intentionally left
about: blank

Andreas Tiefenthaler

Twitter/GitHub: **@pxlpnk**

contentful.com
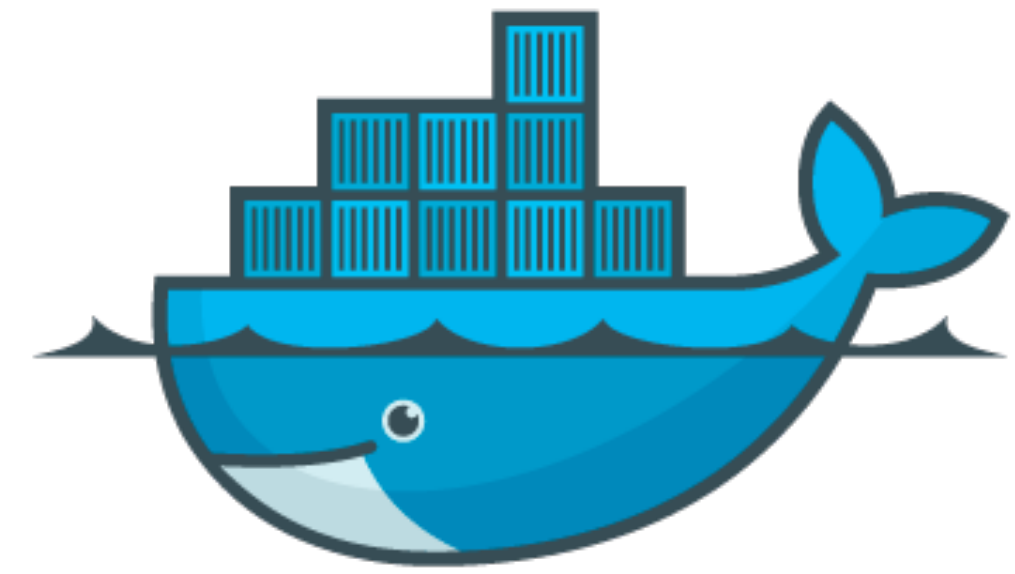
the shipping

What is this about?

- Docker in theory
- technology
- tiny live demo
- that to use it for
- that to not use it for
- some code
- bigger live demo

Docker is an **open-source** engine that
automates the deployment of any application
as a **lightweight**, **portable**, self-sufficient container
that will **run** virtually **anywhere**.

written in **go**-lang

Main Features

Can **encapsulate** any **payload**
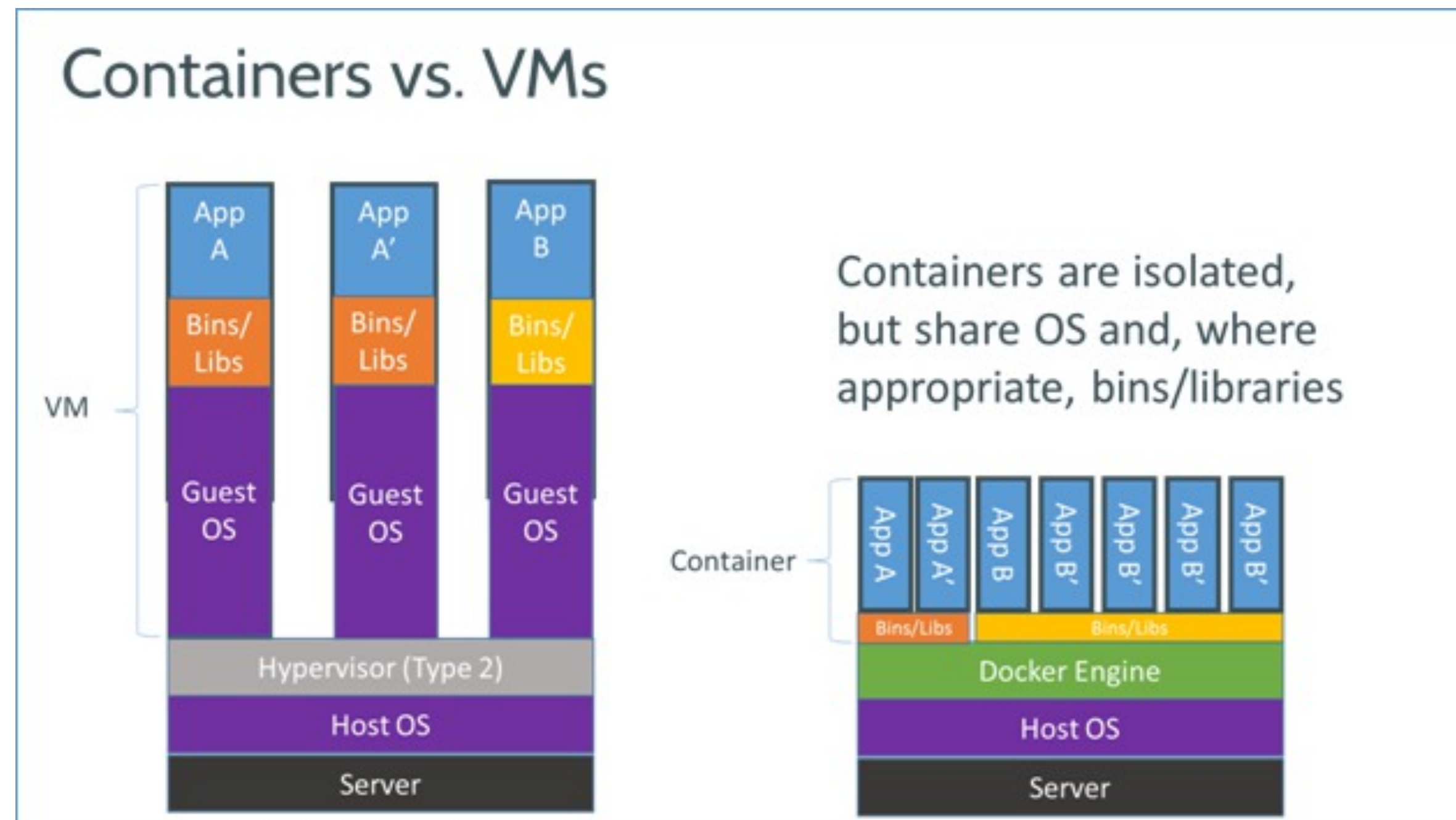and its **dependencies**

Lightweight, virtually no performance or start-up penalty, **quick** to **move** and **manipulate**

Separation of duties
code vs infrastructure

Resource, network, and content **isolation**.
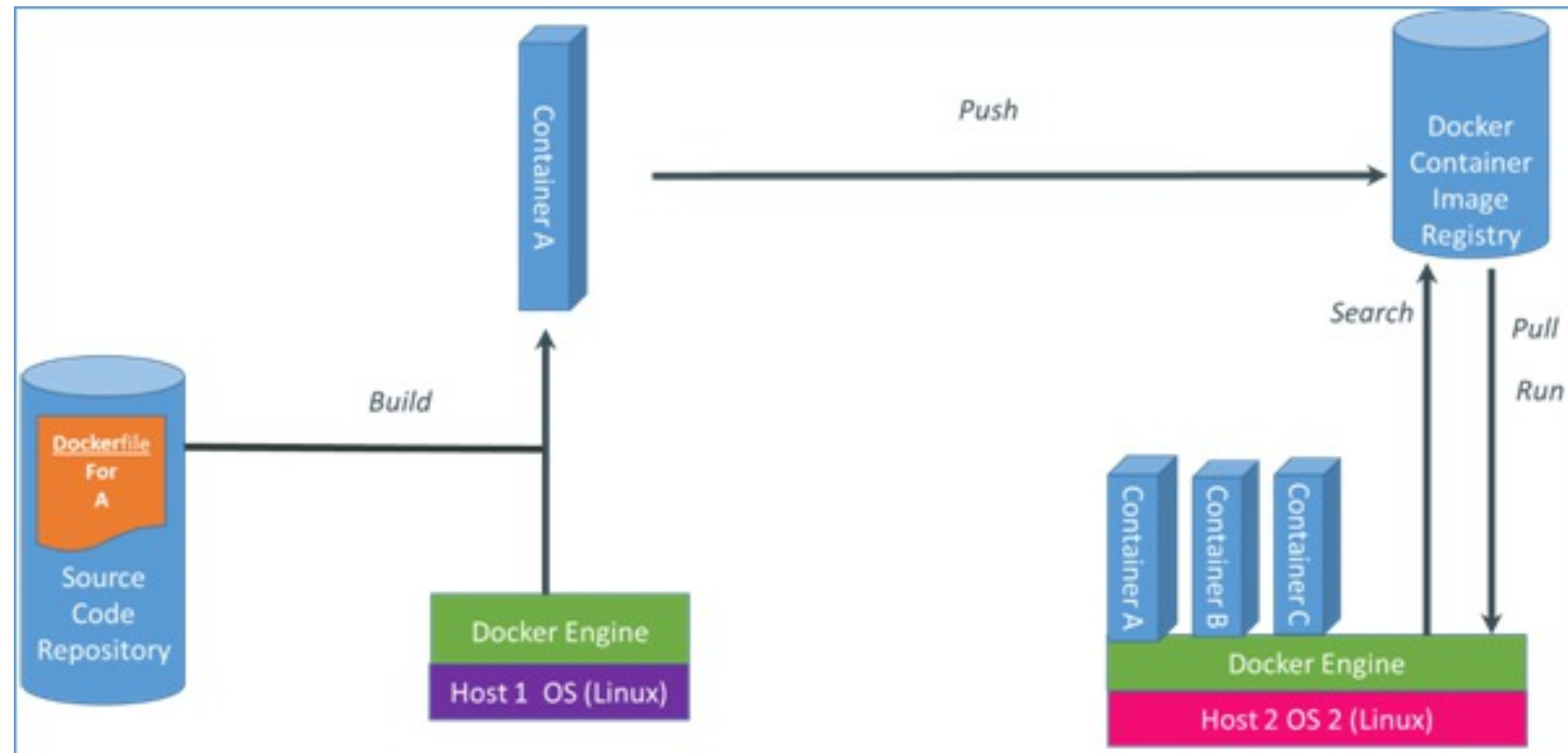Avoids dependency hell

Containers vs VMs

Container?

- Uses Linux Containers - LXC

- cgroups

- Linux Kernel itself

Basic Functions

- build
- modify
- publish
- search
- run containers

First demo:

```
docker run ubuntu /bin/echo hello world
```

Use cases

Automating the packaging and deployment of applications

Creation of lightweight, private PAAS environments

Automated testing and continuous integration/deployment

Deploying and scaling web apps, databases and backend services

Who uses docker?

BLITLINE
IMAGE PROCESSING IN THE CLOUD

ebay

https://flynn.io/

Yandex

wercker

CLOUDFLARE

Travis

rackspace

Some Code

Dockerfiles

```
FROM         ubuntu:12.10
RUN          apt-get -y install htop
ENTRYPOINT   ["/bin/bash"]
```

```
$docker build -t container .
```

Demo

The APP

The APP

- Sinatra Webservice Container

- Sidekiq Worker Container

- Redis Container

Webapp

```ruby
require_relative 'config.rb'
require_relative 'worker'

require 'sinatra'

set :environment, :production unless
ENV['development']

get '/:task/:cost' do |task, cost|
  Worker.perform_async(task, cost)
end
```

Worker

```ruby
require_relative 'config.rb'
require 'sidekiq'

class Worker
  include Sidekiq::Worker

  def perform(task = 'HardTask', cost = 2)
    puts "Working on #{task} for: #{cost}"
    sleep cost.to_i
  end
end
```

Config

```ruby
DB_URL = ENV['DB_PORT_6379_TCP_ADDR']
DB_PORT = 6379
DB_NO = 0
DB_NAMESPACE='catcher'

Sidekiq.configure_client do |config|
  config.redis = { namespace:  DB_NAMESPACE, size: 1,
    url: "redis://#{DB_URL}:#{DB_PORT}/#{DB_NO}" }
end

Sidekiq.configure_server do |config|
  config.redis = { namespace: DB_NAMESPACE,
    url: "redis://#{DB_URL}:#{DB_PORT}/#{DB_NO}" }
end
```

DEMO

Cons

- **Not** production ready

- **Rapidly** changing

- Should not run Database within it

- Does not give you Security Superpowers
  (http://blog.docker.io/2013/08/containers-docker-how-secure-are-they/)

v.0.8

- BTRFS support
- OS X support
- one release per month
- new builder features (Dockerfiles)
- general faster faster

@langziehohr

Danke, thank you, dankjewel

Questions?

Slides: http://an-ti.eu/talks.html

Docker: https://www.docker.io/

Source: https://github.com/pxlpnk/docker

Diagrams taken from docker.io