ITALY
OpenInfra Days

OPENINFRA DAYS
ITALY

Organized by

IRIDEOS

Binario Etico

Under the patronage of

AGID | Agenzia per l'Italia Digitale

Sponsored by

Mellanox TECHNOLOGIES

MESOSPHERE

gci SERVICE FACTORY

Pierre Riteau

Rome, October 3, 2019

Under new management: migrating a running OpenStack to containerisation with Kolla

StackHPC

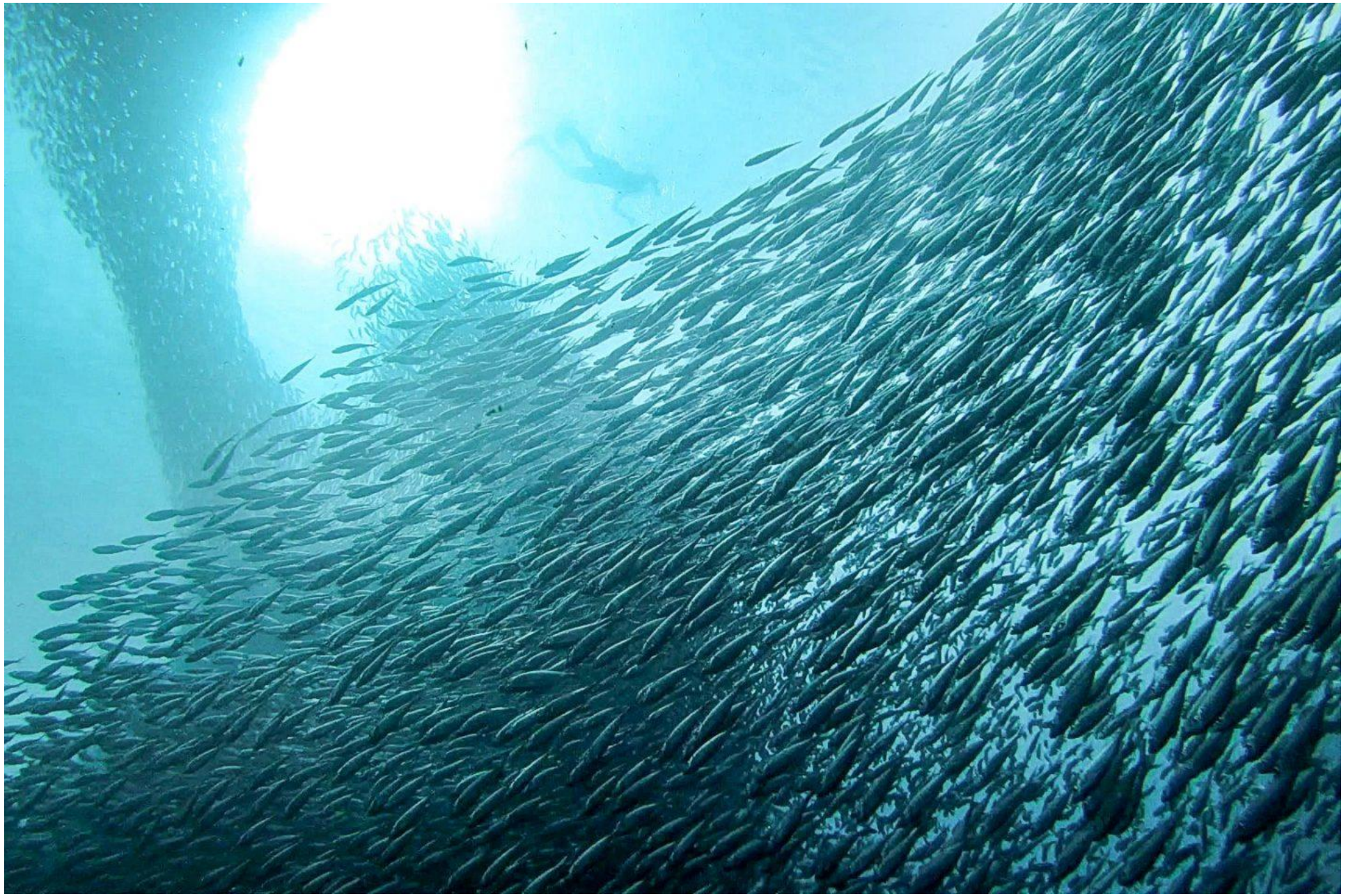
Who am I?

- Senior Technical Lead @ **StackHPC**
- **StackHPC**: Cloud and HPC consultancy
  - Based in Bristol, UK
  - **Deployment** and **support** of OpenStack clouds
  - OpenStack **development** for **research computing**
  - **Core reviewers** in several major OpenStack projects
- **Project Team Lead** of **Blazar** (Reservation as a Service)

BLAZAR
an OpenStack Community Project

Outline

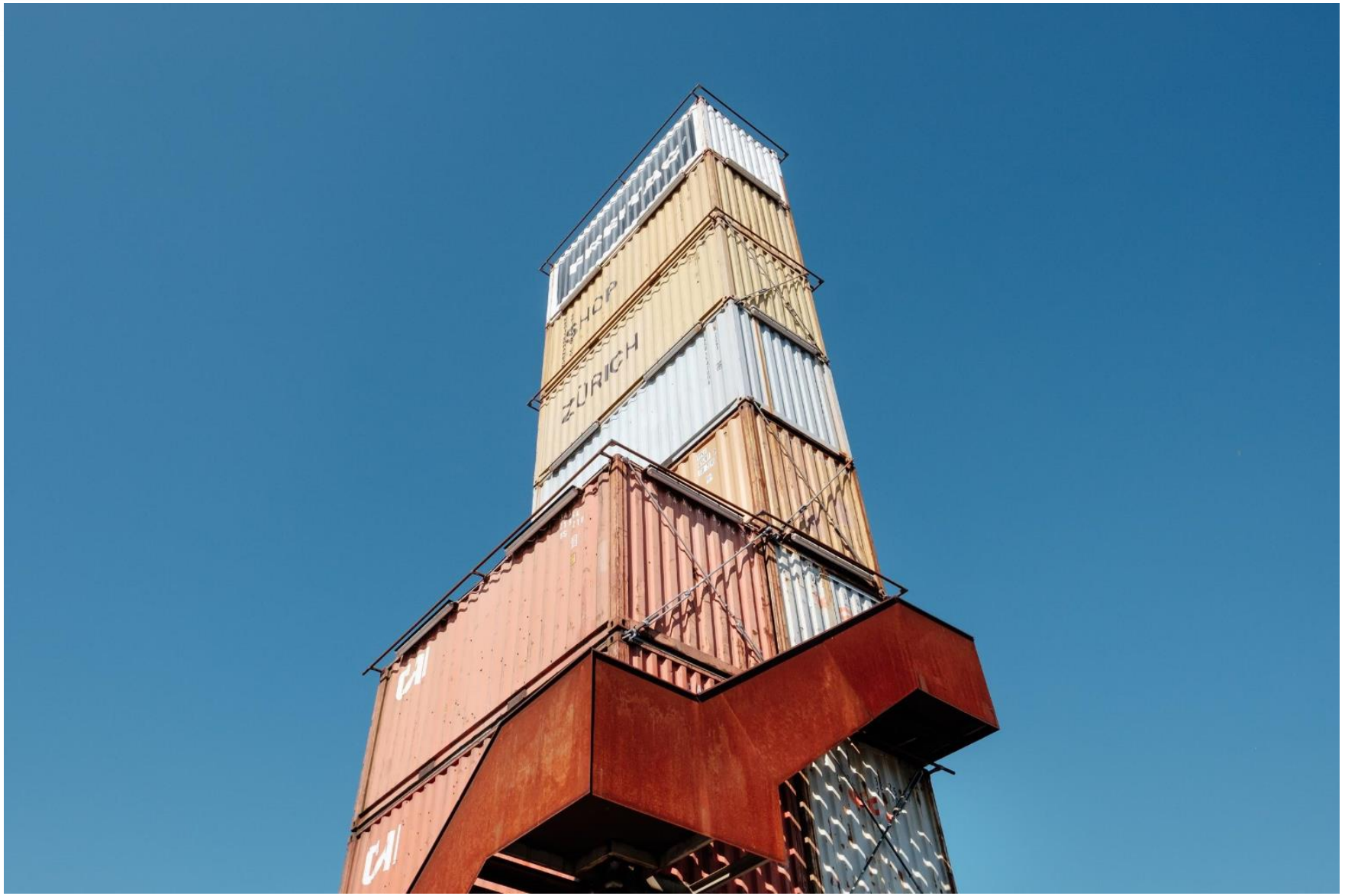
1. OpenStack migration
2. Containerised OpenStack: Kolla and Kayobe
3. Migration strategy
4. Tips & tricks
5. Conclusion



OpenStack migration

Existing OpenStack deployment

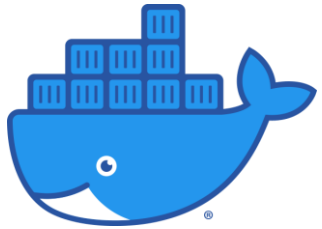
- OpenStack **Queens release**
- Deployed with **CentOS RPMs (RDO)**
- 16 nodes **control plane**
- **HA configuration** with 2 or 3 hosts per service
- **Ceph** storage cluster
  - Used by OpenStack services
  - Including for instance disks
- Around **40 hypervisors**
- Multiple **generations** of hardware
- Gone through several OpenStack upgrades



Containerised OpenStack: Kolla & Kayobe

Why containerise OpenStack?

- OpenStack has lots of components and dependencies
- Many benefits in deploying OpenStack with containers
  - Isolation of dependencies
  - Repeatability of deployment:
    - **Validate** container images in **staging**
    - Deploy **exact same** images in **production**
  - Avoids **divergence** of installed packages over time



OpenStack Kolla (and Kolla Ansible)

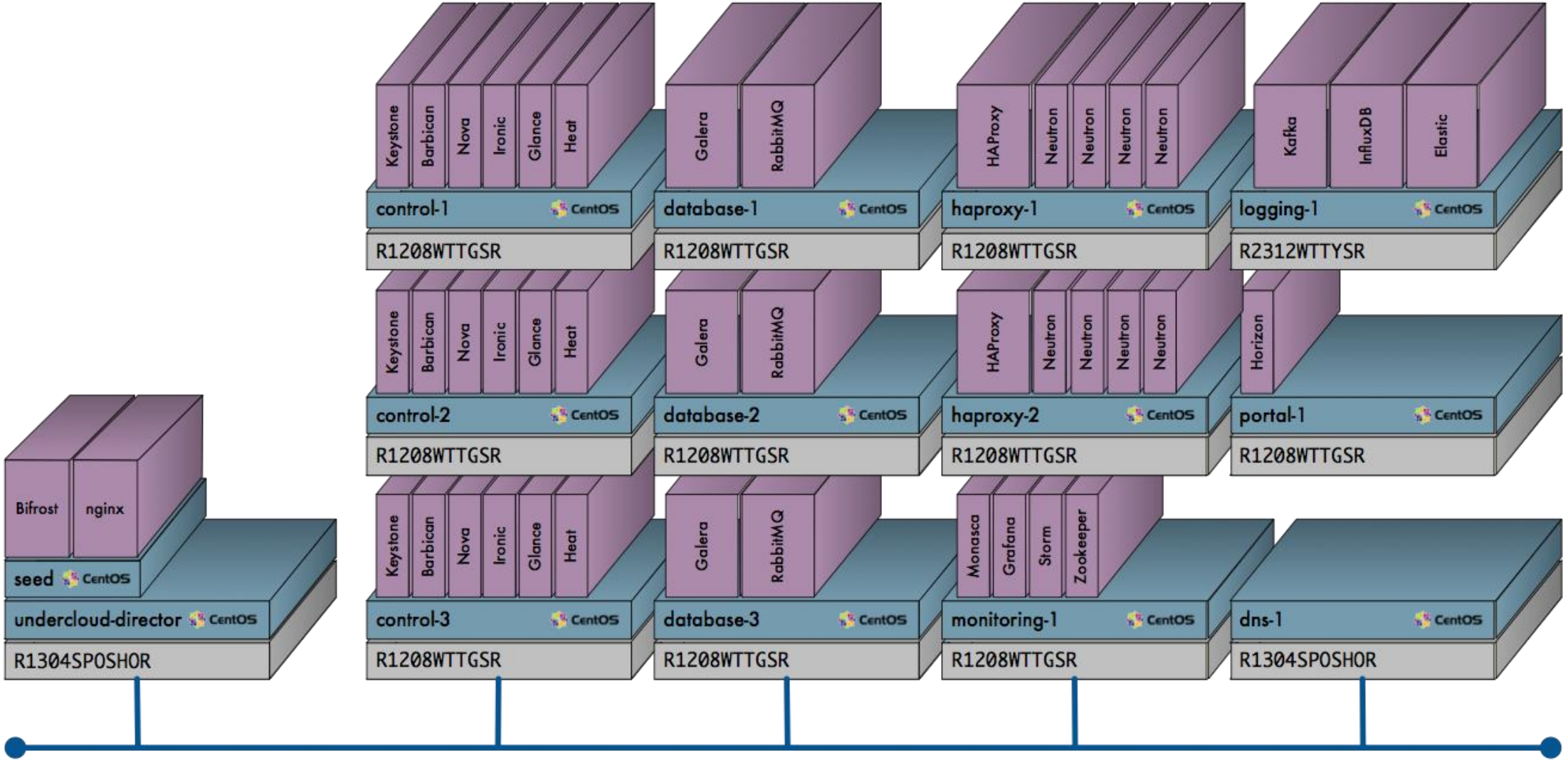
- Kolla: tooling & definitions for building container images
- Kolla Ansible: deployment of Kolla images
- Sane configuration defaults with complete customization
  - Override configuration with INI syntax
- Three principal operations
  - Deploy, reconfigure, upgrade
- Target specific services
  - `kolla-ansible deploy --tags <tags>`
- Target specific hosts
  - `kolla-ansible upgrade --limit <hosts | groups>`
- Learn more at https://docs.openstack.org/kolla/latest/

KOLLA

an OpenStack Community Project

OpenStack Kayobe

- Adds some missing pieces to Kolla Ansible
  - Bare metal deployment of control plane using bifrost
  - Kolla Ansible inventory generation & service placement
  - Configuration of control plane host OS
  - Management of physical network devices
  - Bare metal compute node management
  - A friendly `openstack`-like CLI
- Now **officially** part of the OpenStack Kolla project
- Version 6.0.0 released with **Stein** support this summer
- Learn more at https://docs.openstack.org/kayobe/latest/


Keystone
Barbican
Nova
Ironic
Glance
Heat
Galera
RabbitMQ
HAProxy
Neutron
Neutron
Neutron
Neutron
Kafka
InfluxDB
Elastic
control-1
CentOS
database-1
CentOS
haproxy-1
CentOS
logging-1
CentOS
R1208WTTGSR
R1208WTTGSR
R1208WTTGSR
R2312WTTYSR
Keystone
Barbican
Nova
Ironic
Glance
Heat
Galera
RabbitMQ
HAProxy
Neutron
Neutron
Neutron
Neutron
Horizon
control-2
CentOS
database-2
CentOS
haproxy-2
CentOS
portal-1
CentOS
R1208WTTGSR
R1208WTTGSR
R1208WTTGSR
R1208WTTGSR
Bifrost
nginx
seed
CentOS
undercloud-director
CentOS
R1304SP0SH0R
Keystone
Barbican
Nova
Ironic
Glance
Heat
Galera
RabbitMQ
Monasca
Grafana
Storm
Zookeeper
control-3
CentOS
database-3
CentOS
monitoring-1
CentOS
dns-1
CentOS
R1208WTTGSR
R1208WTTGSR
R1208WTTGSR
R1304SP0SH0R

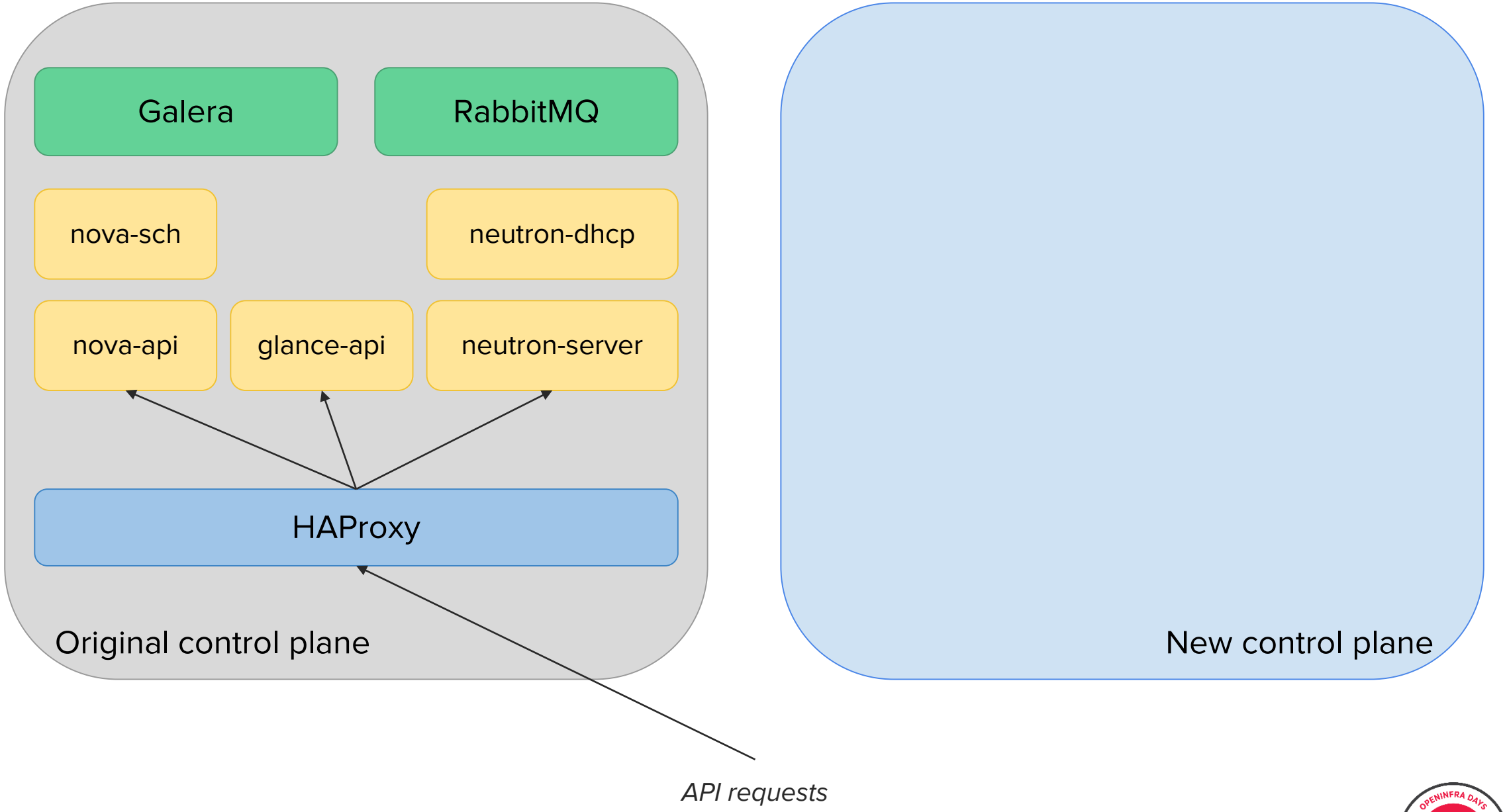
Migration strategy

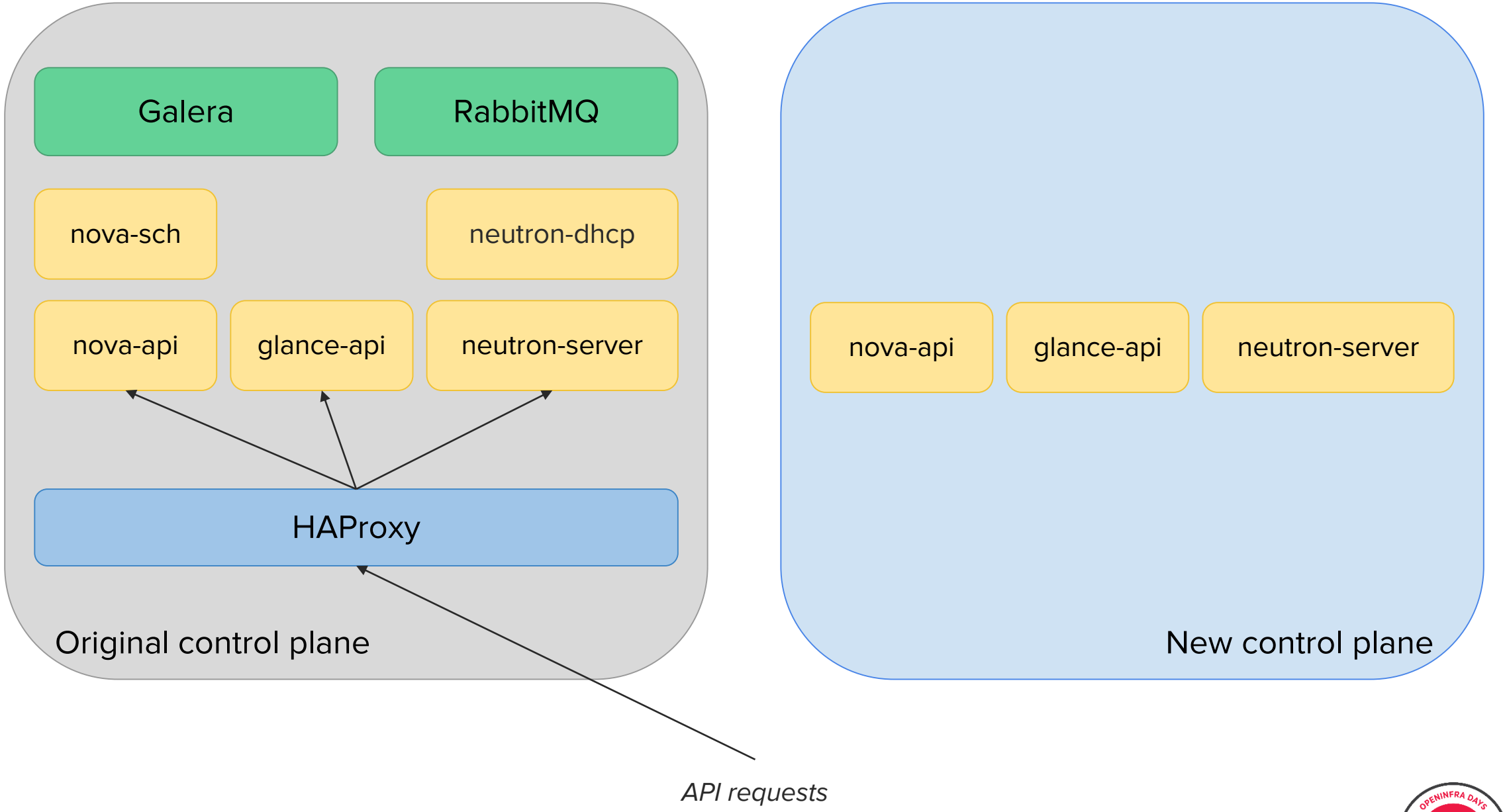
Migration goals

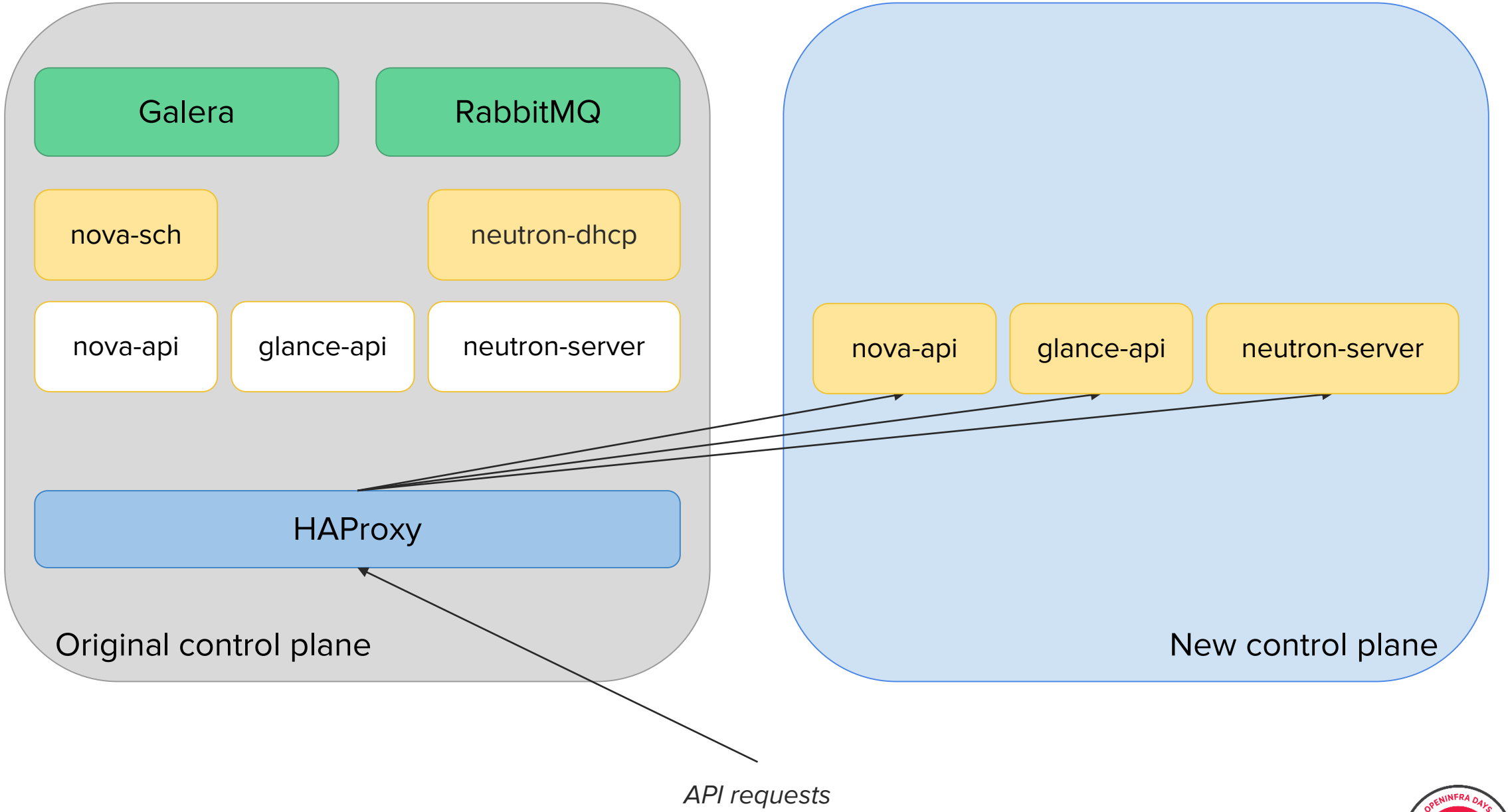
- Transfer control plane services to Kolla on new hardware
- New control plane specifications:
  - 3 **controller** nodes (API services, SQL, RabbitMQ)
  - 2 **network** nodes (Neutron agents, HAProxy)
  - 3 **monitoring** nodes
- Reprovision hypervisors and deploy with Kayobe & Kolla
- **Minimize impact** on end users
  - Replicate existing configuration
  - Keep API downtime short
  - Keep VM network unreachability even shorter

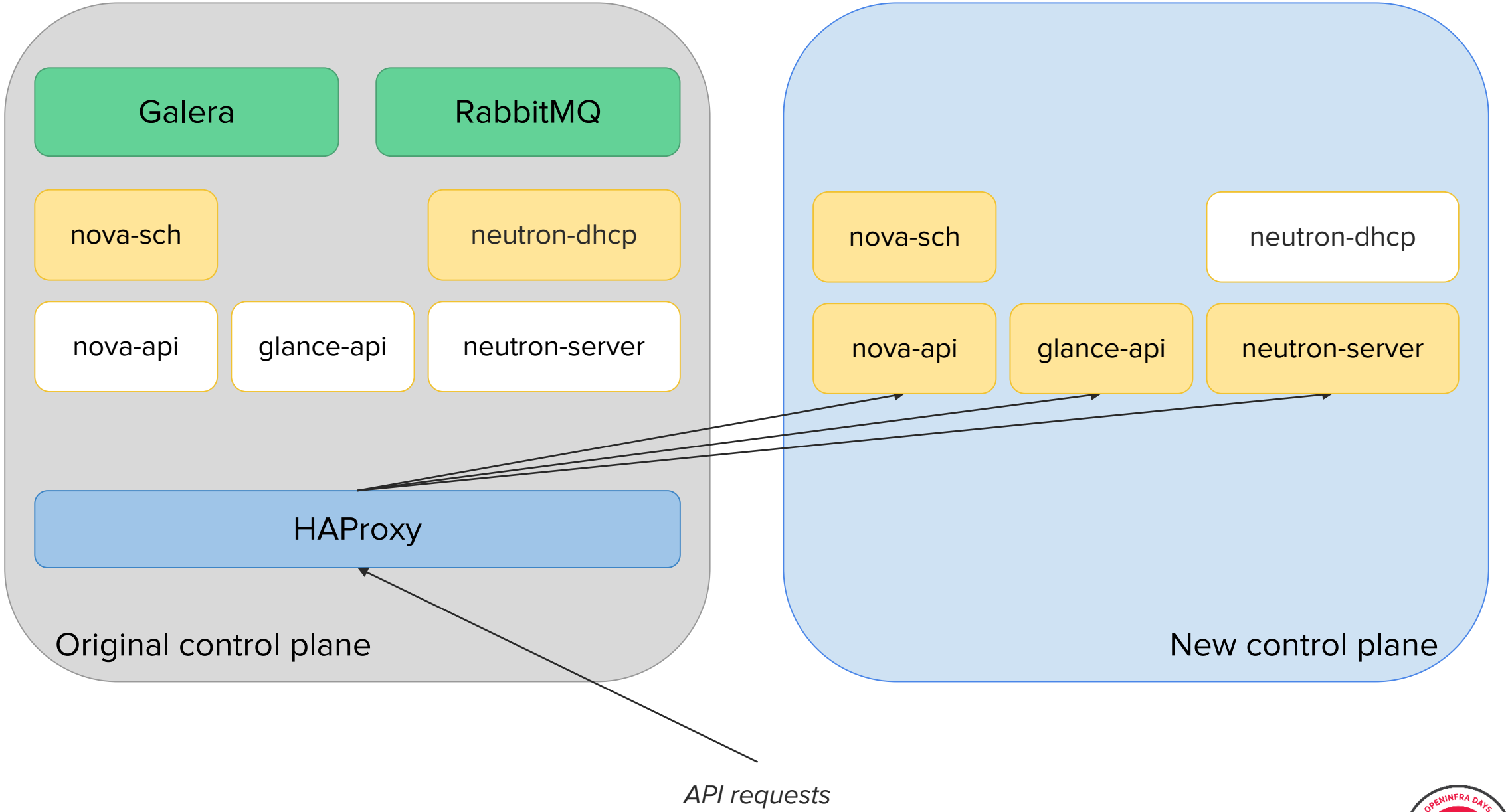
Control plane migration

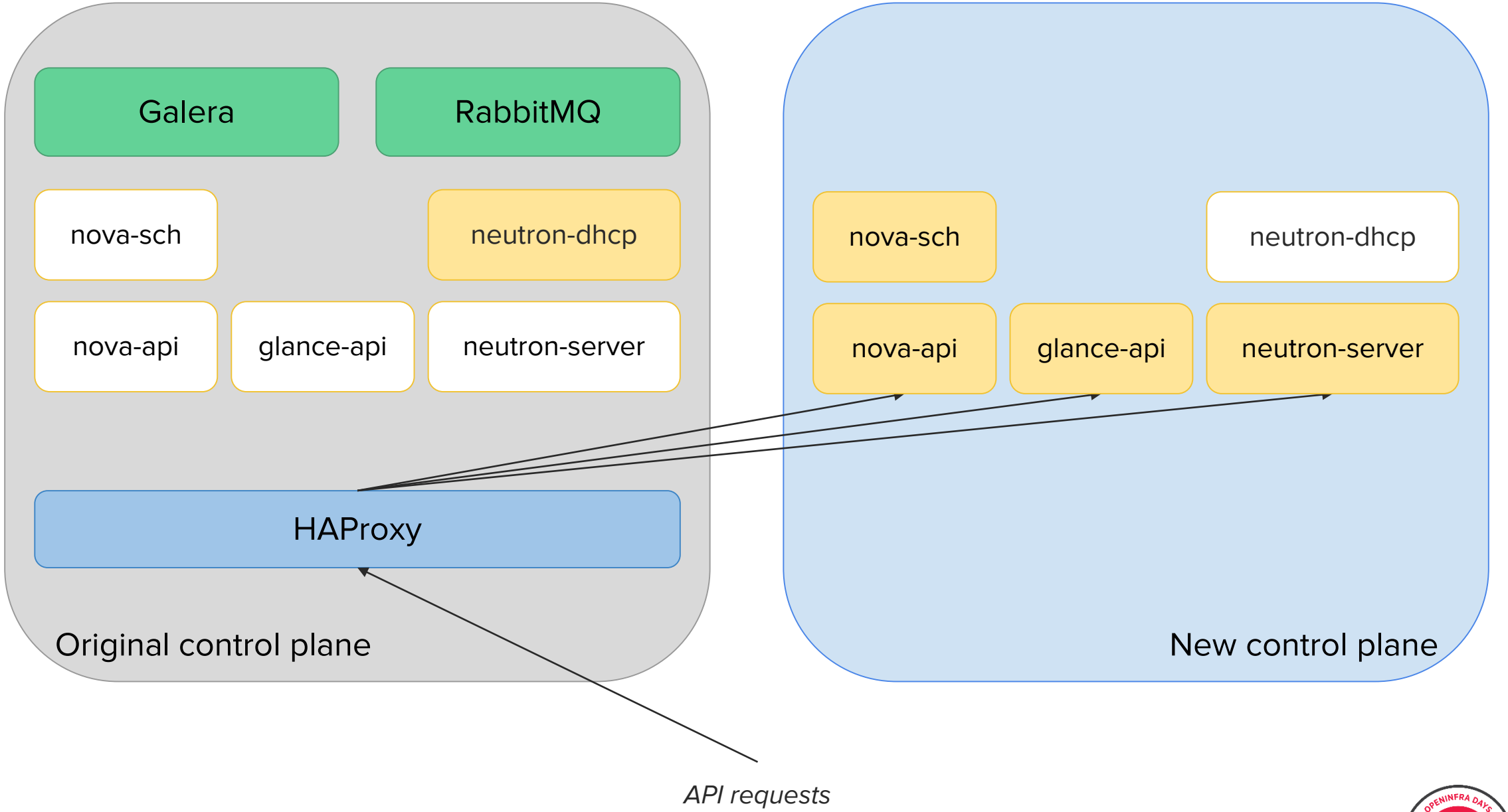
- Configure new hosts with Kayobe
- Deploy OpenStack services with Kolla
- Activate them **progressively** while performing **validation**
- Retire OpenStack services on original control plane
- Short downtime to transfer SQL database

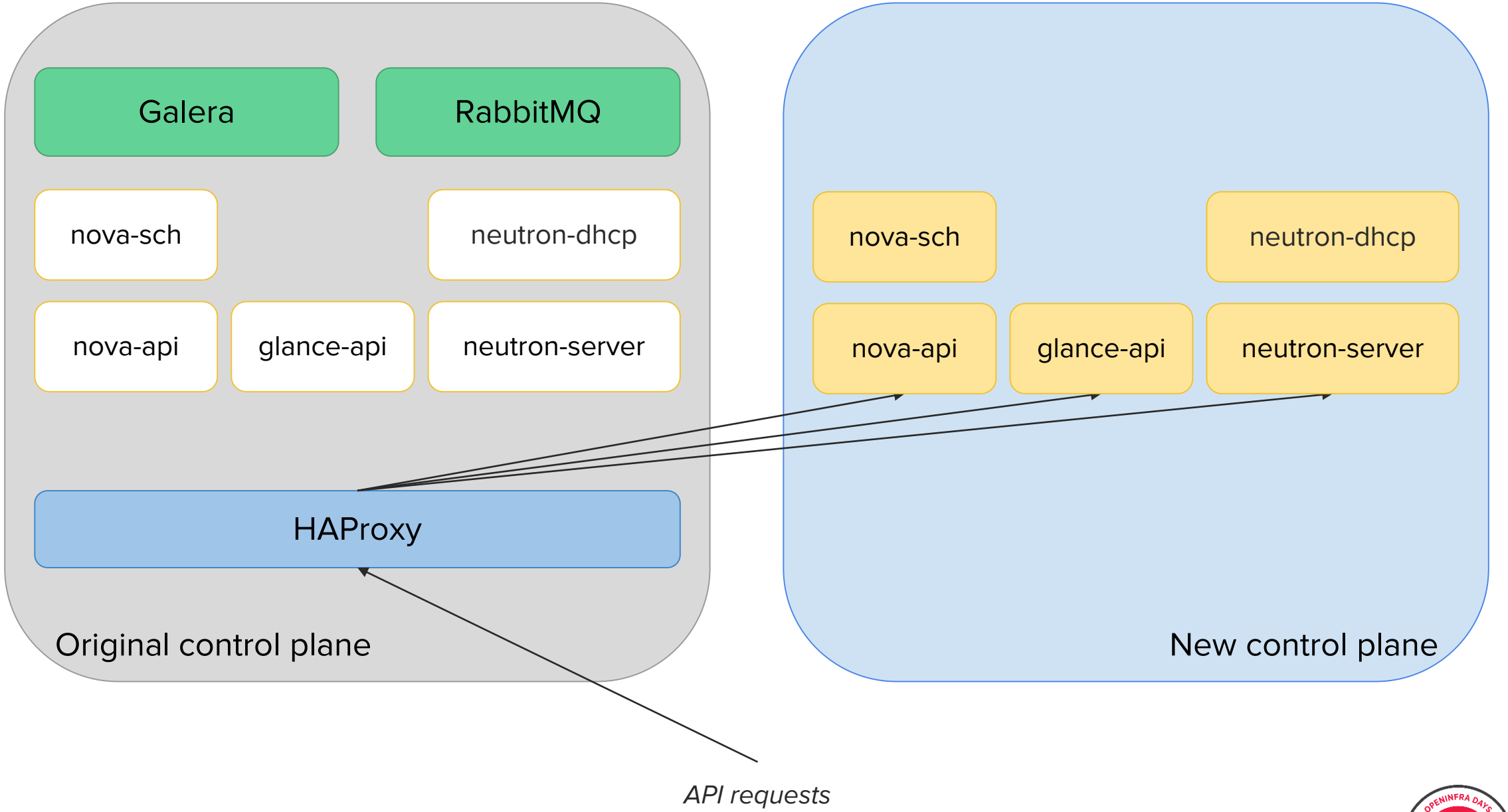

Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
New control plane
API requests

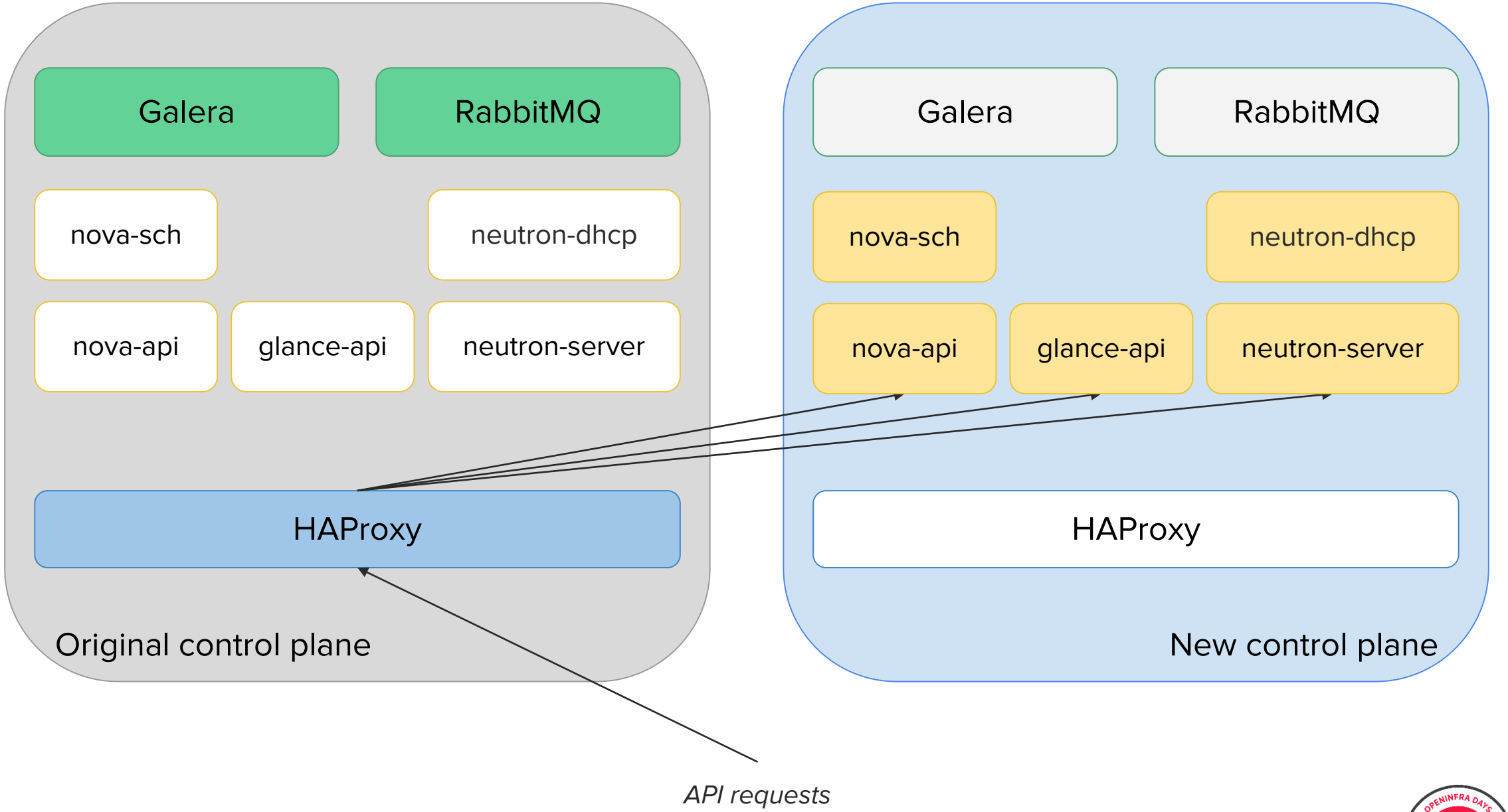

Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
nova-api
glance-api
neutron-server
New control plane
API requests

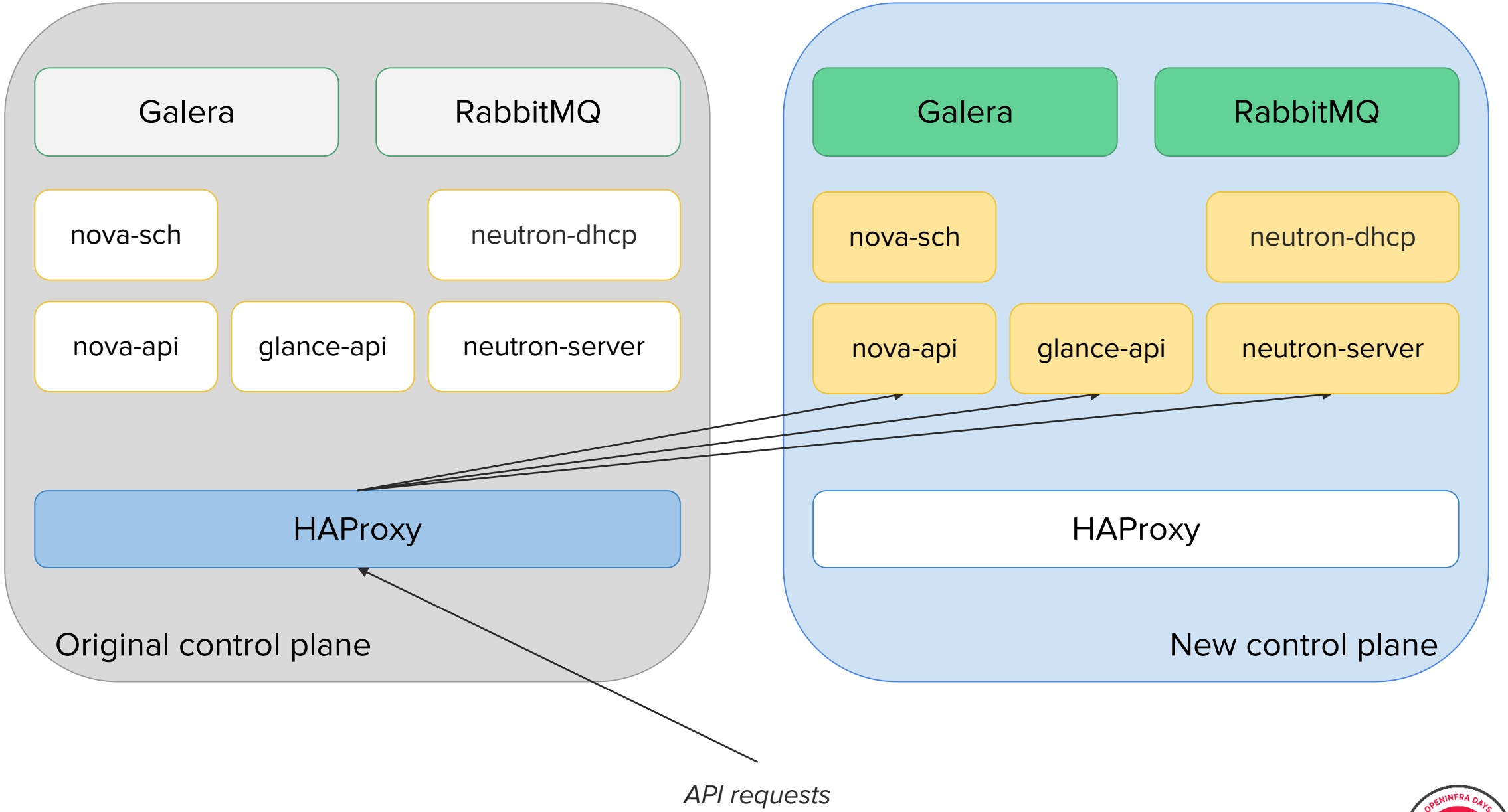

Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
nova-api
glance-api
neutron-server
New control plane
API requests

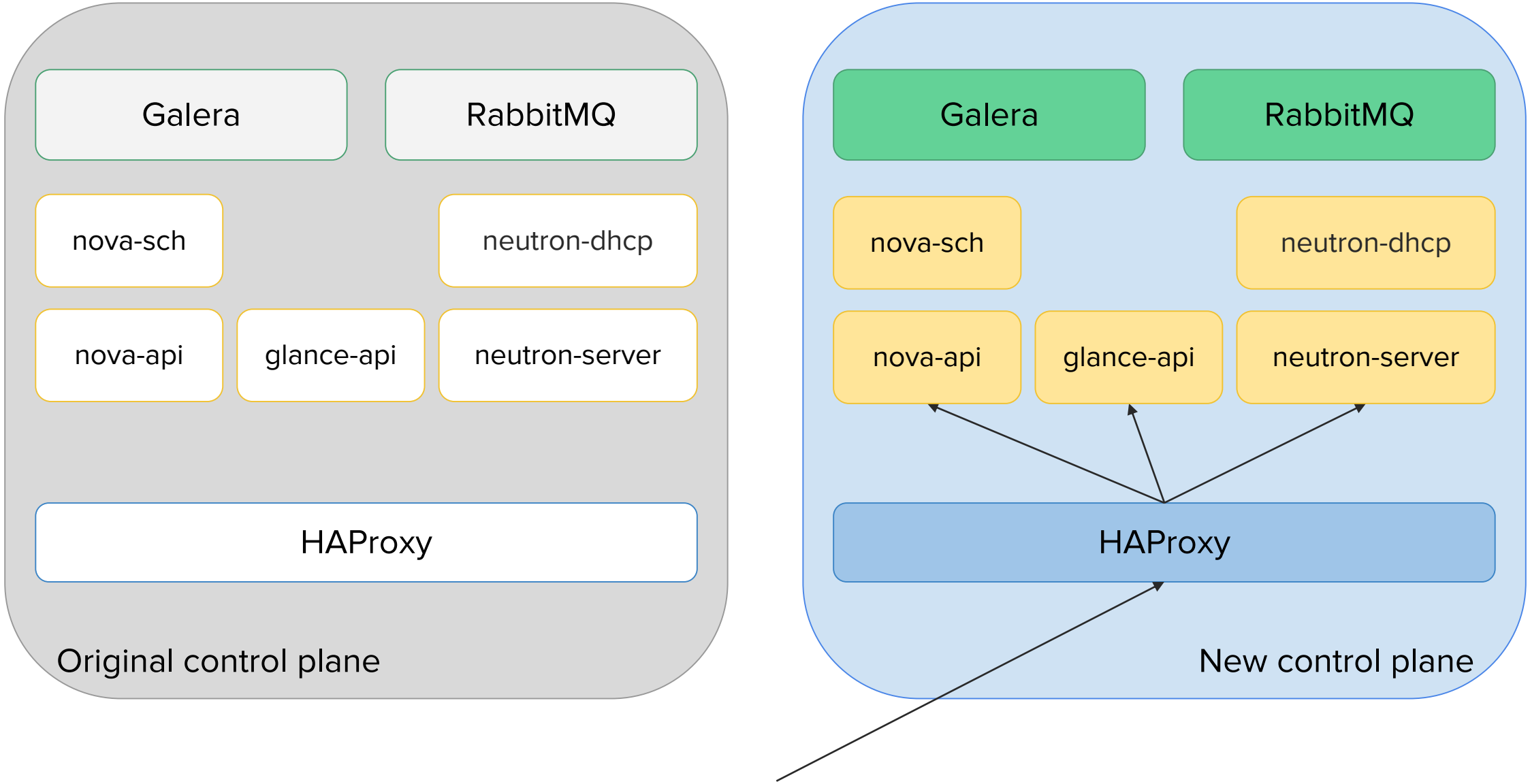

Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
New control plane
API requests


Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
New control plane
API requests


Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
New control plane
API requests

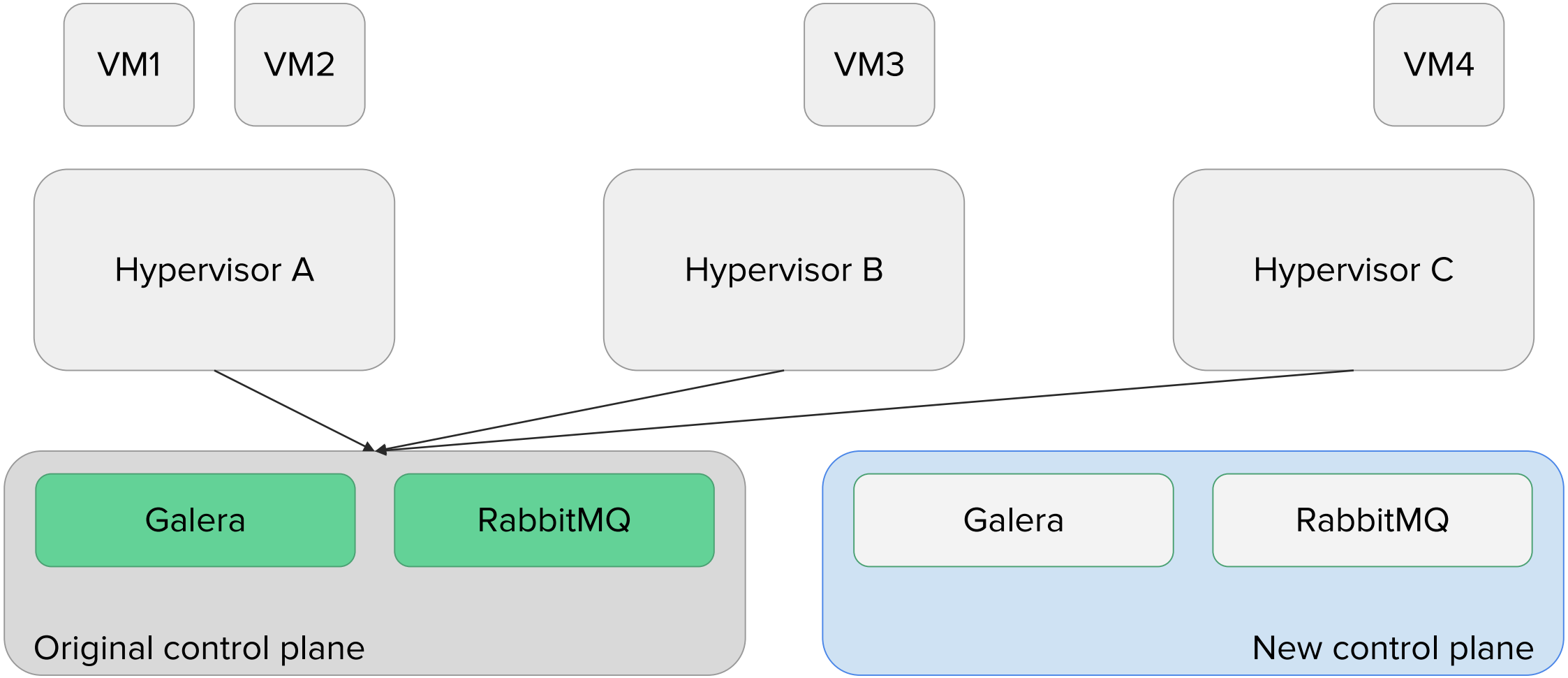

Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
New control plane
API requests

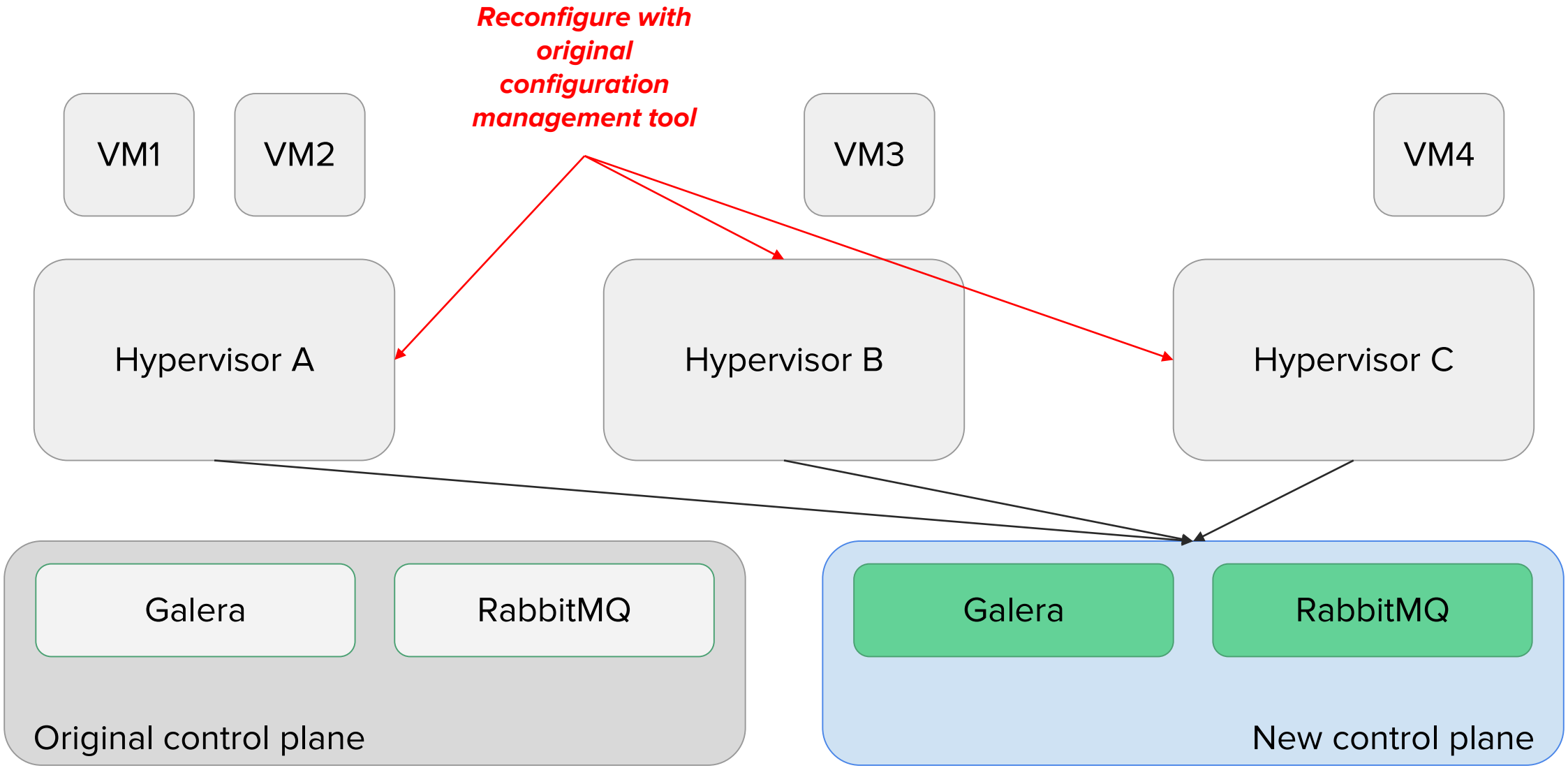

Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
New control plane
API requests

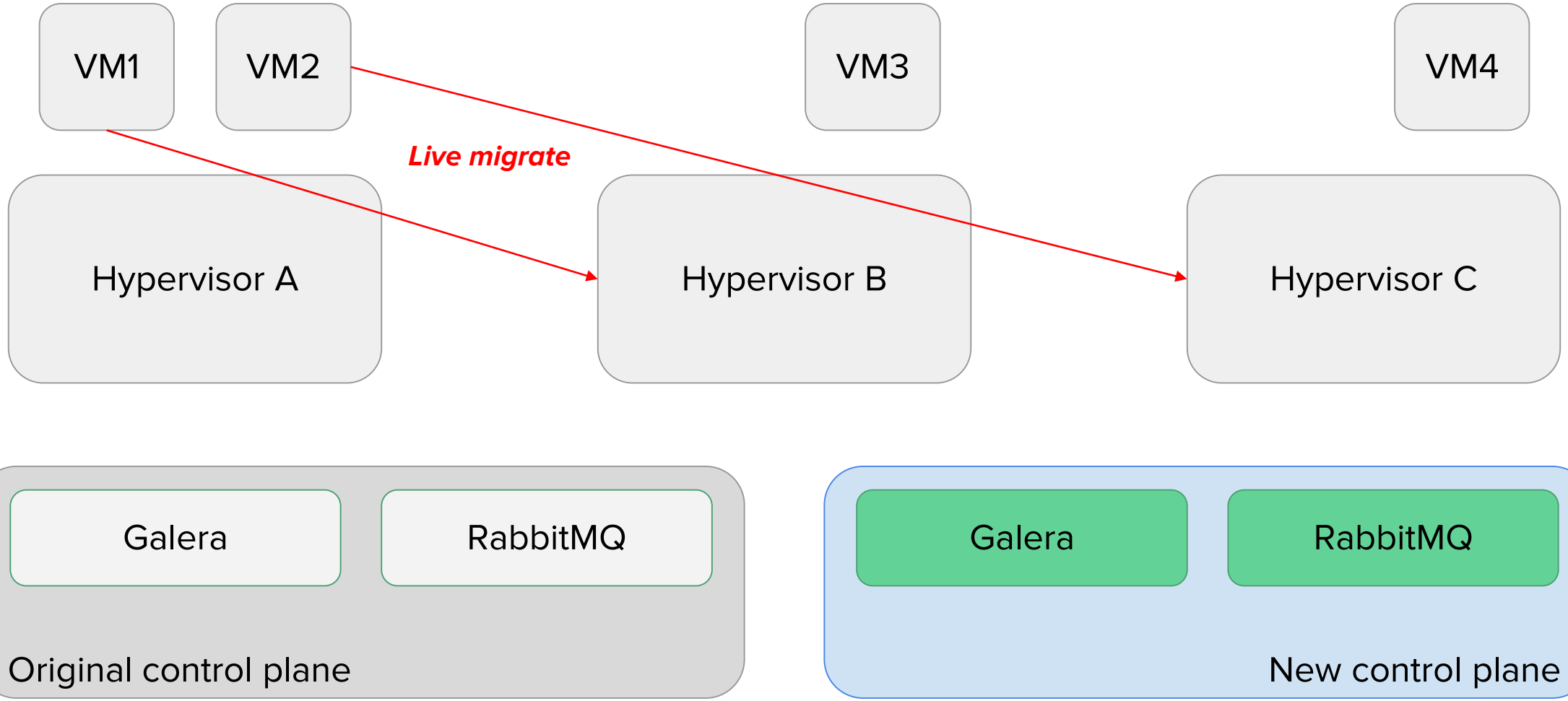

Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
Original control plane
Galera
RabbitMQ
nova-sch
neutron-dhcp
nova-api
glance-api
neutron-server
HAProxy
New control plane
API requests

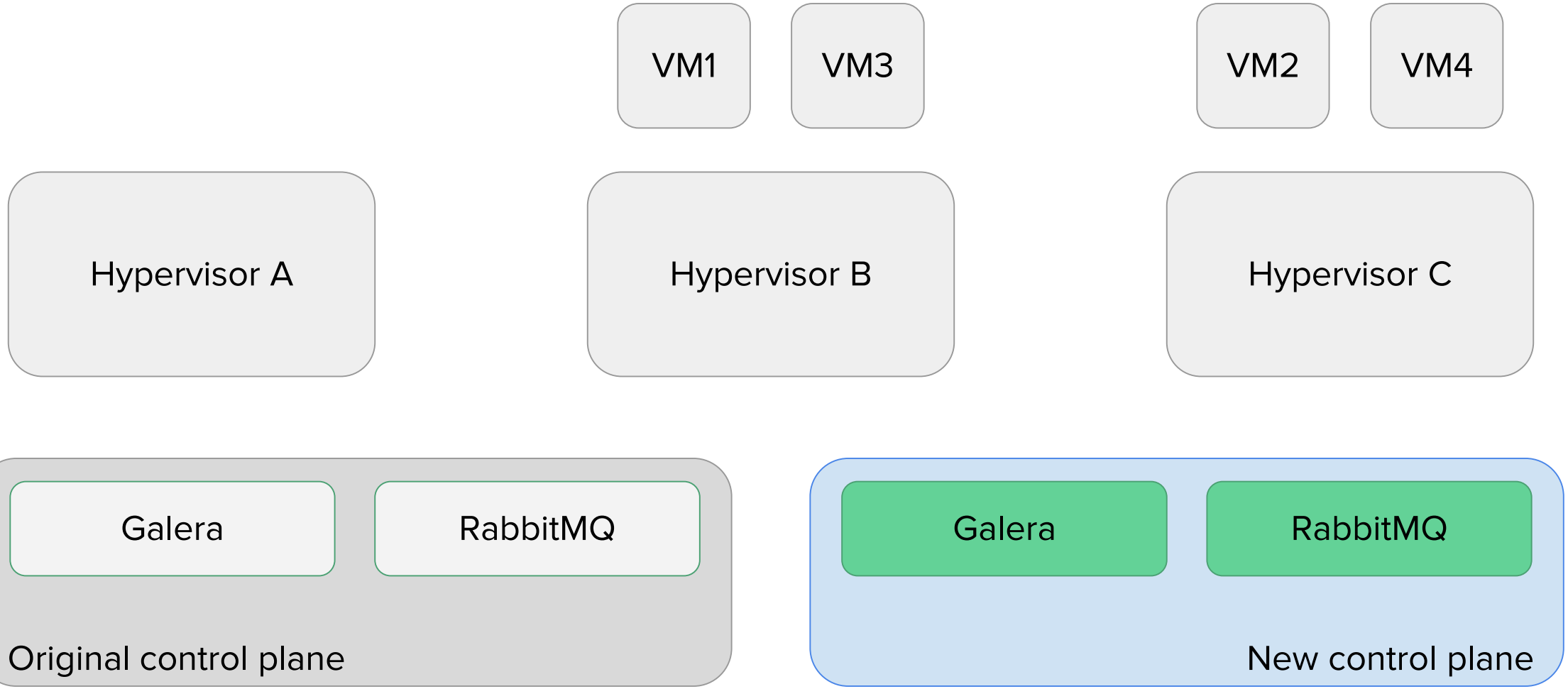
Hypervisors migration

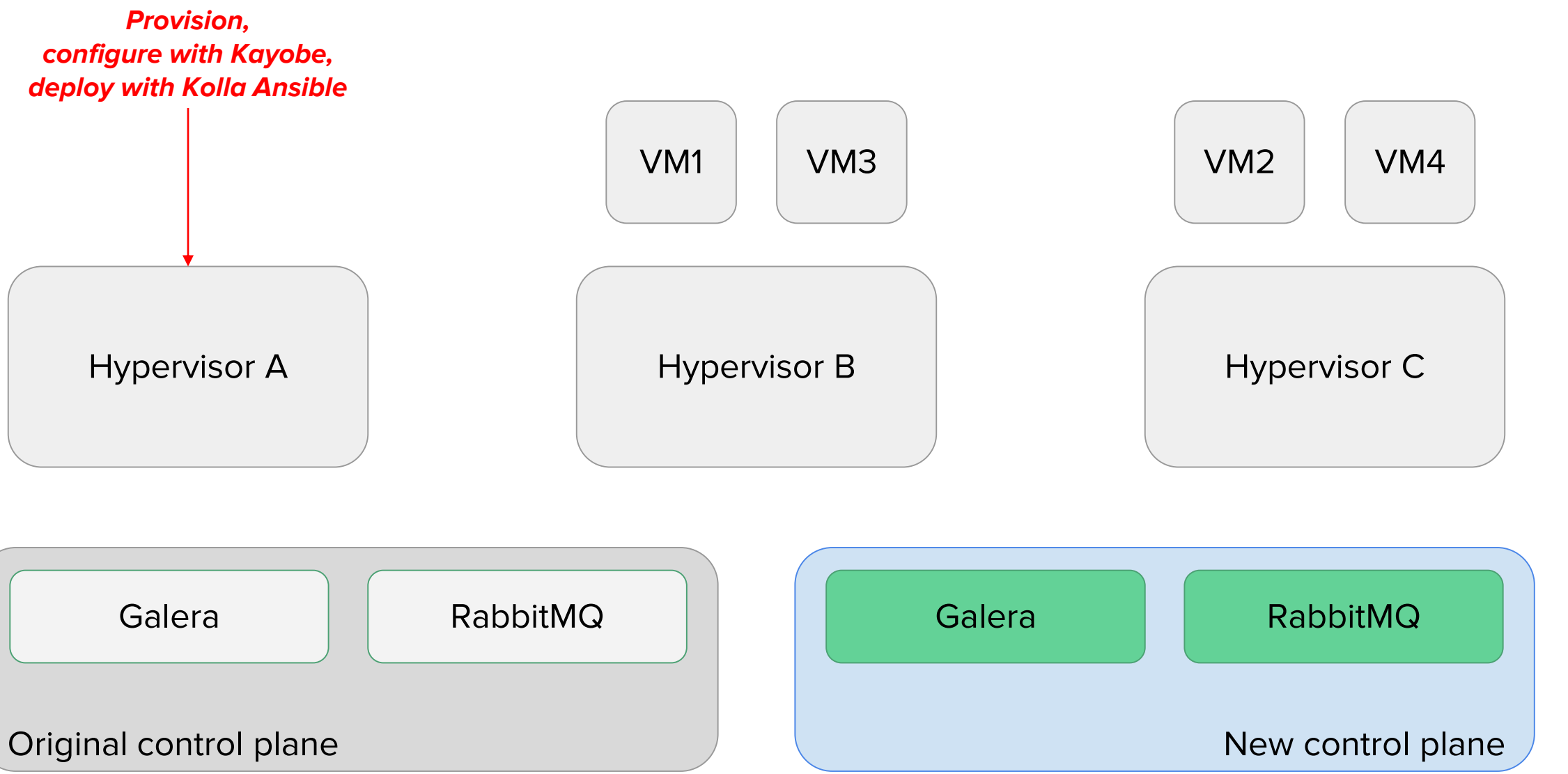
- Live migrate workload to free up hypervisor
- Reprovision node with fresh CentOS 7 image
- Configure node with Kayobe
- Deploy compute node services with Kolla
- Migrate workload back
- **Repeat**

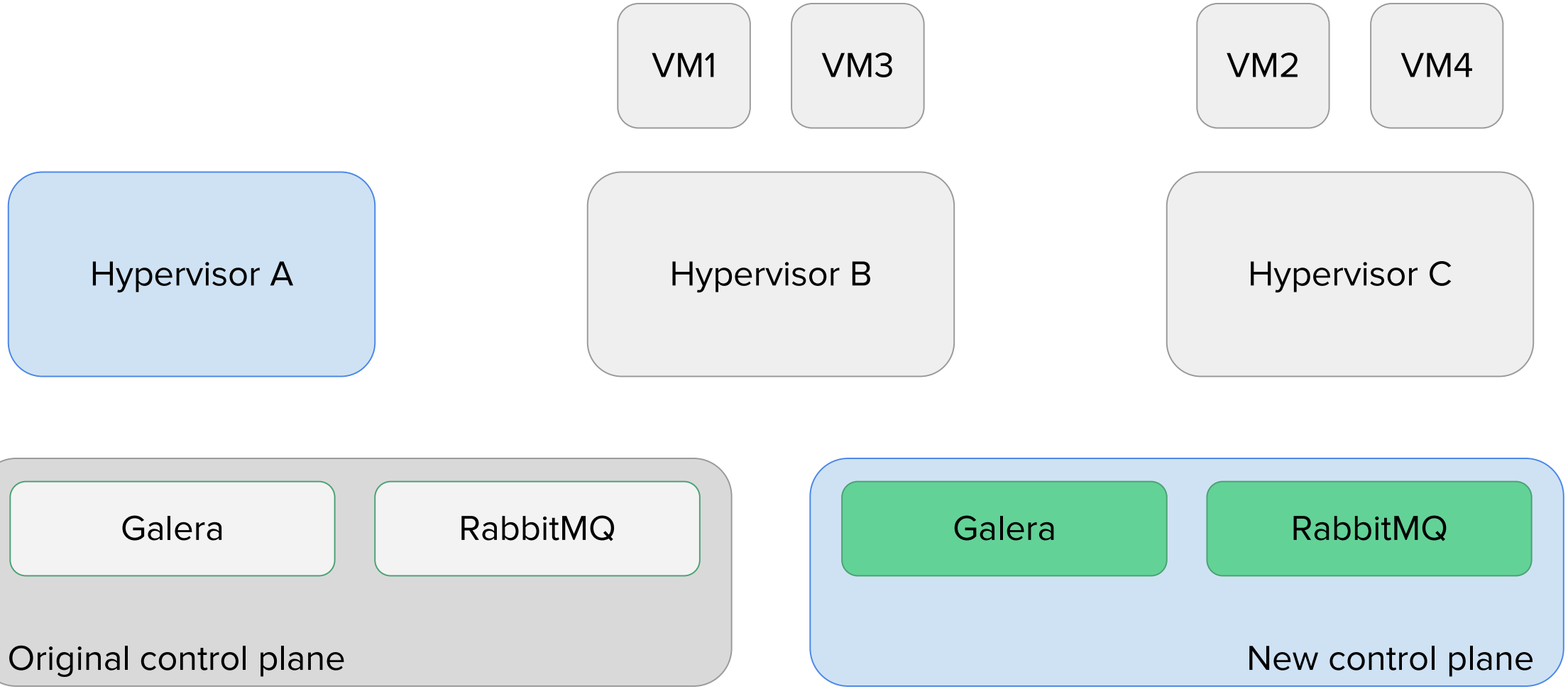

VM1
VM2
VM3
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane

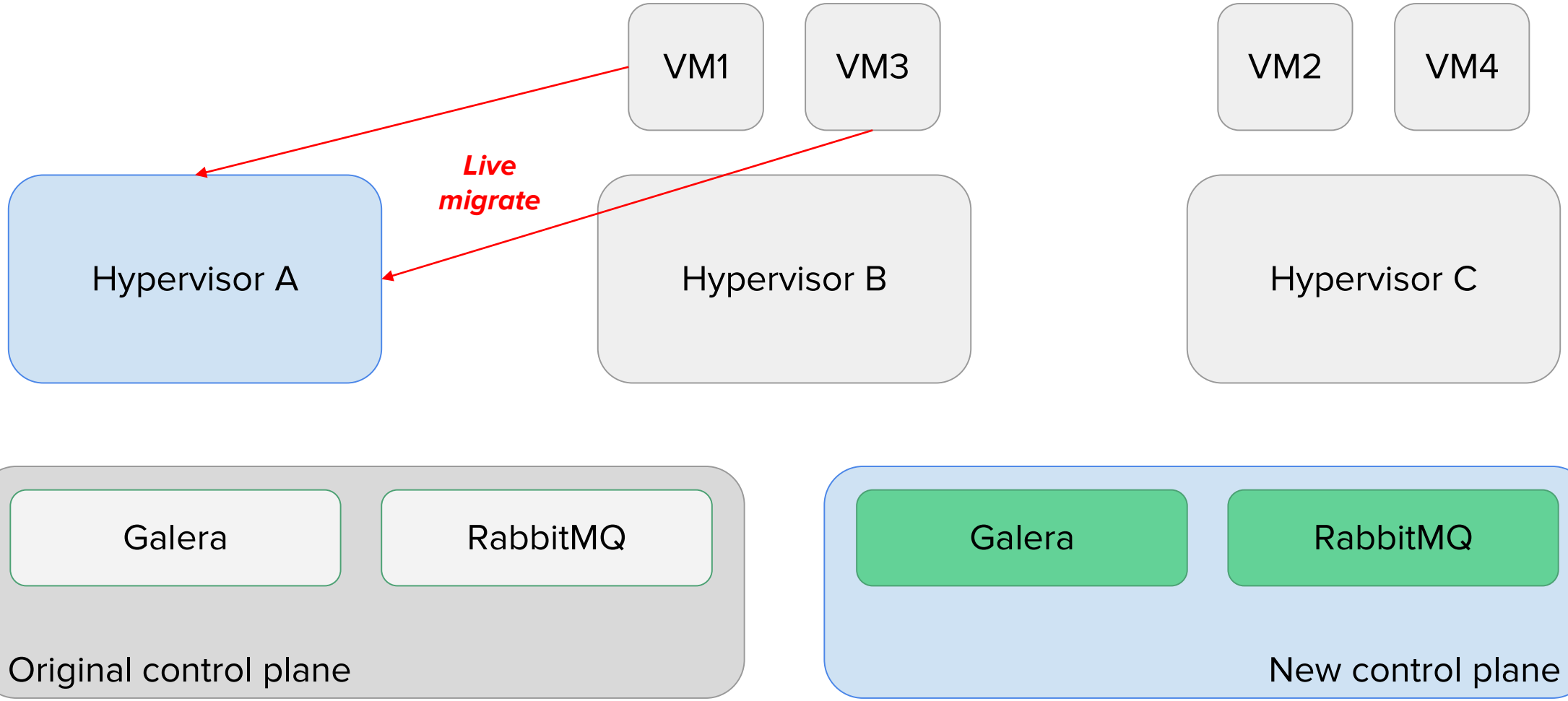

Reconfigure with original configuration management tool
VM1
VM2
VM3
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane

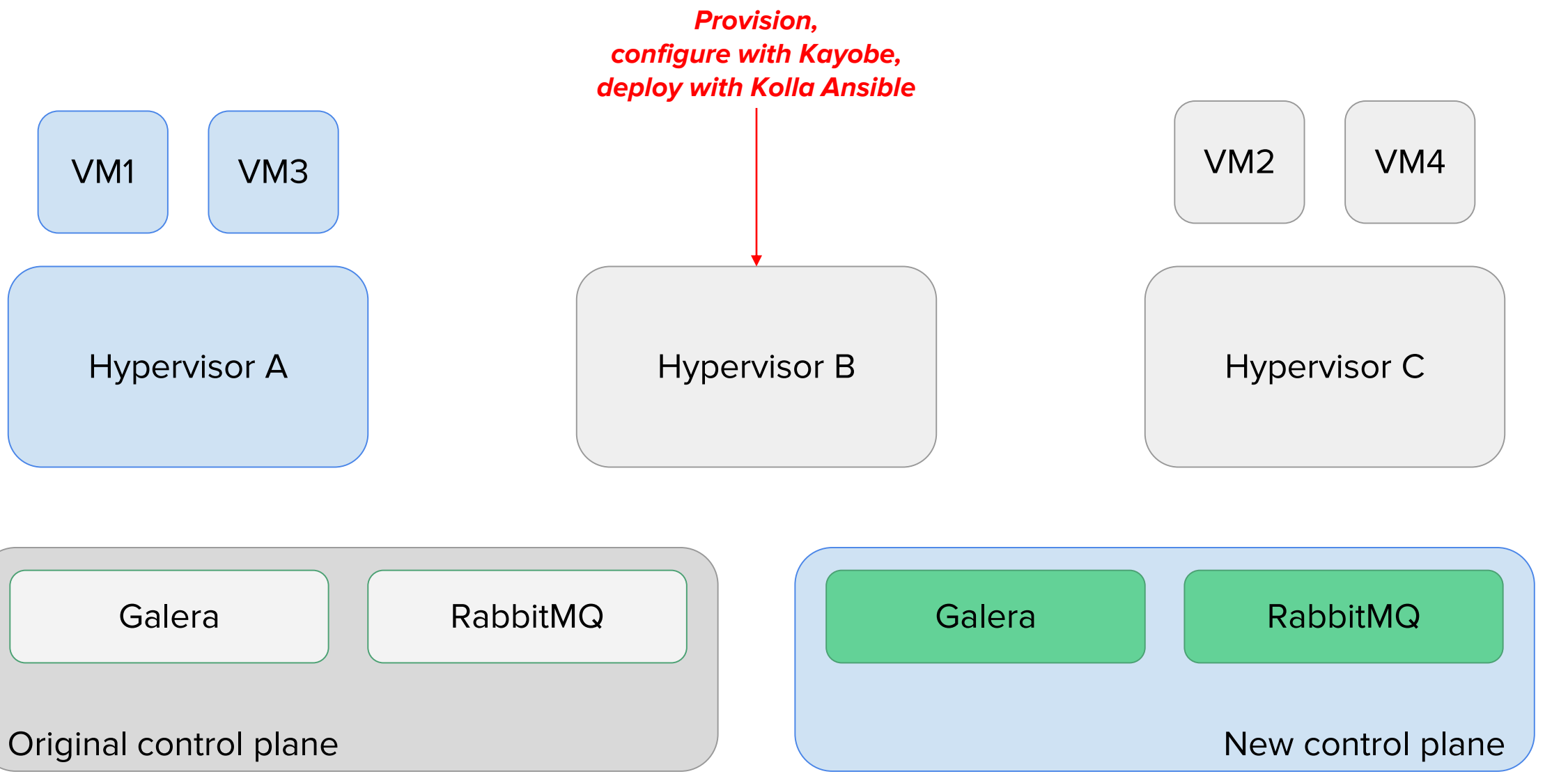

VM1
VM2
VM3
VM4
Live migrate
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane

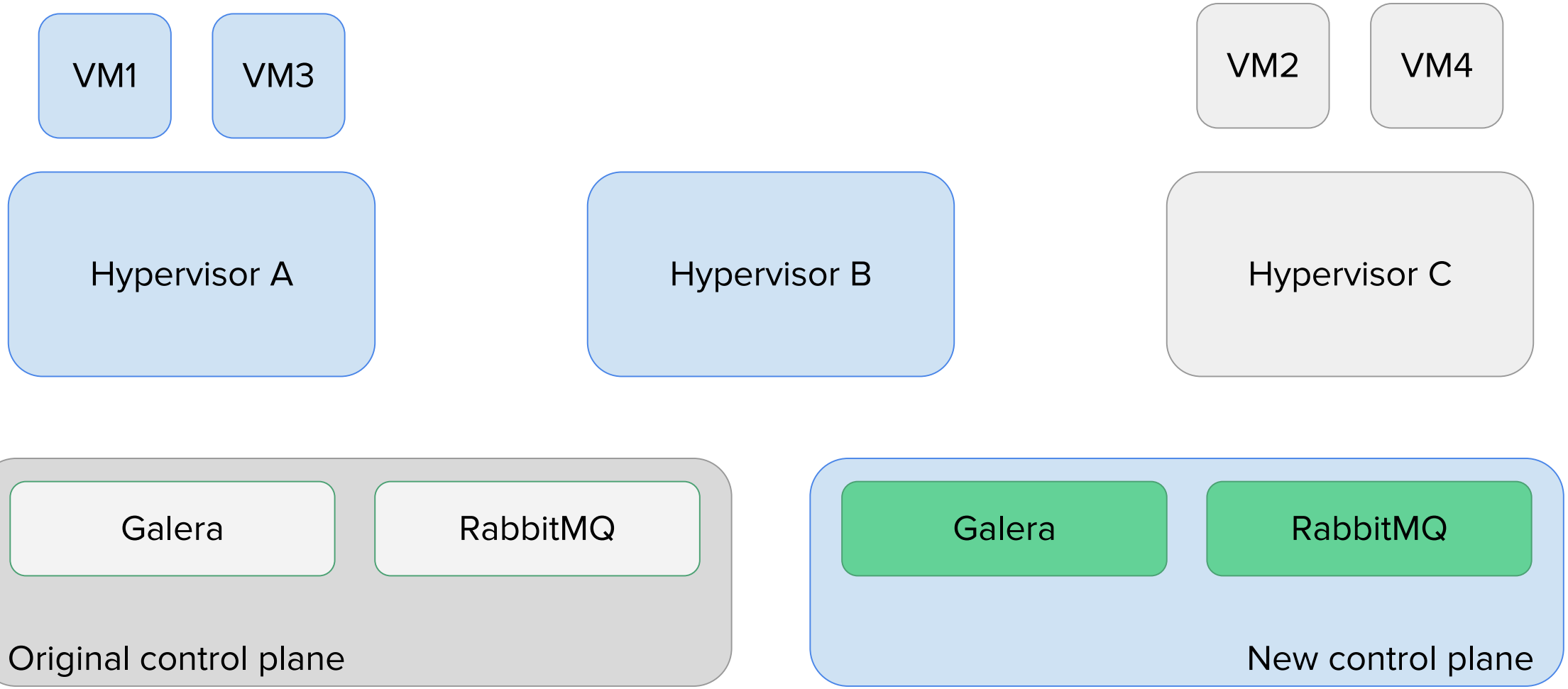

VM1
VM3
VM2
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane

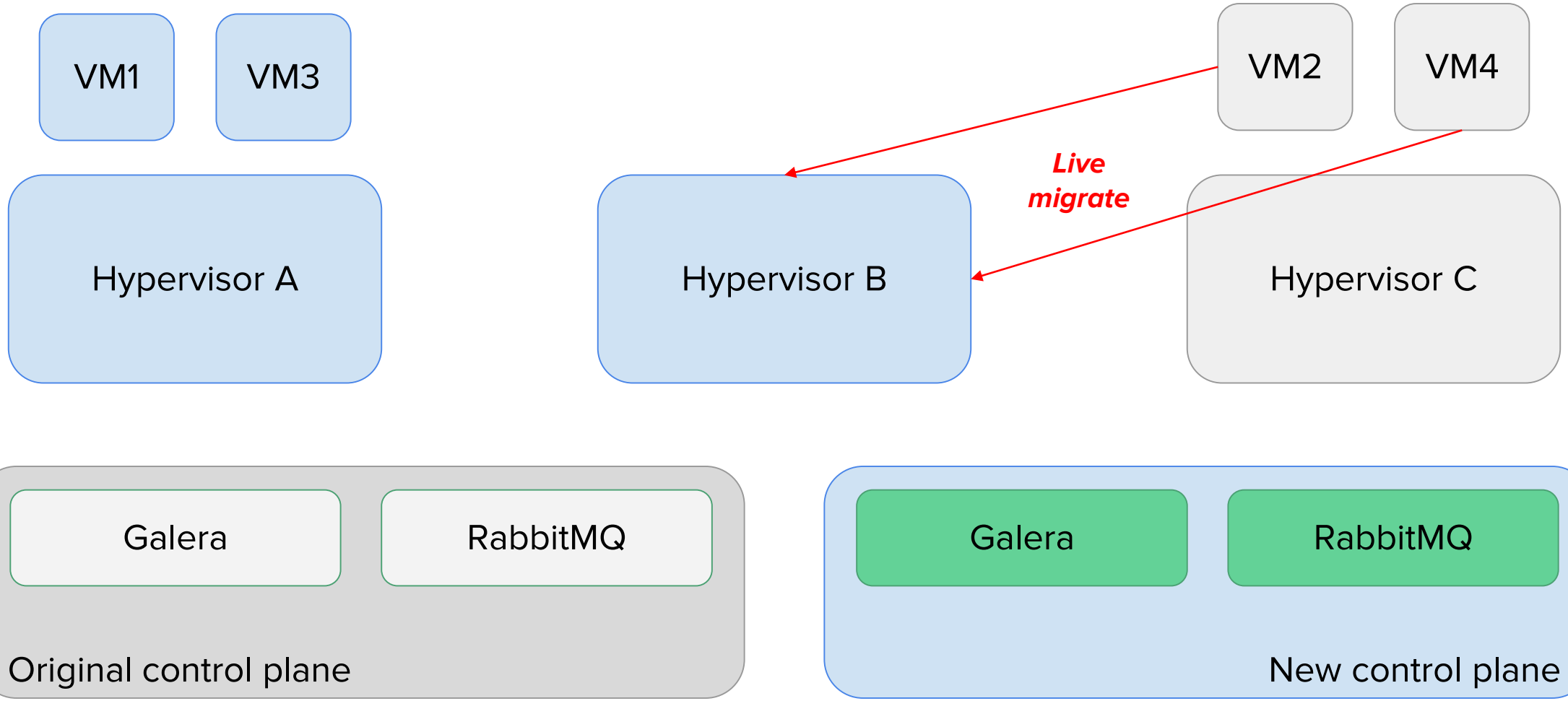

Provision,
configure with Kayobe,
deploy with Kolla Ansible
VM1
VM3
VM2
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane

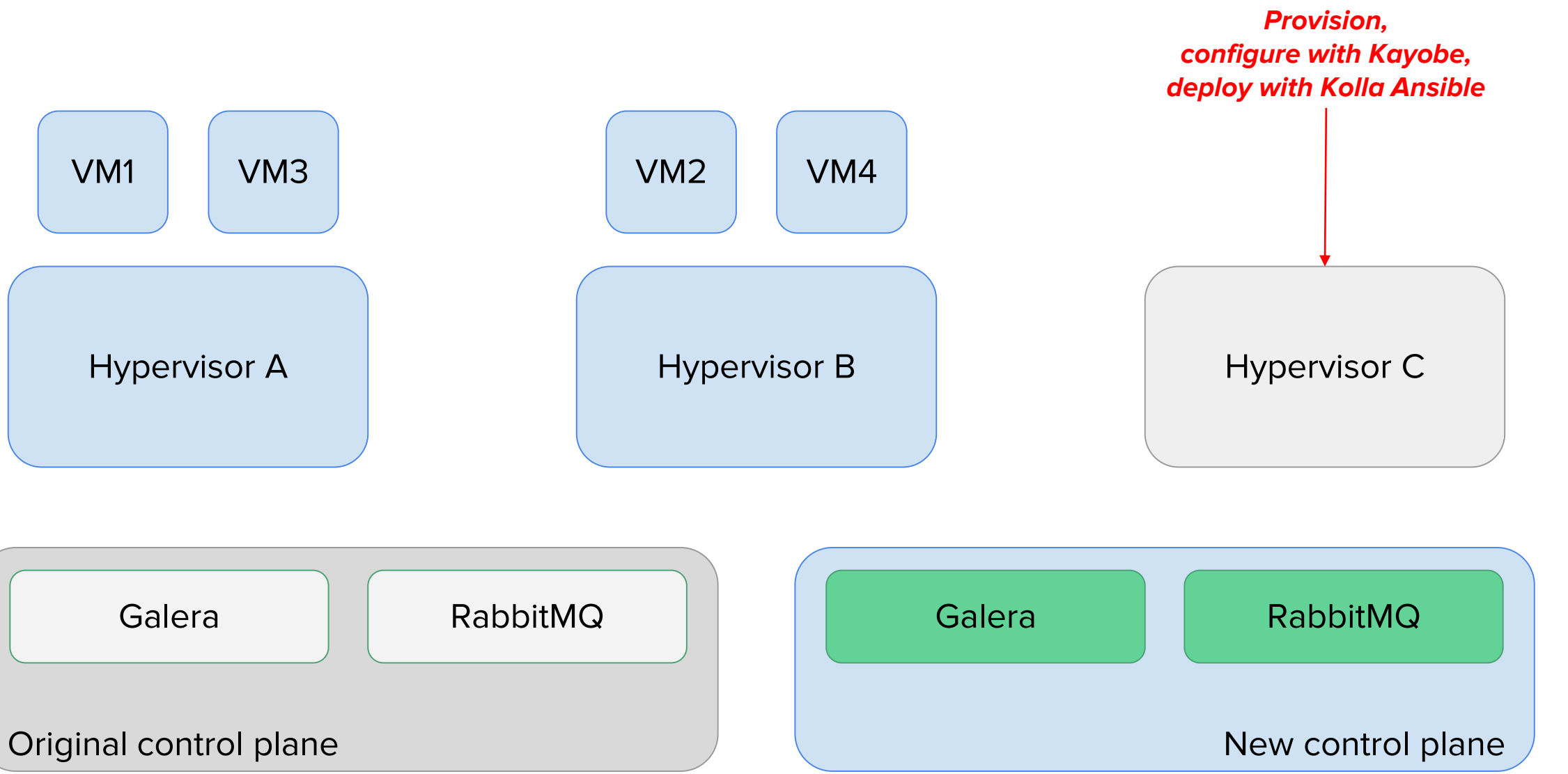

VM1
VM3
VM2
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane

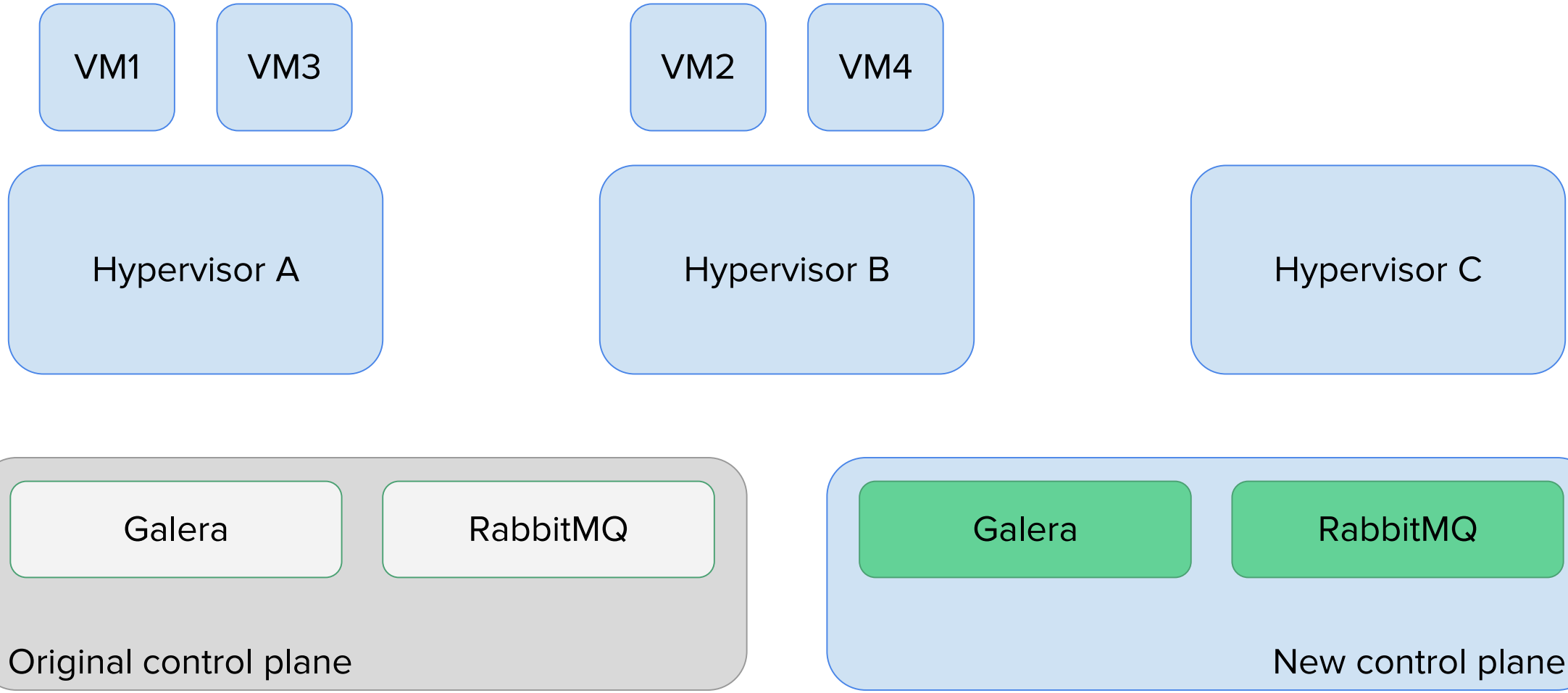

VM1
VM3
VM2
VM4
Live migrate
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane


Provision,
configure with Kayobe,
deploy with Kolla Ansible
VM1
VM3
VM2
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane


VM1
VM3
VM2
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane


VM1
VM3
VM2
VM4
Live migrate
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane


Provision,
configure with Kayobe,
deploy with Kolla Ansible
VM1
VM3
VM2
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane


VM1
VM3
VM2
VM4
Hypervisor A
Hypervisor B
Hypervisor C
Galera
RabbitMQ
Original control plane
Galera
RabbitMQ
New control plane

Tips & tricks

Translating and validating configuration

- Configuration files went through several upgrades
- Deprecated and removed options left over
- New tool in Stein: `oslo-config-validator`
- Example output:


```
ERROR:root:DEFAULT/verbose not found
WARNING:root:Deprecated opt DEFAULT/notify_on_state_change found
WARNING:root:Deprecated opt DEFAULT/notification_driver found
WARNING:root:Deprecated opt DEFAULT/auth_strategy found
WARNING:root:Deprecated opt DEFAULT/scheduler_default_filters found
```


Configuring Kolla Ansible to use existing services

- Override Ansible variables

```
rpc_transport_url: rabbit://user:pass@ctrl01:5672,user:pass@ctrl02:5672
```

- Customise Kolla inventory
  - Create groups with original hosts in Kayobe inventory

```
[ctrl_rabbitmq]
ctrl01 ansible_host=192.168.0.1
ctrl02 ansible_host=192.168.0.2
```

  - Use them in Kolla inventory

```
[rabbitmq:children]
ctrl_rabbitmq
```

Customising Kolla images

- Most of the configuration done via Kolla Ansible
- Sometimes things are hardcoded in Kolla images
- Easy to change them:
  - Fork opendev.org/openstack/kolla
  - Configure Kayobe to use Git repo of fork
  - Modify docker/<component>/<service>/*
  - `kayobe overcloud container image build`
- Only one modification on this deployment:
  - Heat domain name: `s/heat_user_domain/heat/`

Migrating resources to new services

- Cinder
  - Volumes linked to a specific cinder-volume service…
  - … even when using a shared Ceph backend

```
cinder-manage volume update_host \
    --currenthost ctrl01@rbd --newhost newctrl01@rbd
```

- Neutron
  - L3 with High Availability will automatically rebalance
  - Without L3 HA: must migrate to new agents manually

```
openstack network agent remove router --l3 <old-agent-uuid> <router-uuid>
openstack network agent add router --l3 <new-agent-uuid> <router-uuid>
```

Conclusion

In summary

- Many benefits from deploying OpenStack in **containers**
- Extensive **configuration capabilities** in Kolla Ansible
- Allows to closely **replicate** an existing deployment
- **Gradual integration** to reduce risk of migration
- **Live migration** for minimizing **user impact**

Thank you!

@priteau
pierre@stackhpc.com
https://www.stackhpc.com

More details at:

https://www.stackhpc.com/migrating-to-kolla.html