Why Reinvent the Wheel?

Using Murano, Heat, Container Clustering and Ceilometer to provide Auto-Scaling and Enforce Self--Healing Best Practices in Applications

Bruce Basil Mathews
Serg Melikyan
Agenda

- Once Upon a Time
- Virtualization
- Automation
- The Inflection Point
- Redesign Application & Automation
- Redeploy Application
- Getting Containers
- Auto-Scaling
- Self-Healing
High Availability and Disaster Recovery
Virtualization and Automation

Virtualization

IT In Control!

The End User Perspective:
- Not as High Service Levels
- Takes Just as Long
- Requires LOTS of Interaction between End User and IT
- Recovery taken Care of in Hardware not Software

The IT Perspective:
- Better Utilization Levels
- Management in a Pane of Glass
- Added Licensing Costs for Each Feature Used

Automation

IT Give it to you FASTER!
The Inflection Point

Horizon UI

Fuel UI
The New IT Backend..

**AZ1**
- Host Agg 1: Linux
- Fuel
- MongoDB 1
- Network 2

**AZ2**
- Host Agg 2: Windows
- Controller 1
- MongoDB 2

**AZ3**
- Controller 2
- MongoDB 3

**AZ4**
- Controller 3
- Network 1

- Ceph-MON 1
- Ceph-MON 2
- Ceph-MON 3

**Storage**
- Cinder Tier 1 Storage (SSD)
- Cinder Tier 2 Storage (SAS+SSD)
- Cinder Tier 3 Storage (SATA)
The End User Application Use of the New Backend..
End User and IT Perspective on this Change

The End Users Perspective:

- My VMs are Self-Managed, Planning my own Backups and Restores and Next Gen Requirements
- I am More in Control of my Needs
- I Plan for Failure in Software and Placement versus Relying on Hardware Resiliency
- I Consume and Relieve Resources as Needed

The IT Ops Perspective:

- I Plan for Future Growth
- I Enable Future Capabilities
- I Monitor and Manage versus dragging cables and racking components
- I Spend More Time on Planning versus Troubleshooting
Why Run Containers on OpenStack?

Container benefits developers:

- Portability
- Immutability
- Small footprint
- Ease of update (layers)
- Pre-packaged containers
- Light-weight (create, destroy)
- Excellent for micro-services
- Community & enterprise repositories

OpenStack benefits IT/OPS:

- Unified VM, bare-metal, container strategy
- Virtual storage & networking
- Multitenant security & isolation
- Monitoring & management
- Elastic infrastructure
- Push-button deployment of containers & orchestration SW
Running Containers on OpenStack Compute Nodes

If security is more important run containers in VMs.

If density/perf. is more important run containers on bare metal with Ironic.
## Container Frameworks - A Few of the Choices...

<table>
<thead>
<tr>
<th>Framework</th>
<th>What is it?</th>
<th>Status</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| Kubernetes    | Container scheduler & orchestration framework. Works on VMs or bare-metal. Created & open-source by Google in 2015. | Multiple companies contributing, offerings from: RedHat OpenShift, CoreOS Tectonic, Kismatic etc. | - Strong community  
- GCE public cloud option  
- Collaborative with OpenStack and Mirantis  
- Service discovery | - Relatively new; feature incomplete |
| Mesos         | Sophisticated scheduler for distributed systems, created by UC Berkeley AMPLab ~2011, popularized by Twitter and now an Apache project. | Main company behind Mesos is Mesosphere. Sophisticated but heavy scheduler. Orchestrators sit on top. | - Powerful scheduler  
- Proven at large scale  
- Mature  
- Marathon can do some orchestration | - Apps have to port to Mesos  
- Heavy  
- Not native for containers |
| Docker Swarm & Compose | Container scheduling (Swarm) and orchestration (Compose). Created by Docker in 2014. | Still early. Unclear how much traction these projects are getting. | - Uses docker native APIs | - Very new: feature incomplete  
- Not much traction |
What these Changes Mean to You (IT AND End User!)

Services moved and rearranged as needed AND restart almost instantly!

VMs become more static and stable in Operation within the Kubernetes Cluster!
“How to make my container infrastructure reliable & scalable?”
Application Catalog for Openstack

Mission:
Introduce an application catalog to OpenStack, enabling application developers and cloud administrators to publish various cloud-ready applications in a browsable categorized catalog.
Containers Service for Openstack

Mission:
Provide a set of services for management of application containers in a multitenant cloud environment.
Getting Containers

Deploy & Scale by: Murano or Magnum

- REST API for Deployment and Scaling
- Choice of Container Orchestrators
  - Kubernetes
  - Mesos
  - Swarm
- OpenStack-native API and Orchestrator-native API for container scheduling
Murano & Kubernetes

- Just Another Application in the Catalog
- Use Kubernetes v1.0.6
- Scalable & Extensible
- Orchestrator-native API for container management
- API & UI for the Provisioning & Scaling
Telemetry Service for Openstack

Mission:
Reliably collect measurements of the utilization of the physical and virtual resources comprising deployed clouds, persist these data for subsequent retrieval and analysis, and trigger actions when defined criteria are met.
Auto-Scaling

Monitor & Scale: Ceilometer or Zabbix

- Gathering metrics from the VM
- Events on metrics deviation

Concerns?

- Authentication for Magnum & Murano API calls for scaling. Ceilometer plugins?
Self-Healing in Murano

Capabilities provided by Murano:

- Application workflow callable by API
- Imperative workflow language

Workflow may be used for:

- Removing failed node based on monitoring feedback
- Create new node and join to the cluster to replace failed
Platform for Application Management

APIs

Out of Cloud Apps Integration

RBAC & Audit

Monitoring

Create/update

Operate

Consume

Publish

Continuous Delivery

Microservices Tools as a Service

Components

PaaSes

Frameworks

Containers

DBaaS

LBaaS

AMaaS

VPNaaS

FWaaS
Q&A
Do you want more?

Leveraging Kubernetes to Scale Containers in Hybrid Multi-Cloud Clusters
Thursday, October 29 • 11:45am - 12:00pm
Room: Wakakshiba

Magnum: Getting Started for Developers and Sysadmins
Thursday, October 29 • 2:40pm - 3:20pm
Room: Ohka
Thank you

for your time