Introduction to Docker

Using a Raspberry Pi 4

d

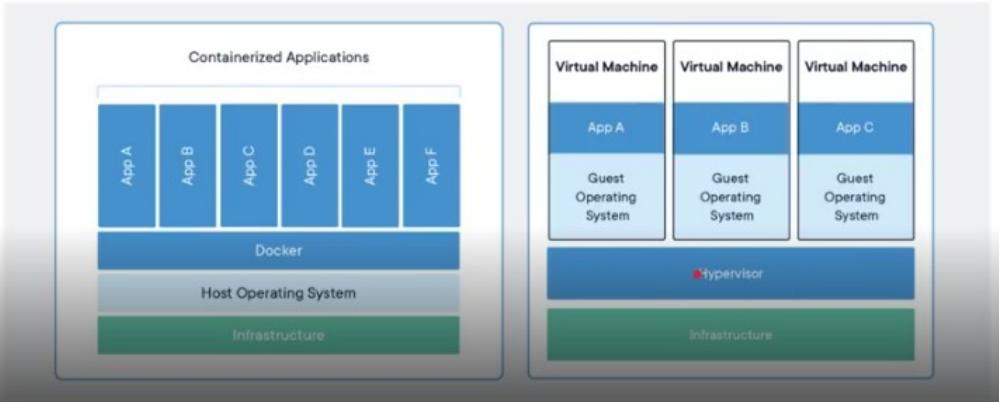
docker

docker.mp4

File | /home/devel/gary_explains_docker/docker.mp4

Container

- Abstraction of user environment that packages code and dependencies together.
- Not a virtual machine



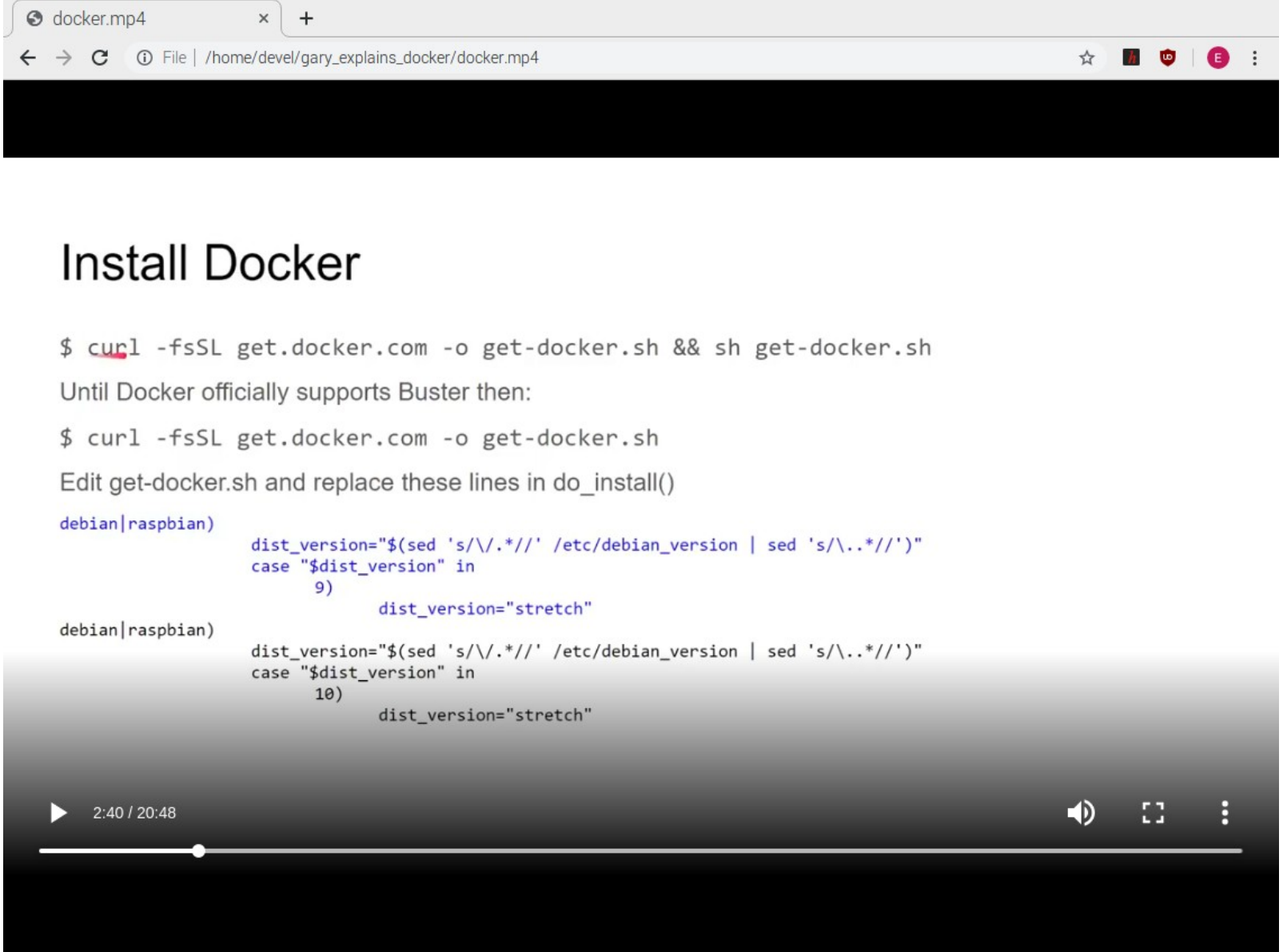
0:38 / 20:48

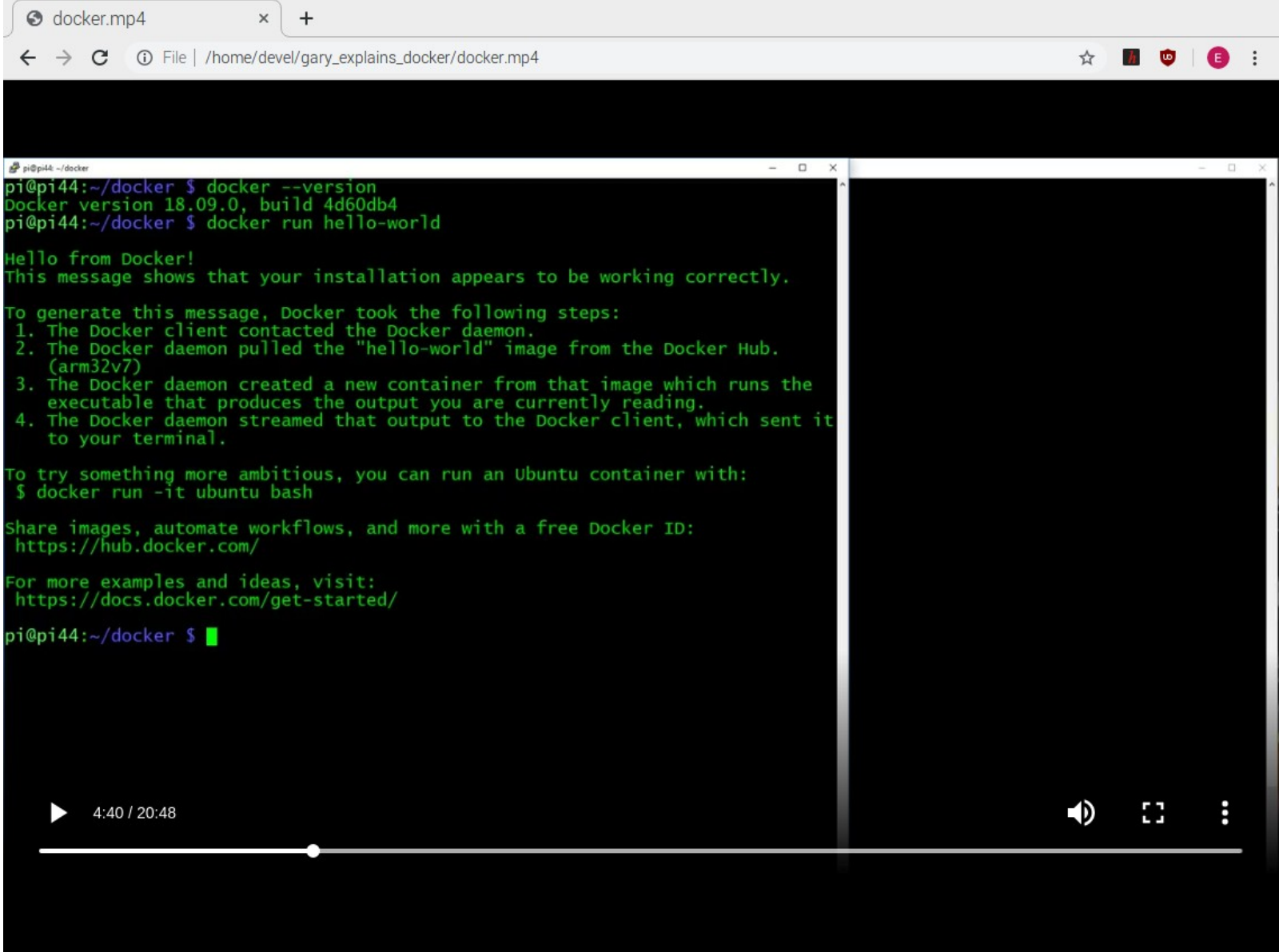
docker.mp4

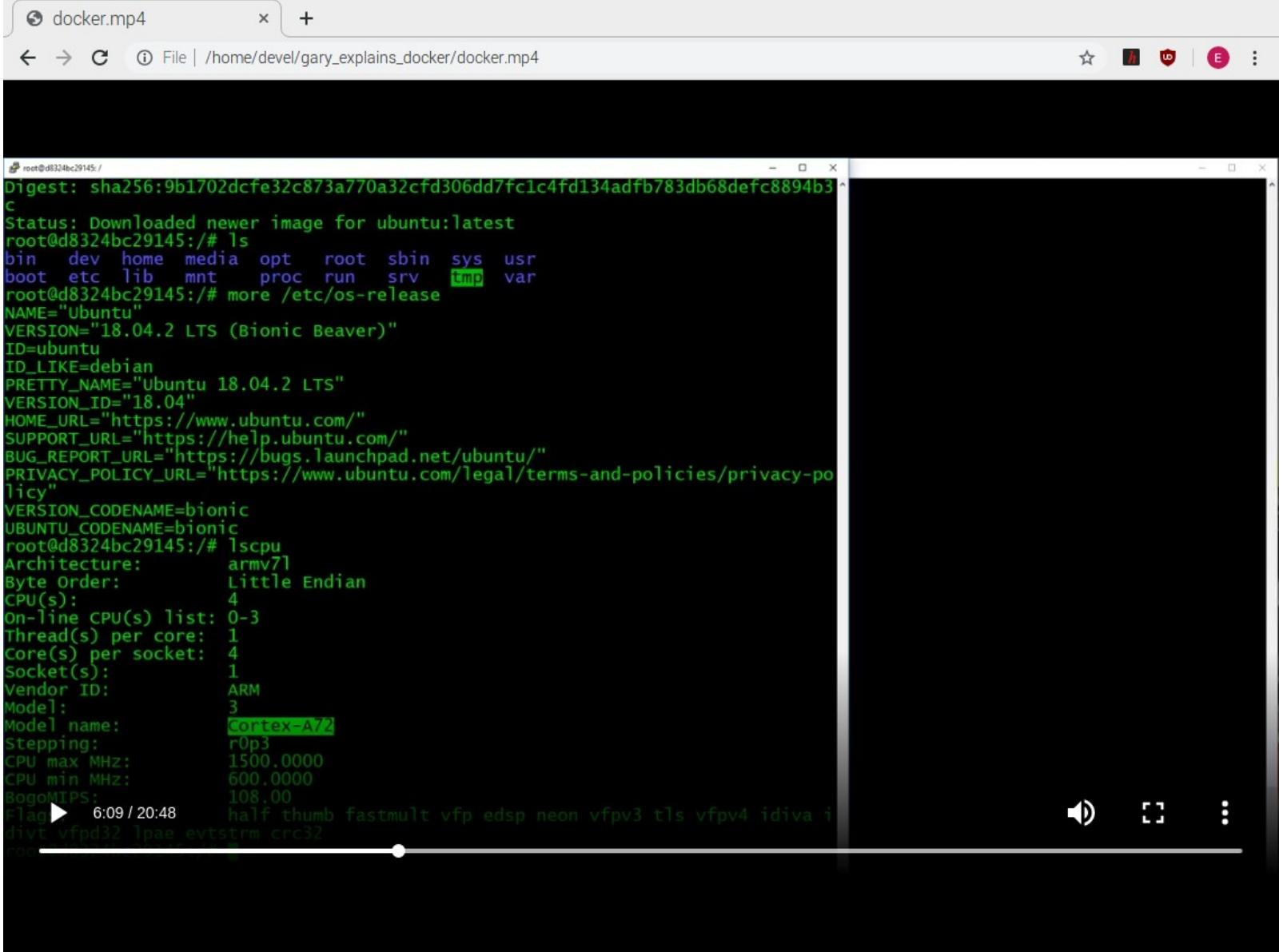
File | /home/devel/gary_explains_docker/docker.mp4

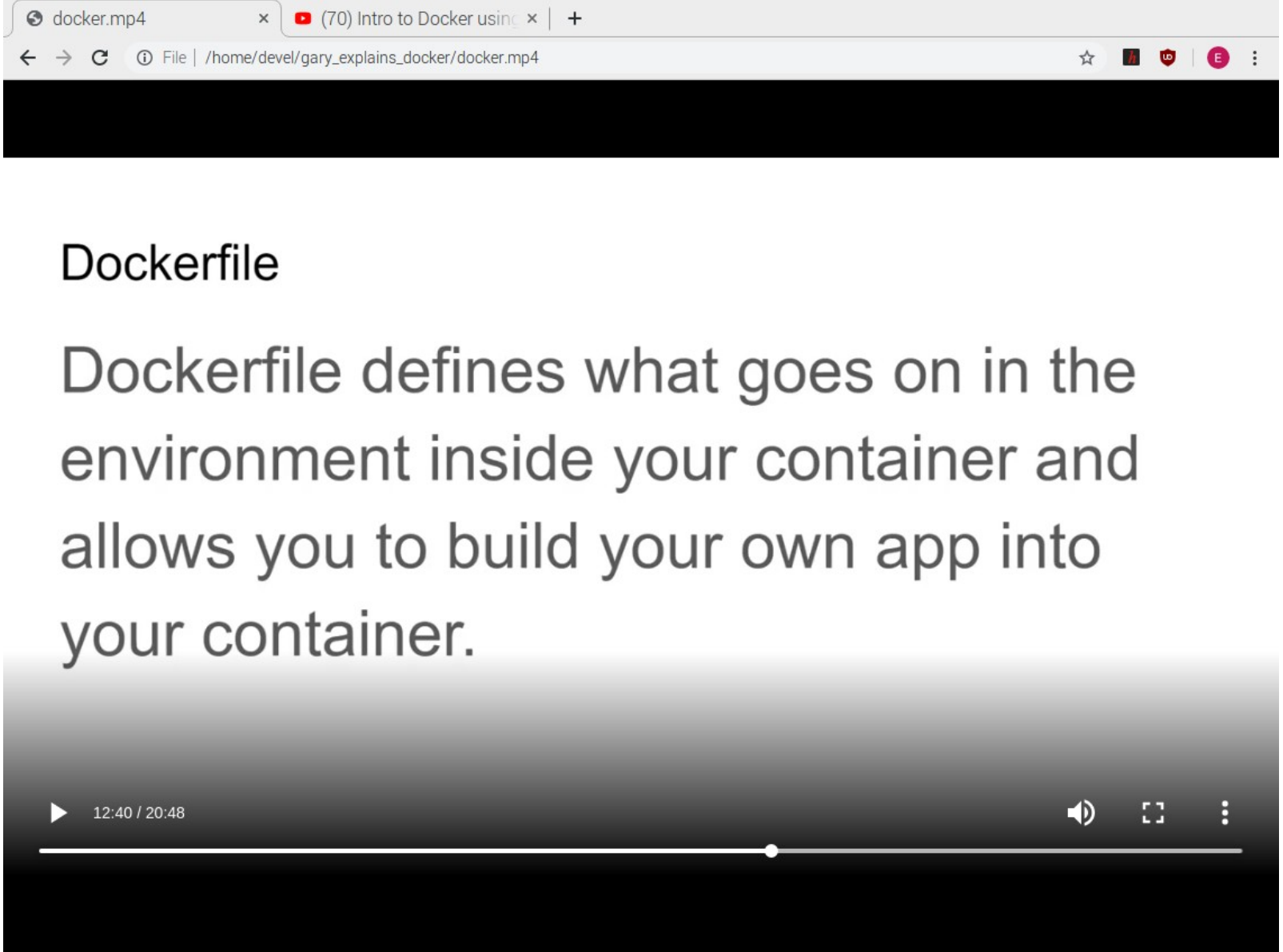
Obligatory picture of a container

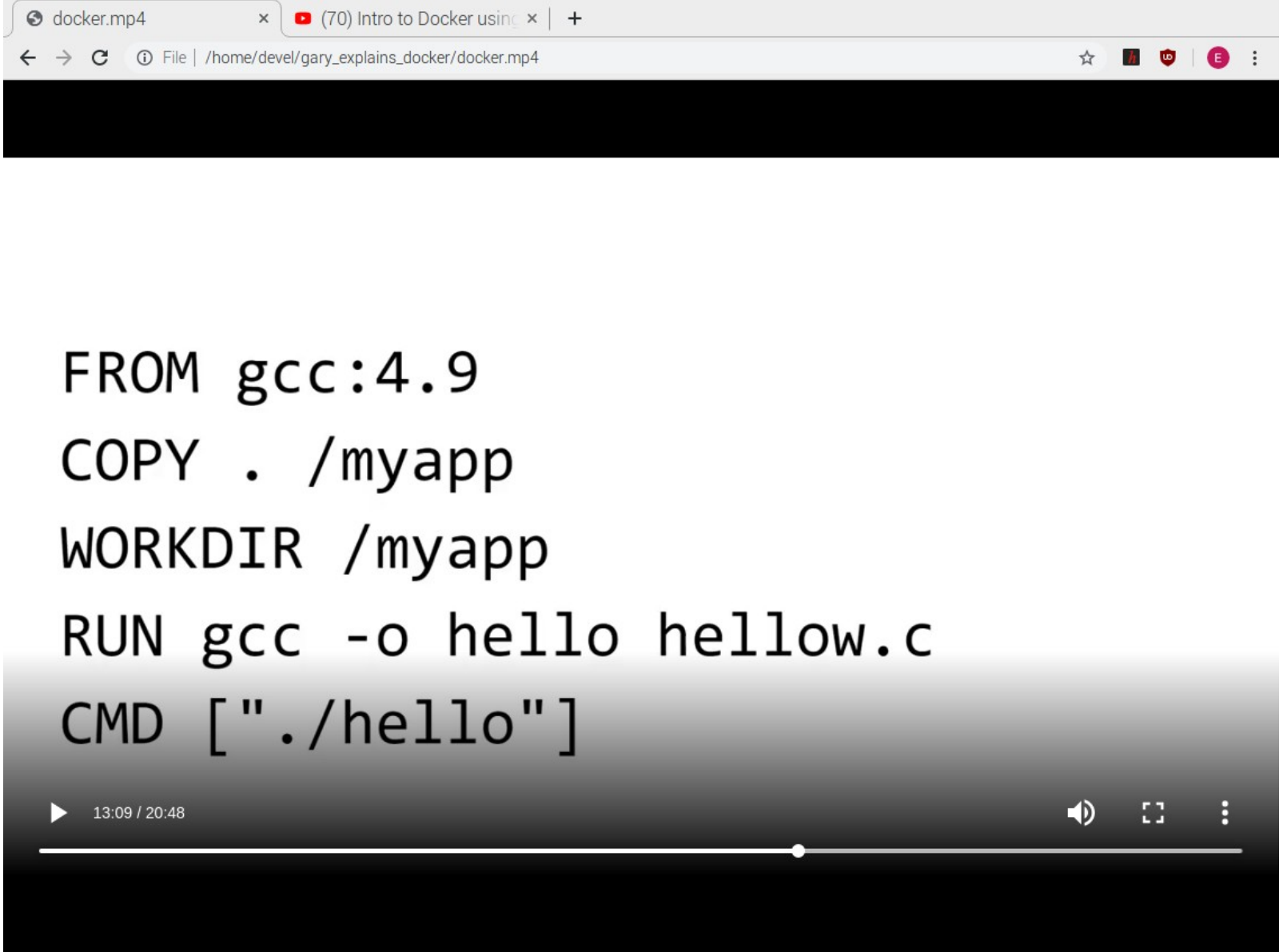
2:02 / 20:48

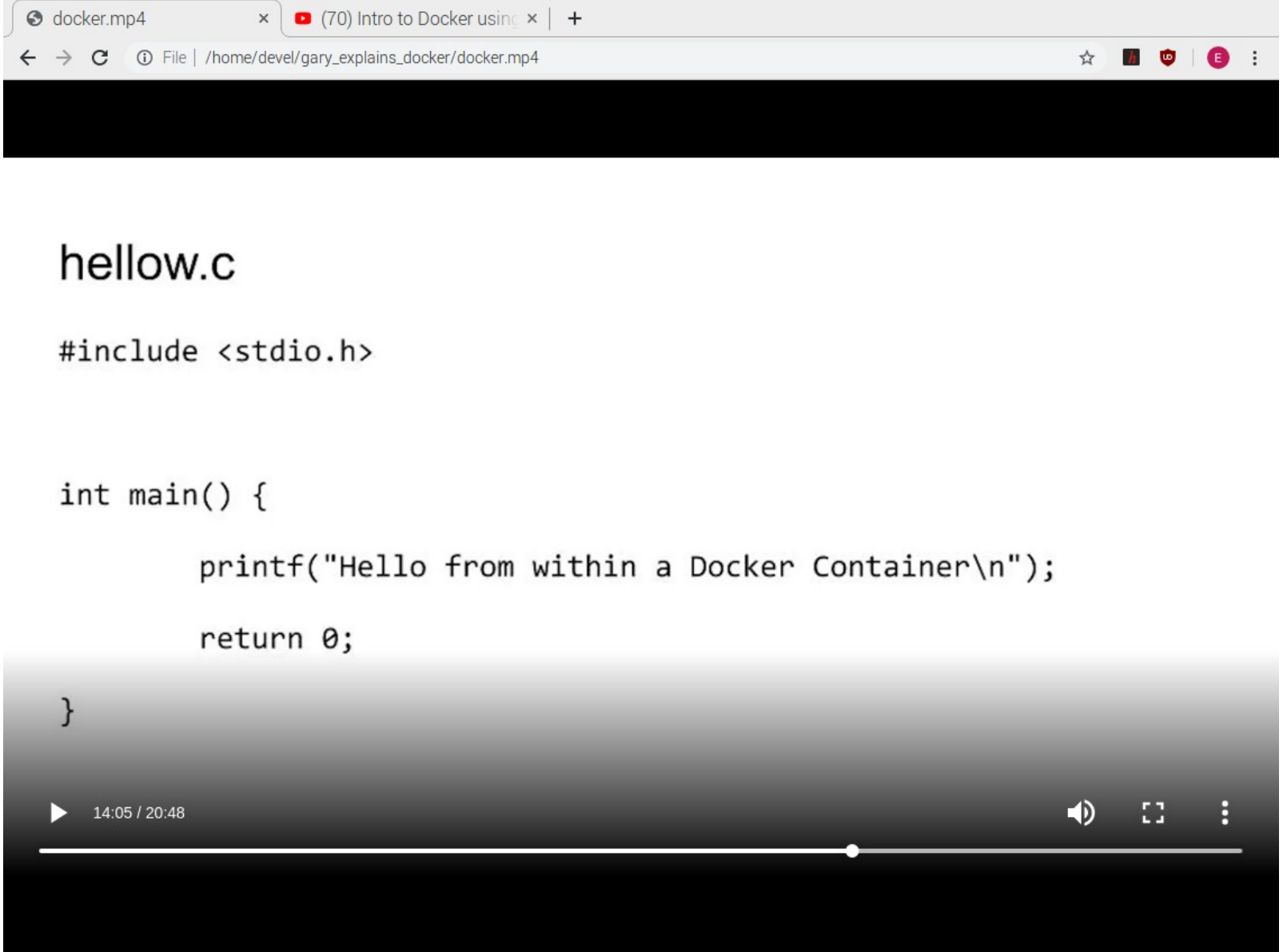

docker.mp4
File | /home/devel/gary_explains_docker/docker.mp4
Install Docker
$ curl -fsSL get.docker.com -o get-docker.sh && sh get-docker.sh
Until Docker officially supports Buster then:
$ curl -fsSL get.docker.com -o get-docker.sh
Edit get-docker.sh and replace these lines in do_install()
debian|raspbian)
dist_version="$(sed 's/\/.*//' /etc/debian_version | sed 's/\..*//')"
case "$dist_version" in
9)
dist_version="stretch"
debian|raspbian)
dist_version="$(sed 's/\/.*//' /etc/debian_version | sed 's/\..*//')"
case "$dist_version" in
10)
dist_version="stretch"
2:40 / 20:48

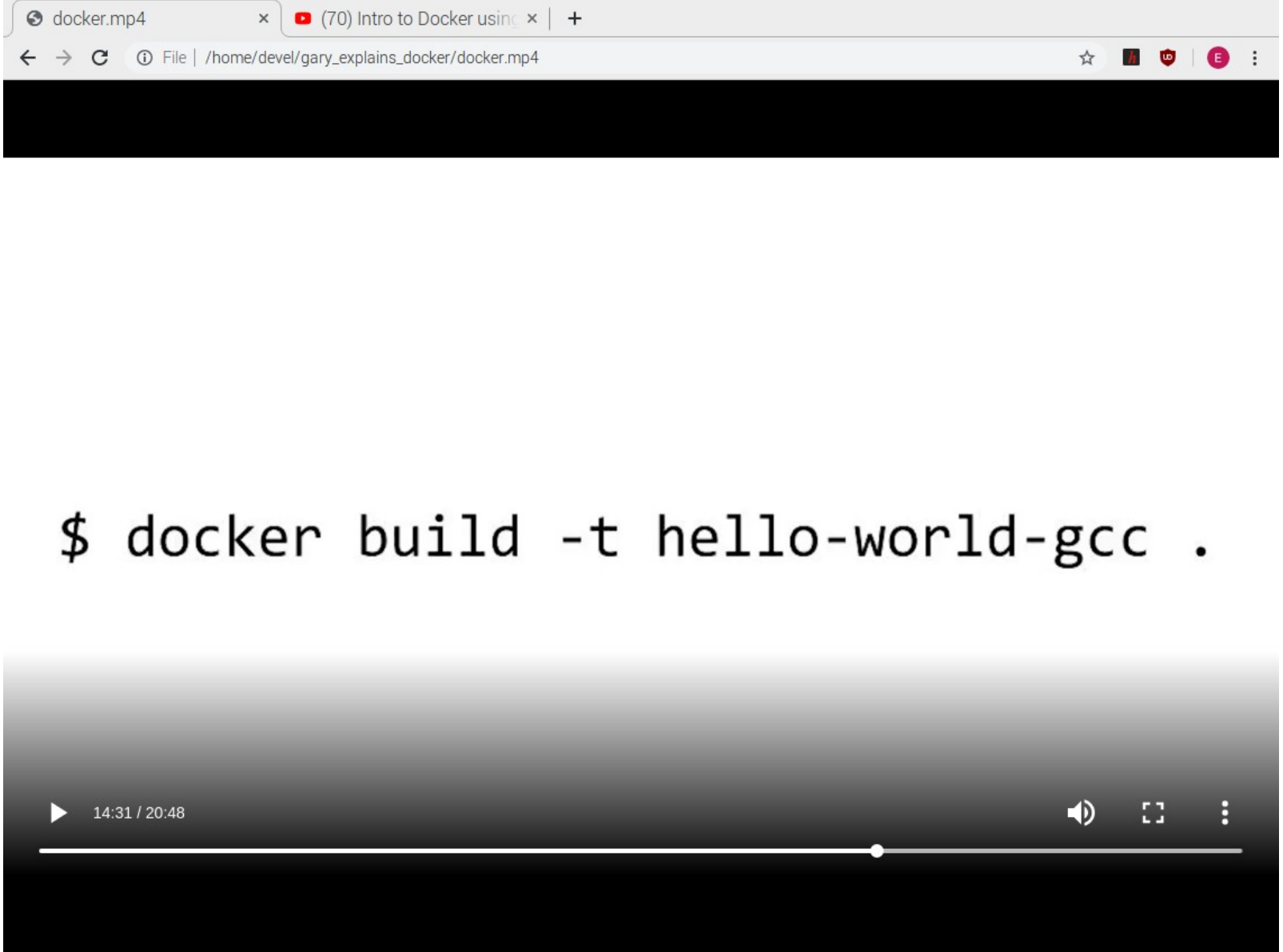

docker.mp4
File | /home/devel/gary_explains_docker/docker.mp4
pi@pi44: ~/docker
pi@pi44:~/docker $ docker --version
Docker version 18.09.0, build 4d60db4
pi@pi44:~/docker $ docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(arm32v7)
3. The Docker daemon created a new container from that image which runs the
executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
to your terminal.
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
pi@pi44:~/docker $
4:40 / 20:48

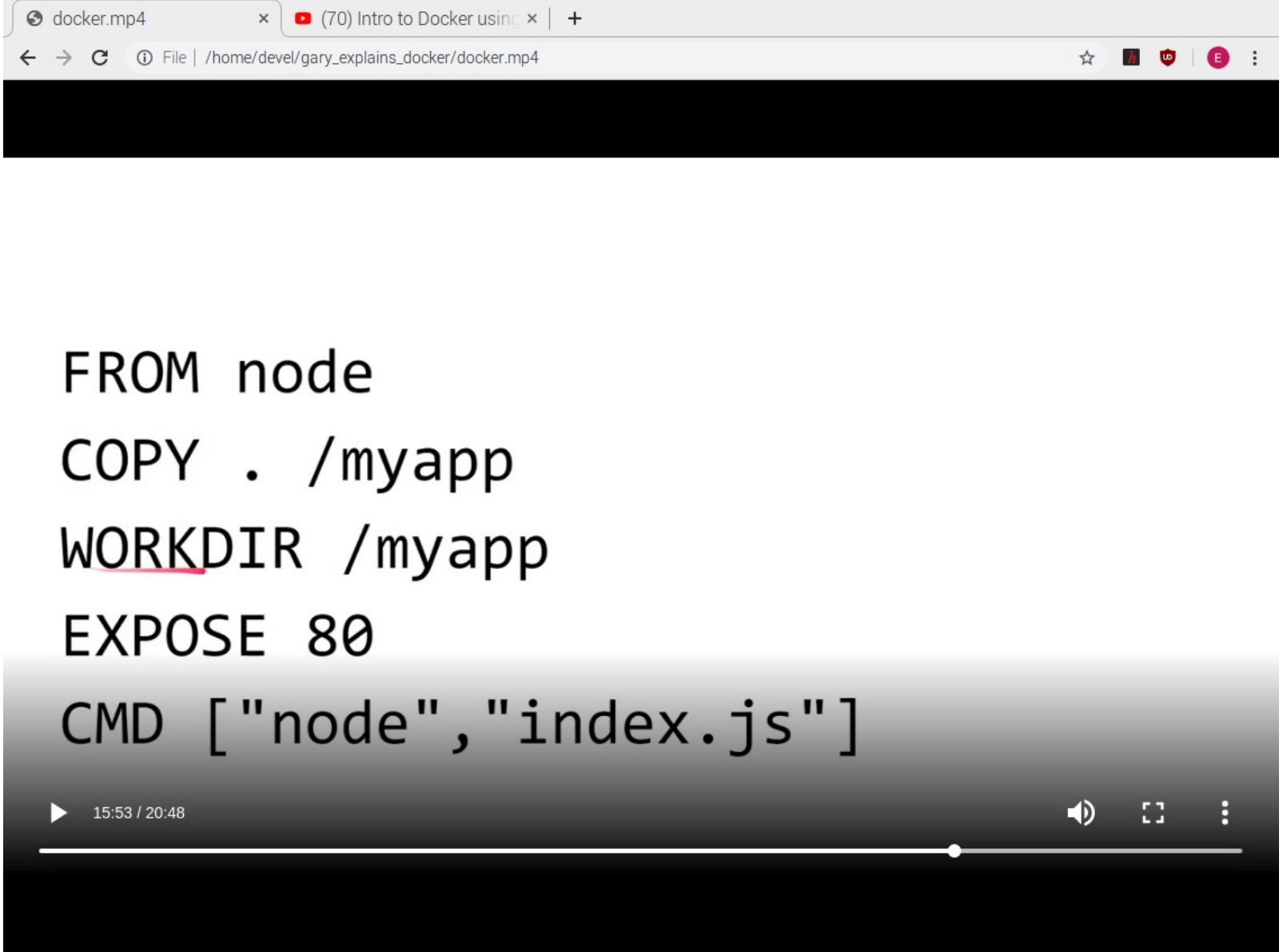

docker.mp4
File | /home/devel/gary_explains_docker/docker.mp4
root@d8324bc29145: /
Digest: sha256:9b1702dcfe32c873a770a32cfd306dd7fc1c4fd134adfb783db68defc8894b3c
Status: Downloaded newer image for ubuntu:latest
root@d8324bc29145:/# ls
bin dev home media opt root sbin sys usr
boot etc lib mnt proc run srv tmp var
root@d8324bc29145:/# more /etc/os-release
NAME="Ubuntu"
VERSION="18.04.2 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.2 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
root@d8324bc29145:/# lscpu
Architecture: armv7l
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 1
Vendor ID: ARM
Model: 3
Model name: Cortex-A72
Stepping: r0p3
CPU max MHz: 1500.0000
CPU min MHz: 600.0000
BogoMIPS: 108.00
Flag half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
6:09 / 20:48


docker.mp4
(70) Intro to Docker usin
File | /home/devel/gary_explains_docker/docker.mp4
Dockerfile
Dockerfile defines what goes on in the environment inside your container and allows you to build your own app into your container.
12:40 / 20:48


docker.mp4
(70) Intro to Docker usin
File | /home/devel/gary_explains_docker/docker.mp4
FROM gcc:4.9
COPY . /myapp
WORKDIR /myapp
RUN gcc -o hello hellow.c
CMD ["./hello"]
13:09 / 20:48


docker.mp4
(70) Intro to Docker usin
File | /home/devel/gary_explains_docker/docker.mp4
hellow.c
#include <stdio.h>
int main() {
printf("Hello from within a Docker Container\n");
return 0;
}
14:05 / 20:48


docker.mp4
(70) Intro to Docker usin
File | /home/devel/gary_explains_docker/docker.mp4
$ docker build -t hello-world-gcc .
14:31 / 20:48


docker.mp4
(70) Intro to Docker usin
File | /home/devel/gary_explains_docker/docker.mp4
FROM node
COPY . /myapp
WORKDIR /myapp
EXPOSE 80
CMD ["node","index.js"]
15:53 / 20:48