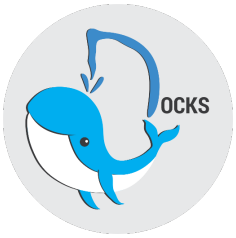
User Manual for Docks

docks-ui 0.0.2

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.

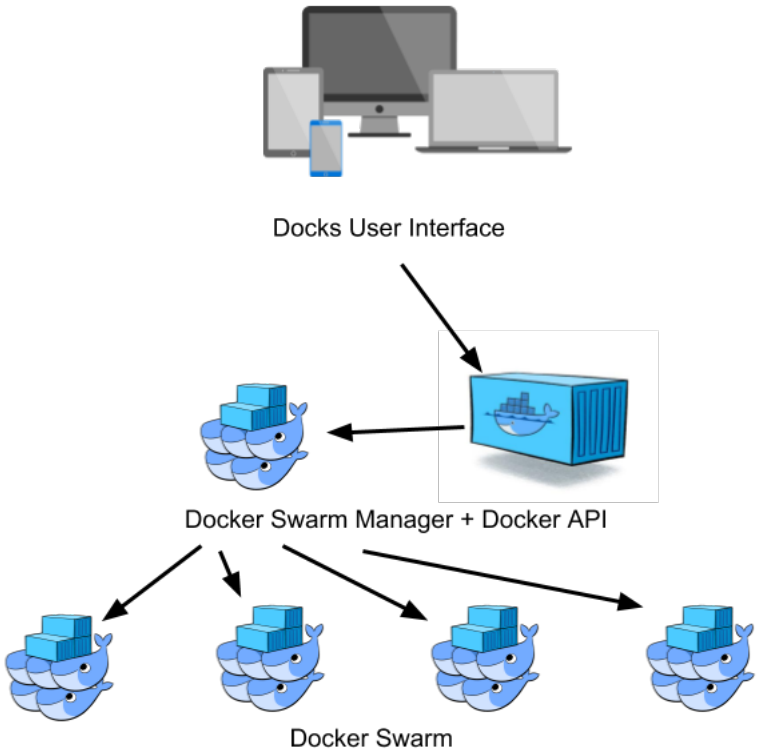
Docks is useful if you want to see a quick overview of the Swarm and all the running containers. Containers can be started and stopped with the click of a button.

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 by deployed on the Manager Node.



3 Installation

1. Install Docker
2. Clone `https://github.com/TripleParity/docks.git` and `cd` into `docks`
3. Run `sudo docker-compose pull` to download the required images
4. Run `sudo docker-compose up -d --force-recreate` to start the Docks stack
   - Note: This might take a while as Docker has to download a large amount of data
   - This will start the **Docks API** on port `8080` and the **Docks UI** on port `4200` .
5. Initialise a swarm using `sudo docker swarm init`
6. Deploy the demo stack to the swarm using `sudo docker stack deploy -c docker-compose-nginx.yml nginx-demo`
   - This will start 1 service with 3 tasks on port `8081` . You can view them at `http://127.0.0.1:8081`
   - The request will be balanced between the three tasks
7. Browse to `http://127.0.0.1:4200` to access the web interface.
   - For more information see the User Manual
8. To stop the running services run the following commands:
   - `sudo docker-compose down -v`
   - `sudo docker stack rm nginx-demo`

4 Getting Started

The web interface will be available at `http://127.0.0.1:4200` after following the Installation instructions. All required setup should be automatically done when starting the Docks stack.

The home screen should display when visiting `http://127.0.0.1:4200`

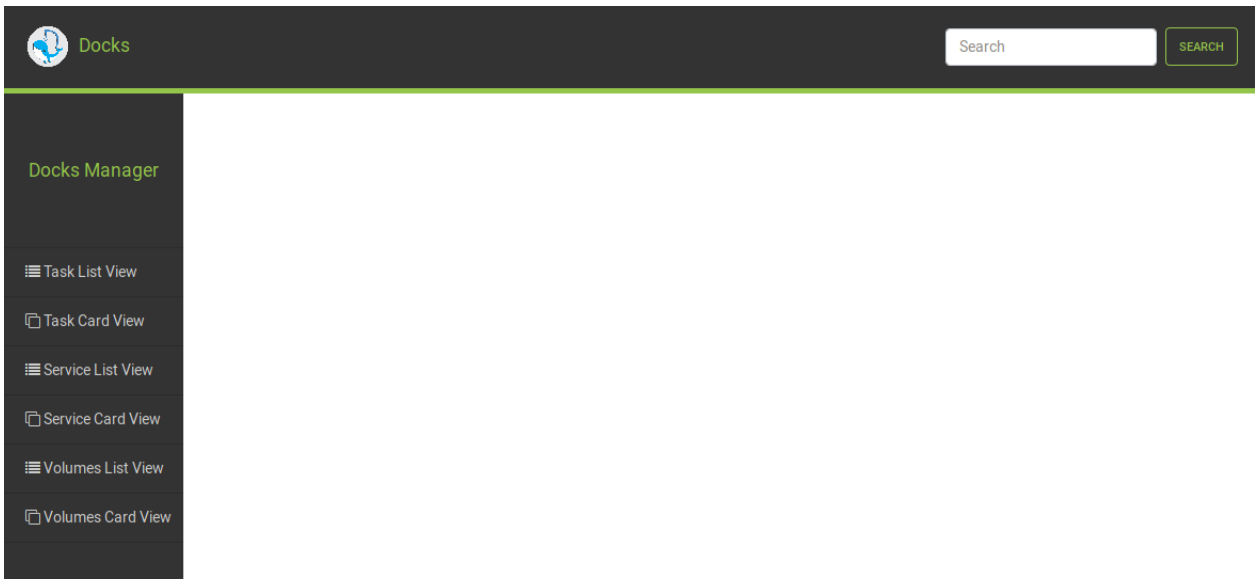


Figure 1: Home Screen

5 Using the System

5.1 Views

All information can be viewed in two ways, as a list or as cards. The list view is more compact but provides less information initially. This is useful for seeing a summary of all running resources. Card view provides more detail without extra information and therefore fewer resources can be seen at a time. Card view is useful if you need to see more detailed information at a quick glance.

5.2 Tasks

Tasks are the containers associated with a running service. On the task list view you can see the task ID and it's state. Upon clicking on a task in the list it will expand to also show the node ID the task is running on, the image the task was created from, the date created and the date update.

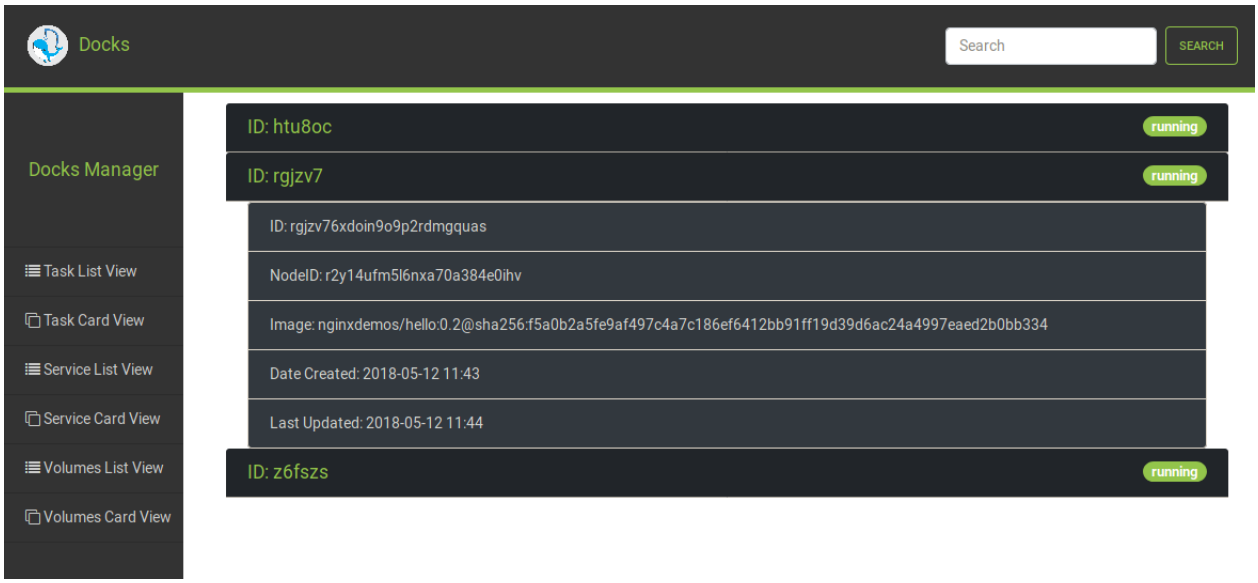


Figure 2: Task list view, with one task expanded

The card view displays the following information:

- ID of the task
- State of the task. Indicated by the circle and card outline colour. Green means the task is running, red indicating it has stopped
- Node ID the task is running on
- Last Updated

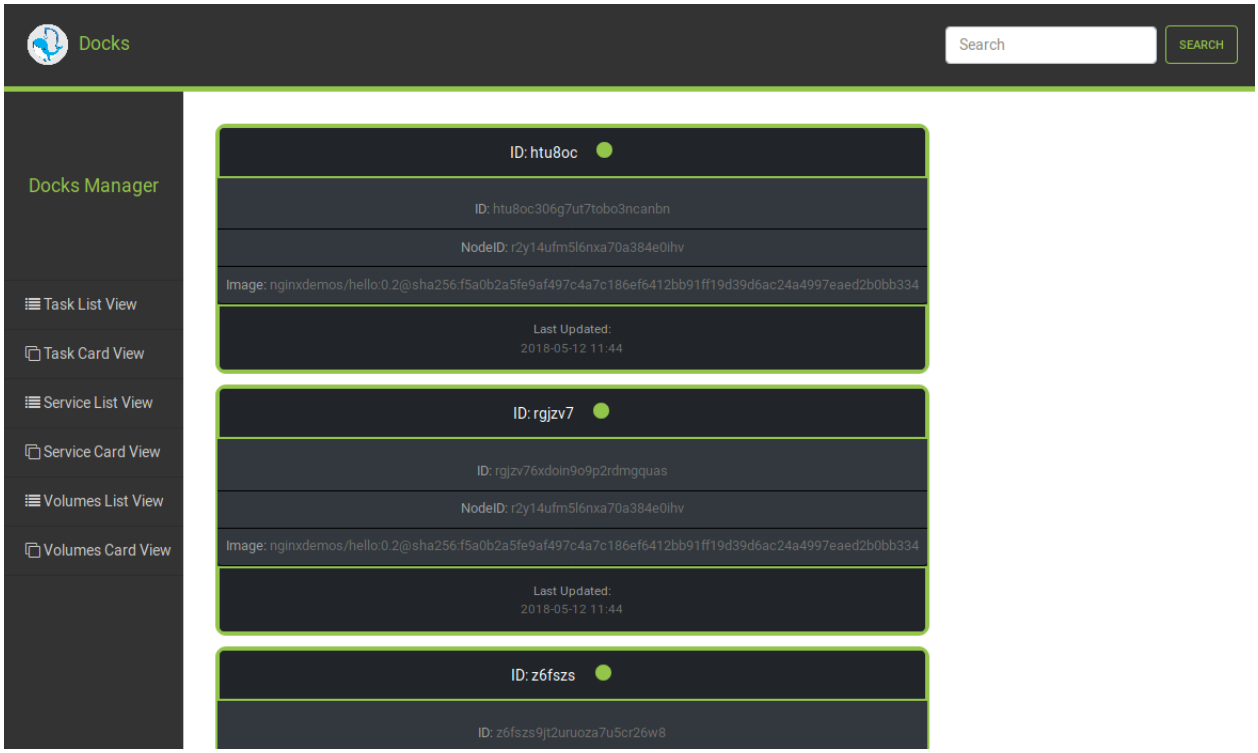


Figure 3: Task card view

Cards can be clicked on to bring up a modal dialog to display the log for that service.

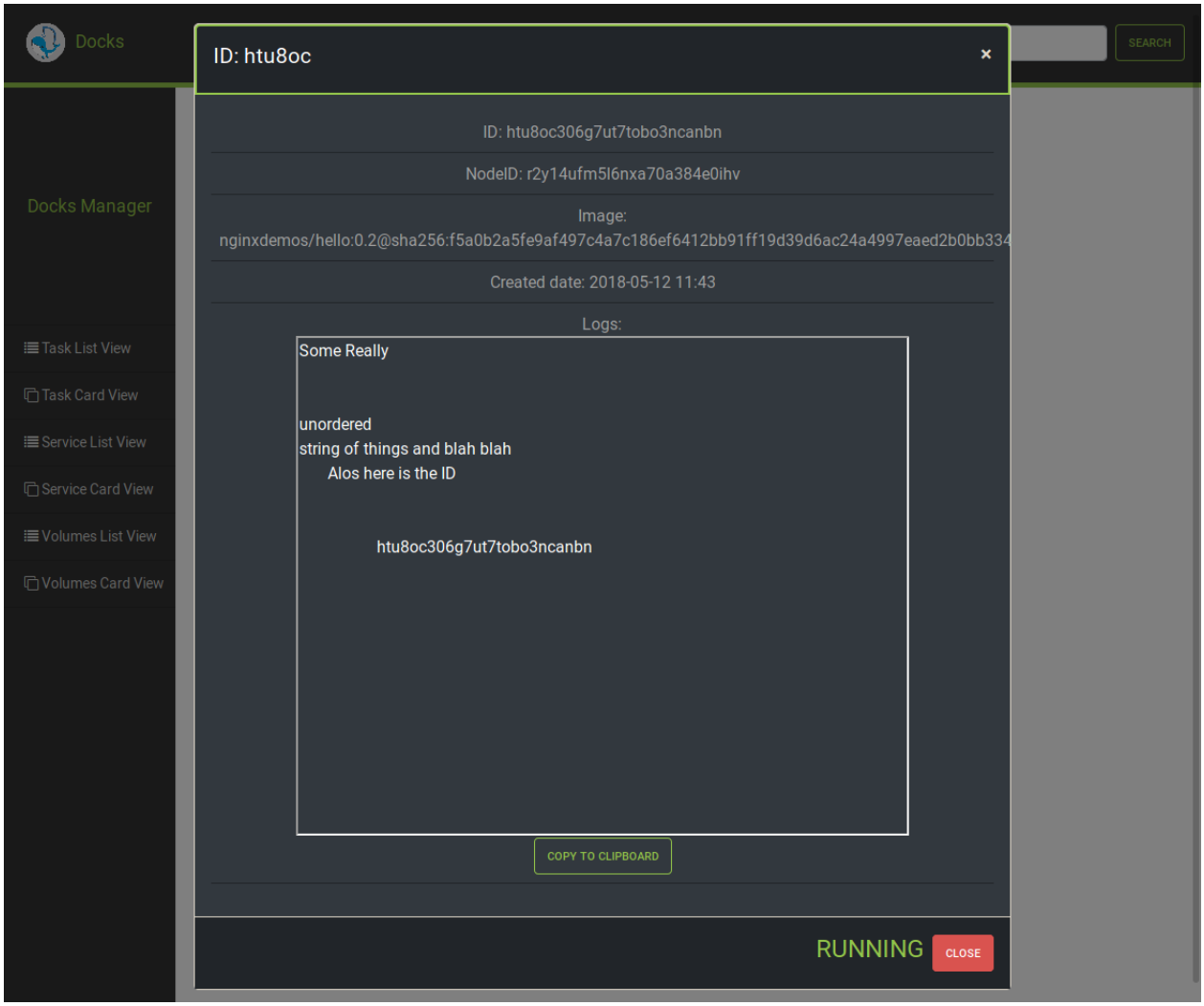


Figure 4: Task modal after clicking on card. Displays task log

5.3 Services

The services list view displays a list of services, which can be clicked on for an expanded view:

- Image the service is created from
- The mode of the service. `vip` - Virtual IP, the service is assigned an IP address and will distribute the requests to the replicated tasks.
- The number of replicas of the service. Each replica represents a task in the swarm.
- Date crated
- Last updated

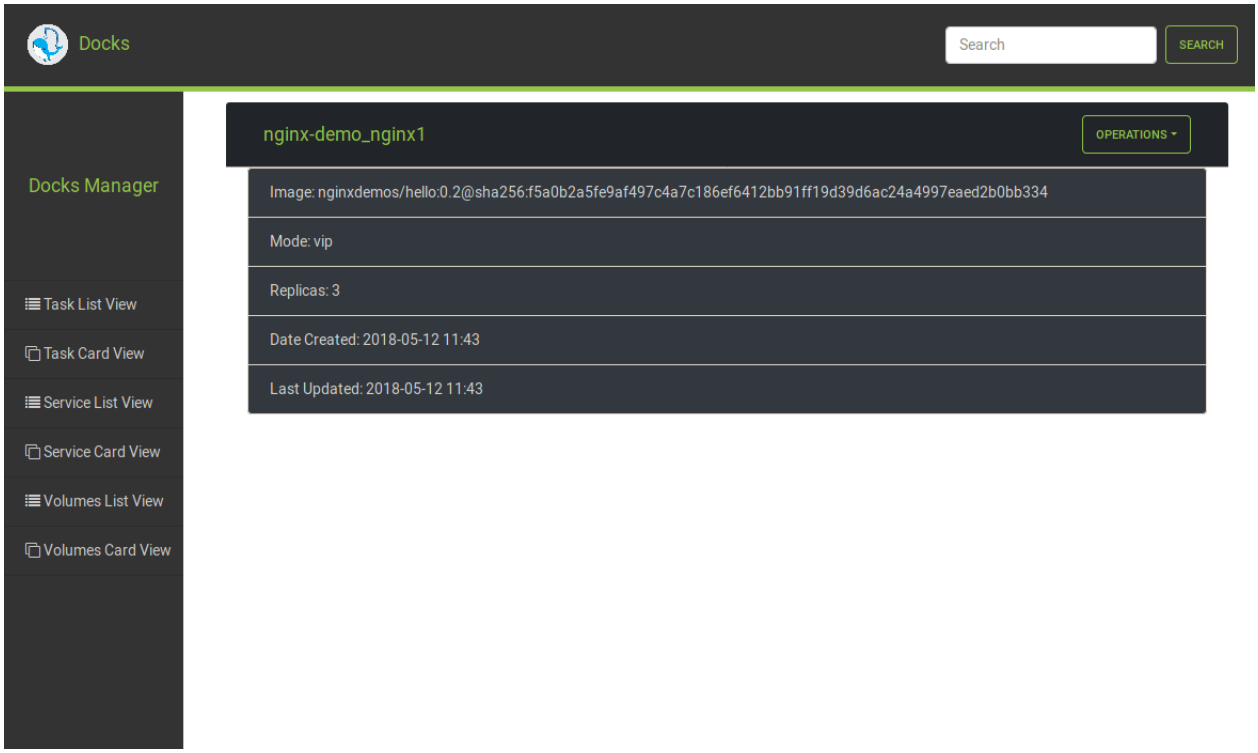


Figure 5: Task list view, with one task expanded

The services card view will display the following information:

- Service name
- Image the service is created from
- The mode of the service. `vip` - Virtual IP, the service is assigned an IP address and will distribute the requests to the replicated tasks.
- The number of replicas of the service. Each replica represents a task in the swarm.
- Last updated

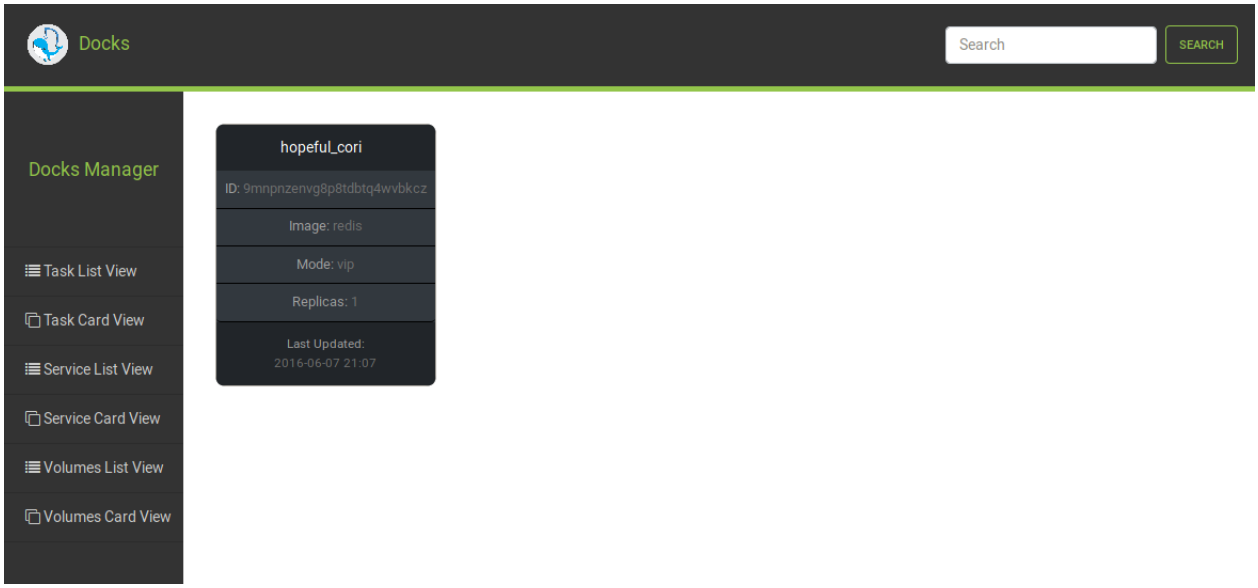


Figure 6: Services card view

When clicking on a Service Card the following additional information will be displayed:

- Ports the services makes use of
- Tasks associated with the service
- Date Created

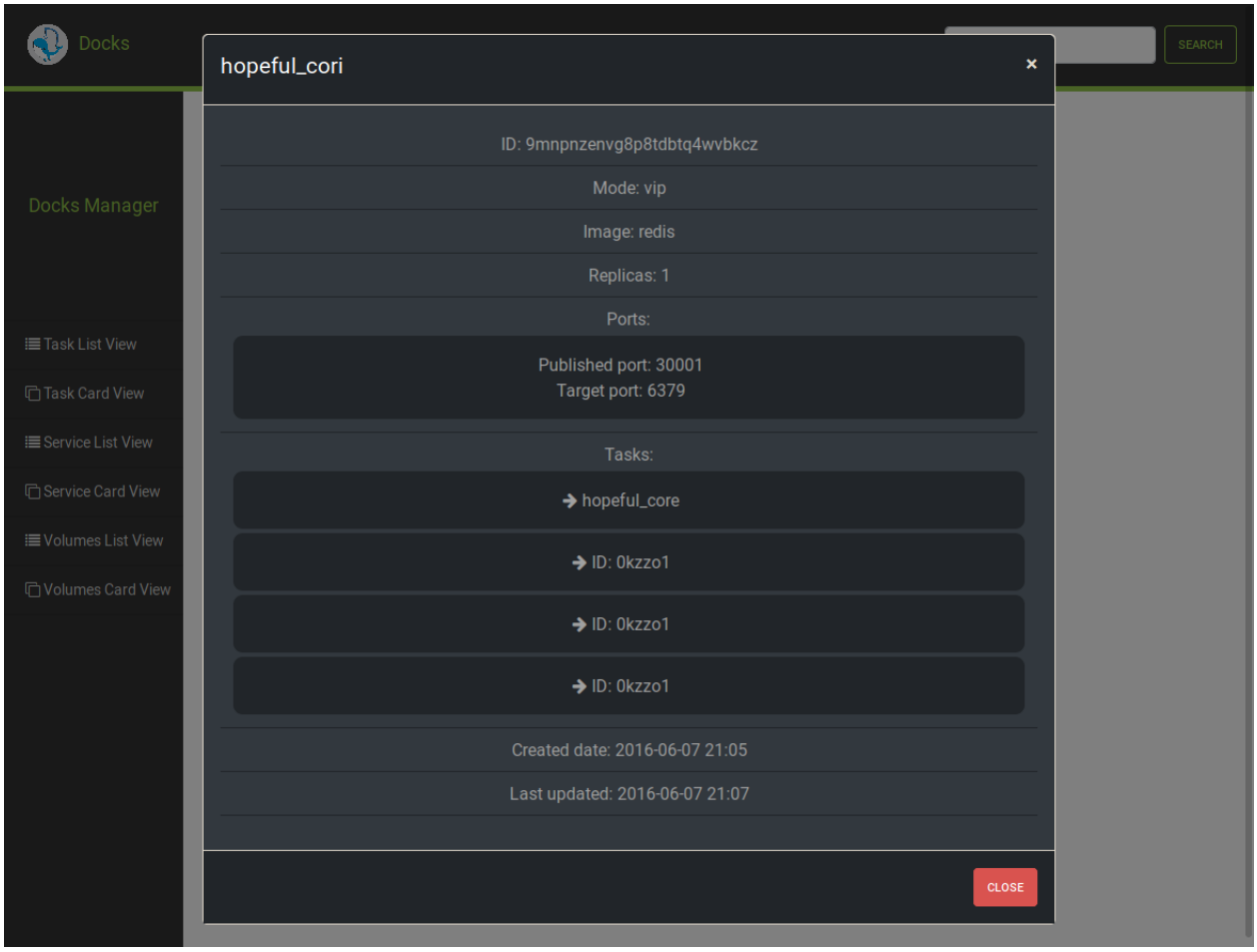


Figure 7: Services card view

5.3.1 Service Operations

From the list view you can perform the following operations by clicking on the `Operations` button:

- Scale the service
- Update the service
- View the service logs
- Remove the service from the swarm

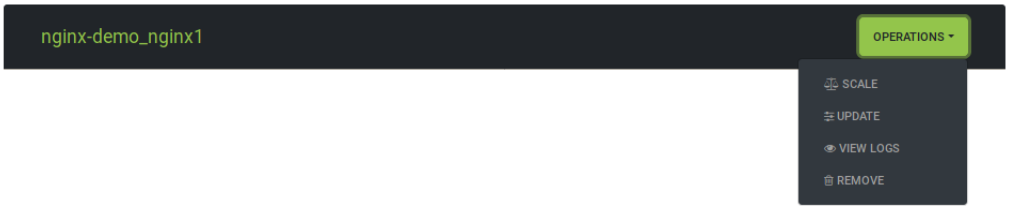


Figure 8: Service list operations

You can scale the service to change the number of replicas (tasks) of the service to run distribute on the swarm. This is useful if the service needs more resources to handle more requests.

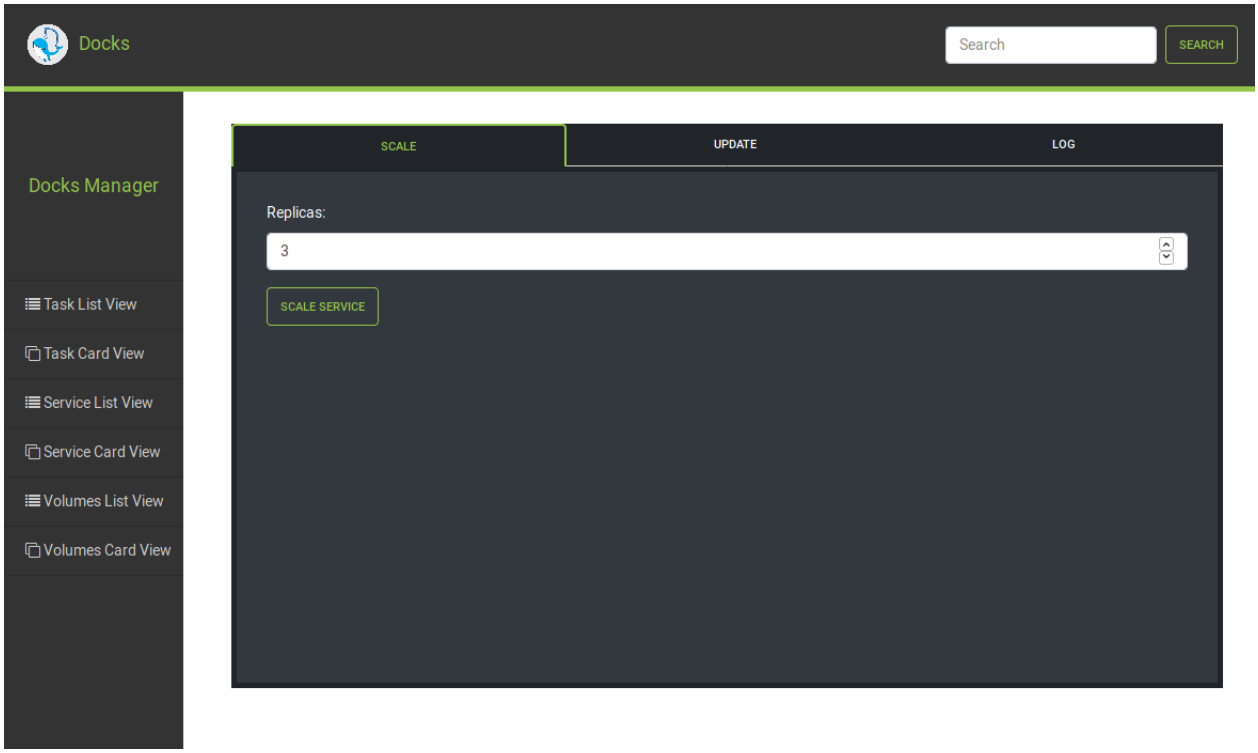


Figure 9: Service list operations

The service name and mode can be changed under the `Update` tab

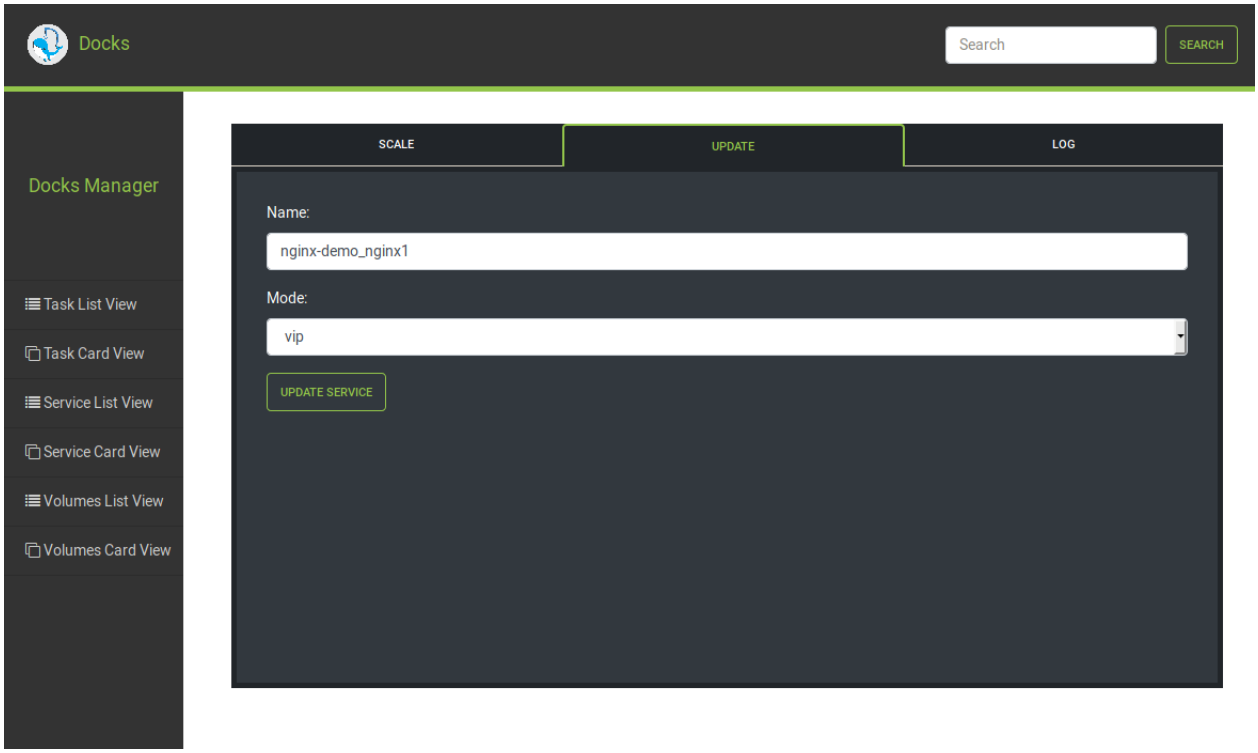


Figure 10: Service list operations

The service log can be viewed under the `Log` tab. This is an aggregate log for all the tasks associated with the service.

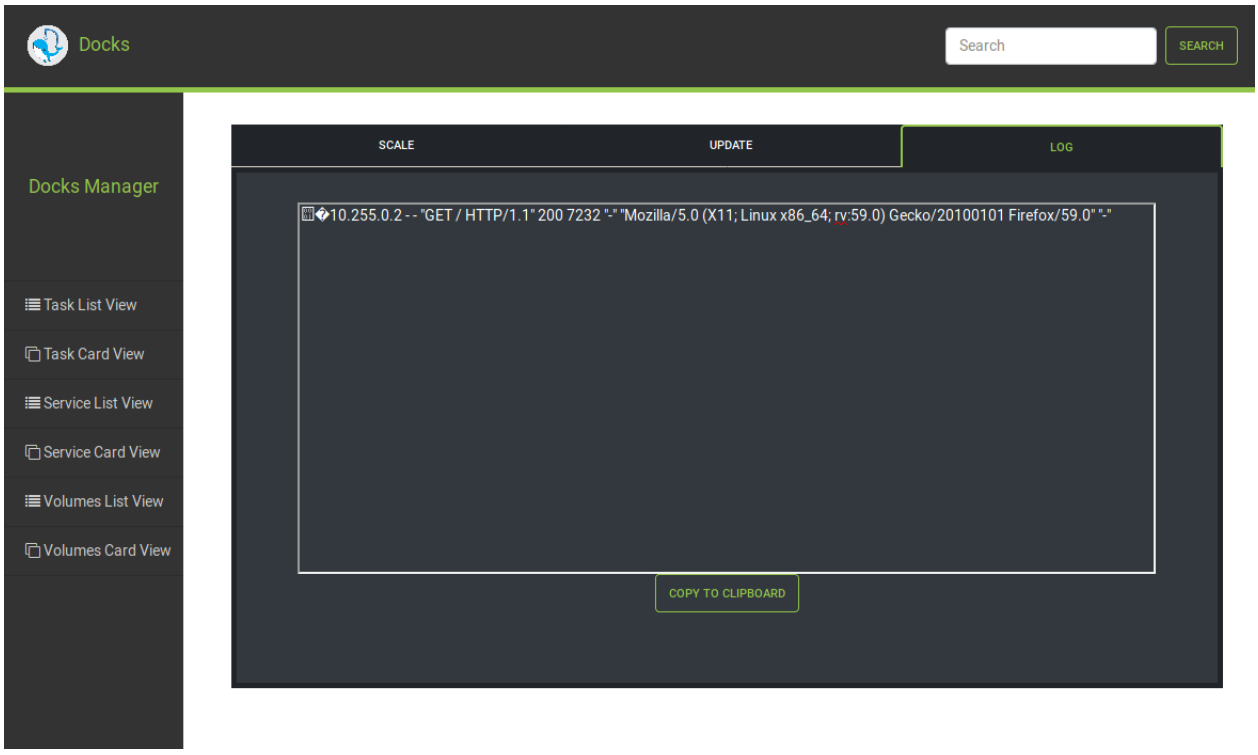


Figure 11: Service list operations

5.4 Volumes

The volumes list view will display the following information when expanded:

- Driver the volume is using.
- Where the volume is mounted on the host
- Scope of the volume. Can be local or swarm wide
- Optional labels
- Further options
- Date created

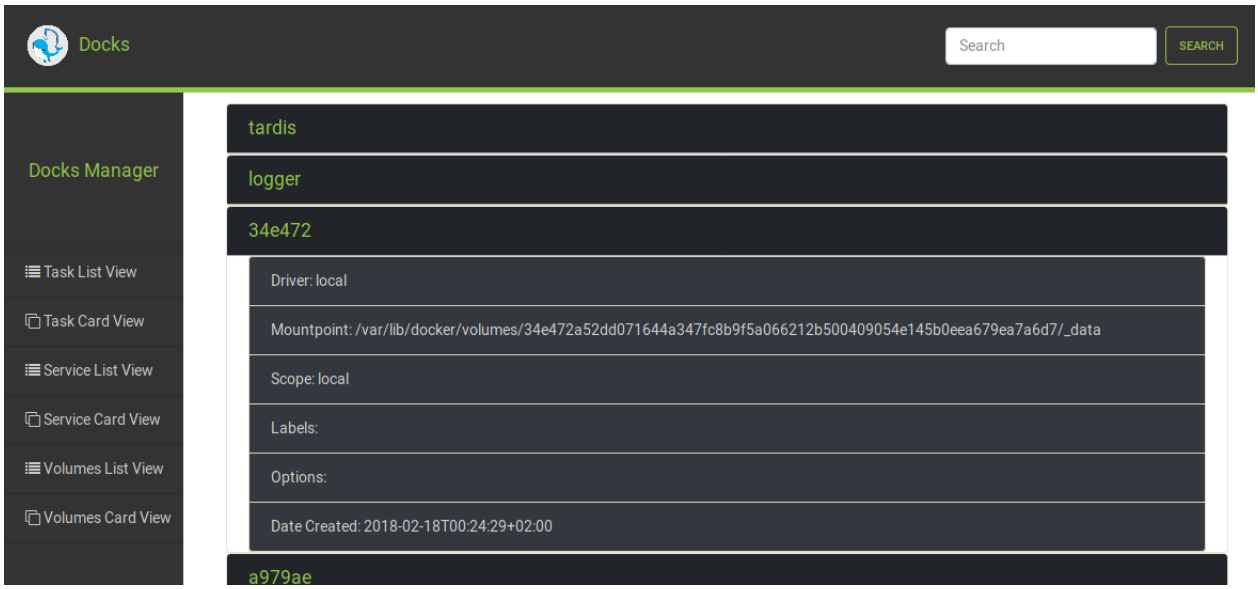


Figure 12: Volume list view, with an expanded item

The volumes card view will display the following information:

- Date created
- Driver the volume is using.
- Where the volume is mounted on the host

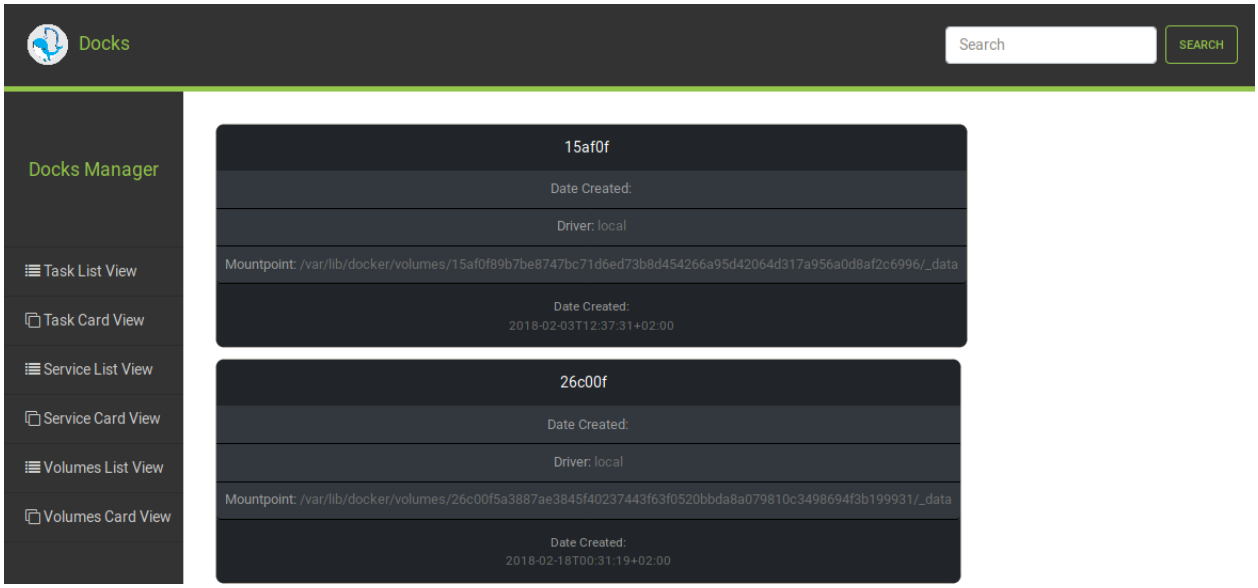


Figure 13: Volume list view, with an expanded item

5.5 Updating Docks

Docks can be updated by running `sudo docker-compose pull` and then `docker-compose up --force-recreate` in the `docks` cloned repository

6 Troubleshooting

6.1 Error: bind: address already in use

Another service is most likely running on port `4200`, `8080` or `8081`. The ports for Docks and nginx-demo can be specified in the *docker-compose.yml* and *docker-compose-yml* files.

For example to run on port 9000 instead of 4200 make the following changes:

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

to

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