docker

In
Production

The current setup

- Hundreds of websites & API’s.
- Serving: Haproxy in front of apache/nginx.
- Backend: Postgres, Mysql, SOLR.
- Workers: Gearman Job Queue.
- CI Jenkins & language specific build utilities.
- Monitoring via file logs, ELK, icinga others…

The goal

For development:

- Reduce project setup time.
- “Free” the framework stack.
- Assume product responsibility.
- Fragment into “microservice”-like architecture.

For production:

- Improve migration ability.
- “Free” the framework stack (2).
- Improve logging aggregation.
- Better resource monitoring.
- Be able to swiftly migrate applications.

The Plan

- Install & configure hardware with cloud-like solution
- Determine VM base image.
- Establish security workflow.
- Pick node clustering solution.
- Resource monitoring/Process monitoring.
- Application level logging.
- CI & Deployment.
- Create an application prototype.
- Create base images for initial application types.
- Create building images for assets.
- Create introductory documentation for the DevOps.
- Build a new application.
- Migrate an existing application.

The barebones

- Fancy powerful servers, I guess?
- OpenNebula
- Own OS (base debian image with docker installe

Clustering

- Docker-swarm
- Private networking between nodes, with TLS or SSL termination.
- No need for TLS, docker over TCP.

Monitoring & Logging

- Resources: https://github.com/google/cadvisor
- Processes

- Logging - Logging directly to log stash at applica level or using logspout to aggregate container lo

logstash

CI & Deployment

- Still using Jenkins.
- Build via images as intermediate steps.
- Use private registry.
- Service discovery via ETC and automatic expery registration via haproxy templates.

…BUT

…why no Mesos, Rancher, <other>?

The verdict