A4M36ISS: Docker

Jiří Pechanec
QA Engineer
September 15th, 2015
Agenda

• Docker architecture
• Docker commands
• Make your own image
  • Dockerfile
• Beyond Docker
  • Composition
  • Orchestration
Containers vs virtualization
Architecture

- Client
  - docker build
  - docker pull
  - docker run

- DOCKER_HOST
  - Docker daemon
  - Containers
  - Images

- Registry
  - Container images
  - NGINX
Image

- Read-only template
- Series of layers
- Union file system
- Structure
  - Base image
  - Instructions in Dockerfile
  - Each instruction create a layer
- Number of layers limited
  - Squashing
Registry

- Storage for images
- Public or private
  - Docker Hub
- Basic operations
  - pull
  - search
  - push
Container

- Composition
  - Operating system
  - User-added files
  - Meta-data
- Upon execution
  - Image is pulled if not present
  - Create container
  - Allocate FS and add R/W layer
  - Allocate network interface
    - Provide IP
  - Execute a process
  - Capture STDOUT/STDERR
Container - Technologies

- Namaspaces
  - pid, net, ipc, mnt, uts
- Control groups
  - Limit resources
- Union FS
  - AUFS, btrfs, vfs, DeviceMapper
Docker commands
Commands

- daemon, events, version
- create, attach, exec, kill, pause, rename, restart, rm, run, start, stop, unpause, wait
- info, inspect, logs, port, ps, stats, top
- build, commit, save
- cp, export, import, load, save
- diff
- history, images, rmi
- load, login, logout, pull, push, search, tag
Advanced concepts

- Persistent data
  - Volumes
    - Bypass UnionFS
    - `VOLUME, -v <container_dir>`
    - `-v <host_dir>:`<container_dir>`

- Container co-operation
  - Port exposure
    - `-p <container_port>:`<host_port>`
  - Linking
    - Name container 1
    - Link container 2 using `--link <name>`<link_alias>`
    - Exposed as environment variables (static) and `/etc/hosts` (dynamic)
Image building
Docker build

- Build command
  - `docker build -t <name> <dir>`

- Dockerfile
  - Template with metadata
  - Format
    - INSTRUCTION arguments
  - Instructions
    - `FROM`, `MAINTAINER`, `LABEL`, `ONBUILD`
    - `RUN`, `ENV`, `ADD`, `COPY`, `USER`, `WORKDIR`
    - `CMD`, `ENTRYPOINT`, `EXPOSE`, `VOLUME`
Next steps

• JBoss images
  • http://www.jboss.org/docker/

• Best practices
  • https://docs.docker.com/articles/dockerfile_best-practices/
  • https://docs.openshift.org/latest/creating_images/guidelines.html
Hierarchy

redhat/jboss-fsw/latest

redhat/jboss-eap/latest

redhat/jboss-jdic:7

redhat/jboss-base

rhel:7

Adds the EAP distribution

Adds the JDK with appropriate ENV variables (JAVA_HOME)

Adds the "jboss" user, environment variables, etc.

Managed by Middleware

Managed by Platform

Adds the EWS distribution with Apache Tomcat

Adds the EWS distribution with Apache HTTPD

Managed by Middleware

Managed by Platform
Beyond Docker
Composition

- Nulecule
  - http://www.projectatomic.io/docs/nulecule/
- Docker Compose
  - https://docs.docker.com/compose/
Orchestration

- Kubernetes
  - http://kubernetes.io/
- OpenShift
  - http://www.openshift.org/
- Docker Swarm
  - https://docs.docker.com/swarm/