Continuous Integration with Jenkins, Docker and Compose

Sandro Cirulli
Platform Tech Lead, Oxford University Press
Agenda

1. Introduction

2. Microservices Architecture

3. CI Workflow

4. Benefits of Docker and Compose in CI

5. Future Work
Introduction
Oxford University Press and Oxford Global Languages
About me

- I work as **Platform Tech Lead at Oxford University Press (OUP)**
- I deal with **system architecture and DevOps** (CI, Docker, AWS, deployment)
- I started using **Docker in November 2014** for developing language resources at OUP
About My Employer

- Oxford University Press (OUP) is a world-renowned dictionary publisher

- OUP launched the Oxford Global Languages (OGL) initiative to digitize under-represented languages

- In August 2015 OUP launched two African languages websites for Zulu and Northern Sotho
container

NOUN

isiphathizinto, isitsha

Examples
Fill this container with water
Gewalisa lesi phathizinto ngamanzi

Be part of the living dictionary

Word of the Day
indlovu
NOUN
FIND OUT WHAT IT MEANS

How do you say literacy in isiZulu?

How do you say snack in isiZulu?

Tell us your translation

Tell us your translation
System Architecture
dev, staging & production environments
System Architecture Challenges

- We needed to **migrate** to more powerful servers and eventually to the cloud.
- Some services were only needed for the **development environment**.
- We wanted to put **system configuration under version control** and replicate it easily on new machines.
- We wanted to **automate building processes**.
Microservices Architecture
dev/staging environments
CI Workflow
Walkthrough
Continuous Integration Workflow

1. Developer commits changes to GitHub
2. Webhook triggers unit tests on Jenkins
3. Passes? (Yes or No)
   - No: Shame developer on Jenkins' monitor view
   - Yes: Build Docker image
4. Start Docker container with Compose
Walkthrough

I will show you:

- **Jenkins jobs** to run:
  - unit tests
  - Docker image build and container restart

- **Shell scripts** invoked by Jenkins

- **Docker Compose file**

- Simulation of
  - successful build
  - unsuccessful build
Demo
Benefits of Docker
Docker & Compose in CI
Benefits of Docker

- Docker **isolates components** and allows their organic growth
- Docker facilitates the replication of dev/staging/production environments
- The system architecture is under version control
- Docker facilitates **blue/green deployment**
Benefits of Compose

- Compose facilitates the orchestration of linked containers
- Compose allows to have a single script for all the environments
- Developers can start/stop/rebuild containers without any deep knowledge of Docker
Future Work

- Deploy Docker containers in master/slave architecture
- Experiment with Amazon EC2 Container Service (ECS)
- Automate, Automate, Automate!
Summary
Summary

I hope I managed to show you:

▶ How we use **Jenkins** to build and start applications running on Docker

▶ How we orchestrate containers with **Docker Compose** inside different environments

▶ How we automated our **CI workflow**
Acknowledgements

Artemis Parvizi

Kal Ahmed

Meritxell Gonzàlez

Matt Kohl
Thank you!

Sandro Cirulli

www.sandrocirulli.net/dockercon2015
www.oxforddictionaries.com/words/oxfordlanguages
sandro.cirulli@oup.com