[Tech Forum] RobotFramework With Docker

Oct 25, 2024
In-Vehicle App. Development Team
Boin Jeong

Index

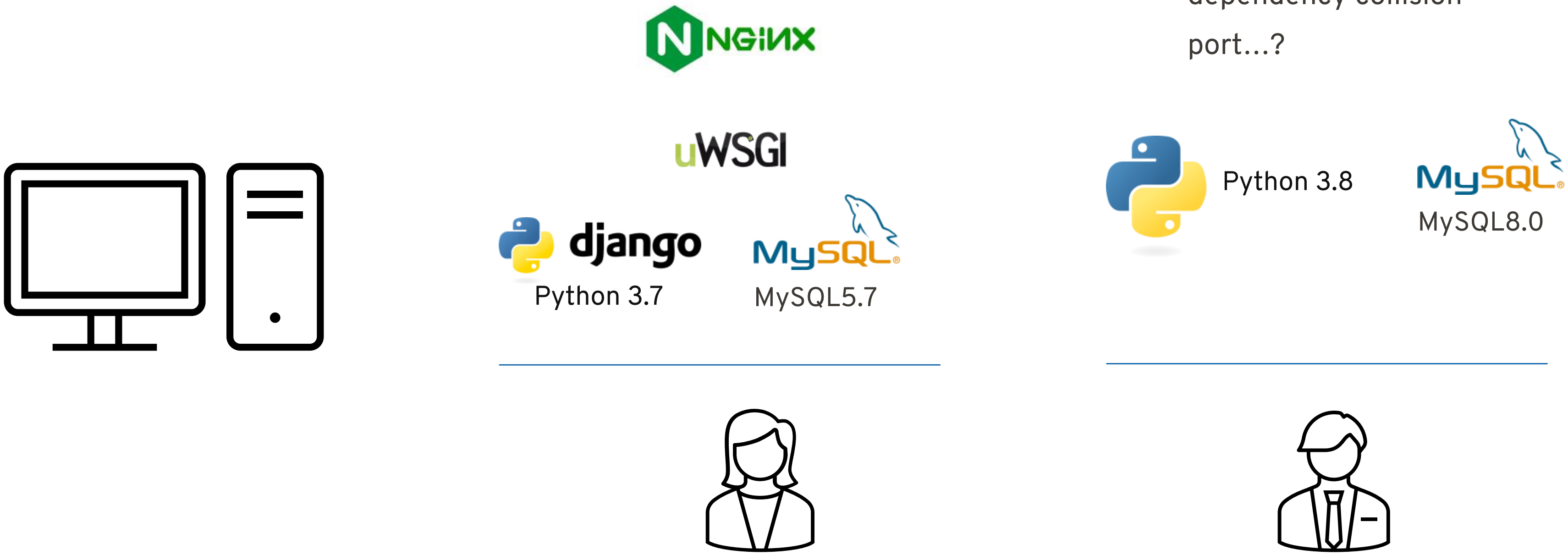
1. What is Docker ?
2. Advantage of using Docker
3. Case Studies in Env Setup
4. Robot Framework with Docker

1. What is Docker

Support that...

dependency collision

port...?



Python 3.7

MySQL5.7

Python 3.8

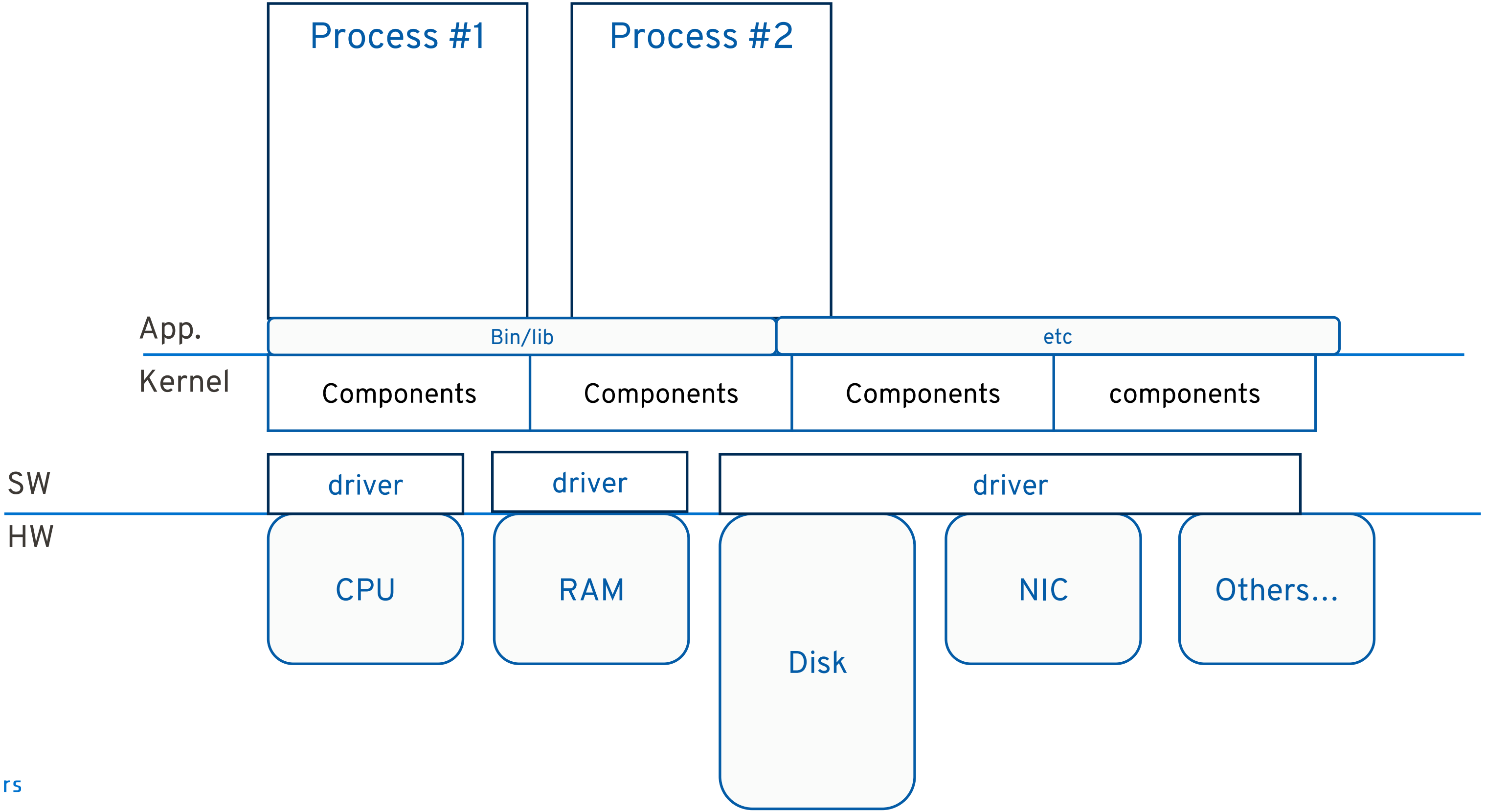
MySQL8.0

1. What is Docker

Why Developers want “Virtualization”

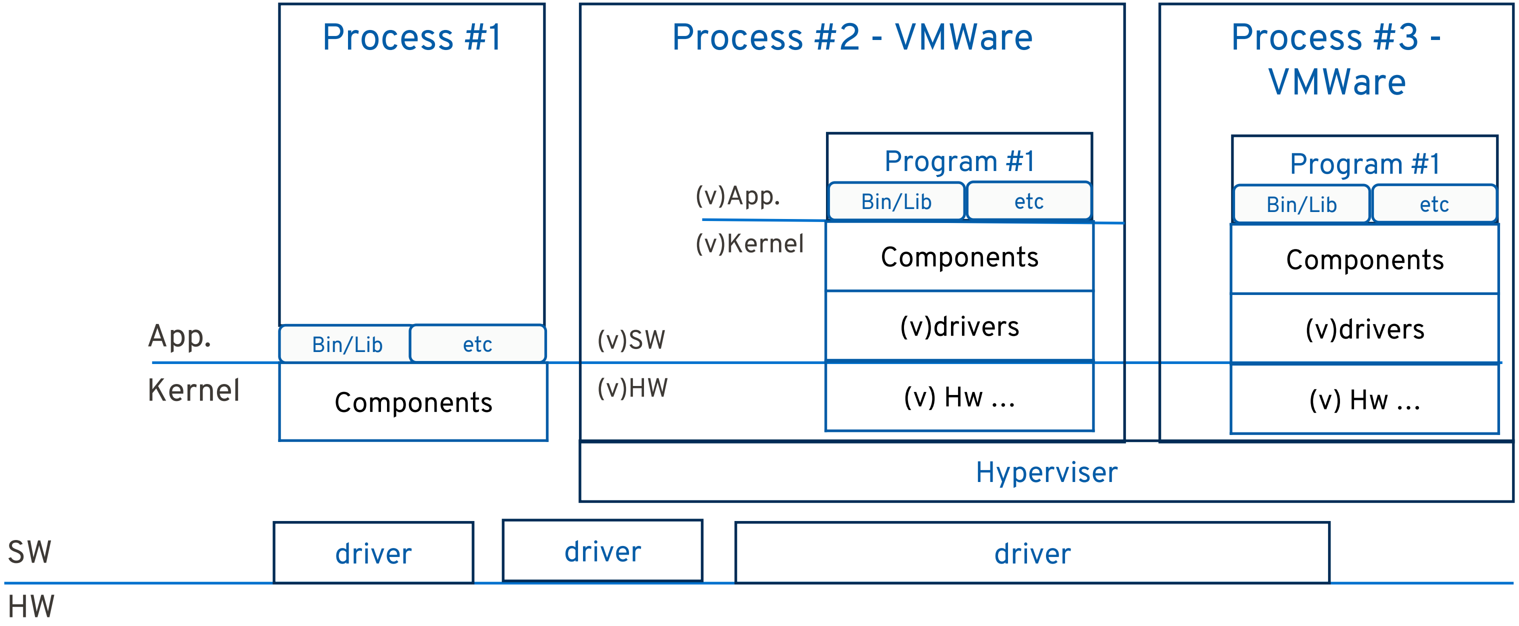
- **Dependency version mismatch:** Python 3.8 vs Python 3.9
- **Port conflicts**
- **System library conflicts**
- **Framework conflicts:** I want to use different versions of Express.js alongside different versions of Node
- **Environment variable conflicts**
- **OS dependency issues**
- **Inconsistency between development, testing, and deployment environments**

1. What is Docker



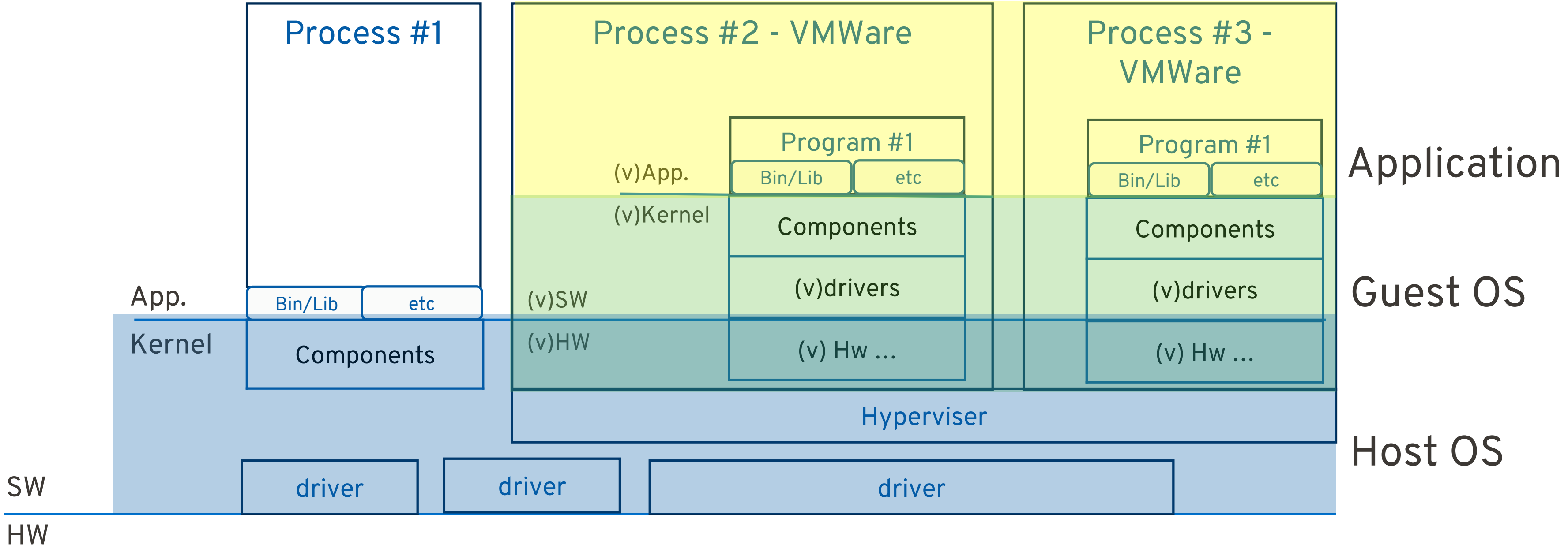
1. What is Docker

Using VM

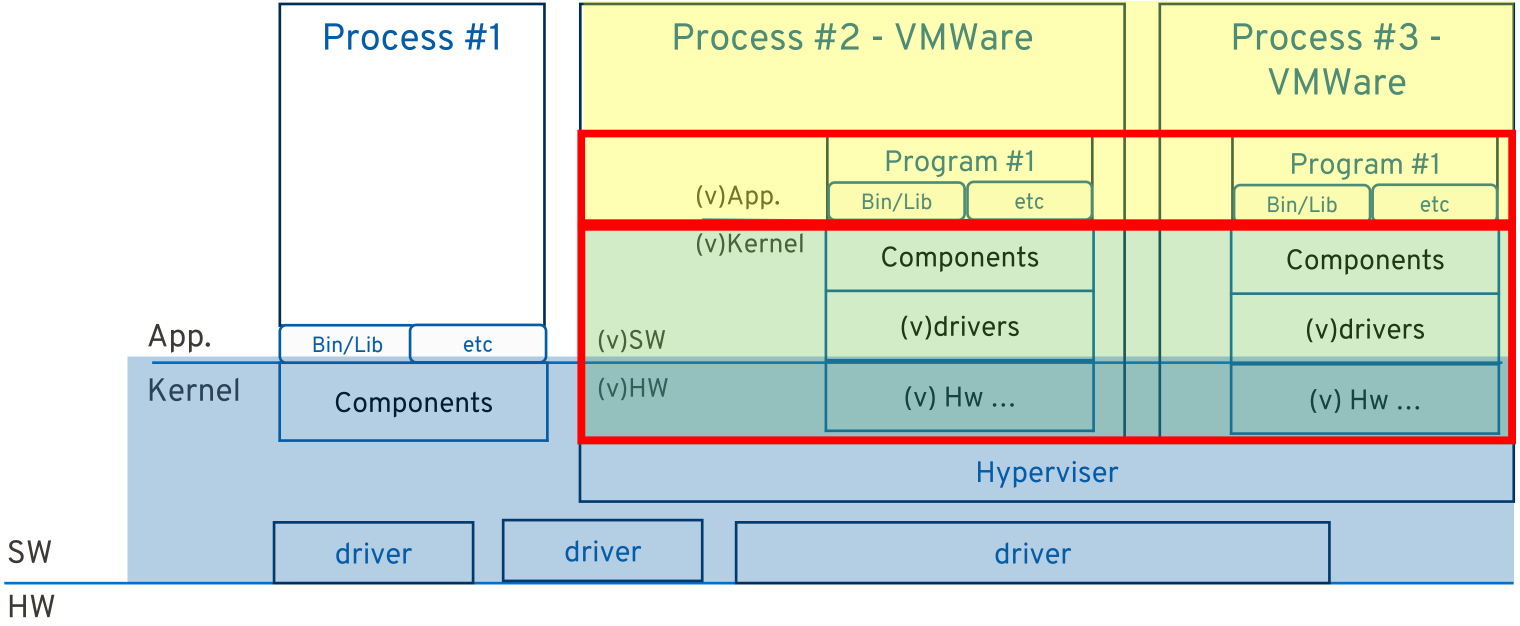


1. What is Docker

Using VM Ware

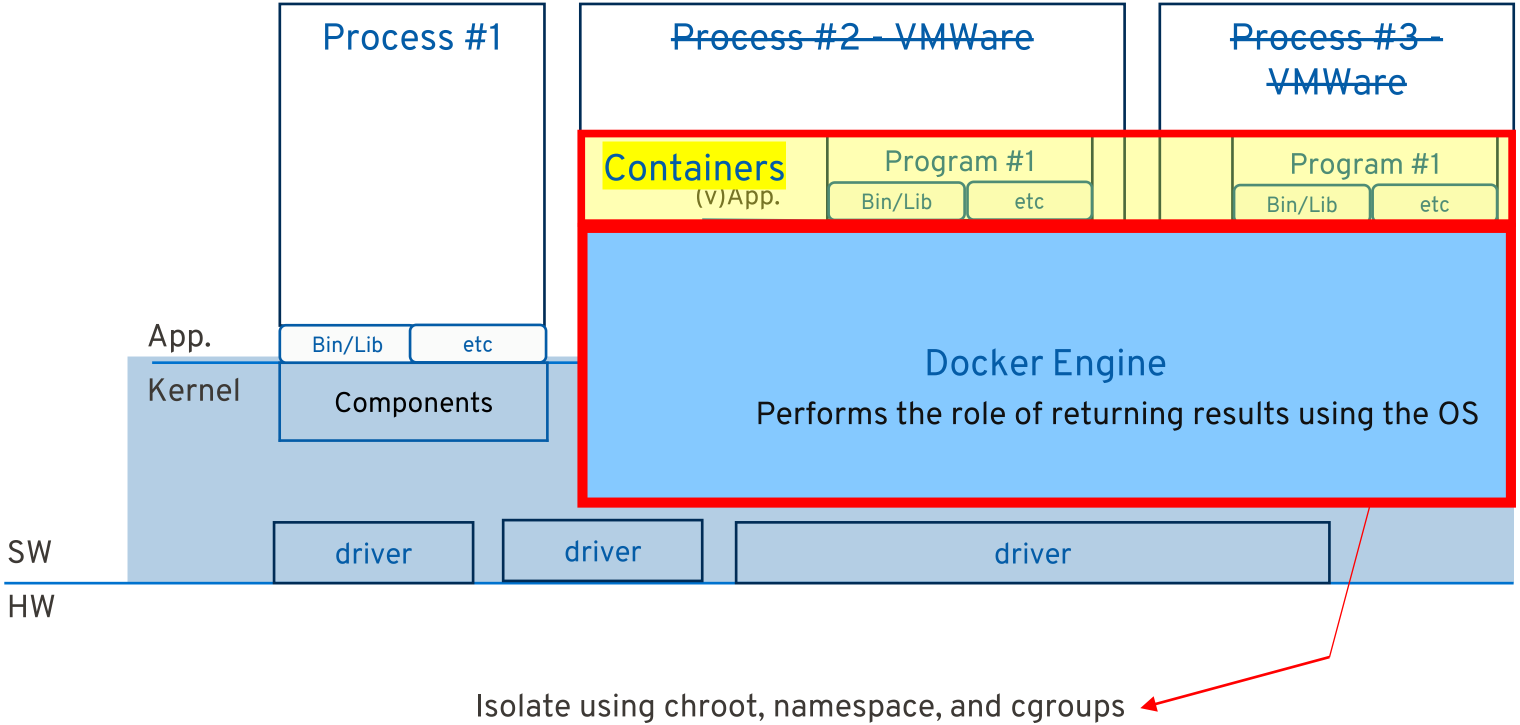


1. What is Docker



Bin/Lib 를 위해서
동일한 guestOS 를
띄워야할까요?

1. What is Docker



Isolate using chroot, namespace, and cgroups

2. Advantages of Using Docker

(Advantages)

- lighter than Virtual Machine
- Independence and scalability of multiple applications
- Standardized dev/test/deploy environment
- You don’t need to setup the environment multiple times !!!

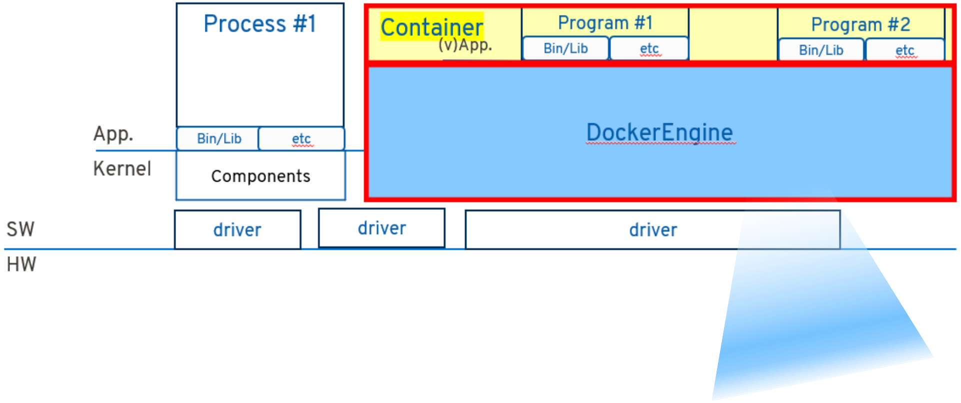
(Disadvantages)

- Storage Issues
- Complex Network configuration between containers
- **Frequent I/O can relatively impact performance negatively**

=> Kubernates : Container Management!

- Service Discovery & Load Balancing
- Storage Ochestration
- rollback/rollout

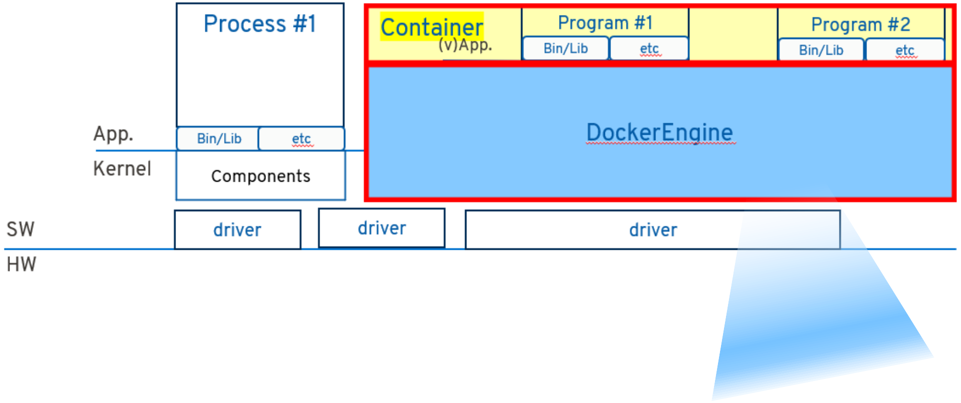
Appendix. Docker inside…



chroot (change root)

- A command that allows a specific process and its child processes to recognize a designated directory as the root, instead of the system’s /root
- When using the filesystem, the user perceives the specified directory as ‘/’

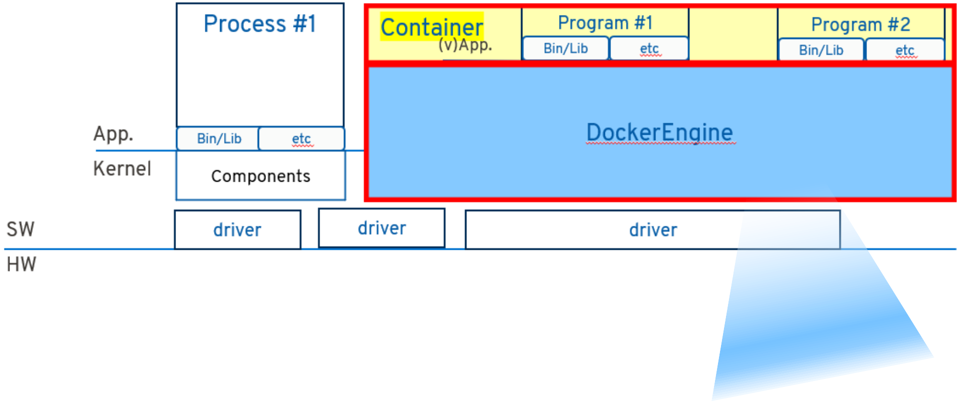
Appendix. Docker inside…



namespace

- Isolates the resources of processes: Allows each container to use resources independently.
- Ensures an independent resource view: Each container has its own PID space and isolates NIC, MNT, UTS, etc.

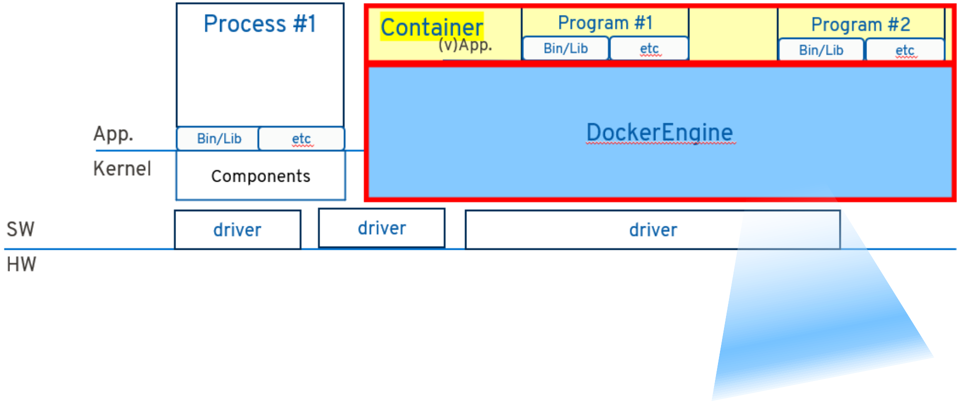
Appendix. Docker inside...



cgroup (control group)

- Limitation, monitoring, and management of resources on a per-process group basis
- CPU, memory, Disk I/O, network bandwidth…

Appendix. Docker inside…

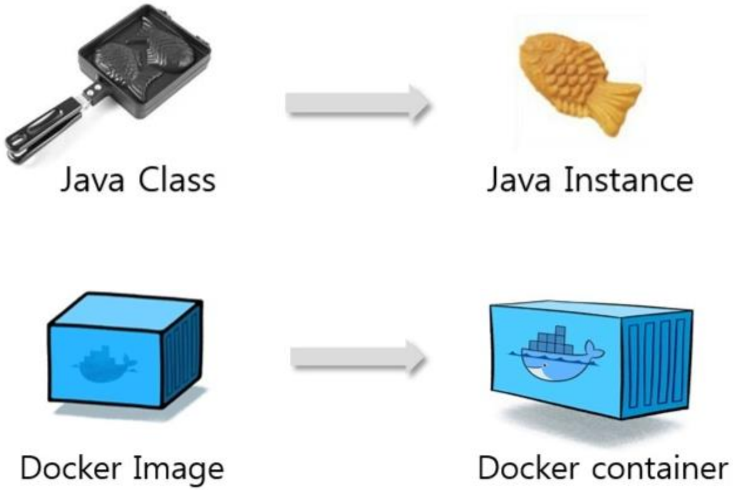


- **chroot** : Isolates files and maintains an independent view of the file system
- **namespace** : Isolates resource (processes, networks, IPC)
- **cgroup** : Manages and limits resources

Appendix. Terms

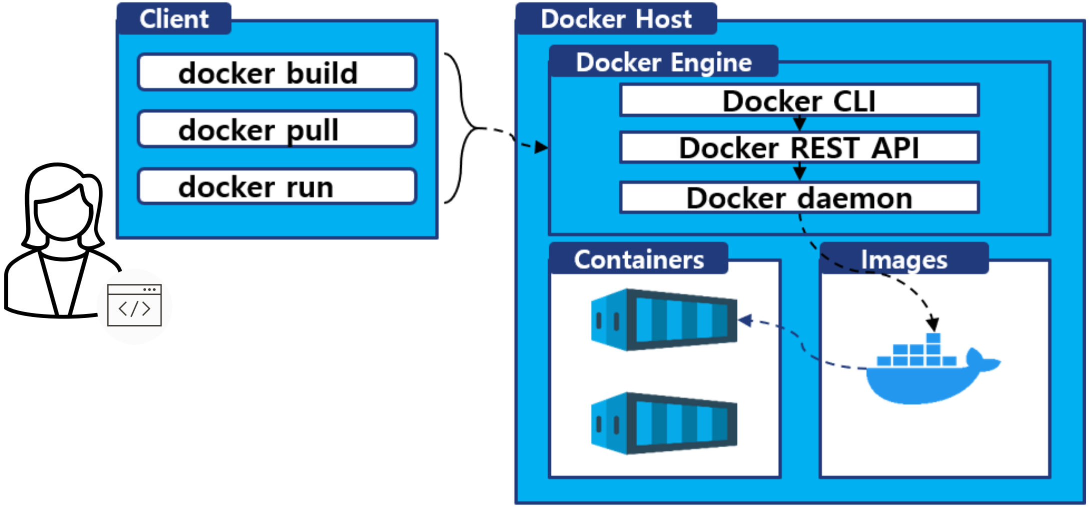
- **Docker** : An open-source project that runs and manages Linux applications in containers using process isolation technology.

- **Image & Container**



Appendix. Terms

- Docker Engine & Docker Daemon



▲그림. 도커 엔진을 포함시킨 도커 아키텍처

3. Case Studies in Env Setup

Dev/Test Environment Setup without Docker

1. Setting up Test Environment

   - Configure environments for applications – Python, Java …

2. **Individual Configuration**

   - Each Developer must set up the env on their own computer

3. Clone the Repository

   - download the latest code

4. Build or Download Executable Files

3. Case Studies in Env Setup

Dev/Test Environment Setup with Docker

1. Install and Setup Docker
2. Pull the Base Image

```
docker pull <base-image-name>
```

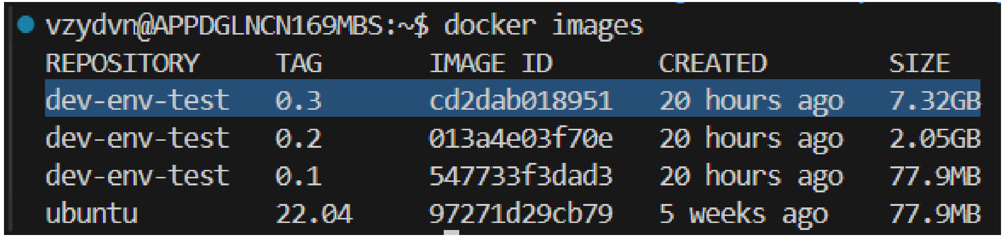
3. Open the Container and Set up the Test Environment

   – configure the environment for application(python, java …) **only Once.**

   - **Each Developer only needs to install the Docker, and the env setup should be done just once!**

4. Commit the Image and Deploy

```
docker commit <container-id> <new_image_name>:<tag>\
```



```
vzydvn@APPDGLNCN169MBS:~$ docker images
REPOSITORY     TAG       IMAGE ID       CREATED         SIZE
dev-env-test   0.3       cd2dab018951   20 hours ago    7.32GB
dev-env-test   0.2       013a4e03f70e   20 hours ago    2.05GB
dev-env-test   0.1       547733f3dad3   20 hours ago    77.9MB
ubuntu         22.04     97271d29cb79   5 weeks ago     77.9MB
```

Issue : The image size is too large.
How should we share/manage it?”

3. Case Studies in Env Setup

Dev/Test Environment Setup with Dockerfile

1. Create a Dockerfile and Define the base image (python, nodejs, java…)
2. Upload and Manage Dockerfile(text) in Github – **light and track changes easily!**
3. Download Dockerfile and Build the image at local

   ```
   docker build -t <image_name>:<tag> <dockerfile_dir>
   ```

4. Run the Docker Container

4. RobotFramework with Docker

Parallel Testing with Appium-Robot Framework Using Android Emulator and Docker

Goal

- Run Multiple Android Emulator locally – GUI / TestBench
- Execute automated test in Parallel – Time Saving

Required Tools

- Android Studio with Emulator
- Docker, **Docker Compose**
- Appium, Robot Framework, pabot(parallel execution tool for Robot Framework)

4. RobotFramework with Docker

What is Docker Compose?

- A tool for defining and running multi-container
- Define multiple containers as a single service in a YAML file

"You don't have to write commands one by one to launch multiple containers!"

4. RobotFramework with Docker

Parallel Testing with Appium-Robot Framework Using Android Emulator and Docker

1. **Write Test Scripts for Parallel Testing and Git Push**

   - Remind that we have to connect to each Appium Container and Emulator

```
*** Variables***
${APPIUM_1}          http://localhost:4723/wd/hub
${APPIUM_2}          https://localhost:4725/wd/hub

*** Test Cases ***
Test on Emulator1
    # Specify the adb-device id
Test on Emulator2
    # Specify the adb-device id
```

4. RobotFramework with Docker

Parallel Testing with Appium-Robot Framework Using Android Emulator and Docker

2. Create the Dockerfile for running Robot Framework Tests

```
FROM python:3.8

RUN pip install robotframework
RUN pip install robotframework-appiumlibrary pabot
…(download github, java, nodejs, etc and setup)

RUN git clone <test code repository> <working dir>
WORKDIR <working dir>
CMD [“pabot”, “--process”, “2”, “test_suite.robot”]
```

4. RobotFramework with Docker

Parallel Testing with Appium-Robot Framework Using Android Emulator and Docker

3. Create Docker Compose Configuration File (.yml)

- Service

```
version: ‘3’
services:
    appium1:
               image : /appium    #or create dockerfile for appium server
               ports :
               (etc… ip, port)
    appium2:

    robot_tests:
               build: <dockerfile dir>
               depends on:
```

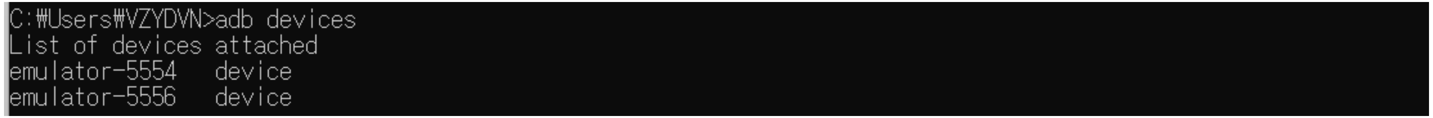
4. RobotFramework with Docker

Parallel Testing with Appium-Robot Framework Using Android Emulator and Docker

4. Running Multiple Android Emulators Locally

- Run multiple emulators, but in the GM Emulator Launcher, you need to import multiple system images to execute them. (ref. GM Launcher User Guide)

- Each instance is assigned two consecutive ports.



```
C:₩Users₩VZYDVN>adb devices
List of devices attached
emulator-5554   device
emulator-5556   device
```

4. Robot Framework with Docker

Parallel Testing with Appium-Robot Framework Using Android Emulator and Docker

5. Execute Parallel Tests

- Run Docker Compose & Parallel Tests

```
docker-compose up --build
```

4. Robot Framework with Docker

Parallel Testing with Appium-Robot Framework Using Android Emulator and Docker

(Advantages)

- Improved Testing Efficiency
- Scalability
- Environment Consistency

+) Building an automated CI/CD pipeline using Jenkins

(Disadvantages/Limits)

- ADB I/O Bottlenecks : frequent communication between containers and ADB
- Complex Setup and maintenance

+) To reduce resource contention, assigning memory limit/CPU cores to each Containers

Q & A