

# Cài và sử dụng minikube

## Mục lục

1. Tải docker về .....	1
2. Cài minikube .....	3
3. Thiết lập PV và Pod mysql .....	4
4. Truy cập và điền dữ liệu trong pod đã chạy.....	6
5. Kiểm tra kết nối đến pod mysql bằng dbear .....	8

## 1. Tải docker về

// update

**sudo apt-get update**

<// cài các gói cần thiết

□ **apt-transport-https:** Cung cấp khả năng truyền tải các gói phần mềm qua HTTPS, giúp bảo mật khi tải các gói từ các kho lưu trữ bên ngoài.

□ **ca-certificates:** Cài đặt các chứng chỉ CA để hệ thống có thể xác thực các kết nối SSL/TLS an toàn.

□ **curl:** Công cụ dòng lệnh dùng để chuyển dữ liệu từ hoặc đến một máy chủ, sử dụng nhiều giao thức khác nhau.

□ **software-properties-common:** Cung cấp các lệnh tiện ích như add-apt-repository, giúp dễ dàng thêm các kho lưu trữ phần mềm mới.

/>

**sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common**

<// Thêm Docker GPG key

Docker ký các gói phần mềm của họ bằng GPG key để đảm bảo tính toàn vẹn và bảo mật. Lệnh này tải xuống và thêm GPG key của Docker vào hệ thống, cho phép hệ thống xác thực các gói Docker khi cài đặt.

/>

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key  
add -
```

</ thêm docker repository

Lệnh này thêm kho lưu trữ chính thức của Docker vào danh sách các nguồn phần mềm của hệ thống. \$(lsb\_release -cs) sẽ tự động thay thế bằng tên mã phát hành của Ubuntu (ví dụ: bionic, focal) để đảm bảo rằng bạn đang nhận được phiên bản phù hợp với hệ thống của mình.

/>

```
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

<// update

Sau khi thêm Docker repository, cần phải cập nhật lại danh sách gói để hệ thống nhận diện các gói mới có sẵn từ kho lưu trữ của Docker.

/>

```
sudo apt-get update
```

```
// cài docker
```

```
sudo apt-get install -y docker-ce
```

<// Khởi động Docker và thiết lập Docker chạy tự động khi khởi động hệ thống

- ☐ `sudo systemctl enable docker`: Lệnh này cấu hình Docker để tự động khởi động khi hệ thống khởi động. Điều này giúp Docker luôn sẵn sàng chạy container khi hệ thống bật lên.

- ☐ `sudo systemctl start docker`: Khởi động dịch vụ Docker ngay lập tức, giúp bạn có thể bắt đầu sử dụng Docker ngay sau khi cài đặt.

/>

**sudo systemctl enable docker**

**sudo systemctl start docker**

</ thêm user vào nhóm docker

Nếu bạn dự định sử dụng Docker làm driver cho Minikube, bạn cần thêm user của bạn vào nhóm Docker và khởi động lại phiên làm việc:

Sau khi thực hiện lệnh trên, khởi động lại phiên terminal của bạn hoặc đăng nhập lại.

/>

**sudo usermod -aG docker \$USER**

**newgrp docker**

// kiểm tra trạng thái và start

**sudo systemctl status docker**

**sudo systemctl start docker**

## 2. Cài minikube

// cài công cụ cần thiết

**sudo apt-get update**

**sudo apt-get install -y curl apt-transport-https**

// cài minikube

**curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64**

**sudo install minikube-linux-amd64 /usr/local/bin/minikube**

// khởi động minikube

**minikube start --driver=docker**

// cài kubectl

**curl -LO "https://dl.k8s.io/release/v1.26.0/bin/linux/amd64/kubectl"**

**chmod +x ./kubectl**

**sudo mv ./kubectl /usr/local/bin/kubectl**

### 3. Thiết lập PV và Pod mysql

Tạo 1 file có tên **mysql\_pv\_pvc.yaml** có nội dung sau:

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: mysql-pv**

**spec:**

**capacity:**

**storage: 5Gi**

**accessModes:**

**- ReadWriteOnce**

**hostPath:**

**path: /mnt/data**

**storageClassName: manual**

**---**

**apiVersion: v1**

**kind: PersistentVolumeClaim**

**metadata:**

**name: mysql-pvc**

**spec:**

**accessModes:**

**- ReadWriteOnce**

**resources:**

**requests:**

**storage: 5Gi**

**storageClassName: manual**

apply file đó

**kubectl apply -f mysql\_pv\_pvc.yaml**

tạo 1 file **mysql\_pod.yaml** để chạy mysql

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: mysql**

**spec:**

**containers:**

**- name: mysql**

**image: mysql:5.7**

**env:**

**- name: MYSQL\_ROOT\_PASSWORD**

**value: ""**

**ports:**

**- containerPort: 3306**

**volumeMounts:**

**- mountPath: /var/lib/mysql**

**name: mysql-storage**

**volumes:**

**- name: mysql-storage**

**persistentVolumeClaim:**

**claimName: mysql-pvc**

cập nhật :

**kubectl apply -f mysql\_pod.yaml**

## 4. Truy cập và điền dữ liệu trong pod đã chạy

Dùng lệnh sau để kết nối đến mysql

**kubectl exec -it mysql bash**

**mysql -u root -p**

sau bước này là ta có thể truy cập vào ứng dụng mysql của chúng ta. Giờ hãy thực hiện sử dụng cơ sở dữ liệu này để thực hiện các công việc mong muốn

ví dụ ta có file sample\_data.sql

**-- Tạo cơ sở dữ liệu mới**

**CREATE DATABASE IF NOT EXISTS SampleDB;**

**USE SampleDB;**

**-- Tạo bảng `Users`**

**CREATE TABLE IF NOT EXISTS Users (**

```
id INT AUTO_INCREMENT PRIMARY KEY,  
username VARCHAR(50) NOT NULL,  
email VARCHAR(100) NOT NULL,  
age INT NOT NULL,  
registered_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

-- Chèn 10 dòng dữ liệu vào bảng `Users`

```
INSERT INTO Users (username, email, age) VALUES ('alice',  
'alice@example.com', 25);
```

```
INSERT INTO Users (username, email, age) VALUES ('bob',  
'bob@example.com', 30);
```

```
INSERT INTO Users (username, email, age) VALUES ('carol',  
'carol@example.com', 28);
```

```
INSERT INTO Users (username, email, age) VALUES ('dave',  
'dave@example.com', 35);
```

```
INSERT INTO Users (username, email, age) VALUES ('eve',  
'eve@example.com', 22);
```

```
INSERT INTO Users (username, email, age) VALUES ('frank',  
'frank@example.com', 40);
```

```
INSERT INTO Users (username, email, age) VALUES ('grace',  
'grace@example.com', 29);
```

```
INSERT INTO Users (username, email, age) VALUES ('heidi',  
'heidi@example.com', 31);
```

```
INSERT INTO Users (username, email, age) VALUES ('ivan',  
'ivan@example.com', 33);
```

```
INSERT INTO Users (username, email, age) VALUES ('judy',  
'judy@example.com', 27);
```

điền dữ liệu:

```
kubectl cp sample_data.sql mysql:/tmp/sample_data.sql
```

```
kubect exec -it mysql -- bash
```

```
mysql -u root -p < tmp/sample_data.sql
```

nếu ta có sẵn dữ liệu ngoài, ta có thể tiến hành export ra file .sql theo các bước sau (đối với mysql workbench):

- Từ menu chính, chọn Server -> Data Export. Bạn cũng có thể tìm thấy tùy chọn này trong thanh công cụ chính.
- Trong cửa sổ Data Export, chọn schema (trong trường hợp này là baitaploncsdl) từ danh sách các schema có sẵn.
- Chọn các bảng (tables) bạn muốn xuất. Nếu bạn muốn xuất toàn bộ schema, bạn có thể chọn tất cả các bảng hoặc chọn Dump Structure and Data để bao gồm cả cấu trúc và dữ liệu.
- Trong phần Export Options, chọn định dạng xuất là SQL.
- Bạn có thể chọn các tùy chọn xuất khác như Export to Self-Contained File (xuất ra file đơn) hoặc Export to Dump Project Folder (xuất ra thư mục chứa nhiều file).
- Chọn đường dẫn lưu file .sql bằng cách nhấp vào nút ... bên cạnh ô Export to Self-Contained File hoặc Export to Dump Project Folder.
- Chọn một thư mục và đặt tên cho file .sql.
- Nhấn nút Start Export để bắt đầu quá trình xuất dữ liệu
- MySQL Workbench sẽ tạo ra file .sql chứa cấu trúc và dữ liệu của schema baitaploncsdl.

Sau khi có file baitaploncsdl.sql ta thực hiện các bước giống việc điền dữ liệu ở trên:

```
kubect cp sample_data.sql mysql:/tmp/sample_data.sql
```

```
kubect exec -it mysql -- bash
```

```
mysql -u root -p < tmp/sample_data.sql
```

## **5. Kiểm tra kết nối đến pod mysql bằng dbear**



Trước khi kiểm tra được connection, ta tạo một service nodeport để có thể kết nối ra ngoài

Tạo 1 file mysql-service để expose port

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: mysql-service**

**labels:**

**app: mysql**

**spec:**

**selector:**

**app: mysql**

**ports:**

**- protocol: TCP**

**port: 3306**

**targetPort: 3306**

**nodePort: 30006 # Cổng trên các node**

**type: NodePort**

apply

**kubectl apply -f mysql-service.yaml**

sau khi apply ta đã có 1 service để kết nối đến các luồng bên ngoài

ta thực hiện lấy IP của minikube bằng lệnh **minikube ip**

vậy ta có cặp IP : minikube IP / port : 30006

tiếp theo đến cài đặt Dbeaver

**sudo add-apt-repository ppa:serge-rider/dbeaver-ce**

**sudo apt-get update**

**sudo apt-get install dbeaver-ce**

sau khi cài đặt thành công, ta vào dbeaver và tạo 1 connection đến pod mysql

host : minikube IP

port :30006

root

password\_root

databases : baitaplonsdl

ta test connect đến connection đó và nếu thành công là hoàn thành.