Containers: Best Practices & Data Management Services

Live Webcast

December 7, 2016
10:00 am PT
The material contained in this presentation is copyrighted by the SNIA unless otherwise noted.

Member companies and individual members may use this material in presentations and literature under the following conditions:

- Any slide or slides used must be reproduced in their entirety without modification
- The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.

This presentation is a project of the SNIA.

Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.

The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.
SNIA Presenters

Andrew Sullivan
Technical Marketing Engineer
NetApp

Keith Hudgins
Technical Alliances
Docker

Alex McDonald
Chair SNIA Cloud Storage Initiative
Agenda

1. Intro to Containers and how they enable DevOps & CI-CD
2. What is Docker?
3. Persistent Storage for Containers
4. Docker Case Studies
Containers 101

- Virtualization of application instead of hardware
- Runs on top of the core OS (Linux or Windows)
- Doesn’t require dedicated CPU, Memory, Network—managed by core OS
- Optimizes Infrastructure—speed and density

"Containerization seems poised to offer both a complement and a viable alternative to server virtualization" (1) IDC
Driving Factors for Containers

- Density & Performance
- Licensing Costs
- Shift to DevOps
- Cloud-native Applications (Scale-out)
- Faster Exploration & Deployment (CI/CD)

"Containerization seems poised to offer both a complement and a viable alternative to server virtualization" - IDC
Stateful vs Stateless

Stateful container apps represent the next big IT challenge (1)

Persistent storage among top issues for container enterprise-readiness in production (2)

Stateful Database applications such as Redis, MySQL, MongoDB among most pulled images on Docker Hub (2)

Importance of Container Orchestration Abilities

<table>
<thead>
<tr>
<th>Long-running applications</th>
<th>Snow</th>
<th>Ice</th>
<th>Rock</th>
<th>Mud</th>
<th>Sand</th>
</tr>
</thead>
<tbody>
<tr>
<td>68%</td>
<td>18%</td>
<td>5%</td>
<td>5%</td>
<td>3%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load balancing</th>
<th>Snow</th>
<th>Ice</th>
<th>Rock</th>
<th>Mud</th>
<th>Sand</th>
</tr>
</thead>
<tbody>
<tr>
<td>58%</td>
<td>10%</td>
<td>11%</td>
<td>8%</td>
<td>3%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Persistent storage</th>
<th>Snow</th>
<th>Ice</th>
<th>Rock</th>
<th>Mud</th>
<th>Sand</th>
</tr>
</thead>
<tbody>
<tr>
<td>38%</td>
<td>32%</td>
<td>14%</td>
<td>16%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Batch applications</th>
<th>Snow</th>
<th>Ice</th>
<th>Rock</th>
<th>Mud</th>
<th>Sand</th>
</tr>
</thead>
<tbody>
<tr>
<td>27%</td>
<td>16%</td>
<td>32%</td>
<td>22%</td>
<td>3%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-contained workloads</th>
<th>Snow</th>
<th>Ice</th>
<th>Rock</th>
<th>Mud</th>
<th>Sand</th>
</tr>
</thead>
<tbody>
<tr>
<td>11%</td>
<td>8%</td>
<td>25%</td>
<td>28%</td>
<td>28%</td>
<td></td>
</tr>
</tbody>
</table>

(1) The New York Survey, March 2016. When evaluating container orchestration tools, how important following requirements? n=107. Due to rounding, figures may not equal 100%.
Does Persistence Matter?

- Containerized applications are still applications
- Not every service requires persistence
- Accessing data at every stage of the application life cycle benefits the application and the business
Persistent Storage - Why

**Data Availability**
Data needs to be always on no matter what happens

**Data Accessibility**
Run containers anywhere without worries about where data is located

**Deployment Cycles**
Traditional storage approaches slow-down innovation

**Storage Costs**
Storage defined by Software using any commodity HW or Cloud
Types of Persistent Data

- Configuration data
- Instance application data
- Shared application data
- as-a-Service data
Configuration Data

- Typically not stored with the application
  - Database details
  - Declaration of purpose
  - Specific tasking
- Passed into application or accessed at startup
  - Environment variables
  - Etcd, Zookeeper, Consul, etc.
Instance Data

- Data created by a container instance which does not need to persist beyond the life of the container
  - Cache data
  - Staging data / scratch space
- Container density can lead to high IO requirements for local drives
Data which is shared across multiple instances of a container image
  - Apache htdocs folder

Data which is shared across services
  - Against 12 Factor principles
  - But, happens anyway
  - E.g. image processing service
as-a-Service Data

- Any storage which is consumed as-a-Service
- DBaaS
  - RDS, DynamoDB
  - Trove
- Object Storage
  - S3
  - Swift
Where is my data?

- Local/host-based persistence
  - Stored on local drives of the host
- Container-based storage services
  - Stored on local drives of the host, but application manages availability and protection
- Array-based storage
  - Enterprise storage array with associated features
Docker Storage Types

- **Registry**: Cold storage of container images
- **Graph**: Active storage of running container images
- **Volume**: Persistent block storage for data
Docker Registry Storage

- Registry service runs as container underneath Docker Engine
- Config data stored via standard Docker volumes
- Images stored via driver
- Native filesystem (We don’t care what’s beneath - attached/remote storage is best!)
- Drivers available for cloud object storage for images (S3, Swift, GCS)
- No heavy lifting required to integrate
Docker Storage
Docker Graph Driver

- Used for actively running containers on a Docker Engine host
- Writes to local filesystem (Can be backed by network storage, depending on driver)
- Image is copy-on-write
- Diff layers are removed when container is deleted

https://docs.docker.com/engine/userguide/storagedriver/selectadriver/
Docker Storage
This is where persistent data lives.
Extremely pluggable.
Network attached storage is extremely useful here.
  - Driver API is simple, easy to implement.
  - [https://docs.docker.com/engine/extend/plugins_volume/](https://docs.docker.com/engine/extend/plugins_volume/)
Supports both software and hardware based storage management.
Agenda

1. Intro to Containers and how they enable DevOps & CI-CD
2. What is Docker?
3. Persistent Storage for Containers
4. Docker Case Studies
Docker Case Studies:
Brief overviews of recent success stories for Docker customers
Case Study: ADP

Before

<table>
<thead>
<tr>
<th>App A</th>
<th>App B</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Service</td>
<td>App Service</td>
</tr>
<tr>
<td>Auth</td>
<td>Auth</td>
</tr>
<tr>
<td>App Reg</td>
<td>App Reg</td>
</tr>
<tr>
<td>Marketplace</td>
<td>Marketplace</td>
</tr>
<tr>
<td>Logging</td>
<td>Logging</td>
</tr>
<tr>
<td>…more</td>
<td>…more</td>
</tr>
</tbody>
</table>

Common services in monoliths are turned turned into base applications stored in the Trusted Registry available to all app teams

After

Teams request into central IT maintained portal/registry to provision infrastructure and pull base images

Monoliths are now micro services applications. Each app has it's own containers based on the same base image
Case Study: SA Home Loans

SA Home Loans uses Docker Datacenter to convert Monoliths to Microservices

Goal
• Convert monolithic .Net applications (built in Mono) into microservices

Result
• Evaluated Docker running small-scale postgres services across 2 nodes
• Docker’s enterprise-class networking and security capabilities were key but impressed with the ease-of-use of Docker Native orchestration - Swarm
• Currently running Docker Datacenter across 4 nodes as they are working to Dockerize all enterprise-class applications in the next few months
Case Study: GSA (Booz Allen)

Challenge
• Migrate away from monolithic application
• Long and cumbersome application development cycles

Solution
• Build a new developer platform (IAE Common Service Platform) with Docker Trusted Registry and commercially supported Docker Engine on AWS

Benefit
• Improved customer centric services Reduced time-to-market
• Improve security and reduce attack surface area
But Wait, There’s More!

- Other SNIA Webcasts on Containers:
  - Intro to Containers, Container Storage and Docker
    https://www.brighttalk.com/webcast/663/217971
  - The State of Storage in the Container World
    https://www.brighttalk.com/webcast/663/225901
  - Stay updated! Join our Containers opt-in email list
After This Webcast

- Please rate this Webcast. We value your feedback.
- This Webcast and a copy of the slides will be on the SNIA Cloud Storage website and available on-demand:
  - [http://www.snia.org/forum/csi/knowledge/webcasts](http://www.snia.org/forum/csi/knowledge/webcasts)
- A Q&A from this webcast, including answers to questions we couldn't get to today, will be on the SNIACloud blog:
- Follow us on Twitter @SNIACloud
Thank you.

Keith.Hudgins@docker.com
Andrew.Sullivan2@netapp.com