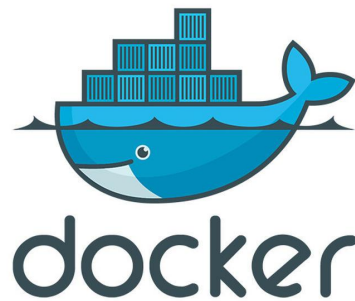




How its made: Docker



A presentation by:
Joar Rutqvist and Pontus Broberg



Content

- What is Docker?
- Docker, “Under the hood”
 - Implementation
- Docker vs VM
- Summary



What is Docker?



What is Docker?

- Open source, Linux based tool
- Uses “Containers” to package your application
- Your application can run on any other Linux machine



Docker, “Under the Hood”



How would you implement docker?

- Same OS
- Separated programs
- No access to host OS

A good start: Processes

- Some isolation
- Good support for resource distribution
- Problem: Shared OS and configuration

Example: Filespace isolation

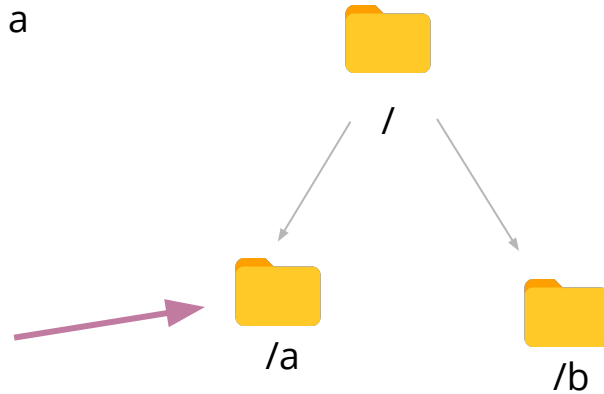
A: Creates a file "**file.txt**" in /

B: Writes a lot of gibberish into any file named **file.txt** in /

Isolating file systems

We can change what a process sees as `/`

Crucially only **a** thinks that `/a` is actually `/`



Before running it we "trick" **a** that **this** is `/`

We may now trick **b** that `/b` is `/`, making it unable to access `/a/file.txt`

We have split **a** and **b**'s file handling into separate **namespaces**

Namespaces: What you can see

- File system *Each container only sees its private sub-filesystem*
- Process IDs
- Users
- Networking
- ...

We also need: Control Groups

Limiting computer resources

- CPU *"give each container 10% of the CPU only"*
- Memory
- Disk I/O
- Device permissions
- ...

WOULD YOU LIKE TO KNOW MORE?

Please check out the talk **Building a container from scratch in Go** by **Liz Rice** from the conference **Container Camp** (you will get it even if you haven't used Go) .

There you will get a quick look into how these ideas might be implemented using the Go programming language.

The video player shows a Go code snippet for a container runner. The code is as follows:


```
1 package main
2
3 import (
4     "fmt"
5     "os"
6     "os/exec"
7 )
8
9 // docker run <container> cmd args
10 // go run main.go run cmd args |
11 func main() {
12 }
13
14 func must(err error) {
15     if err != nil {
16         panic(err)
17     }
18 }
19
```

The video features Liz Rice, a speaker at Container Camp, presenting the talk. The video player interface includes a progress bar at 2:46 / 3:09, a thumbnail of the speaker, and the Container Camp logo. Below the video, the title 'Building a container from scratch in Go - Liz Rice (Microscaling Systems)' is displayed, along with engagement metrics: 80,242 views, 1,916 likes, and 48 comments. The video was uploaded on October 13, 2016. The Channel Name is 'Container Camp' with 4,640 subscribers. A 'PRENUMERERA' button is visible in the bottom right corner.

Building a container from scratch in Go - Liz Rice (Microscaling Systems)

80 242 visningar • 13 okt. 2016

1 916 48 DELA SPARA ...

 Container Camp
4 640 prenumeranter

PRENUMERERA

Docker vs VM

Docker vs Virtual Machine

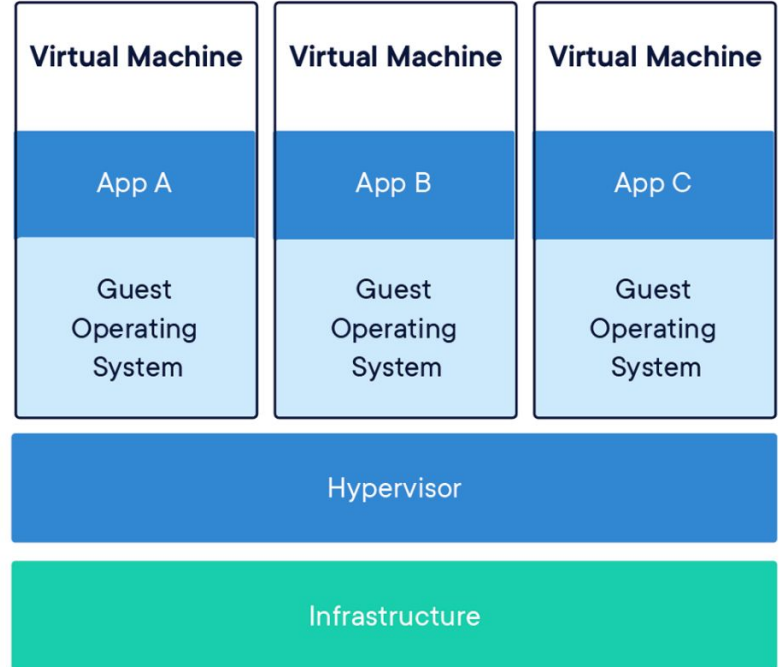
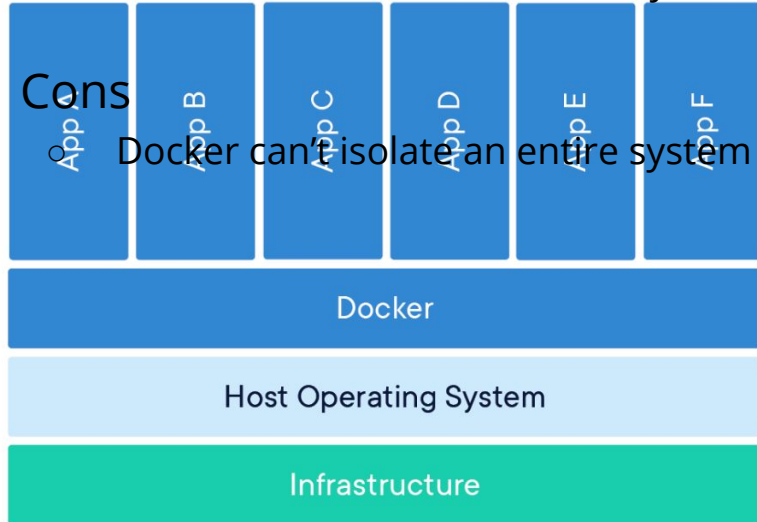
src: https://www.docker.com/resources/what-container#/package_software

- Pros

- Containerized Applications
- Docker is much faster than a VM
- Docker uses much less memory than a VM

- Cons

- Docker can't isolate an entire system



Summary

Summary

Docker provides containers

Container environments are identical everywhere

This is achieved by isolating



Thank You For Listening!

