Gen 2 OpenStack cloud architecture

9th SIG CISS

Adrian Rosinec, Klara Moravcova

Agenda

- What and why Gen 2 (G2) OpenStack?
- G2 Cloud orchestration
- G2 architecture overview
- G1 to G2 OpenStack migration plans

Compute services at e-INFRA CZ

- Batch compute based on PBS (known as Metacentrum)
- **OpenStack IaaS cloud**
- SensitiveCloud - PaaS based on K8S
- Managed Kubernetes – PaaS based on K8S
- Karolina supercomputer

Cloud services as tool to support research

Motivation for G2 architecture?

- End of life of GEN1 installation from 2016
- Reach much lower operation cost, target is 20%, currently around 70%
- Enable frequent OpenStack updates
- Enable Cloud as a service (to support specialized cloud deployments)
- Improve cloud resiliency
- Tune current OpenStack cloud decisions
  - assignment public networks to projects
  - quota assignments, projects governance
  - improve flavor naming and unify functionality from UI and commandline
  - Improve various parameters of cloud (MTUs, storages, GUI, …)

Who will use it

- e-INFRA CZ is research e-infrastructure
  - 200 research/experiment oriented projects
  - 600 users projects in “free tier” (treated as playground)
  - 40+ international projects (through EGI and ELIXIR)
- Main focus on being HPC cloud
  - large flavors, GPUs, fast storage and networking
- Small portion of resources/support dedicated to standard operation but not HA
  - Small VMs, databases, no advanced features like LBaaS, …

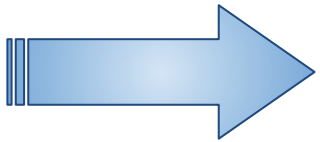
Cloud orchestration

Infrastructure technologies

G1/G2 OpenStack infrastructure technologies

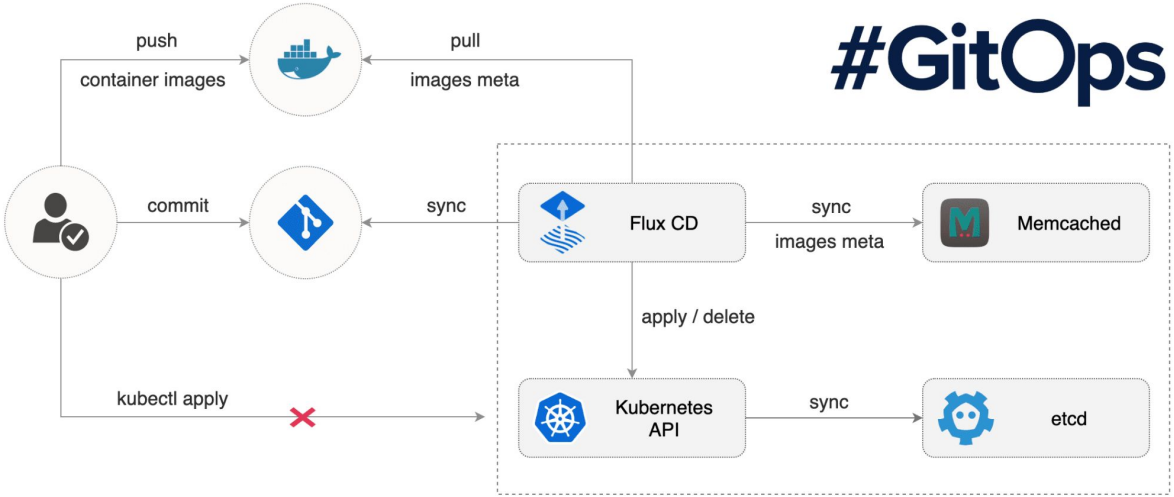
- **G1** Problems:
  - Upstream kolla-ansible split into two parts one in Puppet and the other in Ansible, expensive to track community
  - OpenStack entity life-cycle enforcement - extra code in Ansible

- **G2** Challenges:
  - Stay compatible and in touch with the upstream repositories
  - Manage multiple clouds sustainable GitOps way



G2 OpenStack deployment overview

- Infrastructure is declaratively described in **git** repository which contains set of component releases and configurations
  - Differentiation on helm chart values
- Repository is continuously watched by Flux CD and deployed (server-side) to Kubernetes
  - application life-cycle is guaranteed

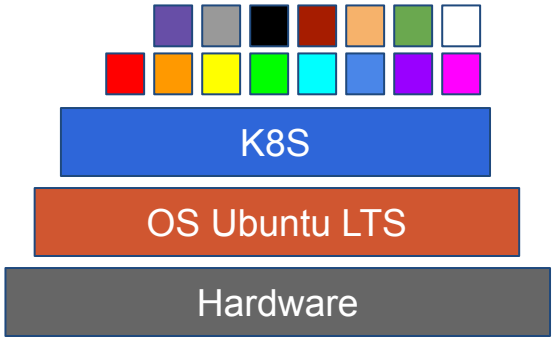


Cloud Architecture

From HW to OpenStack services

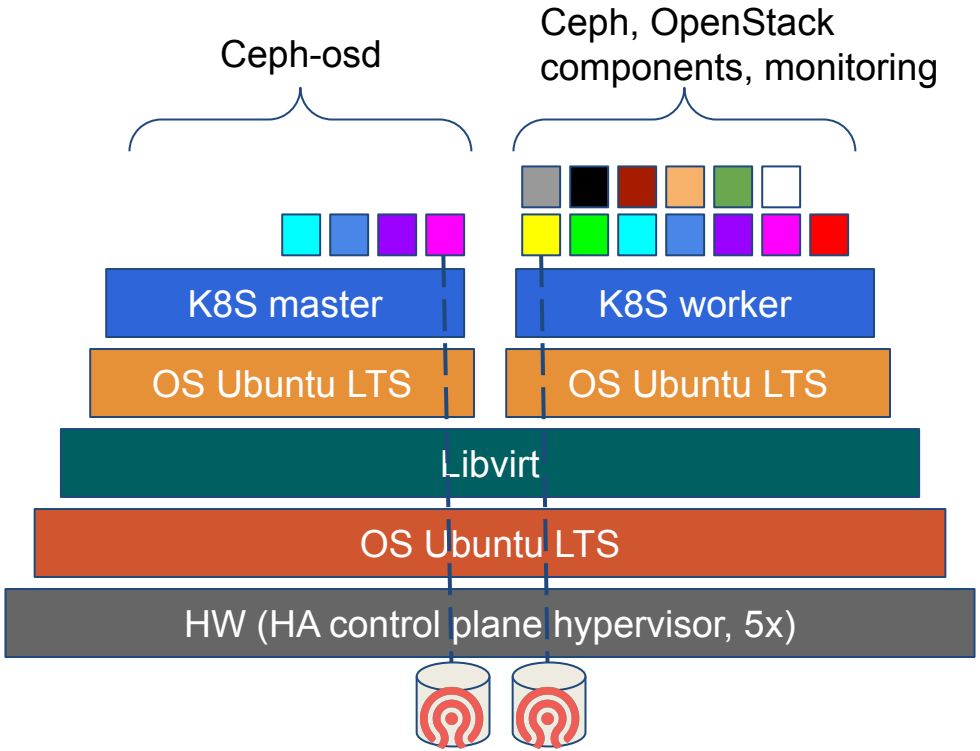
G2 OpenStack architecture overview I

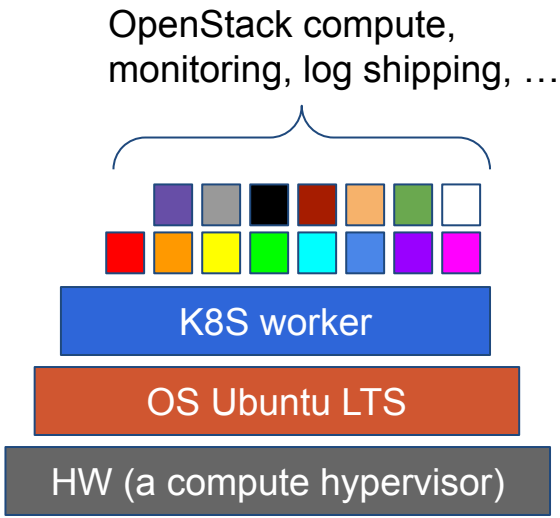
- G2 Cloud comes with **HA control plane**, **ceph nodes** and **compute nodes**
  - **Ceph nodes provide internal ceph network storage** (k8s PV, …)
  - **HA Control plane runs both kubernetes and openstack components**
  - **Compute nodes run openstack compute containers on K8S worker**
- All OpenStack components are built as container images and HELM charts



G2 OpenStack architecture overview II

- G2 Cloud “compact” architecture scheme





G2 OStack Operating System

- **Ubuntu LTS 22.04** as primary operating system
  - Long release support
  - Better tested in Ostack / Kubernetes communities

G2 OStack - Bottom HW, provisioning, initialization I

- **MAAS** - Debian/Ubuntu initial provisioning
  - Initial network configuration
  - Deployment customization using cloud-init
  - Why Canonical MASS?
    - More standard approach than custom scripts

G2 OStack - Bottom HW, provisioning, initialization II

- **Infra-config Ansible** based Debian/Ubuntu provisioning before kubernetes
  - Internal lightweight in-house ansible IaC repository
  - Cleanup after MAAS deployment
  - Basic operating system configuration
  - Ssh keys, networking, packages …
  - Preparation for libvirt VMs, kubespray
  - Storage of node configuration
  - Puppet replacement
  - Firewall rules, VM automated provisioning, etc.

G2 OStack - Kubernetes + HELM + Flux CD

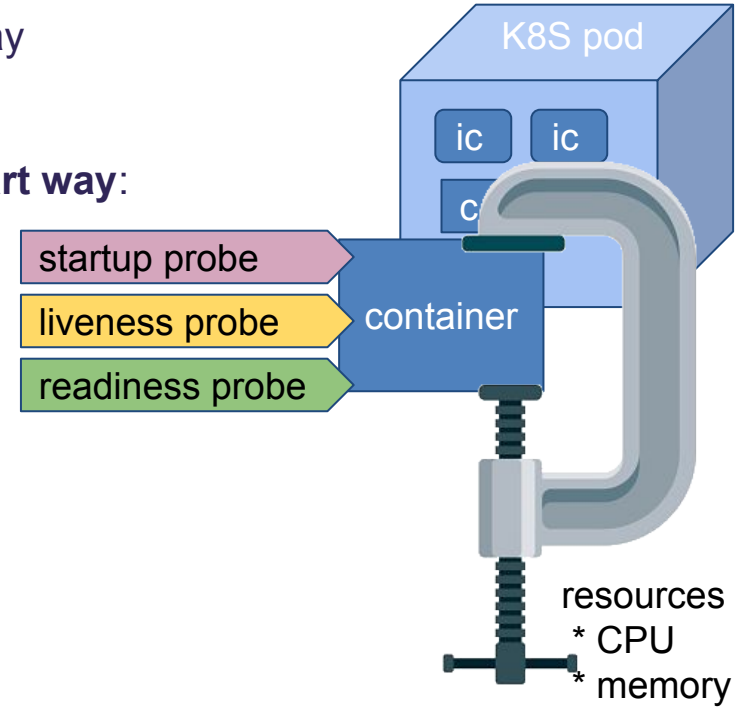
- We use kubespray (Ansible) to deploy vanilla kubernetes on cloud Ubuntu LTS nodes
  - We stick to kubernetes 1.24.x release
- We currently use **kube-vip** addon for kubernetes API & inbound traffic HA
- Kubernetes use internal ceph storage as network persistent storage
- Deployment OpenStack and applications is done via:
  - **HELM**
    - Application packager
  - **Flux CD v2**
    - Continuous Delivery from gitops git repository

kubernetes

flux

G2 OStack - Why Kubernetes under OpenStack?

- OpenStack components are in containers already today (OpenStack-Kolla)
- **K8S provides additional resiliency**.
- **K8S manages container applications state-of-the-art way**:
  - K8S container resources
  - K8S container probes



G2 OStack - Storage improvements

- Internal ceph for platform itself (Glance pool?)
- External “fast” ceph - SSD NVMe pool
- External “slow” ceph, performance will be improved by ceph extension + ceph version upgrade (Nautilus -> Octopus)
- Planned: 100Gbps networking & Flash SSD pool

G1 to G2 migration plans

- Migration G1 -> G2
  - Migration guide will be published, will consist of steps coordinated with HW movement from G1 -> G2
  - Cloud team will be assisting with the migration

Takeaways

- G2 OpenStack cloud uses fresh technologies, shift towards Ubuntu LTS
- G2 OpenStack cloud architecture is build on top of Kubernetes
- We maximize using open-source project (95%) and create minimum code ourselves (5%)
  - We plan to maintain OpenStack much closer to the open-source upstream than before
- We learn from difficulties / misconfigurations we ended up with G1 OpenStack cloud

Thank you for your attention!
Questions?

Please contact us {adrian, moravcova}@cesnet.cz