
Xenia Linux

Luna Deards, Jack Eilles

Jan 07, 2024

INSTALLATION

1	Install Xenia Linux	3
2	Guided installation	5
2.1	Overview	5
2.2	Booting live media	5
2.3	Downloading the installer	5
2.4	Run the installer	5
2.5	Go!	7
3	Manual installation	9
3.1	Overview	9
3.2	Prerequisites	9
3.3	Booting the live media	9
3.4	Setting up disks	9
3.5	Acquiring a root .img	13
3.6	Setting up the new system	14
3.7	Chroot into Xenia and install the bootloader	14
4	Installing applications	17
4.1	Flatpak	17
4.2	Distrobox	18
4.3	foxmerge	18
5	Updating the system	21
5.1	Prerequisites for 0.4 Velox	21
5.2	First-time setup	21
5.3	Updating	22
6	Updating the system - Manual	23
6.1	Preparation	23
6.2	Download the new root	23
6.3	Finalising	24
6.4	Troubleshooting	24
7	Add the Xenia portage overlay	29
7.1	Xenia specific steps	29
7.2	Installing eselect-repository	29
7.3	Add the overlay	29
7.4	Allowing live ebuilds	30
8	Installing Virtual Machine Manager	31

8.1	Creating the container	31
8.2	Installing virt-manager	31
8.3	Starting required services & exporting	32
8.4	Setup the session	32
9	RootFS generation	33
9.1	Xenia setup	33
9.2	foxbuild installation	33
9.3	Creating a root image	33
9.4	Testing the image	34
9.5	Updating the build environment	35
9.6	Customising the image	35
10	Manual rootFS generation	37
10.1	Requirements	37
10.2	First time setup	37
10.3	Create a snapshot	38
10.4	Pick a spec file	38
10.5	Spec file setup	38
10.6	Building	38
10.7	Converting to squashFS	39
10.8	Installation	39
11	Customising the rootfs	41
11.1	Requirements	41
11.2	Packages	41
11.3	USE flags and other portage config	41
11.4	Adding services	42
11.5	Custom shell script	42
11.6	Filesystem overlay	42
11.7	Submitting changes to Xenia	43
12	Filesystem layouts	45
12.1	Traditional label-based layout (generic)	45
12.2	Traditional label-based layout (LVM)	46
12.3	Btrfs	46
12.4	Btrfs with encryption	47
13	OpenRC flatpak workaround	49

Xenia Linux is an immutable OS based on Gentoo Linux. Xenia Linux boots from a SquashFS (the root), much like how a LiveCD works.

Xenia Linux includes the GNOME desktop environment, a selection of filesystems, and versatility with Flatpak, Distrobbox and Portage available to install packages with. Xenia Linux includes your entire system on one file, which can be swapped out for a newer one when updates come.

We named Xenia Linux after the popular Linux mascot who lost to Tux.

The documentation contains guides on installing the system (guided or manual installations), using the system, developing the system and system architecture.

INSTALL XENIA LINUX

There is two ways to install Xenia Linux, *using a installer*, or *performing an installation manually*.

A guided install is easier and will guide you through the process. This is a fast method (disregarding download speeds, an install takes 15 seconds from boot to being at a login screen) and is the recommended method for most users.

A manual install is more similar to a typical Arch/Gentoo installation process, where you format your disks yourself, then install the base system (or the `root.img` in this case). A manual install doesn't provide much benefit apart from bragging rights and learning the internals of Xenia Linux and how it all works under the hood.

If you are in doubt, go with a guided install. If you are feeling adventurous, maybe choose a manual install! It's up to you.

GUIDED INSTALLATION

2.1 Overview

We are so glad you want to install Xenia Linux! The installation can be boiled down to the following steps:

1. Download and boot live media
2. Download the installer
3. Run it!

2.2 Booting live media

The installer relies on certain libraries that are not present in the Gentoo LiveCD, so currently we recommend the use of a [Fedora ISO](#) to use as an installation environment. It is likely other distros will work as well!

Once you are booted into live media, you should setup networking. On Fedora, you can do this in GNOME settings.

2.3 Downloading the installer

Open a terminal and download the installer:

```
liveCD ~ $ git clone https://gitlab.com/xenia-group/installer.git
```

Enter the installer's directory:

```
liveCD ~ $ cd installer
```

2.4 Run the installer

Warning: The installer won't run without root privileges! Make sure you use `sudo` here or become root through other means.

```
liveCD ~ $ sudo ./main.py
```

```
[INFO] Entering interactive mode. Default values are shown wrapped in square brackets,
↳like [this]. Press enter to accept the default.

Enter value for 'drive'. Valid options are ['/dev/nvme0n1', '/dev/nvme1n1', '/dev/sda'].
[]:
```

You will be put into interactive mode of the installer. When asking for values, the installer will give you a list of valid options. The default option will be shown in the square brackets.

2.4.1 Drive

For the drive, pick your drive out of the valid options. If you don't know your drive, you can open another terminal and type `lsblk` to see all the drives on the system and how large they are.

2.4.2 Repository

Stick to the default here unless using `foxbuild` to build images.

2.4.3 Flavour

Here, the flavour of Xenia to be installed can be selected. `gnome-systemd` is the most supported spin and is recommended.

Warning: OpenRC is deprecated! While images are still updated, things might not work as expected and support requests/issues have lower priority. Only use OpenRC if you are prepared to face issues.

2.4.4 Release branch

Release branch defines the branch of Xenia to use. It is important to note that Xenia is rolling, meaning that `unstable` and `main` will run the same package versions. This just defines changes to Xenia itself, such as packages being added to the base set or configuration changes being tested.

It is recommended to stick with the default `main` here.

2.4.5 Filesystem

Here is where you choose which filesystem layout you would like. The default of `btrfs` is *heavily* recommended here, which gives features such as snapshots. Older versions of Xenia Linux used the LVM layout, and the traditional layout is just `ext4` formatted partitions. You can find more information about the different layouts [here](#).

To recap, please use the default of `btrfs`.

2.4.6 Username

Enter a username here that you would like to use as your user. A password will be prompted for later.

2.4.7 Applications

Here, choose either recommended or minimal for packages to install. Recommended will install applications such as a web browser, image viewer, email client and the like with flatpak when you boot into the system. Minimal does not install any extra packages, and the user will need to install these packages themselves through a terminal or GNOME Software.

2.5 Go!

With the installer configured, you'll see the following warning:

```
[WARN] Drive partitioning is about to start. After this process, drive /dev/sda will be ↵  
↵erased. Press enter to continue.
```

When you are ready, press enter to begin the installation!

When the installer is complete, you can reboot into your new installation. You can login with the user you created earlier.

We hope you love Xenia Linux!

MANUAL INSTALLATION

3.1 Overview

This page will act as a rough guideline for installing Xenia Linux manually, without the use of an installer. While this won't show every different way you can setup your system, it will provide the necessary details to get a working system.

This guide will serve as a point-of-reference for those wanting to learn how a complete Xenia system is built and as a guide for those wanting to build their own custom system from scratch.

3.2 Prerequisites

- [Gentoo live media](#)

3.3 Booting the live media

Firstly, boot into the LiveCD and setup networking if applicable. To make sure this guide stays up-to-date, follow the [handbook](#) up to [Configuring the network](#).

3.4 Setting up disks

3.4.1 Xenia expectations

Xenia Linux expects a specific disk layout. For a traditional filesystem, it expects your filesystems to be labelled in a specific manner:

- ROOTS - this is where your root images are stored.
- OVERLAY - this is where system overlays over `/usr`, `/var` and `/etc` are stored.
- HOME - this is where `/home` resides.
- EFI - this is where `/boot/efi` resides.

For Btrfs, it expects the following labels:

- ROOTS - this is where root images, overlays and home is stored.
- EFI - this is where `/boot/efi` resides.

Note: You can find a full list of filesystem layouts [here](#).

3.4.2 Partitioning

First, find the disk to install on with `lsblk`:

```
liveCD ~ # lsblk
```

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda   8:0    0 80G  0 disk
```

Your disk will most likely be `/dev/sda`, `/dev/vda`, `/dev/nvme0n1` or `/dev/mmcblk0`. You can check the `SIZE` column to see if it matches with your disk.

Next, partition the disk with `fdisk`. Replace the disk referenced in the below commands with the one you found above. It should look like this, some text has been redacted for clarity:

```
liveCD ~ # fdisk /dev/sda
```

```
Welcome to fdisk (util-linux 2.38.1).
```

```
Command (m for help):
```

Note: For BIOS systems, we still make an “ESP” as that is where GRUB will store its config. It might be a bit confusing, but specific steps for BIOS/UEFI are marked throughout.

Btrfs

To partition the disk, we will follow the following layout:

- `/dev/sda1` - EFI - 512M
- `/dev/sda2` - ROOTS - remaining space

To achieve this layout with `fdisk`, use the following commands:

UEFI

```
g
n
(prompt enter)
(prompt enter)
+512M
n
(prompt enter)
(prompt enter)
(prompt enter)
w
```

BIOS

```
g
n
(prompt enter)
```

(continues on next page)

(continued from previous page)

```
(press enter)
+512M
n
(press enter)
(press enter)
-1M
n
(press enter)
(press enter)
(press enter)
t
(press enter)
4
w
```

ext4

To partition the disk, we will follow the following layout:

- /dev/sda1 - EFI - 512M
- /dev/sda2 - ROOTS - 8G
- /dev/sda3 - OVERLAY - 20G
- /dev/sda4 - HOME - remaining space

Note: As a rule of thumb, assign at least 4G to ROOTS (otherwise you will not be able to upgrade the system), and around 30% of your remaining space to OVERLAY. The rest can be assigned to /home.

To achieve this layout with `fdisk`, use the following commands:

UEFI

```
g
n
(press enter)
(press enter)
+512M
n
(press enter)
(press enter)
+8G
n
(press enter)
(press enter)
+20G
n
(press enter)
(press enter)
(press enter)
w
```

BIOS

```
g
n
(prompt enter)
(prompt enter)
+512M
n
(prompt enter)
(prompt enter)
+8G
n
(prompt enter)
(prompt enter)
+20G
n
(prompt enter)
(prompt enter)
-1M
n
(prompt enter)
(prompt enter)
(prompt enter)
t
(prompt enter)
4
w
```

3.4.3 Formatting

First, format the EFI System Partition:

```
liveCD ~ # mkfs.vfat -F 32 -n EFI /dev/sda1
```

Btrfs

Next, format the Btrfs partition:

```
liveCD ~ # mkfs.btrfs -L ROOTS /dev/sda2
```

With our disks partitioned, we will mount ROOTS to prepare for the *root.img*:

```
liveCD ~ # mount -L ROOTS /mnt/gentoo
```

```
liveCD ~ # cd /mnt/gentoo
```

Next, we will create the subvolumes for */home* and the filesystem overlays.

```
liveCD ~ # btrfs subvolume create /mnt/gentoo/home
```

```
liveCD ~ # btrfs subvolume create /mnt/gentoo/overlay
```

```
liveCD ~ # btrfs subvolume create /mnt/gentoo/overlay/etc
```

```
liveCD ~ # btrfs subvolume create /mnt/gentoo/overlay/var
```

```
liveCD ~ # btrfs subvolume create /mnt/gentoo/overlay/usr
```

ext4

Next, format the rest of the partitions as ext4:

```
liveCD ~ # mkfs.ext4 -L ROOTS /dev/sda2
```

```
liveCD ~ # mkfs.ext4 -L OVERLAY /dev/sda3
```

```
liveCD ~ # mkfs.ext4 -L HOME /dev/sda4
```


With our disks partitioned, we will mount ROOTS to prepare for the *root.img*:

```
liveCD ~ # mount -L ROOTS /mnt/gentoo
liveCD ~ # cd /mnt/gentoo
```

3.5 Acquiring a root.img

The root image is the heart of a Xenia Linux system. It contains your whole root filesystem.

3.5.1 Picking the root image

There are a few versions of the root image. The main two are:

- **Current**, which is the current stable release of Xenia. Some newer features may not be available, but this is a tested point release
- **Unstable**, which is the latest rolling version of Xenia. This may contain some bugs but will have newer features being prepared for a later release

As a rule of thumb, pick Current if you are new to Xenia Linux or want a more stable, tested system.

3.5.2 Acquiring the root image

Downloading the root in the live media

You can download the root image now. Depending on what release channel you would like, substitute `current` for `unstable`:

systemd

```
liveCD /mnt/gentoo # wget https://repo.xenialinux.com/releases/current/root-systemd.img
liveCD /mnt/gentoo # mv /mnt/gentoo/root-systemd.img /mnt/gentoo/root.img
```

OpenRC

Warning: OpenRC is deprecated (for now) in 0.5 Cana. These images may not work or be present.

```
liveCD /mnt/gentoo # wget https://repo.xenialinux.com/releases/current/root.img
```

Building your own

To learn about building your own root follow the guide [here](#). When you get to the Installing section, host the web server and copy the root over to here.

3.6 Setting up the new system

First, make the directory `/mnt/root`:

```
liveCD /mnt/gentoo # mkdir /mnt/root
liveCD /mnt/gentoo # cd /mnt/root
```

And mount `root.img` there:

```
liveCD /mnt/gentoo # mount -o ro,loop -t squashfs /mnt/gentoo/root.img /mnt/root
```

Mount the ESP:

```
liveCD /mnt/gentoo # mount -L EFI /mnt/root/boot/efi
```

Mount home:

Btrfs

```
liveCD /mnt/gentoo # mount -L ROOTS -o subvol=home /mnt/root/home
```

ext4

```
liveCD /mnt/gentoo # mount -L HOME /mnt/root/home
```

3.7 Chroot into Xenia and install the bootloader

Next, we will chroot into the new system and install the bootloader.

Mount special filesystems:

```
liveCD /mnt/gentoo # mount -t proc /proc /mnt/root/proc
liveCD /mnt/gentoo # mount --rbind /dev /mnt/root/dev
liveCD /mnt/gentoo # mount --rbind /sys /mnt/root/sys
liveCD /mnt/gentoo # mount --bind /run /mnt/root/run
liveCD /mnt/gentoo # mount --make-slave /mnt/root/run
```

Chroot into the new system:

```
liveCD /mnt/gentoo # chroot /mnt/root /bin/bash
```

Note: We install grub with LVM support here, as LVM layouts won't boot without it. Feel free to leave it out if you aren't using LVM.

Install the bootloader:

UEFI

```
liveCD /mnt/gentoo # grub-install --modules=lvm --target="x86_64-efi" --efi-directory="/boot/efi" --boot
```

BIOS

```
liveCD /mnt/gentoo # grub-install --modules=lvm --boot-directory="/boot/efi" /dev/sda # Enter your de
```

Configure the bootloader:

```
liveCD /mnt/gentoo # grub-mkconfig -o /boot/efi/grub/grub.cfg
```

And you are done! Now for the moment of truth:

```
liveCD /mnt/gentoo # reboot
```

On reboot, the password for root will be 87658765XeniaLinux.

INSTALLING APPLICATIONS

On Xenia, the way to install applications is different from the way traditional distributions work. In Xenia, the preferred installation method for GUI applications is Flatpak, for CLI application is Distrobox, and finally the `/usr` overlay for any packages needed on the base system.

4.1 Flatpak

Flatpak is pre-installed and setup ready-to-go. To install applications with Flatpak, you can:

- Use the GUI tool GNOME Software
- Use the CLI

GNOME Software is pre-installed, and it is accessed by searching for “software” in Activities or finding in application view of GNOME.

To use the CLI to install an application, for example Firefox:

```
xenia ~ $ flatpak install firefox
```

```
Looking for matches...
Similar refs found for 'firefox' in remote 'flathub' (system):

 1) app/org.mozilla.firefox.BaseApp/x86_64/20.08
 2) app/org.mozilla.firefox.BaseApp/x86_64/21.08
 3) app/org.mozilla.firefox/x86_64/stable
 4) app/org.mozilla.firefox.BaseApp/x86_64/22.08

Which do you want to use (0 to abort)? [0-4]:
```

Here, you can pick the application to install. In this case, pick 3.

On OpenRC, any Flatpak which launches through DBus (mainly GNOME flatpaks) will not open. Follow the workaround [here](#) if this occurs to you.

4.2 Distrobox

For CLI applications or applications that are not on Flatpak, Xenia ships with Distrobox, a tool to manage containers of other distributions.

Distrobox is similar to Toolbox on Fedora, but gives you access to any distribution you would like. However, we will use the default of Fedora for this guide. You can learn more [here](#).

4.2.1 Setup

First, setup a Distrobox:

```
xenia ~ $ distrobox enter
```

Go through the prompts (you can say yes to everything) to setup a Fedora container. You will now be inside the container.

4.2.2 Installing applications

To install applications, you can use the distributions package manager, in this case DNF:

```
[xenia@my-distrobox ~]$ sudo dnf install emacs
```

4.2.3 Exporting to the host

Distrobox can also export applications to the host. For example, you could install an application like virt-viewer in the container. After this, you can export to the host:

```
[xenia@my-distrobox ~]$ distrobox-export --app virt-viewer
```

4.2.4 Virt-manager (KVM)

A common use-case is using KVM in distrobox instead of cluttering the base system. You can follow the guide [here](#) to set this up.

Alternatively, install GNOME Boxes from Flatpak for a hassle-free solution.

4.3 foxmerge

Note: foxmerge was officially introduced in 0.5 Cana. Please *update your system* if needed.

For applications like VPNs, or drivers such as the proprietary Nvidia drivers, Flatpak or Distrobox will not work.

foxmerge allows for Portage to be used in Xenia. First, add a package:

```
xenia ~ # foxmerge add emacs
```

Then, run an upgrade:

```
xenia ~ # foxmerge upgrade
```

To remove a package:

```
xenia ~ # foxmerge remove emacs
```

On Btrfs systems, `foxmerge` will create snapshots before and after upgrades and removals. If an operation fails, a revert operation will be scheduled.

If something trivial goes wrong, such as a package being keyworded unstable, it is safe to run `foxsnapshot revert --cancel` to cancel the revert, and try again.

As a rule of thumb, do not try to install lots of packages with `foxmerge` (only the ones needed) or try to update the system through it - this will lead to package conflicts in the future.

UPDATING THE SYSTEM

5.1 Prerequisites for 0.4 Velox

On 0.4, the update utility `foxupdate` will not be installed.

First, *install the Xenia overlay*.

Next, run the following command to install `foxupdate`:

```
xenia ~ # mount -o rw,remount /usr
xenia ~ # emaint sync --repo xenia-overlay
xenia ~ # emerge -val foxupdate
```

5.2 First-time setup

First, run `foxupdate`:

```
xenia ~ # foxupdate
```

You will be run through a number of prompts.

- `repository`: This is the URL where the root images are hosted. Unless you are testing your own images, leave this as the default.
- `flavour`: This is what flavour (init system, desktop environment) will be used. As of 0.5, only `gnome-systemd` will be provided.
- `release_branch`: This is what release you want to download. Either use `main` for a stable, tested point release or `unstable` to get the latest features.

If you see an option called `date`, please follow the above instructions to install/update `foxupdate` to the latest version.

`foxupdate` will now show an output showing your choices:

```
[INFO] Config setup!

foxupdate
-----

Repository: https://repo.xenialinux.com/releases/
Flavour: gnome-systemd
Release branch[sub-branch]: unstable

Run foxupdate -u to update!
```

If you would like to change these later, run `foxupdate -c`

5.3 Updating

To update, simply run the following:

```
xenia ~ # foxupdate -u
```

`foxupdate` will only replace the existing root once it has fully downloaded the new one, and when done backs up the existing root to `root.img.bak`.

If an unexpected event occurs where the new image is not bootable, follow the Troubleshooting section [here](#).

UPDATING THE SYSTEM - MANUAL

Warning: If the system is currently on 0.4 Velox, please follow the [guide here](#) to update the system. Updating manually will result in an unbootable system!

6.1 Preparation

Firstly, mount `/roots` as read-write, and enter the directory:

```
xenia ~ # mount -o rw,remount /roots
xenia ~ # cd /roots
```

Backup the current root:

```
xenia /roots # mv /roots/root.img /roots/root.img.bak
```

6.2 Download the new root

Note: Xenia can switch between OpenRC and systemd at will! Use whichever init system you want.

systemd

```
xenia /roots # wget https://repo.xenialinux.com/releases/current/root-systemd.img
xenia /roots # mv /roots/root-systemd.img /roots/root.img
```

OpenRC

Warning: OpenRC is deprecated (for now) in 0.5 Cana. These images may not work or be present.

```
xenia /roots # wget https://repo.xenialinux.com/releases/current/root.img
```

6.3 Finalising

Reboot:

```
xenia /roots # reboot
```

Delete the old root if it is no longer needed:

```
xenia ~ # rm /roots/root.img.bak
```

6.4 Troubleshooting

6.4.1 error: file 'root.img' not found, booting backup root

This error indicates grub cannot find the `root.img`, indicating it no longer exists on `/roots` (or is corrupted).

If a backup was made of the previous root, it can be booted by editing the GRUB menuentry:

Firstly, press enter to continue, then press e to enter the editor. From here, you'll see a menuentry like so:

```

1 menuentry 'Xenia Linux'
2
3     search.fs_label ROOTS roots
4     loopback loop0 ($roots)/root.img
5     load_video
6     if [ "$grub_platform" = xefi ]; then
7         set gfxpayload=keep
8     fi
9     insmod gzio
10    insmod squash4
11    echo      'Loading Linux gentoo-dist ...'
12    linux    (loop0)/boot/vmlinuz ro quiet splash rd.live.squashimg=root.img rd.
↳live.dir=/ rd.live.overlay.overlayfs=1 root=live:/dev/disk/by-label/ROOTS_
↳recovery=false
13    echo      'Loading initial ramdisk ...'
14    initrd   (loop0)/boot/initramfs.img

```

From here, change the occurrences of `root.img` to `root.img.bak`:

```

1 menuentry 'Xenia Linux'
2
3     search.fs_label ROOTS roots
4     loopback loop0 ($roots)/root.img.bak
5     load_video
6     if [ "$grub_platform" = xefi ]; then
7         set gfxpayload=keep
8     fi
9     insmod gzio
10    insmod squash4
11    echo      'Loading Linux gentoo-dist ...'
12    linux    (loop0)/boot/vmlinuz ro quiet splash rd.live.squashimg=root.img.bak_
↳rd.live.dir=/ rd.live.overlay.overlayfs=1 root=live:/dev/disk/by-label/ROOTS_
↳recovery=false

```

(continues on next page)

(continued from previous page)

```

13      echo      'Loading initial ramdisk ...'
14      initrd    (loop0)/boot/initramfs.img

```

Then, press ctrl+x to boot.

6.4.2 Entering emergency mode (0.4 to 0.5 update)

Dracut made a change where the old kernel cmdline we used no longer works in Xenia. If you updated *manually* to 0.5, the system will no longer boot.

First, enter the GRUB editor by pressing e when hovering over Xenia Linux from the GRUB menu. A menuentry will be shown:

```

1  menuentry 'Xenia Linux'
2
3      search.fs_label ROOTS roots
4      loopback loop0 ($roots)/root.img
5      load_video
6      if [ "x$grub_platform" = xefi ]; then
7          set gfxpayload=keep
8      fi
9      insmod gzio
10     insmod squash4
11     echo      'Loading Linux gentoo-dist ...'
12     linux     (loop0)/boot/vmlinuz ro quiet splash rd.live.squashimg=root.img rd.
↵live.dir=/ rd.live.overlay.overlayfs=0 root=live:/dev/disk/by-label/ROOTS_
↵recovery=false
13     echo      'Loading initial ramdisk ...'
14     initrd    (loop0)/boot/initramfs.img

```

Change `rd.live.overlay.overlayfs=0` to `rd.live.overlay.overlayfs=1`:

```

1  menuentry 'Xenia Linux'
2
3      search.fs_label ROOTS roots
4      loopback loop0 ($roots)/root.img
5      load_video
6      if [ "x$grub_platform" = xefi ]; then
7          set gfxpayload=keep
8      fi
9      insmod gzio
10     insmod squash4
11     echo      'Loading Linux gentoo-dist ...'
12     linux     (loop0)/boot/vmlinuz ro quiet splash rd.live.squashimg=root.img rd.
↵live.dir=/ rd.live.overlay.overlayfs=1 root=live:/dev/disk/by-label/ROOTS_
↵recovery=false
13     echo      'Loading initial ramdisk ...'
14     initrd    (loop0)/boot/initramfs.img

```

Then, press ctrl+x to boot.

Fixing it the easy way

Follow the guide [here](#) to update with foxupdate.

Fixing it the hard way

When booted, first mount `/boot/efi`:

```
xenia ~ # mount -L EFI /boot/efi
```

Open `/boot/efi/grub/grub.cfg` in a text editor as root, and find the menuentry for Xenia Linux:

```
1 menuentry 'Xenia Linux' --class xenia --class gnu-linux --class gnu --class os
2 ↪$menuentry_id_option 'gnulinux-simple-/mnt/roots/root.img' {
3     search.fs_label ROOTS roots
4     loopback loop0 ($roots)/root.img
5     load_video
6     if [ "$grub_platform" = xefi ]; then
7         set gfxpayload=keep
8     fi
9     insmod gzio
10    insmod squash4
11    echo 'Loading Linux gentoo-dist ...'
12    linux (loop0)/boot/vmlinuz ro quiet splash rd.live.squashimg=root.img rd.live.
13    ↪dir=/ rd.live.overlay.overlayfs=0 root=live:/dev/disk/by-label/ROOTS recovery=false
14    echo 'Loading initial ramdisk ...'
15    initrd (loop0)/boot/initramfs.img
16 }
17
18 menuentry 'Xenia Linux (Recovery)' --class xenia --class gnu-linux --class gnu --class
19 ↪os $menuentry_id_option 'gnulinux-simple-/mnt/roots/root.img' {
20     search.fs_label ROOTS roots
21     loopback loop0 ($roots)/root.img
22     load_video
23     if [ "$grub_platform" = xefi ]; then
24         set gfxpayload=keep
25     fi
26     insmod gzio
27     insmod squash4
28     echo 'Loading Linux gentoo-dist ...'
29     linux (loop0)/boot/vmlinuz ro quiet splash rd.live.squashimg=root.img rd.live.
30    ↪dir=/ rd.live.overlay.overlayfs=0 root=live:/dev/disk/by-label/ROOTS recovery=true
31    echo 'Loading initial ramdisk ...'
32    initrd (loop0)/boot/initramfs.img
33 }
```

Change the `rd.live.overlay.overlayfs=0` to `rd.live.overlay.overlayfs=1` on both lines:

```
1 menuentry 'Xenia Linux' --class xenia --class gnu-linux --class gnu --class os
2 ↪$menuentry_id_option 'gnulinux-simple-/mnt/roots/root.img' {
3     search.fs_label ROOTS roots
4     loopback loop0 ($roots)/root.img
5     load_video
6     if [ "$grub_platform" = xefi ]; then
```

(continues on next page)

(continued from previous page)

```

6         set gfxpayload=keep
7     fi
8     insmod gzio
9     insmod squash4
10    echo    'Loading Linux gentoo-dist ...'
11    linux   (loop0)/boot/vmlinuz ro quiet splash rd.live.squashimg=root.img rd.live.
↳dir=/ rd.live.overlay.overlayfs=1 root=live:/dev/disk/by-label/ROOTS recovery=false
12    echo    'Loading initial ramdisk ...'
13    initrd  (loop0)/boot/initramfs.img
14 }
15
16 menuentry 'Xenia Linux (Recovery)' --class xenia --class gnu-linux --class gnu --class_
↳os $menuentry_id_option 'gnulinux-simple-/mnt/roots/root.img' {
17     search.fs_label ROOTS roots
18     loopback loop0 ($roots)/root.img
19     load_video
20     if [ "x$grub_platform" = xefi ]; then
21         set gfxpayload=keep
22     fi
23     insmod gzio
24     insmod squash4
25     echo    'Loading Linux gentoo-dist ...'
26     linux   (loop0)/boot/vmlinuz ro quiet splash rd.live.squashimg=root.img rd.live.
↳dir=/ rd.live.overlay.overlayfs=1 root=live:/dev/disk/by-label/ROOTS recovery=true
27     echo    'Loading initial ramdisk ...'
28     initrd  (loop0)/boot/initramfs.img
29 }

```

Save the file.

ADD THE XENIA PORTAGE OVERLAY

The Xenia portage overlay includes multiple tools such as foxsnapshot to be used on a Xenia system.

For development reasons (or to *create a Xenia root image*), the repository can be added to a Gentoo or Xenia system.

7.1 Xenia specific steps

If Xenia Linux is being used, first sync the Gentoo repositories:

Note: The command `emerge-webrsync` is used here as it will fix issues with using `emerge` in the next step.

```
xenia ~ # emerge-webrsync
```

Next, mount `/usr` as read-write:

```
xenia ~ # mount -o rw,remount /usr
```

7.2 Installing eselect-repository

Install `eselect-repository`:

```
xenia ~ # emerge -va eselect-repository
```

7.3 Add the overlay

First, add the overlay:

```
xenia ~ # eselect repository add xenia-overlay git https://gitlab.com/xenia-group/xenia-overlay
```

And sync it:

```
xenia ~ # emaint sync --repo xenia-overlay
```

Done!

7.4 Allowing live ebuids

In their current state, most fox-utils are only available as live ebuids.

To allow `emerge` to use these, add the following to `/etc/portage/package.accept_keywords/xenia`:

```
xenia-tools/* **
```

INSTALLING VIRTUAL MACHINE MANAGER

Warning: This documentation is deprecated. Please use GNOME Boxes if possible.

Virtual Machine Manger, or `virt-manager`, is a popular GUI front-end to `libvirt` (and `KVM/QEMU`) on Linux machines. Installing `virt-manager` is traditionally done on the base system, but this is not advised in Xenia due to potential conflicts (the rule of thumb is to not use the `/usr` overlay if necessary).

Luckily, it is possible to install `virt-manager` on a distrobox, and export it to the host to use virtual machines on Xenia.

8.1 Creating the container

First, create a rootful distrobox with a container that supports init systems - the example given is `registry.opensuse.org/opensuse/tumbleweed:latest`:

Note: The container can be named anything, here it is named `libvirt`.

```
xenia ~ $ distrobox create --root --init --image registry.opensuse.org/opensuse/tumbleweed:latest --nam
```

Next, enter the container

```
xenia ~ $ distrobox enter --root libvirt
```

8.2 Installing virt-manager

Install the `kvm` tools and server:

```
xenia@libvirt:~> sudo zypper install -t pattern kvm_server kvm_tools
```

Install `virt-manager`:

```
xenia@libvirt:~> sudo zypper install virt-manager
```

8.3 Starting required services & exporting

Start libvirtd:

```
xenia@libvirt:~> sudo systemctl enable --now libvirtd
```

Export virt-manager:

```
xenia@libvirt:~> distrobox-export --app virt-manager
```

Virtual Machine Manager is now installed, and it is able to be opened through GNOME like usual - but we still need to setup the QEMU/KVM session.

8.4 Setup the session

Open Virtual Machine Manager through the activities overview on GNOME.

Note: If the container is not started when you open VMM, the startup might take a while. Be patient!

Go to **File > Add Connection...** From here, change the Hypervisor to QEMU/KVM user session, and click Connect.

Done! Now VMM is installed and ready to be used.

ROOTFS GENERATION

To make custom Xenia images, the tool `foxbuild` can be used on a Gentoo system/container or as its own container.

9.1 Xenia setup

As Xenia is immutable and it is advised to not install to the base system when necessary, `distrobox` will be used. This can also be followed on any system with `distrobox`.

First, setup a Gentoo `distrobox` and enter it:

```
xenia ~ # distrobox create --root --image docker.io/gentoo/stage3 --name gentoo
xenia ~ # distrobox enter --root gentoo
```

Setup the container:

```
gentoo ~ # emerge --sync
```

The remainder of this guide takes place from within the Gentoo `distrobox`.

9.2 foxbuild installation

First, *install the Xenia overlay*.

Next, allow for `foxbuild` to be emerged:

```
gentoo ~ # echo "xenia-tools/foxbuild *" >> /etc/portage/package.accept_keywords/foxbuild
```

Install `foxbuild`:

```
gentoo ~ # emerge foxbuild
```

9.3 Creating a root image

`foxbuild` pulls from a git repo to get the required files to build. First, we will use the existing repo that Xenia builds from.

Run `foxbuild`:

```
gentoo ~ # foxbuild
```

Configuration options will appear.

9.3.1 catalyst_repo

Set the git repo to pull from here. Here, we will use the default of `https://gitlab.com/xenia-group/catalyst.git`.

9.3.2 git_branch

The git branch from the repo set above to use. Here, we will use the default of `unstable`.

9.3.3 target

Here, the target set to be built can be specified. This will typically be a spec file such as `stage4-systemd.spec`, but can also be a comma separated list of spec files, changed to only rebuild spec files that have changed in the last commit and `*` to build all targets. We will leave this at the default `stage4-systemd.spec`.

9.3.4 Using a config file

`foxbuild` can also take a config file instead of running interactively.

To do this, write a config file, here called `build.toml`:

```
catalyst_repo = "https://gitlab.com/xenia-group/catalyst.git"
git_branch = "unstable"
target = "stage4-systemd.spec"
```

Then, run `foxbuild` with `-c` specified:

```
gentoo ~ # foxbuild -c build.toml
```

Now, `foxbuild` will start setting up the build environment, generating a Portage snapshot and downloading the required seed `stage3` for the spec selected. It will also show where it is logging. To see the log in real-time, open another terminal and run `tail -f` on the log:

```
gentoo ~ $ tail -f /var/foxbuild/stage4-systemd.spec.log
```

9.4 Testing the image

`foxbuild` creates a repo in `/var/foxbuild/output`. First, expose the repo over a web server:

```
gentoo ~ $ cd /var/foxbuild/output
gentoo ~ $ python3 -m http.server
```

Next, from the installer or using `foxupdate` on a Xenia system, set the repository to `http://localhost:8000/` (or replace `localhost` with the IP/domain of the system hosting the repo).

Then, you can select the available targets that `foxbuild` has built.

9.5 Updating the build environment

foxbuild generates a Portage snapshot if there isn't already one on the system, but doesn't create a new one every time it is run. If the `git_repo` is changed, foxbuild will not pick this up until the build environment has been updated. To update the build environment, run foxbuild with `-u`:

```
gentoo ~ # foxbuild -uc build.toml
```

9.6 Customising the image

First, create a fork of [the catalyst repo](#). Then, follow the customisation guide [here](#) to customise the root.

Once changes have been made, push them to the repo. Then, set the `repository` in foxbuild config to match the repo created.

MANUAL ROOTFS GENERATION

We use catalyst to generate our rootFS images. Catalyst is a build tool made by Gentoo which is used to build Gentoo (and anything based on it). This page will walk you through making a root image of your own.

10.1 Requirements

- Gentoo installation (or chroot)
- `app-arch/pixz` and `sys-fs/squashfs-tools` installed
- Patience

10.2 First time setup

First, install `dev-util/catalyst` on your system.

Note: `dev-util/catalyst` may be masked when you read this. Make sure to unmask it by adding `dev-util/catalyst **` to `/etc/portage/package.accept_keywords/catalyst`

Create the directory `/var/tmp/catalyst/builds/default`:

```
xenia ~ # mkdir -p /var/tmp/catalyst/builds/default
```

Make sure `/var/cache/distfiles` is created:

```
xenia ~ # mkdir -p /var/cache/distfiles
```

Next, clone our catalyst repository:

```
xenia ~ $ git clone https://gitlab.com/xenia-group/catalyst
```

Enter the directory:

```
xenia ~ $ cd catalyst
```

Symlink the config directory:

```
xenia ~/catalyst # ln -sf $PWD/config /var/tmp/catalyst/
```

With catalyst setup, download a [stage3](#). To find which one is needed, look in the spec file you are building under `source_subpath`. For `openrc`, it is usually a standard `openrc stage3`. For `systemd`, it is usually `systemd-mergedusr`.

Finally, *add the Xenia overlay* to your system.

10.3 Create a snapshot

Run a portage sync:

```
xenia ~/catalyst # emerge-webrsync
```

Create the snapshot:

```
xenia ~/catalyst # catalyst -s stable
```

There will be an entry like the following:

```
NOTICE:catalyst:Wrote snapshot to /var/tmp/catalyst/snapshots/gentoo-
↳202a513901854da4c0f3bce7cfd8242e8194689d.sqfs
```

Get the snapshot hash (in this case 202a513901854da4c0f3bce7cfd8242e8194689d), and replace [SNAPSHOT_HASH] with the hash:

```
snapshot_treeish: 202a513901854da4c0f3bce7cfd8242e8194689d
```

10.4 Pick a spec file

Xenia has multiple spec files. For example, there is `stage4-openrc.spec` which will build an openrc GNOME image of Xenia. Or, there is `stage4-systemd-plasma.spec`, which will build a systemd Plasma/KDE image of Xenia.

10.5 Spec file setup

In the spec file selected earlier, change all instances of [CATALYST_DIR] with the directory the catalyst repo is - in this case `/home/xenia/catalyst/`:

```
stage4/root_overlay: /home/xenia/catalyst/overlay
stage4/fsscript: /home/xenia/catalyst/stage4-openrc.sh
```

10.6 Building

Build the stage4:

Note: `-a` here means no auto-resume. We specify this in-case of an error which requires a full rebuild. Auto-resume may be handy if you need to stop for whatever reason and come back later, so feel free to remove this flag

```
xenia ~/catalyst # catalyst -a -f [your spec file].spec
```

Note: If there is any errors building, the first port of call is to download a new stage3. If this doesn't work, try running catalyst again without `-a`. This will get around any issues with post-install of packages (`gtk+` being the main culprit here).

The stage4 will now be in `/var/tmp/catalyst/builds/default/stage4-[arch]-[target].tar.xz`. The target will be the target set in the spec file. The arch will be the architecture of the image (amd64 is most common here, unless `stage4-arm64-*` is being used as a spec file).

10.6.1 Notes

This process will be very slow the first time you build the stage4 - you will be compiling GCC, LLVM, a DE etc. However, catalyst keeps binary packages! So your next build will be very quick.

You can learn about customising your own rootFS [here](#).

10.7 Converting to squashFS

The stage4 you have made is currently a tarball. Let's convert it to a squashFS:

```
xenia ~/catalyst # rm root.img
xenia ~/catalyst # mksquashfs /var/tmp/catalyst/tmp/default/stage4-[arch]-[target] root.img -comp xz
```

10.8 Installation

First, host a server where your `root.img` is located:

```
xenia ~/catalyst $ python3 -m http.server
```

Note: This part is unfinished! There will be better update documentation soon

If you have an existing installation, you can simply replace `root.img` in `/roots` with your own. If you don't, follow the [guided installation](#), replacing the root image URL with your own.

CUSTOMISING THE ROOTFS

You can customise the rootFS with different packages, included files, USE flags - even different init systems. This page will give a small overview of what you can change.

11.1 Requirements

- A catalyst directory, setup according to *manual rootFS generation* or by *foxbuild*

Note: This page will reference files and directories relative to your catalyst directory. On `foxbuild`, this will be a fork of the catalyst repo.

Note: This guide uses the spec name `stage4-openrc.spec` as a placeholder for any spec name. Replace it with the spec being edited.

11.2 Packages

To add, change or remove packages in Xenia, edit `stage4-openrc.spec`:

```
stage4/packages:  
  app-admin/sudo  
  ...  
  sys-process/cronie
```

You can add your own packages here. Make sure they are spelled correctly and you include the package category.

11.3 USE flags and other portage config

11.3.1 Global USE flags

To add a global USE flag, edit `stage4-openrc.spec`:

```
stage4/use:
  -gnome-online-accounts
  ...
  video_cards_vmware
```

You can add USE flags here as well as removing USE flags with `-`.

11.3.2 Package specific USE flags

To add package specific USE flags, edit `config/package.use/stage4`:

```
sys-boot/grub device-mapper
...
```

You can add USE flags here the same way you do in a typical `package.use`. You can also change package masks, licenses and keywords here (for example to enable testing on a specific package).

11.4 Adding services

To add services to Xenia (the equivalent of `rc-update`), edit `stage4-openrc.spec`:

```
stage4/rcadd:
  cupsd|default
  ...
  udev|sysinit
```

You can specify the service here alongside the runlevel in the format `service|runlevel`.

On systemd specs, this is defined in the `fsscript`. See below for more details.

11.5 Custom shell script

To run custom script at the end of the build process, edit `stage4-openrc.sh`:

```
#!/bin/bash
...
```

You can add your own custom script at the end of this file.

11.6 Filesystem overlay

To add arbitrary files to Xenia, such as new backgrounds or configuration for packages, open `overlay` and add any files you want here. For example, to edit `/etc/fstab`, edit `overlay/etc/fstab`.

Multiple overlays can be added. See `stage4-systemd.spec` for an example. It applies both the common overlay that is shared amongst all spec files and a GNOME specific overlay.

These files will be added after catalyst has completed emerging the system.

11.7 Submitting changes to Xenia

If you make a change and you want it merged into Xenia, please follow these guidelines:

- Have a defined purpose to all your changes and reasoning for why it should be added to the base system
- Make sure added files, packages and USE flags are not specific to your system
- Put the services, packages and global USE flags in alphabetical order

You can then submit an MR to the [catalyst repository](#) with your changes.

FILESYSTEM LAYOUTS

Xenia Linux supports a wide range of filesystem layouts and types, but has certain requirements to boot. This page will show the current supported layouts and what is needed to boot.

12.1 Traditional label-based layout (generic)

This layout is the current layout used for LVM setups and can be used on any filesystem that supports labels.

12.1.1 Physical map

```
d1
├─d1p1 - EFI - 512M
├─d1p2 - ROOTS - 8G
├─d1p3 - OVERLAY - around 30% remaining space
└─d1p4 - HOME - all remaining space
```

12.1.2 Description

This layout contains 4 partitions:

- p1 labelled EFI for grub
- p2 labelled ROOTS for root image storage
- p3 labelled OVERLAY for filesystem overlays
- p4 labelled HOME for /home

EFI, ROOTS, OVERLAY and HOME labels are necessary for Xenia to boot this layout. If OVERLAY is not present, Xenia will boot into recovery.

12.2 Traditional label-based layout (LVM)

12.2.1 Physical map

```
d1
├─d1p1 - EFI - 512M
└─d1p2 - LVM - all remaining space
```

12.2.2 Logical map

```
d1p2
├─vg0-roots - ROOTS - 8G
├─vg0-overlay - OVERLAY - around 30% remaining space
└─vg0-home - HOME - all remaining space
```

12.2.3 Description

This layout contains 2 partitions:

- p1 labelled EFI for grub
- p2 for LVM

This layout contains 3 LVs:

- vg0-roots labelled ROOTS for root image storage
- vg0-overlay labelled OVERLAY for filesystem overlays
- vg0-home labelled HOME for /home

EFI, ROOTS, OVERLAY and HOME labels are necessary for Xenia to boot this layout.

12.3 Btrfs

This layout is for Btrfs based systems.

12.3.1 Physical map

```
d1
├─d1p1 - EFI - 512M
└─d1p2 - ROOTS - all remaining space
```

12.3.2 Logical map

```
d1p2 - ROOTS
├─overlay - btrfs subvolume
│  └─etc - btrfs subvolume
│    └─var - btrfs subvolume
│      └─usr - btrfs subvolume
└─home - btrfs subvolume
```

12.3.3 Description

This layout contains 2 partitions:

- p1 labelled EFI for grub
- p2 for Btrfs labelled ROOTS

This layout contains 5 subvolumes alongside the root of the Btrfs:

- root of the Btrfs used for root image storage
- overlay for filesystem overlays
 - etc for etc overlay
 - var for var overlay
 - usr for usr overlay
- home for /home

BTRFS needs to have the root images stored in its root, not in a subvolume. overlay, etc, var, usr and home need to be named as such for Xenia to mount them.

Note: Overlays on a btrfs system look like this:

```
/overlay/etc/etc /overlay/etc/etcw
```

```
/overlay/var/var /overlay/etc/etcw
```

due to both the upper directory and work directory having to be on the same filesystem. Don't worry about making these yourself (foxmount will do it for you on first boot), just make the subvolumes.

12.4 Btrfs with encryption

This layout is for Btrfs/LUKS based systems.

12.4.1 Physical map

```
d1
├─d1p1 - EFI - 512M
├─d1p2 - ROOTS - 8G
└─d1p3 - XENIA - all remaining space
```

12.4.2 Logical map

```
d1p3 - XENIA (crypt)
├─overlay - btrfs subvolume
│   └─etc - btrfs subvolume
│   └─var - btrfs subvolume
│   └─usr - btrfs subvolume
└─home - btrfs subvolume
```

12.4.3 Description

This layout contains 3 partitions:

- p1 labelled EFI for grub
- p2 labelled ROOTS for root image storage
- p3 for Btrfs/LUKS labelled XENIA

This layout contains 5 subvolumes alongside the root of the Btrfs:

- overlay for filesystem overlays
 - etc for etc overlay
 - var for var overlay
 - usr for usr overlay
- home for /home

This layout comprises of a ext4/xfs/any other labelled filesystem for root image storage, and a LUKS encrypted partition formatted as Btrfs for overlays and home.

Note: Overlays on a btrfs system look like this:

```
/overlay/etc/etc /overlay/etc/etcw
```

```
/overlay/var/var /overlay/etc/etcw
```

due to both the upper directory and work directory having to be on the same filesystem. Don't worry about making these yourself (foxmount will do it for you on first boot), just make the subvolumes.

OPENRC FLATPAK WORKAROUND

In OpenRC, flatpaks do not open properly if they are Dbus enabled, as the Dbus service file [isn't able to be found](#).

To fix this, create a symlink like this:

```
xenia ~ $ ln -sf /var/lib/flatpak/exports/share/dbus-1/ /home/$USER/.local/share/
```