

## Measurements

Three type of meters are defined in ceilometer:

Type	Definition
Cumulative	Increasing over time (instance hours)
Gauge	Discrete items (floating IPs, image uploads) and fluctuating values (disk I/O)
Delta	Changing over time (bandwidth)

## Units

1. Whenever a volume is to be measured, SI approved units and their approved symbols or abbreviations should be used. Information units should be expressed in bits ('b') or bytes ('B').
2. For a given meter, the units should NEVER, EVER be changed.
3. When the measurement does not represent a volume, the unit description should always described WHAT is measured (ie: apples, disk, routers, floating IPs, etc.).
4. When creating a new meter, if another meter exists measuring something similar, the same units and precision should be used.
5. Meters and samples should always document their units in Ceilometer (API and Documentation) and new sampling code should not be merged without the appropriate documentation.

Dimension	Unit	Abbreviations	Note
None	N/A		Dimension-less variable
Volume	byte	B	
Time	seconds	s	

Here are the meter types by components that are currently implemented:

## Compute (Nova)

All meters are related to the guest machine, not the host.

Name	Type*	Unit	Resource	Origin**	Support***	Note
instance	g	instance	inst ID	both	1, 2, 3	Existence of instance
instance:<type>	g	instance	inst ID	both	1, 2, 3	Existence of instance <type> (openstack types)
memory	g	MB	inst ID	n	1, 2	Volume of RAM allocated in MB
memory.usage	g	MB	inst ID	p	3	Volume of RAM used in MB
cpu	c	ns	inst ID	p	1, 2	CPU time used
cpu_util	g	%	inst ID	p	1, 2, 3	Average CPU utilisation
vcpus	g	vcpu	inst ID	n	1, 2	Number of VCPUs
disk.read.requests	c	request	inst ID	p	1, 2	Number of read requests
disk.read.requests.rate	g	request/s	inst ID	p	1, 2, 3	Average rate of read requests per second
disk.write.requests	c	request	inst ID	p	1, 2	Number of write requests
disk.write.requests.rate	g	request/s	inst ID	p	1, 2, 3	Average rate of write requests per second
disk.read.bytes	c	B	inst ID	p	1, 2	Volume of reads in B
disk.read.bytes.rate	g	B/s	inst ID	p	1, 2, 3	Average rate of reads in B per second
disk.write.bytes	c	B	inst ID	p	1, 2	Volume of writes in B
disk.write.bytes.rate	g	B/s	inst ID	p	1, 2, 3	Average volume of writes in B per second
disk.root.size	g	GB	inst ID	n	1, 2	Size of root disk in GB
disk.ephemeral.size	g	GB	inst ID	n	1, 2	Size of ephemeral disk in GB
network.incoming.bytes	c	B	iface ID	p	1, 2	Number of incoming

						bytes on a VM network interface
network.incoming.bytes.rate	g	B/s	iface ID	p	1, 2, 3	Average rate per sec of incoming bytes on a VM network interface
network.outgoing.bytes	c	B	iface ID	p	1, 2	Number of outgoing bytes on a VM network interface
network.outgoing.bytes.rate	g	B/s	iface ID	p	1, 2, 3	Average rate per sec of outgoing bytes on a VM network interface
network.incoming.packets	c	packet	iface ID	p	1, 2	Number of incoming packets on a VM network interface
network.incoming.packets.rate	g	packet/s	iface ID	p	1, 2, 3	Average rate per sec of incoming packets on a VM network interface
network.outgoing.packets	c	packet	iface ID	p	1, 2	Number of outgoing packets on a VM network interface
network.outgoing.packets.rate	g	packet/s	iface ID	p	1, 2, 3	Average rate per sec of outgoing packets on a VM network interface

Legend:  
 \*  
 [g]: gauge  
 [c]: cumulative  
 \*\*  
 [p]: pollster  
 [n]: notification  
 \*\*\*  
 [1]: Libvirt support  
 [2]: HyperV support  
 [3]: Vsphere support

Contributors are welcome to extend other virtualization backends' meters or complete the existing ones.

The meters below are related to the host machine.

**Note:** By default, Nova will not collect the following meters related to the host compute node machine. Nova option 'compute\_monitors = ComputeDriverCPUMonitor' should be set in nova.conf to enable meters.

Name	Type	Unit	Resource	Origin	Note
compute.node.cpu.frequency	Gauge	MHz	host ID	notification	CPU frequency
compute.node.cpu.kernel.time	Cumulative	ns	host ID	notification	CPU kernel time
compute.node.cpu.idle.time	Cumulative	ns	host ID	notification	CPU idle time
compute.node.cpu.user.time	Cumulative	ns	host ID	notification	CPU user mode time
compute.node.cpu.iowait.time	Cumulative	ns	host ID	notification	CPU I/O wait time
compute.node.cpu.kernel.percent	Gauge	%	host ID	notification	CPU kernel percentage
compute.node.cpu.idle.percent	Gauge	%	host ID	notification	CPU idle percentage
compute.node.cpu.user.percent	Gauge	%	host ID	notification	CPU user mode percentage
compute.node.cpu.iowait.percent	Gauge	%	host ID	notification	CPU I/O wait percentage
compute.node.cpu.percent	Gauge	%	host ID	notification	CPU utilization

## Network (Neutron)

Name	Type	Unit	Resource	Origin	Note
network	Gauge	network	netw ID	notification	Existence of network
network.create	Delta	network	netw ID	notification	Creation requests for this network
network.update	Delta	network	netw ID	notification	Update requests for this network
subnet	Gauge	subnet	subnt ID	notification	Existence of subnet
subnet.create	Delta	subnet	subnt ID	notification	Creation requests for this subnet
subnet.update	Delta	subnet	subnt ID	notification	Update requests for this subnet
port	Gauge	port	port ID	notification	Existence of port
port.create	Delta	port	port ID	notification	Creation requests for this port
port.update	Delta	port	port ID	notification	Update requests for this port
router	Gauge	router	rtr ID	notification	Existence of router
router.create	Delta	router	rtr ID	notification	Creation requests for this router

router.update	Delta	router	rtr ID	notification	Update requests for this router
ip.floating	Gauge	ip	ip ID	both	Existence of floating ip
ip.floating.create	Delta	ip	ip ID	notification	Creation requests for this floating ip
ip.floating.update	Delta	ip	ip ID	notification	Update requests for this floating ip

## Image (Glance)

Name	Type	Unit	Resource	Origin	Note
image	Gauge	image	image ID	both	Image polling -> it (still) exists
image.size	Gauge	B	image ID	both	Uploaded image size
image.update	Delta	image	image ID	notification	Number of update on the image
image.upload	Delta	image	image ID	notification	Number of upload of the image
image.delete	Delta	image	image ID	notification	Number of delete on the image
image.download	Delta	B	image ID	notification	Image is downloaded
image.serve	Delta	B	image ID	notification	Image is served out

## Volume (Cinder)

Name	Type	Unit	Resource	Origin	Note
volume	Gauge	volume	vol ID	notification	Existence of volume
volume.size	Gauge	GB	vol ID	notification	Size of volume
snapshot	Gauge	snapshot	snap ID	notification	Existence of snapshot
snapshot.size	Gauge	GB	snap ID	notification	Size of snapshot's volume

Make sure Cinder is properly configured first: see [Installing Manually](#).

## Object Storage (Swift)

Name	Type	Unit	Resource	Origin	Note
storage.objects	Gauge	object	store ID	pollster	Number of objects
storage.objects.size	Gauge	B	store ID	pollster	Total size of stored objects
storage.objects.containers	Gauge	container	store ID	pollster	Number of containers
storage.objects.incoming.bytes	Delta	B	store ID	notification	Number of incoming bytes
storage.objects.outgoing.bytes	Delta	B	store ID	notification	Number of outgoing bytes
storage.api.request	Delta	request	store ID	notification	Number of API requests against swift
storage.containers.objects	Gauge	object	str ID/cont	pollster	Number of objects in container
storage.containers.objects.size	Gauge	B	str ID/cont	pollster	Total size of stored objects in container

In order to use storage.objects.incoming.bytes and storage.outgoing.bytes, one must configure Swift as described in [Installing Manually](#). Note that they may not be updated right after an upload/download, since Swift takes some time to update the container properties.

## Orchestration (Heat)

Name	Type	Unit	Resource	Origin	Note
stack.create	Delta	stack	stack ID	notification	Creation requests for a stack successful
stack.update	Delta	stack	stack ID	notification	Updating requests for a stack successful
stack.delete	Delta	stack	stack ID	notification	Deletion requests for a stack successful
stack.resume	Delta	stack	stack ID	notification	Resuming requests for a stack successful
stack.suspend	Delta	stack	stack ID	notification	Suspending requests for a stack successful

To enable Heat notifications configure Heat as described in [Installing Manually](#).

## Energy (Kwapi)

Name	Type	Unit	Resource	Origin	Note
energy	Cumulative	kWh	probe ID	pollster	Amount of energy
power	Gauge	W	probe ID	pollster	Power consumption

## Network (From SDN Controller)

These meters based on OpenFlow Switch metrics. In order to enable these meters, each driver needs to be configured.

Meter	Type	Unit	Resource	Origin	Note
switch	Gauge	switch	switch ID	pollster	Existence of switch
switch.port	Gauge	port	switch ID	pollster	Existence of port
switch.port.receive.packets	Cumulative	packet	switch ID	pollster	Received Packets
switch.port.transmit.packets	Cumulative	packet	switch ID	pollster	Transmitted Packets
switch.port.receive.bytes	Cumulative	B	switch ID	pollster	Received Bytes
switch.port.transmit.bytes	Cumulative	B	switch ID	pollster	Transmitted Bytes
switch.port.receive.drops	Cumulative	packet	switch ID	pollster	Receive Drops
switch.port.transmit.drops	Cumulative	packet	switch ID	pollster	Transmit Drops
switch.port.receive.errors	Cumulative	packet	switch ID	pollster	Receive Errors
switch.port.transmit.errors	Cumulative	packet	switch ID	pollster	Transmit Errors
switch.port.receive.frame_error	Cumulative	packet	switch ID	pollster	Receive Frame Alignment Errors
switch.port.receive.overrun_error	Cumulative	packet	switch ID	pollster	Receive Overrun Errors
switch.port.receive.crc_error	Cumulative	packet	switch ID	pollster	Receive CRC Errors
switch.port.collision.count	Cumulative	count	switch ID	pollster	Collisions
switch.table	Gauge	table	switch ID	pollster	Duration of Table
switch.table.active.entries	Gauge	entry	switch ID	pollster	Active Entries
switch.table.lookup.packets	Gauge	packet	switch ID	pollster	Packet Lookups
switch.table.matched.packets	Gauge	packet	switch ID	pollster	Packet Matches
switch.flow	Gauge	flow	switch ID	pollster	Duration of Flow
switch.flow.duration.seconds	Gauge	s	switch ID	pollster	Duration(seconds)
switch.flow.duration.nanoseconds	Gauge	ns	switch ID	pollster	Duration(nanoseconds)
switch.flow.packets	Cumulative	packet	switch ID	pollster	Received Packets
switch.flow.bytes	Cumulative	B	switch ID	pollster	Received Bytes

## LoadBalancer as a Service (LBaaS)

Meter	Type	Unit	Resource	Origin	Note
network.services.lb.pool	Gauge	pool	pool ID	pollster	Existence of a LB Pool
network.services.lb.vip	Gauge	vip	vip ID	pollster	Existence of a LB Vip
network.services.lb.member	Gauge	member	member ID	pollster	Existence of a LB Member
network.services.lb.health_monitor	Gauge	monitor	monitor ID	pollster	Existence of a LB Health Probe
network.services.lb.total.connections	Gauge	connection	pool ID	pollster	Total connections on a LB
network.services.lb.active.connections	Gauge	connection	pool ID	pollster	Active connections on a LB
network.services.lb.incoming.bytes	Cumulative	B	pool ID	pollster	Number of incoming Bytes
network.services.lb.outgoing.bytes	Cumulative	B	pool ID	pollster	Number of outgoing Bytes

## VPN as a Service (VPNaaS)

Meter	Type	Unit	Resource	Origin	Note
network.services.vpn	Gauge	vpn	vpn ID	pollster	Existence of a VPN service
network.services.vpn.connections	Gauge	connection	connection ID	pollster	Existence of a IPSec Connection

## Firewall as a Service (FWaaS)

Meter	Type	Unit	Resource	Origin	Note
network.services.firewall	Gauge	firewall	firewall ID	pollster	Existence of a Firewall service
network.services.firewall.policy	Gauge	policy	policy ID	pollster	Existence of Firewall Policy

## Ironic Hardware IPMI Sensor Data

IPMI sensor data is not available by default in Ironic. To enable these meters see the [Ironic Installation Guide](#).

Meter	Type	Unit	Resource	Origin	Note
hardware.ipmi.fan	Gauge	RPM	fan sensor	notification	Fan RPM
hardware.ipmi.temperature	Gauge	C	temp sensor	notification	Sensor Temperature Reading

hardware.ipmi.current	Gauge	W	current sensor	notification	Sensor Current Reading
hardware.ipmi.voltage	Gauge	V	voltage sensor	notification	Sensor Voltage Reading

## Dynamically retrieving the Meters via ceilometer client

To retrieve the available meters that can be queried given the actual resource instances available, use the `meter-list` command:

```
$ ceilometer meter-list
```

Name	Type	Resource ID	User ID	Project ID
image	gauge	09e84d97-8712-4dd2-bcce-45970b2430f7		57cf6d9368

## Naming convention

If you plan on adding meters, please follow the convention below:

1. Always use '.' as separator and go from least to most discriminant word. For example, do not use `ephemeral_disk_size` but `disk.ephemeral.size`
2. When a part of the name is a variable, it should always be at the end and start with a ':'. For example do not use `<type>.image` but `image:<type>`, where type is your variable name.
3. If you have any hesitation, come and ask in [#openstack-ceilometer](#)

## User-defined sample metadata for Nova

Users are allowed to add additional metadata to samples of nova meter. These additional metadata are stored in 'resource\_metadata.user\_metadata.\*' of the sample. To do so, users should add nova user metadata prefixed with 'metering.':

```
$ nova boot --meta metering.custom_metadata=a_value my_vm
```

Note: The name of the metadata shouldn't exceed 256 characters otherwise it will be cut off. Also, if it has '.', this will be replaced by a '\_' in ceilometer.

## User-defined sample metadata for Swift

It's possible to add additional metadata to sample of Swift meter as well. You might specify headers whose values will be stored in resource\_metadata as 'resource\_metadata.http\_header\_\$name', where \$name is a name of the header with '-' replaced by '\_'.

This is done using 'metadata\_headers' option in middleware configuration, refer to [Installing Manually](#) for details.

For example, this could be used to distinguish external and internal users. You'd have to implement a custom Swift middleware that sets a proper header and just add it to metadata\_headers.

## OSprofiler data

All messages with event type "profiler.\*" will be collected as profiling data. Using notification plugin `profiler/notifications.py`.

**Note:** Be sparing with heavy usage of OSprofiler, especially in case of complex operations like booting and deleting instance that may create over 100kb of sample data per each request.