

# Trending Emerging Technologies Benefiting Government

## Containerization

*Session will begin at 1:00*

## Trending Emerging Technologies Benefiting Government Containerization Agenda

- **Leveraging Container Technology**  
*Bernard Clairmont, CA Technologies*
- **Containerization at DSNY**  
*Sriram Vasantha, NYC Department of Sanitation*
- **Containerization Demo**  
*Tej Tenmattam, Oracle Public Sector*
- **Speaker Q & A Panel**



# NYS Forum Presentation

Department of Sanitation – Containerization

---

March 2016



# Speaker Bio

## Sriram Vasantha - Development Manager

- Over 18 years in the IT industry designing and building complex n-tiered systems
- Leading SMART Release 2 redesign and development using
  - MicroServices
  - Containerization using Docker
  - Agile Methodology



# Agenda

- Overview of SMART 1.0
- Challenges Before Containerization
- Choosing a Container Technology
- Containerization inside DSNY
- Benefits from Containerization
- Lessons Learned
- Parting Thoughts
- Open Discussion



# Overview of SMART 1.0

- Replacement for Paper Board





# Overview of SMART 1.0 – cont.

- Open-source Software Stack
  - Spring Framework
  - HTML5
  - Websockets
  - Angular JS
  - AMQP
  - REST/SOAP/Webservices
  - RabbitMQ
  - Java8



# Overview of SMART 1.0 – cont.

## Finished Product

**Tuesday Ops-Board**  
Sep 15, 2015

**Tasks - District**

**SIB0**

0000-0800  
SWAN  
R. Chee

0800-1600  
G1  
W. Koppe  
MDA-Watchman  
L. Buono

0600-1400  
DS  
C. Abbate  
XD  
D. Brusella  
FO1  
S. Woolverton  
FO2  
H. Vallibiera  
FO3  
J. Triano  
FO4  
M. Patricia  
MDA-Watchman  
ANNEX  
J. Fronzeph

0700-1500  
G2  
A. Rivera

**SIB1**

0000-0800  
MDA  
G. Maldonado  
T. Smith

1600-0000  
G3  
J. Fennel  
NDS  
J. Calica  
MDA-Watchman  
C. Simoncini

MDA  
W. Arrabona  
C. Almon  
S. Holmes  
G. Maldonado  
T. Smith  
C. Simoncini

0800-1600  
G2B  
M. Polito  
Support-Order to HCF  
HCF  
C. Almon  
MDA-Watchman

**SIB3**

0000-0800  
G1  
A. Buzza

0600-1400  
D/S  
32CR-117  
J. Cocozza  
D/S  
J. Addeo  
XD  
P. Agreste  
FO1  
32CK-326  
P. Maggio  
FO2  
31DC-152  
R. Carroll  
FO3  
31DC-432  
E. Segal

0700-1500  
G2  
C. Jacobson

0800-1600  
G2  
N. Gaglio  
\*FO4  
Wersheimer

1600-0000

Support-Watchman/Security  
G3  
J. Halstoe  
MDA  
S. Barron  
N. Mason  
MDA  
S. Barron  
M. Masch  
Unavailable  
Charts (3)  
R. Lasser  
J. Oarmely  
S. Barron  
XWP (2)  
J. Oarmory  
A. Engelman  
XWOP (1)  
C. Averack  
LODI (1)  
J. Buzza

**SIBO**  
NEW YORK CITY DEPARTMENT OF SANITATION  
SMART Operations Board





# Challenges Before Containerization

## From IT Operation's Perspective

- Provisioning Environments takes days/weeks
- Deployment takes a long time
- Issues during deployment or post deployment, the developer says “but it works on my machine!”
  - Unknown differences between environments
    - Missing Dependencies
    - Version Differences
    - DLL Hell
    - OS Difference (ex Debian for Test vs. RedHat Enterprise for Prod)
    - Network Topology
    - Security Policies
    - Storage
    - Configuration Drift



# Challenges Before Containerization – cont.

## From IT Infrastructure Perspective

- Heterogeneous Deployment environments
  - Public Cloud
  - Private Cloud
- VMs are heavyweight/expensive/slow for large scale out and massive clustering needs
- Infrastructure as Code



# Challenges Before Containerization – cont.

## From a Developer's Perspective

- Longer time for development environment setup
- Difficult to work simultaneously on multiple projects with different versions of the same software (*DLL Hell*)
- Disparity between Production and Developer Sandboxes
- Inconsistency between developer environments in the same team



# Challenges Before Containerization – cont.

## Other Disruptive IT Forces

- Traditional Build/Package/Deployment/Operations Hand-off model is not good enough with these newer and disruptive IT concepts
  - Agile Development
  - MicroServices
  - DevOps



# Choosing a Container Technology

## Key Capabilities/Features to look for

- Portability
- Open Standards
- Caching
- Isolation
- Security
- Registry
- Orchestration/Scheduling
- Monitoring
- Logging
- Image Format
- Performance
- Ecosystem



# Containerization inside DSNY

- Monolith Application broken down into Micro(Business) Services
- Each Service is in its own Container
- Almost everything runs in a container – Services/Build Server/Registry/Orchestration Tool
- Development Environment is identical to a Prod setup except for the difference in capacity
- Created a Private Cloud Container Service using a Container Orchestration Service
- CI Server builds Docker images, pushes it to Registry and deploy those images to our Private Container Cloud Container Service



# Containerization inside DSNY - cont

- Docker Engine/ Docker Machine
- Eco-System surrounding Docker
  - Rancher – Orchestration/Scheduling
  - Artifactory – Docker Registry
  - CloudBees – Continuous Integration Server
  - Gradle – Build Script
  - ELK Stack – Logging
  - Monitoring



# Benefits from Containerization

- Rapid deployment to environment and reduced down time
  - DSNY SMART used to take a minimum of 40 minutes to deploy across an 8-server cluster
  - Now Individual containers can be deployed in a few seconds– containers can be deployed individually
  - Allows for flexible deployment approaches such as rolling deployments to eliminate down-time altogether
- Deploy the container rather than the application
  - Eliminating unknowns that cause, “But it works on my machine!”
  - The same container image built by our CI process gets deployed to all environments
  - Containers are immutable
- Heterogeneous deployment environments
  - Hosts from DOITT, DSNY and external Cloud Vendors can be added to our Private Cloud Container Service easily and seamlessly.





# Benefits from Containerization – cont.

- Ability to treat Infrastructure as code
- Ability to stand up new environments dynamically, quickly and cheaply
- Ability to scale up environments dynamically and quickly
- Development Environment can be setup in an hour vs. day(s)
- Parity between Development and Prod Environments
- Clear separation of Responsibility between Developers and Operations
  - Emergence of DevOps
  - Developers are responsible for what happens inside vs. operations for what happens outside
  - Ability to experiment and use new Technologies



# Benefits from Containerization – cont.

- Ability to achieve true decoupling of MicroServices by Containerizing them
- Ability to Secure environments independently
- Ability to Deploy and run different versions of Application Services at the same time



# Lessons Learned

- Developers need to have DevOps mindset
- Get Management, IT Infrastructure and Operations Buy-In
- There is more to the ecosystem than just the containerization technology of choice
- Go for Open-Source and add Support
- Initial ramp up time required for moving to Containers
- Still a bleeding-edge technology



# Parting Thoughts

- Consider Containers for Application Packaging
- Consider running Containers to increase density on hardware
- VM and Containerization technologies are not competitive but rather complimentary
- Containers are not a silver bullet



# Open Discussion

Questions or Comments?



# Contact Information

## Edmund Lee

Chief Information Officer  
Bureau of Information Technology  
NYC Department of Sanitation  
[elee3@dny.nyc.gov](mailto:elee3@dny.nyc.gov)

## **Trending Emerging Technologies Benefiting Government**

Containerization

Demonstration

**Tej Tenmattam**

Principal Solution Specialist

Oracle Public Sector

[tej.tenmattam@oracle.com](mailto:tej.tenmattam@oracle.com)

# Q & A



**Bernard Clairmont**  
**CA Technologies**



**Tej Tenmattam**  
**Oracle Public Sector**



**Sriram Vasantha**  
**NYC Department of Sanitation**