Web Applications with Eclipse RT and Docker in the Cloud

Johannes Eickhold, Markus Knauer, Florian Waibel
User Perspective

- register
- access
- upload
- manage
- experiment
- play
Implementing our Vision

- one lightweight container per lab
- minimal / no manual preparation per lab
Button button = new Button(parent, SWT.PUSH);
button.setText("Push me!");

{"head":{"requestCounter":1},
"operations": [...
["create","w3","rwt.widgets.Button","parent":"w2","style":["PUSH"],"bounds":[5,5,98,32]],["set","w3","text":"Push me!"],... ]}
Heavy to Lightweight

Isolation Level

- Heavyweight
- Lightweight

Docker

Windows Azure

Amazon Web Services
Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications.
Intro to Docker

Docker uses *containers*

- LXC (Linux Containers)
  - cgroups and namespaces for strong process isolation and resource control

Docker uses a *copy-on-write filesystem*

- AUFS
  - stacking and layering of containers

Docker uses simple configuration language
Intro to Docker: Containers

- Docker Container A
  - Web Application A
    - Tomcat
    - Base Image: Debian
  - API
  - Docker Daemon
  - Console Client

- Docker Container B
  - OSGi Application B
    - Eclipse Virgo
    - Base Image: Ubuntu
  - Host OS (Linux)

- Docker Container C
  - OSGi Application C
    - Eclipse Virgo
    - Base Image: Ubuntu
  - Server Hardware
Intro to Docker: Commands

$ docker version
$ docker search ubuntu
$ docker pull ubuntu

$ docker run ubuntu cat /etc/os-release
$ docker run -t -i ubuntu /bin/bash
$ docker run ubuntu apt-get install -y iputils-ping

$ docker ps -l
$ docker commit --author="Markus Knauer <mknauer@eclipsesource.com>"
    -m="Install ping utility" ID123123123123 mknauer/ping
$ docker run mknauer/ping ping eclipse.org
$ docker run -i -t mknauer/ping /bin/bash
FROM base:ubuntu-12.10
MAINTAINER EclipseSource <admin@eclipsesource.com>

RUN apt-get update && apt-get upgrade -y
RUN apt-get install -y gzip tar curl openjdk-7-jre-headless
RUN useradd -m tomcat
...

EXPOSE 8080

ADD setenv.sh /home/tomcat/bin/setenv.sh

VOLUME ["/home/tomcat/webapps"]
Docker Security

Docker Daemon is running as root!

Beware!

Bad things may happen!
Serving HTTP

SSL for https?
Dynamic Rewriting?
The Backend Server

Developer

:443

User Portal

mongoDB

Trial Application User

:80
Dynamic Reverse Proxy++

http://tabristrial.eclipsesource.com/foo/examples

http://localhost:20002/myapp/examples
Our Technology Selection

- docker
- Virgo from eclipseRT
- NGINX
- Google+ Sign in
Using the Lab

1. register at website
2. receive mail with registration key
3. access user portal with registration key (OAuth)
4. upload app (Web Application)
5. URL under which app is available
6. access app
Docker Container

- Based on Ubuntu image
- Contains OpenJDK 7 + Virgo
- Static and dynamic containers
Inside Docker: Virgo

- Virgo Runtime
- RAP and dependencies pre-installed
- Deployment mechanism for uploaded app
Visit the Labs

DEMO
Conclusions

● Docker proved to be stable and flexible and scalable at the same time

● Inspiring cool technology

● Creating Docker images manually is time consuming and error prone
Outlook

● Put *everything* into Docker containers
● Switch to Virgo RAP Server base-image
● Use Gradle to build Docker images
● Polish web user interfaces and add more management capabilities to user console
● Release as well documented open source template project
Questions?