Docker Commands for Daily life

The following list of Docker commands provide a handy quick reference to tasks you will use regularly.

`$ docker -machine ls`
List all machines.

`$ docker-machine ls -t 60`
Allow 60 seconds to grab all instance needed for multiple cloud deployments.

`$ docker ps`
List running containers.

`$ docker ps -a`
List all stopped containers.

`$ docker ps -s`
List running containers, their size and virtual disk space used.

`docker inspect <friendly-name|container-id>`
Provides more details about a running container, such as IP address.

`docker port <friendly-name|container-id> 6379`
Gives you the external port assigned to an internal port. In the example, will report the external port number mapped/assigned to the internal port of 6379.

`docker logs <friendly-name|container-id>`
Display messages the container has written to standard error or standard out.

`$ docker rm 'containername'`
Removes a stopped container.

$ docker images
List all images downloaded.

$ docker rmi imagename
Remove an image.

$ docker system df
Show Docker disk usage.

$ docker system events
Show real-time events from the server.

$ docker system info
Display system-wide information.

$ docker system prune
$ docker system prune -a --volumes
Remove unused data and clean up your Docker environment. - *Warning any non-running resources will be removed!*

Docker Container Export preserving all file systems layers

The following group of commands deal with the export and import of containers. All data in the container will be saved during export. After Import, the file system will look exactly the same. Volume attachments will not be saved nor copied but can be reattached.

$ docker export --output="latest.tar" containername
Export saving all data in the container.

$ docker import latest.tar imagename
Import the Docker image to a newly created image.

$ docker run --name nameofyourcontainer -d -it imagename /bin/sh
Test run the image as a container.

$ docker exec -it containername /bin/bash
Login to a container for troubleshooting and customization.

$ docker build -t image-name:versionnumber .
Docker Compose file and image created from the directory with the Dockerfile.

$ docker run -d -p 80:80 image-name:ver
Start the image you just built as a container using the internal port 80 mapped to the host port 80

Don McCullough

Solution Architect - DAIR-ATIR

www.canarie.ca

don.mccullough@canarie.ca