RECap: RunEscape Capsule for On-demand Managed Service Delivery in the Cloud

Shripad J Nadgowda, Sahil Suneja, Canturk Isci

IBM T J Watson Research Center
Evolution of application runtimes
(General-purpose → Specialized)

- **General Purpose OS (e.g. Ubuntu)**
  - Apps
  - Deps

- **General Purpose OS (e.g. Ubuntu)**
  - Apps
  - Deps

- **Lightweight OS (e.g. Alpine)**
  - Apps
  - Deps

**Physical Machines/VMs**

**Containers**

**Lightweight runtimes**

**Microcontainers**

But, you promised that containers will be "lightweight" alternatives to VMs?

But, according to standard DevOps practice, container should be immutable?

Can you reduce TCB for secure containers?
Evolution of application runtimes (General-purpose —> Specialized)

Physical Machines/VMs

Containers

Lightweight runtimes

Microcontainers

# of images

0 10 20 30 40 50 60 70 80 90 100 110

general-purpose OS lightweight OS scratch statically compiled
Evolution of application runtimes
(General-purpose —> Specialized)

Physical Machines/VMs

General Purpose OS (e.g. Ubuntu)

Deps

Apps

Containers

General Purpose OS (e.g. Ubuntu)

Deps

Apps

Lightweight runtimes

Lightweight OS (e.g. Alpine)

Deps

Apps

Microcontainers

You really did everything you promised. But now, how do I manage my containers?
3 Rules for Modern Container Cloud

**Rule 1**

*Functionality Dis-aggregation:* Break your traditional monolithic applications into two parts, namely—core application functions and other auxiliary functions.

**Rule 2**

*Use Micro-containers:* Package and run your core application functions through micro-containers for safer execution.

**Rule 3**

*On-demand Auxiliary functions:* Enable delivery common auxiliary functions as on-demand managed services on cloud.
What are these auxiliary functions?

**System and Application Administration**
- Typical cronjobs: malware scans, logrations, ntpupdate
- Application Utilities: MD5 checksum, archival with tar

**Debugging**
- Ad-hoc debugging: gdb, strace, tcpdump, iperf

**Monitoring**
- System metrics: CPU, Memory
- Application Metrics: # of connections, # of requests

How to deliver these functions on-demand and securely to running containers?
Inspiration…

Serverless
or
Function-as-a-Service
Introducing RunEscape Capsule (RECap)  
(also stands for Capability Redemption)

Capsule

- This itself is a micro-container
- Encapsulates auxiliary function and all its dependencies together
- Existing tools and techniques (e.g. Dockerfiles) can be leveraged to create an image

RunEscape

- Capsule are dynamically-attached to app containers as a sidecar container
- Capsule is attached ONLY for the duration of running the function
- Capsule is then detached or Escape the app container
Although agnostic to the underlying cloud substrate, we are currently designing RECap for Kubernetes.
RECap: System Design

Affinity between capsule and app container is established through K8s labels
RECap: System Design

A special node agent “caplet” manages lifecycle of capsule containers
# RECap: Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Image Build</th>
<th>App Deploy</th>
<th>Docker Exec</th>
<th>Capsule Run(\text{Escape})</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function exists in app container</strong></td>
<td>0</td>
<td>0</td>
<td>0.083s</td>
<td>0</td>
</tr>
<tr>
<td><strong>Function does not exist in app container</strong></td>
<td>5.24s</td>
<td>0.29s</td>
<td>0.083s</td>
<td>0</td>
</tr>
<tr>
<td><strong>Capsule Image is present on the node</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.243s</td>
</tr>
<tr>
<td><strong>Capsule Image is not present on the node</strong></td>
<td>6.2s</td>
<td>0</td>
<td>0</td>
<td>0.243s</td>
</tr>
</tbody>
</table>
Recap on RECap

**Trend**

Breaking monolithic applications into Microservices and delivering them in Microcontainers

**Challenge**

Ability to implement common system administration, debugging, monitoring functions on-demand

**Solution**

RunEscape Capsule is a framework that promotes a cloud-native solution for on-demand managed service delivery
RECap: Discussion

What is the criteria for deciding which functions can be de-coupled from application and delivered through RECap?

Is it safe to dynamically execute on-demand functions in the application context?

Whom does RECap is really going to help? Developer, Administrator, Cloud provider, Everyone?
Thank You

nadgowda@us.ibm.com