

EC2 > 安全組 > click 已有的安全組 > 傳入規則 > 編輯規則 > HTTP - 源: 0.0.0.0/0 >

```
$ nmap 127.0.0.1
```

```
$ docker pull komavideo/deeplearnaws-web:latest
```

```
$ docker run --name deeplearnaws-web -p 80:3000 -d komavideo/deeplearnaws-web:latest
```

```
$ docker container ls -a
```

```
$ docker logs -f deeplearnaws-web
```

```
$ docker container stop deeplearnaws-web
```

```
$ docker container rm deeplearnaws-web
```

```
$ docker ps
```

```
# 編輯調整
```

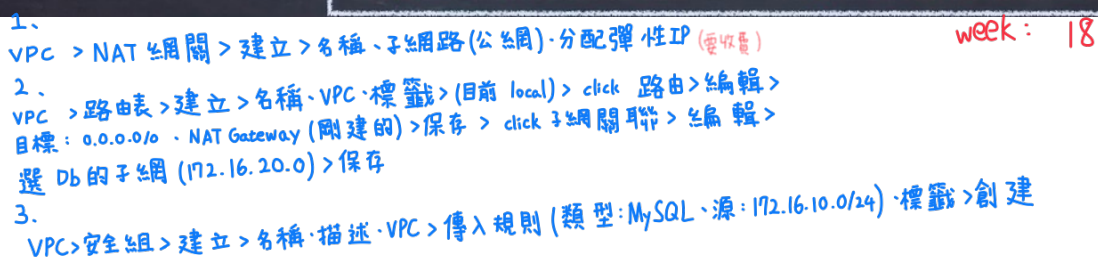
```
$ docker exec -it deeplearnaws-web sh
```

```
$ ls
```

```
> vi app.js
```

```
> exit
```

```
$ docker container restart deeplearnaws-web
```



EC2 > 安全組 > (db) > 傳入規則 > SSH - 172.16.10.0/24 > 保存

後的 DB 要用成 Image (所以用 AMI 做)

EC2 > AMI > 啟動 > Config (網路、子網 > DB、自動分配 IP: 禁用) > 網路接口 (172.16.20.10) > storage > 標籤 > 現有安全組 (選 Db) > 啟動 > launch > 選擇現有 key pair (並打勾) > 之後要把 key pair copy 到 172.16.10.10 的 host 上

先 ssh 到 172.16.10.10 > vi deeplearnaaws-ssh-key.pem (複製本機的內容) >
chmod 400 deeplearnaaws-ssh-key.pem > ssh 到 172.16.20.10 >

如果 error (Add correct host key in /home/ec2-user/.ssh/known-hosts 則下指令 `rm -fr /home/ec2-user/.ssh/known-hosts`)

`> sudo yum update -y` (查看 172.16.20.10 是否可以連 Internet)

* 在公有網段安裝 MySQL 數據庫實例

數據庫端(deeplearnaws-db1)

```
$ docker pull komavideo/deeplearnaws-mysql:latest
```

```
$ docker run --name deeplearnaws-mysql -e MYSQL_ROOT_PASSWORD=12345678 -p 3306:3306 -d --restart=always komavideo/deeplearnaws-mysql:latest
```

```
$ docker ps
```

```
# 進入容器bash-shell
```

```
$ docker exec -it deeplearnaws-mysql bash -p
```

```
>mysql -u root -p -h 127.0.0.1
```

```
>pass:12345678
```

```
mysql> show databases;
```

```
mysql> use blogdb;
```

```
mysql> select * from user;
```

```
mysql> exit;
```

Web端(deeplearnaws-web1)

```
``bash
```

```
$ mkdir myapp
```

```
$ cd myapp
```

```
$ npm init -y
```

```
$ npm install mysql --save
```

```
$ nano main.js
```

```
...(main.js)
```

```
$ node main.js
```

```
...
```

main.js

```
const mysql = require('mysql');
```

```
const connection = mysql.createConnection({
```

```
  host: '172.16.20.10',
```

```
  user: 'root',
```

```
  password: '12345678',
```

```
  database: 'blogdb'
```

```
});
```

```
connection.connect((err) => {
```

```
  if (err) {
```

```
    console.log('error connecting: ' + err.stack);
```

```
    return;
```

```
  }
```

```
  connection.query(
```

```
    'SELECT * FROM user', (error, results) => {
```

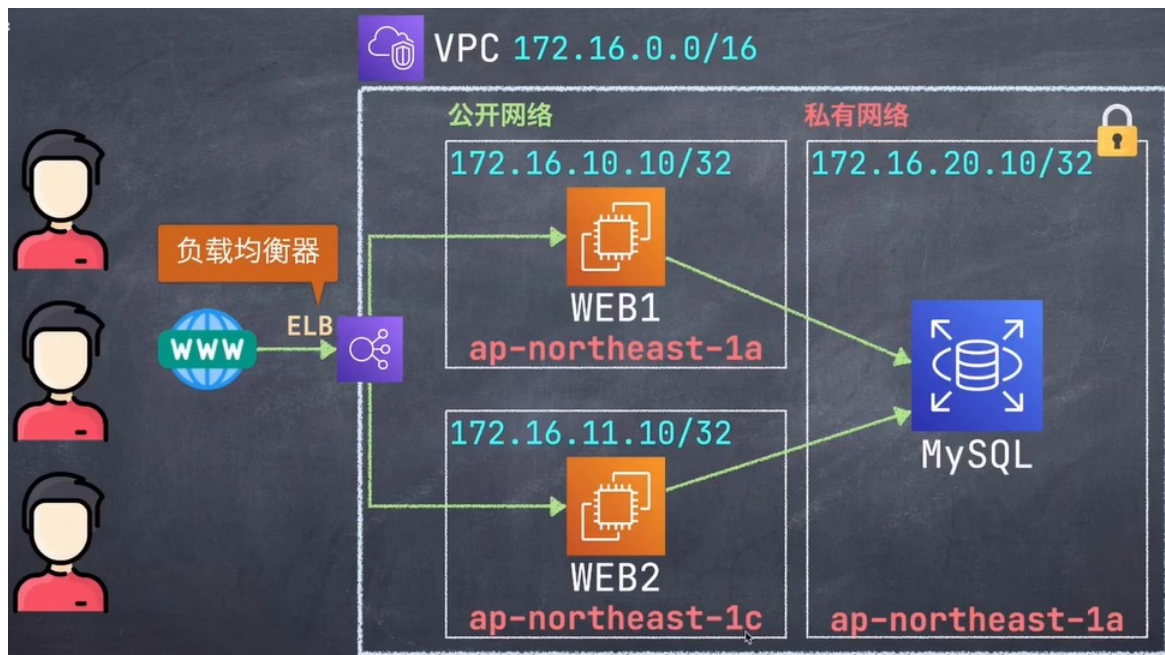
```
      console.log(results);
```

```
      process.exit()
```

```
    }
```

```
  );
```

```
});
```



VPC > 子網 > 建立 > VPC ID、子網名稱、可用區(2c)、IP → 172.16.11.0/24、標籤 > 創建 > 子網(2c) > 路由表 > 編輯 > 選 路由表ID(web-rtb) > 儲存
 EC2 > AMI > 啟動 > Config (網路-子網(web-2c)、自動分配IP → 啟用 > 網路接口 (172.16.11.10) > storage > 標籤 > 現有安全組 (選 web) > 啟動 > 現有 key pair (打勾)
 ssh 到 172.16.11.10 > 執行 week 17 (pull web on DockerHub、run、exec、restart)

建立目標組 (含 web1、web2)、建屬於目標組的 ALB

打開 web1、web2 instance > EC2 > 目標組 > 建立 > 實例、名稱、VPC、HTTP

運行狀況檢查 (路徑 → /、高級設置: 間隔 → 10)、標籤 > 選 web1、web2 並註冊 > 創建

EC2 > 負載均衡器 > 建立 > HTTP 建立 > 名稱、面向內部 internet、偵聽器、Mapping (VPC、web1、web2) 標籤、
 配置安全組 (創建新安全、名稱、描述、80 port)、Listener & routing (現有目標組) > 創建

此時在 ALB 中, DNS 名稱是 web1、web2 的 URL