샷을 삭제합니다..."
SNAPSHOTS=$(aptly snapshot list -raw)
if [ -n "$SNAPSHOTS" ]; then
    aptly snapshot drop $SNAPSHOTS
    echo "   - 모든 스냅샷이 삭제되었습니다."
else
    echo "   - 생성된 스냅샷이 없습니다."
fi

# --- 3. 모든 미러 삭제 ---
echo -e "\n>> 3. 모든 미러를 삭제합니다..."
MIRRORS=$(aptly mirror list -raw)
if [ -n "$MIRRORS" ]; then
    aptly mirror drop $MIRRORS
    echo "   - 모든 미러가 삭제되었습니다."
else
    echo "   - 생성된 미러가 없습니다."
fi

# --- 4. 데이터베이스 정리 ---
echo -e "\n>> 4. Aptly 데이터베이스를 정리합니다..."
aptly db cleanup

echo -e "\n### 원복 작업 완료 ###"
echo "모든 aptly 미러, 스냅샷, 게시된 리포지토리가 삭제되었습니다."
```

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/debian%EA%B3%84%EC%97%B4_%EC%B5%9C%EC%8B%A0_%EB%A0%88%ED%8F%AC%EC%A7%80%ED%86%A0%EB%A6%AC_%EB%AF%B8%EB%9F%AC_%EA%B5%AC%EC%84%B1

Last update: **2025/07/08 14:28**