Amirkabir University of Technology
(Tehran Polytechnic)

Cloud Computing

OpenStack Nova Architecture

Seyyed Ahmad Javadi
sajavadi@aut.ac.ir

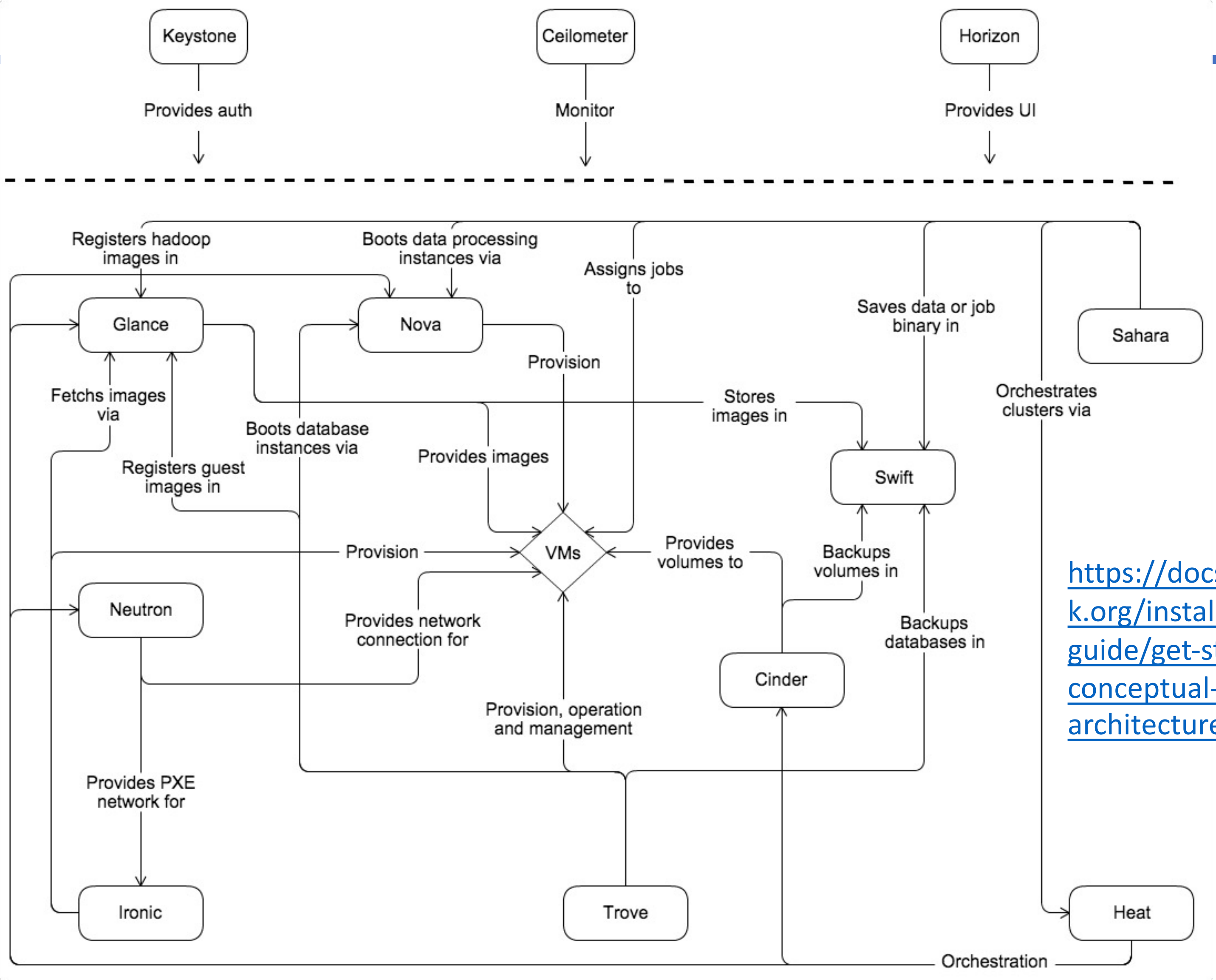
Spring 2023

https://www.slideshare.net/HaimAteya/an-intrudction-to-openstack-2017

https://docs.openstack.org/security-guide/introduction/introduction-to-openstack.html

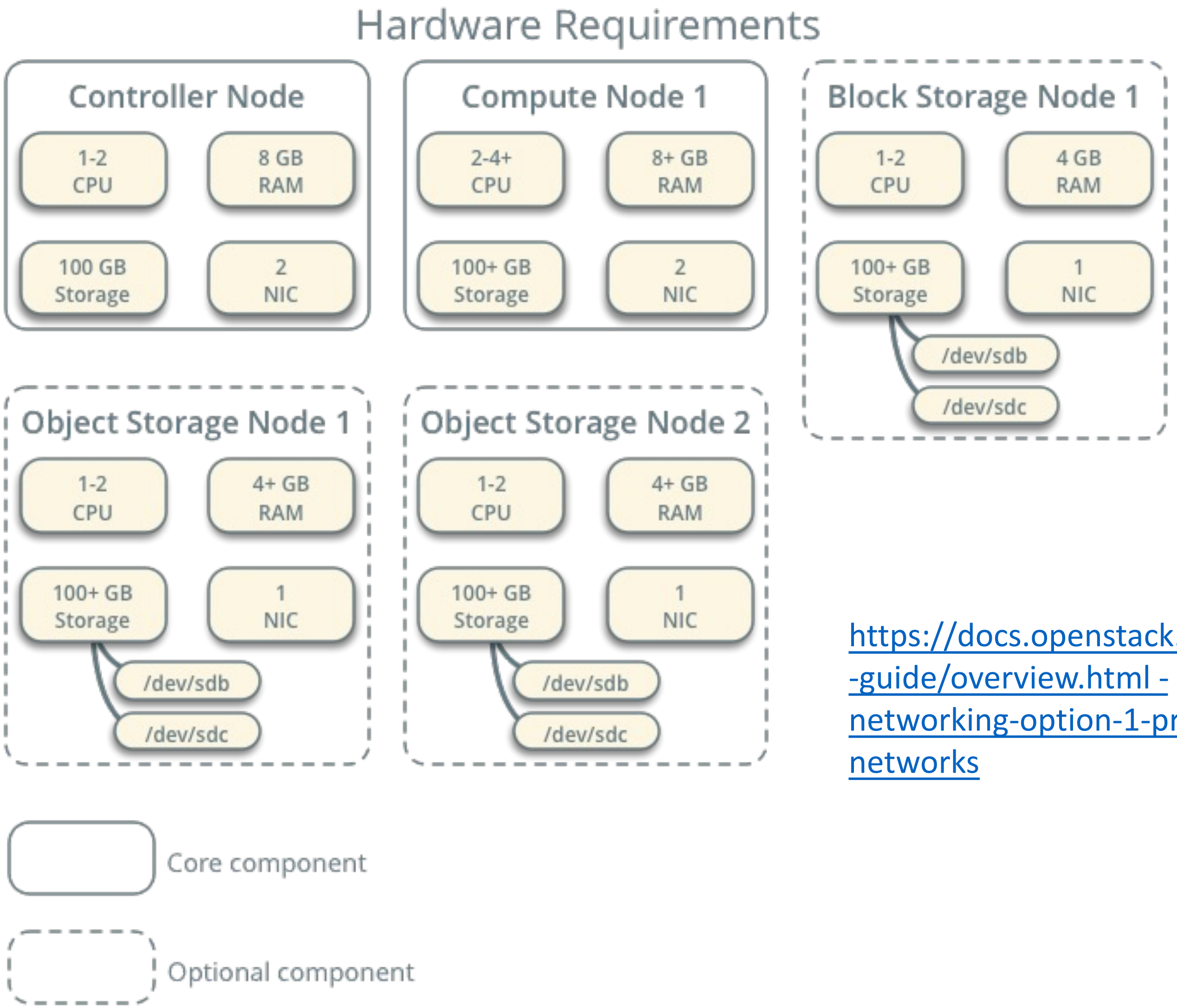
Overview

OpenStack Conceptual Architecture



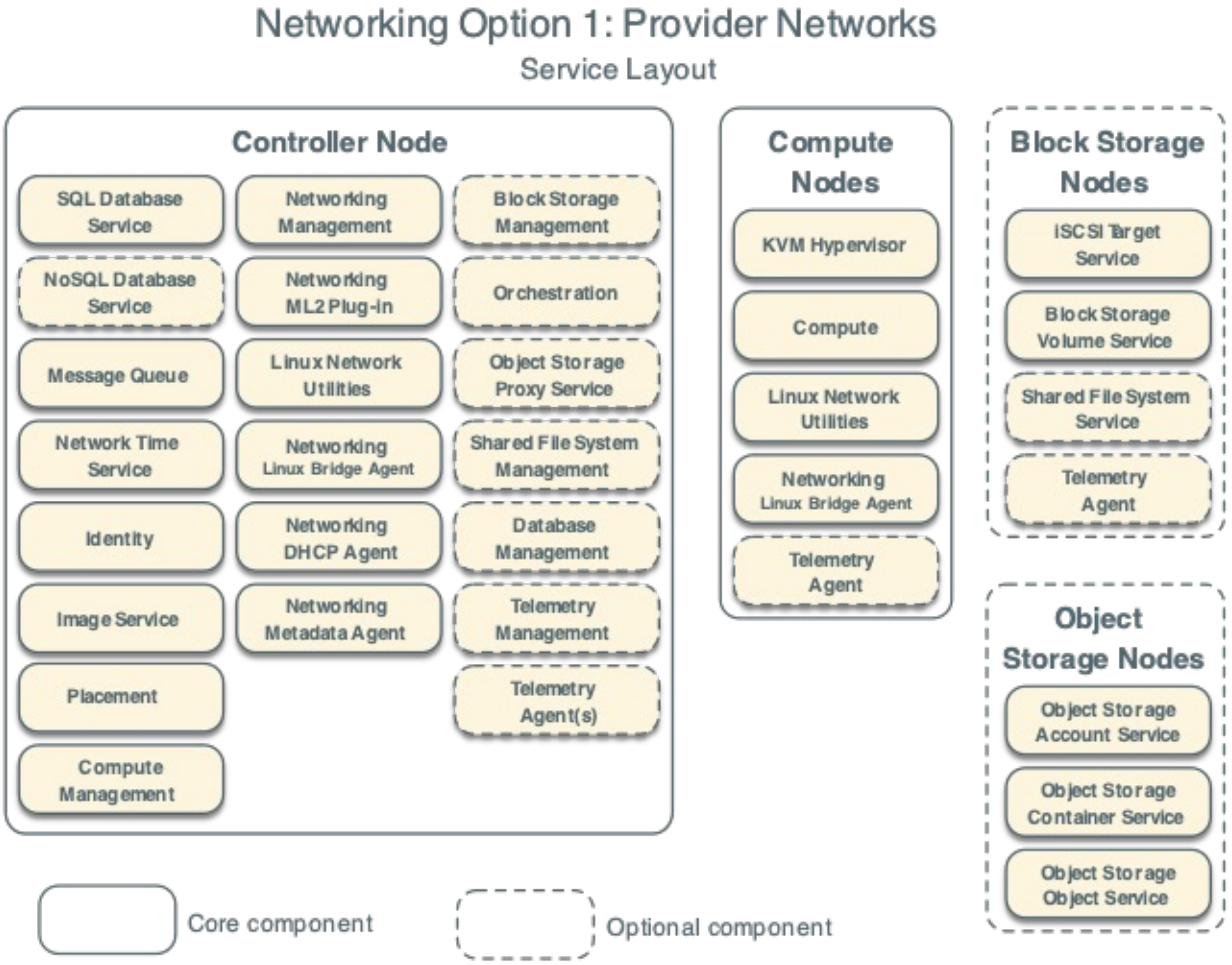
https://docs.openstack.org/install-guide/get-started-conceptual-architecture.html

OpenStack Installation



[https://docs.openstack.org/install-guide/overview.html - networking-option-1-provider-networks](https://docs.openstack.org/install-guide/overview.html#networking-option-1-provider-networks)

OpenStack Installation



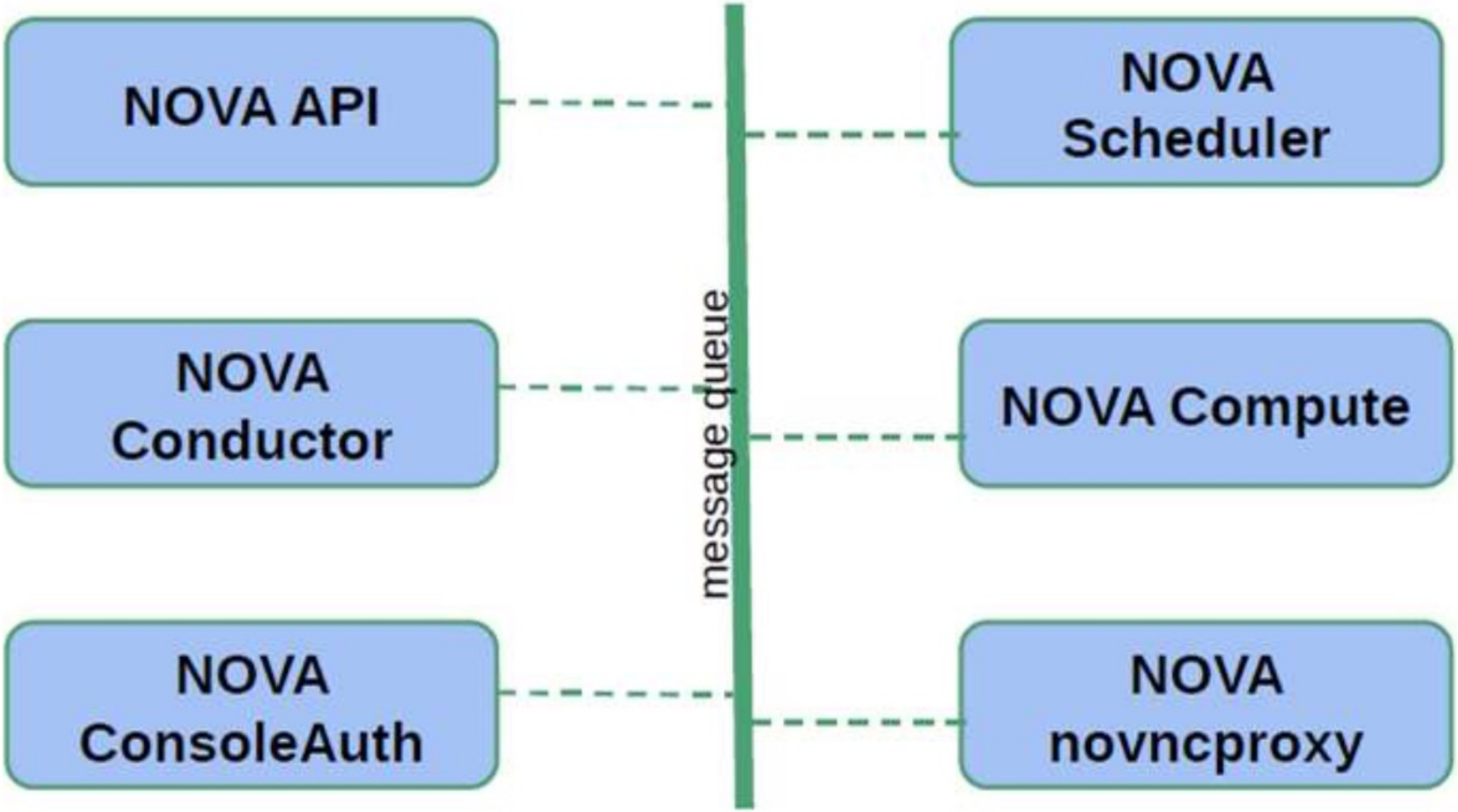
Run Kubernetes Cluster on OpenStack

- Check slides and watch the video
  - https://object-storage-ca-ymq-1.vexxhost.net/swift/v1/6e4619c416ff4bd19e1c087f27a43eea/www-assets-prod/summits/27/presentations/24157/slides/OpenInfra-Summit-Shanghai-OpenShift-on-OpenStack.pdf
  - https://www.youtube.com/watch?v=DuBYWXTnnsg
  - https://www.youtube.com/watch?v=uiplRQ2pQfc&t=176s

Nova

- Provided compute as service
- The main part of an IaaS system
- It is designed to manage and automate pools of computer resources
- Compute’s architecture is designed to scale horizontally

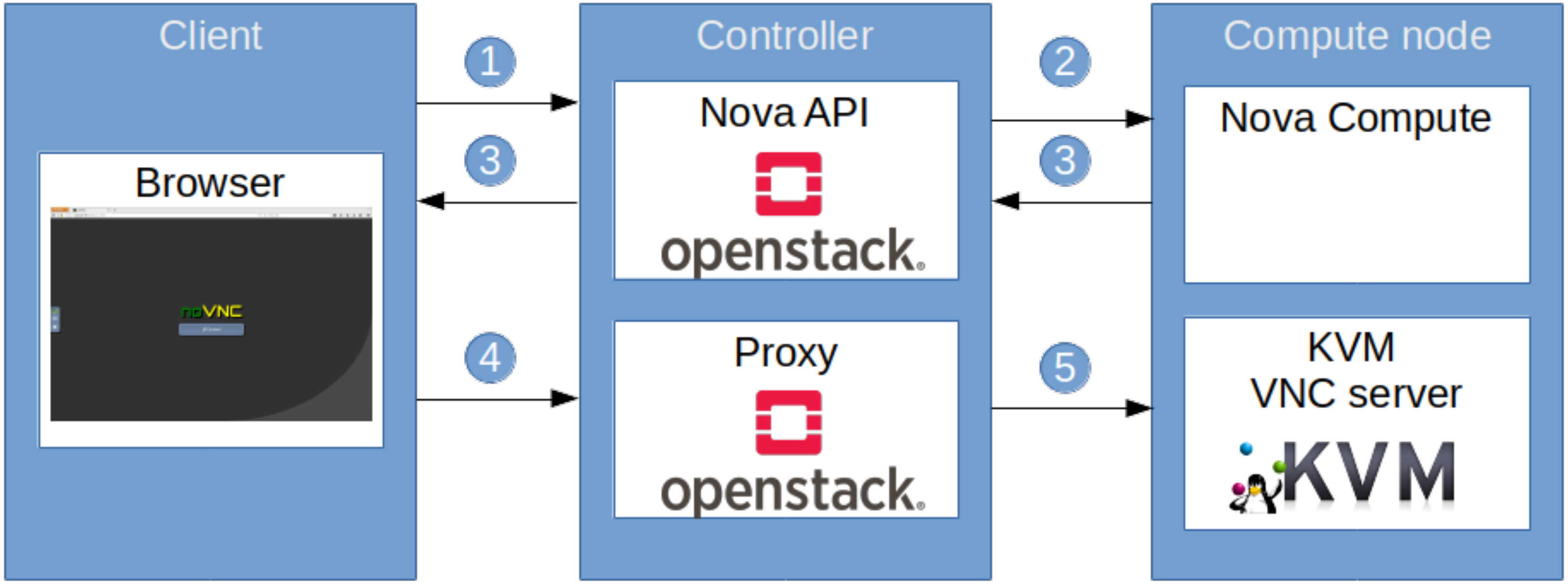
Nova Components



Nova Components

- Nova-conductor
  - Provides database-access support for Compute nodes
- Nova-consoleauth
  - Handles console authentication
- Nova-novncproxy
  - Provides a VNC proxy for browsers

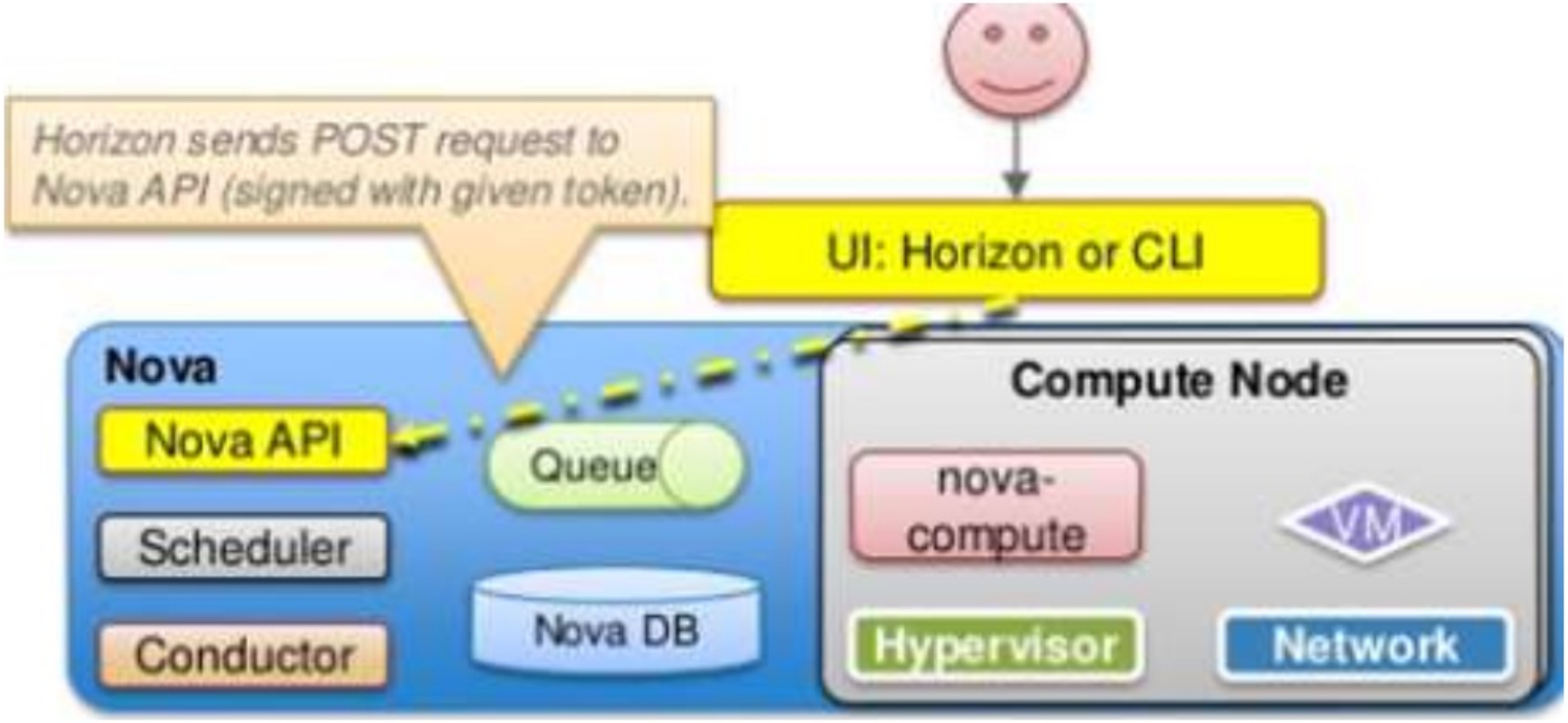
The Nova VNC proxy



https://leftasexercise.com/2020/02/14/openstack-nova-installation-and-overview/

NOVA API

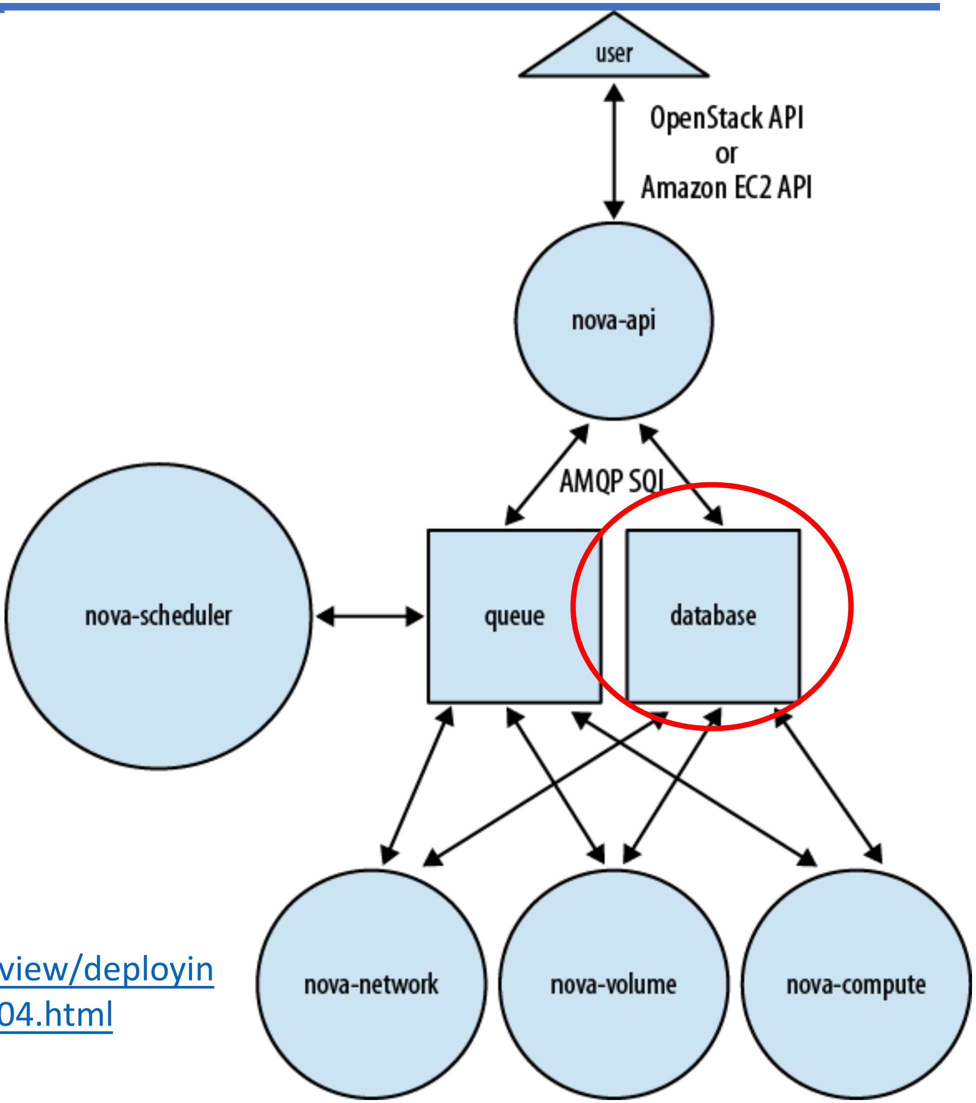
- NOVA-API is responsible to provide an API for users and services to interact with NOVA



Horizon sends POST request to Nova API (signed with given token).

On Compute Node

- There is a periodic task (Resource Tracker), which collects host information.
- This information is then stored to ***database***

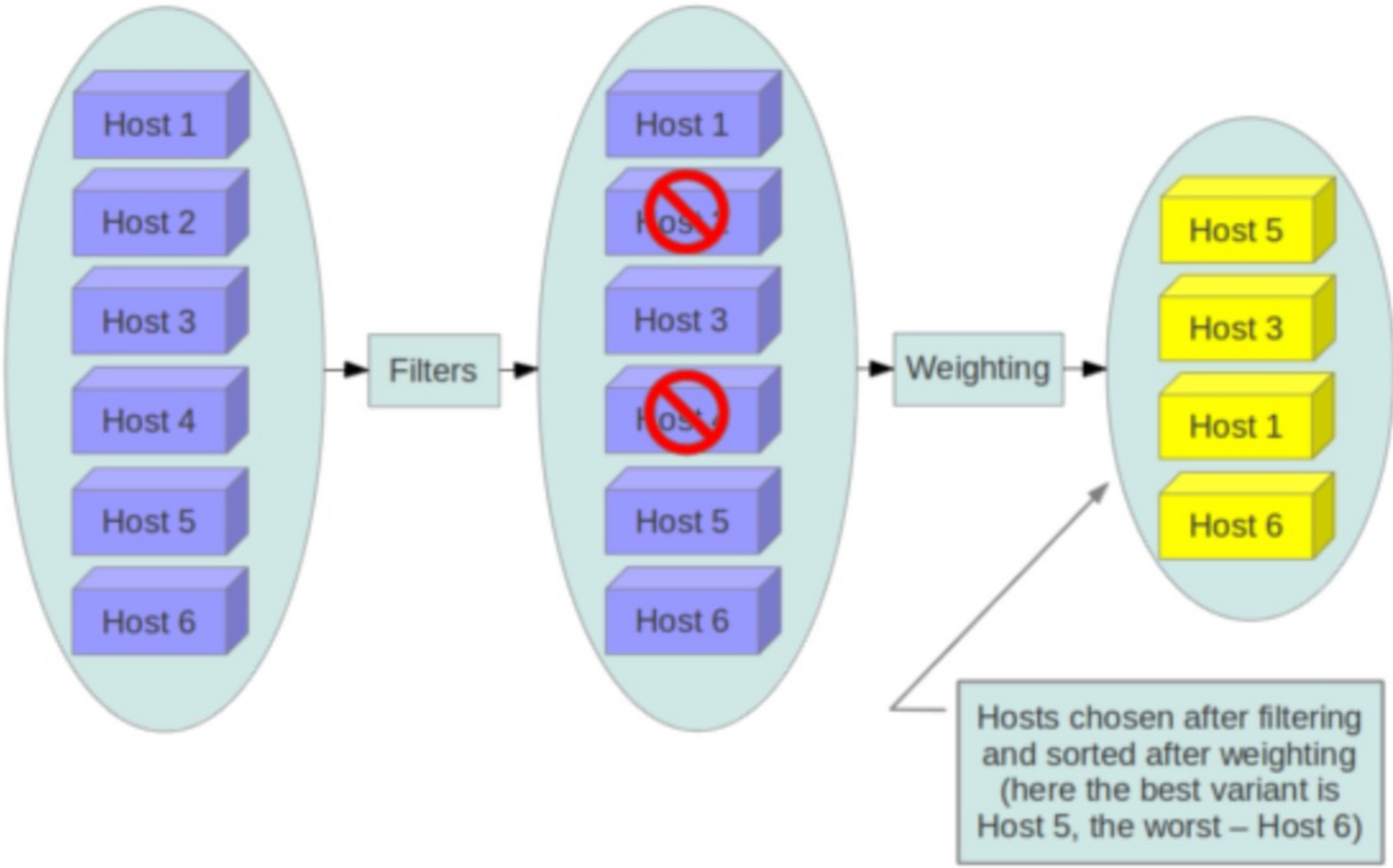


https://www.oreilly.com/library/view/deploying-openstack/9781449311223/ch04.html

On Controller Node

- Request from nova API reaches conductor
- Conductor interacts with the scheduler
- Scheduler ***uses filters*** to identify the best node
    - From the information stored in ***database***
- Selected host information is sent back to conductor
- Conductor uses the compute queue and directs it to selected host
- The compute node then launches the instance

Filters and Weights



Hosts chosen after filtering and sorted after weighting (here the best variant is Host 5, the worst – Host 6)

Some Common Filters

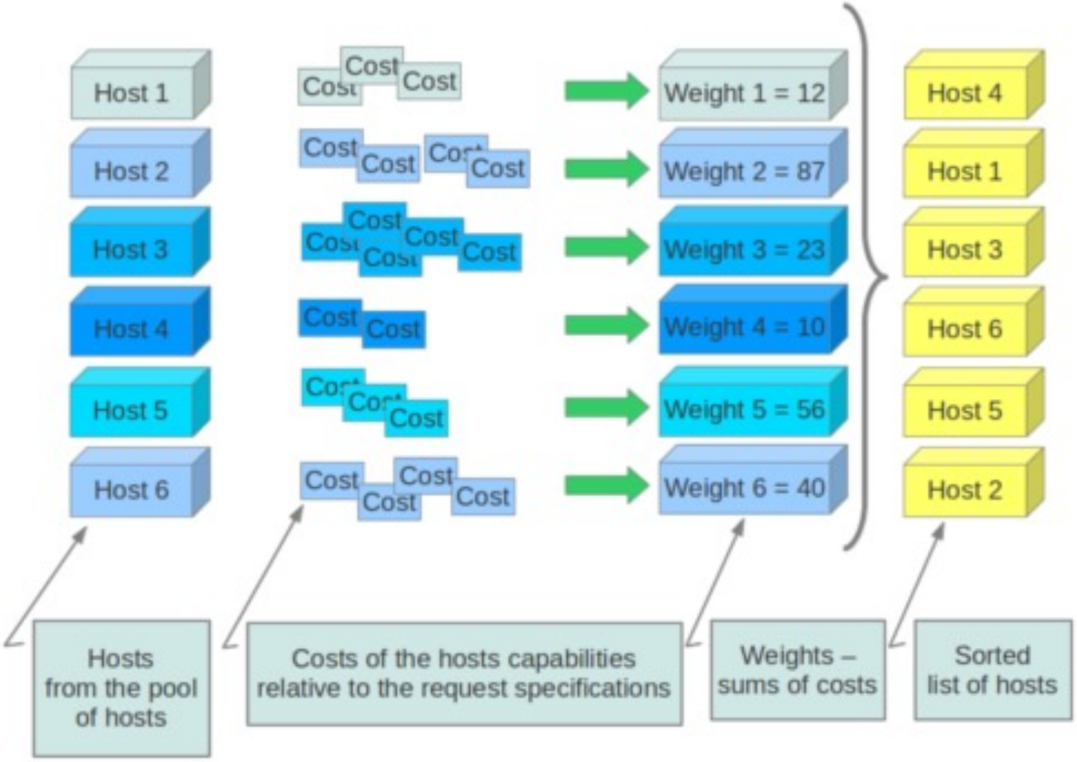
- **AvailabilityZoneFilter**
  - Return hosts where node_availability_zone name is the same as the one requested
- **RamFilter**
  - Return hosts where (free_ram * ram_allocation_ration) is greater than requested ram.
- **ComputerFilter**
  - Return hosts where asked instance_type (with extra_specs) match capabilities

Some Common Filters (cont.)

- **DiskFilter**
  - Returns hosts with sufficient disk space available for root and ephemeral storage.
- **RetryFilter**
  - Filters out hosts that have already been attempted for scheduling purposes.

Weights

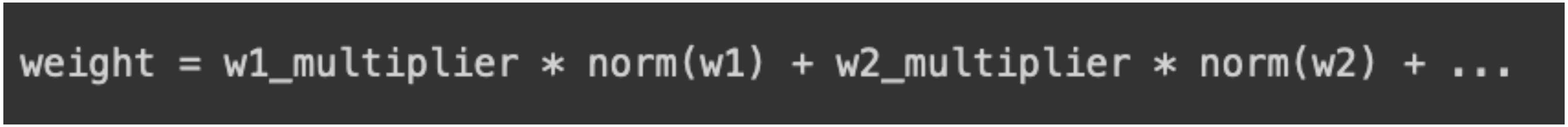
- Scheduler applies cost function on each host & calculates the weight.



https://docs.openstack.org/nova/latest/admin/scheduling.html

Some Possible Cost Functions

- Considering free RAM among filtered hosts.
  - Highest free RAM wins.
- Considering least workload (e.g., IO ops) among filtered hosts.
- Can consider any specific metric we want to consider in a similar fashion.
  - Can be enabled from configuration file.



```
weight = w1_multiplier * norm(w1) + w2_multiplier * norm(w2) + ...
```

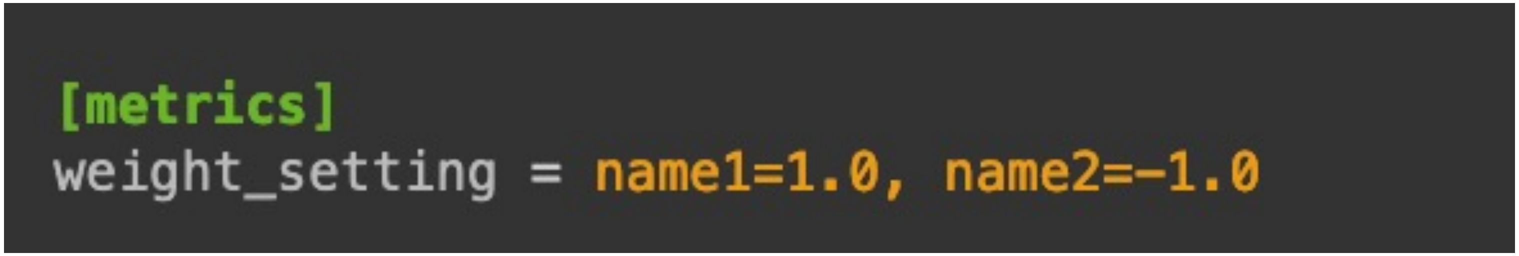
Weights (cont.)

- **RAMWeigher**
  - Compute weight based on available RAM on the compute node. Sort with the largest weight winning.
- **CPUWeigher**
  - Compute weight based on available vCPUs om the compute node. Sort with the largest weight winning.
- **DiskWeigher**
  - Hosts are weighted and sorted by free disk space with the largest weight wininng.

Weights (cont.)

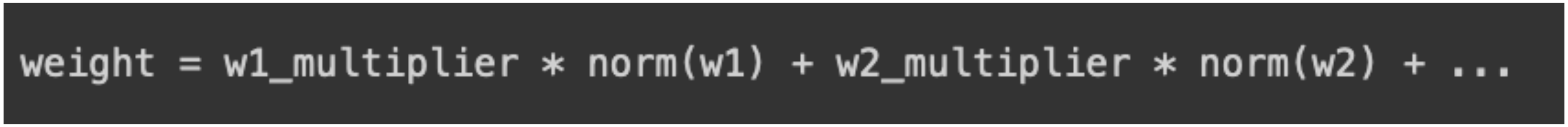
MetricWeigher

- This weigher can compute the weight based on the compute node host’s various metrics.
- The to-be weighed metrics and their weighing ration are specified in the configuration file as the followings:



```
[metrics]
weight_setting = name1=1.0, name2=-1.0
```

General Cost Function



```
weight = w1_multiplier * norm(w1) + w2_multiplier * norm(w2) + ...
```

Metric	Range
CPU utilization	(0, 100) usage percentage
Outbound network traffic	(0, 10^9) byte per second

Least Loaded Server with No Normalization

Weight (Load) = 1 * (CPU utilization) + 1* (Outbound network traffic)

	CPU utilization	Outbound network traffic
Host1	95	100000
Host2	10	100090

Least Loaded Server Without Normalization

Weight (Load) = 1 * (CPU utilization) + 1* (Outbound network traffic)

	Weight
Host1	(1 * 95) + (1 * 100000) = 100095 ✓
Host2	(1*10) + (1* 100090) = 100100

Host1 is selected!

Not good ☹

Min-Max Normalization

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Getting Back to the Previous Example

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

	CPU utilization	Outbound network traffic
Host1	95	100000
Host2	10	100090

	CPU utilization	Outbound network traffic
Host1	(95-0)/(100-0)=0.95	(100000-0)/(10^9-0)=0.0001
Host2	(10-0)/(100-0)=0.1	(100090-0)/(10^9-0)=0.00010009

Least Loaded Server with Normalization

Weight (Load) = 1 * norm(CPU utilization) +

1* norm(Outbound network traffic)

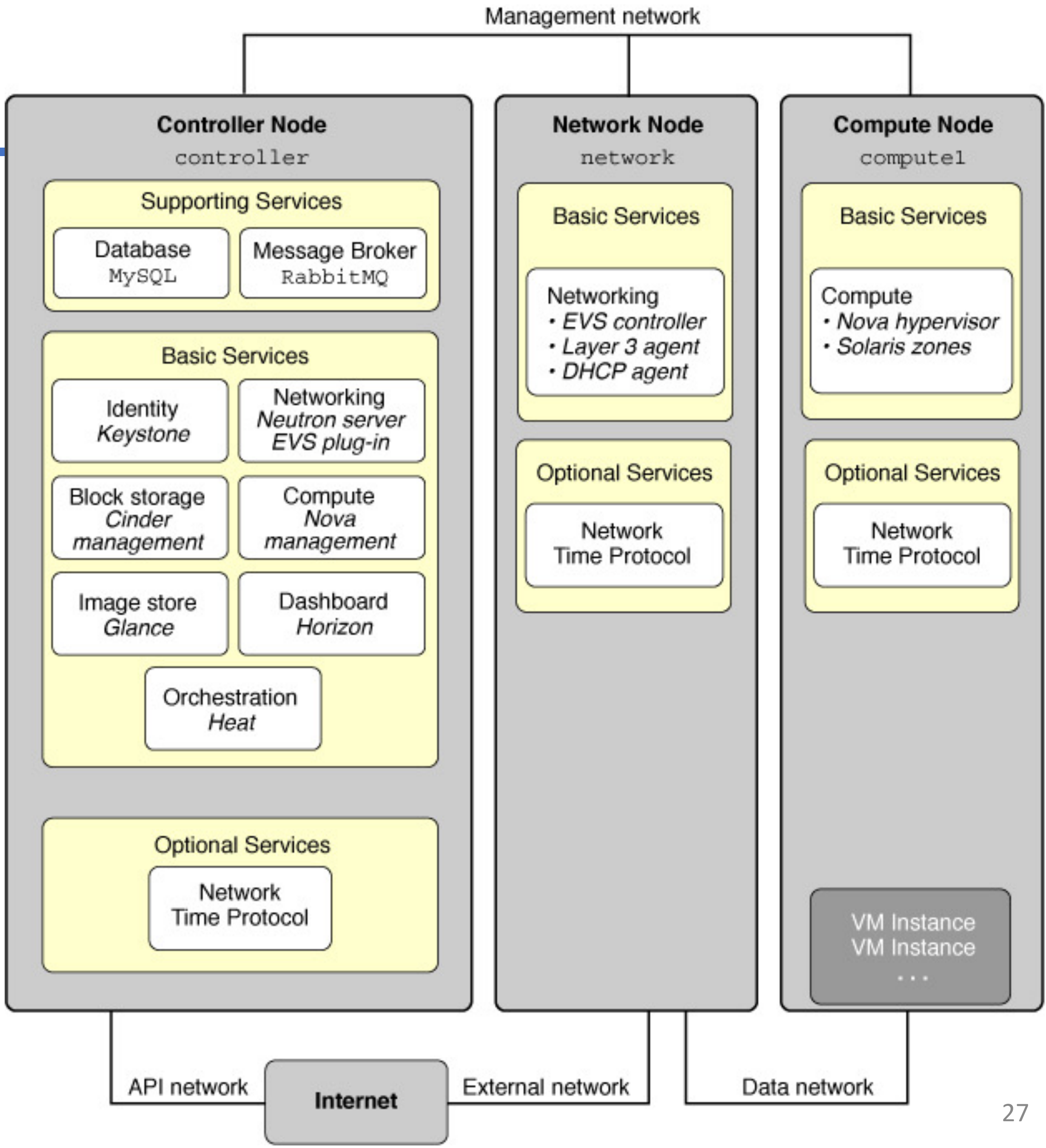
	Weight
Host1	(1 * 0.95) + (1 * 0.0001) = 0.9501
Host2	(1*0.1) + (1* 0.00010009) = 0.10010009✓

Host2 is selected!

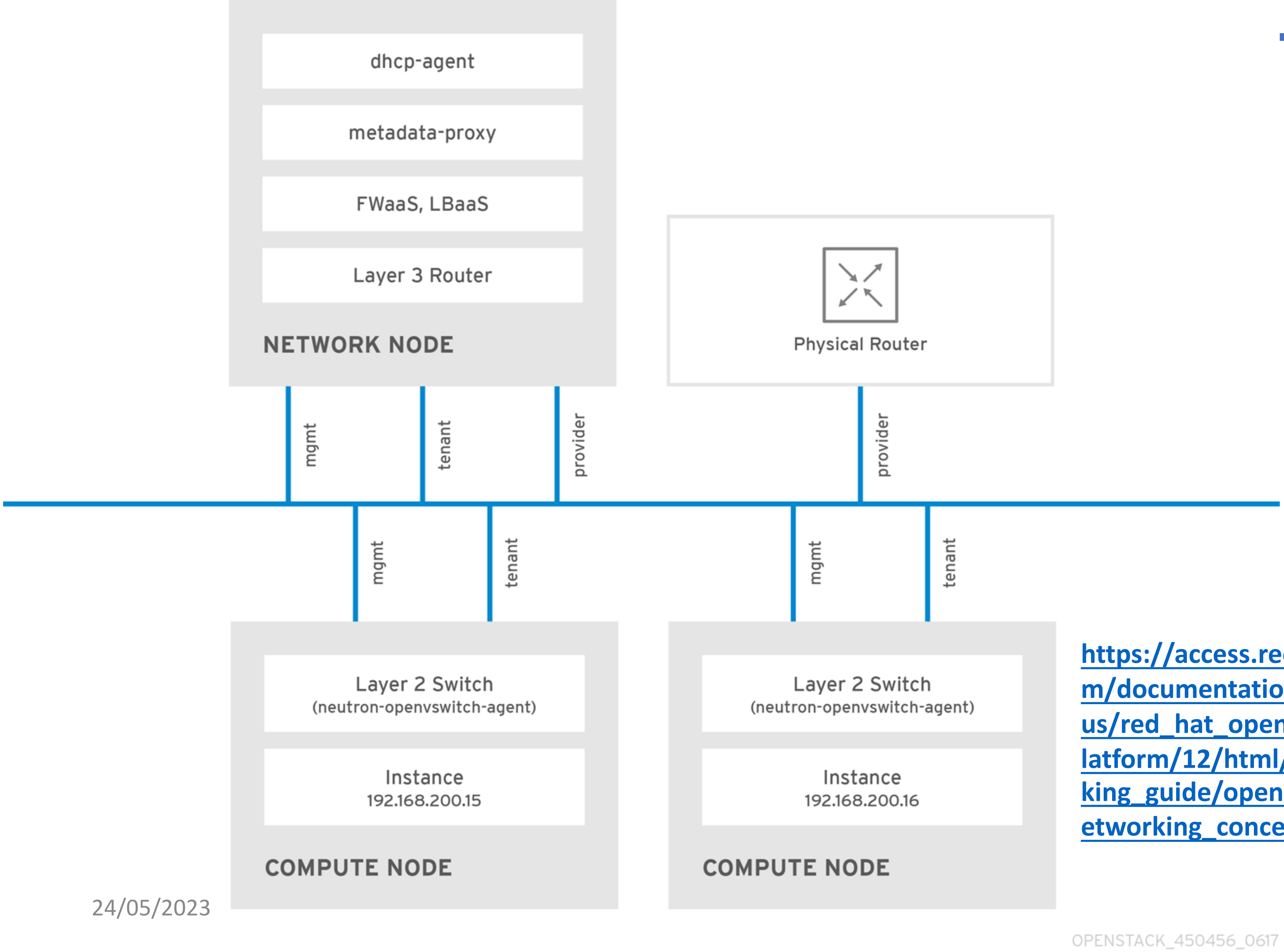
Good job :)

Recap

https://docs.oracle.com/cd/E36784_01/html/E54155/archover.html



Recap (cont.)



https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/12/html/networking_guide/openstack_networking_concepts