

Using Docker to Deploy your Servoy Applications

by Scott Butler

iTech
Professionals

01



Docker

What is it?
Why should I use it?
Who am I to be telling you about it?

02



Create a Docker File

Understand the Format.
Use it with Servoy.

Our Goals Today

The nuts & bolts. Things you can take with you, and
apply to your environment tomorrow!

03



Advanced Topics

Link containers together.
Add configurable options.

04



Deploy!

To the cloud.
And on premise.

About Me

Has it really been that long?



Started developing in Servoy in 2005

Servoy version 2.2! It was just a baby.

Migrated legacy enterprise apps.



Senior Technical / Sales Engineer for Servoy US

From 2007 – 2009. Project Management,
Software conversions, Trainings, etc.



Started iTech Professionals in 2009

Dedicated Servoy Consulting, Support and
Development.

iTech
Professionals

Daily Active Users

1,000 +

Number of Servoy Servers

15

Deployment Types

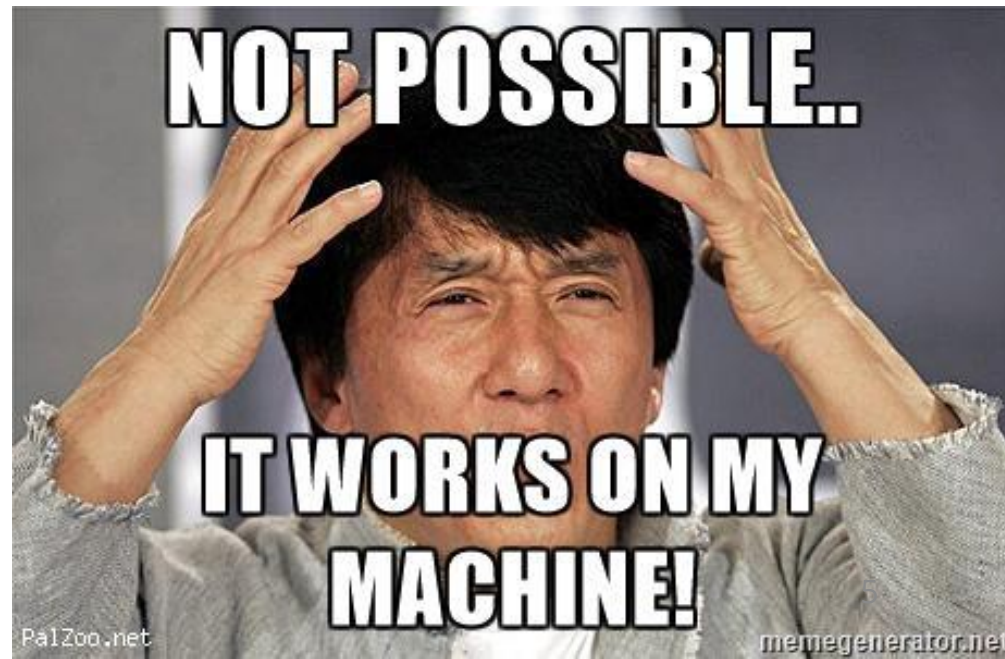
?



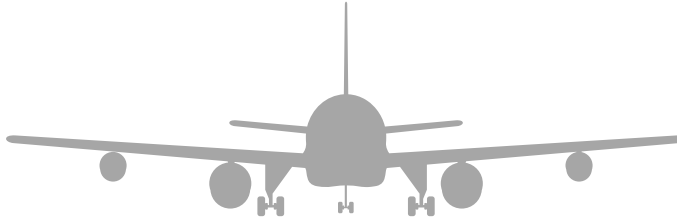
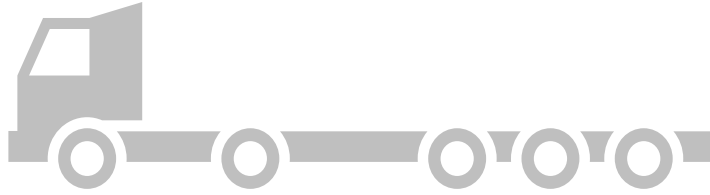
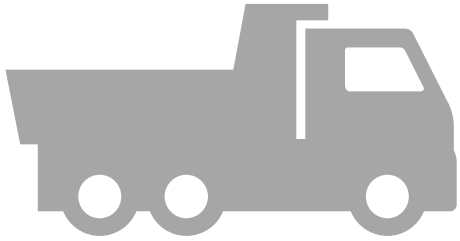
Deployment Matrix of Hell

		Server Purpose		
		Development	Staging	Production
Location / Type	Your Location	✓	✓	✓
	Customer's Location	✓	✓	✓
	Public Cloud	✓	✓	✓
	Private Cloud	✓	✓	✓
	Virtual Machine	✓	✓	✓
	Cluster	✓	✓	✓
	Laptop / Runtime	✓	✓	✓

This Leads to Unreliable Deployments

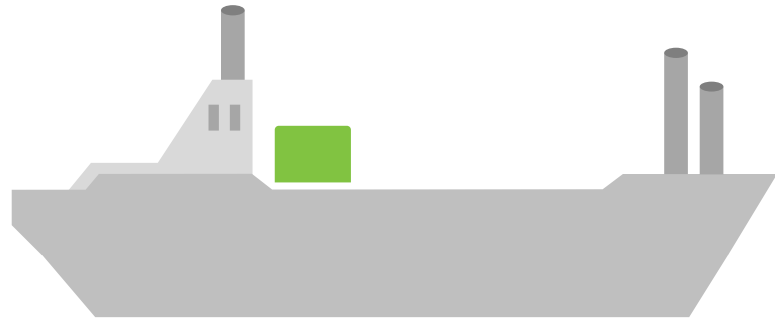
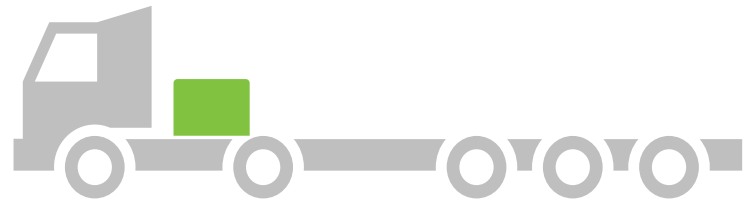
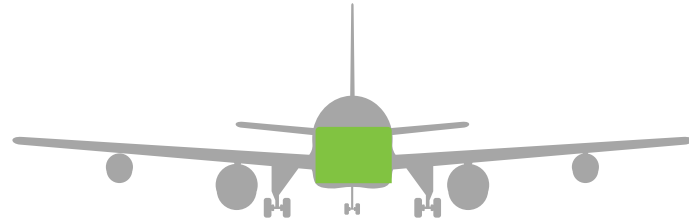
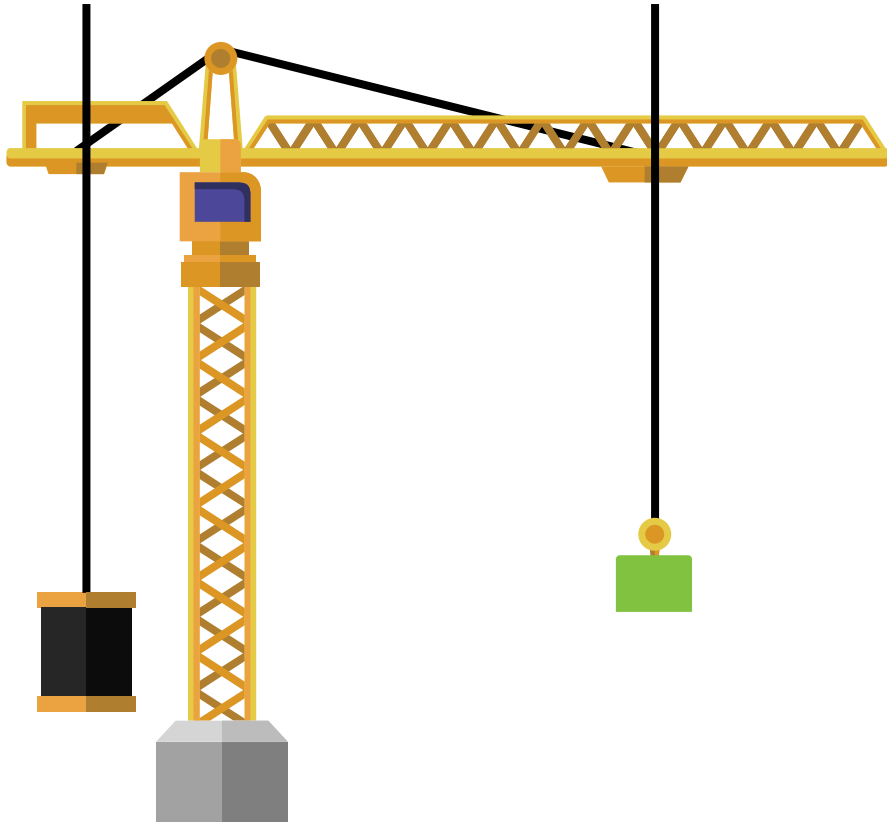


This isn't a new problem...



Containers!

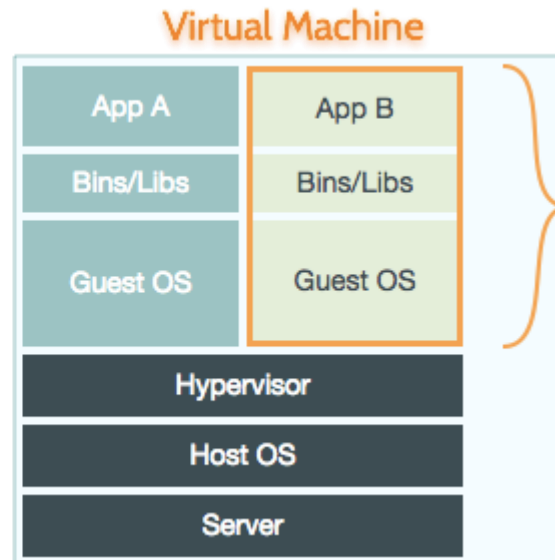
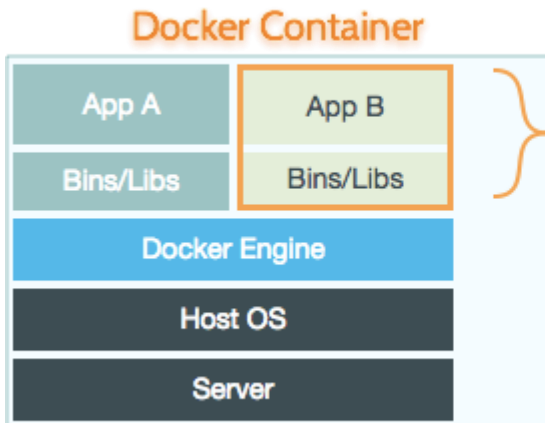
Standardized units to package everything in



Docker

Its more than just a VM!

- Software Container Platform
 - Build, Ship, Run. Any app, Anywhere.
 - Open Source
- Additional abstraction layer of OS level virtualization
 - The OS isn't replicated on each Docker container!
 - Only libraries and settings required to make your software work.
 - Makes them MUCH more efficient and lightweight.

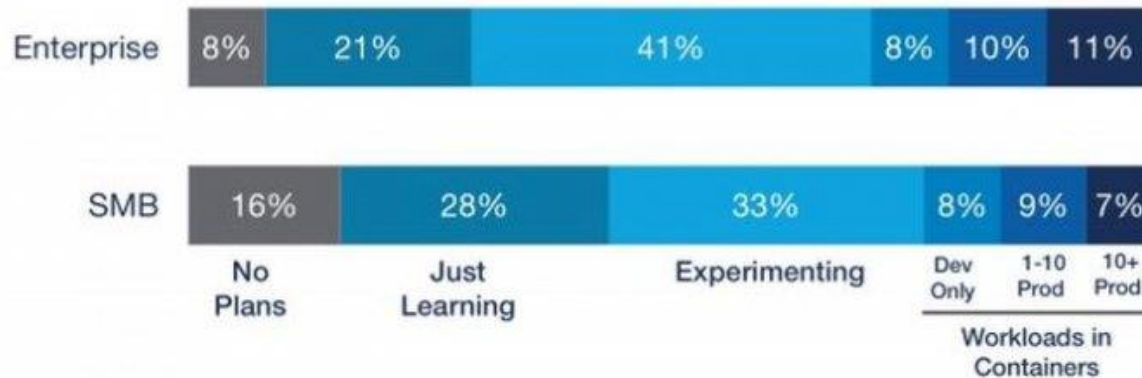


Docker Popularity

What a Difference 3 Years Makes



Container Usage of Respondents

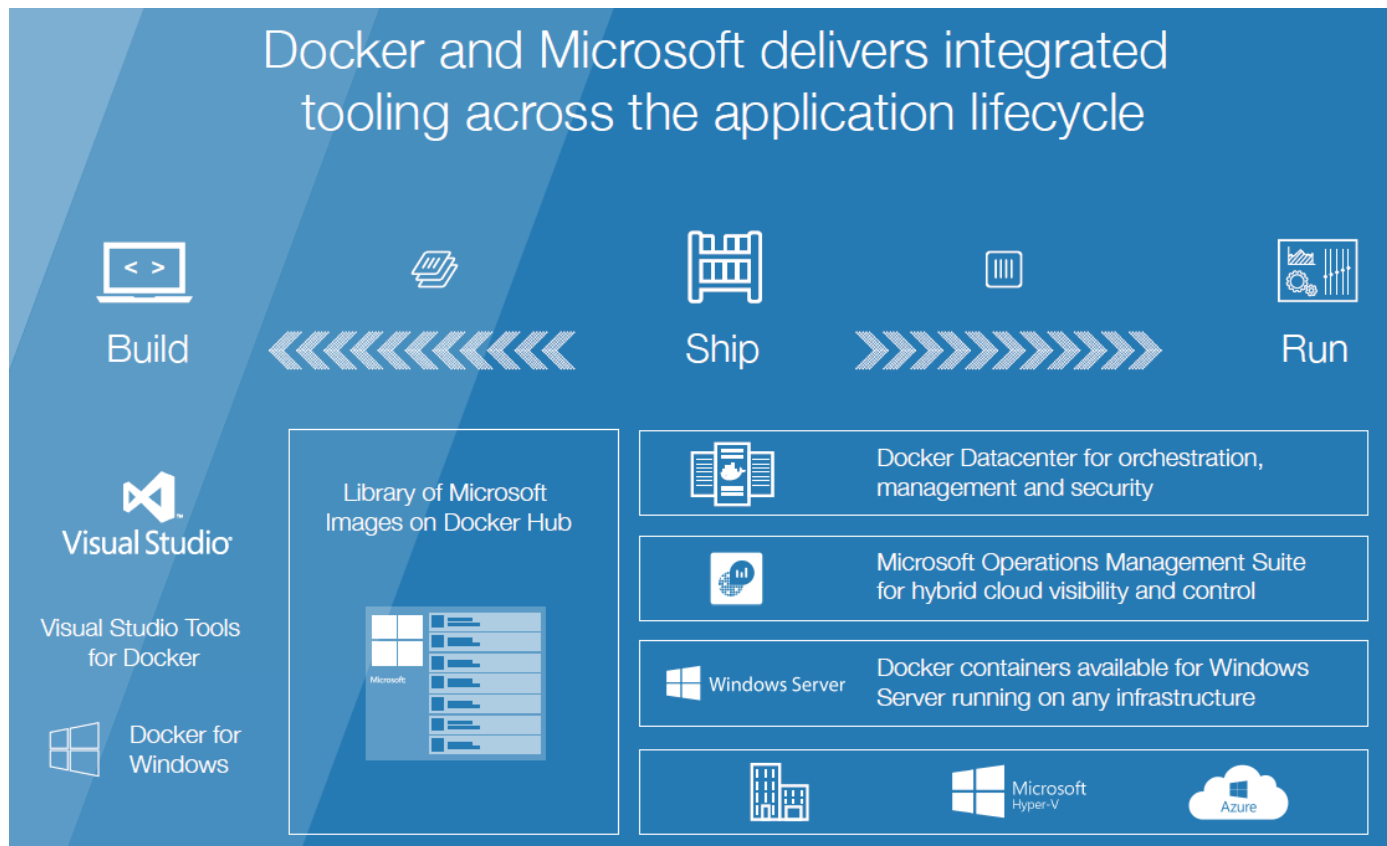


Source: RightScale 2016 State of the Cloud Report

Microsoft & Docker

From First Release to Native Windows Server Support in 4 Years!

- Native support for Windows Containers on Windows Server 2016
- Native Support for Linux Containers on Windows Server 2016
 - Just announced at DockerCon last month!
- Docker Enterprise Partnership



Have I convinced you that you need Docker yet?

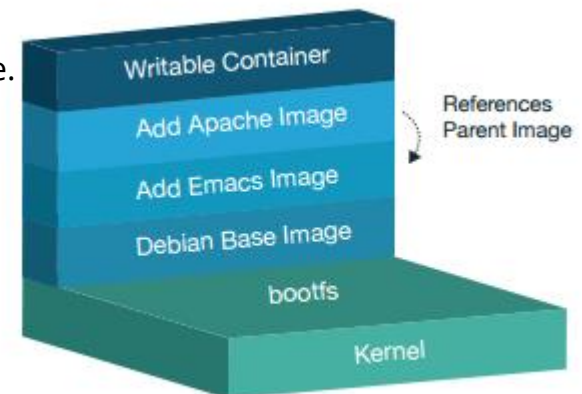
Next up...

Creating a Docker File



Docker Terminology

- Dockerfile
 - A text document containing the instructions (command lines) to create the Image.
 - Uses inheritance!
- Image
 - An ordered collection of filesystems, built up in layers.
 - Layers usually created from Dockerfiles
 - Each image has a name, and gets registered in some registry (public or private)
 - Read-only layer used to build a Container.
 - When started, has a default process it runs.
- Container
 - Runtime instance of an Image
 - You **can** make changes to the container, and save it back as an image.
 - Don't do this!
- Registry
 - Hosting location of the Images. Default is Docker Hub.
- Repository
 - Library of variations available for the Images under different Tags (versions, etc)



Getting Started

Workflow Example

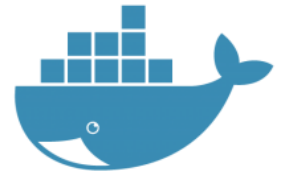
GitHub

- MyProject
 - Dockerfile
 - myOtherFile.txt



Configure Setup
Tags/Folders/Branches

DockerHub



- username/MyProject
 - v1
 - v2



Authenticate for private DockerHub Projects

DockerCloud



```
docker pull username/myproject:v1
```

Anywhere with Docker



```
docker pull username/myproject:v1
```

Getting Started

How to do this on your own?

- Create your own Dockerfiles, Extend ours!
 - Pure Servoy (Web and Smart)
 - FROM: `itechpros/servoy:servoy-oraclejava`
 - NGClient (Web and Smart)
 - FROM: `itechpros/servoy:tomcat-oraclejava`
 - Add your resources
 - Property file/WAR, plugins, etc
 - Bundle it with Docker Compose to spin up DB instances with Servoy
 - Deploy it to Docker Cloud, Portainer, or your favorite tool
- Each “Service” should be its own Docker Container!

Dockerfile

The commands for your image

- DockerFile has its own set of available commands
- Example

```
FROM tomcat:7
```

```
MAINTAINER Scott Butler <scott@itechpros.com>
```

```
# OS Updates and base installs that are useful
```

```
RUN yum update -y && \
```

```
yum install -y \
```

```
    wget \
```

```
    gzip \
```

```
    tar
```

```
# Add files to the image from source control
```

```
ADD tomcat.conf    /usr/local/tomcat/conf
```

```
# Open Ports
```

```
EXPOSE 8080
```

```
# Define default command.
```

```
CMD ["/my_startup.sh"]
```

Example for Today

Multiple configurations with multiple deployment scenarios

- Servoy 8 Standard (non-WAR)
- Tomcat Server with Servoy 8 WAR Deployment
- Preconfigured Database on Amazon Cloud
- Dynamic Postgres Database
- Several Deployment Variations
 - Local and Cloud

Servoy / Preconfigured DB

Dockerfile Part 1 of 3: [itechpros/servoy:servoy-oraclejava](#)

```
FROM store/oracle/serverjre:8
```

```
MAINTAINER Scott Butler <scott@itechpros.com>
```

```
# Environment variables
```

```
ENV SERVOY_VERSION_MAJOR ${SERVOY_VERSION_MAJOR:-8}
```

```
ENV SERVOY_INSTALL_DIR ${SERVOY_INSTALL_DIR:-/usr/home/servoy}
```

```
ENV JAVA_XMS ${JAVA_XMS:-64m}
```

```
ENV JAVA_XMX ${JAVA_XMX:-1024m}
```

```
ENV TIME_ZONE ${TIME_ZONE:-America/New_York}
```

```
# Bundle installer config and scripts
```

```
ADD servoy_install_${SERVOY_VERSION_MAJOR}.xml servoy_install.xml
```

```
ADD servoy_docker.sh ${SERVOY_INSTALL_DIR}/application_server/
```

```
ADD servoy_server_docker_${SERVOY_VERSION_MAJOR}.sh
```

```
${SERVOY_INSTALL_DIR}/application_server/servoy_server_docker.sh
```

```
# Add RPM for windows fonts
```

```
ADD msttcore-fonts-2.0-3.noarch.rpm msttcore-fonts-2.0-3.noarch.rpm
```

Servoy / Preconfigured DB

Dockerfile Part 2 of 3: [itechpros/servoy:servoy-oraclejava](#)

```
#Set Timezone
```

```
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone && \
```

```
# OS Updates and base installs that are useful. net-tools is used by UserManager plugin
```

```
yum update -y && \
```

```
yum install -y wget unzip tar gzip gunzip net-tools gettext \
```

```
# Install envplate templating engine for environment variable substitution
```

```
curl -sLo /usr/local/bin/ep https://github.com/kreuzwerker/envplate/releases/download/v0.0.8/ep-linux && \
```

```
chmod +x /usr/local/bin/ep && \
```

```
# Install Windows fonts, like Arial and Verdana
```

```
rpm -Uvh msttcore-fonts-2.0-3.noarch.rpm && \
```

```
# Cleanup
```

```
yum clean all && \
```

```
# Install Servoy
```

```
wget --output-document servoy_installer.jar --no-cookies --no-check-certificate
```

```
"http://www.servoyguy.com/docker/download/servoy.php?version=$SERVOY_VERSION_MAJOR" && \
```

```
java -jar servoy_installer.jar servoy_install.xml && \
```

```
rm -f servoy_installer.jar && \
```

```
rm -f servoy_install.xml && \
```

Servoy / Preconfigured DB

Dockerfile Part 3 of 3: [itechpros/servoy:servoy-oraclejava](#)

```
#set permissions
chmod +x $SERVOY_INSTALL_DIR/application_server/servoy_docker.sh && \
chmod +x $SERVOY_INSTALL_DIR/application_server/servoy_server_docker.sh

# Add properties after servoy install so we overwrite it
ADD servoy.properties $SERVOY_INSTALL_DIR/application_server/

# Set environment variables.
ENV HOME $SERVOY_INSTALL_DIR/

# Define working directory.
WORKDIR $SERVOY_INSTALL_DIR/application_server

# Open Ports
EXPOSE 8080
EXPOSE 1099

# Define default command.
CMD [ "./servoy_docker.sh" ]
```

Servoy / Preconfigured DB

Extra Resources

/servoy_docker.sh

Wrapper to fill in variable into any files before executing startup scripts

```
#!/bin/sh echo "Running Servoy Start Script"
echo "Filling templates with environmental variables"
ep servoy_server_docker.sh
ep servoy.properties

echo "Starting Servoy"
./servoy_server_docker.sh
```

Servoy 8.2 Will support configurable variables in the properties file!

/servoy_server_docker.sh

Replica of official servoy_server.sh with env variables for java memory.

```
#!/bin/sh
if [ -z "$CMD_LINE_START" ]; then
    CMD_LINE_START="java"
fi

while true
do
    $CMD_LINE_START -Djava.awt.headless=true -Drmi.whitelist.config=com.servoy.:com.sebster. -Xmx${JAVA_XMX} -
Xms${JAVA_XMS} -XX:MaxPermSize=128m -XX:MetaspaceSize=200M -XX:MaxMetaspaceSize=200M -XX:MinMetaspaceFreeRatio=0 -
XX:MaxMetaspaceFreeRatio=100 -classpath <BunchOfJars...> com.servoy.j2db.server.main.ApplicationServer "$@"
    EXITCODE=$?
    if [ "$EXITCODE" != 99 ] && [ "x$EXITCODE" != "x$ADDITIONAL_RESTART_CODE" ]; then exit $EXITCODE; fi
done
```

Servoy / Preconfigured DB

Deploy

- Deploy to Local with Kitematic
- For Developer local testing of your Dockerfiles
- Install Docker on your machine
- Install Docker Toolbox
 - Includes Kitematic!
 - Available on Mac and Windows

Servoy / Preconfigured DB / Extending

Dockerfile

```
FROM itechpros/servoy:servoy-oraclejava  
MAINTAINER Scott Butler <scott@itechpros.com>
```

```
# Add my property file
```

```
ADD servoy.properties $SERVOY_INSTALL_DIR/application_server/
```

```
# Define default command.
```

```
CMD [ "./servoy_docker.sh" ]
```

- Add your own Servoy property file pointing to your cloud DB's
- Add your Servoy plugins/beans/drivers/etc
- Variations for Dev/Staging/Prod through variables or different property files

Servoy / Preconfigured DB / Extending

Deploy

- Deploy to hosted server with Portainer
- Portainer:
 - Open Source Docker Project
 - Docker Container to manage other Docker Containers
 - Launch Containers
 - More configurable options than Kitematic
 - View Stats
 - Connect to Terminal
 - Manage Containers/Volumes/Images
- Demo
 - <http://208.113.166.142:9000>
 - [itechpros/servoy:servoy_generic_demo_preconfigured](http://208.113.166.142:9000/itechpros/servoy:servoy_generic_demo_preconfigured)
 - <http://208.113.166.142:8080/servoy-webclient/>

Servoy / Extending with Dynamic DB

Linking Containers Together

- Docker Compose- Create multiple containers and link them together

/docker-compose.yml

version: '3'

services:

servoy:

container_name: Servoy8

image: 'itechpros/servoy:servoy_generic_demo_dynamic_db'

environment:

- JAVA_XMS=128m

- JAVA_XMX=1024m

links:

- db

depends_on:

- db

ports:

- '8080:8080'

db:

container_name: Postgres

image: 'itechpros/servoy:servoy_generic_demo_dynamic_db_pg'

environment:

- POSTGRES_DB=servoyworld_2017

- POSTGRES_PASSWORD=demo

volumes:

- /container_volumes/pg:/var/lib/postgresql/data

Servoy / Extending with Dynamic DB

Linking Containers Together

- Dockerfile: itechpros/servoy:servoy_generic_demo_dynamic_db_pg

```
FROM postgres:9.6
```

```
ADD servoyworld_db.sql /docker-entrypoint-initdb.d/
```

Servoy / Extending with Dynamic DB

Linking Containers Together

- Dockerfile: itechpros/servoy:servoy_generic_demo_dynamic_db

```
FROM itechpros/servoy:servoy-oraclejava
```

```
MAINTAINER Scott Butler <scott@itechpros.com>
```

```
# Add my property file and my startup file
```

```
ADD servoy.properties $SERVOY_INSTALL_DIR/application_server/
```

```
ADD wait-for-postgres.sh $SERVOY_INSTALL_DIR/application_server/
```

```
# Add Postgres Client 9.6 and special script to wait for postgres to startup
```

```
RUN wget -O pgyum.rpm https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm && \
```

```
yum localinstall -y pgyum.rpm && \
```

```
yum install -y postgresql96 && \
```

```
chmod +x $SERVOY_INSTALL_DIR/application_server/wait-for-postgres.sh
```

```
# Add our postgres path to path so our wait-for-postgres.sh works
```

```
ENV PATH="${PATH}:/usr/pgsql-9.6/bin"
```

```
# Define an Entrypoint, Runs before the command
```

```
ENTRYPOINT ["/wait-for-postgres.sh", "db", "5432"]
```

```
# Define default command.
```

```
CMD ["/servoy_docker.sh"]
```

- Deploy to server manually and demo with Portainer

Servoy / Extending with Dynamic DB

Linking Containers Together (Recap)

- Docker Compose- Create multiple containers and link them together

/docker-compose.yml

version: '3'

services:

servoy:

container_name: Servoy8

image: 'itechpros/servoy:servoy_generic_demo_dynamic_db'

environment:

- JAVA_XMS=128m

- JAVA_XMX=1024m

links:

- db

depends_on:

- db

ports:

- '8080:8080'

db:

container_name: Postgres

image: 'itechpros/servoy:servoy_generic_demo_dynamic_db_pg'

environment:

- POSTGRES_DB=servoyworld_2017

- POSTGRES_PASSWORD=demo

Servoy / Extending with Dynamic DB

Linking Containers Together

- Docker Cloud- Create multiple containers and link them together
/docker-cloud.yml

db:

environment:

- POSTGRES_DB=servoyworld_2017
- POSTGRES_PASSWORD=demo

image: 'itechpros/servoy:servoy_generic_demo_dynamic_db_pg'

servoy:

environment:

- JAVA_XMS=128m
- JAVA_XMX=1024m

image: 'itechpros/servoy:servoy_generic_demo_dynamic_db'

links:

- db

ports:

- '8080:8080'

- Deploy to Docker Cloud demo

Servoy WAR

Required for NGClient Deployment

- Based off Tomcat Installation
- Servoy 8 WAR- Preconfigured DB
- Servoy 8 WAR- Extended with Preconfigured DB
- Servoy 8 WAR- Extended with Dynamic Postgres DB
 - Deploy to a hosted server with Docker Cloud

Servoy WAR / Preconfigured DB

Dockerfile Part 1 of 3: **itechpros/servoy:tomcat-oraclejava**

```
FROM store/oracle/serverjre:8
MAINTAINER Scott Butler <scott@itechpros.com>

# Environment variables

ENV CATALINA_HOME /usr/share/tomcat

ENV SERVOY_USER_HOME ${SERVOY_USER_HOME:-/home/servoy}

ENV TOMCAT_VERSION_MAJOR ${TOMCAT_VERSION_MAJOR:-8}

ENV TOMCAT_VERSION_FULL ${TOMCAT_VERSION_FULL:-8.5.15}

ENV TOMCAT_USERNAME ${TOMCAT_USERNAME:-tomcatAdmin}

ENV TOMCAT_PASSWORD ${TOMCAT_PASSWORD:-tomcatPassword!}

ENV TOMCAT_MAX_FILE_SIZE ${TOMCAT_MAX_FILE_SIZE:-300000000}

ENV JAVA_XMS ${JAVA_XMS:-128m}

ENV JAVA_XMX ${JAVA_XMX:-1024m}

ENV CATALINA_OPTS -Xms${JAVA_XMS} -Xmx${JAVA_XMX} -DSERVOY_USER_HOME=${SERVOY_USER_HOME}

ENV TIME_ZONE ${TIME_ZONE:-America/New_York}

# Add RPM for windows fonts

ADD msttcore-fonts-2.0-3.noarch.rpm msttcore-fonts-2.0-3.noarch.rpm
```

Servoy WAR / Preconfigured DB

Dockerfile Part 2 of 3: itechpros/servoy:tomcat-oraclejava

```
#Set Timezone
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone && \

# OS Updates and base installs that are useful. net-tools is used by UserManager plugin
yum update -y && \
yum install -y wget unzip tar gzip gunzip net-tools gettext && \

# Install envplate templating engine for environment variable substitution
curl -sLo /usr/local/bin/ep https://github.com/kreuzwerker/envplate/releases/download/v0.0.8/ep-linux && \
chmod +x /usr/local/bin/ep && \

# Install Windows fonts, like Arial and Verdana
rpm -Uvh msttcore-fonts-2.0-3.noarch.rpm && \

# Cleanup
yum clean all && \

# Create Servoy Home Dir and make sure we have full access
mkdir -p $SERVOY_USER_HOME && \
chmod +rwx $SERVOY_USER_HOME && \

# Add Tomcat
mkdir $CATALINA_HOME && \
wget -O $CATALINA_HOME/apache-tomcat.tar.gz http://www-us.apache.org/dist/tomcat/tomcat-
$TOMCAT_VERSION_MAJOR/v$TOMCAT_VERSION_FULL/bin/apache-tomcat-$TOMCAT_VERSION_FULL.tar.gz && \
tar -zxvf $CATALINA_HOME/apache-tomcat.tar.gz -C $CATALINA_HOME --strip-components=1
```

Servoy WAR / Preconfigured DB

Dockerfile Part 3 of 3: [itechpros/servoy:tomcat-oraclejava](#)

```
# Add custom config files
```

```
ADD tomcat.conf $CATALINA_HOME/conf/
```

```
ADD manager.xml $CATALINA_HOME/conf/Catalina/localhost/
```

```
ADD tomcat-users.xml $CATALINA_HOME/conf/
```

```
ADD web.xml $CATALINA_HOME/webapps/manager/WEB-INF/
```

```
ADD tomcat_start.sh $CATALINA_HOME/
```

```
# Make the startup executable
```

```
RUN chmod +x $CATALINA_HOME/tomcat_start.sh
```

```
# Define default working directory
```

```
WORKDIR $CATALINA_HOME/webapps
```

```
# Open Ports
```

```
EXPOSE 8080
```

```
EXPOSE 8443
```

```
# Substitute Environment variables in the config files and start services within the custom script
```

```
CMD $CATALINA_HOME/tomcat_start.sh
```


Servoy WAR / Preconfigured DB / Extending

Dockerfile

```
FROM itechpros/servoy:tomcat-oraclejava
```

```
MAINTAINER Scott Butler <scott@itechpros.com>
```

```
ENV WAR_CONTEXT ${WAR_CONTEXT:-servoy}
```

```
# Add my war file
```

```
RUN wget -O $CATALINA_HOME/webapps/$WAR_CONTEXT.war http://servoyguy.com/docker/demo/servoy8_samplecrm.war
```

```
# Define default command.
```

```
CMD $CATALINA_HOME/tomcat_start.sh
```

Servoy WAR / Dynamic DB / Extending

Dockerfile

```
FROM itechpros/servoy:tomcat-oraclejava
MAINTAINER Scott Butler <scott@itechpros.com>

ENV WAR_CONTEXT ${WAR_CONTEXT:-servoy}

ADD wait-for-postgres.sh $CATALINA_HOME/

# Add Postgres Client 9.6 and special script to wait for postgres to startup

RUN wget -O pgyum.rpm https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm && \
yum localinstall -y pgyum.rpm && \
yum install -y postgresql96 && \
chmod +x $CATALINA_HOME/wait-for-postgres.sh && \

# Add my war file

wget -O $CATALINA_HOME/webapps/$WAR_CONTEXT.war http://servoyguy.com/docker/demo/svySampleCRM_dockerCompose.war

# Add our postgres path to path so our wait-for-postgres.sh works

ENV PATH="${PATH}:/usr/pgsql-9.6/bin"

# Define an Entrypoint, Runs before the command

ENTRYPOINT ["../wait-for-postgres.sh", "db", "5432"]

# Define default command.

CMD ["../tomcat_start.sh"]
```

Servoy / Extending with Dynamic DB

Linking Containers Together WAR (Recap)

- Docker Compose- Create multiple containers and link them together

/docker-compose.yml

version: '3'

services:

servoy:

container_name: Servoy8

image: 'itechpros/servoy:servoy_war_demo_dynamic_db'

environment:

- JAVA_XMS=128m

- JAVA_XMX=1024m

links:

- db

depends_on:

- db

ports:

- '8080:8080'

db:

container_name: Postgres

image: 'itechpros/servoy:servoy_war_demo_dynamic_db_pg'

environment:

- POSTGRES_DB=servoyworld_2017

- POSTGRES_PASSWORD=demo

Servoy / Extending with Dynamic DB

Linking Containers Together

- Docker Cloud- Create multiple containers and link them together
/docker-cloud.yml

db:

environment:

- POSTGRES_DB=servoyworld_2017
- POSTGRES_PASSWORD=demo

image: 'itechpros/servoy:servoy_war_demo_dynamic_db_pg'

servoy:

environment:

- JAVA_XMS=128m
- JAVA_XMX=1024m

image: 'itechpros/servoy:servoy_war_demo_dynamic_db'

links:

- db

ports:

- '8080:8080'

- Deploy to Docker Cloud demo

Getting Started

Workflow Example

GitHub

- MyProject
 - Dockerfile
 - myOtherFile.txt



Configure Setup
Tags/Folders/Branches

DockerHub



- username/MyProject
 - v1
 - v2



Authenticate for private DockerHub Projects

DockerCloud



```
docker pull username/myproject:v1
```

Anywhere with Docker



```
docker pull username/myproject:v1
```

To Do

Still more work to do

- Better Documentation
- Support for Servoy 8.2, variables in the property file
- More testing of lower versions of Servoy (5-7)
- Hook into Jenkins for automated unit testing

Docker Gotcha's

Don't get burned by these!

- Disk Persistence
 - You loose all your data when your container restarts
 - By default
 - Use Volumes to save your data
 - volumes:
 - - /path/on/host:/path/on/container/data
 - Many other options available in the Docker Community
- When your default Docker process stops, your Docker container stops.
 - There are special situations to this (see my systemd images)
 - Docker is designed for the container to mirror a service.
 - So if your default process is not running, your Docker container stops.

Docker Gotcha's

Don't get burned by these!

- Bugs
 - Its Open Source, there are issues
 - ENTRYPOINT in Docker Compose wipes out CMD in Dockerfile
 - <https://github.com/docker/compose/issues/3140>
 - Good Community, quick responses
- Hardware Resources
 - “The cloud” isn't infinite, and its still running on hardware somewhere
 - Watch out for running out of Disk Space!
 - Create routines to purge out old unused images and volumes

Docker Gotcha's

Don't get burned by these!

- Security
 - Passwords in configuration files?
 - Docker Compose provides a “env_file” option
 - Pass in a path to an Environment file
 - Environment file contains secure info, not kept in source control
 - Developers keep it in a secure location
 - Used when running Docker Compose commands

Using Docker to Deploy your Servoy Applications

by Scott Butler

Questions?
Drinks?