

RACKSPACE WEBMAIL DEVELOPMENT WITH DOCKER COMPOSE

BY JON ARMSTRONG



RACKSPACE WEBMAIL

- **Web-based email and collaboration for businesses**
- **Monolithic architecture, moving to microservices**
- **A monolith with many moving parts**

LOTS OF DEPENDENCIES

- **PHP / Javascript**
- **NodeJS**
- **Python (Flask)**
- **Java**
- **Dev tools**
- **Proxies**
- **App servers**
- **Identity Service**
- **MySQL**
- **IMAP**
- **SMTP**
- **Redis**
- **Memcache**

PREVIOUS DEV ENVIRONMENT

- **Rackspace cloud servers**
- **Custom VPN**
- **Image existing dev's environment for new devs**
- **Partially automated provisioning and configuration**
- **Shared instances of services**

NEW DEV ENVIRONMENT

Goals:

- **Faster ramp up for developers**
- **Easier to share changes**
- **Make dev environment more like production**
- **Allow development on local workstation**
 - Cross platform compatible

CHALLENGES

- Parts of application date to mid 2000s or earlier
- Dev tools poorly understood
- Need to keep releasing software 😊

DOCKER COMPOSE

Began moving to Docker Compose mid-2015

- **Lightweight wrapper around Docker**
- **Simple service orchestration**
 - Built-in networking between containers
 - Quickly reset environment to known starting state
- **File-based configuration**
 - No external service
 - Maintain in version control

BONUS BENEFITS

- **Understanding and documenting dev tools**
- **Easier to publish services for other teams**
- **Partial deployments of new services**

BASIC PROJECT STRUCTURE

```
|— docker
|   |— image1
|   |— image2
|— docker-compose.yml
|— src
|   |— project1
|   |— project2
```

DEMO

