Ubiquitous Edge Platform

Lincoln Bryant
Rob Gardner

OSG All Hands Meeting, 15 March 2016

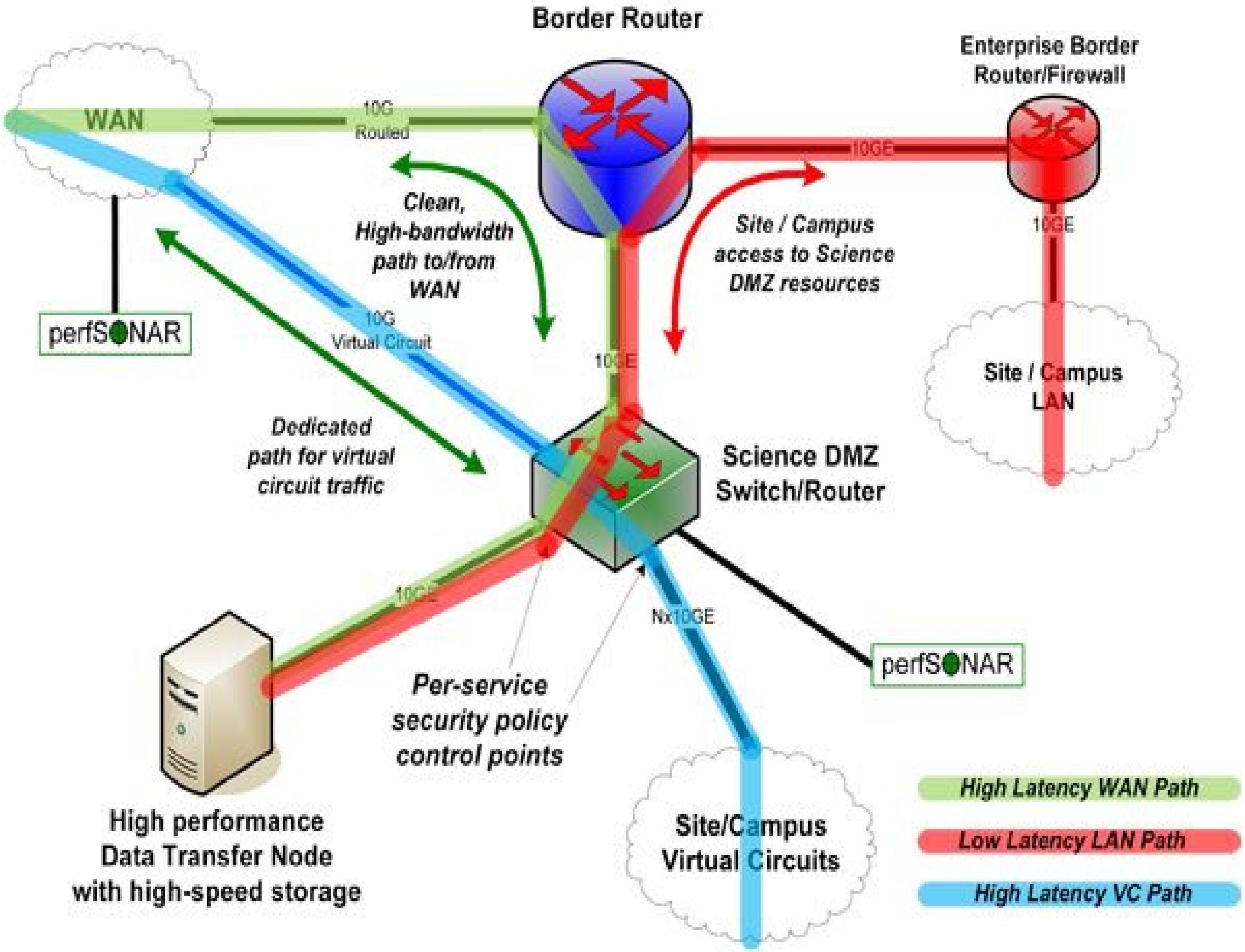
Ubiquitous & Easy “CI Substrate”

- Pioneer a new phase of advanced cyberinfrastructure deployment, allowing sites to flexibly evolve and sustain both on-premise and commercial cloud-based infrastructure
- Hosted services, such as CEs, data caches, squid, etc., could be centrally deployed onto “CI substrates” within a trusted CI zones and remotely operated, upgraded, and optimized for performance
- Extend to shared, opportunistic university clusters and cloud resources

Distributed Virtualized Data Centers

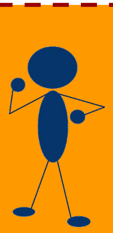
- Reduce IT footprint and ops burden
  - Centralize deployment & ops; reduce local admin cost
- Explore virtualized data center frameworks
  - E.g. container management over bare metal or VMs
- “Blue sky” goal
  - Establish a “trusted pattern” for a “CI substrate” on sites
  - Create distributed virtualized data center(s) overlaying the fabric substrate

Canonical SciDMZ

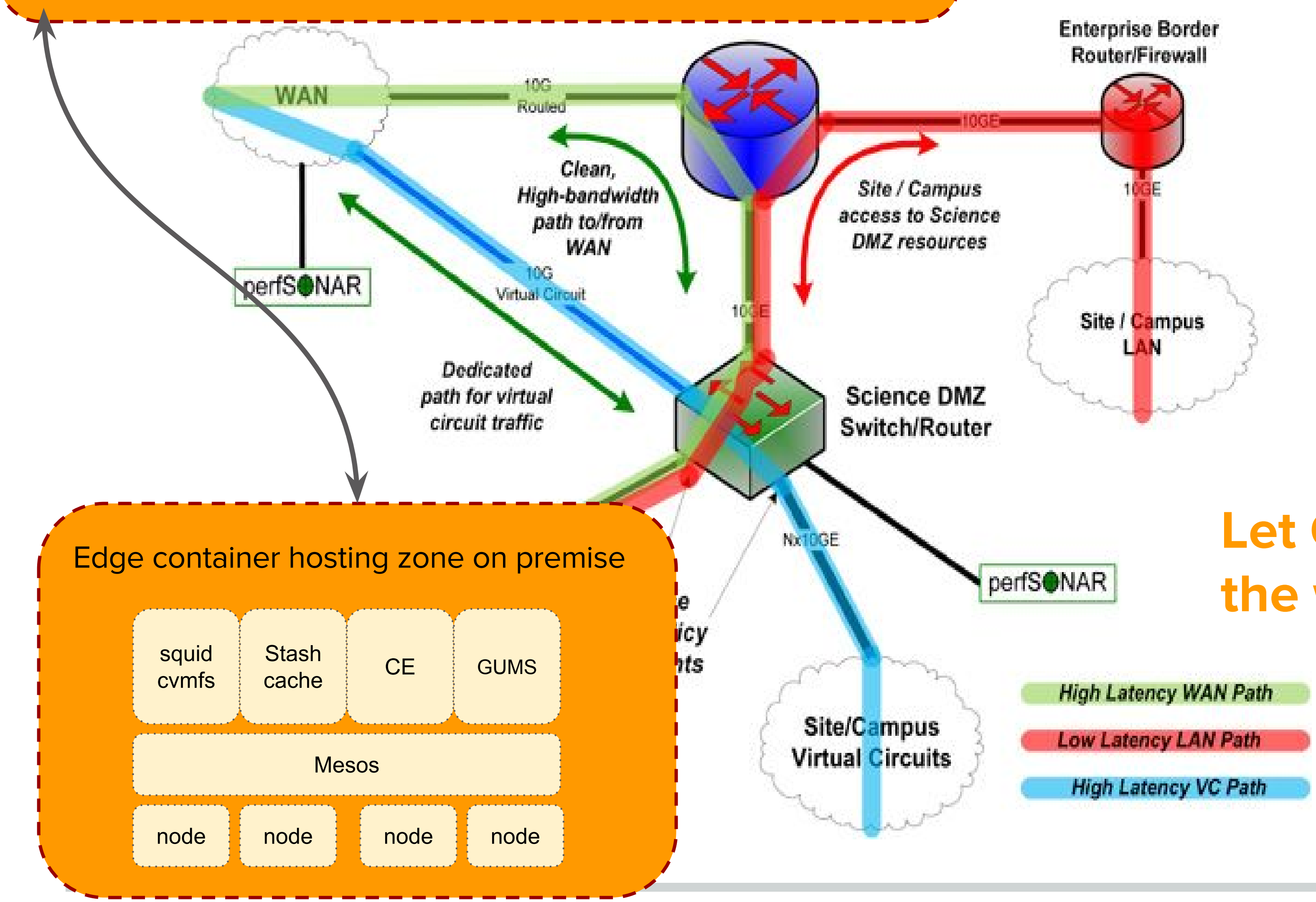

Border Router
Enterprise Border Router/Firewall
WAN
10G Routed
10GE
Clean, High-bandwidth path to/from WAN
Site / Campus access to Science DMZ resources
10GE
perfSONAR
10G Virtual Circuit
10GE
Site / Campus LAN
Dedicated path for virtual circuit traffic
Science DMZ Switch/Router
10GE
Nx10GE
perfSONAR
Per-service security policy control points
High performance Data Transfer Node with high-speed storage
Site/Campus Virtual Circuits
High Latency WAN Path
Low Latency LAN Path
High Latency VC Path

CI Connect central ops console:

```
$ slate install osg-squid.3.3 --sites UAB Wellesley Wyo
```



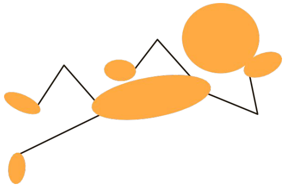
SuperDMZ



Edge container hosting zone on premise

squid cvmfs	Stash cache	CE	GUMS
Mesos			
node	node	node	node

Let OSG do the work!



Deploying research software at the edge

Open Science Grid

perfSONAR

XRootD

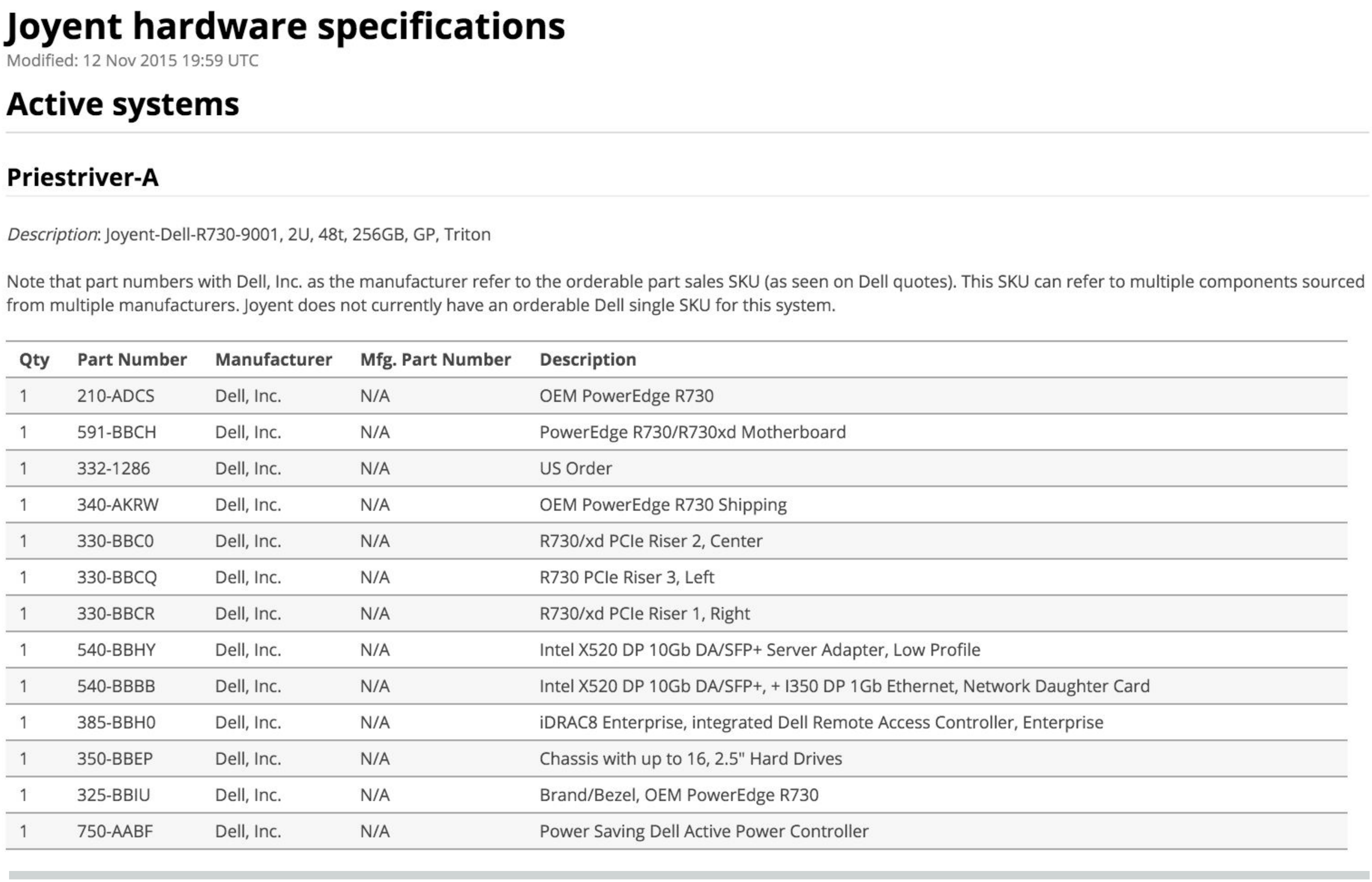
globus

Your favorite project here!

Hardware

- Produce reference specification for supportability reasons
  - No more than 2-3 vendor options.
- Hybrid cloud providers like Joyent have done a really good job in this space. Something similar to:
  - https://docs.joyent.com/private-cloud/hardware/specs

Hardware



Joyent hardware specifications

Modified: 12 Nov 2015 19:59 UTC

Active systems

Priestriver-A

Description: Joyent-Dell-R730-9001, 2U, 48t, 256GB, GP, Triton

Note that part numbers with Dell, Inc. as the manufacturer refer to the orderable part sales SKU (as seen on Dell quotes). This SKU can refer to multiple components sourced from multiple manufacturers. Joyent does not currently have an orderable Dell single SKU for this system.

Qty	Part Number	Manufacturer	Mfg. Part Number	Description
1	210-ADCS	Dell, Inc.	N/A	OEM PowerEdge R730
1	591-BBCH	Dell, Inc.	N/A	PowerEdge R730/R730xd Motherboard
1	332-1286	Dell, Inc.	N/A	US Order
1	340-AKRW	Dell, Inc.	N/A	OEM PowerEdge R730 Shipping
1	330-BBC0	Dell, Inc.	N/A	R730/xd PCIe Riser 2, Center
1	330-BBCQ	Dell, Inc.	N/A	R730 PCIe Riser 3, Left
1	330-BBCR	Dell, Inc.	N/A	R730/xd PCIe Riser 1, Right
1	540-BBHY	Dell, Inc.	N/A	Intel X520 DP 10Gb DA/SFP+ Server Adapter, Low Profile
1	540-BBBB	Dell, Inc.	N/A	Intel X520 DP 10Gb DA/SFP+, + I350 DP 1Gb Ethernet, Network Daughter Card
1	385-BBH0	Dell, Inc.	N/A	iDRAC8 Enterprise, integrated Dell Remote Access Controller, Enterprise
1	350-BBEP	Dell, Inc.	N/A	Chassis with up to 16, 2.5" Hard Drives
1	325-BBIU	Dell, Inc.	N/A	Brand/Bezel, OEM PowerEdge R730
1	750-AABF	Dell, Inc.	N/A	Power Saving Dell Active Power Controller

Operating system

- Many choices to evaluate in this area
- Traditional distributions:
  - Flavors of RHEL, Debian, Ubuntu, etc
- Upcoming projects building around containers:
  - CoreOS, Boot2Docker, RancherOS, Project Atomic
- Exotic alternatives:
  - SmartOS (Solaris-based, emulating Linux kernel ABI)

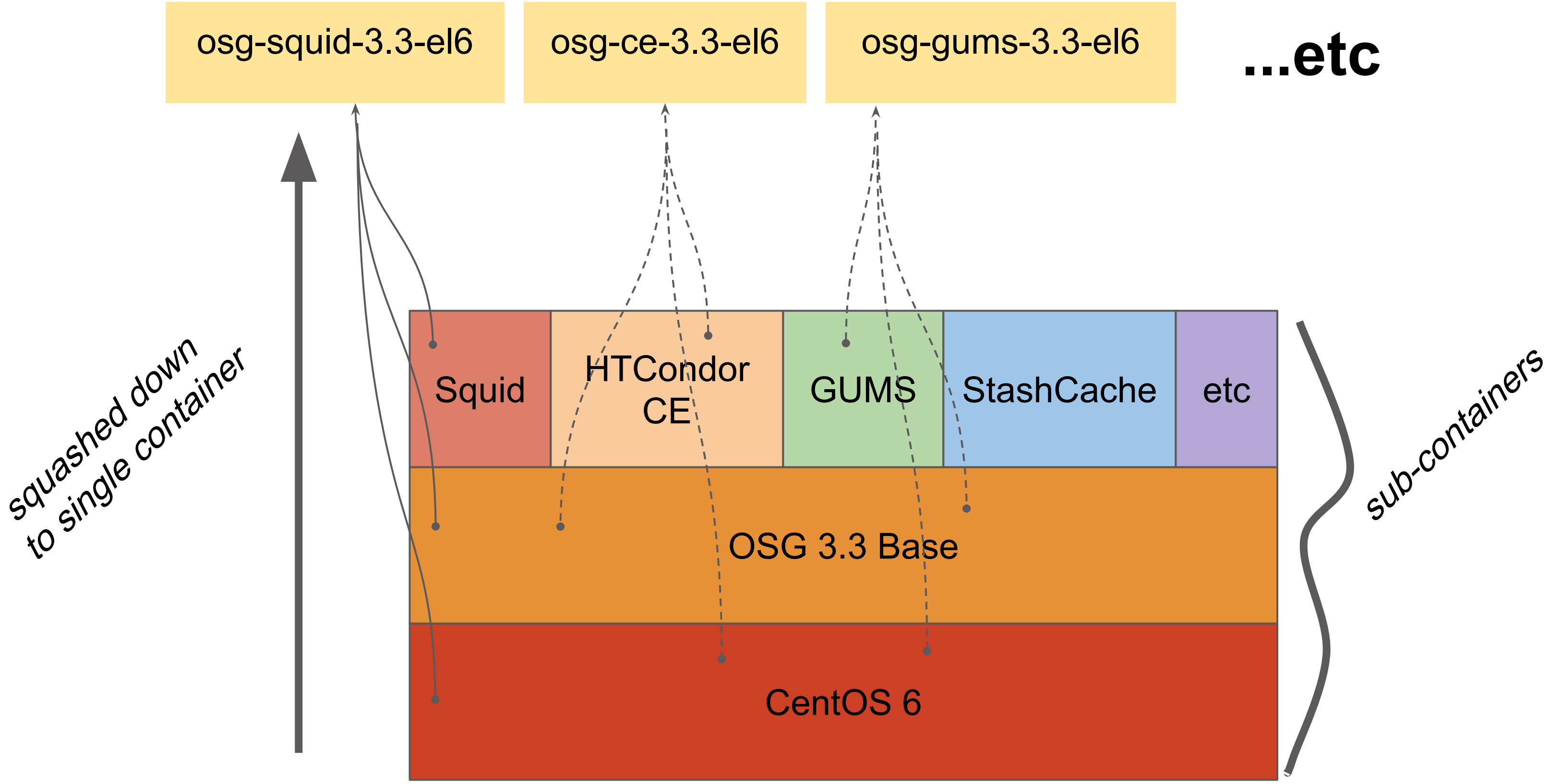
Software

- Microservices-y architecture
  - Follow the Docker model of 1 application per container
- Service discovery and configuration tools
  - Consul, etcd, and others.
- Scheduling / cluster management
  - Kubernetes, Docker Swarm, Fleet, Mesos
  - (HTCondor?)

Software

- Dockerized applications created, vetted, maintained by central operations team.
  - Pushed by operators down to subscribed sites
  - Or, depending on use case, pulled by local site admins without interaction with central.
- Built-in monitoring/analytics
  - Every service should get its own set of applicable collectors
  - Leverage our existing monitoring scripts and expertise.

Containerizing Services



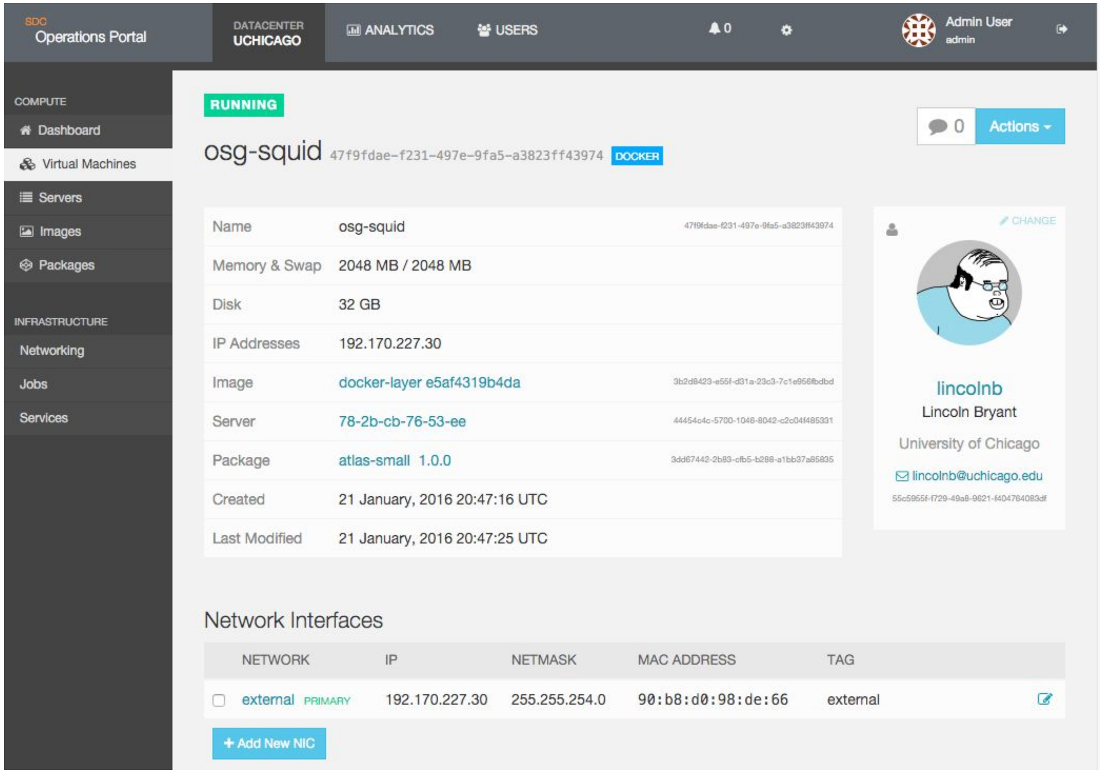
Example: Frontier Squid - Dockerfile

```
FROM lincolnbryant/osg-base-3.3-el6
MAINTAINER Lincoln Bryant <lincolnb@uchicago.edu>
# See https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallFrontierSquid

RUN yum install -y frontier-squid initscripts

VOLUME ["/var/cache/squid"]

COPY customize.sh /etc/squid/customize.sh
RUN chown squid: /etc/squid/customize.sh && chmod +x /etc/squid/customize.sh

EXPOSE 3128 3401

CMD /sbin/runuser -s /bin/bash squid /usr/sbin/fn-local-squid.sh start && tail -f
/var/log/squid/*.log
```

Example: Frontier Squid - Launching

- The container can be launched on another machine, or via Docker’s remote API to a cloud resource

```
$ docker run -p 3128:3128/tcp -p 3401:
3401/udp -ti -e IP_BLOCKS="10.0.0.0/8
192.170.226.0/23" -e MEMORY_MB=2048 -e
CACHE_MB=32768 lincolnbryant/osg-squid-3.3-
el6

Generating /etc/squid/squid.conf

Initializing Cache...

2016/01/21 20:45:07| Creating Swap
Directories

Starting 1 Frontier Squid...

done


...
```



osg-squid 47f9fdae-f231-497e-9fa5-a3823ff43974

Name	osg-squid	47f9fdae-f231-497e-9fa5-a3823ff43974
Memory & Swap	2048 MB / 2048 MB	
Disk	32 GB	
IP Addresses	192.170.227.30	
Image	docker-layer e5af4319b4da	3b2d8423-e55f-d31a-23c3-7c1e956fbdbd
Server	78-2b-cb-76-53-ee	44454c4c-5700-1046-8042-c2c04f485331
Package	atlas-small 1.0.0	3dd67442-2b83-cfb5-b288-a1bb37a85835
Created	21 January, 2016 20:47:16 UTC	
Last Modified	21 January, 2016 20:47:25 UTC	

lincolnb
Lincoln Bryant
University of Chicago
lincolnb@uchicago.edu
55c5965f-f729-49a8-9621-f404764083df

Network Interfaces

	NETWORK	IP	NETMASK	MAC ADDRESS	TAG
□	external PRIMARY	192.170.227.30	255.255.254.0	90:b8:d0:98:de:66	external

Benefits for OSG

- Could potentially deploy CEs, SEs, caching proxies, etc all within “the box”.
    - Best known versions and configurations get automatically pushed to downstream
    - Updates should be atomic, so rollbacks are easy.
- Containerization effort putting more eyes on existing OSG Documentation and builds
    - Example: Patches submitted for GUMS to build on EL7
        - https://github.com/opensciencegrid/gums/pull/27

Thoughts on automation

- Is it possible for me to stand up, then destroy an entire OSG site in an automated way?
    - Site needs to be registered with OIM, certificates issued, etc.
- Many points where human interaction is currently needed.
- Need to separate approvals (necessarily requiring human interaction) from configuration/setup

Security concerns

- Who has root on the machine?
- Can trusted users allocate resources and start containers remotely?
- Is Docker secure enough to be used? Many claims of a busted security model.
  - User namespaces and unprivileged containers seem to be semi-working in new kernels? (Affects OS choice!)
- Ultimately: What is the correct privilege separation between owner and operator?

Other considerations

- What does the networking configuration look like?
    - Do standard installers (Anaconda) cover the majority of network configurations for initial bootstrapping?
    - Private control channel / VPN?
    - Require public IP(s)?
- Can we use this platform as a testbed for things like SDN?
- What does it look like when we have multiple nodes per site?

Summary

- Platform for “edge” services on Science DMZs with well-defined reference hardware.
- Container-based applications, maintained by a central team
- Built-in service discovery, configuration, and monitoring
- Flexible, adaptable to the needs of other projects.

Thank you!

Questions?