
gentoo_configs

Config files for my gentoo installation.

Gentoo Installation

Booting

Download the iso from gentoo site, and burn it to a disc. Booting from the live installation media, boot into the gentoo kernel

```
1 boot: gentoo
```

Network

Check if the network is setup automatically with `ifconfig`. Test the network with ping.

```
1 ping -c 3 www.google.com
```

Partition Scheme (UEFI)

When installing Gentoo on a system that uses UEFI to boot the operating system (instead of BIOS), then it is important that an EFI System Partition (ESP) is created. The instructions for parted below contain the necessary pointers to correctly handle this operation.

```
1 mkfs.fat -F 32 /dev/sda2
```

We are going to follow the following partitioning scheme for our gentoo installation.

Partition	Filesystem	Size
/dev/sda1	(bootloader)	2M
/dev/sda2	ext2 (or fat32 if UEFI is being used)	128M
/dev/sda3	(swap)	512M or higher
/dev/sda4	ext4	Rest of the disk

We use cfdisk for partitioning the disk

```
1 cfdisk /dev/sda
```

Creating Filesystems

We use ext4 filesystem for our partitions.

```
1 mkfs.ext4 /dev/sda2
2 mkfs.ext4 /dev/sda4
```

mkfs.ext4 command comes from the package `sys-fs/e2fsprogs`
mkfs.ntfs command comes from the package `sys-fs/ntfs-3g`

Activating SWAP partition

```
1 mkswap /dev/sda3
2 swapon /dev/sda3
```

Mounting the ROOT partition

```
1 mount /dev/sda4 /mnt/gentoo
```

Installing the Gentoo installation files

Installing a stage tarball

We first set the date and time

```
1 ntpd -qg
2 date
```

We are going to use the multilib(both 32 and 64 bit support) tarball for our installation. We download the tarball using a terminal based web browser “links”.

```
1 cd /mnt/gentoo
2 links https://www.gentoo.org/downloads/mirrors/
```

Unpacking the stage tarball

```
1 tar xpvf stage3-*.tar.xz --xattrs-include='*.*' --numeric-owner
```

Configuring compile options

ptimize Gentoo, it is possible to set a couple of variables which impacts the behavior of Portage, Gentoo's officially supported package manager. All those variables can be set as environment variables (using export) but that isn't permanent. To keep the settings, Portage reads in the `/etc/portage/make.conf` file, a configuration file for Portage.

You can read up more on this on the [gentoo handbook.] (<https://wiki.gentoo.org/wiki/Handbook:AMD64/Installation/St>)

I have make.conf file setup specifically for my system. And it is available in this repository. We are going to copy it to `/etc/portage` and use it. The same make.conf file would not work for all systems, as I have it setup specifically for my current running system.

Installing the Gentoo base system

Chrooting

Select a mirror for downloading your installation files

```
1 mirrorselect -i -o >> /mnt/gentoo/etc/portage/make.conf
```

Gentoo ebuild repository

```
1 mkdir --parents /mnt/gentoo/etc/portage/repos.conf
2 cp /mnt/gentoo/usr/share/portage/config/repos.conf /mnt/gentoo/etc/
  portage/repos.conf/gentoo.conf
```

We already have a repos.conf in this repository setup as my personal preferences, so we can use that.

Copy DNS info

```
1 cp --dereference /etc/resolv.conf /mnt/gentoo/etc/
```

Mounting the necessary filesystems

```
1 mount --types proc /proc /mnt/gentoo/proc
2 mount --rbind /sys /mnt/gentoo/sys
3 mount --make-rslave /mnt/gentoo/sys
4 mount --rbind /dev /mnt/gentoo/dev
5 mount --make-rslave /mnt/gentoo/dev
```

Entering the new environment

```
1 chroot /mnt/gentoo /bin/bash
2 source /etc/profile
3 export PS1="(chroot) ${PS1}"
```

Mounting the boot partition

```
1 mount /dev/sda2 /boot
```

Configuring Portage

Installing a Gentoo ebuild repository snapshot from the web

```
1 emerge-webrsync
```

We are also going to update the ebuild repository

```
1 emerge --sync
```

Choosing the right profile

A profile is a building block for any Gentoo system. Not only does it specify default values for USE, CFLAGS, and other important variables, it also locks the system to a certain range of package versions. These settings are all maintained by Gentoo's Portage developers.

We check the profiles using the following command

```
1 eselect profile list
```

Viewing the list, we select a profile as per our needs

```
1 eselect profile set 2
```

Updating the @world set

At this point, it is wise to update the system's @world set so that a base can be established.

This following step is necessary so the system can apply any updates or USE flag changes which have appeared since the stage3 was built and from any profile selection:

```
1 emerge --ask --verbose --update --deep --newuse @world
2
3 OR, emerge -DuvaN @world
```

Configuring the USE variable

You can view your USE variables using the following

```
1 emerge --info
```

You can also read up on USE flags more here

```
1 less /var/db/repos/gentoo/profiles/use.desc
```

We already have our USE flags configured in our make.conf file. But Setting the flags is very important as it gives a lot of control over support for what is installed on your system and what is not, and makes your system lightweight.

Timezone

```
1 ls /usr/share/zoneinfo
2 echo "Europe/Brussels" > /etc/timezone
3 emerge --config sys-libs/timezone-data
```

Configure locales

```
1 nano -w /etc/locale.gen
```

Uncomment the following for English

```
1 en_US ISO-8859-1
2 en_US.UTF-8 UTF-8
```

And then, generate the locales

```
1 locale-gen
```

Locale selection

```
1 eselect locale list
```

From the list, select the preferred locale

```
1 eselect locale set <locale number>
```

Now, reload the environment.

```
1 env-update && source /etc/profile && export PS1="(chroot) ${PS1}"
```