IBM Power Systems

Container vs Virtualization

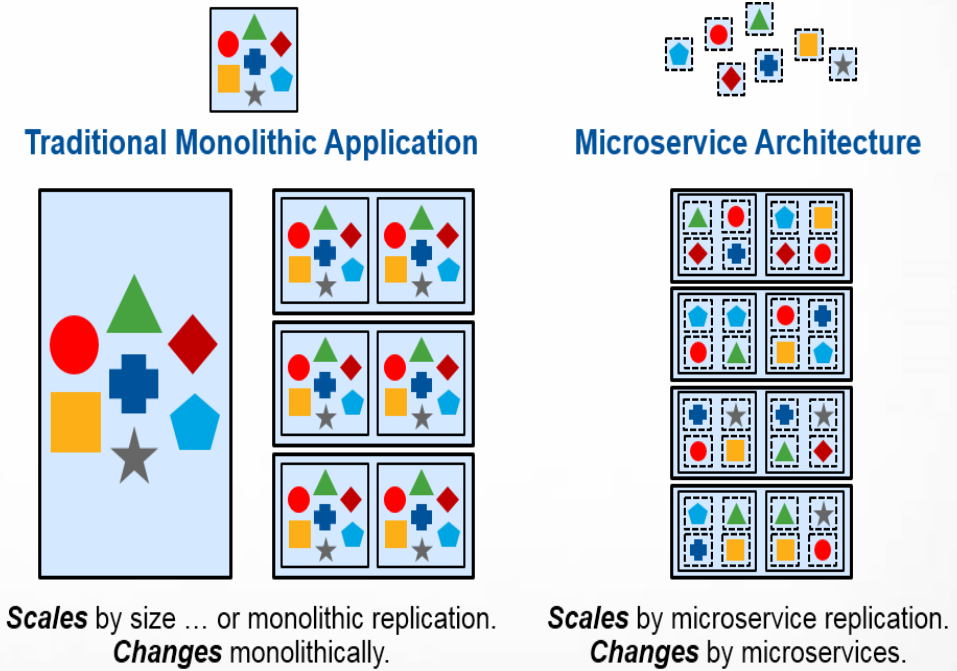
Sébastien Chabrolles
IT specialist Linux on POWER
IBM Client Center - Montpellier
(s.chabrolles@fr.ibm.com)

Journée Common Romandie
Genève – Nov 2016

Cloud Native Applications are being built with Microservices Using Containers

Cloud Native applications are characterized by the following:

- DevOps and continuous integration / continuous delivery methods
- User experience centric principles
- Use of microservice architecture, and API-centric approach to applications
- Multisource data collection mechanisms from sources such as social media streams or IOT sensors.
- Built-in analytics capabilities intended to help inform/develop process
- Low-risk technology investments that avoid lock-in



Source: Gartner

IT Must Evolve to Stay Ahead of Demands

Development Process	Application Architecture	Deployment & Packaging	Application Infrastructure
Waterfall	Monolithic	Physical Servers	Datacenter
Agile	N-Tier	Virtual Servers	Hosted
DevOps	**Microservices**	**Containers**	**Cloud**

Power is Cloud ready

Orchestration & Management

oVirt

kubernetes

vmware

openstack CLOUD SOFTWARE

Apache MESOS

juju

Consolidation technologies

KVM

LXC

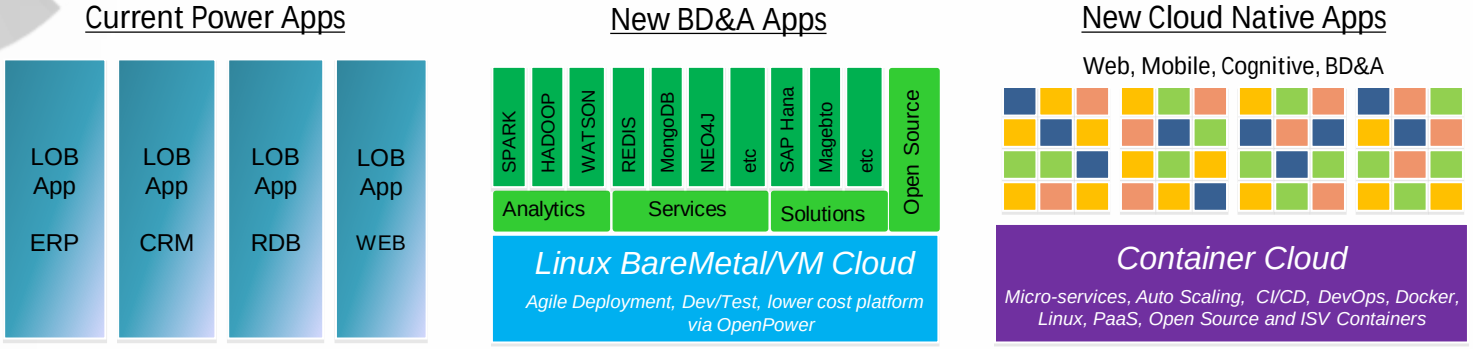
LXD machine containers from Ubuntu

IBM PowerVM

docker

… and many others are supported on Power Systems

Power Systems are a Great Platform for Cloud Native Solutions

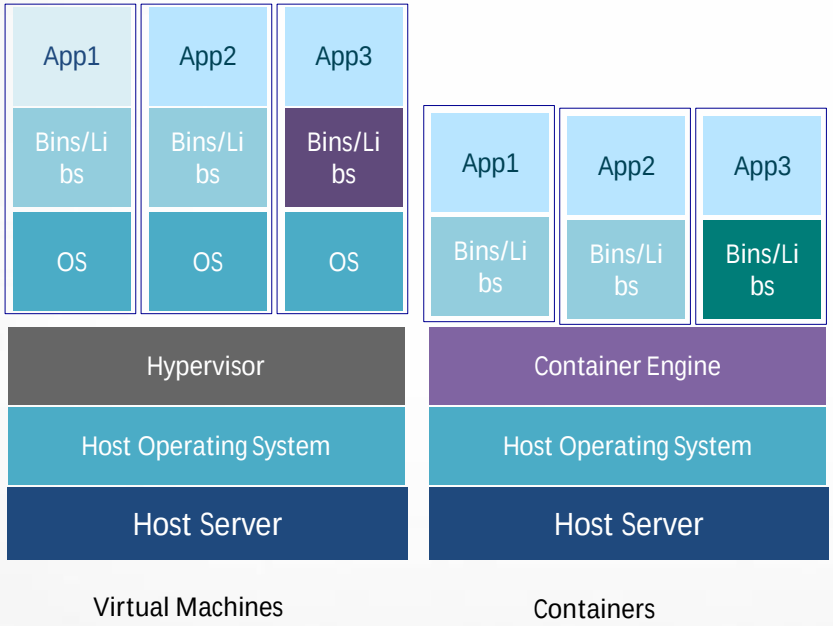


- Extending the value of current Power Solutions by leveraging Hybrid Cloud Technologies
- Connecting Cloud Style Apps to existing business Workloads to support Mobile and Cloud Consumers
- Exploiting the value of Business Data by adding new BD&A Capabilities

- All Major Open Source BD&A on Power
- LE Linunx (RHEL, Ubuntu) enables quick access to new BD&A technologies
- Thousands of packages available
- Accelerators (FPGA & GPU) via OpenPOWER can dramatically speed performance

- Full Docker Ecosystem available on Power
- RedHat and Ubuntu LE speed porting to Power
- Node.js, Java, Go, PHP Optimized on Power
- Thousands of packages are available to power as docker files being created and upstreamed
- Strong Customer Interest and desire to move into production with container clouds

Containers and VMs Solve Different Problems



Containers is more light and better performance:

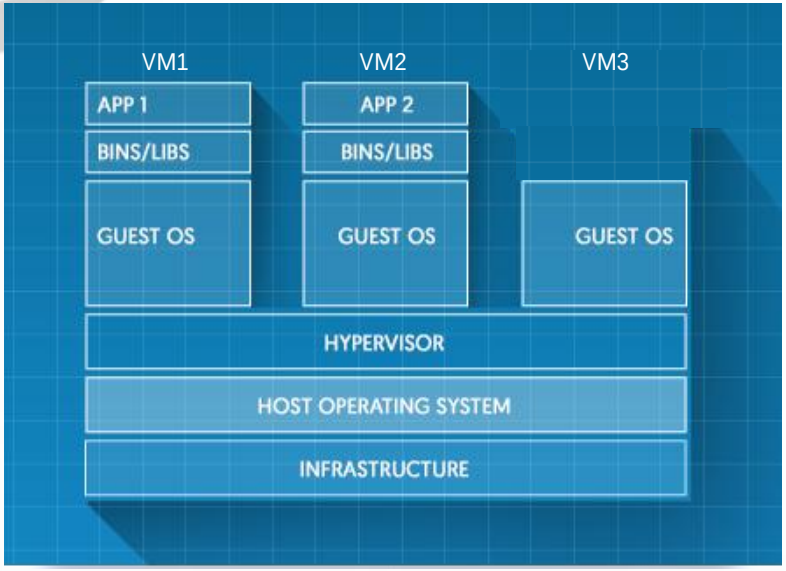
- Portability: VM (Gigabytes) vs. Container (Megabytes), VMs are constraint to Hypervisor and hardware-emulation
- Performance: Containers can boot and restart in seconds, compared to minutes for virtual machines. And no extra overhead of a hypervisor and guest OS makes containers consume less CPU and memory.
- Management cost: Each VM requires a full functional operating system, and then extra management for them.

Great advantage to use containers in:

- DevOps
- Batch computing
- Lightweight PaaS
- Microservices

Containers vs VMs

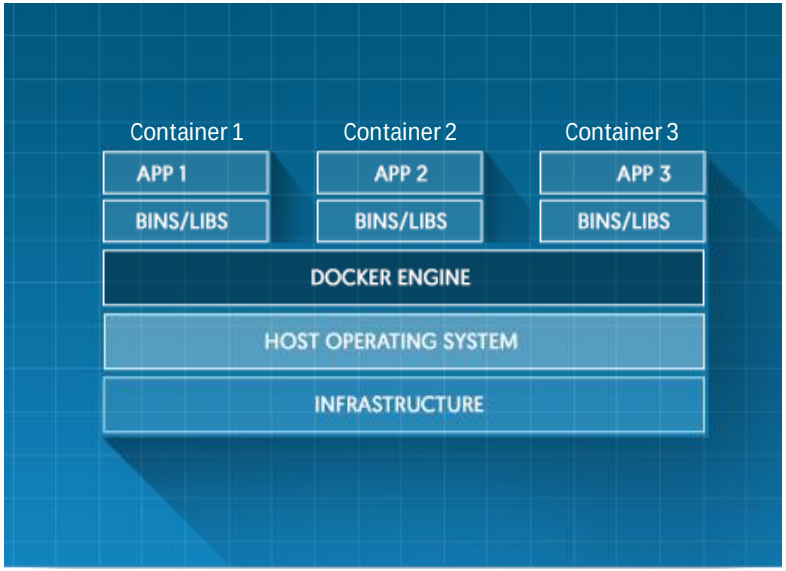
Virtual Machines



Virtualization Pro:

- Better Security / Isolation
- Allow different Kernel between VMs
- Not Limited to Linux OS

Containers



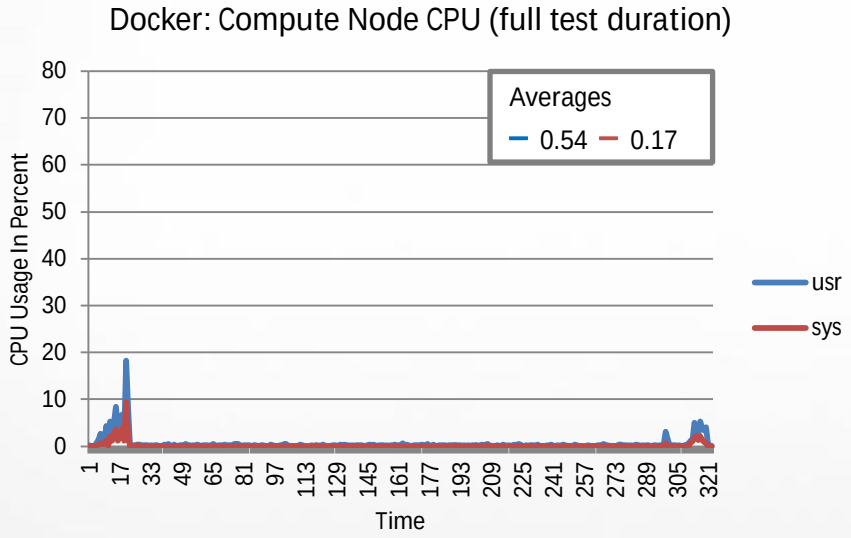
Containers Pro:

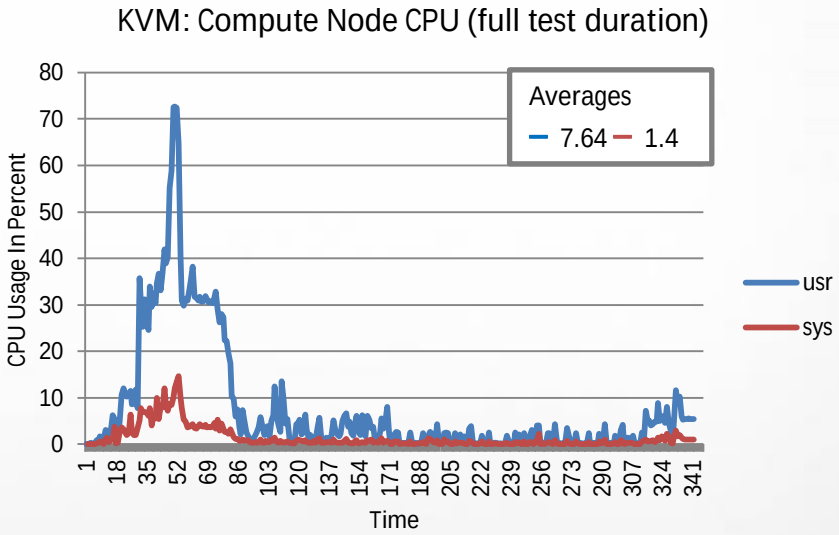
- Better resources utilization
- Less overhead compare to VM
- Light compare to VM
- Very FAST START : No Boot
- No special hypervizor mode access required
  => could be nested without performance impact.

Cloudy Performance

Starting 15 Container vs 15 VM (impact on system)

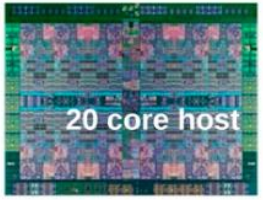
"Starting 15 VM on a single server consumes x3 more Memory and x10 CPU than 15 Container."





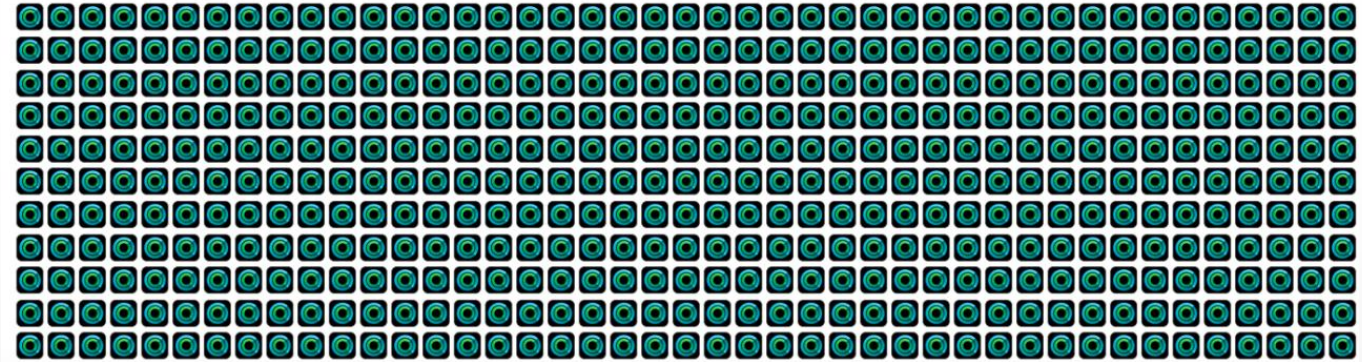
Docker Containers running on Power have Superior Density

- Wider, Faster Memory Interface, Faster Cores with More Threads
- Split-Core Mode supports Interactive Web style Apps better
- Greater Density of Containers per systems lowers Cost
- Better Throughput and Latency
- OpenPower ecosystem offers wide range of Open HW Platforms



Total: 10,011 Containers on One System:

Ubuntu (8028), Node.js (991), Wordpress (992)



Demo Shown at DockerCon Europe 2015

IBM POWER8 with Docker Delivers Superior Cloud Performance

- *Docker's* low overhead and quick deployments allow service providers to support many different types of users at once.
- IBM POWER8 with Docker delivers superior cloud economics through higher container density, better throughput and better transaction latencies.

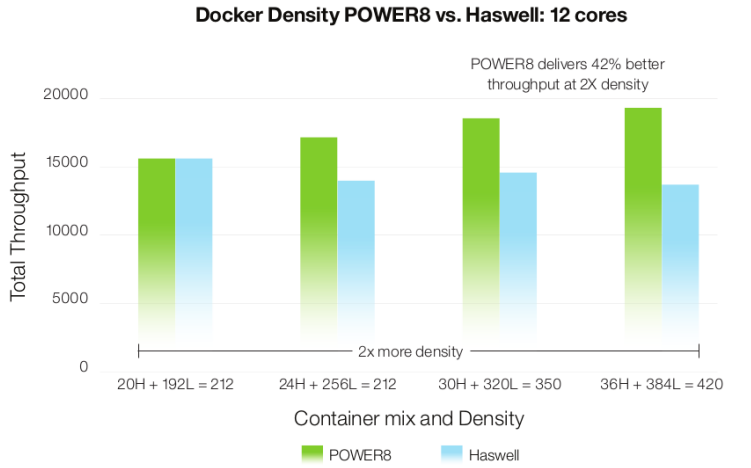
<u>Highlights:</u>

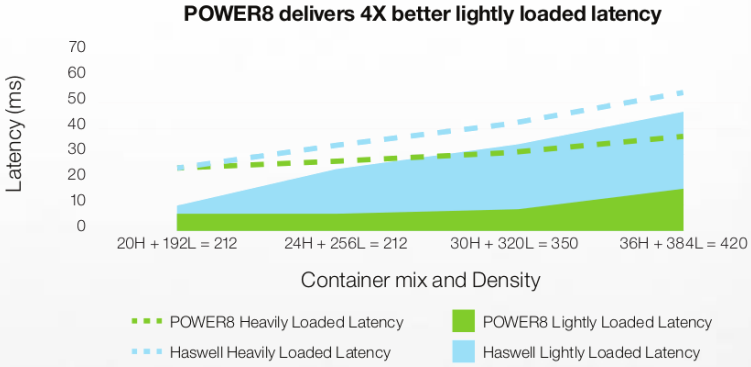
POWER8 with Docker delivers better performance than the compared Haswell, for 12 cores:

Up to 42% better throughput

Up to 4X better latency

2X container density





What is Docker?

"Docker is a software containerization platform"

Running on Linux **platforms…**

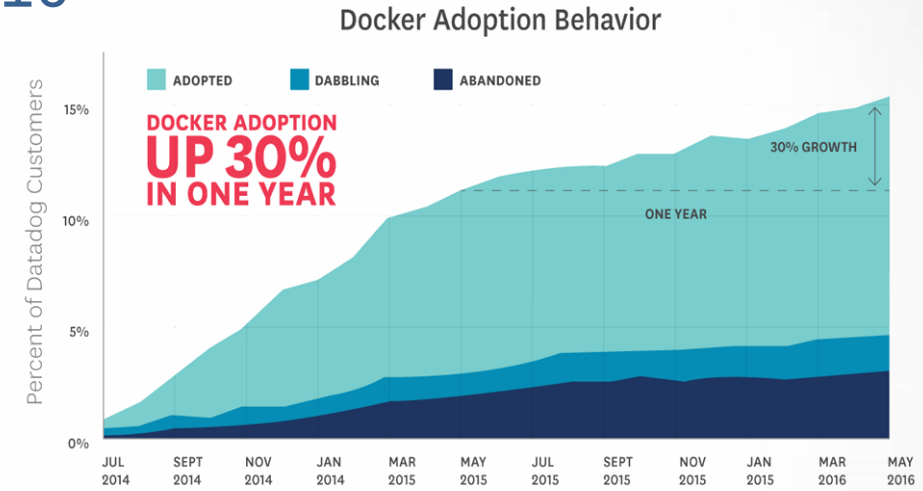
… **built** with Golang & Open Source!

Started in 2013 (3 years old)
But very popular ! => Quick & strong adoption

Docker Adoption Behavior in 2016

> ***"With*** thousands of companies using Datadog to track their infrastructure, we can see software trends emerging in real time***."***
>
> DATADOG

- 30% increase in Docker adoption in one year
- Docker runs on 10% of the hosts, up from 2% compared to 18 months ago
- Docker is mostly used by large companies with a large number of hosts
- The number of containers running in production quintuples (= 5x) 9 months after initial deployment



Source: Datadog

Source: Datadog

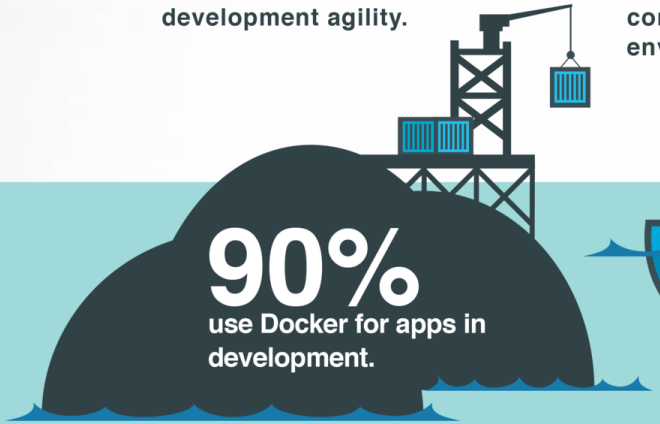
Source : https://www.datadoghq.com/docker-adoption/

Who are Docker adopter ?

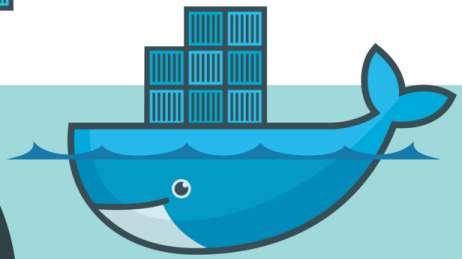
65%
use Docker to deliver development agility.

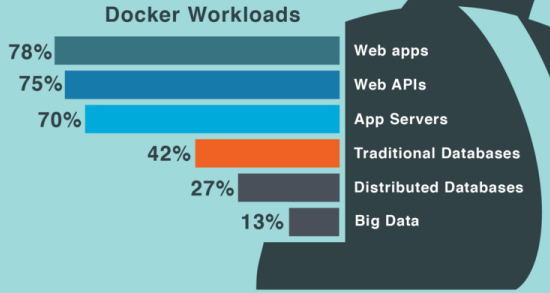
48%
use Docker to control app environments.

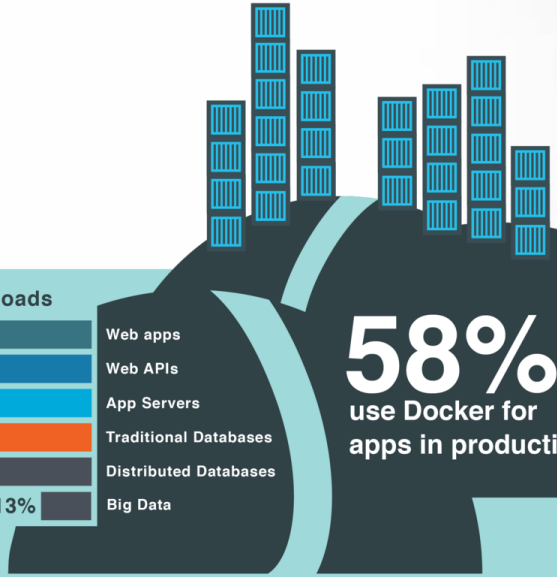
41%
use Docker to achieve app portability.



90%
use Docker for apps in development.







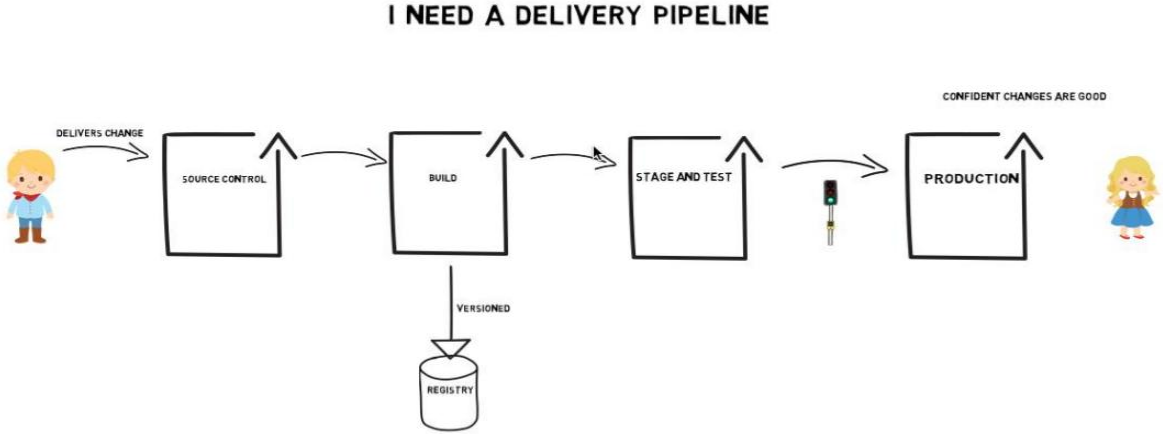
58%
use Docker for apps in production.

90%
plan dev environments around Docker.

80%
plan DevOps around Docker.

docker

Docker is not just another virtualization technology.



docker =

Build Ship Run

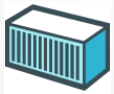
How it works – Docker Basics

Engine: Runs on Linux, it provides the operating environment for Docker containers.

Image: Read-only templates for containers, stored and managed in a registry. Once instantiated a container is created.

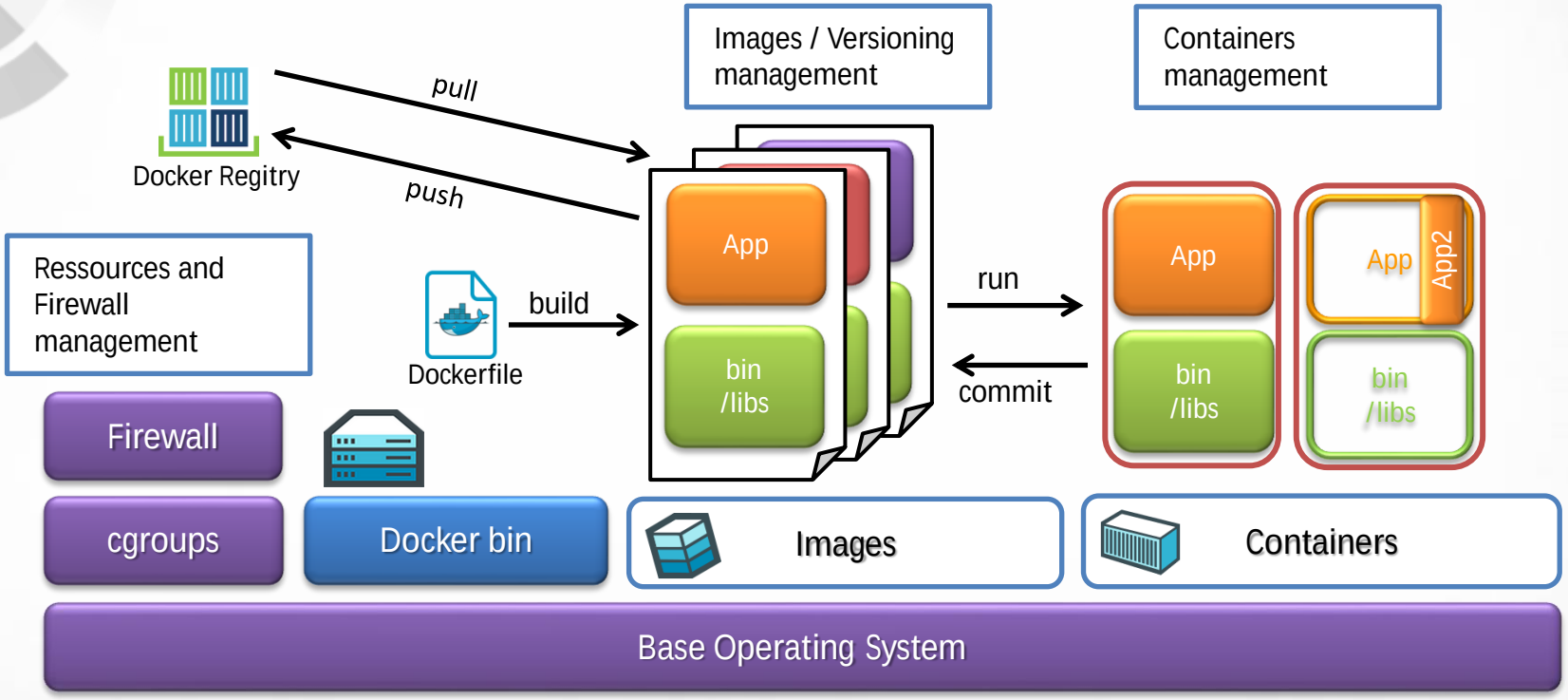
Dockerfile: Defines a Docker image as if it was code; used to re-build an image.

Registry: A service that allows to store and manage Docker images.



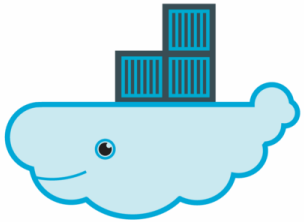
Container: Standard unit to package an application and its dependencies: binaries, libraries, system tools... So that it can be moved between environments and run without changes.

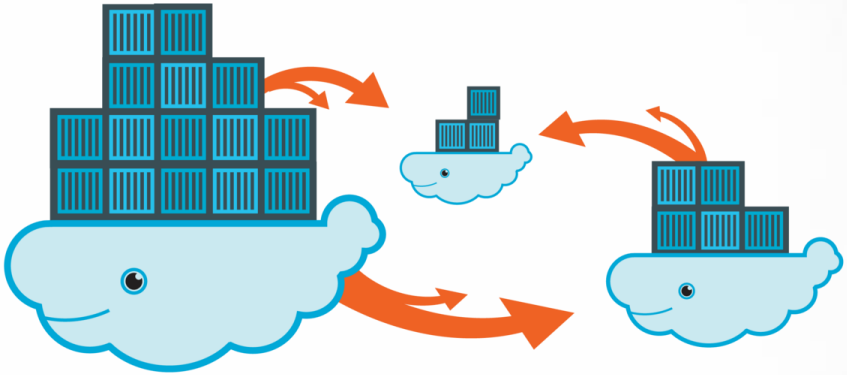
How it works – Docker Basics



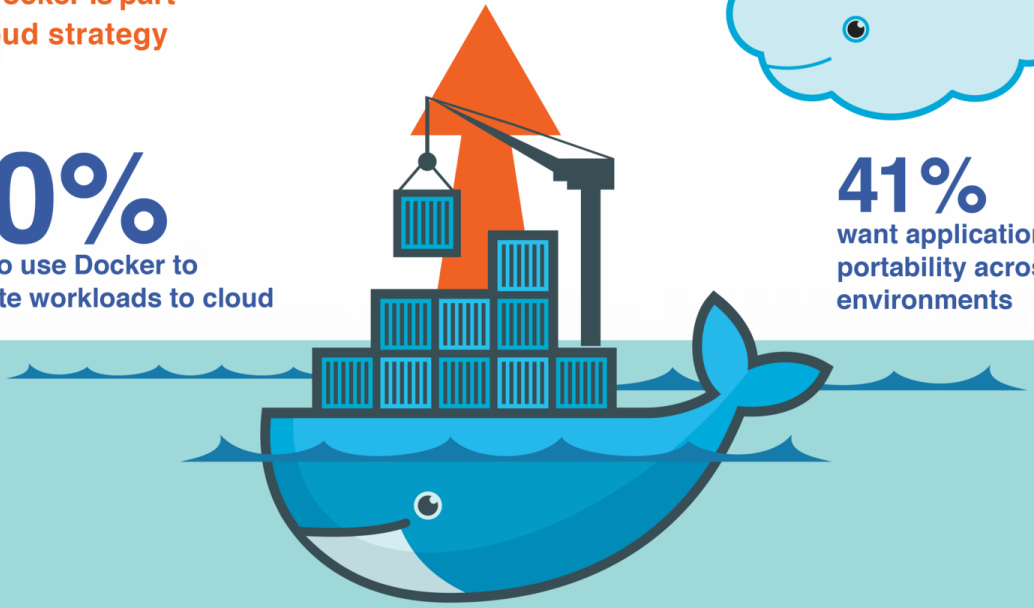
Docker adoption (Cloud)

80%
say Docker is part of cloud strategy





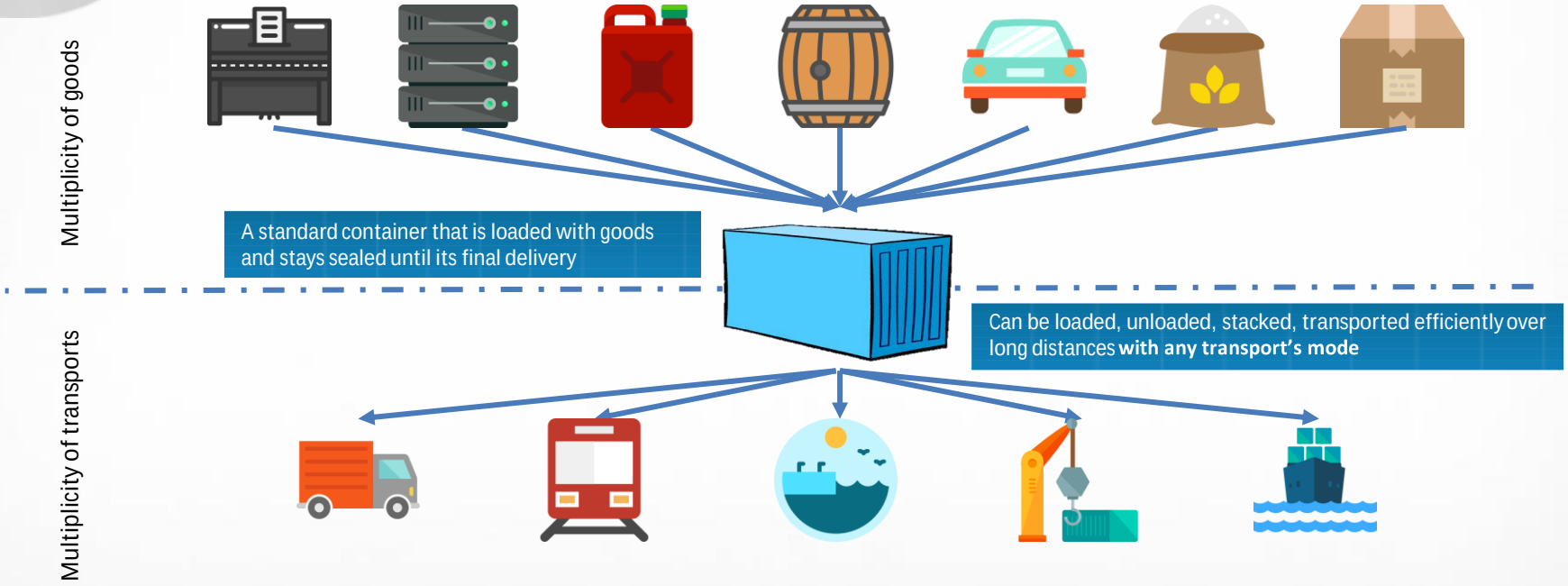
60%
plan to use Docker to migrate workloads to cloud



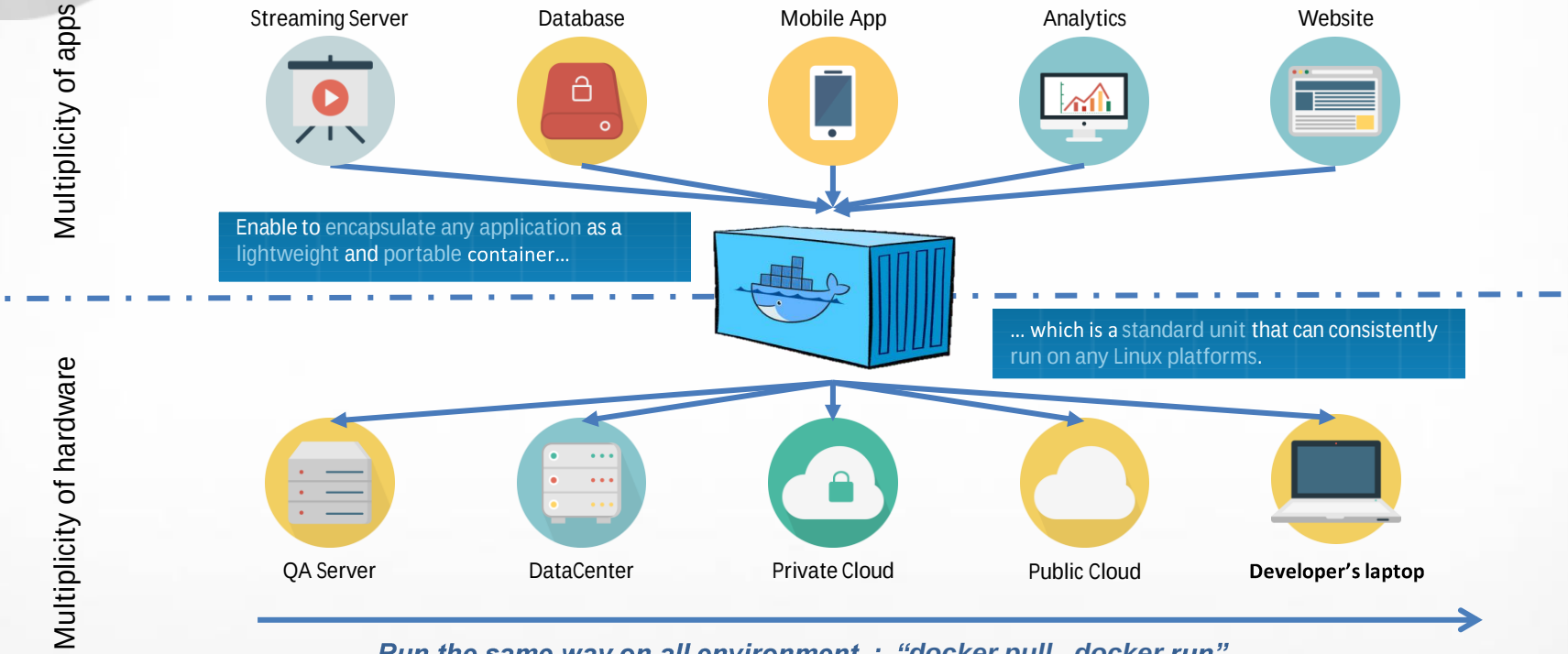
41%
want application portability across environments

35+%
want to avoid cloud vendor lock-in

Docker: Application portability. (the Container analogy)

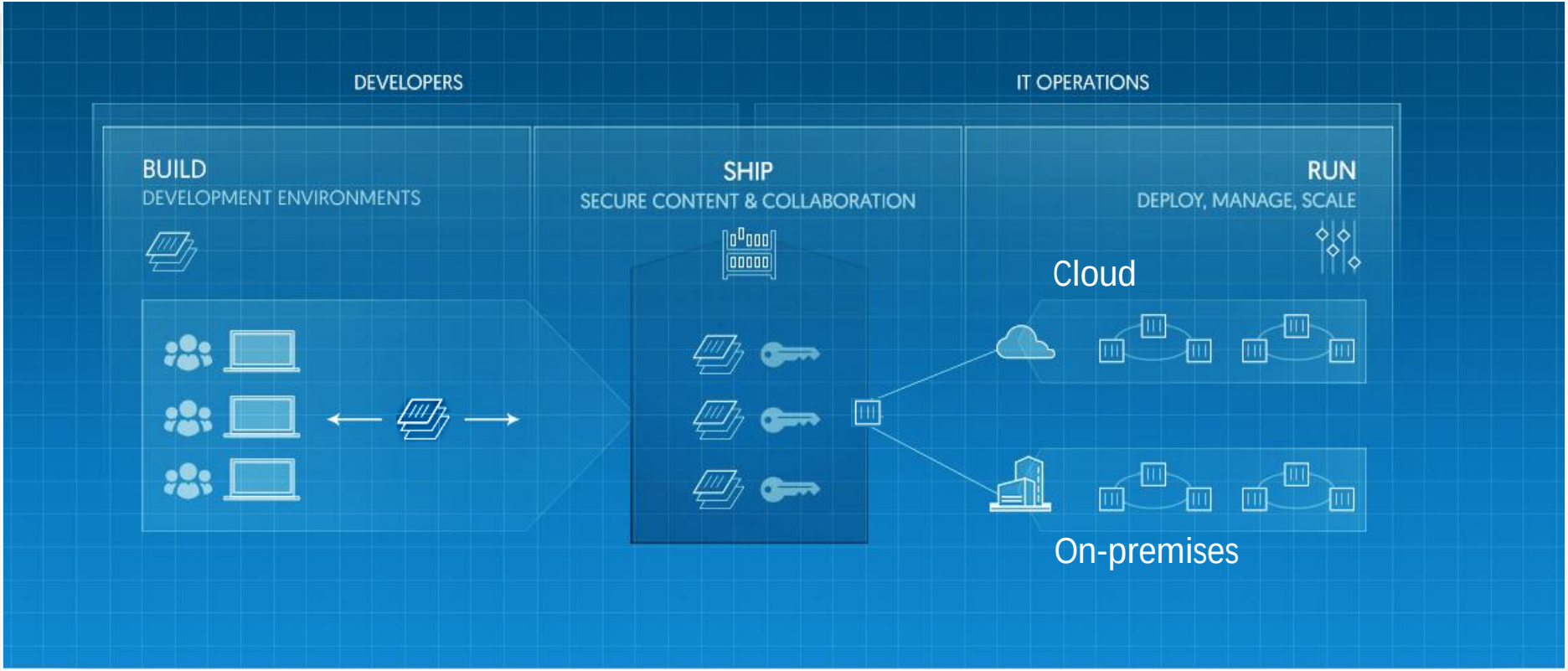


Docker: Application portability. (the Container analogy)



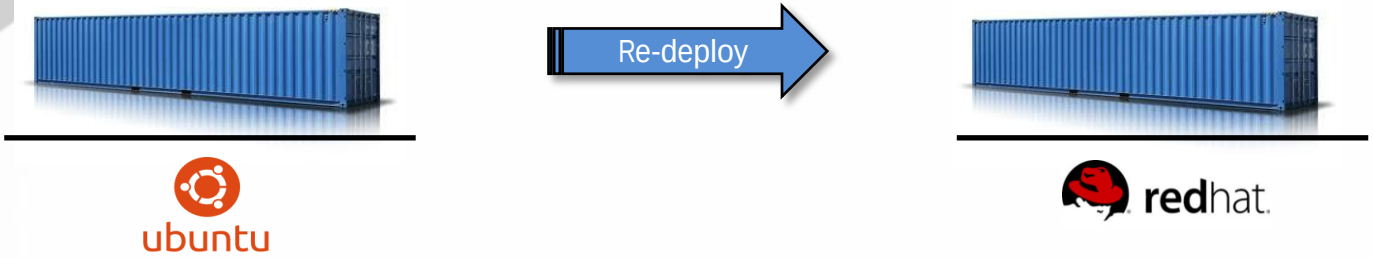
Run the same way on all environment : "docker pull , docker run"
No Operating System dependency : *(all application dependencies are in the container)*

Simplify deployment Process for Hybrid Cloud

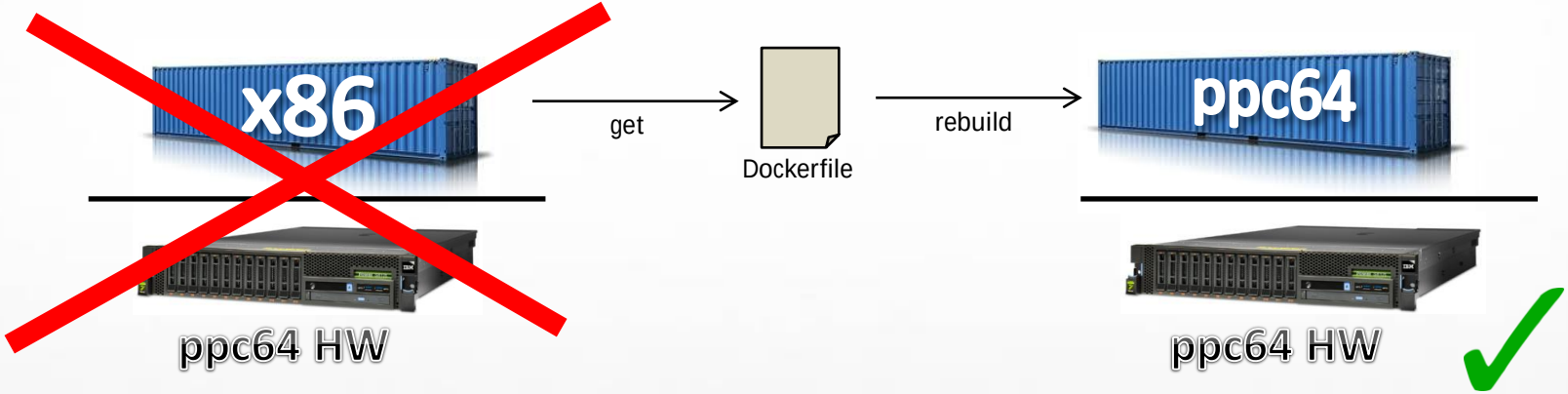


Important note: "Docker is cool but not Magic"

- Dockerized Application are OS independent



- … But they are Architecture/platform dependent (container is binary).



- Container can be easily rebuild from its Dockerfile.

Docker on non-x86 platform.

- Docker user experience is identical across platforms (CLI, API)
- Containers in binary form are not portable, either:
  - Binary code for Specific architecture needs to be complied into container

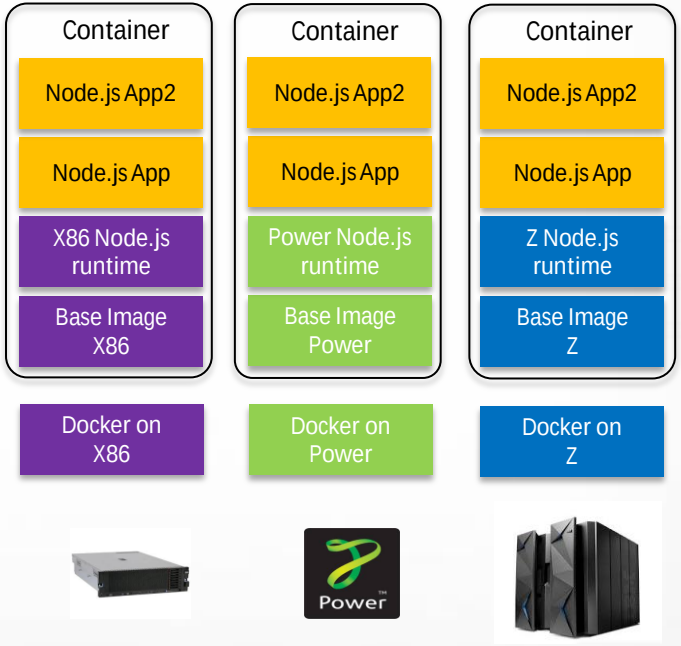
Docker linux packages for power are provided :

- In ubuntu for Power (since 15.04)
- For rpm based
  - In the Power Productivity tool : https://www-304.ibm.com/webapp/set2/sas/f/lopdiags/yum.html
  - By Unicamp repo : ftp://ftp.unicamp.br/pub/ppc64el/rhel/7_1/
- Docker latest binary for all arch are available here: https://master.dockerproject.org

Docker images for non x86 arch :

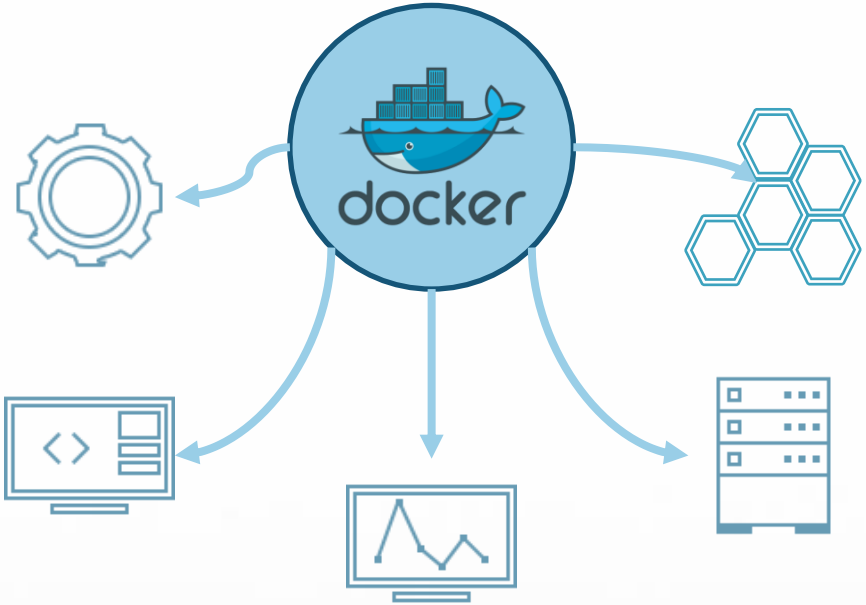
- POWER8 (ppc64le): https://hub.docker.com/u/ppc64le/
- System z (s390x): https://hub.docker.com/u/s390x/

Today, the top 250 most popular Docker containers are available



Try Docker on Power : https://www.ibm.com/developerworks/community/blogs/fe313521-2e95-46f2-817d-44a4f27eba32/entry/ubuntu_core_on_docker_for_power?lang=en

Docker Use Cases



CI/CD

Develop and test applications faster within any environment

DevOps

Break down barriers between Dev and Ops

Microservices

Embrace microservices architecture with containers technology

Infrastructure Optimization

Reduce infrastructure costs

Big data

Run big data workloads within containers

IBM **Client Center**
Montpellier

Follow us!

- Website: ibm.com/ibm/clientcenter/montpellier/
- Twitter: @IBMCCMPL
- YouTube: IBM Client Center Montpellier
- Pinterest: ibm-client-center-Montpellier
- IBM connections: w3-connections.ibm.com/communities/community/ibmccmpl