Oracle Container Cloud
Case Study
Jagusztin László
The change challenge

- Quality of changes
- Speed of changes
- Frequency of changes
- Dependency of changes
- Cost of changes
Solution in the logistics industry

A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.

...in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another.
Solution in IT: Application Containers

An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container...

...that can be manipulated using standard operations and run consistently on virtually any hardware platform.
Docker Technology

The runtime infrastructure will be based on container technology (docker), meaning that:

• Many application can run on same machine not interfere with each other
• To move (or clone) an application to another machine is very easy
• The only app specific requirements for running host are:
  – Proper network infrastructure for app traffic (to fulfill the connection matrix)
  – Proper disk infrastructure for app data (e.g., shared, well sized, io)
Docker application container features:

• Application virtualization
• Deployment only in DEV
• No shipment in higher envs
• Containers with applications are exactly the same in all envs
• Application specific configuration in independent GIT repositories
• Central log collecting and monitoring
Flexible IaaS strategy
Oracle Container Service
Demo application
Container cloud is part of my cloud services
Test container cloud service is created
...with one manager and one worker node

<table>
<thead>
<tr>
<th>Overview</th>
<th>Service Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Nodes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nodes</th>
<th>OCPUs</th>
<th>Memory</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>15 GB</td>
<td>70 GB</td>
</tr>
</tbody>
</table>

**Status:** Ready  
**Version:** 17.1.5-1047

### OCCS Manager Component

- **State:** Ready  
- **Management node Compute Shape:** oc3

**Resources**

- **Host Name:** test-occs-mgr-1  
  - **Public IP:** 141.145.40.18  
  - **Instance:** Runs test manager 1  
  - **OCPUs:** 1  
  - **Memory:** 7.5 GB  
  - **Storage:** 35 GB

### OCCS Worker Component

- **State:** Ready  
- **Worker node Compute Shape:** oc3

**Resources**

- **Host Name:** test-occs-wkr-1  
  - **Public IP:** 141.145.40.244  
  - **Instance:** Runs test worker 1  
  - **OCPUs:** 1  
  - **Memory:** 7.5 GB  
  - **Storage:** 35 GB
Manager node running the console dashboard app
Configure external and internal registries, GIT repo.

<table>
<thead>
<tr>
<th>Actions</th>
<th>URL</th>
<th>Description</th>
<th>Email</th>
<th>Username</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>141.145.40.244:8443/occspec</td>
<td>Configuration Repo (git)</td>
<td><a href="mailto:lajos.sarecz@oracle.com">lajos.sarecz@oracle.com</a></td>
<td>occspoc</td>
</tr>
<tr>
<td>Edit</td>
<td>141.145.40.244:8444/occspec</td>
<td>Artifact Repo</td>
<td></td>
<td>occspoc</td>
</tr>
<tr>
<td>Edit</td>
<td>index.docker.io</td>
<td>Public Docker Hub Registry</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Create a new service
Use yaml format to specify container startup params.
Containers can be started or stopped from the GUI.
We can check container logs from GUI

Container Logs

Standard Out:

2017-03-20T14:02:35.059171447Z Launching GlassFish on Felix platform
2017-03-20T14:02:36.034453603Z Registered com.sun.enterprise.glassfish.bootstrap.osgi.EmbeddedOSGiGlassFishRuntime@1cf
2017-03-20T14:02:37.001911924Z
2017-03-20T14:02:37.001960208Z
2017-03-20T14:02:37.001965751Z
2017-03-20T14:02:37.001970178Z ## LogManagerService.postConstruct : rootFolder=/glassfish4/glassfish
2017-03-20T14:02:37.001975458Z ## LogManagerService.postConstruct : templateDir=/glassfish4/glassfish/lib/templates
2017-03-20T14:02:37.001981518Z ## LogManagerService.postConstruct : src=/glassfish4/glassfish/lib/templates/logging
2017-03-20T14:02:37.001986031Z ## LogManagerService.postConstruct : dest=/glassfish4/glassfish/domains/domain1/conf:

Standard Error:

2017-03-20T14:02:34.765967982Z Mar 20, 2017 2:02:34 PM com.sun.enterprise.admin.launcher.GFLauncherLogger info
2017-03-20T14:02:34.766046120Z INFO: JVM invocation command line:
2017-03-20T14:02:34.766054154Z /usr/lib/jvm/java-openjdk/bin/java
2017-03-20T14:02:34.766060371Z -cp
2017-03-20T14:02:34.766065813Z /glassfish4/glassfish/modules/glassfish.jar
2017-03-20T14:02:34.766070680Z -XX:+UnlockDiagnosticVMOptions
2017-03-20T14:02:34.766075593Z -XX:NewRatio=2
2017-03-20T14:02:34.766080589Z -XX:MaxPermSize=192m
2017-03-20T14:02:34.766095078Z -Xmx512m
2017-03-20T14:02:34.766101077Z -javaagent:/glassfish4/glassfish/lib/monitor/flashlight-agent.jar
2017-03-20T14:02:34.766106343Z -client
2017-03-20T14:02:34.766114102Z -Djavax.xml.accessExternalSchema=all
2017-03-20T14:02:34.766116599Z -Djavax.net.ssl.trustStore=/glassfish4/glassfish/domains/domain1/config/cacerts.jks
2017-03-20T14:02:34.766121857Z -Djdk.corba.allowOutputStreamSubclass=true
2017-03-20T14:02:34.766126604Z -Dfelix.fileinstall.dir=/glassfish4/glassfish/modules/autostart/
2017-03-20T14:02:34.766131895Z -Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo
2017-03-20T14:02:34.766141619Z -Dcom.sun.aas.installRoot=/glassfish4/glassfish
2017-03-20T14:02:34.766146157Z -Dcom.sun.aas.debug=true
2017-03-20T14:02:34.766146257Z -Dsun.net.http.socket.defaultmaxconnections=64
2017-03-20T14:02:34.766146357Z -Dsun.net.http.socket.httpmaxconnections=64
2017-03-20T14:02:34.766146457Z -Dsun.net.http.socket.cachingproxy=org.apache.felix.httpproxy.CachingProxy
2017-03-20T14:02:34.766146557Z -Dsun.net.http.socket.proxy=org.apache.felix.httpproxy.CachingProxy
2017-03-20T14:02:34.766146657Z -Dsun.net.http.socket.proxy.config=org.apache.felix.httpproxy.CachingProxy
We can connect to these nodes using ssh client.
We can also use `cmdline` for check and control.
Running container on internet, you can try it now!

https://141.145.40.244/DataExchangeConverter/
Next steps

- Implement security hardening
- Build VPN connection
- New applications
- Test engines
Thank You!

Questions