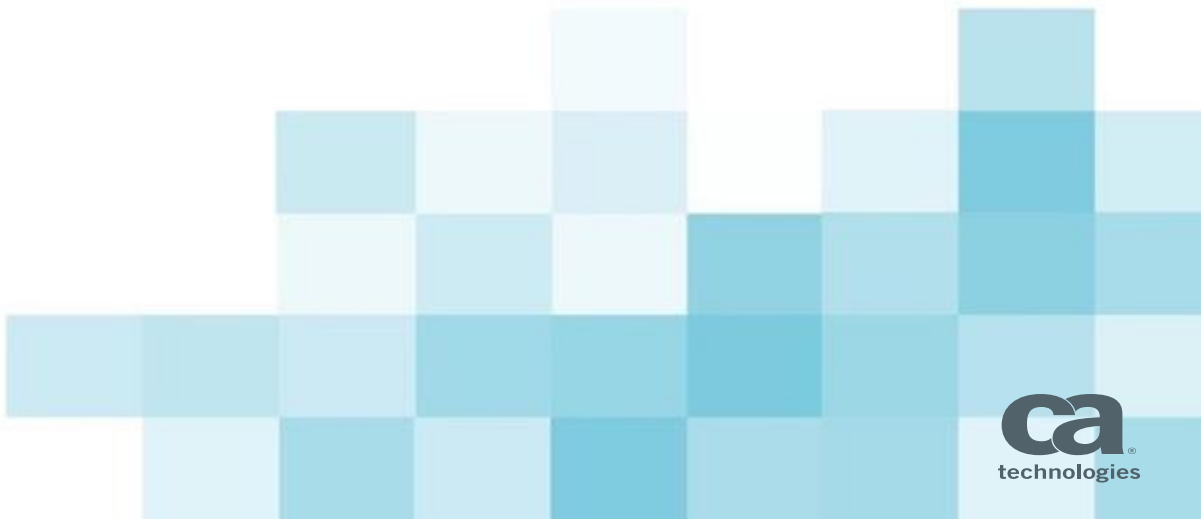


Automation for Container and Cloud-based applications

September 26, 2016



Legal Statement

© 2016 CA. All rights reserved. CA confidential & proprietary information. For CA, CA Partner and CA Customer use only. No unauthorized use, copying or distribution. All names of individuals or of companies referenced herein are fictitious names used for instructional purposes only. Any similarity to any real persons or businesses is purely coincidental. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. These Materials are for your informational purposes only, and do not form any type of warranty. The use of any software or product referenced in the Materials is governed by the end user's applicable license agreement. CA is the manufacturer of these Materials. Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-19(c)(1)-(2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Workload Automation –Application Deployment Models

Use Case & Solution Concept



Application Developer,
Operations

Use Case: As an application developer, I need to easily monitor and automate workloads for my container and cloud-based applications so that I can deliver applications quicker and realize better business outcomes.

Biz Agility

- ✓ Enable Bi-modal IT
- ✓ Manage SaaS/On-Prem workloads
- ✓ Implement DevOps Practices for Faster Delivery

Automate bi-modal workloads from single solution

Automation of Container & Cloud based apps

- Container and cloud friendly application automation
- Support bimodal infrastructures
- API-based communication for monitoring and automating workloads
- Support provisioning of containers and cloud VMS for on-demand workloads
- End-point monitoring and control
- Automated triggering based on container, cloud machine provisioning
- Support variety of container managers, cloud providers
- Metering, resource monitoring and analytics

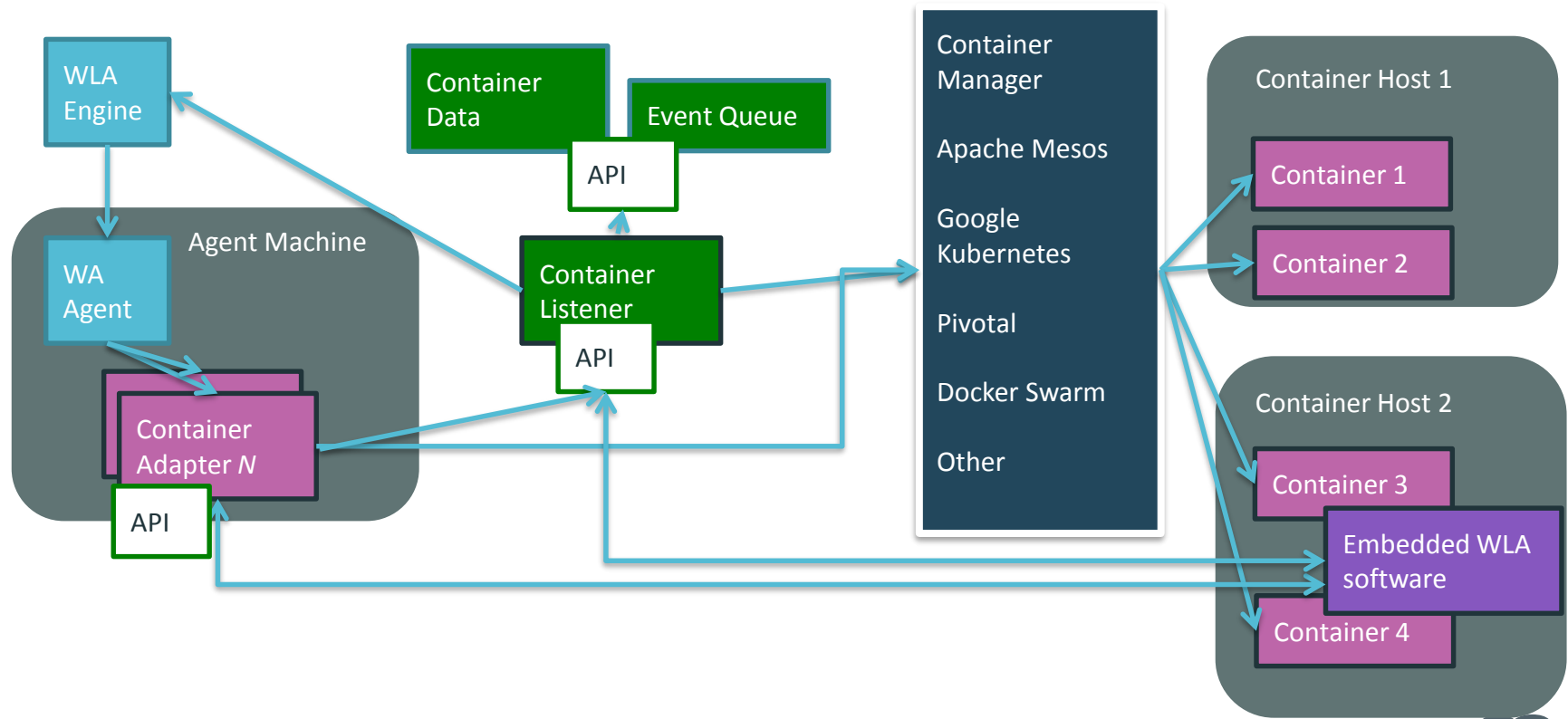
Containers: Use Cases

- **Execute processes inside containers without SSH**
 - Orchestrate a container to run a single process
 - Run sequential and concurrent processes inside active containers
- **Monitor and control commands running inside containers**
 - Synchronize duration of WLA jobs with processes being executed
 - Kill job -> Stop/kill associated processes and optionally, stop containers
 - Real-time access to output generated by associated processes
 - Track host and system metrics for commands running inside containers
- **Monitor container deployment and trigger workloads**
 - Track relevant containers being started and stopped
 - Trigger workloads when relevant containers are active
 - Track host and system metrics for relevant containers

Containers: Use Cases

- **Analysis and Analytics**
 - View output generated by processes and containers that are no longer running
 - Measure resource consumption across versions of containers and deployment location
 - Assign optimal resource values before deploying containers
- **Container Managers**
 - Support industry-standard managers such as Pivotal, Google Kubernetes, Mesos, Swarm
 - Support home-grown managers such Morgan Stanley Treadmill
- **User Experience**
 - Integrate with WLA interfaces for single point of monitoring and control
 - Support job flows across containerized processes and currently supported job types across all infrastructures

Proposed Architecture



WLA Container Orchestrator: Listener

- **Capabilities with Container Managers**
 - Read Docker registries
 - Fetch data from various container managers
 - Provide ability to connect to customized container managers
 - Track status of relevant application containers
 - Track host and system metrics for active relevant containers
- **Capabilities with Container Data**
 - Store association between containers and container managers
 - Store relevant API calls and configurations for each container manager
 - Update container data as fetched from container managers
 - Sync container data in case of listener processes recovery

WLA Container Orchestrator: Listener

- **UI/API capabilities**
 - Fetch data regarding container images, container status, metrics.
 - Update container status, metrics
 - Fetch data regarding configured container managers
 - Update configuration of container managers
 - Fetch data regarding configured WLA engines
 - Update configuration of WLA engines

WLA Container Orchestrator: Adapter

- **Capabilities with Container Managers**
 - Start a container and run processes
 - Run a process inside an active container
 - Stop/kill processes and/or containers
 - Fetch real-time output from submitted processes inside containers
- **Capabilities with WLA Engine**
 - Accept details about container and process to run
 - Pass through information to WLA engines via System Agent
 - Run as a normally supported job type

WLA Container Orchestrator: Embedded Software

- **Capabilities with Containers**

- Easily embed inside containers
- Run a process inside an active container
- Stop/kill processes and/or containers
- Fetch real-time output from submitted processes
- Track host and system metrics for submitted processes

- **Capabilities with Listener**

- Self-register to accept jobs
- Show container host as final destination of container processes