Containers

Jan Hajek
v-jahaje@microsoft.com
@hajekj
Microsoft: The open-source company

Summary: Microsoft loves Linux, is adopting Docker for its servers, and just bought Revolution Analytics, the biggest open-source R statistical language company. This is not your dad’s Microsoft.

By Steven J. Vaughan-Nichols for Linux and Open Source | January 26, 2015 -- 19:04 GMT (11:04 PST)

Follow @sjvn | 7,962 followers | Get the ZDNet Cloud newsletter now
Bash on Ubuntu on Windows
Start from an image in Azure Marketplace

- Azure Endorsed Linux Distributions
  - Published, maintained and supported by partners, curated & tested by Microsoft
  - Standard Images
    - Customers contact Linux vendor/partner for Linux support
    - Azure-related platform issues supported by Microsoft
  - Premium Images
    - Microsoft engages the Linux vendor/partner on behalf of the customer for support
    - Includes updates, patches, and support through 24x7 web, email, chat and phone
    - SUSE SLES 11SP3 only
Where containers came from?
a very brief History of Shipping
Cargo transport pre-1960

- Multiplicity of goods
- Multiplicity of methods for transporting & storing

Do I worry about how goods interact?

Can I transport quickly and smoothly?
# Cargo transport pre-1960

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Barrels" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image2.png" alt="Crates" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image3.png" alt="Cars" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image4.png" alt="Barrels" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image5.png" alt="Pianos" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image6.png" alt="Bales" /></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Microsoft Azure
The Shipping Container analogy brings a solution

- Multiplicity of goods
- Multiplicity of methods for transporting & storing

Do I worry about how goods interact?
Can I transport quickly and smoothly?
so

What's

The Big Deal?
containers for code
Containers for code

Microsoft Azure

Application code
- dev

Infrastructure code
- test

Docker

- production
- staging
density & efficiency
Data center without containers
Data center with containers

Microsoft Azure
build once
run anywhere
micro services
A canonical PaaS platform

Author

Service manifest

Service code

Tooling

Programming model

Microsoft Azure

Run

Application manifest

Application

Image

Health

Messaging

Identity

ALM

Naming

State

Routing

Reliability

Coordinator

Request

Offer

Scheduler

Nodes

Repository / distribution

Package manager

Discovery

Community
The End
What Docker resolves?
The Shipping Container analogy brings a solution

Multiplicity of Stacks

Static website
nginx 1.5 + modsecurity + openssl + bootstrap 2

Queue
Redis + redis-sentinel

Web frontend
Ruby + Rails + sass + Unicorn

Analytics DB
hadoop + hive + thrift + OpenJDK

User DB

Do services and apps interact appropriately?

Multiplicity of hardware environments

Development VM

Can I migrate smoothly and quickly?

Customer Data Center

QA server

Public Cloud

Production Cluster

Contributor’s laptop

Microsoft Azure
Heterogeneity turns deployment into a nightmare

|----------------|--------|-----------|--------------------|----------------|--------------|----------------------|------------------|

Microsoft Azure
But it works for me...
The Shipping Container analogy brings a solution

Multiplicity of Stacks
- Static website
- User DB
- Web frontend
- Queue
- Analytics DB

Do services and apps interact appropriately?

Multiplicity of hardware environments
- Development VM
- QA server
- Customer Data Center
- Public Cloud
- Production Cluster
- Contributor's laptop

Can I migrate smoothly and quickly?
Heterogeneity turns deployment into a nightmare

<table>
<thead>
<tr>
<th>Static website</th>
<th>Web frontend</th>
<th>Background workers</th>
<th>User DB</th>
<th>Analytics DB</th>
<th>Queue</th>
<th>Dev VM</th>
<th>QA Server</th>
<th>Single Prod Server</th>
<th>Onsite Cluster</th>
<th>Public Cloud</th>
<th>Contributor's laptop</th>
<th>Customer Servers</th>
</tr>
</thead>
</table>

Microsoft Azure
Comparison: Containers vs. VMs

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

...result is significantly faster deployment, much less overhead, easier migration, faster restart.
Container Ecosystem

Container Run-Time
- Windows Server
- Linux

(Docker API/Client)

Container Images
- Windows
- (Docker images)

Image Repository
- (Docker Hub/Trusted Repositories)
Windows and Containers

- Windows Server introduces two container types:
  - Windows Server Containers
  - Hyper-V Containers

- Both provide high degree of Windows Server application compatibility

- Both integrated with Docker Engine so compatible with Docker Client and Docker APIs
Use Hyper-V Containers when:
- The host doesn’t trust the code it’s hosting in the container
- You want a version of Windows different than what’s running in the container host
Cattle Not Pets!
Q&A

Jan Hajek
v-jahaje@microsoft.com
@hajekj