Containerizing legacy applications

OSCON Container Day
Jose De la Rosa – Linux Engineering
May 17th, 2016
Who am I?

- Linux Engineer (@jdelaros1)
- At Dell for 11+ years
- Emerging Technologies Evangelist
- Team’s mission statement: To make Linux “just work” on Dell notebooks, desktops and servers.
Containers are for microservices

- Microservices are awesome! 😊
- They are modular, immutable, scalable, etc.
- Example: launch web server, database server, middleware server and connect them all together.
- You can easily orchestrate them, cluster them, run them on bare metal, VMs, on the cloud.
But what about *macroservices*?

• What if you have a **large**, legacy application and you want to break it up and containerize it? Do it?

• Sure. If you have time and $, then do it.

• However, if time and $ are not available, should you place your application as-is inside a container??

• If you can do it with minimum effort, keep your environment secure and it *solves a problem* for you, then yes, containerize your legacy application as-is.
OpenManage Server Administrator (OMSA)

• OMSA is an in-band systems management tool, used to manage and monitor Dell PowerEdge servers.
• It’s been around since 1995.
• Ships with every server we sell.
• Services include web server, event managers, inventory collectors, event loggers, ipmi, snmpd, etc.
### Modular Enclosure Information

#### Chassis Information

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>PowerEdge VRTX</td>
</tr>
<tr>
<td>Lock</td>
<td>Not Present</td>
</tr>
<tr>
<td>Service Tag</td>
<td>G6428Y1</td>
</tr>
<tr>
<td>Express Service Code</td>
<td>35137671625</td>
</tr>
</tbody>
</table>

#### CMC Information

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Chassis Management Controller (CMC)</td>
</tr>
<tr>
<td>Description</td>
<td>This component provides a complete set of functions to manage the server modules present in the modular enclosure</td>
</tr>
<tr>
<td>Version</td>
<td>1.31</td>
</tr>
<tr>
<td>IP Address</td>
<td>10.9.162.241</td>
</tr>
<tr>
<td>IP Address Source</td>
<td>Dynamic Source</td>
</tr>
<tr>
<td>IP Address Type</td>
<td>IPv4</td>
</tr>
<tr>
<td>Remote Connect Interface</td>
<td>Launch the CMC Web Interface</td>
</tr>
</tbody>
</table>
### Processors Information

**Instructions:**
Click the connector name to view its capabilities and cache details.

<table>
<thead>
<tr>
<th>Status</th>
<th>Connector Name</th>
<th>Processor Brand</th>
<th>Processor Version</th>
<th>Current Speed</th>
<th>State</th>
<th>Core Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>CPU1</td>
<td>Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz</td>
<td>Model 45 Stepping 7</td>
<td>2200 MHz</td>
<td>Present</td>
<td>8</td>
</tr>
<tr>
<td>✔️</td>
<td>CPU2</td>
<td>Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz</td>
<td>Model 45 Stepping 7</td>
<td>2200 MHz</td>
<td>Present</td>
<td>8</td>
</tr>
</tbody>
</table>
Total of Memory Array(s)

- Total Installed Capacity: 49152 MB
- Total Installed Capacity Available to the OS: 48091 MB
- Total Maximum Capacity: 1572864 MB

Details of Memory Array 1

<table>
<thead>
<tr>
<th>Status</th>
<th>Connector Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DIMM_A1</td>
<td>DDR3 - Synchronous Registered (Buffered)</td>
<td>8192 MB</td>
</tr>
<tr>
<td></td>
<td>DIMM_A2</td>
<td>DDR3 - Synchronous Registered (Buffered)</td>
<td>8192 MB</td>
</tr>
<tr>
<td></td>
<td>DIMM_A3</td>
<td>DDR3 - Synchronous Registered (Buffered)</td>
<td>8192 MB</td>
</tr>
<tr>
<td></td>
<td>DIMM_A4</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIMM_A5</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIMM_A6</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIMM_A7</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIMM_A8</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIMM_A9</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIMM_A10</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIMM_A11</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIMM_A12</td>
<td>[Not Occupied]</td>
<td></td>
</tr>
</tbody>
</table>
Why containerize OMSA?

We wanted to solve two problems:

- **Problem 1**: OMSA creates a large footprint on host
  - Installs 89 RPM packages, including dependencies.
  - Upgrading or uninstalling can be cumbersome.
- **Problem 2**: Limited Linux distribution support
  - Select distros only (RHEL and SLES).
  - All others distros (i.e. Ubuntu and Debian) at best get unsupported, semi-tested builds.
Challenges to containerizing OMSA

Not an ideal application to run in a container:

• Many services (not just one)
• Large size (~700 MB)
• Has dependencies on kernel modules 😞
• Needs access to server’s hardware
Yes we can

FROM centos:centos7
MAINTAINER Jose De la Rosa "https://github.com/jose-delarosa"

ENV PATH $PATH:/opt/dell/srvadmin/bin:/opt/dell/srvadmin/sbin
ENV USER root
ENV PASS password

# Install missing packages needed for OpenManage
RUN yum -y update & & \\ yum -y install gcc wget perl passwd which tar libstdc++.so.6 compat-libstdc++-33.i686 glibc.i686

# Set login credentials
RUN echo "$USER:$PASS" | chpasswd

# Add OMSA repo. Let's use this DSU version with a known stable OMSA.
RUN wget -q -O - http://linux.dell.com/repo/hardware/DSU_16.02.00/bootstrap.cgi | bash

# Let's "install all", however we can select specific components instead
RUN yum -y install srvadmin-all & & yum clean all

# Prevent daemon helper scripts from making systemctl calls
ENV SYSTEMCTL_SKIP_REDIRECT=1

# Restart application to ensure a clean start
CMD srvadmin-services.sh restart & & tail -f /opt/dell/srvadmin/var/log/openmanage/dcsys64.xml
It “just works”

- 100% functionality
- No footprint on host
- Installation, upgrade and rollback now as easy as “docker run, docker stop and docker rm”
- It runs on *any* Linux distro that supports Docker
- No loss in performance
Problem #1 → Win #1

$ yum install srvadmin-all
Resolving Dependencies

→ Running transaction check
  → Package srvadmin-all.x86_64 0:8.3.0-1908.9058.e17 will be installed
  → Processing Dependency: srvadmin-webserver = 8.3.0 for package: srvadmin-all-8.3.0-1908.9058.e17.x86_64
  → Processing Dependency: srvadmin-storageservices = 8.3.0 for package: srvadmin-all-8.3.0-1908.9058.e17.x86_64
  → Processing Dependency: srvadmin-standardAgent = 8.3.0 for package: srvadmin-all-8.3.0-1908.9058.e17.x86_64
  → Processing Dependency: srvadmin-server-snmp = 8.3.0 for package: srvadmin-all-8.3.0-1908.9058.e17.x86_64
  → snip→
  → Processing Dependency: perl(HTML::Entities) for package: perl-libwww-perl-6.05-2.e17.noarch
  → Processing Dependency: perl(HTML::HeadParser) for package: perl-libwww-perl-6.05-2.e17.noarch
  → Processing Dependency: perl(URI::Escape) for package: perl-libwww-perl-6.05-2.e17.noarch
  → Processing Dependency: perl(Digest::MDS) for package: perl-libwww-perl-6.05-2.e17.noarch
  → Processing Dependency: srvadmin-storage-c11 = 8.3.0 for package: srvadmin-storageservices
  → Processing Dependency: srvadmin-storage = 8.3.0 for package: srvadmin-storageservices-c11
  → Package srvadmin-storageservices-snmp.x86_64 0:8.3.0-1908.9058.e17 will be installed
  → Processing Dependency: srvadmin-storage-snmp = 8.3.0 for package: srvadmin-storageservices-snmp-e17.x86_64
  → Processing Dependency: srvadmin-isvc-snmp = 8.3.0 for package: srvadmin-storageservices-snmp-e17.x86_64
  → Processing Dependency: srvadmin-idrac-snmp = 8.3.0 for package: srvadmin-storageservices-snmp-e17.x86_64

Dependencies Resolved

→ snip→

Running transaction
Installing : srvadmin-smcommon-8.3.0-1908.9058.e17.x86_64 1/89
Installing : srvadmin-racadm-8.3.0-1908.9058.e17.x86_64 2/89
Installing : srvadmin-xlmsup-8.3.0-1908.9058.e17.x86_64 3/89
Installing : srvadmin-hapl-8.3.0-1908.9058.e17.x86_64 4/89
Installing : srvadmin-argtable2-8.3.0-1908.9058.e17.x86_64 5/89
Installing : 1:perl-Compress-Zlib-2.061-4.e17.x86_64 6/89
Installing : srvadmin-sysfsutil-8.3.0-1908.9058.e17.x86_64 7/89
Installing : perl-Encode-Locale-1.03-5.e17.noarch 8/89

→ snip→

Installing : perl-IO-Socket-IP-0.21-4.e17.noarch 82/89
Installing : perl-IO-Socket-SSL-1.94-3.e17.noarch 83/89
Installing : perl-Net-HTTP-6.06-2.e17.noarch 84/89
Installing : perl-libwww-perl-6.05-2.e17.noarch 85/89
Installing : sbilm-sfcb-1.3.16-12.e17_0.x86_64 86/89
Installing : srvadmin-tunnelprovider-8.3.0-1908.9058.e17.x86_64 87/89
Installing : srvadmin-standardAgent-8.3.0-1908.9058.e17.x86_64 88/89
Installing : srvadmin-all-8.3.0-1908.9058.e17.x86_64 89/89

Complete!

$ docker run --privileged -d -p 1311:1311 --net=host \ -v /lib/modules/\uname -r*:/lib/modules/\uname -r` \ jdelarosi/openmanage
Problem #2 → Win #2
Some caveats

- Must run in *privileged* mode to access host’s hardware.
- Kernel module dependencies. Yuck.
- Container image is almost 1GB, including OS.
Moral of the story

• A legacy and bloated application can be containerized with minimum effort and without changing a single line of code.

• Didn’t exactly follow containerization best practices.

• Didn’t do anything fancy or clever.

BUT MOST IMPORTANTLY:

• We **solved** an existing customer problem.
Going from POC to supported solution

• Our current challenge is convincing marketing and development teams to embrace containers.
• Slowly getting there, but not fast enough.
• “No resources available”.
• “No customer demand”.
• Change is hard 😞
What else can you expect from Dell?

• Expect to see more Dell applications (existing and new) delivered via containers.

• Dell is working with Linux partners to ensure the entire container ecosystem (i.e. container-only operating systems, orchestration and clustering tools) “just work” on Dell platforms and software solutions.
Other Dell projects

• We are working on several projects in different areas, including Internet-Of-Things (IoT)

• In one particular IoT project, up to 25 services are being containerized
Storage Solutions

- Flocker driver for Dell Storage Center
- Enables persistent block storage in Dell Storage Center
- github.com/dellstorage/storagecenter-flocker-driver
Learn more

http://linux.dell.com/docker