

Rancher documentation

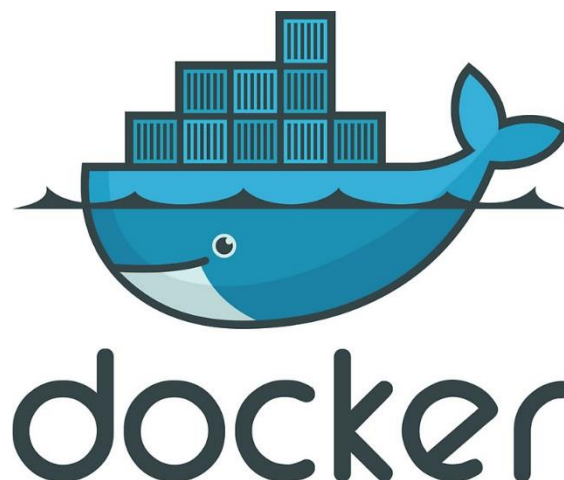


Table des matières

Prérequis	1
Installation.....	2
Installer la base de donnée directement dans le conteneur (tout-en-1).....	2
Installer base de données sur un autre serveur MYSQL	2
Configurer l'authentification	2
Langues.....	2
Désactiver l'analyse des statistiques.....	2
Ajouter des hosts.....	3
La Stack.....	5
Menu stack	5
Création d'une stack.....	5
Configuration d'une stack	6
Configuration d'un Load Balancer	6
Pour aller plus loin.....	8
Gestion des conteneurs et hosts.....	8

Prérequis

- 1 serveur orchestrateur (VM) – 4GB
- 1 ou plusieurs hosts (VM) – 4 GB/VM
- Docker installé sur toutes les machines (script d'installé [ici](#))

Installation

Rancher va se lancer en tant que conteneur. Un conteneur « Rancher serveur » sera installé sur l'orchestrateur. Un conteneur « client » sera installé sur les hosts.

Pour fonctionner Rancher a besoin d'une base de donnée MYSQL. Il est possible de l'installer directement dans le conteneur d'installe ou de l'externaliser. Dans tous les cas il faudra lancer la commande d'installation sur le serveur orchestrateur. Voici les 2 méthodes :

Installer la base de donnée directement dans le conteneur (tout-en-1)

```
docker run -d --restart=always -p 8080:8080 rancher/server
```

Installer base de données sur un autre serveur MYSQL

```
docker run -d --restart=unless-stopped -p 8080:8080 rancher/server \
  --db-host myhost.example.com --db-port 3306 --db-user username --db-
  pass password --db-name cattle
```

Après quelques minutes, se connecter à l'interface web : [http://\[IP:8080\]](http://[IP:8080]) pour accéder à Rancher.

Configurer l'authentification

Par défaut, Rancher ne fournit pas de système d'authentification installé. Pour cela il faut se rendre dans le menu « Admin » ; « Access Control ». Sélectionner ensuite « Local », il ne reste plus qu'à remplir les champs et c'est parti !

Langues

Vous pourrez changer la langue depuis le menu d'authentification, ou en bas à droite sur toutes les pages. Personnellement je le laisse en anglais.

Désactiver l'analyse des statistiques

Pour ne pas envoyer d'informations sur son système, se rendre dans « Admin » ; « Settings ».

Partie Statistics « Décocher » → Save

Ajouter des hosts

Les hosts vont être les machines clientes du cluster qui vont héberger les applications. Il faudra donc plusieurs hosts pour avoir une redondance.

Pour ajouter un hôte suivre les étapes suivantes :
Se rendre dans le menu « Infrastructure » ; « Hosts »
Cliquer sur le bouton « Add Host ».

Hosts: Add Host


Host Registration URL

What base URL should hosts use to connect to the Rancher API?

☒ **This site's address:**

☐ **Something else:**

Don't include `/v1` or any other path, but if you are doing **SSL termination** in front of Rancher, be sure to use `https://`.

 Are you sure all the hosts you will create will be able to reach `http://192.168. [REDACTED]` ?
It looks like a private IP or local network.

Save

Vous devriez voir apparaître l'adresse IP du serveur orchestrateur, et c'est tout à fait normal. EN effet ce menu initialise la gestion du cluster. Sélectionner donc « This site's address » et vérifier que l'adresse IP et le port sont corrects. Ensuite valider avec le bouton « Save ».



Manage available machine drivers

- 1 Start up a Linux machine somewhere and install a supported version of Docker on it.
- 2 Make sure any security groups or firewalls allow traffic:
 - o From and To all other hosts on UDP ports 500 and 4500 (for IPsec networking)
- 3 Optional: Add labels to be applied to the host.

⊕ Add Label
- 4 Specify the public IP that should be registered for this host. If left empty, Rancher will auto-detect the IP to use. This generally works for machines with unique public IPs, but will not work if the machine is behind a firewall/NAT or if it is the same machine that is running the `rancher/server` container.

1.1.1.1
- 5 Copy, paste, and run the command below to register the host with Rancher:

```
sudo docker run -e CATTLE_AGENT_IP="1.1.1.1" --rm --privileged -v /var/run/docker.sock:/var/run/docker.sock -v /var/lib/rancher:/var/lib/rancher rancher/agent:v1.2.10 http://192.168.68.68/v1/scripts/02100988118f90a73397:1514678400000:vdc6CHMFTe1idoYwFqsgge3XgE
```
- 6 Click close below. The new host should pop up on the `Hosts` screen within a minute.

Close

Sur cette étape vous devez remplir le champs en encadré rouge avec une adresse IP d'un host du cluster.

Une fois ceci fait, il faudra copier coller le contenu de la commande **sur le serveur Host** pour installer le client Rancher.

→ Répéter la manipulation pour chaque host à ajouter.

Après quelques instant l'host apparaitra avec tous les conteneurs systèmes qui ont été installés dessus. Pour information le client Rancher installe plusieurs conteneurs afin gérer l'host : réseau, management, data etc...

Pour cacher ses informations depuis l'interface, décocher « Show System ».

La Stack

La stack correspond à la pile désignant un ensemble de service. Ces services vont être déployés par le biais de conteneurs sur les Hosts.

Par défaut vous trouverez une stack nommé « Default », mais nous allons créer notre propre stack !

Menu stack

Le menu stack est constitué de 3 sous menu :

- **All** : Contient toutes les stacks.
- **User** : Contient toutes les stacks crée « maison », personnelles.
- **Infrastructure** : Contient toutes les stacks portant sur le système et l'infrastructure Rancher.

Création d'une stack

Nous allons donc crée notre stack !

Cliquer sur « Stack » ; « All »

Cliquer sur le bouton « Add Stack ».

The screenshot shows the 'Add Stack' form in Rancher. It includes a 'Name' field with the value 'Nginx' and a 'Description' field with the value 'Nginx description'. Below these fields is a section labeled 'OPTIONAL: IMPORT COMPOSE' which contains two sub-sections: 'Optional: docker-compose.yml' and 'Optional: rancher-compose.yml'. Each sub-section has a text area for the file contents and a blue upload button. At the bottom of the form is a section labeled 'ADVANCED OPTIONS' with a dropdown arrow. At the very bottom are 'Create' and 'Cancel' buttons.

Donnez-lui simplement un nom et une description.

Rendez-vous maintenant dans « Stack » ; « User ». Vous devriez voir votre nouvelle stack. Cliquer dessus pour la configurer !

Configuration d'une stack

Comme je l'ai dit précédemment la stack est un ensemble de service. Nous allons créer notre premier service, pour cela cliquer sur le bouton « Add Service ».

The screenshot shows the 'Add Service' configuration page in Docker Swarm. At the top, there's a 'Scale' section with a slider and two radio buttons: 'Run 1 container' (selected) and 'Always run one instance of this container on every host'. Below this is a button labeled 'Nginx-service' and a link '+ Add Sidekick Container'. The main form has two columns: 'Name' with a text input containing 'Nginx-service', and 'Description' with a text input containing 'Nginx-service description'. Below the 'Name' input is a 'Select Image*' dropdown menu showing 'nginx:latest'. To the right of the 'Description' input is a checkbox labeled 'Always pull image before creating' which is checked. At the bottom right, there is a '+ Port Map' button.

Ce service crée un conteneur avec une instance du serveur web NGINX, simple pour vérifier le fonctionnement il suffit de se connecter en http.

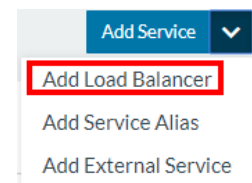
Vous remarquez que nous n'avons pas encore rediriger les flux (par le biais des numéros de ports), donc notre conteneur n'est pas encore accessible.

Configuration d'un Load Balancer

Il y a 2 manière de procéder pour rediriger les flux par le biais des ports.

- **SOIT** vous rediriger les ports d'un conteneur, et dans ce cas se sera CE conteneur qui aura le service
- **SOIT** vous utilisez un Load balancer. Le load balancer va permettre de rediriger les fluxs, donc les ports vers plusieurs conteneurs → **Meilleure solution**

Pour créer le Load-Balancer, rendez-vous dans votre stack (Nginx) et à coté de « Add Service » dérouler les sous-menu avec la petite flèche bleue et sélectionner Load-Balancer.



Add Load Balancer ⓘ

Scale

☒ Run 1 container
 ☐ Always run one instance of this container on every host

Name **Description**

Port Rules ⊕ Add Service Rule ⊕ Add Selector Rule

Access*	Protocol*	Request Host	Port*	Path	Target*	Port*
Public	HTTP	e.g. example.com	80	e.g. /foo	Nginx/Nginx-service	80

Host and Path rules are matched top-to-bottom in the order shown. Backends will be named randomly by default; to customize the generated backends, provide a name and then refer to that in the custom haproxy.cfg. Show custom backend names. Show host IP address options.

[SSL Termination](#)
[Stickiness](#)
[Custom haproxy.cfg](#)
[Labels](#)
[Scheduling](#)

There are no SSL/TLS ports configured.

[Create](#) [Cancel](#)

Choisissez 1 conteneur, c'est suffisant pour rediriger le trafic. Donnez-lui un nom, une description.

Après c'est la partie la plus importante :

- Pour le « Request Host » donc le serveur Host
 - ❖ Vous pouvez rediriger un nom de domaine précis
 - ❖ Choisissez le port 80 puisque nous voulons rediriger le port 80 du serveur Web Nginx.
- Pour la « Target » donc le conteneur
 - ❖ Sélectionner le service vers lequel vous voulez rediriger le trafic
 - ❖ Sélectionner le port sur lequel le service est actif, pour le serveur web Nginx c'est 80.

Créez le Load-Balancer et attendez quelques instants (il est possible que son statut soit en « Initialize », attendez que la configuration se termine).

Vous devriez avoir cette vision de la stack Nginx.

Stack: 🔗 Nginx ▼ Add Service ▼ 🔗 Active ⚙️ ⋮

Description: Nginx Description

⚙️ Active	Load-balancer ⓘ	To: Nginx-service Ports: 80/tcp	Load Balancer	1 Container	⊕ ⓘ
🔗 Active	Nginx-service ⓘ	Image: nginx:latest	Service	1 Container	⊕ ⓘ

Il est maintenant possible d'augmenter le nombre de conteneur, et le trafic sera réparti entre les différents conteneurs.

Pour visualiser le résultat cliquer sur « 80/tcp ». Cela ouvrira un nouvel onglet avec l'adresse IP de l'host hébergeant le Load-Balancer.

Conclusion :



Pour accéder aux services peu importe leurs configurations (hosts, IP...), ce qui compte est le load-balanceur puisque c'est LUI qui va rediriger vers le service.

Pour aller plus loin

Gestion des conteneurs et hosts


On peut remarquer que lors de la création de nouveaux conteneurs, ils sont créés aléatoirement sur les hosts. Ce qui peut poser problème, comme par exemple pour le load-balancer. Pour pallier à ce problème il existe les **Labels**. Le label est une variable affectée à un host, qui déterminera les conteneurs qui lui seront affectés automatiquement.

Pour la configurer, rendez-vous dans le menu « Infrastructure » ; « Hosts ».

Cliquer sur  et  Edit

Partie « Require Container Label » :

Require Container Label

 Add Required Label

Key

io.rancher.container.system

Value

Value

Key

Value

ProTip: Paste one or more lines of key=value pairs into any key field for easy bulk entry.

If provided, containers must have one or more of the given labels in order to be eligible for scheduling onto this host. If the value is left empty, a container with a matching key is eligible regardless of value.

`io.rancher.container.system` is required to allow system containers on to the host.

Save

Cancel

Laisser la première valeur. Vous pouvez en ajouter autant que vous voulez, voici des exemples de variables :

Key	Value	Commentaires
io.rancher.stack.name	<name_of_a_stack>	Tous les services de la stack seront déployés sur cet host (à part le load balancer). Attention cependant cet host ne pourra plus héberger d'autres conteneurs que ceux du service concerné. (Il est possible d'attribuer plusieurs labels pour pallier à ce problème).