VRIJE UNIVERSITEIT BRUSSEL

5TH VLAIO TETRA OPENCLOUDEDGE USER MEETING

INFRASTRUCTURE CHANGES & SOFTWARE UPGRADES

25 February 2022

Steffen Thielemans & Luca Gattobigio – VUB OpenCloudEdge team

AGENDA

OpenCloudEdge hands-on workshops

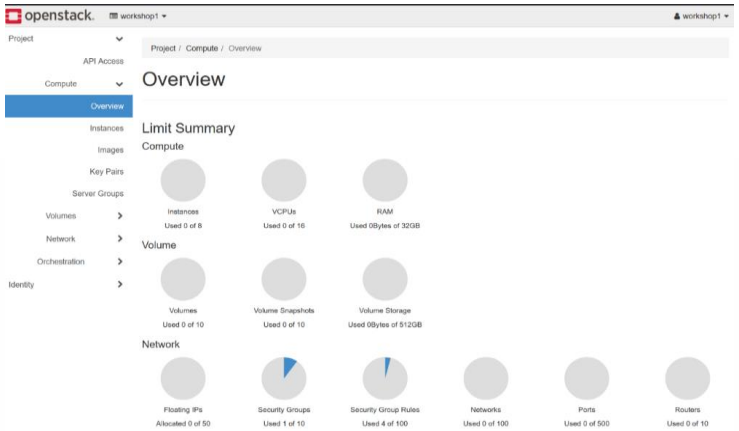
Infrastructure changes & Host OS migrations

OpenStack version upgrades with Kolla-Ansible

Terraform & Consul

End seminar

OPENCLOUDEDGE HANDS-ON WORKSHOPS



Part 1: Set up & **consume** elementary OpenStack project environment

- Tenant network, SSH key, Deploy cloud instances from image
- Volume snapshots of modified cloud instances
- Deploy additional instances from this volume snapshot

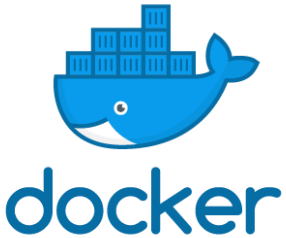
Part 2: Set up and **deploy** multi-node OpenStack infrastructure using **Kolla-Ansible**

- *OpenStack-inside-OpenStack*: Deployed inside compute instances provided by OpenStack
- Set up host machines, Ansible & Kolla-Ansible configuration
- Deploy OpenStack with Kolla containers
- Consume the OpenStack-inside-OpenStack

KOLLA
an OpenStack Community Project

We will schedule additional hands-on workshop events upon request!

OPENCLOUDEDGE HANDS-ON WORKSHOPS

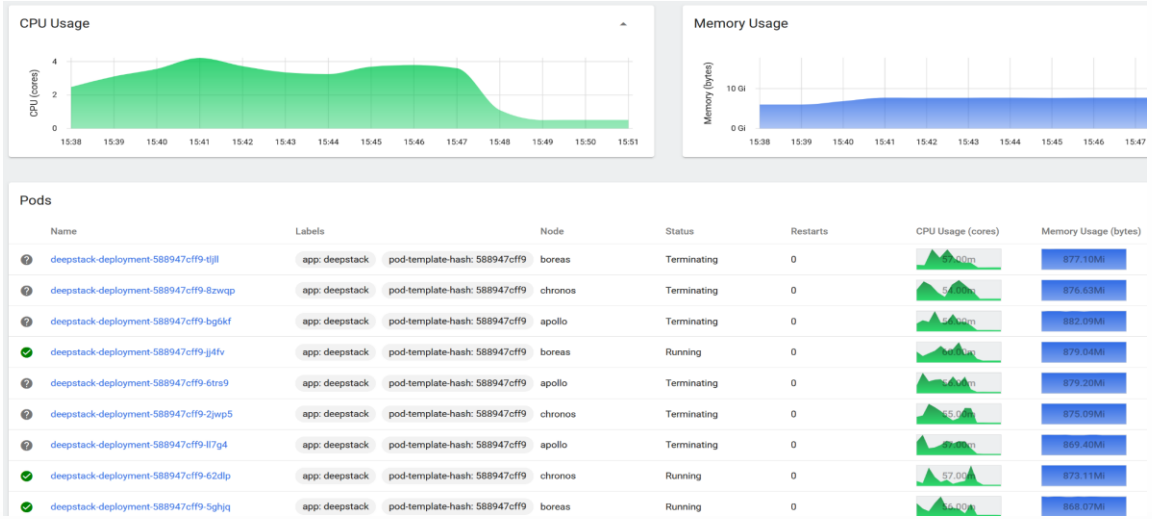


kubernetes

Brief introduction to **Docker** containers

Deployment of multi-node **Kubernetes** environment

- Simplified testing/development with **MicroK8s**



Kubernetes interaction and deployments

- *Kubectl* and Kubernetes Dashboard
- *JupyterHub* from **Helm** package manager
- Porting containerized webapp to autonomous **horizontally scalable** Kubernetes deployment

We will schedule additional hands-on workshop events upon request!

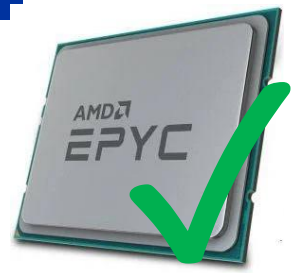
OPENCLOUDEDGE HANDS-ON WORKSHOPS

- **Software installation:** Terraform CLI.
- **Hashicorp Configuration Language:** usage and explanation.
- **Basic commands:** init, plan, apply, destroy.
- **Infrastructure orchestration:** deploy, change, delete.
- **Use of variables and functions:** input, output, count.
- **Multiple providers:** docker, OpenStack, authentications, providers comparison (example with AWS).
- **Examples:** modules, infrastructure import, deployment of a webserver.

We will schedule additional hands-on workshop events upon request!

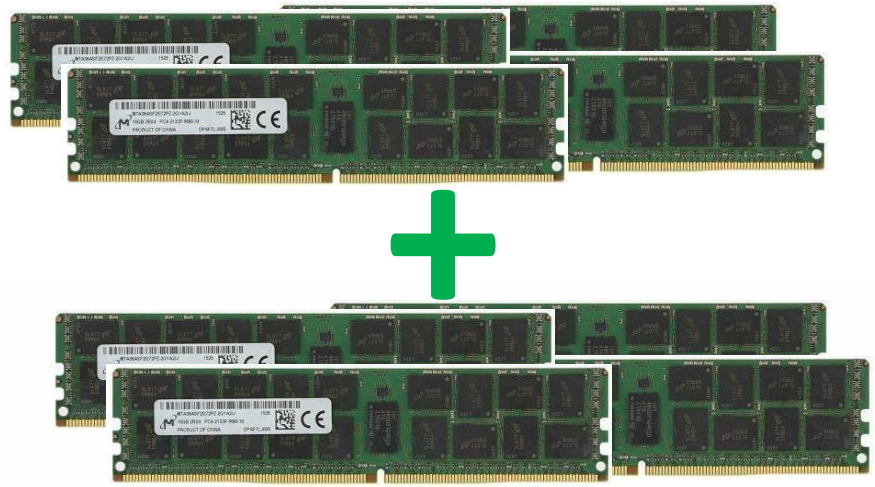
INFRASTRUCTURE CHANGES

UPGRADED SERVER CONFIGURATION

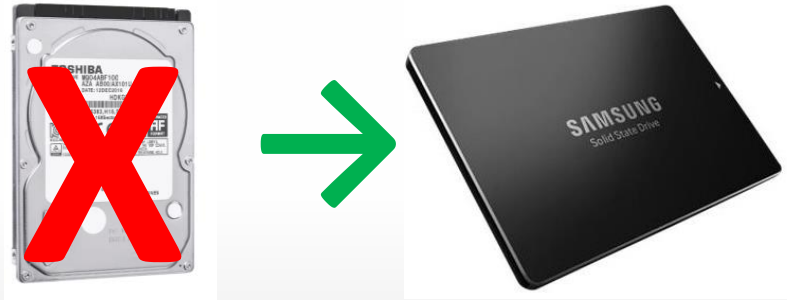


Heterogenous HPE DL325 Gen10 servers

- 16 core AMD EPYC 7302
- 64 GB RAM → 128 GB RAM
  - **Memory constraints**: base load 20 – 30 GB (OpenStack + Ceph)
  - Bare-metal Kubernetes: NO SWAP memory



- ~~12-disk Toshiba 2TB **2,5" HDDs** in Ceph cluster~~
  - **Abysmal performance** under load: latency > 1000 ms
- 8-disk Samsung 2TB PM883 **enterprise SSD** in Ceph cluster
  - Consistent < 10 ms latency, Ceph I/O exceeding 1000 MB/s



- 4x 1 Gbps networking → 2x 10 Gbps networking (SFP+)
  - Ceph distributed storage network & OpenStack management network

HOST OS MIGRATIONS

CENTOS WENT EOL

Initial host OS: **CentOS 8** (Red Hat community release)

- End of Life: 31 December 2021

Suggested upgrade paths:

- **RHEL 8**: commercial release
- ***CentOS 8 Stream***: rolling release "beta" upstream version of RHEL

We want a free stable/LTS host OS with OpenStack support

→ **Ubuntu Focal 20.04 LTS** (EOL April 2025)

HOST OS MIGRATION PROCESS

HIGH AVAILABILITY THROUGHOUT THE MIGRATION PROCESS

ceph ✓

- Migrate one/few servers at a time to ensure redundancy.
- Remove old server(s), migrate to new OS, add new server(s)
  - Existing OSD configuration can be migrated to avoid double rebalancing.

kubernetes ✓ (When following the correct removal procedure)

- Cordon & drain old server(s), remove worker node from Kubernetes cluster
  - If **control-plane node**: follow procedure to remove from *etcd* distributed key-value store (e.g. `kubeadm destroy`)
  - Manual removal inside the *etcd* store IF the prior control-plane server not correctly removed
- Migrate to new OS
- Add server(s) again (e.g. `kubeadm join`)

HOST OS MIGRATION PROCESS

HIGH AVAILABILITY THROUGHOUT THE MIGRATION PROCESS

openstack

with Kolla-Ansible: **X** (Possible but with issues)

KOLLA
an OpenStack Community Project

Preparation via OpenStack CLI:

- Disable OpenStack services (compute, network, storage) on node(s) to be migrated
- Migrate active resources to remaining nodes

Push updated OpenStack configuration with Kolla-Ansible:

- `Kolla-ansible -i [inventory-to-server-to-be-removed] destroy` — *ONLY on the server(s) to be migrated!*
- `Kolla-ansible -i [inventory-to-remaining-servers] deploy` — *Updated configuration for operation without the removed node(s)*

The OpenStack cluster remains with **obsoleted database references** to the removed node(s)

- These references trigger **errors** when adding the migrated node again (using same hostname & IP addr)
- Requires **manual** removal of various entries from the OpenStack underlying SQL databases
  - Could be automated with some research effort

HOST OS MIGRATION PROCESS

HIGH AVAILABILITY THROUGHOUT THE MIGRATION PROCESS

openstack

with Kolla-Ansible: **X** (Possible but with issues)

KOLLA
an OpenStack Community Project

Reinstall server(s) with new OS supported by Kolla-Ansible

- OpenStack Victoria supported OS: CentOS 8, RHEL 8, Debian *Buster* 10 and Ubuntu *Focal* 20.04

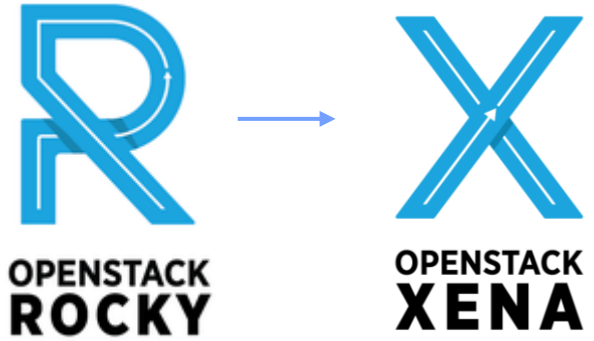
Deploy migrated node(s) with Kolla-Ansible & update cluster configuration

- *Kolla-ansible -i [inventory-to-all-servers] deploy*

Kolla-Ansible can manage OpenStack clusters on **mixed** supported operating systems

- DO NOT specify the ***openstack_release* variable** in *globals.yml*
- Specify ***openstack_release* variable** individually for each node in the Ansible inventory file(s)

OPENSTACK VERSION UPGRADES



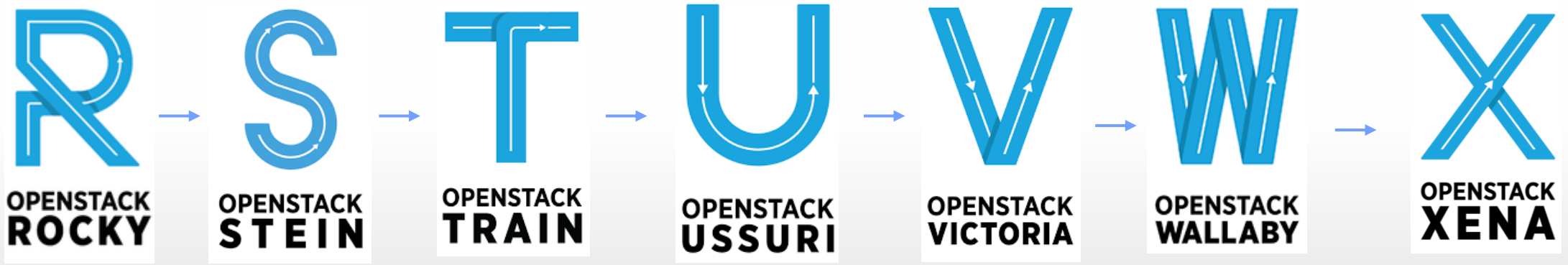
Upgrades: OpenStack ***Rocky*** (Aug 2018) → OpenStack **Xena** (Oct 2021)

Test environment:

- Single-node OpenStack cluster with Ubuntu *bionic* 18.04 LTS
- **Kolla-Ansible** using *all-in-one* Ansible inventory file

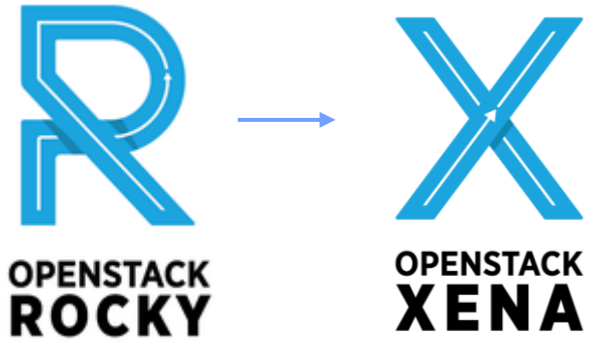
Kolla-Ansible provides a **version-by-version** upgrade path

- Not possible/not recommended to skip versions between upgrades



OPENSTACK VERSION UPGRADES

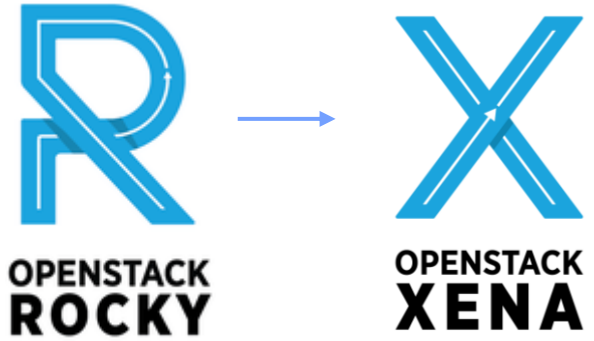
KOLLA-ANSIBLE UPGRADE PROCEDURE (IN GENERAL)



- Install latest available Kolla-Ansible version for corresponding OpenStack release
  - ***Rocky*** = Kolla-Ansible **7**.2.1; ***Stein*** = Kolla-Ansible **8**.3.0; ***Train*** = Kolla-Ansible **9**.3.2; etc.
  - See Kolla-Ansible release notes and/or PyPI repository
- Upgrade & migrate Kolla-Ansible's passwords file
  - Generate new passwords (sometimes required for new/modified features)
  - Merge with the existing passwords
- Upgrade & migrate Ansible inventory file(s) (sometimes required for new/ modified features)
- Upgrade *globals.yml* (primarily *openstack_release* variable)
- Run `kolla-ansible -i all-in-one upgrade`

OPENSTACK VERSION UPGRADES

DOWNTIME DURING UPGRADES (IN GENERAL)



Brief downtime OpenStack components during the service upgrades

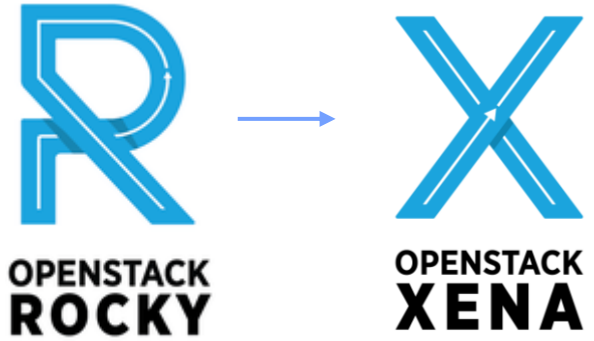
- Downtime in the various OpenStack management services
  - Restarts of services using upgraded Kolla containers
  - Horizon, Keystone, Neutron, etc. momentarily offline
  - Typically < 1 minute downtime per service

Cloud instances (Virtual Machines) **remain operational** during the upgrades ✓

- Nova's *libvirt/KVM/QEMU* back-end not directly affected by the OpenStack upgrades.
- **Network connectivity briefly interrupted**
  - Neutron & OpenVSwitch service containers are upgraded and restarted
  - Restricted to few seconds of downtime

OPENSTACK VERSION UPGRADES

EXPERIENCED ISSUES/EXCEPTIONS



Train → Ussuri

- Kolla-Ansible for Train: **Python 2**, Ussuri: **Python 3**
  - Install Python 3, PIP3, and Python3 versions of Ansible and Kolla-Ansible

Ussuri → Victoria

- Ussuri @ **Ubuntu 18.04**, Victoria @ **Ubuntu 20.04**
  - 1. upgrade to Victoria on 18.04, 2. perform OS upgrade, 3. **Reboot** to new OS
- DOWNTIME (OpenStack Services + Virtual Machines) during reboot procedure

Victoria → Wallaby

- Upgraded minimum **Docker version requirements**
- *Kolla-Ansible bootstrap-servers*
  - Upgrades and restarts Docker → restarts all OpenStack services
- DOWNTIME (OpenStack Services) during Docker restart

TERRAFORM

What is it and why use it?

HashiCorp Terraform

Write, Plan, and Create Infrastructure as Code

- Opensource Infrastructure-as-Code software tool
- Efficient deployment, orchestration and automation
- One tool to manage any resource, regardless of where, with more than 500 providers (public/private clouds, network appliances, PaaS, SaaS)
- Excellent to handle multi-cloud / hybrid cloud scenarios

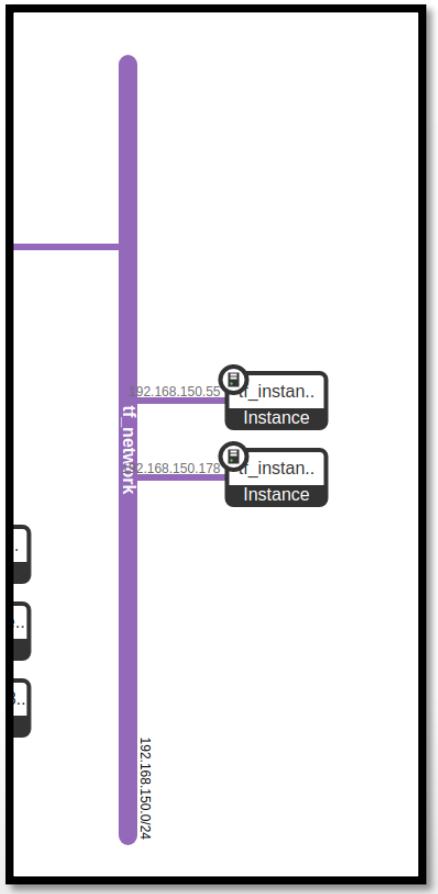
Infrastructure-as-Code (IaC)

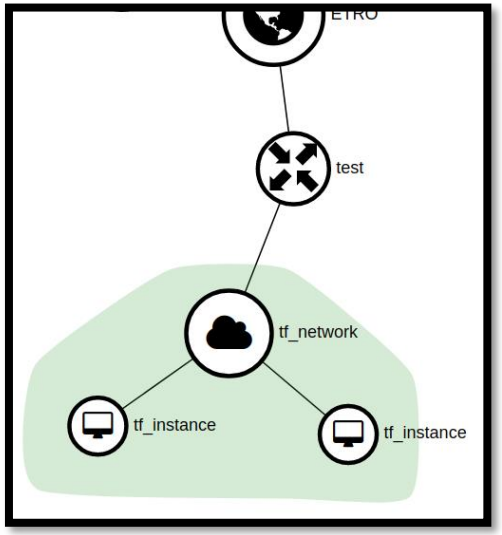
```
provider "openstack" {}

resource "openstack_compute_instance_v2" "tf_instance" {
  count = var.instances_per_subnet
  name = "tf_instance"
  image_name = "example"
  flavor_name = "m1.tiny"
  network {
    name = openstack_networking_network_v2.tf_network.name
  }
}

resource "openstack_networking_network_v2" "terraform_network" {
  name = "tf_network"
}

resource "openstack_networking_subnet_v2" "terraform_subnet" {
  name = "tf_subnet"
  network_id = openstack_networking_network_v2.tf_network.id
  cidr = "192.168.1.0/24"
}

resource "openstack_blockstorage_volume_v2" "tf_volume" {
  count = var.instances_per_subnet
  name = "tf_volume"
  size = 2
}

resource "openstack_compute_volume_attach_v2" "tf_attach" {
  count = var.instances_per_subnet
  instance_id = openstack_compute_instance_v2.tf_instance[count.index].id
  volume_id = openstack_blockstorage_volume_v2.tf_volume[count.index].id
}

resource "openstack_networking_floatingip_v2" "public_network" {
  pool = "ETRO"
}
```

Configuration file example

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status
tf_instance	cirros	192.168.150.178	m1.tiny	-	Active
tf_instance	cirros	192.168.150.55	m1.tiny	-	Active

Name	Description	Size	Status	Group	Type	Attached To
tf_volume	-	2GiB	In-use	-	__DEFAULT__	/dev/vdb on tf_instance
tf_volume	-	2GiB	In-use	-	__DEFAULT__	/dev/vdb on tf_instance

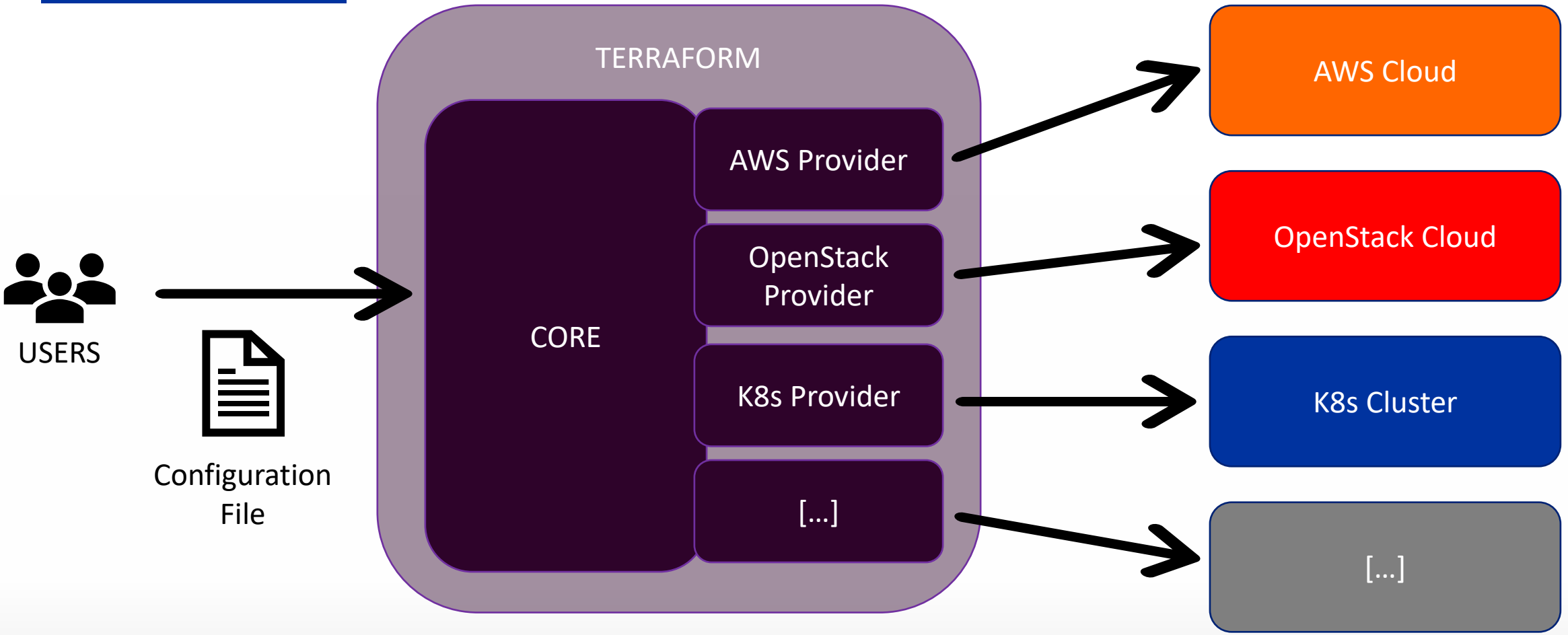




Infrastructure example

TERRAFORM ARCHITECTURE

Core and providers



TERRAFORM

Use cases examples

- Cloud orchestration
- Automation and replication
- Multi-cloud deployment
- Cloud migration

```
terraform {
required_version = ">= 0.14.0"
  required_providers {
    openstack = {
      source = "terraform-provider-openstack/openstack"
      version = "~> 1.35.0"
    }
  }
}

provider "openstack" {
  user_name = var.user_name
  tenant_name = var.tenant_name
  password = var.password
  auth_url = var.auth_url
}

resource "openstack_networking_network_v2" "net" {
  name = var.network_name
}

resource "openstack_networking_subnet_v2" "subnet" {
  name = var.subnet_name
  network_id = openstack_networking_network_v2.net.id
  cidr = var.cidr_value
}

resource "openstack_compute_instance_v2" "instance" {
  name = var.instance_name
  image_name = var.image_name
  flavor_name = var.flavor_name
  network {
    name = openstack_networking_network_v2.net.name
  }
}

resource "openstack_blockstorage_volume_v2" "vol" {
  name = var.volume_name
  size = var.volume_size
}

resource "openstack_compute_volume_attach_v2" "vol_attach" {
  instance_id = openstack_compute_instance_v2.instance.id
  volume_id = openstack_blockstorage_volume_v2.vol.id
}
```

OpenStack

```
terraform {
required_version = ">= 0.14.0"
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.0"
    }
  }
}

provider "aws" {
  region = var.region
  access_key = var.access_key
  secret_key = var.secret_key
}

resource "aws_vpc" "net" {
  tags = {Name = var.network_name}
  cidr_block = var.cidr_net_block
}

resource "aws_subnet" "subnet" {
  tags = {Name = var.subnet_name}
  vpc_id     = aws_vpc.net.id
  cidr_block = var.cidr_value
}

resource "aws_instance" "instance" {
  tags = {Name = var.instance_name}
  ami = var.image_name
  instance_type = var.flavor.name
  subnet_id = aws_subnet.subnet.id
}

resource "aws_ebs_volume" "vol" {
  tags = {Name = var.volume_name}
  size = var.volume_size
}

resource "aws_volume_attachment" "vol_attach" {
  instance_id = aws_instance.instance.id
  volume_id = aws_ebs_volume.vol.id
}
```

AWS

TERRAFORM WORKSHOP

Basics and advanced topics

- **INIT**

Initialize Terraform and look for providers

- **PLAN**

Overview of what to execute to realize what is described in the configuration files

- **APPLY**

Perform the operations as planned

- **DESTROY**

Deallocate and destroy all the resources

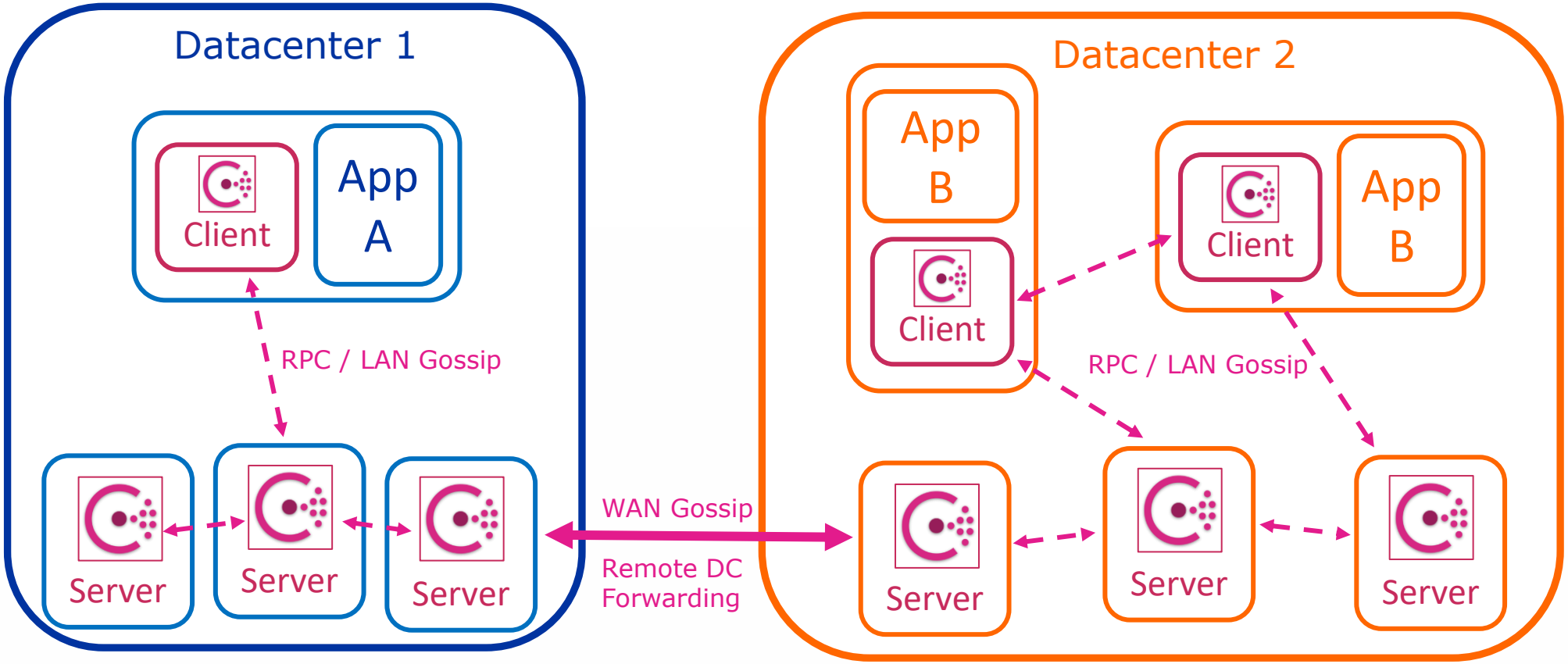
```
  Enter a value: yes

openstack_networking_secgroup_v2.secgroup: Creating...
openstack_networking_floatingip_v2.ws_floating_ip: Creating...
openstack_networking_network_v2.workshop_network: Creating...
openstack_networking_router_v2.ws_router: Creating...
openstack_networking_secgroup_v2.secgroup: Creation complete after 0s [id=4c0d3741-1e58-4a4a-9e92-d2450ec2d1f9]
openstack_networking_secgroup_rule_v2.secgroup_rule: Creating...
openstack_networking_secgroup_rule_v2.secgroup_rule: Creation complete after 0s [id=401cec09-360d-477a-aac1-b948b4d910d3]
openstack_networking_network_v2.workshop_network: Creation complete after 5s [id=c1fb2002-a930-48b3-9476-08ac18708f06]
openstack_networking_subnet_v2.workshop_subnet: Creating...
openstack_compute_instance_v2.workshop_instance: Creating...
openstack_networking_floatingip_v2.ws_floating_ip: Creation complete after 6s [id=47ae9b31-423c-49ca-bc7d-b15f94e41e16]
openstack_networking_router_v2.ws_router: Creation complete after 6s [id=ed6c0809-6ac5-48a4-b07e-b1457811a628]
openstack_networking_subnet_v2.workshop_subnet: Creation complete after 5s [id=49522f7d-0815-43de-b219-74b3b0d6ab2a]
openstack_networking_router_interface_v2.router_interface: Creating...
openstack_compute_instance_v2.workshop_instance: Still creating... [10s elapsed]
openstack_networking_router_interface_v2.router_interface: Creation complete after 7s [id=a9044234-ac3a-4f49-a03c-bc5af5f1d840]
openstack_compute_instance_v2.workshop_instance: Creation complete after 12s [id=ff18f067-4ad9-4383-97e1-67a1a88e750e]
openstack_compute_floatingip_associate_v2.fip_associate: Creating...
openstack_compute_floatingip_associate_v2.fip_associate: Creation complete after 2s [id=10.20.29.149/ff18f067-4ad9-4383-97e1-67a1a88e750e/]

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.

Outputs:

floating_ip = "10.20.29.149"
```

CONSUL

What is it and why use it?

- Open-source tool for deploying a service mesh (HCL language, Terraform integration)
- Multiple features: Centralized registry, service discovery, health checks, zero trust network, load-balancer, Key-Value store...
- Can be easily deployed in both VMs and containers
- Perfect for multi-cloud environment

Consul architecture



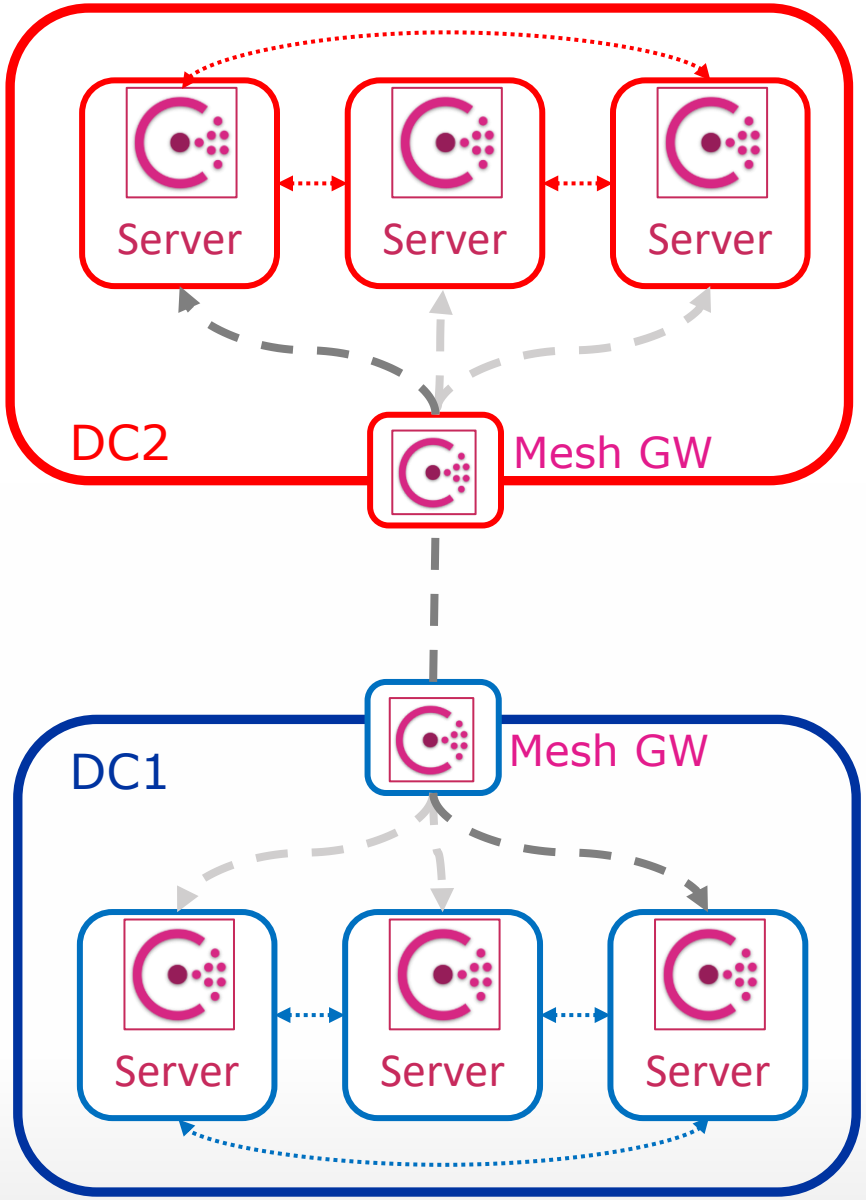
Name	Datacenter	IP	Status
App A	1	192.168.1.45	Alive
App B	2	10.0.0.54, 10.0.0.98	Alive, Alive

Name	Datacenter	IP	Status
App A	1	192.168.1.45	Alive
App B	2	10.0.0.54, 10.0.0.98	Alive, Alive

Consul features

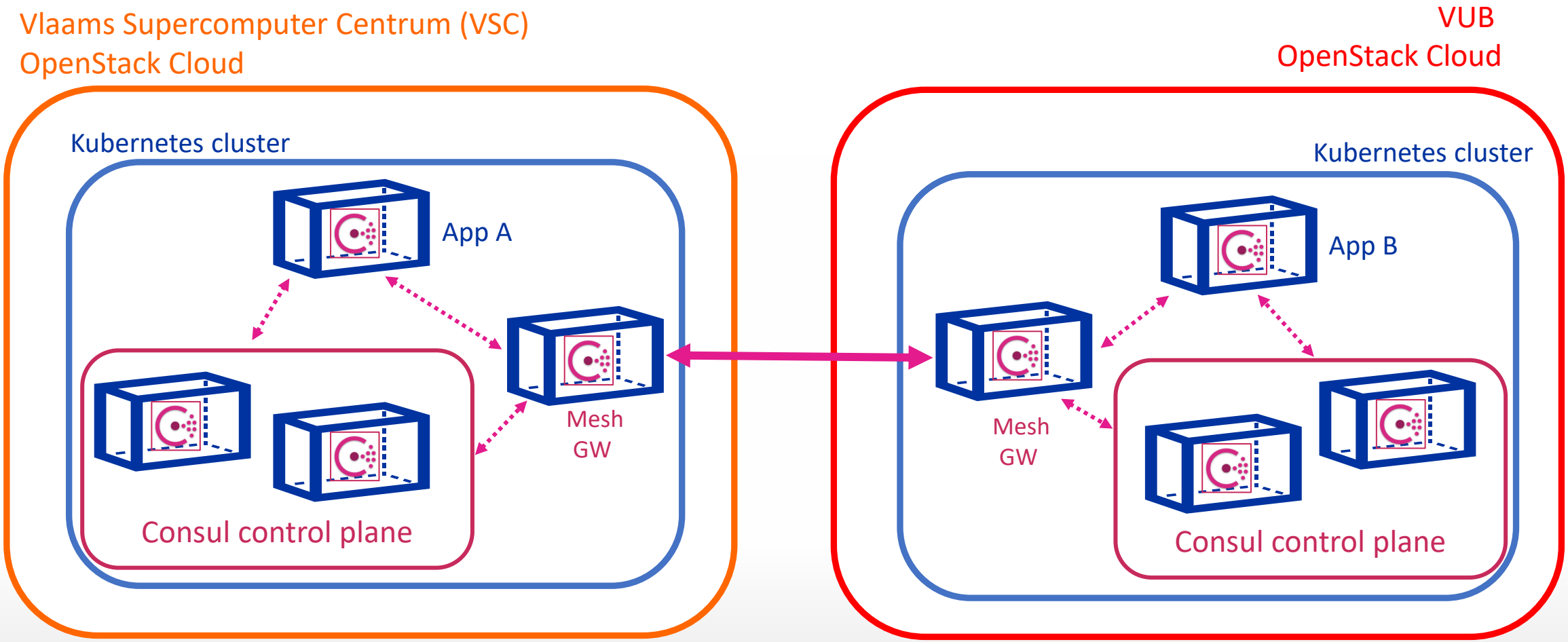
Multi-cloud Service Mesh

- Multi-cloud service mesh
- Mesh gateways
- Datacenter federation
- mTLS (mutual Transport Layer Security)

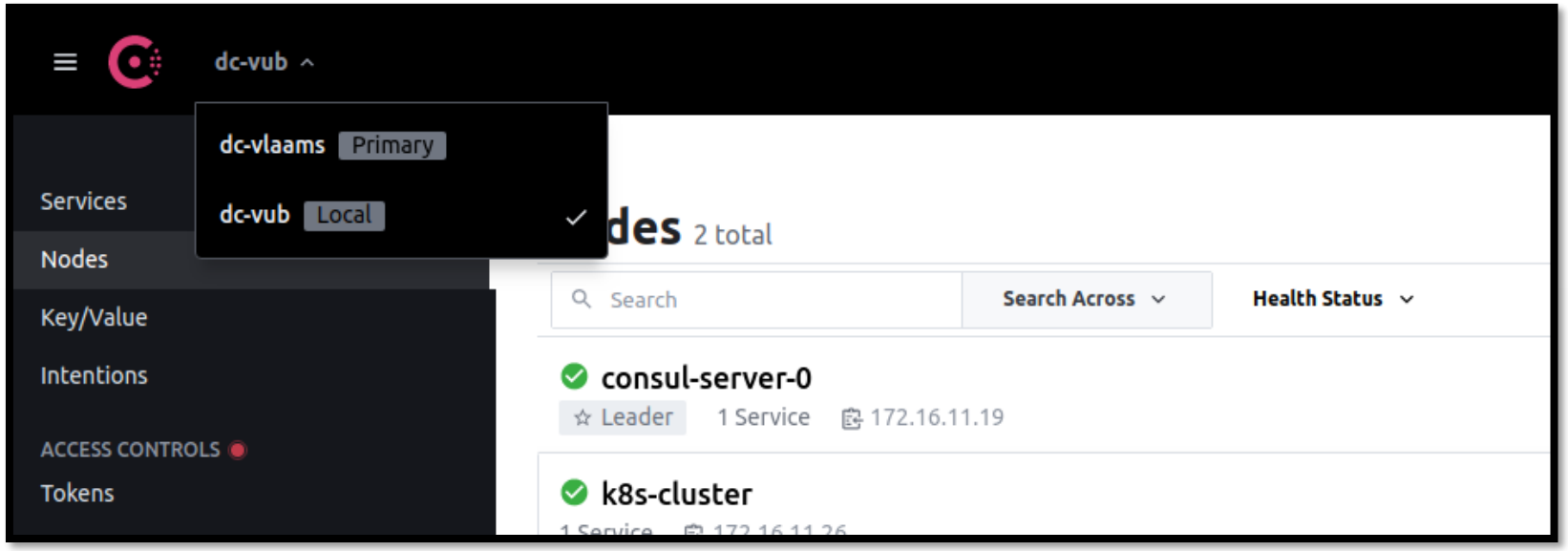


MULTI-CLOUD CONNECTION

K8s cluster federation



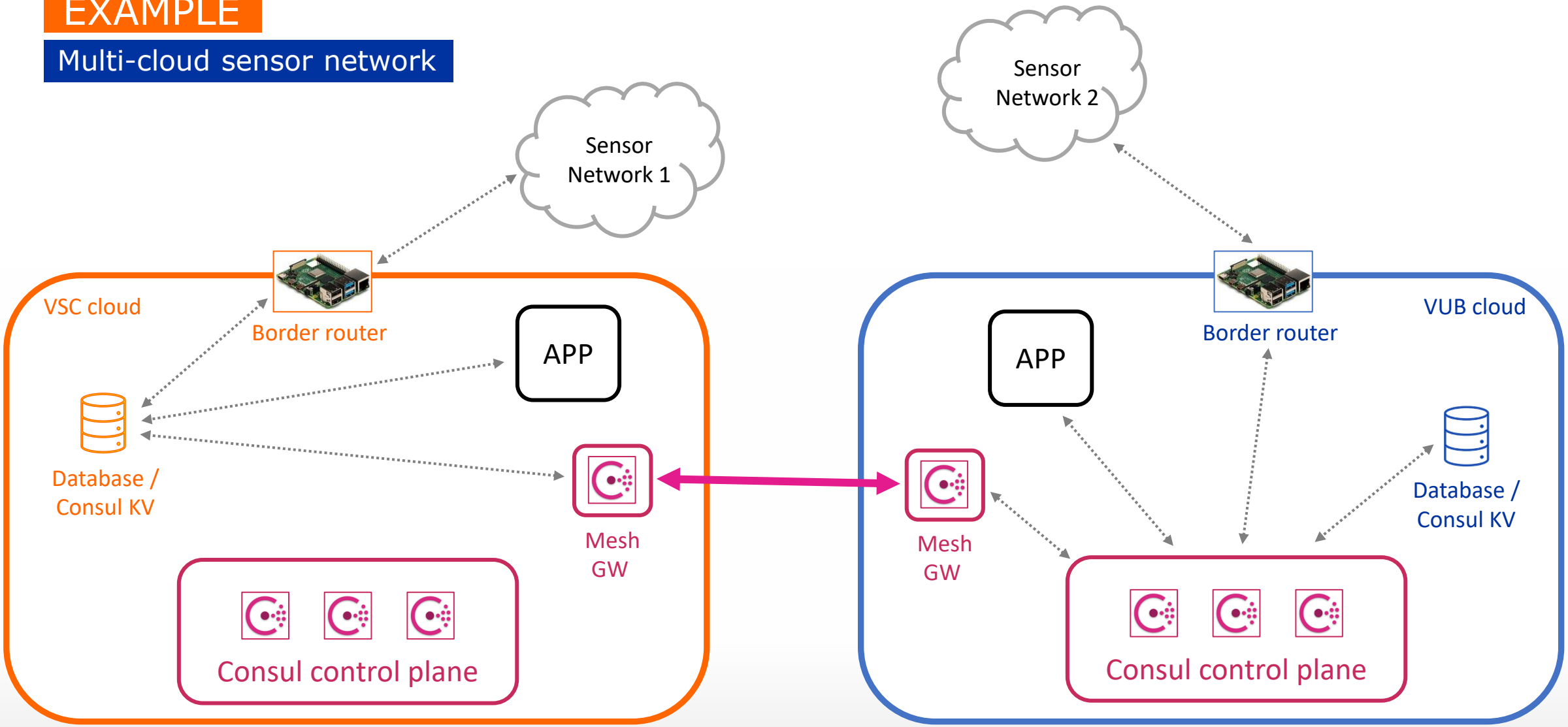
Multi-cloud connection




```
ubuntu@k8s-cluster:~$ kubectl exec statefulset/consul-server -- consul members -wan
Node                       Address              Status  Type    Build   Protocol  DC         Partition  Segment
consul-server-0.dc-vlaams  192.168.223.97:8302  alive   server  1.10.2  2         dc-vlaams  default    <all>
consul-server-0.dc-vub     172.16.11.19:8302    alive   server  1.11.1  2         dc-vub     default    <all>
```

EXAMPLE

Multi-cloud sensor network



dsp valley

LSEC
LEADERS IN SECURITY

VUB VRIJE UNIVERSITEIT BRUSSEL

PRELIMINARY AGENDA

13h30 - 14h00: Welcome with coffee

14h00 - 14h40: Approach and experiences TETRA OpenCloudEdge project (Steffen Thielemans & Luca Gattobigio & Priscilla Benedetti, VUB)

14h40 - 15h00: Vlaams Supercomputer Centrum Tier-1 Cloud - Typical use cases (Ewald Pauwels, VSC-UGent)

15h00 - 15h15: Insights on OpenStack and Kubernetes from AXS Guard (Alex Ongena, AXS Guard)

15h15 - 15h30: Nomad as alternative for Kubernetes (Kris Buytaert, Inuits)

15h30 - 16h00: Coffee break

16h00 - 16h15: Best practices and challenges related to containers and microservices (Cloud Security Alliance)

16h15 - 16h30: OpenShift expertise in the Flemish context (Piros)

16h30 - 16h45: Enterprise cloud native computing with Red Hat OpenShift (Westpole)

16h45 - 18h30: Networking drink

Looking for ways on how to scale your current and future IT needs? Thoughts about **Cloud** migration and how to best balance pros and cons? Eager to learn how **open-source** technology fits in?

You are cordially invited to the TETRA OpenCloudEdge seminar, which will take place on Thursday afternoon, **24 February**, in a **hybrid** physical & online format.

Industry and academic speakers will present experience with **OpenStack**, **Kubernetes** and related open-source cloud technologies. They will discuss advantages and pitfalls as well as presenting opportunities.

The physical venue location is **DSP Valley, Esperantolaan 4, 3001 Leuven**.
A link for online participation will be sent together with the final programme.

We are looking forward to your attendance.
DSP Valley, LSEC and TETRA OpenCloudEdge team