Docker Container

Agenda

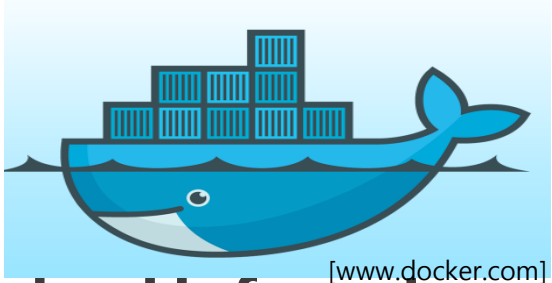
- What is Docker?
  - Docker vs. Virtual Machine
  - History, Status, Run Platforms
  - Hello World
- Images and Containers
- Volume Mounting, Port Publishing, Linking
- Around Docker, Docker Use Cases
- Hands-On Workshop

What is Docker?

Docker is an open-source project th at automates the deployment of ap plications inside software containers , by providing an additional layer of abstraction and automation of oper ating system–level virtualization o n Linux.

[Source: en.wikipedia.org]

Docker: Name

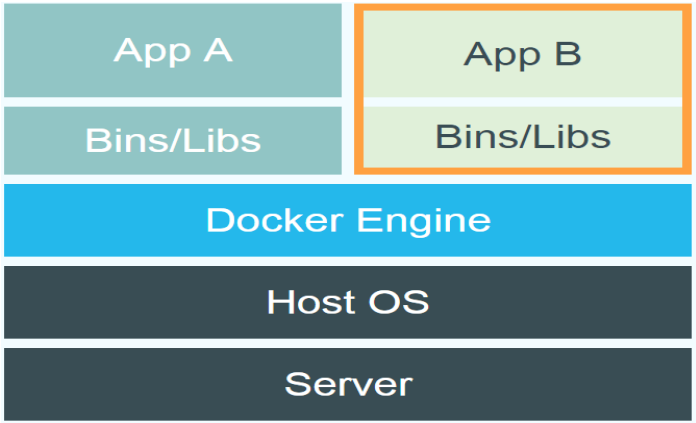

[www.docker.com]

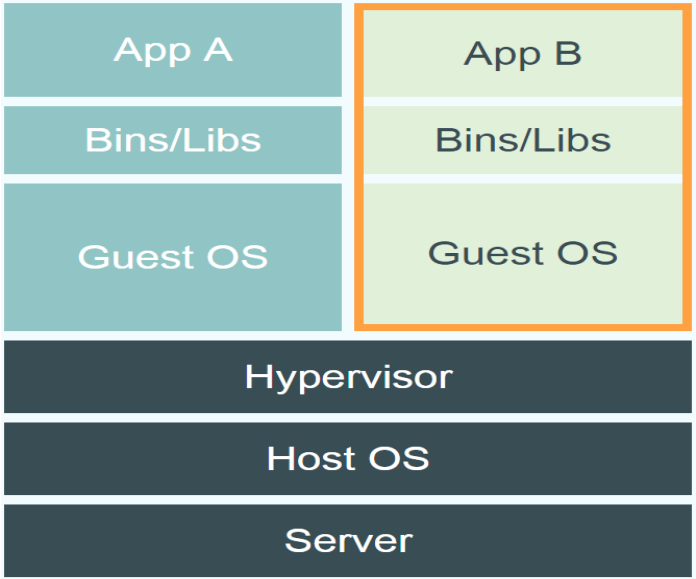
docker [naut.]: der Dockarbeiter, der Hafenarbeiter

Source: leo.org

- Provide a uniformed wrapper around a software package: *«Build, Ship and Run Any App, Anywhere»* [www.docker.com]
  - Similar to shipping containers: The container is always the same, regardless of the contents and thus fits on all trucks, cranes, ships, …

Docker vs. Virtual Machine

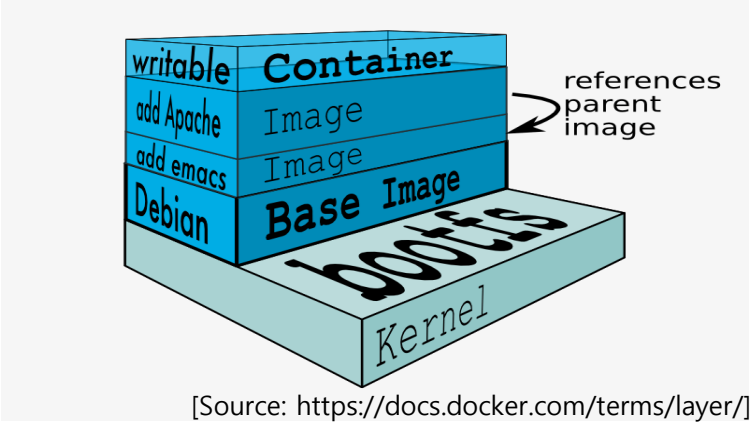




Source: https://www.docker.com/whatisdocker/

Docker Technology

- libvirt: Platform Virtualization
- LXC (LinuX Containers): Multiple is olated Linux systems (containers) on a single host
- Layered File System

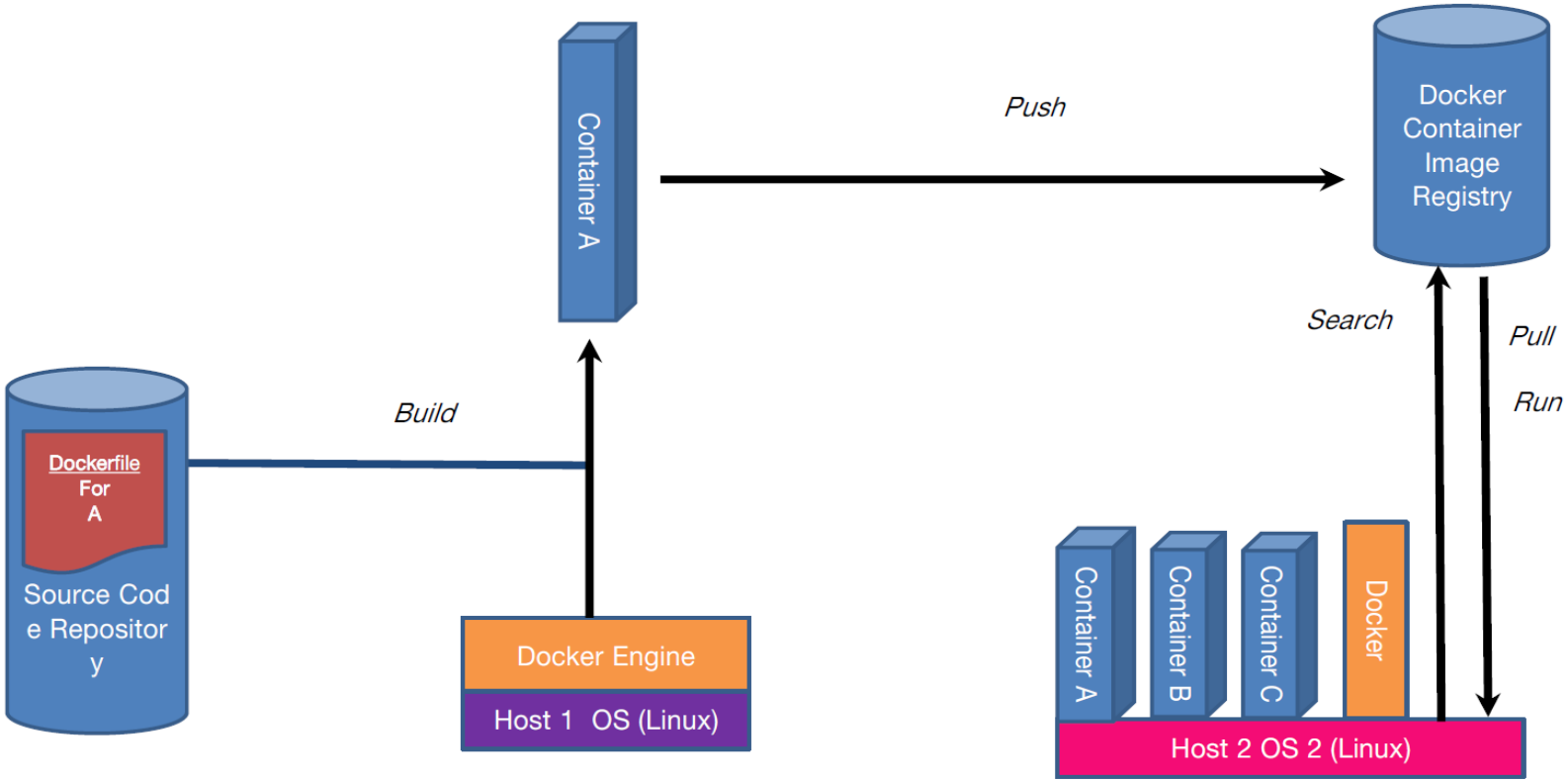


[Source: https://docs.docker.com/terms/layer/]

Docker History

- 2013-03: Releases as Open Source
- 2013-09: Red Hat collaboration (Fedora, RHEL, OpenShift)
- 2014-03: 34th most starred GitHub project
- 2014-05: JAX Innovation Award (most innovative open technology)

Docker Operation Env.



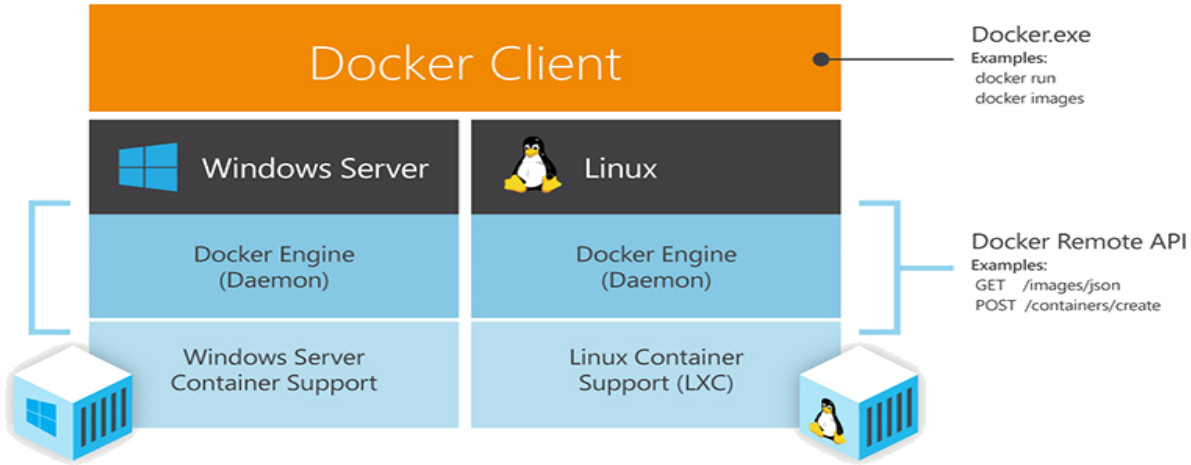
Why Docker?

- Lightweight Virtualization

	Ship	Manual deployment	Automated deployment	Boot
Bare Metal	Days	Hours	Minutes	Minutes
Virtualization (VM)	Minutes	Minutes	Seconds	Less than minutes
Lightweight Virtualization (Docker)	seconds	Minutes	Seconds	Seconds

Run Platforms

- Various Linux distributions (Ubuntu, Fedora, RHEL, Centos, openSUSE,,)
- Cloud (Amazon EC2, Google Compute Engine, Rackspace)
- Microsoft integrate Docker with Windows (Azure Platform)



Hello World

Simple Command - Ad-Hoc Container

- `docker run ubuntu echo Hello World`
  - `docker images [-a]`
  - `docker ps -a`

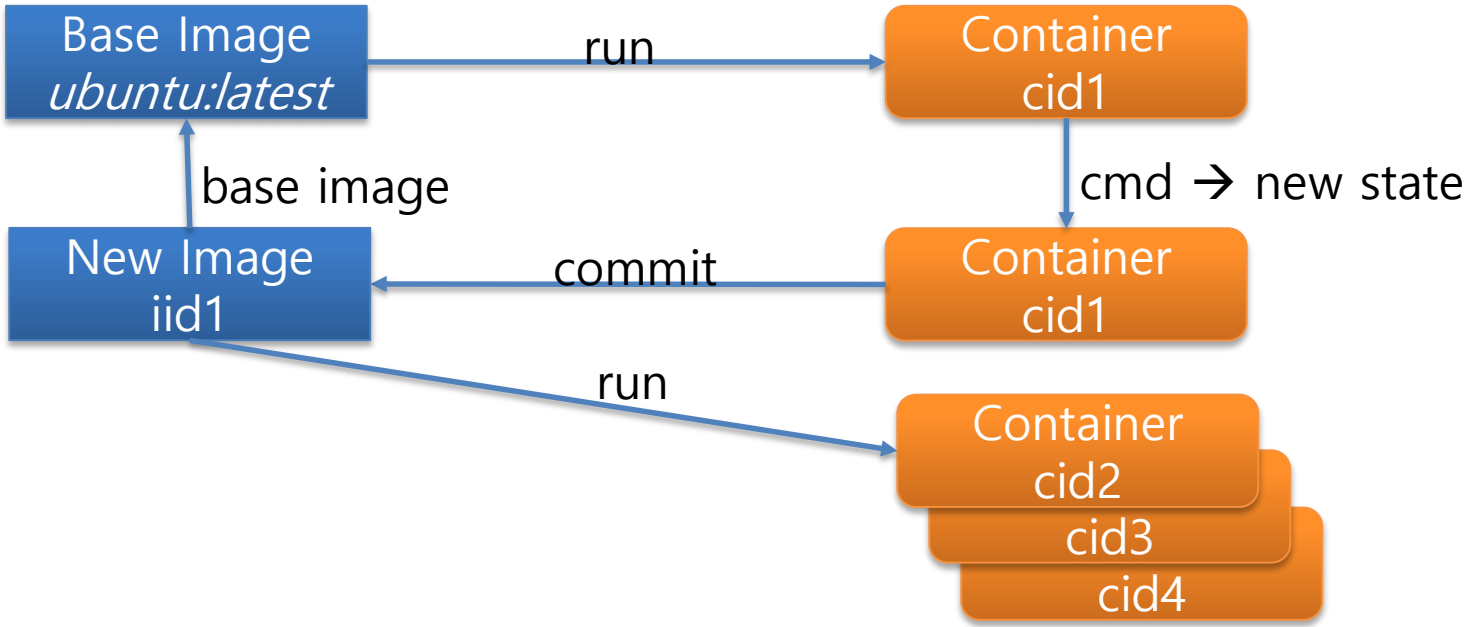
Terminology - Image

- Persisted snapshot that can be run
  - *images:* List all local images
  - *run*: Create a container from an image and execute a command in it
  - *tag*: Tag an image
  - *pull*: Download image from repository
  - *rmi*: Delete a local image
    - This will also remove intermediate images if no longer u sed

Terminology - Container

- Runnable instance of an image
  - *ps:* List all running containers
  - *ps –a*: List all containers (incl. stopped)
  - *top*: Display processes of a container
  - *start*: Start a stopped container
  - *stop*: Stop a running container
  - *pause*: Pause all processes within a container
  - *rm*: Delete a container
  - *commit*: Create an image from a container

Image vs. Container



Dockerfile

- Create images automatically using a build script: «Dockerfile»
- Can be versioned in a version con trol system like Git or SVN, along with all dependencies
- Docker Hub can automatically bui ld images based on dockerfiles on Github

Dockerfile Example

- Dockerfile:
  - ```
    FROM ubuntu
    ENV DOCK_MESSAGE Hello My World
    ADD dir /files
    CMD ["bash", "someScript"]
    ```
- `docker build [DockerFileDir]`
- `docker inspect [imageId]`
    ```

Mount Volumes

- `docker run -ti` **`-v /hostLog :/log`** `ubuntu`
- Run second container: Volume ca n be shared
  - `docker run -ti --volumes-fro m firstContainerName ubuntu`

Publish Port

- `docker run -t -p 8080:80 ubuntu nc -l 80`
  - Map container port 80 to host port 8080
  - Check on host: nc localhost 8080
- Link with other docker container
  - `docker run -ti --link containerName:alias ubuntu`
  - See link info with `set`

Around Docker

- Docker Images: Docker Hub
- Vagrant: «Docker for VMs»
- Automated Setup
  - Puppet, Chef, Ansible, ...
- Docker Ecosystem
  - skydock / skydns
  - fig

Docker Hub

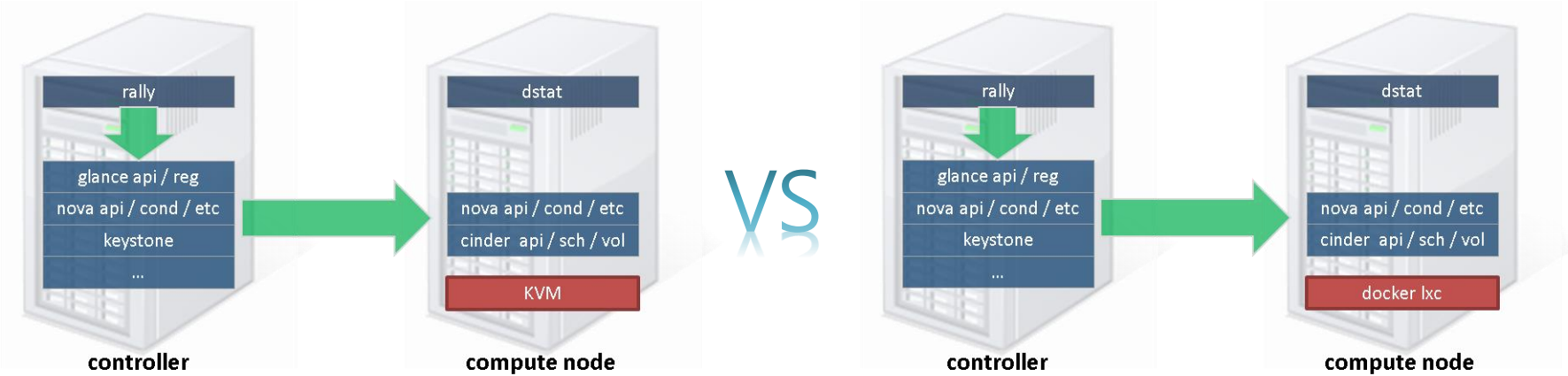
- Public repository of Docker images
  - https://hub.docker.com/
  - docker search [term]
- Automated: Has been automatically built from Dockerfile
  - Source for build is available on GitHub

Docker Use Cases

- Development Environment
- Environments for Integration Tests
- Quick evaluation of software
- Microservices
- Multi-Tenancy
- Unified execution environment (dev → test → prod (local, VM, cloud, ...)

VM vs Docker Performance Evaluation (IBM Data)

- OpenStack Controller with KVM and Docker

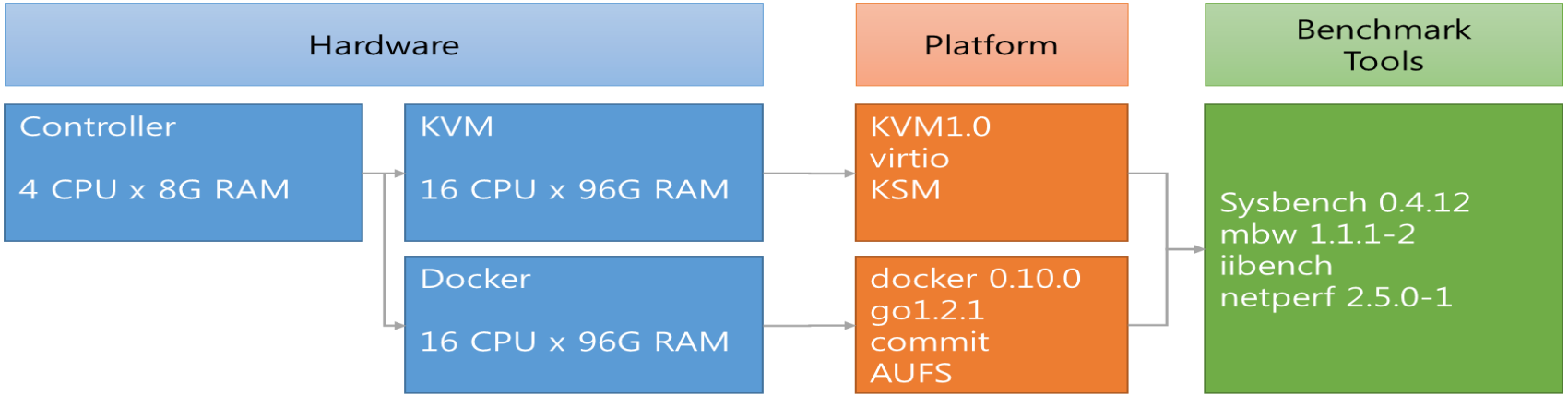


KVM with OpenStack

Docker with OpenStack

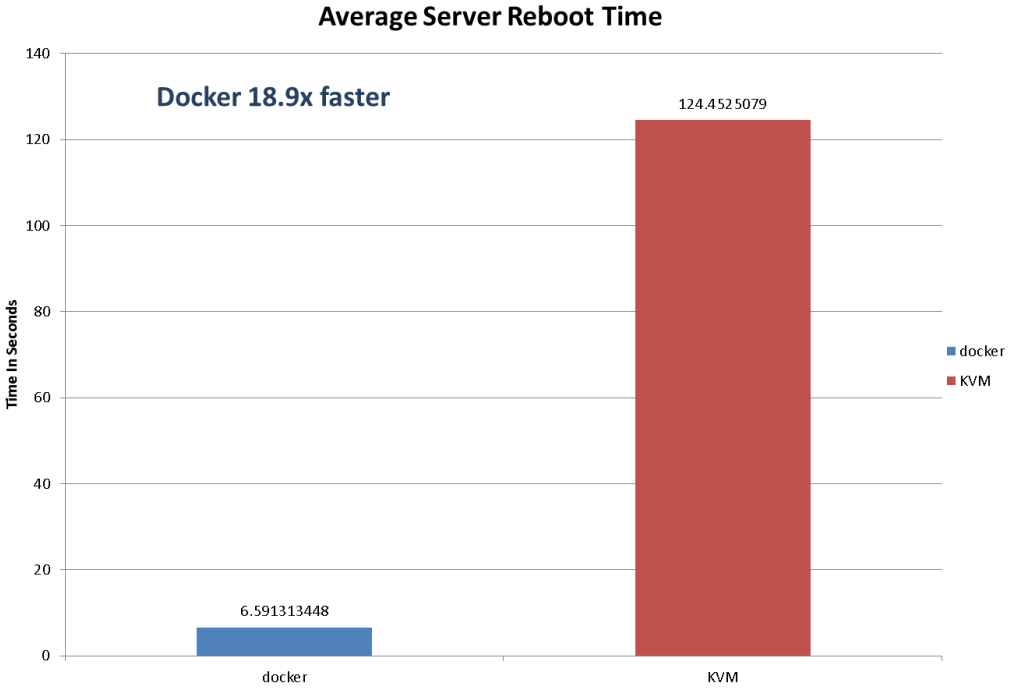
VM vs Docker Performance Evaluation (IBM Data)

- Test Env.



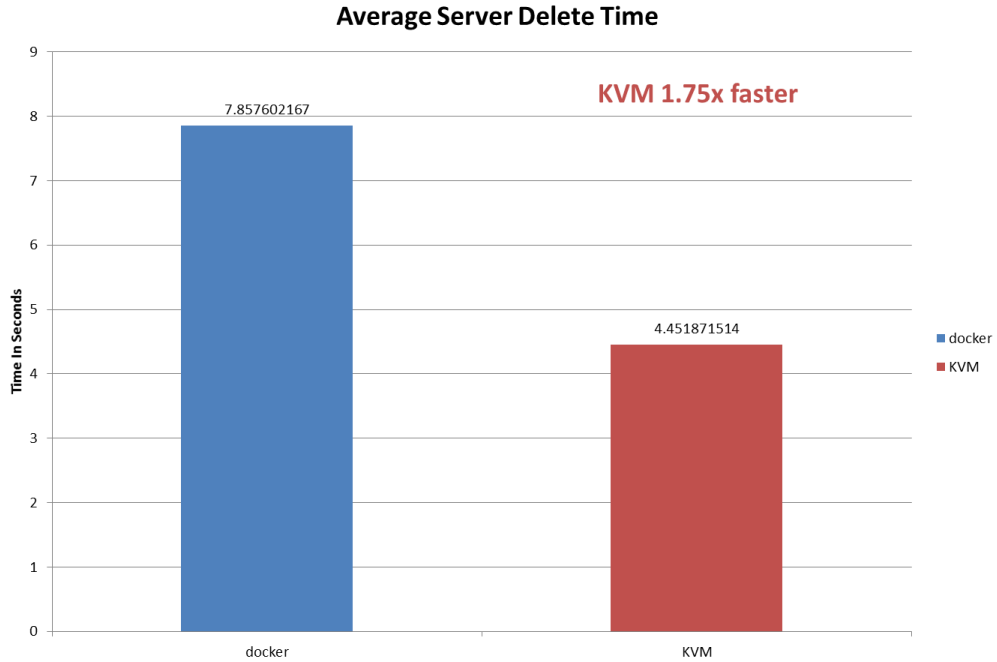
VM vs Docker Performance Evaluation (IBM Data)

- Reboot Test



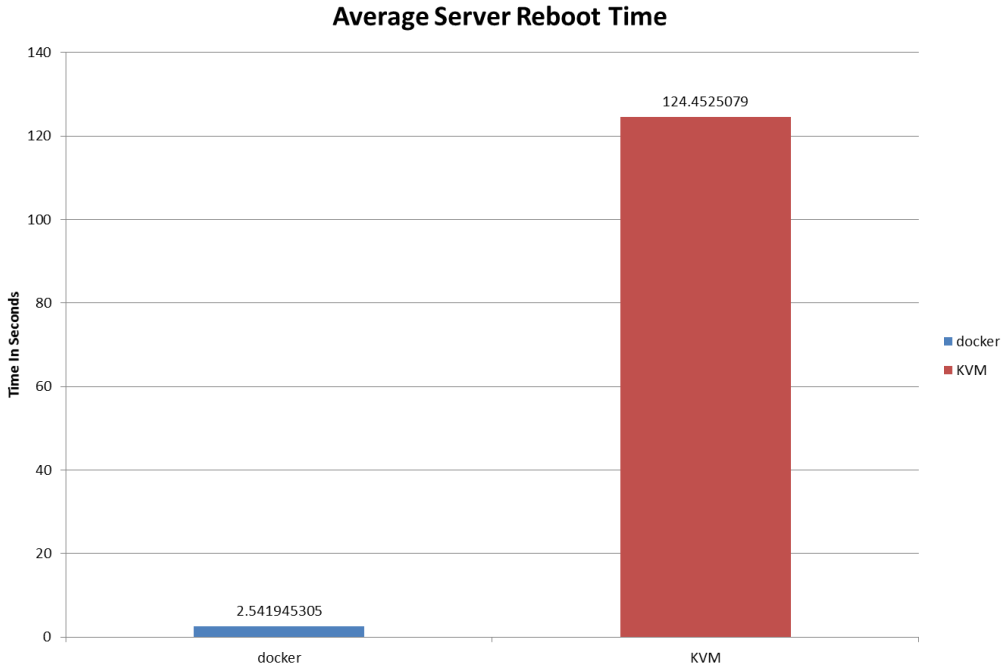
VM vs Docker Performance Evaluation (IBM Data)

- Delete Test



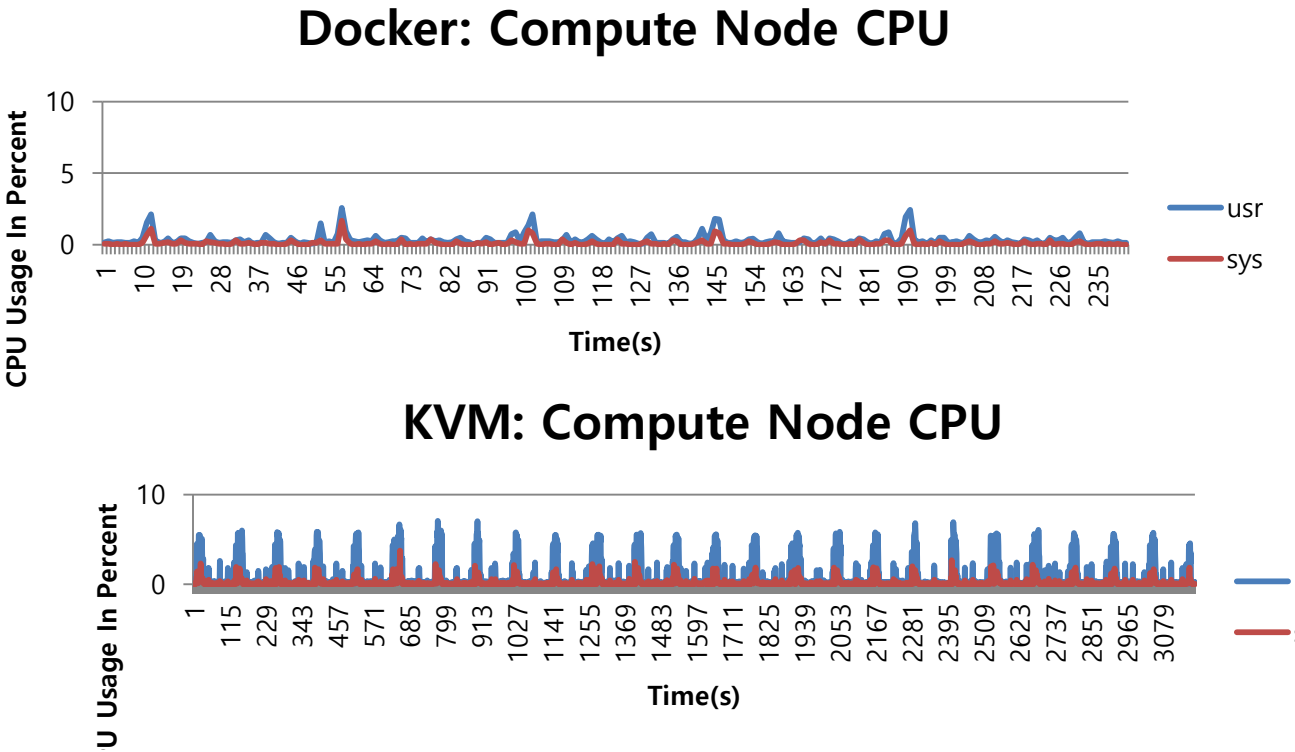
VM vs Docker Performance Evaluation (IBM Data)

- Reboot Test



VM vs Docker Performance Evaluation (IBM Data)

- Resource Usage



Documentation

- Docker homepage: https://www.docker.com/
  - Introduction: https://www.docker.com/whatisdocker/
  - Online tutorial: https://www.docker.com/tryit/
  - Installation and user guide: https://docs.docker.com/
- InfTec TecBoard: https://inftec.atlassian.net/wiki/display/TEC/Docker
  - Includes this presentation