Specific Linux distributions, system virtual machines, RV64GQVH_S, hardware-assisted virtualisation and OS-level virtualisation with Docker

Adam Boult (www.bou.lt)

March 23, 2024

Contents

I Introduction **2**

1 Introduction **3**

II Package management **4**

2 Debian, Advanced Package Tool (APT), and dpkg **5**

3 Fedora, RPM Package Manager, Yellowdog Updater Modified (yum) and Dandified YUM (DNF) **8**

4 Arch linux and pacman **9**

5 Gentoo, portage and emerge **13**

6 Slackware **14**

7 NixOS and Nix **15**

8 Void Linux and the X Binary Package System (XBPS) **16**

9 OpenSUSE and zypper **17**

10 openWRT and opkg **18**

11 Alpine Linux **19**

III Distributions based on other distributions **20**

12 Ubuntu, Personal Package Archives (PPAs) and Snap **21**

IV Meta distributions **22**

13 Qubes **23**

V Other package managements **24**

14 Flatpak **25**

15 AppImage **26**

VI Virtual Machines **27**

16 QEMU and Kernel-based Virtual Machine (KVM) **28**

VII Hypervisors **30**

VIII OS-level virtualization with Docker **31**

17 Docker **32**

Part I

Introduction

Chapter 1

Introduction

1.1 Introduction

1.1.1 Introduction

/etc/os-release

Part II

Package management

Chapter 2

Debian, Advanced Package Tool (APT), and dpkg

2.1 Installing and uninstalling with apt-get

2.1.1 apt-get

package updates

to get list of packages possible updates

```
apt-get update
```

to upgrade packages

```
apt-get upgrade
```

debian install package

```
apt-get install <package>
```

purge removes conf files too

```
apt-get remove <package>
apt-get purge <package>
```

2.1.2 sources.list

/etc/apt/sources.list

/etc/apt/sources.list.d/

2.2 Building from source

2.2.1 Installing from source

rebuild package from source:

Download the source:

```
apt-get source <package>
```

gets dependencies of building package

```
apt-get build-dep <package>
```

```
dpkg-buildpackage -rfakeroot -uc -b (from source code folder)
```

the above results in a .deb file

```
dpkg -i <package_file>.deb
```

2.3 Other

2.3.1 apt

apt exists as alternative to apt-get and apt-cache. front end for it, somewhat more user friendly

2.3.2 SORT

```
apt list --installed
```

dpkg, apt, aptitude, sources.list

2.3.3 apt-src

Streamlined way to install from source.

```
apt-src install <package>
```

downloads vanilla source, dsc (?), a .changes file, source tree?

```
apt-src build <package>
dpkg --install <path to compiled .deb>
```

or:

```
apt-src --build install <package>
```

does all in one line

2.3.4 apt-mirror

not installed by default on debian or ubuntu

Allows you to set up a mirror repository of packages.

Can then point sources.list for systems to point to the mirror.

Chapter 3

Fedora, RPM Package Manager, Yellowdog Updater Modified (yum) and Dandified YUM (DNF)

3.1 Introduction

3.1.1 RPM Package Manager (RPM)

.rpm files

3.1.2 Yellowdog Updater Modified (yum)

3.1.3 Dandified YUM (DNF)

3.1.4 Hosting a local repository using reposync

fedora silverblue: immutable version of fedora. flatpak for stuff?

Chapter 4

Arch linux and pacman

4.1 pacman

4.1.1 pacman -S: Installing and updating packages

pacman -S (sync) is family of commands

to refresh packages and upgrade (u upgrade, y, download database from remote)

```
pacman -Syu
```

install it

```
pacman -Syu my_package
```

This syncs as installing, which is safer. The following is less safe but can also be done:

```
pacman -S my_package
```

You can search for packages:

```
pacman -Ss string_in_package
```

4.1.2 pacman -S: Managing the cache

cache is stored in pacman in /var/cache/pacman/pkg/

/etc/pacman.conf

can clear cache of uninstalled packages with

```
pacman -Sc
```

double clean to be more aggressive (remove cache of installed packages)

```
pacman -Scc
```

4.1.3 pacman -Q

Query

Info on a package, including what installed packages depends on it.

```
pacman -Qi <package>
```

list of explicitly installed:

```
pacman -Qe
```

To see packages installed without the official repository (eg AUR) use

```
pacman -Qm
```

Packages which were installed as dependencies

```
pacman -Qd
```

Packages which are dependencies and orphans

```
pacman -Qdt
```

list of local files associated with package:

```
pacman -Ql <package_name>
```

4.1.4 pacman -R

remove it

```
pacman -Rns my_package
```

The -s flag removes dependencies which are no longer needed.

The -n flag removes config files.

Can just remove the package and not dependencies:

```
pacman -R my_package
```

4.2 Manually installing packages using the Arch Build System (ABS) and makepkg

4.2.1 Getting the build instructions

```
git clone https://gitlab.archlinux.org/archlinux/packaging/packages/apache.git
```

4.2.2 makepkg

From the folder with the PKGBUILD, can run makepkg

/etc/makepkg.conf

makepkg -si to sync dependencies and install

Running "pacman -syu" will replace these packages with the repository ones. The PKGBUILD file can be amended to prevent that.

4.3 The Arch User Repository (AUR)

4.3.1 Introduction

Can pull from AUR.

```
git clone https://aur.archlinux.org/gzdoom.git
```

And then build as before.

As these packages are not in the official repositories, running pacman -Syu will not replace them.

Packages can be updated by pulling the git repo and rerunning makepkg as before.

Can see explicitly installed non-pacman using

```
pacman -Qm
```

4.4 Yet another yogurt (yay)

4.4.1 Yet another yogurt (yay)

Can use as a wrapper around pacman, and to download and update packages from the AUR.

Can update yay pacakges with

```
yay -Syu
```

Or just running:

```
yay
```

To just update AUR packages:

```
yay -Sau
```

Can search eg for firefox by doing

```
yay firefox
```

If you know the exact name you can do

```
yay -Syu firefox
```

Can uninstall

```
yay -Rns firefox
```

to remove unneeded dependencies.

```
yay -Yc
```

to remove cache, on both yay and pacman

```
yay -Sc
```

See status of installed packages

```
yay -Ps
```

4.5 Other tools

4.5.1 paccache

paccache is a separate package. Not available by default.

can remove all but last 3 with either of:

```
paccache -r
paccache -rk3
```

To remove all uninstalled:

```
paccache -rk0 remove all uninstalled
```

4.5.2 asp

Not installed by default. Alternative. to get build instructions:

```
asp export <package>
```

to get code:

```
asp checkout <package>
```

Chapter 5

Gentoo, portage and emerge

5.1 Introduction

5.1.1 Emerge

5.1.2 Portage

Chapter 6

Slackware

6.1 Introduction

6.1.1 Introduction

Chapter 7

NixOS and Nix

7.1 Introduction

7.1.1 Introduction

Chapter 8

Void Linux and the X Binary Package System (XBPS)

8.1 Introduction

8.1.1 Introduction

Chapter 9

OpenSUSE and zypper

9.1 Introduction

9.1.1 Introduction

To update packages:

zypper patch

To install a package

zypper install mplayer

To remove a package

zypper remove mplayer

Chapter 10

openWRT and opkg

10.1 Introduction

10.1.1 OpenWRT

10.1.2 libreCMC

Chapter 11

Alpine Linux

11.1 Introduction

11.1.1 Introduction

```
apk update
apk upgrade
```

These can be combined.

```
apk -U upgrade
```

Can install new packages.

```
apk add vim
```

Part III

Distributions based on other distributions

Chapter 12

Ubuntu, Personal Package Archives (PPAs) and Snap

12.1 Introduction

12.1.1 Introduction

Based on Debian

12.1.2 Personal Package Archives (PPAs)

12.1.3 Snap

12.1.4 Other

vi on ubuntu is actually vim

Part IV

Meta distributions

Chapter 13

Qubes

13.1 Introduction

13.1.1 Introduction

Part V

Other package managements

Chapter 14

Flatpak

14.1 Introduction

14.1.1 Introduction

List installed packages

```
flatpak list
```

To update packages

```
flatpak update
```

uninstall:

```
flatpak uninstall org.gimp.GIMP
```

Chapter 15

AppImage

15.1 Introduction

15.1.1 Introduction

Part VI

Virtual Machines

Chapter 16

QEMU and Kernel-based Virtual Machine (KVM)

16.1 Introduction

16.1.1 QEMU

```
qemu-img create -f raw ./image\_file 4G
qemu-img create -f qcow2 ./image\_file 4G
```

Can also use dd or fallocate.

Can resize

```
qemu-img resize disk_image +10G
qemu-img resize --shrink disk_image -10G
```

Install:

```
qemu-system-x86_64 -cdrom iso_image -boot order=d -drive file=disk_image,format=raw
```

Run

```
qemu-system-x86_64 options disk_image
```

16.1.2 KVM

```
qemu-system-x86_64 -accel kvm -cdrom iso_image -boot order=d -drive file=disk_image,format=r

qemu-system-x86_64 -accel kvm options disk_image
```

16.1.3 3D drivers

```
-device virtio-vga-gl
```

Part VII

Hypervisors

Part VIII

OS-level virtualization with Docker

Chapter 17

Docker

17.1 Pulling docker images and running them as containers

17.1.1 Pulling images

```
docker pull alpine:latest
```

List images:

```
docker image ls
```

Or:

```
docker images
```

To remove an image:

```
docker image remove alpine:latest
```

To remove all images (without an associated container):

```
docker image prune --all
```

17.1.2 Running images as containers

If the image is not already pulled, it will automatically be pulled, and so there is generally no need to manually pull images.

```
docker container create --name container_name alpine:latest
docker create --name container_name alpine:latest
```

If no name is provided, a random one will be created.

Once a container has been created, it can be started.

List containers. The "a" flag makes it show all containers, not just those running.

```
docker ps -a
```

```
docker container start container_name
docker start container_name
```

We can run it interactively and with a TTY.

```
docker container start --interactive --tty container_name
docker start -it container_name
```

Run can be used instead of create and start.

```
docker container run -it --name container_name alpine:latest
docker run -it --name container_name alphine:latest
```

Stopping containers.

```
sudo docker kill $(sudo docker ps -q)
```

Removing containers.

```
sudo docker rm $(sudo docker ps -a -q)
sudo docker system prune -af (this does much more than other stuff, saved lots of space. wha
```

17.1.3 Working without root

17.2 Building images from dockerfiles

17.2.1 Docker files and building images

First, build the images.

```
docker build -t "ae:tensorflow" -f ./docker/tf/Dockerfile_jetson.gpu .
```

Build

```
docker build -t localhost:32000/homepage-nodejs -f ./docker/web/Dockerfile .
```

17.3 Detaching containers

17.3.1 Detaching

```
docker run --detach
docker run -d
```

17.3.2 Running on reboot

making things start on reboot

```
docker run -d --restart=always
```

17.3.3 SSH into detached containers

```
docker exec -it <container_name> /bin/bash
```

17.4 Registry

17.4.1 Pushing images to repos

```
docker run -d -p 5000:5000 --restart=always --name registry registry:2
docker tag ubuntu:16.04 localhost:5000/my-ubuntu
docker push localhost:5000/my-ubuntu
docker pull localhost:5000/my-ubuntu
docker image remove localhost:5000/my-ubuntu

sudo docker push localhost:32000/homepage-nodejs

docker container stop registry
docker container stop registry && docker container rm -v registry
```

17.5 Volumes

17.5.1 Introduction

17.6 Network

17.6.1 Introduction

17.7 Docker compose

17.7.1 Docker Compose

```
sudo docker-compose build --no-cache
sudo docker-compose up --detach
```

can delete old databases if interacting badly

```
sudo rm -rf /data/db/artificialeconomist_mongo
```