Debian: Offline Package Installation

If you need to add services or packages to a VM that has no internet access to pull down packages, here's an offline method.

NOTE: These steps were performed and vetted when updating a template that was already deployed inside an isolated network (without internet access). See this page: Debian VM Template

Needed Elements

For this method, we need a few pieces to get the job done.

- Sudo access on the target VM (the one without internet access).
- Access to a surrogate VM that has internet visibility

> NOTE: In order to correctly retrieve packages that an offline VM requires, the surrogate VM needs to be the same Distro and very close to the same OS version as the target VM.
> This ensures that you fetch the correct secondary dependencies (of the target package) that are needed (by the same Distro/version).

- The ability to ferry a set of package files across the network boundary to the isolated, target VM.

Process Steps

For demonstrations purposes, we will attempt to perform an offline install of SubVersion on a Debian 13.4 VM, using a surrogate Debian 13.5 VM with internet access.

Surrogate VM

Log into the surrogate VM.

From an SSH session, create a download folder where you will download packages, and navigate to it.

```
mkdir -p ~/Desktop/temppkg/subversion
cd ~/Desktop/temppkg/subversion
```

Once inside the download folder, we will download the desired package and secondary dependencies.\

Apt-RDepends

> NOTE: Since any package that you want to install, has dependencies of its own, we will be downloading these secondary dependencies, as well.

> NOTE: Downloading dependency sets, requires apt-rdepends, which is not part of a normal OS build.
> This next step will install it.
> If you skip it, you will get an error message, like this:
> Command 'apt-rdepends' not found, but can be installed with: sudo apt install apt-rdepends

Before downloading the offline dependency set, you will need to ensure that apt-rdepends is installed on the surrogate VM.
Install apt-rdepends, with this:

```
sudo apt update
sudo apt install -y apt-rdepends
```

With apt-rdepends installed, we can safely download the target package and its secondary dependencies.

Package Download

The following command will download the subversion package and all of its dependencies needed for a Debian 13.4 OS.

```
apt-get download $(apt-rdepends subversion | grep -v "^ ")
```

> NOTE: You can change the package name, in the above command, to get the dependency set for any package.

Here's what the downloaded dependency set (for subversion) looks like, today:

```
glwhite@gitlab01:~/Desktop/temppkg/subversion$ ls -l
total 11304
-rw-r--r-- 1 glwhite glwhite   49432 Mar 15  2025 gcc-14-base_14.2.0-19_amd64.deb
-rw-r--r-- 1 glwhite glwhite  104232 Sep  7  2024 libaprlt64_1.7.5-1_amd64.deb
-rw-r--r-- 1 glwhite glwhite   89444 Jan 16  2025 libaprutillt64_1.6.3-3+b1_amd64.deb
-rw-r--r-- 1 glwhite glwhite 2847424 Mar  3 02:41 libc6_2.41-12+deb13u2_amd64.deb
-rw-r--r-- 1 glwhite glwhite   25044 Mar  9 19:04 libcom-err2_1.47.2-3+b10_amd64.deb
-rw-r--r-- 1 glwhite glwhite   90072 Jan 16  2025 libcrypt1_1%3a4.4.38-1_amd64.deb
-rw-r--r-- 1 glwhite glwhite  703676 Oct 15  2024 libdb5.3t64_5.3.28+dfsg2-9_amd64.deb
-rw-r--r-- 1 glwhite glwhite  108204 Jul 17  2025 libexpat1_2.7.1-2_amd64.deb
-rw-r--r-- 1 glwhite glwhite   72772 Mar 15  2025 libgcc-s1_14.2.0-19_amd64.deb
-rw-r--r-- 1 glwhite glwhite   75176 Aug  7  2024 libgdbm6t64_1.24-2_amd64.deb
-rw-r--r-- 1 glwhite glwhite  138232 Mar 13  2025 libgssapi-krb5-2_1.21.3-5_amd64.deb
-rw-r--r-- 1 glwhite glwhite   81540 Mar 13  2025 libk5crypto3_1.21.3-5_amd64.deb
-rw-r--r-- 1 glwhite glwhite    9456 Apr 13  2025 libkeyutils1_1.6.3-6_amd64.deb
-rw-r--r-- 1 glwhite glwhite  326308 Mar 13  2025 libkrb5-3_1.21.3-5_amd64.deb
-rw-r--r-- 1 glwhite glwhite   33036 Mar 13  2025 libkrb5support0_1.21.3-5_amd64.deb
-rw-r--r-- 1 glwhite glwhite   63900 Mar  7  2025 liblz4-1_1.10.0-4_amd64.deb
-rw-r--r-- 1 glwhite glwhite   57512 Feb 28  2025 libsasl2-2_2.1.28+dfsg1-9_amd64.deb
-rw-r--r-- 1 glwhite glwhite   19768 Feb 28  2025 libsasl2-modules-db_2.1.28+dfsg1-9_amd64.deb
-rw-r--r-- 1 glwhite glwhite   52992 Oct 29  2024 libserf-1-1_1.3.10-3+b1_amd64.deb
-rw-r--r-- 1 glwhite glwhite  914268 Mar  1 05:28 libsqlite3-0_3.46.1-7+deb13u1_amd64.deb
-rw-r--r-- 1 glwhite glwhite 2445272 Apr  3 11:53 libssl3t64_3.5.5-1~deb13u2_amd64.deb
-rw-r--r-- 1 glwhite glwhite 1422564 Apr  6  2025 libsvn1_1.14.5-3_amd64.deb
-rw-r--r-- 1 glwhite glwhite   60960 Oct 29  2024 libutf8proc3_2.9.0-1+b2_amd64.deb
-rw-r--r-- 1 glwhite glwhite   37612 May  9  2025 libuuid1_2.41-5_amd64.deb
-rw-r--r-- 1 glwhite glwhite   27096 Jan  4  2025 libxxhash0_0.8.3-2_amd64.deb
-rw-r--r-- 1 glwhite glwhite  303836 Mar 13  2025 libzstd1_1.5.7+dfsg-1_amd64.deb
-rw-r--r-- 1 glwhite glwhite  310940 Apr  3 11:53 openssl-provider-legacy_3.5.5-1~deb13u2_amd64.deb
-rw-r--r-- 1 glwhite glwhite  961896 Apr  6  2025 subversion_1.14.5-3_amd64.deb
-rw-r--r-- 1 glwhite glwhite   88892 Oct 29  2024 zlib1g_1%3a1.3.dfsg+really1.3.1-1+b1_amd64.deb
glwhite@gitlab01:~/Desktop/temppkg/subversion$
```

Target VM

Next, you need to move the dependency set over to the target VM, by whatever means you have. For me, I copied them out of the surrogate, via SCP.
Then, they were ferried across the network boundary as committed changes to an SVN repository (visible on both sides).

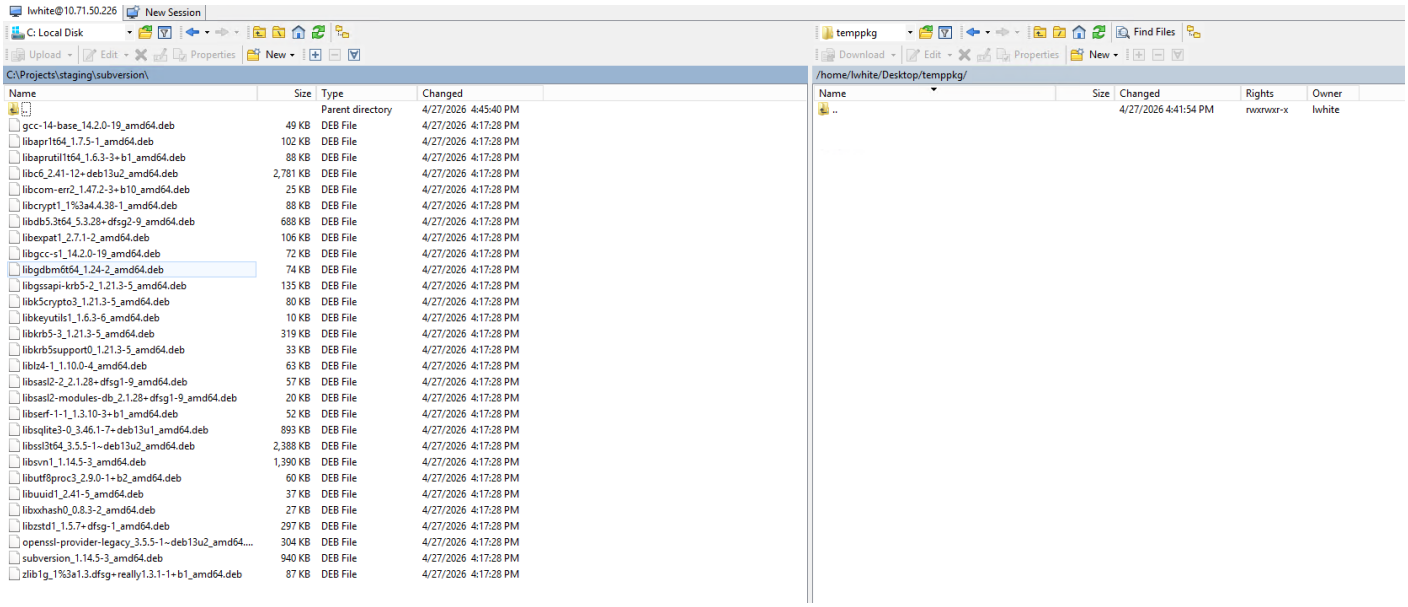
Next, log into the target VM, and create a download folder, where you will push the dependency set.

```
login as: lwhite
lwhite@10.71.50.226's password:
Linux localhost 6.12.74+deb13+1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.74-2 (2026-03-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
lwhite@localhost:~$ cd ~/Desktop/
lwhite@localhost:~/Desktop$ ls -l
total 4
drwxrwxr-x 4 lwhite lwhite 4096 Apr 24 19:09 dotnet-test
lwhite@localhost:~/Desktop$ mkdir ./temppkg
lwhite@localhost:~/Desktop$ cd temppkg/
lwhite@localhost:~/Desktop/temppkg$
```

Then, another SCP push to a temporary folder on the target VM.



Once the packages are on the target VM, open an SSH session to it, and navigate to the download folder.

Confirm the dependency set is there:

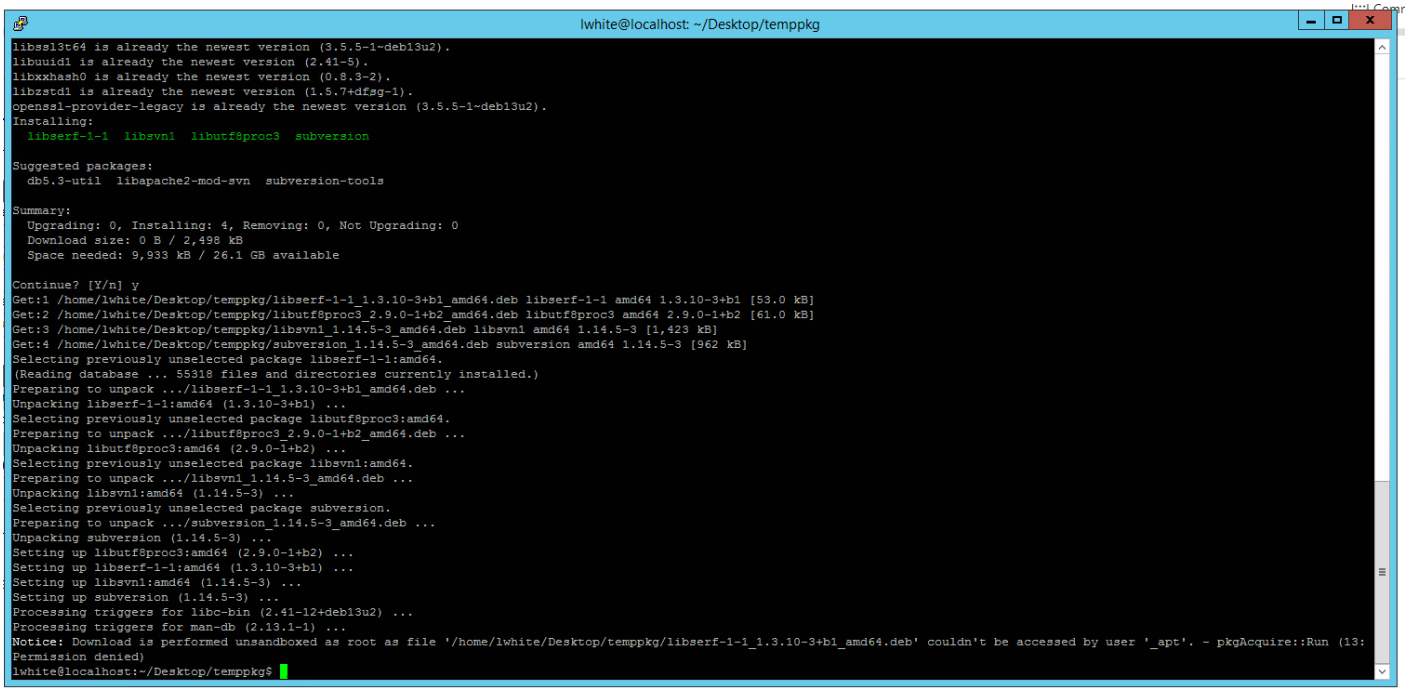
```
lwhite@localhost:~/Desktop$ mkdir ./temppkg
lwhite@localhost:~/Desktop$ cd temppkg/
lwhite@localhost:~/Desktop/temppkg$ ls -l
total 11304
-rw-r--r-- 1 lwhite lwhite   49432 Apr 27 16:17 gcc-14-base_14.2.0-19_amd64.deb
-rw-r--r-- 1 lwhite lwhite  104232 Apr 27 16:17 libapr1t64_1.7.5-1_amd64.deb
-rw-r--r-- 1 lwhite lwhite   89444 Apr 27 16:17 libaprutil1t64_1.6.3-3+b1_amd64.deb
-rw-r--r-- 1 lwhite lwhite 2847424 Apr 27 16:17 libc6_2.41-12+deb13u2_amd64.deb
-rw-r--r-- 1 lwhite lwhite   25044 Apr 27 16:17 libcom-err2_1.47.2-3+b10_amd64.deb
-rw-r--r-- 1 lwhite lwhite   90072 Apr 27 16:17 libcrypt1_1:4.4.38-1_amd64.deb
-rw-r--r-- 1 lwhite lwhite  703676 Apr 27 16:17 libdb5.3t64_5.3.28+dfsg2-9_amd64.deb
-rw-r--r-- 1 lwhite lwhite  108204 Apr 27 16:17 libexpat1_2.7.1-2_amd64.deb
-rw-r--r-- 1 lwhite lwhite   72772 Apr 27 16:17 libgcc-s1_14.2.0-19_amd64.deb
-rw-r--r-- 1 lwhite lwhite   75176 Apr 27 16:17 libgdbm6t64_1.24-2_amd64.deb
-rw-r--r-- 1 lwhite lwhite  138232 Apr 27 16:17 libgssapi-krb5-2_1.21.3-5_amd64.deb
-rw-r--r-- 1 lwhite lwhite   81540 Apr 27 16:17 libk5crypto3_1.21.3-5_amd64.deb
-rw-r--r-- 1 lwhite lwhite    9456 Apr 27 16:17 libkeyutils1_1.6.3-6_amd64.deb
-rw-r--r-- 1 lwhite lwhite  326308 Apr 27 16:17 libkrb5-3_1.21.3-5_amd64.deb
-rw-r--r-- 1 lwhite lwhite   33036 Apr 27 16:17 libkrb5support0_1.21.3-5_amd64.deb
-rw-r--r-- 1 lwhite lwhite   63900 Apr 27 16:17 liblz4-1_1.10.0-4_amd64.deb
-rw-r--r-- 1 lwhite lwhite   57512 Apr 27 16:17 libsasl2-2_2.1.28+dfsg1-9_amd64.deb
-rw-r--r-- 1 lwhite lwhite   19768 Apr 27 16:17 libsasl2-modules-db_2.1.28+dfsg1-9_amd64.deb
-rw-r--r-- 1 lwhite lwhite   52992 Apr 27 16:17 libserf-1-1_1.3.10-3+b1_amd64.deb
-rw-r--r-- 1 lwhite lwhite  914268 Apr 27 16:17 libsqlite3-0_3.46.1-7+deb13u1_amd64.deb
-rw-r--r-- 1 lwhite lwhite 2445272 Apr 27 16:17 libssl3t64_3.5.5-1~deb13u2_amd64.deb
-rw-r--r-- 1 lwhite lwhite 1422564 Apr 27 16:17 libsvn1_1.14.5-3_amd64.deb
-rw-r--r-- 1 lwhite lwhite   60960 Apr 27 16:17 libutf8proc3_2.9.0-1+b2_amd64.deb
-rw-r--r-- 1 lwhite lwhite   37612 Apr 27 16:17 libuuid1_2.41-5_amd64.deb
-rw-r--r-- 1 lwhite lwhite   27096 Apr 27 16:17 libxxhash0_0.8.3-2_amd64.deb
-rw-r--r-- 1 lwhite lwhite  303836 Apr 27 16:17 libzstd1_1.5.7+dfsg-1_amd64.deb
-rw-r--r-- 1 lwhite lwhite  310940 Apr 27 16:17 openssl-provider-legacy_3.5.5-1~deb13u2_amd64.deb
-rw-r--r-- 1 lwhite lwhite  961896 Apr 27 16:17 subversion_1.14.5-3_amd64.deb
-rw-r--r-- 1 lwhite lwhite   88892 Apr 27 16:17 zlib1g_1:1.3.dfsg+really1.3.1-1+b1_amd64.deb
lwhite@localhost:~/Desktop/temppkg$
```

> Warning: To prevent an apt elevation warning, read ahead on how to set permissions on the dependency folder, first.

From inside the folder with the dependencies, run this, to install all of them:

```
sudo apt install ./*.deb
```

The installation ran for a good bit, and finished.
But, it did have, what appears to be, an error.



At the tail of the installation scroll, you may see an `_apt` notice that looks like an error. See above.
That final notice is not a failure.
It just means APT could not read the local `.deb` files as the low-privilege `_apt` user, so it read them as root instead.

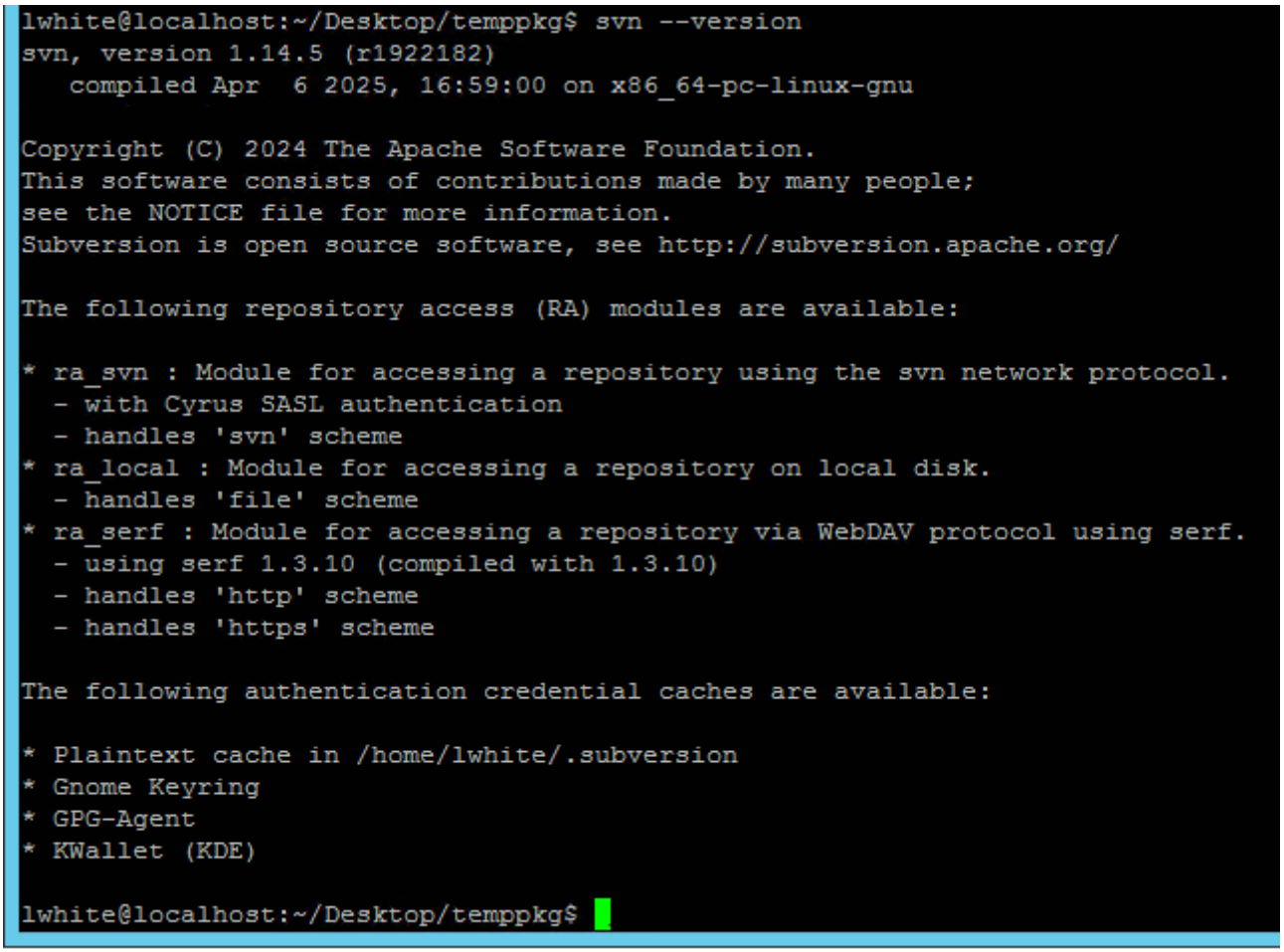
That notice can be prevented by making the folder/files world-readable before installing.
Run this to make the downloaded packages and folder world-readable:

```
chmod -R a+rX ~/Desktop/temppkg
```

Validation

Since we were installing SubVersion, we can confirm its installation, with this:

```
svn --version
```



Cleanup

Once installed, you can remove the packages and download folder, with this:

```
cd ~/Desktop
rm -rf ~/Desktop/temppkg
```

Revision #6
Created 27 April 2026 20:18:24 by glwhite
Updated 28 April 2026 13:12:08 by glwhite