

SHIFTING THE RACK

Greg Swift



Overview

- Who Rackspace is and what we do
- What we need
- How we are getting there
- What we learned



WHO IS RACKSPACE

Who is Rackspace

- Managed Services
- Fanatical Support



What does that mean?

- Managed Dedicated Servers (Bare Metal)
- Public Cloud offering via OpenStack
- *Four* Private cloud offerings
 - Rackspace OpenStack
 - RedHat Deployed OpenStack
 - VMware
 - Azure
- *Three* Managed Public Cloud services
 - AWS
 - GCP
 - Azure
- Managed Security, Datastores, Email, and more!



No really...

- Hundreds of small IT departments
- Pick a Rackspace product/platform, any one
- Pick a best practice, any best practice
- Pick a toolset, any toolset
- Go as fast as you want
- Be *Agile*



With Great Power comes...

- Changing teams can be like changing companies
- For some teams, context switching their own stacks is like changing companies
- Compliance reporting is an adventure
- Even with a single wiki for most docs, tribal knowledge rules



WHAT WE NEED

All is not lost...

- Best practices becoming standard practice
- Lots of interest in not managing entire stack
- Openness to a primary tool, even if its only a 90%'er
- Realizing that we going further together is more productive than faster apart.



Goals

- Developers are the SME for this application. Let them be.
- Get Operations out of the delivery path. Let them run the platform and supporting services.
- Simplify management of fleet.
- Maintain compliance objectives.
- Actually move faster.



HOW WE ARE GETTING THERE

IaaS

- Utilize one of our largest IaaS products
- Try to stay ahead of demand



First pass

- Inhouse built PaaS written in Ruby on top of Mesos/Marathon
- Heroku like and supported buildpacks
- Easily integrated GitHub Deployment API and Slack
- One project suite recently passed 3000 deployments in a year
- Development team turn over and drop off of support
- Never solved
 - logging
 - persistent storage
 - UI or cli (curl all the things)



Second pass

- Deployed OpenShift Origin 1.4 and upgraded to 1.5
- Started with GlusterFS but ran into issues deploying Aggregated logging and Cassandra on it. Switched to Ceph.
- Went production for non-critical workloads after 3 months
- Jenkins became a top consumer quickly



Successes

- New applications have gone to production without dedicated ops involvement in a fraction of the time as before
- QE was able to migrate their performance test harness with throughput of 15 million requests from our internal OpenStack solution in days.



Current Footprint

- >300 Projects
- ~150 Sandbox
- ~40 CICD Projects (Jenkins)
- 6 Production
- 1 Customer Facing Production



LESSONS LEARNED

LB != router

- Make sure your *openshift_hosted_router_replicas* count matches the number of schedulable nodes in that region.
 - Default region is infra
 - Set by *openshift_router_selector*
- Lb nodes run non-containerized HAProxy for the main openshift api on port 8443
- You can have the same hosts perform both actions, but they are not the same
- Do over: Separate nodes named and labeled to act as routers.



Quotas

- Start tight by default, but pay attention and shift as necessary
- Use resource limiting (cpu and memory), but you have to update all the default templates and quick starts since they dont include requests
- Do over: Implement resource limits as part of main quotas at start.



Resources (*openshift_node_kubelet_args*)

```
# image-gc-high-threshold: Trigger garbage collection at 80% disk utilization
# image-gc-low-threshold: Try to maintain 65% available disk through garbage collection
# The next 2 come from this article:
# https://docs.openshift.org/latest/admin_guide/out_of_resource_handling.html#out-of-resource-hard-eviction-thresholds
# https://docs.openshift.org/latest/admin_guide/out_of_resource_handling.html#out-of-resource-allocatable
# TODO: eviction-hard: memory.available<500Mi - if we are running out of memory start heavy handed eviction
# system-reserved: memory=1.5Gi - maintain this much capacity for system use
# To enforce the system reserved it appears you have to enable enforcement with cgroups
# https://docs.openshift.org/1.5/admin_guide/allocating_node_resources.html#node-enforcement
# cgroup-driver: systemd
# We thought we needed these but they break things:
# cgroups-per-qos: true
# enforce-node-allocatable: "pods" ( must be at this time )
openshift_node_kubelet_args={'pods-per-core': ['10'], 'image-gc-high-threshold': ['80'], 'image-gc-low-threshold': ['65'], 'system-reserved': ['memory=1.5Gi'], 'cgroup-driver': ['systemd']}
```



Roadshow

- OpenShift community has an amazing resource that Red Hat has taken on the road... the OpenShift Roadshow.
- <https://github.com/osevg/workshopper-content>
- You can run it yourself

```
oc new-project roadshow
oc new-app osevg/workshopper -e \
  'WORKSHOPS_URLS=https://github.com/osevg/workshopper-workshops/blob/master/roadshow.yml'
oc expose workshopper
```



Helm

- Development groups using Helm/Tiller
- Tracking upstream support for multi-tenancy



Community

- OpenShift Commons Slack
- Openshift-ansible on gitter.im
- StackOverflow tag 'openshift-origin'



Random notes

- Pay attention to SDN networks
openshift_master_portal_net
osm_cluster_network_cidr
- Track the git hash of openshift-ansible you deploy with
- Use ansible OC module to handle post deploy changes
- Follow the work to use Helm as



Thank You

Greg Swift

greg.swift@rackspace.com

gregswift and rackergs on github

xaeth on twitter

