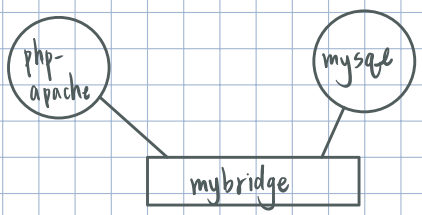
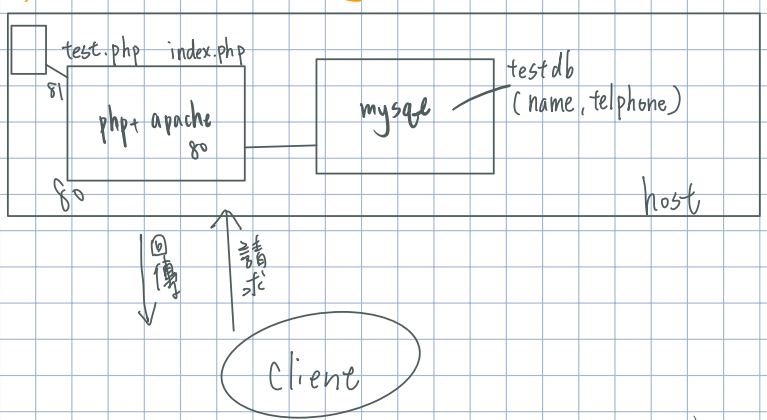


事前準備工作

```
# docker pull mysql
# docker pull radys/php-apache:7.4
```

多容器彼此互相合作、通訊



創造橋接 net

```
# docker network create -d bridge mybridge
# mkdir mydbdata
```

創 mysql 容器

```
# docker run -d --network mybridge --name hello-mysql -e
MYSQL_ROOT_PASSWORD=123456 -v /root/mydbdata:/var/lib/mysql mysql
```

設資料庫密碼　資料存在主地端

```
# docker ps
# docker exec -it hello-mysql bash
# mysql -u root -p
```

進入 db

password: 123456

```
> show databases;
```

顯示目前的 db

```
> create database [NAME] testdb;
```

創造 db

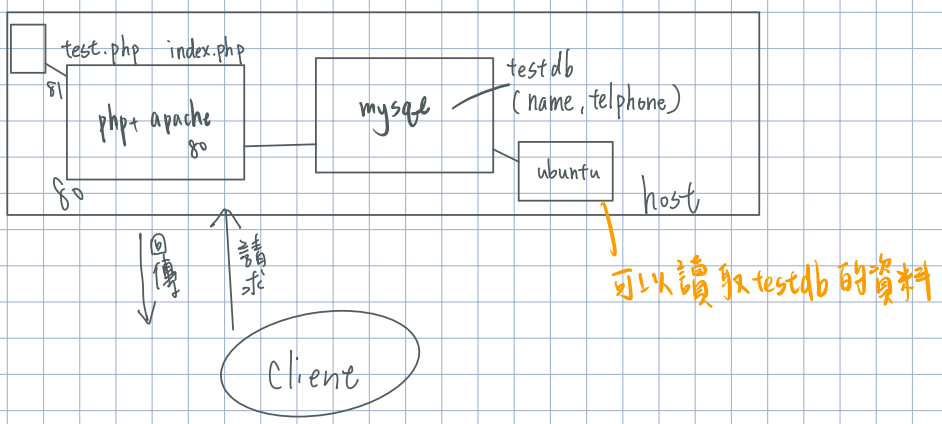
```
> show databases;

> use testdb;          切換到testdb

建立表格
> create table [NAME] addrbook (name varchar(50) not null, phone varchar(10));
                                  可變動的大小
                  第1欄位,姓名,最多50個字,不能空

插入data
> insert into addrbook (name, phone) values ("Tom", "12345");
> insert into addrbook (name, phone) values ("Amy", "123");

查看table裡data
> select * from addbook;


# mkdir myphp

創 php-apache容器
# docker run -d -p 80:80 --network mybridge -v /root/myphp:/var/www/html
  --name hello-php radys/php-apache:7.4

# cd myphp

# vim test.php
  <?php
  phpinfo();
  ?>

Chrome search 192.168.249.138/test.php

√

寫一個程式,連到mysql把資料讀出來,用網頁顯示data給客戶端

# vim index.php

  <?php

  $servername = "hello-mysql";
  $username = "root";
  $password = "123456";
  $dbname = "testdb";

  $conn = new mysqli($servername, $username, $password, $dbname);

  if ($conn->connect_error)
```

```
    die("Connection failed: " . $conn->connect_error);
}

$sql = "select name, phone from addrbook";
$result = $conn->query($sql);

if ($result->num_rows > 0){          讀取資料
    while($row = $result->fetch_assoc()){
        echo "name:" . $row["name"]. "  phone:" .$row["phone"]. "<br>";
    }
} else {
    echo "0 result";
}

?>
```

chrome search centos7 ip /index.php

name: Tom, phone: 12345
name: Amy, phone: 123

在另一個容器連到資料庫




```
# docker run -it --network mybridge ubuntu

/# apt update
  # apt install mysql-client

  # mysql -u root -p -h hello-mysql
                        資料庫所在的位置
```

```
> show databases;
> use testdb;
> select * from addrbook;
```

持久化：資料保存在主地端

保存方式：① host volume：在主地指明路徑掛載上去，要找空間建立資料夾然後對應到容器

```
# docker run -d -p 80:80 --network mybridge -v /root/myphp:/var/www/html/
  --name hello-php radys/php-apache:7.4
```

volume

② named volume：先創造一個 named volume，之後只指明掛載 named volume

```
# docker volume ls
# docker volume create [NAME] myvol1
# docker volume ls
    myvol1
# docker volume inspect myvol1
    Mountpoint: /var/lib/docker/volumes/myvol1/_data
# docker run -it -v myvol1:/mydata --name test1 busybox sh
  /#cd mydata
    /# echo "123" > 123.txt
    /# exit

# cd /var/lib/docker/volumes/myvol1/_data
# ls
  123.txt
# cat 123.txt
  123
# docker run -it -v myvol1:/mydata --name test2 busybox sh
  /#cd mydata
    /# ls
      123.txt
    /# exit
```

```
# docker run -d -p 81:80 --network mybridge -v /root/myphp:/var/www/html/
```

--name hello-php1 radys/php-apache:7.4

Chome

192.168.249.138:81/index.php