

INSTALLATION ET UTILISATION « DOCKER COMPOSE »

```
root@osboxes:~# apt-get install curl
root@osboxes:~# curl -L
"https://github.com/docker/compose/releases/download/1.9.0/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
root@osboxes:~# chmod +x /usr/local/bin/docker-compose
root@osboxes:~# docker-compose --version
docker-compose version 1.9.0, build 2585387
root@osboxes:~# nano docker-compose.yml
```

```
version: '2'

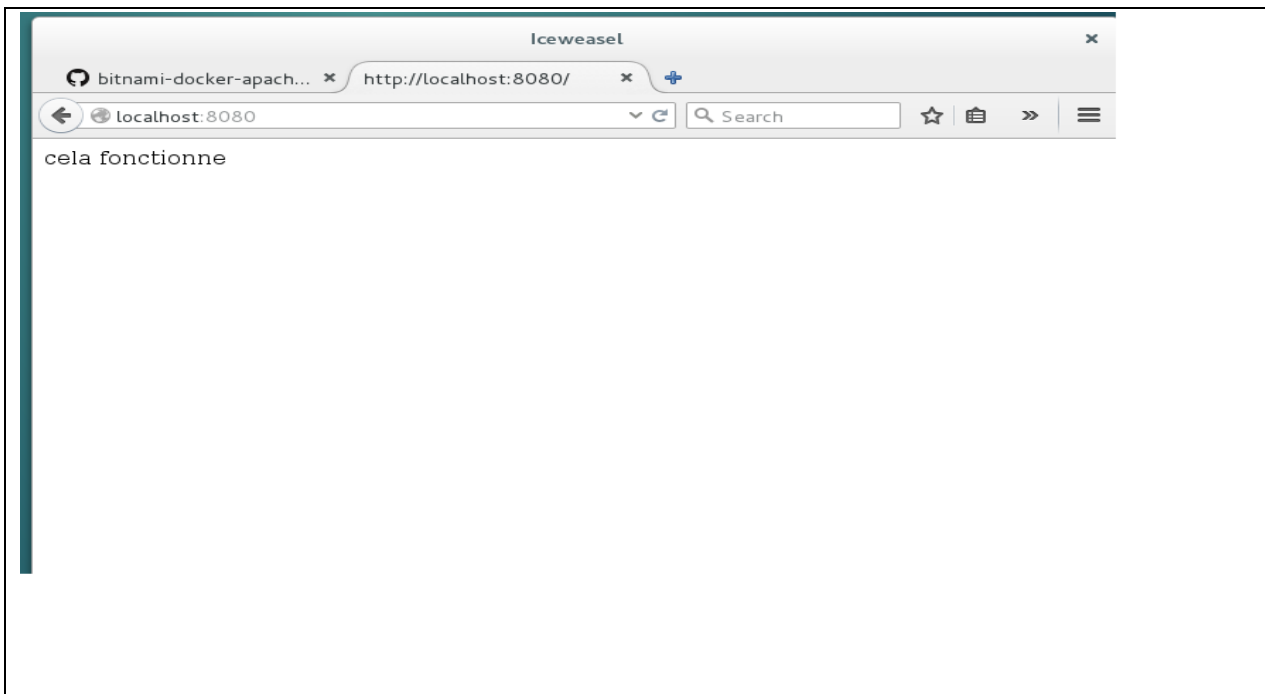
services:
  apache:
    image: 'grassa/apache:latest'
    ports :
      - "8080:80"
    volumes:
      - '/root/Docker/Apache/website:/var/www/html/'
```

```
root@osboxes:~# docker-compose up
Creating network "root_default" with the default driver
Creating root_apache_1
Attaching to root_apache_1
apache_1 | AH00558: apache2: Could not reliably determine the server's
fully qualified domain name, using 172.18.0.2. Set the 'ServerName'
directive globally to suppress this message
```

Taper **Ctrl-c** pour arrêter

```
root@osboxes:~# docker-compose up -d
Recreating root_apache_1
root@osboxes:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	STATUS	PORTS	NAMES
afa9d6180aa7	grassa/apache:latest	"/usr/sbin/apache2 -D"	Up 11 seconds	0.0.0.0:8080->80/tcp	root_apache_1



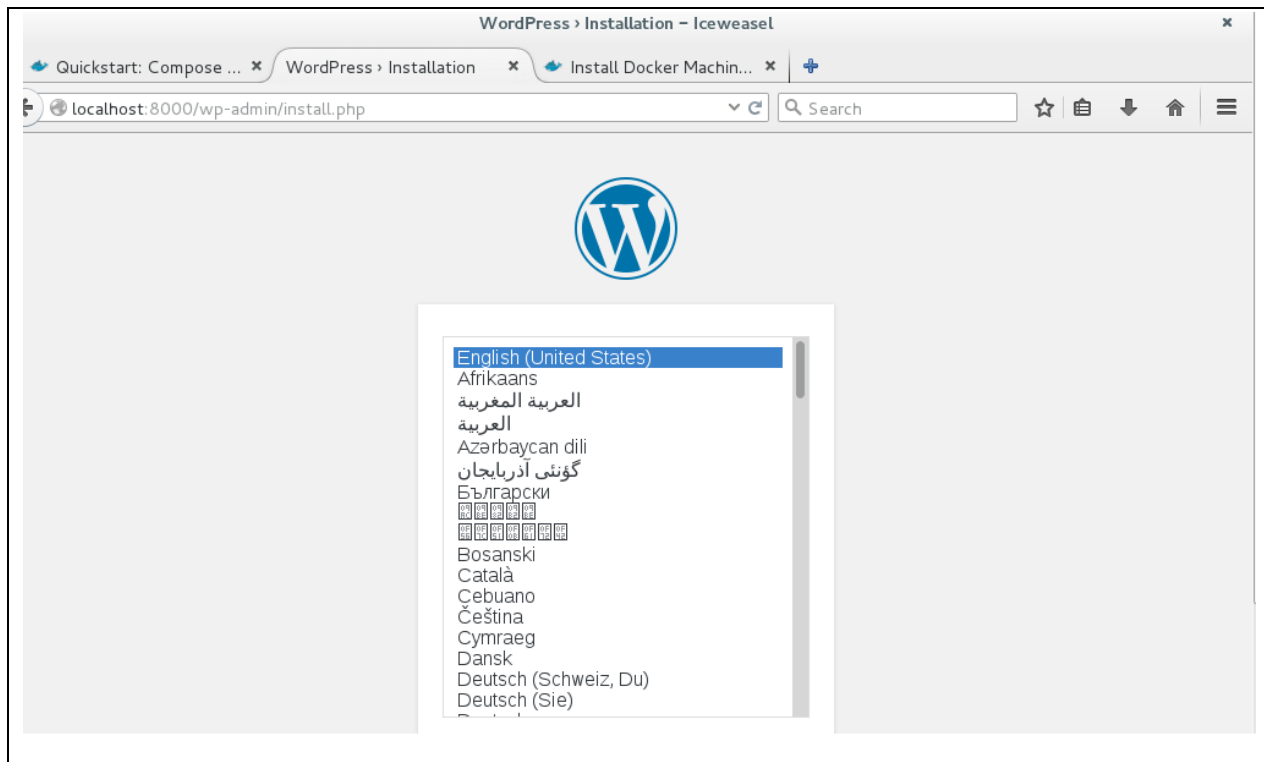
```
root@osboxes:~# mkdir wordpress
root@osboxes:~# nano wordpress/docker-compose.yml
```

```
version: '2'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: wordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_PASSWORD: wordpress
volumes:
  db_data:
```

```
root@osboxes:~# cd wordpress
root@osboxes:~# docker-compose up -d
```



INSTALLATION D'UN ORCHESTRATEUR POUR DOCKER « RANCHER »

Afin d'installer Rancher sur notre serveur nous avons quelques prérequis à satisfaire. Rancher est composé de trois grandes parties :

- Installation de Docker
- Un **serveur maître** qui dispose d'un *frontend Web*, un serveur dédié à la tâche, une machine virtuelle, ou toute autre chose.
- Un **serveur slave** à faire tourner les conteneurs docker/rancher qui doit impérativement pouvoir dialoguer avec le serveur maître et disposé également au minimum de 1 Go de RAM .

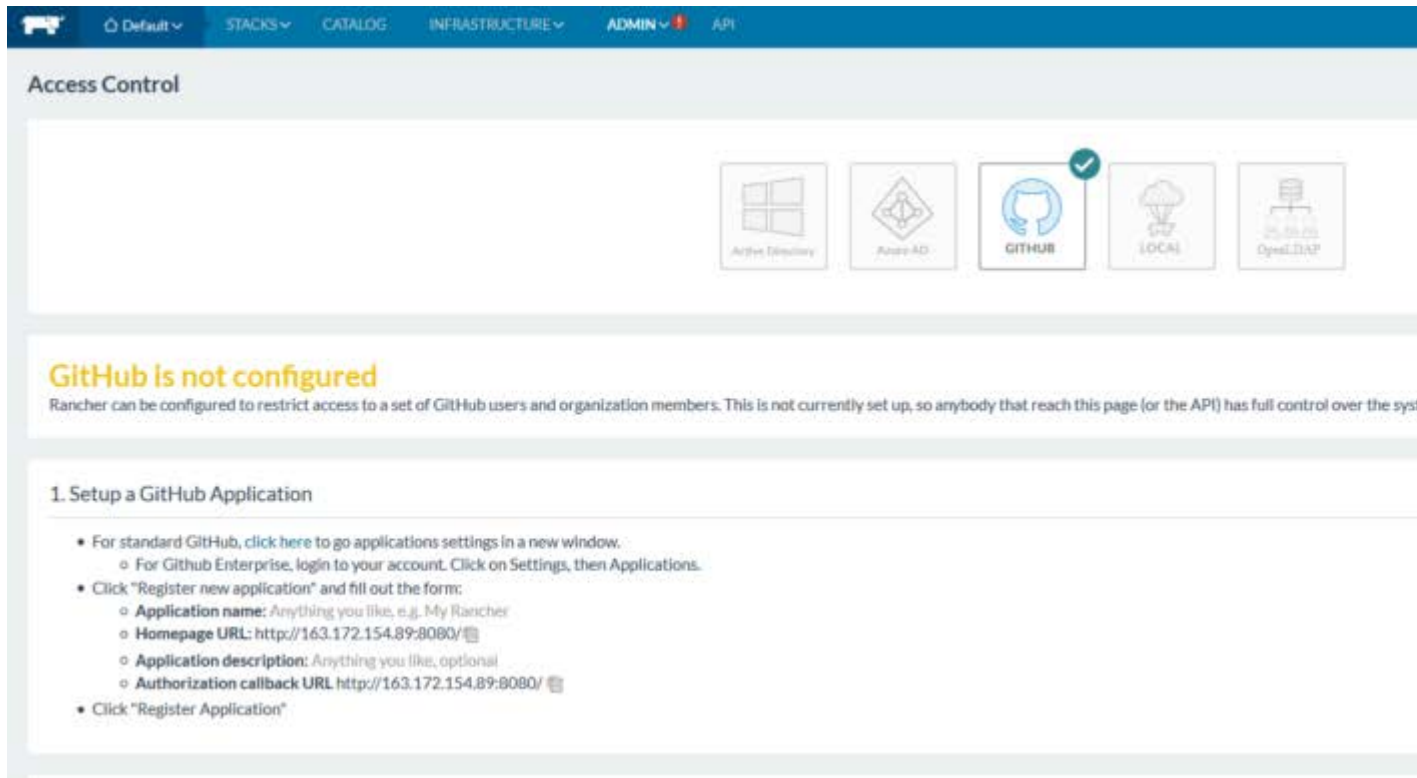
Installation serveur Maitre :

```
1. docker run -d --restart=always -p 8080:8080 rancher/server
2. docker ps
#récupération du contenerid afin de vérifier le déroulement de l'installation
3. docker logs -f containerid (ex : a93e2157b7b8)
```

Un peu de patience rancher s'installe tout seul mais il lui faut un peu de temps. Une fois que les opérations sont terminées rancher est disponible à l'adresse suivante http://ip_de_votre_serveur:8080

Lors de notre première visite, Rancher nous demande de rajouter un nouvel hôte, ignorer cette étape, dans un premier temps nous allons configurer l'authentification.

Pour ce faire, Allez dans « Admin » puis « Access Control ». La page affichée devrait ressembler à celle-ci



The screenshot shows the Rancher 'Access Control' page. At the top, there is a navigation bar with links for Default, STACKS, CATALOG, INFRASTRUCTURE, ADMIN (with a red notification icon), and API. Below the navigation bar, the 'Access Control' section displays five authentication providers: Active Directory, Azure AD, GITHUB, LOCAL, and OpenLDAP. The GITHUB provider is highlighted with a green checkmark, indicating it is the selected or active provider. Below this, a yellow banner states 'GitHub is not configured' with a warning icon. The text below the banner reads: 'Rancher can be configured to restrict access to a set of GitHub users and organization members. This is not currently set up, so anybody that reach this page (or the API) has full control over the system.' Below the banner, the section '1. Setup a GitHub Application' provides instructions for setting up a GitHub application, including links to GitHub settings and a list of required fields: Application name, Homepage URL, Application description, and Authorization callback URL.

Access Control

Active Directory Azure AD GITHUB LOCAL OpenLDAP

GitHub is not configured

Rancher can be configured to restrict access to a set of GitHub users and organization members. This is not currently set up, so anybody that reach this page (or the API) has full control over the system.

1. Setup a GitHub Application

- For standard GitHub, [click here](#) to go applications settings in a new window.
 - For Github Enterprise, login to your account. Click on Settings, then Applications.
- Click "Register new application" and fill out the form:
 - **Application name:** Anything you like, e.g. My Rancher
 - **Homepage URL:** <http://163.172.154.89:8080/>
 - **Application description:** Anything you like, optional
 - **Authorization callback URL** <http://163.172.154.89:8080/>
- Click "Register Application"

Sélectionner **Local** pour activer l'authentification grâce à PAM. Puis renseigner un compte utilisateur et un mot de passe (ex : admin/system)

Access Control

Active Directory Azure AD GITHUB LOCAL OpenID.DAP

Local Authentication is **not configured**

Rancher can be configured to restrict access to a set of accounts defined in the Rancher database. This is not currently set up, so anybody that reach this page (or the API) has full control over the system.

1. Setup an Admin user

This user will become the admin that has full control over Rancher.

Login Username* admin

Password* *****

Full Name admin

Confirm Password* *****

2. Enable Access Control

Click to enable access control and log in.

Enable Local Auth

Nous voici désormais avec un serveur Rancher prêt à l'usage.

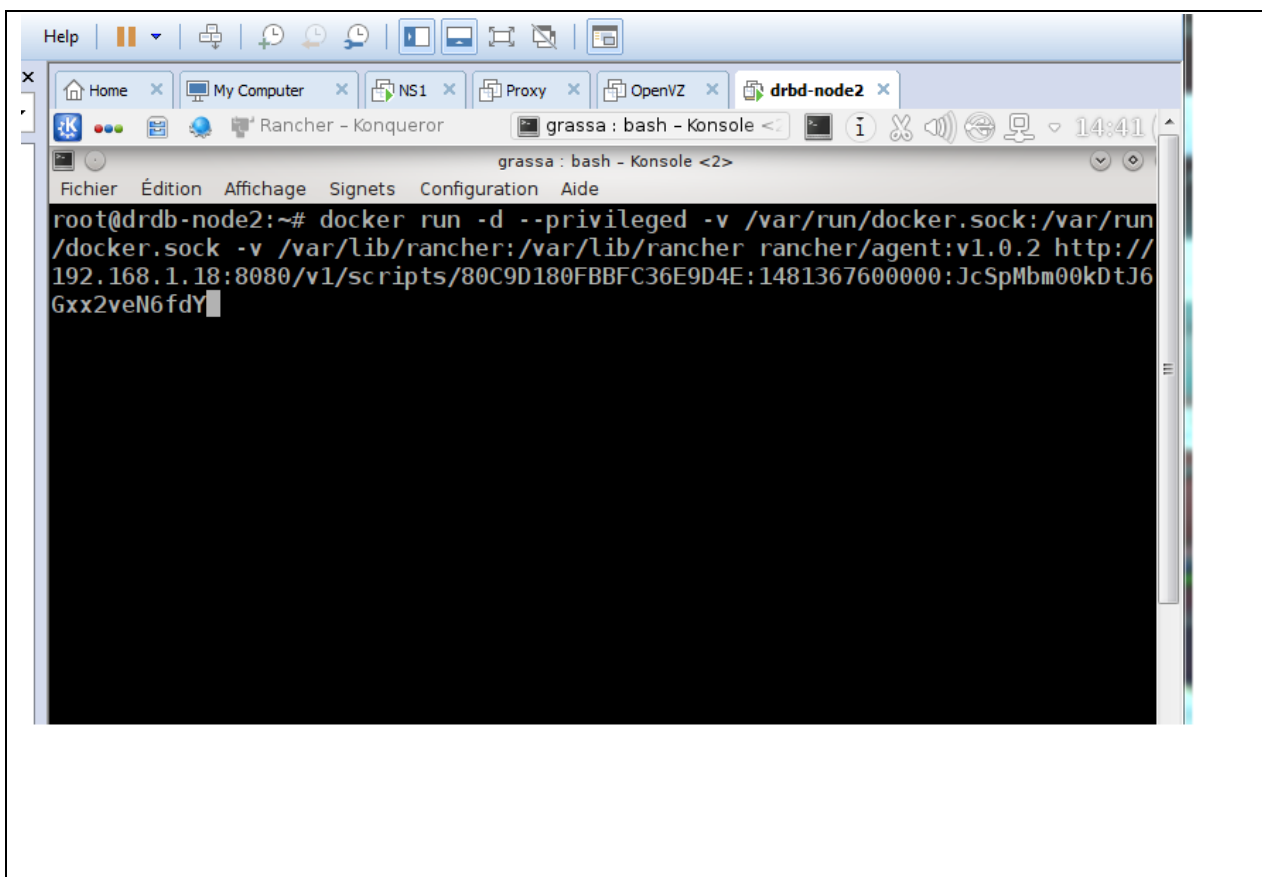
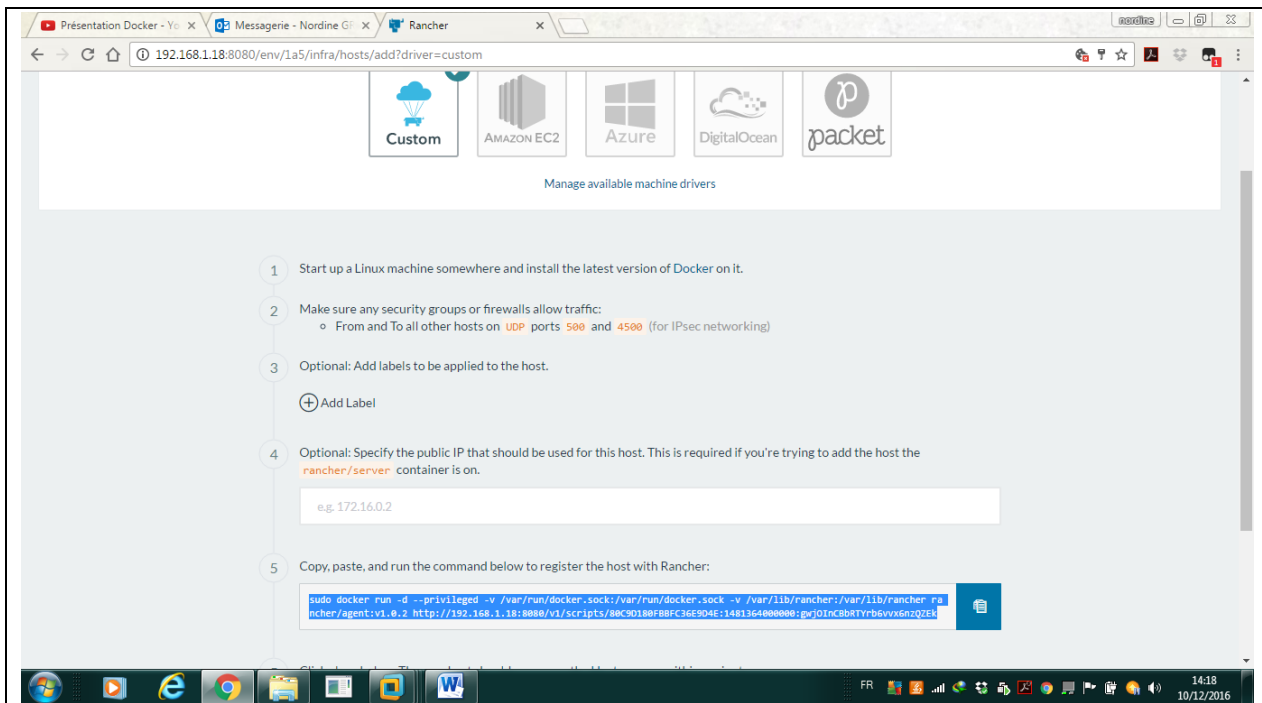
Installation d'un serveur slave :

Prérequis :

Docker installé

Pour pouvoir commencer à héberger des services, Rancher a besoin d'un serveur esclave qui servira de serveur de conteneurs Docker..

Sur le web panel allez dans « INFRASTRUCTURE » puis dans « Hosts ». Enfin, cliquer sur le bouton « Add Host ». Une nouvelle page s'ouvre vous avez juste à **récupérer la ligne de commande dans la section 5** et la coller sur votre serveur esclave. Voici un exemple :



Voici une capture sur un serveur master montrant 2 serveurs slaves dans lesquels tournent des containers Docker.

