|  |
| --- |
| **Build VyOS**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-vyos) |

**Prerequisites**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#prerequisites)

There are different ways you can build VyOS.

Building using a [Docker](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-docker) container, although not the only way, is the easiest way as all

dependencies are managed for you. However, you can also set up your own build machine and

run a [Native Build](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-native).

|  |
| --- |
| **Note** |

|  |
| --- |
| Starting with VyOS 1.2 the release model of VyOS has changed. VyOS is now **free as in** |

|  |
| --- |
| **speech, but not as in beer**. This means that while VyOS is still an open source project, the |

|  |
| --- |
| release ISOs are no longer free and can only be obtained via subscription, or by contributing to |

|  |
| --- |
| the community. |

|  |
| --- |
| The source code remains public and an ISO can be built using the process outlined in this |

|  |
| --- |
| chapter. |

This will guide you through the process of building a VyOS ISO using [Docker](https://www.docker.com/). This process has

been tested on clean installs of Debian Jessie, Stretch, and Buster.

**Docker**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#docker)

Installing [Docker](https://www.docker.com/) and prerequisites:

|  |
| --- |
| $ sudo apt-get update  $ sudo apt-get install -y apt-transport-https ca-certificates curl gnupg2 software-properties-common $ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add - $ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb\_release -cs) stable"  $ sudo apt-get update  $ sudo apt-get install -y docker-ce |
| To be able to use [Docker](https://www.docker.com/) without sudo , the current non-root user must be added to   |  | | --- | |  |   the docker group by calling: sudo usermod -aG docker yourusername .   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| **Hint** |

|  |  |  |
| --- | --- | --- |
| Doing so grants privileges equivalent to the root user! It is recommended to remove the non-   |  | | --- | |  |   root user from the docker group after building the VyOS ISO. See also [Docker as non-root](https://docs.docker.com/engine/install/linux-postinstall).   |  | | --- | |  | |

|  |
| --- |
| **Note** |

|  |
| --- |
| The build process needs to be built on a local file system, building on SMB or NFS shares will |

|  |
| --- |
| result in the container failing to build properly! VirtualBox Drive Share is also not an option as |

|  |
| --- |
| block device operations are not implemented and the drive is always mounted as “nodev” |

**Build Container**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-container)

The container can be built by hand or by fetching the pre-built one from DockerHub. Using the

pre-built containers from the [VyOS DockerHub organisation](https://hub.docker.com/u/vyos) will ensure that the container is

always up-to-date. A rebuild is triggered once the container changes (please note this will take 2-

3 hours after pushing to the vyos-build repository).

Dockerhub[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#dockerhub)

To manually download the container from DockerHub, run:

|  |
| --- |
| $ docker pull vyos/vyos-build:crux # For VyOS 1.2  $ docker pull vyos/vyos-build:equuleus # For VyOS 1.3 $ docker pull vyos/vyos-build:sagitta # For VyOS 1.4 $ docker pull vyos/vyos-build:current # For rolling release |

Build from source[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-from-source)

The container can also be built directly from source:

|  |
| --- |
| # For VyOS 1.2 (crux)  $ git clone -b crux --single-branch https://github.com/vyos/vyos-build # For VyOS 1.3 (equuleus)  $ git clone -b equuleus --single-branch https://github.com/vyos/vyos-build # For VyOS 1.4 (sagitta)  $ git clone -b sagitta --single-branch https://github.com/vyos/vyos-build |

|  |
| --- |
| $ cd vyos-build  $ docker build -t vyos/vyos-build:crux docker # For VyOS 1.2  $ docker build -t vyos/vyos-build:current docker # For rolling release |

|  |
| --- |
| **Note** |
| Since VyOS has switched to Debian (11) Bullseye in its current branch, you will require   |  | | --- | |  | |

|  |
| --- |
| individual container for *current*, *equuleus* and *crux* builds. |

**Tips and Tricks**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#tips-and-tricks)

You can create yourself some handy Bash aliases to always launch the latest - per release train

|  |  |
| --- | --- |
| (*current* or *crux*) - container. Add the following to your .bash\_aliases file:   |  | | --- | |  | |

|  |
| --- |
| alias vybld='docker pull vyos/vyos-build:current && docker run --rm -it \ -v "$(pwd)":/vyos \  -v "$HOME/.gitconfig":/etc/gitconfig \  -v "$HOME/.bash\_aliases":/home/vyos\_bld/.bash\_aliases \  -v "$HOME/.bashrc":/home/vyos\_bld/.bashrc \  -w /vyos --privileged --sysctl net.ipv6.conf.lo.disable\_ipv6=0 \  -e GOSU\_UID=$(id -u) -e GOSU\_GID=$(id -g) \  vyos/vyos-build:current bash' |

|  |  |  |
| --- | --- | --- |
| alias vybld\_crux='docker pull vyos/vyos-build:crux && docker run --rm -it \ -v "$(pwd)":/vyos \  -v "$HOME/.gitconfig":/etc/gitconfig \  -v "$HOME/.bash\_aliases":/home/vyos\_bld/.bash\_aliases \  -v "$HOME/.bashrc":/home/vyos\_bld/.bashrc \  -w /vyos --privileged --sysctl net.ipv6.conf.lo.disable\_ipv6=0 \  -e GOSU\_UID=$(id -u) -e GOSU\_GID=$(id -g) \  vyos/vyos-build:crux bash'  Now you are prepared with two new aliases vybld and vybld\_crux to spawn your development   |  | | --- | |  |  |  | | --- | |  | |

containers in your current working directory.

|  |
| --- |
| **Note** |

|  |
| --- |
| Some VyOS packages (namely vyos-1x) come with build-time tests which verify some of the |

|  |  |  |  |
| --- | --- | --- | --- |
| internal library calls that they work as expected. Those tests are carried out through the Python  Unittest module. If you want to build the vyos-1x package (which is our main development   |  | | --- | |  |   package) you need to start your Docker container using the following argument: --   |  | | --- | |  |  |  | | --- | |  |   sysctl net.ipv6.conf.lo.disable\_ipv6=0 , otherwise those tests will fail. |

**Native Build**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#native-build)

To build VyOS natively you require a properly configured build host with the following Debian

versions installed:

 Debian Jessie for VyOS 1.2 (crux)

 Debian Buster for VyOS 1.3 (equuleus)

 Debian Bullseye for VyOS 1.4 (sagitta)

To start, clone the repository to your local machine:

|  |
| --- |
| # For VyOS 1.2 (crux)  $ git clone -b crux --single-branch https://github.com/vyos/vyos-build |

|  |
| --- |
| # For VyOS 1.3 (equuleus)  $ git clone -b equuleus --single-branch https://github.com/vyos/vyos-build |

|  |
| --- |
| # For VyOS 1.4 (sagitta)  $ git clone -b sagitta --single-branch https://github.com/vyos/vyos-build |
| For the packages required, you can refer to the docker/Dockerfile file in the [repository](https://github.com/vyos/vyos-build).   |  | | --- | |  |   The ./build-vyos-image script will also warn you if any dependencies are missing.   |  | | --- | |  | |

Once you have the required dependencies installed, you may proceed with the steps described in [Build ISO](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-iso).

**Build ISO**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-iso)

Now as you are aware of the prerequisites we can continue and build our own ISO from source. For this we have to fetch the latest source code from GitHub. Please note as this will differ for both *current* and *crux*.

|  |
| --- |
| # For VyOS 1.2 (crux)  $ git clone -b crux --single-branch https://github.com/vyos/vyos-build |

|  |
| --- |
| # For VyOS 1.3 (equuleus)  $ git clone -b equuleus --single-branch https://github.com/vyos/vyos-build |

|  |  |
| --- | --- |
| # For VyOS 1.4 (sagitta)  $ git clone -b sagitta --single-branch https://github.com/vyos/vyos-build  Now a fresh build of the VyOS ISO can begin. Change directory to the vyos-build directory and   |  | | --- | |  | |

run:

|  |
| --- |
| $ cd vyos-build  # For VyOS 1.2 (crux)  $ docker run --rm -it --privileged -v $(pwd):/vyos -w /vyos vyos/vyos-build:crux bash |

|  |
| --- |
| # For VyOS 1.3 (equuleus)  $ docker run --rm -it --privileged -v $(pwd):/vyos -w /vyos vyos/vyos-build:equuleus bash |

|  |
| --- |
| # For VyOS 1.4 (sagitta)  $ docker run --rm -it --privileged -v $(pwd):/vyos -w /vyos vyos/vyos-build:sagitta bash # For MacOS (crux, equuleus, sagitta)  $ git clone https://github.com/vyos/vyos-utils-misc  $ cd build-tools/macos-build |

|  |
| --- |
| # For VyOS 1.2 (crux)  $ os=jessie64 branch=crux make build |

|  |
| --- |
| # For VyOS 1.3 (equuleus)  $ os=buster64 branch=equuleus make build |

|  |
| --- |
| # For VyOS 1.4 (sagitta)  $ os=buster64 branch=sagitta make build |

Start the build:

|  |
| --- |
| # For VyOS 1.2 (crux) and VyOS 1.3 (equuleus) |

|  |
| --- |
| vyos\_bld@8153428c7e1f:/vyos$ ./configure --architecture amd64 --build-by "j.randomhacker@vyos.io" vyos\_bld@8153428c7e1f:/vyos$ sudo make iso |

|  |  |  |
| --- | --- | --- |
| # For VyOS 1.4 (sagitta)  vyos\_bld@8153428c7e1f:/vyos$ sudo make clean  vyos\_bld@8153428c7e1f:/vyos$ sudo ./build-vyos-image iso --architecture amd64 --build-by "j.randomhacker@vyos.io"  When the build is successful, the resulting iso can be found inside the build directory as live-   |  | | --- | |  |  |  | | --- | |  | |

|  |
| --- |
| image-[architecture].hybrid.iso . |

Good luck!

|  |
| --- |
| **Hint** |

|  |
| --- |
| Building VyOS on Windows WSL2 with Docker integrated into WSL2 will work like a charm. |

|  |
| --- |
| No problems are known so far! |

**Customize**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#customize)

This ISO can be customized with the following list of configure options. The full and current list

|  |  |
| --- | --- |
| can be generated with ./build-vyos-image --help :   |  | | --- | |  | |

|  |
| --- |
| $ vyos\_bld@8153428c7e1f:/vyos$ sudo ./build-vyos-image --help  I: Checking if packages required for VyOS image build are installed  usage: build-vyos-image [-h] [--architecture ARCHITECTURE]  [--build-by BUILD\_BY] [--debian-mirror DEBIAN\_MIRROR]  [--debian-security-mirror DEBIAN\_SECURITY\_MIRROR]  [--pbuilder-debian-mirror PBUILDER\_DEBIAN\_MIRROR]  [--vyos-mirror VYOS\_MIRROR] [--build-type BUILD\_TYPE]  [--version VERSION] [--build-comment BUILD\_COMMENT] [--debug] [--dry-run] [--custom-apt-entry CUSTOM\_APT\_ENTRY] [--custom-apt-key CUSTOM\_APT\_KEY] [--custom-package CUSTOM\_PACKAGE]  [build\_flavor] |

|  |
| --- |
| positional arguments:  build\_flavor Build flavor |

|  |
| --- |
| optional arguments:  -h, --help show this help message and exit  --architecture ARCHITECTURE  Image target architecture (amd64 or arm64)  --build-by BUILD\_BY Builder identifier (e.g. jrandomhacker@example.net) --debian-mirror DEBIAN\_MIRROR  Debian repository mirror  --debian-security-mirror DEBIAN\_SECURITY\_MIRROR  Debian security updates mirror  --pbuilder-debian-mirror PBUILDER\_DEBIAN\_MIRROR  Debian repository mirror for pbuilder env bootstrap  --vyos-mirror VYOS\_MIRROR  VyOS package mirror  --build-type BUILD\_TYPE  Build type, release or development  --version VERSION Version number (release builds only)  --build-comment BUILD\_COMMENT |

|  |
| --- |
| Optional build comment  --debug Enable debug output  --dry-run Check build configuration and exit  --custom-apt-entry CUSTOM\_APT\_ENTRY  Custom APT entry  --custom-apt-key CUSTOM\_APT\_KEY  Custom APT key file  --custom-package CUSTOM\_PACKAGE  Custom package to install from repositories |

**ISO Build Issues**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#iso-build-issues)

There are (rare) situations where building an ISO image is not possible at all due to a broken package feed in the background. APT is not very good at reporting the root cause of the issue.

Your ISO build will likely fail with a more or less similar looking error message:

|  |
| --- |
| The following packages have unmet dependencies:  vyos-1x : Depends: accel-ppp but it is not installable  E: Unable to correct problems, you have held broken packages. P: Begin unmounting filesystems...  P: Saving caches...  Reading package lists...  Building dependency tree...  Reading state information...  Del frr-pythontools 7.5-20210215-00-g8a5d3b7cd-0 [38.9 kB] Del accel-ppp 1.12.0-95-g59f8e1b [475 kB]  Del frr 7.5-20210215-00-g8a5d3b7cd-0 [2671 kB]  Del frr-snmp 7.5-20210215-00-g8a5d3b7cd-0 [55.1 kB]  Del frr-rpki-rtrlib 7.5-20210215-00-g8a5d3b7cd-0 [37.3 kB] make: \*\*\* [Makefile:30: iso] Error 1  (10:13) vyos\_bld ece068908a5b:/vyos [current] # |

To debug the build process and gain additional information of what could be the root cause, you need to use *chroot* to change into the build directry. This is explained in the following step by step procedure:

|  |
| --- |
| vyos\_bld ece068908a5b:/vyos [current] # sudo chroot build/chroot /bin/bash |

We now need to mount some required, volatile filesystems

|  |
| --- |
| (live)root@ece068908a5b:/# mount -t proc none /proc (live)root@ece068908a5b:/# mount -t sysfs none /sys (live)root@ece068908a5b:/# mount -t devtmpfs none /dev |

We now are free to run any command we would like to use for debugging, e.g. re-installing the failed package after updating the repository.

|  |
| --- |
| (live)root@ece068908a5b:/# apt-get update; apt-get install vyos-1x Get:1 file:/root/packages ./ InRelease  Ign:1 file:/root/packages ./ InRelease  Get:2 file:/root/packages ./ Release [1235 B]  Get:2 file:/root/packages ./ Release [1235 B]  Get:3 file:/root/packages ./ Release.gpg  Ign:3 file:/root/packages ./ Release.gpg  Hit:4 http://repo.powerdns.com/debian buster-rec-43 InRelease |

|  |
| --- |
| Hit:5 http://repo.saltstack.com/py3/debian/10/amd64/archive/3002.2 buster InRelease Hit:6 http://deb.debian.org/debian bullseye InRelease  Hit:7 http://deb.debian.org/debian buster InRelease  Hit:8 http://deb.debian.org/debian-security buster/updates InRelease  Hit:9 http://deb.debian.org/debian buster-updates InRelease  Hit:10 http://deb.debian.org/debian buster-backports InRelease  Hit:11 http://dev.packages.vyos.net/repositories/current current InRelease  Reading package lists... Done  N: Download is performed unsandboxed as root as file '/root/packages/./InRelease' couldn't be accessed by user '\_apt'. - pkgAcquire::Run (13: Permission denied)  Reading package lists... Done  Building dependency tree  Reading state information... Done  Some packages could not be installed. This may mean that you have  requested an impossible situation or if you are using the unstable  distribution that some required packages have not yet been created  or been moved out of Incoming.  The following information may help to resolve the situation: |

|  |
| --- |
| The following packages have unmet dependencies:  vyos-1x : Depends: accel-ppp but it is not installable  E: Unable to correct problems, you have held broken packages. |

Now it’s time to fix the package mirror and rerun the last step until the package installation

succeeds again!

**Linux Kernel**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#linux-kernel)

The Linux kernel used by VyOS is heavily tied to the ISO build process. The

|  |  |  |  |
| --- | --- | --- | --- |
| file data/defaults.json hosts a JSON definition of the kernel version used kernel\_version and   |  | | --- | |  |  |  | | --- | |  |   the kernel\_flavor of the kernel which represents the kernel’s LOCAL\_VERSION. Both together   |  | | --- | |  | |

form the kernel version variable in the system:

|  |  |
| --- | --- |
| vyos@vyos:~$ uname -r  6.1.52-amd64-vyos | |
|  | Accel-PPP |
|  | Intel NIC drivers |
|  | Inter QAT |

Each of those modules holds a dependency on the kernel version and if you are lucky enough to

receive an ISO build error which sounds like:

|  |
| --- |
| I: Create initramfs if it does not exist.  Extra argument '6.1.52-amd64-vyos'  Usage: update-initramfs {-c|-d|-u} [-k version] [-v] [-b directory] Options:  -k version Specify kernel version or 'all'  -c Create a new initramfs  -u Update an existing initramfs  -d Remove an existing initramfs  -b directory Set alternate boot directory  -v Be verbose |

|  |
| --- |
| See update-initramfs(8) for further details.  E: config/hooks/live/17-gen\_initramfs.chroot failed (exit non-zero). You should check for errors. |

The most obvious reasons could be:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | | --- | |  |   vyos-build repo is outdated, please git pull to update to the latest release kernel   |  | | --- | |  | |

version from us.

 You have your own custom kernel *\*.deb* packages in the *packages* folder but

neglected to create all required out-of tree modules like Accel-PPP, Intel QAT or

Intel NIC drivers   
**Building The Kernel**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#building-the-kernel)

The kernel build is quite easy, most of the required steps can be found in

|  |  |
| --- | --- |
| the vyos-build/packages/linux-kernel/Jenkinsfile but we will walk you through it.   |  | | --- | |  | |

Clone the kernel source to *vyos-build/packages/linux-kernel/*:

|  |
| --- |
| $ cd vyos-build/packages/linux-kernel/  $ git clone https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git |
| Check out the required kernel version - see vyos-build/data/defaults.json file (example uses   |  | | --- | |  | |

kernel 4.19.146):

|  |
| --- |
| $ cd vyos-build/packages/linux-kernel/linux  $ git checkout v4.19.146  Checking out files: 100% (61536/61536), done.  Note: checking out 'v4.19.146'. |

|  |
| --- |
| You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout. |

|  |
| --- |
| If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example: |

|  |
| --- |
| git checkout -b <new-branch-name> |

|  |  |
| --- | --- |
| HEAD is now at 015e94d0e37b Linux 4.19.146  Now we can use the helper script build-kernel.sh which does all the necessary voodoo by   |  | | --- | |  | |

applying required patches from the *vyos-build/packages/linux-kernel/patches* folder, copying our

|  |  |
| --- | --- |
| kernel configuration x86\_64\_vyos\_defconfig to the right location, and finally building the Debian   |  | | --- | |  | |

packages.

|  |
| --- |
| **Note** |

|  |
| --- |
| Building the kernel will take some time depending on the speed and quantity of your CPU/cores |

|  |
| --- |
| and disk speed. Expect 20 minutes (or even longer) on lower end hardware. |
| (18:59) vyos\_bld 412374ca36b8:/vyos/vyos-build/packages/linux-kernel [current] # ./build-kernel.sh I: Copy Kernel config (x86\_64\_vyos\_defconfig) to Kernel Source  I: Apply Kernel patch: /vyos/vyos-build/packages/linux-kernel/patches/kernel/0001-VyOS-Add-linkstate-IP-device-attribute.patch  patching file Documentation/networking/ip-sysctl.txt  patching file include/linux/inetdevice.h  patching file include/linux/ipv6.h  patching file include/uapi/linux/ip.h  patching file include/uapi/linux/ipv6.h  patching file net/ipv4/devinet.c  Hunk #1 succeeded at 2319 (offset 1 line).  patching file net/ipv6/addrconf.c  patching file net/ipv6/route.c  I: Apply Kernel patch: /vyos/vyos-build/packages/linux-kernel/patches/kernel/0002-VyOS-add-inotify-support-for-stackable-filesystems-o.patch  patching file fs/notify/inotify/Kconfig  patching file fs/notify/inotify/inotify\_user.c  patching file fs/overlayfs/super.c  Hunk #2 succeeded at 1713 (offset 9 lines).  Hunk #3 succeeded at 1739 (offset 9 lines).  Hunk #4 succeeded at 1762 (offset 9 lines).  patching file include/linux/inotify.h  I: Apply Kernel patch: /vyos/vyos-build/packages/linux-kernel/patches/kernel/0003-RFC-builddeb-add-linux-tools-package-with-perf.patch  patching file scripts/package/builddeb  I: make x86\_64\_vyos\_defconfig  HOSTCC scripts/basic/fixdep  HOSTCC scripts/kconfig/conf.o  YACC scripts/kconfig/zconf.tab.c  LEX scripts/kconfig/zconf.lex.c  HOSTCC scripts/kconfig/zconf.tab.o  HOSTLD scripts/kconfig/conf  #  # configuration written to .config  #  I: Generate environment file containing Kernel variable  I: Build Debian Kernel package  UPD include/config/kernel.release  /bin/sh ./scripts/package/mkdebian  dpkg-buildpackage -r"fakeroot -u" -a$(cat debian/arch) -b -nc -uc  dpkg-buildpackage: info: source package linux-4.19.146-amd64-vyos  dpkg-buildpackage: info: source version 4.19.146-1  dpkg-buildpackage: info: source distribution buster  dpkg-buildpackage: info: source changed by vyos\_bld <christian@poessinger.com>  dpkg-buildpackage: info: host architecture amd64  dpkg-buildpackage: warning: debian/rules is not executable; fixing that  dpkg-source --before-build .  debian/rules build  make KERNELRELEASE=4.19.146-amd64-vyos ARCH=x86 KBUILD\_BUILD\_VERSION=1 KBUILD\_SRC=  SYSTBL arch/x86/include/generated/asm/syscalls\_32.h |

...

dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory

|  |
| --- |
| dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: binaries to analyze should already be installed in their package's directory dpkg-shlibdeps: warning: package could avoid a useless dependency if  /vyos/vyos-build/packages/linux-kernel/linux/debian/toolstmp/usr/bin/trace  /vyos/vyos-build/packages/linux-kernel/linux/debian/toolstmp/usr/bin/perf were not linked against libcrypto.so.1.1 (they use none of the library's symbols)  dpkg-shlibdeps: warning: package could avoid a useless dependency if  /vyos/vyos-build/packages/linux-kernel/linux/debian/toolstmp/usr/bin/trace  /vyos/vyos-build/packages/linux-kernel/linux/debian/toolstmp/usr/bin/perf were not linked against libcrypt.so.1 (they use none of the library's symbols)  dpkg-deb: building package 'linux-tools-4.19.146-amd64-vyos' in '../linux-tools-4.19.146-amd64-vyos\_4.19.146-1\_amd64.deb'.  dpkg-genbuildinfo --build=binary  dpkg-genchanges --build=binary >../linux-4.19.146-amd64-vyos\_4.19.146-1\_amd64.changes dpkg-genchanges: warning: package linux-image-4.19.146-amd64-vyos-dbg in control file but not in files list  dpkg-genchanges: info: binary-only upload (no source code included)  dpkg-source --after-build .  dpkg-buildpackage: info: binary-only upload (no source included) |

In the end you will be presented with the kernel binary packages which you can then use in your custom ISO build process, by placing all the *\*.deb* files in the vyos-build/packages folder where they will be used automatically when building VyOS as documented above.

Firmware[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#firmware)

If you upgrade your kernel or include new drivers you may need new firmware. Build a

|  |  |
| --- | --- |
| new vyos-linux-firmware package with the included helper scripts.   |  | | --- | |  | |

|  |
| --- |
| $ cd vyos-build/packages/linux-kernel  $ git clone https://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git $ ./build-linux-firmware.sh  $ cp vyos-linux-firmware\_\*.deb ../ |

This tries to automatically detect which blobs are needed based on which drivers were built. If it

|  |  |  |
| --- | --- | --- |
| fails to find the correct files you can add them manually to vyos-build/packages/linux-kernel/build-   |  | | --- | |  |  |  | | --- | |  |   linux-firmware.sh : |

|  |
| --- |
| ADD\_FW\_FILES="iwlwifi\* ath11k/QCA6390/\*/\*.bin" |

**Building Out-Of-Tree Modules**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#building-out-of-tree-modules)

Building the kernel is one part, but now you also need to build the required out-of-tree modules

|  |  |
| --- | --- |
| so everything is lined up and the ABIs match. To do so, you can again take a look at vyos-   |  | | --- | |  | |

|  |  |
| --- | --- |
| |  | | --- | |  |   build/packages/linux-kernel/Jenkinsfile to see all of the required modules and their selected |

versions. We will show you how to build all the current required modules.

Accel-PPP[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#accel-ppp)

First, clone the source code and check out the appropriate version by running:

|  |
| --- |
| $ cd vyos-build/packages/linux-kernel  $ git clone https://github.com/accel-ppp/accel-ppp.git |

We again make use of a helper script and some patches to make the build work. Just run the following command:

|  |
| --- |
| $ ./build-accel-ppp.sh  I: Build Accel-PPP Debian package  CMake Deprecation Warning at CMakeLists.txt:3 (cmake\_policy):  The OLD behavior for policy CMP0003 will be removed from a future version of CMake. |

|  |
| --- |
| The cmake-policies(7) manual explains that the OLD behaviors of all policies are deprecated and that a policy should be set to OLD only under specific short-term circumstances. Projects should be ported to the NEW behavior and not rely on setting a policy to OLD. |

|  |
| --- |
| -- The C compiler identification is GNU 8.3.0 |

|  |
| --- |
| ... |

|  |  |  |  |
| --- | --- | --- | --- |
| CPack: Create package using DEB  CPack: Install projects  CPack: - Run preinstall target for: accel-ppp  CPack: - Install project: accel-ppp  CPack: Create package  CPack: - package: /vyos/vyos-build/packages/linux-kernel/accel-ppp/build/accel-ppp.deb generated.  After compiling the packages you will find yourself the newly generated *\*.deb* binaries in vyos-   |  | | --- | |  |  |  | | --- | |  |   build/packages/linux-kernel from which you can copy them to the vyos-build/packages folder for   |  | | --- | |  | |

inclusion during the ISO build.

Intel NIC[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#intel-nic)

The Intel NIC drivers do not come from a Git repository, instead we just fetch the tarballs from our mirror and compile them.

Simply use our wrapper script to build all of the driver modules.

|  |
| --- |
| ./build-intel-drivers.sh  % Total % Received % Xferd Average Speed Time Time Time Current Dload Upload Total Spent Left Speed |

|  |
| --- |
| 100 490k 100 490k 0 0 648k 0 --:--:-- --:--:-- --:--:-- 648k I: Compile Kernel module for Intel ixgbe driver |

|  |
| --- |
| ... |

|  |  |  |  |
| --- | --- | --- | --- |
| I: Building Debian package vyos-intel-iavf  Doing `require 'backports'` is deprecated and will not load any backport in the next major release. Require just the needed backports instead, or 'backports/latest'.  Debian packaging tools generally labels all files in /etc as config files, as mandated by policy, so fpm defaults to this behavior for deb packages. You can disable this default behavior with --deb-no-default-config-files flag {:level=>:warn}  Created package {:path=>"vyos-intel-iavf\_4.0.1-0\_amd64.deb"}  I: Cleanup iavf source  After compiling the packages you will find yourself the newly generated *\*.deb* binaries in vyos-   |  | | --- | |  |  |  | | --- | |  |   build/packages/linux-kernel from which you can copy them to the vyos-build/packages folder for   |  | | --- | |  | |

inclusion during the ISO build.

Intel QAT[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#intel-qat)

The Intel QAT (Quick Assist Technology) drivers do not come from a Git repository, instead we just fetch the tarballs from 01.org, Intel’s open-source website.

Simply use our wrapper script to build all of the driver modules.

|  |
| --- |
| $ ./build-intel-qat.sh  % Total % Received % Xferd Average Speed Time Time Time Current Dload Upload Total Spent Left Speed  100 5065k 100 5065k 0 0 1157k 0 0:00:04 0:00:04 --:--:-- 1157k I: Compile Kernel module for Intel qat driver  checking for a BSD-compatible install... /usr/bin/install -c  checking whether build environment is sane... yes  checking for a thread-safe mkdir -p... /bin/mkdir -p  checking for gawk... gawk  checking whether make sets $(MAKE)... yes |

|  |
| --- |
| ... |

|  |  |  |  |
| --- | --- | --- | --- |
| I: Building Debian package vyos-intel-qat  Doing `require 'backports'` is deprecated and will not load any backport in the next major release. Require just the needed backports instead, or 'backports/latest'.  Debian packaging tools generally labels all files in /etc as config files, as mandated by policy, so fpm defaults to this behavior for deb packages. You can disable this default behavior with --deb-no-default-config-files flag {:level=>:warn}  Created package {:path=>"vyos-intel-qat\_1.7.l.4.9.0-00008-0\_amd64.deb"}  I: Cleanup qat source  After compiling the packages you will find yourself the newly generated *\*.deb* binaries in vyos-   |  | | --- | |  |  |  | | --- | |  |   build/packages/linux-kernel from which you can copy them to the vyos-build/packages folder for   |  | | --- | |  | |

inclusion during the ISO build.

**Packages**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#packages)

If you are brave enough to build yourself an ISO image containing any modified package from our GitHub organisation - this is the place to be.

Any “modified” package may refer to an altered version of e.g. vyos-1x package that you would like to test before filing a pull request on GitHub.

Building an ISO with any customized package is in no way different than building a regular (customized or not) ISO image. Simply place your modified *\*.deb* package inside   
the *packages* folder within *vyos-build*. The build process will then pickup your custom package and integrate it into your ISO.

**Troubleshooting**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#troubleshooting)

Debian APT is not very verbose when it comes to errors. If your ISO build breaks for whatever reason and you suspect it’s a problem with APT dependencies or installation you can add this small patch which increases the APT verbosity during ISO build.

|  |
| --- |
| **diff --git i/scripts/live-build-config w/scripts/live-build-config**  **index 1b3b454..3696e4e 100755** --- i/scripts/live-build-config  +++ w/scripts/live-build-config  **@@ -57,7 +57,8 @@ lb config noauto \**  --firmware-binary false \  --updates true \  --security true \ - --apt-options "--yes -oAcquire::Check-Valid-Until=false" \  + --apt-options "--yes -oAcquire::Check-Valid-Until=false -oDebug::BuildDeps=true -oDebug::pkgDepCache::AutoInstall=true \  + -oDebug::pkgDepCache::Marker=true -oDebug::pkgProblemResolver=true -oDebug::Acquire::gpgv=true" \  --apt-indices false  "${@}"  """ |

**Virtualization Platforms**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#virtualization-platforms)

**QEMU**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#qemu)

Run the following command after building the ISO image.

|  |
| --- |
| $ make qemu |

**VMware**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#vmware)

Run the following command after building the QEMU image.

|  |
| --- |
| $ make vmware |

**Packages**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-packages)

VyOS itself comes with a bunch of packages that are specific to our system and thus cannot be found in any Debian mirror. Those packages can be found at the [VyOS GitHub project](https://github.com/vyos) in their source format can easily be compiled into a custom Debian (*\*.deb*) package.

The easiest way to compile your package is with the above mentioned [Docker](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-docker) container, it includes all required dependencies for all VyOS related packages.

Assume we want to build the vyos-1x package on our own and modify it to our needs. We first need to clone the repository from GitHub.

|  |
| --- |
| $ git clone https://github.com/vyos/vyos-1x |

**Build**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#id5)

Launch Docker container and build package

|  |
| --- |
| # For VyOS 1.3 (equuleus, current)  $ docker run --rm -it --privileged -v $(pwd):/vyos -w /vyos vyos/vyos-build:current bash |

|  |
| --- |
| # Change to source directory  $ cd vyos-1x |

|  |
| --- |
| # Build DEB  $ dpkg-buildpackage -uc -us -tc -b |

After a minute or two you will find the generated DEB packages next to the vyos-1x source directory:

|  |
| --- |
| # ls -al ../vyos-1x\*.deb -rw-r--r-- 1 vyos\_bld vyos\_bld 567420 Aug 3 12:01 ../vyos-1x\_1.3dev0-1847-gb6dcb0a8\_all.deb-rw-r--r-- 1 vyos\_bld vyos\_bld 3808 Aug 3 12:01 ../vyos-1x-vmware\_1.3dev0-1847- gb6dcb0a8\_amd64.deb |

**Install**[](https://docs.vyos.io/en/latest/contributing/build-vyos.html#install)

To take your newly created package on a test drive you can simply SCP it to a running VyOS instance and install the new *\*.deb* package over the current running one.

Just install using the following commands:

|  |
| --- |
| vyos@vyos:~$ dpkg --install /tmp/vyos-1x\_1.3dev0-1847-gb6dcb0a8\_all.deb (Reading database ... 58209 files and directories currently installed.) |

|  |
| --- |
| Preparing to unpack .../vyos-1x\_1.3dev0-1847-gb6dcb0a8\_all.deb ...  Unpacking vyos-1x (1.3dev0-1847-gb6dcb0a8) over (1.3dev0-1847-gb6dcb0a8) ... Setting up vyos-1x (1.3dev0-1847-gb6dcb0a8) ...  Processing triggers for rsyslog (8.1901.0-1) ... |

You can also place the generated *\*.deb* into your ISO build environment to include it in a custom iso, see [Linux Kernel](https://docs.vyos.io/en/latest/contributing/build-vyos.html#build-custom-packages) for more information.

|  |
| --- |
| **Warning** |

|  |
| --- |
| Any packages in the packages directory will be added to the iso during build, replacing the |

|  |
| --- |
| upstream ones. Make sure you delete them (both the source directories and built deb packages) if |

|  |
| --- |
| you want to build an iso from purely upstream packages. |