

DGSI Training DAY 1

- 9:00 - 10:20 [*Presentation*]
 - Introduction: why, what, where, how, platforms, setup
- 10:20 - 10:40 COFFEE BREAK
- 10:20 - 12:00 [*Hands-on*]
 - Basic commands & information about Docker
- 12:00 - 14:00 LUNCH BREAK
- 14:00 - 15:30 [*Hands-on*]
 - Docker image manipulation
- 15:30 - 15:45 COFFEE BREAK
- 15:45 - 17:00 [*Hands-on*]
 - Docker container run-time networking

History

1979: Unix V7 *chroot*

2000: FreeBSD *Jails* system partitions with IP

2001: Linux VServer *Jails* resource partitions on patched Linux kernel

2004: Oracle (SUN) Solaris *Containers* system zones, cloning from ZFS

2006: Google Process *Containers* renamed to *cgroups*
























































2008: *Linux Containers* (LXC) using *cgroups* and namespaces

2013: Google's *Let Me Contain That For You* (LMCTFY)->*libcontainer*

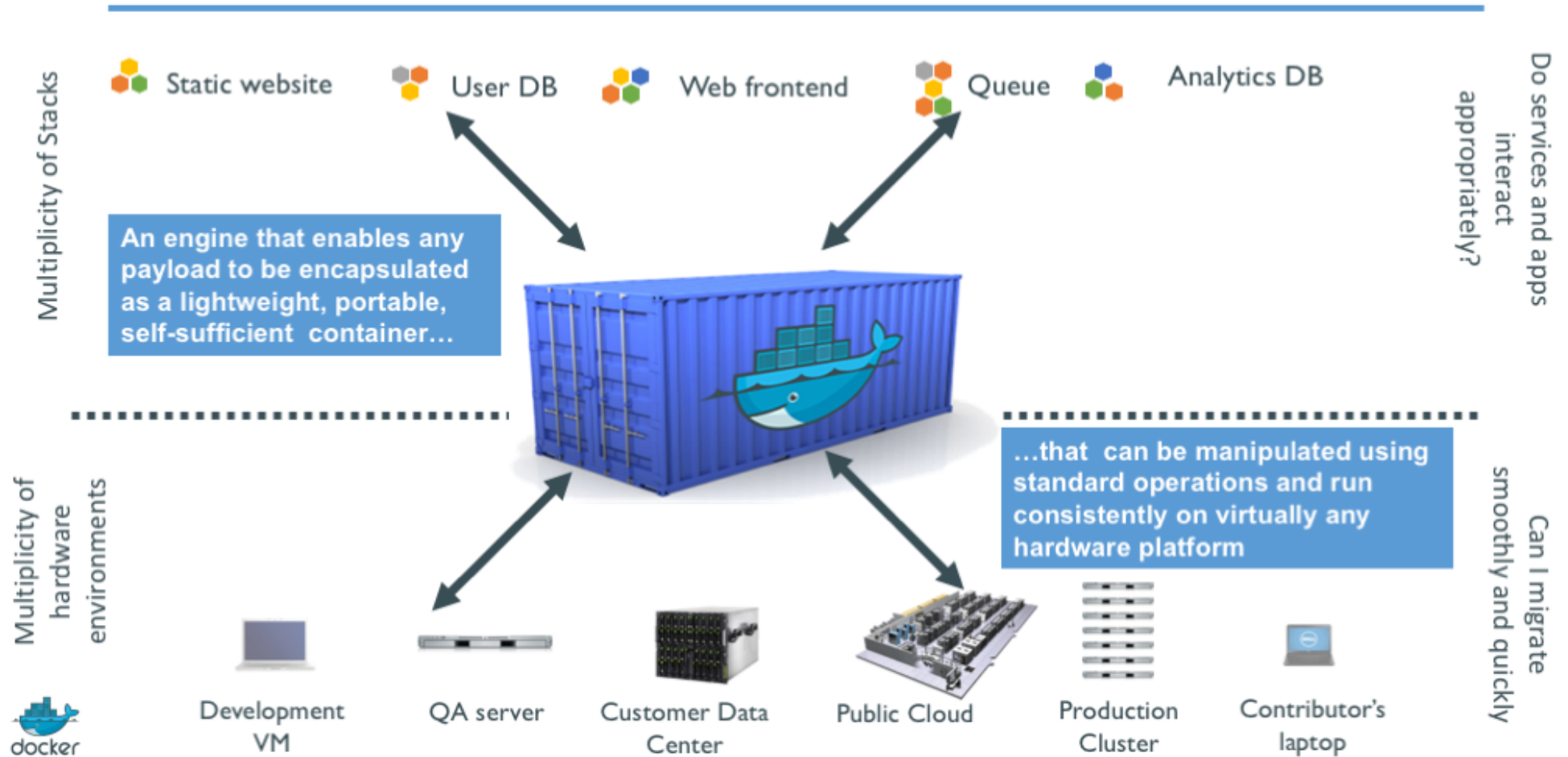
2013: Docker, Docker CLI, Docker Hub = full ecosystem

2014 CoreOS Rkt (*Rocket*)

Matrix from hell

	Static website							
	Web frontend							
	Background workers							
	User DB							
	Analytics DB							
	Queue							
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
								

Shipping containers



Separation of Concerns

- Dan the Developer

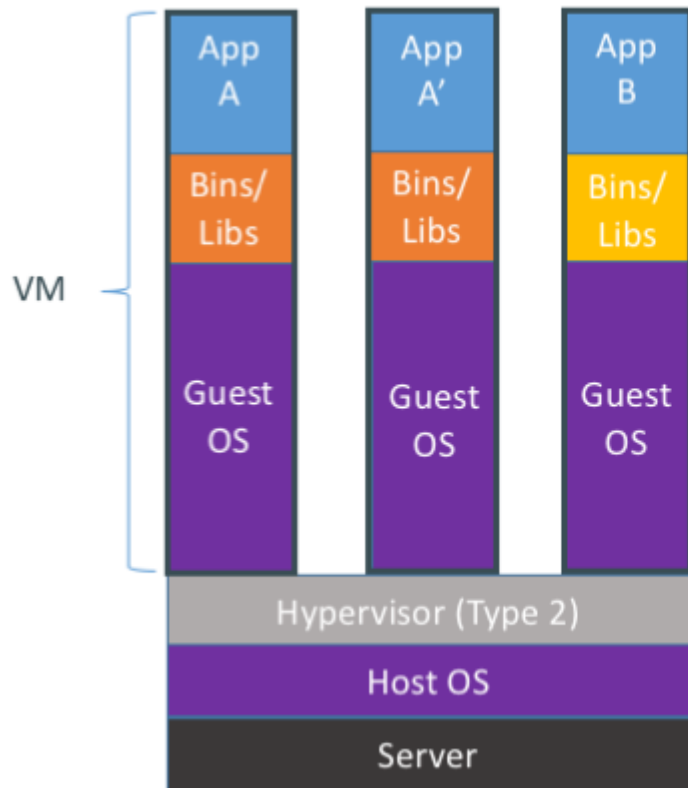
- Worries about what's "inside" the container
 - His code
 - His Libraries
 - His Package Manager
 - His Apps
 - His Data
- All Linux servers look the same

- Oscar the Ops Guy

- Worries about what's "outside" the container
 - Logging
 - Remote access
 - Monitoring
 - Network configuration
- All containers start, stop, copy, attach, migrate, etc. the same way

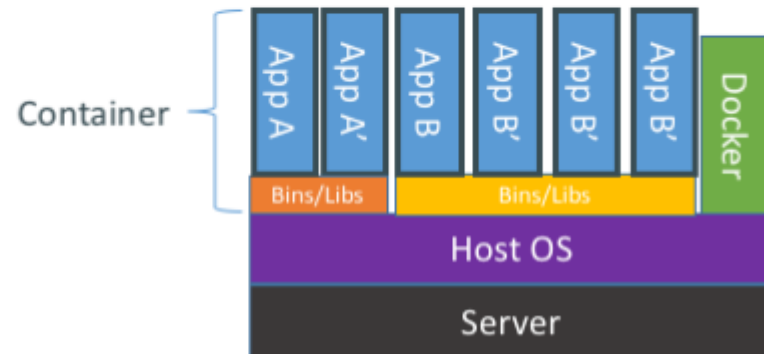


Virtualization vs. Isolation

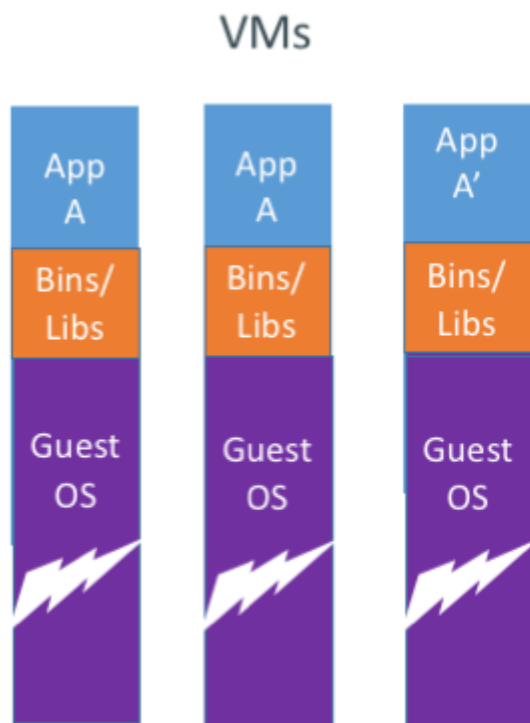


Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



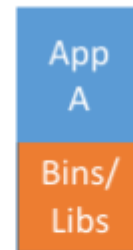
VMs vs. Docker containers



VMs

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

Containers



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'

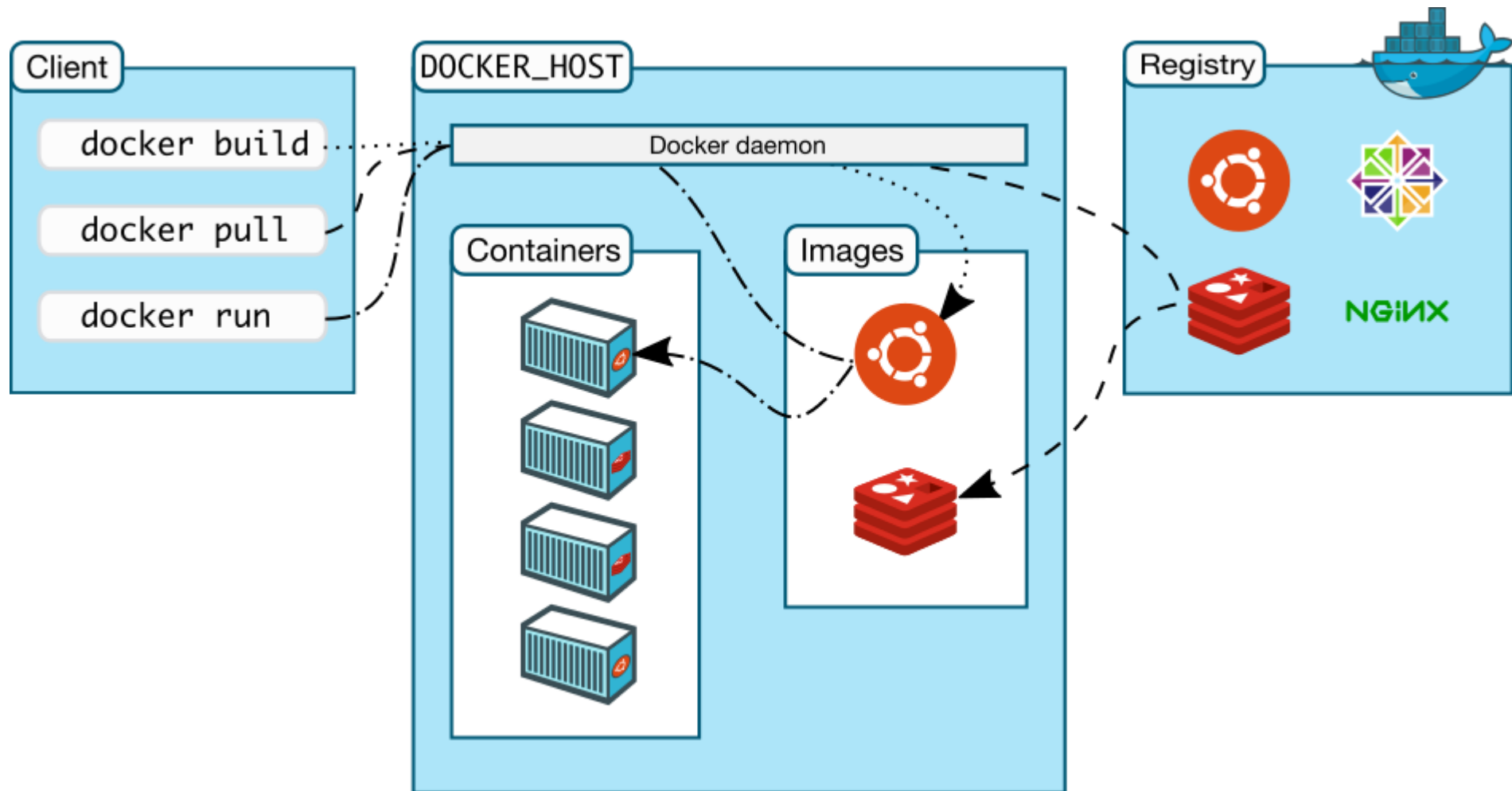
Why Docker

- Run everywhere
 - Regardless of kernel version (2.6.32+)
 - Regardless of host distro
 - Physical or virtual, cloud or not
 - Container and host architecture must match*
- Run anything
 - If it can run on the host, it can run in the container
 - i.e. if it can run on a Linux kernel, it can run

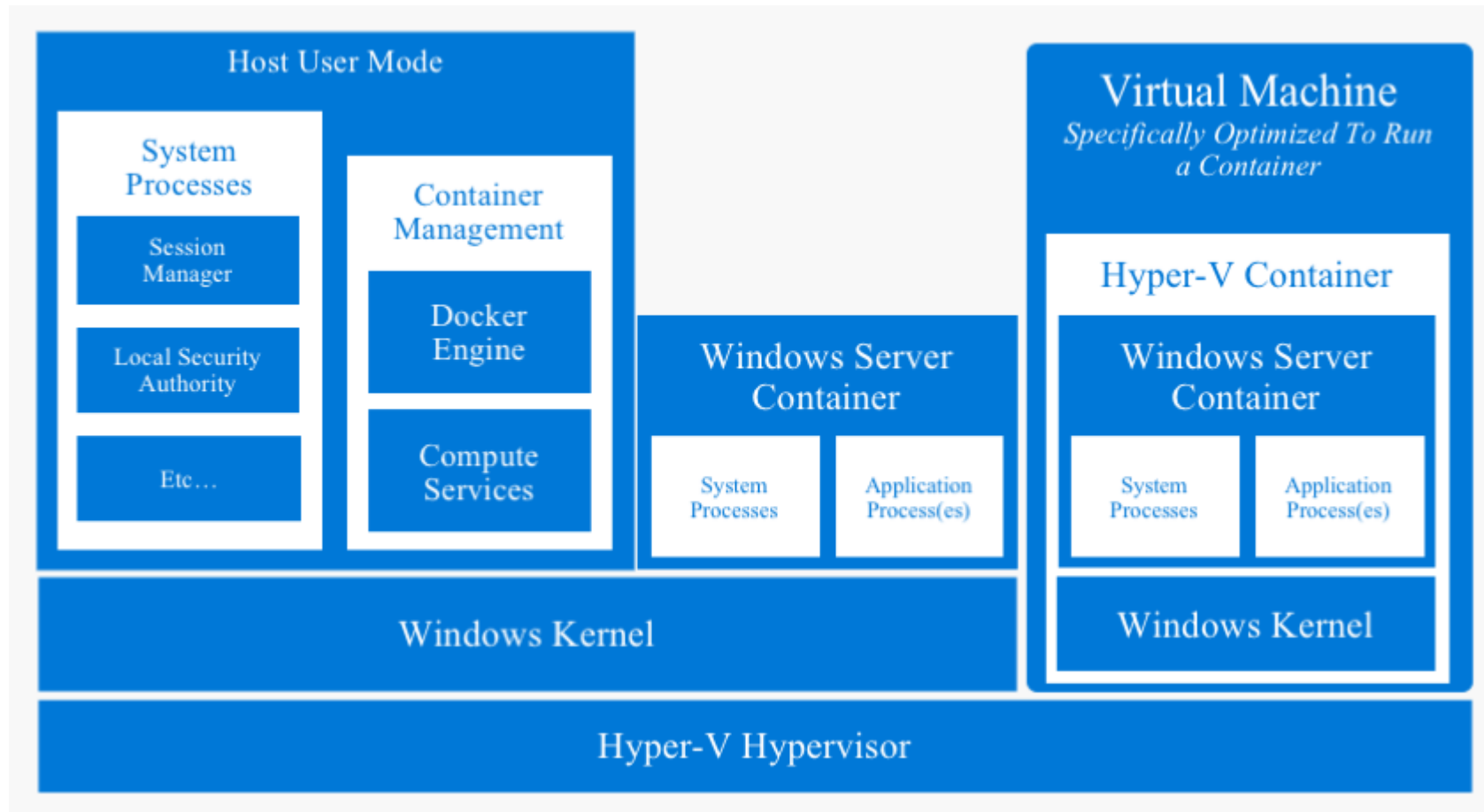
What is Docker

- High Level—It's a lightweight VM
 - Own process space
 - Own network interface
 - Can run stuff as root
 - Can have its own /sbin/init (different from host)
 - <<machine container>>
- Low Level—It's chroot on steroids
 - Can also not have its own /sbin/init
 - Container=isolated processes
 - Share kernel with host
 - No device emulation (neither HVM nor PV) from host)
 - <<application container>>

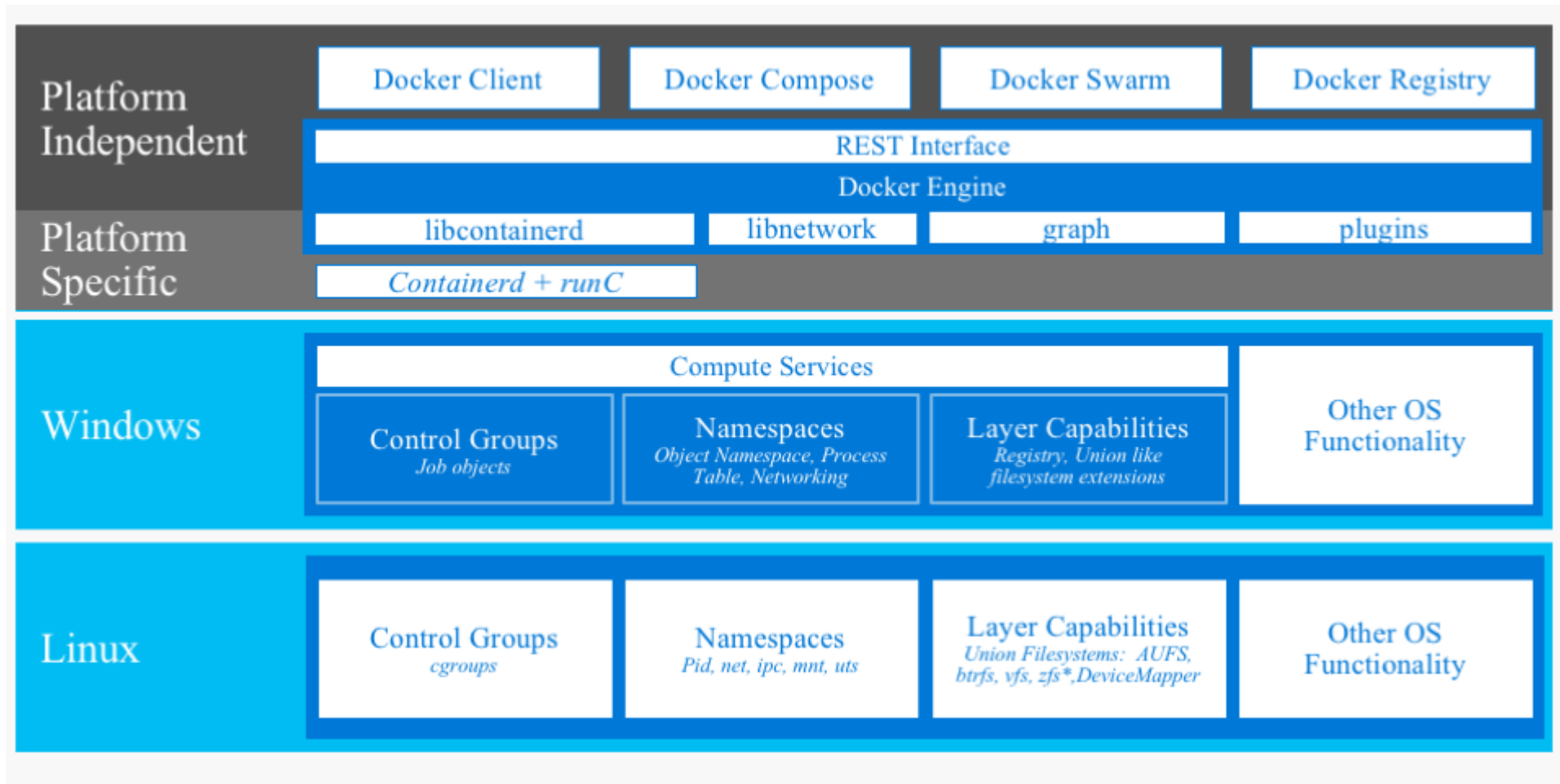
Docker on Linux



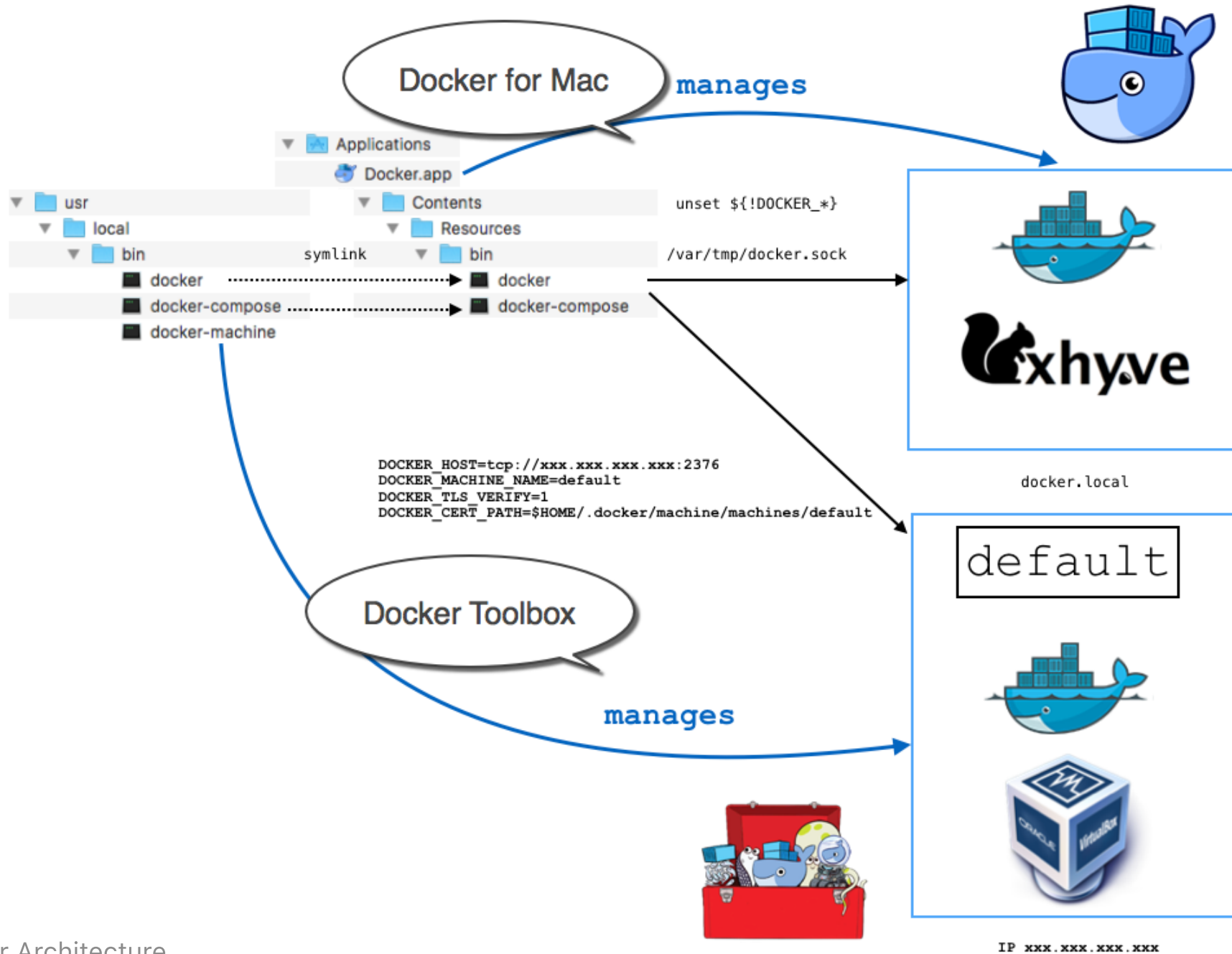
Docker on Windows



Comparison Linux vs. Windows



Docker on Mac OS X



Linux (URL)

```
$ subscription-manager register
$ subscription-manager repos \
    --enable=rhel-7-server-rpms \
    --enable=rhel-7-server-extras-rpms \
    --enable=rhel-7-server-optional-rpms

$ yum install docker \
    device-mapper-libs device-mapper-event-libs
$ systemctl start docker.service
$ systemctl enable docker.service
$ systemctl status docker.service
docker.service - Docker Application Container Engine
   Loaded: loaded (/usr...stem/docker.service; enabled;...
   Drop-in: /usr/lib/systemd/system/docker.service.d
            |-flannel.conf
   Active: active (running) since ...
   Docs: http://docs.docker.com
  Main PID: 13495 (sh)
   CGroup: /system.slice/docker.service
           └─13495 /bin/sh -c /usr/bin/docker-current ...

...

```

Mac OS X (URL)

Install Docker for Mac

Estimated reading time: 6 minutes

Docker for Mac is a [Docker Community Edition \(CE\)](#) app. The Docker for Mac install package includes everything you need to run Docker on a Mac. This topic describes pre-install considerations, and how to download and install Docker for Mac.

Already have Docker for Mac? If you already have Docker for Mac installed, and are ready to get started, skip to [Get started with Docker for Mac](#) for a quick tour of the command line, preferences, and tools.

Looking for Release Notes? [Get release notes for all versions here.](#)

Download Docker for Mac

If you have not already done so, please install Docker for Mac. You can download installers from the Stable or beta channel.

Both Stable and Edge installers come with [experimental features in Docker Engine](#) enabled by default and configurable on [Docker Daemon preferences](#) for experimental mode. We recommend that you disable experimental features for apps in production.

On both channels, we welcome your [feedback](#)) to help us as the apps evolve.

For more about Stable and Edge channels, see the [FAQs](#).

Stable channel

This installer is fully baked and tested. This is the best channel to use if you want a reliable platform to work with. These releases follow the Docker Engine stable releases.

On this channel, you can select whether to send usage statistics and other data.

Stable builds are released once per quarter.

[Get Docker for Mac \(Stable\)](#)

Edge channel

This installer provides the latest Edge release of Docker for Mac and Engine, and typically offers new features in development. Use this channel if you want to get experimental features faster, and can weather some instability and bugs. We collect all usage data on Edge releases across the board.

Edge builds are released once per month.

[Get Docker for Mac \(Edge\)](#)

Windows (URL)

```
$version = (Invoke-WebRequest -UseBasicParsing  
https://raw.githubusercontent.com/docker/docker/master/VERSION).  
Content.Trim()
```

```
Invoke-WebRequest
```

```
"https://master.dockerproject.org/windows/amd64/docker-$(  
-UseBasicParsing  
-OutFile "$env:TEMP\docker.zip"
```

```
Expand-Archive -Path "$env:TEMP\docker.zip"  
-DestinationPath $env:ProgramFiles
```

```
$env:path += ";$env:ProgramFiles\Docker"
```

```
$existingMachinePath = [Environment]::GetEnvironmentVariable(  
"Path", [System.EnvironmentVariableTarget]::Machine)
```

```
[Environment]::SetEnvironmentVariable(  
"Path", $existingMachinePath + ";$env:ProgramFiles\Docker",  
[EnvironmentVariableTarget]::Machine)
```

```
dockerd --register-service
```

```
Start-Service Docker
```


Docker basic workflow

