

# Apt

- [Enable backports in Debian 11](#)
- [Enable testing repo in stable Debian 11](#)

# Enable backports in Debian

## 11

Sometimes the packages in Debian stable are way too old. In that case, you can use the backported repository to install packages in newer versions. It is not recommended to install every single package from the backported repo to keep your distro stable in the long term.

Instead only pick packages you need. Use the `-t` option with `apt` to force installation from the backported repo.

### Enable backports in Debian

Open `/etc/apt/sources.list` in your favorite editor.

```
$ sudo vi /etc/apt/sources.list
```

Add a line containing the following:

```
deb http://deb.debian.org/debian bullseye-backports main
```

Update apt configuration

```
$ sudo apt update
```

Check apt priority configuration and make sure the backports repo has lower priority than the other repositories.

```
$ apt-cache policy
```

```
Package files:
```

```
100 /var/lib/dpkg/status
```

```
release a=now
```

```
500 https://nginx.org/packages/mainline/debian bullseye/nginx amd64 Packages
```

```
release v=11.0,o=nginx,a=stable,n=bullseye,l=nginx,c=nginx,b=amd64
```

```
origin nginx.org
```

```
100 http://deb.debian.org/debian bullseye-backports/main amd64 Packages
```

```
release o=Debian Backports,a=bullseye-backports,n=bullseye-backports,l=Debian
Backports,c=main,b=amd64
origin deb.debian.org
500 http://security.debian.org/debian-security bullseye-security/main amd64 Packages
release v=11,o=Debian,a=stable-security,n=bullseye-security,l=Debian-Security,c=main,b=amd64
origin security.debian.org
500 http://deb.debian.org/debian bullseye/main amd64 Packages
release v=11.0,o=Debian,a=stable,n=bullseye,l=Debian,c=main,b=amd64
origin deb.debian.org
Pinned packages:
```

As you can see, the backported repo has lower number than the rest. It is a bit more complicated than that (you can check apt's man page), but for the purposes of this, let's just say that **lower means lower priority**.

# Enable testing repo in stable Debian 11

You want to keep a stable Debian 11 distro (install and upgrade all packages from stable by default), but there's a few packages you want to use that are in another repo (testing or unstable). Here's how to do it (probably correctly).

## Enable testing in Debian

Adding backports is easy, it's just another line in `/etc/apt/sources.list`. Unfortunately, it is not as simple as that with testing or unstable. For these to work properly (and not change all packages to unstable), we need to play a bit with repository priorities.

We will use something called Apt-Pinning, which you can read more about in the [Debian Handbook](#).

## Edit apt preferences file

Create apt preferences file (if it doesn't exist already) and open it with your favorite file editor.

```
$ sudo vi /etc/apt/preferences
```

Now add the following content into it:

```
Package: *  
Pin: release a=stable  
Pin-Priority: 900  
  
Package: *  
Pin: release o=Debian  
Pin-Priority: -10
```

This should give the stable repository high enough priority that all packages will be installed and upgraded from stable by default. To install package from other repo (testing or unstable), it has to be explicitly specified with `-t` option of apt.

## Replace bullseye with stable in sources.list

This is how my `sources.list` in `/etc/apt` looks like now:

```
# Generated by distrobuilder
deb http://deb.debian.org/debian bullseye main

deb http://security.debian.org/debian-security bullseye-security main

deb http://deb.debian.org/debian bullseye-updates main

# Official Nginx repo
deb https://nginx.org/packages/mainline/debian/ bullseye nginx
```

We can see that each line contains a reference to the release codename *bullseye*.

```
$ sudo vi /etc/apt/sources.list
```

Replace all references to *bullseye* with *stable* (except for Nginx) like this:

```
# Generated by distrobuilder
deb http://deb.debian.org/debian stable main

deb http://security.debian.org/debian-security stable-security main

deb http://deb.debian.org/debian stable-updates main

# Official Nginx repo
deb https://nginx.org/packages/mainline/debian/ bullseye nginx
```

**WARNING** – It is not recommended to use *stable* in the *sources.list*. *Stable* always refers to the stable repository of the **current** Debian version. Once the next Debian version is released in a few years and you upgrade your system, **it will break stuff**, because everything will suddenly update to the newest version. **Keep that in mind.**

Avoid using [stable](#) in your *sources.list* as that results in nasty surprises and broken systems when the next release is made; upgrading to a new release should be a deliberate, careful action and editing a file once every two years is not a burden.

## Update your system

```
$ sudo apt update
```

```
Get:1 http://security.debian.org/debian-security stable-security InRelease [44.1 kB]
Get:2 http://deb.debian.org/debian stable InRelease [113 kB]
Get:3 http://deb.debian.org/debian stable-updates InRelease [36.8 kB]
Get:4 http://security.debian.org/debian-security stable-security/main amd64 Packages [31.1 kB]
Get:5 http://security.debian.org/debian-security stable-security/main Translation-en [16.8 kB]
Get:6 http://deb.debian.org/debian stable/main amd64 Packages [8,178 kB]
Get:7 http://deb.debian.org/debian stable/main Translation-en [6,241 kB]
Hit:8 https://nginx.org/packages/mainline/debian bullseye InRelease
Fetched 14.7 MB in 7s (2,254 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

## Add testing lines to sources.list

```
$ sudo vi /etc/apt/sources.list
```

Copy the 3 lines with stable in them like this:

```
# Generated by distrobuilder
deb http://deb.debian.org/debian stable main
deb http://deb.debian.org/debian stable main

deb http://security.debian.org/debian-security stable-security main
deb http://security.debian.org/debian-security stable-security main

deb http://deb.debian.org/debian stable-updates main
deb http://deb.debian.org/debian stable-updates main

# Official Nginx repo
deb https://nginx.org/packages/mainline/debian/ bullseye nginx
```

and add testing to each of the other lines, like this:

```
# Generated by distrobuilder
deb http://deb.debian.org/debian stable main
deb http://deb.debian.org/debian testing main

deb http://security.debian.org/debian-security stable-security main
deb http://security.debian.org/debian-security testing-security main
```

```
deb http://deb.debian.org/debian stable-updates main
deb http://deb.debian.org/debian testing-updates main

# Official Nginx repo
deb https://nginx.org/packages/mainline/debian/ bullseye nginx
```

## Update your system again

```
$ sudo apt update
```

## Confirm the testing repo is working

Query search for a package that has different versions across stable and testing, e.g wireguard

```
$ apt-cache show wireguard
```

You can see that there are now 2 packages available, each at different version.

```
Package: wireguard
Version: 1.0.20210424-1
Installed-Size: 17
Maintainer: Daniel Kahn Gillmor <dkg@fifthhorseman.net>
Architecture: all
Depends: wireguard-modules (>= 0.0.20191219) | wireguard-dkms (>= 0.0.20200121-2), wireguard-tools (>= 1.0.20210424-1)
Description-en: fast, modern, secure kernel VPN tunnel (metapackage)
WireGuard is a novel VPN that runs inside the Linux Kernel and uses state-of-the-art cryptography (the "Noise" protocol). It aims to be faster, simpler, leaner, and more useful than IPSec, while avoiding the massive headache. It intends to be considerably more performant than OpenVPN. WireGuard is designed as a general purpose VPN for running on embedded interfaces and super computers alike, fit for many different circumstances. It runs over UDP.
.
This metapackage explicitly depends on both the kernel module and the userspace tooling.
Description-md5: bd6dd7a30cf34800a40219e3d2df9dc3
Homepage: https://www.wireguard.com
Section: net
Priority: optional
```

```
Filename: pool/main/w/wireguard/wireguard_1.0.20210424-1_all.deb
Size: 8196
MD5sum: fcf9917e3a6cc6c2588d4d0310d631be
SHA256: 4b19f03e77c3ab82e9510ef9b7abe1b35c39d509ba0a792328fd3b6f6e060adc

Package: wireguard
Version: 1.0.20210223-1
Installed-Size: 17
Maintainer: Daniel Kahn Gillmor <dkg@fifthhorseman.net>
Architecture: all
Depends: wireguard-modules (>= 0.0.20191219) | wireguard-dkms (>= 0.0.20200121-2), wireguard-tools (>= 1.0.20210223-1)
Description-en: fast, modern, secure kernel VPN tunnel (metapackage)
WireGuard is a novel VPN that runs inside the Linux Kernel and uses state-of-the-art cryptography (the "Noise" protocol). It aims to be faster, simpler, leaner, and more useful than IPSec, while avoiding the massive headache. It intends to be considerably more performant than OpenVPN. WireGuard is designed as a general purpose VPN for running on embedded interfaces and super computers alike, fit for many different circumstances. It runs over UDP.
.
This metapackage explicitly depends on both the kernel module and the userspace tooling.
Description-md5: bd6dd7a30cf34800a40219e3d2df9dc3
Homepage: https://www.wireguard.com
Section: net
Priority: optional
Filename: pool/main/w/wireguard/wireguard_1.0.20210223-1_all.deb
Size: 8164
MD5sum: fa92e03f62c6d6528cc770a9e97a141a
SHA256: d07b7f466a90a232f7a8a1750bbf7e5872555d41854789fc0eb6b4f0538b6ebf
```

## Confirm the correct apt priorities

We should also check if apt pulls the correct (stable) package by default and only installs from testing when explicitly requested. Run apt with `--dry-run` to show what would be installed without installing anything.

```
$ sudo apt install wireguard --dry-run
```

Comparing the output from `apt-cache show wireguard` to this confirms that apt is trying to install the stable version (20210223-1)

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  wireguard-tools
Suggested packages:
  openresolv | resolvconf
The following NEW packages will be installed:
  wireguard wireguard-tools
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Inst wireguard-tools (1.0.20210223-1 Debian:11.0/stable [amd64])
Inst wireguard (1.0.20210223-1 Debian:11.0/stable [all])
Conf wireguard-tools (1.0.20210223-1 Debian:11.0/stable [amd64])
Conf wireguard (1.0.20210223-1 Debian:11.0/stable [all])
```

## Try installing package from testing

Now install the same package from testing, by specifying the `-t` option and the name of the testing repository.

```
sudo apt install -t testing wireguard --dry-run
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  wireguard-tools
Suggested packages:
  openresolv | resolvconf
The following NEW packages will be installed:
  wireguard wireguard-tools
0 upgraded, 2 newly installed, 0 to remove and 163 not upgraded.
Inst wireguard-tools (1.0.20210424-1 Debian:testing [amd64])
Inst wireguard (1.0.20210424-1 Debian:testing [all])
Conf wireguard-tools (1.0.20210424-1 Debian:testing [amd64])
Conf wireguard (1.0.20210424-1 Debian:testing [all])
```

This would install version 20210424-1 from testing.