



# Scaling OSBS

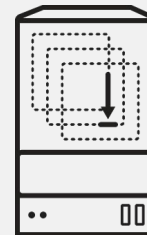
A Hybrid Cloud Story

Michael Bonnet

Technical Lead, DevOps Development, Products and Technologies

January 18, 2017

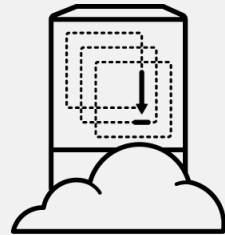
# The OpenShift Build System



**Custom image** that runs in OpenShift to build layered images.

- Running on a **single-node** OpenShift “cluster”.
- Communicates **directly** with the Docker daemon.
- Tightly **integrated** with a Koji plugin.
- Significant **bottleneck** in the process of building and releasing containers.

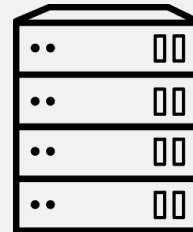
# Advantages of OpenShift Dedicated



OpenShift Dedicated provides a **managed cluster**, hosted in AWS, where custom images can be deployed.

- Avoids the **complexity** of running your own multi-node OpenShift cluster.
- Allows for easily **scaling up** the resources in the cluster.
- OpenShift upgrades are handled by the operations team, with **zero downtime**.

# Making OSBS Multi-Node Capable



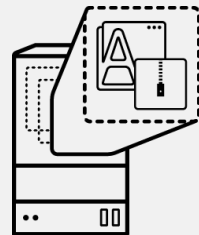
OSBS had been run internally on a single node, and that assumption was **built in** to the tools.

- Node had internal **NFS** volumes mounted.
- Images were pulled from the node into Koji over **HTTP**, requiring httpd to be running.

OSBS was redesigned to be a Koji **Content Generator**.

- Pushes images (and associated metadata) into Koji using a **new API**.
- No longer requires the Koji plugin to know the details of the OSBS **environment**.

# Deployment



The OpenShift Dedicated cluster had to be **customized** to enable OSBS operation.

- Connected to the Red Hat network using the existing **VPN** infrastructure.
  - Connections were only allowed to specific internal **IP addresses**, for security.
- Internal **authentication** mechanism (Kerberos) was not available in cluster.
  - Used Google **OpenID** Connect.
  - Created an OpenShift **service account** for Koji to launch builds in OSBS.
- Custom **configuration** of the cluster was required.
  - Ability to pull/push to an internal **image registry**.
  - Use of **privileged containers** (required by atomic-reactor).
- Production cluster uses identical configuration as the test cluster, but **more nodes**.
  - Building images is **resource-intensive**.

# Things We Learned



- OpenShift Dedicated imposes **constraints**.
  - Code changes to OSBS have made it more **portable** across OpenShift instances.
  - Streamlining the workflow and reducing unnecessary copies also **improved performance**.
- IP address **conflicts** between OpenShift Dedicated and the Red Hat network.
  - Need to be aware of **address ranges** when requesting the cluster.
- Heavy load can cause a node to get **evicted** from the cluster.
  - Unable to change the node eviction timeout, so **concurrent build** limit is lower.
- Running in both internal and cloud environments improves **resilience**.
  - Problems in one **environment** will not affect the other.

# Questions?

(Thanks to Tim Waugh who provided the content for this presentation.)



# THANK YOU



[plus.google.com/+RedHat](https://plus.google.com/+RedHat)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[twitter.com/RedHatNews](https://twitter.com/RedHatNews)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)



# Links

- OpenShift Dedicated: <https://www.openshift.com/dedicated/>
- OSBS: <https://github.com/projectatomic/osbs-client>
- Koji: <https://pagure.io/koji>
  - Content Generators: [https://docs.pagure.org/koji/content\\_generators/](https://docs.pagure.org/koji/content_generators/)
- Google OpenID Connect:  
<https://developers.google.com/identity/protocols/OpenIDConnect>

# OSBS Build Workflow

