

# CONTAINERISE EVERYTHING

why you should stop installing  
development environment locally





**I'M DANIEL DEMMEL (@DAAAAN)**

*I work at*  
**WISWU**

*We design, build and launch  
digital products*



# TALK OUTLINE

- A brief history of the shipping industry
- What is containerisation, why do we need it and how it will change software development
- A practical introduction to Docker



*Handling bulk  
goods used to be  
messy and labour  
intensive...*





*...until Malcolm McLean had enough by 1955 and came up with standardised, multi-modal containers.*





**BUT WHY DO I  
CARE ABOUT THE  
SHIPPING  
INDUSTRY?**





# CONTAINERISATION BROUGHT US GLOBALISATION



# SHIPPING CONTAINERS PROVIDE:

- Encapsulation
- Standardised interfaces
- Easy scaling





**:SHIPIT:**



**SO WHAT IS THE  
PROBLEM WITH  
DEVELOPMENT  
ENVIRONMENTS?**



*This is your  
development  
machine with the  
dependencies  
for all projects,  
tools, libraries  
and versions  
installed on it:*

**IO.JS 2.4**  
**JAVA 6** **NODE 0.8**  
**PHP 4** **MYSQL 4**  
**LESS 2.5**

**SASS 3.4**  
**RUBY 2.1**  
**POSTGRESQL 9**

**HERE BE DRAGONS**

**LIBSASS 3.2**  
**NODE 0.12**  
**JAVA 7**  
**RUBY 1.9**  
**PHP 5** **MYSQL 5**  
**IMAGEMAGICK 6**



*But it could be  
neat like  
Malcolm's port:*



...

**JAVA 6**

**RUBY 1.9**

**PHP 5**

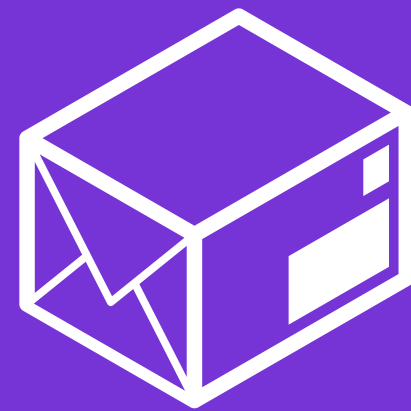
**NODE 0.8**

**POSTGRESQL 9**

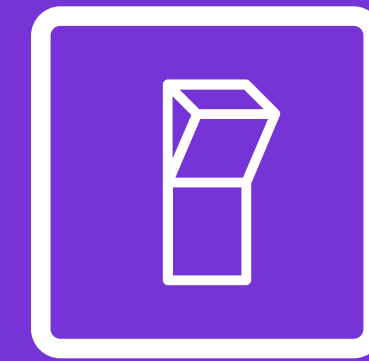
**MYSQL 5**



# APPROACHES TO HANDLE DIFFERENT ENVIRONMENTS



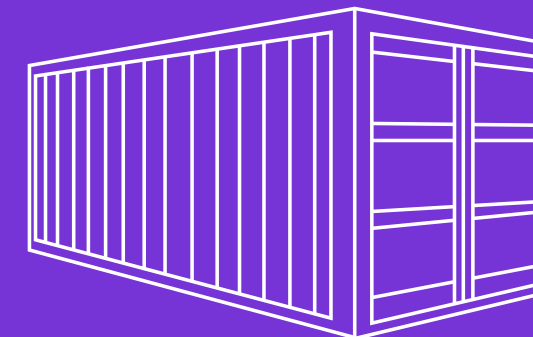
Package managers



Version switchers



Virtualisation



Containerisation





# PACKAGE MANAGERS

- Libraries installed locally for each project
- Easy to forget including existing globally installed tools into project package list
- Falls apart with legacy projects
- For ex: Bundler, NPM, Maven





# VERSION SWITCHERS

- Picks you the right runtime version to execute your code in per folder / project
- Gets messy with a fair amount of magic to make things happen (and go wrong)
- For ex: RVM, Rbenv, NVM, JSelect

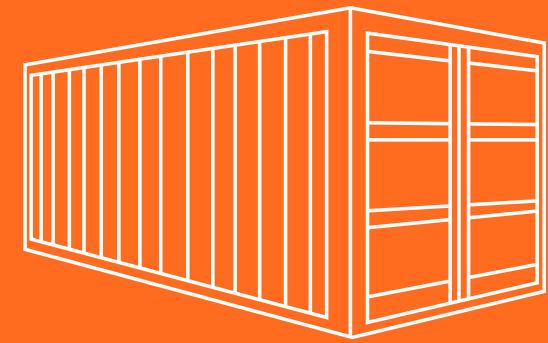




# VIRTUALISATION

- Totally independent virtual operating systems
- Can perfectly mirror production environment
- Mostly scripted dependency installation
- Portable image snapshots
- For ex: Virtualbox, Vagrant, Puppet, Chef, Salt

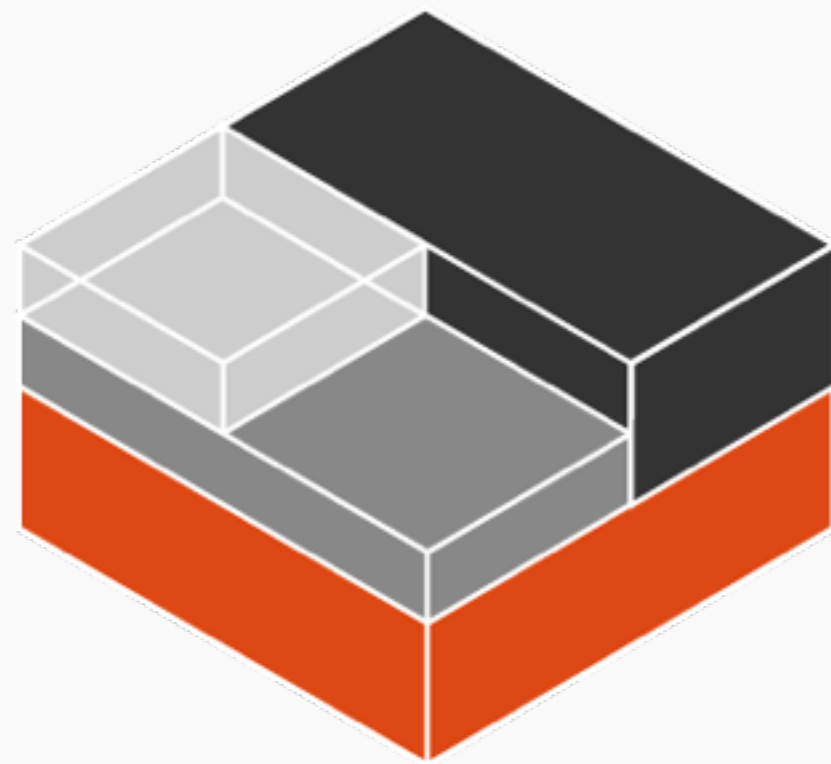




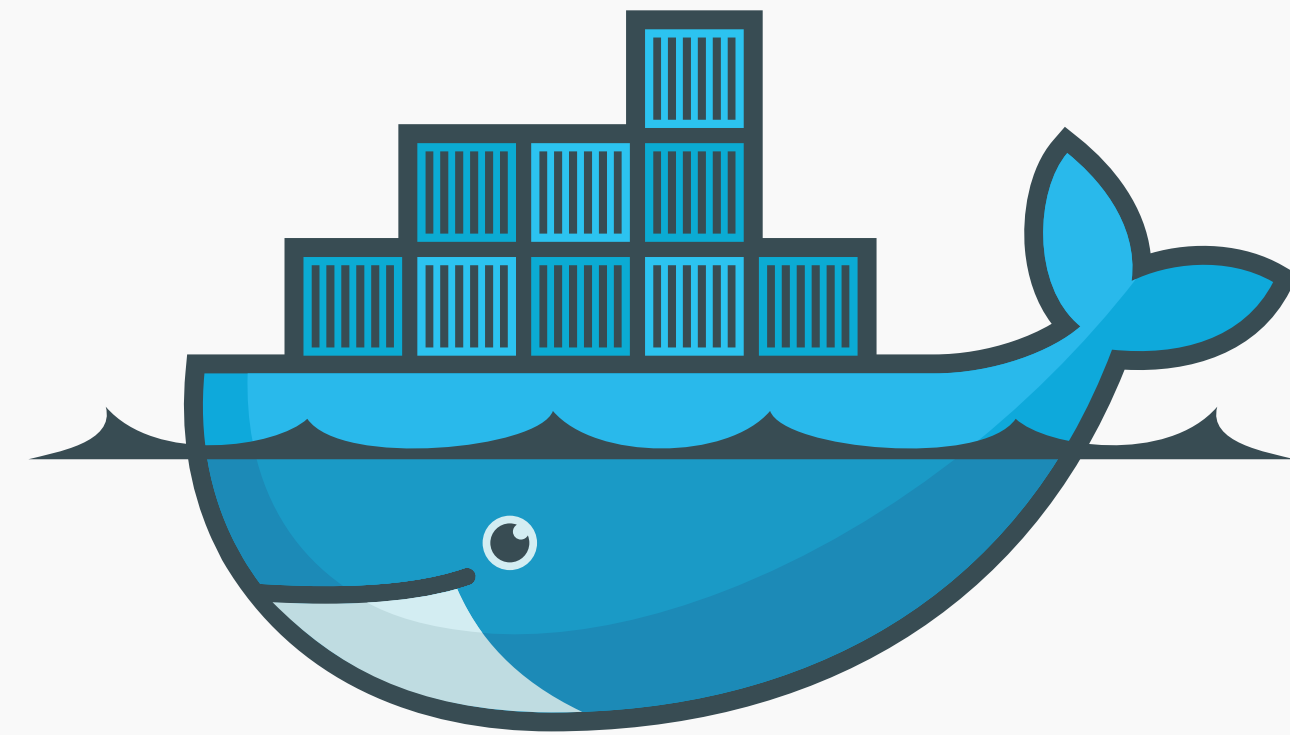
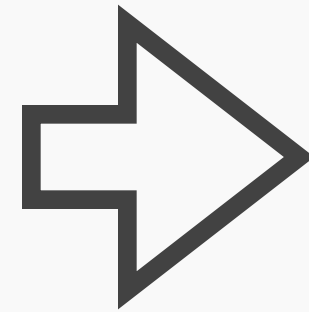
# CONTAINERISATION

- Isolated environments reusing host OS core
- Much smaller and no overhead of virtualisation
- Pretty much instant creation and restart
- For ex: LXC, Docker, Open Container Initiative





**LXC**



**docker**



# DOCKER ECOSYSTEM

- Open source with big and active community
- Addresses several levels of the problem
- Makes containerisation accessible to all



# DOCKER (ENGINE)

- Abstraction and UX around Linux Containers
- REST API to interact with CLI and remotes
- Dockerfile as build recipe
- Portable and incremental images
- Uses Virtualbox + Boot2Docker on Win / Mac



# DOCKER HUB / REGISTRY

- Github of Docker with public and private repos
- A place to store / share container snapshots
- Fork or build on top of existing containers
- API hooks to integrate build / testing pipelines



# KITEMATIC

- A simple GUI to see containers created locally and available on Docker Hub
- Installs everything you need to get started
- Mac only, but Windows version is coming
- Not that useful if you want to customise or build on containers, little info shown in search



# DOCKER MACHINE

- Makes it easy to manage containers and hosts
- Works both locally and with cloud providers
- A quick way to do simple deploys

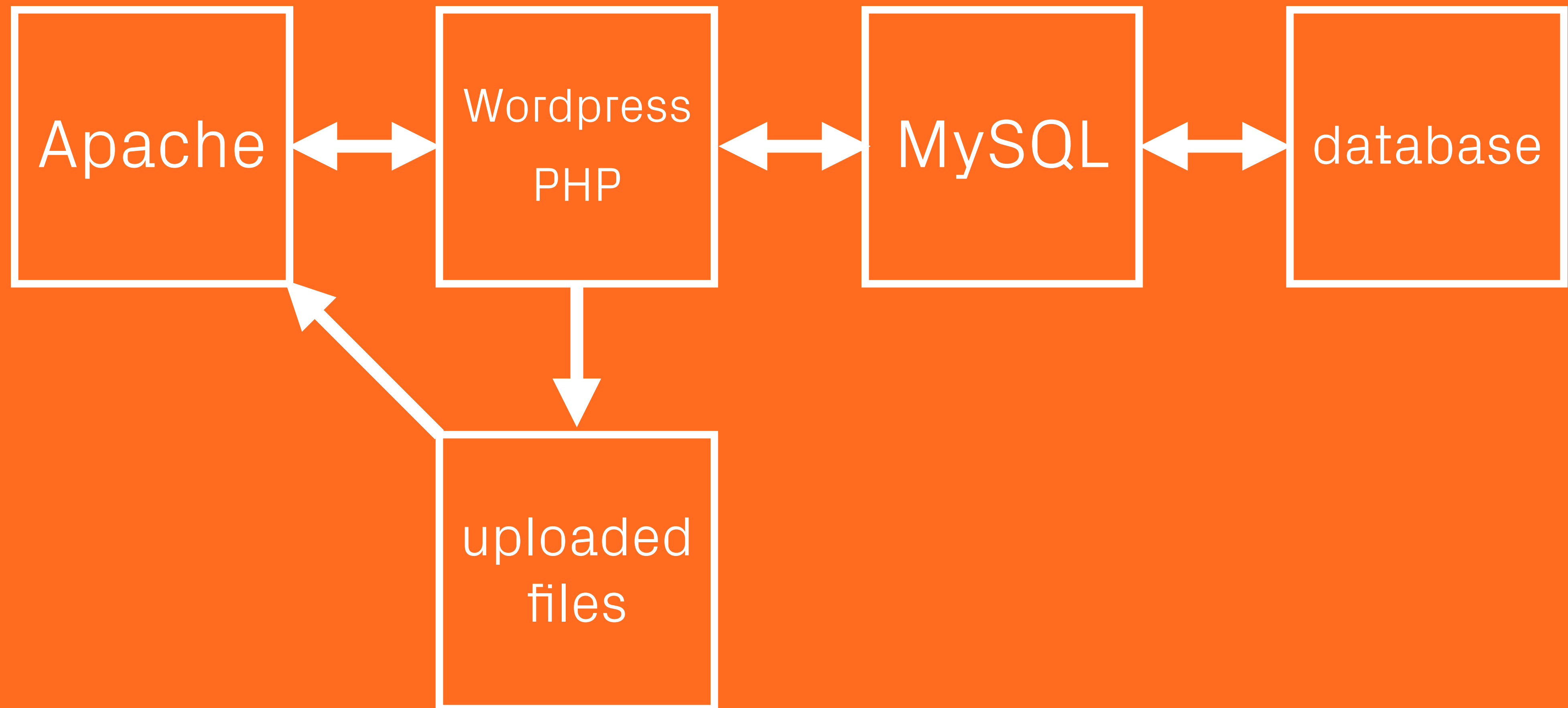


# DEMO (FINALLY!)

- Walkthrough of Docker boilerplate repo
  - React + Sass front end app
  - Express / Node server
  - Gulp build chain
  - package.json, Dockerfile, Makefile
- Terminal + browser demo
- Kitematic



# MICROSERVICES





“ Containerisation will start a similar **revolution in open source** as Git did, by making it much simpler to take any code and start compiling it right away



# QUESTIONS?

- Yes, we're hiring, do get in touch: [dain@ustwo.com](mailto:dain@ustwo.com)
- Links
  - Talk: <http://www.danieldemmel.me/blog/2015/08/14/containerise-everything/>
  - Git repo: <https://github.com/ustwo/docker-node-boilerplate>
  - Dockerfile best practices: [https://docs.docker.com/articles/dockerfile\\_best-practices/](https://docs.docker.com/articles/dockerfile_best-practices/)



# CREDITS

- Talk is Creative Commons Attribution 4.0 International
- Image credits as attributed on respective slides
- Vector icons from the Noun project
  - **Package:** <https://thenounproject.com/search/?q=package&i=10996>
  - **Switch:** <https://thenounproject.com/search/?q=switch&i=79501>
  - **Virtualisation:** <https://thenounproject.com/term/data-center-server-virtualization/95507/>
  - **Container:** <https://thenounproject.com/term/shipping-container/116963/>