Containers:
The Future of Virtualization & SDDC

Anil Vasudeva
President & Chief Analyst
IMEX Research.com

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA and the author.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - SNIA and author must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

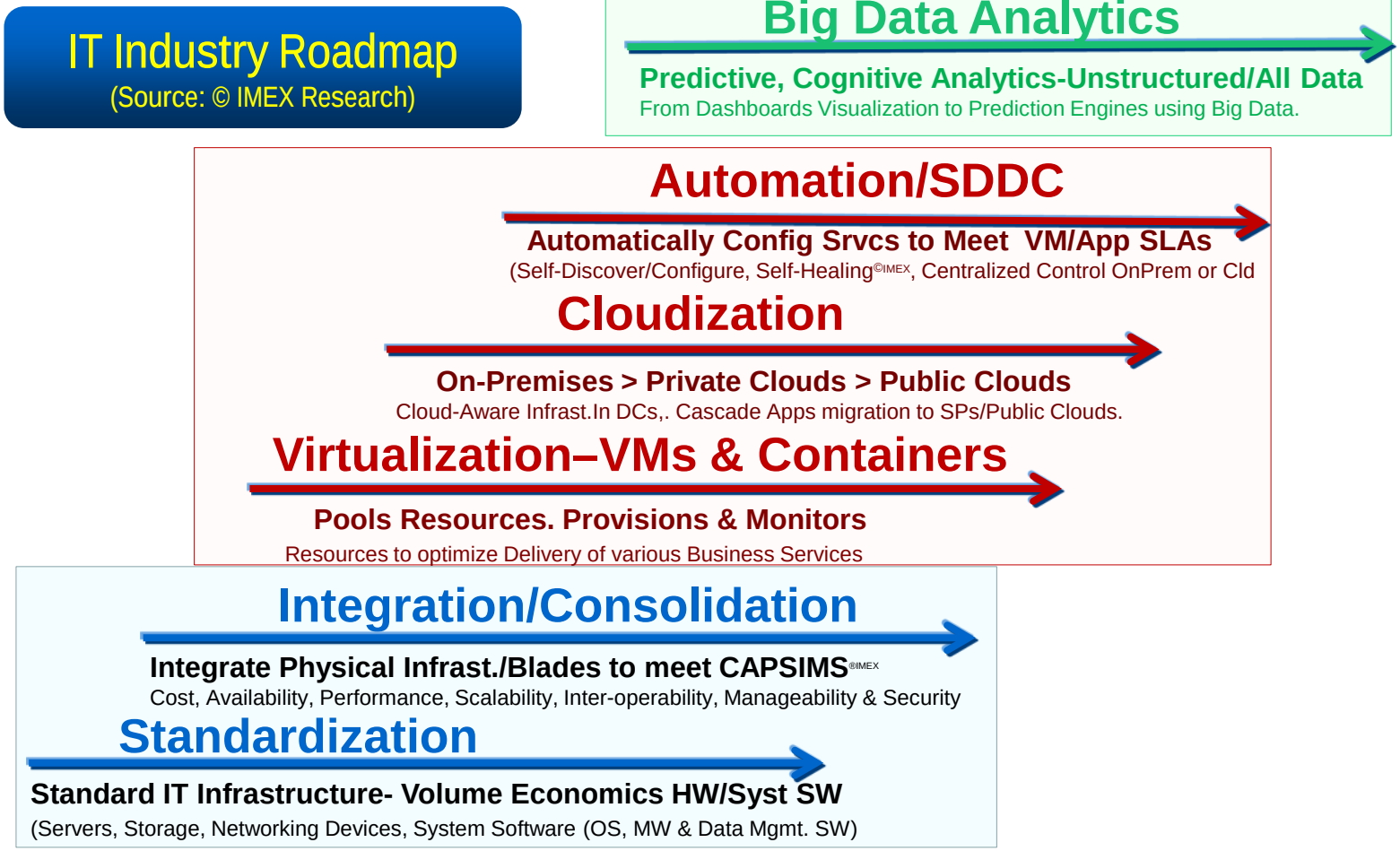
NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Abstract

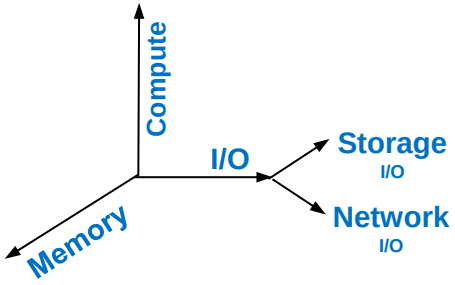
Containers: The Future of Virtualization & SDDC

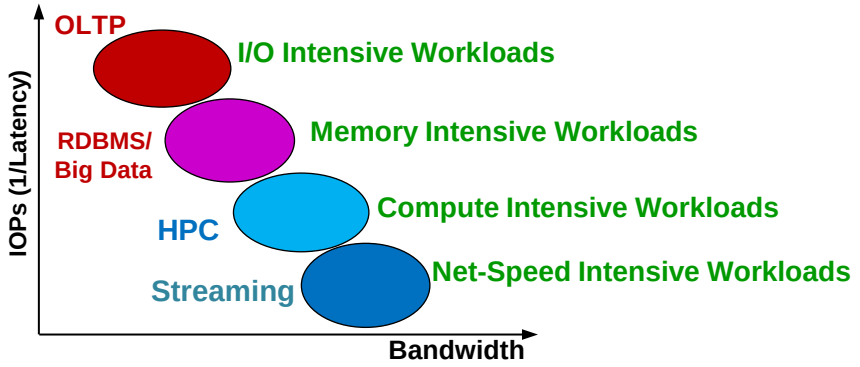
- This session will appeal to Data Center Managers, Development Managers, and those that are seeking a fundamental understanding of the future of virtualization and next generation software defined data centers both on-premise and in clouds. The session delves into the emergence of containers as the next generation of virtualization to run multiple applications without a hypervisor and as the light weight platform that will extend to software defined data centers (SDDC). Their rapid adoption by both startups and major IT vendors is a testament to their “Build and Configure once , run anywhere” portability between On-Premise DCs to Clouds., The session is targeted to bring a clear understanding of this key technology to developers as well as systems admins and cloud service providers

IT Industry Roadmap



Data Center IT Issues & Solutions

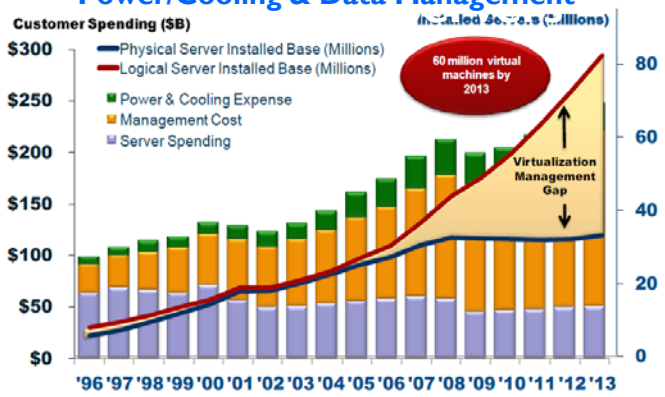




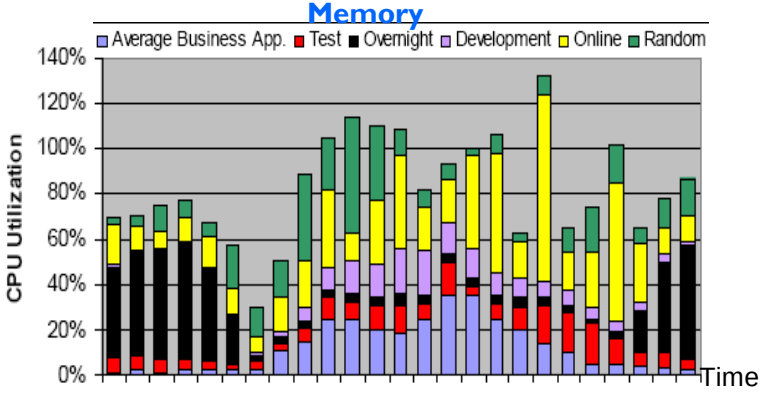
Issue	Industry Solution
• VMs/Hypervisor treated all apps as same ✓ Created Blender Effect ✓ Poor Performance of some VMs	• Workload I/O Optimized Infrastructure • Solid State/Flash for I/O intensive Apps (e.g. OLTP) • Application segregated by Frequency of use into Hot/Warm/Cold Storage
• Each Application is different and depending on its SLA/mission criticality needs in the Data Center or Cloud	• Must identify various Metrics to meet SLA needs such as Availability, Performance, Cost, Security etc. (e.g. Performance metrics - Latency vs Bandwidth)
• Storage must meet 2 major criteria of ✓ Data Protection ✓ Storage Efficiency	• Implement technologies ✓ RAID, Continuous Availability, ✓ Compression, Encryption,
• Real-Time IT/Databases needed for both ✓ On-line Transaction Processing to ✓ Real-Time Deep Queries ✓ Cloud Queries Exploding ✓ VMs heavy due having to load 10-20 MB of OS from storage slowing queries	• Implement New technologies ✓ RDBMS/Big Data/Virtualization VMs, ✓ Columnar Compression ✓ Use of Lightweight Containers with no Hypervisor or not having to load OS – witness Google containers using 7000 queries per second

Virtualization: Impact on IT Infrastructure

WW Spending on Power/Cooling & Data Management

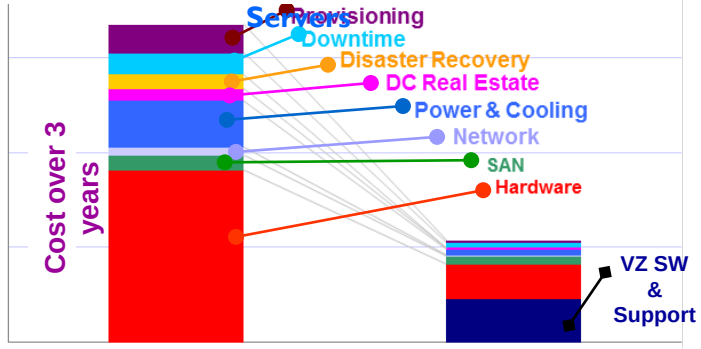


Multiple VMs create I/O Blender & Bottlenecks in Physical Servers sharing Memory



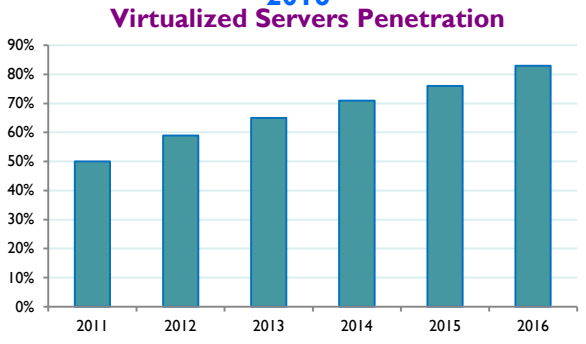
Virtualization: TCO

995 Pre-Virtualization (VZ) Servers → 78 VZ Servers



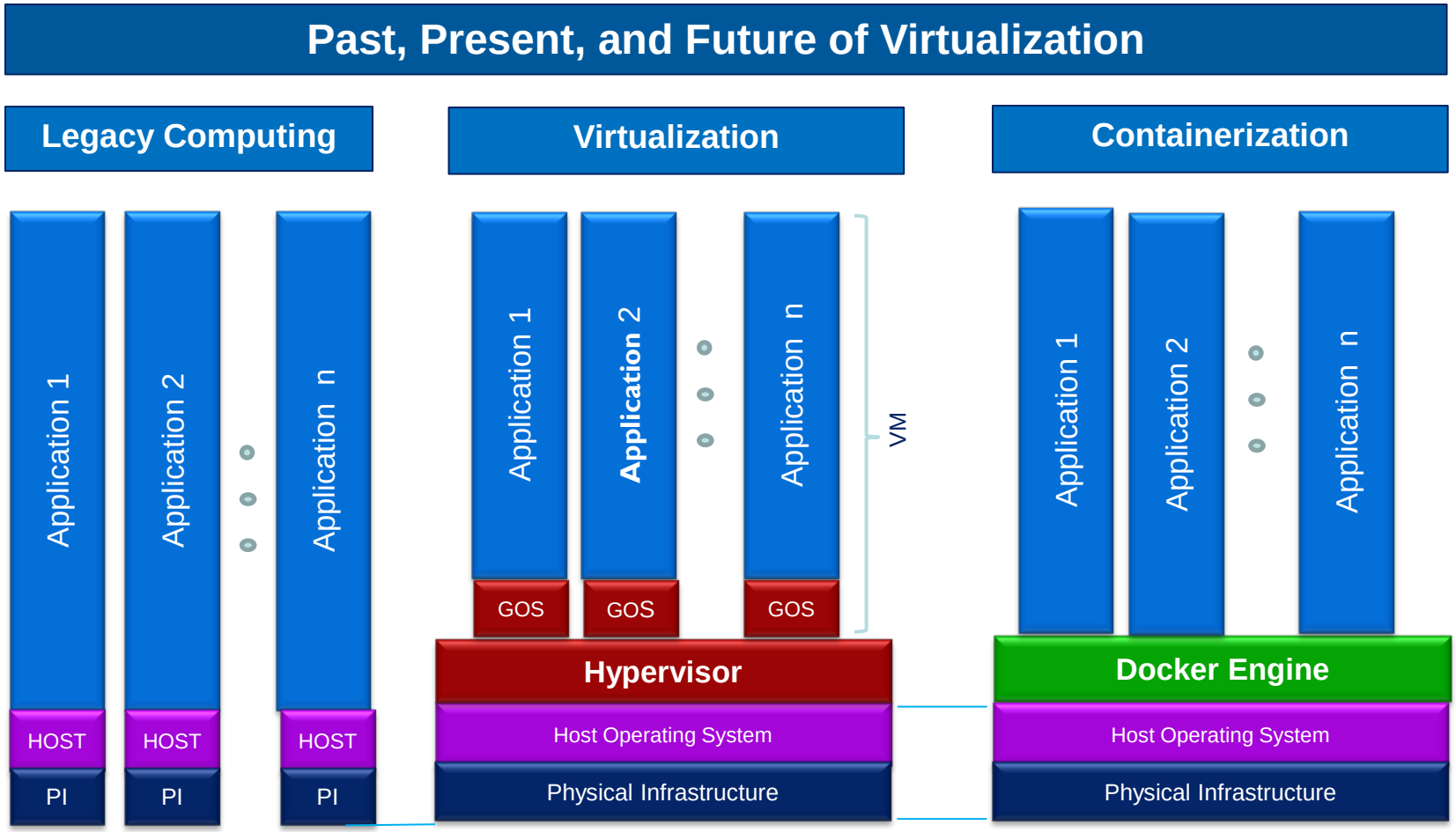
Virtualized Server Penetration

90% of all servers in 2011 growing to 65% by 2016



Virtualization led Workload Consolidation provided >20x Savings in RE, Pwr., HA

Past, Present & Future of Virtualization



Legend: DE=Docker Engine; G-OS=Guest Operating System; HOST=Host Operating System; PI=Physical Infrastructure

Containers: In Perspective

Container = Hottest Trend in Data Center Innovation
Container = Opening next wave of optimization of computing power usage post VMs
Container = Lightweight OS Level Virtualization
Container = Runs Multiple Apps without a Hypervisor
Container = Build and Configure Once , Run Anywhere
Container = Key to Google Search Engine speeds
Google - World's largest Linux Containers User
Google Search launches 7000 containers/sec
Search App in Container launches in msec

State of Containers Industry

Containers: An open source app portability platform that packages app and its dependent component parts into a hardware isolated container
Containers provide a more efficient use of Resources:: Eliminates Hypervisor reducing virtualization overhead while still allowing separation and isolation of multiple tasks on one host without relying on a separate OS, saving huge amount of HW resources compared to bulky VMs which emulate physical servers complete with OS
Allows Easy Application Portability between different hosts within DCs, DC to DC/ DC to Cloud (Private, Public, Hybrid) / Cloud to Cloud
Rapidly gaining adoption: Over15000 aps have been "containerized" and available on Docker Hub repositories, ready for server admins and cloud server providers to deploy both as free and paid options.
Significant buy-in from both start-ups and large vendors: :
Repositories – Ubuntu, Redis, MySQL, mongo DB etc.
OS Vendors - Red Hat, SUSE, CoreOS, Boot2Docker - Mac & Windows …
Clouds – IBM, Amazon, Microsoft, Rackspace, Google, Intel etc.
Major Users - Paypal, eBay, Google, Spotify etc..

Containers: Pros & Cons

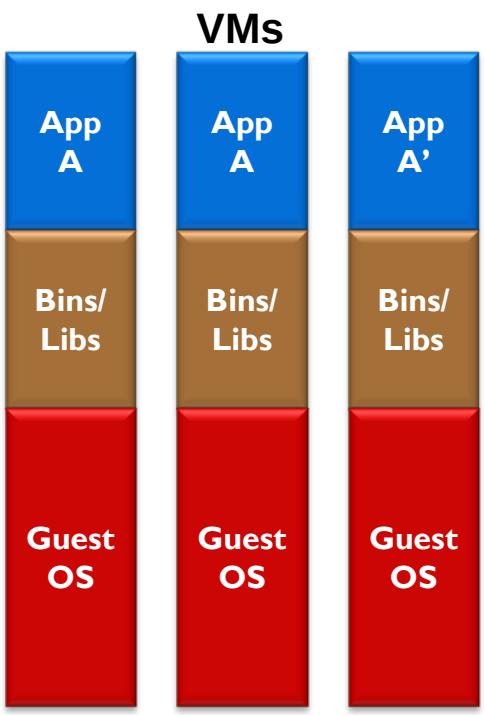
Pros

- **Containers pack more computing workloads onto a single server**
- **Launch capacity for new computing jobs in a split second vs minutes for VMs since no loading of OS from storage required**
- **Appealing to Virtualization Admins and Cloud Providers to maximize consolidation and improve resources efficiency in multi-tenant price-sensitive environments**
- **Developers able to rapidly move their applications from concept to production since no need to boot up an OS instance of Linux or Windows for every app run**
- **Efficiency driven from speed allows improvements in CapEx (Buy less HW) or OpEx (Build/Rent less DC space, Fewer people in operations/maintenance).**
- **Speed lets DCs respond quickly in case of a sudden spike in business activity (explosion in searches/queries or Black Friday promotion driven sales orders)**

Cons

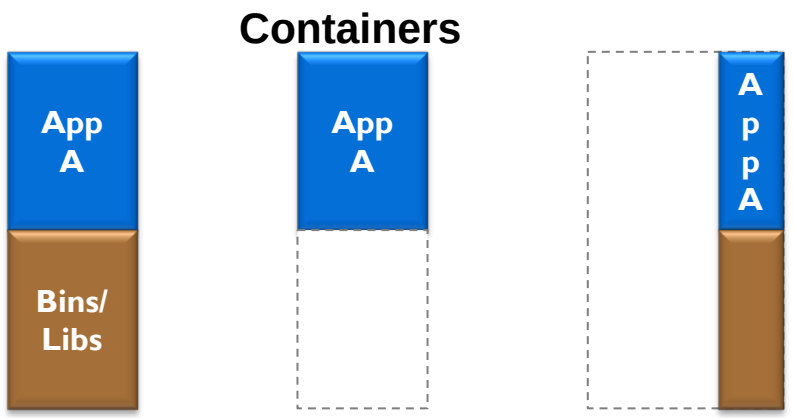
- Shared back-end Host OS failures can create system-wide outages
- Can't provide a virtual instance of windows on a Linux Server
- Uneasy co-existence with VMs in the long haul
- Much as Containers give each app running on a server its own isolated environment to run, but those containers all share host servers' OS With severe dependencies.

VMs vs. Containers



VMs

Every app, every copy of an app, and every slight modification of the app requires a new virtual server.



Original App

No OS to take up space, resources, or require restart

Copy of App

No OS, can share bins/libs

Modified App

Copy on write capabilities allow us to only save the diffs between Container A and Container A'

VMs vs Containers Characteristics 1

Metric	VM	Containers
Infrastructure	• 100s of VMs can be put on a single host server each running an application w its own Guest OS • Strong Management systems available to deal with 100s of VMs per server	• Lightweight, require Less Memory space • Fast Launch time • 1000s of containers can be loaded onto a Host. Containerized apps share Host OS's kernel to execute work • Containers can become ultimate form of compute intensive low power, dense computing environment
Performance	• Slower than Containers since need to retrieve 10-20 GB of OS from storage per VM launch	• Fast creation without shackle of retrieving OS • Workload in Containers use Host OS kernel • Container can boot up in 0.5 seconds • Activates code instantly to test or launch add'l ecommerce capacity
Interoperability	• Systems designs fully compatibility with existing operations of enterprise DCs	• Apps are formatted in a standard way to be placed in a container. • Once in a container, each type of app moves around the network in same way
Maturity	• Robust, Highly Developed • Proven, even in mission critical workloads	• Less Mature • Kinks not worked out yet • Cos. Still working on a creative Management System

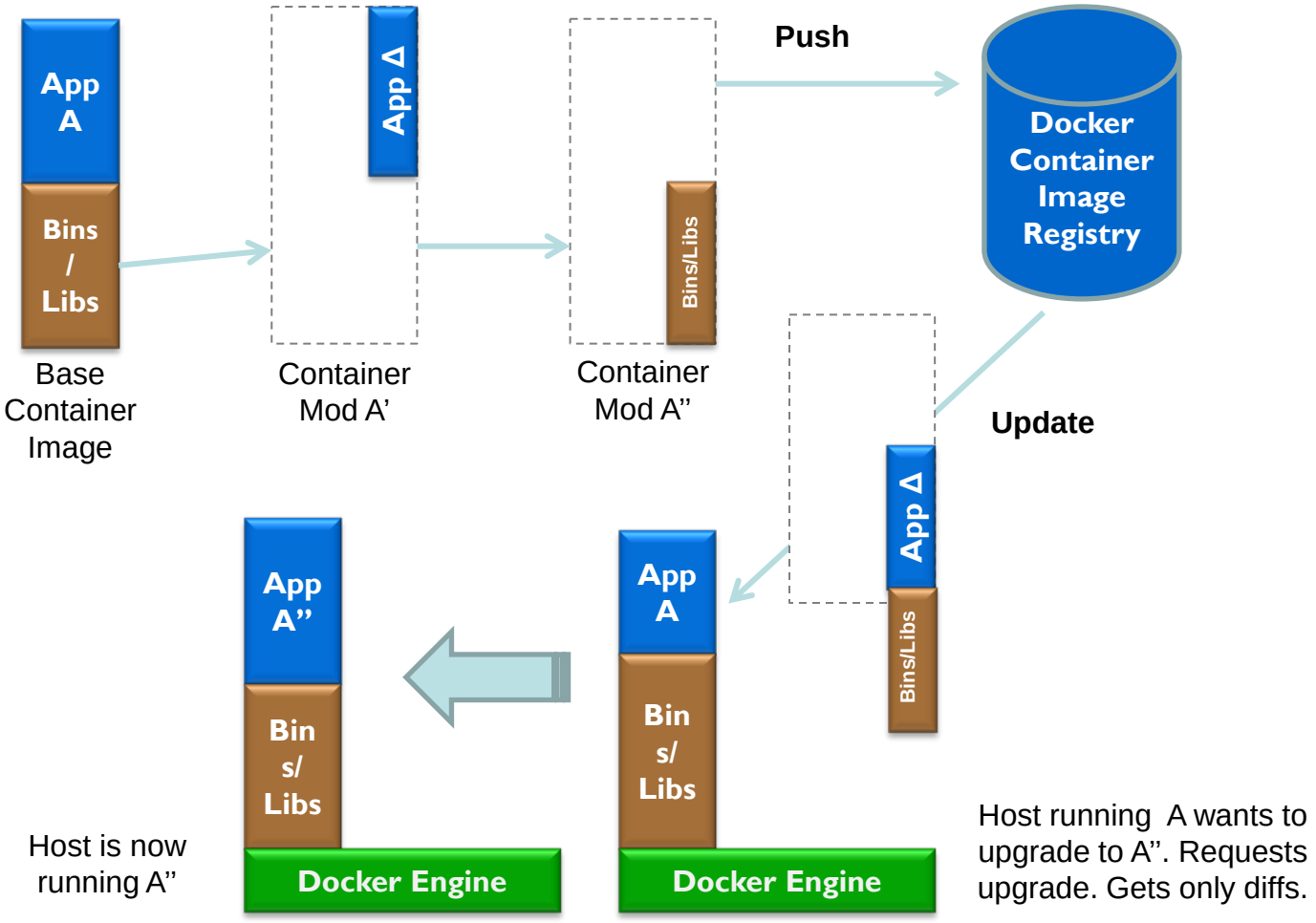
VMs vs Containers Characteristics 2

Metric	VM	Containers
Security	• Security of a dedicated OS • Harder Logical Boundaries • Hypervisor controlled dedicated pathway between App/Guest OS and Physical Infrastructure	• High security concerns • Containers share CPU, Memory, Disk in close proximity to each other • Containers talk to each other using shared memory - one container can steal neighbor's data or spread its malicious code to neighbors
Portability	• Highly portable between systems running same hypervisor (such as ESX, Hyper-V, Xen or KVM)	• Highly portable but bound to a certain version of an OS. • An app inside a container can't move to another host OS, only where there is copy of same OS w same version level
Multi-tenancy	• VM's Guest OS can be different than physical host's OS allowing different types of Apps (Technical on Mac vs Marketing on Windows) to run on same Physical Host	• Higher Level isolation with many apps running under the host OS and all containers sharing certain OS's kernel and certain Libs • Proven barriers to keep containers colliding each other • Once in a container, each type of app moves around the network in the same way

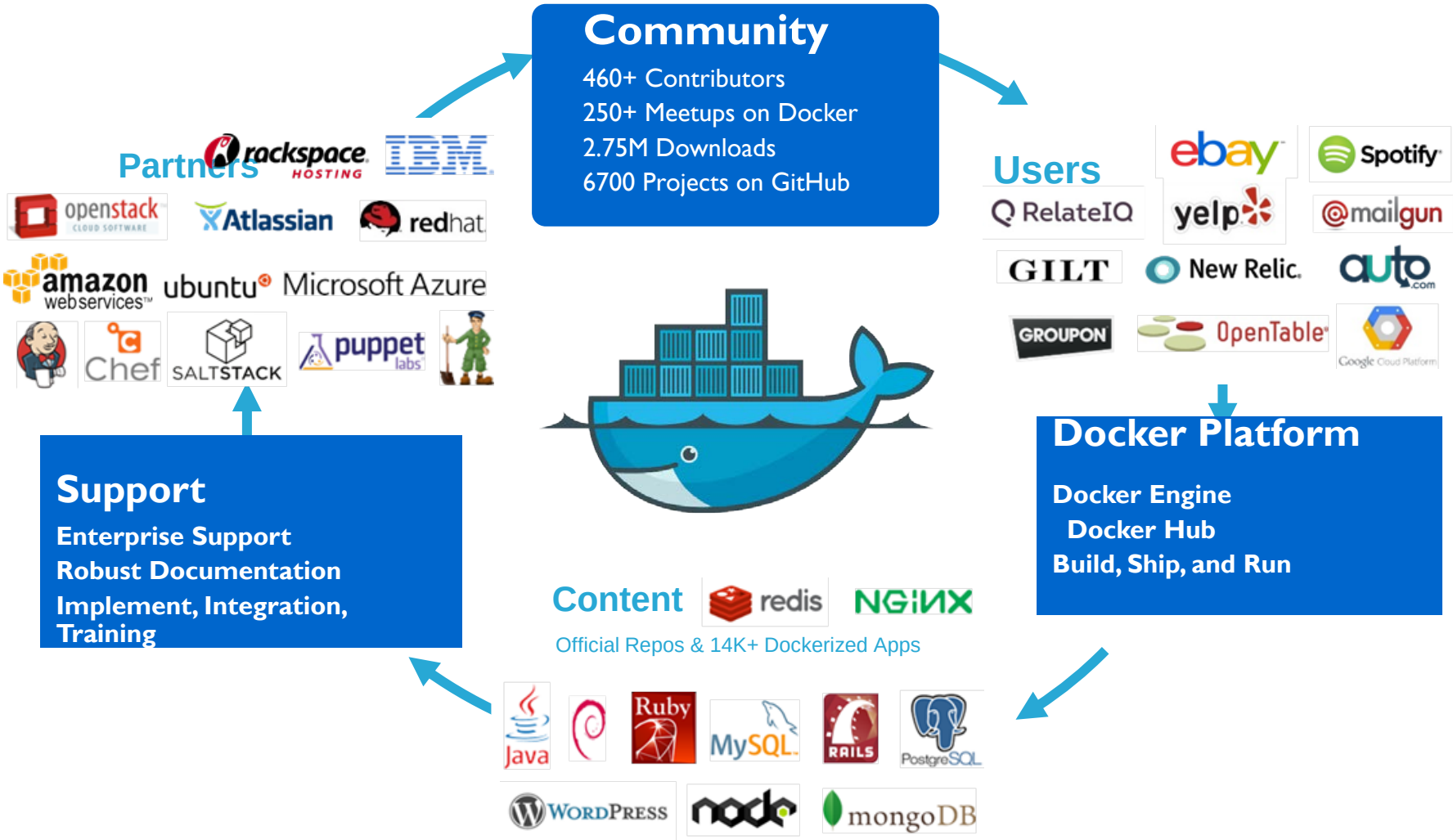
VMs vs Containers Characteristics 3

Metric	VM	Containers
Architecture	• VMs package files that can move from one host to another over internal networks or internet	• Built in layers and accessed independently.- a code change affecting one layer can be executed without touching other layers, making changes less dangerous vs. monolithic architecture in which an error can stall whole application • Big apps can be broken into smaller ones running in their own containers, maintained & upgraded independently • Google/Kubernetes for Docker Manages multi-containers adoptable by enterprises running clusters of containers for large apps across multiple computers.
Adoption & Futures	• Highly Developed and adopted in 50-60% of all large IT shops • Robust, proven, even in mission critical workloads • Multiple offerings by several vendors • Being adopted as the foundation in next generation DCs and Clouds via Software Defined Data Centers SDDC – SDC/SDS/ SDN	• Google's efforts to get container in Linux kernel & Docker making its creation & movements in open source helped developers rally around them as a standard, avoiding proliferations • Docker formatting engine now a standard with lots of tools/ workflows • By YE2014 -100 million downloads, 730 contributors • Support from Linux startups and large companies (IBM, MS, RH, Docker, Google etc.) • IBM Docker Hub Enterprise with Middleware allowing developers install preformatted dockerized containers in their won DCs and invoke the services through a link. • Stack multiple micro-services on a host using containers

Containers Lightweight Operation



Docker Ecosystem



Docker: Open Source Platform

SNIA Global Education

Any App

+ 14K apps
+ 6K projects

API

Docker Engine

open source software at the heart of the Docker platform

Docker Hub Enterprise

cloud-based platform services for distributed applications

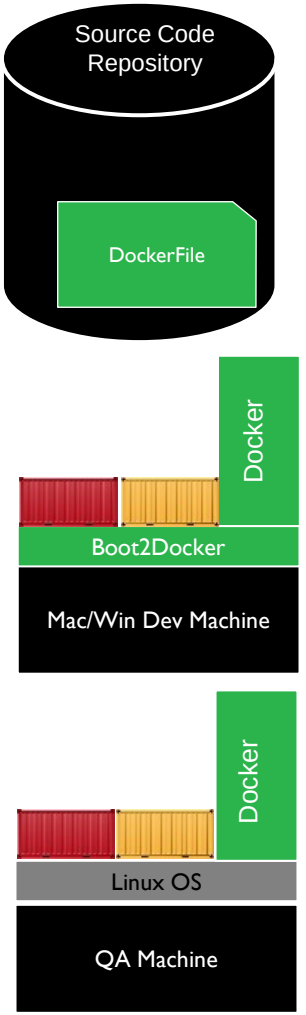
API

Any infrastructure

- Physical
- Virtual cloud

Docker: Engine

- Stable core
- All major Linux platforms and distributions
- Execution engine plug-ins: LXC, libcontainer
- Filesystem plug-ins: AUFS, BTRFS, device mapper
- Host networking, link hostnames
- boot2docker: Mac OS X and Windows
- Support for SELinux and AppArmor
- TLS auth, systemd slices, release hashes



Distributed Applications

Contributor's Laptop

Development VM

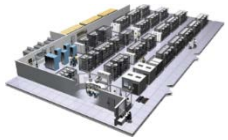
Production Servers

QA Server

Disaster Recovery

Production Cluster

Customer Data Center

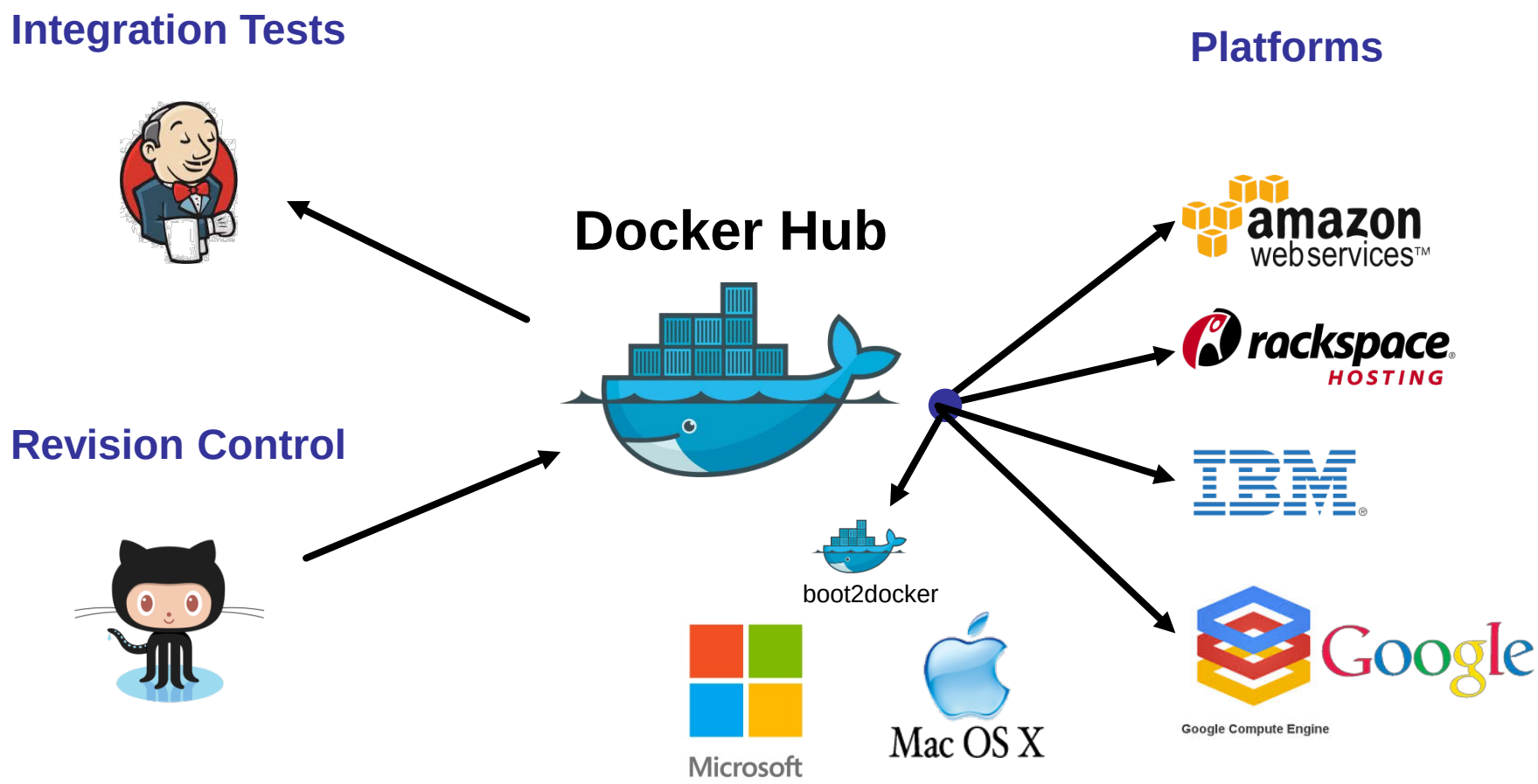


Public Cloud

Official Repositories

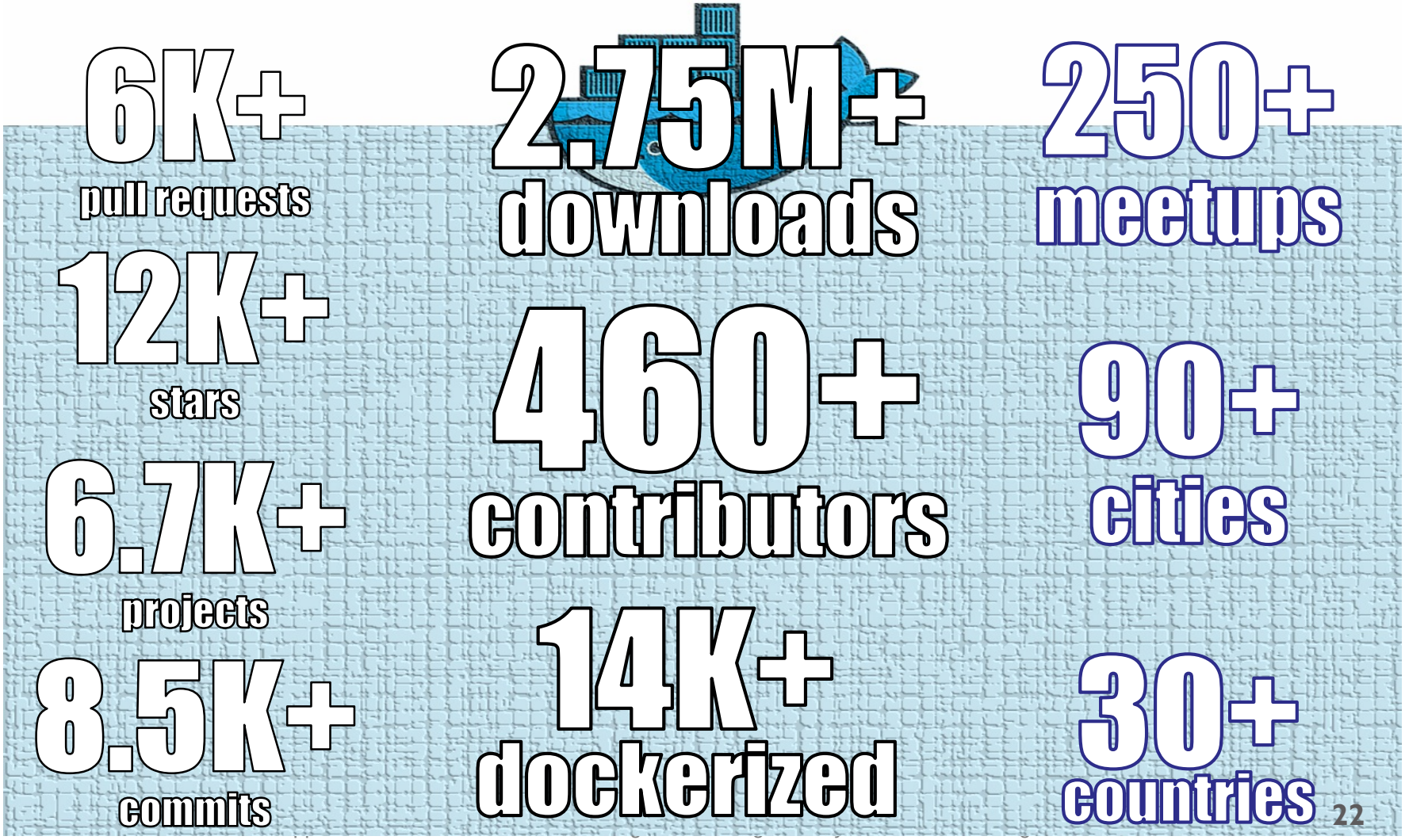
- DBs, Web servers, OSes and more
- Free on Docker Hub Registry
- Optimized and tuned for Docker
- Maintained and supported
- Publisher program

Container Deployment Infrastructure



Open Source Community

SNIA Global Education



Containers – In Perspective

- Containers extending virtualization to next generation computing bringing in speed with thousands of containers launch able within seconds on same hardware lowering CapEx and OpEx for enterprises and cloud service providers.
- Several start-ups and all major IT vendors endorsing it.
- Allows developers to quickly create, ready-to-run ”containerized” applications and enterprises to offer preconfigured, ready to run infrastructure apps to VARs and System Integrators to add industry vertical apps
- Makes managing and deploying applications much easier

Attribution & Feedback

The SNIA Education Committee thanks the following Individuals for their contributions to this Tutorial.

Authorship History

Original Author :
Anil Vasudeva, IMEX Research.com

Additional Contributors

Anil Vasudeva
M.R.Pamidi Ph.D.
Joseph L. White

Please send any questions or comments regarding this SNIA Tutorial to ***tracktutorials@snia.org***