NIST Special Publication 800-180 (DRAFT)

NIST Definition of Microservices, Application Containers and System Virtual Machines

Anil Karmel
Ramaswamy Chandramouli
Michaela Iorga

This publication is available free of charge

COMPUTER SECURITY

National Institute of Standards and Technology
U.S. Department of Commerce

NIST Special Publication 800-180 (DRAFT)

NIST Definition of Microservices, Application Containers and System Virtual Machines

Anil Karmel
C2 Labs, Inc.
Reston, VA

Ramaswamy Chandramouli
Michaela Iorga.
Computer Security Division
Information Technology Laboratory

This publication is available free of charge

February 2016

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

Authority

This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3541 *et seq.*, Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-180
Natl. Inst. Stand. Technol. Spec. Publ. 800-180, 12 pages (February 2016)
CODEN: NSPUE2

This publication is available free of charge

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST Computer Security Division publications, other than the ones noted above, are available at http://csrc.nist.gov/publications.

Comments on this publication may be submitted to:

Public comment period: *February 18, 2016* through *March 18, 2016*

All comments are subject to release under the Freedom of Information Act (FOIA).

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: sec-cloudcomputing@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Abstract

Many variations and definitions of application containers exist in industry, causing considerable confusion amongst those who attempt to explain what a container is. This document serves to provide a NIST-standard definition to application containers, microservices which reside in application containers and system virtual machines. Furthermore, this document explains the similarities and differences between a Services Oriented Architecture (SOA) and Microservices as well as the similarities and differences between System Virtual Machines and Application Containers.

Keywords

Application Containers; System Virtual Machines; Microservices; Services Oriented Architecture

Acknowledgements

Audience

The intended audience of this document is system planners, program managers, technologists, and others as consumers or providers of cloud services.

Compliance with NIST Standards and Guidelines

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2014, Public Law 113-283.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), "Securing Agency Information Systems," as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

Executive Summary

Ubiquitous deployment of server or hardware virtualization has created a good understanding of the semantics of the term Virtual Machines (VMs). Similarly, the web services deployment paradigm that has been in vogue since the 1990's to the 2000's has created a fair agreement on what constitutes a Service-Oriented Architecture (SOA).

However, a relatively recent trend is operating system-level virtualization using the concept of application containers that run as isolated user space processes on top of an OS's kernel. Because of the close similarity between the core function provided by application containers and VMs (i.e., isolation), there is a need to provide a formal definition of both these terms and outline their similarities and differences. Further, these application containers are self-contained application packages and are built using OS/library/binary components each providing an OS-level capability.

Applications are decomposed into discrete components based on capabilities as opposed to services and placed into application containers with the resulting deployment paradigm called a Microservices Architecture. This Microservices Architecture, in turn, bears many similarities with SOAs in terms of their modular construction and hence formal definitions for these two terms are also needed in order to promote a common understanding among various stakeholders in this technology space such as system architects, integrators etc.

Table of Contents

List of Appendices

List of Figures

List of Tables

1 Introduction

A trend since the early 2000's in data centers used for in-house enterprise applications and cloud computing services is the increasing adoption of Hardware or Server Virtualization. Hardware virtualization enables running multiple computing stacks called System Virtual Machines (S-VMs) on a single physical host. A S-VM in the context of hardware virtualization is made up of a complete computing stack (or engine) consisting of one or more applications, Operating System (called the Guest OS) and virtual hardware. S-VMs are able to perform their tasks due to an intervening hardware emulation layer or hypervisor that runs between the S-VMs and the hardware of the physical host.

Another trend is to virtualize applications at the OS layer. Just like multiple S-VMs run on the same physical hardware, in this context, multiple instances of an entity called "Application Containers" run on top of an OS's kernel in user space. Just like hardware virtualization allows multiple OS instances to run on a single physical host, application container technology allows multiple isolated user space instances (processes) to be run on a single host. Application containers are made of up application code (e.g., webserver or DBMS server) which has access to a collection of libraries/binaries that represent an OS's core capabilities. Each library component provides a traditional OS function such as memory, namespace and processes needed for that application code to work. The application container, when deployed, provides an execution environment for applications in the form of isolated processes.

Application components that are placed into a container leverage a Microservices architecture. A Microservices architecture can be contrasted with a Service-oriented architecture (SOA) wherein Microservices consist of small, stateless, loosely coupled and isolated processes built around capabilities as opposed to services. Microservices are independently deployable in Application Containers, use less resources and can be created, destroyed, started and stopped far faster than in a SOA.

Based on the discussion above, it should be clear that we need a formal definition of the building blocks of these emerging technologies such as Application Containers & Microservices architecture as well as their closely related counterparts – S-VMs & SOA along with an explanation of similarities and differences. The objective of this document is to provide those definitions, similarities and differences so as to create a common understanding of the semantics of these terms.

2 Background: Service-Oriented Architecture

Assembling an enterprise-scale solutions or individual system from distributed services is a well-established architectural approach referred to as service-oriented architecture (SOA) [2]. A SOA is an architectural pattern for integrating business processes and supporting IT infrastructure wherein application components are decomposed into self-contained services that communicate with each other using a communications protocol and a set of well-defined Application Programming Interfaces (APIs), independent of any vendor, product or technology.

SOA allows services to be reused and combined to address changing business priorities.

3 Definition of Microservices

Microservices: *A microservice is a basic element that results from the architectural decomposition of an application's components into loosely coupled patterns consisting of self-contained services that communicate with each other using a standard communications protocol and a set of well-defined APIs, independent of any vendor, product or technology.*

Microservices are built around capabilities as opposed to services, builds on SOA and is implemented using Agile techniques. Microservices are typically deployed inside Application Containers.

4 Similarities and Differences between SOA and Microservices

SOA and Microservices share several similarities and differences that are outlined below.

Table 1 – Comparison of Services Oriented Architecture and Microservices

Services Oriented Architecture	Microservices
Self-contained, monolithic services	Small, decomposed, isolated and independently deployable services
Communications between services occur through an enterprise service bus	Communications between services occur through lightweight, standard communications protocols and interfaces
Stateful and requires mapping of service dependencies when changes are introduced	Stateless and less fragile when changes are introduced
Longer start/stop times	Quick start/stop times
Built around services	Built around capabilities

5 Definition of Application Containers

Application Containers: *An Application Container is a construct designed to package and run an application or its' components running on a shared Operating System.*

Application Containers are isolated from other Application Containers and share the resources of the underlying Operating System, allowing for efficient restart, scale-up or scale-out of applications across clouds. Application Containers typically contain Microservices.

6 Definition of System Virtual Machines (S-VM)

System Virtual Machines: *A System Virtual Machine (S-VM) is a software implementation of a complete system platform that supports the execution of a complete operating system and corresponding applications in a cloud.*

Each S-VM serves as an efficient, isolated duplicate of a real machine running on a cluster of physical machines.

7 Similarities and Differences between S-VMs and Application Containers

S-VMs abstract the Operating System from the underlying hardware, allowing for multiple Operating Systems and Application to share a single system's physical compute resources. Application Containers abstract the Application from the underlying Operating System, allowing for multiple Applications to share a single system's Operating System and underlying physical compute resources

The following figure depicts the difference between System Virtual Machines and Application Containers

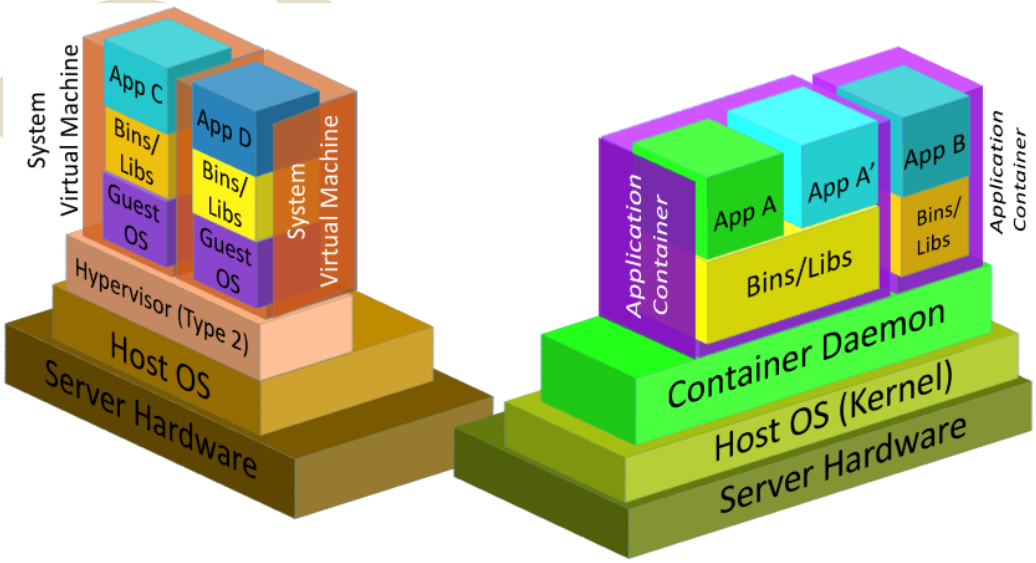

Figure 1 – Differences between S-VMs and Application Containers

249 ## Appendix A—Acronyms

250 Selected acronyms and abbreviations used in this paper are defined below.

API	Application Programming Interface
OS	Operating System
SOA	Service-Oriented Architecture
S-VM	System Virtual Machine

251

252

253 ## Appendix B—References

[1] Federal Information Security Management Act of 2002, Pub. L. 107-347 (Title III), 116 Stat 2946. http://www.gpo.gov/fdsys/pkg/PLAW-107publ347/pdf/PLAW-107publ347.pdf.

[2] *Executing SOA: A Practical Guide for the Service-Oriented Architect,* IBM Press, 2008, 240pp. https://books.google.com/books?id=VIrz5v4MMkgC

254