



# DOCKSAL

More Dev / Less Ops



# Hi there!



**I'm a Technical Lead at Kanopi Studios.**

@seanedietch / sean\_e\_dietrich / sean.e.dietrich

Maintainer on the Docksal Project

Drupal Development > 10 years

First Time @ Drupal Corn



# What we will be going over...

- About Docksal
- Installing Docksal
- Adding Docksal to Projects
- Extending Docksal
- Demo
- Q&A



# What we won't be going over...

- Docker specifics/configurations - [docs.docker.com](https://docs.docker.com)
- Configuring CI/CD Services - CircleCI, TravisCI, Gitlab, Pipelines, etc.
- Setting Up Docksal Sandboxes - Contact Me
- Writing Bash Scripting - [bit.ly/bash-scripting](https://bit.ly/bash-scripting)
- Other similar software - (Lando, DDev, Lagoon, etc.)



# History



# Bare Metal

- Local servers, MAMP, WAMP, XAMPP, ...
- Hard to manage multiple versions
- Hard to synch with prod
- Can't share with other devs
- "Works for me"



# VMs

- VMs: Drupal VM, Acquia Dev Desktop, custom
- Monolithic
- Lots of disk, memory
- Maintenance is hard
- Big files, slow to provision



# Containers

- Smaller / Faster
- Modular
- Maintainable
- Shareable
- Portable
- Scalable



# Development Operations

DevOps



# What is DevOps?

**DevOps** is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes



# DEVOPS



What my friends think I do.



What my mom thinks I do.



What society thinks I do.



What my boss thinks I do.



What I think I do.



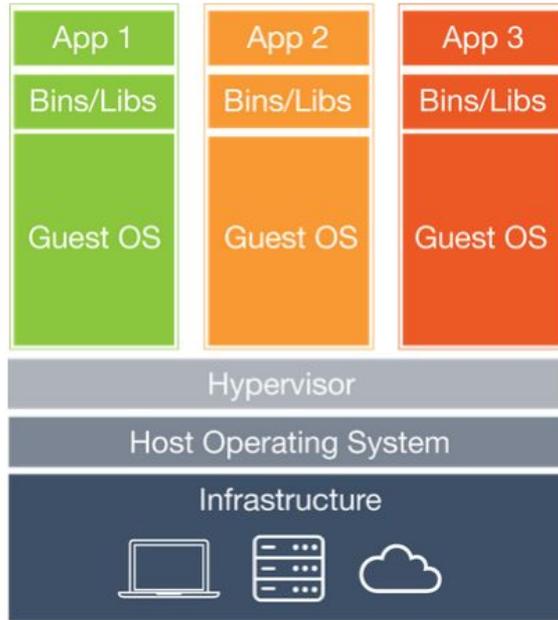
What I actually do.



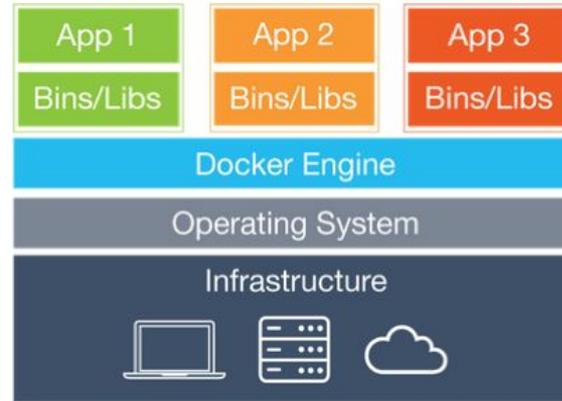
# Docker



# Why Docker?



Virtual Machines

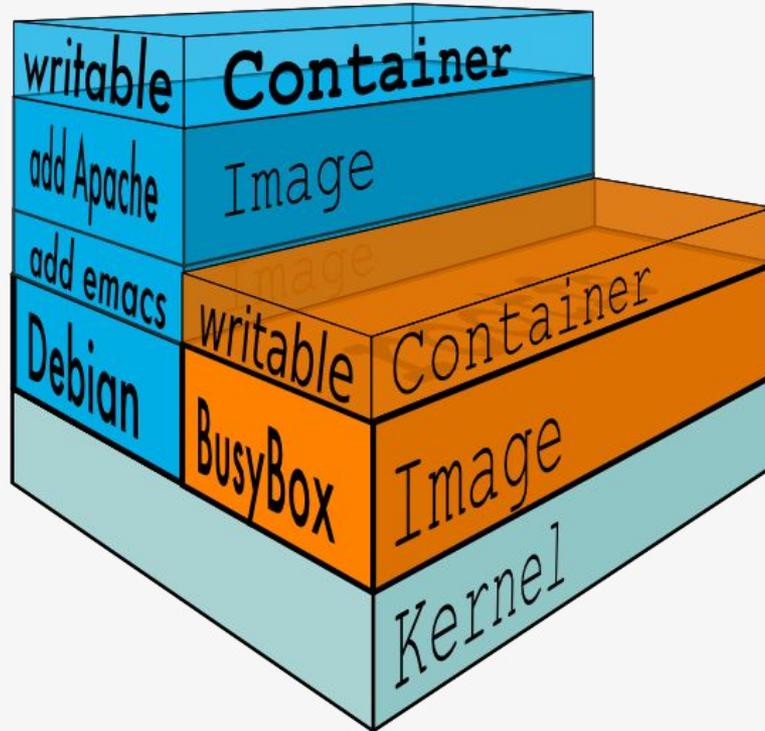


Containers

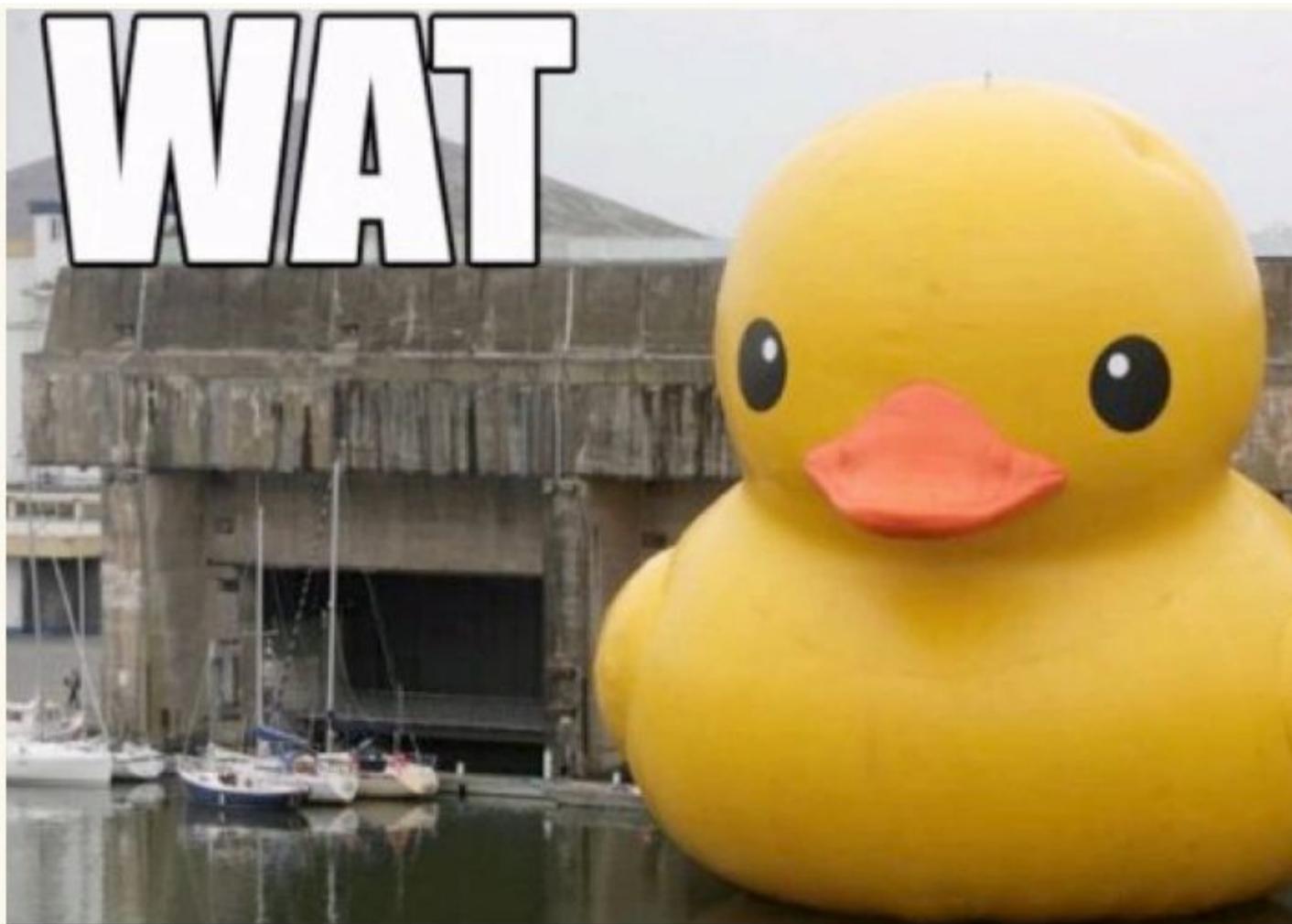




# Why Docker?



**WAT**



# Why Docker?

Installing

Configuration

Multiple Versions

Multiple Projects

Environment Templates



# Do you find yourself ...

Duplicating efforts

1 time install software

1+ hrs to onboard other devs

Too long to set up local environment



## Going Insane yet?

# What's Docksal?

## About:

- Helps manage dev environments
- Minimizes configuration
- Consistent
- Efficient Onboarding

## Features:

- Docker based
- Default AMP Stack
- Easy Installation
- Cross Platform
- Extendable



# Installing Docksal



# Installing

## MacOS / Linux

```
curl -fsSL https://get.docksal.io | sh
```

Using Vagrant? Read before installing: [bit.ly/docksal-vagrant](https://bit.ly/docksal-vagrant)

## Windows

Requires a Linux Subsystem: Babun (Recommend)

Ubuntu (Experimental) [bit.ly/docksal-windows-ubuntu](https://bit.ly/docksal-windows-ubuntu)



# Docksal with Projects



# Docksal with Projects

## Project Install Wizard

```
fin project create
```

## Existing Project

```
mkdir .docksal  
fin start
```



**Drupal 8 Composer - [bit.ly/2yFxufR](https://bit.ly/2yFxufR)**

**Other Boilerplates - [bit.ly/docksal-examples](https://bit.ly/docksal-examples)**

Why Docksal?  
Why not Docker?



# Commands w/ Docker

## **Docker4Drupal**

### **# Drush, Console, Composer:**

```
$ docker-compose exec --user 82 php drush status -r /var/www/html/web
$ docker-compose exec --user 82 php drupal list
$ docker-compose exec --user 82 php composer list
```

### **# Execute Command in PHP Container:**

```
$ docker-compose exec --user 82 php pwd
```

### **# MySQL shell:**

```
$ docker-compose exec mariadb /usr/bin/mysql -uroot -p"password"
```

### **# DB import/export:**

#### **# Import: Add my-db-dump.sql before bringing up container**

```
$ docker-compose exec mariadb sh -c \  
    'exec mysqldump -uroot -p"password" my-db' > my-db.sql
```

### **# Logs**

```
$ docker-compose logs
```



# Commands w/ Docksal

## # Drush, Console, Composer:

```
$ fin drush status  
$ fin drupal list  
$ fin composer list
```

## # Execute command in 'cli' container:

```
$ fin exec pwd
```

## # MySQL shell:

```
$ fin db cli
```

## # DB import/export:

```
$ fin db import db.sql  
$ fin db dump db.sql
```

## # Logs:

```
$ fin logs cli
```



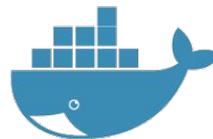
# Extending Docksal



# Tools & Services Available



AND MORE!



There's an image for that!

# Extending

## Commands / Addons

- Speed up our workflow
- Shareable
- Reusable
- Target Specific Environments

## Apache

- File Proxy vs. Stage File Proxy

## PHP

- Increase Memory
- Inject Database Credentials
- Increase File Post

## MySQL

- Enable Slow Query Logging
- Modify my.cnf



# Commands

Commands are scriptable actions that help simplify a project/developers workflow.

## Gulp Wrapper - [bit.ly/2Jc8ouR](https://bit.ly/2Jc8ouR)

```
#!/usr/bin/env bash
#: exec_target = cli
## Run Gulp Commands on the theme
## Usage: fin gulp [arguments]
# Abort if anything fails
set -e
cd ${PROJECT_ROOT}/web/themes/custom/site_theme
gulp "$@"
```



# Commands (cont.)

## Refresh Database - [bit.ly/2shwj2h](https://bit.ly/2shwj2h)

```
#!/usr/bin/env bash
#: exec target = cli
## Pull Database Using Drush (full reset of db)
## Usage: fin refresh [environment id]
set -e
ENVIRONMENT=${1:-prod}
cd /var/www/${DOCROOT}
DBFILE="/tmp/site.${ENVIRONMENT}.sql"
if [ ! -f $DBFILE ] || [ ! -z $(find $DBFILE -mmin +60) ] ; then
    if [ ! -f $DBFILE ]; then
        echo "Database file does not exist."
    elif [ ! -z $(find $DBFILE -mmin +60) ]; then
        echo "Database file is older then 1 hour"
    fi
    echo "Exporting ${ENVIRONMENT} Database..."
    drush @remote.${ENVIRONMENT} sql-dump > $DBFILE
fi
echo "Truncating Database..."
drush sql-drop -y
echo "Importing Database..."
drush sql-cli < $DBFILE
```



# Addons

<a href="#">adminer</a>	<a href="#">Adminer</a> database management tool	MySQL
<a href="#">codeclimate</a>	<a href="#">CodeClimate</a> code quality tool	
<a href="#">mailhog</a>	<a href="#">Mailhog</a> e-mail capture service for current project	
<a href="#">phpunit</a>	Creates a phpunit.xml file and runs PHPUnit tests	Drupal
<a href="#">pma</a>	<a href="#">PhpMyAdmin</a> database management tool	MySQL
<a href="#">redis</a>	Add <a href="#">Redis</a> to current project	
<a href="#">sequelpro</a>	Launches <a href="#">SequelPro</a> with the connection information for current project.	macOS
<a href="#">solr</a>	<a href="#">Apache Solr</a> search service for current project	
<a href="#">uli</a>	Generate one time login url for current site	Drupal

See more @ [github.com/docksal/addons](https://github.com/docksal/addons)



# K.I.S.S.

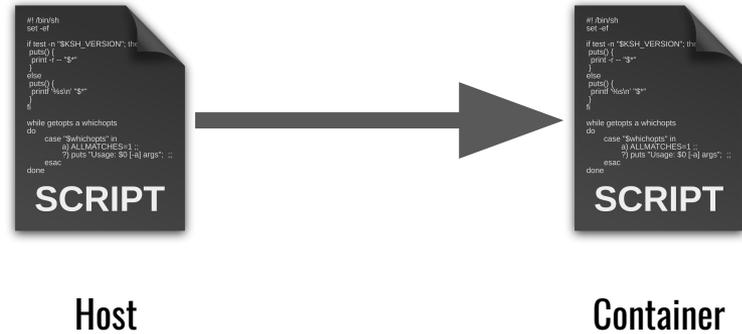
Keep It Simple Superhero

# YES YOU!

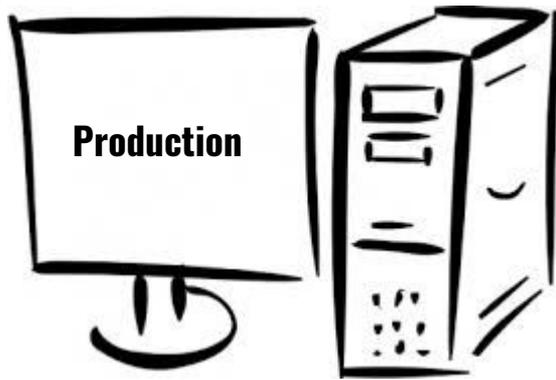


# One and Done ...

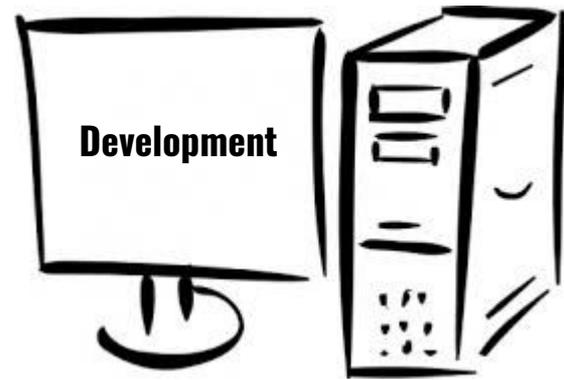
- Run `fin init`
- `init` is a command
- Runs steps
  - Destroy
  - Initialize
  - Install
  - Set Up



Goal!!!



=



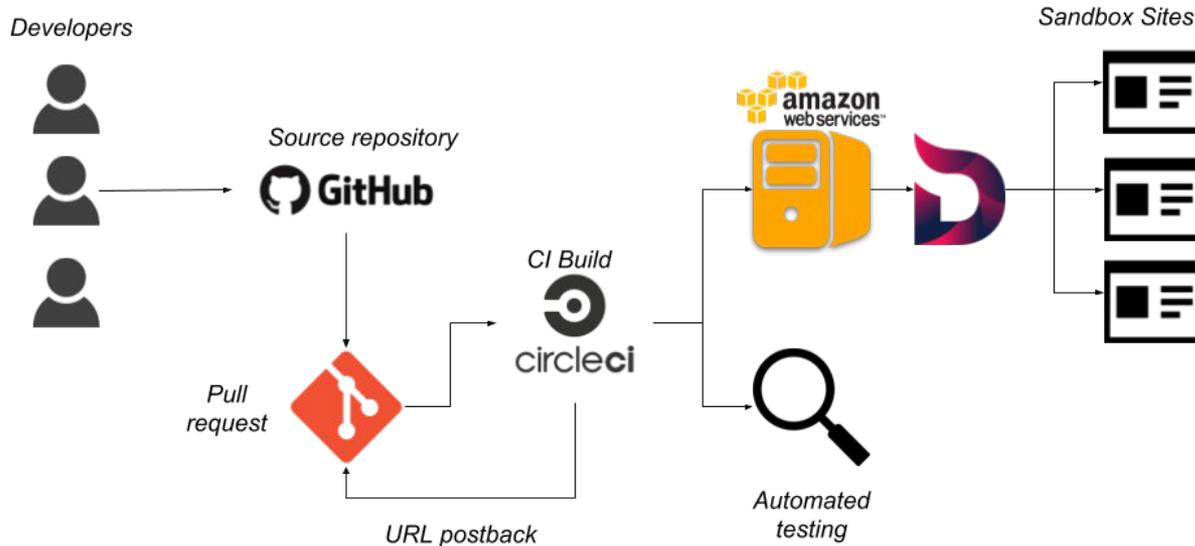
# Demos

1. Starting new project
2. Converting project from local AMP
3. Onboarding new developers



# Docksal and QA Delivery

- Enhanced code reviews
- Continuous QA
- Automated VRT/BDD testing
- Demos



# 2018 BADCamp

## SANDBOX CI/CD ENVIRONMENTS FOR EVERYONE

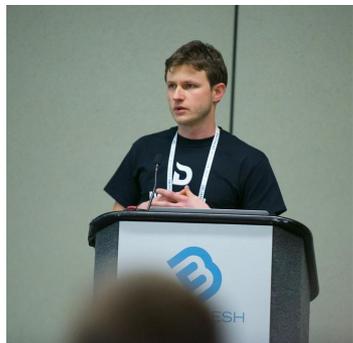
Leonid Makarov

<https://bit.ly/2ONgSYR>

If your project is hosted with one of the BIG-3 Drupal hosting providers, then your team is most likely already enjoying some sort of sandboxing solution provided by them. You commit code, a CI build runs, and a new sandbox (CD / multidev / on-the-fly) environment is magically provisioned for you. Life is good!

BUT What if your project isn't hosted on one of those hosting companies?

Every development team should be able to use these features in a simple and coherent way regardless of where their projects are hosted. How?



Questions?

