OpenStack: Arquitetura e Operação

Aula Teórica nº4

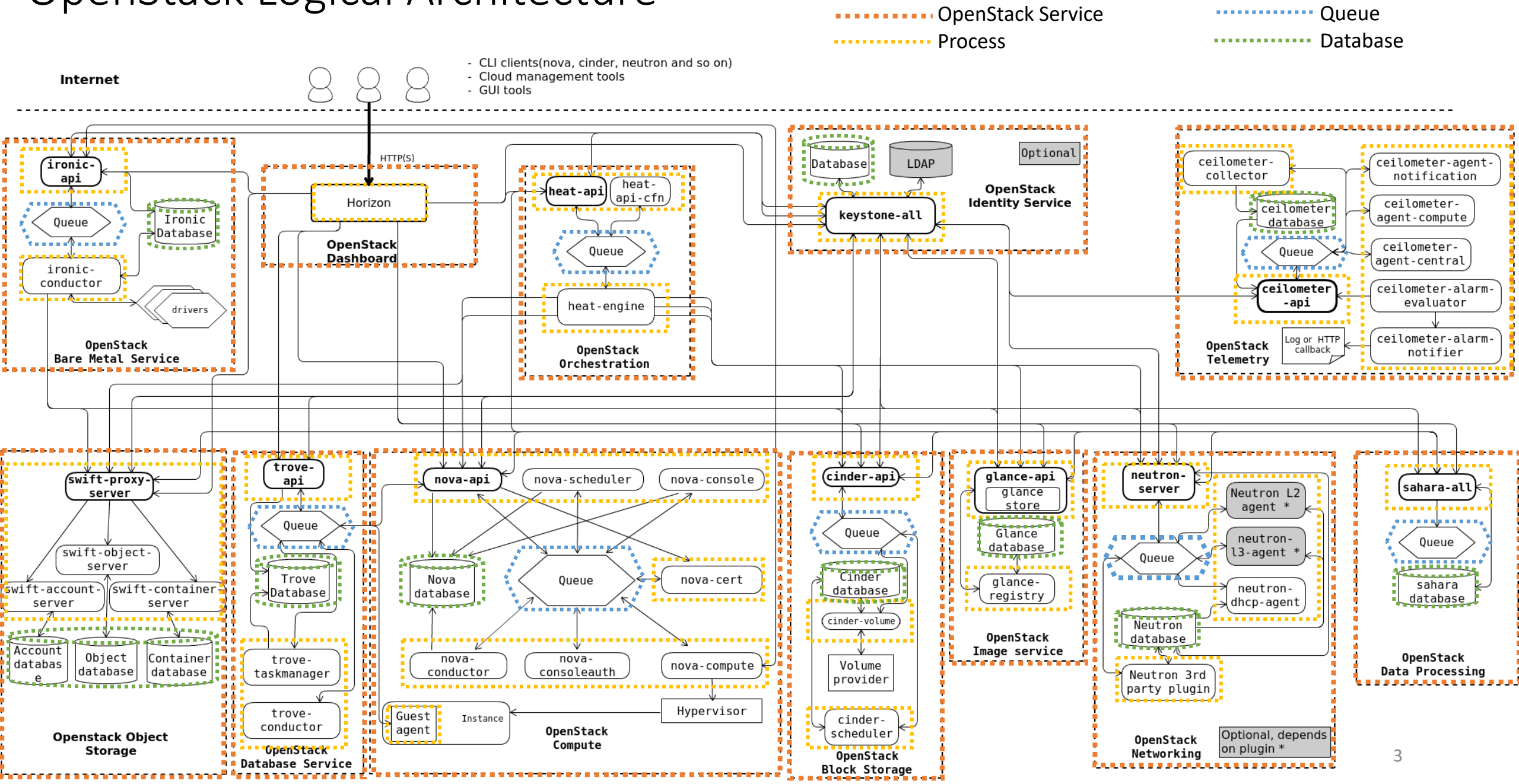
2020/2021

OpenStack Architecture – SW Components*

- **OpenStack Services** (Keystone, Nova, Glance, Cinder, Neutron, Horizon, Swift, Ironic, Trove, Heat, Sahara, Ceilometer, …)
- **Processes** (services within the OpenStack services)
- **Support services**
  - AMQP message broker – communication between the components of a service
  - SQL Database – stores built-in and runtime information
  - Memcached – token caching
  - NTP – Network Timing Protocol
  - Etcd – a distributed reliable key-value store
  - iSCSI - SCSI disk protocol tunneled within Ethernet

* Most relevant components

OpenStack Logical Architecture



Services & Processes

- OpenStack consists of several independent parts: services
  - Nova, Glance, Neutron, Keystone, Cinder, Swift, Horizon, Ironic, Heat, …
- Services are composed of several processes
- All services have at least one API process
  - which listens for API requests, preprocesses them and passes them on to other parts of the service
  - Except for the Identity service, the actual work is done by distinct processes

The queue (AMQP message broker)

- Used for all communication between the processes and daemons of one service (the communication between the two different services in OpenStack uses service endpoints)
- Usually implemented with RabbitMQ or ZeroMQ

Database

- Most of the OpenStack services use an SQL database to store the build-time, and run-time states for a cloud infrastructure, such as instance status, networks, projects
- The most tested and preferable databases to use in OpenStack are MySQL, MariaDB, and PostgreSQL

memcached

- Keystone uses memcached to cache tokens
- The memcached service typically runs on the controller node
- For production deployments, is recommend to enable a combination of firewalling, authentication, and encryption to secure it

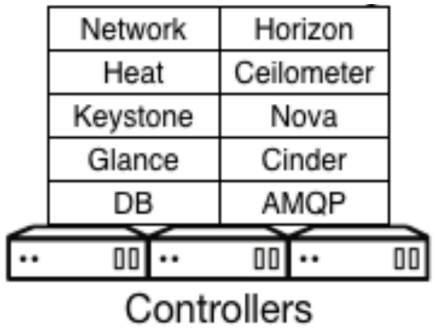
NTP

- A time synchronization package, such as NTP, is a prerequisite, as OpenStack services depend on consistent and synchronized time between the controller, network and compute nodes
- For example, the Nova service should synchronize the time across the hosts to avoid time conflicts when scheduling VM provisions on the compute nodes
- Also, other services will experience similar issues when the time is not synchronized

General Compute Cloud – Design Model

A firewall, switches and load balancers on the public facing network connections

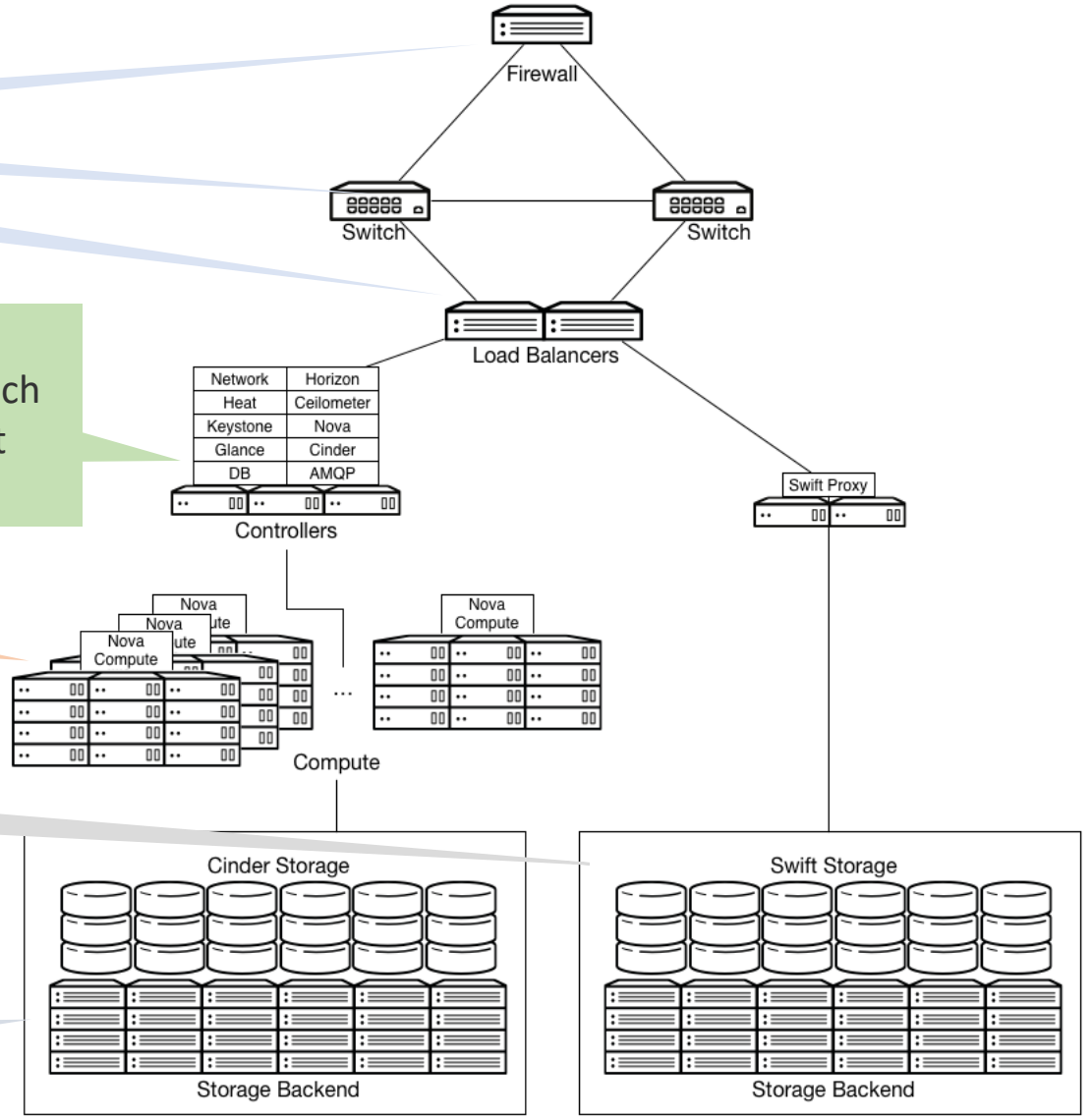
OpenStack Controller service running Image service, Identity service, Networking service, combined with support services such as MariaDB and RabbitMQ, configured for high availability on at least three controller nodes



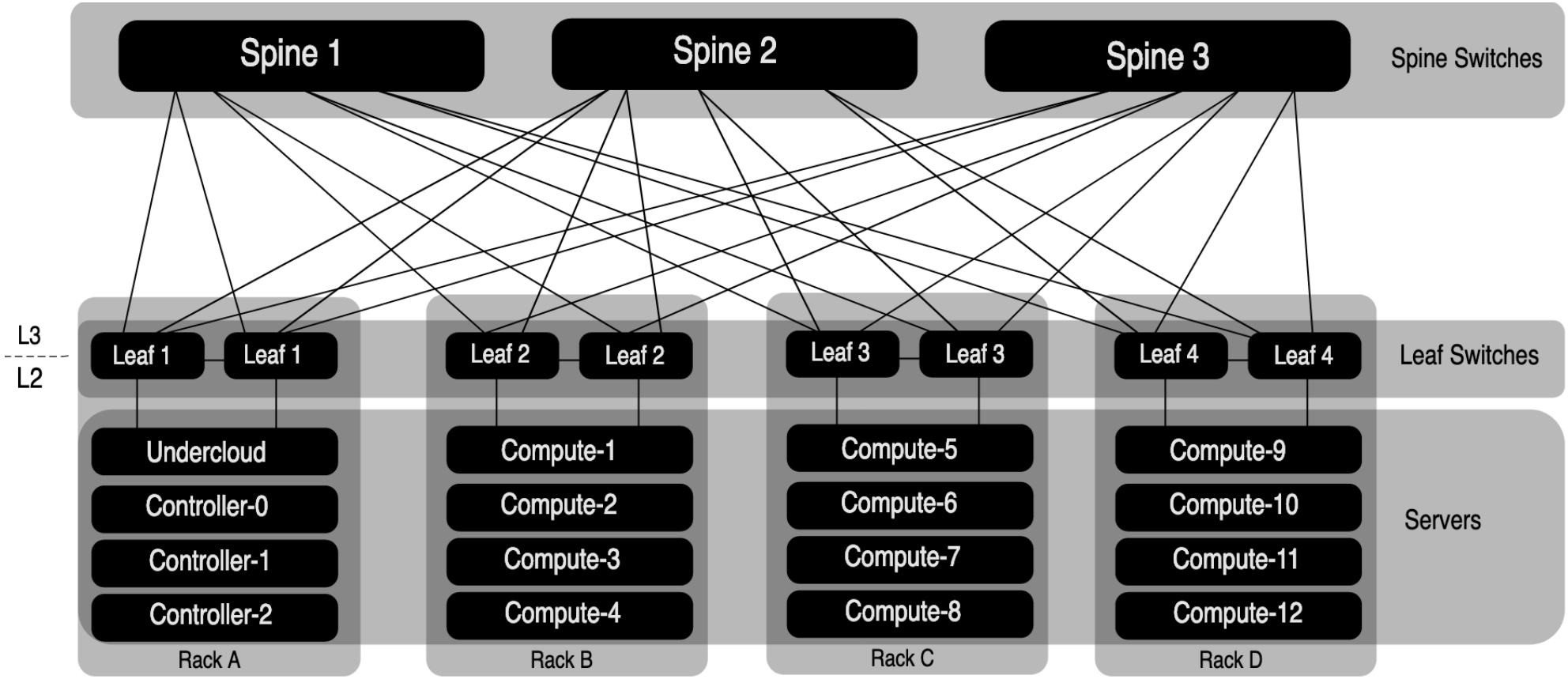
OpenStack compute nodes running the KVM hypervisor

OpenStack Object Storage for serving static objects (such as images)

OpenStack Block Storage for use by compute instances, requiring persistent storage (such as databases for dynamic sites)



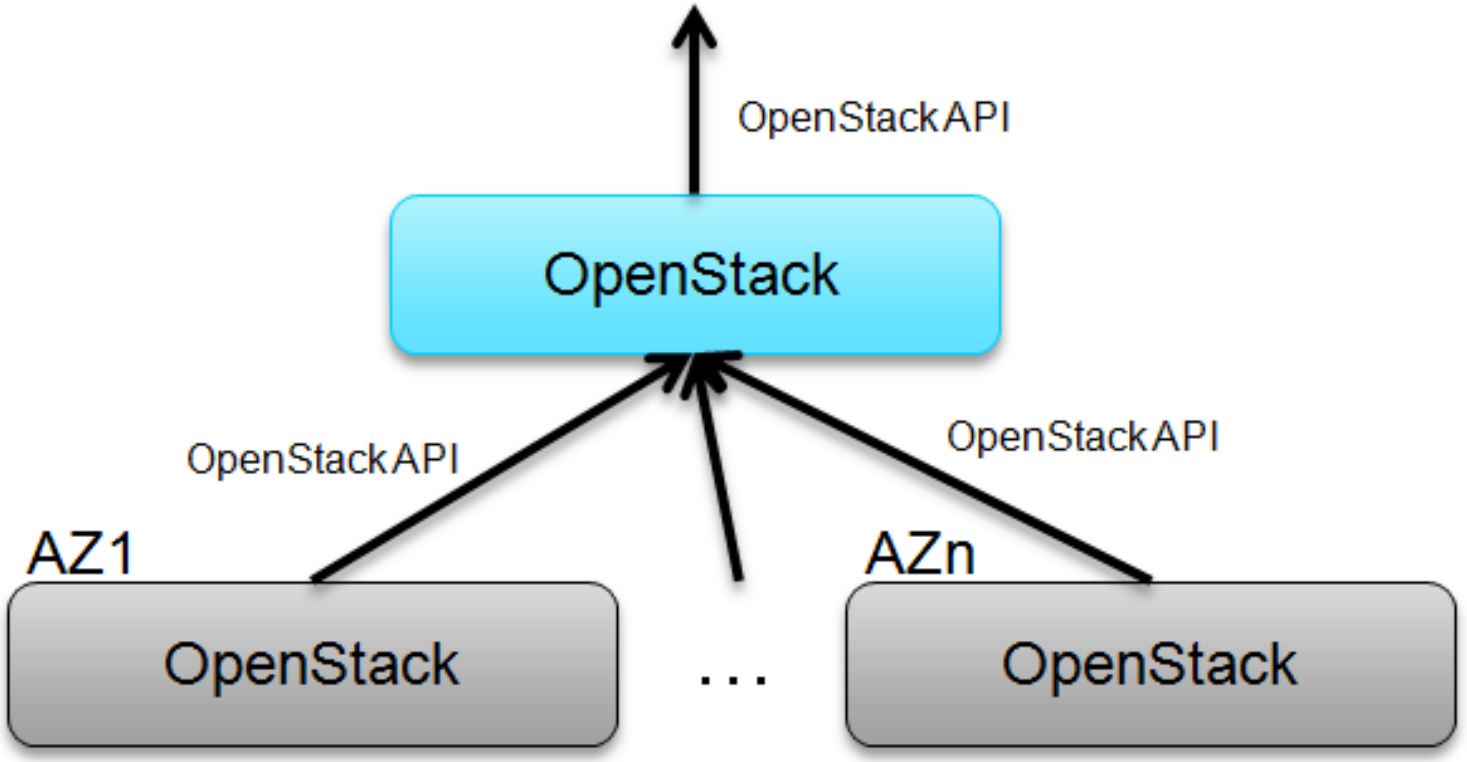
Network HW Requirements



- The leaf-spine design model permits to add additional bandwidth as well as scale out to additional racks
- Select network HW that provides port density and speed for future growth as workload demands increase
- Important to evaluate the redundancy of the solution to provide high availability (HA)

OpenStack Cascading Solution

- Designed for multi-site OpenStack clouds integration
- Solves possible scalability limitations of OpenStack being able to provide a cloud service with millions of instances geographically distributed in many data centers



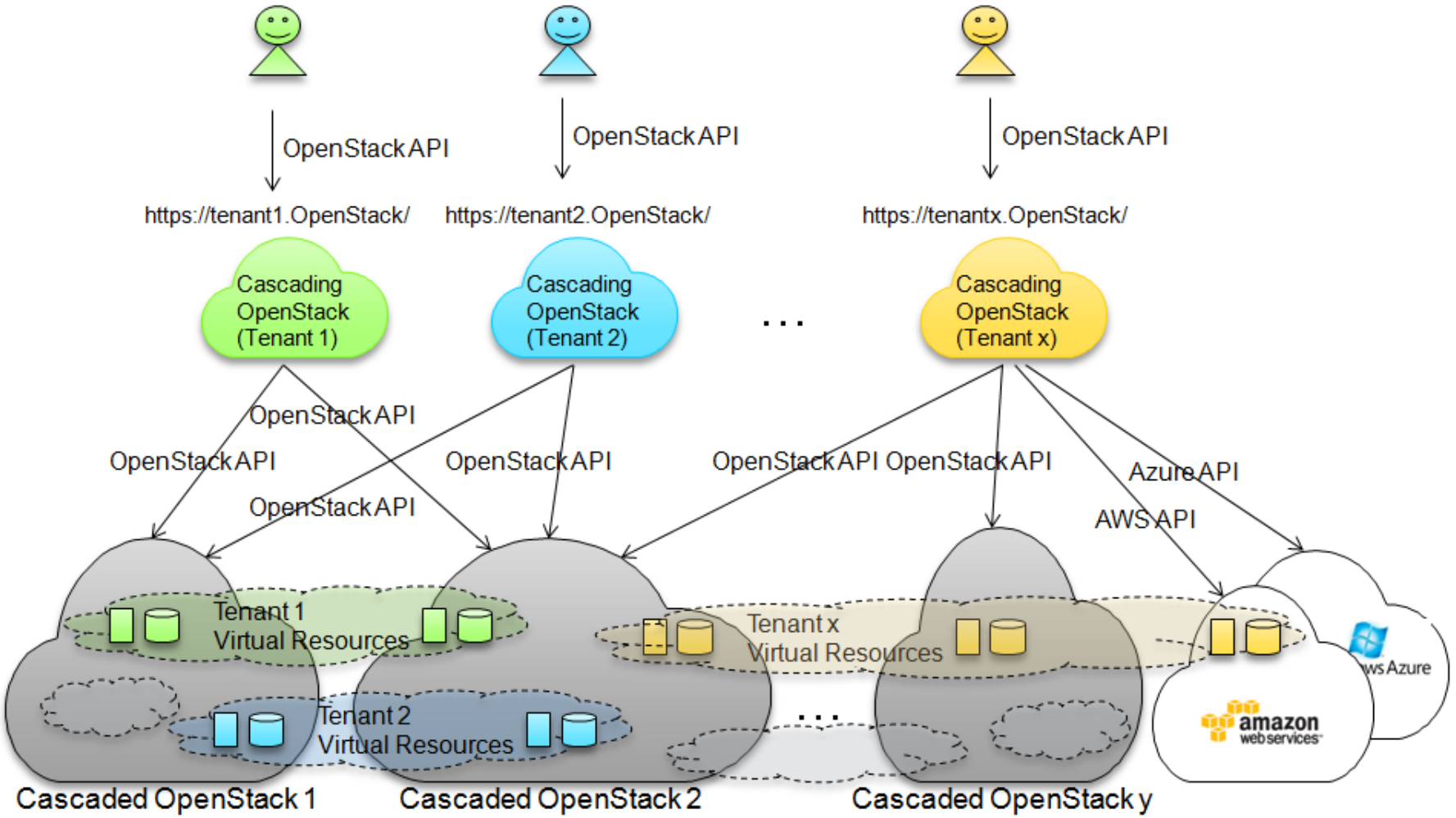
Cascading OpenStack

Cascaded OpenStack

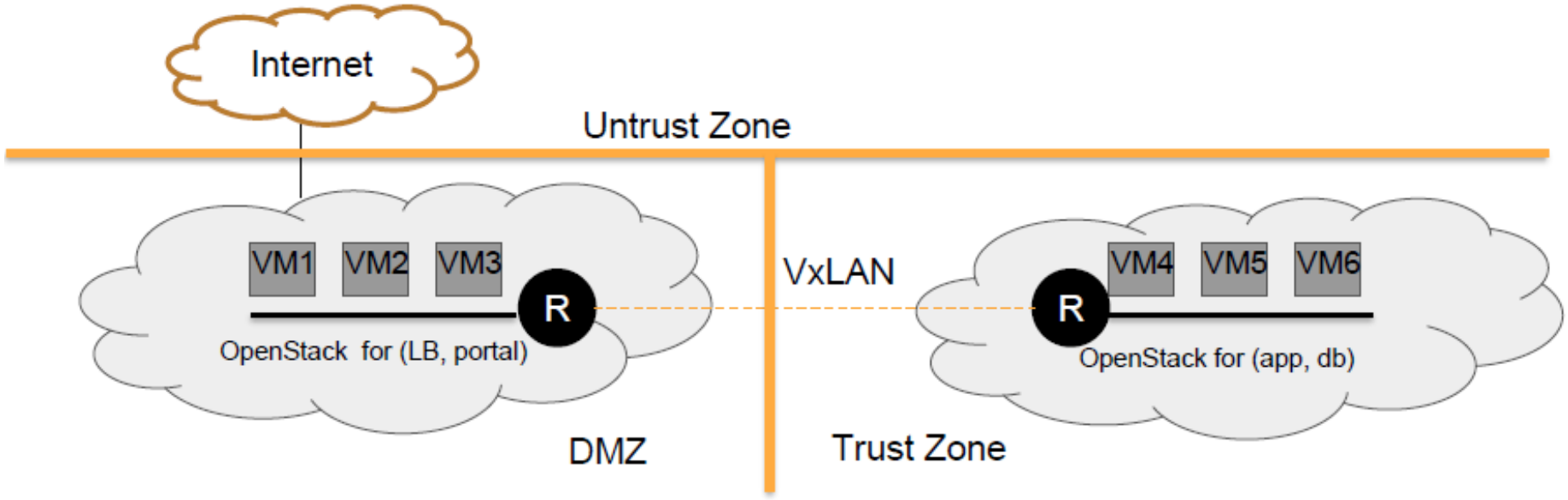
Cascading OpenStack: providing API and scheduling orchestration and networking of Cascaded OpenStacks

Cascaded OpenStack: Provisioning the VM, Volume and virtual Networking resources

Tenant level virtual OpenStack service over hybrid or federated or multiple OpenStack based clouds



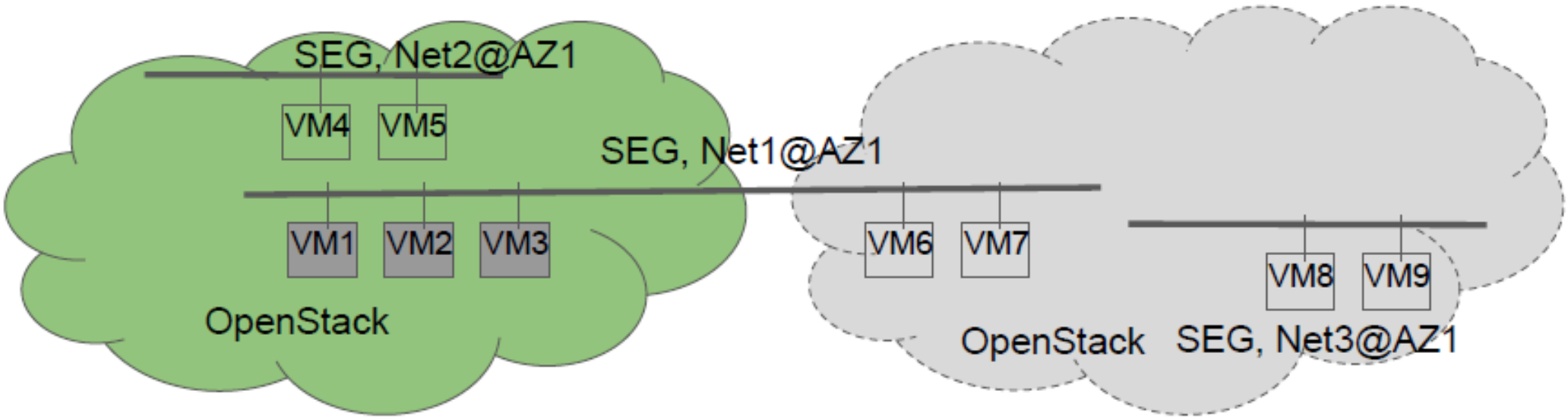
Use Case: Isolation of East-West Traffic



Financial application has different requirements from security aspect, separate OpenStack instance in different zone, tenant level networking automation requirements:

1. East-west L3 traffic for the tenant should be isolated, use VxLAN to connect the routers in different OpenStack instance.
2. Security group should work for VMs in different OpenStack instances.
3. IP/Mac address management to avoid conflict.

Use Case: Cloud Capacity Expansion

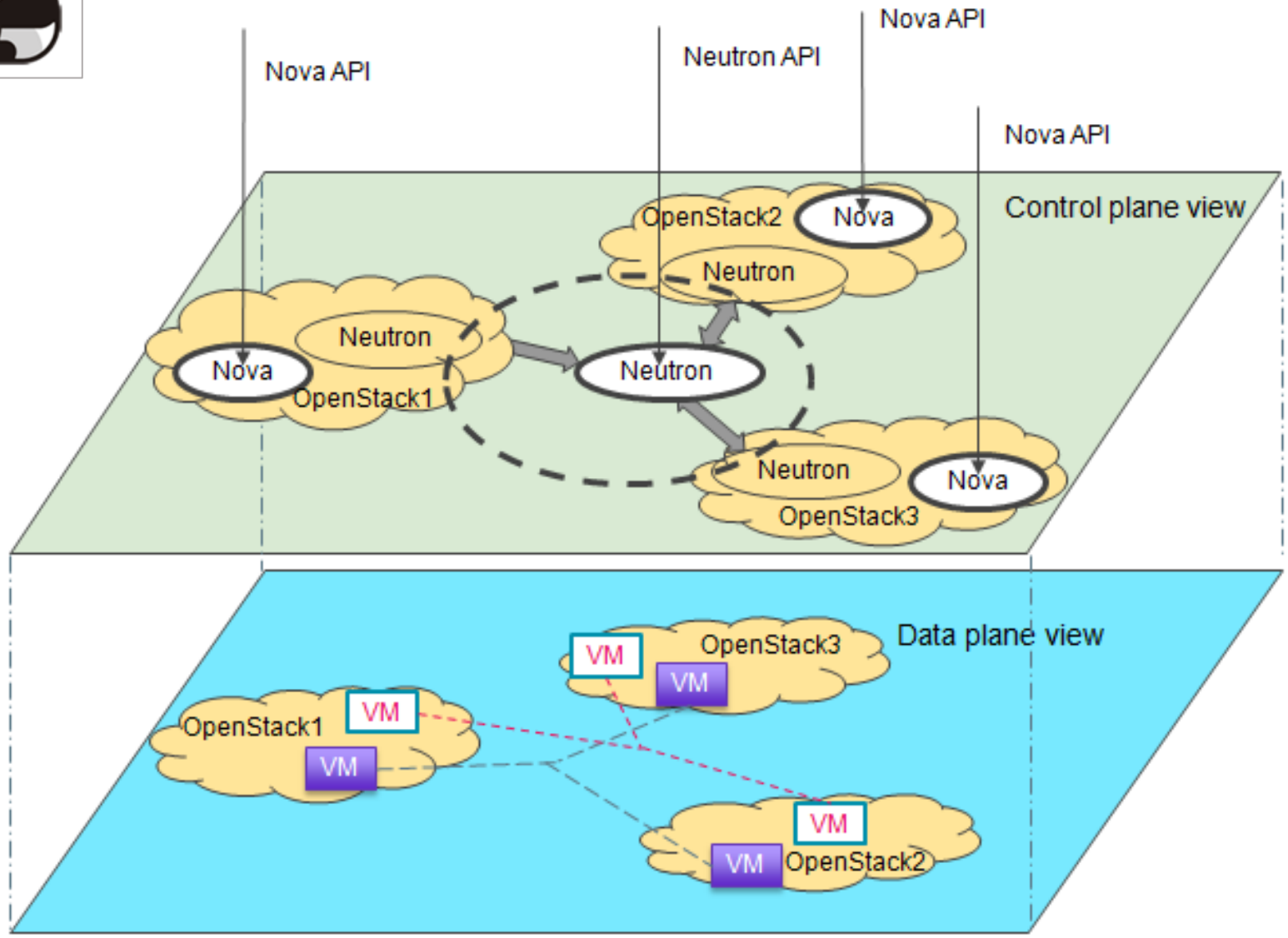


Tenant level networking automation requirements:

1. Add new VMs into same L2 network in the new OpenStack instance.
2. Security group should work for VMs in different OpenStack instances.
3. IP/Mac address management to avoid conflict.
4. East-west L3 traffic for the tenant should be isolated.

Tricircle Service

- First appeared in OpenStack **Ocata** release (feb 2017)
- Dedicated for networking automation across Neutron in multi-region OpenStack deployments
- Enable the creation of global network abstracting networking resources across multiple OpenStack clouds
- From the data plane view, all instances are provisioned in different clouds but can be inter-connected via the global abstract networking resources (with tenant level isolation)



OpenStack Operations & Troubleshooting

Validating service status in **devstack**

```
devstack@openstack:~$ service --status-all | egrep 'rabbit|mysql|mem|ntp'
 [ + ]  memcached
 [ + ]  mysql
 [ + ]  ntp
 [ + ]  rabbitmq-server
```

+ Active
- Inactive

```
devstack@openstack:~$ service devstack@etcd status
● devstack@etcd.service - Devstack devstack@etcd.service
   Loaded: loaded (/etc/systemd/system/devstack@etcd.service; enabled; vendor preset:
enabled)
   Active: active (running) since Fri 2019-05-03 19:59:48 UTC; 4 days ago
 Main PID: 27165 (etcd)
    Tasks: 8 (limit: 4664)
   CGroup: /system.slice/system-devstack.slice/devstack@etcd.service
           └─27165 /opt/stack/bin/etcd --name openstack --data-dir /opt/stack/data/etcd -
-initial-cluster-state new --in

May 03 19:59:48 openstack etcd[27165]: e9ecc4b07ea38320 received MsgVoteResp from
e9ecc4b07ea38320 at term 2
May 03 19:59:48 openstack etcd[27165]: e9ecc4b07ea38320 became leader at term 2
...
```

```
devstack@openstack:~$ service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-05-06 08:42:17 UTC; 1 day 12h ago
 Main PID: 26317 (mysqld)
    Tasks: 55 (limit: 4664)
   CGroup: /system.slice/mysql.service
           └─26317 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

May 06 08:42:16 openstack systemd[1]: Starting MySQL Community Server...
May 06 08:42:17 openstack systemd[1]: Started MySQL Community Server.
```

OpenStack Services in **devstack**

- **systemd** is a system and service manager that brings up and maintains userspace services
- **CGroup:** /system.slice/system-devstack.slice

Linux Control Group (cgroup) tree

Slice unit configuration (a slice unit is a concept for hierarchically managing resources of a group of processes)

```
devstack@openstack:~$ more /etc/systemd/system/devstack@
```

cinder
- devstack@c-api.service
- devstack@c-sch.service
- devstack@c-vol.service

devstack@dstat.service
devstack@etcd.service

glance
- devstack@g-api.service
- devstack@g-reg.service

devstack@keystone.service

keystone

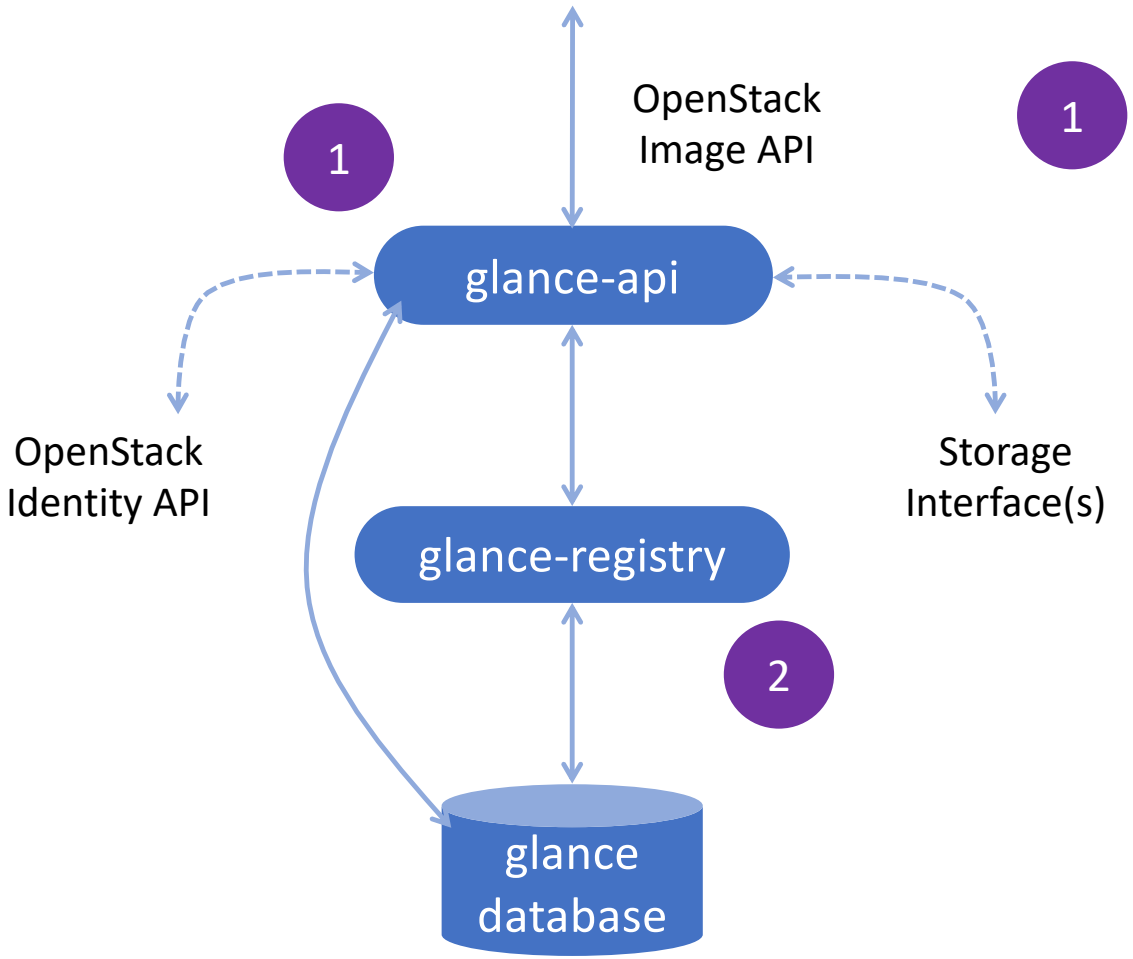
Nova
- devstack@n-api-meta.service
- devstack@n-api.service
- devstack@n-cauth.service
- devstack@n-cond-cell1.service
- devstack@n-cpu.service
- devstack@n-novnc.service
- devstack@n-sch.service
- devstack@n-super-cond.service

devstack@placement-api.service

Neutron
- devstack@q-agt.service
- devstack@q-dhcp.service
- devstack@q-l3.service
- devstack@q-meta.service
- devstack@q-svc.service

/lib/systemd/system/apache2.service Horizon

Glance Service in **devstack**




```
devstack@openstack:~$ more
/etc/systemd/system/devstack@g-
devstack@g-api.service 1
devstack@g-reg.service 2
```

```
devstack@openstack:~$ more
/etc/systemd/system/devstack@g-api.service

[Unit]
Description = Devstack devstack@g-api.service

[Service]
RestartForceExitStatus = 100
NotifyAccess = all
Restart = always
KillMode = process
Type = notify
ExecReload = /bin/kill -HUP $MAINPID
ExecStart = /usr/local/bin/uwsgi --procname-prefix
glance-api --ini /etc/glance/glance-uwsgi.ini
User = devstack
SyslogIdentifier = devstack@g-api.service


[Install]
WantedBy = multi-user.target
```

ExecStart devstack@g-api.service

1

ExecStart = /usr/local/bin/uwsgi --procname-prefix glance-api --ini /etc/glance/glance-uwsgi.ini

```
devstack@openstack:~$ more /etc/glance/glance-uwsgi.ini

[uwsgi]
socket-timeout = 30
http-keepalive = false
http-auto-chunked = true
http-chunked-input = true
http-raw-body = true
chmod-socket = 666
lazy-apps = true
add-header = Connection: close
buffer-size = 65535
worker-reload-mercy = 90
hook-master-start = unix_signal:15 gracefully_kill_them_all
thunder-lock = true
plugins = python
enable-threads = true
exit-on-reload = false
die-on-term = true
master = true
processes = 2
http-socket = 127.0.0.1:60999
wsgi-file = /usr/local/bin/glance-wsgi-api
```

uWSGI is a software application that "aims at developing a full stack for building hosting services".[2] It is named after the Web Server Gateway Interface (WSGI), which was the first plugin supported by the project.

```
devstack@openstack:~$ sudo lsof -i -P -n | grep 60999
uwsgi     21503        devstack    4u  IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
uwsgi     21507        devstack    4u  IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
uwsgi     21508        devstack    4u  IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
```

master

worker 1

worker 2

python script

ExecStart devstack@g-reg.service 2

ExecStart = /usr/local/bin/glance-registry --config-file=/etc/glance/glance-registry.conf

```
devstack@openstack:~$ more /etc/glance/glance-registry.conf

[DEFAULT]
...
transport_url = rabbit://stackrabbit:devstack@10.0.2.15:5672/
use_syslog = False
bind_host = 0.0.0.0
debug = True

[database]
connection =
mysql+pymysql://root:devstack@127.0.0.1/glance?charset=utf8

[paste_deploy]
flavor = keystone

[keystone_authtoken]
memcached_servers = localhost:11211
signing_dir = /var/cache/glance/registry
cafile = /opt/stack/data/ca-bundle.pem
project_domain_name = Default
project_name = service
user_domain_name = Default
password = devstack
username = glance
auth_url = http://10.0.2.15/identity
auth_type = password

[oslo_messaging_notifications]
driver = messagingv2
```

/var/lib : Variable state information

```
devstack@openstack:~$ sudo su
root@openstack:/home/devstack# cd /var/lib/lxcfs/cgroup/name=systemd/system.slice/system-devstack.slice
root@openstack:/var/lib/lxcfs/cgroup/name=systemd/system.slice/system-devstack.slice# ls
cgroup.clone_children   devstack@g-api.service         devstack@n-cpu.service          devstack@q-l3.service
cgroup.procs            devstack@g-reg.service         devstack@n-novnc.service        devstack@q-meta.service
devstack@c-api.service  devstack@keystone.service      devstack@n-sch.service          devstack@q-svc.service
devstack@c-sch.service  devstack@n-api-meta.service    devstack@n-super-cond.service   notify_on_release
devstack@c-vol.service  devstack@n-api.service         devstack@placement-api.service  tasks
devstack@dstat.service  devstack@n-cauth.service       devstack@q-agt.service
devstack@etcd.service   devstack@n-cond-cell1.service  devstack@q-dhcp.service
```

1 devstack@g-api.service

```
root@openstack:/var/lib/lxcfs/cgroup/name=systemd/system.slice/system-devstack.slice# more devstack@g-api.service/
cgroup.clone_children  cgroup.procs          notify_on_release      tasks
root@openstack:/var/lib/lxcfs/cgroup/name=systemd/system.slice/system-devstack.slice# more devstack@g-
api.service/cgroup.procs
21503
21507
21508
```

```
devstack@openstack:~$ sudo lsof -i -P -n | egrep '21503|21507|21508'
uwsgi     21503        devstack    4u  IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
uwsgi     21507        devstack    4u  IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
uwsgi     21507        devstack    9u  IPv4  390316      0t0  TCP 127.0.0.1:60038->127.0.0.1:11211 (ESTABLISHED)
uwsgi     21507        devstack   12u  IPv4  390362      0t0  TCP 10.0.2.15:52366->10.0.2.15:5672 (ESTABLISHED)
uwsgi     21508        devstack    4u  IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
uwsgi     21508        devstack    9u  IPv4  390339      0t0  TCP 127.0.0.1:60050->127.0.0.1:11211 (ESTABLISHED)
uwsgi     21508        devstack   12u  IPv4  390347      0t0  TCP 10.0.2.15:52358->10.0.2.15:5672 (ESTABLISHED)
uwsgi     21508        devstack   13u  IPv4  500501      0t0  TCP 10.0.2.15:54470->10.0.2.15:5672 (ESTABLISHED)
```

Listening@

Nova:

- **API** 8775, 35949
- **VNC proxy** 6080

Glance:

- **API server** 60999
- **Registry server** 9191

Neutron: 6969, 6633 (OF)

OVSDB: 6640

iSCSI (Cinder, Nova, Glance): 3260

Apache: 80

RabbitMQ: 4369, 25672, 5672

MySQL: 3306

Etcd: 2379, 2380

memcached: 11211

```
devstack@openstack:~$ sudo lsof -i -P -n | grep LISTEN
COMMAND     PID            USER   FD   TYPE  DEVICE SIZE/OFF NODE NAME
epmd       1461          rabbitmq    3u   IPv4   24946      0t0  TCP *:4369 (LISTEN)
epmd       1461          rabbitmq    4u   IPv6   24947      0t0  TCP *:4369 (LISTEN)
beam.smp   1585          rabbitmq   44u   IPv4   25658      0t0  TCP *:25672 (LISTEN)
beam.smp   1585          rabbitmq   54u   IPv6   26226      0t0  TCP *:5672 (LISTEN)
uwsgi      1897          devstack    4u   IPv4  418408      0t0  TCP 127.0.0.1:35949 (LISTEN)
uwsgi      1897          devstack    5u   IPv4  418409      0t0  TCP *:8775 (LISTEN)
uwsgi      1903          devstack    4u   IPv4  418408      0t0  TCP 127.0.0.1:35949 (LISTEN)
uwsgi      1904          devstack    4u   IPv4  418408      0t0  TCP 127.0.0.1:35949 (LISTEN)
uwsgi      1905          devstack    5u   IPv4  418409      0t0  TCP *:8775 (LISTEN)
nova-novn  2495          devstack    3u   IPv4  425830      0t0  TCP *:6080 (LISTEN)
tgtd       6537              root    6u   IPv4  428604      0t0  TCP *:3260 (LISTEN)
tgtd       6537              root    7u   IPv6  428605      0t0  TCP *:3260 (LISTEN)
apache2    9397              root    4u   IPv6  445310      0t0  TCP *:80 (LISTEN)
systemd-r 12120 systemd-resolve   13u   IPv4  301308      0t0  TCP 127.0.0.53:53 (LISTEN)
sshd      17191              root    3u   IPv4  233686      0t0  TCP *:22 (LISTEN)
sshd      17191              root    4u   IPv6  233697      0t0  TCP *:22 (LISTEN)
glance-re 20757          devstack    3u   IPv4  389480      0t0  TCP *:9191 (LISTEN)
glance-re 21493          devstack    3u   IPv4  389480      0t0  TCP *:9191 (LISTEN)
uwsgi     21503          devstack    4u   IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
uwsgi     21507          devstack    4u   IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
uwsgi     21508          devstack    4u   IPv4  389698      0t0  TCP 127.0.0.1:60999 (LISTEN)
apache2   23718          www-data    4u   IPv6  445310      0t0  TCP *:80 (LISTEN)
apache2   23719          www-data    4u   IPv6  445310      0t0  TCP *:80 (LISTEN)
neutron-s 24081          devstack    5u   IPv4  393800      0t0  TCP *:9696 (LISTEN)
neutron-s 24221          devstack    5u   IPv4  393800      0t0  TCP *:9696 (LISTEN)
neutron-s 24222          devstack    5u   IPv4  393800      0t0  TCP *:9696 (LISTEN)
neutron-s 24223          devstack    5u   IPv4  393800      0t0  TCP *:9696 (LISTEN)
neutron-s 24224          devstack    5u   IPv4  393800      0t0  TCP *:9696 (LISTEN)
neutron-o 25882          devstack    3u   IPv4  398925      0t0  TCP 127.0.0.1:6633 (LISTEN)
mysqld    26317             mysql   15u   IPv4  672643      0t0  TCP *:3306 (LISTEN)
etcd      27165              root    5u   IPv6  353298      0t0  TCP *:2380 (LISTEN)
etcd      27165              root    6u   IPv4  353299      0t0  TCP 10.0.2.15:2379 (LISTEN)
ovsdb-ser 28179              root   18u   IPv4  316128      0t0  TCP 127.0.0.1:6640 (LISTEN)
memcached 28590         memcache   26u   IPv4  355658      0t0  TCP 127.0.0.1:11211 (LISTEN)
```

Validating **glance** status in **devstack**

```
devstack@openstack:~$ service devstack@g-api status
● devstack@g-api.service - Devstack devstack@g-api.service
   Loaded: loaded (/etc/systemd/system/devstack@g-api.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-05-03 20:05:08 UTC; 4 days ago
 Main PID: 21503 (uwsgi)
   Status: "uWSGI is ready"
    Tasks: 6 (limit: 4664)
   CGroup: /system.slice/system-devstack.slice/devstack@g-api.service
           ├─21503 glance-apiuWSGI master
           ├─21507 glance-apiuWSGI worker 1
           └─21508 glance-apiuWSGI worker 2

<logs>

devstack@openstack:~$ service devstack@g-reg status
● devstack@g-reg.service - Devstack devstack@g-reg.service
   Loaded: loaded (/etc/systemd/system/devstack@g-reg.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-05-03 20:05:05 UTC; 4 days ago
 Main PID: 20757 (glance-registry)
    Tasks: 2 (limit: 4664)
   CGroup: /system.slice/system-devstack.slice/devstack@g-reg.service
           ├─20757 /usr/bin/python /usr/local/bin/glance-registry --config-file=/etc/glance/glance-registry.conf
           └─21493 /usr/bin/python /usr/local/bin/glance-registry --config-file=/etc/glance/glance-registry.conf

<logs>
```

Logging using **devstack**

- devstack runs all services as systemd unit file
- to check the logs of the different services, use the journalctl utility

Usage for **glance-api:**

Complete log of the service

```
devstack@openstack:~$ sudo journalctl –u devstack@g-api.service
```

```
devstack@openstack:~$ sudo journalctl –f –u devstack@g-api.service
```

With option –f Show only the most recent journal entries, and continuously print new entries as they are appended to the journal

Logging example 1/2

```
devstack@openstack:~$ sudo journalctl -f -u devstack@g-api.service
-- Logs begin at Fri 2018-04-27 19:34:32 UTC. -
<last logs>
```

```
devstack@openstack:~$ cd devstack/
devstack@openstack:~/devstack$ source openrc demo demo
WARNING: setting legacy OS_TENANT_NAME to support cli tools.
devstack@openstack:~/devstack$ openstack image list
+--------------------------------------+--------------------------+--------+
| ID                                   | Name                     | Status |
+--------------------------------------+--------------------------+--------+
| 8fe8afd8-81e4-4378-a28a-40e019270b8e | cirros-0.3.5-x86_64-disk | active |
+--------------------------------------+--------------------------+--------+
```

Logging example 2/2

```
May 08 02:14:11 openstack devstack@g-api.service[21503]: DEBUG glance.api.middleware.version_negotiation
[None req-24b010e8-839a-4de0-9553-1d091cef6639 demo demo] Determining version of request: GE
May 08 02:14:11 openstack devstack@g-api.service[21503]: DEBUG glance.api.middleware.version_negotiation
[None req-24b010e8-839a-4de0-9553-1d091cef6639 demo demo] Using url versioning {{(pid=21507)
May 08 02:14:11 openstack devstack@g-api.service[21503]: DEBUG glance.api.middleware.version_negotiation
[None req-24b010e8-839a-4de0-9553-1d091cef6639 demo demo] Matched version: v2 {{(pid=21507)
May 08 02:14:11 openstack devstack@g-api.service[21503]: DEBUG glance.api.middleware.version_negotiation
[None req-24b010e8-839a-4de0-9553-1d091cef6639 demo demo] new path /v2/images {{(pid=21507)
May 08 02:14:11 openstack devstack@g-api.service[21503]: [pid: 21507|app: 0|req: 77/154] 127.0.0.1 ()
{36 vars in 720 bytes} [Wed May  8 02:14:11 2019] GET /v2/images => generated 671 bytes in 191 msecs
(HTTP/1.1 200) 4 headers in 156 bytes (1 switches on core 0)
```

CLI debug mode

```
devstack@openstack:~/devstack$ openstack --debug image list
START with options: [u'--debug', u'image', u'list']
REQ: curl -g -i -X GET http://10.0.2.15/identity -H "Accept: application/json" -H "User-Agent: osc-lib/1.10.0
keystoneauth1/3.5.0 python-requests/2.18.4 CPython/2.7.15rc1"
Starting new HTTP connection (1): 10.0.2.15
http://10.0.2.15:80 "GET /identity HTTP/1.1" 300 268
http://10.0.2.15:80 "POST /identity/v3/auth/tokens HTTP/1.1" 201 3331
http://10.0.2.15:80 "POST /identity/v3/auth/tokens HTTP/1.1" 201 3331
http://10.0.2.15:80 "GET /image/v2/images HTTP/1.1" 200 671
RESP: [200] Connection: close Content-Length: 671 Content-Type: application/json Date: Wed, 08 May 2019 03:39:33 GMT Server:
Apache/2.4.29 (Ubuntu) x-openstack-request-id: req-b0f9d9e5-10dd-4fa2-8c00-8d708415788e
RESP BODY: {"images": [{"status": "active", "name": "cirros-0.3.5-x86_64-disk", "tags": [], "container_format": "bare",
"created_at": "2019-05-03T20:05:13Z", "size": 13267968, "disk_format": "qcow2", "updated_at": "2019-05-03T20:05:13Z",
"visibility": "public", "self": "/v2/images/8fe8afd8-81e4-4378-a28a-40e019270b8e", "min_disk": 0, "protected": false, "id":
"8fe8afd8-81e4-4378-a28a-40e019270b8e", "file": "/v2/images/8fe8afd8-81e4-4378-a28a-40e019270b8e/file", "checksum":
"f8ab98ff5e73ebab884d80c9dc9c7290", "owner": "6a4681c883d54628b5104c83d4aea388", "virtual_size": null, "min_ram": 0,
"schema": "/v2/schemas/image"}], "schema": "/v2/schemas/images", "first": "/v2/images"}
http://10.0.2.15:80 "GET /image/v2/images?marker=8fe8afd8-81e4-4378-a28a-40e019270b8e HTTP/1.1" 200 69
RESP: [200] Connection: close Content-Length: 69 Content-Type: application/json Date: Wed, 08 May 2019 03:39:34 GMT Server:
Apache/2.4.29 (Ubuntu) x-openstack-request-id: req-288ce1d7-570a-4ac1-9b6e-289ac8082fe2
RESP BODY: {"images": [], "schema": "/v2/schemas/images", "first": "/v2/images"}
GET call to http://10.0.2.15/image/v2/images?marker=8fe8afd8-81e4-4378-a28a-40e019270b8e used request id req-288ce1d7-570a-
4ac1-9b6e-289ac8082fe2
+--------------------------------------+--------------------------+--------+
| ID                                   | Name                     | Status |
+--------------------------------------+--------------------------+--------+
| 8fe8afd8-81e4-4378-a28a-40e019270b8e | cirros-0.3.5-x86_64-disk | active |
+--------------------------------------+--------------------------+--------+
clean_up ListImage:
END return value: 0
```

Logging and CLI debug together (example):

Prepare the loggging for Glance and Keystone in 2 different terminals:

- `devstack@openstack:~/devstack$ sudo journalctl -f -u devstack@g-api.service`
- `devstack@openstack:~/devstack$ sudo journalctl -f -u devstack@keystone.service`

Execute the CLI command (or through the GUI via Horizon):

- `devstack@openstack:~/devstack$ openstack --debug image list`