CH4 Basic Software & Tool

Software and Tool

- Open Source License
- Develop Tool
  - → Geany, gedit, vim
  - → Git
  - → diff, patch
- Build Code Tool
  - → ARM Toolchain
  - → make
  - → automake, autoconfig

Software and Tool

- Network
  - → nmcli, ethtool, wpa_supplicant
  - → SSH, SSHFS
  - → NFS
- Bluetooth
  - → hciconfig, bluetoothctl
- Media Software
  - → gstreamer
  - → ALSA Tool - aplay, arecord

Software and Tool

- Bus
  - → I2C – I2cset, i2cget, i2cdump
  - → USB – lsusb

Package Manage

Debian Package Management

- dpkg : command-line tool for handling packages
- dpkg --help
  - → [CMD] dpkg -i ${PACKAGE_NAME} : install packag
  - → [CMD] dpkg -r ${PACKAGE_NAME} : remove package

Ubuntu Package Management

- apt-get : command-line tool for handling packages
- apt-get --help
  - → [CMD] apt-get update
  - → [CMD] apt-get install ${PACKAGE_NAME}
  - → [CMD] apt-get remove ${PACKAGE_NAME}
  - → [CMD] apt-get autoremove
  - → [CMD] apt-get clean

Editor

Editor

- Geany
  - [CMD] sudo apt-get install geany
- Vim
  - [CMD] sudo apt-get install vim
- Gedit
  - [CMD] sudo apt-get install gedit

Diff Tool

diff and patch

- diff - compare files line by line
  - Create a patch file
    - diff -Nuar file_a file_b > c.patch
      - -N, treat absent files as empty
      - -a, --text
      - -u, output NUM (default 3) lines of unified context
      - -r, recursively compare any subdirectories found
- patch - apply a diff file to an original
  - apply a patch file
    - patch ./hello_1.c < ./tmp.patch
  - Reverse a patch file
    - patch -R ./hello_1.c < tmp.patch

Source Code Version Control

Git

- https://git-scm.com/book/zh-tw/v1/
- 版本控制
- 程式回溯
- 管理多人共同開發

Makefile

Makefile

- Simplify compile command
- Automation compile, linker program source
- It can update source in accordance with the dependence

Compile a Hello_World

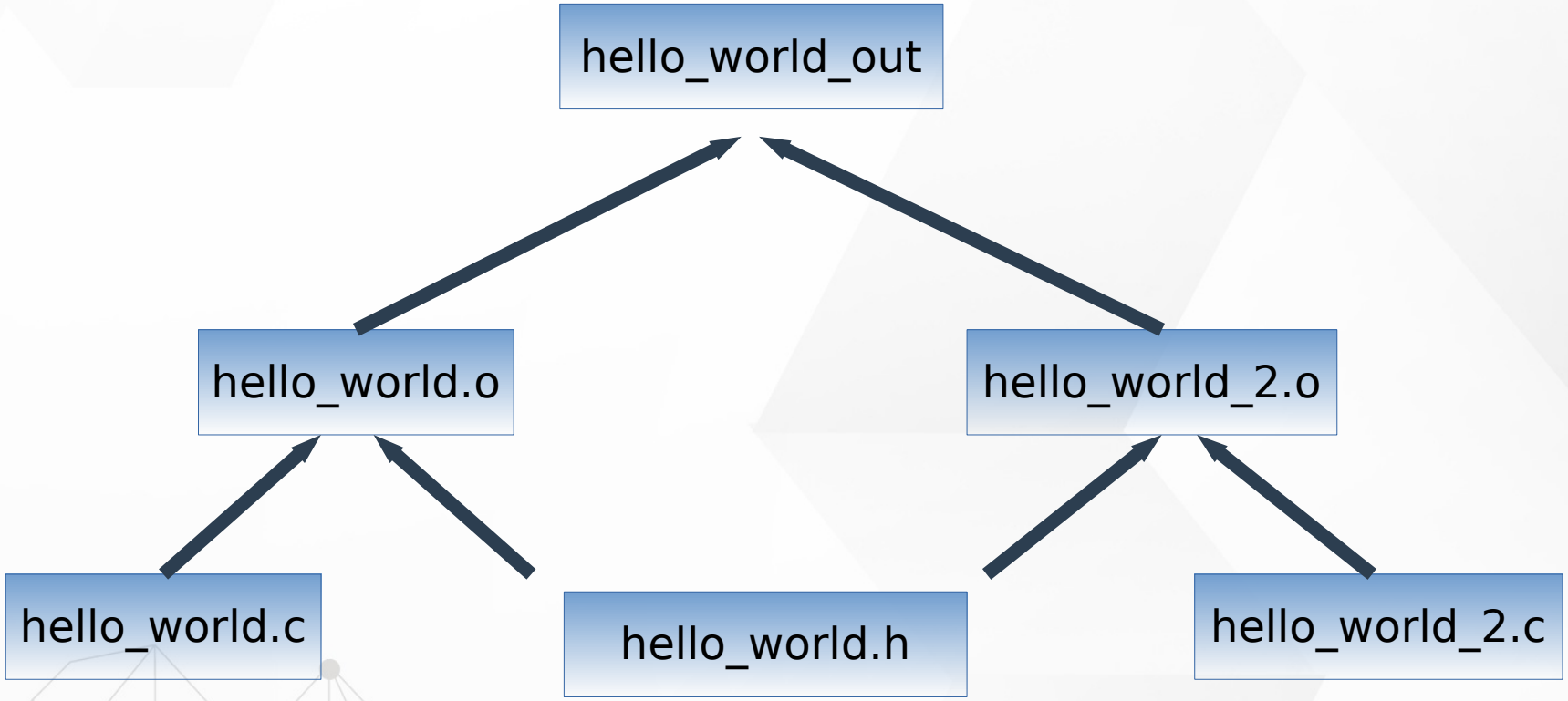
A	B	C
aarch64-linux-gnu-gcc	-o helloworld	./hello.c

A : ARM C Compile

B : ARM C Compile Parameter
(Output name)

C : C source code

Another Sample



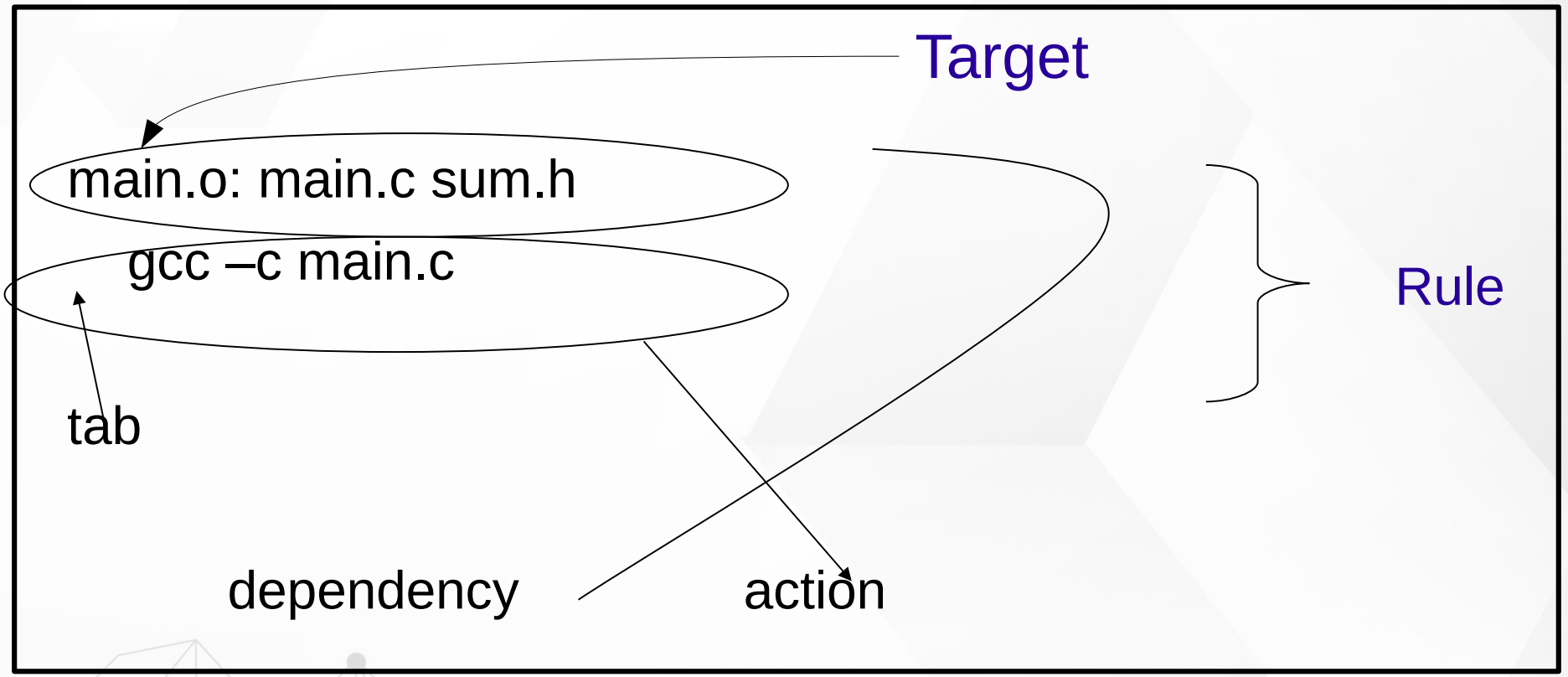
Compile Another Sample

- Step 1 : gcc -c hello_world.c
- Step 2 : gcc -c hello_world_2.c
- Step 3 : gcc -o hello_world hello_world.o hello_world_2.o

Another Sample - Makefile

```
CC=$(CROSS_COMPILE)gcc


all: hello_world

hello_world: hello_world.o hello_world_2.o
	$(CC) -o hello_world hello_world.o hello_world_2.o

hello_world.o: hello_world.c
	$(CC)  -c hello_world.c

hello_world_2.o: hello_world_2.c
	$(CC)  -c hello_world_2.c

clean:
	rm -r *.o
	rm hello_world
```

Rule Syntax



make 在編譯時，若發現 target 比較新，
也就是 dependencies 都比 target 舊，
那麼將不會重新建立 target，如此可以避免不必要的編譯動作

Rule Syntax

```
CC=$(CROSS_COMPILE)gcc

CFLAGS=-Wl,-Map,out.map -lpthread -lm

all: hello_world

hello_world: hello_world.o hello_world_2.o
	$(CC) -o hello_world hello_world.o hello_world_2.o

hello_world.o: hello_world.c
	$(CC) $(CFLAGS) -c hello_world.c

hello_world_2.o: hello_world_2.c
	$(CC) $(CFLAGS) -c hello_world_2.c

clean:
	rm -r *.o
	rm hello_world
```

hello_world_ex1

```
CC=$(CROSS_COMPILE)gcc

AA ='1234' '5678'
AA := 'DDDD'

$(info AA=$(AA))

CFLAGS=-Wl,-Map,out.map -lpthread -lm

all: hello_world

hello_world: hello_world.o
	$(CC) -o hello_world hello_world.o

hello_world.o: hello_world.c
	$(CC) $(CFLAGS) -c hello_world.c

clean:
	rm -r *.o
	rm hello_world
```

: Tab

Assignment Operators

=	定義一個 需做遞迴展開的 變數型態
:=	定義一個 立即運作的 變數型態
+=	將 指定值， 續加在 原變數中
?=	如果之前 無任何設定該變數， 即現在設定， 否則 跳過設定 (就是不做任何事)

Assignment Operators Sample 1

```
AA ='1234' '5678'
BB = ${AA}
AA = '789'
AA += 'ABCDE'
```

Output

```
AA='789' 'ABCDE'
BB='789' 'ABCDE'
```

Assignment Operators Sample 2

```
AA ='1234' '5678'
BB := ${AA}
AA = '789'
AA += 'ABCDE'
```

Output

```
AA='789' 'ABCDE'
BB='1234' '5678'
```

Assignment Operators Sample 3

```
AA ='1234' '5678'
BB := ${AA}
AA = '789'
AA ?= 'ABCDE'
```

Output

```
AA='789'
BB='1234' '5678'
```

Command-Line Options

- -C dir, --directory= dir
  - make changes the current working directory to dir before it does anything else. If the command line includes multiple -C options, each directory specified builds on the previous one

- -j [number] , --jobs[= number]
  - Run multiple commands in parallel

Bus Tool

Device Tool

- Real Time Clock
  - hwclock
    - -s, --hctosys set the system time from the RTC
    - -r, --show display the RTC time
- USB
  - lsusb
- Block Device
  - lsblk
- I2c-tools
  - I2cdump, i2cdetect, i2cget, i2cset

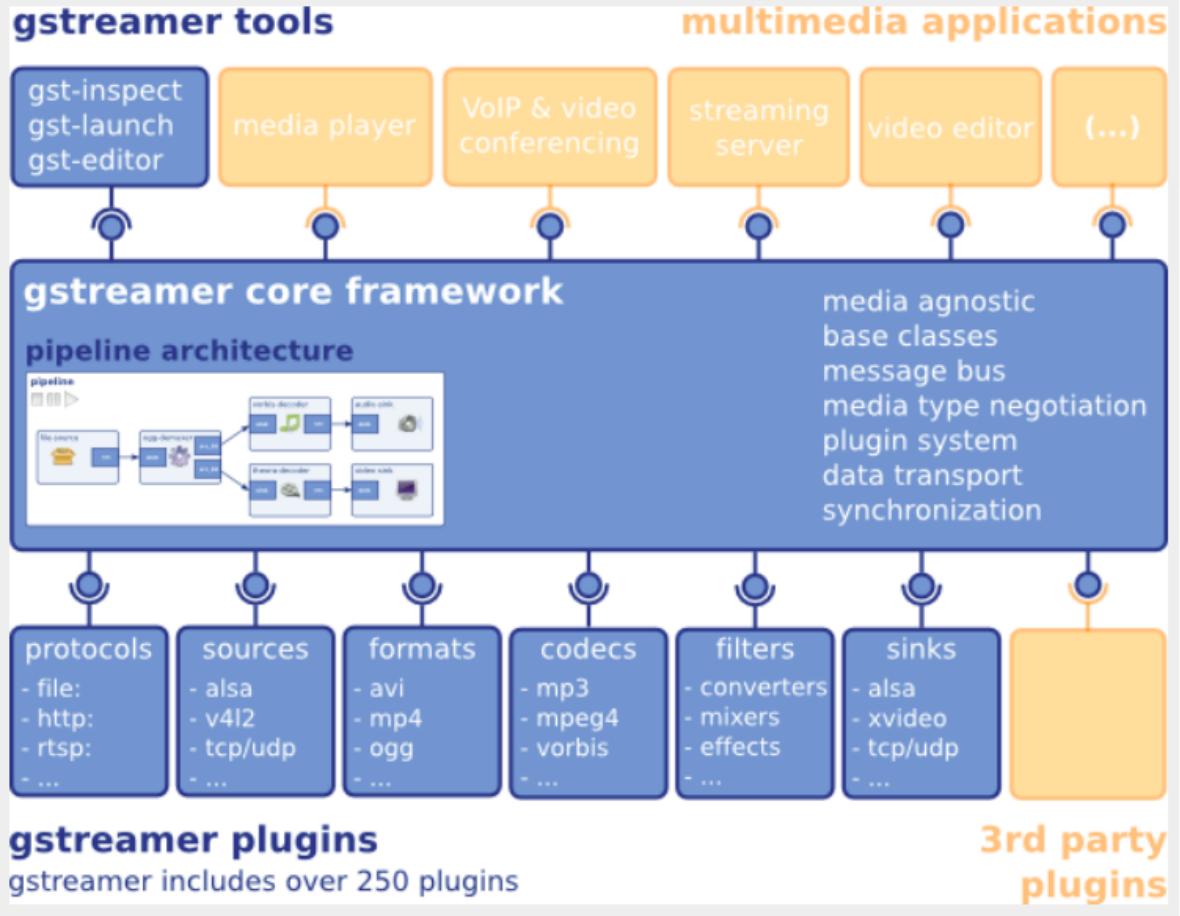
Media Tool

Gstreamer

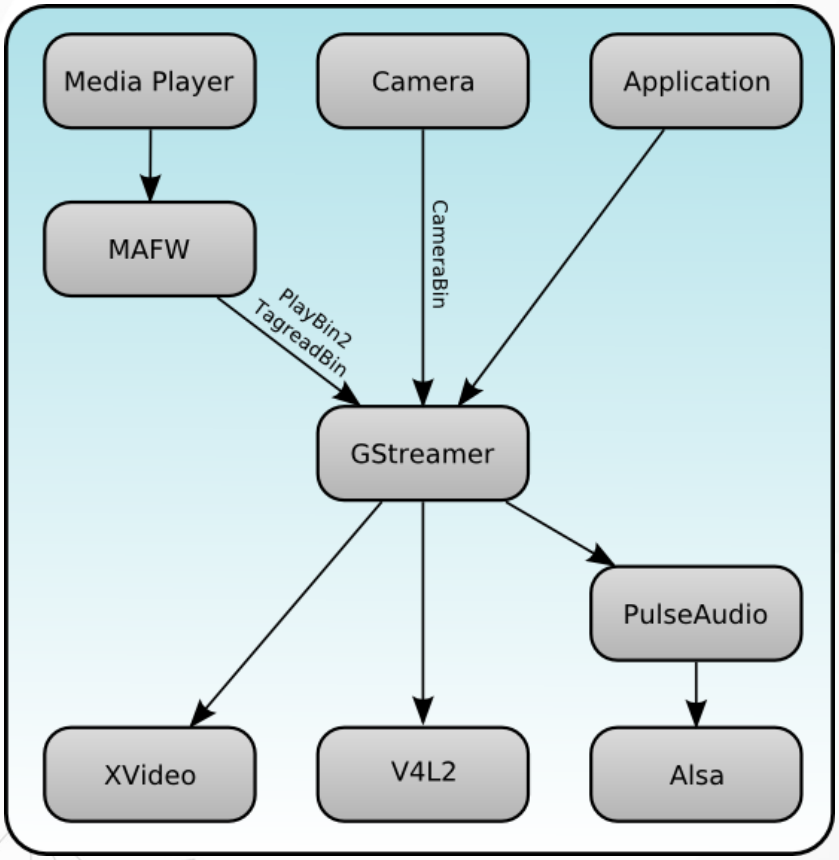
Open Source Multimedia Framework

- https://gstreamer.freedesktop.org/
- Documentation
  - Tutorials

Block Diagram



Block Diagram



http://maemo.org/development/sdks/maemo_5_beta_docs/using_multimedia_components/

Overview

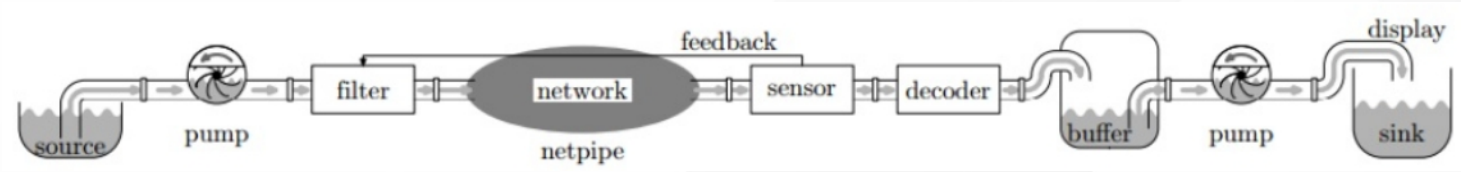
- GStreamer is a **framework** for creating streaming media applications
- The framework is based on **plugins** that will provide the various codec and other functionality

Overview

- **Gst-plugins-base:**
  an essential exemplary set of elements
- **Gst-plugins-good:**
  a set of good-quality plug-ins under LGPL
- **Gst-plugins-ugly:**
  a set of good-quality plug-ins that might pose distribution problems
- **Gst-plugins-bad:**
  a set of plug-ins that need more quality

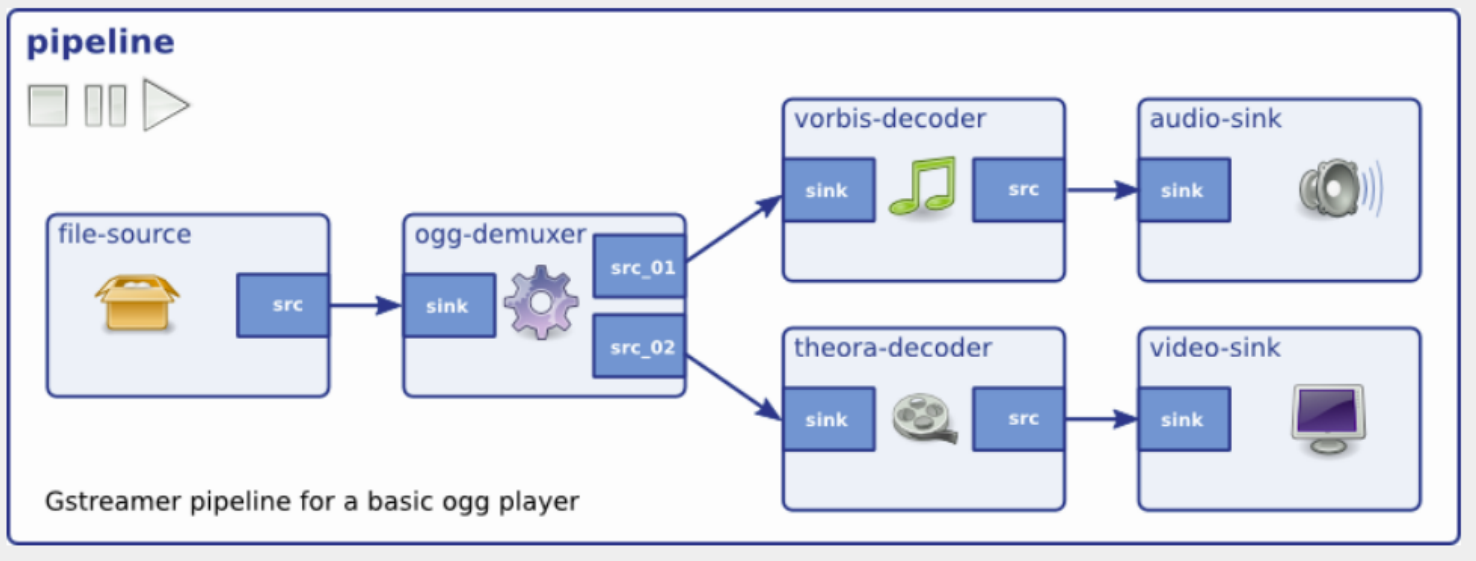
Gstreamer Pipe

A streamer pipe



https://www.slideshare.net/rodrigo_costa/gstreamer-an-overview

Gstreamer Pipe



Gstreamer - Tool

- Gst-inspect-1.0
  - Print supported plug-in
- Gst-launch-1.0
  - Gstreamer tester
- Gst-typefind-1.0
  - Check file for gstreamer plug-in type

Gstreamer - Tool

- Gst-inspect-1.0
  - Check what kind of videosink in Rockpi4b
  - gst-inspect-1.0 | grep -i videosink

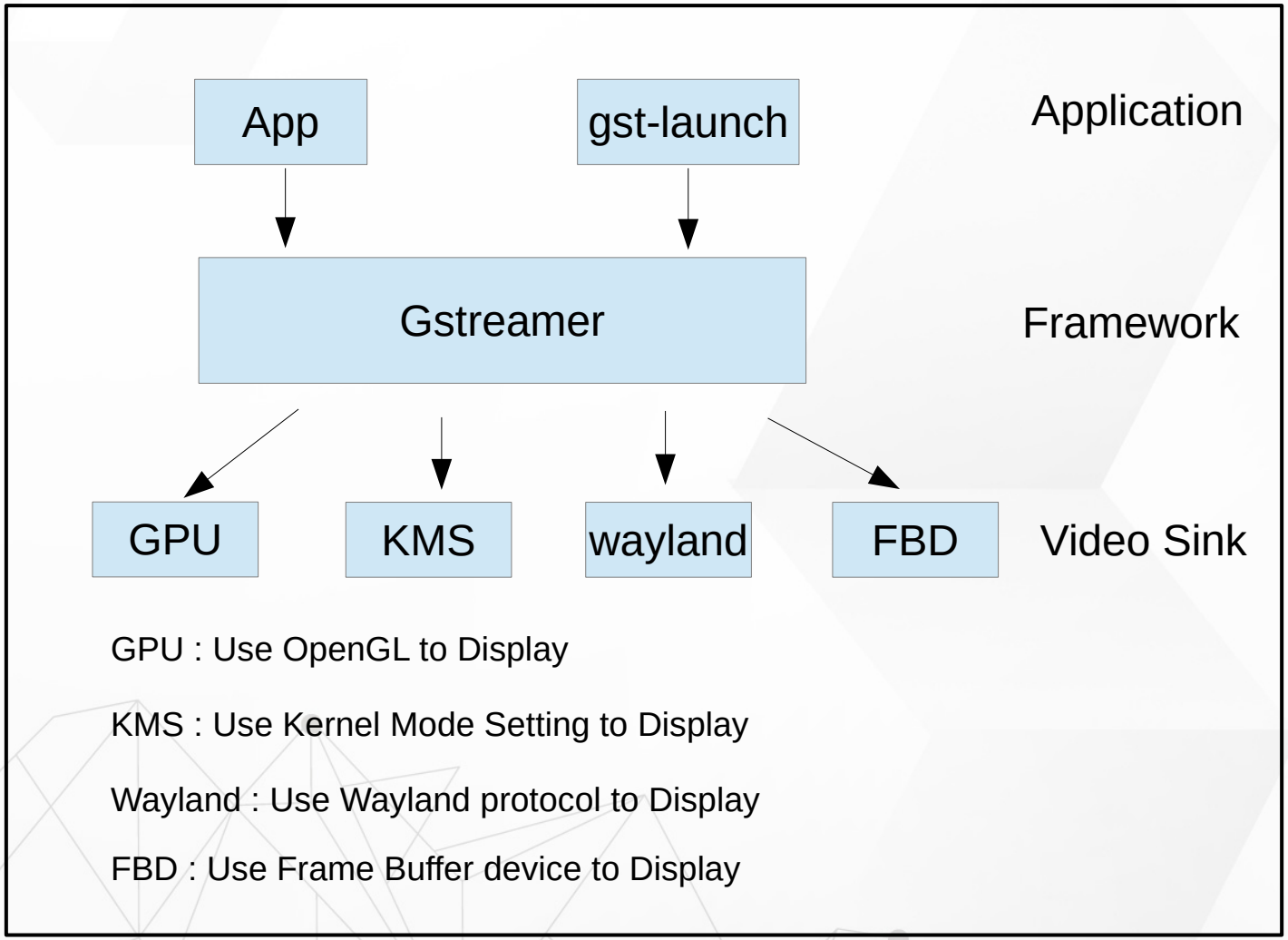
```
rock@rockpi4b:~$ gst-inspect-1.0 | grep -i videosink
debugutilsbad:  fakevideosink: Fake Video Sink
debugutilsbad:  fpsdisplaysink: Measure and show framerate on videosink
inter:  intervideosink: Internal video sink
decklink:  decklinkvideosink: Decklink Video Sink
autodetect:  autovideosink: Auto video sink
rock@rockpi4b:~$
```

Gstreamer - Tool

- Gst-typefind-1.0
  - Check Serenity-DVD-320x240.m4v what kind of file type in gstreamer
  - gst-typefind-1.0 ./Serenity-DVD-320x240.m4v

```
rock@rockpi4b:~$ gst-typefind-1.0 ./Serenity-DVD-320x240.m4v
./Serenity-DVD-320x240.m4v - video/quicktime, variant=(string)iso
```

Gstereamer - Video

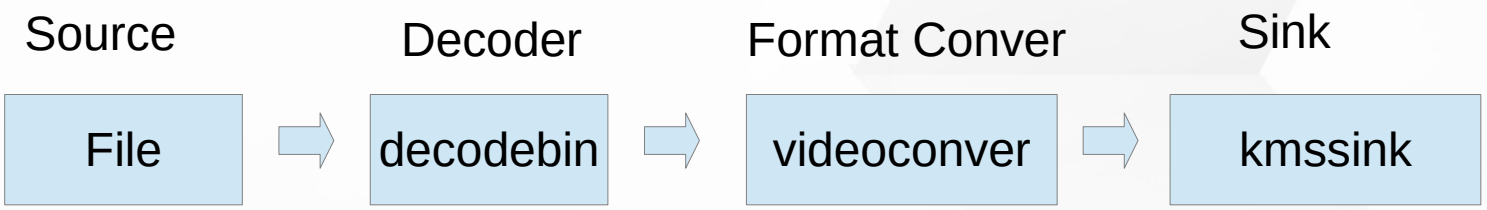


Play Video Test Stream


```
gst-launch-1.0 videotestsrc ! video/x-raw, width=1280, height=720 ! kmssink
```

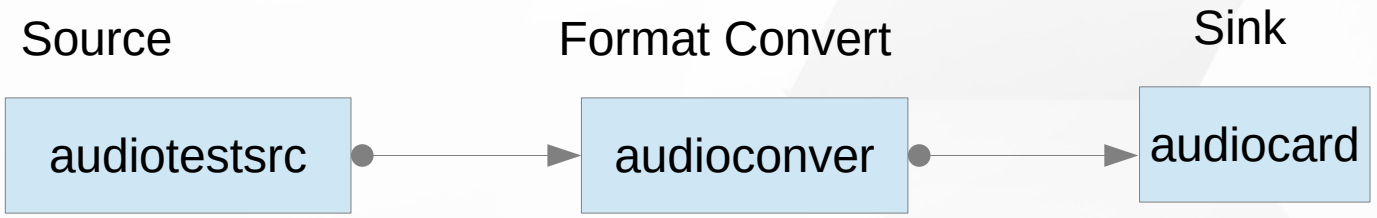
Play a H.264 video

```
gst-launch-1.0 filesrc location=/tmp/Serenity-DVD-320x240.m4v ! \
decodebin name=dec ! \
videoconvert ! \
kmssink
```



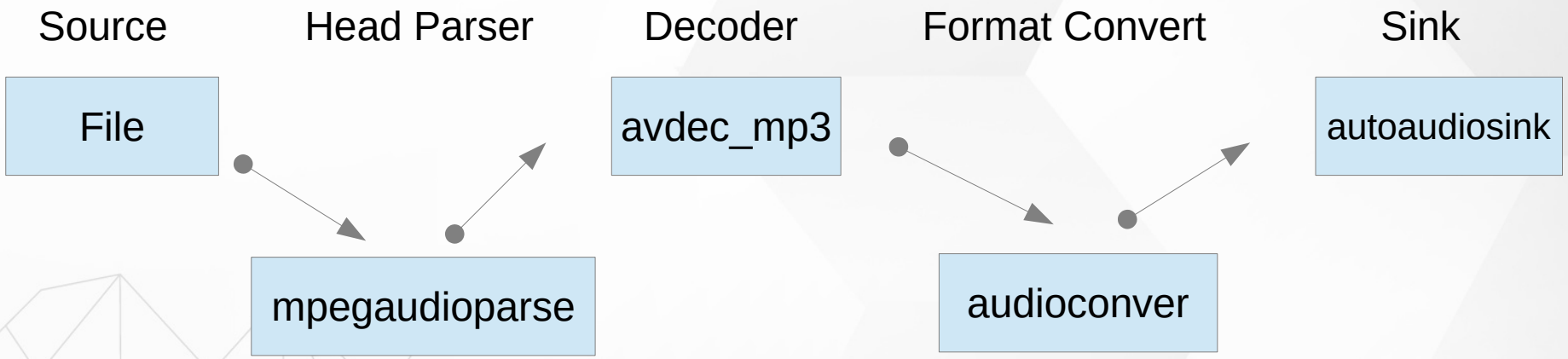
Play a Audio Test

```
gst-launch-1.0 audiotestsrc ! audioconvert ! alsasink device-name=rockchipes8316c
```



Play a MP3

```
gst-launch-1.0 filesrc location="/home/cadtc/audio/piano2-CoolEdit.mp3" ! \
mpegaudioparse ! \
mpg123audiodec ! \
audioconvert ! autoaudiosink
```



ALSA Tool

- ALSA Utile
  - **aplay**
    - Play a WAV file
  - **arecord**
    - Record a sound
  - **alsamixer**
    - A graph tool for adjusting audio gain
  - **amixer**
    - A console tool for adjusting audio gain

ALSA Tool

ALSA Utile

- **aplay**
  - aplay -Dsysdefault:CARD=rockchipes8316c /usr/share/sounds/alsa/Front_Center.wav
  - aplay -Dsysdefault:CARD=HDMICODEC /usr/share/sounds/alsa/Front_Center.wav
- **arecord**
  - arecord -Dhw:0,0 -r 44100 -t wav -f CD -d 5 /tmp/test.wav
- **alsamixer**
  - alasmixer
- **amixer**
  - amixer scontrols | less
  - amixer sget 'HP' 0%
  - amixer sset 'HP' 0%

WiFi and Network

Network Tool

- ifconfig → Network setting
- ping → Network package test
- iperf3 → perform network throughput tests
- dhcpc → used for automatic retrieving of

WIFI

- NetworkManager Command Line Tool
  - nmcli
- Show / manipulate wireless devices and their configuration
  - iw
- For connecting to a WPA/WPA2 network
  - wpa_supplicant

SSH

SSH

- Secure SHell protocol
- SSH Client
- SSH Server

SSH

- SSH Client
  - # sudo apt-get install ssh
  - References

NFS

NFS

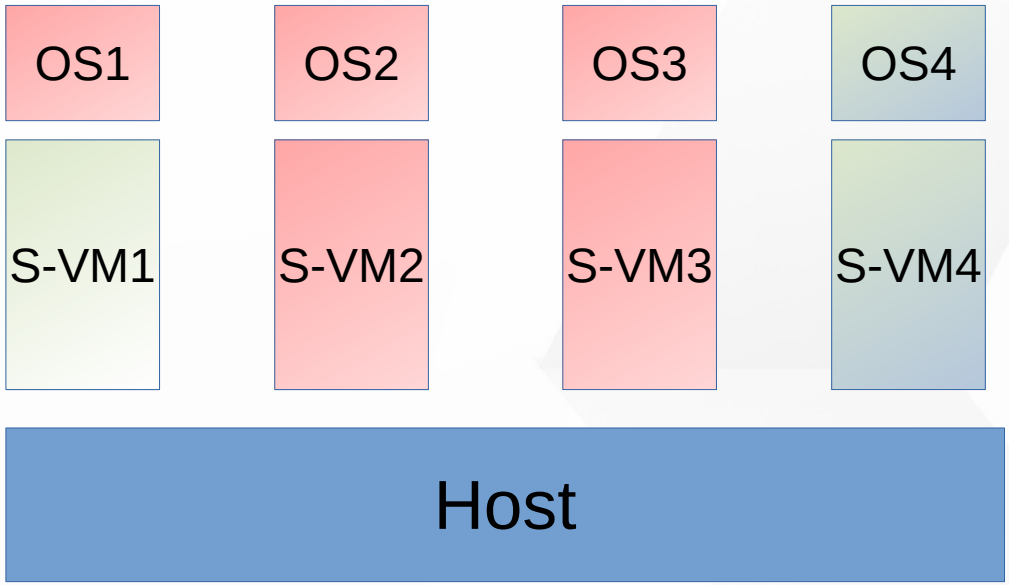
- Network File System
- NFS Client
- NFS Server

Docker

What is Docker

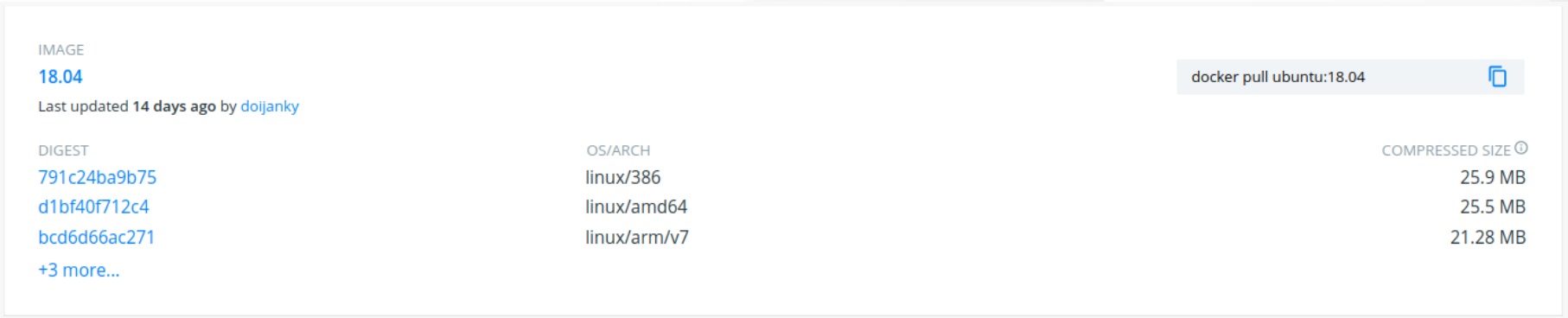
- https://docs.microsoft.com/zh-tw/dotnet/architecture/containerized-lifecycle/what-is-docker
- 簡單的說
  - Docker ： 模擬 作業系統的 “ User space”, 沒有底層
  - VirtualBox ： 模擬 整個 作業系統 , 包含 底層 與 硬體
- 優點
  - 對於開發 User space 的程式方便交換環境 , 快速
- 缺點
  - 無法處理 底層 或是 硬體 相關 部份

What is Docker



Dicker Hub

- https://hub.docker.com/
- https://hub.docker.com/_/ubuntu
- Ubuntu 官方 已在 Docker Hub 建立不同版本的 docker image
- 直接從 Docker Hub 下載需要的新環境
  - 例如 ：Ubuntu 1804



Install a Docker in Ubuntu

- Install docker in Ubuntu
  - $ sudo apt-get install docker
- Docker help
  - $ docker --help

How to use Docker

- Pull a Docker image that in Docker Hub
  - For example : ubuntu 18.04
  - $ docker pull ubuntu:18.04

```
slash@slash-HD631-Q87CRM:~$ dicker images
No command 'dicker' found, did you mean:
 Command 'ticker' from package 'ticker' (universe)
 Command 'docker' from package 'docker.io' (universe)
dicker: command not found
slash@slash-HD631-Q87CRM:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
slash@slash-HD631-Q87CRM:~$ docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
f08d8e2a3ba1: Extracting [>                                                  ]  294.9kB/26.7MB
3baa9cb2483b: Download complete
94e5ff4c0b15: Download complete
1860925334f9: Download complete
```

How to use Docker-2

- Check docker Image in local
  - $ docker images

```
slash@slash-HD631-Q87CRM:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              18.04               6526a1858e5d        2 weeks ago         64.2MB
slash@slash-HD631-Q87CRM:~$
```

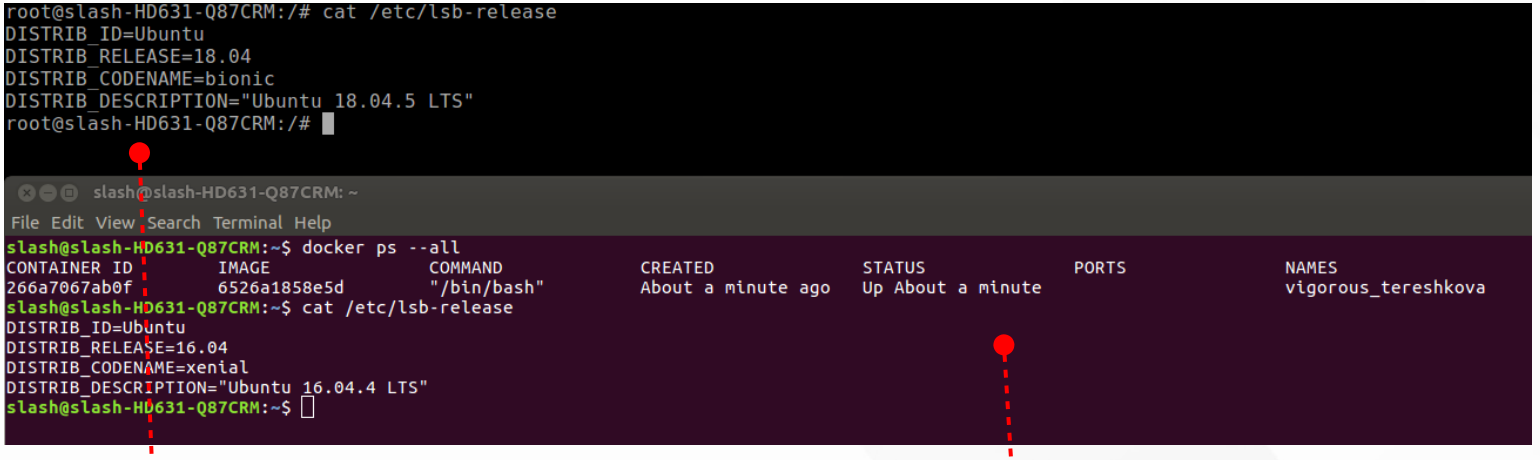
- start to run docker Image in local
  - $ docker run --help
  - $ docker run --net=host -i -t 6526a1858e5d /bin/bash

Enter docker content

```
slash@slash-HD631-Q87CRM:~$ docker run -i -t --net=host 6526a1858e5d /bin/bash
root@slash-HD631-Q87CRM:/#
```

How to use Docker-3

- Check docker content with docker command



Docker content (ubuntu 18.04)

Host (ubuntu 20.04)

How to use Docker-4

- Install package that you want in docker image
  - For example
    - $ apt-get install net-tools

```
root@slash-HD631-Q87CRM:/# apt-get install  net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 5 not upgraded.
Need to get 194 kB of archives.
After this operation, 803 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.60+git20161116.90da8a0-1ubuntu1 [194 kB]
Fetched 194 kB in 2s (115 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package net-tools.
(Reading database ... 4046 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
root@slash-HD631-Q87CRM:/#
```

How to use Docker-5

- **Commit** a docker content and push it to Docker Hub
  - 1. check which docker content that you want to commit
    - $ docker ps --all

```
slash@slash-HD631-Q87CRM:~$ docker ps --all
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
266a7067ab0f        6526a1858e5d        "/bin/bash"         16 hours ago        Up 12 minutes                           vigorous_tereshkova
slash@slash-HD631-Q87CRM:~$
```

  - 2. check REPOSITORY and TAG of docker image current
    -

```
slash@slash-HD631-Q87CRM:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              18.04               6526a1858e5d        2 weeks ago         64.2MB
```

  - 3. commit docker content
    - $ docker commit -m "test" 266a7067ab0f ubuntu:18.04

```
slash@slash-HD631-Q87CRM:~$ docker commit -m "test" 266a7067ab0f ubuntu:18.04
sha256:769ce8b9f38081b4cedf764b51e070ba4032b8146f26cf5c5d6726a3f20b831a
slash@slash-HD631-Q87CRM:~$ rm -rf ./build-fb/^C
slash@slash-HD631-Q87CRM:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              18.04               769ce8b9f380        11 seconds ago      94.9MB
ubuntu              <none>              6526a1858e5d        2 weeks ago         64.2MB
```

How to use Docker-6

- **Import** a Local Docker Image (Load)
  - [CMD] docker load < docker_image_file.tar
- Save a Local Docker Image (Export)
  - [CMD] docker save ${IMAGE_ID} > docker_image_file.tar
- Delete a docker **Content**
  - $ docker ps --all
  - $ docker rm ${Content ID}
- Delete a docker **Image**
  - $ docker images
  - $ docker rmi ${IMAGE ID}

A Sample Docker Usage

- **docker_info.sh**
- Set environment variables
  - -e --env
  - -e DISPLAY
- Execute application when login to docker content
  - -c --cpu-shares
  - -c "${DOCKER_BUILD_CMD}"
- Work directory
  - -w --workdir
  - -w ${DOCKER_WORKDIR}

A Sample Docker Usage

- Login with user name
  - -u --user
  - -u ${DOCKER_LOGIN_ID}
- Mount a local disk space in docker content
  - -v --volume
  - -v ${LOCAL_FOLDER}:${DOCKER_FOLDER}
- Allocate a pseudo-TTY
  - -t --tty
- Keep STDIN open even if not attached
  - -i --interactive