

# User Manual for Docks

TripleParity

## Contents

### 1 System Overview

Docks is a system to manage a Docker Swarm using a Web User Interface. It provides a visual representation of a Docker Swarm, allowing users to manage the Swarm without using the Command Line Interface.

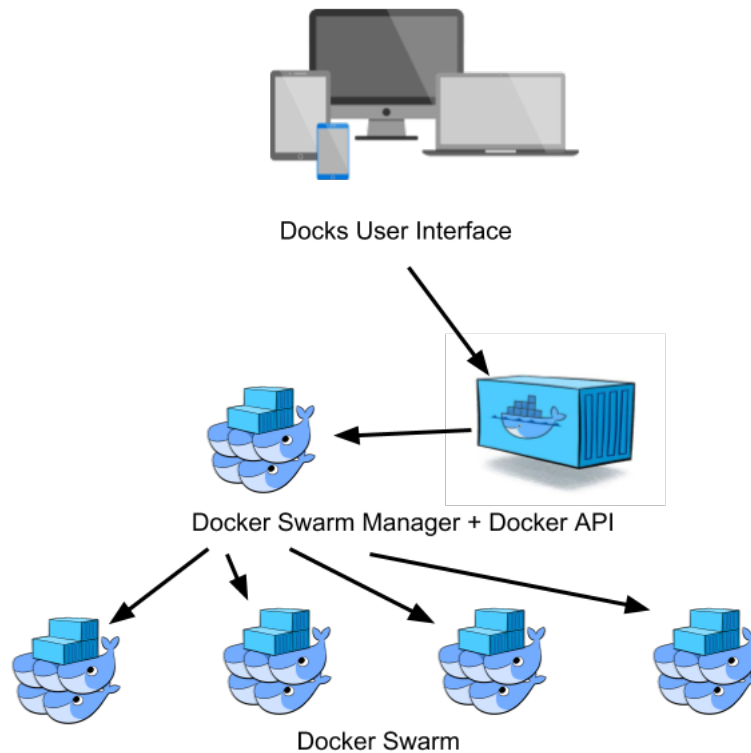
Docks is useful if you want to see a quick overview of the Swarm and all the running containers. Containers can be started and stopped with the click of a button.

System Administrators will be able to effectively manage their Swarm and scale services as required. Docks also appeals to users wishing to manage their services remotely from a web browser.

### 2 System Configuration

Docks consists of two subsystems; the Docks API server and the Docks UI.

The Docks API server has to be on a Manager Node in the Docker Swarm to be able to manage other nodes. The Docks UI also has to be served from the same Domain Name as the Docks API, in most cases it will also be deployed on the Manager Node.



### 3 Installation

1. Install Docker (<https://docs.docker.com/install/>)
2. Install Docker Compose (<https://docs.docker.com/install/>)
3. Make sure the Docker Daemon is running (<https://docs.docker.com/config/daemon/>)
4. Clone the docks-demo repository using git (<https://github.com/TripleParity/docks-demo>)
5. Run `sudo docker-compose up -d` in the docks-demo folder that was cloned. This will start the Docks API on port '8080' and the web server on port '4200'. To view the output of the Dock API and web server, omit the '-d' flag. This process may take a while since Docker has to download the images for Docks
6. Browse to <http://127.0.0.1:4200> to access the web interface.
7. To stop the Docks UI and Docks API run `sudo docker-compose down` in the docks-demo folder to stop the running containers (docks and docks-ui)

### 4 Getting Started

When using the docks-demo repository the UI will be available at <http://127.0.0.1:4200> after following the Installation instructions.

You will be presented with the following login screen:

A login form titled "Login" with two input fields: "Username \*" and "Password \*". Below the fields is a purple "Login" button.

Figure 1: Docks Login Screen

Default credentials:

- Username: "admin"
- password: "admin"

After logging in you will be presented with the Dashboard:



Figure 2: Dashboard Navigation bar



Figure 3: Views

The containers view will show running and stopped containers. From this view you can start and stop existing containers or remove them.

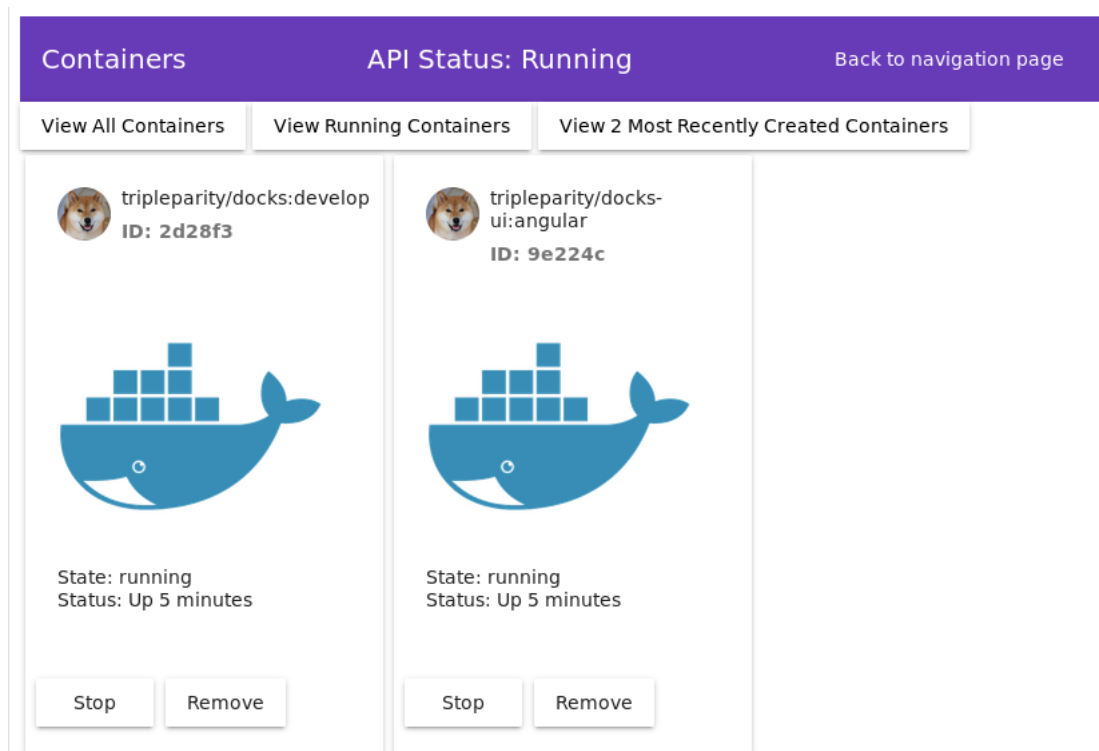


Figure 4: Running containers

## 5 Using the System

### 5.1 Managing Containers

#### 5.1.1 Start and Stop container

From the containers view, you can start existing containers that are not running or stop running containers. This is useful if you no longer require the service running in the container but want to keep it around.

#### 5.1.2 Remove containers

If you no longer need the service provided by the container, or the data in the container layer (data in volumes will not be deleted) then you can click on the "Remove" button to remove the container from the swarm. This

cannot be undone.

## 5.2 Stacks



Figure 5: Stacks Dashboard

You can paste a *docker-compose.yml* file into the text box. The 'Create Stack' button will deploy the stack to the Docker Swarm. You can then manage the containers in the Containers dashboard.

## 5.3 Updating Docks

The current process to update Docks is to run

```
docker-compose pull
```

and then

```
docker-compose up --force-recreate
```

# 6 Troubleshooting

## 6.1 Error: bind: address already in use

Another service is most likely running on port 4200 or 8080. The ports for Docks can be specified in the *docker-compose.yml* file by changing for example

```
ports:
  - 4200:80
```

to

```
ports:
  - 9000:80
```

to run on port 9000 on the host machine.

## 6.2 404 Not Found

URLS such as <http://127.0.0.1:4200/index> cannot be directly visited, you first need to visit <http://127.0.0.1:4200>