Red Hat Container Technology Strategy

Containers are so 2014

Clayton Coleman
Daniel Riek
April 2017

What we told you earlier:

The future of the Linux OS is a scale-out cluster-as-computer platform for fully orchestrated multi-container apps, providing an abstraction layer across underlying infrastructure, and breaking the vertical integration of proprietary cloud.

Computing is Ubiquitous

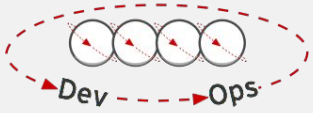
Open Source & The Cloud fuel the cycle of Digital Transformation

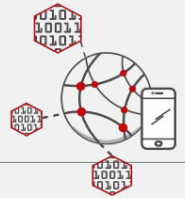
“The future is already here — it's just not very evenly distributed.”

- **William Gibson**

Bridging Across The Mega Trends

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



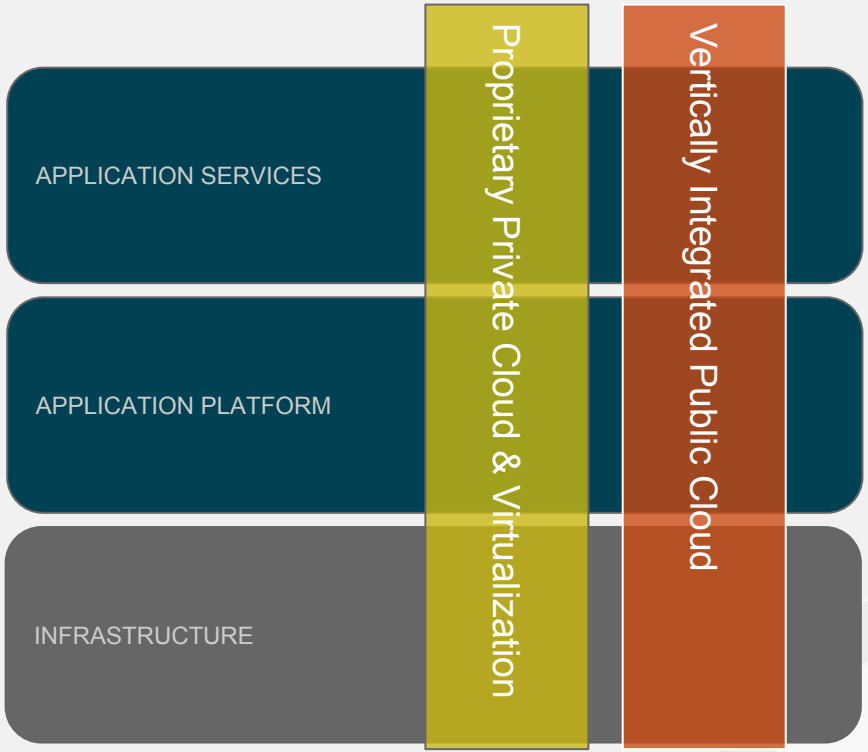


Cloud is Vertical Integration

Public and Private Cloud platforms drive value through vertical Integration from the Infrastructure up.

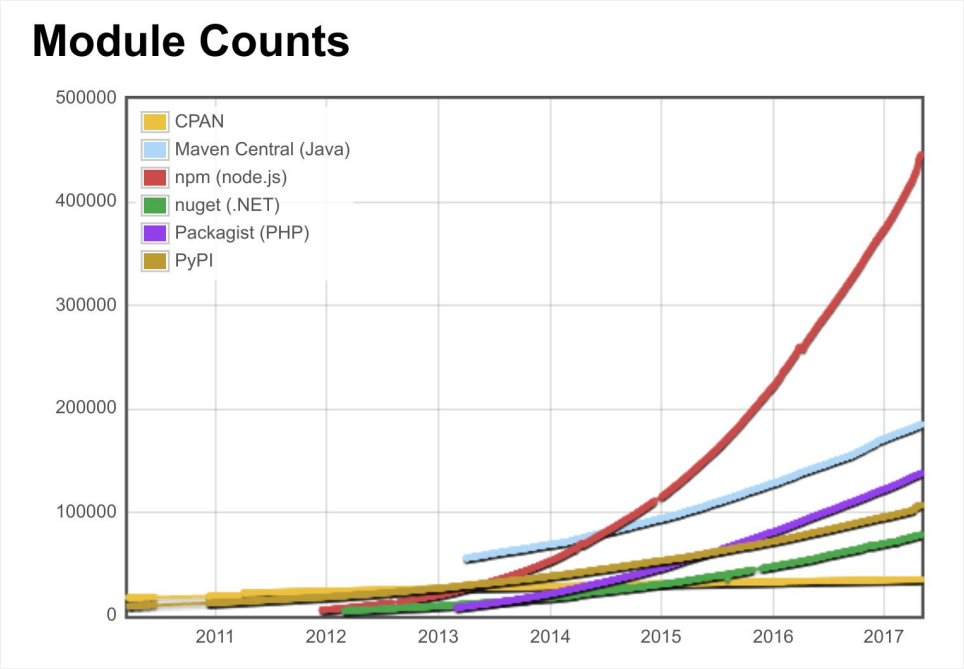
- Infrastructure elasticity.
- Operational excellence by default.
- Ready to use services.
- Cost can be tied to actual consumption.

Vertical integration allows focus on the business differentiators rather than the underlying plumbing.



Software Stack Complexity Keeps Growing

Everyone pays a Strategy Tax, Red Hat's Strategy Tax is Open Source

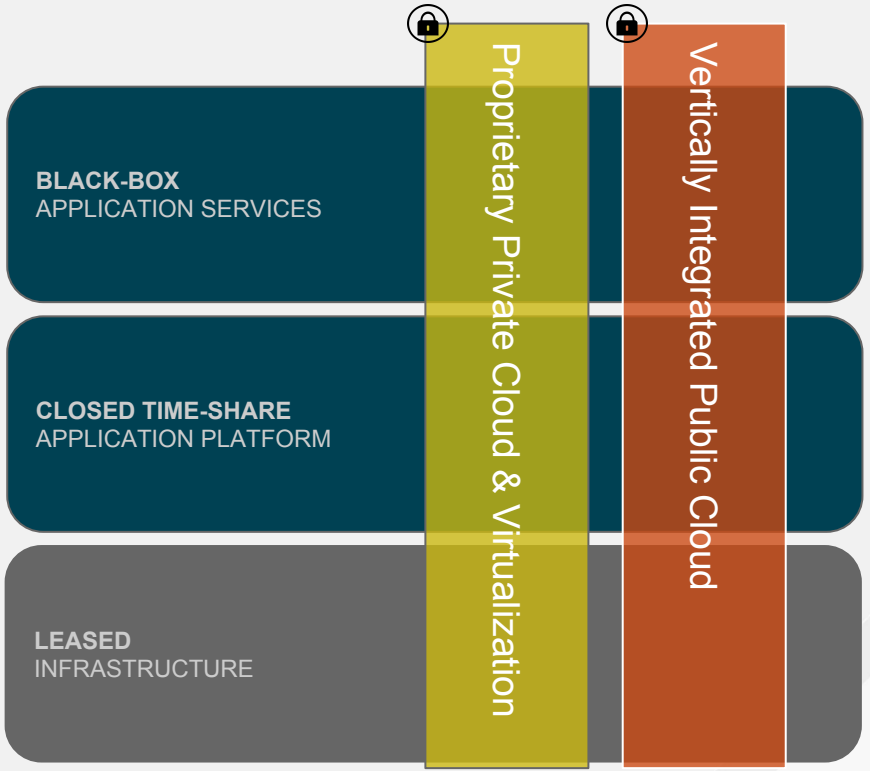


Source: http://www.modulecounts.com/

The New Mainframe?

Vertical Integration is inherently selective. With centralization it can become limiting.

- The exponential and revolving growth of Open Source innovation will at some point disrupt every service.
- This includes the public cloud.
- How to pivot if one is all-in on vertical integration?



Cloud vs Open Source

Cousins, not twins.

Open Source is the sustaining catalyst of the third industrial revolution. It is impossible for any one company to control how software is built and evolves - communities cannibalize all proprietary innovation.

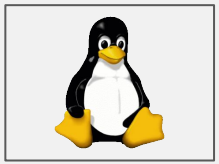
To empower developers and innovators we explicitly build and sustain those communities.

Infrastructure exists only to run applications. Why do we keep designing services focused on infrastructure?

One More Thing...

Let's reflect on what the Cloud is about

What is in the Box?



RED HAT SUMMIT

APPLICATION!

Red Hat ~~Container~~ Technology Strategy

In Reality, It's About Applications - An Analogy

A Container

Application in a Container

The Application User

More Containers

Million Dollar Question:

What did the user care about? The Container or the Application?

So Containers Don’t Matter?

Not quite...

The App-Centric Platform

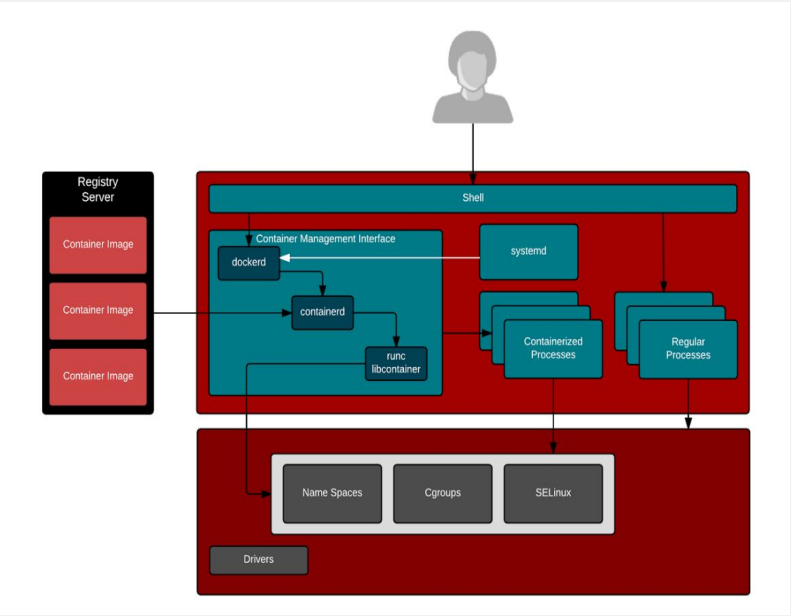
Red Hat's Container Technology Strategy: Make Everything Boring

Your core business should be exciting, not the task of operating the application stack supporting it!

Boring things you should just be able to use

- Containers
- Kubernetes
- High Availability
- Scaleability
- Cloud Services
- Security
- Application Content
- Lifecycle
- Day 2+ Operations

Containers Are Linux



- ‘Containers’ are just a name for a set of kernel features, that have been in Linux since up to a decade.
- Running containers just are Linux processes.
- Linux user space runtime is inside the container.
- Containerized apps inherit
  - Linux Security
  - Linux Performance
  - Linux Reliability
  - Linux Ecosystem
- OCI (aka Docker) provides a package format and runtime standard for Linux Containers.

Mission: Make Containers Boring

Containers are a tool to serve your application

Just like RHEL, Containers should be...

- Predictable
- Reliable
- Implicit
- Consistent
- Safe & Secure
- Ubiquitous

Aggregate packaging for binary stacks and free you from Dependency Hell.

The default way to run your application processes as a core part of the Red Hat Enterprise Linux OS.

Kubernetes Makes Applications Great

A Container does not stand alone

Even in traditional IT, Applications are multi-tier. This translates to multiple services.

Kubernetes commoditizes application operations patterns:

- Cluster orchestration
- Service Linking
- High-Availability
- Scalability
- Portability
- Resource management

For both, traditional stateful and cloud native applications.

Kubernetes Has Won

* You may just not have realized it yet

Why?

It’s about the applications, and the community, and the patterns, and the technology.

It’s about being good enough and willing to grow in different directions.

It’s about true Open Source.

Mission: Make Kubernetes Boring

Because we are about to take off

If you're sitting on top of fifteen hundred metric tons of rocket fuel, you want...

- Predictable
- Self-tuning
- Self-correcting
- Extensible and flexible
- Easily monitorable
- Secure all the way down

How: Application Security

We live in a hostile world, and we need an immune system

Security is becoming harder - how do we make it easier AND safer?

- Bring together the layers of security that already exist
- End to end network *and* transport security, partitions and policy
- Improve the supply chain and find problems faster (in situ)
- Applications that automatically receive and rotate secrets
- Tie identity of applications to cluster - not new, Kerberos has done this for years

How: Run everything ever, all together

Relationship status: It's complicated

What kinds of applications will you run?

- 12-factor microservices which value **agility and discovery**
- Lift-and-shift vintage IT and line-of-business applications which value **adaptation**
- Even lighter frameworks, like functions-as-a-service, which value **efficiency**
- Stateful tools and components which value **stability and reliability**
- Big data, batch, and high-perf workloads which value **performance and scale**

How: New Development Paradigms

Does anyone **really** want to learn how to program in a new framework?

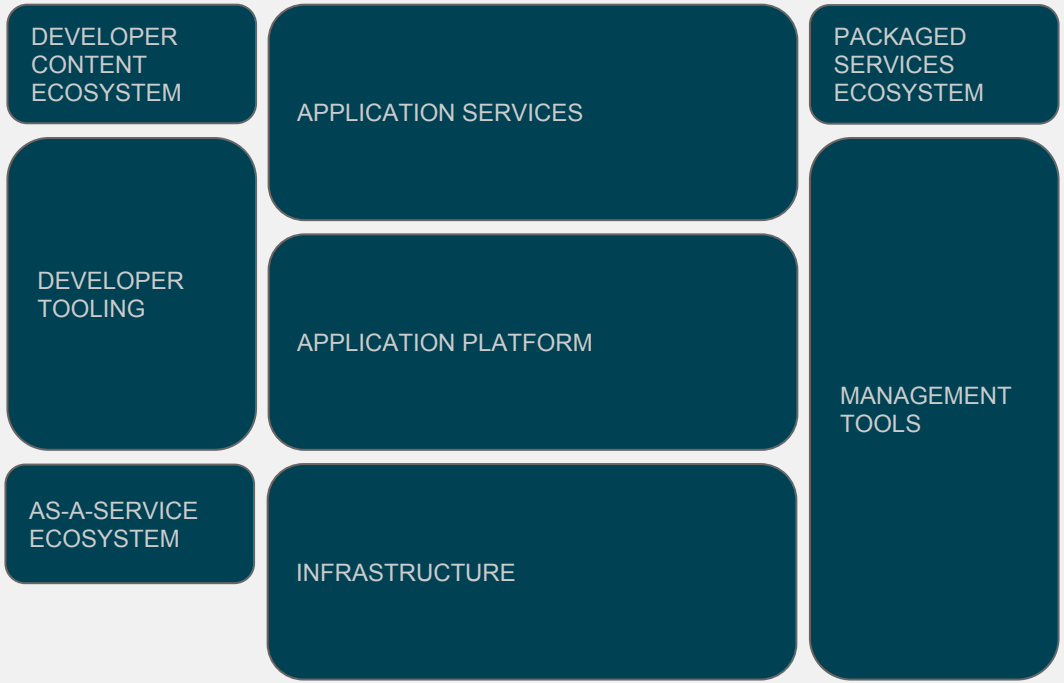
The only constant is change. What benefits developers?

- Massive communities of practice
- Workflows that ease iteration and connect the developers to their applications
- New patterns and tools that dramatically reduce complexity
- The event driven, data aware, fully connected ecosystem

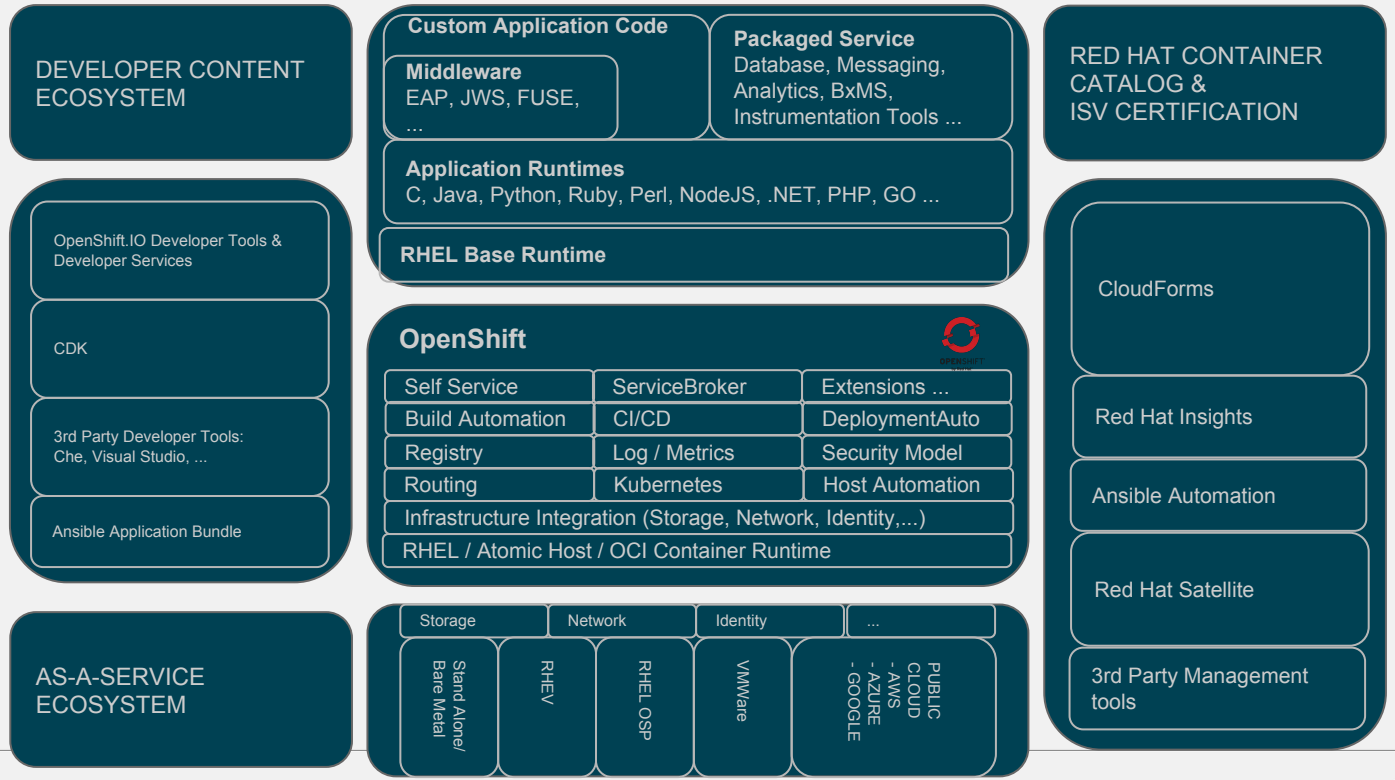
Serverless, FUSE, Kafka, microservices, envoy, Netflix OSS, react, SPIFFE,

The App-Centric Stack

Bringing it all together.



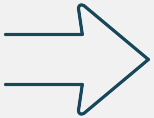
The OpenShift Stack



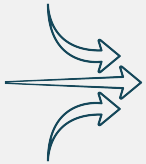
Red Hat Vision for Application Centric IT

Enable Customers & Partners

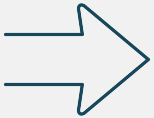
Any Service, Anywhere



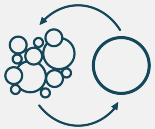
Common abstraction and integration layer



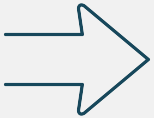
Augment Vertically Integrated Cloud



Commoditize all infrastructure



Empower developers everywhere



Standardize application (development) life cycle

Leverage Red Hat's experience and earned trust from breaking the proprietary Unix and Mainframe lock-in.

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.