Sudo yum install –y yum-utils device-mapper –persistent-data lvm2

Add the repository

Sudo yum –config –manager –add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

Yum install docker-ce

Sudo systemctl enable docker

Sudo systemctl start docker

Sudo docker run hello-world

Scripts for this installation can be found at:

[https://get.docker.com](https://get.docker.com/)

and

<https://github.com/docker/docker-install>

To pull the script to install docker

wget –q0- <https://get.docker.com> |sh

==============

Docker cheat sheet

-d to run container in background

-p to publish the network port for the container

Docker run –it –d Ubuntu (to Run a docker image)

Docker ps –a ( to see list of running container)

Docker exec –it <image id> bash – to execute and go to container

Docker stop <id>

Docker commit <container id> <name of docker hub repo/docker image>

Docker push <name of docker hub repo/docker image>

Docker rm <container id>

Docker rmi <image id >

=====================

Create a backup that can then be used with docker load.

$ docker save busybox > busybox.tar

$ ls -sh busybox.tar

2.7M busybox.tar

$ docker save --output busybox.tar busybox

$ ls -sh busybox.tar

2.7M busybox.tar

$ docker save -o fedora-all.tar fedora

$ docker save -o fedora-latest.tar fedora:latest

Save an image to a tar.gz file using gzip

You can use gzip to save the image file and make the backup smaller.

docker save myimage:latest | gzip > myimage\_latest.tar.gz

Cherry-pick particular tags

You can even cherry-pick particular tags of an image repository.

$ docker save -o ubuntu.tar ubuntu:lucid ubuntu:saucy

==============================

Docker images

Docker run –it –d Ubuntu ( to run docker images in background)

Docker ps –a (to list containers)

Docker exec –it <container id> bash

Created a directory

Docker stop <container id>

Docker rm <container id > - to remove a stopped container

Docker commit <container id> user id/image name

Docker ps

Docker push user id/image name (provide credentials)

Docker stop rm <comtainer id>

Docker ps

Docker images

Docker rmi <image id>

Docker rmi –f <image id>

=============================

Useful links

<https://runnable.com/docker/java/dockerize-your-java-application>

<https://stackify.com/guide-docker-java/>

<https://hub.docker.com/_/openjdk>

<https://github.com/docker-library/openjdk>

<https://runnable.com/docker/java/dockerize-your-java-application>

<https://github.com/docker-library/openjdk/blob/master/8/jdk/Dockerfile>

<https://github.com/wsargent/docker-cheat-sheet> - Cheat sheet

<https://phusion.github.io/baseimage-docker/>

<https://medium.com/edureka/docker-commands-29f7551498a8>

<https://medium.com/@migueldoctor/how-to-create-a-custom-docker-image-with-jdk8-maven-and-gradle-ddc90f41cee4>

# <https://codefresh.io/docker-tutorial/java_docker_pipeline/> - Crafting the perfect Java Docker build flow

<https://medium.com/@hudsonmendes/docker-spring-boot-choosing-the-base-image-for-java-8-9-microservices-on-linux-and-windows-c459ec0c238>

<https://runnable.com/docker/java/dockerize-your-java-application>

<https://developers.redhat.com/blog/2019/02/26/create-java-8-runtime-container-image/>

Docker Swarm – Provides native clustering capabilities to turn a group of docker engines into a single virtual docker engine. With these pooled resources we can scale out the application.

Docker manager its responsible to check app/services are running on container.

# Difference between save and export

As I described in my last post (<http://tuhrig.de/difference-between-save-and-export-in-docker>), there are **two ways to persist a Docker images or container**:

* A Docker image can be saved to a tarball and loaded back again. This will preserve the history of the image.

|  |  |
| --- | --- |
| 1  2  3  4  5 | # save the image to a tarball  docker save <IMAGE NAME> > /home/save.tar    # load it back  docker load < /home/save.tar |

* A Docker container can be exported to a tarball and imported back again. This will not preserve the history of the container.

|  |  |
| --- | --- |
| 1  2  3  4  5 | # export the container to a tarball  docker export <CONTAINER ID> > /home/export.tar    # import it back  cat /home/export.tar | docker import - some-name:latest |

================================

Docker tag Ubuntu docker.jfrogdev.com/Ubuntu

Docker push docker.jfrogdev.com/Ubuntu

Docker pull docker.jfrogdev.com/Ubuntu

=============================