Generally Useful Docker Commands

Remove all Docker Containers

Stop the container(s):

```
cd ~/mydocker-repo-folder;
docker-compose down;
```

Delete all containers :

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

Delete all volumes:

```
docker volume rm $(docker volume ls -q)
```

Delete all networks:

```
docker network rm $(docker network ls -q)
```

Kill a specific container :

```
docker container kill [CONTAINER_NAME]
```

Saving and Restoring Docker Containers

In cases where you cannot for whatever reason build docker containers on your local system, do not fear. Docker allows you to save and import backed up images of containers.

Saving Containers

It is a good habit to routinely save containers. Just open a terminal and use the docker save command. Example here:

```
docker save -o ~/Desktop/my_docker_image.tar laravel
```

Once that is saved you can share it with other developers or keep it as a personal backup. You can also share it with another developer directly using JustBeamIt.

Restoring from a Container Image

If one of your containers is acting wonky, you can get the name and image id with the following command:

```
docker images
```

You can see the image name and id in the list.

DeepinScreenshot_select-area_20200902153536.png

If the container is running, you can shut it all down with “docker-compose down”. Then you can delete the offending container with the docker rm command. Here is an example:

```
# kill docker compose
cd ~/my-docker-folder;
docker-compose down;


docker image rm 3f8c96702c14
```

Now you can load a new container to replace the broken one. To do this you will need to get an image from another developer or use one you previously saved.

To load the container from the image use the docker load command. Example here:

```
docker load -i ~/Desktop/my_docker_image.tar
```

Running multiple services in one container

In my case I want to serve some pages with php and others with nodejs within the same container. This saves a lot of build time and memory. So here is what I add to my Dockerfile

```
CMD /config_items/run.sh
```

Then in the file run.sh I start php, nginx and run a nodejs app all in different threads using a single ampersand to run each command in a differrent thread. This lets me run as many processes as needed concurrently.

```
service php8.0-fpm start & nginx -g 'daemon off;' & cd /var/www/pslamp-blog && npm run start
```

In cases where you cannot for whatever reason build docker containers on your local system, do not fear. Docker allows you to save and import backed up images of containers.

Saving Containers

It is a good habit to routinely save containers. Just open a terminal and use the docker save command. Example here:

```
docker save -o ~/Desktop/my_docker_image.tar laravel
```

Once that is saved you can share it with other developers or keep it as a personal backup.

Restoring from a Container Image

If one of your containers is acting wonky, you can get the name and image id with the following command:

```
docker images
```

docker image rm IDOFBADCONTAINER

Now you can load a new container to replace the broken one. To do this you will need to get an image from another developer or use one you previously saved.

To load the container from the image use the docker load command. Example here:

```
docker load -i ~/Desktop/my_docker_image.tar
```

Revision #2
Created 2024-06-14 01:00:57 UTC by peterd
Updated 2024-06-14 01:02:04 UTC by peterd