

# Kodi 21 Debian DLBox

Debian 12.11 Bookworm Kodi + Docker download box

## BIOS settings

- set SATA mode to AHCI (anything but raid!)
- enable SGX
- Set secure boot to audit
- Change boot order to USB first
- enable wake on LAN

## Debian and Kodi install

Boot Debian 12.11 bookworm via USB and install as following:

- configure language + country
- setup wifi or ethernet as desired
- hostname kodi
- domain empty
- root pw kodi2025
- user full name kodi
- username kodi
- password kodi
- disk: guided use entire disk (not lvm or encrypted!)
- all in one partition
- packet manager UK, default mirror, no proxy
- don't participate in package usage survey
- only install ssh server and standard system utilities. no desktop environment!
- remove USB stick and reboot

Log in as root and set up ssh for easier configuration from a remote system:

```
ssh-keygen  
ssh-copy-id user@targethost
```

check ip address with "ip addr" set up any internal portforwarding if required

on remote system, ssh to system using kodi@IP-ADDR

```
su -  
apt install zip unzip
```

Set bash fancy prompt and prefs:

```
cat >> /etc/bash.bashrc << "EOD"
```

```
# set a fancy prompt colour
if [ "$LOGNAME" = root ] || [ "`id -u`" -eq 0 ] ; then
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;31m\]\u\[\033[00m\]@\[\033[
01;32m\]\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
else
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u\[\033[00m\]@\[\033[
01;32m\]\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
fi

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval
"$$(dircolors -b)"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# Warn if removing more than 3 files or recursively
alias rm='rm -I'

#exclude docker overlay from df command
alias df='/usr/bin/df -x overlay'

EOD

source /etc/bash.bashrc
```

## Set VIM preferences

```
cat > /etc/vimrc << "EOD"
"placeholder as /usr/share/vim/vim80/defaults.vim is loaded if no ~/.vimrc
exists
if has("syntax")
    syntax on
endif
```

```
set tabstop=4      " The width of a TAB is set to 4.
                  " Still it is a \t. It is just that
                  " Vim will interpret it to be having
                  " a width of 4.

set shiftwidth=4   " Indents will have a width of 4

set softtabstop=4  " Sets the number of columns for a TAB

set expandtab      " Expand TABs to spaces

set nocompatible  " Fix telnet issues where cursor keys add A B etc

set mouse-=a      " Disable automatic visual mode when selecting with
mouse

set copyindent     " Disable auto indent when copying indented text

" Only do this part when compiled with support for autocommands.
if has("autocmd")

    " Enable file type detection.
    " Use the default filetype settings, so that mail gets 'tw' set to 72,
    " 'cindent' is on in C files, etc.
    " Also load indent files, to automatically do language-dependent
indenting.
    " Revert with ":filetype off".
    filetype plugin indent on

    " Put these in an autocmd group, so that you can revert them with:
    " ":augroup vimStartup | au! | augroup END"
augroup vimStartup
    au!

    " When editing a file, always jump to the last known cursor position.
    " Don't do it when the position is invalid or when inside an event
handler
    " (happens when dropping a file on gvim).
autocmd BufReadPost *
    \ if line("\") >= 1 && line("\") <= line("$") |
    \   exe "normal! g\"" |
    \ endif

augroup END

endif " has("autocmd")
EOD
cp /etc/vimrc /root/.vimrc
```

Disable IPv6:

```
cat >> /etc/sysctl.conf << "EOD"
# Disable IPv6
net.ipv6.conf.all.disable_ipv6 = 1
#

EOD
sysctl -p
```

Install X server for Intel GPUs and sound system

```
apt-get install software-properties-common xorg xserver-xorg-legacy alsa-
utils mesa-utils git librtmp1 libmad0 lm-sensors libmpeg2-4 avahi-daemon
libnfs13 libva2 vainfo i965-va-driver dbus-x11 pastebinit xserver-xorg-
video-intel intel-media-va-driver firmware-linux
```

# Allow anybody to start the x server:

```
#dpkg-reconfigure xserver-xorg-legacy
cat > /etc/X11/Xwrapper.config << "EOD"
allowed_users=anybody
needs_root_rights=yes
EOD
```

# add kodi user to all required groups

```
usermod -a -G cdrom,audio,video,plugdev,users,dialout,dip,input,render kodi
```

# allow kodi user to shutdown, suspend, upgrade system - enables option in kodi to shutdown/restart etc

```
cat >> /etc/polkit-1/rules.d/50-kodi.rules << "EOD"
polkit.addRule(function(action, subject) {
    if (subject.user == "kodi") {
        polkit.log("action=" + action);
        polkit.log("subject=" + subject);
        if (action.id.indexOf("org.freedesktop.login1.") == 0) {
            return polkit.Result.YES;
        }
        if (action.id.indexOf("org.freedesktop.udisks.") == 0) {
            return polkit.Result.YES;
        }
        if (action.id.indexOf("org.freedesktop.udisks2.") == 0) {
            return polkit.Result.YES;
        }
        if (action.id.indexOf("org.freedesktop.upower.") == 0) {
            return polkit.Result.YES;
        }

        if (action.id.indexOf("org.debian.apt.upgrade-packages") == 0) {
            return polkit.Result.YES;
        }
    }
});
```

```
    }
    if (action.id.indexOf("org.debian.apt.update-cache") == 0) {
        return polkit.Result.YES;
    }
}
});
EOD
```

/usr/sbin where the reboot command is located is not in user paths by default, so add:

```
cat >> /home/kodi/.profile << "EOD"
```

```
PATH="/usr/sbin:$PATH"
EOD
```

# system d service that actively waits for network connection:

```
cat >> /etc/systemd/system/kodi.service << "EOD"
[Unit]
Description = kodi-standalone using xinit
Requires = dbus.service
After = systemd-user-sessions.service sound.target network-online.target
mysql.service
Wants = mysql.service

[Service]
User = kodi
Group = kodi
Type = simple
PAMName=login
ExecStart = /usr/bin/xinit /usr/bin/dbus-launch --exit-with-session env
KODI_AE_SINK=ALSA /usr/bin/kodi-standalone -- :0 -nolisten tcp vt7
Restart = always
RestartSec = 5
#Restart = on-abort

[Install]
WantedBy = multi-user.target
EOD
```

In case of use of AV receiver and/or HDMI connection and device disappearing when receiver/tv turned off, add this to X config to disable power management:

```
cat >> /etc/X11/xorg.conf.d/10-monitors.conf << "EOD"
Section "Extensions"
    Option "DPMS" "Disable"
EndSection
EOD
```

# setting nice priority of kodi to -1

```
sed -i '/# End of file/ikodi - nice -1' /etc/security/limits.conf
```

# fake display manager service

```
ln -s /etc/systemd/system/kodi.service /etc/systemd/system/display-manager.service
```

Add Debian Multimedia repository for Kodi

```
cat >> /etc/apt/sources.list.d/deb-multimedia.sources << "EOD"
Types: deb
URIs: https://www.deb-multimedia.org
Suites: bookworm bookworm-backports
Components: main
Signed-By: /etc/apt/trusted.gpg.d/deb-multimedia-keyring.asc
EOD
```

# <https://www.deb-multimedia.org/pool/main/d/deb-multimedia-keyring/>

```
wget
https://www.deb-multimedia.org/pool/main/d/deb-multimedia-keyring/deb-multimedia-keyring_2024.9.1_all.deb
dpkg -i deb-multimedia-keyring*.deb
```

# lower priority of deb multimedia repo

```
cat >> /etc/apt/preferences.d/99deb-multimedia << "EOD"
Package: *
Pin: origin www.deb-multimedia.org
Pin-Priority: 1
EOD
apt update
```

# Install Kodi

```
apt install kodi kodi-inputstream-adaptive kodi-vfs-libarchive kodi-vfs-rar
-t "o=Unofficial Multimedia Packages"
```

# Set Kodi boot logo and silent boot

```
sed -i 's/#quiet_boot="0"/quiet_boot="1"/' /etc/grub.d/10_linux
sed -i 's/#vt_handoff="0"/vt_handoff="1"/' /etc/grub.d/10_linux
```

```
cat > /etc/default/grub << "EOD"
GRUB_DEFAULT=0
GRUB_TIMEOUT=0
GRUB_TIMEOUT_STYLE=hidden
```

```
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_GFXPAYLOAD_LINUX=keep
GRUB_DISABLE_OS_PROBER=true
#GRUB_CMDLINE_LINUX="i915.modeset=1 console=tty12"
GRUB_CMDLINE_LINUX=""
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash vt.global_cursor_default=0"
GRUB_GFXMODE=1920x1080
EOD
```

```
update-grub
```

```
echo FRAMEBUFFER=y > /etc/initramfs-tools/conf.d/framebuffer
git clone https://github.com/solbero/plymouth-theme-kodi-animated-logo
cd plymouth-theme-kodi-animated-logo
apt-get install -y fakeroot plymouth-label plymouth-themes
./build.sh
dpkg -i plymouth-theme-kodi-animated-logo.deb
# update-alternatives: using /usr/share/plymouth/themes/kodi-animated-
logo/kodi-animated-logo.plymouth to provide
/usr/share/plymouth/themes/default.plymouth (default.plymouth) in auto mode
plymouth-set-default-theme kodi-animated-logo
update-initramfs -u
update-grub
```

```
# for intel gpus prevent flickering by preserving framebuffer of bios/bootloader and prevent flickering
```

```
echo "options i915 fastboot=1" > /etc/modprobe.d/i915.conf
```

```
# set to 8x18 or largest font size
```

```
dpkg-reconfigure console-setup
```

```
# set up unattended system upgrades
```

```
apt-get install -y unattended-upgrades
cat >> /etc/apt/apt.conf.d/50unattended-upgrades << "EOD"
Unattended-Upgrade::Origins-Pattern {
    "o=Debian,n=bullseye";
    "o=Debian,n=bullseye-updates";
    "o=Debian,n=bullseye-proposed-updates";
    "o=Debian,n=bullseye-security";
    "o=Debian,n=bullseye,l=Debian-Security";
    "o=Unofficial Multimedia Packages,a=stable";
    "o=Unofficial Multimedia Packages,a=stable-backports";
```

```
//To enable ALL updates, use this:
```

```
//     "site=*";
};
```

```
#Don't force updates
```

```
Unattended-Upgrade::AutoFixInterruptedDpkg "false";
```

```
#set mail to whoever should receive it and make sure bsd-mailx or sendmail
#is installed and able to send mails
Unattended-Upgrade::Mail "root";
```

```
#configure reboot if desired.
Unattended-Upgrade::Automatic-Reboot "false";
EOD
```

```
cat >> /etc/apt/apt.conf.d/20auto-upgrades << "EOD"
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::AutocleanInterval "3";
APT::Periodic::Unattended-Upgrade "1";
EOD
```

```
# mount ext usb at same point and add to fstab
```

```
# get uuid
blkid /dev/sda1
#add to fstab:
UUID=20e3cd6b-df8a-44a0-bf3b-4e2c59413e99 /mnt/usb/ ext4
defaults,auto,users,rw 0 0
```

```
cat >> /etc/udev/rules.d/99-mount.rules << "EOD"
ACTION=="add", SUBSYSTEMS=="usb", SUBSYSTEM=="block",
ENV{ID_FS_USAGE}=="filesystem", RUN{program}+="/usr/bin/systemd-mount --no-
block --automount=yes --collect $devnode /mnt/usb"
EOD
/etc/init.d/udev restart
```

```
# set primary wifi network interface:
```

```
vi /etc/network/interfaces
```

```
iface wlp3s0 inet static
    wpa-ssid ssid-name
    wpa-psk wifipassword
    address 192.168.1.100
    netmask 255.255.255.0
    gateway 192.168.1.254
    dns-nameservers 192.168.1.254 8.8.8.8
```

```
# Set maximum 100MB journald log files
```

```
sed -i 's/#SystemMaxUse=/SystemMaxUse=500M/' /etc/systemd/journald.conf
mkdir -p /etc/systemd/journald.conf.d/
cat >> /etc/systemd/journald.conf.d/max-size.conf << "EOF"
[Journal]
SystemMaxUse=100M
MaxRetentionSec=7day
```

```
EOF
```

```
systemctl restart systemd-journald
```

## Docker install

Assumptions: download dir: /media/downloads media storage: /mnt/usb/

```
mkdir -p /opt/audiobookshelf-storage/{metadata,config}
mkdir /opt/heimdall
mkdir /opt/portainer
mkdir /opt/prowlarr
mkdir /opt/qbittorrent
mkdir /opt/radarr
mkdir /opt/sonarr
mkdir -p /media/downloads/{complete,incomplete,watch}
mkdir /mnt/usb/movies
mkdir /mnt/usb/tv
mkdir /mnt/usb/audiobooks
mkdir /mnt/usb/ebooks
```

## Docker compose files

```
mkdir /root/docker

cat >> /root/docker/docker-audiobookshelf.yml << "EOD"
name: audiobookshelf
services:
  audiobookshelf:
    image: advplyr/audiobookshelf
    container_name: audiobookshelf
    ports:
      - 13378:80
    volumes:
      - /mnt/usb/ebooks:/ebooks
      - /mnt/usb/audiobooks:/audiobooks
      - /opt/audiobookshelf-storage/metadata:/metadata
      - /opt/audiobookshelf-storage/config:/config
    restart: unless-stopped
EOD

cat >> /root/docker/docker-flaresolVERR.yml << "EOD"
name: flaresolVERR
services:
  flaresolVERR:
```

```
image: ghcr.io/flaresolVERR/flaresolVERR:latest
# DockerHub mirror flaresolVERR/flaresolVERR:latest
#image: 21hsmw/flaresolVERR:fixlooping
#image: docker.io/library/oui:oui
#image: ghcr.io/flaresolVERR/flaresolVERR:latest
#image: ghcr.io/flaresolVERR/flaresolVERR:pr-1282
#jcolfej/flaresolVERR:pr-1272
#image: alexfozor/flaresolVERR:pr-1300
#image: alexfozor/flaresolVERR:pr-1300-experimental
container_name: flaresolVERR
environment:
  - LOG_LEVEL=debug
  - LOG_HTML=false
  - CAPTCHA_SOLVER=none
  - TZ=Europe/London
  - LANG=en_GB
ports:
  - "8191:8191"
restart: unless-stopped
```

EOD

```
cat >> /root/docker/docker-heimdall.yml << "EOD"
```

```
name: heimdall
services:
  heimdall:
    image: lscr.io/linuxserver/heimdall:latest
    container_name: heimdall
    environment:
      - PUID=1007
      - PGID=1007
      - TZ=Europe/London
    volumes:
      - /opt/heimdall:/config
    ports:
      - 80:80
      - 443:443
    restart: unless-stopped
```

EOD

```
cat >> /root/docker/docker-portainer.yml << "EOD"
```

```
name: portainer
services:
  portainer:
    image: portainer/portainer-ce:latest
    container_name: portainer
    restart: always
    environment:
      - TZ=Europe/London
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /opt/portainer:/data
```

```
ports:
  - 8000:8000
  - 9443:9443/tcp
```

EOD

```
cat >> /root/docker/docker-prowlarr.yml << "EOD"
```

```
name: prowlarr
services:
  prowlarr:
    image: lscr.io/linuxserver/prowlarr:develop #or latest
    container_name: prowlarr
    environment:
      - PUID=1001
      - PGID=1001
      - TZ=Europe/London
    volumes:
      - /opt/prowlarr:/config
    ports:
      - 9696:9696
    restart: unless-stopped
```

EOD

```
cat >> /root/docker/docker-qbittorrent.yml << "EOD"
```

```
name: qbittorrent
services:
  qbittorrent:
    image: lscr.io/linuxserver/qbittorrent:latest
    container_name: qbittorrent
    environment:
      - PUID=1004
      - PGID=1001
      - TZ=Europe/London
      - WEBUI_PORT=9092
      - TORRENTING_PORT=6881
    volumes:
      - /opt/qbittorrent:/config
      - /media/downloads:/downloads
      - /mnt/usb/tv:/tv
      - /mnt/usb/movies:/movies
    ports:
      - 9092:9092
      - 6881:6881
      - 6881:6881/udp
    restart: unless-stopped
```

# notes:

# login admin, password random shown via docker logs qbittorrent

# changing ports requires changing webui/torrenting env as well

EOD

```
cat >> /root/docker/docker-radarr.yml << "EOD"
```

```
name: radarr
services:
  radarr:
    image: lscr.io/linuxserver/radarr:develop
    container_name: radarr
    environment:
      - UMASK_SET=022
      - PUID=1003
      - PGID=1001
      - TZ=Europe/London
    volumes:
      - /mnt/usb/movies:/movies
      - /media/downloads/complete:/downloads
      - /opt/radarr:/config
    ports:
      - 7878:7878
    restart: unless-stopped
```

EOD

```
cat >> /root/docker/docker-sonarr.yml << "EOD"
```

```
name: sonarr
services:
  sonarr:
    image: lscr.io/linuxserver/sonarr:latest
    container_name: sonarr
    environment:
      - PUID=1002
      - PGID=1001
      - TZ=Europe/London
    volumes:
      - /mnt/usb/tv:/data
      - /media/downloads/complete:/downloads
      - /opt/sonarr:/config
    ports:
      - 8989:8989
    restart: unless-stopped
```

EOD

## install docker community edition

```
apt-get -y install curl unzip dialog
curl -fSSL get.docker.com | sh
```

## Setup docker containers and start

```
adduser --system --uid 1007 --no-create-home --group "heimdall"
chown heimdall:heimdall /opt/heimdall-config -R
```

```
docker compose -f docker-heimdall.yml up -d

groupadd --gid 1001 media
adduser --system --uid 1001 --no-create-home --ingroup "media" "prowlarr"
chown prowlarr:media /opt/prowlarr
docker compose -f docker-prowlarr.yml up -d

adduser --system --uid 1004 --no-create-home --ingroup "media" "bittorrent"
mkdir -p /media/downloads/{complete,incomplete,watch}
chmod 775 /media/downloads/{complete,incomplete,watch}
chown bittorrent:media /media/downloads -R
docker compose -f docker-transmission.yml up -d
mkdir -p /opt/qbittorrent/
chown bittorrent:media /opt/qbittorrent
docker compose -f docker-qbittorrent.yml up -d

mkdir /mnt/usb/tv
mount /mnt/usb/tv
adduser --system --uid 1002 --no-create-home --ingroup "media" "sonarr"
chown sonarr:media /opt/sonarr/
docker compose -f docker-sonarr.yml up -d

mkdir /mnt/usb/movies
mkdir /media/downloads
adduser --system --uid 1003 --no-create-home --ingroup "media" "radarr"
chown radarr:media /opt/radarr/
docker compose -f docker-radarr.yml up -d

docker compose -f docker-portainer.yml up -d

docker compose -f docker-flaresolVERR.yml up -d

docker compose -f docker-audiobookshelf.yml up -d
```

## Docker auto update

```
curl -L
https://github.com/regclient/regclient/releases/latest/download/regctl-linux
-amd64 >/usr/local/bin/regctl
chmod 755 /usr/local/bin/regctl
```

```
cat >> /root/docker/update-all-containers.sh << "EOD"
#!/usr/bin/env bash
script_dir=$(dirname "$0")

#curl -L
https://github.com/regclient/regclient/releases/latest/download/regctl-linux
-amd64 >/usr/local/bin/regctl
#chmod 755 /usr/local/bin/regctl
```

```
### If not in PATH, set full path. Else just "regctl"
regbin="/usr/local/bin/regctl"
### options to allow exclude:
while getopts "e:" options; do
  case "${options}" in
    e) Exclude=${OPTARG} ;;
    *) exit 0 ;;
  esac
done
shift "$((OPTIND-1))"
### Create array of excludes
IFS=',' read -r -a Excludes <<< "$Exclude" ; unset IFS

SearchName="$1"

for i in $(docker ps --filter "name=$SearchName" --format '{{.Names}}') ; do
  ### Looping every item over the list of excluded names and skipping:
  for e in "${Excludes[@]}" ; do [[ "$i" == "$e" ]] && continue 2 ; done
  #printf ". "
  RepoUrl=$(docker inspect "$i" --format '{{.Config.Image}}')
  LocalHash=$(docker image inspect "$RepoUrl" --format '{{.RepoDigests}}')
  ### Checking for errors while setting the variable:
  if RegHash=$(($regbin image digest --list "$RepoUrl" 2>/dev/null) ; then
    if [[ "$LocalHash" = *"$RegHash"* ]] ; then
      NoUpdates+=("$i");
    else
      GotUpdates+=("$i");
      docker rm -f "$i" 2>&1
      docker compose -f "$script_dir/docker-$i.yml" pull -q 2>&1
    >/dev/null

    if [ ! $? -eq 0 ]
    then
      echo; echo;
      echo "Could not pull image for $i."
    else
      docker compose -f "$script_dir/docker-$i.yml" up -d 2>&1
    >/dev/null

    if [ ! $? -eq 0 ]
    then
      echo; echo;
      echo "Could not start new container for $i."
    else
      docker image prune -f 2>&1 >/dev/null
      echo "Updated $i."
    fi
  fi
fi
else
  echo "Container has errors: $i"
fi
```

```
done
EOD

chmod +x /root/docker/update-all-containers.sh

cat >> /etc/cron.daily/docker-containers << "EOD"
#!/bin/bash
/root/docker/update-all-containers.sh
EOD

chmod 755 /etc/cron.daily/docker-containers
```

## App Configurations

### qbittorrent

The password for the initial username "admin" is auto-generated by qbittorrent docker log in as root

```
docker logs qbittorrent
```

note the random password and log in using those details via browser on <http://192.168.1.100:9092/>

settings:

```
webui: change username and password "something"
downloads: "set Merge trackers to existing torrent"
downloads: set "append .!qB extension to incomplete files"
downloads: Default Torrent Management Mode "automatic"
downloads: Default save path "/downloads/complete"
downloads: tick "Keep incomplete torrents in" "/downloads/incomplete"
downloads: set automatically add torrents from "/downloads/watch"
connection: set "global max connections" to 1200
bittorrent: set max active downloads to 30, max active uploads to 10, max
active torrents to 50
bittorrent: set seeding limits when ratio reaches 0.1, total time 60 minutes
bittorrent: auto add trackers to new downloads:
udp://tracker.opentrackr.org:1337/announce
udp://tracker.torrent.eu.org:451
udp://open.stealth.si:80/announce
udp://opentracker.i2p.rocks:6969
```

### radarr

set forms login page auth method, username/password something settings: show advanced options!

- media management
  - colon replacement: replace with space dash space
  - movie folder format for large collections: “{Movie TitleFirstCharacter}/{Movie Title} ({Release Year})”
  - enable “delete empty movie folders”
  - import extra files and set to “srt”
  - set “unmonitor deleted files”
  - add root folder “/movies”
- profiles
  - HD-720p: set upgrades allowed, upgrade until bluray-720p
  - HD-1080p: set upgrades allowed, upgrade until bluray-1080p, untick remux-1080p
  - HD-720p/1080p: set upgrades allowed, upgrade until bluray-1080p, untick remux-1080p
  - any: set upgrades allowed, upgrade until bluray-480p, remove all 4k and telecine/sync/cam
  - delete SD profile
  - ultra-hd: set upgrades allowed, upgrade until bluray-2160p, remove remux
  - add release profiles:
    - dub/sub: must not contain: Hchdrip, Korsub, Azerbaijani Dubbed, Uzbek Dubbed
    - cam: must not contain: Camrip, Hdcam
    - Not multi: must not contain: iTA-ENG, TeeWee, ITA-ENG, ITA ENG
    - 3D/xvid: must not contain: 3D, xvid
- quality:
  - HDTV-720 min 14.1
  - WEBDL-720 min 19.4
  - WEBRip-720 min 19.4
  - Bluray-720 min 19.4
  - HDTV-1080 min 23
  - WEBDL-1080 min 25, max 135.5
  - WEBRIP-1080 min 25
  - Bluray-1080 min 44
- download clients
  - qBittorrent, 192.168.1.100, port 9092, username/password something
  - remotepathmappings: 192.168.1.100, remote “/downloads/complete”, local “/downloads/”
- connections
  - kodi on grab/file import/file upgrade/rename only, host 192.168.1.100 port 8080, user/pass kodi, gui notification, update lib, clean lib, always up
- metadata
  - kodi enable, tick “use movie nfo”
- general: copy api key for heimdall and prowlarr
  - disable usage data
- UI
  - set first day of week and uk date/time formats

## sonarr

set forms login page auth method, username/password something show advanced options! settings:

- media management
  - multi episode style: extend

- add root folder /data
- profiles
  - delete SD, set allow upgrades for all.
  - add release profile “nodub” must not contain dubbed
- download clients
  - qBittorrent, 192.168.1.100, port 9092, username/password something
  - remotepathmappings: 192.168.1.100, remote “/downloads/complete/”, local “/downloads/”
- connections
  - kodi on grab/file import/file upgrade/rename, series delete, episode file delete, episode file delete for upgrade only, host 192.168.1.100 port 8080, user/pass kodi, gui notification, update lib, clean lib, always up
- metadata
  - kodi enable, untick series metadata, episode metadata
- general: copy api key for heimdall and prowlarr
  - disable usage data
- UI
  - set first day of week and uk date/time formats

## prowlarr

set forms login page auth method, username/password something settings: show advanced options!

- indexer proxies
  - flaresolrerr  
tag flaresolrerr  
<http://192.168.1.100:8191/>
- download clients
  - qbittorrent  
192.168.1.100  
port 9092  
add user/password,
- general
  - disable use proxy
  - log level: info
  - disable send anonymous usage data
- copy API key from general tab and add to heimdall
- UI
  - set to uk date/time format
- apps
  - radarr  
add and remove only  
prowlarr <http://192.168.1.100:9696>  
radarr <http://192.168.1.100:7878>  
enter api key
  - sonarr  
add and remove only  
prowlarr <http://192.168.1.100:9696>  
sonarr <http://192.168.1.100:8989>

enter api key

Indexers:

- ext torrent: base ext.to, tag flaresolVERR
- thepiratebay
- kickasstorrents.to: base ninjaproxy1.com, tag flaresolVERR

## Audiobookshelf

root user: root, pw something settings: set uk date format users: add user account api key: add title heimdall, for root user. add to heimdall backups: enable automatic backups, keeping 5 Libraries

- add Books, Audiobooks, icon headphones, audible.com, /audiobooks folder
  - settings: audiobooks only
- add Books, ebooks, icon books, google books, /ebooks folder
  - settings: disable use square book covers

## Kodi

settings → sidebar - set to expert settings System:

- Power saving
  - put display to sleep when idle: 30 minutes

Media:

- library:
  - add sources:
    - movies - /mnt/usb/movies movies
    - tvshows - /mnt/usb/tv tv
    - scan to library
- general:
  - allow file renaming and deletion
- videos
  - show info for unwatched items - disable episode thumb

Interface:

- screensaver: black
- regional: set to UK
- skin - configure skin: disable all entries not required (pvr, games, addons, etc)

services

- control
  - enable, set to kodi kodi and require auth
- Addons
  - install from repo

- program-addons: backup, up next
- services: kodi check previous episode, kodi playback resumer, stinger scene notification, unpause jumpback
- subtitles: opensubtitles.com, opensubtitles.org, subscene

configure addons: backup:

```
general: 10 backups, compressed, progress display: none
rempte dir: /home/kodi/backup (create backup folder in home dir)
file selection: all but game saves
scheduling enabled, run on startup if missed, every day
```

command line as kodi user:

```
#disable kodi splash screen: vi ~/.kodi/userdata/advancedsettings.xml
```

```
<advancedsettings>
  <splash>>false</splash>
</advancedsettings>
```

## Estuary MOD V2 Omega

```
cd ~
# https://repo.kodinerds.net/ red button at top
wget
https://repo.kodinerds.net/addons/repository.kodinerds/repository.kodinerds-7.0.1.7.zip
```

in kodi interface → settings/system → addons → unknown sources enable addons → install from zip file → home folder → kodinerds repo zip addons → install from repo → kodinerds repo → look & feel → estuary mod v2 omega

skin settings home menu

- minimise main menu
- show item label on top
- customise menu items → delete all non-required entries
- show smaller icons for categories widgets
- display of episodes: watched/total
- greyed out watched items

library

- greyed out watched items

extras

- disable splash screen

necessary add-ons:

- global search

- genre icons
- studio icons

artworks:

- video genres icon pack: movie genres icons for estuary mod v2 matrix
- studio icons pack: studio icons - white

```
cd ~
wget
https://github.com/AnonTester/kodi-repo/raw/master/repository.anontester/repository.anontester-1.0.0.zip
```

addons → install from zip file → home folder → anontester repo zip addons → install from repo → wuff repo → services → after Watch

##### hardware decoding notes, probably not required any longer system →Player→Processing:  
Enable HQ Scalers for scaling above: 20% Allow hardware acceleration - VAAPI: on Use Mpeg-2 VAAPI: Yes Use Mpeg-4 VAAPI: if you like UseVC-1 VAAPI: on Use VP8 VAAPI: on (if your device supports it) Use VP9 VAAPI: on (if your device supports it) Use HEVC VAAPI: on (if your device supports it)  
#####

## Static IP

set static IP

```
vi /etc/dhcp/dhclient.conf
```

```
interface eth0 static ip_address=192.168.0.100/24 static routers=192.168.0.1 static domain_name_servers=192.168.0.1 8.8.8.8
```

```
interface wlan0 static ip_address=192.168.0.100/24 static routers=192.168.0.1 static domain_name_servers=192.168.0.1 8.8.8.8
```

## Kodi Customisation

Exclude in-progress movies from unwatched list:

```
sed -i '/<\rule>/a\ \ \ \ <rule field="inprogress" operator="false" \/>'
~/.kodi/addons/skin.estuary.modv2/playlists/unwatched_movies.xsp
```

Add option to close Video OSD immediately when resuming:

```
cd ~/.kodi/addons/skin.estuary.modv2
xml/SkinSettings.xml: <onclick
```

```
condition="String.IsEqual(Skin.String(autoclose_osd),31579)">Skin.SetString(
autoclose_osd,31578)</onclick>
xml/SkinSettings.xml:           <onclick
condition="String.IsEqual(Skin.String(autoclose_osd),31584)">Skin.SetString(
autoclose_osd,31579)</onclick>
xml/SkinSettings.xml:           <onclick
condition="!String.IsEmpty(Skin.String(autoclose_osd)) +
!String.IsEqual(Skin.String(autoclose_osd),31578) +
!String.IsEqual(Skin.String(autoclose_osd),31579) +
!String.IsEqual(Skin.String(autoclose_osd),31580) +
!String.IsEqual(Skin.String(autoclose_osd),31581) +
!String.IsEqual(Skin.String(autoclose_osd),31582) +
!String.IsEqual(Skin.String(autoclose_osd),31583) +
!String.IsEqual(Skin.String(autoclose_osd),31584)">Skin.SetString(autoclose_
osd,31578)</onclick>

xml/Variables.xml:           <value
condition="String.IsEqual(Skin.String(autoclose_osd),31579)">$LOCALIZE[31579
]</value>

language/resource.language.en_gb/strings.po
msgctxt "#31579"
msgid "0 seconds"
msgstr ""

# other closing times defined in xml/Custom_1110_CloseVideo0SD.xml
```

## YouTube addon

The Kodi YouTube addon requires a google account and a personal API key and OAuth credentials.

Add the “unofficial” repository as the version in the official Kodi repository is outdated and likely will not work. Links as per <https://github.com/anxdpanic/plugin.video.youtube/wiki/Installation>

```
wget
"https://ftp.fau.de/osmc/osmc/download/dev/anxdpanic/repositories/repository
.yt.unofficial-2.0.7.zip"
```

Install repo in Kodi via settings → addons → install from zip file → home folder → yt unofficial zip

Create API key and OAuth credentials: <https://github.com/anxdpanic/plugin.video.youtube/issues/1095>  
<https://forum.kodi.tv/showthread.php?tid=356934&pid=3224383#pid3224383>  
<https://github.com/anxdpanic/plugin.video.youtube/issues/439>

<https://console.developers.google.com/>

How to add API key and OAuth credentials: <https://ytaddon.page.link/keys>

Enable API page, then enter the key and credentials on: <http://192.168.1.11:50152/youtube/api>

Kodi forum support thread: <https://forum.kodi.tv/showthread.php?tid=356934&page=223>

From:  
<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:  
<http://wuff.dyndns.org/doku.php?id=kodi:kodi21-debian-dlbox&rev=1753814849>

Last update: **2025/07/29 19:47**

