National Synchrotron Light Source II

Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

Virtualization at NSLS-II

Dr. Marcus D. Hanwell

Group Leader, AST Lead, Software Architect

NSLS2-SDCC Workshop, BNL

16 September, 2022

Early History of Virtualization

- Two main flavors—virtual machines and containers
  - 1999 gave us bullet time and VMWare 1.0!
  - 2000 FreeBSD 4 with the first version of chroot jails!
- Then came widespread 64 bit and new CPU architectures
  - 2003 AMD64 later known as x86_64
  - 2003 Xen open source x86 hypervisor
- Pivoting to servers and proto-cloud/hosted services
  - 2006 first VMWare Server release

More Recent Virtualization Events

- New players arrive in 2007's virtualization scene
  - Open source kvm released—required hardware support
  - Virtualbox Open Source Edition released under GPL license
- Containers burst onto the scene in 2013
  - Docker was released upon the world and containers were cool
- Container orchestration wars ensued
  - 2016 Docker went native on Windows (always native on Linux)
  - 2019 Windows subsystem for Linux, containers blurred...

Central Infrastructure

- Satellite server for provisioning, configuration of RHEL systems
  - Tower and now Ansible Automation Platform for improved automation
- VMWare clusters centrally and at some beamlines
- Central Lustre file store for most "big data"
  - Accessed via workstations, Globus, SFTP
- Services such as MongoDB, Kafka, Prefect, ...
- Looking at Kubernetes and container use for future facing work

NSLS-II & VMWare

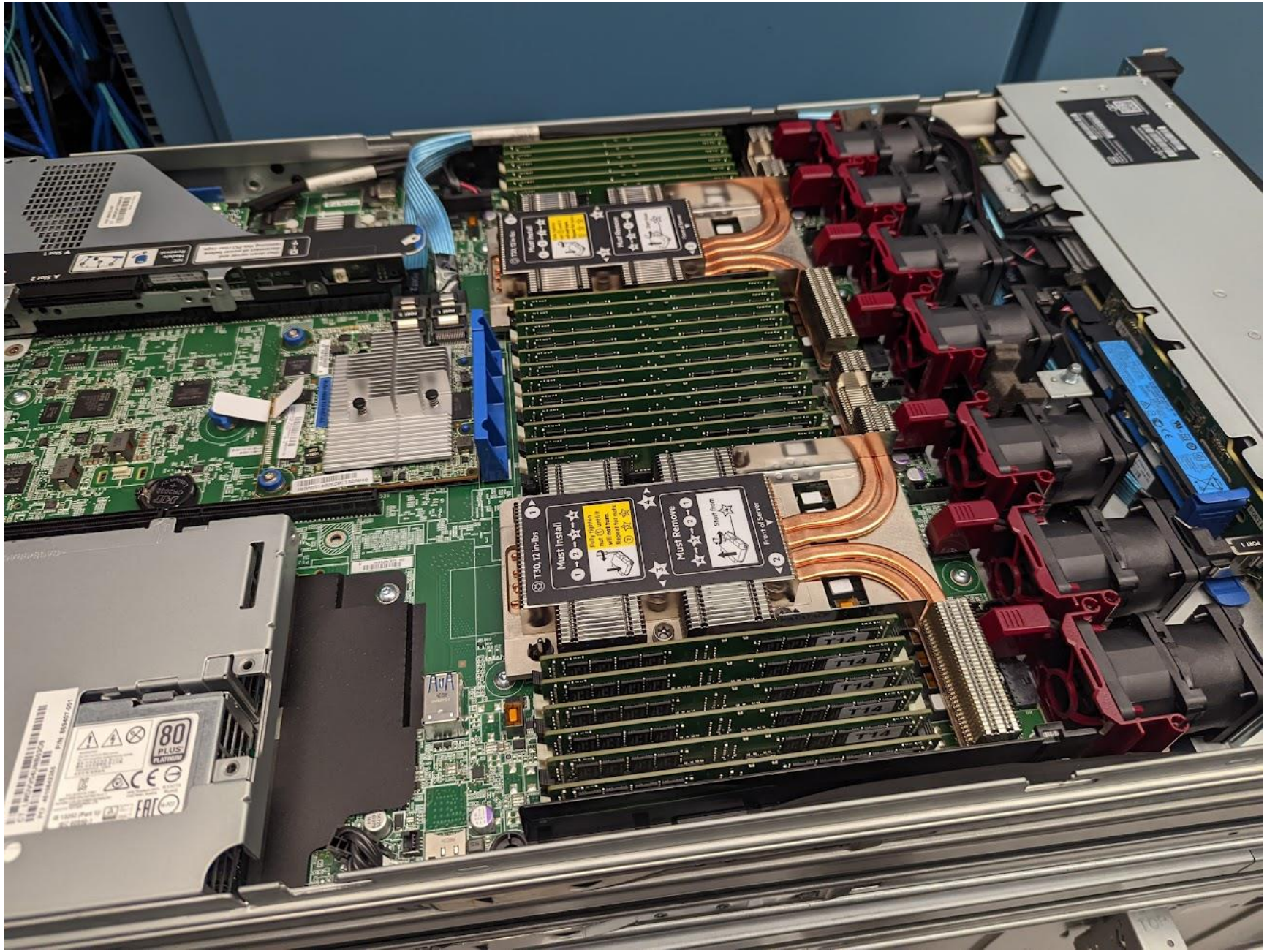
A tour of the NSLS-II VMWare infrastructure

VMWare Deployments at NSLS-II

- Data Center VMWare Server Instances
  - N2SN—9 nodes hosting many of our core services
  - N2SN Data Services—6 nodes for data centric services
  - NSLS2ACC—8 nodes for accelerator services
  - 515—3 nodes at SDCC for new services
- A small 3 node instance on the Campus Network
- LOB Lab 1 hosts a 2 node development and 2 node test cluster
- Beamline VMWare Server at beamlines
  - Mostly two node at this stage, XPD, PDF, SRX, HEX and BMM

Integration with Satellite

- VMWare Server and Satellite talk!
  - Provision virtual machines using satellite
  - Look very similar in both deployment and use
  - Can access the same networks, reuse the same Ansible roles!
- Extends to Tower and now Automation Platform
  - Source their inventory from the same place
  - Offer configuration management, deployment, etc


Must Install
Must Remove
80 PLUS PLATINUM

Increased Fault Tolerance

- Increased fault tolerance for critical services
  - Normal VMs will fail over to a different hypervisor automatically
    - Looks like a reboot of the machine, often before people have time to react
  - Highly available VMs constantly synchronize their memory
    - Hypervisor failing looks like a short pause in communication usually
  - Data center hypervisors using redundant switches and network!
    - Even with complete switch failure they use the other switch and continue
- Partnering with networking to offer more robust solutions

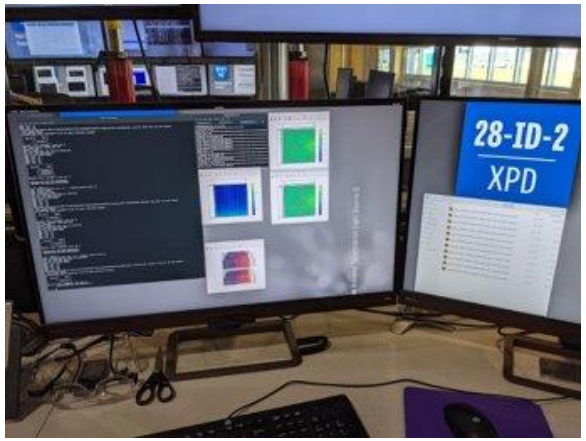
Backups and Utilization

- Ability to backup, snapshot, rollback
  - ITD CommVault taking daily images of VMs
    - Full machine backups to Azure automatically and transparently
  - VMWare can snapshot images live before changes are made
    - Rollback to last "good" version is possible
- Much better utilization of hardware with simpler replacement
  - Size the clusters to the workload that is required
  - Replacing old/faulty nodes is seamless, images migrate around

Development VMs and Windows

- Staff can create development VMs for development
  - Avoid mixing production and development together
  - Usually isolated on the right networks
  - Identical software environments thanks to Satellite, RHEL, etc
- Windows VMs as needed along with other OSes
  - They use virtual networks to connect to the VLANs required
  - Can be isolated, left shutdown until needed, snapshot and restart
  - Many appliances offer VMWare images for quick deployment

Computational View of a Beamline



- Detectors and motors networked, serial or USB
- IOC servers on the "EPICS" VLAN control the beamline
  - Cameras/detectors live on the "CAM" stream images mainly
  - Instruments on an "INST" VLAN for general instruments
- Some specialized servers (some now VMs)
  - GPFS clusters, processing, MongoDB, channel archiving
- Workstations on the "SCI" VLAN used to control beamlines
  - Usually dual-homed with "EPICS" to talk to IOC services
  - Can be accessed remotely with Guacamole or NX

Future Work: Hardware Passthrough

- Can we virtualize specialized machines with PCI/PCI express?
  - Looking at hardware passthrough to Linux/Windows
  - Running vendor drivers, kernel modules, etc
- Offering GPU capabilities for specialized workloads
  - Some experimental nodes in place for experimenting
  - Whole GPU, partial GPU, etc
- Some of this is better suited to containerization...

Containers and Kubernetes

Future facing work

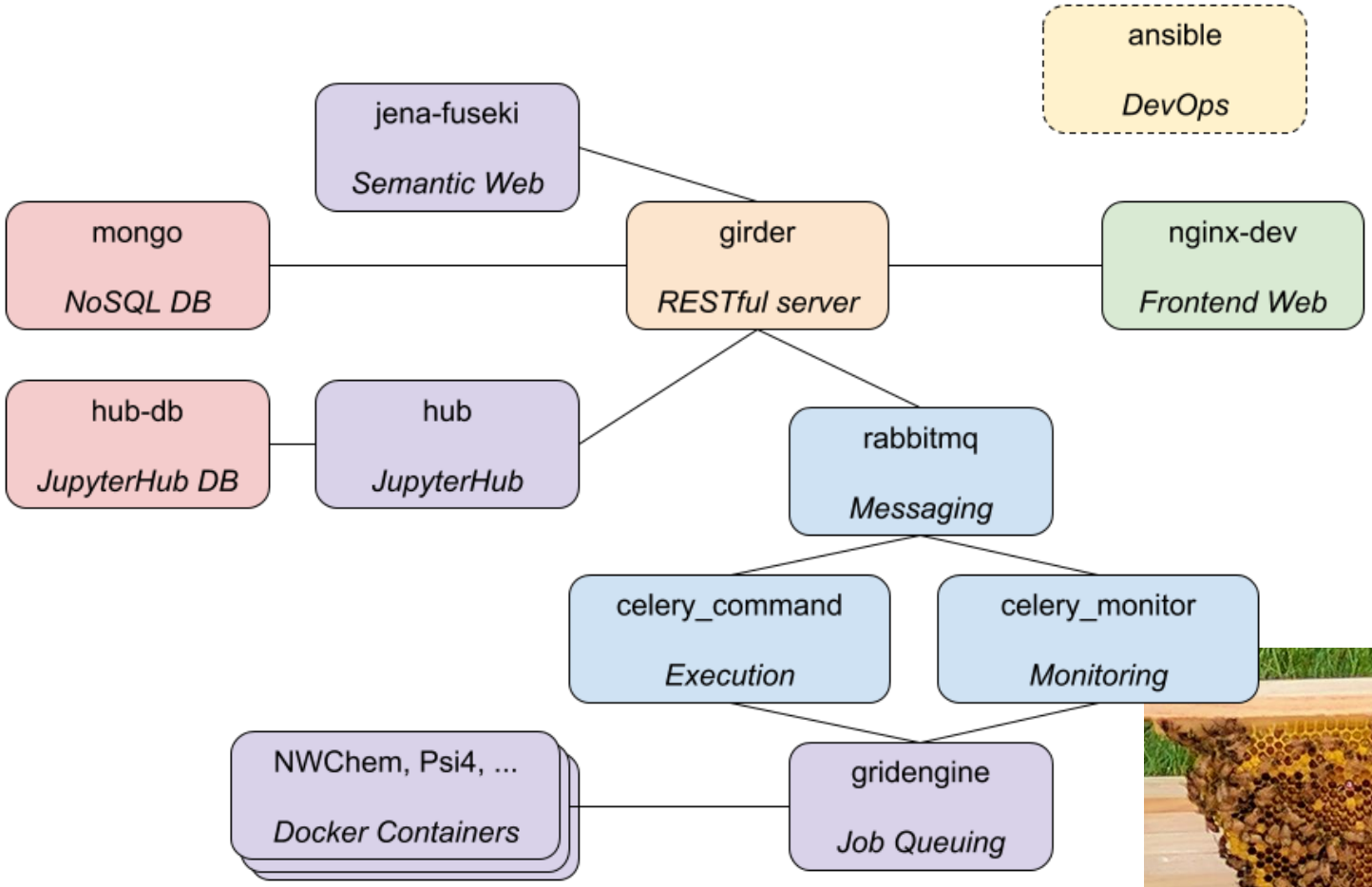
Why Containers

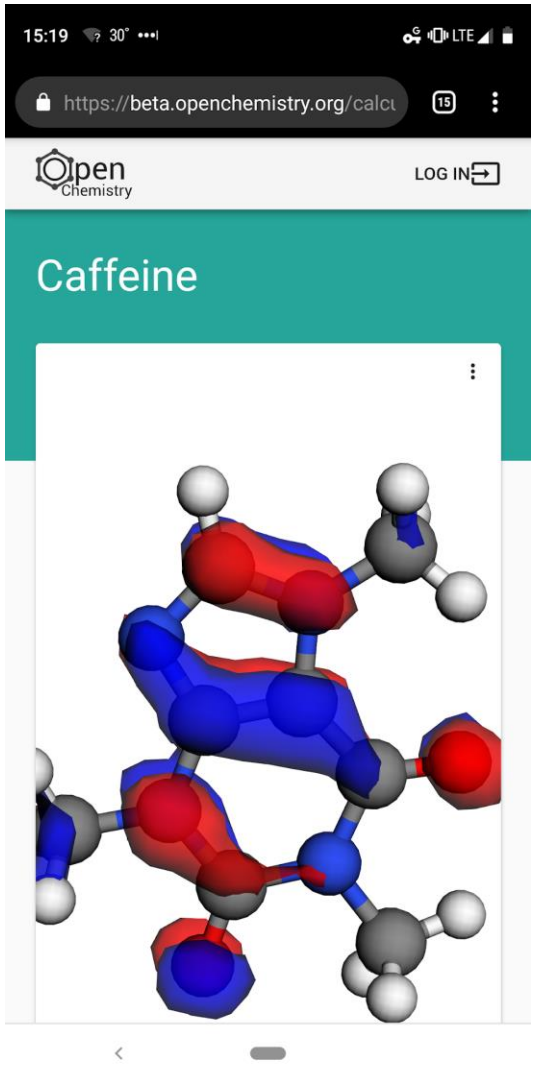
- Virtual machines are "ancient" and hugely wasteful!
  - Virtual kernel, virtual network stack, etc within a hypervizor
  - Whole operating system installed thinking and running like it is real
- Containers are the modern, leaner, more reproducible solution
  - Just enough of the operating system/software environment
  - Require less resources, no "boot up time", quite complete isolation
  - Fully reproducible and isolated container building environment
- Different and require new tools that are less familiar

Kubernetes and NSLS-II

- Deploying some experimental capabilities for testing
- OpenShift, Rancher, and other solutions are out there
- Container builds are version controlled
  - All steps contained within Git for containers
  - Base container, config, changes, layers,
- Kubernetes, ansible and other pieces for orchestration
  - Most of the web now runs on containers
  - Possible to experiment with small things, build up over time

My Open Chemistry Work: Containers





Running Multiple Containers in ~2017

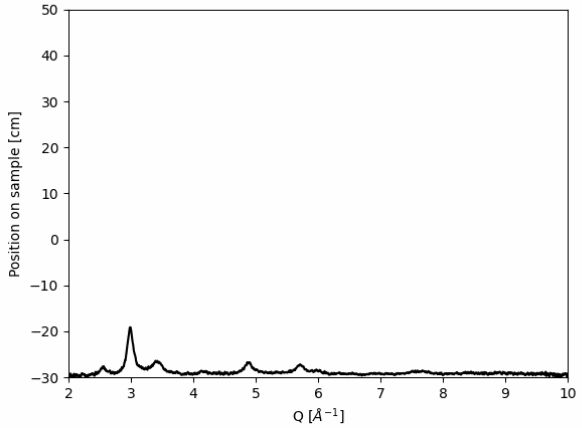
```
(jupyter) marcus@adamantium:~/src/mongochemdeploy master$ ocdeploy pull
Executing: docker-compose -f docker-compose.yml -f ../girder/docker-compose.yml
-f ../jupyterhub/docker-compose.yml -f ../jena/docker-compose.yml pull
WARNING: The OPENCHEMISTRYPY variable is not set. Defaulting to a blank string.
WARNING: The JUPTERLAB_APP_DIR variable is not set. Defaulting to a blank string
.
Pulling jena-fuseki    ...
Pulling gridengine     ... pull complete
Pulling rabbitmq       ... done
Pulling celery_command ... extracting (1.0%)
Pulling celery_monitor ... extracting (1.0%)
Pulling hub-db         ...
Pulling hub            ...
Pulling girder         ... extracting (100.0%)
Pulling nginx-dev      ... extracting (100.0%)
Pulling ansible        ... done
Pulling mongo          ... done
```

Experiment Enabled by VMs

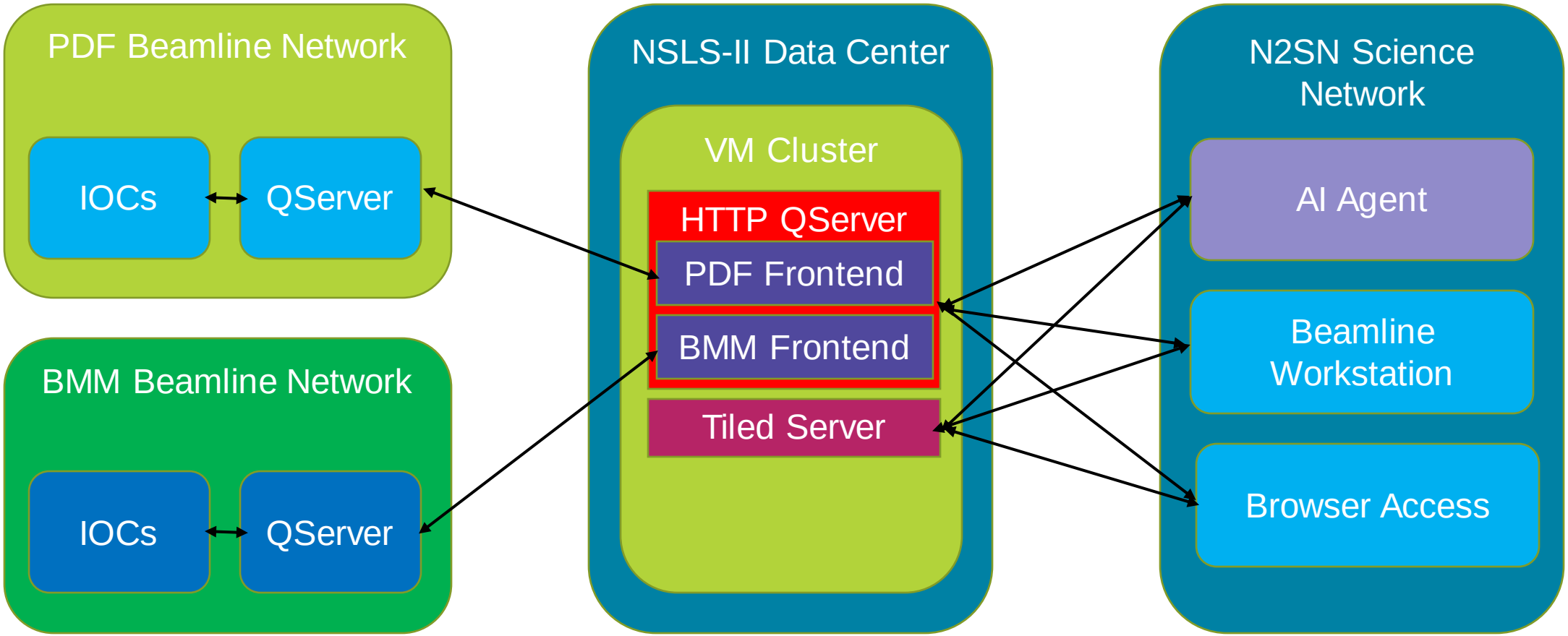
Largely deployed and hosted on VMWare server (central and beamline)

Building Out Distributed Experiments

- Real push on getting the infrastructure ready
- Beamtime is precious – use it wisely
  - AI can help maximize results with automation
- Analyze and understand in experiment time
  - Make decisions using AI to drive the experiment
- Proof of concept honing the distributed tools
- Toward a true human-AI augmented interface



Infrastructure for Multimodal Experiment



Future Plans

Where are we headed?

More Automation

- Pushing on Ansible for even greater automation
- Using VMs in place of physical machines where feasible
- Looking towards software containers in the longer term
  - Leads to much better more reproducible software and services
  - Offers improved development, testing and encapsulation
- Balancing research, experimentation, agility and security
- Physical machine failures should cause minimal downtime

Questions

- DSSI has been using VMWare for a while now
  - Migrating more and more physical machines over
  - Initially central services and other data center applications
- Containers and an eye to the future
  - Not all or nothing—right tool for the right job
  - Industry has largely already moved to containers for services
- Virtualization in either form offers much better hardware use
  - Greater utilization, resilience, failover, versioning, roll backs, ...