The future of data centers

Red Hat’s Office of Technology’s perspective

Stephen Watt
Red Hat Office of Technology
5/2/17
The Red Hat Office of Technology

Chris Wright
Vice President and Chief Technologist

Charter:
To provide Red Hat with long-term technology perspectives around emerging technologies and products, as well as an internal hub to foster and evangelize forward-looking, pre-commercialized innovation.
Who are we?

**Dan Pratl**: Chief of Staff

**Sage Weil**: Storage

**Vincent Batts**: Containers
Who are we?

Mike Bursell: Security
Monty Taylor: CI / CD
Suneel Marthi: ML & AI
Who are we?

Bill Burns: NFV Partner Eng

Andre Fredette: ODL & NFV

Russell Bryant: OpenStack
Who are we?

Miki Kenneth: Research  
Deb Bryant: OSAS  
Steve Watt: Emerging Technologies

... and many others
Containers, Kubernetes
& the Data Center
S/W based innovation is increasing
Historically, an enterprise responding to a threat or opportunity looked a little like this...
But with an innovation deluge, it’s starting to look a lot more like this...

Different Disruptive Technologies

- Innovators: 2%
- Early Adopters: 13.5%
- Early Majority: 34%
- Late Majority: 34%
- Laggards: 16%

Incumbent Technology
But applications are hard to deploy, which makes it costly to experiment and innovate.

A huge step forward for single server applications.

Enter Containers

A way to include environmental configuration and the runtime in a way that is easily packaged and can be rapidly deployed.
And is reshaping our expectations of what we call the OS

Traditional OS (RHEL)
- Repos
  - Yum install RPMs (Additional Packages)
  - Base Packages
  - Kernel

Container OS (RHEL Atomic)
- Registry
- Pull & Run Containers
  - host-extensions
  - Container Lifecycle Manager
  - Kernel
But now we’re seeing a proliferation of container runtimes

So we’re continuing to strategically invest in OCI so that the industry can have a standard interchange format as an abstraction to build on for all of these runtimes
Containers by themselves are great, but most new open source innovation is designed as a distributed system with deployment documentation that is mediocre at best. We needed a solution for clusters.

Enter
With Kubernetes, the Cluster is now the Computer

From many large clusters to just one

Spark Cluster

Master

Worker(s)

Cassandra Cluster

Master

Worker(s)

MySQL

Single Instance

Registry

Master

Worker(s)
This means one can run software defined storage platforms **IN** Kubernetes along with all your other containerized applications.

* [https://github.com/gluster/gluster-kubernetes/](https://github.com/gluster/gluster-kubernetes/)

** [https://github.com/ceph/ceph-docker/tree/master/examples/kubernetes/gce](https://github.com/ceph/ceph-docker/tree/master/examples/kubernetes/gce)
And all these containers need to communicate efficiently using standardized network interfaces

So we’re investing in Kuryr to allow Kubernetes to use Neutron as a CNI plugin if you’re running on OpenStack and also building CNI Plugins for OpenDaylight and OVN as well as contributing to those SDNs directly.
Which is great for on-premise, but Kubernetes still has a few challenges ahead to incentivize applications away from Cloud Provider lock-in

“By 2025, 80% of all workloads will run on the public cloud”

Citigroup analyst report
Amazon’s quick time-to-solution via their Service Catalog has wide appeal but creates a natural lock-in
There is an opportunity to create a compelling alternative by:

- Creating a great Application Portability experience across Cloud Providers (Investing in Kubernetes Federation)

- Exposing a rich ecosystem of services to Kubernetes Applications (Investing in Kubernetes Service Broker/Service Catalog)

  - S105156 - Open Service Broker API: Enabling microservices in the enterprise (Wednesday, May 3, 4:30 PM - 5:15 PM, Room 102A)
Exploring Application Portability with Kubernetes Federation & Pac-Man
In order to explore cross-cloud application portability we decided that we first need to build a reference stateful web application.

It turns out that Jim and I love Pac-Man and that we could add state to the PacMan application by persisting the high scores in MongoDB.
We then modified the Pac-Man high scores to detect and reflect the cloud provider and zone the game was running in and then containerized Pac-Man so we could deploy it along with containerized Mongo in Kubernetes.

We then built 3 separate Kubernetes Clusters in AWS, GCE and Azure respectively and plugged them into the Kubernetes Federation Control Plane and then used that to deploy the Application across all the cloud providers.
Federation Control Plane

pacman.ifontlabs.com

Amazon Web Services

Google Cloud Platform

Microsoft Azure
Federation Control Plane

pacman.ifontlabs.com

amazon web services

Google Cloud Platform

Microsoft Azure
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHatNews
LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.