

Docker Workshop

Appendix 1

Working with Rancher Desktop

1	REFERENCES	1
2	GENERAL CONFIGURATION	1
3	KUBERNETES CONFIGURATION	2
4	IMAGES DASHBOARD UI.....	4
4.1	FILTERING AND SORTING IMAGES	4
4.2	DELETING IMAGES	5
4.3	PULLING IMAGES.....	5
4.4	BUILDING IMAGES FROM A DOCKERFILE	6
5	CONTAINERS DASHBOARD UI.....	6
5.1	FILTERING AND SORTING CONTAINERS	7
5.2	STARTING AND STOPPING CONTAINERS	7
5.3	DELETING CONTAINERS.....	8

1 References

The main official reference for Rancher Desktop

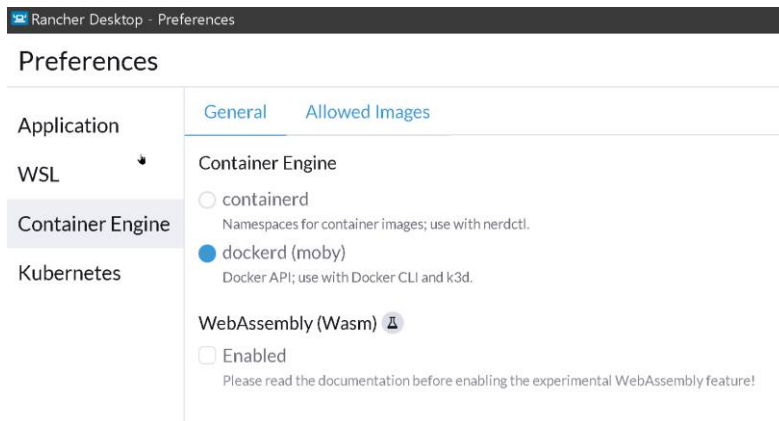
<https://docs.rancherdesktop.io/>

Reference for working with the UI

<https://docs.rancherdesktop.io/ui/general>

2 General Configuration

By default, Rancher is configured to use `dockerd` (moby) as the container engine; this is the open source version of the engine used by Docker.



With this engine selected, you can use your typical Docker commands via:

`docker command`

If you use the `containerd` engine, then you would precede your typical Docker commands with:

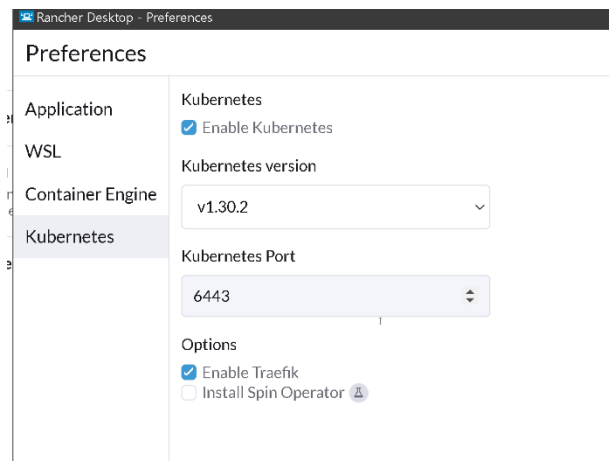
`nerdctl command`

instead

3 Kubernetes configuration

Rancher Desktop uses K3s, a lightweight Kubernetes distribution. This is already activated when Rancher Desktop starts. K3s is implemented through a set of containers that are generated from existing images.

Rancher also uses the Traefik ingress controller to control incoming traffic into the Kubernetes cluster.



With all the above options enabled by default, the following `localhost` ports will be bound for use

- 80 (Traefik)
- 443 (Traefik)
- 6443 (Rancher)

Therefore, when working with K3s in Rancher, ensure you do not perform any mapping from a Docker container or a Kubernetes pod to these ports on `localhost`.

With K3s enabled, whenever you run the standard Docker CLI commands to view images or containers:

```
docker images
```

```
docker ps
```

you will see all these additional images and containers that are associated with K3s, in addition to any of your own images that you have pulled from DockerHub and any of your own containers that you have started from them.

These additional K3s related images / containers will clutter your listing of images or containers.

To work around this without removing the K3s related images and containers, you can view the images / containers using the Dashboard UI, which will automatically exclude the K3s containers but the K3s images will still display.

If you are working purely using CLI commands, you can alternatively use a `select-string` Powershell function to exclude the K3s related images / containers, for e.g.

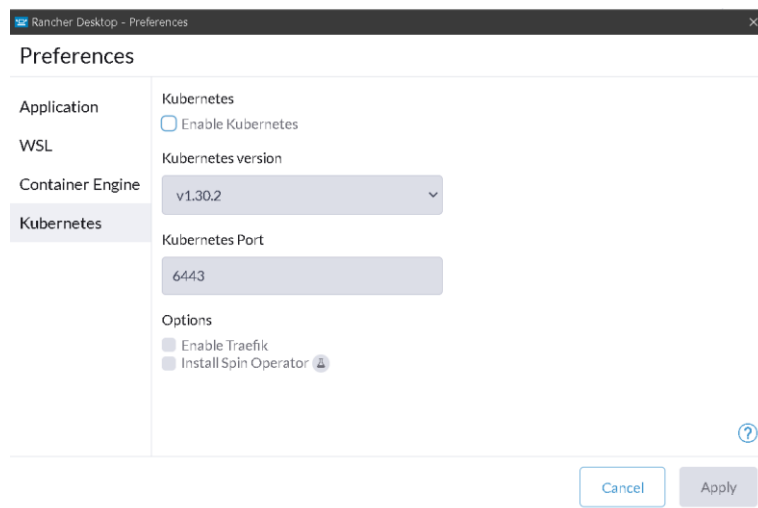
```
docker images | select-string -Pattern "rancher" -NotMatch
```

```
docker ps -a | select-string -Pattern "k8s" -NotMatch
```

The other approach is to temporarily disable the K3s cluster by going to Preferences and

- uncheck Enable Kubernetes
- uncheck Enable Traefik

and then click Apply to restart Rancher Desktop.



When Rancher restarts again, you can subsequently remove any existing K3s related containers and images by running these CLI commands in the given sequence:

```
docker rm -f $(docker ps -a -q)
```

```
docker rmi -f $(docker images -q)
```

Then check again for all containers and images with:

```
docker images
```

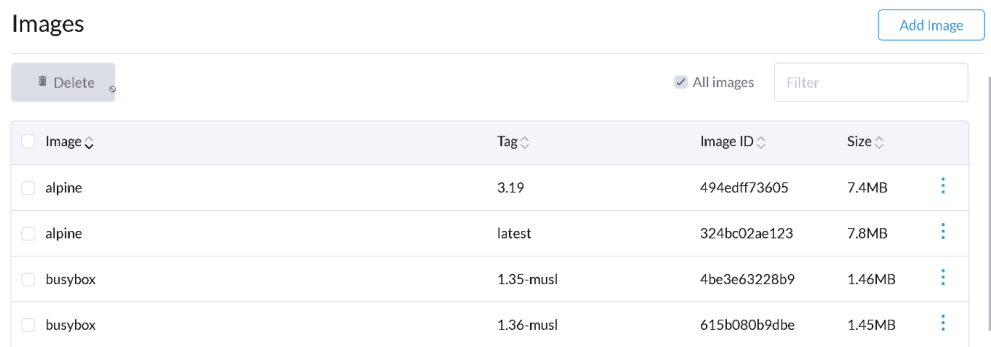
```
docker ps -a
```

You should not see anything there anymore.

IMPORTANT: If you use this approach, remember to enable Kubernetes again in Preferences when you need to start working on the Kubernetes in a workshop or at your workplace.

4 Images dashboard UI

You can manage your images directly from the dashboard as an alternative to the standard Docker CLI commands.

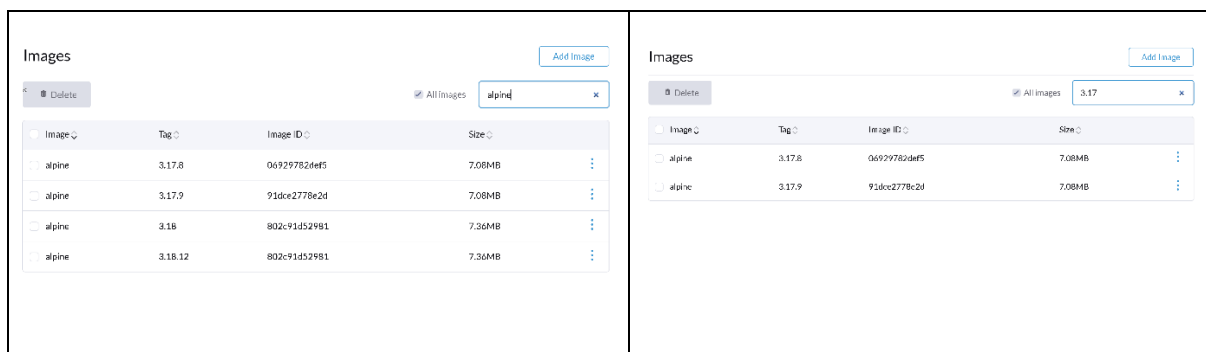


You can perform the following image related functions from the UI

- Filter and sort on existing images in the local registry
- Select specific images to delete (equivalent to `docker rmi`)

4.1 Filtering and sorting images

You can filter on image or tag names



You can sort (ascending / descending order) on any one of the column headers (Image, Tag, ImageID and Size)

Image ↕	Tag ↕	Size ↕
<input type="checkbox"/> alpine	1.35-musl	7.36MB
<input type="checkbox"/> alpine	1.36-musl	7.36MB
<input type="checkbox"/> alpine	1.37.0-glibc	7.08MB
<input type="checkbox"/> alpine	1.37.0-glibc	7.08MB
<input type="checkbox"/> busybox	3.17.8	

4.2 Deleting images

Examples of selecting and deleting images

Images

Delete

1 selected

Image ↕

Tag ↕

busybox

latest

<input type="checkbox"/>	alpine	3.18	802c91d52981	7.36MB
<input checked="" type="checkbox"/>	alpine	3.17.9	91dce2778e2d	7.08MB

Delete

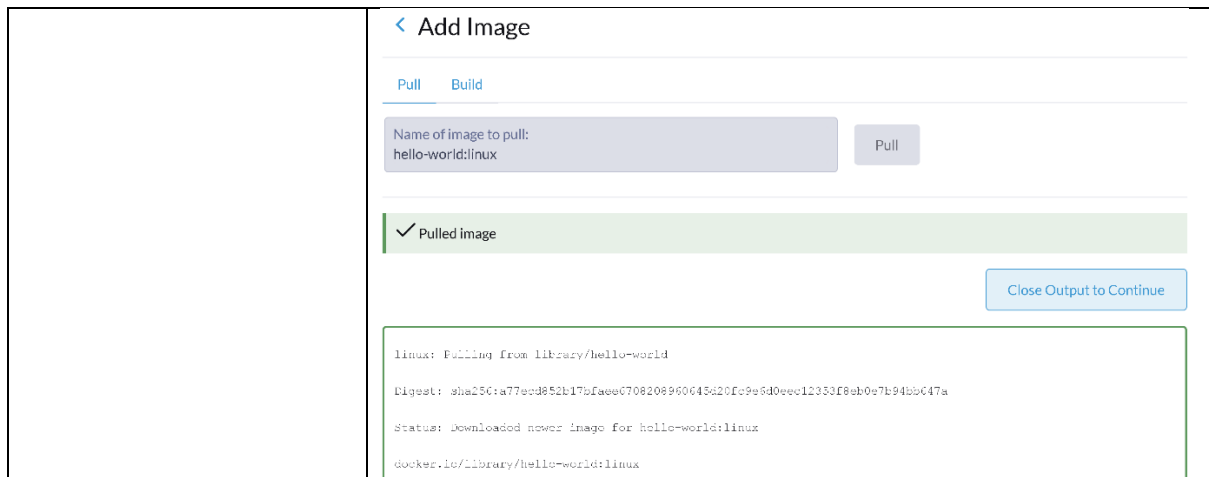
Scan...

IMPORTANT: To view latest updates to the list of images due to addition of new images or deletion of existing images, move away from the Images main dashboard view by clicking on another one of the menu options (e.g. General, Containers, Port Forwarding), wait a while and then click back on Images.

4.3 Pulling images

You can also add an image by pulling it from DockerHub (equivalent to `docker pull`)

Images <input type="button" value="Add Image"/>	
<input checked="" type="button" value="Actions"/> <input checked="" type="checkbox"/> All Images <input type="text" value="Filter"/>	



IMPORTANT: To view latest updates to the list of images due to addition of new images or deletion of existing images, move away from the Images main dashboard view by clicking on another one of the menu options (e.g. General, Containers, Port Forwarding), wait a while and then click back on Images.

4.4 Building images from a Dockerfile

Not done yet ????

???? ??? or by running a build on a Dockerfile contained in a specific directory (equivalent to `docker build`)

5 Containers dashboard UI

You can manage your containers directly from the dashboard as an alternative to the standard Docker CLI commands.

Containers

<div> <div>Stop</div> <div>Start</div> <div>Delete</div> </div> <div>Filter</div>				
<input type="checkbox"/> State	Name ↕	Image ↕	Port(s) ↕	Uptime ↕
<input type="checkbox"/> running	superman	alpine		Up 9 minutes ⋮
<input type="checkbox"/> exited	goofy_rubin	hello-world		⋮

You can perform the following container related functions from the UI

- Determine which containers are stopped or running
- Selecting specific containers to stop, start and delete (functionally equivalent to `docker stop`, `docker start` and `docker rm`)
- Filtering on containers based on their names

5.1 Filtering and sorting containers

You can filter on the state (running, exited), name or image column values

Containers

Stop

running

<input type="checkbox"/>	State	Name	Image	Port(s)	Uptime
<input type="checkbox"/>	running	charming_williams	alpine		Up 5 seconds

Containers

Stop

Actions

exited

<input type="checkbox"/>	State	Name	Image	Port(s)	Uptime
<input type="checkbox"/>	exited	blissful_kare	hello-world		

Containers

Stop

Actions

hello

<input type="checkbox"/>	State	Name	Image	Port(s)	Uptime
<input type="checkbox"/>	exited	blissful_kare	hello-world		

Containers

Stop

charming

<input type="checkbox"/>	State	Name	Image	Port(s)	Uptime
<input type="checkbox"/>	running	charming_williams	alpine		Up 5 seconds

You can sort on the Name, Image, Port and Uptime column values as well.

5.2 Starting and stopping containers

You can select and then start and stop existing containers

Containers

Stop

Start

Delete

1 selected

<input checked="" type="checkbox"/>	State	Name
<input checked="" type="checkbox"/>	running	superman

Containers

Stop

Start

Delete

1 selected

<input type="checkbox"/>	State	Name
<input type="checkbox"/>	running	superman
<input type="checkbox"/>	running	charming_williams
<input checked="" type="checkbox"/>	exited	blissful_kare

IMPORTANT: To view latest updates to the list of containers due to the starting or stopping of existing containers, move away from the Containers main dashboard view by clicking on another one of the menu options (e.g. General, Images, Port Forwarding), wait a while and then click back on Containers. Some containers may also take a while to become active after starting, depending on the base OS and applications contained within them.

5.3 Deleting containers

You can also select and delete containers that have stopped.

Containers

<div>Start Delete 1 selected</div>		
✓ State	Name ↕	Image ↕
✓ exited	busy_mendel	hello-world