Overlay networking with **OpenStack Neutron** in Public Cloud environment

Trex Workshop 2015

NEBULA PILVI 9.0

nebula

About

- Presenter
  - Anton Aksola (aakso@Twitter,IRCNet,Github)
  - Network Architect @Nebula Oy, started in 2005
  - Currently working heavily with OpenStack
  - Focusing in Networking and Software Development
- Nebula Oy
  - ISP, hosting and IT service company established in 1997
  - Turnover in 2013 was ~26M€
  - 120 employees currently
  - Offering ranging from Cloud Services to Managed Services and traditional IT services

OpenStack?

In a nutshell

- Collection of software projects for providing cloud services
- Core projects allow you to run Infrastructure as a Service cloud
  - Compute (Nova)
  - Storage (Cinder and Swift)
  - Network (Neutron)
  - Identity (Keystone)
- Other projects include
  - Dashboard (Horizon)
  - Orchestration (Heat)
  - Telemetry (Ceilometer)
  - Database (Trove)
  - … and many more

Network - Neutron

- Network orchestration framework that provides essential and supporting network services to OpenStack cloud
- Core functionality
  - Network connectivity
  - SDN: user defined arbitrary topologies
  - Basic IPAM
- Supporting services such as DHCP, DNS, Perimeter FW, Security Groups and VPN
- Consists of multiple plugins and drivers both commercial and open source
- Unified northbound API

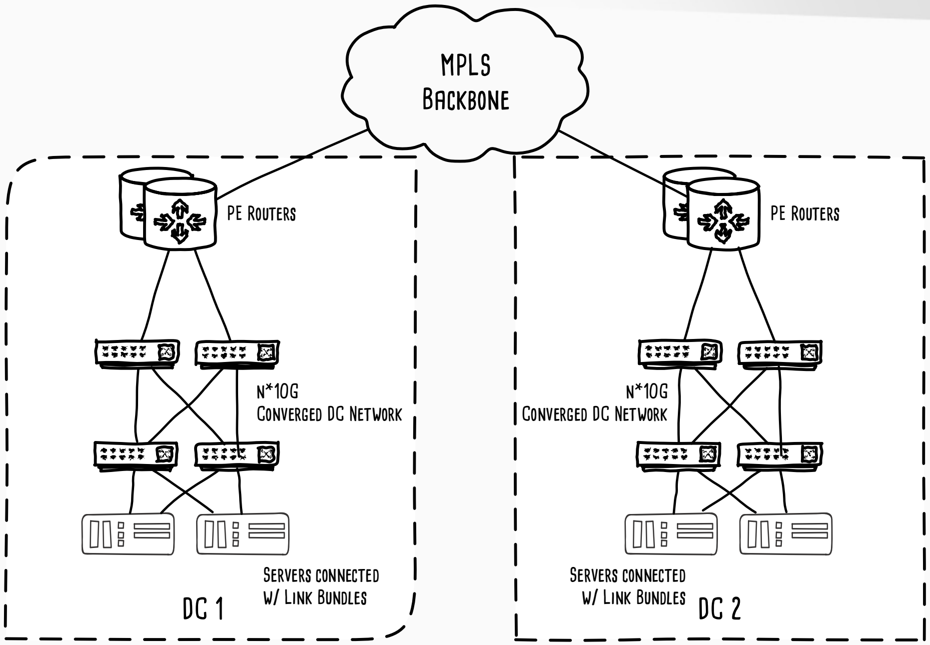
Our Public Cloud journey

- We started researching for an alternative cloud platform in Autumn 2013
  - Before this we've had offering based on commercial products
  - Legacy systems also included Xen and Hyper-V based virtualization with static network configuration
- Main targets from network point of view
  - Flexible: users must be able to provision network resources on demand (segments, subnets and interfaces)
  - Fault tolerant: solution needs to be available on two distinct datacenters
  - Secure: user separation must be built-in
  - Scalable: must support large amount of configurations and performance must be in par with current offering

nebula

Starting point

- We had two almost identical datacenters with good network gear
- Ability to use AToM, VPLS and L3VPNs over the MPLS backbone
- So pretty good base
- But:
    - Legacy systems use VLANs for customer separation
    - VLAN ids are DC significant
    - VLAN and VPLS/L3VPN provisioning is static and done by the operator with provisioning scripts
    - No existing APIs



Networking models in Neutron

- In 2013 Neutron supported following models:
  - Flat: all compute instances join to a single network. No customer isolation is possible and no SDN features are available
  - VLAN: users can create custom networks and neutron allocates segmentation ids (VLANs) from predefined ranges
  - Overlay: same model as with VLANs but an overlay protocol (GRE/VXLAN) is used to transport customer traffic between hypervisors
- Neutron implements classic L2 network segments in all operating modes
  - Other commercial public clouds have their specific solutions: Amazon has VPC and Microsoft has Hyper-V Network Virtualization
- L3 functions are handled by L3-agents

nebula

Comparing VLAN and Overlay models

VLAN

Pros

- Pretty straight-forward solution – what we have done for ages
- Predictable performance
- Ability for legacy servers to join to the customer network

Cons

- Limited number of segments available (4094). Need to coordinate with existing allocations
- QinQ cannot be used as we want true separation in MAC level
- PBB push/pop is not supported in hypervisors so we cannot leverage that
- Every VLAN needs to be mapped to a unique VPLS instance for Inter-DC connectivity. PBB-VPLS is also a possibility but we didn't have support for it.
- Need to pre-provision all possible VLANs or integrate with network gear

OVERLAY

Pros

- Much more segments available. In VXLAN the VNID field is 24bits long
- Only one segment required between hypervisors for customer traffic
- L3VPN can be utilized for Inter-DC traffic instead of VPLS
- True separation in MAC level
- Network topology agnostic

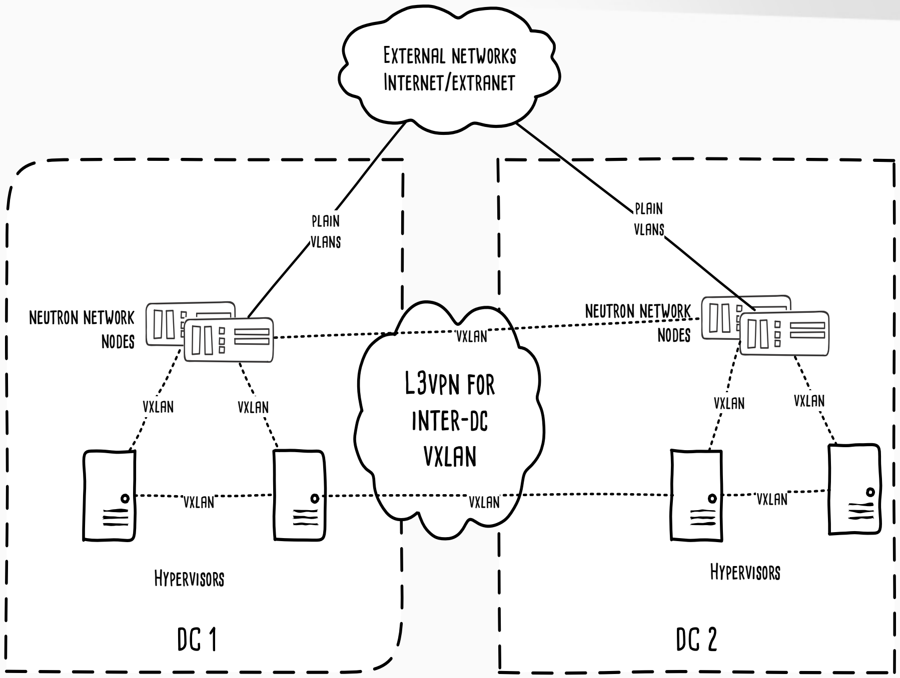
Cons

- Unknown performance. The overhead with VXLAN is much more severe than with MPLS or GRE
- Lack of visibility. Current policies do not apply
- Joining legacy servers to customer networks is not possible
- BUM unicast replication

nebula

Our setup

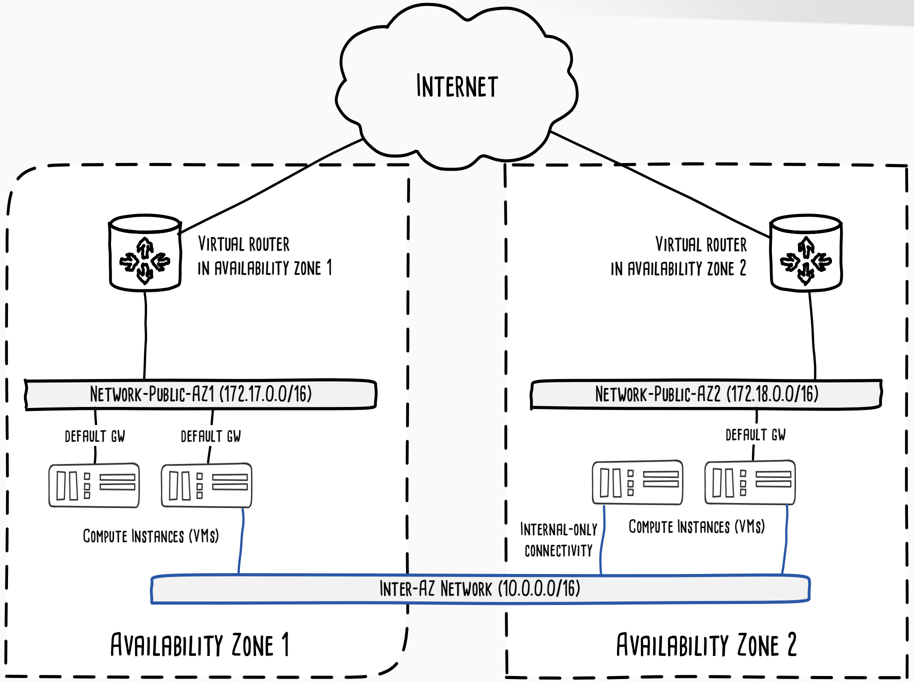
- We chose VXLAN for encapsulation as it was becoming a industry standard
- Currently only software components participate in VXLAN. This can change in the future as VXLAN in HW switches is becoming more common
- We use MPLS-VPN to transport the VXLAN traffic between DCs. This gives us separation, some security and more flexible ways to manage the traffic
- External access is handled by network nodes that route between cloud networks and traditional networks



nebula

Example topology inside OpenStack

- Users can create complex network topologies with multiple Virtual Routers and networks spanning availability zones
- Inter-AZ networks can be used, for example, database clustering or similar applications
- Floating addresses can be allocated to virtual routers that will do 1-to-1 NAT to a desired VM
- Virtual Routers can also do SNAT for VMs that do not have floating address. This can be used for fetching updates from the Internet etc.



nebula

L2 forwarding in OpenStack

- Modular Layer2 plugin introduced in Havana release
- Defines type and mechanism drivers to handle specific tasks
- Type drivers include: flat, local, gre, vlan and vxlan
- Mechanism drivers: openvswitch, linuxbridge, ofagent, l2pop and multiple commercial options for example cisco, arista and nuage
- Multiple drivers can be loaded at the same time

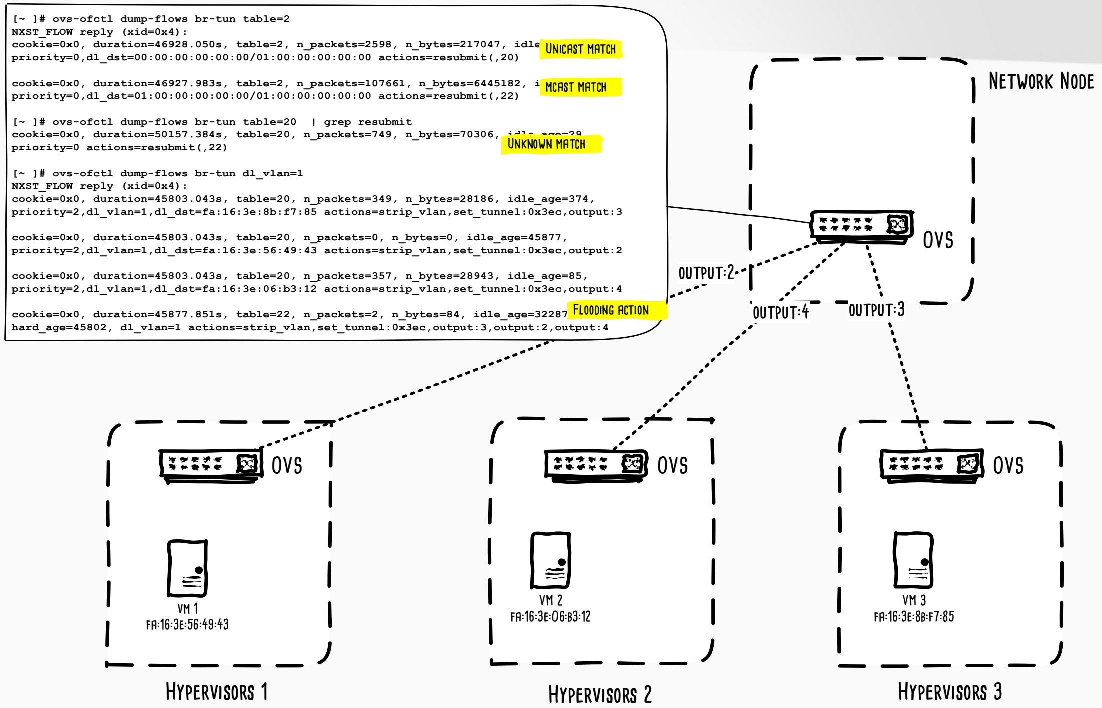
- As we wanted a pure OSS solution, openvswitch was our choice
- Open vSwitch is a flexible switching solution for Linux that runs in userland but has datapath-support in Linux Kernel

Open vSwitch + L2 Population

- Open vSwitch supports traffic steering with OpenFlow rules
- This allows L2 population mechanism driver to be implemented in OVS
- As every single endpoint is known inside the cloud, all L2 forwarding entries can be pre-populated to nodes that need them
- Unknown unicast flooding is reduced
- BUM traffic should be minimized in overlay model as it becomes unicast traffic in hypervisor egress
  - VXLAN RFC defines *Broadcast Communication and Mapping to Multicast* but it is not currently implemented in OVS

L2 Forwarding example

- L2 forwarding entries are populated as OpenFlow entries by the **neutron-openvswitch-agent**
- set_tunnel:0x3ec defines the VNID for the traffic
- output is the logical VXLAN tunnel
- The last entry is the BUM entry as it has multiple output actions
- BUM entry includes only hypervisors that have VMs in this particular logical network (partial mesh)
- There is still source learning so L2population is purely an optimization feature



OpenStack deployment

Challenges - General

- As a whole, OpenStack deployment can be quite difficult and time consuming. This applies especially to getting the system stable
- Most of our issues have been software issues
  - Things are getting better by the day though
  - Good dev and qa environments are your friends
- OpenStack by nature very distributed system so there is really no logical central management point
  - This is why configuration needs to be consistent in every component
  - Do not try to install OpenStack by hand, use automation frameworks such as Chef, Puppet or Ansible

OpenStack deployment

Challenges – VXLAN

- VXLAN is still quite new technology, the first Internet Draft was released in Feb 2012
- It was designed as a encapsulation for virtualized datacenters
- It provides some entropy in the outer UDP header for hashing in L2 bundles or ECMP
  - Thus it fits better to an existing network than (NV)GRE
- Support has been available in Linux Kernel and OVS for quite some time now
- There are still problems…

OpenStack deployment

Challenges – VXLAN

- …which are not VXLAN problems but Ethernet problems
- As we moved to 10G NICs in servers we got dependent on NIC offload features
- With 1514B ethernet frame size the packet rate will be over 800Kpps
- If we need to process every packet individually we quickly max out our CPUs for interrupt handling
- Features such as GSO (tx) and GRO (rx) help us to handle the traffic by combining packets belonging to a same flow
- But HW assisted features tend to only work if the IP protocol is TCP

OpenStack deployment

Challenges – VXLAN

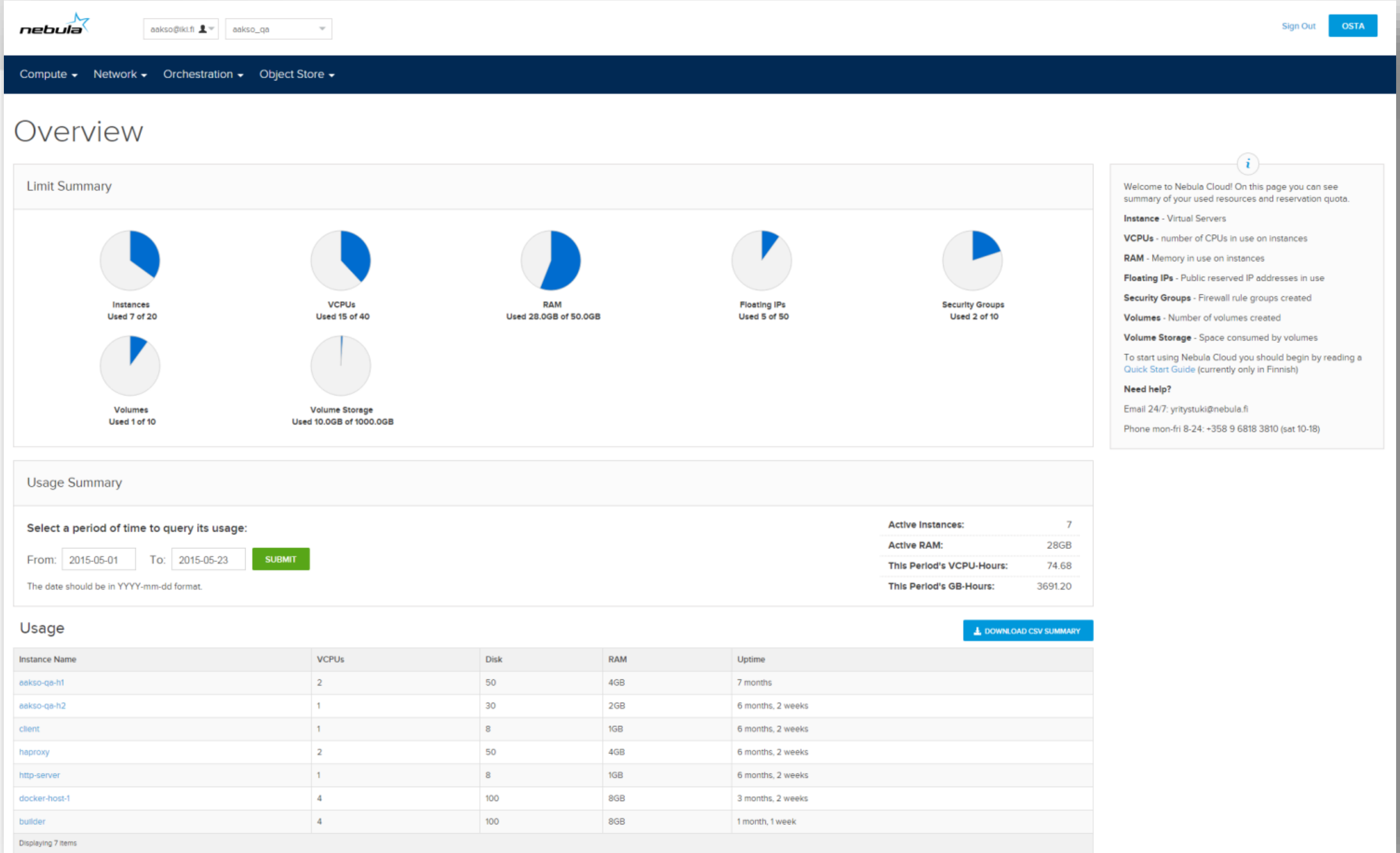
- With VXLAN we need to look deeper into the packet
- Support in NICs is still quite rare
- Support in Linux kernel is quite new so recent Kernel is required (upstream >= 3.14)
- Expect things to improve as more users choose VXLAN
- Traditional VLANs are still the only possibility if you need the best performance possible

Word from the sponsor

- Nebula launched public cloud offering *Nebula Cloud 9.0* in November 2014
- First public cloud in Finland based in OpenStack
- We offer
  - True multi-datacenter solution. Infrastructure is not shared between availability zones. Networks can be terminated to either availability zone
  - Compute instances ranging 1CPU 1GB to 16CPU 128GB
  - Block storage in SSD, SAS and archive grades
  - Object Storage based on CEPH where data is replicated to multiple datacenters
  - SDN network services
  - Dashboard, Orchestration and telemetry services
- Standard OpenStack APIs available

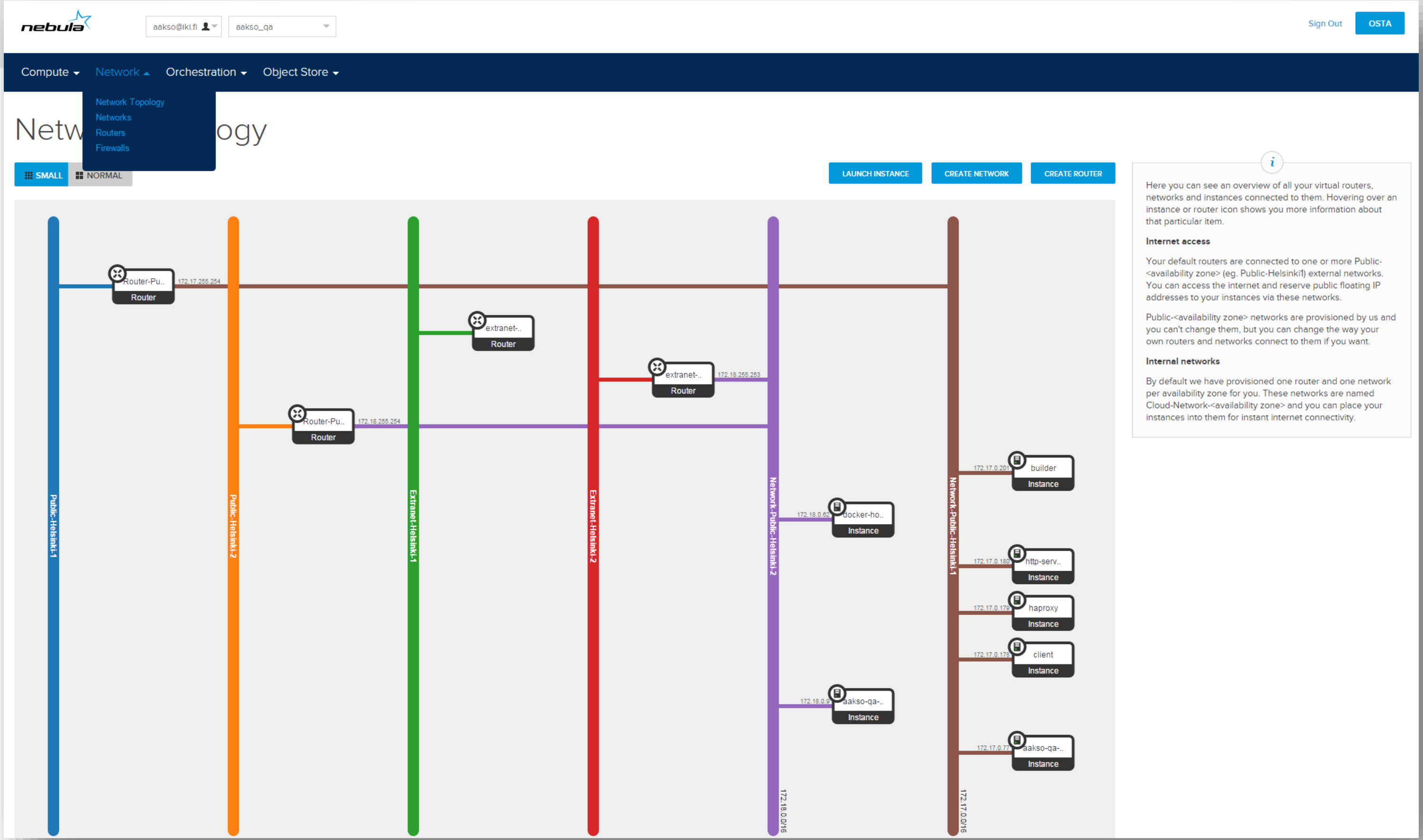
Dashboard

NEBULA PILVI 9.0



nebula

Dashboard



aakso@iki.fi
aakso_qa
Sign Out
OSTA
Compute
Network
Orchestration
Object Store
Network Topology
Networks
Routers
Firewalls
Netw ogy
SMALL
NORMAL
LAUNCH INSTANCE
CREATE NETWORK
CREATE ROUTER
Public-Helsinki-1
Public-Helsinki-2
Extranet-Helsinki-1
Extranet-Helsinki-2
Network-Public-Helsinki-2
Network-Public-Helsinki-1
Router-Pu..
Router
extranet-..
Router
builder
Instance
docker-ho..
http-serv..
haproxy
client
aakso-qa-..
172.18.0.0/16
172.17.0.0/16
Here you can see an overview of all your virtual routers, networks and instances connected to them. Hovering over an instance or router icon shows you more information about that particular item.
Internet access
Your default routers are connected to one or more Public-<availability zone> (eg. Public-Helsinki1) external networks. You can access the internet and reserve public floating IP addresses to your instances via these networks.
Public-<availability zone> networks are provisioned by us and you can't change them, but you can change the way your own routers and networks connect to them if you want.
Internal networks
By default we have provisioned one router and one network per availability zone for you. These networks are named Cloud-Network-<availability zone> and you can place your instances into them for instant internet connectivity.

CLI

```
[mem: 600/1497MB - load: 0.00 - up: 71 days, 17:00  :: development-s4 :: ]
[aakso@cloudadmin.s4.mng.dev - /dev/pts/4] [2015-05-23T13:53:29+0300]
[~ ]$ neutron net-list
+--------------------------------------+-----------------------------+-----------------------------------------------------+
| id                                   | name                        | subnets                                             |
+--------------------------------------+-----------------------------+-----------------------------------------------------+
| 8a733b7c-30d1-4ec5-bf57-7a9a5874a7f8 | Network-Public-Helsinki-1   | fc3d42fc-50f6-4532-ac95-90c558f946b5 172.17.0.0/16 |
| d808be11-3667-422c-8332-3d873899b44e | Public-Helsinki-1           | ed265e60-29b0-4f1e-a679-10fc05e7e0bd                |
+--------------------------------------+-----------------------------+-----------------------------------------------------+
[mem: 600/1497MB - load: 0.00 - up: 71 days, 17:00  :: development-s4 :: ]
[aakso@cloudadmin.s4.mng.dev - /dev/pts/4] [2015-05-23T13:53:32+0300]
[~ ]$ openstack server list --name helsinki-1_2
+--------------------------------------+-------------+--------+--------------------------------------+
| ID                                   | Name        | Status | Networks                             |
+--------------------------------------+-------------+--------+--------------------------------------+
| 4a705148-8215-4b4d-8727-a0fb28c6e8d9 | helsinki-1_2 | ACTIVE | Network-Public-Helsinki-1=172.17.0.11 |
+--------------------------------------+-------------+--------+--------------------------------------+
[mem: 600/1497MB - load: 0.00 - up: 71 days, 17:00  :: development-s4 :: ]
[aakso@cloudadmin.s4.mng.dev - /dev/pts/4] [2015-05-23T13:53:35+0300]
[~ ]$ openstack server show 4a705148-8215-4b4d-8727-a0fb28c6e8d9
+--------------------------------------+------------------------------------------------------------------+
| Field                                | Value                                                            |
+--------------------------------------+------------------------------------------------------------------+
| OS-DCF:diskConfig                    | AUTO                                                             |
| OS-EXT-AZ:availability_zone          | helsinki-1                                                       |
| OS-EXT-STS:power_state               | 1                                                                |
| OS-EXT-STS:task_state                | None                                                             |
| OS-EXT-STS:vm_state                  | active                                                           |
| OS-SRV-USG:launched_at               | 2015-05-21T13:17:43.000000                                       |
| OS-SRV-USG:terminated_at             | None                                                             |
| accessIPv4                           |                                                                  |
| accessIPv6                           |                                                                  |
| addresses                            | Network-Public-Helsinki-1=172.17.0.11                            |
| config_drive                         |                                                                  |
| created                              | 2015-05-21T13:17:34Z                                             |
| flavor                               | nbl-n1-tiny (100)                                                |
| hostId                               | dbe47cf160c6143254e5ed4330df418c4f0672b80c9f4eaad2fae09a         |
| id                                   | 4a705148-8215-4b4d-8727-a0fb28c6e8d9                             |
| image                                | CentOS 7 (9cc53123-9941-4550-b21e-1b8872f81eee)                  |
| key_name                             | aakso-cloudtest-laptop                                           |
| name                                 | helsinki-1_2                                                     |
| os-extended-volumes:volumes_attached | []                                                               |
| progress                             | 0                                                                |
| properties                           |                                                                  |
| security_groups                      | [{u'name': u'default'}]                                          |
| status                               | ACTIVE                                                           |
| tenant_id                            | fc225bb18cab434e9ede654e5def7e67                                 |
| updated                              | 2015-05-21T13:17:43Z                                             |
| user_id                              | 4ef2fdf7d9dd41388e29d8e4a9d99022                                 |
+--------------------------------------+------------------------------------------------------------------+
[mem: 600/1497MB - load: 0.00 - up: 71 days, 17:00  :: development-s4 :: ]
[aakso@cloudadmin.s4.mng.dev - /dev/pts/4] [2015-05-23T13:53:48+0300]
[~ ]$
```

Thank you

Visit **www.nebulacloud.fi** for more information