

Scout and Docker

Scout UGM 2016, Ludwigsburg

Scout and Docker

Benefits

- Share your Application as a set of Docker Images
- Formalized Description of Setup to run App
- Pre-existing Images for 3rd-Party Images (Tomcat, ...)
- ...

Preparations

Installs

- Docker (and get some good Material about it)
- Eclipse Scout (and Docker Tooling for Eclipse)

Doing

- Create a Scout «Hello World»
- «Dockerize»

THE DOCKER BOOK

CONTAINERIZATION IS THE NEW VIRTUALIZATION



SIMPLE

A hands-on book that teaches you Docker™.



SCALABLE

Start small with a single container and then build on what you learn to deploy multi-container applications.



UP-TO-DATE

Current, accurate and up-to-date.



DIFFERENT

Written for both developers and sysadmins with real-world examples and use cases.

[MARKETS](#)[SEARCH](#)[ADVANCED SEARCH](#)[SEARCH](#)[MORE LIKE THIS](#)

- Eclipse Java EE Developer Tools
- doclipser
- Eclipse Web Developer

Eclipse Docker Tooling

[Eclipse Project](#)

★ 21

💬 0

[Install](#)[Details](#)[Screenshots](#)[Metrics](#)[Errors](#)[External Install Button](#)

The Eclipse Docker Tooling plugin provides the ability to manage Docker images and containers from within the Eclipse IDE.

Categories: [Linux Tools](#), [Tools](#)

Tags: [docker](#), [Containers](#), [Virtualization](#), [fileExtension_Dockerfile](#)

[ADDITIONAL DETAILS](#)

Eclipse Versions: Neon (4.6), Mars (4.5)

Platform Support: Windows, Mac, Linux/GTK

Organization Name: [Eclipse.org](#)

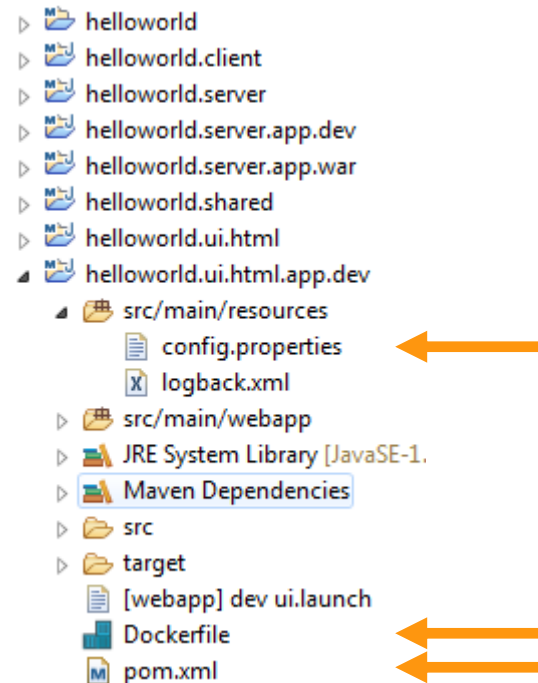
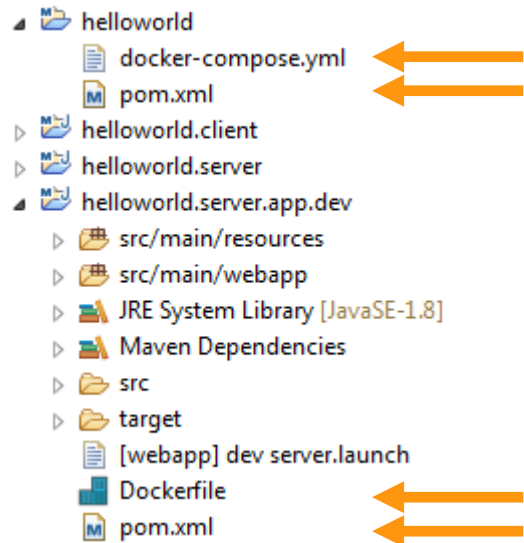
Date Created: Tue, 2016-02-02 16:42

Development Status: Production/Stable

License: FPI

Overview of Changes

Dockerize the «Hello World»



Root pom.xml

Add docker-maven-plugin

```
<name>My Application Parent Module</name>


<properties>
  <org.eclipse.scout.apps.helloworld.version>${project.version}</org.eclipse.scout.apps.helloworld.version>
  <jdk.source.version>1.8</jdk.source.version>
  <org.eclipse.scout.rt.version>6.0.100.RC4</org.eclipse.scout.rt.version>
  <docker.image.prefix>eclipse scout</docker.image.prefix>
</properties>

<build>
  <plugins>
    <plugin>
      <groupId>com.spotify</groupId>
      <artifactId>docker-maven-plugin</artifactId>
      <version>0.4.13</version>
      <configuration>
        <imageName>${docker.image.prefix}/${docker.image.name}</imageName>
        <dockerDirectory>.</dockerDirectory>
        <resources>
          <resource>
            <!-- https://github.com/spotify/docker-maven-plugin/issues/68 -->
            <!-- <targetPath>/</targetPath -->
            <directory>${project.build.directory}</directory>
            <include>${project.build.finalName}.war</include>
          </resource>
        </resources>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>
```



helloworld.server.app.dev

Add Dockerfile



```
<name>My Application Server WAR Development</name>
<properties>
  <docker.image.name>helloworld-server</docker.image.name>
</properties>
```

```
Dockerfile
1 FROM tomcat:jre8
2 MAINTAINER BSI-Business-Systems-Integration-AG
3
4 RUN echo "nameserver 8.8.8.8" > /etc/resolv.conf
5
6 ADD helloworld.server.app.dev-1.0.0-SNAPSHOT.war /usr/local/tomcat/webapps/helloworld.war
7
8 CMD ["catalina.sh", "run"]
```


helloworld.html.ui.app.dev

```
config.properties ✕  
1 ### Application theming  
2 scout.ui.theme=application  
3  
4 ### Application Config  
5 scout.application.version=1.0.0-SNAPSHOT  
6 scout.application.name=My Application  
7  
8 ### Service tunnel  
9 # scout.server.url=http://localhost:8080  
10 scout.server.url=http://app-server:8080/helloworld  
11 scout.auth.privatekey=MD0CAQAwEAYHkoZIZj0CAQYFK4EEAAoEJjAkAgEBBB9TsmjaGpeY2T4XD7kJpaG76Ku19q  
12
```

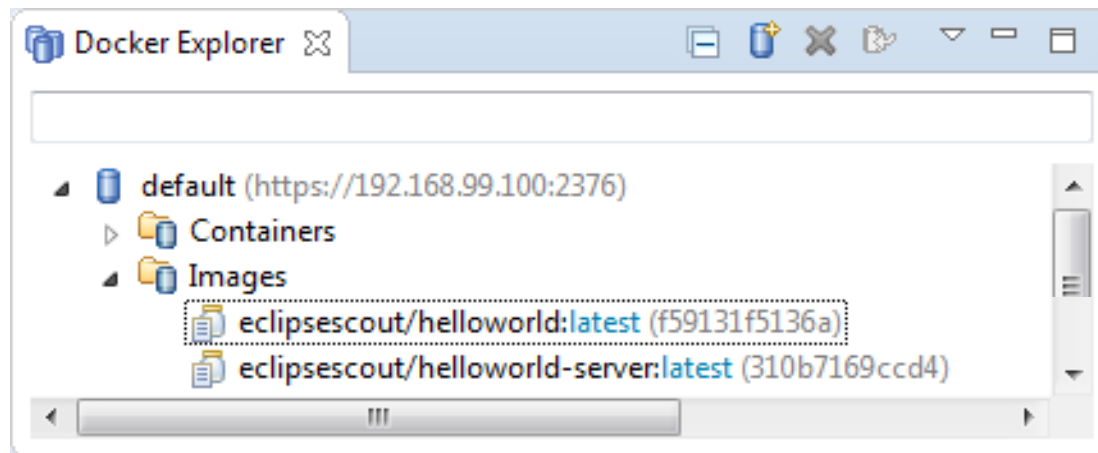
we will reuse **app-server** in the **docker-compose.yml** file

```
<name>My Application HTML UI Server Development</name>  
<properties>  
  <docker.image.name>helloworld</docker.image.name>  
</properties>
```

```
Dockerfile ✕  
1 FROM tomcat:jre8  
2 MAINTAINER BSI-Business-Systems-Integration-AG  
3  
4 RUN echo "nameserver 8.8.8.8" > /etc/resolv.conf  
5  
6 ADD helloworld.html.ui.app.dev-1.0.0-SNAPSHOT.war /usr/local/tomcat/webapps/helloworld.war  
7  
8 CMD ["catalina.sh", "run"]
```

Build the Docker Images

- `mvn clean package` for Root `pom.xml`
- `mvn clean package docker:build`
 - `helloworld.server.app.dev`
 - `helloworld.html.ui.app.dev`

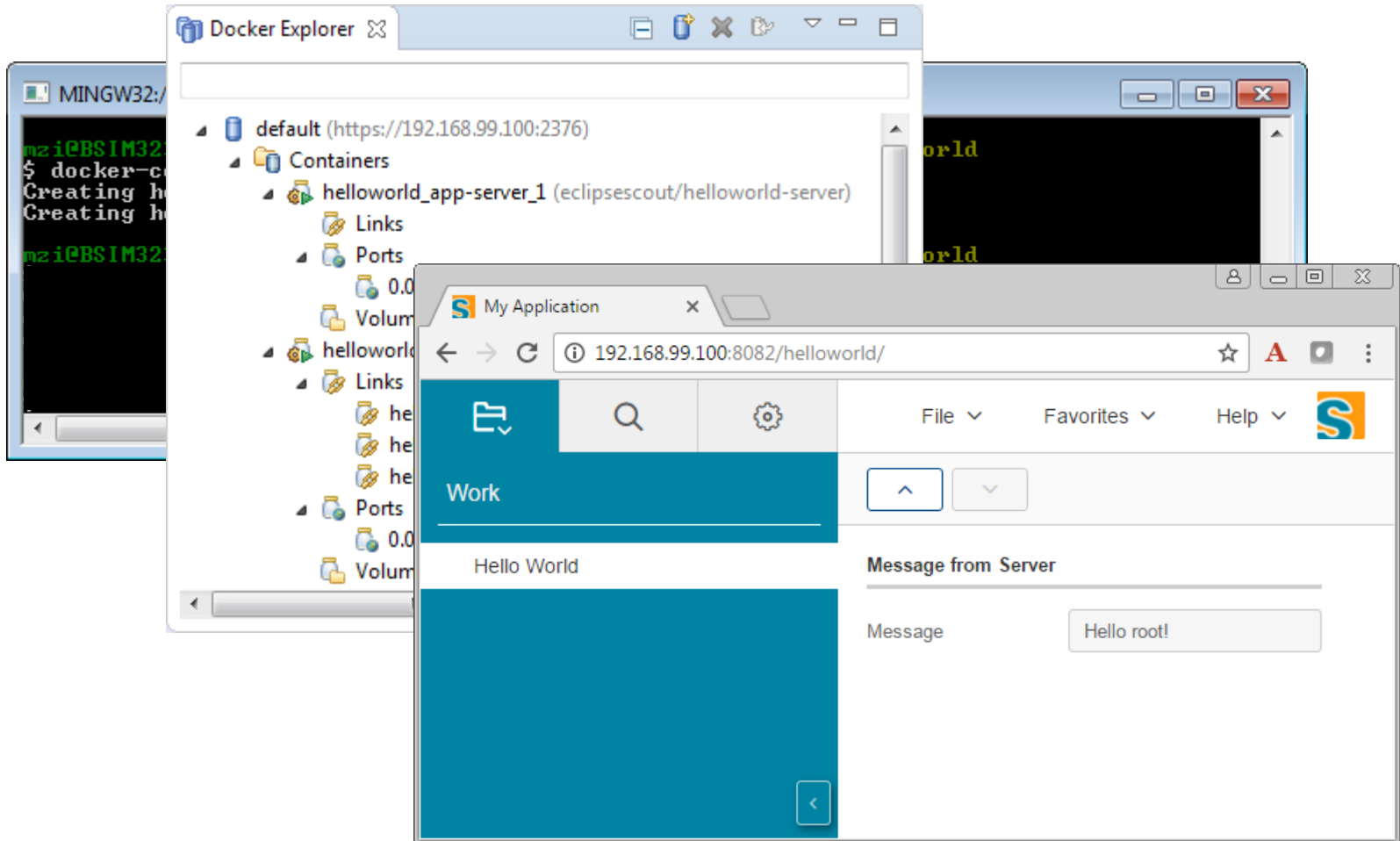


Define Multi-Container Setup docker-compose.yml

```
docker-compose.yml
1 web-server:
2   image: eclipsescout/helloworld
3   ports:
4     - "8082:8080"
5   links:
6     - app-server
7
8 app-server:
9   image: eclipsescout/helloworld-server
10  ports:
11    - "8080:8080"
12
```

remember **app-server** from the config.properties?

Run the «Hello World» docker-compose up



Thanks

@EclipseScout