Mittuniversitetet
MID SWEDEN UNIVERSITY

Administration of UNIX-like systems
Containers and Docker

Lennart Franked

20 november 2023

Informationsystem and Technology
Mid Sweden University

Mittuniversitetet
MID SWEDEN UNIVERSITY

Containers

1. Containers

2. Docker

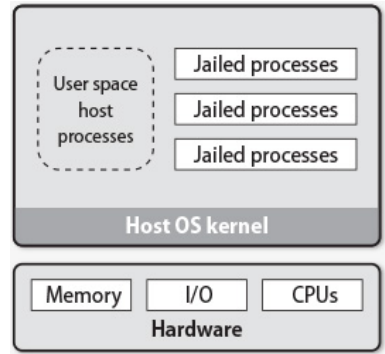
Mittuniversitetet
MID SWEDEN UNIVERSITY

Introduction

- OS-level Virtualization
- Does not use a hypervisor, relies on kernel
- Isolating processes
- Low resource overhead.

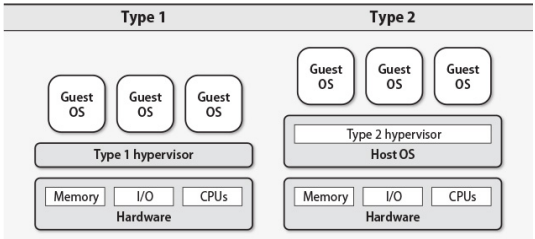
Containerization

- Private namespace
- Shares the kernel with the host



Figur: Containerization [1, Ch. 24 p.905]

Virtual Machine vs Container



Figur: Comparing virtual machines with containers [1, Ch. 24. p.906]

Containers usage scenarios

- Applications might need a lot of dependencies to run.
- Specific version of the interpreter
- Requirement of other softwares
- Other software/services that is run on the same machine, might have conflicting dependencies, which might make it difficult to run them on the same host.

Mittuniversitetet
MID SWEDEN UNIVERSITY

Containers usage scenarios II

- Popular to run applications in the cloud
- Easy to move an application from different machines
- If it is a service, more instances of the service might have to be created on demand.
- By keeping all the requirements packaged into one container, it allows us to easily move our applications between different hosts.

Namespaces

Namespaces

- Private namespace isolates the containerized process, e.g
  - Mount — private/Isolated root file system
  - User ID — Isolates user identification
  - PID — Isolate processes
  - Networking — Virtualize the network stack
- Possible to integrate different parts from the host system into the container.

Control Groups

Cgroups

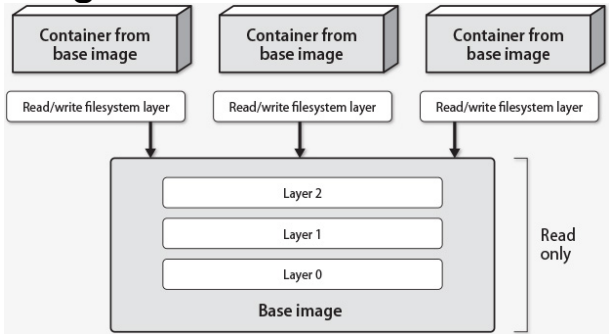
- Control Groups (cgroups) limits use of resources e.g
  - CPU
  - Memory
  - Disk
  - I/O
- Capabilities allows a container to access some kernel operations and system calls

Container Images

Container Image

- Template for a container
- Union filesystem organized to resemple the root file system
- Overlays multiple filesystem.

Container Images



Figur: Container overlays [1, Ch. 25, p.918]

Docker

Docker

Docker

- One of the more common container solutions used today.
- Server/Client application.
- `docker` — Managing docker.
- `dockerd` — Docker daemon.
- Can be run on the same system, or on separate systems.

docker and dockerd

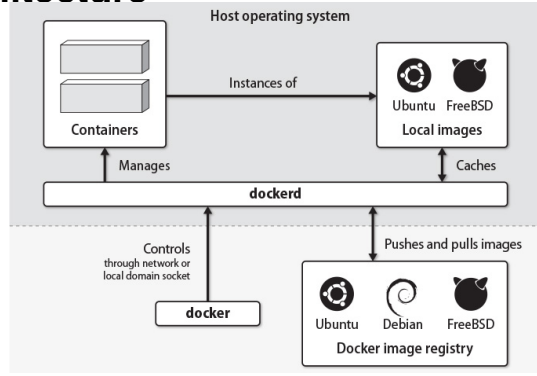
`docker`

- Executable command
- Management interface for dockerd

`dockerd`

- Responsible for creating and running images and containers,
- setup filesystem,
- networking.
- By default, docker images can be found at /var/lib/docker

Docker Architecture



Figur: Docker architecture [1, Ch. 25 P. 920]

Docker CLI

Subcommand	What it does
docker info	Displays summary information about the daemon
docker ps	Displays running containers
docker version	Displays extensive version info about the server and client
docker rm	Removes a container
docker rmi	Removes an image
docker images	Displays local images
docker inspect	Displays the configuration of a container (JSON output)
docker logs	Displays the standard output from a container
docker exec	Executes a command in an existing container
docker run	Runs a new container
docker pull/push	Downloads images from or uploads images to a remote registry
docker start/stop	Starts or stops an existing container
docker top	Displays containerized process status

Figur: Docker commands [1, Ch. 25 P. 921]

Docker CLI Examples

List Docker Images

```
root@host:~# docker image ls
REPOSITORY          TAG       IMAGE ID         CREATED        SIZE
ubuntu/bind9        latest    ba2f61203e87     8 days ago     131MB
root@host:~#
```

Access to prompt in a container

```
root@host:~# docker exec -it bind9_cali /bin/bash
root@647871d44ece:/#
```

Storage

- Container image contains mostly for static files.
  - Application code, libraries, supporting OS files
- UnionFS allows containers to read/write
- Difficult to administer
- Ephemeral storage
  - Tied to the life cycle of the running container

Volumes

- Used for persistent storage that is used and generated by docker
- Configuration files
- User data
- Mounting a path into the container
- `-v /var/docker/bind9_cali/etc/bind:/etc/bind`

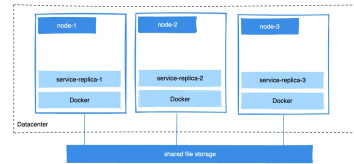
Mittuniversitetet
MID SWEDEN UNIVERSITY

Volumes

- Easy to backup and migrate
- Can be placed on remote hosts
- Easy to manage

Sharing volumes

- Volumes can be shared between containers
- Commonly used for replicating services
- All replicas need to have access to same information



Figur: Share data between machines [2]

Networking

Network drivers

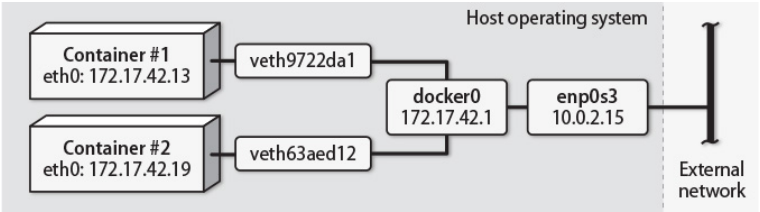
- Bridge
- Host
- IPvlan
- Macvlan
- Overlay
- None

Networking

Bridge

- Link layer device
- Software bridge
- Allows containers to communicate with other containers in same bridge.
- vSwitch
- Docker host will route to outside network.

Docker Network Architecture



Figur: Docker bridge network [1, Ch. 25 P. 926]

Networking

Host

- Let's the container share the host network stack.
- No isolation
- No need for NAT or proxy
- Better performance

Networking

IPvlan

- Share the same MAC
- Gives each container their own IP
- Each container have their own routing table.

Macvlan

- Unique MAC per container.
- Unique IP per container.
- Compatible with 802.1q trunks

Networking

Overlay

- Distributed network over multiple docker hosts
- For example sharing a bridge between multiple hosts
- Used when running Docker in swarm mode

None

- No network
- Container have a local purpose
- E.g working with files available on volumes

Mittuniversitetet
MID SWEDEN UNIVERSITY

Referenser

[1] Evi Nemeth m. fl. *Unix and Linux system administration handbook*. Fifth edition. Boston: Addison-Wesley/Pearson, 2017. ISBN: 9780134277554.

[2] *Share data between machines*. Accessed: 2023-11-15. URL: https://docs.docker.com/storage/volumes/#share-data-between-machines.

Mittuniversitetet
MID SWEDEN UNIVERSITY