



Application Deployment and  
Management at Scale with 1&1

**Matt Baldwin**

CEO

@stackpointcloud

dockercon 16

Who is 1&1?

# What We Do

Customers push their own code, we manage the stack and infrastructure.

- **Secure, multi-tenant environment**
- **70,000 Machines**
- **Millions of Active Websites**
- **7 Global Data Centers**

# Our Challenges

“One Stack Fits All” approach to application hosting and management:

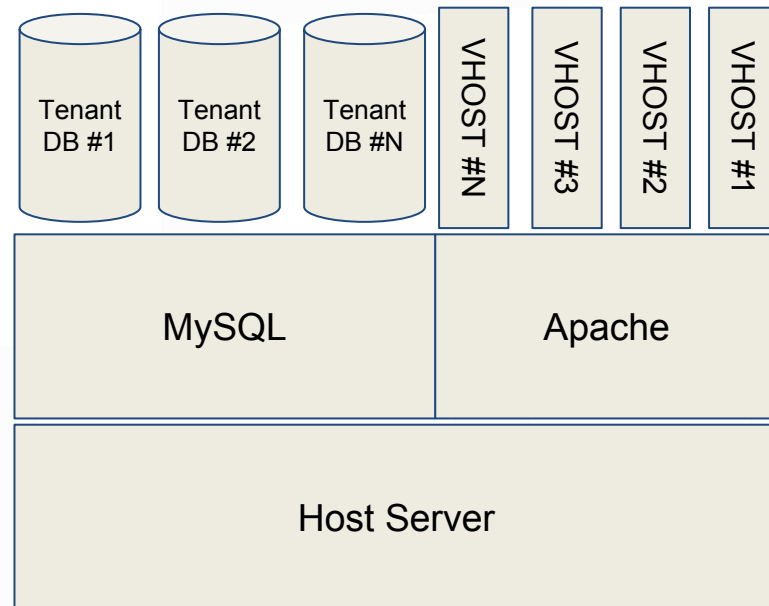
- **Rigid, monolithic**
- **Innovation is challenging**
- **Painful update process for shared components.**
- **Large effort to maintain several Linux kernels due to varied infrastructures.**

# 1&1's Future is Docker

# Current Architecture

Monolithic Hosting carves up resources and multi-tenants components.

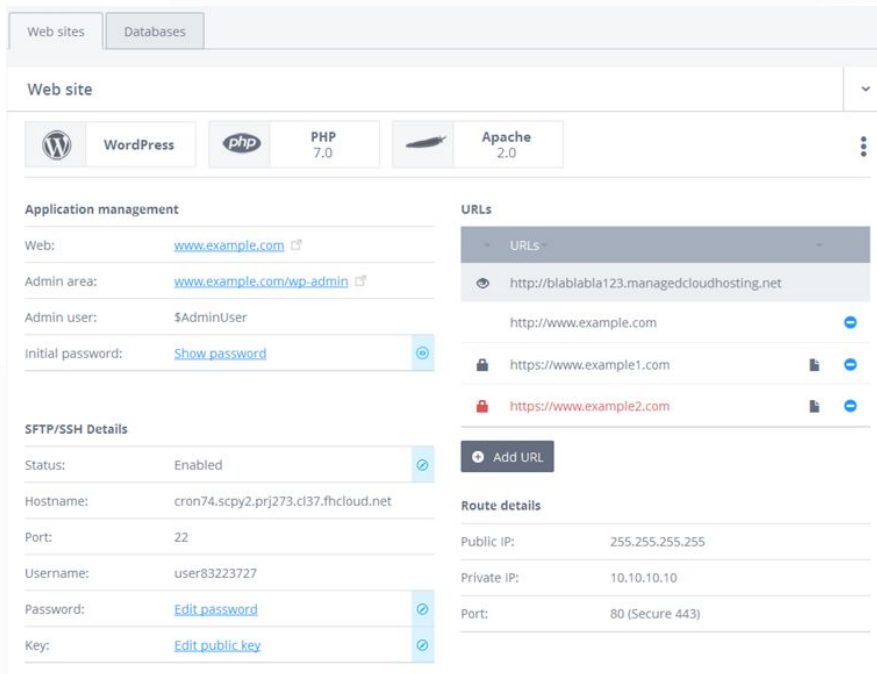
- **Single MySQL Instance**
- **Single Apache Instance**
- **Single Host**
- **Shared CPU and Memory**
- **All services access the same storage**



# Our Next Gen Goal

More choice, better scalability -- customer need drives our design goals.

- Richer Components - Caches, Queues, etc.
- Broader SQL and NoSQL support
- Reduce SPOF
- Workload portability
- Eliminate Resource Contention



The screenshot displays a cloud management interface for configuring a web site. At the top, there are tabs for 'Web sites' and 'Databases'. Below this, a dropdown menu shows 'Web site'. The configuration area includes several components: WordPress, PHP 7.0, and Apache 2.0. The 'Application management' section contains fields for 'Web' (www.example.com), 'Admin area' (www.example.com/wp-admin), 'Admin user' (\$AdminUser), and 'Initial password' (with a 'Show password' link). The 'SFTP/SSH Details' section shows 'Status' as 'Enabled', 'Hostname' as cron74.scpy2.prj273.ci37.fhcloud.net, 'Port' as 22, 'Username' as user83223727, 'Password' (with an 'Edit password' link), and 'Key' (with an 'Edit public key' link). The 'URLs' section lists 'http://blablabla123.managedcloudhosting.net', 'http://www.example.com', 'https://www.example1.com', and 'https://www.example2.com', each with a lock icon and a refresh icon. An 'Add URL' button is also present. The 'Route details' section shows 'Public IP' as 255.255.255.255, 'Private IP' as 10.10.10.10, and 'Port' as 80 (Secure 443).

# Future Architecture



We broke the monolith up: shared components are now containers:

- **Everything is a container.**
- **Mix and match web server and database solutions.**
- **Immediate component scalability.**
- **Components can be added ad-hoc.**
- **Leverage container ecosystem for solutions.**

Select custom configuration



Select managed database:

 None	<b>MariaDB</b>  Latest stable ▾	MySQL Latest stable ▾
---	---	--------------------------



# Our Solution

# Provide Choice

Containers allow us to quickly add curated components:







- **Customers create projects.**
- **Application stacks are either assembled or pre-built.**
- **Usual suspects are supported.**
- **Support scripting languages in use across all sites we host.**

## Create project

Name

New project

Select type of project

 WordPress ?	 Drupal 7.0	 Magento ?
 Stack 1 PHP 7.0 / Apache / MariaDB	 Stack 2 Ruby on Rails 4 / Phusion Passenger / MySQL	 Custom Define your own project configuration.

Select custom configuration

Language



Select scripting language:

 None	<b>PHP</b> 7.0 Latest stable	Ruby Latest stable	Python 3x Latest stable	Perl 5 Latest stable
---	---------------------------------	-----------------------	----------------------------	-------------------------

Name

My project

Configuration

Scripting Language:

Create

# Better Isolation

The traditional model forced us to multi-tenant the system via permissions, users, groups, and so on. Now:

- **User projects are deployed to their own dedicated resources.**
- **Strongly isolated networks.**
- **Strongly isolated storage.**
- **1&1 Manages Upgrades, Environment.**

The screenshot shows a 'SELECT CUSTOM CONFIGURATION' interface. At the top, there's a progress bar with four steps: Language, Framework, Web server, and Database. Below this is a 'Summary' section with the following details:

Scripting language:	PHP 7.0 - Latest stable
Web framework:	Zend 2 - Latest stable
Managed web server:	NGINX - Latest stable
Managed database:	MariaDB - Latest stable

Below the summary is a 'Select size' section with three options: 4XL, 5XL, and Flex. The Flex option is selected and highlighted in blue. The specifications for each size are as follows:

Size	vCore	RAM	SSD
4XL	12 vCore	32GB RAM	360GB SSD
5XL	16 vCore	48GB RAM	500GB SSD
Flex	4 vCore	8GB RAM	500GB SSD

Below the size selection, there are three sliders for CPU, RAM, and SSD, each with a numeric input field. The CPU slider is set to 4, RAM to 8, and SSD to 500. On the right side of the interface, there's a 'Management fee' section with a 'TOTAL' and a 'Create' button.

# Better Management

Our own management overhead decreases as usability and choice increases for our users.

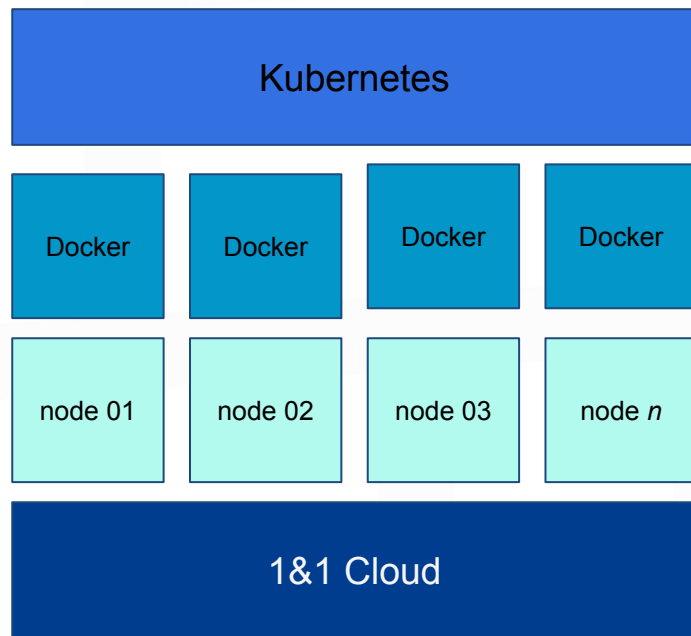
- **Upgrading components has no downtime.**
- **Users have more knobs -- regions, frameworks, kernel versions.**
- **Nodes management is now invisible.**

The screenshot displays a configuration interface for a cloud service. At the top, it says "Select custom configuration". Below this, there are four tabs: "Language", "Framework", "Web server", and "Database". The "Database" tab is selected, showing a progress indicator. Underneath, it says "Select managed database:" and presents two options: "MariaDB" and "MySQL". The "MariaDB" option is selected and highlighted in blue, with a "Latest stable" dropdown menu below it. The "MySQL" option also has a "Latest stable" dropdown menu. Below this section, there is an "Advanced options" section with a "Show" button. Under "Advanced options", there is a "Data Centre" section with five options: "Germany", "Spain", "Europe", "United States", and "United Kingdom". The "Germany" option is selected and highlighted in blue.

# Eliminate Failure Points

Treat hosts as fungible.

- Servers are cluster members.
- Containers are scheduled to nodes.
- Containers go to where resources are available.
- Components can be scaled independently of each other.
- Dynamic Cluster Routing



# Make Storage Easy

Since we manage the app, we manage the storage:

- **During an upgrade we move the volume to the new container.**
- **Docker images are tested and verified before user environments are updated.**
- **We perform and test the data migration.**

# Lifecycle Management

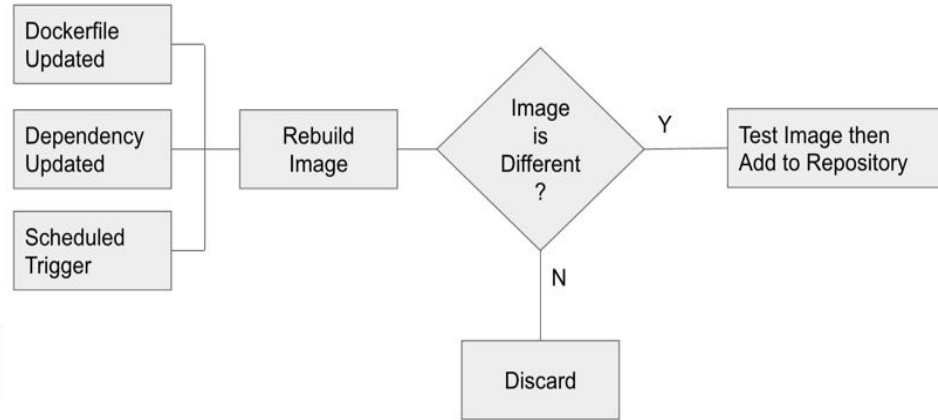
We manage the life of the application deployment for the customers:

- **Containers are automatically upgraded when new versions ship.**
- **Customer content is mounted into their containers externally.**
- **Containers are validated via a bespoke test suite.**

# Automate Proaction

Traditionally it was a manual process to update customer environments.

- We watch repositories for the latest, stable versions.
- Map app environment dependencies.
- Rebuild when a change is scheduled or spotted.





Application  
Library  
(Git / Gitlab)

Image  
Repository  
(Dockerhub)

CI Stack  
(Drone)

User Interface

Multi-tenant API

Cluster Build /  
Lifecycle Tooling

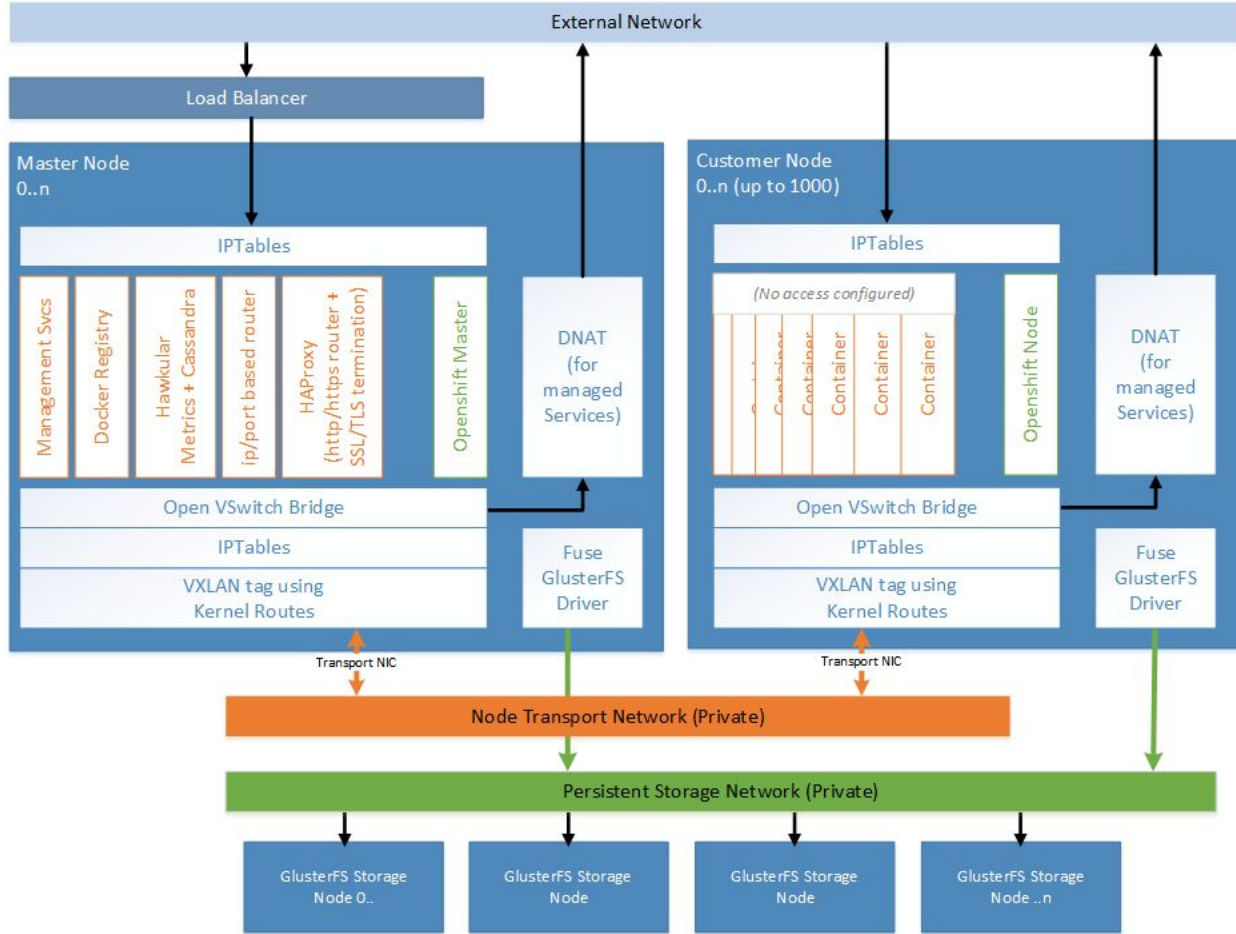
Hosting Cluster(s)

Openshift

FreelIPA

Kubernetes

Gluster



# Live Demo

# Help Us

- Chat with 1&1 at Booth #G5
- Register for the Closed Beta and win a trip to join 1&1 at ContainerDays EU in Hamburg
- Email: [baldwin@stackpoint.io](mailto:baldwin@stackpoint.io)
- Twitter: [@baldwinmathew](https://twitter.com/baldwinmathew)

**Thank you!**

