Docker Aufgaben 2

#INSY #STO

Aufgabe 1:

3 Hauptvarianten vom Node Image:

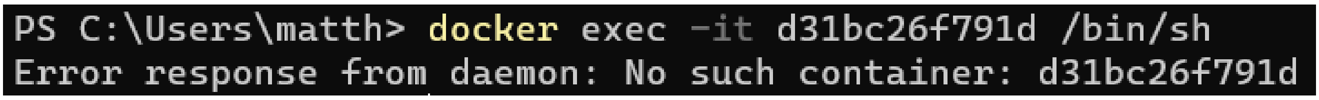
- node
  - Standard node version
  - Groß
  - Hat schon viele dependencies vorinstalliert
  - 952 MB
- node-alpine
  - Basierend auf Alpine-Linux (lightweight)
  - Enthält nur das notwendigste
  - 178 MB
- node-slim
  - Oft basierend auf Debian, CentOS
  - Enthält ebenfalls nur die notwendigsten Abhängigkeiten
  - Könnte größer als das Alpine-Image sein
  - ~188 MB

Image herunterladen und umbenennen:

```
docker pull node:alpine
docker image tag node:alpine small-node
```

Container erstellen und nach dem schließen löschen:

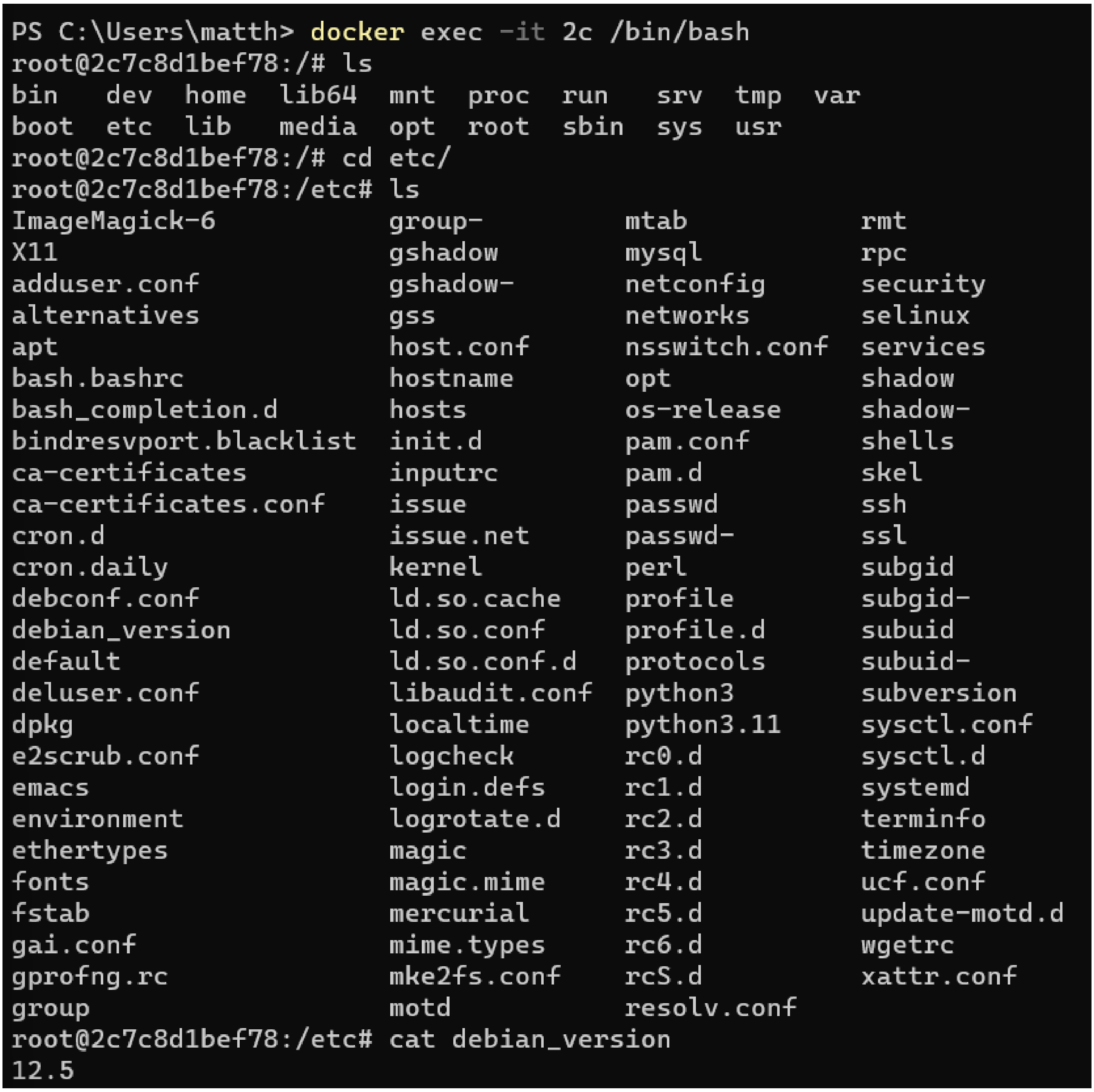
```
docker run --rm -it small-node
```



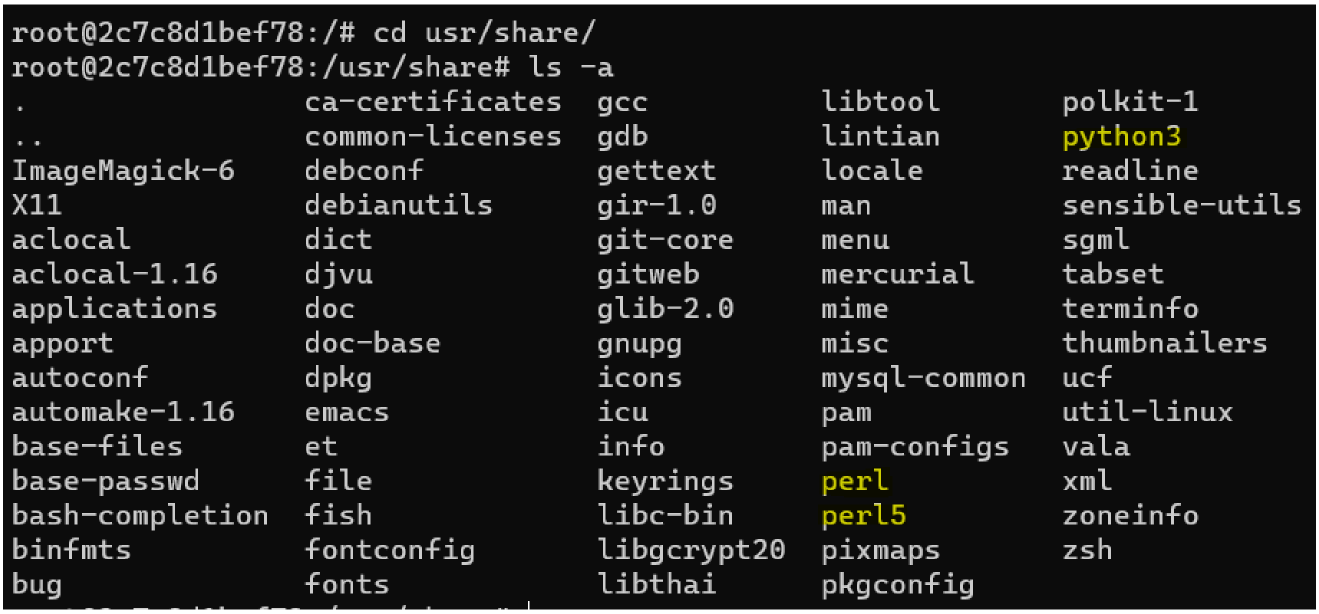
```
PS C:\Users\matth> docker exec -it d31bc26f791d /bin/sh
Error response from daemon: No such container: d31bc26f791d
```

Aufgabe 2:

```
docker pull node
docker run -it node
docker exec -it <container_ID> /bin/bash
```

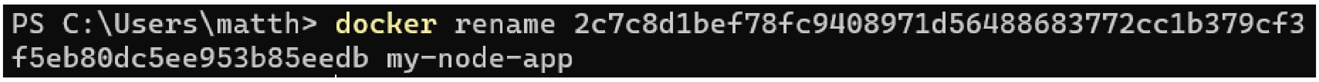


```
PS C:\Users\matth> docker exec -it 2c /bin/bash
root@2c7c8d1bef78:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@2c7c8d1bef78:/# cd etc/
root@2c7c8d1bef78:/etc# ls
ImageMagick-6           group-         mtab           rmt
X11                     gshadow        mysql          rpc
adduser.conf            gshadow-       netconfig      security
alternatives            gss            networks       selinux
apt                     host.conf      nsswitch.conf  services
bash.bashrc             hostname       opt            shadow
bash_completion.d       hosts          os-release     shadow-
bindresvport.blacklist  init.d         pam.conf       shells
ca-certificates         inputrc        pam.d          skel
ca-certificates.conf    issue          passwd         ssh
cron.d                  issue.net      passwd-        ssl
cron.daily              kernel         perl           subgid
debconf.conf            ld.so.cache    profile        subgid-
debian_version          ld.so.conf     profile.d      subuid
default                 ld.so.conf.d   protocols      subuid-
deluser.conf            libaudit.conf  python3        subversion
dpkg                    localtime      python3.11     sysctl.conf
e2scrub.conf            logcheck       rc0.d          sysctl.d
emacs                   login.defs     rc1.d          systemd
environment             logrotate.d    rc2.d          terminfo
ethertypes              magic          rc3.d          timezone
fonts                   magic.mime     rc4.d          ucf.conf
fstab                   mercurial      rc5.d          update-motd.d
gai.conf                mime.types     rc6.d          wgetrc
gprofng.rc              mke2fs.conf    rcS.d          xattr.conf
group                   motd           resolv.conf
root@2c7c8d1bef78:/etc# cat debian_version
12.5
```

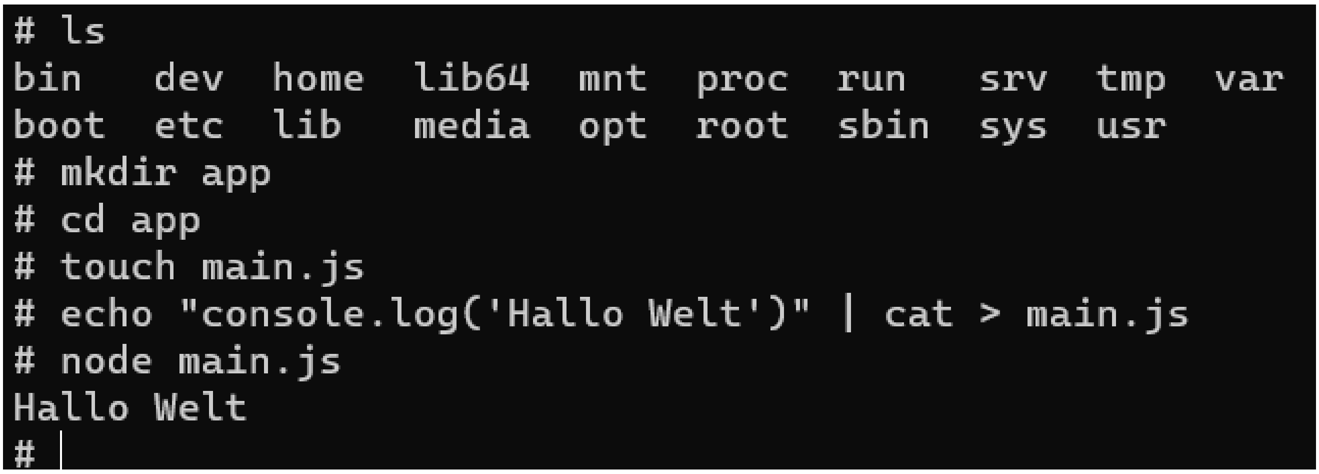


```
root@2c7c8d1bef78:/# cd usr/share/
root@2c7c8d1bef78:/usr/share# ls -a
.                ca-certificates  gcc          libtool       polkit-1
..               common-licenses  gdb          lintian       python3
ImageMagick-6    debconf          gettext      locale        readline
X11              debianutils      gir-1.0      man           sensible-utils
aclocal          dict             git-core     menu          sgml
aclocal-1.16     djvu             gitweb       mercurial     tabset
applications     doc              glib-2.0     mime          terminfo
apport           doc-base         gnupg        misc          thumbnailers
autoconf         dpkg             icons        mysql-common  ucf
automake-1.16    emacs            icu          pam           util-linux
base-files       et               info         pam-configs   vala
base-passwd      file             keyrings     perl          xml
bash-completion  fish             libc-bin     perl5         zoneinfo
binfmts          fontconfig       libcrypt20   pixmaps       zsh
bug              fonts            libthai      pkgconfig
```

Aufgabe 3:



```
PS C:\Users\matth> docker rename 2c7c8d1bef78fc9408971d56488683772cc1b379cf3
f5eb80dc5ee953b85eedb my-node-app
```



```
# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
# mkdir app
# cd app
# touch main.js
# echo "console.log('Hallo Welt')" | cat > main.js
# node main.js
Hallo Welt
# 
```