Docker II - Judgement Day

PUG Challenge Americas 2016

Thomas Hansen
whoGlooo, Inc.
Introduction

Thomas Hansen
CEO & Co-Founder of whoGloo, Inc.

Working with OpenEdge since 1995

Working with Docker since 2014 :-)

- Core part of the nodeSpeed development platform and application runtime environments
- Used for internal OpenEdge projects
What is Docker?

DOCKER PROJECTS
Specialized tools to help developers build modern, distributed applications.

DOCKER ENGINE
Create and runs Docker Containers

DOCKER COMPOSE
Define multi-container applications

DOCKER REGISTRY
Open source Docker image distribution [not included]

DOCKER MACHINE
Automated Docker Provisioning

DOCKER SWARM
Host clustering and container scheduling

KITEMATIC
Desktop User Interface for Docker

© 2016 www.whogloo.com
Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries.

...result is significantly faster deployment, much less overhead, easier migration, faster restart.
Docker Engine
Docker Engine

**docker create**: create new container from image

**docker run**: run container

**docker exec**: execute a program inside a running container

**docker stop** / **start** / **restart**

**docker inspect**: inspect the structure of a container

**docker cp**: copy data between host and containers

**docker rm**: remove docker container

**docker rmi**: remove docker image
Docker Compose

GET STARTED IN 3 STEPS:

1. WRITE YOUR DOCKERFILE
   - WORKDIR /code
   - ADD requirements.txt /code/
   - RUN pip install -r requirements.txt
   - ADD ./code
   - CMD python app.py

2. WRITE YOUR COMPOSE.YML FILE
   - web:
     - build: .
     - links:
       - db
     - ports:
       - "8000:8000"
   - db:
     - image: postgres

3. RUN YOUR APP
   - $ docker-compose up
Docker Compose

“Recipes” for projects with multiple containers

```yaml
openedge:
    image: whogloo/openedge116
    volumes_from:
        - openedge_dev_data
    ports:
        - 9090:9090

openedge_dev_data:
    image: busybox
    volumes:
        - /data
```
Docker Hub

Cloud service (registry)

Share Applications

Automate workflows

Assemble apps from components
Docker Registry

“The Registry is a stateless, highly scalable server side application that stores and lets you distribute Docker images.”

You should use the Registry if you want to:

● Tightly control where your images are being stored

● Fully own your images distribution pipeline

● Integrate image storage and distribution tightly into your in-house development workflow

(Source: docker.com)
Docker Machine

Automates all the provisioning and installation tasks for a single Docker host

$ docker-machine create -d virtualbox dev
$ docker-machine create --driver amazonec2 aws-sandbox
Docker Swarm

Host clustering and container scheduling
Docker Swarm

Create a swarm

docker-machine create -d virtualbox \
--swarm --swarm-master \
--swarm-discovery token:// swarm-master

Create more docker hosts for swarm

docker-machine create -d virtualbox \
--swarm --swarm-discovery token:// swarm-node-00
Docker Cloud

“Docker Cloud is a hosted service that provides a Registry with build and testing facilities for Dockerized application images, tools to help you set up and manage your host infrastructure, and deployment features to help you automate deploying your images to your infrastructure.”
Docker Volumes

Volume drivers exist for
- AWS S3
- GlusterFS
- Flocker
- Azure File Storage
- VMWare
- NFS
- ....

Complete list:
https://github.com/docker/docker/blob/master/docs/extend/plugins.md
Docker for Windows

- Docker for Windows and Kitematic currently in beta
- Download and test today
Docker Store

- “a marketplace for trusted and validated dockerized software – free, open source and commercial”

- Private Beta just launched
Docker & OpenEdge

The following slides contain information of a highly sensitive nature - of which you will have no recollection when you leave...
Need OpenEdge?

Need OpenEdge to run databases, AppServer, WebSpeed?

● Install Docker
● Build images
● Push images
● Pull images to any machine
● Configure containers
● Run
OpenEdge & Docker

Quickly pull and use OpenEdge on any supported platform
No installation on host required!

- Run & maintain databases
- Run AppServers
- Run WebSpeed
- Execute ad-hoc OpenEdge jobs
OpenEdge & Docker

- Test applications on multiple OpenEdge versions/platforms - **on one host without installing** - just pull and run
OpenEdge & Docker

- Develop
- Test
- Build
- Deploy
- Install
OpenEdge in Docker examples
Roundtable TSMS in Docker

- **volumes:**
  - /usr/rtb115
  - /usr/openedge116
  - /usr/wrk/116

- **--volumes-from**
  - rtbws-data
  - rtb115:11.5.02
  - rtb-db-demo:11.5

- **--link** rtb-db-server:db

- **volumes:**
  - /data/db/rtb115:
  - /data/rtbws:

© 2016 www.whogloo.com
Development Containers

appserver116:latest
volumes:
- /usr/openedge116
- /usr/wrk/116

--volumes-from
rtbws-data
volumes:
- /data/rtbws:/data/rtbws

--link db-server1:db1

openedge116:latest
volumes:
- /usr/openedge116
- /usr/wrk/116

--volumes-from
dev-db1-data
volumes:
- /data/db/db1:/data/db/db1

--volumes-from
dev-db2-data
volumes:
- /data/db/db2:/data/db/db2

--volumes-from
openedge116:latest
volumes:
- /usr/openedge116
- /usr/wrk/116

© 2016 www.whogloo.com
Shared memory

- **app1-oell6-appserver**
  - (appserver116:latest)
  - `volumes`:
    - `/usr/openedge116`
    - `/usr/wrk/116`

- **rtbws-data**
  - (busybox)
  - `volumes`:
    - `/data/rtbws:/data/rtbws`

- **dev-db1-data**
  - (busybox)
  - `volumes`:
    - `/data/db/db1:
      - `/data/db/db1`

- **dev-db2-data**
  - (busybox)
  - `volumes`:
    - `/data/db/db2:
      - `/data/db/db2`
OpenEdge Demo
Docker Build

After application deployment and build - put it into a container

```bash
echo "Deleting existing server dirs..."
rm -fr server-runtime

echo "Copying new server dirs..."
cp -R ${buildRootDir}/ server-runtime
dos2unix server/*.sh

echo "Building docker image..."
docker build -t ${imageName} .
```
Docker Build

# Dockerfile TO CREATE App Runtime container.

FROM busybox

COPY ./server-runtime /data/app/runtime

VOLUME /data/app
Use

# Pull app runtime images from repo
docker pull whogloo/app:03.00.10

# Create data container from tagged image
docker create \
    --name=cust1-app-runtime-data \         nodeable/app-server-runtime:
    03.00.10

# Start runtime containers using data container
docker run \
    ... \
    --volumes-from cust1-app-runtime-data \
    ... 
    whogloo/appserver116
Docker Pros

- Fast, flexible & scalable
- Huge traction – changes and tools coming out all the time
- Abstraction of services into micro services
- Image layers
- Runs on many platforms – even Raspberry Pi!
- Content and hardware agnostic
- Separation of duties
Docker Cons

- Keep an eye on disk space!
- Concepts can be complicated to start with
- Lack of graphical tools – command line
- Lack of dynamic port exposure
- Many containers to keep track of
- Not available on Windows – yet
- Lack of OpenEdge support
Learning about Docker

- https://www.docker.com
- https://docs.docker.com
- What You Need to Know about Docker - Free eBook (until 7 pm EDT June 27 2016) https://www.packtpub.com/packt/offers/free-learning
Questions

Email: thomas@whogloo.com
Twitter: @whogloo
LinkedIn: https://www.linkedin.com/company/whogloo
Facebook: https://www.facebook.com/whogloo