**Docker Swarm**

Trong quá trình phát triển, quản lý, scale cũng như deploy project của bạn với việc dùng lệnh của Docker để deploy thì ban đầu project nhỏ chỉ cần chạy một host(vps) nên không có vấn đề gì cả. Tuy nhiên khi project đó vì một số yêu cầu hoặc lý do gì đó phải cần thêm nhiều host hoặc rất nhiều host(vps). Lúc này bạn khó có thể quản lý, scale và cũng không thể nào dùng lệnh để đi deploy lên từng con host(vps) điều đó rất vật vã. Biết được nỗi lòng đó Docker đã phát triển thêm cho ta cái gọi là **Docker Swarm.**

**Khái niệm:**

* **Docker Swarm** là công cụ native clustering cho Docker. Cho phép ta có thể gom một số Docker host lại với nhau thành dạng cụm (cluster) và ta có xem nó như một máy chủ Docker ảo (virtual Docker host) duy nhất. Và một Swarm là một cluster của một hoặc nhiều Docker Engine đang chạy. Và Swarm mode cung cấp cho ta các tính năng để quản lý và điều phối cluster.

Các tính năng của **Docker Swarm:**

* **Cluster management integrated with Docker Engine:** Quản lý cluster với Docker Engine bằng việc sử dụng Docker CLI để tạo swarm.
* **Decentralized design:**Docker Swarm được thiết kế dạng phân cấp. Thay vì xử lý sự khác biệt giữa các roles của node tại thời điểm triển khai, Docker Engine xử lý bất kỳ chuyên môn hoá nào khi runtime. Bạn có thể triển khai cả hai loại node: managers và worker bằng Docker Engine.
* **Declarative service model:** Docker Engine sử dụng phương thức khai báo để cho phép bạn define trạng thái mong muốn của các dịch vụ khác nhau trong stack ứng dụng của bạn. VD: Bạn có thể mô tả ứng dụng bao gồm: web front-end với service message queueing và database back-end.
* **Scaling:**Đối với mỗi service bạn có thể khai báo số lượng task mà bạn muốn run. Khi bạn scale up hoặc down thì swarm manager sẽ tự động thêm hoặc xoá task để duy trì trạng thái mong muốn.
* **Desired state reconciliation:**Hãy hình dung bạn thiết lập một service run 10 replicas của một container và một worker machine (host/vps) đang giữ 2 trong số 10 replicas đó gặp sự cố bị crash, lúc này swarm manager sẽ tiến hành tạo thêm 2 replicas mới để thay để cho 2 replicas đã bị crash đó và tiến hành chuyển 2 replicas mới này cho các worker đang run.
* **Multi-host networking:**Bạn có thể chỉ định một overlay network cho các services của mình. Swarm manager sẽ tự động gán địa chỉ IP cho các container trên overlay network khi nó khởi tạo và cập nhật application.
* **Service discovery:**Swarm manager node gán mỗi service trong swarm một DNS duy nhất và bạn có thể truy vấn được thông qua DNS này.
* **Load balancing:** Có thể expose các port cho các services tới load balance để giao tiếp với bên ngoài.
* **Secure by default:**Các service giao tiếp với nhau thông qua giao thức bảo mật TLS. Bạn có thể tuỳ chỉnh sử dụng chứng chỉ ký tự root hoặc chứng chỉ từ một custom root CA.
* **Rolling updates:**Swarm giúp bạn update image của service một cách hoàn toàn tự động. Swarm manager giúp bạn kiểm soát độ trễ giữa service deploy tới các node khác nhau và bạn có thể rolling back bất cứ lúc nào.

**Kiến trúc Swarm:**

**A diagram of a company

Description automatically generated**

Bao gồm các**Manager**và các**Worker.** Người dùng có thể khai báo trạng thái mong muốn của nhiều service để chạy trong Swarm sử dụng YAML files.

**A diagram of a server

Description automatically generated**

* **Swarm:**là một cluster của một hoặc nhiều Docker Engine đang run (cụ thể ở đây là các node) trong chế độ Swarm, thay vì phải chạy các container bằng câu lệnh thì ta sẽ thiết lập các services để phân bổ các bản replicas tới các node.
* **Node:**Một node là một máy vật lý hay máy ảo đang run phiên bản Docker Engine trong chế độ Swarm. Node sẽ gồm hai loại: Manager Node và Worker Node.
* **Manager Node:**Là node nhận các define service từ user, nó quản lý và điều phối các task đến các node Worker. Theo mặc định node Manager cũng được coi là node Worker.
* **Worker Node:**là node nhận và thực thi các task từ node Manager.
* **Service:**Một service xác định image của container và số lượng các replicas (bản sao) mong muốn khởi chạy trong swarm.
* **Task:**là một tác vụ mà node worker phải thực hiện. Tác vụ này sẽ do node Manager phân bổ xuống. Một task mang một Docker Container và các lệnh để chạy bên container.

**Cách thiết lập máy ảo:**

### **1. Thiết lập máy ảo**

Bạn có thể sử dụng phần mềm tạo máy ảo như **VirtualBox**. Dưới đây là hướng dẫn chi tiết:

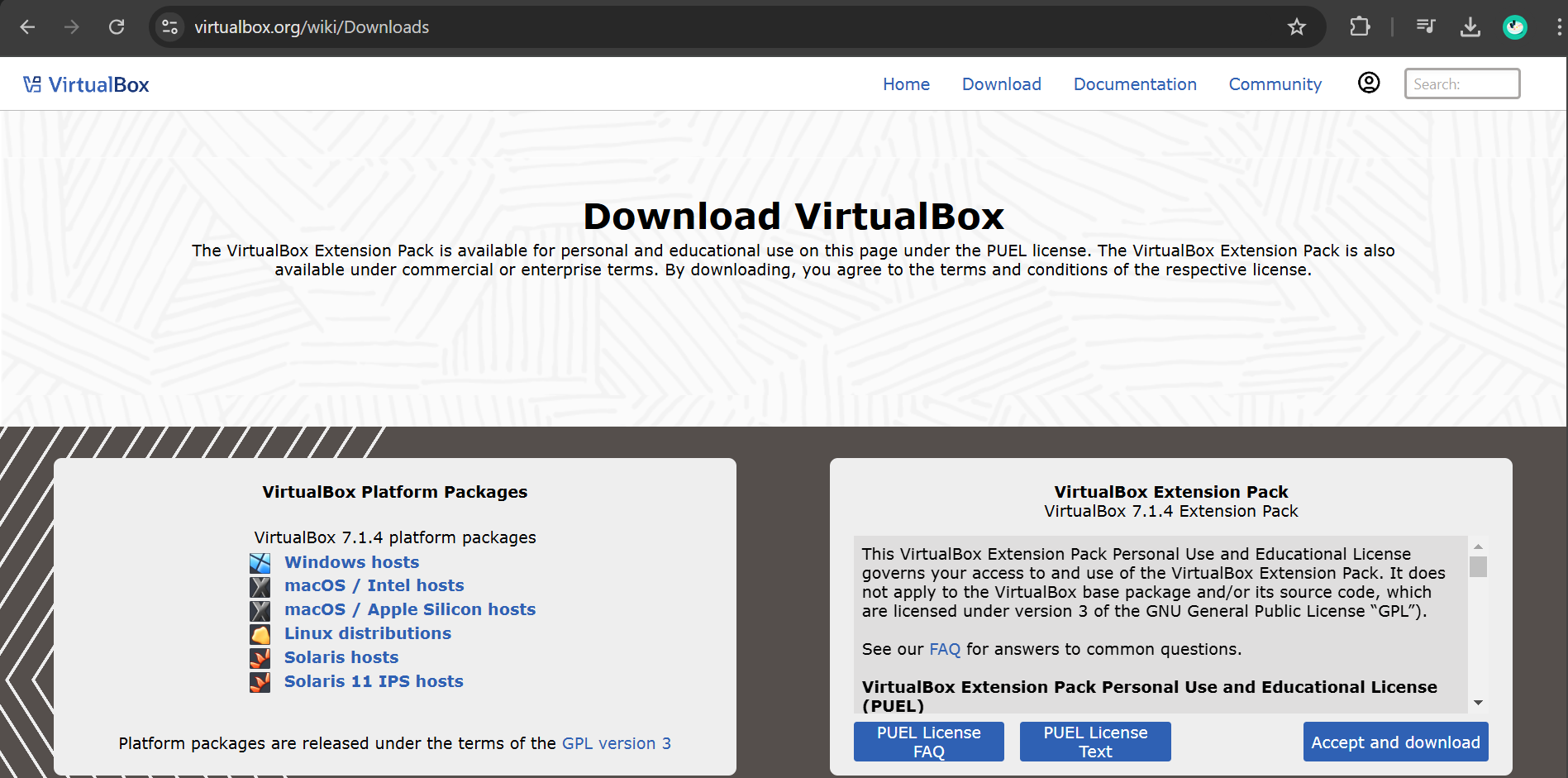
**1.1. Tải và cài đặt phần mềm máy ảo**

* **VirtualBox:**  
  Tải từ trang chính thức: <https://www.virtualbox.org>.  
  Nhấn vào Download như hình:

A screenshot of a computer

Description automatically generated

Sẽ có giao diện như sau:



Tải **Windows hosts** và chọn **Accept and download.** Sau khi tải xong sẽ được 2 file sau:



Khởi động **VirtualBox** rồi chạy **Oracle\_VirtualBox\_Extension\_Pack7.1.4.vbox-extpack** thì sẽ hiển thị hộp thoại sau và nhấn **Install:**

**A screenshot of a computer error

Description automatically generated**

**1.2. Tạo một máy ảo mới**

1. Mở VirtualBox

A screenshot of a computer

Description automatically generated

1. Chọn **New**.

A screenshot of a computer

Description automatically generated

1. Cấu hình các thông số:
   * Name: Tên máy ảo (vps1)
   * Type: Linux

A screenshot of a computer

Description automatically generated

* + Chọn Hard Disk 🡪 Use an Existing Virtual Hard Disk File

A screenshot of a computer

Description automatically generated

Nhấn vào đường link để tải tập tin máy ảo Vitual Box (VB) yêu cầu cho dự án: <https://drive.google.com/file/d/1aioXBdyeswbmLHjZR2_opZk7deaiyC1E/view?usp=sharing>

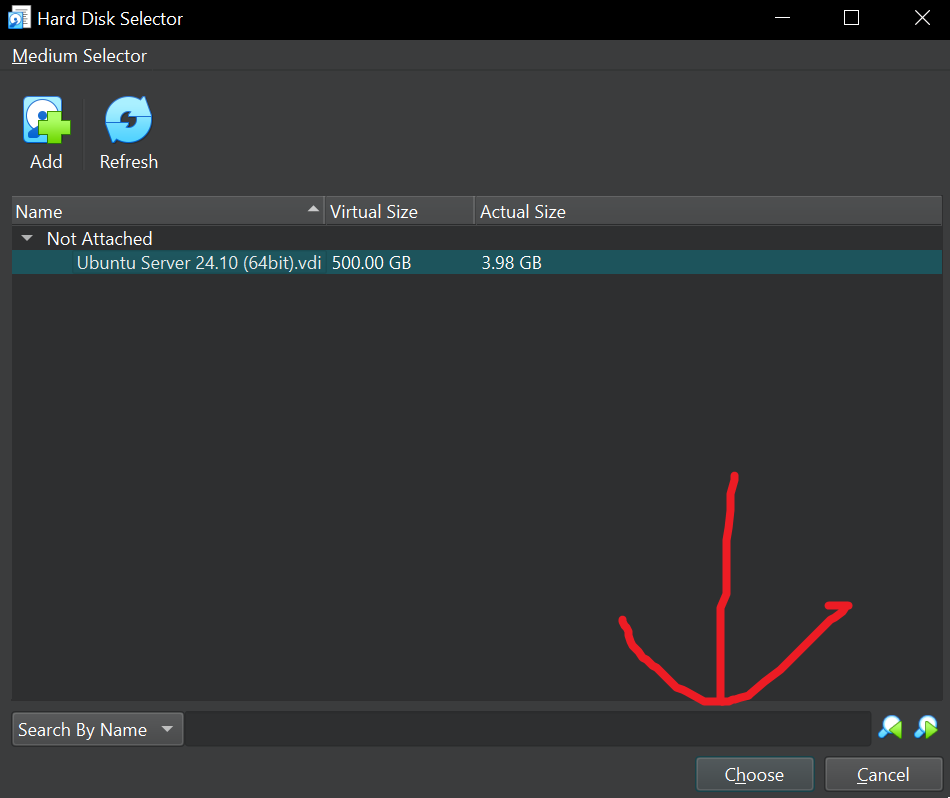
A screenshot of a computer

Description automatically generatedBấm tải xuống và giải nén file 64bit.7z  
Sau khi giải nén xong nhấn vào Add:

Chọn file vừa mới giải nén:

A screenshot of a computer program

Description automatically generated



Cuối cùng nhấn Finish:

A screenshot of a computer

Description automatically generated

Sẽ tạo ra được máy ảo và nhấn Start để khởi động máy ảo:

A screenshot of a computer

Description automatically generated

**Cài đặt Docker trên Ubuntu:**

**Bước 1: Kết nối ssh từ máy thật sang máy ảo Ubuntu Server:**

A screenshot of a computer

Description automatically generated

Start máy ảo, sau đó **Khởi động Open SSH trên máy ảo Ubuntu Server:**

**Hướng dẫn kiểm tra và cài đặt SSH trên máy ảo Ubuntu Server**

1. **Kiểm tra phiên bản SSH Client**
   * Kiểm tra phiên bản SSH Client: ssh -V
   * Nếu chưa cài SSH Client, thực hiện cài đặt: sudo apt install openssh-client
   * **Ghi chú:** SSH Client được dùng để kết nối từ máy ảo đến máy khác qua SSH.
2. **Kiểm tra phiên bản SSH Server**
   * Kiểm tra phiên bản SSH Server: sshd -V
   * Nếu chưa cài SSH Server, thực hiện cài đặt:sudo apt install openssh-server
   * **Ghi chú:** SSH Server dùng để mở cổng, cho phép các máy khác kết nối đến máy ảo.
3. **Thao tác với dịch vụ SSH**
   * Kích hoạt SSH: sudo systemctl enable ssh
   * Tắt SSH: sudo systemctl disable ssh
   * Kiểm tra trạng thái SSH: sudo systemctl status ssh
   * Khởi chạy SSH Server: sudo systemctl start ssh
   * Dừng SSH Server: sudo systemctl stop ssh

Sau khi cài đặt và kích hoạt dịch vụ SSH Server, đảm bảo rằng máy thật có thể kết nối tới máy ảo bằng cách:

* Kiểm tra địa chỉ IP của máy ảo: ifconfig -a
  + Thì địa chỉ IP (inet): 192.168.102.101

**A screen shot of a computer

Description automatically generated**

**Bước 2: Kết nối ssh đến máy ảo Ubuntu Server từ máy thật Windows**

* Mở command prompt dưới quyền Administrator
* Kết nối ssh từ máy thật sang máy ảo Ubuntu Server: ssh osboxes@<ubuntu server ip>

**Bước 3: Tiến hành cài đặt Docker trên Ubuntu**

1. **Cập nhật hệ thống server**: sudo apt-get update

2. **Cài đặt chứng chỉ:** sudo apt-get install ca-certificates curl gnupg lsb-release

3. **Cài đặt chứng chỉ ssl:** sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

4. **Thiết lập cấu hình:** echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

5. **Cập nhật hệ thống cho các gói cấu hình mới:** sudo apt-get update

6**. Cài đặt Docker CE:** sudo apt-get install docker-ce docker-ce-cli containerd.io

**7. Kiểm tra phiên bản Docker CE:** sudo docker --version

A black background with white text

Description automatically generated

**8. Kiểm tra xem có các image:** sudo docker image ls

