Using Cluster API to provide managed Kubernetes on OpenStack

Matt Pryor, Senior Tech Lead, StackHPC
OpenInfra Days Europe @ CERN, 6th June 2024

StackHPC

StackHPC

Kubernetes is 10 today!

StackHPC Company Overview

StackHPC

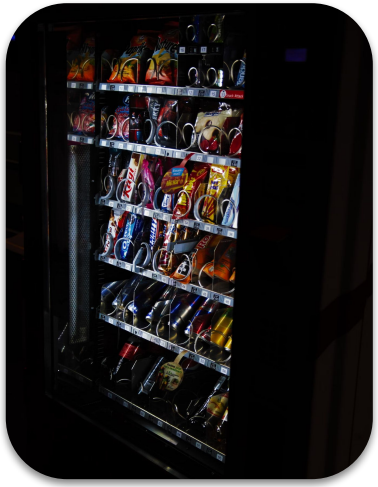
- Formed 2016, based in Bristol, UK
  - Based in Bristol with presence in Oxford, Cambridge, France and Poland
  - Currently around 30 people
- Founded on HPC expertise
  - Software Defined Networking
  - Systems Integration
  - OpenStack Development and Operations
- Motivation to transfer this expertise 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 Three Pillars

Reconfigurable and isolated infrastructure

Performance to extract maximum value

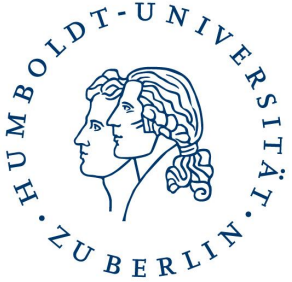


Self-service platforms

Open Source Co-Development

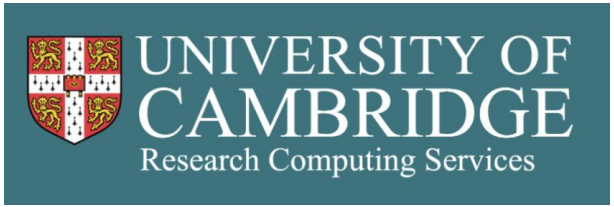
StackHPC

JASMIN



Catalyst Cloud

GRAPHCORE

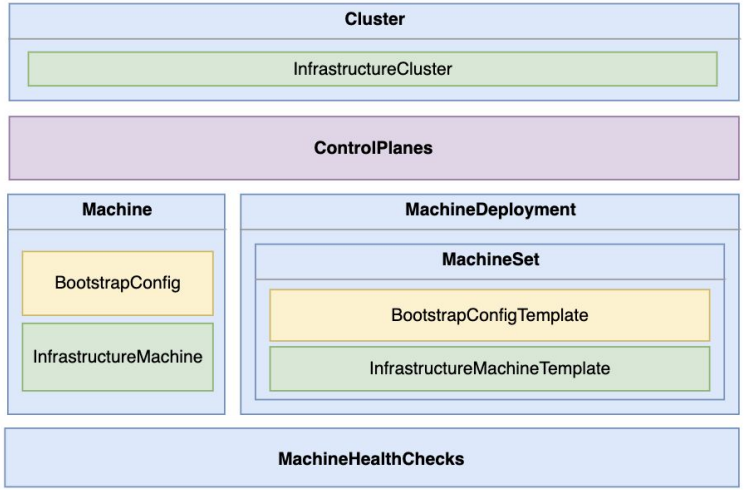


DiRAC

What is Cluster API (CAPI)?

StackHPC

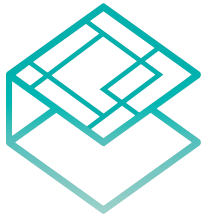
- Declarative API to manage Kubernetes clusters
- Multiple infrastructure providers
  - Provision machines, load balancers, networks
  - OpenStack, public cloud, bare metal
- Kubernetes deployed using kubeadm
- Provider-agnostic auto-healing, auto-scaling and rolling upgrade
- Management cluster runs CAPI controllers
- Workload clusters managed by CRUD operations on management cluster



Handling cluster addons

StackHPC

- Cluster needs addons to be fully functional
  - Networking (CNI)
  - Cloud integrations (CCM, CSI)
  - Ingress controller
  - Monitoring and logging
  - GPU and NIC drivers
- StackHPC addon provider
  - Open-source
  - Declarative interface for addons
  - `HelmRelease` and `Manifests` CRDs
  - Templating of properties from CAPI resources



CNI

CONTAINER STORAGE INTERFACE

KUBERNETES INGRESS NGINX

Prometheus

CAPI Helm charts

StackHPC

- Cluster API clusters composed of several resources with references to each other
- Addons need to be wired up correctly
- Use Helm to template resources for a cluster
  - Knowledge encapsulated in charts
  - Charts are open-source
  - Simplified interface for consumers
  - Can be reused in multiple contexts
  - Tested on Kubernetes versions up to 1.30
- Cluster API now has `ClusterClass`
  - Extremely promising but still officially experimental
  - Will be integrated with CAPI Helm charts in future

```
kubernetesVersion: 1.29.4
machineImageId: <id>

machineSSHKeyName: <name>

clusterNetworking:
  externalNetworkId: <id>

controlPlane:
  machineFlavor: <name>

nodeGroups:
  - name: md-0
    machineFlavor: <name>
    machineCount: 2
```

Magnum integration

StackHPC

What is Magnum?

StackHPC

- OpenStack project for managing container orchestration engines (COEs)
- Allows users to provision COEs in their OpenStack project to manage container-based workloads
  - Cluster templates define the available configurations
  - Clusters are where containers can be scheduled
- Designed to support multiple COEs using drivers
  - Existing drivers for Kubernetes, Swarm and Mesos
  - Swarm and Mesos drivers now deprecated
- REST server and conductor that communicate via AMQP

openstack®

MAGNUM

an OpenStack Community Project

Existing Kubernetes driver issues

StackHPC

- Uses Heat to provision resources
  - Heat project is increasingly unloved
- Kubernetes deployed using non-standard tools
- Configuration using bash scripts
  - Brittle and difficult to debug
- Bespoke auto-healing and auto-scaling
- Upgrading a cluster almost never works
- Difficult to maintain
- Slow to support new Kubernetes versions

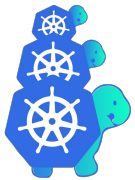
Magnum CAPI Helm driver

StackHPC

- Magnum architecture allows for multiple drivers
- Implement brand new driver that speaks Cluster API
  - Under OpenStack governance on OpenDev
  - Uses CAPI Helm charts to template resources
  - Benefit from upstream development and testing
  - Greatly reduce the amount of brittle, bespoke code
  - Not tied to Magnum release cycle (can evolve faster)
  - Deployed in production, e.g. Catalyst Cloud Kubernetes service
- Existing interfaces will continue to work
  - OpenStack CLI, Horizon plugin, Magnum Terraform plugin

MAGNUM

an OpenStack Community Project



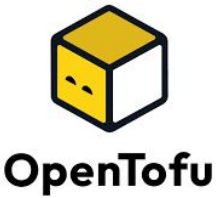
Cluster API

User-friendly Kubernetes in Azimuth

StackHPC

What is Azimuth?

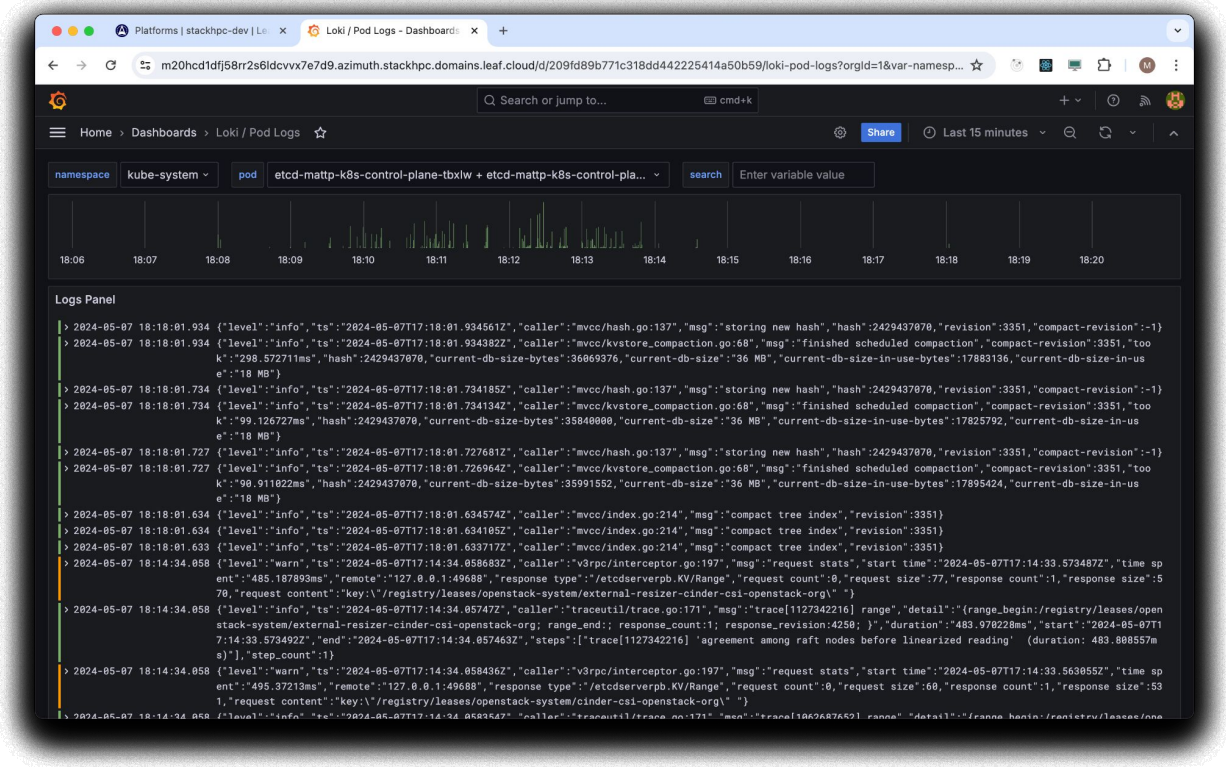
- Web portal for self-service platforms
- Configurable catalogue of curated platforms
  - StackHPC reference platforms
  - Apply site-specific optimisations
  - Automation using standard tools
- Platform services exposed using Zenith
  - Tunneling application proxy
  - No public IP required
  - SSO and TLS
- Manage platform users with Keycloak



Kubernetes

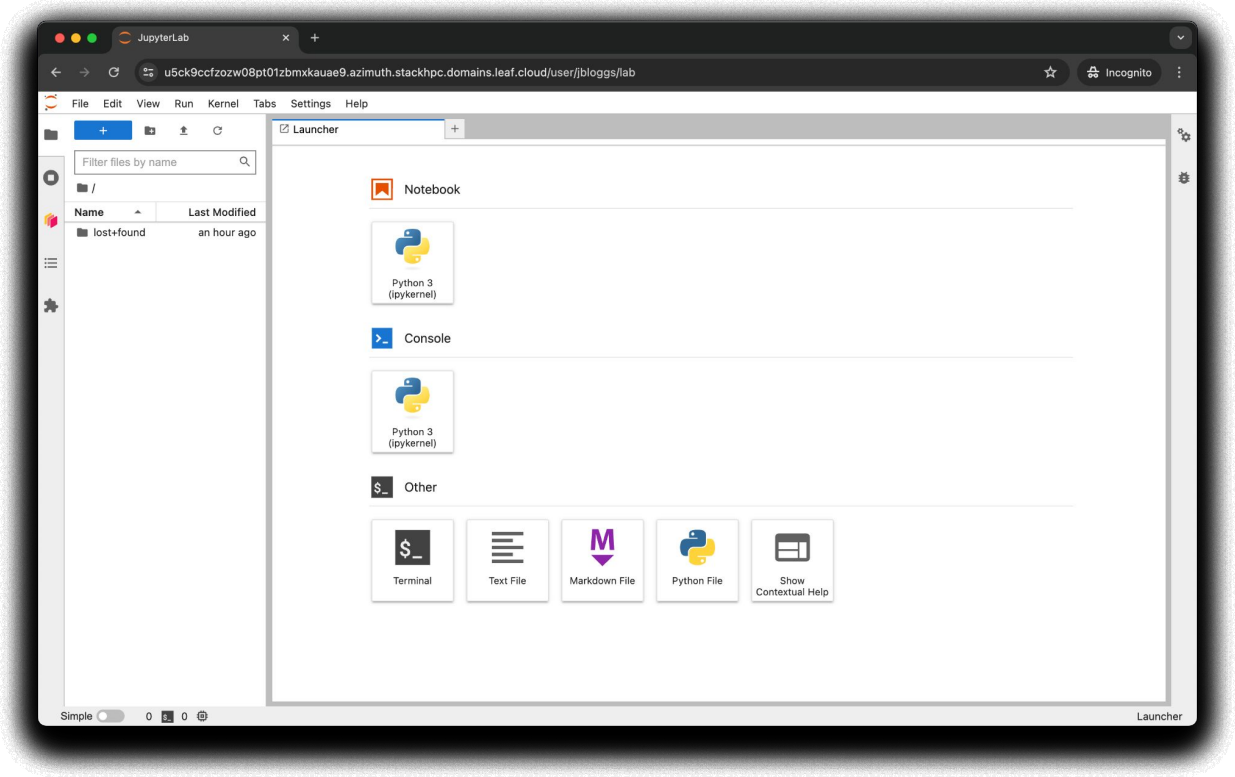
StackHPC

- Built on Cluster API and CAPI Helm charts
- Simple user interface
- HA control plane
- Multiple node groups
- Autoscaling, autohealing, rolling upgrades
- NVIDIA GPU and NIC support
- Kubernetes dashboard
- Monitoring and logging
- Secure access via Zenith



DaskHub

- Runs on Kubernetes
- Apps managed using `HelmRelease` addon resources
- Each user gets their own notebook server
- Secure access via Zenith
- Grant access to external users using tenancy Keycloak realm
- Dask clusters for parallel computing using Dask Gateway



GitOps-managed Kubernetes

StackHPC

What is GitOps?

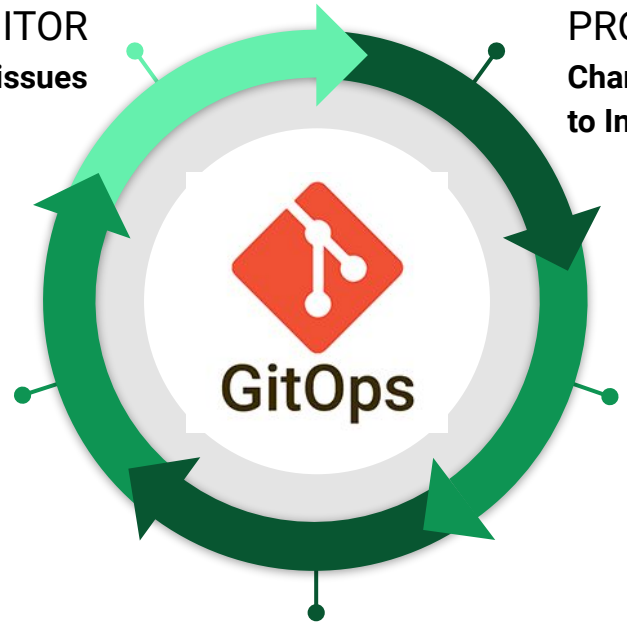
StackHPC

MONITOR

Monitor the running system for issues

PROPOSE CHANGES

Changes proposed via a merge request to Infrastructure-as-Code repo



DEPLOY

Changes applied to staging
Production can be automatic or require manual confirmation

TEST AND REVIEW

Changes tested automatically and reviewed by another human

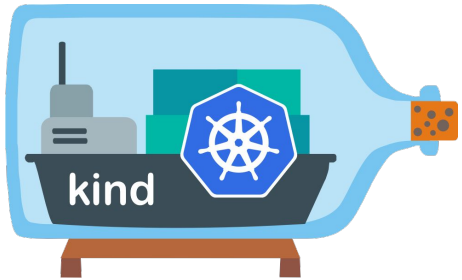
APPROVE AND MERGE

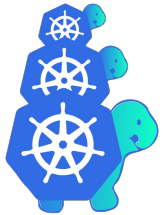
Once approved, changes are merged to main

Self-managed Cluster API cluster

StackHPC

- *Recap: Cluster API deploys clusters by creating resources in a management cluster*
- But - Cluster API clusters can manage themselves!
- Must be bootstrapped using a "pivot" process
  - Provision ephemeral Kubernetes cluster, e.g. kind
  - Install Cluster API controllers on ephemeral cluster
  - Provision a cluster using Cluster API resources on the ephemeral cluster (e.g. using CAPI Helm charts)
  - Install Cluster API controllers on Cluster API cluster
  - Pause reconciliation on ephemeral cluster
  - Move Cluster API resources to Cluster API cluster
  - Resume reconciliation on Cluster API cluster





Kubernetes Cluster API

GitOps-managed Kubernetes cluster

StackHPC

- Cluster API resources are just Kubernetes resources
- Bootstrap a self-managed Cluster API cluster
- Resources created using CAPI Helm charts
- Install Flux CD (or Argo) on cluster
  - Can also be self-managed
- Manage Helm release for cluster using Flux Helm resources or an Argo Application
- Changes to cluster configuration in git applied to cluster by Flux and reconciled by Cluster API

flux

argo

How to get started?

StackHPC

How to get started?

- Azimuth
  - Open-source (Apache 2.0)
  - Can be installed on any OpenStack cloud
  - https://stackhpc.github.io/azimuth-config/try/
- Magnum CAPI Helm driver
  - Open-source, under OpenStack governance
  - Install driver into Magnum Python environment
  - Needs a Cluster API management cluster to point to
  - Requires suitable images to be available
- StackHPC can help!

StackHPC

The Rise of the HPC Cloud

Thank You

https://www.stackhpc.com