

OpenStack, RDMA and K8s

Bread, Oil and Vinegar?



John Garbutt, Principal Engineer

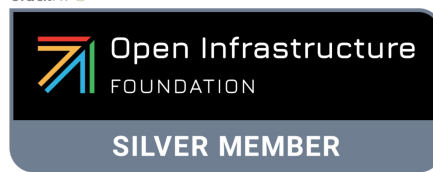
StackHPC

StackHPC Company Overview

StackHPC

- Formed 2016, based in Bristol, UK
 - Based in Bristol with presence in Oxford, Cambridge, France and Poland
 - Currently around 25 people
- Founded on HPC expertise
 - Software Defined Networking
 - Systems Integration
 - OpenStack Development and Operations
 - Growing Staff in AI/ML
- Motivation to transfer this expertise into Cloud to address HPC & HPDA (AI)
- “Open” Modus Operandi
 - Upstream development of OpenStack capability
 - Scientific-WG engagement for the Open Infrastructure Foundation
- Hybrid Cloud Enablement

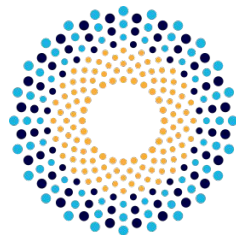
StackHPC



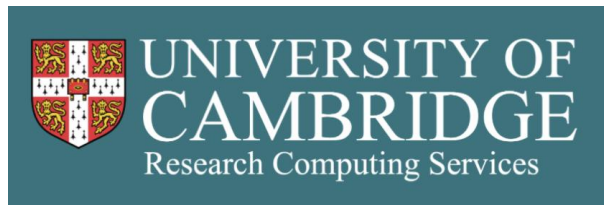
Thank you!

StackHPC

JASMIN



iris



DiRAC

Why **OpenStack** and **Kubernetes**?

Sharing Diverse Infrastructure



Reconfigurable Infrastructure (with Isolation)



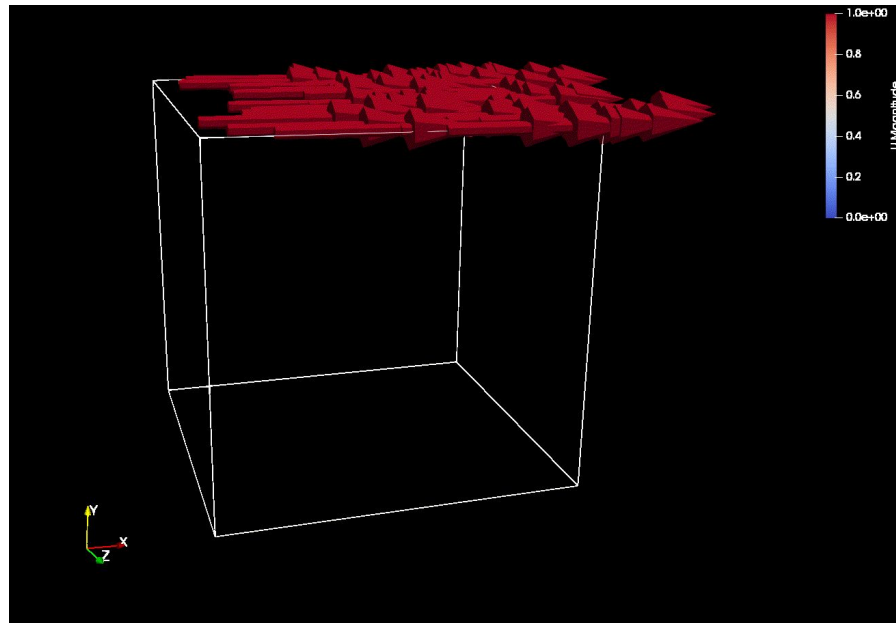
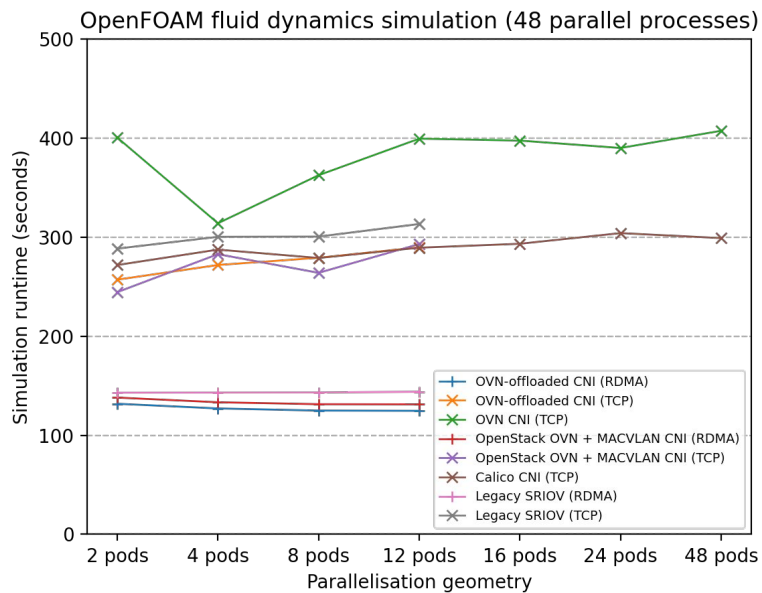
Many Apps built for Kubernetes



Why RDMA networking?

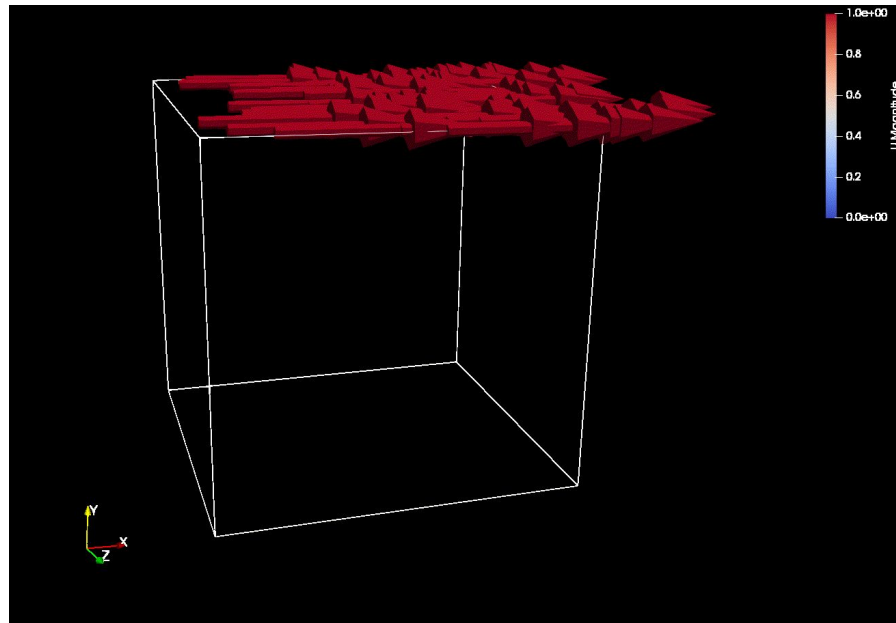
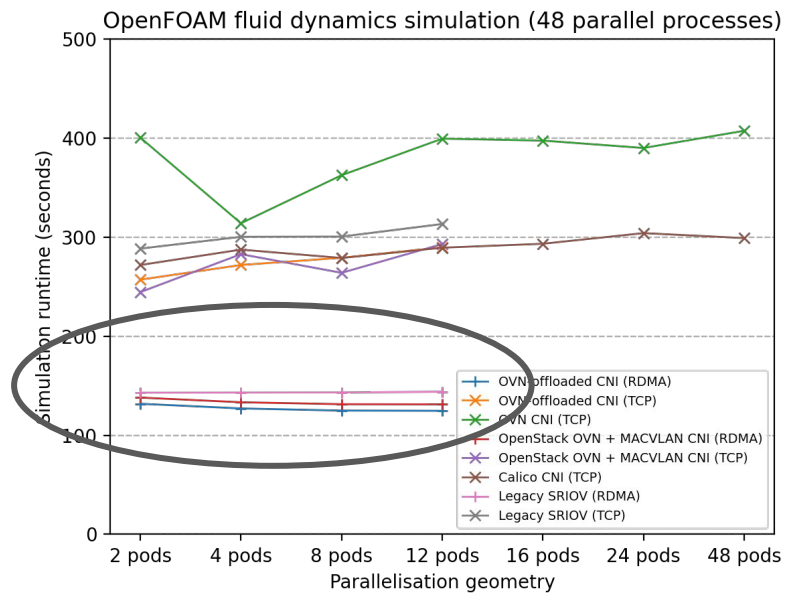
Why Remote Direct Memory Access?

OpenFOAM via kube-perftest



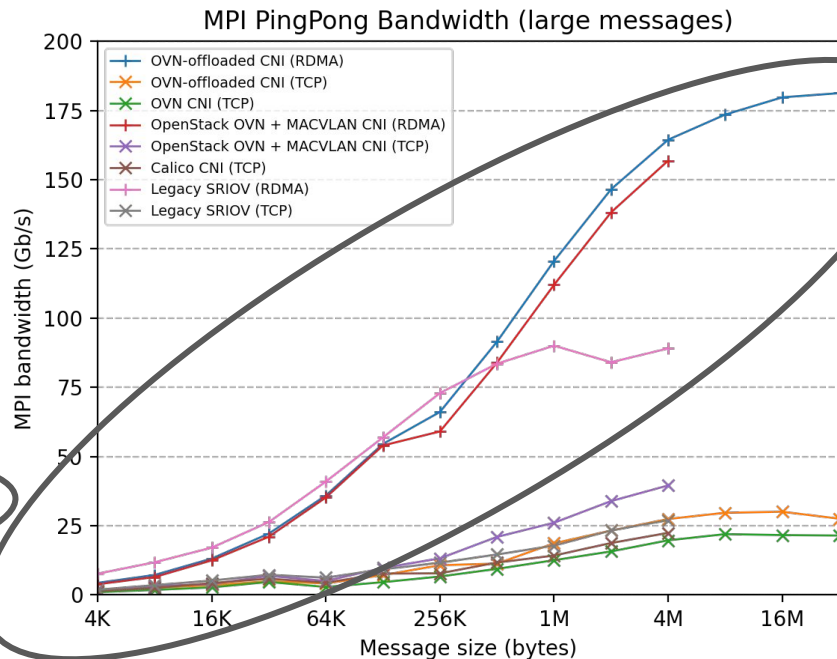
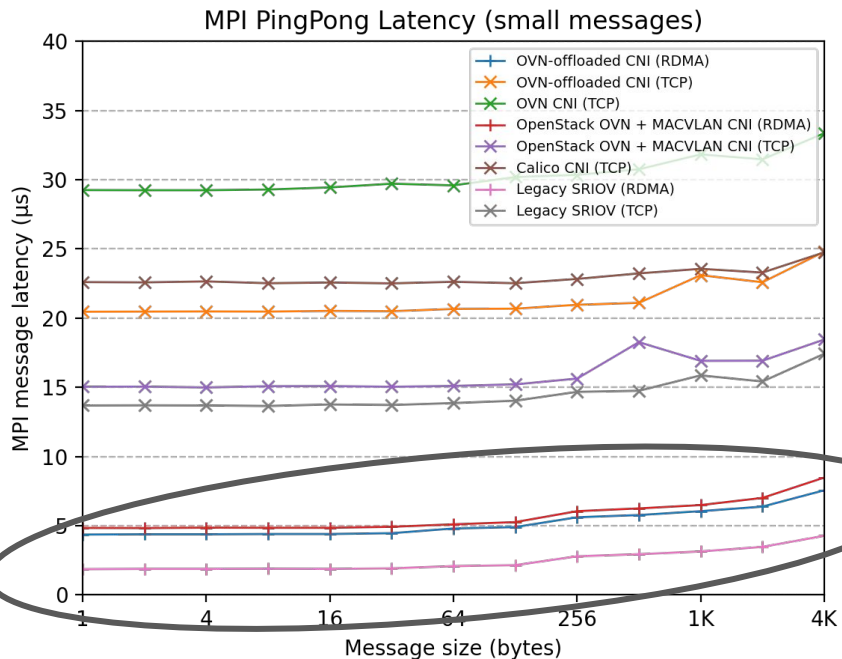
<https://github.com/stackhpc/kube-perftest>

OpenFOAM via kube-perftest



<https://github.com/stackhpc/kube-perftest>

KubeCon: Five ways with a CNI



<https://github.com/stackhpc/kube-perftest>



Repeatable on-demand:
Azimuth using K8s Cluster API

StackHPC

Azimuth Science Platforms

StackHPC

The screenshot shows the 'Create a new platform' dialog in the Azimuth interface. The dialog is divided into two main sections: 'Pick a platform type' and 'Configure platform'. The 'Pick a platform type' section contains six platform options, each with a logo, a brief description, and a 'Select' button.

- DaskHub**: Multi-user Jupyter notebook environment with Dask integration.
- Jupyter Notebook**: Interactively explore Jupyter Notebooks from an existing GitHub, GitLab, Zenodo or Figshare repository. Powered by repo2docker.
- JupyterHub**: Multi-user Jupyter notebook environment.
- Kubernetes**: Kubernetes cluster with optional addons including Kubernetes dashboard, monitoring and ingress.
- Linux Workstation**: Linux workstation (Ubuntu 20.04) accessible via a web browser.
- Linux Workstation (with SSH access)**: Linux workstation (Ubuntu 20.04) accessible via a web browser and by SSH.
- Slurm**: Batch cluster running the Slurm workload manager, the Open OnDemand web interface, and custom monitoring.

The 'Configure platform' section is currently empty. At the bottom right of the dialog is a 'Next' button. The background interface shows the 'iris' logo, 'Other Clouds' dropdown, and a sidebar with navigation options: Platforms, Quotas, Project metrics, Advanced, Switch tenancy, and SSH public key. A 'New platform' button and a 'Refresh' button are visible on the right side of the background interface.

Create Kubernetes using Cluster API

Create a new Kubernetes cluster

Cluster name
Cluster name
Must contain lower-case alphanumeric characters and dash (-) only.

Cluster template
Select a Kubernetes cluster template...
The template determines the Kubernetes version for the cluster.

Control Plane Size
Select a size...
The size to use for the Kubernetes control plane node(s).

Node Groups

Name	Node Size	Node Count
No node groups configured yet.		

+ Add node group

Cluster Addons

- Enable Kubernetes Dashboard?
Allows you to view and manage resources in your cluster using a web browser.
- Enable cluster monitoring?
Enables collection of cluster metrics and web-based dashboards for visualisation.

Advanced Options

- Enable auto-healing?
If enabled, the cluster will try to remediate unhealthy nodes automatically.
- Enable Kubernetes Ingress?
Allows the use of [Kubernetes Ingress](#) to expose services in the cluster via a load balancer.
Requires an external IP for the load balancer.

+ Create cluster

Kubernetes based Platforms

StackHPC

The screenshot displays the 'iris' platform interface. A modal window titled 'Create a new platform' is open, showing the configuration for a 'DaskHub' platform. The background interface includes a sidebar with navigation options like 'Platforms', 'Quotas', 'Project metrics', 'Advanced', 'Switch tenancy', and 'SSH public key'. The main content area shows a list of platforms, with 'john-dask-demo' selected. The modal window contains the following fields and options:

- Platform type:** DaskHub (Multi-user Jupyter notebook environment with Dask integration).
- Platform name:** A text input field with a note: 'Must contain lower-case alphanumeric characters and dash (-) only.'
- Kubernetes cluster:** A dropdown menu with 'john-test-stage' (Kubernetes version: 1.24.2) selected.
- Application version:** A dropdown menu with '0.1.0-dev.0.main.23' selected.
- Notebook CPUs:** A text input field with the value '1'.
- Notebook RAM:** A text input field with the value '2' and a 'GB' unit selector.
- Notebook storage:** A text input field with the value '10' and a 'GB' unit selector.

At the bottom of the modal, there are 'Back' and '+ Create platform' buttons. The background interface also features a 'New platform' button and a 'Refresh' button.

Azimuth Science Platforms

StackHPC

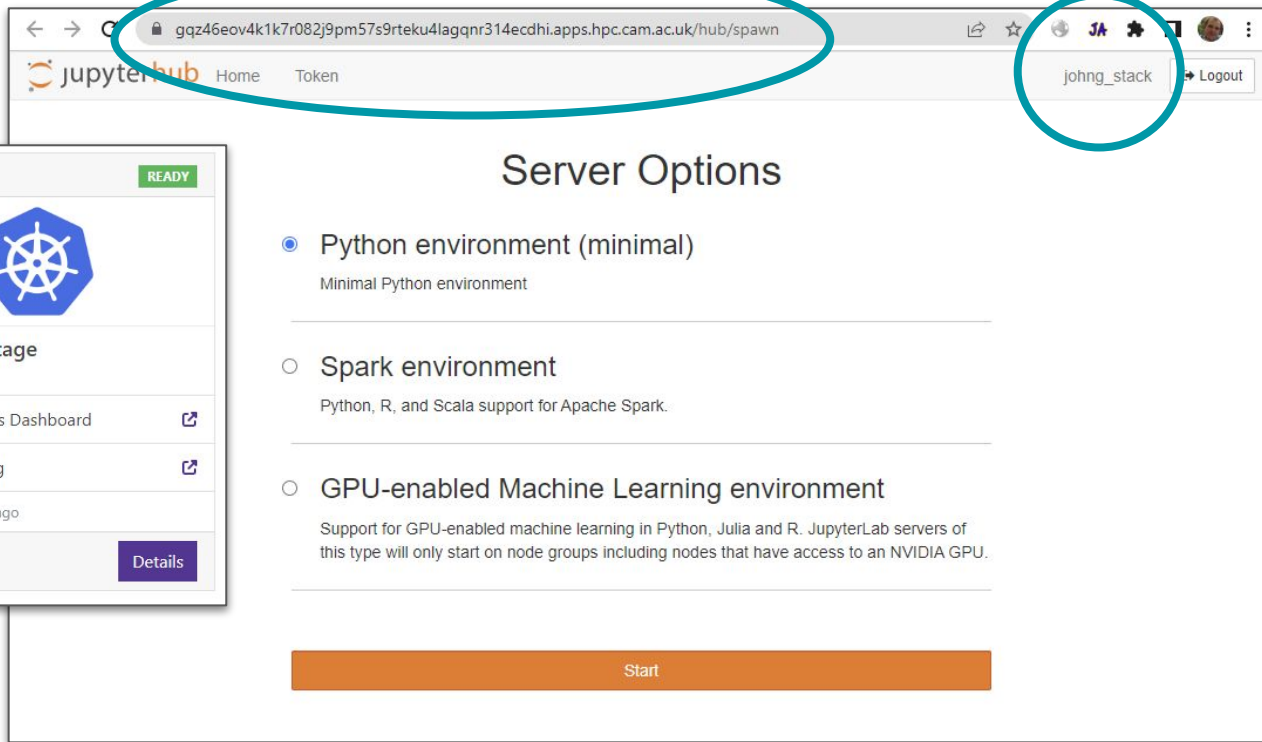
The screenshot displays the 'iris' cloud portal interface. The top navigation bar includes the 'iris' logo, 'Other Clouds', 'Cloud Metrics', 'Documentation', and 'Sign out (johng_stack)'. A left sidebar contains navigation options: 'Platforms', 'Quotas', 'Project metrics', 'Advanced', 'Switch tenancy', and 'SSH public key'. The main content area is titled 'rcp-cloud-portal-demo' and features a 'New platform' button and a 'Refresh' button. Four platform cards are shown, each with a 'READY' status:

- johng-big-laptop** (Linux Workstation): Includes 'Web console' and 'Monitoring' links. Updated 7 days ago.
- johng-voila** (Jupyter Notebook): Includes 'Jupyter Notebook' and 'Monitoring' links. Updated 7 days ago.
- k8s-demo** (Kubernetes): Includes 'Applications', 'Jupyter Notebook', 'Kubernetes Dashboard', and 'Monitoring' links. Created 7 days ago.
- mattp-slurm** (Slurm): Includes 'Open OnDemand' and 'Monitoring' links. Updated 8 days ago.

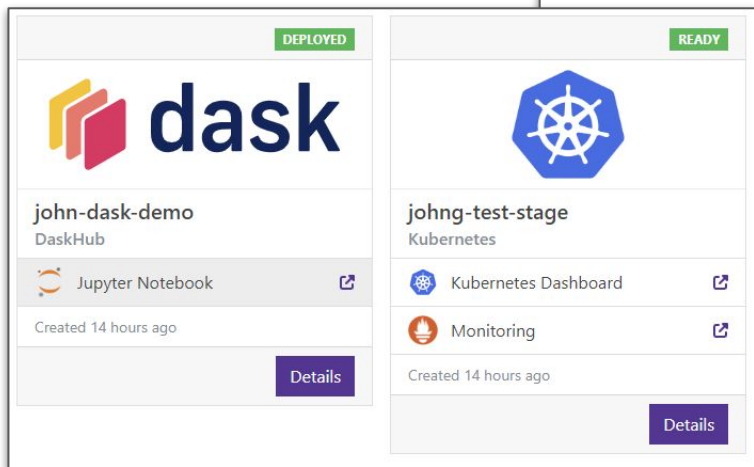
The 'iris' logo is located in the bottom right corner of the interface.

JupyterHub with Zenith Powered SSO

StackHPC



The screenshot shows a web browser window with the URL `gqz46eov4k1k7r082j9pm57s9rteku4lagqnr314ecdhi.apps.hpc.cam.ac.uk/hub/spawn` highlighted in a teal oval. The browser's address bar also shows the user `johnh_stack` and a `Logout` button, also highlighted in a teal oval. The page content includes a `jupyterhub` logo and navigation links for `Home` and `Token`. The main heading is `Server Options`, followed by three radio button options: `Python environment (minimal)`, `Spark environment`, and `GPU-enabled Machine Learning environment`. A large orange `Start` button is at the bottom.



The image shows two panels. The left panel, titled `john-dask-demo` with a `DEPLOYED` status, features the `dask` logo and a `Jupyter Notebook` icon. The right panel, titled `johng-test-stage` with a `READY` status, features the Kubernetes logo and icons for `Kubernetes Dashboard` and `Monitoring`. Both panels include a `Details` button and indicate they were created 14 hours ago.

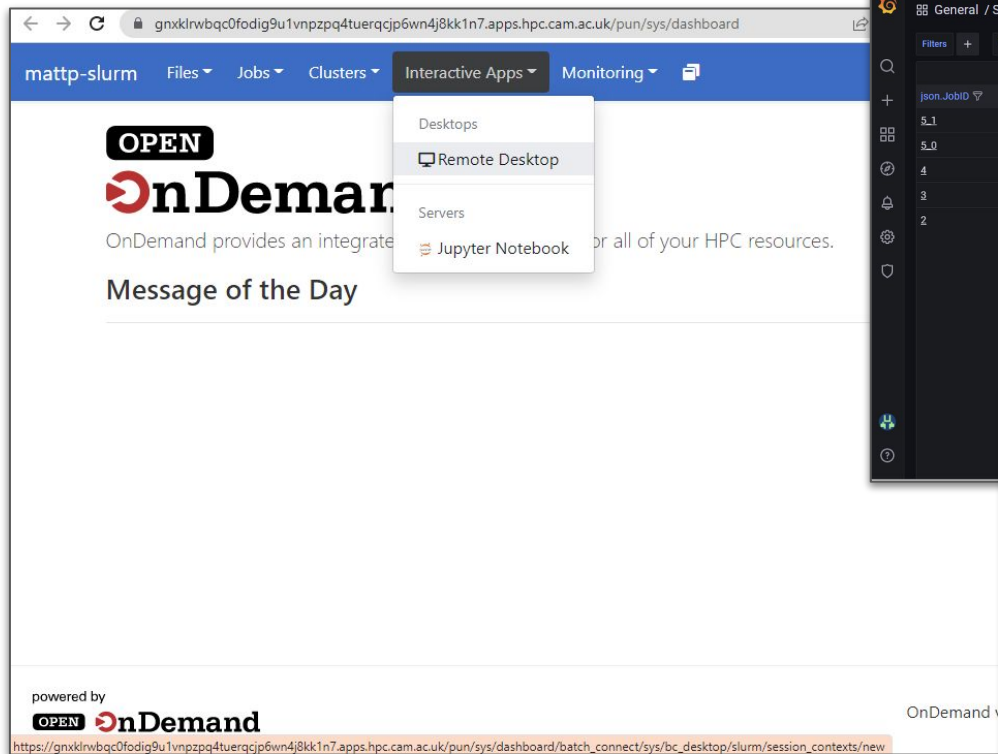
Kubernetes using Cluster API

The image displays a multi-layered view of the Kubernetes dashboard. The background shows a sidebar with navigation options like 'Kubernetes / Networking / Workload' and 'Kubernetes / Persistent Volumes'. Overlaid on this is a main dashboard window titled 'Workloads' showing a 'Workload Status' section with four green circles representing 'Running: 4' Deployments and 'Running: 5' Pods. In the foreground, a modal window titled 'Kubeconfig for k8s-demo' is open, providing instructions to use the configuration file with the 'kubectl' command-line tool. The modal includes buttons for 'Copy to clipboard', 'Download', and 'Regenerate', and a code block containing the following configuration:

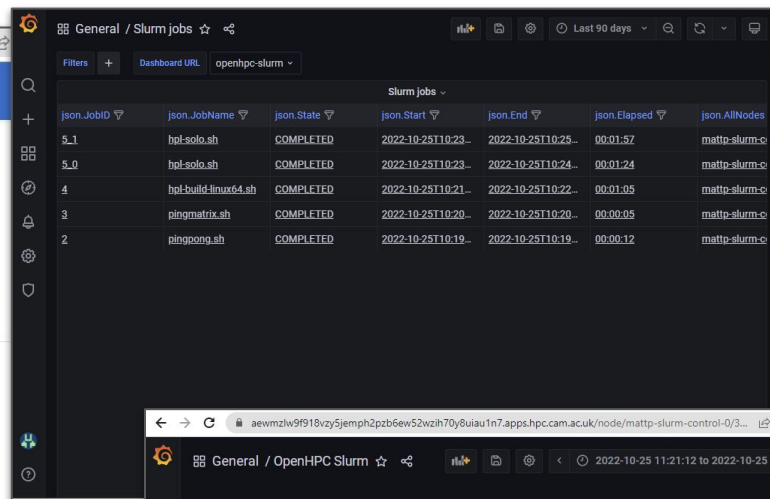
```
apiVersion: v1
clusters:
- cluster:
```

Slurm with Open OnDemand

StackHPC

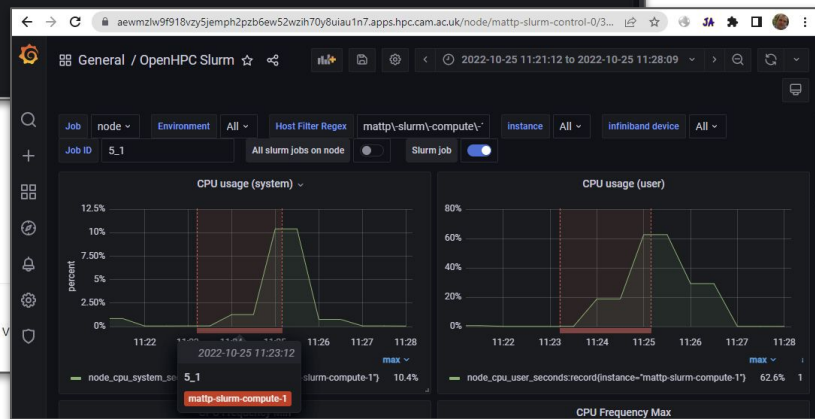


The screenshot shows the Open OnDemand dashboard interface. The browser address bar displays `gnxklrwbqc0fodig9u1vnpzpq4tuerqjp6wn4j8kk1n7.apps.hpc.cam.ac.uk/pun/sys/dashboard`. The navigation bar includes 'mattp-slurm', 'Files', 'Jobs', 'Clusters', 'Interactive Apps', and 'Monitoring'. The 'Interactive Apps' dropdown menu is open, showing options for 'Desktops', 'Remote Desktop', 'Servers', and 'Jupyter Notebook'. The main content area features the 'OPEN OnDemand' logo and a 'Message of the Day' section. At the bottom, it is powered by 'OPEN OnDemand' with a URL: `https://gnxklrwbqc0fodig9u1vnpzpq4tuerqjp6wn4j8kk1n7.apps.hpc.cam.ac.uk/pun/sys/dashboard/batch_connect/sys/bc_desktop/slurm/session_contexts/new`.



The screenshot shows the 'General / Slurm jobs' monitoring dashboard. It features a table of active Slurm jobs with the following columns: 'json.JobID', 'json.JobName', 'json.State', 'json.Start', 'json.End', 'json.Elapsed', and 'json.AllNodes'. The table lists five jobs, all in a 'COMPLETED' state.

json.JobID	json.JobName	json.State	json.Start	json.End	json.Elapsed	json.AllNodes
5_1	hpl-solo.sh	COMPLETED	2022-10-25T10:23...	2022-10-25T10:25...	00:01:57	mattp-slurm-c
5_0	hpl-solo.sh	COMPLETED	2022-10-25T10:23...	2022-10-25T10:24...	00:01:24	mattp-slurm-c
4	hpl-build-linux64.sh	COMPLETED	2022-10-25T10:21...	2022-10-25T10:22...	00:01:05	mattp-slurm-c
3	pingmatrix.sh	COMPLETED	2022-10-25T10:20...	2022-10-25T10:20...	00:00:05	mattp-slurm-c
2	pingpong.sh	COMPLETED	2022-10-25T10:19...	2022-10-25T10:19...	00:00:12	mattp-slurm-c



Bigger Laptop via Guacamole

StackHPC

The image displays the Apache Guacamole web interface. On the left, the 'Create a new platform' form is visible, featuring a 'Linux Workstation' option with a penguin icon. The form includes fields for 'Platform name', 'Workstation Size', and 'Data volume size (GB)'. A 'Create platform' button is at the bottom right of the form.

In the center, a browser window shows the 'RECENT CONNECTIONS' section. A notification bubble indicates that text and images were copied to the clipboard. Below this, the 'ALL CONNECTIONS' section lists 'desktop' and 'shell'.

On the right, a platform card for 'johng-big-laptop' is shown. It features a penguin icon, a 'READY' status, and options for 'Web console' and 'Monitoring'. The card also indicates it was 'Updated 8 days ago' and has a 'Details' button at the bottom.

Single VM with repo2docker

StackHPC

The image illustrates the workflow for creating a single VM with repo2docker. It is divided into three main sections:

- Create a new platform:** A form where you can configure a Jupyter Notebook environment. It includes fields for Platform name, Notebook Repository, and Jupyter Notebook size. A 'Create platform' button is at the bottom right.
- Terminal:** A terminal window showing the execution of git commands to clone a repository and update the title. The output shows a commit message: 'Merge pull request #8 from fcollonval/patch-1' and 'Update title to fit better in binder doc site'.
- READY Card:** A card indicating the platform is ready. It features the Jupyter logo, the name 'johng-voila Jupyter Notebook', and a 'Details' button.

How do you get **RDMA** in **LOKI**?

Three Steps to RDMA in LOKI

StackHPC

LOKI = Linux, OpenStack and Kubernetes Infrastructure

1. RDMA inside OpenStack servers
2. Kubernetes clusters on OpenStack
3. RDMA inside K8s pods



openstack®



kubernetes

Step 1:
RDMA inside OpenStack servers

How to get RDMA?

StackHPC

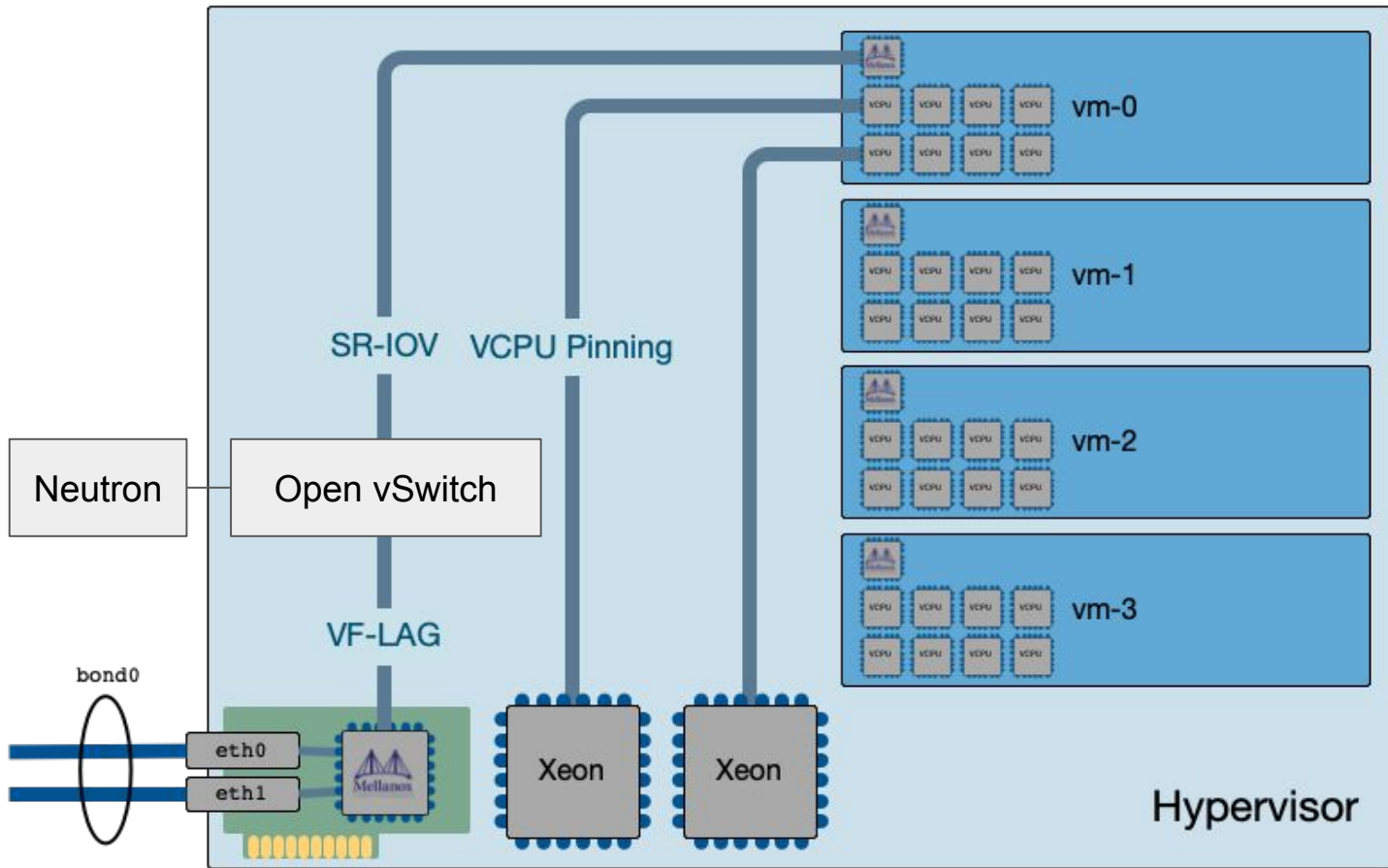
- Baremetal servers (via Nova and Ironic)
 - ... it is just a physical server, all options are possible
 - But you don't want untrusted users having root
- Virtual Machines need drivers for a real NIC
- PCI passthrough a real NIC
 - Dedicated NIC (PF passthrough)
- Legacy SR-IOV
 - Dedicate NIC(s) for SR-IOV
 - Multiple VFs on provider VLAN
- Mellanox VF-LAG
 - bond shared with hypervisor host
 - Full bond bandwidth of the bond in VM
 - OVS hardware offload flows
- Live migration isn't ideal (ignoring vDPA and mdevs, for now)



openstack®

PF = Physical Function
VF = Virtual Function

SR-IOV = Single Root I/O Virtualization



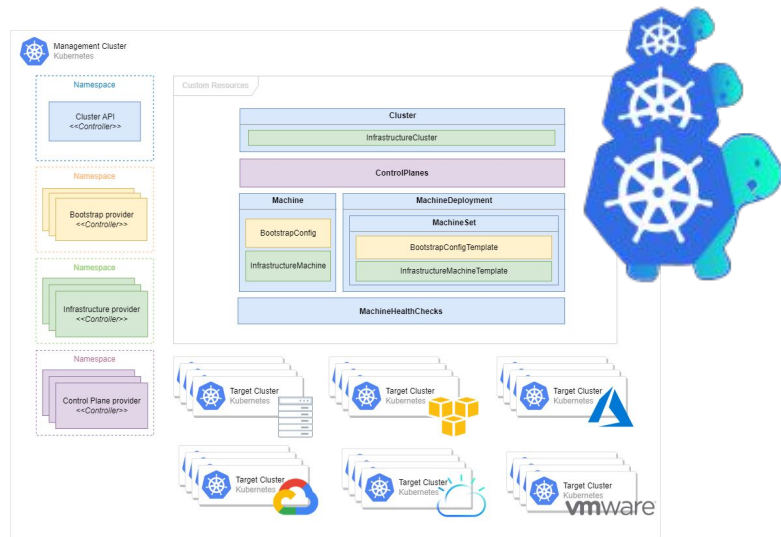
<https://www.stackhpc.com/vflag-kayobe.html>

Step 2:
Create K8s on OpenStack

Kubernetes on OpenStack

StackHPC

- K8s Cluster API
 - Describe Cluster in K8s CRDs
 - CAPI - OpenStack Provider
 - Bootstrap and Management Clusters
 - New OpenStack Magnum driver soon
- K8s Cloud Provider
 - CAPO - OpenStack provider
 - Cinder CSI
 - Octavia Load balancers
- Add ons
 - CNI, monitoring, GPU drivers ...



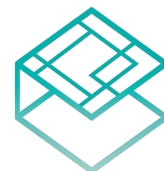
<https://github.com/stackhpc/capi-helm-charts>

Step 3:
RDMA inside k8s pods

RDMA Pods in OpenStack VMs

StackHPC

- OpenStack VM
 - Primary CNI network, using virtual NIC
 - Second SR-IOV VF-LAG
- Multus CNI
 - Pods opt into an additional network
- MACVLAN CNI
 - Additional MAC and IPs on VF-LAG NIC
 - Whereabouts IPAM address range
 - ... currently port security off
- Future ideas
 - Automatic addition of allowed address pairs



CNI



MULTUS

<https://github.com/Mellanox/network-operator#macvlannetwork-crd>

How to get involved?

StackHPC

Contributions are very welcome!

StackHPC

Please try Azimuth:

<https://stackhpc.github.io/azimuth-config/try/>

<https://github.com/stackhpc/azimuth>

<https://github.com/stackhpc/zenith>

K8s Cluster API Helm charts:

<https://github.com/stackhpc/capi-helm-charts>

kube-perftest:

<https://github.com/stackhpc/kube-perftest>

To read our blog or get in touch:

<https://www.stackhpc.com/>

Questions?

john.garbutt@stackhpc.com

StackHPC