Microclouds for Fragmented Markets

Getting OpenStack everywhere!

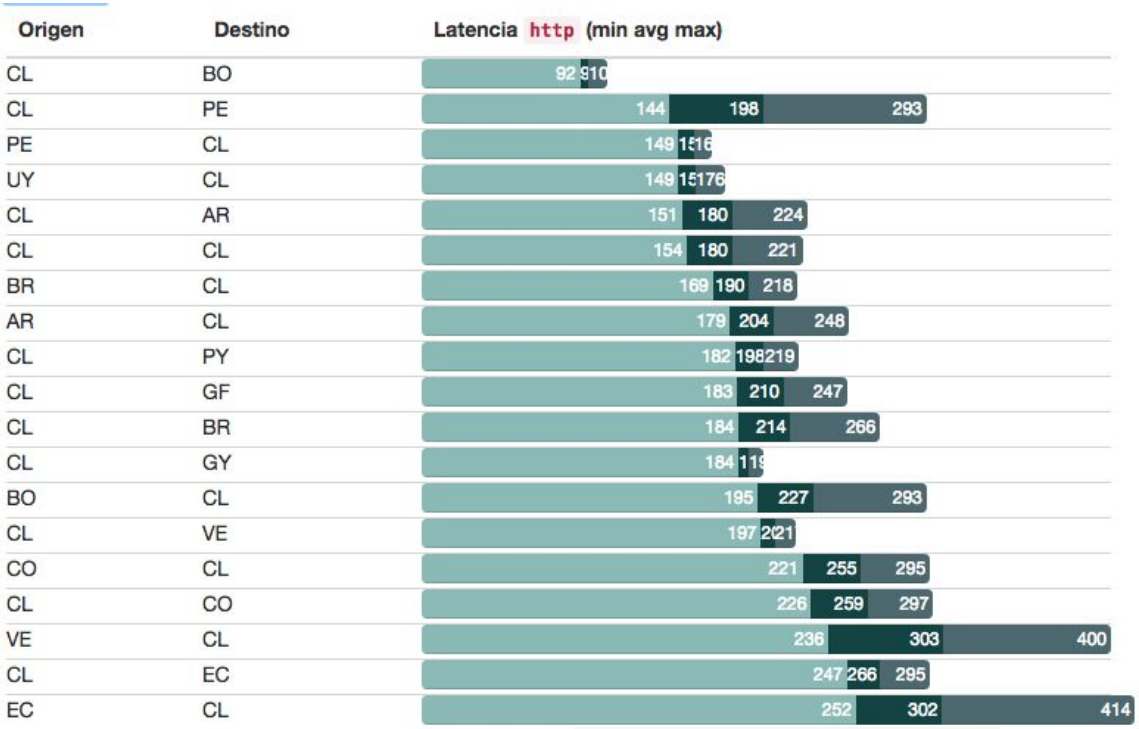
Benjamin Diaz - Cloud Engineer at Whitestack

What are fragmented markets?

- A marketplace where there is no one company that can exert enough influence to move the industry in a particular direction.
- The market consists of several small to medium-sized companies that compete with each other and large enterprises.
- For example, most of Latin American markets.

In Latam, there are still some network fragmentation issues

From Chile, RTT to neighboring countries is in average ~ 180-240 ms



Origen	Destino	Latencia http (min avg max)
CL	BO	92 910
CL	PE	144 198 293
PE	CL	149 151
UY	CL	149 151 176
CL	AR	151 180 224
CL	CL	154 180 221
BR	CL	169 190 218
AR	CL	179 204 248
CL	PY	182 198 219
CL	GF	183 210 247
CL	BR	184 214 266
CL	GY	184 11
BO	CL	195 227 293
CL	VE	197 201
CO	CL	221 255 295
CL	CO	226 259 297
VE	CL	236 303 400
CL	EC	247 266 295
EC	CL	252 302 414

Source:
https://simon.lacnic.net/reports/region/

Proyecto Simón

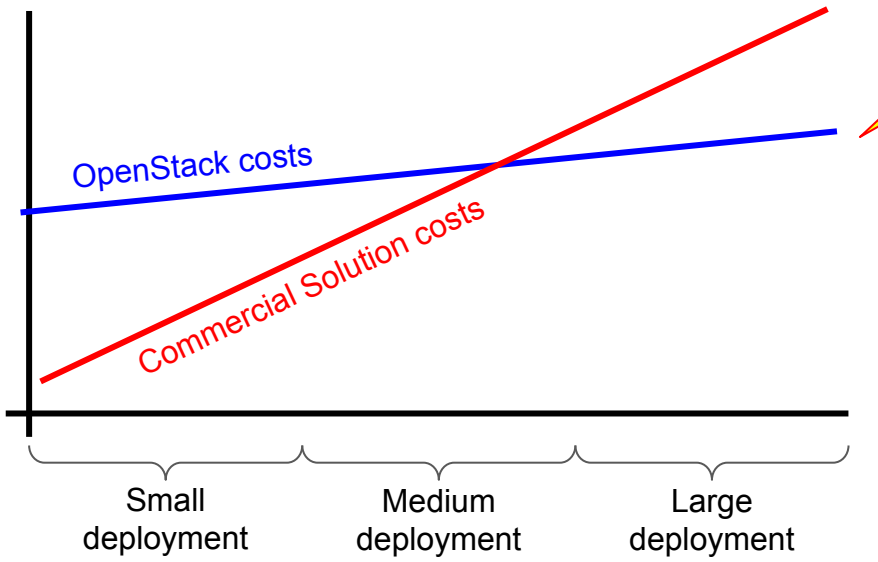
lacniclabs

Having a couple of datacenters in the region, does not work

It makes total sense to deploy in-country Clouds...

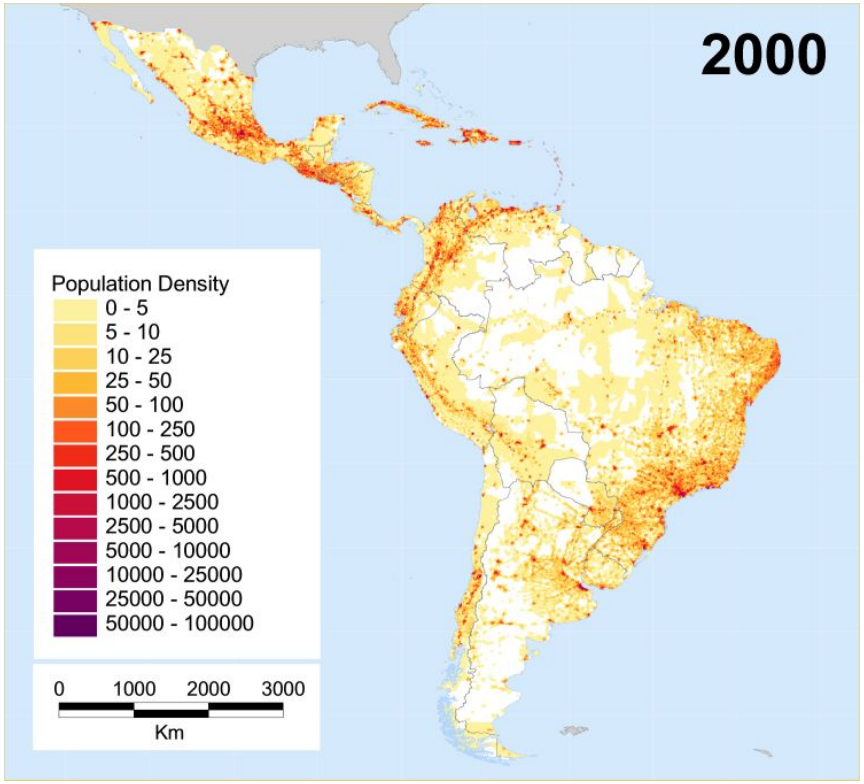
However, these markets are small, making OpenStack deployments difficult..

There is a financial and expertise barrier, to enter the OpenStack business



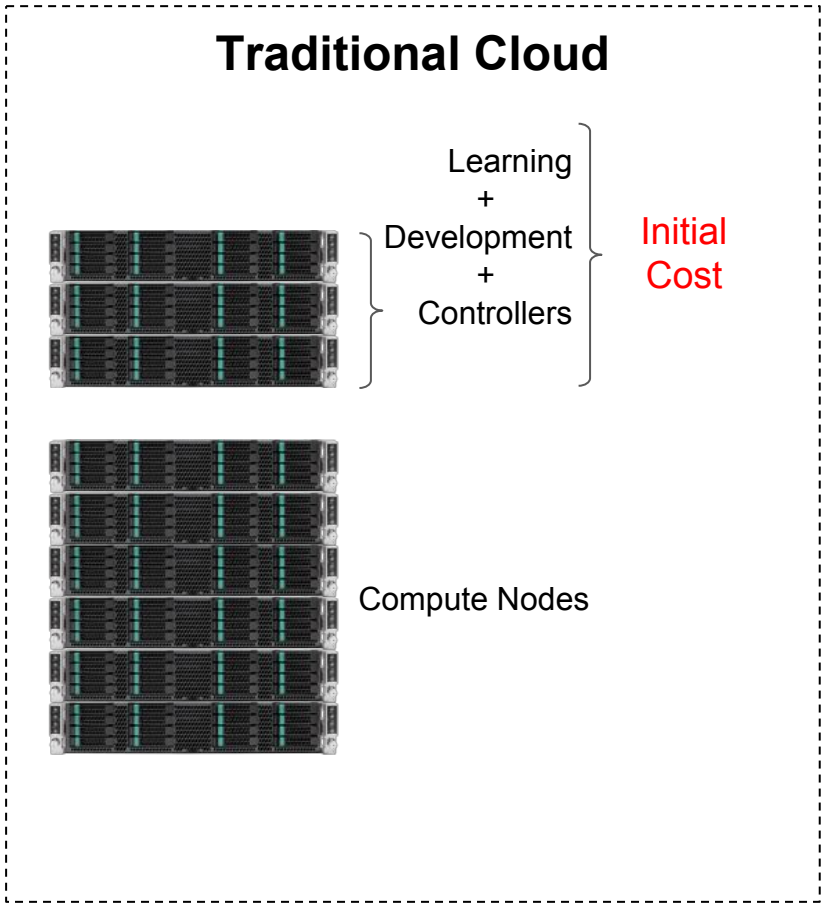
There is a significant entry cost, and then growth costs are moderate

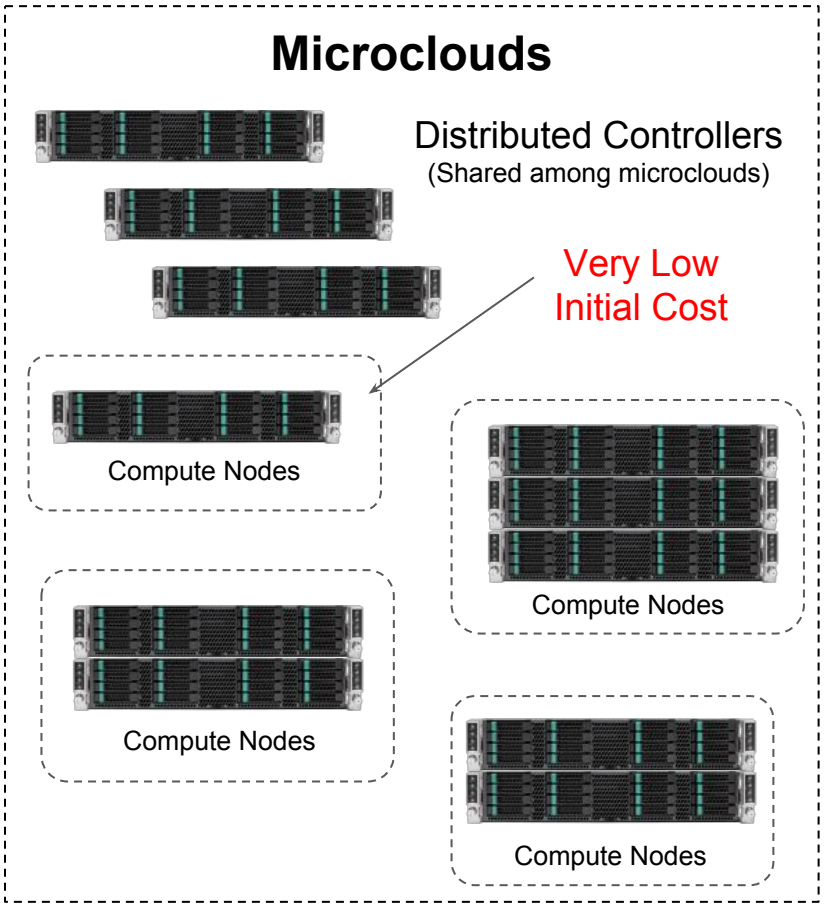
Low Density
80% of Latin American Markets require small deployment



So, let's build Microclouds

Let's reduce the initial investment, by sharing controller services among multiple microclouds



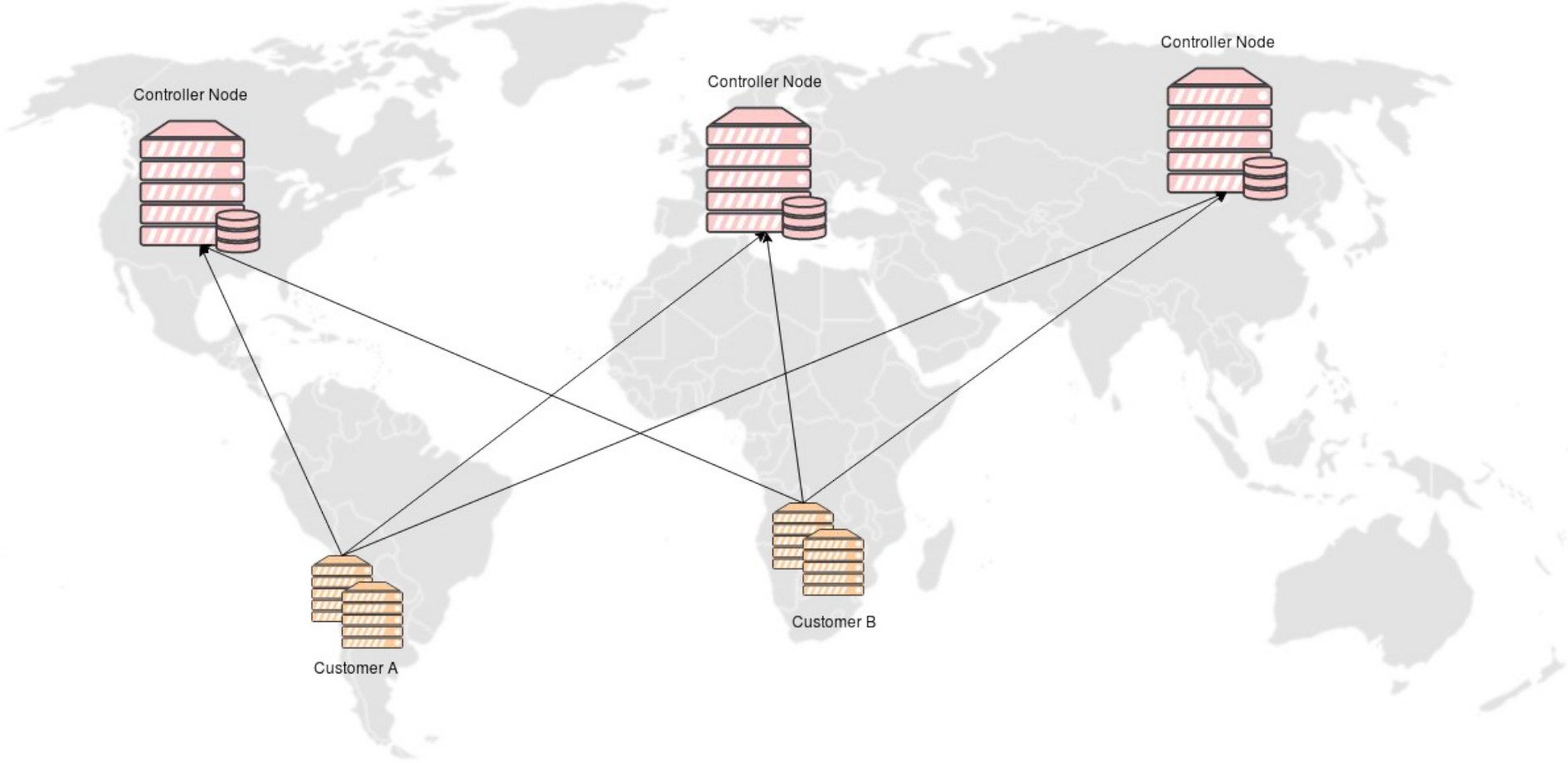


What do we mean by Microclouds?

- Openstack controller components as a service (OCCaaS)[1]
- Multiple geographically distributed nodes containing multiple OpenStack controller services.
- Customers provide only **compute, network** and/or **storage** services.
- All of this achieved with already existing software.

1. No idea how to pronounce that.

Microclouds general view



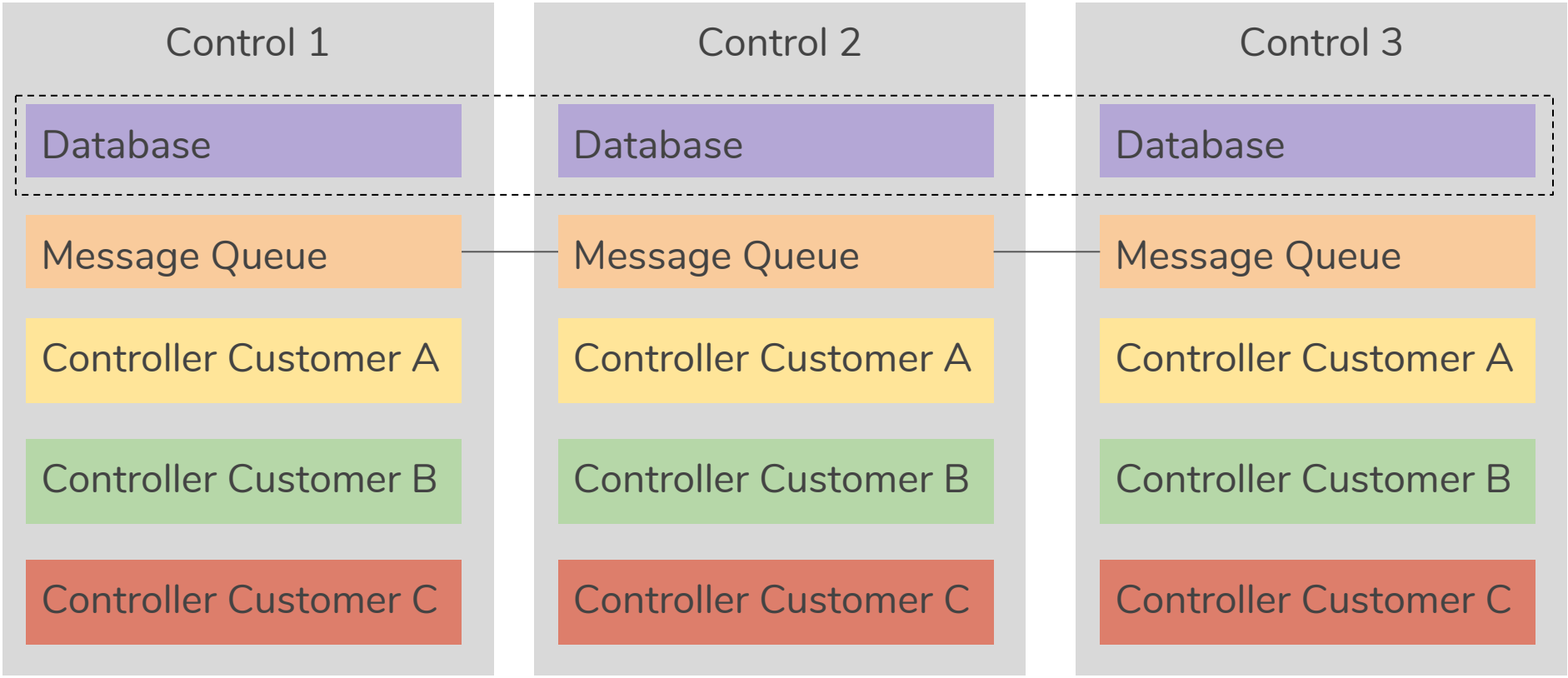
Why?

- Reduce entry barrier to cloud technology.
  - Low infrastructure costs.
  - Low operational costs.
  - Easy deployment.
- Become a gateway for future local OpenStack deployments.
- We all love challenges, don’t we?

What are these challenges?

- Provide High Availability across geographically distributed controller nodes.
- Setup database cluster in a high latency environment.
- Find a way to redirect messages to other nodes in case of service failure.
- Present customers a simple deployment procedure.

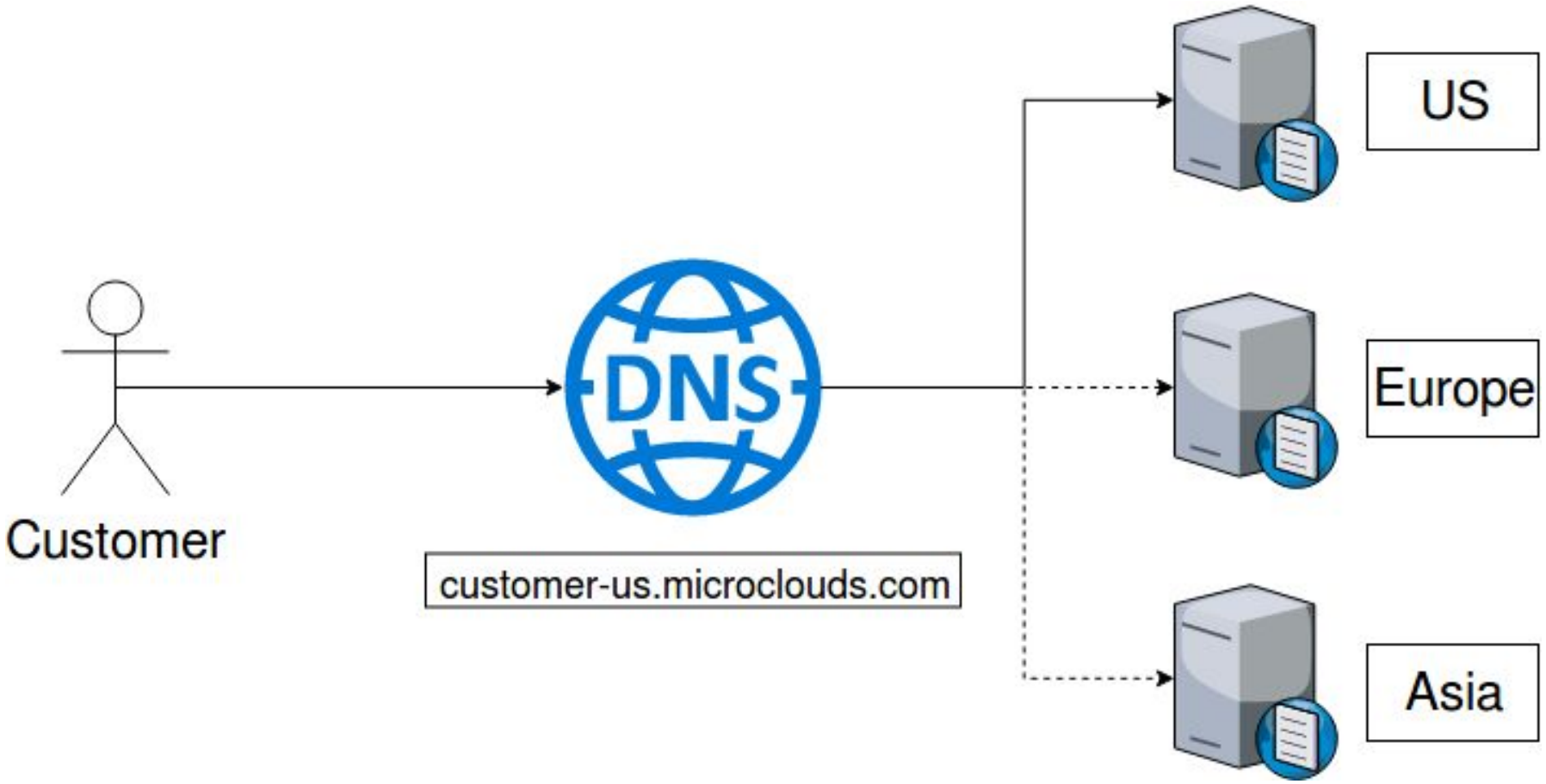
Controller Overview



High Availability

- Every customer will be assigned a primary controller node with lowest latency to compute nodes.
- Other controllers will be assigned as backup.
- How to handle individual services failures?
- How to handle complete node failures?

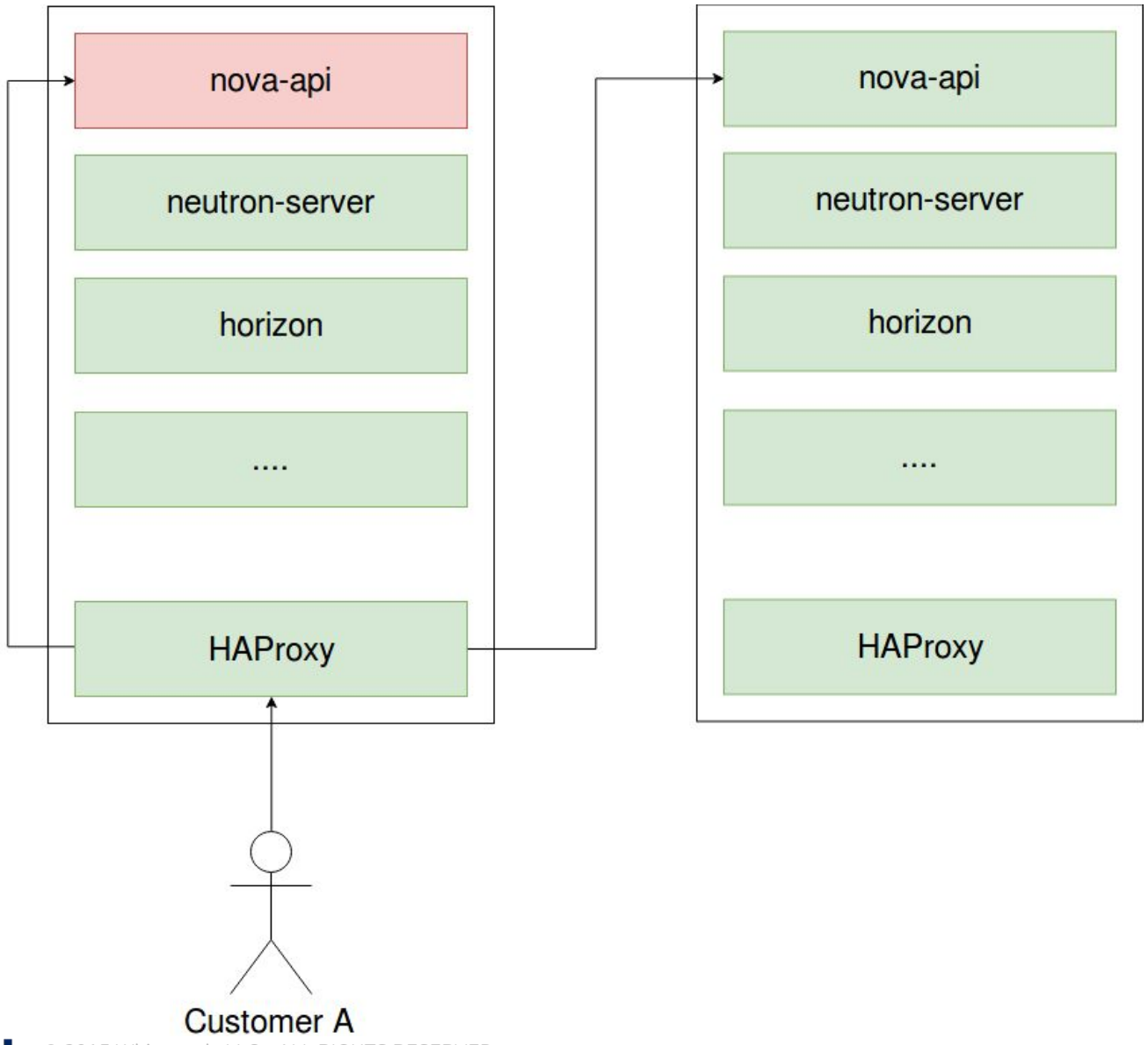
DNS Failover



DNS Failover

- One big caveat: DNS Cache.
- Time To Live (TTL) is not a certain solution.
- ISPs and browsers have to respect the TTL configured.
- Why not a central Load Balancer?
  - Latency
  - Single Point of Failure

Service Failover



Service Failover

- Services can be configured as backup, only getting traffic if the service in the primary node fails.
- If the primary node comes back up, it can be configured so sessions with backup nodes are closed.
- There are cases when you may actually need load balancing features due to node capacity.
- This could be done through custom healthchecks.

Configuration


```
listen nova_api
  bind 10.0.0.2:8774
  http-request del-header X-Forwarded-Proto if { ssl_fc }
  server example 10.0.0.3:8774 check inter 2000 rise 2 fall 5 on-marked-up
shutdown-backup-sessions
  server example 10.0.0.4:8774 check inter 2000 rise 2 fall 5 backup
  server example 10.0.0.5:8774 check inter 2000 rise 2 fall 5 backup

listen mariadb
  mode tcp
  timeout client 3600s
  timeout server 3600s
  option tcplog
  option tcpka
  option mysql-check user haproxy post-41
  bind 10.0.0.2:3306
  server example 10.0.0.3:3306 check inter 2000 rise 2 fall 5
  server example 10.0.0.4:3306 check inter 2000 rise 2 fall 5 backup
  server example 10.0.0.5:3306 check inter 2000 rise 2 fall 5 backup
```

Database

- MariaDB Galera Cluster.
- Native WAN cluster support.
- Penalty for synchronizing the nodes over a high-latency link is only incurred at commit time.
- No use of distributed locking, so each row-level lock does not have to be communicated across datacenters.
- Built-in Encryption through SSL.
- Network timeouts can also be configured to tolerate transient WAN outages.

Configuration

GALERA CLUSTER

- Increase keepalive timeouts to prevent cluster partitioning.
- The following parameters can tolerate 30 second connectivity outages.
  - ```
    wsrep_provider_options = "evs.keepalive_period = PT3S;
    evs.suspect_timeout = PT30S;
    evs.inactive_timeout = PT1M;
    evs.install_timeout = PT1M"
    ```
    ```

Configuration

- Adjust periods and timeouts to be higher than the highest Round Trip Time measurements between nodes in your cluster.
- For example, evs.join_retrans_period:

```
wsrep_provider_options="evs.join_retrans_period=PT0.5S"
```

- To take RTT measurements, you can use ping on each cluster node to ping the others.

Configuration

GALERA CLUSTER

- $ ping -c 3 192.168.1.2

  PING 192.168.1.2 (192.168.1.2) 58(84) bytes of data.

  64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.736 ms

  64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.878 ms

  64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=12.7 ms

  --- 192.168.1.2 ---

  3 packets transmitted, 3 received, 0% packet loss, time 2002ms

  rtt min/avg/max/mdev = 0.736/4.788/12.752/5.631 ms

Configuration

- If each controller node is a LAN cluster, it is possible to tell Galera Cluster how **nodes are grouped by physical proximity** using the gmcast.segment setting.
- Messages are sent between two datacenters only once, avoiding duplication in inter-data center traffic. The message will then be relayed internally within the datacenter.
- State Snapshot Transfers (SST) and Incremental Snapshot Transfers (IST) will favor using a donor node inside the same datacenter.

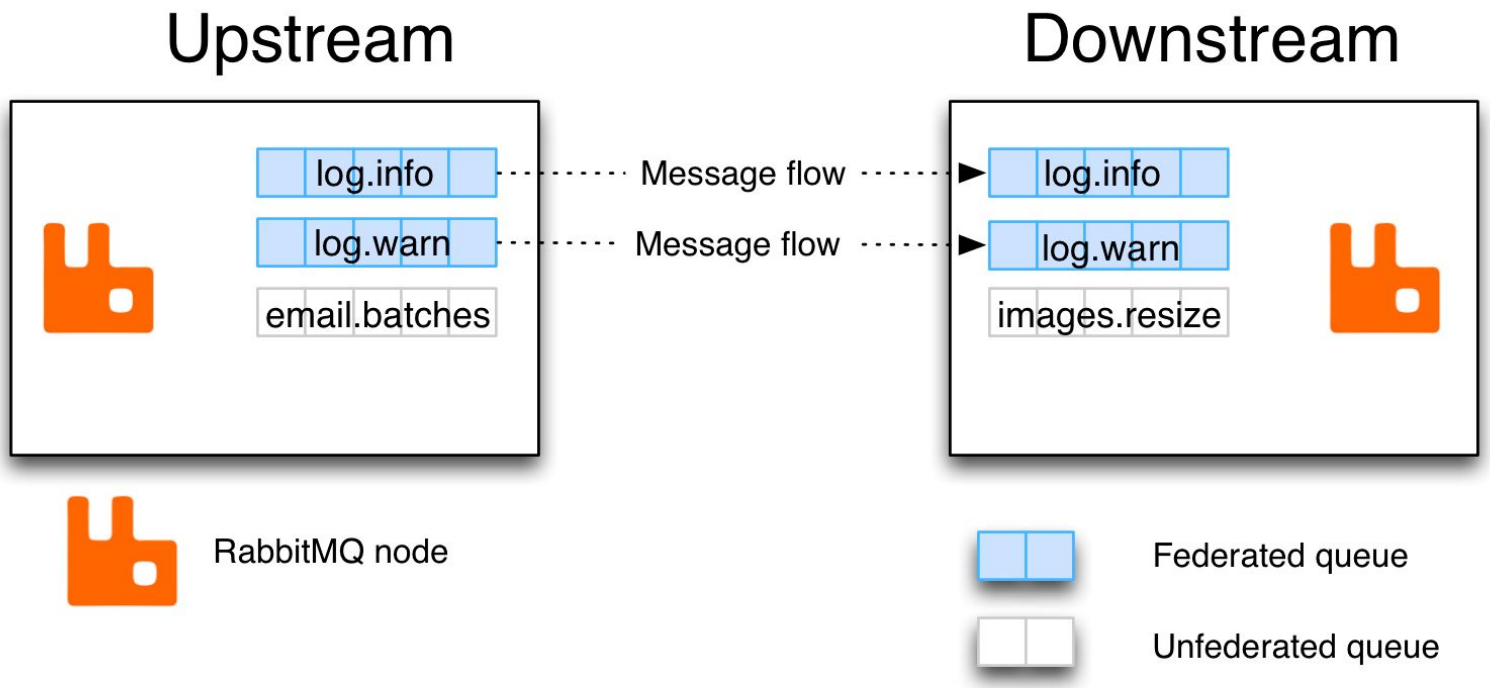
Message Queue

- Shared RabbitMQ across deployments.
- Virtual Hosts provide multitenancy.
- Support in oslo.messaging.
- `transport_url = driver://[user:pass@]host:port[,[userN:passN@]hostN:portN]/`**`virtual_host`**`?query`

Message Queue

RabbitMQ

- How to handle services failures from the message bus perspective?
- RabbitMQ Federated Queues

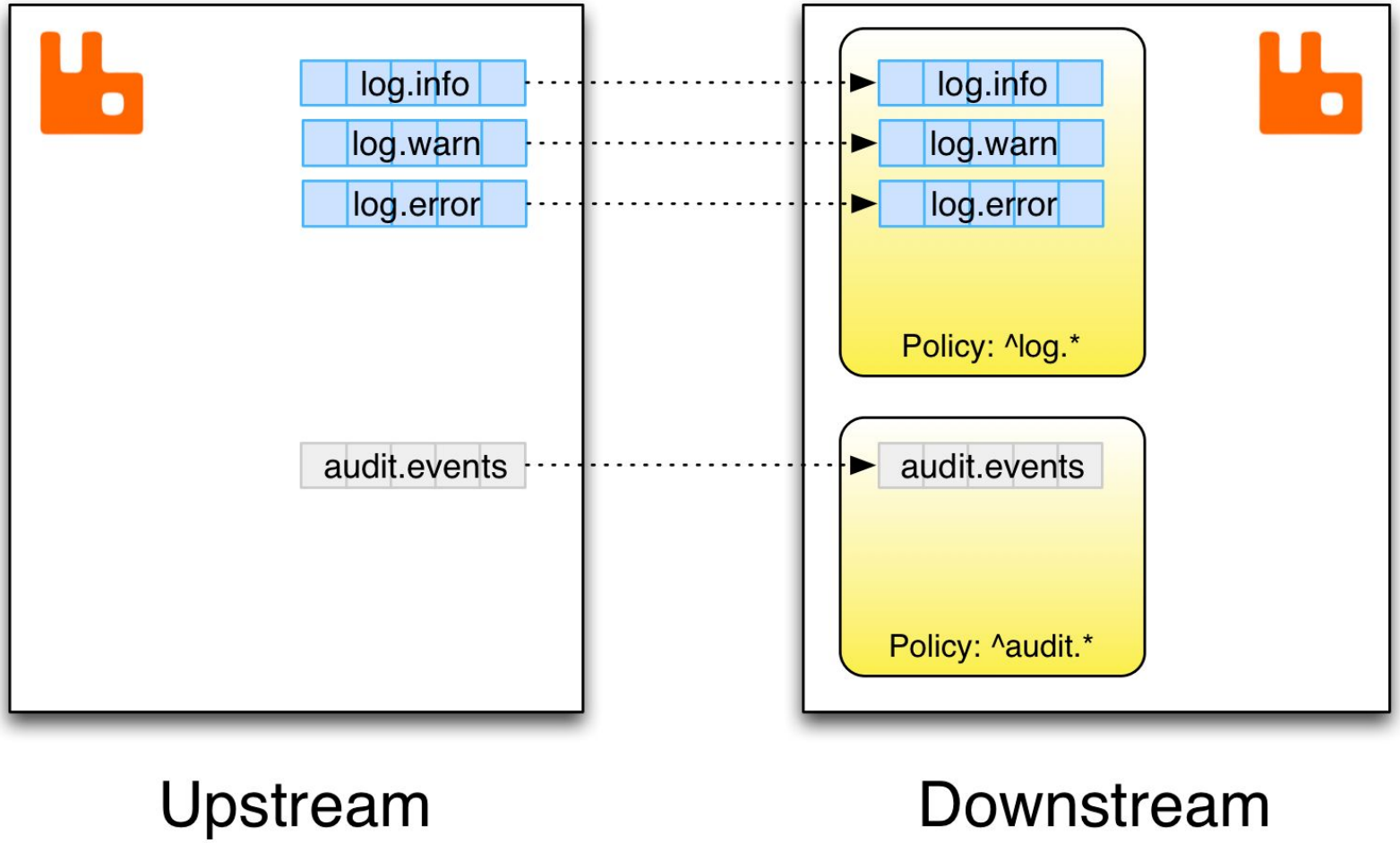


Message Queue

- The federated queue will only retrieve messages when it has run out of messages locally, it has consumers that need messages, and the upstream queue has "spare" messages that are not being consumed.
- Configuration is done by declaring policies. A policy is a pattern that queue names are matched against. Matching queues will be federated.
- In our scenario we will federate all queues by using a wildcard policy.

Message Queue

RabbitMQ



Configuration

RabbitMQ

- Enable Federation Plugin:

```
rabbitmq-plugins enable rabbitmq_federation
```

- Enable Federation Management Plugin:

```
rabbitmq-plugins enable rabbitmq_federation_management
```

- In each node, define other nodes as upstream:

```
rabbitmqctl set_parameter federation-upstream my-upstream
'{"uri":"amqp://server-name","expires":3600000}'
```

- Define policy:

```
rabbitmqctl set_policy --apply-to exchanges federate-me
"^amq\." '{"federation-upstream-set":"all"}'
```

Deployment

KOLLA
an OpenStack Community Project

- Ansible is the solution to all problems in this world.
- Kolla-Ansible is the best way of deploying OpenStack.
- Easy to build on top of it to adequate it to our use case.
- It can handle today multiple simultaneous OpenStack deployments.
- https://github.com/openstack/kolla-ansible/

Future Ideas

- Consider replacing RabbitMQ with Apache QPID and setting up a router in the microcloud nodes.
- Setting another load balancer in the microcloud.
- Share more services between microclouds than just the database and message bus.
- Have any idea? Let us know!

Gracias!

bdiaz@whitestack.com

USA
Whitestack, LLC.
Brickell Bayview Center
80 SW 8th Street, Suite 2000,
Miami, FL 33130
EEUU

LATAM
Whitestack Latam
Apoquindo 4700 piso 11
Las Condes, 7560969
Chile