SITE UPDATE SURF

THE DAPHNE CLUSTER

SANDER ZIJLSTRA

16/09/2022

Who we are

- **SURF**: Since 01/10/2020 SURFsara, SURFnet, SURFmarket & SURFburo are SURF
  - *SURF is the collaborative organisation for IT in Dutch education and research*
- The **DAPHNE** Openstack cluster is aimed at offering **IaaS** resources to internal service teams which provide services to end-users (researchers mainly).
- We're part of SURF's **Research Development** domain which is focused on data processing, managed services for researchers, storage solutions and data analytics on private infrastructure and commercial cloud (AWS).
- Within the Research Development domain, we are members of the **Distributed Data Processing** team (DDP).

What we have

- **DAPHNE** is a collection of separated Openstack instances which we call regions, but they do not share Keystone.
  - Production-01 : 238 Compute nodes and ~14k cores, 90 GPUs
  - Production-02: 85 Compute nodes and ~5k cores, 80 GPUs
  - Small service cluster: 10 Compute nodes with local storage only
  - 2 tiny clusters for development and staging activities
  - CEPH storage cluster (+ceph-fs): 13 PB

What we have

- Some **DAPHNE** specifics:
  - OS Release: Rocky (2019) -> Stein (2020) -> Train (2022)
  - Containerized services (kolla) on CentOS
  - Networking:
    - OVS
    - DVR is enabled
    - Jumbo frames on all networks
  - Multi domain: service teams are assigned a domain for their resources
  - CEPH storage by default, but many hypervisors are set to using local NVMe; especially for the GRID clusters
  - CEPHfs is provided directly to one GRID cluster, no use of Manilla (yet)

Who we serve

- SURF Research Cloud: self-service cloud services (~PaaS)
- Spider: Grid cluster(s) tailored for specific users or use cases.
- Gina: Grid compute cluster (slurm)
- MS4: managed services for various small & big projects, among:
  - iRODS & Yoda
- SDA: data analytics platforms based on K8S
- Internal use cases:
  - Small development environments
  - Proof of concept environments
  - Staging for software upgrade testing

What we use

- Git: all code and automated deployment setups
- Foreman: bare metal deployment
- Terraform: user project management, cluster configurations
- Ansible (+Kolla): bare-metal provisioning & Openstack
- Zabbix: monitoring
- Grafana: trends
- Rally: Testing Openstack during changes

What we are doing

- Current activities:
  - CentOS 7 -> CentOS Stream 8
  - K8SaaS -> cluster-api (capi)
  - L3VPN
  - Regular BAU:
    - CEPH Cluster upgrade & new CEPH cluster
    - Hardware replacement and additions
- Planned activities (short future):
  - Upgrade to Ussuri/Victoria
  - Ironic

What we are doing

- Planned/Interested-in activities (mid to long term):
  - OoK8S
- Future (long-term)
  - Routing to the host
  - Multi-site
  - OVN

How we deal with

- Upgrades:
  - In-place upgrades using kolla-ansible
- Accounting:
  - Limited to none, internal users are diverse and each handle it differently on their own.
  - Internal users budget for a #Cores , Storage and/or #GPUs which is then “billed”
- User management:
  - Internal teams are assigned a domain and a GitLab project which manages projects, quota and users using Terraform. We approve the MR and push the pipeline that runs the Terraform apply.
- Security:
  - regular patching
  - port scanning & connection monitoring
  - FW and/or ACL in front of API’s and hosts next to Security Groups

SURF SARA

What we struggle with

- Neutron:
  - We have had our share of “strange” networking issues and even though we more or less know packet flow and related configurations we do not seem to get grip when certain issues occur.
    - We had issues where MTU’s seem to be “reset” from jumbo but we couldn’t find where. Moving the router solved the problem.
    - We had issues with connections breaking due to faulty MAC addresses constantly being programmed into the OVS bridge to time out later on. Again by moving the associated instances, it was solved.
  - No associated upstream bugs were found which seemed related.
  - So we (still) miss some knowledge in the OVS flows, network namespaces and all the other parts which make up the Neutron networking.

SURF SARA

What we struggle with

- Maintenance:
  - We can live migrate large parts of our workload, next to instances with local storage which can’t.
  - But still live-migration doesn’t always work when dealing with CPU differences. AMD->Intel doesn’t work ofcourse but we also seem to have issues between daul-socket and single-socket hosts.
    - Except for GPU nodes we are standardized on AMD currently.
  - We use the `host-model` setting in Libvirt as most users want a CPU as close to the actual one even on regular cloud instances.
  - Compute hosts serving instances with GPUs and for GRID usage have `host-passthrough`, so migration is even harder, hence we do not offer that.
- How do you deal with this??

What we struggle with

- Domain admins:
  - We struggle a bit with adding projects/users and setting quota, flavor access, network RBAC etc
  - Everything is managed using terraform currently, but manual.
  - We want to improve on this and are looking into a **domain admin default policy**, and level-2 keystone quotas,
    - no idea for flavor and network access
    - no quota per flavor/aggregate is really missing
- Any ideas or input how to solve this??