

Development

Meeting Docker

Outline

- Editors
- Development Environment
- Local, VM, Docker
- First Steps with Docker

Editors

- Need to write files
 -
- No requirement to use product X or Y
 - You should like it
 - Have features you like
 - Code highlighting
 - Useful templates
- I frequently change editors
 - Free
 - Feature
 - Visual Studio Code

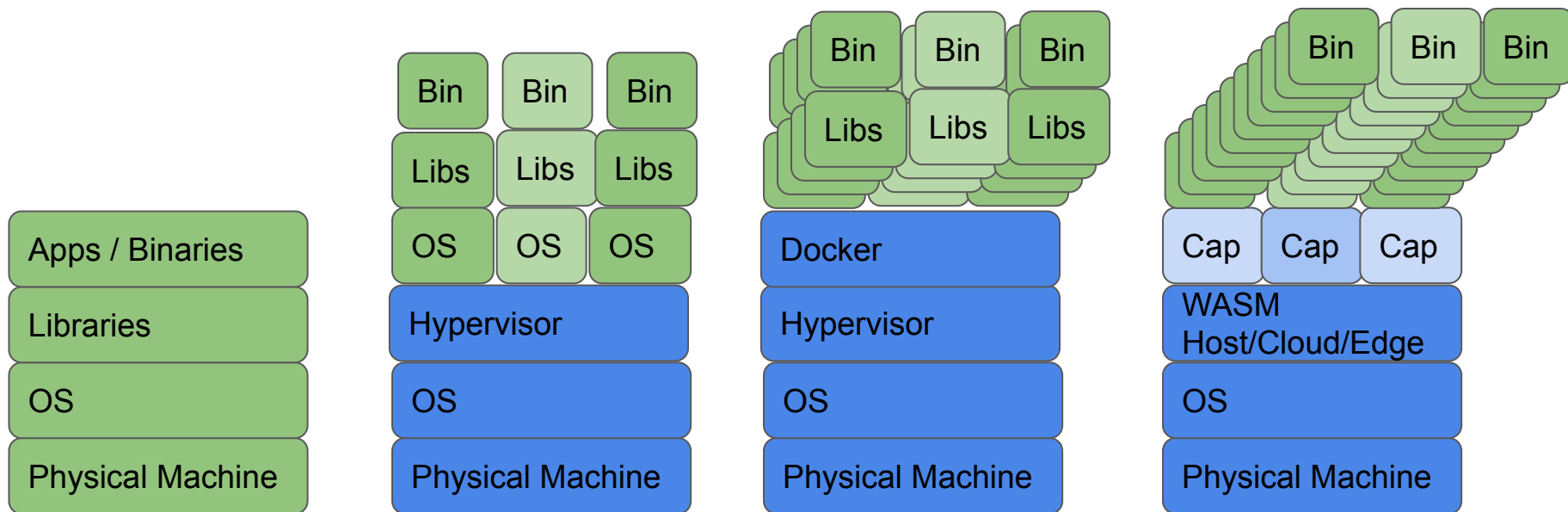
Software Development

- Development Environment, Production Environment
 - DEV == PROD?
- Need for different languages/frameworks
- Need to run networks

Developing

- Local
- VM
- Containers -> **Docker**
- WebAssembly

Development



Computer:
PC, Server, ...

Virtual machine

Container

WebAssembly

Local

- Create local environment
 - Install servers/libraries
 -
- Advantages:
 - You are in control
 - Customize everything
 - Low overhead
 -
- BUT

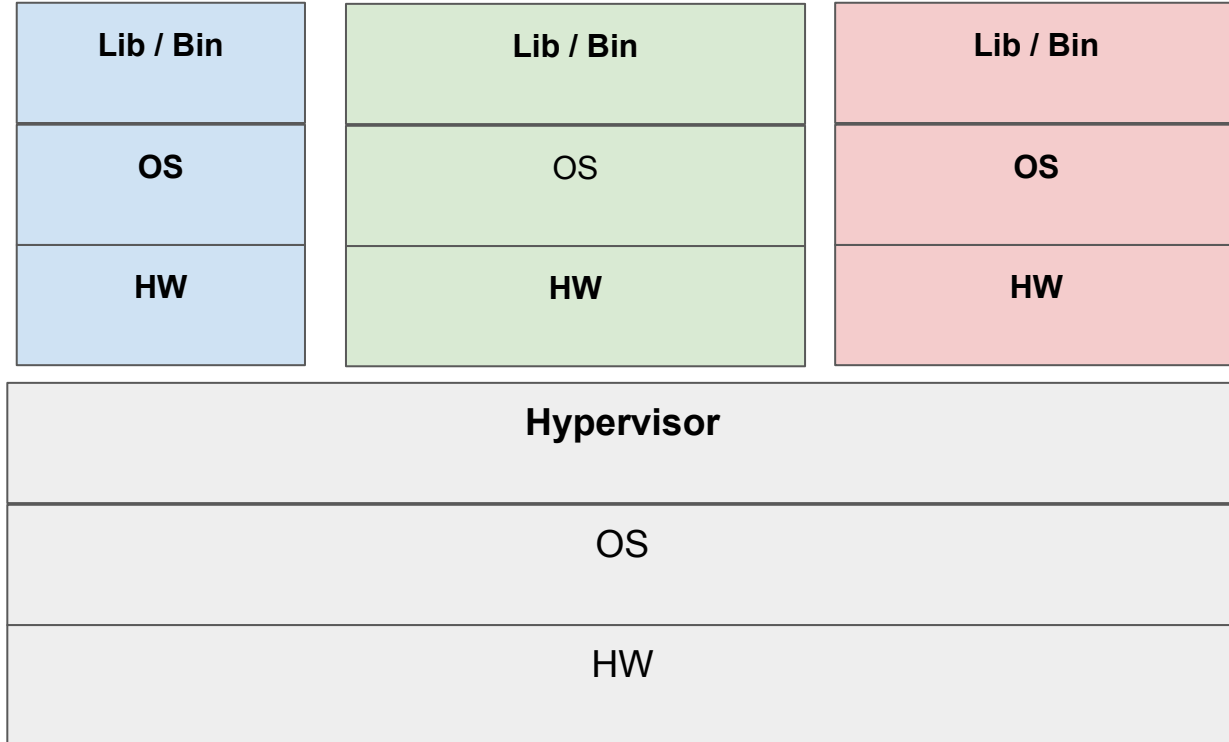
Disadvantages

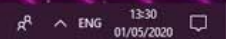
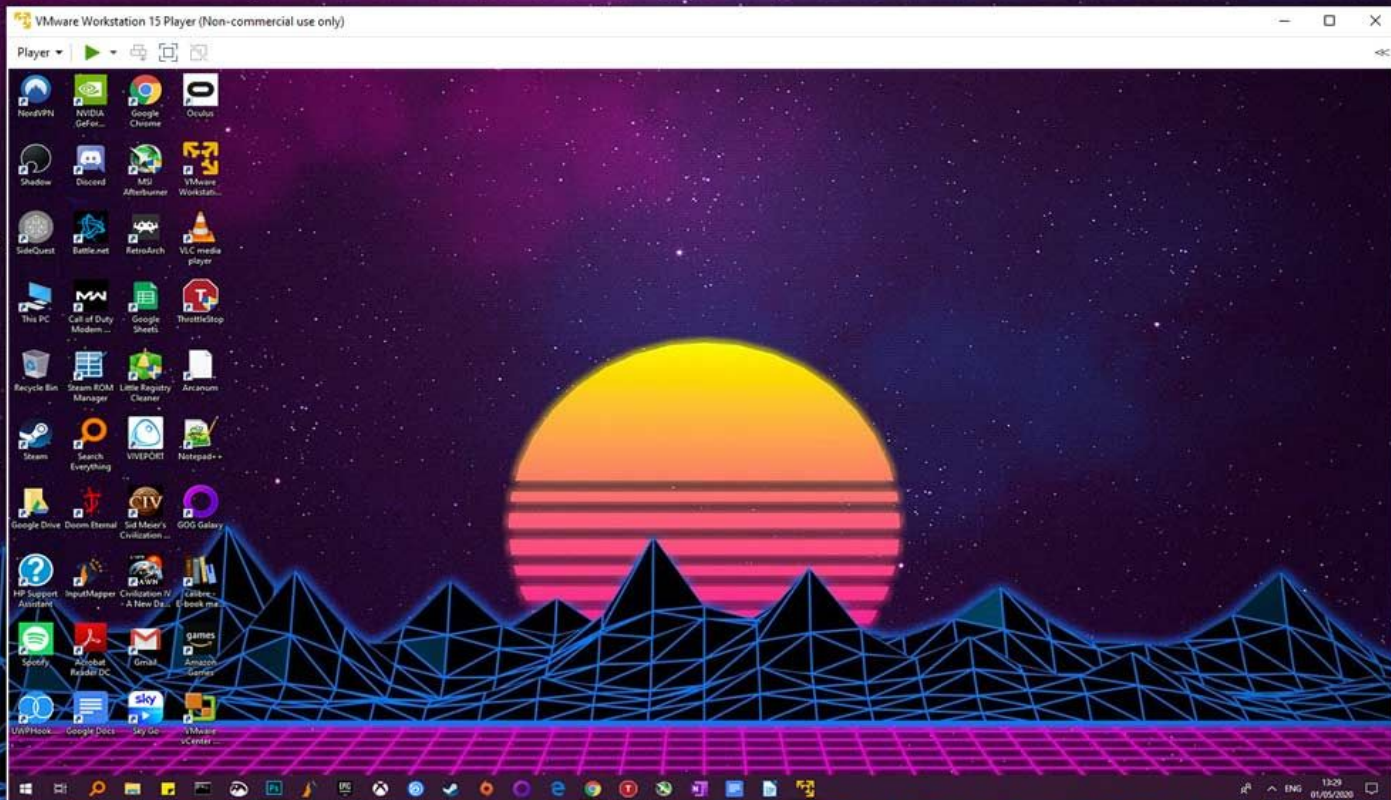
- Installations
- Library versions
 - Different projects may need different versions/subversions
- Directory separator
 - / A / B
 - \ A \ B
- Case Sensitivity
 - Does upper/lower case matter?
- File encoding
- Permissions ...

Yes your IDE can help and yes code repository can help

Virtual Machines

- Virtualize computers
 - Simulate machines





```
Preview File Edit View Go Tools Bookmarks Window Help
Terminal — telnet — 80x24

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.
Hercules version 3.06 built on Jan 30 2009 21:35:31
running on gojira.local (Darwin-9.6.0.Darwin Kernel Version 9.6.0: Mon Nov 24 17:37:00 1386 MP+2)
Connected to device 0:001F

VM/370 VERSION 06 LEVEL 00 PLC 0029; 09/07/05 09:42:40

NOW 08:06:54 PST MONDAY 02/09/09
CHANGE TOD CLOCK (YES NO) :HMCTE006A Enter input for console device 001F
NO
08:06:58 START ((COLD WARM CKPT FORCE) (DRAIN)) (SHUTDOWN) :HMCTE006A Enter input for console device 001F
COLD
08:07:00 AUTO LOGON *** OPERATOR USERS = 001 BY SYSTEM
DMKCP19571 STOR 0204SK, MUC 180K, DVM 01712K, TRR 032K, FREE 0124K, V=R 00000K
08:07:00 FILES: NO ROR, NO PAT, NO PUN
enable all
08:07:04 08:07:04 COMMAND COMPLETE
```

```
x3270-2 127.0.0.1:3270

File Options

VM/370 ONLINE

VV VV MM MM
VV VV MM MM
VV VV MM MM
VV VV MM MM
3333333333 7777777777777777 00000000
333333333333 7777777777777777 0000000000
33 VV33 77VV 77 00MM 00
VV33 VV 77MM 00MM 00
33 VV 77MM 00MM 00
3333VV VV 77 MM 00MM 00
3333 VVVV 77 MM 00MM 00
33 VV 77 MM 00MM 00
33 33 77 00 00
33 33 77 00 00
333333333333 77 0000000000
333333333333 77 00000000

RUNNING
```

Terminal — hercules — 80x57

Hercules	CPU0000:	0%	S/370	Peripherals	
030E0000	00000000	24..H.....	U	Addr Modl Type Assignment	
PSW			R	000C 3505 ROR * intrq	
			B	000D 3525 PCH punch00d.txt oscil	
			C	000E 1403 PAT print00d.txt crlf	
030E0000	00000310	7FFFFFF8 F00C1000	D	001F 3215 CON 127.0.0.1	
	0	1	2	3	
00000000	00000005	00000000 00000160	E	0040 3270 DSP 127.0.0.1	
	4	5	6	7	
FFFFFFF	0000022F	00010000 00026400	G	0500 3420 TAPE dmhddr.ows [1:00000000]	
	8	9	10	11	
00010E00	00019E00	5001AF10 00000000	H	0501 3420 TAPE VMREL6.ddr.ows [1:000000]	
	12	13	14	15	
GPR	CR	MR	FFR	I	0502 3420 TAPE CPAL6.ddr.ows [1:000000]
ADDRESS: 00000000 DATA: 00000000				J	0131 3330 DASD VMREL6.3330-1 [404 cyls]
				K	0132 3330 DASD CPAL6.3330-1 [404 cyls]

0.00 0 STO DIS RST

MIPS \$10/s

STA STP EXT IPL PWR

CPU
0

Virtual Machines

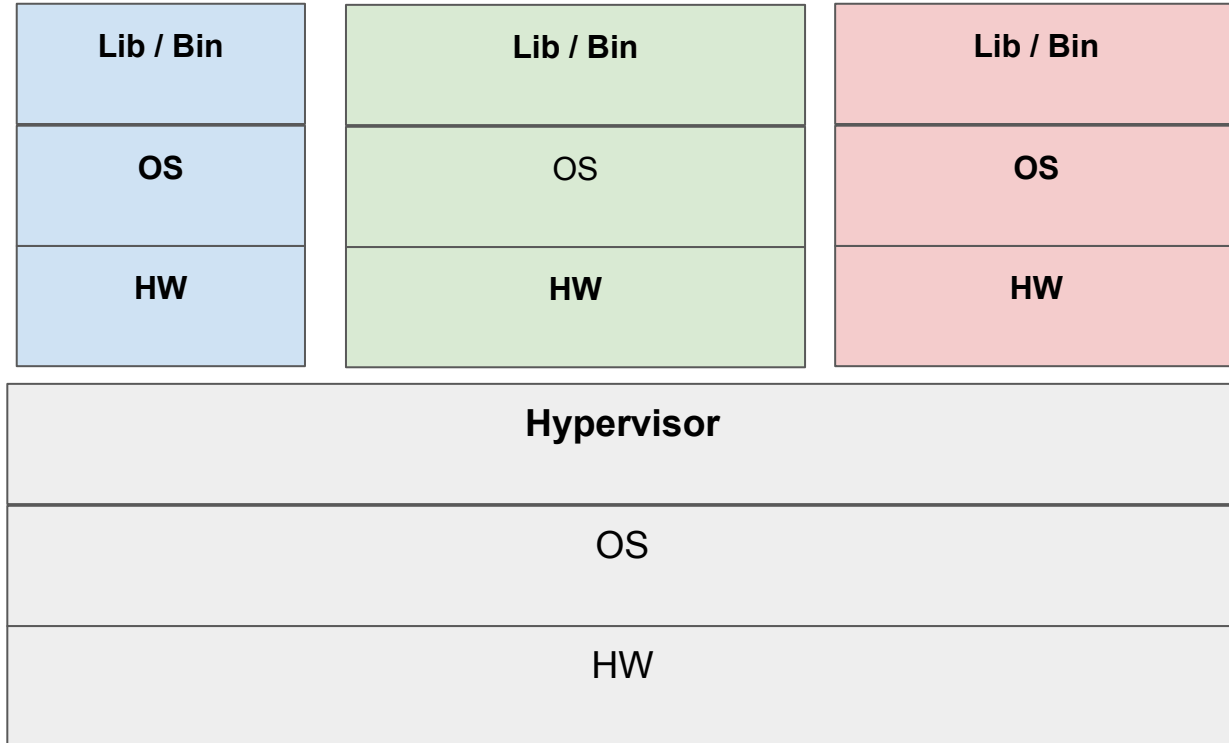
- Create virtual machines that match production environment !
- Advantages:
 - Total isolation
 - Replication of production environment
 - Widely used
 -
- BUT
 - Resource intensive => expensive
 - Why???

Containers !!

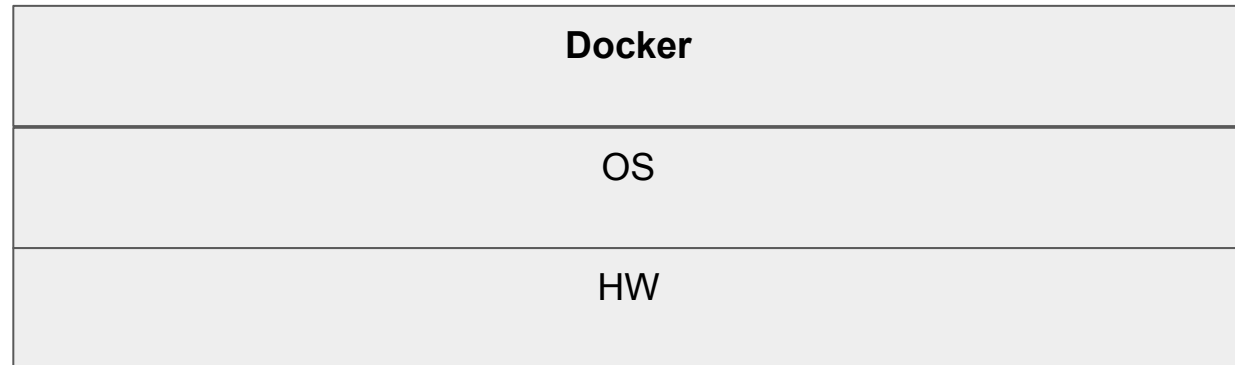
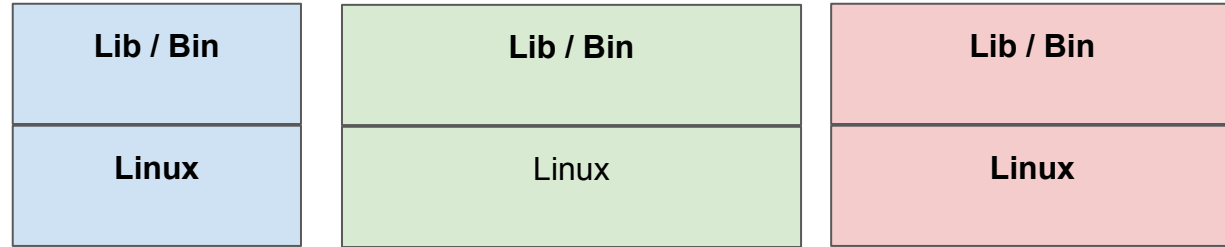
- Define compute environments => containers
- “containerize applications”
- Many Container Solutions
- Docker is most popular
 - Cloud Vendors have own solutions
 - Docker compatible
- 2 basic options
 - Docker Playground:
 - <https://labs.play-with-docker.com/>
 - Need to create a free account
 - Local Install
 - <https://www.docker.com/>



VM

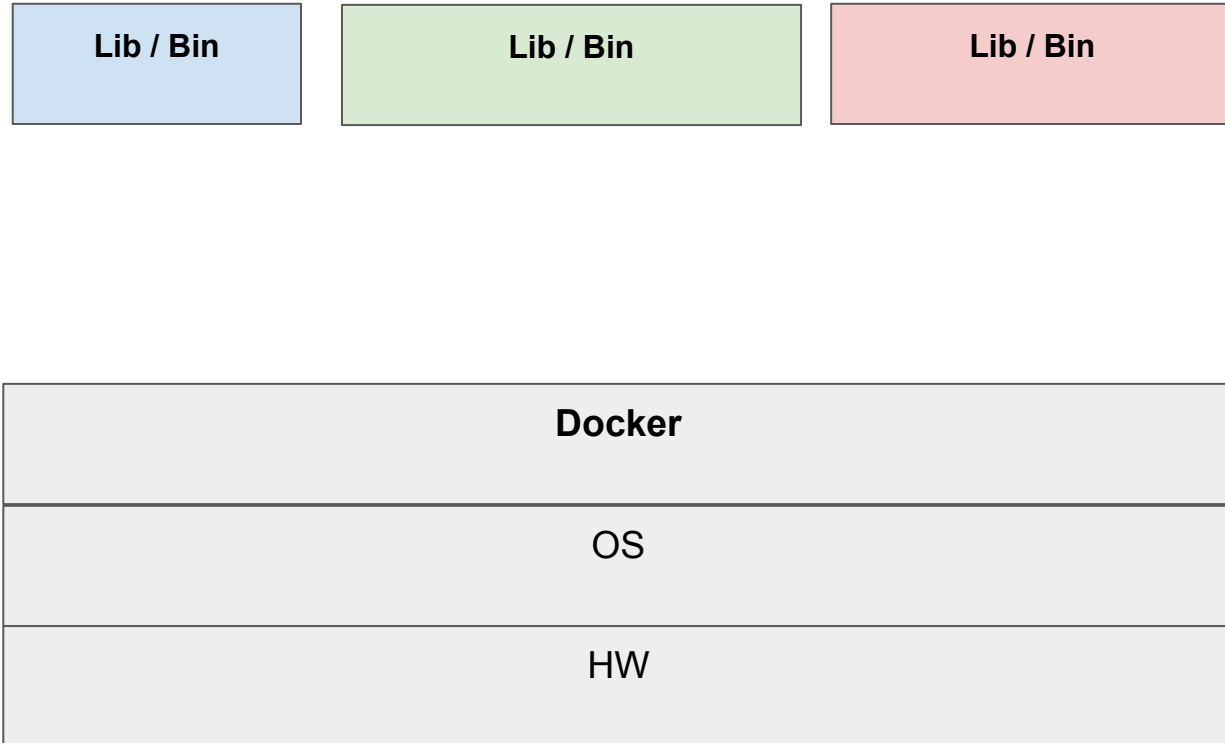


Container



© Ralph Deters - 2023

Container



Docker

- Widely used approach
- Active community
- Key concepts
 - Dockerfile
 - Image
 - Container
- How to get docker?
 - <https://www.docker.com/products/docker-desktop>
- What kind of “images” are there?
 - <https://hub.docker.com/search?q=&type=image>

Steps

- 1) Get base-image with **pull** command
 - a) Something that is close to what you want
- 2) Write a Dockerfile (script) to modify/enrich base image
- 3) Create new image with **build** command
- 4) Run new image with **run** command

Let's install a Node.js server with docker

Let's start

- Get image
 - `docker pull node:latest`
- Write your Dockerfile

Dockerfile

FROM node:latest

EXPOSE 8080

CMD ["/bin/bash"]

Why Expose???

- Docker containers are locked down
- You have to actively enable capabilities (e.g. allow use of port 8080)



...

- Build our own image
 - `docker build -t ralph/node .`

- Run Container (run instance of image)
 - `docker run -it ralph/node`

You also have to grant capabilities to the runtime

- 1) Ports**
- 2) File System**

```
docker run  
-p 80:8080  
-v /Users/ralph/node/nodejs:/usr/src/app/  
--rm  
--name a1  
-it ralph/node
```


Change run command

- Open port
 - `-p 80:8080`
- Mount directory/volume
 - `-v /Users/ralph/node/nodejs:/usr/src/app/`
- Remove container after it stopped
 - `--rm`
- Give container a name
 - `--name a1`
-

Starting Container

1. `docker run -p 80:8080 -it ralph/node`
2. `docker run -p 80:8080 --rm -it ralph/node`
3. `docker run -p 80:8080 --rm --name a1 -it ralph/node`
4. `docker run -p 80:8080 -v
/Users/ralph/node/nodejs:/usr/src/app/ --rm --name a1
-it ralph/node`

Configuring Node.js

```
cd /usr/src/app/
```

Create config file of the server

```
npm init
```

Add library (package)

```
npm add express
```

Create Code for Server

First install editor

`apt-get update`

`apt-get install nano`

Now create file

server.js

```
'use strict';
```

```
var express = require('express'); var app = express();
```

```
app.get('/', (req, resp) => { console.log(req.originalUrl); resp.send('hello world'); });
```

```
app.get('/hello', (req, resp) => { console.log(req.originalUrl); resp.send('hello'); });
```

```
app.use('/web', express.static('pages'));
```

```
app.listen(8080);
```

Start Server

Npm add express

node server.js

Kill server => ctrl c

Create pages & add a file

```
mkdir pages
```

```
cd pages
```

```
nano ralph.txt
```

```
cd ..
```

```
node server.js
```

Kill server (ctrl c) & kill container (exit)