

docker

Docker - containerization of Oracle

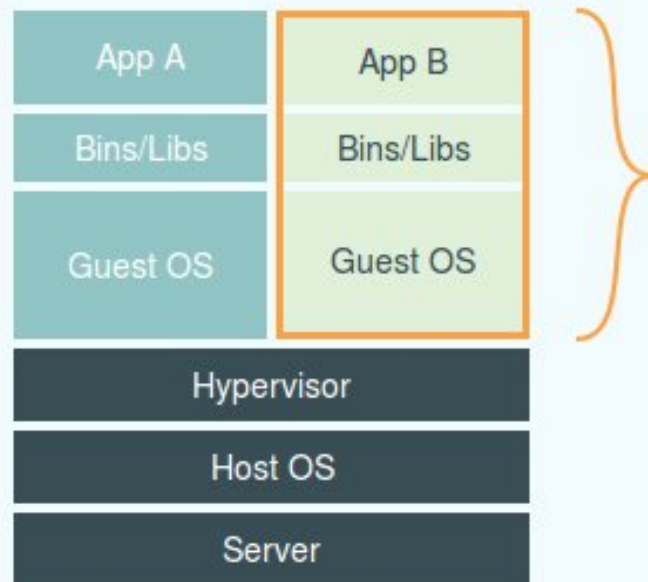
Get started with Docker
and get your DB running.

Oracle Stockholm TechTalk
Ulf Hellström

Ulf Hellström (About me)

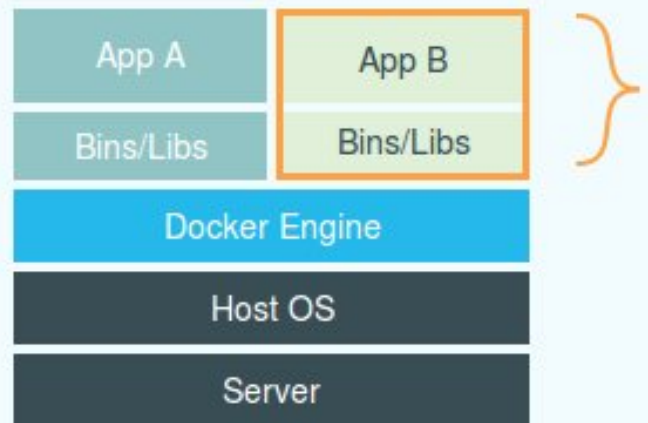
- ❖ More than 25 years of passion for Oracle database.
(version 5 - 12)
- ❖ Programming is a kind of Art. (Microcontrollers /
Assembler / C / Python / PLSQL / LiveCode /
JavaScript)
- ❖ Consultant working for Miracle as a generic-specialist
in Development / Architecture / DBA / Performance /
Apex / Rest / Engineered Systems.

Docker versus virtualization



Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.



Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

Where to get Docker ?

- ❖ <https://www.docker.com/>

Pro's and Con's

- ❖ Easy to get started (Very simple installation and even support on Windows)
- ❖ Mature project with lots of support
- ❖ Easier to get things running then virtual machines
- ❖ 1000's of ready made images on Docker Hub.
- ❖ Link containers together and port forwarding support.
- ❖ Easy to create own images to contribute to the community.
- ❖ Support for on-premise as well as cloud (Azure,AWS,Oracle)

Pro's and Con's (continue..)

- ❖ Memory hog (Don't try to run containerized Oracle on less than 8GB unless Oracle XE)
- ❖ Virtual file space can take allot of disk space even for a small machine (can be altered).
- ❖ Since no other kernel then host O/S some things can be a challenge like if you really need ssh.
- ❖ Persistence on image level by default. Easy to forget to "commit" changes to the image if you need it for all containers.

Areas for usage

- ❖ Education
- ❖ Demo (since very easy to get to 0-state.)
- ❖ Development / Test environment
- ❖ Production (not really when it comes to Oracle DB due to patching is time-consuming when it comes to containers..)

Oracle's support for Docker

- ❖ Docker supported images on GitHub (<https://github.com/oracle/docker-images>)
- ❖ Oracle support for docker in the cloud (Oracle Container Cloud Service) Released 2017.

Hello-World the Docker Way

- ❖ `docker run hello-world` (That's it!!)

Ubuntu the Docker way

- ❖ `docker pull ubuntu` (Download and install)
- ❖ `docker run -d --name ulf-ubuntu ubuntu tail -f /dev/null` (Start in background use `-it` for interactive mode)
- ❖ `docker exec -it ulf-ubuntu /bin/bash` (Connect a session. Remember no ssh)
- ❖ `docker ps -a` (Shows running and exited container)
- ❖ `docker images` (Shows installed images)

A very tiny bit about Networking

- ❖ Every running container get's a virtual ip that can be used for forwarding rules around connectivity to the outside world.
- ❖ To find the ip of a running container you could use
 - ❖ `docker ps -a` (Find the CONTAINER ID)
 - ❖ `docker inspect <CONTAINER ID>`

Persistence..

- ❖ Docker is persistent on image level not on the container level.
- ❖ On container level changes lives between stop and start of named container but is lost as soon as you cleanup you named container from the ps list.

Persistence demo

```
docker run -d --name ulf-ubuntu ubuntu tail -f /dev/null or (docker start ulf-ubuntu if the container is named and already in the ps list)
docker exec -ti ulf-ubuntu /bin/bash
vim (no vim installed so will not work)
apt-get update
apt-get install vim
```

vim works!

```
docker run -d --name ulf-ubuntu1 ubuntu tail -f /dev/null
docker exec -ti ulf-ubuntu1 /bin/bash
vim (No vim available, vim only installed on the first running container)
```

```
docker stop ulf-ubuntu
docker stop ulf-ubuntu1
```

```
docker start ulf-ubuntu
docker exec -ti ulf-ubuntu /bin/bash
vim (should work, since vim installed in the first container)
```

```
docker stop ulf-ubuntu
docker rm ulf-ubuntu
docker run -d --name ulf-ubuntu tail -f /dev/null
docker exec -ti ulf-ubuntu /bin/bash
vim (vim is no longer since only installed on container but never committed to the image)
```

Persistence..How to get it glued?

- ❖ `docker run -d --name ulf-ubuntu ubuntu tail -f /dev/null`
- ❖ `apt-get update`
- ❖ `apt-get install vim`
- ❖ verify it works!
- ❖ Exit the machine and then commit (image level) the changes with.
 - ❖ `docker commit -m "ulf-ubuntu" `docker ps -l -q` ubuntu`

Power means responsibility.

- ❖ Since persistence comes with commit you can plan ahead generating demo containers that always get's to a 0-state when stopped.

Persistence and Database

- ❖ Changes to databases are lost if the physical files are stored within the container and the container is removed after stop.
- ❖ This is where "Virtual filesystem" comes in... You can map a filesystem on the host with the "-v host directory:containerdirectory" flag.
- ❖ Storing the database on virtual filesystem means it is stored outside the container and we get a database that remember the changes between remove of the container and changes of the image or even between binary compatible images.

Oracle docker image

- ❖ Regular updated
- ❖ Supports more then just Oracle and MySQL databases (11.2.0.2,12.1.0.2,12.2.0.1 and XE when it comes to the database)

Demo Oracle 12.2 on Docker

- ❖ Download the repo from GitHub and read the README.md for the OracleDatabase
- ❖ Download the Oracle binaries from OTN and place the zip file in the OracleDatabase / dockerfiles / <version> subcatalog
- ❖ Run the buildDockerImage.sh script
- ❖ Start the docker in interactive mode to build the database
- ❖ Stop the container and start as a daemon with port forwarding for the Oracle listener.

DEMO

- ❖ Quick demo (Pre-recorded to save time to build and start the container)
- ❖ How to connect from the outside world thru sqlnet. (If you know port forwarding in VirtualBox or VMware you know what to do and how to get it to work)

Examples

- ❖ Start a Oracle 12c container after the database been created
- ❖ `docker run -d --name <name> -p 1522:1521 -v <host path>:/opt/oracle/oradata oracle/database:12.2.0.1-ee`

Examples

- ❖ Connect to a Oracle 12c container after the container has been started.
- ❖ `docker exec -ti <name> /bin/bash`

Examples

- ❖ Commit changes to a Oracle 12c container after the container has been modified by adding or changing something within the container.
- ❖ `docker commit -m "<name>" `docker ps -l -q` oracle/
database:12.2.0.1-ee`

Examples

- ❖ `docker -log <name>` Gives dmesg log of the machine
- ❖ `docker -top <name>` Monitoring
- ❖ `docker ps -a` Checks for running and previous used containers
- ❖ `docker rm <name>` cleanup previous used containers
- ❖ `docker rmi <image id>` remove an image completely
- ❖ `docker —help <parameter>` (For the rest)

Q&A

❖ Questions ?