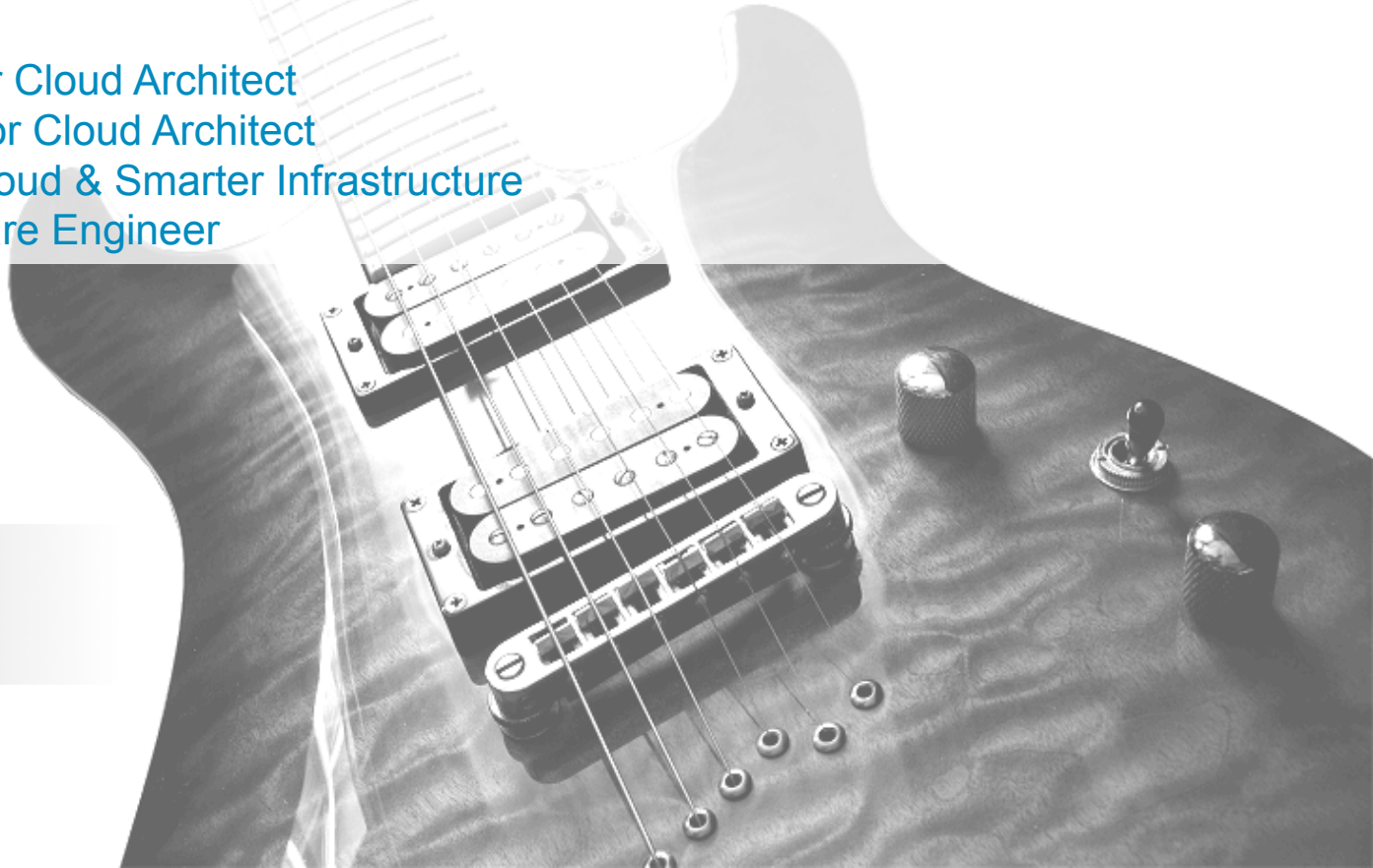
A Practical Approach to Deploying a Highly Available and Optimally Performing OpenStack

Manuel Silveyra Senior Cloud Architect
Shaun Murakami Senior Cloud Architect
Jeffrey Yang STSM, Cloud & Smarter Infrastructure
Tony Yang Staff Software Engineer



OpenStack Summit
May 12-16, 2014
Atlanta, Georgia

Agenda

- Active-Passive HA
- Demo
- Active-Active HA
- HA Orchestration with Heat and Chef
- Questions

Active-Passive High Availability

Introduction and Background

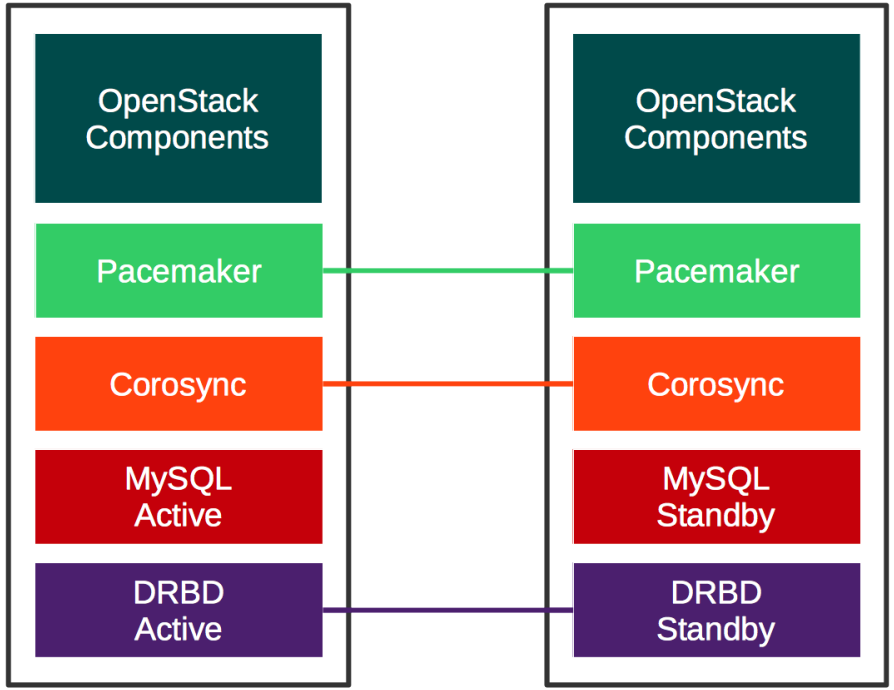
Goal:

- Database High Availability
- Data Persistence
- Persistent IP Addressing

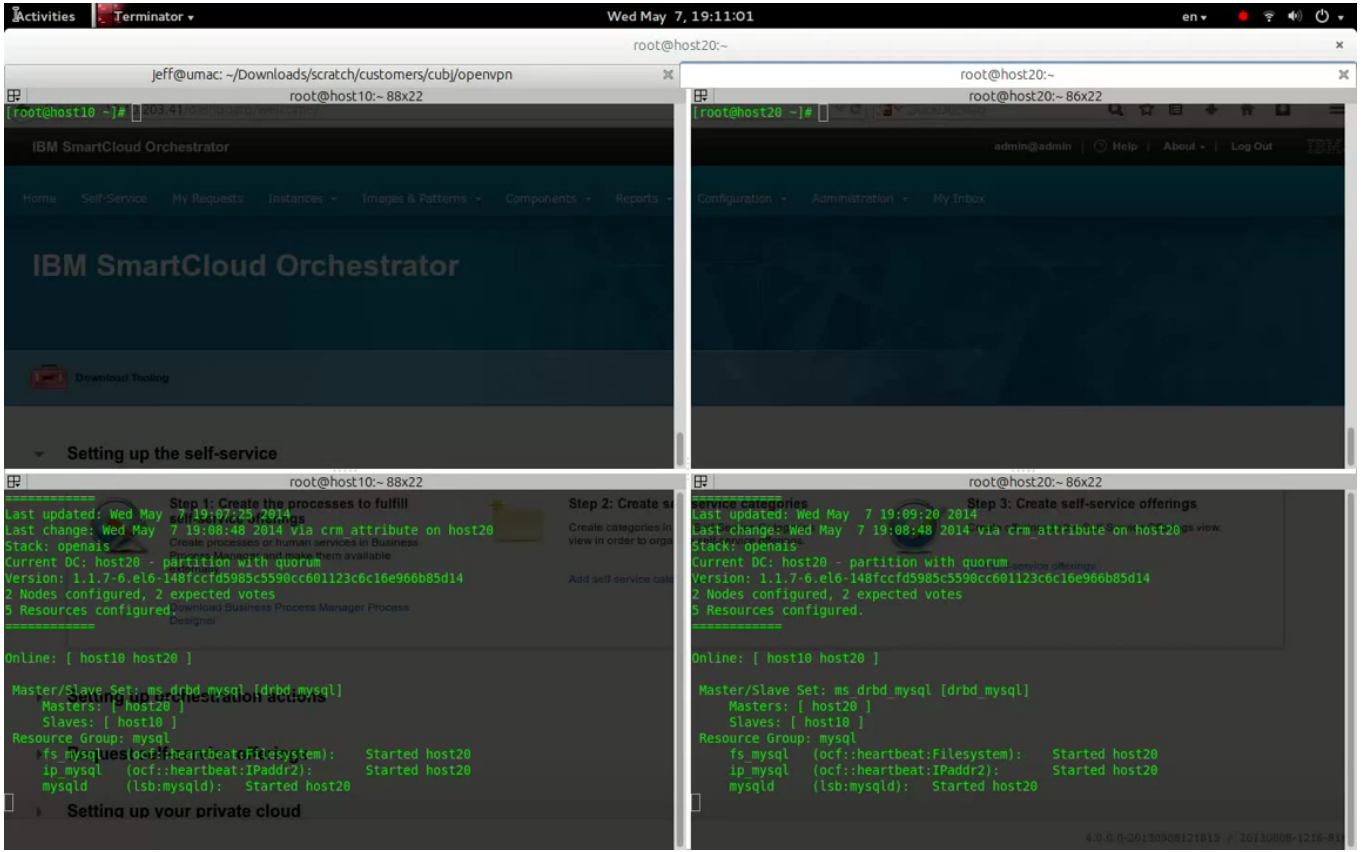
Architectural Decisions:

- For a Production Environment
- Data Persistence was Paramount

High Level Architecture



Demo



Active-Active High Availability

Introduction and Background

Goal:

- Improve the stability, reliability, and scalability over previous OpenStack deployments.
- Provide a robust platform for PaaS workloads.

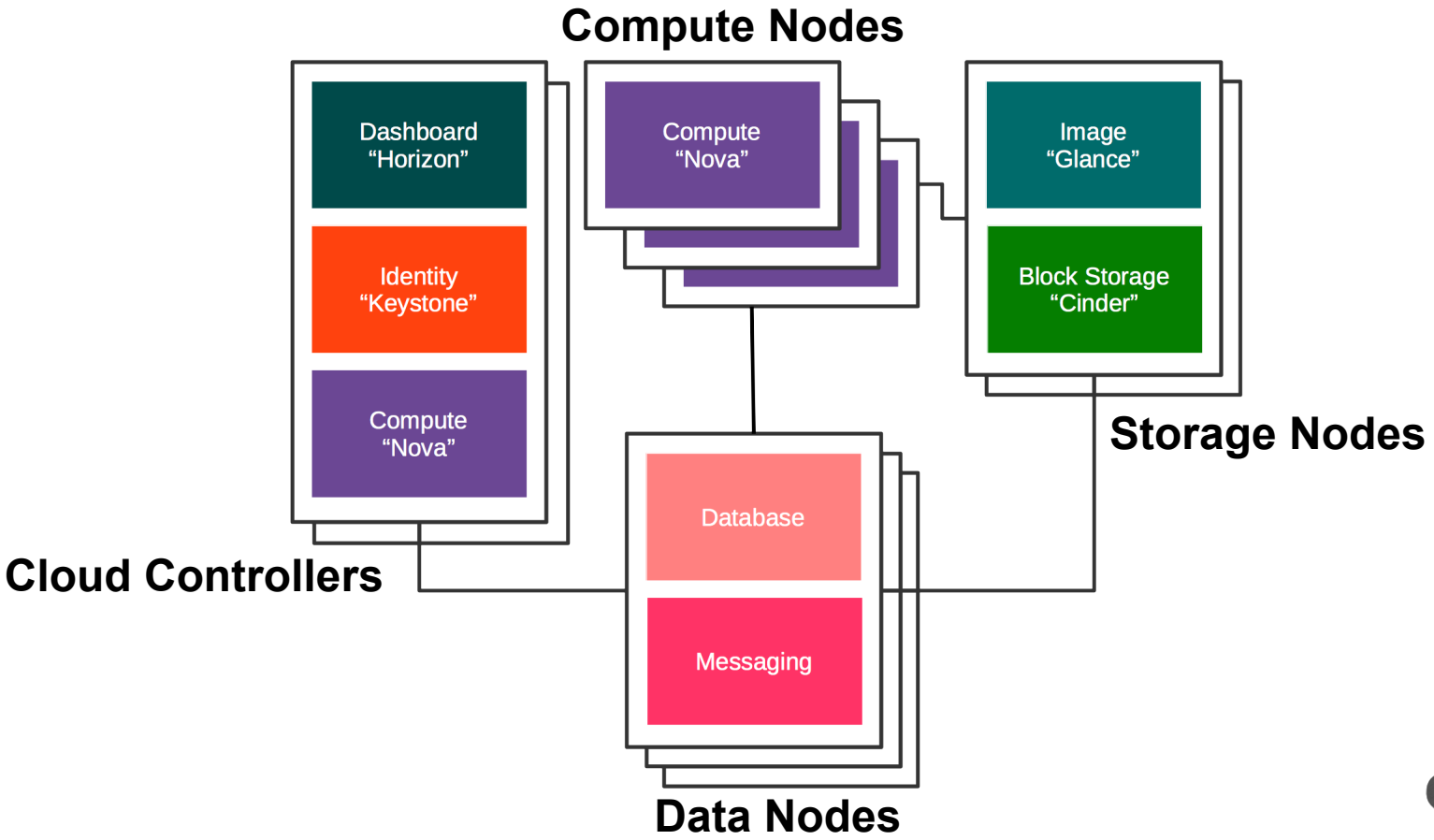
Cloud Foundry Workload Characteristics:

- Bursty deployments
- Large storage consumption
- High network I/O
- High API utilization

Architectural Decisions

- **Scale Out** vs. Scale Up
  - Makes it possible to meet workload capacity demands
  - Compliments Cloud Foundry's resilient architecture
- **Active-Active** vs. Active-Passive HA
  - Distributed utilization
  - Improved response time
  - Improved failover time

Architecture Overview



Messaging HA – RabbitMQ Clustering

- RabbitMQ Clustering is easy to set up:

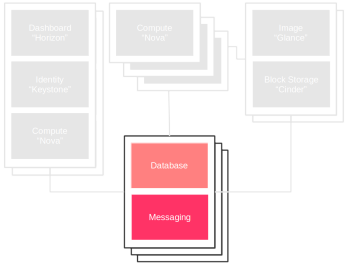
  Copy .erlang.cookie file to all servers

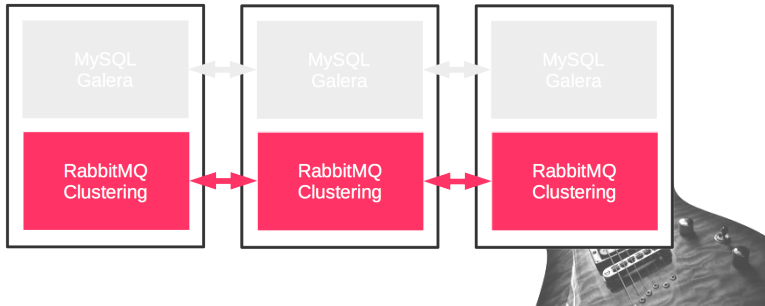
```
rabbitmq-server -detached
rabbitmqctl stop_app
rabbitmqctl join_cluster <server>
rabbitmqctl start_app
```

- Define the HA queues (for version 3+):

```
rabbitmqctl set_policy HA '^(?!amq\.).*' '{"ha-mode": "all"}'
```

- RabbitMQ Monitoring helps diagnose some performance issues

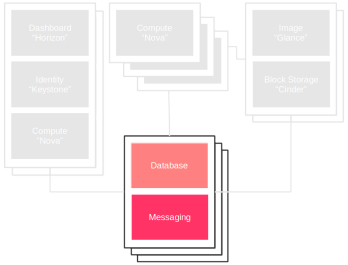


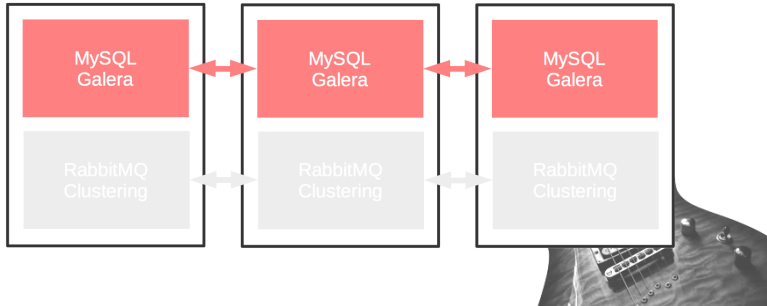


Database HA – MySQL & Galera

- Galera replication works…
  - Except when multiple nodes try to update the same row, then Galera returns a deadlock.
- Use Active/Standby configuration for the cluster whenever you will write to the database.
- Performance tweaks:

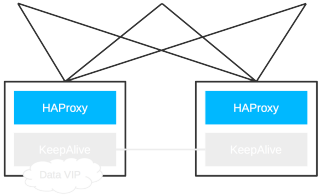
```
max_connections=1000
key_buffer_size=2048M
innodb_buffer_pool_size=4096M
thread_cache_size=32
table_cache=1024
```





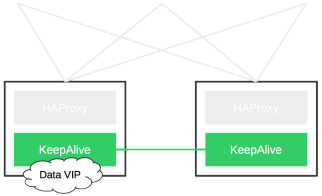
Load Balancing with a Virtual IP

- The use of a load balancer allows us to quickly and easily scale-out and manage services behind them



HAProxy – Load balancer

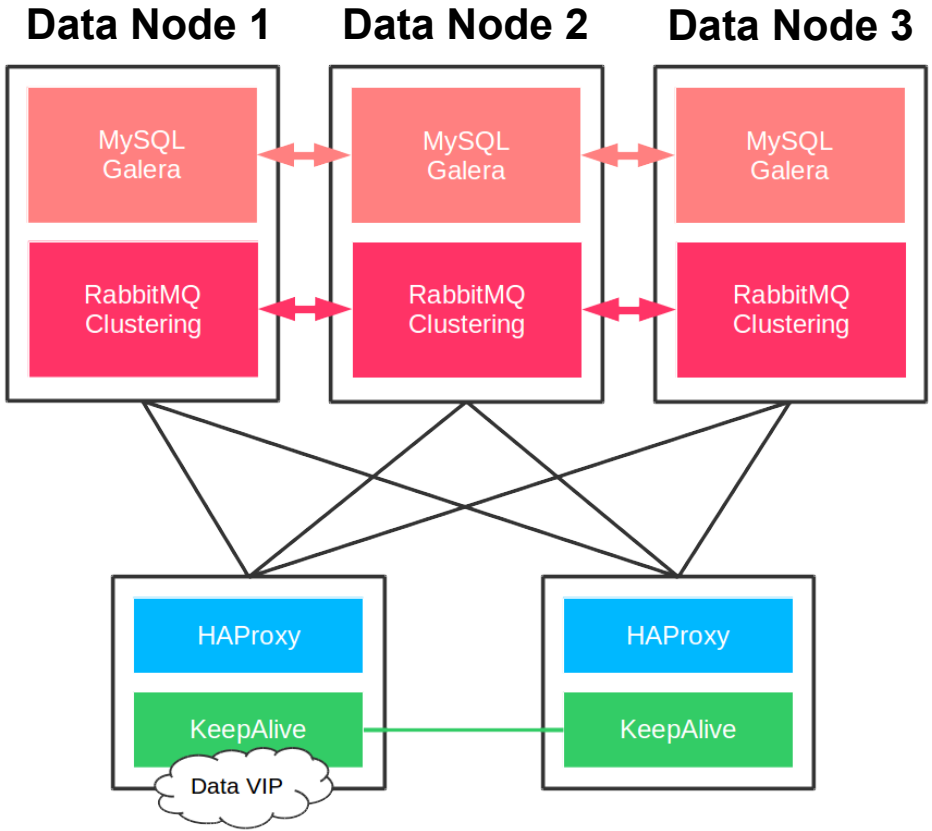
- The Stats functionality is a great way to monitor and debug the environment
- Timeouts matter. We're still tweaking so your suggestions would be welcome.
  - In our data node, we found that much longer timeouts worked best.
  - For the other services, defaults available in the web worked well.



KeepAlived – Manages the virtual IP

- Remember to have unique `virtual_router_id`'s for each cluster in your environment

HA across the Data Nodes

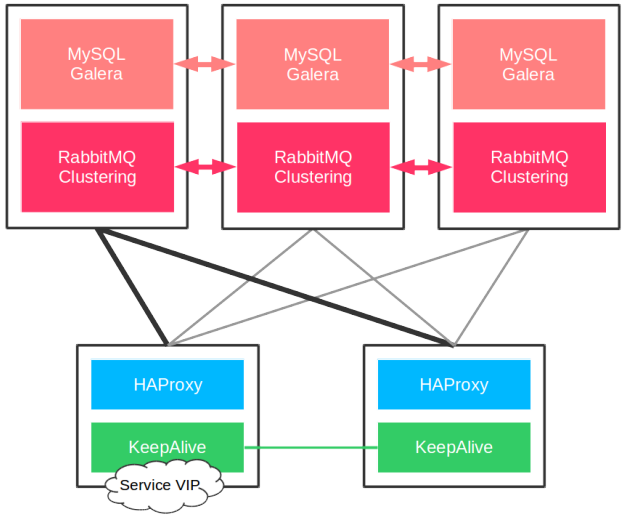


Primary Active-Passive Configuration

HAProxy configuration

```
listen mysql-cluster
    bind *:3306
    mode tcp
    option tcpka
    option mysql-check user haproxy_check
    balance leastconn
    server mysql-1 10.81.25.194:3306 check
    server mysql-2 10.81.25.195:3306 check backup
    server mysql-3 10.81.25.196:3306 check backup
```

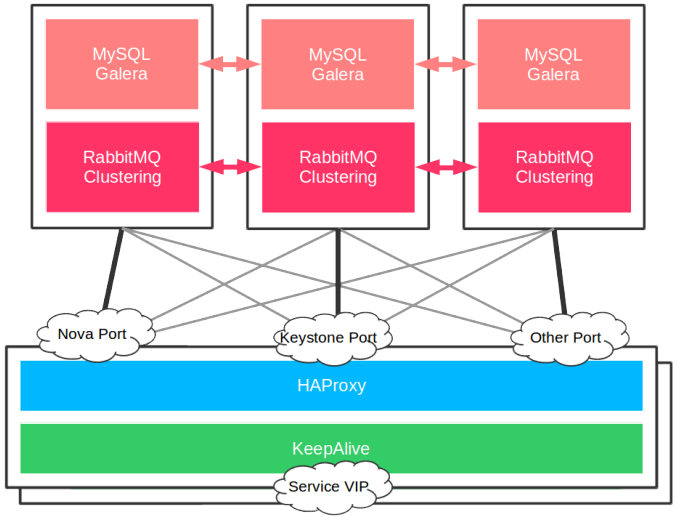
Active-Active data replication



Single primary target

Distributed Active-Passive Configuration

Active-Active data replication



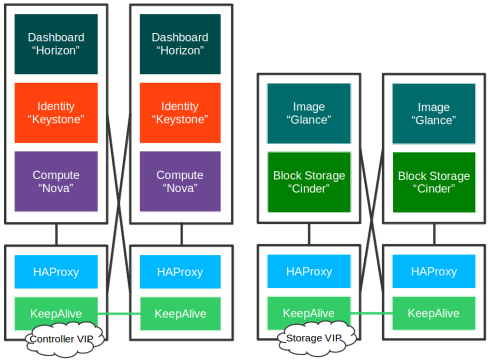
Multiple primary targets

HAProxy configuration

```
listen mysql-cluster-nova | keystone | etc
    bind *:3307 | 3308 | 3309
    mode tcp
    option tcpka
    option mysql-check user haproxy_check
    balance leastconn
    server mysql-1 10.81.25.194:3306 check         | backup | backup
    server mysql-2 10.81.25.195:3306 check backup  |        | backup
    server mysql-3 10.81.25.196:3306 check backup  | backup |
```

OpenStack Services

- OpenStack services should be registered in Keystone with the corresponding VIP as it's target IP



Services

Filter Filter

Name	Service	Host	Enabled
cinder	volume	10.81.72.201	Enabled
glance	image	10.81.72.201	Enabled
nova	compute	10.81.72.200	Enabled
keystone	identity (native backend)	10.81.72.200	Enabled

Displaying 4 items

Storage LB VIP

Contoller LB VIP

Cloud Controller HA

- SQL configuration in service.config files should point to their ports

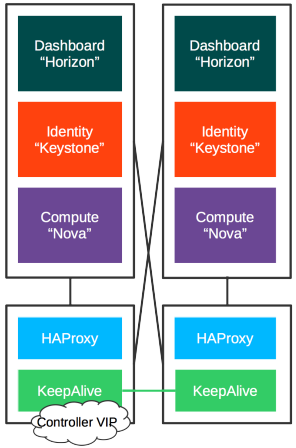
```
sql_connection = mysql://svc_user:svc_password@mysql_lb_ip:port/service_database
```

- Enable HA queues in service.config files

```
rpc_backend = nova.openstack.common.rpc.impl_kombu
rabbit_hosts = 10.81.25.194:5672,10.81.25.195:5672,10.81.25.196:5672
rabbit_ha_queues = True
```

- Haproxy Configuration
  - /etc/haproxy/haproxy.conf

```
listen service_name
  bind *:service_port
  balance roundrobin
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 controller1_IP:service_port check inter 2000 rise 2 fall 5
  server controller2 controller2_IP:service_port check inter 2000 rise 2 fall 5
```



Storage Node HA

- SQL configuration in service.config files should point to their ports

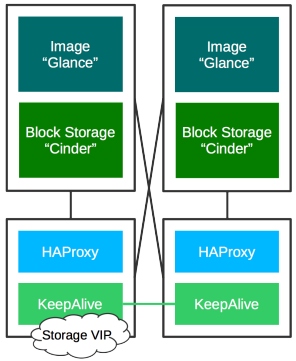
```
sql_connection = mysql://svc_user:svc_password@mysql_lb_ip:port/service_database
```

- Enable HA queues in service.config files

```
rpc_backend = nova.openstack.common.rpc.impl_kombu
rabbit_hosts = 10.81.25.194:5672,10.81.25.195:5672,10.81.25.196:5672
rabbit_ha_queues = True
```

- Haproxy Configuration
  - /etc/haproxy/haproxy.conf

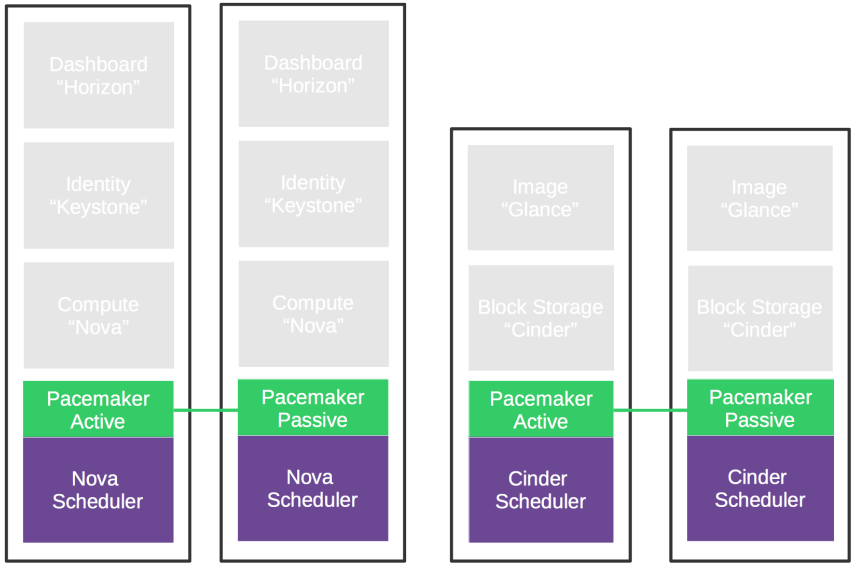
```
listen service_name
  bind *:service_port
  balance source
  option  tcpka
  option  httpchk
  option  tcplog
  server controller1 controller1_IP:service_port check inter 2000 rise 2 fall 5
  server controller2 controller2_IP:service_port check inter 2000 rise 2 fall 5
```



Active-Passive Nova/Cinder Schedulers

Pacemaker – Cluster Resource Manager

- Disable STONITH
- Ignore the quorum policy
- Set resource sticky-ness to prevent resource fallbacks



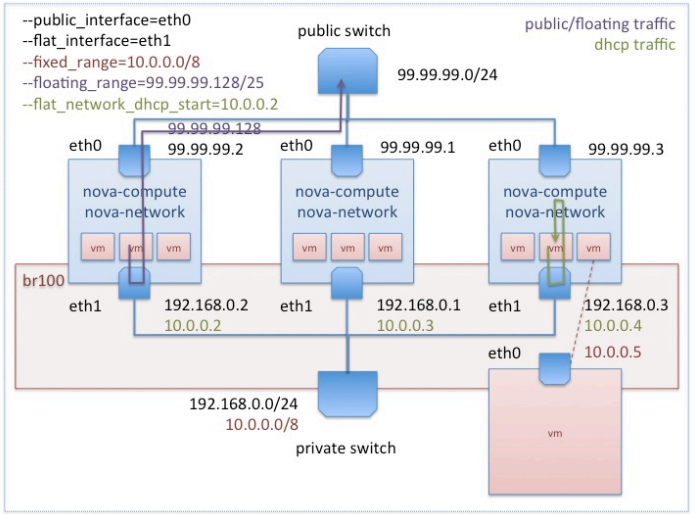
Network HA – Nova Network Multi Host

- No single point of failure
- Each compute node acts as its own gateway
- Failure of a compute node will not affect VMs on other nodes

- Compute Host must run the following services:
  - openstack-nova-compute
  - openstack-nova-network
  - openstack-nova-api-metadata

- Nova Configuration
  - /etc/nova/nova.conf

```
[Default]
multi_host=True
send_arp_for_ha=True
update_dns_entries=True
dns_update_periodic_interval=60
```



Lessons Learned

- Nova and Cinder Scheduler in HA environments
  - The service(s) listens on the bus and responds to messages.
  - No coordination between services.
  - Example: Duplication of provision requests to a single compute node.
  - Solution: A way to guarantee a single server will respond to a request.
- MySQL/Galera write locks
  - Simultaneous writes can cause errors or deadlocks.
  - Solution: Provide OpenStack with a way to write to a port and read from a different port.
- Out of the box configurations are too broad.
  - Further investigation into tuning options is required.
- A couple of bugs have been opened with (and fixed in) OpenStack related to `rabbit_hosts`
  - Our suggestion was to manually define different order per configuration file, but this has been recently fixed.
  - https://review.openstack.org/#/c/81962/

HA Orchestration with Heat and Chef

Problems

- An installer should be there to streamline the installation/configuration steps

- An installer might not be enough…
  - Usually designed from a development/test perspective
  - Not able to satisfy real, complex production need – network topologies, high availability, etc.

Heat + Chef

- Chef to manipulate a single node
  - Cookbooks, roles, environments
- Heat to manage the whole deployment
  - Templates

Deployment service

- Under cloud + over cloud
- Under cloud
  - An all-in-one OpenStack deployment
  - Used to spawn over clouds
- Over cloud
  - The actual service
  - Described by templates
  - Where all possibilities lie

Deployment service (cont.)

Resources

```
"control": {
      "Type": "IBM::SCO::Node",
      "Properties": {
            "Address": "CENTRAL2_ADDR",
            "User": "root",
            "Password": "passw0rd",
            "KeyFile": "/home/heat/.ssh/zq_key”
      },
      "Metadata": {
            "chef-runlist": "role[primary]",
            "order": 1
      }
},
```

```
"standby": {
      "Type": "IBM::SCO::Node",
      "Properties": {
            "Address": "CENTRAL3_ADDR",
            "User": "root",
            "Password": "passw0rd",
            "KeyFile": "/home/heat/.ssh/zq_key”
      },
      "Metadata": {
            "chef-runlist": "role[secondary]",
            "order": 2
      }
},
```

Environments

```
"orchestration": {
    "debug": "ENABLE_DEBUG",
    "identity_service_chef_role": "os-identity",
    "rabbit_server_chef_role": "os-ops-messaging”
},

...

"network": {
    "service_type": "NETWORK_TYPE",
    "network_manager": "NETWORK_MANAGER”
},
```

Parameters

```
"EnableOSDebug": {
    "Description": "Enable OpenStack debug mode",
    "Type": "String",
    "Default": "false”
},
"NetworkManager": {
    "Description": "Network manager type of OpenStack",
    "Type": "String",
    "Default": "nova.network.manager.VlanManager”
},
```

Outputs

```
"ENABLE_DEBUG": {
    "Value": {
        "Ref": "EnableOSDebug”
    }
},
"NETWORK_MANAGER": {
    "Value": {
        "Ref": "NetworkManager”
    }
},
```

Questions