**TÊN ĐỀ TÀI: TÌM HIỂU VÀ TRIỂN KHAI PORTAINER DOCKER**

**MÃ NHÓM: 175A071167-175A071398**

1. **Danh sách thành viên & Công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ & tên** | **MSSV** | **Công việc** | **Tiến độ** |
| Bùi Văn Đức | 175A071398 | * Tìm hiểu và cài đặt Docker * Cài đặt và sử dụng Portainer Docker. | 60% |
| Nguyễn văn Hữu | 175A071167 | * Cài đặt docker và sử dụng cơ bản. * Tìm hiểu về Portainer Docker | 60% |

1. **Nội dung nghiên cứu**

**2.1. Giới thiệu**

Trước khi hiểu Portainer Docker là gì chúng ta phải hiểu khái quát về Docker.

* + 1. **Docker**
  1. **Khái niệm**
* Theo nguồn wikipedia: "Docker là một dự án mã nguồn mở giúp tự động triển khai các ứng dụng Linux và Windows vào trong các container ảo hóa.".
* Theo các trang mạng: "Docker là một open platform cung cấp cho người sử dụng những công cụ và service để người sử dụng có thể đóng gói và chạy chương trình của mình trên các môi trường khác nhau một cách nhanh nhất."
  1. **Lợi ích của Docker**
* Với Docker, chúng ta có thể đóng gói mọi ứng dụng vd như webapp, backend, MySQL, BigData…thành các containers và có thể chạy ở “hầu hết” các môi trường vd như Linux, Mac, Window…
* Docker Containers có một API cho phép quản trị các container từ bên ngoài. Giúp cho chúng ta có thể dễ dàng quản lí, thay đổi, chỉnh sửa các container.
* Hầu hết các ứng dụng Linux có thể chạy với Docker Containers.
* Docker Containers có tốc độ chạy nhanh hơn hẳn các VMs truyền thống (theo kiểu Hypervisor). Điều này là một ưu điểm nổi bật nhất của Docker.
  1. **Một số khái niệm trong docker**
* **Docker Client:** là cách mà bạn tương tác với docker thông qua command trong terminal. Docker Client sẽ sử dụng API gửi lệnh tới Docker Daemon.
* **Docker Daemon:** là server Docker cho yêu cầu từ Docker API. Nó quản lý images, containers, networks và volume.
* **Docker Volumes:** là cách tốt nhất để lưu trữ dữ liệu liên tục cho việc sử dụng và tạo apps.
* **Docker Registry:** là nơi lưu trữ riêng của Docker Images. Images được push vào registry và client sẽ pull images từ registry. Có thể sử dụng registry của riêng bạn hoặc registry của nhà cung cấp như : AWS, Google Cloud, Microsoft Azure.
* **Docker Hub:** là Registry lớn nhất của Docker Images ( mặc định). Có thể tìm thấy images và lưu trữ images của riêng bạn trên Docker Hub ( miễn phí).
* **Docker Repository:** là tập hợp các Docker Images cùng tên nhưng khác tags. VD: golang:1.11-alpine.
* **Docker Networking:** cho phép kết nối các container lại với nhau. Kết nối này có thể trên 1 host hoặc nhiều host.
* **Docker Compose:** là công cụ cho phép run app với nhiều Docker containers 1 cách dễ dàng hơn. Docker Compose cho phép bạn config các command trong file *docker-compose.yml* để sử dụng lại. Có sẵn khi cài Docker.
* **Docker Swarm:** để phối hợp triển khai container.
* **Docker Services:** là các containers trong production. 1 service chỉ run 1 image nhưng nó mã hoá cách thức để run image — sử dụng port nào, bao nhiêu bản sao container run để service có hiệu năng cần thiết và ngay lập tức.
* **Docker Machine**:tạo ra các Docker engine trên máy chủ.
* **Docker image**: một dạng tập hợp các tệp của ứng dụng, được tạo ra bởi Docker engine. Nội dung của các Docker image sẽ không bị thay đổi khi di chuyển. Docker image được dùng để chạy các Docker container.
* **Docker Container**: một dạng runtime của các Docker image, dùng để làm môi trường chạy ứng dụng.
* **Dockerfile**: là file config cho Docker để build ra image. Nó dùng một image cơ bản để xây dựng lớp image ban đầu. Một số image cơ bản: python, unbutu and alpine. Sau đó nếu có các lớp bổ sung thì nó được xếp chồng lên lớp cơ bản. Cuối cùng một lớp mỏng có thể được xếp chồng lên nhau trên các lớp khác trước đó.
  + 1. **Portainer Docker**
  1. **Khái niệm**

Là một công cụ dùng để quản lý các app Docker trên Local bằng UI. Portainer giải quyết cho chúng ta vấn đề về việc quản lý các docker apps và thao tác chúng được thuận tiện hơn trên giao diện UI.

* 1. **Ưu điểm của Portainer Docker**
* Siêu nhẹ tốn chừng 10MB RAM
* Các thao tác quản lý Container / Images / Volumes / Networks rất thuận tiện.
  1. **Cài đặt Portainer Docker**

**Bước 1 Cài đặt Docker và Docker Compose**

Docker có hai phiên bản:   **Phiên bản doanh nghiệp (EE)**  và  **Phiên bản cộng đồng (CE)**. Trong bài này sẽ hướng dẫn cài đặt docker bản Docker CE

Đầu tiên là cập nhật apt-get bằng câu lệnh sau

$ sudo apt update

Tiếp theo, chạy câu lệm bên dưới để tải khóa chính thức GPG của Docker

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

$ sudo apt-key fingerprint 0EBFCD88

Kết quả như sau:

Output:

**pub rsa4096 2017-02-22 [SCEA]**

**9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88**

**uid [ unknown] Docker Release (CE deb) <docker@docker.com>**

**sub rsa4096 2017-02-22 [S]**

Câu lệnh bên dưới để thêm kho lưu trữ **ổn định** của nó  vào Ubuntu, Để thêm  kho lưu trữ **hàng đêm** hoặc  **kiểm tra** , hãy thêm từ **nightly** hoặc  **test** (hoặc cả hai) sau khi từ  **stable**trong các lệnh bên dưới

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

Để xóa các phiên bản docker cũ, sử dụng câu lệnh sau:

$ sudo apt-get remove docker docker-engine docker.io containerd runc

Để hiện danh sách các phiên bản docker-ce, sử dụng các lệnh sau:

**$ apt-cache madison docker-ce**

Output:

**docker-ce | 5:19.03.8~3-0~ubuntu-bionic | https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages**

**docker-ce | 5:19.03.7~3-0~ubuntu-bionic | https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages**

**docker-ce | 5:19.03.6~3-0~ubuntu-bionic | https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages**

**docker-ce | 5:19.03.5~3-0~ubuntu-bionic | https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages**

**....**

Để tự động tải phiên bản mới nhất, chúng ta chỉ cần sử dụng lệnh sau:

$ sudo apt update

$ sudo apt-get install docker-ce docker-ce-cli containerd.io

Thêm user vào nhóm Docker và khởi động lại

$ sudo usermod -aG docker $USER

$ sudo reboot

Kiểm tra version của docker, chúng ta sử dụng câu lệnh sau:

**$ docker version**

Output :

**Client:**

**Version: 18.09.5**

**API version: 1.39**

**Go version: go1.10.8**

**Git commit: e8ff056**

**Built: Thu Apr 11 04:43:57 2019**

**OS/Arch: linux/amd64**

**Experimental: false**

Chạy image hello-world

**$ sudo docker run hello-world**

Output:

**Hello from Docker!**

**This message shows that your installation appears to be working correctly.**

**Tải Docker Compose**

Trên Ubuntu Linux, bạn có thể tải xuống bản nhị phân Docker Compose từ đường dẫn  <https://github.com/docker/compose/releases>Dưới đây là câu lệnh tải Docker Compose với bản 1.24.0

$ sudo curl -L "https://github.com/docker/compose/releases/download/**1.24.0**/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

Sau khi tải xuống, chúng ta cần cấp quyền thực thi cho tệp tin nhị phân và liên kết đến **/usr/binary**

$ sudo chmod +x /usr/local/bin/docker-compose

$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

Đề kiểm tra phiên bản của docker-compose

**$ docker-compose –version**

Output:

**docker-compose version 1.24.0, build 0aa59064**

**Bước 2: Cài đặt Portainer**

**$ sudo docker run -d -p 8000:8000 -p 9000:9000 --name=portainer --restart=always -v /var/run/docker**

**.sock:/var/run/docker.sock -v portainer\_data:/data portainer/portainer**

Output:

**Unable to find image 'portainer/portainer:latest' locally**

**latest: Pulling from portainer/portainer**

**d1e017099d17: Pull complete**

**a7dca5b5a9e8: Pull complete**

**Digest: sha256:4ae7f14330b56ffc8728e63d355bc4bc7381417fa45ba0597e5dd32682901080**

**Status: Downloaded newer image for portainer/portainer:latest**

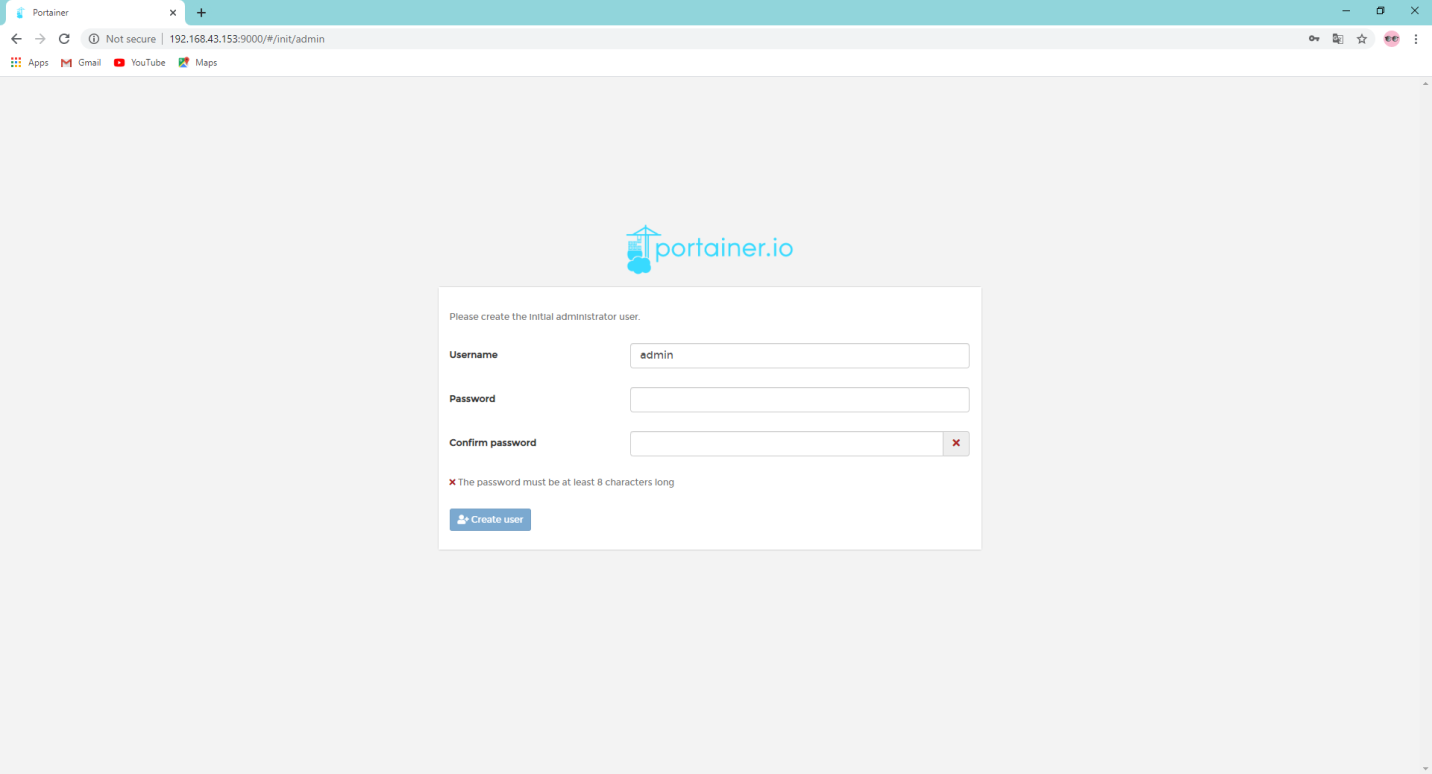
**2b2221cde5a6da53c375026ffea183596bc5d532bfdd02ffc58dcaa9bc94b93d**

$ cd ~/

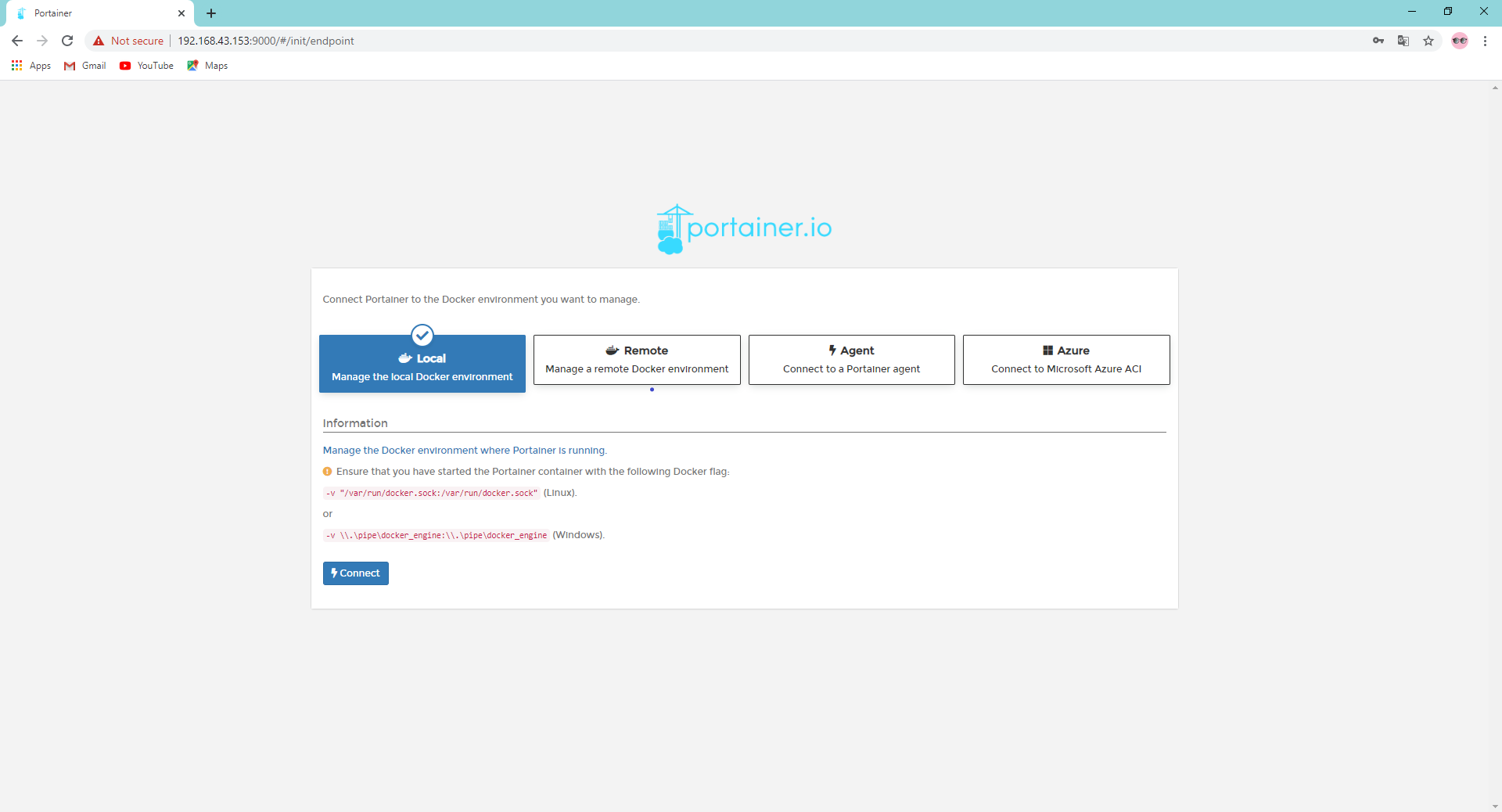
$ docker volume create portainer\_data

**Bước 3: Truy cập Portainer Portal**

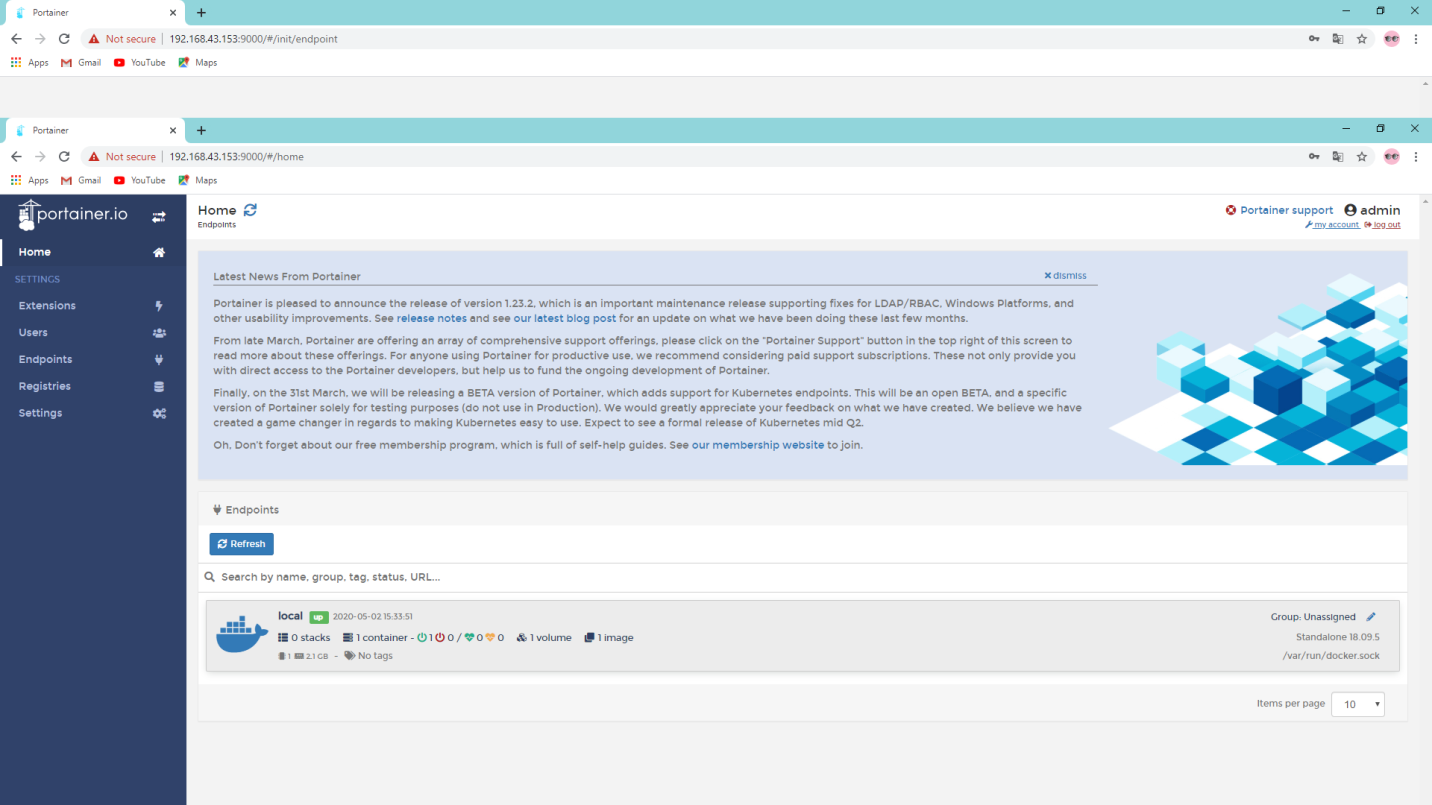
Truy cập với cổng #9000: http://localhost:9000(nếu trên Server truy cập với đường dẫn <ip của server>:9000)



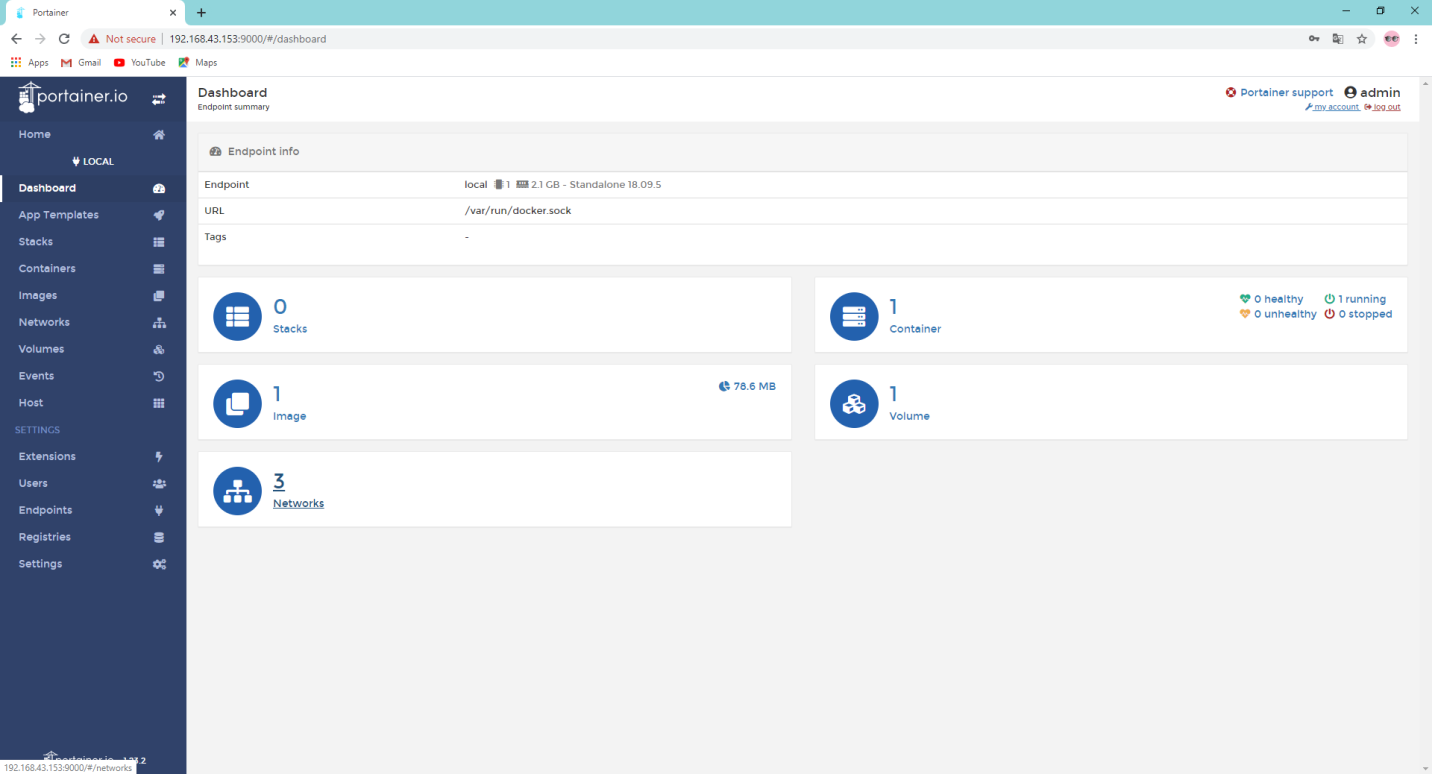
*Tạo mật khẩu cho admin*

**

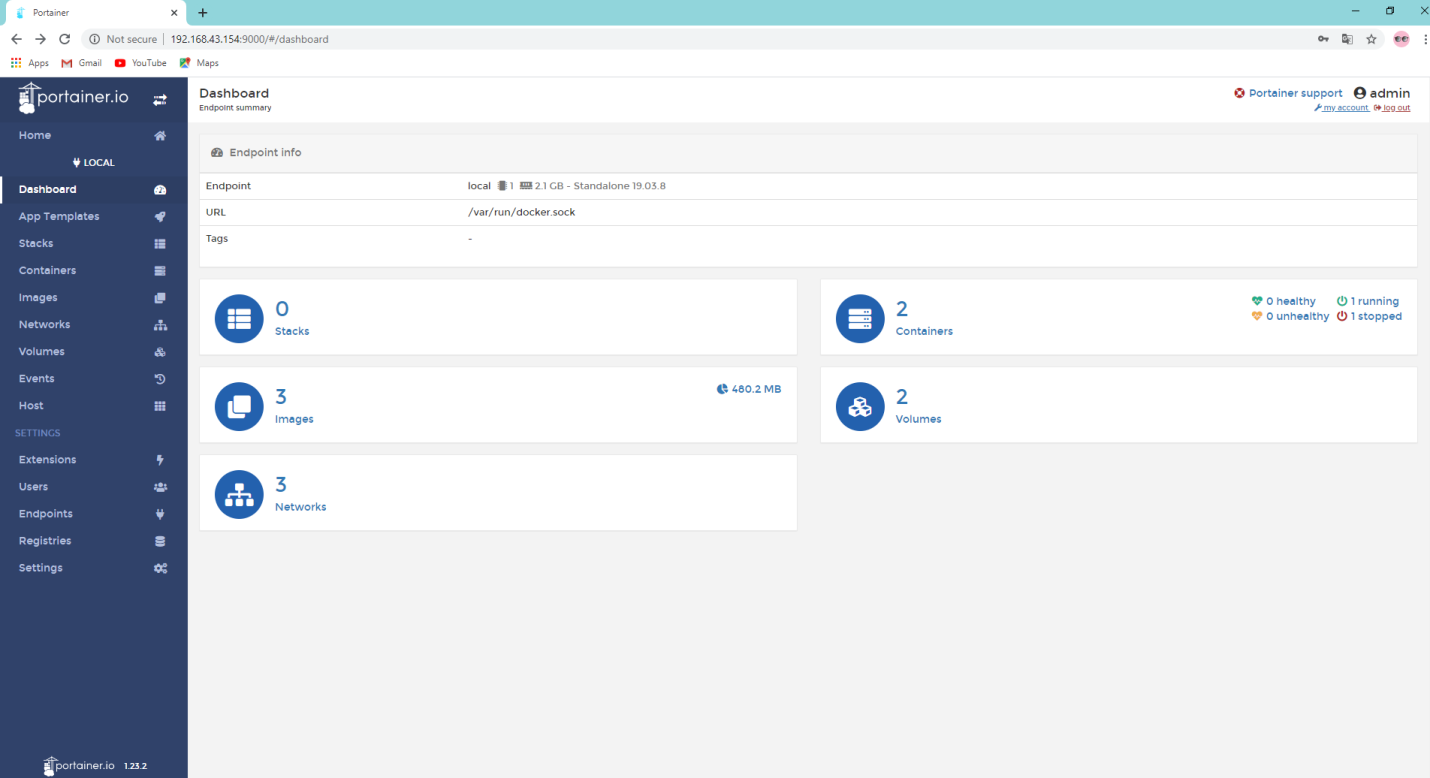
*Chọn môi trường Docker muốn quản lý*



*Trang chủ*

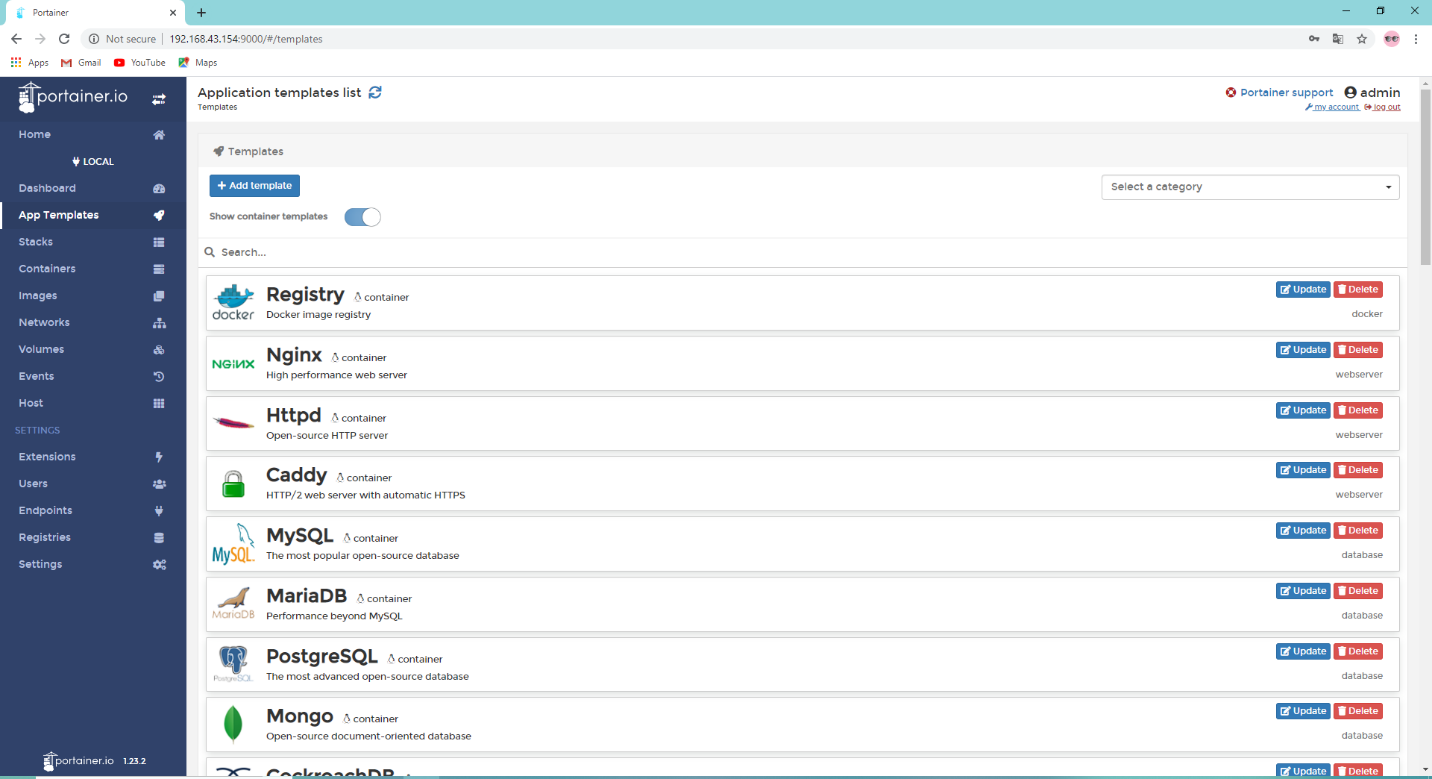


* 1. **Quản trị dự án Docker với Portainer**
     1. **Giới thiệu thanh công cụ**
  2. **Dashboard**



Với Dashboard là bảng điều khiển của dự án. Tại đây, chúng ta có thể xem tổng quát xem docker có bao nhiêu Stacks, Containers, Images, Volumes, Network. Cũng có thể xem có bao nhiêu container đang running hay stop.

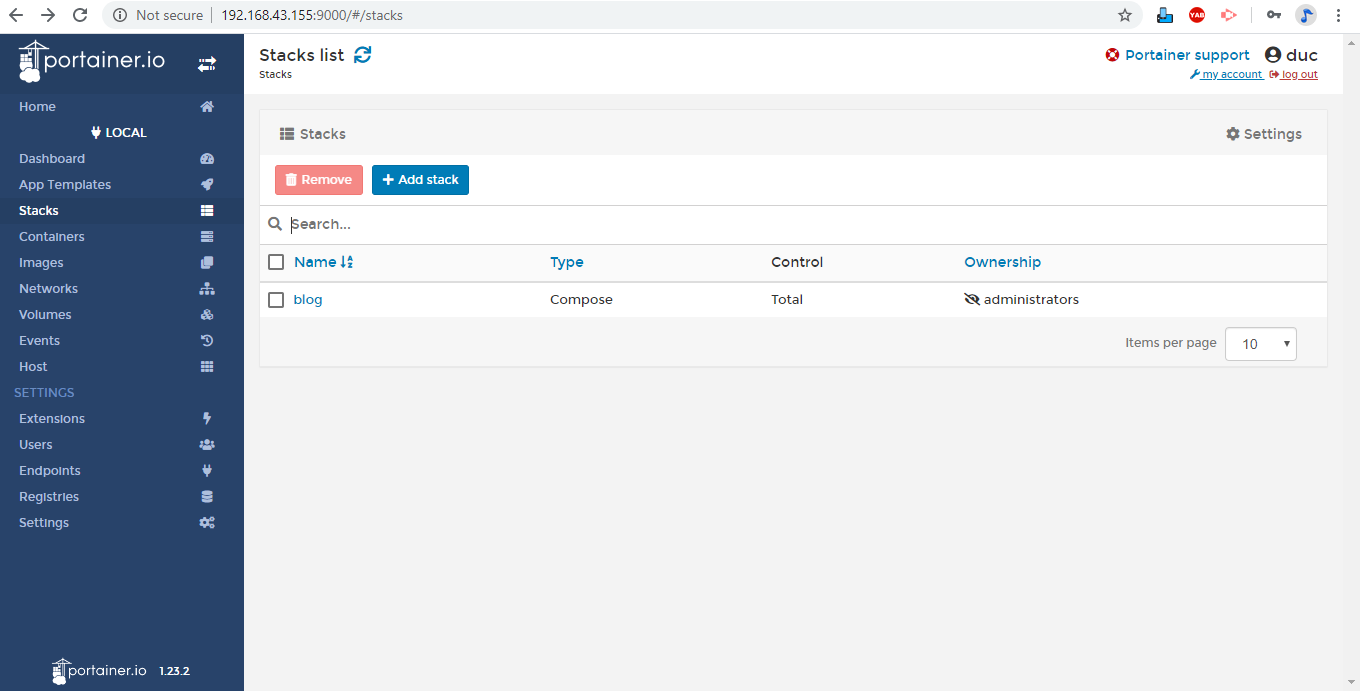
* 1. **App Templates**



**Tại đây ta có thể tạo các templates có sẵn trong Portainer Docker**

* 1. **Stack**

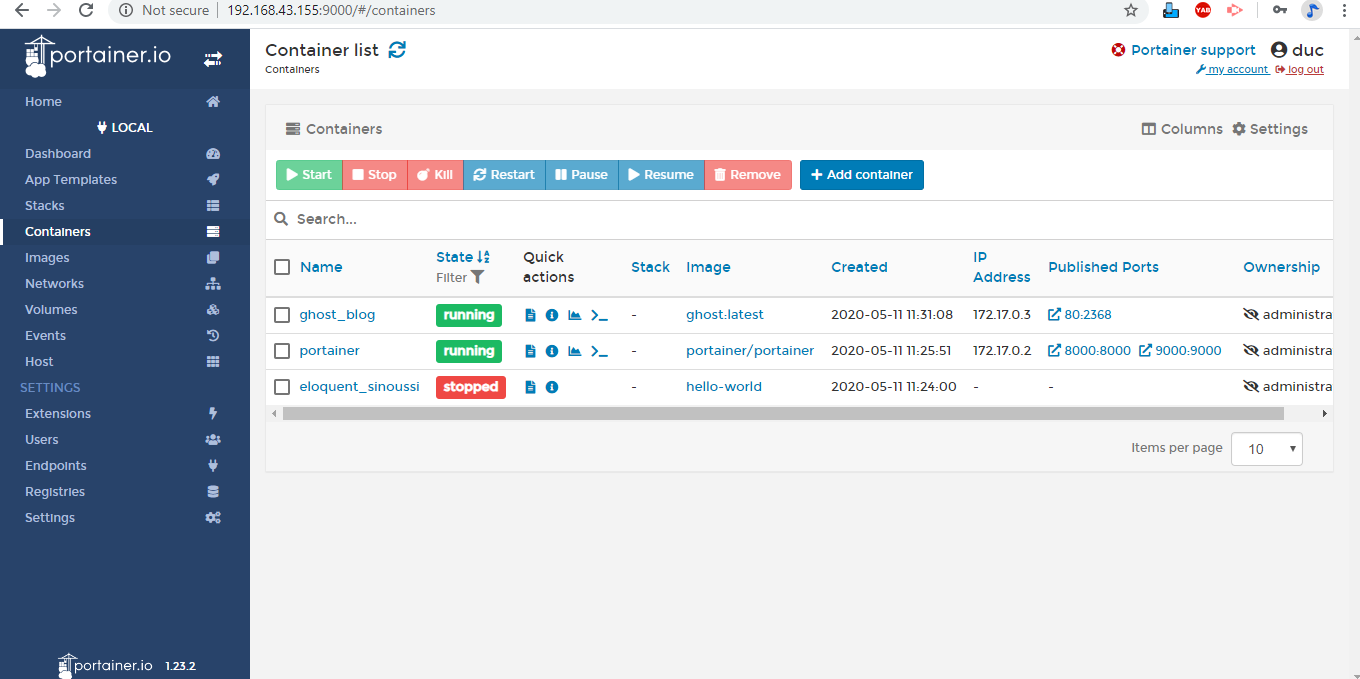
Và đây là stacks sau khi chúng ta đã chạy docker-compose.yml bằng command hay deploy trên dashboard



* 1. **Container**

sau khi chạy xong docker-compose thì hiện tại stacks của mình gồm các containers

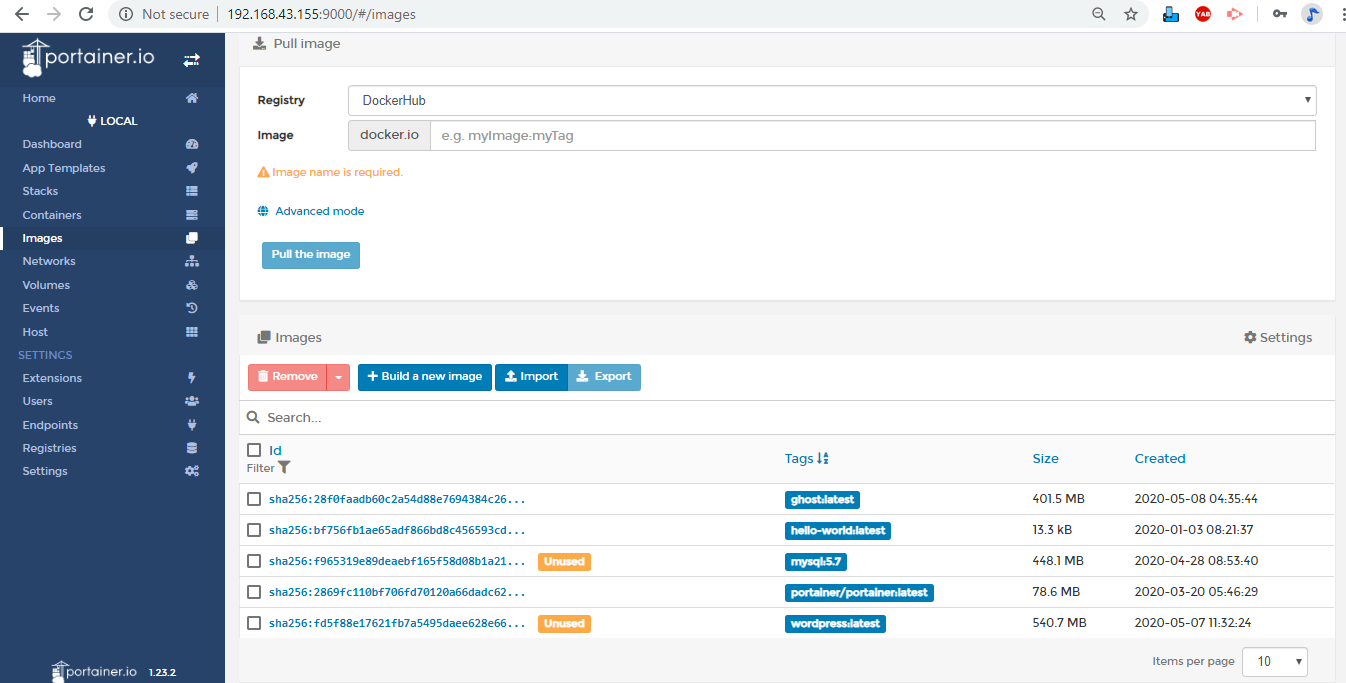
như bên dưới



Bây giờ các bạn có thể start, stop, hay thâm chí remove các container mà không phải gõ command line nữa.

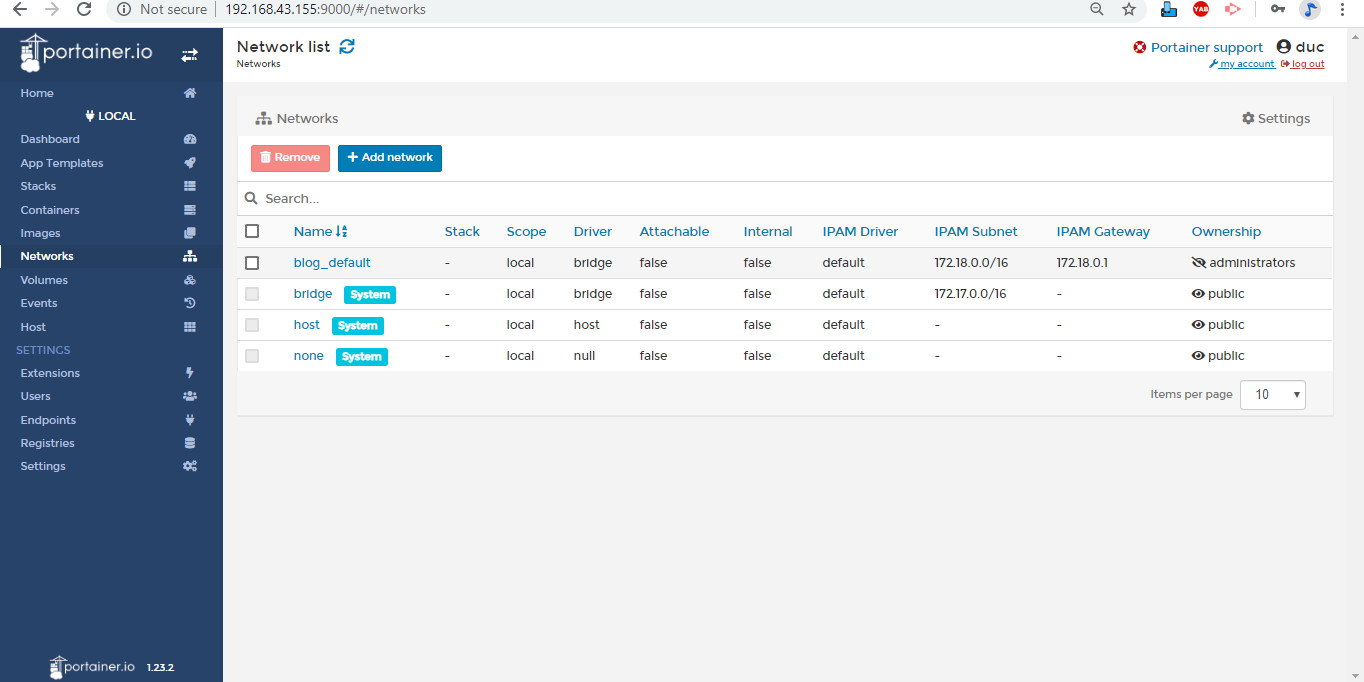
* 1. Image

Chúng ta có thể thao tác tất cả các công cụ them, build, xóa của images



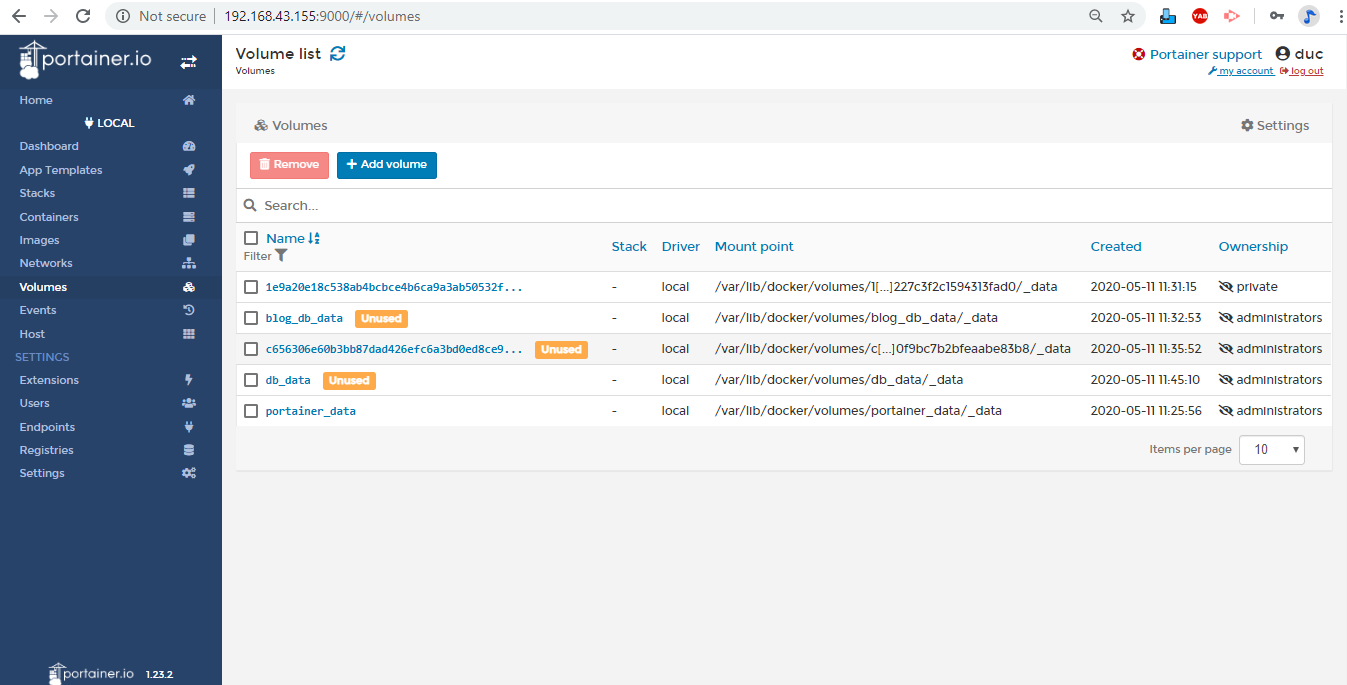
* 1. Networks

Quản trị mạng



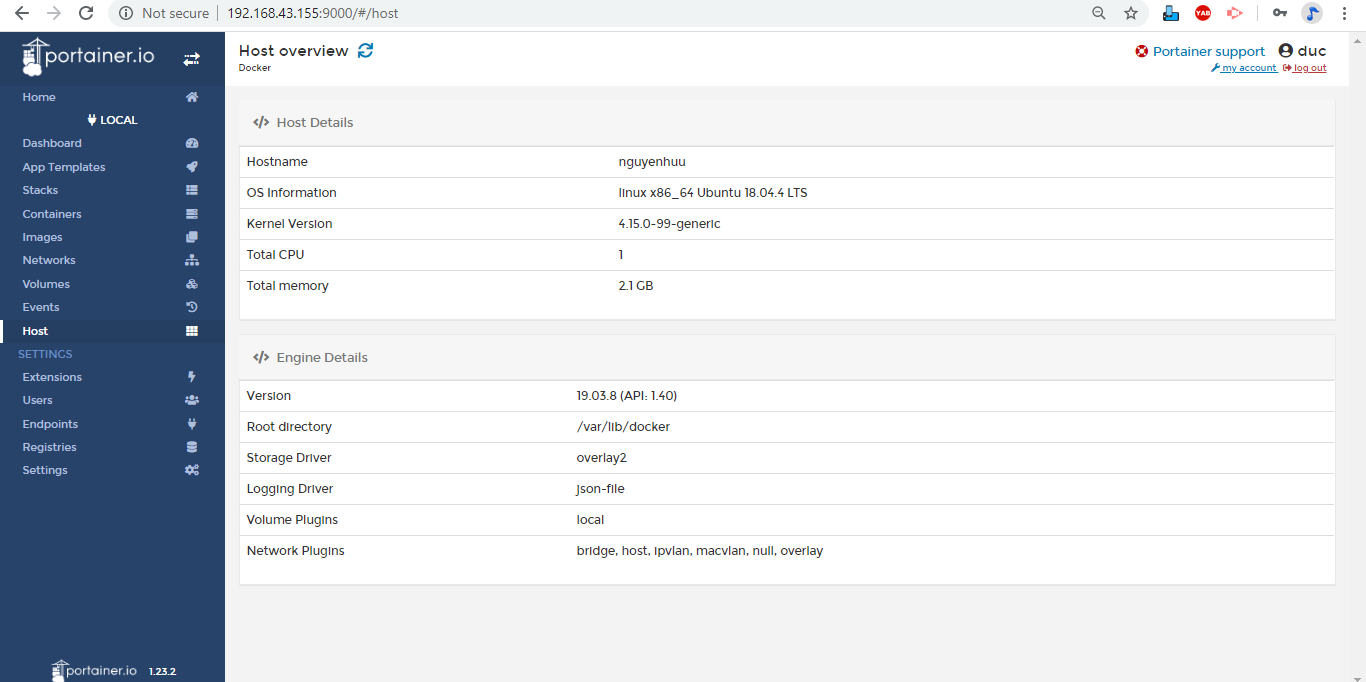
* 1. Volumes

Làm việc với docker Data



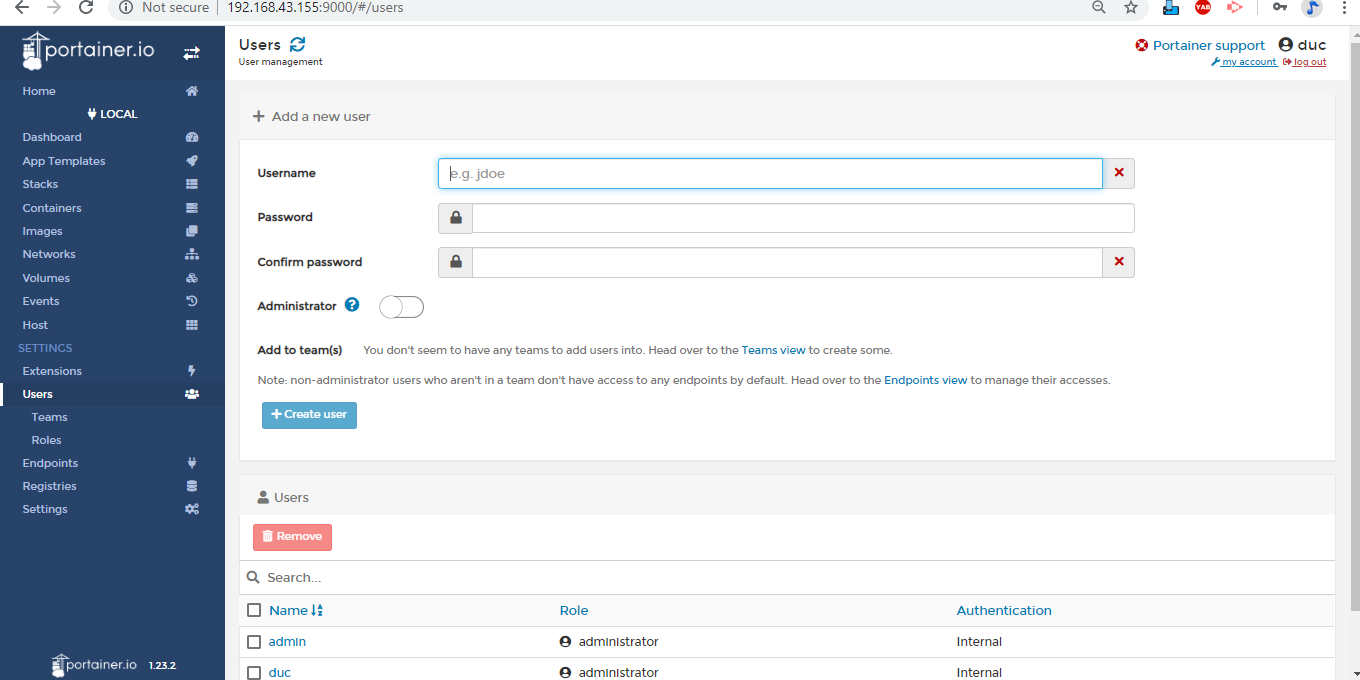
* 1. Host

Giới thiệu thong tin chi tiết máy Host

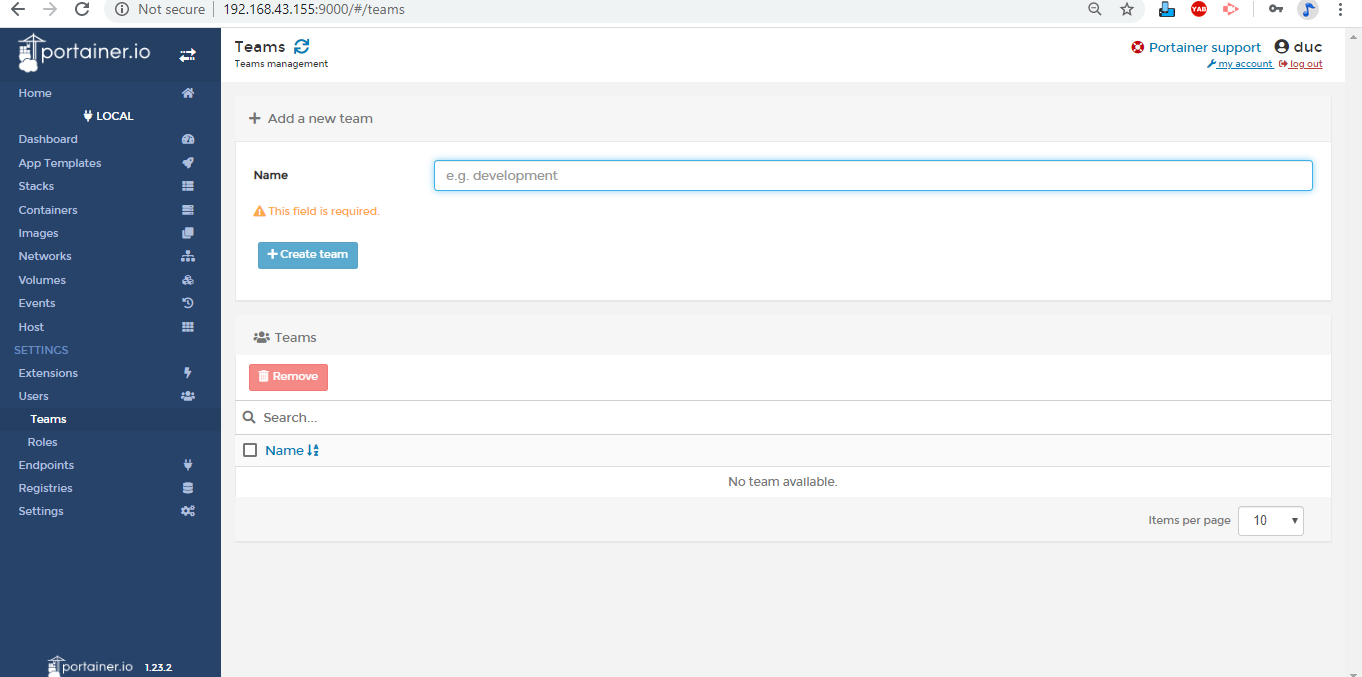


* 1. Users

a)Quản lý users: thêm user, xóa user



b)Phân nhóm user



Phân quyền user

Endpoint administrator: Kiểm soát hoàn toàn tất cả các tài nguyên

Helpdesk : Truy cập chỉ đọc của tất cả các tài nguyên

Read-only user : Truy cập chỉ đọc các tài nguyên được chỉ định

Standard user: Kiểm soát hoàn toàn các tài nguyên

