

Retired Draft

Warning Notice

The attached draft document has been RETIRED. NIST has discontinued additional development of this document, which is provided here in its entirety for historical purposes.

Retired Date January 21, 2025

Original Release Date February 18, 2016

Retired Document

Status Initial Public Draft (IPD)

Series/Number NIST Special Publication 800-180

Title NIST Definition of Microservices, Application Containers and System Virtual Machines

Publication Date February 2016

Additional Information See <https://csrc.nist.gov/projects/cloud-computing> for cloud computing-related resources from the NIST Cybersecurity and Privacy Program.

2

3

4 **NIST Definition of Microservices,**
5 **Application Containers and**
6 **System Virtual Machines**

7

8

9 Anil Karmel
10 Ramaswamy Chandramouli
11 Michaela Iorga

12

13

14

15

16

17

18

19 This publication is available free of charge

20

21

22

COMPUTER SECURITY

23

24

25

23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

46
47
48
49
50
51
52

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

53

Authority

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

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

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

70 This publication is available free of charge
71

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

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

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

85 **Comments on this publication may be submitted to:**

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

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

88 National Institute of Standards and Technology
89 Attn: Computer Security Division, Information Technology Laboratory
90 100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
91 Email: sec-cloudcomputing@nist.gov

92

93

Reports on Computer Systems Technology

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

104

Abstract

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

112

Keywords

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

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138

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.

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

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

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

157 **Table of Contents**

158 **Executive Summary v**

159 **1 Introduction 1**

160 **2 Background: Service-Oriented Architecture 2**

161 **3 Definition of Microservices 2**

162 **4 Similarities and Differences between SOA and Microservices 2**

163 **5 Definition of Application Containers 3**

164 **6 Definition of System Virtual Machines (S-VM)..... 3**

165 **7 Similarities and Differences between S-VMs and Application Containers 3**

166 **List of Appendices**

167

168 **Appendix A— Acronyms 4**

169 **Appendix B— References 5**

170 **List of Figures**

171

172 **Figure 1 – Differences between S-VMs and Application Containers 3**

173

174 **List of Tables**

175 **Table 1 – Comparison of Services Oriented Architecture and Microservices 2**

176

1 Introduction

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

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

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

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

208 **2 Background: Service-Oriented Architecture**

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

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

216 **3 Definition of Microservices**

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

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

224 **4 Similarities and Differences between SOA and Microservices**

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

226 **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

227 5 Definition of Application Containers

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

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

233 6 Definition of System Virtual Machines (S-VM)

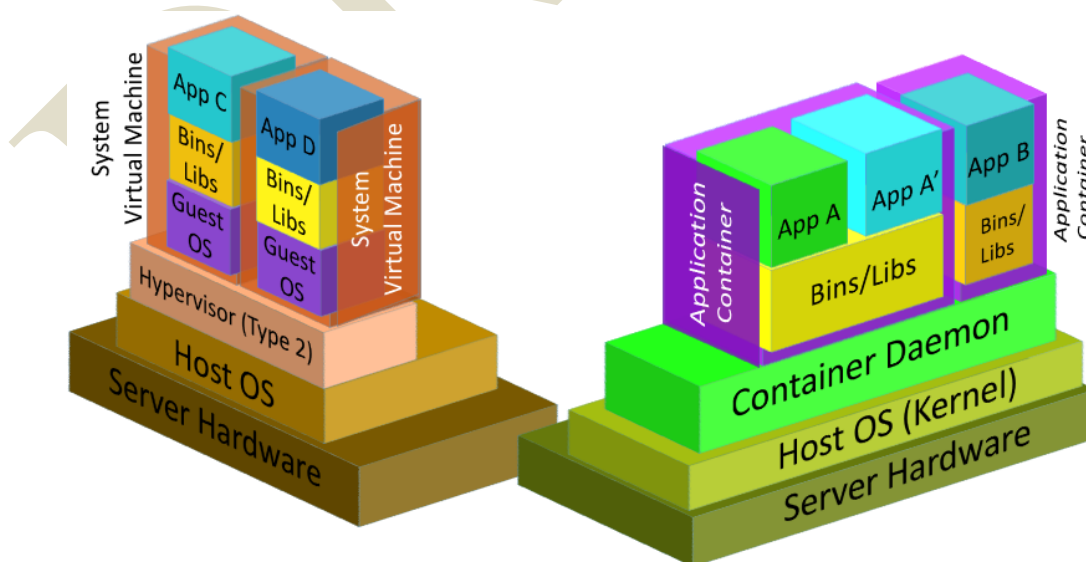
234 **System Virtual Machines:** *A System Virtual Machine (S-VM) is a software implementation of a*
235 *complete system platform that supports the execution of a complete operating system and*
236 *corresponding applications in a cloud.*

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

239 7 Similarities and Differences between S-VMs and Application Containers

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

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



247

248

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

DRAFT

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=Vlrz5v4MMkgC>

254

DRAFT