

Debian

- [System Einstellungen](#)
- [bash](#)
- [Linksammlung Docs](#)
- [fail2ban](#)
- [Debian Upgrade auf neuen Release](#)
- [missing firmware](#)
- [Backports](#)
- [Kernel updaten](#)
- [Paket selber bauen - dpkg-buildpackage](#)
- [MySQL - MariaDB](#)
- [Paketmanager - apt](#)

System Einstellungen

Root Passwort zurücksetzen

Quelle: https://www.thomas-krenn.com/de/wiki/Linux_Root_Passwort_wiederherstellen

Standard Editor ändern

Der Befehl sieht so aus: `update-alternatives --config editor`

Die Ausgabe dazu:

```
root@adminsrv: /# update-alternatives --config editor
There are 9 choices for the alternative editor (providing /usr/bin/editor).
```

Selection	Path	Priority	Status

* 0	/usr/bin/joe	70	auto mode
1	/bin/nano	40	manual mode
2	/usr/bin/jmacs	50	manual mode
3	/usr/bin/joe	70	manual mode
4	/usr/bin/jpico	50	manual mode
5	/usr/bin/jstar	50	manual mode
6	/usr/bin/mcedit	25	manual mode
7	/usr/bin/rjoe	25	manual mode
8	/usr/bin/vim.basic	30	manual mode
9	/usr/bin/vim.tiny	15	manual mode

```
Press <enter> to keep the current choice[*], or type selection number: 6
```

Ich benutze am liebsten den Editor vom Paket "mc", das MidnightCommander heißt.

MC - internen Editor benutzen

Quelle: <https://askubuntu.com/a/16782>

Zeitzone

Quelle: <https://wiki.debian.org/TimeZoneChanges>

bash

Arbeiten mit Variablen

Arrays

Array erstellen

```
# Array mit virtuellen Maschinen anlegen
machines=( vm01, vm02, vm03)

# Erstellt eine Liste mit den virtuellen Maschinen
IFS=';' read -r -a arrhosts <<< $(echo $machines | tr ',' ';')
```

Array iterieren

```
echo "Hier eine Liste der virtuellen Maschinen : "
i=0
for vm in "${arrhosts[@]}"
do
    echo $i" - "$vm
    i=$((i+1))
done

#Ausgabe:
0 - vm01
2 - vm02
3 - vm03
```

Scripte mit Parametern

Beispiel mit Abfrage eines bestimmten Wertes der Parametern

<https://stackoverflow.com/a/16496491>

```
#!/bin/bash

usage() { echo "Usage: $0 [-s <45| 90>] [-p <string>]" 1>&2; exit 1; }

while getopts ":s:p:" o; do
    case "${o}" in
        s)
            s=${OPTARG}
            (( s == 45 || s == 90 )) || usage
            ;;
        p)
            p=${OPTARG}
            ;;
        *)
            usage
            ;;
    esac
done
shift $(( OPTIND-1 ))

if [ -z "${s}" ] || [ -z "${p}" ]; then
    usage
fi

echo "s = ${s}"
echo "p = ${p}"
```

Ausgabe:

```
$ ./myscript.sh
Usage: ./myscript.sh [-s <45| 90>] [-p <string>]

$ ./myscript.sh -h
Usage: ./myscript.sh [-s <45| 90>] [-p <string>]
```

```
$ ./myscript.sh -s "" -p ""
Usage: ./myscript.sh [-s <45| 90>] [-p <string>]

$ ./myscript.sh -s 10 -p foo
Usage: ./myscript.sh [-s <45| 90>] [-p <string>]

$ ./myscript.sh -s 45 -p foo
s = 45
p = foo

$ ./myscript.sh -s 90 -p bar
s = 90
p = bar
```

Beispiel mit getopt

<http://mywiki.woledge.org/BashFAQ/035>

```
#!/bin/sh
# Usage info
show_help() {
cat << EOF
Usage: ${0##*/} [-hv] [-f OUTFILE] [FILE]...
Do stuff with FILE and write the result to standard output. With no FILE
or when FILE is -, read standard input.

-h          display this help and exit
-f OUTFILE  write the result to OUTFILE instead of standard output.
-v          verbose mode. Can be used multiple times for increased
            verbosity.

EOF
}

# Initialize our own variables:
output_file=""
verbose=0

OPTIND=1

# Resetting OPTIND is necessary if getopt was used previously in the script.
# It is a good idea to make OPTIND local if you process options in a function.
```

```

while getopts hvf: opt; do
    case $opt in
        h)
            show_help
            exit 0
            ;;
        v)  verbose=$(( verbose+1))
            ;;
        f)  output_file=$OPTARG
            ;;
        *)
            show_help >&2
            exit 1
            ;;
    esac
done
shift "$((OPTIND-1))"    # Discard the options and sentinel --

# Everything that's left in "$@" is a non-option.  In our case, a FILE to process.
printf ' verbose=<%d>\noutput_file=<%s>\nLeftovers: \n' "$verbose" "$output_file"
printf ' <%s>\n' "$@"

# End of file

```

Erstellen von Passwoertern

Quelle: <https://www.howtogeek.com/30184/10-ways-to-generate-a-random-password-from-the-command-line/>

Generate a Random Password

For any of these random password commands, you can either modify them to output a different password length, or you can just use the first x characters of the generated password if you don't want such a long password. Hopefully you're using a password manager like [LastPass](#) anyway so you don't need to memorize them.

This method uses SHA to hash the date, runs through base64, and then outputs the top 32 characters.

```
“ date +%s | sha256sum | base64 | head -c 32 ; echo
```

This method used the built-in /dev/urandom feature, and filters out only characters that you would normally use in a password. Then it outputs the top 32.

```
“ < /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c${1:-32};echo;
```

This one uses openssl's rand function, which may not be installed on your system. Good thing there's lots of other examples, right?

```
“ openssl rand -base64 32
```

This one works a lot like the other urandom one, but just does the work in reverse. Bash is very powerful!

```
“ tr -cd '[:alnum:]' < /dev/urandom | fold -w30 | head -n1
```

Here's another example that filters using the strings command, which outputs printable strings from a file, which in this case is the urandom feature.

```
“ strings /dev/urandom | grep -o '[:alnum:]' | head -n 30 | tr -d '\n'; echo
```

Here's an even simpler version of the urandom one.

```
“ < /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c6
```

This one manages to use the very useful dd command.


```
“ dd if=/dev/urandom bs=1 count=32 2>/dev/null | base64 -w 0 | rev | cut -b 2-  
| rev
```

You can even create a random left-hand password, which would let you type your password with one hand.

```
“ </dev/urandom tr -dc '12345!@#$%qwertyQWERTasdfgASDFGzxcvZXCVB' | head -c8;  
echo ""
```

If you're going to be using this all the time, it's probably a better idea to put it into a function. In this case, once you run the command once, you'll be able to use *randpw* anytime you want to generate a random password. You'd probably want to put this into your `~/.bashrc` file.

```
“ randpw(){ < /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c${1:-16};echo;}
```

You can use this same syntax to make any of these into a function—just replace everything inside the `{ }`

And here's the easiest way to make a password from the command line, which works in Linux, Windows with Cygwin, and probably Mac OS X. I'm sure that some people will complain that it's not as random as some of the other options, but honestly, it's random enough if you're going to be using the whole thing.

```
“ date | md5sum
```

Yeah, that's even easy enough to remember.

bash oneliner commands

* alle durch salt generierten Icinga config files in einer Datei schreiben und den salt header entfernen

* apply rules in `/etc/icinga2/zones.d/global-templates/notifications/slack/` :

```
workdir="/etc/icinga2/zones.d/global-templates/notifications/slack";for file in $(ls
${workdir}); do echo ${file}; cat ${workdir}/${file} | sed -E -e '/^#.*|^\/*|^*\//d';echo
"";echo "" ; done > /tmp/slack.log
```

Linksammlung Docs

- Ansible: <https://docs.ansible.com/>
- Grafana: <http://docs.grafana.org/>
- Prometheus: <https://prometheus.io/docs>
- Telegraf von Influx: <https://docs.influxdata.com/telegraf/v1.7/introduction/installation/>
- PostgreSQL: <https://www.postgresql.org/docs>
- Graylog: <http://docs.graylog.org/en/2.4/>
- Elasticsearch DB: <https://www.elastic.co/guide/index.html>

fail2ban

Anzeigen von gebannten IPs eines JAILs:

```
# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed:    6
| `-- File list:      /var/log/auth.log
`- Actions
   |- Currently banned: 0
   |- Total banned:    1
   `-- Banned IP list:
```

sshd ist hier ein sog. JAIL also Gefängnis.

IP entbannen

```
# fail2ban-client set sshd unbanip 192.168.1.100
192.168.1.100
```

Debian Upgrade auf neuen Release

Quelle: <https://www.cyberciti.biz/faq/update-upgrade-debian-9-to-debian-10-buster/>

Upgrade Debian 9 to Debian 10 Buster

The procedure is as follows:

1. Backup your system.
2. Update existing packages and reboot the Debian 9.x system.
3. Edit the file `/etc/apt/sources.list` using a text editor and replace each instance of **stretch** with **buster**.
4. Update the packages index on Debian Linux, run: `sudo apt update`
5. Prepare for the operating system upgrade, run: `sudo apt upgrade`
6. Finally, update Debian 9 to Debian 10 buster by running: `sudo apt full-upgrade`
7. Reboot the Linux system so that you can boot into Debian 10 Buster
8. Verify that everything is working correctly.

Let us see all command in details.

Step 1. Backup your system

It is crucial to backup all data and system configurations. Cloud-based VMs can be quickly backup and restore using snapshots. I use [rsnapshot](#), which is the perfect solution for making backups on the local or remote servers. [Check os version in Linux:](#)

```
lsb_release -a
```

Sample outputs:

```
No LSB modules are available.
Distributor ID: Debian
Description: Debian GNU/Linux 9.9 (stretch)
```

```
Release: 9.9  
Codename: stretch
```

Note down the [Linux kernel version](#) too:

```
uname -mrs
```

Sample outputs:

```
Linux 4.9.0-9-amd64 x86_64
```

Step 2. Update installed packages

Type the following [apt command](#) or [apt-get command](#):

```
sudo apt update  
sudo apt upgrade  
sudo apt full-upgrade  
sudo apt --purge autoremove
```

OR

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get full-upgrade  
sudo apt-get --purge autoremove
```

[Reboot the Debian 9.x stretch](#) to apply the kernel and other updates:

```
sudo reboot
```

Step 3. Update /etc/apt/sources.list file

Before starting the upgrade you must reconfigure APT's source-list files. To view current settings using the [cat command](#):

```
cat /etc/apt/sources.list
```

Sample outputs:

```
deb http://cdn-aws.deb.debian.org/debian stretch main  
deb http://security.debian.org/debian-security stretch/updates main  
deb http://cdn-aws.deb.debian.org/debian stretch-updates main
```

The stretch indicates that we are using an older version. Hence, we must change all the references in this file from Stretch to Buster using a text editor such as vim:

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

I prefer to use sed tool, but first backup all config files using the [cp command](#):

```
sudo cp -v /etc/apt/sources.list /root/  
sudo cp -rv /etc/apt/sources.list.d/ /root/  
sudo sed -i 's/stretch/buster/g' /etc/apt/sources.list  
sudo sed -i 's/stretch/buster/g' /etc/apt/sources.list.d/*  
### see updated file now ###  
cat /etc/apt/sources.list
```

```
admin@ip-172-26-10-123:~$ sudo sed -i 's/stretch/buster/g' /etc/apt/sources.list  
admin@ip-172-26-10-123:~$ cat /etc/apt/sources.list  
deb http://cdn-aws.deb.debian.org/debian buster main  
deb http://security.debian.org/debian-security buster/updates main  
deb http://cdn-aws.deb.debian.org/debian buster-updates main  
admin@ip-172-26-10-123:~$
```

© www.cyberciti.biz

APT source-list files updated to use buster

Updating the package list

Simply run:

```
sudo apt update
```

```
admin@ip-172-26-10-123:~$ sudo apt update  
Get:1 http://security.debian.org/debian-security buster/updates InRelease  
Get:2 http://security.debian.org/debian-security buster/updates/main amd64  
Get:3 http://security.debian.org/debian-security buster/updates/main Translation-en  
Get:4 http://cdn-aws.deb.debian.org/debian buster InRelease [118 kB]  
Get:5 http://cdn-aws.deb.debian.org/debian buster-updates InRelease [46.1 kB]  
Get:6 http://cdn-aws.deb.debian.org/debian buster/main amd64 Packages [71.2 kB]  
Get:7 http://cdn-aws.deb.debian.org/debian buster/main Translation-en [52.0 kB]  
Fetched 14.1 MB in 3s (3,778 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
310 packages can be upgraded. Run 'apt list --upgradable' to see them.  
admin@ip-172-26-10-123:~$
```

© www.cyberciti.biz

Step 4. Minimal system upgrade

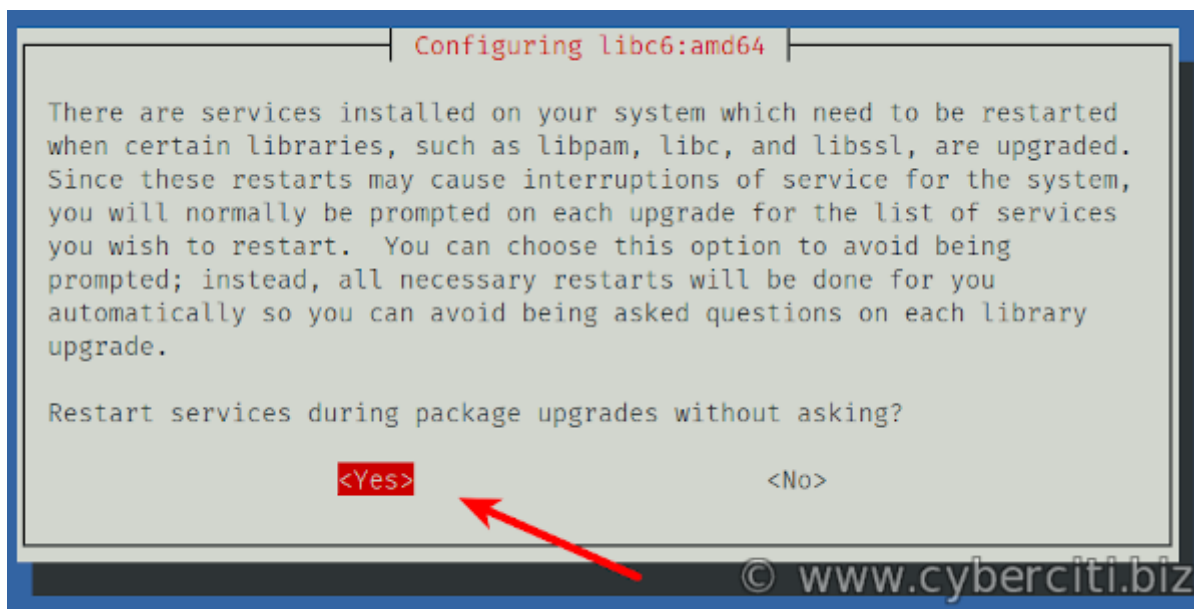
A two-part process is necessary to avoid the removal of large numbers of packages that you want to keep. Therefore, first run the following:

```
sudo apt upgrade
```

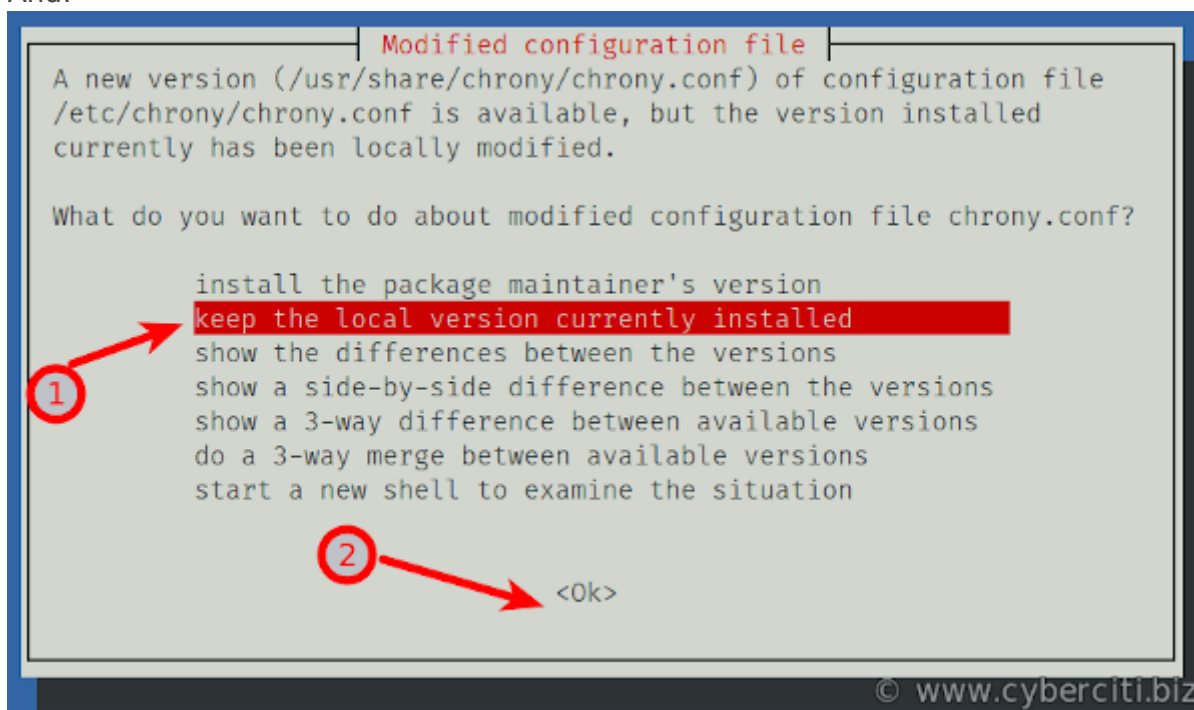
```
admin@ip-172-26-10-123:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  dh-python libbind9-140 libdns162 libicu57 libisc160 libisccc140 libisccfg140 liblwres1
  libtomcrypt0 libtommath1 libwebpmux2 python-pyasn1 python3-prettytable python3.5 pytho
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  apparmor dirmgr e2fsprogs-l10n eatmydata exim4-base exim4-config exim4-daemon-light f
  gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm guile-2.2-libs libargon2-1 l
  libdns-export1104 libdns1104 libeatmydata1 libefivar1 libevent-2.1-6 libex
  libgirepository-1.0-1 libgnutls-dane0 libgpg-error-l10n libgsasl7 libicu63 libimagequa
  libisccfg163 libjansson4 libjson-c3 libkyotocabinet16v5 liblmbd0 libltdl7 liblwres161
  libncursesw6 libnftables0 libnftnl11 libnss-systemd libntlm0 libpam-cap libpcre2-8-0 l
  libpython3.7-minimal libpython3.7-stdlib libtinfo6 libuchardet0 libunbound8 libunistri
  linux-image-4.19.0-5-amd64 mailutils mailutils-common mariadb-common mysql-common nfta
  python2-minimal python3-asn1crypto python3-certifi python3-dbus python3-debconf python
  python3-jsonschema python3-lib2to3 python3-olefile python3.7 python3.7-minimal runit-h
The following packages have been kept back:
  curl
The following packages will be upgraded:
  adduser apt apt-listchanges apt-transport-https apt-utils awscli base-files base-passw
  bzip2 ca-certificates chrony cloud-init coreutils cpio cron dash dbus debconf debconf-
  diffutils distro-info-data dmidecode dmsetup docutils-common dpkg e2fslibs e2fsprogs e
  gnupg gnupg-agent gpgv grep groff-base grub-common grub-pc grub-pc-bin grub2-common gz
  initramfs-tools initramfs-tools-core iproute2 iptables iputils-ping irqbalance isc-dhc
  krb5-locales less libacl1 libapparmor1 libapt-inst2.0 libapt-pkg5.0 libassuan0 libattr
  libbz2-1.0 libc-bin libc-l10n libc6 libcap-ng0 libcap2 libcomerr2 libcurl3-gnutls libd
  libedit2 libelf1 libestr0 libexpat1 libfastjson4 libfdisk1 libffi6 libfontconfig6 libf
  libglb2.0-data libgmp10 libgnutls30 libgpg-error0 libgpm2 libgssapi-krb5-2 libhogweed
  libjpeg62-turbo libk5crypto3 libkeyutils1 libklibc libkmod2 libkrb5-3 libkrb5support0
  liblocale-gettext-perl liblognorm5 liblz4-1 liblzma5 libmagic-mgc libmagic1 libmount1
  libnetfilter-contrack3 libnettle6 libnewt0.52 libnftnl0 libnftnl1 libnftnl2 libnftnl3 li
  libpam-runtime libpam-systemd libpam0g libpaper-utils libpaper1 libpcap0.8 libpcre3 li
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib libpython3-stdlib libreadlin
  libsasl2-modules-db libseccomp2 libselinux1 libsemanage-common libsemanage1 libsepol1
  libssl1.1 libstdc++6 libsystemd0 libtasn1-6 libtext-charwidth-perl libtext-iconv-perl
  libwebp6 libwebpdemux2 libwrap0 libxapian3 libxml2 libxtables12 libyaml-0-2 linux-bas
  lsb-base lsb-release man-db manpages mime-support mount multiarch-support nano ncurses
  openssh-client openssh-server openssh-sftp-server openssl passwd perl-base pinentry-cu
  python-cffi-backend python-chardet python-cryptography python-enum34 python-idna pytho
  python-pyasn1 python-requests python-setuptools python-six python-urllib3 python2.7 py
  python3-boto python3-botocore python3-cffi-backend python3-chardet python3-configobj p
  python3-debianbts python3-docutils python3-httplib2 python3-idna python3-jinja2 python
  python3-markupsafe python3-minimal python3-oauthlib python3-pil python3-pkg-resources
  python3-pygments python3-pysimplesoap python3-reportbug python3-requests python3-roman
  python3-six python3-urllib3 python3-yaml readline-common reportbug rsyslog screen sed
  systemd-sysv sysvinit-utils tar taskset taskset-data tcpdump tzdata ucf udev unattended
  vim-runtime vim-tiny wget whiptail xdg-user-dirs xml-core xxd xz-utils zlib1g
309 upgraded, 99 newly installed, 0 to remove and 1 not upgraded.
Need to get 205 MB of archives.
After this operation, 577 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

© www.cyberciti.biz

Just follow on-screen instructions. During the upgrade process, you may get various questions, like “Do you want to restart the service?” OR “keep or erase config options” and so on.



And:



Step 5. Upgrading Debian 9 to Debian 10

In addition, minimum upgrades we need to do full upgrades to finish the whole Debian 9 to Debian 10 update process. This is the main part of the upgrade. In other words, execute the following command to perform a complete upgrade of the system, installing the newest available versions of all packages, and resolving all possible dependency:

```
sudo apt full-upgrade
```

```
admin@ip-172-26-10-123:~$ sudo apt full-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  dh-python libbind9-140 libdns162 libicu57 libisc160 libisccc140 libisccfg140 libl
  libtomcrypt0 libtommath1 libwebpmux2 linux-image-4.9.0-8-amd64 python-pyasnl pyth
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  libcurl3
The following NEW packages will be installed:
  libcurl4
The following packages will be upgraded:
  curl
1 upgraded, 1 newly installed, 1 to remove and 0 not upgraded.
Need to get 596 kB of archives.
After this operation, 134 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

[Reboot the Linux system](#) to boot into Debian Linux 10 buster, issue:

```
sudo reboot
```

Step 6. Verification

It is time to confirm the upgrade. Run:

```
uname -r
```

```
lsb_release -a
```

Sample outputs:

```
No LSB modules are available.
Distributor ID: Debian
Description: Debian GNU/Linux 10 (buster)
Release: 10
Codename: buster
```

Finally, clean up outdated packages using the [apt command/apt-get command](#):

```
sudo apt --purge autoremove
```

```
admin@ip-172-26-10-123:~$ lsb_release -a
```

```
No LSB modules are available.
```

```
Distributor ID: Debian
```

```
Description:    Debian GNU/Linux 10 (buster)
```

```
Release:        10
```

```
Codename:       buster
```

```
admin@ip-172-26-10-123:~$ uname -r
```

```
4.19.0-5-amd64
```

```
admin@ip-172-26-10-123:~$ sudo apt --purge autoremove
```

```
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

```
The following packages will be REMOVED:
```

```
dh-python* libbind9-140* libdns162* libicu57* libisc160* libisccc140* libiscfg  
libtomcrypt0* libtommath1* libwebpmux2* linux-image-4.9.0-8-amd64* python-pyasn
```

```
0 upgraded, 0 newly installed, 18 to remove and 0 not upgraded.
```

```
After this operation, 256 MB disk space will be freed.
```

```
Do you want to continue? [Y/n] y
```

```
(Reading database ... 47902 files and directories currently installed.)
```

```
Removing dh-python (3.20190308) ...
```

```
Removing libbind9-140:amd64 (1:9.10.3.dfsg.P4-12.3+deb9u5) ...
```

```
Removing libisccc140:amd64 (1:9.10.3.dfsg.P4-12.3+deb9u5) ...
```

```
Removing libdns162:amd64 (1:9.10.3.dfsg.P4-12.3+deb9u5) ...
```

```
Removing libicu57:amd64 (57.1-6+deb9u2) ...
```

```
Removing libisccc140:amd64 (1:9.10.3.dfsg.P4-12.3+deb9u5) ...
```

```
Removing libisc160:amd64 (1:9.10.3.dfsg.P4-12.3+deb9u5) ...
```

```
Removing liblwres141:amd64 (1:9.10.3.dfsg.P4-12.3+deb9u5) ...
```

```
Removing python3.5 (3.5.3-1+deb9u1) ...
```

```
Removing libpython3.5-stdlib:amd64 (3.5.3-1+deb9u1) ...
```

```
Removing python3.5-minimal (3.5.3-1+deb9u1) ...
```

```
Unlinking and removing bytecode for runtime python3.5
```

```
Removing libpython3.5-minimal:amd64 (3.5.3-1+deb9u1) ...
```

```
Removing libtomcrypt0:amd64 (1.17-9) ...
```

```
Removing libtommath1:amd64 (1.1.0-3) ...
```

```
Removing libwebpmux2:amd64 (0.5.2-1) ...
```

```
Removing linux-image-4.9.0-8-amd64 (4.9.110-3+deb9u6) ...
```

```
/etc/kernel/postrm.d/initramfs-tools:
```

```
update-initramfs: Deleting /boot/initrd.img-4.9.0-8-amd64
```

```
/etc/kernel/postrm.d/zz-update-grub:
```

```
Generating grub configuration file ...
```

```
Found linux image: /boot/vmlinuz-4.19.0-5-amd64
```

```
Found initrd image: /boot/initrd.img-4.19.0-5-amd64
```

```
Found linux image: /boot/vmlinuz-4.9.0-9-amd64
```

```
Found initrd image: /boot/initrd.img-4.9.0-9-amd64
```

```
done
```

```
Removing python-pyasnl (0.4.2-3) ...
```

```
Removing python3-prettytable (0.7.2-4) ...
```

```
Processing triggers for libc-bin (2.28-10) ...
```

```
Processing triggers for man-db (2.8.5-2) ...
```

```
Processing triggers for mime-support (3.62) ...
```

```
(Reading database ... 42652 files and directories currently installed.)
```

```
Purging configuration files for python3.5-minimal (3.5.3-1+deb9u1) ...
```

```
Purging configuration files for libpython3.5-minimal:amd64 (3.5.3-1+deb9u1) ...
```

```
Purging configuration files for linux-image-4.9.0-8-amd64 (4.9.110-3+deb9u6) ...
```

```
admin@ip-172-26-10-123:~$
```

Conclusion

And there you have it. We have successfully upgraded to Debian Linux 10. Debian project also posted an in-depth guide [here](#) that explains other issues one might face during installation.

missing firmware

Jeder kennt es, wenn man sich ein Update zieht und diverse Treiber nicht verfügbar sind.

Auf dieser Seite will ich zu meinen Fällen Links und Lösungen zeigen wie ich die Herausforderungen lösen konnte.

Mein System

Ich benutze nur noch Debian als Linux, da es sich einfach aus meiner Sicht am besten verwalten lässt und nicht überladen ist. Nun gibt es hier 2 Herausforderungen:

Zum Einen sind im aktuellen Stable Release oft nicht die aktuellsten Treiber und Software Pakete eingebaut.

Und zum Anderen tun sich die Hersteller leider heutzutage immer noch schwer mit dem Support von Linux. Darum ärgert man sich immer wieder mit fehlenden Treibern herum. Vor allem bei Laptops ist das häufiger das Problem.

Aus diesem Grund benutze ich auf meinem Laptop (von Tuxedo) einen sehr aktuellen Kernel und versuche auch immer den neuesten zu verwenden, da hier meisten für ganz aktuelle Hardware die Treiber vorhanden sind. Wenn auch manchmal noch im Teststadium.

```
saphir: ~/ $ uname -a
```

```
Linux saphir 5.6.0-0.bpo.2-amd64 #1 SMP Debian 5.6.14-2-bpo10+1 (2020-06-09) x86_64 GNU/Linux
```

Meine fehlenden Pakete

```
W: Possible missing firmware /lib/firmware/rtl_nic/rtl8125a-3.fw for module r8169
W: Possible missing firmware /lib/firmware/rtl_nic/rtl8168fp-3.fw for module r8169
W: Possible missing firmware /lib/firmware/i915/icl_dmc_ver1_09.bin for module i915
W: Possible missing firmware /lib/firmware/i915/tgl_dmc_ver2_04.bin for module i915
W: Possible missing firmware /lib/firmware/i915/skl_huc_2.0.0.bin for module i915
W: Possible missing firmware /lib/firmware/i915/bxt_huc_2.0.0.bin for module i915
W: Possible missing firmware /lib/firmware/i915/kbl_huc_4.0.0.bin for module i915
W: Possible missing firmware /lib/firmware/i915/glk_huc_4.0.0.bin for module i915
W: Possible missing firmware /lib/firmware/i915/kbl_huc_4.0.0.bin for module i915
W: Possible missing firmware /lib/firmware/i915/cml_huc_4.0.0.bin for module i915
```

```
W: Possible missing firmware /lib/firmware/i915/cml_guc_33.0.0.bin for module i915
W: Possible missing firmware /lib/firmware/i915/icl_huc_9.0.0.bin for module i915
W: Possible missing firmware /lib/firmware/i915/ehl_huc_9.0.0.bin for module i915
W: Possible missing firmware /lib/firmware/i915/ehl_guc_33.0.4.bin for module i915
W: Possible missing firmware /lib/firmware/i915/tgl_huc_7.0.3.bin for module i915
W: Possible missing firmware /lib/firmware/i915/tgl_guc_35.2.0.bin for module i915
```

Wegfindung

Ich bin dann auf der Suche nach dem fehlenden Firmwares auf folgenden Bugreport gestoßen und fand dann durch diesen Post hier ein Repo für Linux Kernel mit Firmware Modulen.

<https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=947356#30>

Repo Linux Firmware Module

https://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git/plain/rtl_nic

Repo-Ordner für meine fehlenden Module

rtl_nic

https://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git/plain/rtl_nic

i915

<https://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git/plain/i915/>

Lösungsweg

1. Git Repo von der [Projektseite](#) klonen in einen lokalen Ordner
 1. mkdir /home/USER/GIT/linux_firmware
 2. cd /home/USER/GIT/linux_firmware
 3. git clone git://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git
2. Module in die entsprechenden Ordner des Systems kopieren und alle vorhanden überschreiben

1. z.b. i915:

1. `/bin/cp -frv /home/USER/GIT/linux_kernel/linux-firmware/i915/*
/lib/firmware/i915/`

3. initramfs aktualisieren und die neuen Module laden lassen

1. `update-initramfs -u`

1. Parameter "-u" -> Update an existing initramfs

Backports

Backports aktivieren

Repo für Backports hinterlegen `/etc/apt/sources.list.d/backports.list`:

```
# buster backports
deb http://http.debian.net/debian buster-backports main
```

Danach die Paketverwaltung neu laden:

```
apt-get update
```

Paket aus den Backports installieren:

```
apt-get install PAKET -t buster-backports
```


Kernel updaten

Vorbereitungen

Backports aktivieren: <https://wiki.freakylabs.de/books/linux/page/backports>

Nach verfügbaren Kernel Versionen suchen

```
apt-cache search linux-image
----- gekürzte Liste -----
linux-headers-4.19.0-10-amd64 - Header files for Linux 4.19.0-10-amd64
linux-headers-4.19.0-10-cloud-amd64 - Header files for Linux 4.19.0-10-cloud-amd64
linux-headers-4.19.0-10-rt-amd64 - Header files for Linux 4.19.0-10-rt-amd64
linux-headers-4.19.0-11-amd64 - Header files for Linux 4.19.0-11-amd64
linux-headers-4.19.0-11-cloud-amd64 - Header files for Linux 4.19.0-11-cloud-amd64
linux-headers-4.19.0-11-rt-amd64 - Header files for Linux 4.19.0-11-rt-amd64
.... snipped ....
linux-image-5.5.0-0.bpo.2-amd64 - Linux 5.5 for 64-bit PCs (signed)
linux-image-5.5.0-0.bpo.2-cloud-amd64 - Linux 5.5 for x86-64 cloud (signed)
linux-image-5.6.0-0.bpo.2-amd64 - Linux 5.6 for 64-bit PCs (signed)
linux-image-5.6.0-0.bpo.2-cloud-amd64 - Linux 5.6 for x86-64 cloud (signed)
linux-image-5.6.0-0.bpo.2-rt-amd64 - Linux 5.6 for 64-bit PCs, PREEMPT_RT (signed)
linux-image-5.7.0-0.bpo.2-amd64 - Linux 5.7 for 64-bit PCs (signed)
linux-image-5.7.0-0.bpo.2-cloud-amd64 - Linux 5.7 for x86-64 cloud (signed)
```

Kernel Version installieren

```
apt-get install linux-image-5.7.0-0.bpo.2-amd64
```

Paket selber bauen - dpkg-buildpackage

Beispiel Paket: libssh2-1

Problem:

Debian Buster -> Saltstack Master Anbindung per SSH Key an Gitlab

Vorgehensweise:

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

apt-get update

apt-get install devscripts debhelper-compat libgcrypt20-dev zlib1g-dev chrpath

mkdir /root/build

cd /root/build

apt-get source libssh2-1

vim /root/build/libssh2-1.9.0/debian/control
-> Build-Depends: debhelper-compat (= 13) => Build-Depends: debhelper (>= 12)
-> vim /root/build/libssh2-1.9.0/debian/compat => 10 als Inhalt einfügen

dch -i

libssh2 (1.9.0-2.1) stable; urgency=medium

** Non-maintainer upload.

- David Fritsch <darkentik@gmx.de> Sat, 27 Mar 2021 20:14:59 +0100

dpkg-buildpackage
```

==> DEB Paket: libssh2-1_1.9.0-2.1_amd64.deb

MySQL - MariaDB

Datenbank mysql wiederherstellen

Quelle: <https://stackoverflow.com/questions/8911115/how-to-recover-recreate-mysqls-default-mysql-database>

```
mysql_install_db
```

```
mysqld --initialize
```

CREATE OR REPLACE USER

Quelle: <https://mariadb.com/kb/en/create-user/#or-replace>

GRANT PRIVILEGES

Quelle: <https://phoenixnap.com/kb/how-to-create-mariadb-user-grant-privileges>

Password with special characters

Quelle: <https://www.tutorialspoint.com/set-special-characters-for-password-while-creating-a-new-mysql-user>

```
create user 'yourUserName'@'yourHostName' identified by 'yourSpecialCharacterPassword';
```

Paketmanager - apt

Paket Neuinstallation und Neuerstellung von Dateien und Ordnern

Quelle: <https://askubuntu.com/questions/66533/how-can-i-restore-configuration-files>

1. Find out [what package installed the config file](#):

```
$ dpkg -S unity-greeter.conf
unity-greeter: /etc/lightdm/unity-greeter.conf
```

As you can see, the name of the package is `unity-greeter`.

If you deleted a directory, like `/etc/pam.d`, you can list every package that added to it by using the directory path:

```
$ dpkg -S /etc/pam.d
login, sudo, libpam-runtime, cups-daemon, openssh-server, cron, policykit-1, at,
samba-common, ppp, accountsservice, dovecot-core, passwd: /etc/pam.d
```

2. Run the following command, replacing `<package-name>` with the name of the package:

```
sudo apt install --reinstall -o Dpkg::Options::="--force-confask,confnew,confmiss"
<package-name>
```

And for restoring the directory:

```
sudo apt install --reinstall -o Dpkg::Options::="--force-confask,confnew,confmiss"
$(dpkg -S /etc/some/directory | sed 's/,//g; s/.*//')
```

3. If everything worked as expected, you should get a message:

```
Configuration file `/etc/lightdm/unity-greeter.conf', does not exist on system.
Installing new config file as you requested.
```

4. A Practical example when needing to reinstall all of the PulseAudio configuration files:

```
apt-cache pkgnames pulse | xargs -n 1 apt-get -o Dpkg::Options::="--force-confmiss"  
install --reinstall
```