

HW2 Report (Deploy a simple service via Docker Compose)

A. Please simply explain your code

1. 為了滿足檔案一定要由 internet 傳遞，我先寫了 server.py 和 client.py，client 和 server 都採用了 python 的 twisted framework，並定義了自己的 GET & POST，以確保穩定性。
2. 在用 localhost 確定完正確性後，我開始將這兩個檔案包成 docker image，在 server 的 image 需要包括 JDK >9 和 gcc >=7 和 twisted 等套件，並且要有 python & twisted 的 package，而 client 的 image 則僅需要包括 twisted。
3. 最後，我使用了 docker-compose 來將兩個 image 給包起來，並且在 docker-compose 中宣告了 volume、新的 bridge 等設定，完成這次作業。

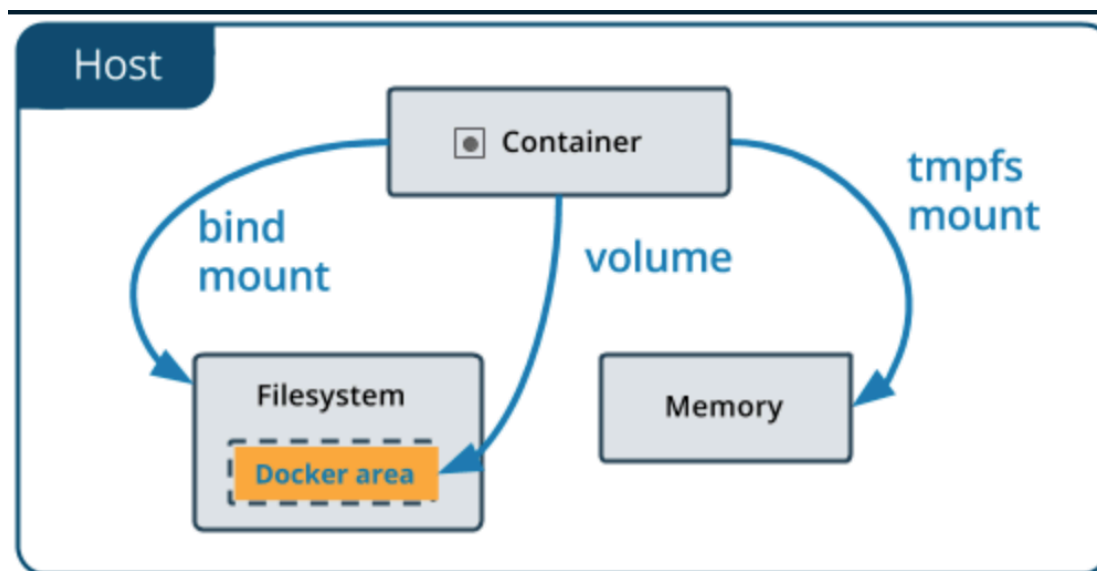
B. What is the main difference between container and VM

1. VM 和 container 最大的差別在於三點：
 1. VM 通常指一個完整的電腦環境，使用者可以將其當成一台獨立的電腦，Container 通常只將需要的服務都加進去 image 中而已，所以空間較小。
 2. VM 使用 hypervisor 來轉換使得使用者誤以為自己在真正的硬體上實作，而 container 則是透過 Container Engine 來跑服務。
 3. Container 的相容性近乎原生的系統，VM 則需要持續更新 hypervisor。

C. What are the three method for container to store data in host machine, what is the difference between them?

以下的三個方法都是 container 可以將資料儲存在 host machine 的方法，差別在於

1. Volumes: 會將資料儲存在 docker engine 維護的 docker area 中，方便 container 之間的溝通
 2. Bind Mount: 會將 host OS 的 folder bind 到 container 中的其他資料夾，讓 host 的資料可以和 container 中的同步
 3. Tmpfs Mount: 會直接將不太需要讓使用者知道的資訊放到 Memory 中。
- 由下圖可以很好的示意三種儲存的差異。




```

        avg:                                2.01ms
        max:                                3.57ms
        approx. 95 percentile:              2.02ms

Threads fairness:
    events (avg/stddev):                   10000.0000/0.00
    execution time (avg/stddev):           20.0835/0.00

yclo@ubuntu-vm:~$ sysbench --test=cpu --cpu-max-prime=20000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 20000

Test execution summary:
    total time:                            20.1745s
    total number of events:                 10000
    total time taken by event execution:    20.1734
    per-request statistics:
        min:                                2.00ms
        avg:                                2.02ms
        max:                                9.34ms
        approx. 95 percentile:              2.07ms

Threads fairness:
    events (avg/stddev):                   10000.0000/0.00
    execution time (avg/stddev):           20.1734/0.00

```

```
10240.00 MB transferred (47903.91 MB/sec)
```

Baseline

```

Test execution summary:
    total time:                            0.2138s
    total number of events:                 1310720
    total time taken by event execution:    0.1512
    per-request statistics:
        min:                                0.00ms
        avg:                                0.00ms
        max:                                0.06ms
        approx. 95 percentile:              0.00ms

```

```

Threads fairness:
    events (avg/stddev):                   1310720.0000/0.00
    execution time (avg/stddev):           0.1512/0.00

```

```
yclo@ubuntu-vm:~$ sysbench --test=memory --memory-block-size=8K --memory-total-size=1G run
sysbench 0.4.12: multi-threaded system evaluation benchmark
```

```

Running the test with following options:
Number of threads: 1

```

```

Doing memory operations speed test
Memory block size: 8K

```

```
Memory transfer size: 1024M
```

```

Memory operations type: write
Memory scope type: global
Threads started!
Done.

```

```
Operations performed: 131072 (1451301.66 ops/sec)
```

```
1024.00 MB transferred (11338.29 MB/sec)
```

Live Migration