Slurm Appliance

A batch HPC cluster defined as code for OpenStack clouds

Steve Brasier

StackHPC

StackHPC Company Overview

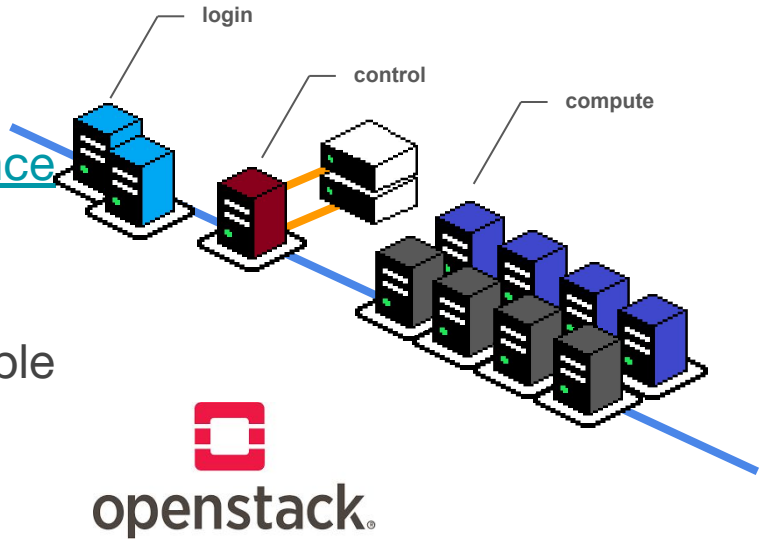
- Formed 2016, based in Bristol, UK
  - Based in Bristol with presence in Oxford, Cambridge, France and Poland
  - Currently around 30 people
- Founded on HPC expertise
  - Software Defined Networking
  - Systems Integration
  - OpenStack Development and Operations
- Motivation to transfer this expertise into Cloud to address HPC & HPDA (AI)
- “Open” Modus Operandi
  - Upstream development of OpenStack capability
  - Consultancy/Support to end-user organizations in managing HPC service transition
  - Scientific-WG engagement for the Open Infrastructure Foundation
- Hybrid Cloud Enablement

StackHPC

StackHPC

Overview

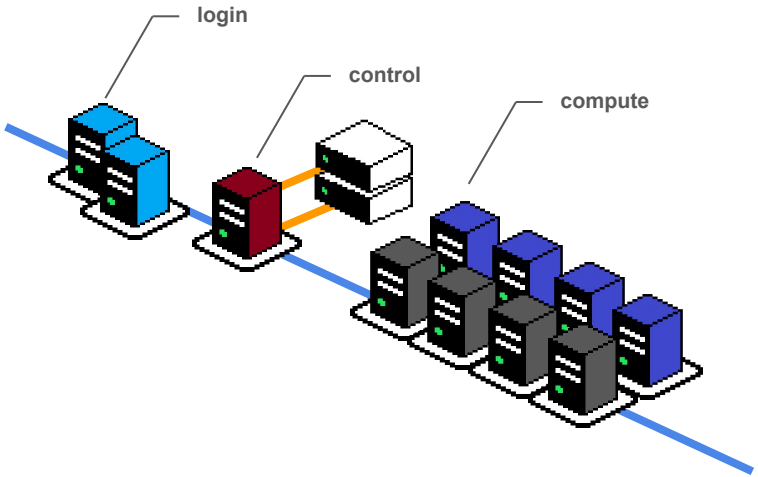
- Batch HPC environment on OpenStack
- Entirely defined by code
  github.com/stackhpc/ansible-slurm-appliance
- Multi-user long-lived site cluster
- Shared, extendable and open-source
- Useable & sensible defaults, but customisable



Overview

- Slurm batch scheduler
- Monitoring
- Web-based portal
- Workload support

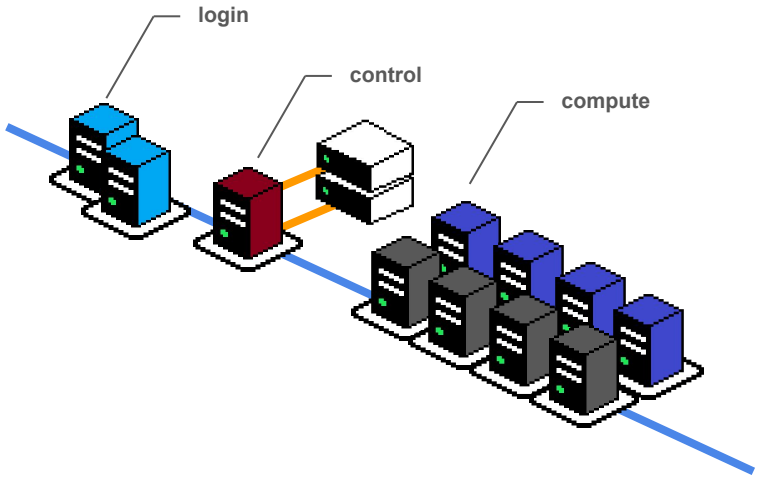
openHPC
Grafana Prometheus
OPEN OnDemand
APPTAINER
EESSI Rocky Linux™



StackHPC

Overview

- **Slurm batch scheduler**
- Monitoring
- Web-based portal
- Workload support

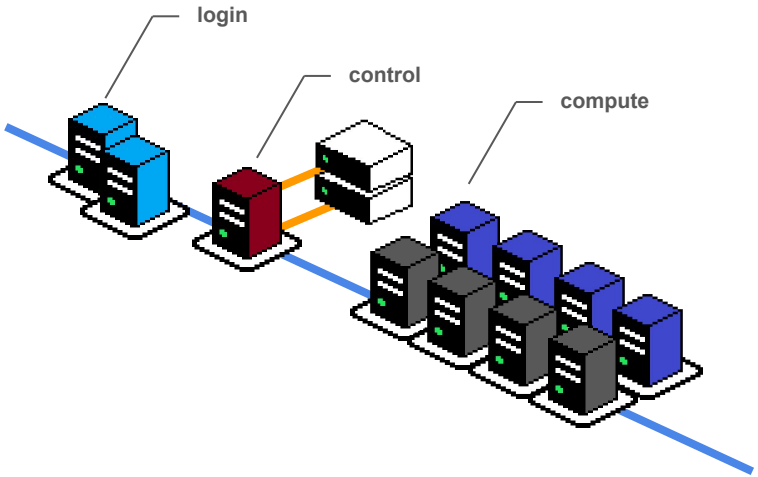


- Accounting database
- Login control
- LBNL Node Heath Checks
- Topology-aware scheduling

StackHPC

Overview

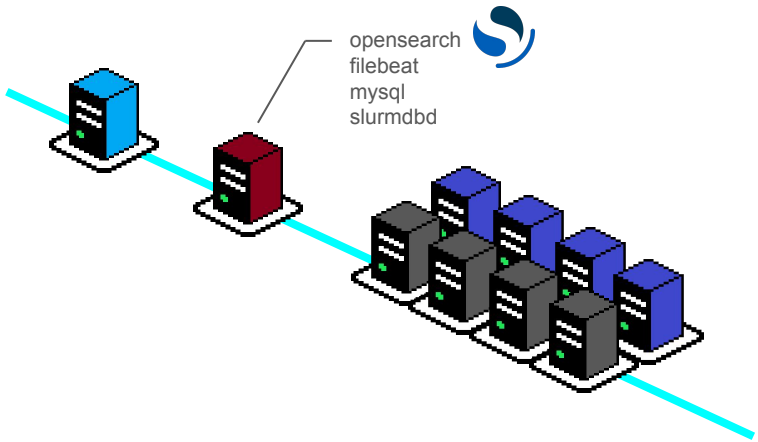
- Slurm batch scheduler
- **Monitoring**
- Web-based portal
- Workload support



StackHPC

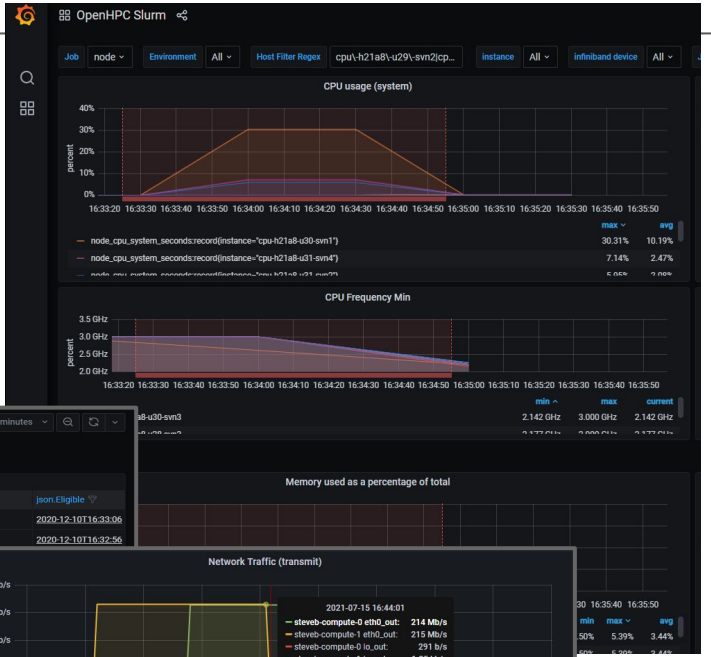
Monitoring

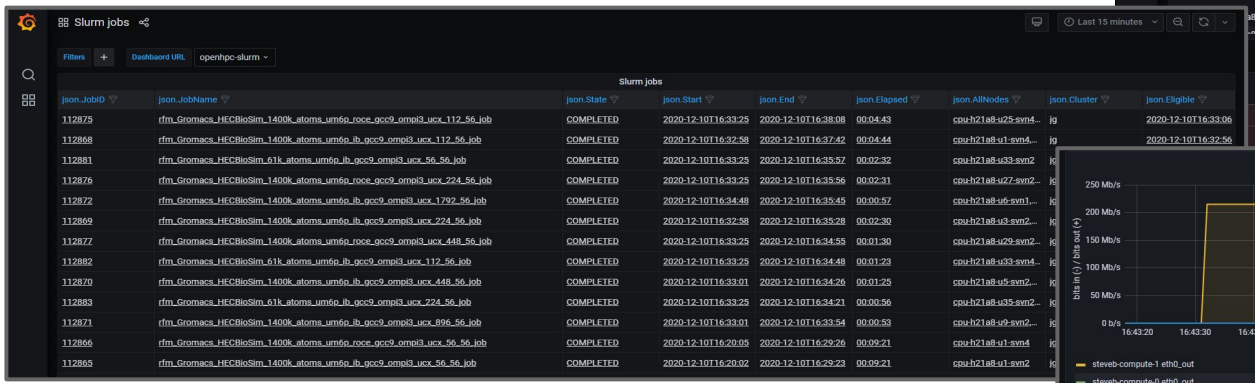
- “Standard” Linux metrics stack of node exporter + Prometheus + Grafana
- SlurmDBD + MySQL + Filebeat + OpenSearch to integrate with Slurm jobs
- Alertmanager

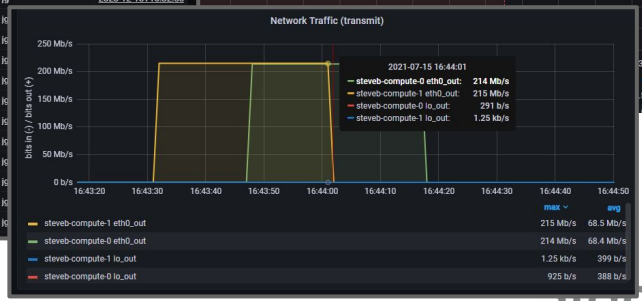


Monitoring

- “Standard” Linux metrics stack of node exporter + Prometheus + Grafana
- SlurmDBD + MySQL + Filebeat + OpenSearch to integrate with Slurm jobs
- Alertmanager

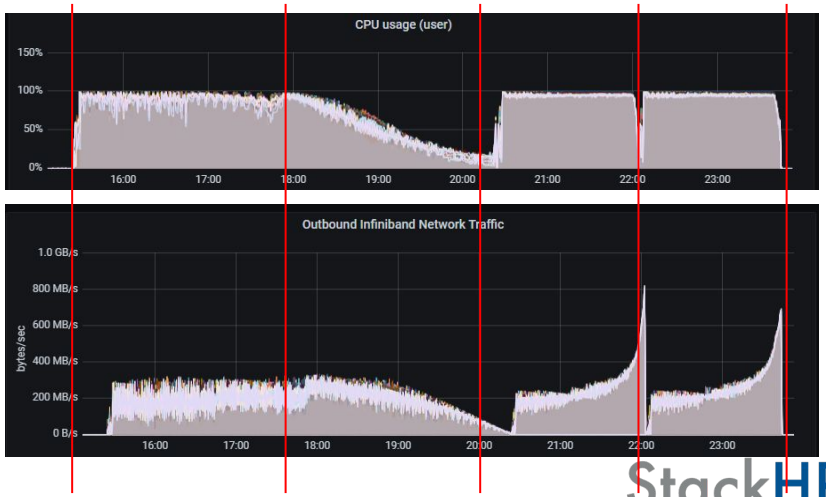






Monitoring

- “Standard” Linux metrics stack of node exporter + Prometheus + Grafana
- SlurmDBD + MySQL + Filebeat + OpenSearch to integrate with Slurm jobs
- Alertmanager
- Plus other prometheus exporters:
  - Open Ondemand
  - Slurm
- LBNL Node Health Checks
  - Set node DOWN if it loses mounts etc …



StackHPC

Overview

- Slurm batch scheduler
- Monitoring
- **Web-based portal**
- Workload support

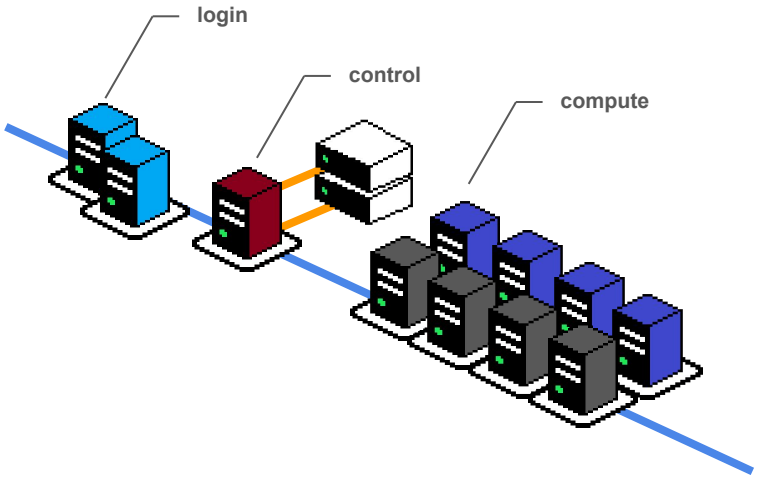
openHPC

Grafana Prometheus

OPEN OnDemand

APPTAINER

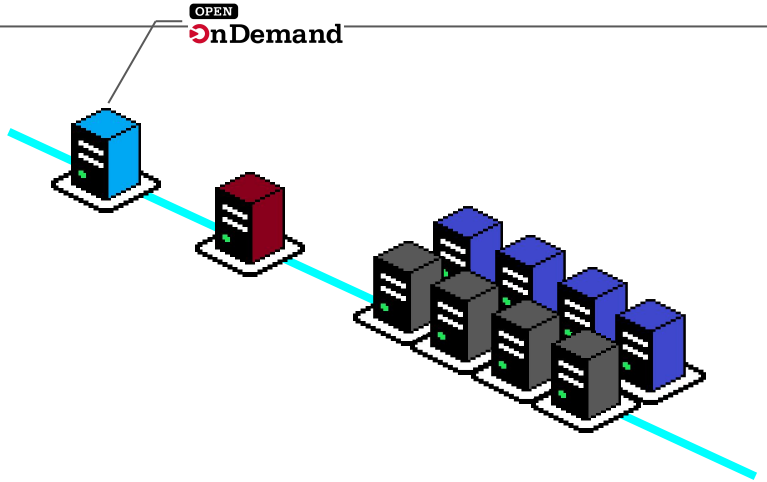
EESSI Rocky Linux™

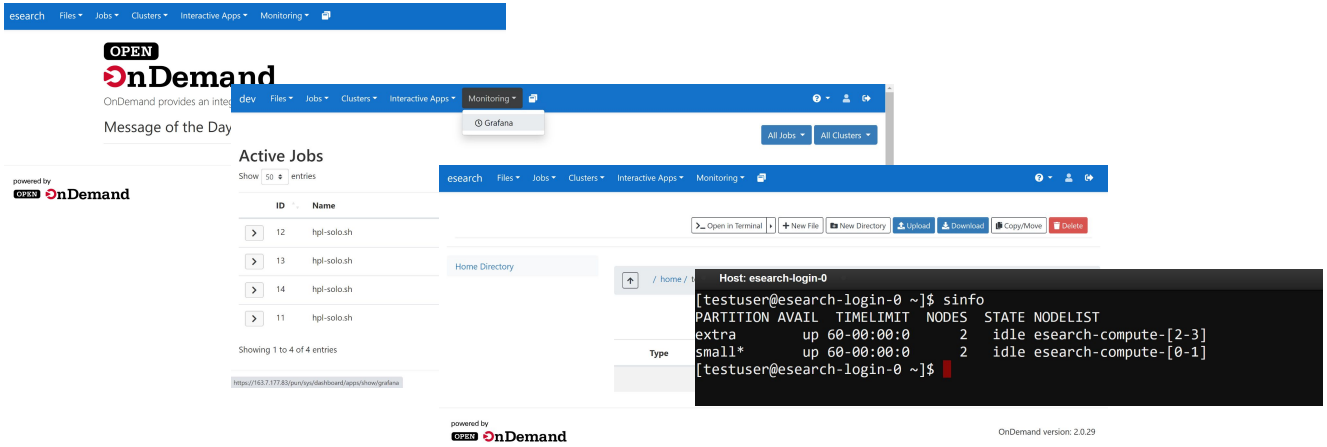


StackHPC

User Portal

- Web-based Open Ondemand user portal
  - Shell
  - File explorer
  - Monitoring (proxied)
  - Job templates & submission etc

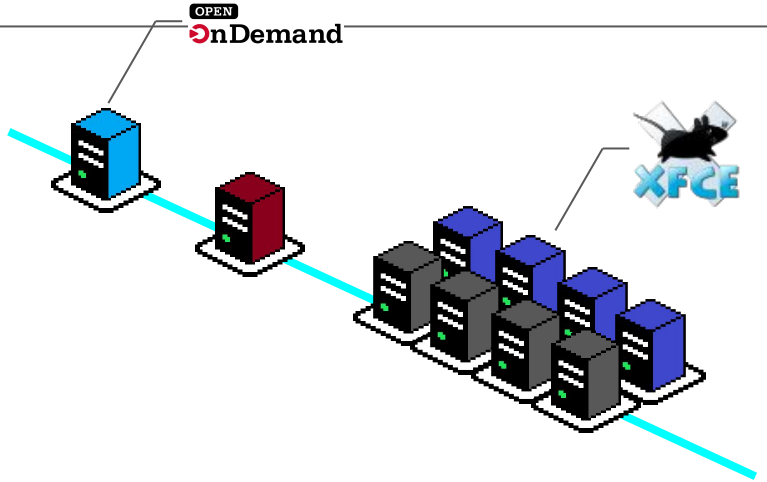


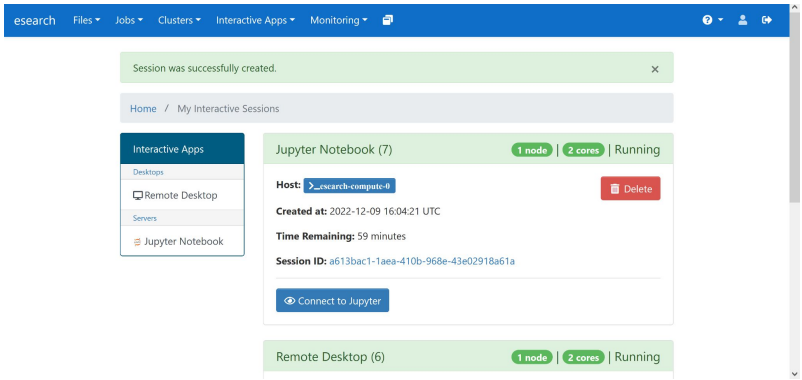


StackHPC

User Portal

- Preconfigured compute node “apps”
  - Remote desktop
  - Jupyter notebook
  - Rstudio
  - Codeserver
  - Matlab*





StackHPC

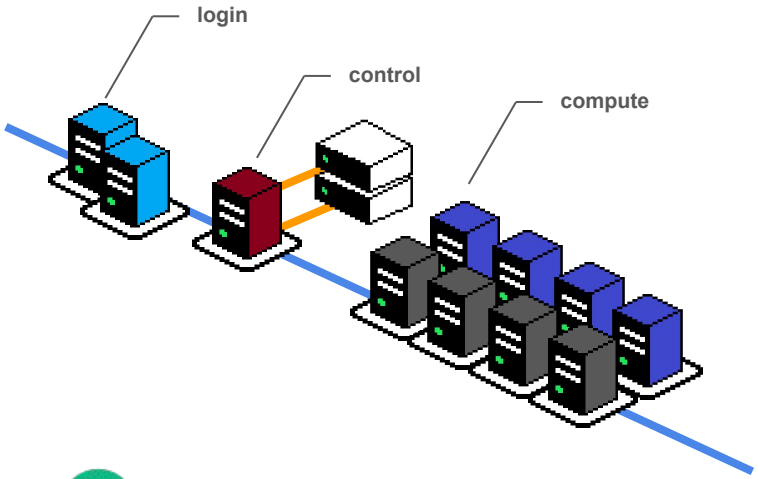
Overview

- Slurm batch scheduler
- Monitoring
- Web-based portal
- **Workload support**

openHPC

Grafana Prometheus

OPEN OnDemand



EESSI
EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

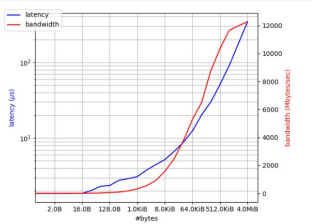
APPTAINER

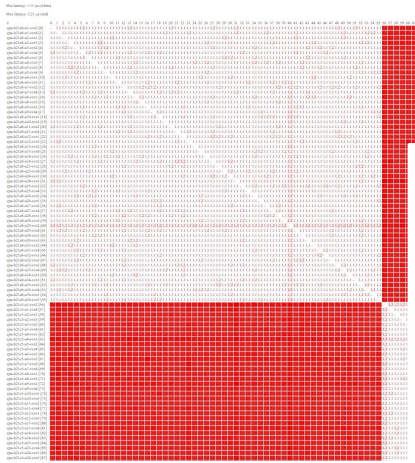
Rocky Linux™

Built-in MPI-based tests

- pingpong, pingmatrix, HPL
- Known-good (OpenHPC-based) MPI software stack - test environment & h/w
- Post-deploy acceptance/confirmation
- Useful for debugging

```
Summary for pingpong using 2x scheduler-selected nodes in 'baremetal' partition, job ID 1, device 'mlx5_bond_0:1':

Nodes: compute-[0-1]
Zero-size msg latency: 1.87 us
Max bandwidth: 12266.95 Mbytes/s (98.13560000000001 Gbit/s)
```





Min latency: 1.11 µs (white)

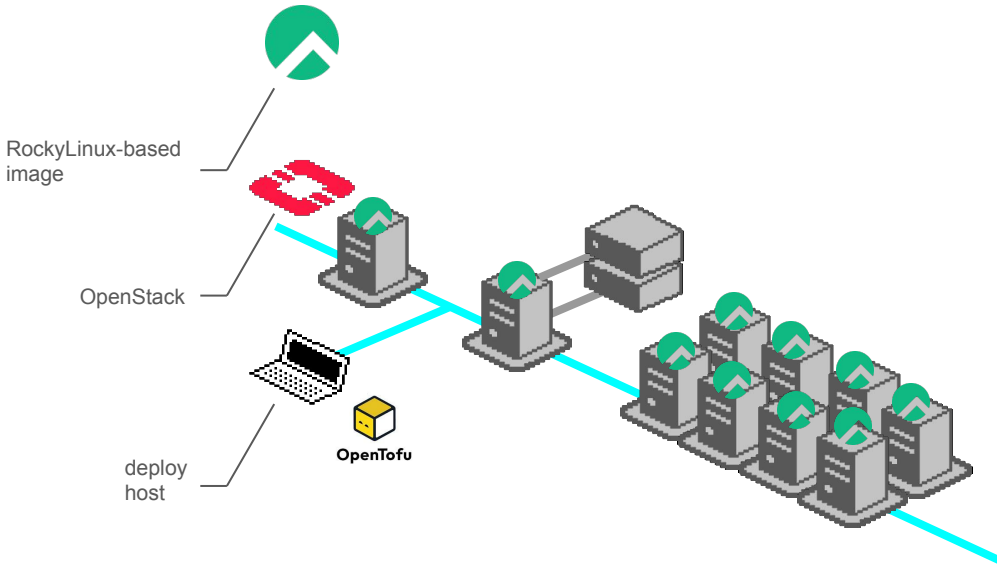
Max latency: 2.21 µs (red)

#	0	1	2	3	4	5	6	7	8	9	10	11	12	13
cpu-h21a8-u1-svn2 [0]	-	1.1	1.1	1.1	1.1	1.1	1.2	1.1	1.1	1.1	1.1	1.1	1.1	1.2
cpu-h21a8-u1-svn4 [1]	1.1	-	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
cpu-h21a8-u2-svn1 [2]	1.1	1.1	-	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.2	1.1	1.1	1.1
cpu-h21a8-u2-svn3 [3]	1.1	1.1	1.1	-	1.1	1.1	1.1	1.1	1.2	1.1	1.1	1.1	1.1	1.1
cpu-h21a8-u3-svn2 [4]	1.1	1.1	1.1	2.1	1.1	-	1.1	1.1	1.1	1.2	1.2	1.1	1.1	1.1
cpu-h21a8-u3-svn4 [5]	1.2	1.1	1.1	1.1	1.1	1.1	-	1.2	1.1	1.2	1.1	1.1	1.1	1.2
cpu-h21a8-u4-svn1 [6]	1.1	1.1	1.1	1.1	1.1	1.1	1.2	-	1.1	1.1	1.1	1.1	1.2	1.1
cpu-h21a8-u4-svn3 [7]	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	-	1.1	1.1	1.1	1.1	1.1
cpu-h21a8-u5-svn2 [8]	1.1	1.1	1.1	1.1	1.2	1.2	1.2	1.1	1.1	-	1.1	1.1	1.1	1.2
cpu-h21a8-u5-svn4 [9]	1.1	1.1	1.1	1.1	1.1	1.2	1.1	1.1	1.1	1.1	-	1.1	1.1	1.1
cpu-h21a8-u6-svn1 [10]	1.1	1.1	1.1	2.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	-	1.1	1.1
cpu-h21a8-u6-svn3 [11]	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.2	1.1	1.1	1.1	1.1	-	1.1
cpu-h21a8-u7-svn2 [12]	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	-
cpu-h21a8-u7-svn4 [13]	1.2	1.1	1.1	1.1	1.1	1.1	1.2	1.1	1.1	1.2	1.1	1.1	1.1	1.1

```
Summary for hpl-solo (4 nodes) job 11 using mlx5_0:1:
  Max:  35.163 gflops
  Min:  34.3 gflops
  Mean: 34.579750000000004 gflops

Individual node results (gflops):
dev-compute-1: 34.3
dev-compute-2: 35.163
dev-compute-3: 34.401
dev-compute-4: 34.455
```

StackHPC

Deployment Process

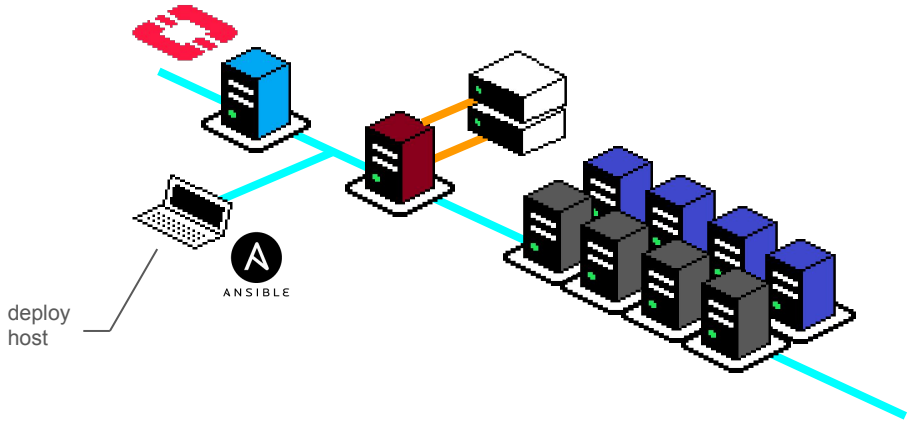


OpenTofu creates infrastructure and Ansible inventory file

```
module "cluster" {
 source                    ="../../site/tofu/"
 environment_root          = var.environment_root
 cluster_name              ="staging"
  login = {
   head = {
     nodes          = ["login-0"]
     flavor         ="vm.cpu.himem.eighth"
     fip_addresses = ["198.51.100.19"]
     fip_network    ="external"
   }
 }

 compute = {
   saphrapid_himem = {
     nodes                = ["standard-0", "standard-1", "standard-2"]
     flavor               ="vm.cpu.himem.v2.full"
   }
   a100_gpu = {
     nodes                = ["a100-0"]
     flavor               ="vm.gpu.4x.a100.full"
   }
 }
}
```

StackHPC

Deployment Process



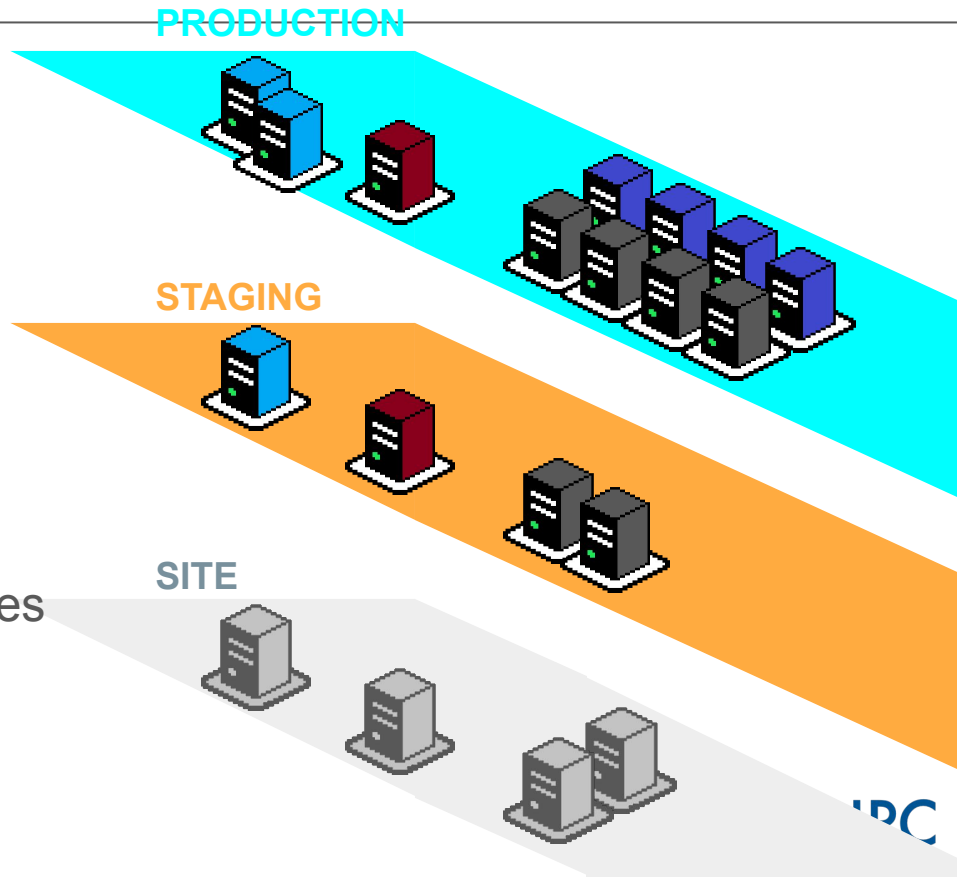
Ansible configures instances into a cluster

```
- name: Ensure OpenHPC repos
  ansible.builtin.yum_repository : "{{ item }}"
  loop: "{{ openhpc_repos }}"

- name: Install required Slurm packages
  ansible.builtin.dnf :
   name: "{{ openhpc_slurm_pkglist  }}"

...

- name: Template slurm.conf
  template:
   src: "{{ openhpc_slurm_conf_template }}"
   dest: "{{ openhpc_slurm_conf_path }}"
   owner: root
   group: root
   mode: '0644'
  when: openhpc_enable.control | default(false)
  notify:
   - Restart slurmctld service
...
```

StackHPC

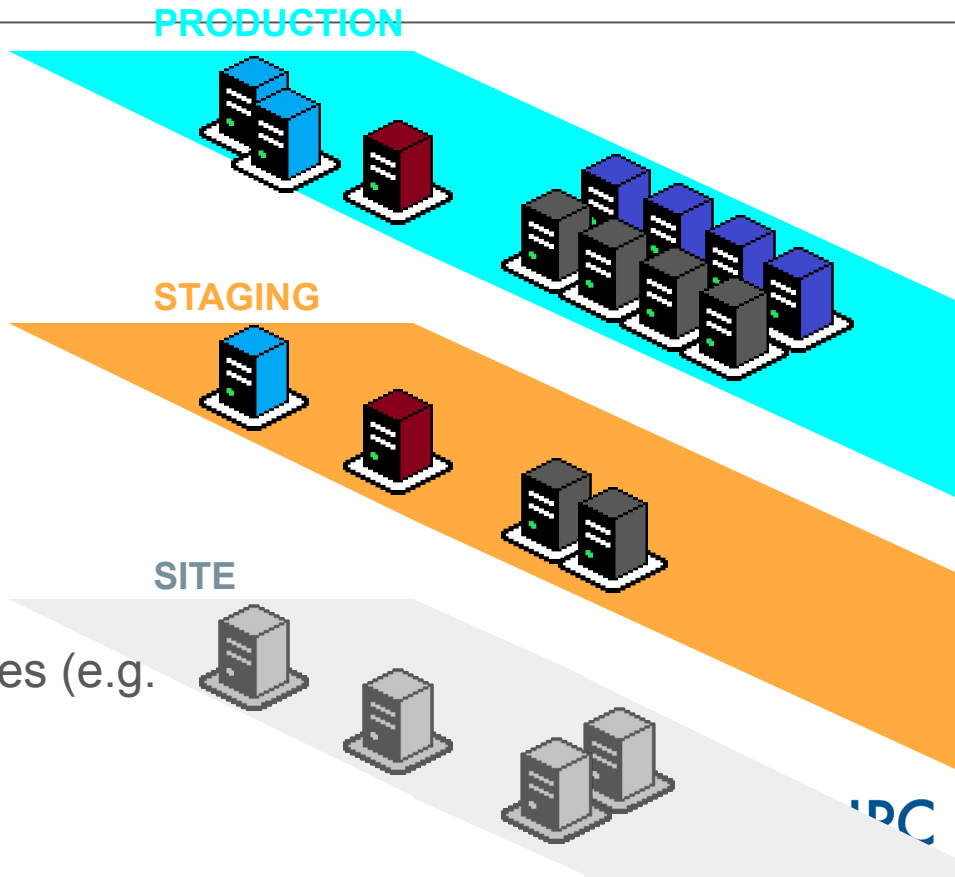
Environments

- Configuration drawn from multiple, layered Ansible inventories
  - *common* - StackHPC defaults
  - *site* - per-site defaults
  - *production, staging*, …
- Shared aspects only defined once
- Minimal differences
  - Cluster name
  - Floating IPs + external DNS names
  - Numbers of nodes



Environments

- Git workflows
- Upgrades via reimaging
  - Confidence on production roll-out
  - But changes are disruptive
- Easy development
- Minimise resources required
  - Fewer nodes
  - Smaller instances
  - VMs instead of baremetal
  - Temporarily move limited resources (e.g. GPUs)



EESSI Integration

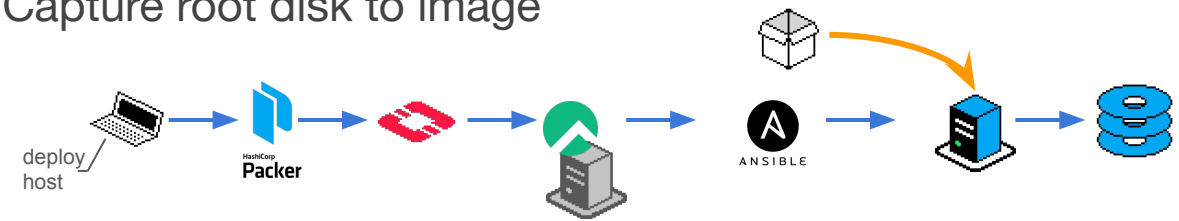
- Since pilot repo days …
- Docs converted to Ansible

Docs are great - thank you!

- By default:
  - Configure Squid proxy on control node
  - Configure clients to use it
  - Check for NVIDIA GPUs and link host libraries
- Obviously requires outbound HTTP …
  - Client cluster with only HTTP**S** allowed out
  - Open PR to add CVMFS Stratum 1 server

Image Build

- Using upstream repos isn't repeatable …
- Hashicorp Packer
  - Launch VM in OpenStack
  - Run (some) Ansible
    - DNF installs
    - Template out files, …
  - Capture root disk to image
- Adds complexity to Ansible
- *State* preserved via remote storage

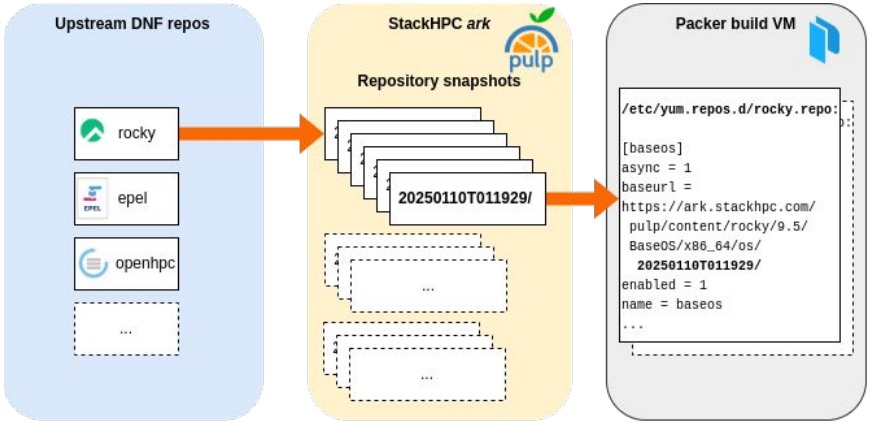


Package Problems …

- Tests in staging at client site using Lustre: Image missing RDMA verbs header file
  - Latest NVIDIA OFED version clashed with OpenHPC libfabric package required for MPI packages
- New NVIDIA CUDA packages released
  - CI image builds fail - not compatible with the NVIDIA drivers
- RockyLinux 9.5 released
  - Packages from RockyLinux 9.4 disappear from mirrors
  - Our CI tests found issue with latest podman package
- **No version of NVIDIA OFED which worked with RockyLinux 9.5**

Image Build

- Using upstream repos (still) isn't repeatable …
- Use package mirror with "snapshots" of upstream repositories
- StackHPC "Ark": Pulp server for OpenStack packages



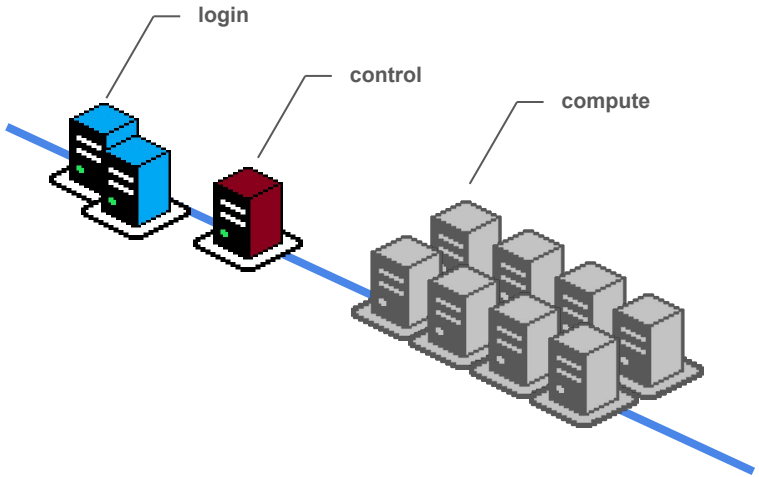
- *Repeatable* builds

StackHPC

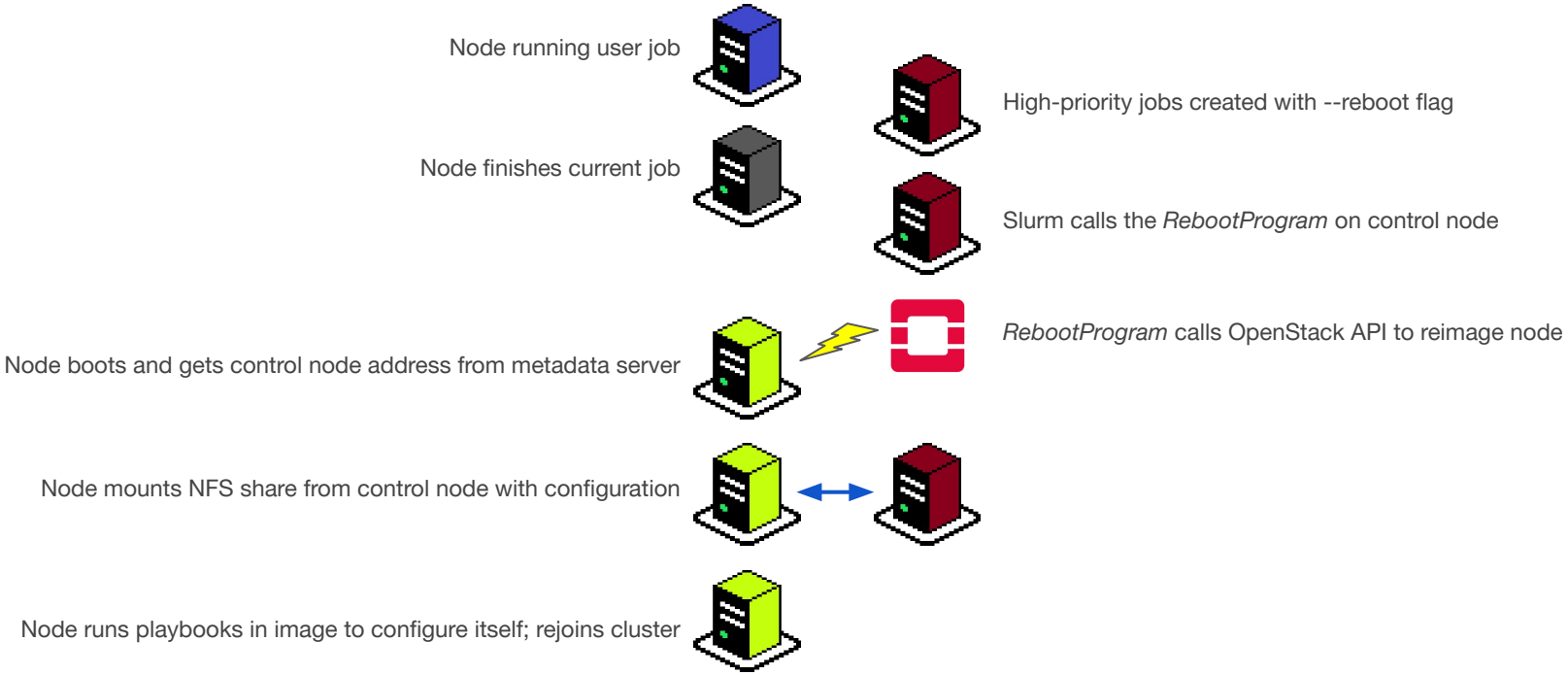
Lower-disruption upgrades

Update via reimage DRAINs compute nodes

- Use the scheduler to reimage compute nodes between user jobs
- Need Ansible to configure nodes?
  - Build code into the image
  - Provide config via metadata + NFS



Lower-disruption upgrades



Features

IAM and access control
- Templated users/groups
- fail2ban & firewalld
- FreeIPA
- LDAP / sssd
- sshd
- opkssh: SSH via OIDC identities

Filesystems
- NFS
- Lustre
- GPFS (Client-specific)
- Arbitrary mount and tmpfs configuration
- CephFS and OpenStack Manila

Slurm integrations
- OpenHPC
- LBNL Node Heath Check
- Topology-aware scheduling
- MySQL

Portals
- Open OnDemand

Networking
- /etc/resolv.conf configuration
- Early boot gateway/default route configuration
- Proxy configuration
- /etc/hosts configuration
- Squid

Software
- NVIDIA GPU drivers and CUDA
- NVIDIA DOCA-OFED
- Podman
- EESSI
- Apptainer
- Local Pulp mirror
- Xfce desktop
- Remote apps

System
- journald
- Persistent hostkeys
- TuneD
- CA cert management
- chrony
- mdadm raid root disks
- systemd dropin configuration

StackHPC