

# Astra Linux Docker

•  
•  
•

 . :

- [Docker Astra Linux 1.7](#);
- [Docker](#);
- [Docker- Astra Linux Special Edition?](#);
- [Docker](#) .



:

- [Astra Linux Special Edition .10015-01 2022-0819SE17 \( 1.7.2\)](#)

...

, Docker, (. [synaptic](#)) :

```
sudo apt install debootstrap docker.io
```

:

1. chroot-;
2. chroot-;
3. chroot- Docker.

:

1. chroot- debootstrap. chroot- () ncurses-term, mc, locales, nano, gawk, lsb-release, acl, perl-modules-5.28. :  
a. , , Astra Linux Special Edition - . chroot- Astra Linux /var/docker-chroot.



/var chroot- , Astra Linux Special Edition [2022-0819SE17 \( 1.7.2\)](#) 3:63:-1:ccnr, . Astra Linux ( /var). ( ( - )).



Astra Linux Special Edition --components=main,contrib,non-free , Astra Linux Special Edition .

Astra Linux Special Edition .10015-01 ( 1.7) 1.7\_x86-64, :

```
sudo debootstrap \
--include ncurses-term,mc,locales,nano,gawk,lsb-release,acl,
perl-modules-5.28 \
--components=main,contrib,non-free 1.7_x86-64 \
/var/docker-chroot \
http://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-main
```

, https- ( apt-transport-https ).

- b. (), , :

```
sudo debootstrap --verbose orel /var/docker-chroot file:///srv/repo
/orel
```

2. , chroot, :  
a. /etc/resolv.conf /etc/apt/sources.list chroot:

```
sudo cp /etc/resolv.conf /var/docker-chroot/etc/resolv.conf
sudo cp /etc/apt/sources.list /var/docker-chroot/etc/apt/sources.
list
```

, , , /var/docker-chroot/etc/apt/sources.list :

```
sudo nano /var/docker-chroot/etc/apt/sources.list
```

 Docker-

() :

astra-sec-level dockerd. 1 6, :

- 1—5: ;
- 6: , , .

:

- :
  - /etc/docker :

```
sudo mkdir -p /etc/docker
```

- /etc/docker/daemon.json :

```
{
  "debug" : true,
  "astra-sec-level" : 6
}
```

- :
  - :

```
sudo systemctl edit docker
```

- :

```
[Service]
Environment="DOCKER_OPTS=--astra-sec-level 6"
```

- docker:

```
sudo systemctl restart docker
```

b. chroot- , , , chroot- chroot (, locales ):

```
sudo chroot /var/docker-chroot
apt update
apt dist-upgrade
echo "ru_RU.UTF-8 UTF-8" >> /etc/locale.gen
echo "en_US.UTF-8 UTF-8" >> /etc/locale.gen
locale-gen
update-locale ru_RU.UTF-8
exit
```

3. :

```
sudo tar -C /var/docker-chroot -cpf - . | \
sudo docker import - wiki/astralinux:se \
  --change "ENV PATH /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr
/bin:/sbin:/bin" \
  --change 'CMD ["/bin/bash"]' \
  --change "ENV LANG=ru_RU.UTF-8"
```

```
-C      --change,  PATH, -, (/bin/bash), -, .
```

,:

1. :

```
sudo docker images
```

2. :

a. Astra Linux :

```
sudo docker run -it --rm wiki/astralinux:se
```

```
#!/bin/sh
program=$(basename $0)
version=1.0

set -e

pkg_missing=false
for required_pkg in docker.io debootstrap; do
  if ! dpkg -l $required_pkg >/dev/null 2>/dev/null; then
    printf 'Please install %s package\n' $required_pkg
    pkg_missing=true
  fi
done
if $pkg_missing; then
  exit 1
fi

# Check docker can be run without sudo
docker version 2>&1 >/dev/null ||\
  (printf 'Please run with sudo or add your account to `docker` group\n';\
  exit 1)

usage="\
Usage:
  $program -v
          Print program version

  $program -r REPOSITORY [-c CODENAME] -i IMAGE_NAME [-b]

          Create Docker image IMAGE_NAME based on REPOSITORY with CODENAME

  -v          Print version
  -r REPOSITORY Address of the repository
  -c CODENAME  Codename (specified in $REPOSITORY/dists)
  -i IMAGE_NAME Name of the image being created
  -b          Install base Astra Linux packages

default CODENAME is \"stable\""

invalid_args() {
  echo "${usage}" 1>&2
  exit 1
}

REPO=$REPO
IMAGE=$IMAGE
CODENAME="${CODENAME:-stable}"
```

```

install_base_pkgs=false

while getopts 'r:c:i:vb' option; do
    case $option in
        r)
            REPO=$OPTARG
            ;;
        i)
            IMAGE=$OPTARG
            ;;
        c)
            CODENAME=$OPTARG
            ;;
        b)
            install_base_pkgs=true
            ;;
        v)
            echo $program $version
            ;;
        ?)
            invalid_args
            ;;
    esac
done

if [ -z $REPO ]; then
    echo Please specify -r \(repository\) argument
fi
if [ -z $IMAGE ]; then
    echo Please specify -i \image\) argument
fi
if [ -z $REPO ] || [ -z $IMAGE ]; then
    invalid_args
fi

ROOTFS_IMAGE="$IMAGE-rootfs"

TMPDIR=`mktemp -d`
cd $TMPDIR

cleanup() {
    cd $HOME
    # debootstrap leaves mounted /proc and /sys folders in chroot
    # when terminated by Ctrl-C
    sudo umount $TMPDIR/proc $TMPDIR/sys >/dev/null 2>/dev/null || true
    # Delete temporary data at exit
    sudo rm -rf $TMPDIR
}
trap cleanup EXIT

sudo -E debootstrap --no-check-gpg --variant=minbase \
    --components=main,contrib,non-free "$CODENAME" ./chroot "$REPO"

echo "deb $REPO $CODENAME contrib main non-free" | sudo tee ./chroot/etc/apt/sources.list

docker rmi "$ROOTFS_IMAGE" 2>/dev/null || true

sudo tar -C chroot -c . | docker import - "$ROOTFS_IMAGE"

docker rmi "$IMAGE" 2>/dev/null || true

if $install_base_pkgs; then
    cmd="echo Installing base packages && apt-get install -y parsec parsec-tests linux-astra-modules-common
astra-safepolicy lsb-release acl perl-modules-5.28 ca-certificates"
else
    cmd="true"
fi

docker build --network=host --no-cache=true -t "$IMAGE" - <<EOF
FROM $ROOTFS_IMAGE
ENV TERM xterm-256color
ENV DEBIAN_FRONTEND noninteractive
RUN apt-get update
RUN $cmd
WORKDIR /
CMD bash
EOF

printf 'Docker image "%s" has been generated\n' "$IMAGE"
exit 0

```

