#

# From oslo.messaging

#

# Size of RPC connection pool. (integer value)

#rpc\_conn\_pool\_size = 30

{%- if \_data.rpc\_conn\_pool\_size is defined %}

rpc\_conn\_pool\_size = {{ \_data.rpc\_conn\_pool\_size }}

{%- endif %}

# The pool size limit for connections expiration policy (integer

# value)

#conn\_pool\_min\_size = 2

{%- if \_data.conn\_pool\_min\_size is defined %}

conn\_pool\_min\_size = {{ \_data.conn\_pool\_min\_size }}

{%- endif %}

# The time-to-live in sec of idle connections in the pool (integer

# value)

#conn\_pool\_ttl = 1200

{%- if \_data.conn\_pool\_ttl is defined %}

conn\_pool\_ttl = {{ \_data.conn\_pool\_ttl }}

{%- endif %}

# ZeroMQ bind address. Should be a wildcard (\*), an ethernet

# interface, or IP. The "host" option should point or resolve to this

# address. (string value)

#rpc\_zmq\_bind\_address = \*

# MatchMaker driver. (string value)

# Possible values:

# redis - <No description provided>

# sentinel - <No description provided>

# dummy - <No description provided>

#rpc\_zmq\_matchmaker = redis

# Number of ZeroMQ contexts, defaults to 1. (integer value)

#rpc\_zmq\_contexts = 1

# Maximum number of ingress messages to locally buffer per topic.

# Default is unlimited. (integer value)

#rpc\_zmq\_topic\_backlog = <None>

# Directory for holding IPC sockets. (string value)

#rpc\_zmq\_ipc\_dir = /var/run/openstack

# Name of this node. Must be a valid hostname, FQDN, or IP address.

# Must match "host" option, if running Nova. (string value)

#rpc\_zmq\_host = localhost

# Number of seconds to wait before all pending messages will be sent

# after closing a socket. The default value of -1 specifies an

# infinite linger period. The value of 0 specifies no linger period.

# Pending messages shall be discarded immediately when the socket is

# closed. Positive values specify an upper bound for the linger

# period. (integer value)

# Deprecated group/name - [DEFAULT]/rpc\_cast\_timeout

#zmq\_linger = -1

# The default number of seconds that poll should wait. Poll raises

# timeout exception when timeout expired. (integer value)

#rpc\_poll\_timeout = 1

{%- if \_data.rpc\_poll\_timeout is defined %}

rpc\_poll\_timeout = {{ \_data.rpc\_poll\_timeout }}

{%- endif %}

# Expiration timeout in seconds of a name service record about

# existing target ( < 0 means no timeout). (integer value)

#zmq\_target\_expire = 300

# Update period in seconds of a name service record about existing

# target. (integer value)

#zmq\_target\_update = 180

# Use PUB/SUB pattern for fanout methods. PUB/SUB always uses proxy.

# (boolean value)

#use\_pub\_sub = false

# Use ROUTER remote proxy. (boolean value)

#use\_router\_proxy = false

# This option makes direct connections dynamic or static. It makes

# sense only with use\_router\_proxy=False which means to use direct

# connections for direct message types (ignored otherwise). (boolean

# value)

#use\_dynamic\_connections = false

# How many additional connections to a host will be made for failover

# reasons. This option is actual only in dynamic connections mode.

# (integer value)

#zmq\_failover\_connections = 2

# Minimal port number for random ports range. (port value)

# Minimum value: 0

# Maximum value: 65535

#rpc\_zmq\_min\_port = 49153

# Maximal port number for random ports range. (integer value)

# Minimum value: 1

# Maximum value: 65536

#rpc\_zmq\_max\_port = 65536

# Number of retries to find free port number before fail with

# ZMQBindError. (integer value)

#rpc\_zmq\_bind\_port\_retries = 100

# Default serialization mechanism for serializing/deserializing

# outgoing/incoming messages (string value)

# Possible values:

# json - <No description provided>

# msgpack - <No description provided>

#rpc\_zmq\_serialization = json

# This option configures round-robin mode in zmq socket. True means

# not keeping a queue when server side disconnects. False means to

# keep queue and messages even if server is disconnected, when the

# server appears we send all accumulated messages to it. (boolean

# value)

#zmq\_immediate = true

# Enable/disable TCP keepalive (KA) mechanism. The default value of -1

# (or any other negative value) means to skip any overrides and leave

# it to OS default; 0 and 1 (or any other positive value) mean to

# disable and enable the option respectively. (integer value)

#zmq\_tcp\_keepalive = -1

# The duration between two keepalive transmissions in idle condition.

# The unit is platform dependent, for example, seconds in Linux,

# milliseconds in Windows etc. The default value of -1 (or any other

# negative value and 0) means to skip any overrides and leave it to OS

# default. (integer value)

#zmq\_tcp\_keepalive\_idle = -1

# The number of retransmissions to be carried out before declaring

# that remote end is not available. The default value of -1 (or any

# other negative value and 0) means to skip any overrides and leave it

# to OS default. (integer value)

#zmq\_tcp\_keepalive\_cnt = -1

# The duration between two successive keepalive retransmissions, if

# acknowledgement to the previous keepalive transmission is not

# received. The unit is platform dependent, for example, seconds in

# Linux, milliseconds in Windows etc. The default value of -1 (or any

# other negative value and 0) means to skip any overrides and leave it

# to OS default. (integer value)

#zmq\_tcp\_keepalive\_intvl = -1

# Maximum number of (green) threads to work concurrently. (integer

# value)

#rpc\_thread\_pool\_size = 100

{%- if \_data.rpc\_thread\_pool\_size is defined %}

rpc\_thread\_pool\_size = {{ \_data.rpc\_thread\_pool\_size }}

{%- endif %}

# Expiration timeout in seconds of a sent/received message after which

# it is not tracked anymore by a client/server. (integer value)

#rpc\_message\_ttl = 300

{%- if \_data.rpc\_message\_ttl is defined %}

rpc\_message\_ttl = {{ \_data.rpc\_message\_ttl }}

{%- endif %}

# Wait for message acknowledgements from receivers. This mechanism

# works only via proxy without PUB/SUB. (boolean value)

#rpc\_use\_acks = false

{%- if \_data.rpc\_use\_acks is defined %}

rpc\_use\_acks = {{ \_data.rpc\_use\_acks }}

{%- endif %}

# Number of seconds to wait for an ack from a cast/call. After each

# retry attempt this timeout is multiplied by some specified

# multiplier. (integer value)

#rpc\_ack\_timeout\_base = 15

{%- if \_data.rpc\_ack\_timeout\_base is defined %}

rpc\_ack\_timeout\_base = {{ \_data.rpc\_ack\_timeout\_base }}

{%- endif %}

# Number to multiply base ack timeout by after each retry attempt.

# (integer value)

#rpc\_ack\_timeout\_multiplier = 2

{%- if \_data.rpc\_ack\_timeout\_multiplier is defined %}

rpc\_ack\_timeout\_multiplier = {{ \_data.rpc\_ack\_timeout\_multiplier }}

{%- endif %}

# Default number of message sending attempts in case of any problems

# occurred: positive value N means at most N retries, 0 means no

# retries, None or -1 (or any other negative values) mean to retry

# forever. This option is used only if acknowledgments are enabled.

# (integer value)

#rpc\_retry\_attempts = 3

{%- if \_data.rpc\_retry\_attempts is defined %}

rpc\_retry\_attempts = {{ \_data.rpc\_retry\_attempts }}

{%- endif %}

# List of publisher hosts SubConsumer can subscribe on. This option

# has higher priority then the default publishers list taken from the

# matchmaker. (list value)

#subscribe\_on =

# Size of executor thread pool when executor is threading or eventlet.

# (integer value)

# Deprecated group/name - [DEFAULT]/rpc\_thread\_pool\_size

#executor\_thread\_pool\_size = 64

{%- if \_data.executor\_thread\_pool\_size is defined %}

executor\_thread\_pool\_size = {{ \_data.executor\_thread\_pool\_size }}

{%- endif %}

# Seconds to wait for a response from a call. (integer value)

#rpc\_response\_timeout = 60

{%- if \_data.rpc\_response\_timeout is defined %}

rpc\_response\_timeout = {{ \_data.rpc\_response\_timeout }}

{%- endif %}

# The network address and optional user credentials for connecting to

# the messaging backend, in URL format. The expected format is:

#

# driver://[user:pass@]host:port[,[userN:passN@]hostN:portN]/virtual\_host?query

#

# Example: rabbit://rabbitmq:password@127.0.0.1:5672//

#

# For full details on the fields in the URL see the documentation of

# oslo\_messaging.TransportURL at

# https://docs.openstack.org/oslo.messaging/latest/reference/transport.html

# (string value)

#transport\_url = <None>

{%- set rabbit\_port = \_data.get('port', 5671 if \_data.get('ssl',{}).get('enabled', False) else 5672) %}

{%- if \_data.members is defined %}

transport\_url = rabbit://{% for member in \_data.members -%}

{{ \_data.user }}:{{ \_data.password }}@{{ member.host }}:{{ member.get('port', rabbit\_port) }}

{%- if not loop.last -%},{%- endif -%}

{%- endfor -%}

/{{ \_data.virtual\_host }}

{%- else %}

transport\_url = rabbit://{{ \_data.user }}:{{ \_data.password }}@{{ \_data.host }}:{{ rabbit\_port }}/{{ \_data.virtual\_host }}

{%- endif %}

# DEPRECATED: The messaging driver to use, defaults to rabbit. Other

# drivers include amqp and zmq. (string value)

# This option is deprecated for removal.

# Its value may be silently ignored in the future.

# Reason: Replaced by [DEFAULT]/transport\_url

#rpc\_backend = rabbit

# The default exchange under which topics are scoped. May be

# overridden by an exchange name specified in the transport\_url

# option. (string value)

#control\_exchange = openstack

{%- if \_data.control\_exchange is defined %}

control\_exchange = {{ \_data.control\_exchange }}

{%- endif %}