e-INFRA CZ

Beskar Cloud: Openstack deployment on top of Kubernetes

Adrian Rosinec <adrian@ics.muni.cz>
Cloud Engineer at CESNET

13.11.2023, 12th SIG-CISS

cesnet

MUNI
CERIT-SC

VŠB TECHNICKÁ UNIVERZITA OSTRAVA | IT4INNOVATIONS NÁRODNÍ SUPERPOČÍTAČOVÉ CENTRUM

Agenda

- Cloud compute service
- Motivation for the new architecture
- OpenStack distribution overview
- **Current status**

Cloud compute service

- e-INFRA CZ is national research e-infrastructure
  - 200 research/experiment oriented allocations
  - 600 users projects in “free tier” (treated as playground)
  - **50+ international projects (through EGI and ELIXIR)**
- 300 HV, 10K CPU, 200TB RAM
- Main focus on being “HPC cloud”
  - large flavors (up to 128 CPU), GPUs (NVIDIA A40), fast storage (local NVMEs) and networking
- Portion of resources/support dedicated to standard operation
  - Small VMs, databases, websites + features like LBaaS, …
  - Nowadays not preferred - VMs are “heavy tool”

Motivation for the new architecture

- End of life of GEN1 installation from 2016
- Custom made solution for configuring OpenStack
  - “puppet-kolla” = not supported by community
  - Openstack kolla orchestrated by puppet
- Enable Cloud as a service (to support specialized cloud deployments, BYOC)
- Improve cloud resiliency, frequent updates
- Add second location in Czechia (Supercomputer cloud nodes in Ostrava, CZ)
- Get ready for ISO 27K and Health data
  - Requirements for auditability, change management, …

National GEO context

- Brno & Ostrava – working cloud locations
- Prague – maybe in the future

Prague
Ostrava
Brno

Partnership with commercial partner TAIKUN

- Taikun Cloud,
- Since 2018, based in Prague, Czech Republic
- Main product Taikun.cloud
  - SaaS DevSecOps platform
  - To manage OpenStack / Kubernetes clusters
- Members of Cloud native foundation
- Utilizing OpenStack for themselves, and building OS for customers
  - motivation to build low-effort OpenStack management

OpenStack distribution as a result

- The cloud way of orchestrating OpenStack services in containers
- Based on **open-source projects** (no inhouse development of new tools)
  - MAAS, Ubuntu OS
  - Ansible (infra-config)
  - Kubernetes (from Ansible Kubespray)
  - Openstack-helm and openstack-helm-infra
  - FluxCD
  - Prometheus, Grafana,..
  - CEPH storage
  - OpenStack Entity Management (declarative way)
- Getting all of those projects working can be tedious, in **one repository** Beskar.cloud is set of
  - Values for kubespray
  - Settings for infra-config
  - Helm charts for useful components
  - FluxCD recommended settings

Supported by the community

- The cloud way of building and making stuff in recent years
- Sharing experience from operating the Beskar.cloud OpenStack
- Building shared knowledge base
- Regular community calls
- Discord as one of communication channels (WIP)

What’s in the stack?

- Bare–metal provisioning
  - **MAAS** - install OS, basic networking, …
- Server provisioning
  - **Ansible (infra-config, kubespray)**
- Infrastructure management
  - **Kubernetes**
    - stable orchestrator / workload distributor of OpenStack components
  - **HELM** + **Flux CD**
    - App configuration converted into HELM values
    - App deployment described declaratively
- OpenStack entity management
  - **Terraform**
- Monitoring and logging stack
  - **Prometheus and Grafana**
- Storage backend
  - **CEPH cluster (for undercloud and cloud)**

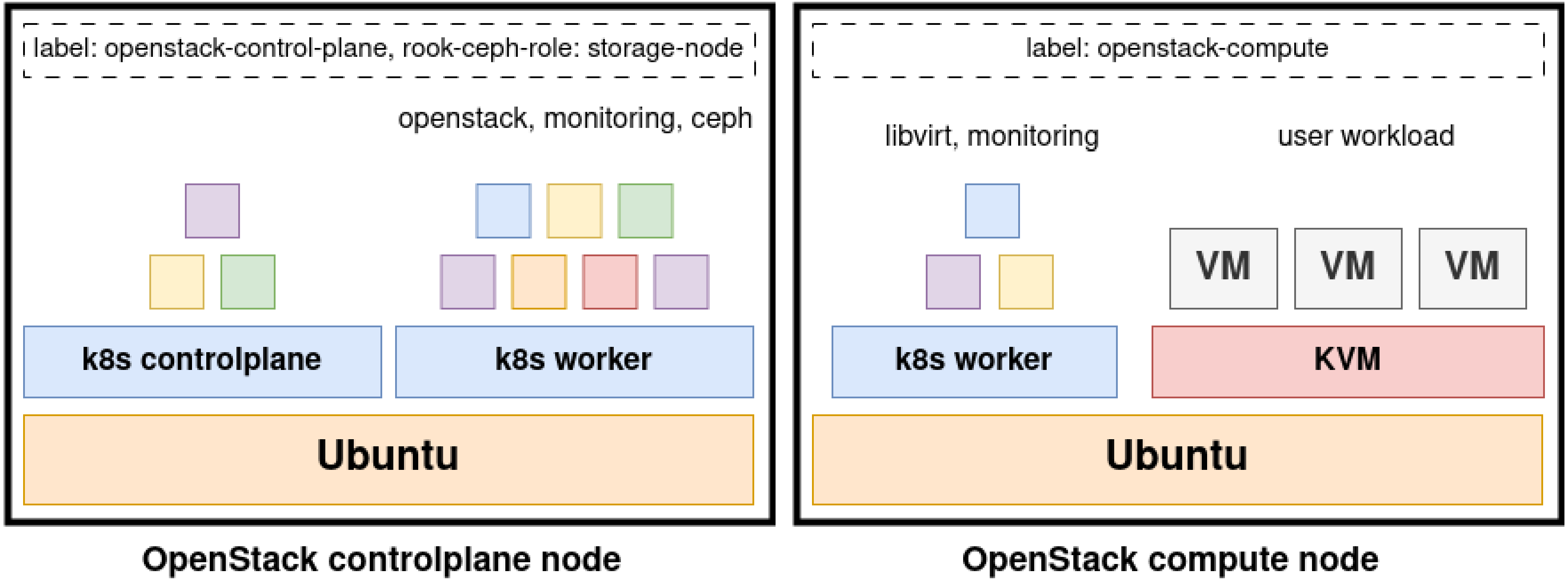
Cloud Architecture

From HW to OpenStack services

Architecture

- Kubernetes installed on every hypervisor
  - 2 types of hypervisor workload:
    - **Compute**
      - Running **K8S worker** with *Nova, OVS, …*
      - Virtualisation service (KVM, …) to run VMs
      - *Containerized **libvirt** with mounted node's KVM socket*
    - **Control plane**
      - Running **K8S controller** with *Horizon, Heat, Cinder, Keystone, Prometheus, …*
      - Ingress to publish HTTPS - Openstack API / Dashboard

Architecture



Site management

And key principles

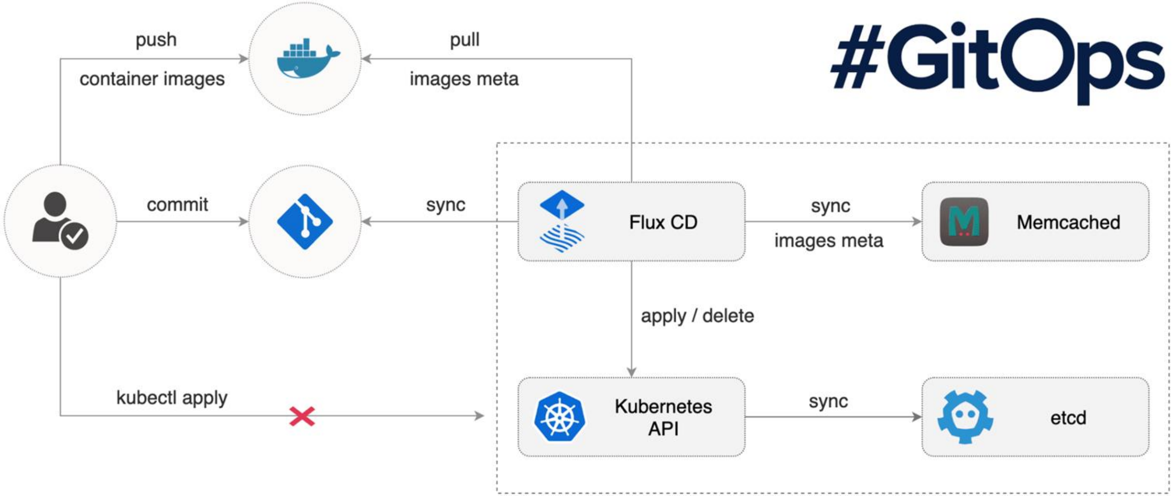
Pre-GitOps era

- Infrastructure management is done
  - **manually by administrators**
  - via set of **custom scripts**

- Problems:
  - **Configuration file duplicities**
  - **Lack of automation**
  - **Non-standard custom management approaches**
  - **Secret management**
  - **Manual life-cycle management**

Managing the site

- Infrastructure is declaratively described in git:
  - K8S cluster definition and all manifests
  - Helm charts for OpenStack components + values
  - OpenStack entities
- Repository is continuously watched by Flux CD and deployed (server-side) to Kubernetes



#GitOps

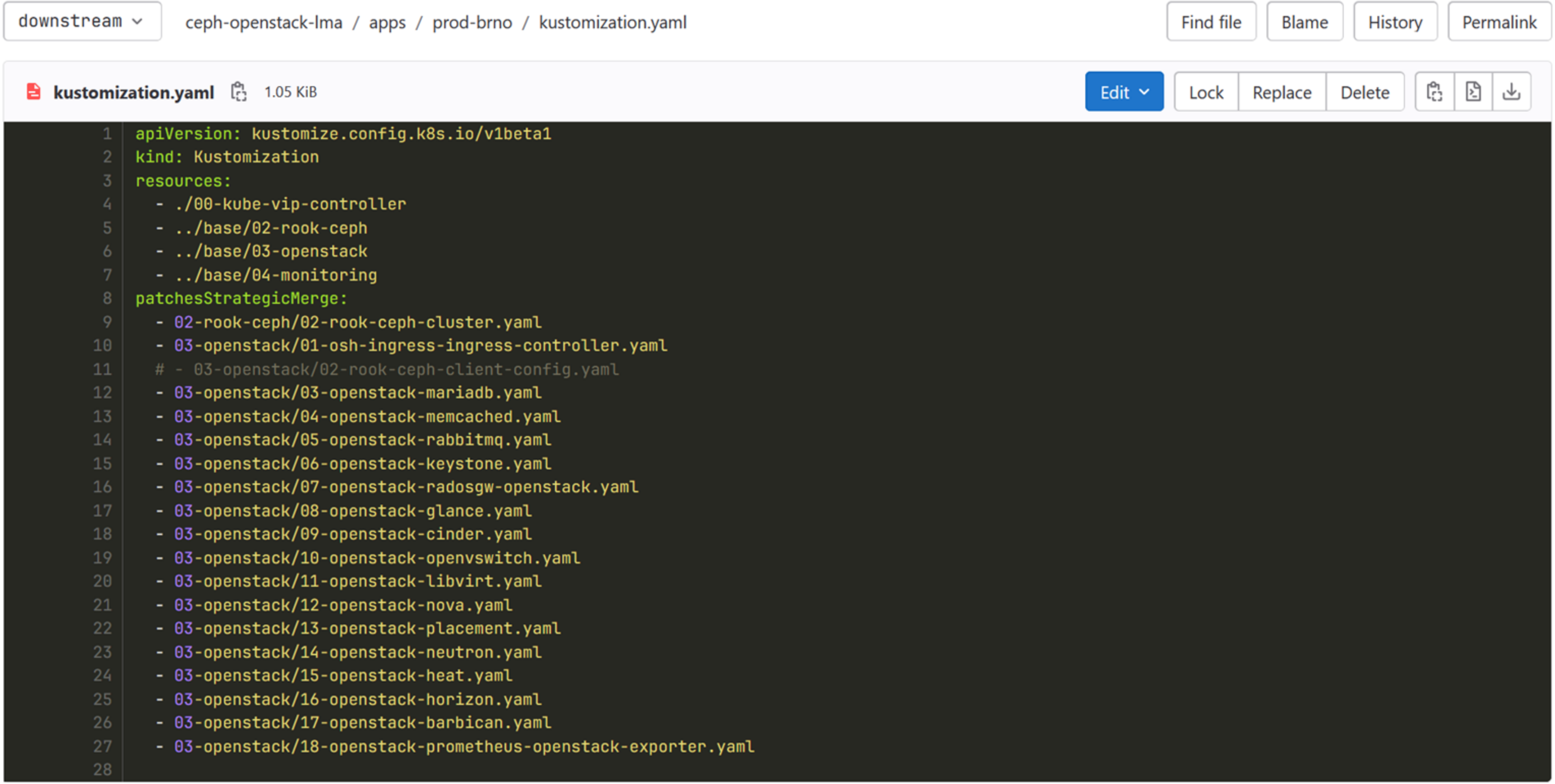
Git repo structure

```
├── apps
│   ├── base
│   ├── demo-cluster-10a
│   ├── demo-cluster-10b
│   ├── prod-brno
│   ├── prod-brno-obsoleted
│   └── prod-ostrava
├── clusters
│   ├── README
│   ├── demo-cluster-10a
│   ├── demo-cluster-10b
│   ├── prod-brno
│   ├── prod-brno-obsoleted
│   └── prod-ostrava
├── infrastructure
│   ├── 00-sources
│   ├── 01-cluster-ingress
│   ├── 02-debugger
│   └── kustomization.yaml
```

Git repo structure

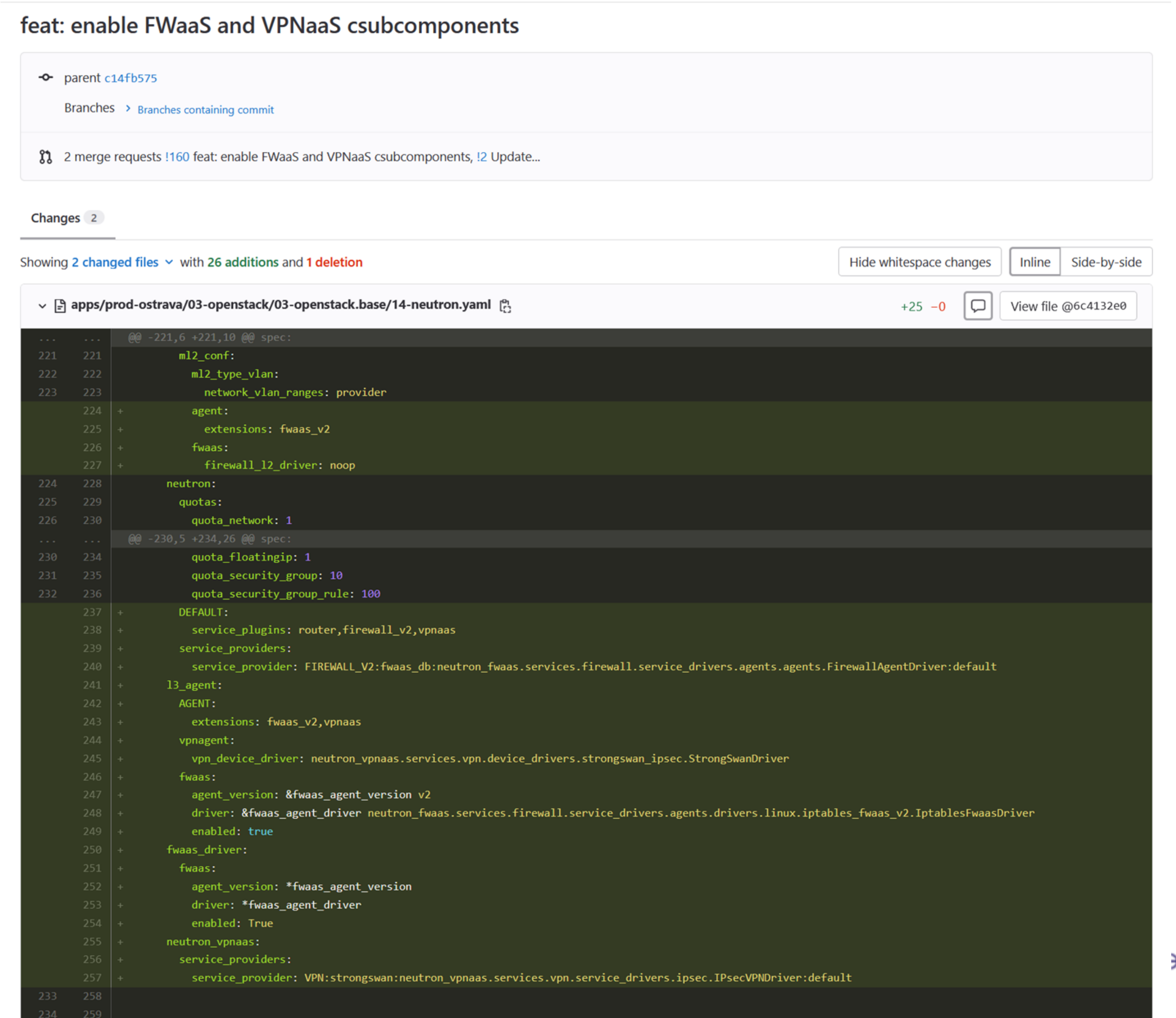
```
apps/prod-brno
├── 01-support
│   ├── 01-kube-vip-controller
│   ├── 02-edge-proxy
│   ├── 03-welcome-hub
│   ├── 04-gitops-server
│   └── kustomization.yaml
├── 02-ceph
│   ├── 01-rook-ceph
│   ├── 02-rook-ceph-external
│   └── kustomization.yaml
├── 03-openstack
│   ├── 03-openstack.base
│   ├── 03-openstack.specific
│   └── kustomization.yaml
└── 04-lma
           ├── 04-lma.base
           └── kustomization.yaml
```

```
apps/prod-brno/03-openstack
├── 03-openstack.base
│   ├── 00-common-configmap.yaml
│   ├── 00-common-encryptedsecret.yaml
│   ├── 00-namespace.yaml
│   ├── 01-ingress-controller.yaml
│   ├── 02-ceph-cluster-config.yaml
│   ├── 03-mariadb-backup.yaml
│   ├── 03-mariadb.yaml
│   ├── 04-memcached.yaml
│   ├── 05-rabbitmq.yaml
│   ├── 06-esaco.yaml
│   ├── 06-keystone-apache-oidc-metadata-encryptedsecret.yaml
│   ├── 06-keystone.yaml
│   ├── 08-ceph-client-glance-key-images-rbd-keyring-encryptedsecret.yaml
│   ├── 08-glance.yaml
│   ├── 09-ceph-client-cinder-backup-rbd-keyring-encryptedsecret.yaml
│   ├── 09-ceph-client-cinder-volume-rbd-keyring-encryptedsecret.yaml
│   ├── 09-cinder.yaml
│   └── kustomization.yaml
├── 03-openstack.specific
│   ├── 03-mariadb.yaml
│   ├── 05-rabbitmq.yaml
│   └── 08-glance.yaml
└── kustomization.yaml
```



```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
  - ./00-kube-vip-controller
  - ../base/02-rook-ceph
  - ../base/03-openstack
  - ../base/04-monitoring
patchesStrategicMerge:
  - 02-rook-ceph/02-rook-ceph-cluster.yaml
  - 03-openstack/01-osh-ingress-ingress-controller.yaml
  # - 03-openstack/02-rook-ceph-client-config.yaml
  - 03-openstack/03-openstack-mariadb.yaml
  - 03-openstack/04-openstack-memcached.yaml
  - 03-openstack/05-openstack-rabbitmq.yaml
  - 03-openstack/06-openstack-keystone.yaml
  - 03-openstack/07-openstack-radosgw-openstack.yaml
  - 03-openstack/08-openstack-glance.yaml
  - 03-openstack/09-openstack-cinder.yaml
  - 03-openstack/10-openstack-openvswitch.yaml
  - 03-openstack/11-openstack-libvirt.yaml
  - 03-openstack/12-openstack-nova.yaml
  - 03-openstack/13-openstack-placement.yaml
  - 03-openstack/14-openstack-neutron.yaml
  - 03-openstack/15-openstack-heat.yaml
  - 03-openstack/16-openstack-horizon.yaml
  - 03-openstack/17-openstack-barbican.yaml
  - 03-openstack/18-openstack-prometheus-openstack-exporter.yaml

```



feat: enable FWaaS and VPNaaS csubcomponents

parent c14fb575

Branches › Branches containing commit

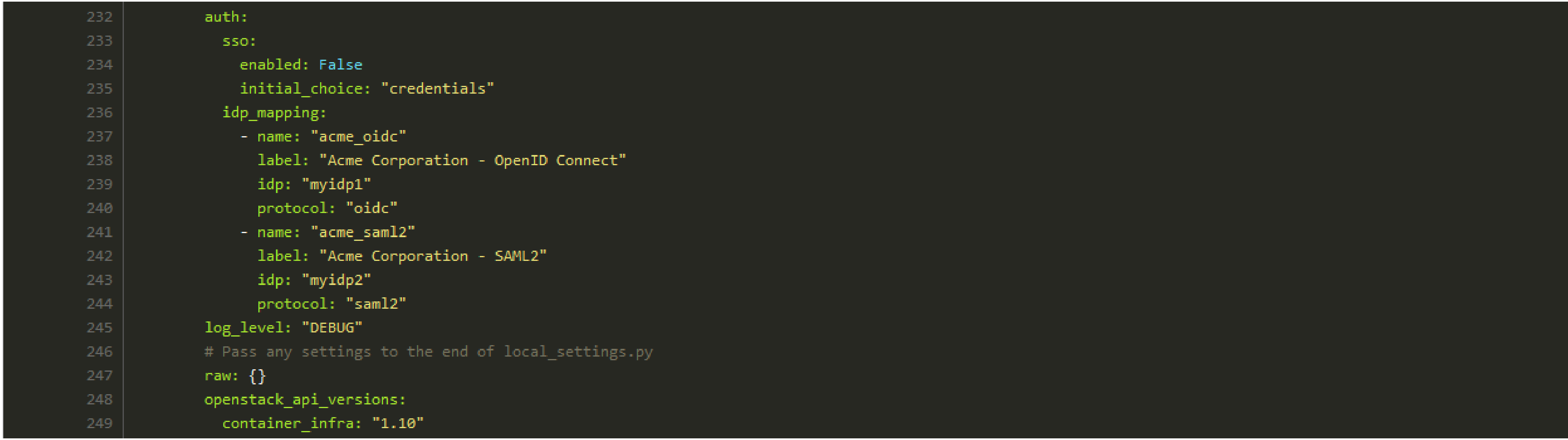
2 merge requests !160 feat: enable FWaaS and VPNaaS csubcomponents, !2 Update...

Showing 2 changed files with 26 additions and 1 deletion

apps/prod-ostrava/03-openstack/03-openstack.base/14-neutron.yaml +25 −0

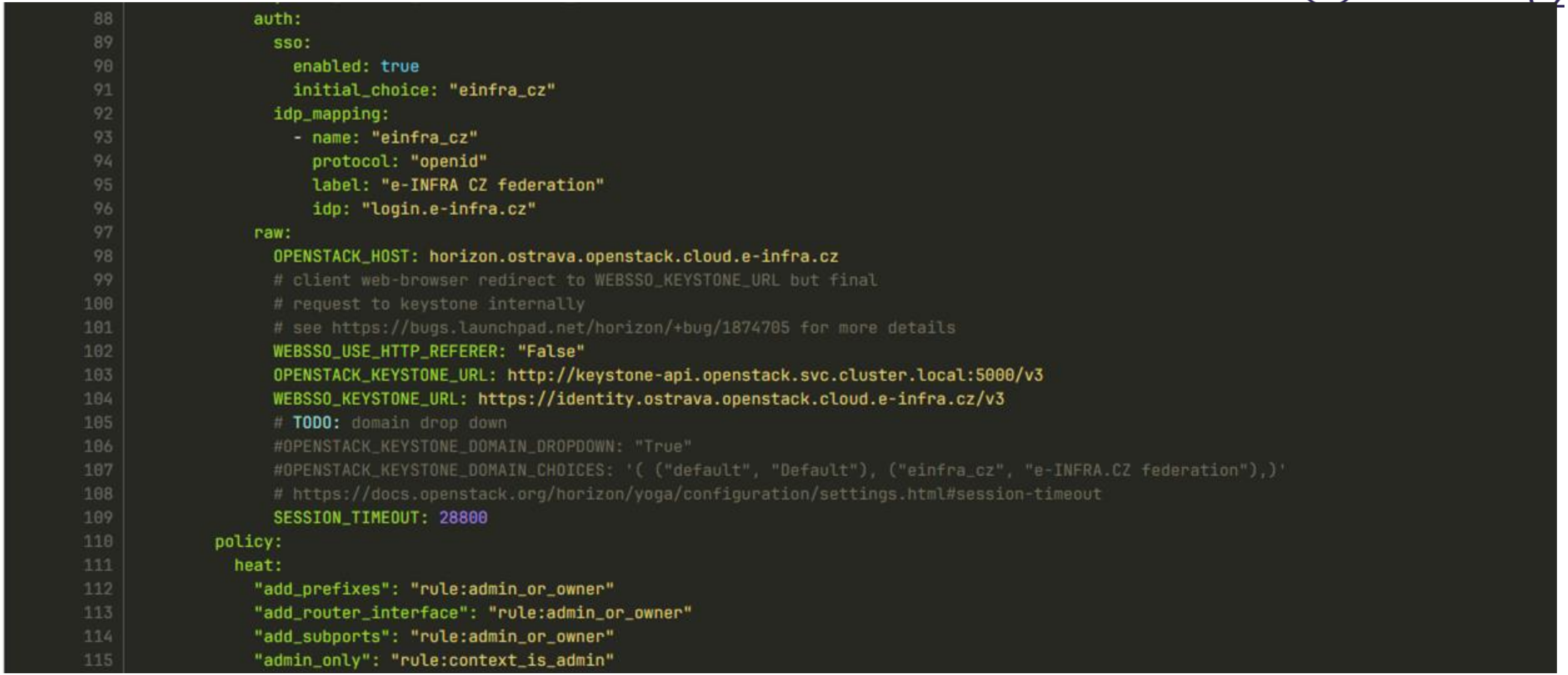
```diff
@@ -221,6 +221,10 @@ spec:
       ml2_conf:
         ml2_type_vlan:
           network_vlan_ranges: provider
+        agent:
+          extensions: fwaas_v2
+        fwaas:
+          firewall_l2_driver: noop
     neutron:
       quotas:
         quota_network: 1
@@ -230,5 +234,26 @@ spec:
         quota_floatingip: 1
         quota_security_group: 10
         quota_security_group_rule: 100
+      DEFAULT:
+        service_plugins: router,firewall_v2,vpnaas
+      service_providers:
+        service_provider: FIREWALL_V2:fwaas_db:neutron_fwaas.services.firewall.service_drivers.agents.agents.FirewallAgentDriver:default
+    l3_agent:
+      AGENT:
+        extensions: fwaas_v2,vpnaas
+      vpnagent:
+        vpn_device_driver: neutron_vpnaas.services.vpn.device_drivers.strongswan_ipsec.StrongSwanDriver
+      fwaas:
+        agent_version: &fwaas_agent_version v2
+        driver: &fwaas_agent_driver neutron_fwaas.services.firewall.service_drivers.agents.drivers.linux.iptables_fwaas_v2.IptablesFwaasDriver
+        enabled: true
+    fwaas_driver:
+      fwaas:
+        agent_version: *fwaas_agent_version
+        driver: *fwaas_agent_driver
+        enabled: True
+    neutron_vpnaas:
+      service_providers:
+        service_provider: VPN:strongswan:neutron_vpnaas.services.vpn.service_drivers.ipsec.IPsecVPNDriver:default
```

Default Horizon HELM values



```
        auth:
          sso:
            enabled: False
            initial_choice: "credentials"
          idp_mapping:
            - name: "acme_oidc"
              label: "Acme Corporation - OpenID Connect"
              idp: "myidp1"
              protocol: "oidc"
            - name: "acme_saml2"
              label: "Acme Corporation - SAML2"
              idp: "myidp2"
              protocol: "saml2"
        log_level: "DEBUG"
        # Pass any settings to the end of local_settings.py
        raw: {}
        openstack_api_versions:
          container_infra: "1.10"
```

e-INFRA CZ modification



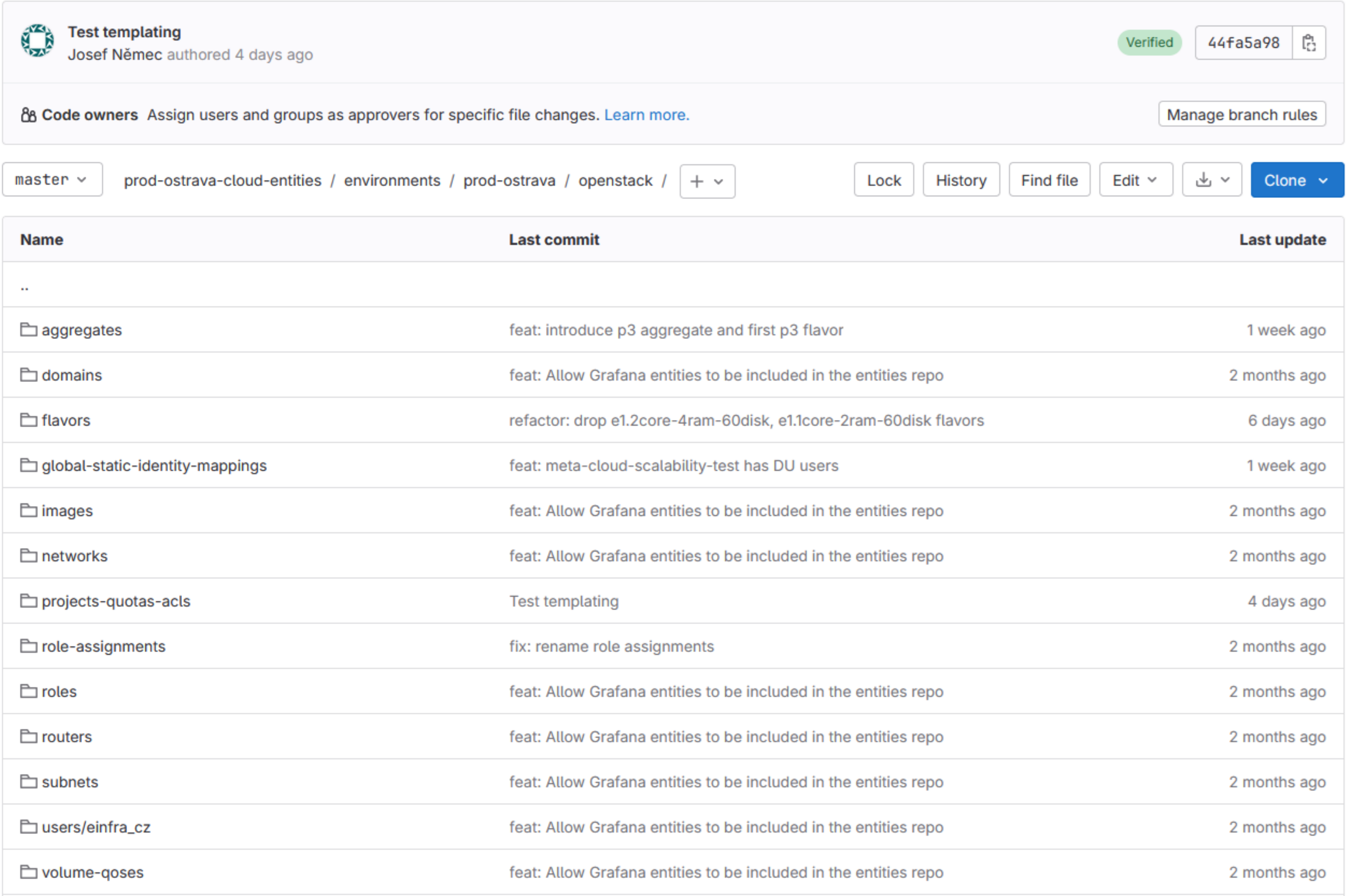
```
    auth:
      sso:
        enabled: true
        initial_choice: "einfra_cz"
      idp_mapping:
        - name: "einfra_cz"
          protocol: "openid"
          label: "e-INFRA CZ federation"
          idp: "login.e-infra.cz"
    raw:
      OPENSTACK_HOST: horizon.ostrava.openstack.cloud.e-infra.cz
      # client web-browser redirect to WEBSSO_KEYSTONE_URL but final
      # request to keystone internally
      # see https://bugs.launchpad.net/horizon/+bug/1874705 for more details
      WEBSSO_USE_HTTP_REFERER: "False"
      OPENSTACK_KEYSTONE_URL: http://keystone-api.openstack.svc.cluster.local:5000/v3
      WEBSSO_KEYSTONE_URL: https://identity.ostrava.openstack.cloud.e-infra.cz/v3
      # TODO: domain drop down
      #OPENSTACK_KEYSTONE_DOMAIN_DROPDOWN: "True"
      #OPENSTACK_KEYSTONE_DOMAIN_CHOICES: '( ("default", "Default"), ("einfra_cz", "e-INFRA.CZ federation"),)'
      # https://docs.openstack.org/horizon/yoga/configuration/settings.html#session-timeout
      SESSION_TIMEOUT: 28800
policy:
  heat:
    "add_prefixes": "rule:admin_or_owner"
    "add_router_interface": "rule:admin_or_owner"
    "add_subports": "rule:admin_or_owner"
    "admin_only": "rule:context_is_admin"
```

Entities management

Single source of truth in git repo

Terraform as orchestrator

- **Reproducibility and auditing**
  - Admins are not using OpenStack CLI/API and custom scripts
  - Modification of **cluster-wide** settings are done using commits and PR with reviews
- Terraform OpenStack Provider
  - Developed and maintained by the OpenStack community
- Managed entities:
  - Flavors
  - Networks
  - Host aggregates
  - Images
  - Keystone - domains, projects, …



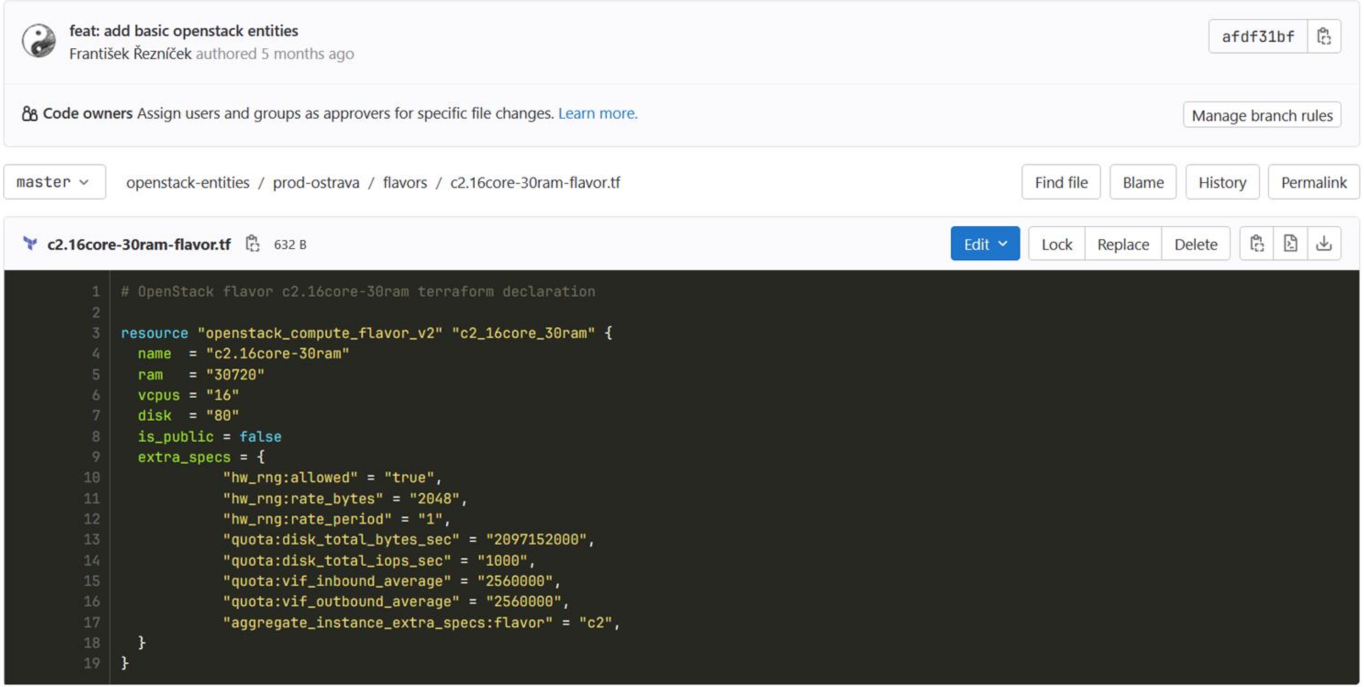
Test templating
Josef Němec authored 4 days ago

44fa5a98

Code owners Assign users and groups as approvers for specific file changes. Learn more.

prod-ostrava-cloud-entities / environments / prod-ostrava / openstack /

Name	Last commit	Last update
..		
aggregates	feat: introduce p3 aggregate and first p3 flavor	1 week ago
domains	feat: Allow Grafana entities to be included in the entities repo	2 months ago
flavors	refactor: drop e1.2core-4ram-60disk, e1.1core-2ram-60disk flavors	6 days ago
global-static-identity-mappings	feat: meta-cloud-scalability-test has DU users	1 week ago
images	feat: Allow Grafana entities to be included in the entities repo	2 months ago
networks	feat: Allow Grafana entities to be included in the entities repo	2 months ago
projects-quotas-acls	Test templating	4 days ago
role-assignments	fix: rename role assignments	2 months ago
roles	feat: Allow Grafana entities to be included in the entities repo	2 months ago
routers	feat: Allow Grafana entities to be included in the entities repo	2 months ago
subnets	feat: Allow Grafana entities to be included in the entities repo	2 months ago
users/einfra_cz	feat: Allow Grafana entities to be included in the entities repo	2 months ago
volume-qoses	feat: Allow Grafana entities to be included in the entities repo	2 months ago



feat: add basic openstack entities

František Řezníček authored 5 months ago

afdf31bf

Code owners Assign users and groups as approvers for specific file changes. Learn more.

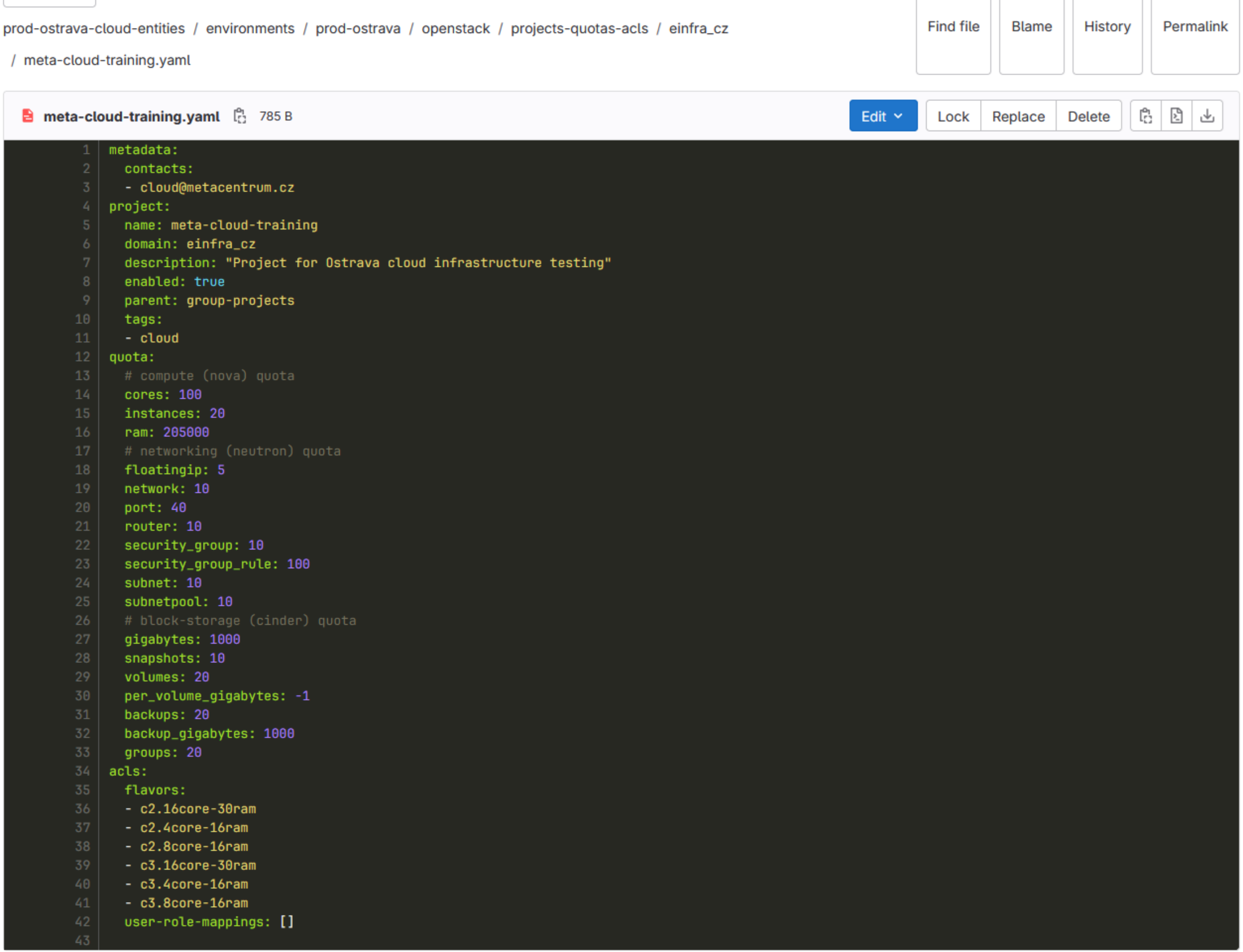
master

openstack-entities / prod-ostrava / flavors / c2.16core-30ram-flavor.tf

c2.16core-30ram-flavor.tf 632 B

```
# OpenStack flavor c2.16core-30ram terraform declaration

resource "openstack_compute_flavor_v2" "c2_16core_30ram" {
  name  = "c2.16core-30ram"
  ram   = "30720"
  vcpus = "16"
  disk  = "80"
  is_public = false
  extra_specs = {
            "hw_rng:allowed" = "true",
            "hw_rng:rate_bytes" = "2048",
            "hw_rng:rate_period" = "1",
            "quota:disk_total_bytes_sec" = "2097152000",
            "quota:disk_total_iops_sec" = "1000",
            "quota:vif_inbound_average" = "2560000",
            "quota:vif_outbound_average" = "2560000",
            "aggregate_instance_extra_specs:flavor" = "c2",
  }
}
```



prod-ostrava-cloud-entities / environments / prod-ostrava / openstack / projects-quotas-acls / einfra_cz / meta-cloud-training.yaml

meta-cloud-training.yaml

```yaml
metadata:
  contacts:
  - cloud@metacentrum.cz
project:
  name: meta-cloud-training
  domain: einfra_cz
  description: "Project for Ostrava cloud infrastructure testing"
  enabled: true
  parent: group-projects
  tags:
  - cloud
quota:
  # compute (nova) quota
  cores: 100
  instances: 20
  ram: 205000
  # networking (neutron) quota
  floatingip: 5
  network: 10
  port: 40
  router: 10
  security_group: 10
  security_group_rule: 100
  subnet: 10
  subnetpool: 10
  # block-storage (cinder) quota
  gigabytes: 1000
  snapshots: 10
  volumes: 20
  per_volume_gigabytes: -1
  backups: 20
  backup_gigabytes: 1000
  groups: 20
acls:
  flavors:
  - c2.16core-30ram
  - c2.4core-16ram
  - c2.8core-16ram
  - c3.16core-30ram
  - c3.4core-16ram
  - c3.8core-16ram
  user-role-mappings: []

```

Beskar is deployed

- e-INFRA CZ
  - Primary deployment in second datacenter of Czech Republic (in Ostrava)
  - 30 HV, part of **CZ Karolina Supercomputer Cluster**
- TAIKUN
  - Hosting internal services and SaaS platform for customers on Beskar.cloud

TODO for Q1-2/2024

- At e-INFRA CZ
  - Migration of Brno site (current; the one with 300HV)
  - as seamless as possible
  - without user interaction if possible
  - Shared CEPH cluster for seamless migrations (moving VMs from “old” to “new”)
  - Migration of OpenStack database (keeping projects/domains/…)
  - Shared network pool
- Ostrava and Brno built as distinct sites
  - Primary motivation - to simplify deployment - no shared/synced OpenStack databases between two(or more) cities
  - User will have options to use it as multi-cloud rather than “regions/availability zones”

Prague
Ostrava
Brno

This is the cloud way!

- The Beskar.cloud OpenStack distribution
  - set of **open-source projects**, documented and **prepared to deploy OpenStack** cluster
- Community endeavour
  - **share and unify** experience of **operating and managing** OpenStack clusters
- Deploy OpenStack cloud of any size using Beskar.cloud distribution
- Documentation and code:
  https://github.com/beskar-cloud/

- Contact us if questions at cloud@metacentrum.cz or adrian@ics.muni.cz