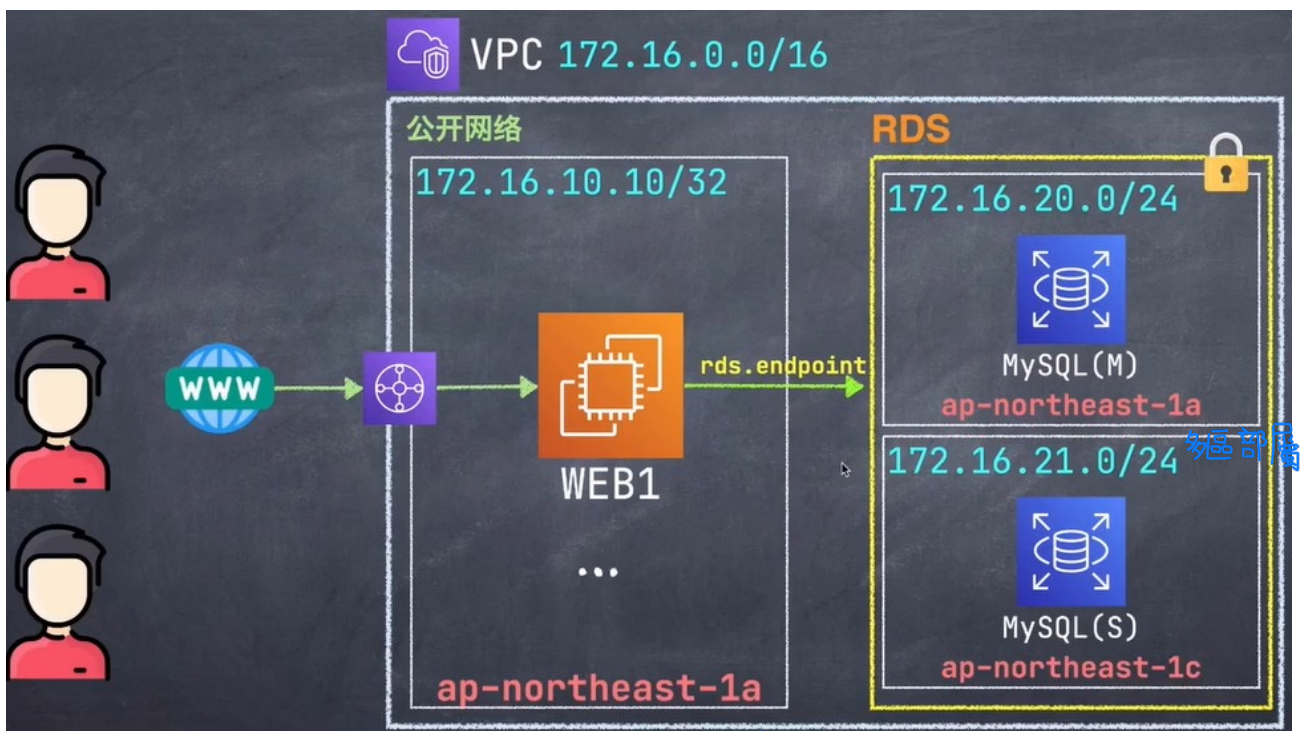


传统方式	EC2	RDS
应用程序	应用程序	应用程序
DB扩容	DB扩容	DB扩容
数据备份	数据备份	数据备份
引擎升级	引擎升级	引擎升级
OS维护	OS维护	OS维护
服务器维护	服务器维护	服务器维护

讓開發人員避免
管理的麻煩



VPC > 子網 > 建立 > VPC ID · 名稱 · 可用區 · CIDR · 標籤 > 創建

VPC > 安全組 > 建立 > 名稱 · 傳入規則

RDS > 子網組 > 建立 > 名稱 · VPC > 新增子網 (選多個可用區 · 子網 → 172.16.21.0 - 172.16.20.0) > 創建

RDS > 參數組 > 建立 > MySQL 5.7 · 名稱 > 創建 > click 可查看參數

RDS > 選項組 > 建立 > 名稱 · 引擎 · 版本 > 創建

RDS > 創建數據庫 > 標準建立、MySQL 5.7.31、免費 > 設置 (個體識別符、用戶名、密碼) >
DB 實例大小 (db.t2.micro) > SSD、啟用自動擴充 > 連接 (VPC、子網組、公開訪問否、現存主組) >
DB 身份驗證 (剛學用密碼身份) > 其他組態 (名稱、參數組、選項組 { 每自動備份 * 花費 } >
創建 > 終端節點 (URL)
end-point

MySQL 客戶端工具, 連接到 MySQL RDS

EC2 建台 server > ssh -i xxx.pem ec2-user@ip > 以下指令

```
$ sudo yum -y update
$ sudo yum -y install mysql
# 下面是 rds end point url
$ mysql -h mysqldev.cbcbjcywwn.ap-northeast-1.rds.amazonaws.com -u root -p
Enter password:
mysql> show databases;
mysql> use blogdb;
mysql> create table blogdb.user (id int, name varchar(255));
mysql> show tables;
mysql> insert into blogdb.user values (1, 'Koma');
mysql> insert into blogdb.user values (2, 'Xiaoma');
mysql> insert into blogdb.user values (3, 'MySql');
mysql> select * from blogdb.user;
mysql> exit;
```

連結MySQL phpMyAdmin 工具

```
$ vi docker-compose.yml
...
version: '3'
services:
  phpmyadmin:
    image: phpmyadmin:latest
    container_name: web_phpmyadmin
    ports:
      - 80:80
    environment:
```

- PMA_HOST=RDS的endpoint url
- PMA_USER=當時aws上設定的(root)
- PMA_PASSWORD=當時aws上設定的

...

編譯服務

\$ sudo docker-compose build

容器啟動

\$ sudo docker-compose up -d

查詢狀態

\$ sudo docker-compose ps

在自己的local端上 輸入URL(ec2)

把上一期的phpMyAdmin container關掉(因為80端口)

接著開啟NodeJs. Web

從DockerHub上獲取deeplearnaws-web image

\$ docker pull komavideo/deeplearnaws-web:latest

\$ docker run --name deeplearnaws-web -p 80:3000 -d --restart=always komavideo/deeplearnaws-web:latest

\$ docker container ls -a

編輯

\$ docker exec -it deeplearnaws-web sh

>vi app.js

...

修改 host

mysqldev.cbcbjcywwn.ap-northeast-1.rds.amazonaws.com

...

\$ docker container restart deeplearnaws-web

在自己的local端上 輸入 URL

在自己的local端上 輸入 URL/blog